aws

User Guide

Amazon SageMaker Unified Studio



Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon SageMaker Unified Studio: User Guide

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

..... xxvi
What is Amazon SageMaker Unified Studio? 1
How does it work? 1
Get Started 2
Access Amazon SageMaker Unified Studio 3
Configure credentials 4
Sign up for an AWS account 4
Configure SSO credentials with IAM Identity Center 4
Set up your domain as an administrator 6
What you will do 6
Step 1: Open SageMaker Unified Studio in the console 6
Step 2: Configure the login role 7
Step 3: Configure the execution role 8
Step 4: Configure storage and encryption 9
Step 5: Explore your new domain 10
Step 6: Create a project and add members 11
What you learned 11
Create a project and add members 13
Step 1: Enter details 13
Step 2: Add members 15
Navigate Amazon SageMaker Unified Studio 17
Navigation Panel 17
Overview 17
Data analytics 17
AI/ML 17
Integrated development environments (IDEs) 18
Domain Management 18
Jump into your data and models 18
Build with sample data 18
Change the display mode 18
Change the display language 19
Getting started tutorials 21
Run your first SQL query 21
What you will do 21

Step 1: Open the query editor ... 22
Step 2: Browse your data ... 23
Step 3: Run a SQL query ... 24
Step 4: Explore your results ... 25
Ask in natural language ... 26
What you learned ... 28
Analyze and visualize data ... 28
What you will do ... 29
Step 1: Open a notebook ... 29
Step 2: Load the data ... 31
Step 3: Clean the data and calculate statistics ... 32
Step 4: Analyze patterns by state ... 33
Step 5: Create a visualization ... 34
Use the Data Agent to generate code ... 36
What you learned ... 39
Build a data pipeline with Visual ETL ... 39
What you will do ... 40
Step 1: Create a visual ETL job ... 40
Step 2: Add a data source ... 42
Step 3: Add transformations ... 43
Step 4: Add a target ... 45
Step 5: Save and run the job ... 46
Step 6: Verify the output ... 48
What you learned ... 50
Automate a data pipeline with Workflows ... 51
What you will do ... 51
Step 1: Create a workflow ... 52
Step 2: Add an Athena query task ... 53
Step 3: Add a visual ETL job task ... 55
Step 4: Save and run the workflow ... 57
Step 5: Verify the output ... 58
What you learned ... 59
Train an ML model ... 60
What you will do ... 60
Step 1: Open the sample notebook ... 60
Step 2: Set up and load the data ... 62

Step 3: Explore the churn problem ... 64
Step 4: Prepare features and train models ... 66
Step 5: Understand what drives churn ... 69
Step 6: Save the model ... 70
What's next ... 71
What you learned ... 71
Core capabilities ... 73
Notebooks ... 74
Overview ... 74
Key capabilities ... 75
Roles and permissions ... 75
Create and manage notebooks ... 76
Overview ... 76
Procedure ... 76
Examples ... 77
Work with cells ... 79
Overview ... 79
Procedure ... 79
Visualize and explore data ... 85
Overview ... 85
Procedure ... 85
Manage compute environments ... 86
Overview ... 86
Procedure ... 86
Spark Compute ... 88
Overview ... 88
Select a Spark runtime ... 88
Spark UI monitoring ... 89
Use AWS Glue with Spark Connect ... 90
Use EMR Serverless with Spark Connect ... 90
Connect to data sources ... 93
Overview ... 93
Prerequisites ... 93
Procedure ... 93
Import and export notebooks ... 95
Overview ... 95

Import a notebook ... 95
Export a notebook ... 96
Schedule and automate notebook runs ... 97
Overview ... 97
Running a notebook in the background ... 97
Creating a recurring schedule ... 100
Viewing, editing, and deleting schedules ... 101
Parameterizing notebooks ... 103
Orchestrating notebooks with workflows ... 104
Debugging failed runs ... 106
Use the sample notebook ... 107
Workflows ... 109
Serverless ... 109
Visual workflows ... 110
Provisioned ... 113
Workflow environments ... 114
Visual Workflows ... 118
Code workflows ... 125
Query Editor ... 132
Supported query engines ... 132
Write, run, and view query results ... 132
Create a querybook ... 132
Write a query ... 133
Run a sample query from the data explorer ... 133
Run a query ... 134
View results ... 135
Create a visualization ... 135
Export query results ... 136
Generate SQL with the Data Agent ... 137
Open the Data Agent panel ... 137
Ask a question and generate SQL ... 138
Add generated SQL to your querybook ... 139
Multi-turn conversations ... 139
Fix errors with the Data Agent ... 139
Save, schedule, and review queries ... 140
Save a querybook to the project ... 140

Share with project members ... 141
Create a schedule ... 142
Review and manage scheduled queries ... 145
Review query history ... 145
Visual ETL ... 150
Supported connectors for Visual ETL ... 151
Supported transforms for Visual ETL ... 151
Add current timestamp transform ... 152
Aggregate transform ... 153
Array to columns transform ... 154
Autobalance processing transform ... 155
Change columns transform ... 156
Concatenate columns transform ... 157
Custom code transform ... 158
Custom visual transform ... 159
Derived column transform ... 163
Drop columns transform ... 164
Drop duplicates transform ... 165
Drop nulls transform ... 168
Explode array or map into rows transform ... 168
Evaluate data quality transform ... 170
Extract JSON path transform ... 174
Fill nulls transform ... 175
Filter transform ... 176
Flatten transform ... 177
Format timestamp transform ... 178
Identifier transform ... 178
Intersect transform ... 179
Join transform ... 179
Limit tranform ... 180
Lookup transform ... 180
Parse JSON column transform ... 181
Pivot rows into columns transform ... 182
Regex extract transform ... 182
Rename columns transform ... 183
Select columns transform ... 183

Split string transform ... 184
SQL query transform ... 185
To timestamp transform ... 186
Union transform ... 187
Unpivot columns into rows transform ... 188
UUID transform ... 188
Visual ETL for IAM-based domains ... 189
Key features for IAM-based domains ... 189
Create a Visual ETL job in IAM-based domains ... 189
Authoring a Visual ETL job using generative AI in IAM-based domains ... 190
Scheduling and running visual jobs in IAM-based domains ... 190
Visual ETL for Identity Center-based domains ... 191
Key features for Identity Center-based domains ... 191
Creating a Visual ETL job in Identity Center-based domains ... 191
Authoring a Visual ETL job using generative AI in Identity Center-based domains ... 192
Scheduling and running visual jobs in Identity Center-based domains ... 193
Using both external data and fine-grained data ... 195
Best practices for Visual ETL in Identity Center-based domains ... 197
Data preview ... 198
Authoring with Glue data preview ... 198
Authoring with SageMaker data preview ... 198
SageMaker Data Agent ... 200
Prerequisites ... 200
Basic usage ... 200
Getting started with the SageMaker Data Agent for Notebook ... 201
Conversation history ... 201
Getting started with the SageMaker Data Agent for Query Editor ... 202
Using the SageMaker Data Agent for data analytics tasks ... 207
Data exploration and discovery ... 207
Building multi-step analytical workflows ... 208
Cell level inline code modification ... 208
Referencing previous cell results ... 209
Error handling and fix with the SageMaker Data Agent ... 210
Business context integration ... 210
Prerequisites ... 211
How it works ... 211

Types of questions you can ask ... 211
Understanding search results ... 212
Handling inaccessible assets ... 213
In-cell code generation ... 213
Supported tools ... 213
Best practices for the SageMaker Data Agent ... 213
Security and access control for the SageMaker Data Agent ... 214
Required IAM permissions to use SageMaker Data Agent ... 214
Disable the SageMaker Data Agent in IdC domains ... 214
Data storage in the SageMaker Data Agent ... 215
Content that is stored by SageMaker Data Agent ... 215
Content that is not stored by SageMaker Data Agent ... 215
AWS Regions where content is processed and stored ... 215
Service improvement ... 215
Content that AWS may use for service improvement ... 215
Content that AWS does not use for service improvement ... 216
How to opt out ... 216
Cross-region processing in the SageMaker Data Agent ... 216
Cross-region inference ... 216
Supported regions for cross-region inference ... 217
Regions using global cross-region inference ... 218
Code spaces ... 219
Code spaces in IAM domains ... 219
Creating a code space in IAM domains ... 219
Managing code spaces in IAM domains ... 220
Configuring a code space in IAM domains ... 221
Amazon Q support in code spaces for IAM domains ... 221
Code spaces in Identity Center domains ... 221
Creating a code space in Identity Center domains ... 221
Editing a code space in Identity Center domains ... 223
Deleting a code space in Identity Center domains ... 224
Amazon Q support in code spaces for Identity Center domains ... 224
Bring your own image (BYOI) ... 225
Dockerfile specifications ... 225
How to BYOI ... 227
Launch your custom image in Amazon SageMaker Unified Studio ... 228

Speed up container startup with SOCI ... 229
Detach and clean up custom image resources ... 229
Using Amazon SageMaker Unified Studio Library for Python ... 229
Using ClientConfig ... 230
Domain ... 230
Project ... 231
Utility Methods ... 237
Execution APIs ... 250
JupyterLab ... 254
Managing configurations ... 255
Accessing metadata ... 259
Performing Git operations ... 259
Using the Amazon Q data integration in AWS Glue ... 260
Running SQL and Spark code ... 260
Visualizing results ... 261
Data Sharing Across Compute Environments ... 263
Code Editor ... 265
Performing Git operations ... 266
Checking the version ... 267
Connections and extensions ... 267
Using Strands Agents in Amazon SageMaker Unified Studio ... 267
Local IDE Support ... 270
Overview ... 270
Prerequisites ... 270
VS Code Setting in Amazon SageMaker Unified Studio ... 271
Authenticating directly in Local IDE ... 272
Link to VS Code from Amazon SageMaker Unified Studio ... 273
Kiro in Amazon SageMaker Unified Studio ... 274
Link to Kiro from Amazon SageMaker Unified Studio ... 275
Cursor in Amazon SageMaker Unified Studio ... 276
Link to Cursor from Amazon SageMaker Unified Studio ... 277
Connecting to Amazon SageMaker Unified Studio Spaces ... 278
Domain and Project Switching ... 279
Refresh Project Metadata ... 279
AWS Toolkit Experience in Remote Session ... 280
Data sources and connections ... 281

Working with data 282
Bringing existing resources into Amazon SageMaker Unified Studio 283
AWS Glue Data Catalogs 283
Amazon S3 data 283
Amazon Athena workgroups and saved queries 283
Amazon EMR on EC2 clusters 283
Amazon Redshift clusters and Amazon Redshift Serverless workgroups 284
Amazon SageMaker AI resources 284
AWS IAM roles 284
Data in Identity Center-based domains 285
S3 data 285
Adding Amazon S3 data 285
Sharing Amazon S3 data 297
Lakehouse architecture 299
Data and catalog connections in IAM-based domains 300
Prerequisites 300
Supported data connections 300
Databases and data warehouses 300
Storage 301
AWS resources created by connections 301
Connecting to a new data source 301
Domain and project VPC configuration 301
SSL connection 302
Connecting to Amazon S3 302
Connecting to Amazon Redshift 303
VPC Requirements 303
Steps to connect to Amazon Redshift from Amazon SageMaker Unified Studio 304
Steps to connect to Amazon SageMaker Unified Studio from Amazon Redshift console pages 305
Steps to connect to Amazon SageMaker Unified Studio from Amazon Redshift Query Editor v2 305
Redshift Access using IAM Authentication 306
Troubleshooting Database Access 306
Data integration and processing 308
Data processing jobs 309
Benefits of using jobs 309

Creating jobs ... 309
Job authoring methods ... 310
To create a new job ... 310
Running jobs ... 316
Run Methods ... 316
How to run a job ... 317
Monitoring jobs and job runs ... 318
Monitoring capabilities ... 319
Job runs view ... 319
To access job monitoring ... 321
Viewing the details of a job run ... 321
Viewing CloudWatch metrics for AWS Glue jobs ... 323
Limitations ... 324
Best Practices ... 324
Next Steps ... 324
Data quality monitoring ... 326
View data quality results in data processing jobs ... 326
View data quality results in Visual ETL run details ... 327
How in-transit and at-rest evaluations differ ... 328
Data visualization and sharing ... 330
Amazon Quick in Amazon SageMaker Unified Studio ... 331
Overview ... 270
Limitations ... 331
Getting started ... 331
Share dashboards ... 335
AI and machine learning ... 336
Amazon Bedrock in SageMaker Unified Studio ... 337
Discover Amazon Bedrock in SageMaker Unified Studio playgrounds ... 337
Build generative AI apps ... 338
Find serverless models with the model catalog ... 339
What is a prompt? ... 340
Inference parameters ... 341
System instructions ... 343
Prompt engineering guides ... 343
Experiment with the playgrounds ... 344
Chat with a model in the chat playground ... 345

Chat with an app in the chat playground ... 348
Generate an image with the image and video playground ... 349
Generate a video ... 358
Access shared generative AI assets in a playground ... 361
Build a chat agent app ... 362
Create a chat agent app with Amazon Bedrock ... 363
Deploy a chat agent app ... 372
Share a chat agent app ... 375
Build a flow app ... 377
Create a flow app with Amazon Bedrock ... 379
Define inputs with expressions ... 388
Use logic nodes to control flow ... 392
Use a chat agent app in a flow app ... 394
Flow nodes ... 396
Reuse and share prompts ... 406
Create a prompt ... 406
Add a prompt to a flow app ... 409
Modify a prompt ... 410
Delete a prompt ... 411
Share a prompt version ... 412
Evaluate the performance of a model ... 413
Create a model evaluation job ... 414
Model evaluation task types ... 417
Prompt datasets for model evaluation ... 424
Review a model evaluation job ... 429
Add a Knowledge Base to your Amazon Bedrock app ... 432
Create a Knowledge Base component ... 433
Add a Knowledge Base component to a chat agent app ... 448
Add a Knowledge Base component to a flow app ... 450
Synchronize a Knowledge Base ... 451
Safeguard your app with a guardrail ... 452
Guardrail policies ... 452
Create a guardrail component ... 455
Add a guardrail component to a chat agent app ... 457
Add a guardrail component to a flow app ... 457
Call functions from your Amazon Bedrock chat agent app ... 459

Function schema ... 460
Authentication methods ... 464
Create a function component ... 465
Add a function component to an Amazon Bedrock chat agent app ... 466
Use app history to view and restore app versions ... 467
Use your app outside of Amazon SageMaker Unified Studio ... 468
App export files ... 468
Export your app ... 470
Deploy an exported app ... 470
Run a deployed app ... 474
Get started with Amazon Bedrock in SageMaker Unified Studio ... 475
Chat with a model in the chat playground ... 476
Machine learning ... 479
Machine learning in Identity Center-based domains ... 481
Discover Jumpstart models ... 482
Build models in JupyterLab ... 482
Train models ... 483
Use inference endpoints to deploy models ... 483
Pipelines ... 485
Model registry ... 490
Track experiments using MLflow ... 492
HyperPod clusters ... 499
Partner AI apps ... 501
Machine Learning Workflows in IAM-based domains ... 501
Discover Foundation models ... 502
Deploy Foundation models ... 502
Track experiments using MLflow ... 503
Create and monitor custom training jobs ... 505
Manage Registered Models ... 506
Feature Store ... 508
Common use cases ... 521
Train and deploy an XGBoost model ... 521
Overview ... 521
Prerequisites ... 522
Step 1: Open your notebook and configure your project bucket ... 522
Step 2: Download and explore the dataset ... 523

Step 3: Prepare the data for XGBoost ... 524
Step 4: Train the XGBoost model ... 525
Step 5: Track and register the model with MLflow ... 527
Step 6: Deploy the model to a real-time endpoint ... 527
Step 7: Test predictions ... 528
Troubleshooting ... 528
Step 8: Clean up ... 529
Platform and administration ... 530
Projects ... 531
Create a new project ... 532
Step 1: Project name and description ... 532
Step 2: Customize parameters ... 533
Step 3: Review ... 534
Next steps ... 535
Get project details ... 535
View the SageMaker AI domain details associated with your project ... 536
Edit a project ... 536
Update a project ... 537
Move a project ... 538
Delete a project ... 539
Add project members ... 540
Remove project members ... 541
Using a project with trusted identity propagation enabled ... 542
Create a new project with trusted identity propagation ... 542
Working with Amazon Athena with trusted identity propagation ... 543
Connect to existing Amazon Redshift clusters or Serverless endpoints with trusted identity propagation ... 543
Source control and file storage ... 545
S3 storage ... 545
Key benefits of S3 storage: ... 545
Git-based storage ... 546
Key benefits of Git-based storage include: ... 546
Storage working in different tools ... 546
Web-based tools ... 547
Space-based tools ... 547
Managing storage resources ... 549

Storage organization ... 549
Working with files across multiple tools ... 549
Transitioning between storage types ... 549
Limitations ... 331
Compute ... 551
Amazon Athena ... 552
Gaining access to Amazon Athena resources ... 552
Connecting to an existing Amazon Athena resource ... 556
Amazon Redshift ... 556
Gaining access to Amazon Redshift resources ... 557
Connecting to an existing Amazon Redshift resource ... 562
Creating a new Amazon Redshift Serverless compute resource ... 563
Removing an Amazon Redshift compute connection ... 563
Amazon EMR on EC2 ... 564
Adding a new Amazon EMR on EC2 cluster ... 564
Adding an existing Amazon EMR on EC2 cluster ... 565
Using an Amazon EMR on EC2 cluster ... 572
Monitoring Amazon EMR on EC2 clusters ... 573
Configuring TIP ... 573
Configuring user background sessions ... 575
Terminating and removing an Amazon EMR on EC2 cluster ... 576
Spark History Server ... 576
Get started adding on-demand Amazon EMR on EC2 instances ... 576
Amazon EMR on EKS ... 580
Spark History Server for Amazon EMR on EKS in Amazon SageMaker Unified Studio ... 580
Adding a new Amazon EMR on EKS virtual cluster ... 580
Using an Amazon EMR on EKS virtual cluster ... 581
Removing an Amazon EMR on EKS virtual cluster ... 582
EMR Serverless ... 582
Adding a new EMR Serverless application ... 582
Configuring permission mode ... 583
Configuring user background sessions ... 584
Deleting applications ... 585
Glue ETL ... 585
Configuring permission mode ... 586
Configuring user background sessions ... 587

Limitations of fine-grained permission mode ... 587
CI/CD ... 590
Key concept ... 590
Manifest ... 590
Stages and targets ... 590
Bundles ... 591
Catalog support ... 591
CLI commands ... 591
Prerequisites ... 593
Getting started ... 593
Install the CLI ... 593
Create a manifest ... 593
Validate your configuration ... 594
Deploy ... 594
Run post-deployment validation ... 594
Deployment modes ... 594
Direct deployment ... 594
Bundle-based deployment ... 595
Cross-region and cross-account ... 595
CI/CD integration ... 595
Sample IAM policy ... 595
Failure recovery ... 596
Current limitations ... 597
Additional resources ... 597
Kiro CLI ... 598
Prerequisites ... 598
To sign in and start Kiro CLI ... 598
Using Kiro CLI ... 599
Capabilities ... 600
Example: Data catalog ... 600
Example: Project context ... 600
Troubleshooting ... 601
Additional information ... 601
Administration ... 603
Events and notifications ... 604
Events via the dedicated inbox in the Amazon SageMaker Unified Studio ... 604

Catalog and Governance ... 610
Catalog in IDC-based domains ... 611
Data governance ... 611
Create a business glossary ... 613
Edit a business glossary ... 614
Delete a business glossary ... 615
Create a term in a glossary ... 616
Edit a term in a glossary ... 617
Delete a term in a glossary ... 618
Create a metadata form ... 618
Edit a metadata form ... 619
Delete a metadata form ... 620
Create a field in a metadata form ... 621
Edit a field in a metadata form ... 622
Delete a field in a metadata form ... 622
Restricted asset classification ... 623
Data products ... 625
Create new data products ... 626
Publish data products ... 627
Edit data products ... 628
Unpublish data products ... 629
Delete data products ... 630
Subscribe to a data product ... 630
Review a subscription request and grant a subscription to a data product ... 631
Republish data products ... 632
Data inventory and publishing ... 633
Configure Lake Formation permissions for Amazon SageMaker Unified Studio ... 634
Create custom asset types ... 635
Create a data source for AWS Glue ... 637
Create a data source for Amazon Redshift ... 642
Create a data source for Amazon SageMaker AI ... 646
Edit a data source ... 649
Delete a data source ... 650
Publish assets to the catalog from the project inventory ... 650
Share assets ... 652
Manage inventory and curate assets ... 652

Manually create an asset ... 654
Unpublish an asset from the catalog ... 655
Delete an asset ... 656
Manually start a data source run ... 657
Asset versioning ... 658
Data quality ... 658
Using machine learning and generative AI in Amazon SageMaker Unified Studio ... 662
Analyze your subscribed data with external analytics applications via JDBC connection ... 667
Metadata enforcement rules for publishing ... 670
Data discovery, subscription, and consumption ... 671
Search for data in the catalog ... 672
Request subscription to assets ... 674
Approve or reject a subscription request ... 675
Revoke an existing subscription ... 677
Cancel a subscription request ... 678
Unsubscribe from an asset ... 678
Grant access to managed AWS Glue Data Catalog assets ... 679
Grant access to managed Amazon Redshift assets ... 680
Grant access for approved subscriptions to unmanaged assets ... 681
Metadata enforcement rules for subscription requests ... 685
Fine-grained access control to data ... 686
Limitations ... 687
Create row filters ... 687
Create column filters ... 688
Delete row or column filters ... 689
Edit row or column filters ... 689
Grant access with filters ... 690
Exporting asset metadata ... 691
Requirements ... 692
Start exporting asset metadata ... 693
Asset table schema ... 694
Querying asset tables ... 696
Catalog in IAM-based domains ... 701
Browse and explore data ... 702
Browse catalog assets ... 702

Browse S3 buckets ... 704
Manage S3 bucket contents ... 705
Search and discover data ... 705
Search results page ... 706
Filter search results ... 707
View an asset's detail page ... 708
Request access ... 709
Request a subscription ... 710
What happens after approval ... 711
View and manage your subscriptions ... 711
Cancel a pending request ... 712
Unsubscribe from a table ... 713
Manage and govern the catalog ... 713
Approve and manage access ... 716
Create and manage business glossaries ... 723
Standardize metadata using metadata forms ... 730
Curate and enrich asset metadata ... 736
Publish data ... 753
Custom asset types ... 755
Data quality in the catalog ... 763
Create rulesets ... 763
Navigate to the Data quality tab ... 763
Create a ruleset ... 763
Write rules using DQDL ... 764
Migrate existing rulesets to your project ... 766
Run rulesets and view results ... 766
Run a ruleset ... 766
View run results ... 766
View evaluation run details ... 767
Review the dashboard ... 769
Schedule evaluations ... 770
Work with multiple rulesets ... 772
Data lineage ... 773
What is OpenLineage? ... 773
Data lineage support ... 774
Data lineage support matrix ... 775

Visualizing data lineage in IAM mode ... 779
Generating lineage ... 782
Node types ... 782
Column lineage (Timestamp mode only) ... 783
Graph controls ... 784
Known limitations ... 785
Visualizing data lineage in IdC mode ... 785
Aggregated lineage view ... 787
Test drive data lineage ... 789
Data lineage authorization ... 790
Automate lineage capture from data connections ... 790
Configure automated lineage capture for AWS Glue (Lakehouse) connections ... 791
Configure automated lineage capture for Amazon Redshift connections ... 792
Automate lineage capture from tools ... 792
Capture lineage for Spark executions in Visual ETL ... 793
Capture lineage for AWS Glue Spark executions in Notebooks ... 794
Capture lineage from EMR Spark executions ... 796
Permissions required for data lineage ... 797
Read permissions to view lineage ... 797
Write permissions to publish lineage ... 797
Publishing data lineage programmatically ... 799
The importance of the sourceIdentifier attribute to lineage nodes ... 801
Linking dataset lineage nodes with assets imported into Amazon SageMaker Unified Studio ... 807
Data lineage troubleshooting ... 808
Not seeing lineage graph for events published programmatically ... 809
Not seeing lineage for assets even though importLineage is shown as true in AWS Glue datasource ... 813
Not seeing lineage for assets even though importLineage is shown as true in Amazon Redshift datasource ... 813
Troubleshooting lineage for lineage events published from AWS Glue ETL jobs/vETL/Notebooks ... 815
Troubleshooting lineage for lineage events published from EMR-S/EC2/EKS ... 818
Third-party catalog integrations ... 821
Atlan integration ... 821
Capabilities ... 822

How it works ... 822
Collibra integration ... 822
Capabilities ... 823
How it works ... 823
Alation integration ... 824
Capabilities ... 824
How it works ... 824
Catalog and governance use cases ... 825
From raw tables to a governed data catalog ... 825
The scenario ... 825
The approach ... 826
The end result ... 833
Applying this pattern to your own data ... 835
Reference and resources ... 837
Terminology and concepts ... 838
Troubleshooting ... 851
Troubleshooting issues related to subscriptions in Amazon SageMaker Unified Studio ... 851
Amazon EBS Volume Depletion with Local Notebook Execution ... 854
Domain ... 854
SAML Identity Provider Email Issue ... 855
Project Creation Failure ... 855
Data Explorer Visibility Issue ... 856
Data Catalog Visibility Issue ... 856
Connection to Amazon RDS MySQL in Existing VPC ... 856
Visual ETL Flow Column Selection ... 857
Invalid or expired auth token when accessing an IDE ... 857
JupyterLab Configure Magic Error ... 858
Local IDE troubleshooting ... 858
Quotas ... 861
Release notes ... 863
July 2026 ... 863
July 21, 2026 ... 863
July 17, 2026 ... 864
July 9, 2026 ... 864
July 9, 2026 ... 865
July 9, 2026 ... 865

July 7, 2026 ... 866
July 7, 2026 ... 866
July 2, 2026 ... 867
June 2026 ... 867
June 17, 2026 ... 867
June 9, 2026 ... 867
June 4, 2026 ... 868
June 4, 2026 ... 868
June 3, 2026 ... 868
June 3, 2026 ... 869
June 3, 2026 ... 869
June 1, 2026 ... 870
May 2026 ... 870
May 22, 2026 ... 870
May 21, 2026 ... 871
May 21, 2026 ... 871
May 22, 2026 ... 871
May 20, 2026 ... 872
May 13, 2026 ... 872
May 11, 2026 ... 873
May 7, 2026 ... 873
April 2026 ... 874
April 23, 2026 ... 874
April 22, 2026 ... 875
April 21, 2026 ... 876
April 7, 2026 ... 876
April 6, 2026 ... 877
April 3, 2026 ... 877
April 1, 2026 ... 878
March 2026 ... 878
March 31, 2026 ... 878
March 30, 2026 ... 879
March 25, 2026 ... 879
March 18, 2026 ... 880
March 17, 2026 ... 880
March 9, 2026 ... 881

March 5, 2026 ... 881
March 3, 2026 ... 882
February 2026 ... 883
February 4, 2026 ... 883
January 2026 ... 883
January 30, 2026 ... 883
January 20, 2026 ... 884
December 2025 ... 884
December 2, 2025 ... 884
November 2025 ... 885
November 21, 2025 ... 885
November 14, 2025 ... 886
November 10, 2025 ... 886
November 6, 2025 ... 887
October 2025 ... 887
October 27, 2025 ... 887
October 12, 2025 ... 887
October 11, 2025 ... 888
September 2025 ... 888
Septemeber 18, 2025 ... 888
September 12, 2025 ... 888
Septemeber 08, 2025 ... 889
August 2025 ... 890
August 22, 2025 ... 890
August 7, 2025 ... 890
August 5, 2025 ... 891
July 2025 ... 891
July 16, 2025 ... 891
July 15, 2025 ... 891
July 16, 2025 ... 891
July 15, 2025 ... 892
July 15, 2025 ... 892
July 15, 2025 ... 892
July 15, 2025 ... 893
July 15, 2025 ... 893
July 15, 2025 ... 893

July 3, 2025 ... 894
July 3, 2025 ... 894
July 3, 2025 ... 894
July 25, 2025 ... 894
July 23, 2025 ... 895
July 2, 2025 ... 895
June 2025 (Additional) ... 895
June 30, 2025 ... 895
June 25, 2025 ... 895
June 2025 ... 896
June 12, 2025 ... 896
June 19, 2025 ... 896
June 5, 2025 ... 896
June 2, 2025 ... 897
May 2025 ... 897
May 30, 2025 ... 897
May 30, 2025 ... 897
May 29, 2025 ... 897
May 23, 2025 ... 898
May 22, 2025 ... 898
May 12, 2025 ... 898
May 12, 2025 ... 899

What is Amazon SageMaker Unified Studio?

As a part of the next generation of Amazon SageMaker, the Amazon SageMaker Unified Studio is a unified development experience that brings together AWS data, analytics, artificial intelligence (AI), and machine learning (ML) services. It provides a place to build, deploy, execute, and monitor workflows from a single interface. This helps drive collaboration across teams and facilitate agile development.

In Amazon SageMaker Unified Studio, administrators manage the users and groups that can access Amazon SageMaker Unified Studio, and they set up resources for teams to use. When your administrator grants you access to Amazon SageMaker Unified Studio, you can contribute to Amazon SageMaker Unified Studio projects. Within Amazon SageMaker Unified Studio projects, you can collaborate on business use cases by creating and sharing data, computation work, and other resources.

This guide provides information about using Amazon SageMaker Unified Studio to create, contribute to, and manage projects. For more information about the administrator experience and flow for creating, configuring, and managing your domains, see the Amazon SageMaker Unified Studio Administrator Guide.

How does it work?

After your administrator has set up your domain and invited you to use Amazon SageMaker Unified Studio, you can start creating, managing, and using projects.

The flow for Amazon SageMaker Unified Studio users is as follows:

- Gain access to Amazon SageMaker Unified Studio by configuring your single sign-on (SSO) or IAM credentials and using the domain URL from your administrator.
- Create a new project or navigate to a project that you have been added to.
- Use the tools and resources within Amazon SageMaker Unified Studio to build, share, and execute applications within your project.

aws

User Guide

Amazon SageMaker Unified Studio

Get Started



Access Amazon SageMaker Unified Studio

For you to get started with Amazon SageMaker Unified Studio, your admin must create a domain in the Amazon SageMaker Unified Studio console and provide you with a URL. For more information, see the Amazon SageMaker Unified Studio Administrator Guide.

When you have the URL from your admin, you can sign in to Amazon SageMaker Unified Studio in one of the following ways:

- By using your AWS IAM credentials. For more information, see the section called "Sign up for an AWS account".
- If your admin has configured single sign-on (SSO) access, you can also sign in to Amazon SageMaker Unified Studio using SSO credentials that you configure with IAM Identity Center or through an identity provider. For more information, see the section called "Configure SSO credentials with IAM Identity Center".

Note

Amazon SageMaker Unified Studio supports the following browsers:

Browser	Version
Microsoft Edge	Latest 3 major versions
Google Chrome	Latest 3 major versions
Apple Safari	Latest 3 major versions

JupyterLab IDE requires third-party cookies to be allowed in your Amazon SageMaker Unified Studio domain. For more information, see the section called "Invalid or expired auth token when accessing an IDE".

Configure credentials

If you want to sign in to Amazon SageMaker Unified Studio using AWS IAM user or SSO credentials using IAM Identity Center, follow the instructions in the optional prerequiste sections below.

> **Note**
>
> You only need one method to sign in to Amazon SageMaker Unified Studio. If you have already configured an AWS account or SSO credentials that work with the domain URL you received from your admin, you can skip the steps in this section.

Topics

- Sign up for an AWS account
- Configure SSO credentials with IAM Identity Center

Sign up for an AWS account

To get started with AWS, you need an AWS account. For information about creating an AWS account, see Getting started with an AWS account in the *AWS Account Management Reference Guide*.

Configure SSO credentials with IAM Identity Center

You can use SSO with IAM Identity Center or with an identity provider. To use SSO with IAM Identity Center, work with your admin to get added to their IAM Identity Center directory and set up your SSO credentials.

The process is as follows:

1. After your admin adds your user information to their IAM Identity Center directory, you receive an email with your username and configuration instructions for single sign-on (SSO). Use the link in the email to set your password for SSO.
2. Your admin creates configurations and adds you to a domain using the Amazon SageMaker Unified Studio console. They then copy the link to that domain from the Amazon SageMaker Unified Studio console and send it to you. Use the domain URL from your admin to navigate to Amazon SageMaker Unified Studio.

3. Sign in to Amazon SageMaker Unified Studio with the SSO username and password that you configured in step 1.
4. If your admin's IAM Identity Center is configured to require multi-factor authentication (MFA), set up and use an MFA device. Follow the instructions on the screen to register or use an MFA device as needed, or contact your admin for support. For more information about MFA device enforcement, see Configure MFA device enforcement in the IAM Identity Center User Guide.

You are then able to view Amazon SageMaker Unified Studio landing page, where you can create new projects and view projects that you have been added to.

Set up your domain as an administrator

Time: 10 minutes

Prerequisites: An AWS account with administrator access. If you don't have an account, see the prerequisites section.

Outcome: You create a SageMaker Unified Studio domain, set up a project, and add a member so your team can start working with data and AI.

What you will do

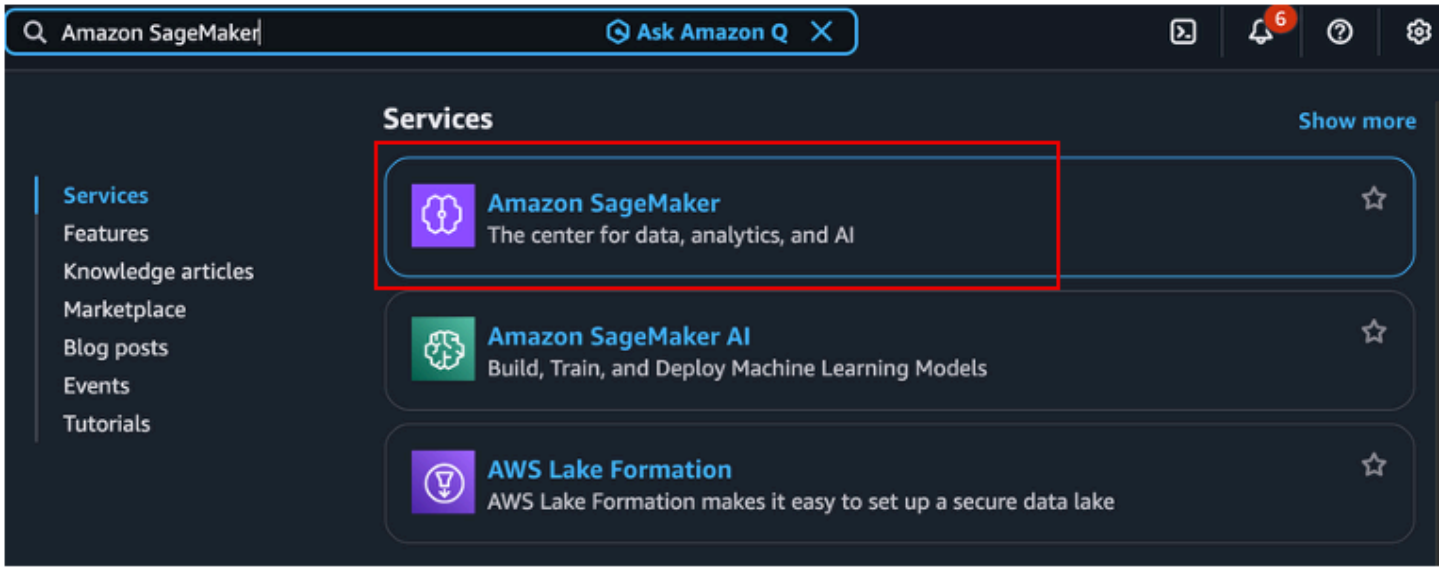
In this tutorial, you will:

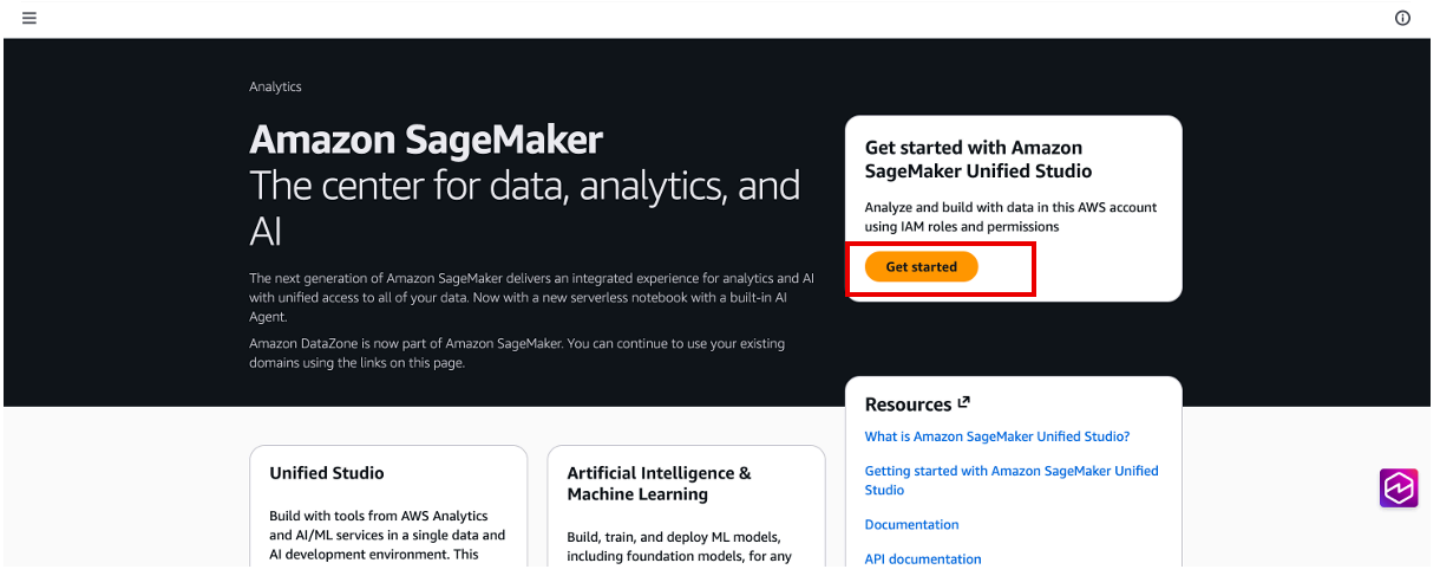
- Open the SageMaker Unified Studio setup in the AWS console
- Configure login and execution IAM roles
- Set up data storage and encryption options
- Create your domain
- Create a project and add a team member

This tutorial walks you through setting up an IAM-based domain in SageMaker Unified Studio. An IAM-based domain uses AWS Identity and Access Management (IAM) roles for authentication and access control. Only one IAM-based domain is available per AWS account per Region.

Step 1: Open SageMaker Unified Studio in the console

1. Sign in to the AWS Management Console.
2. In the search bar at the top, type `SageMaker` and choose **Amazon SageMaker** from the results.
3. On the SageMaker landing page, choose **Get started**.

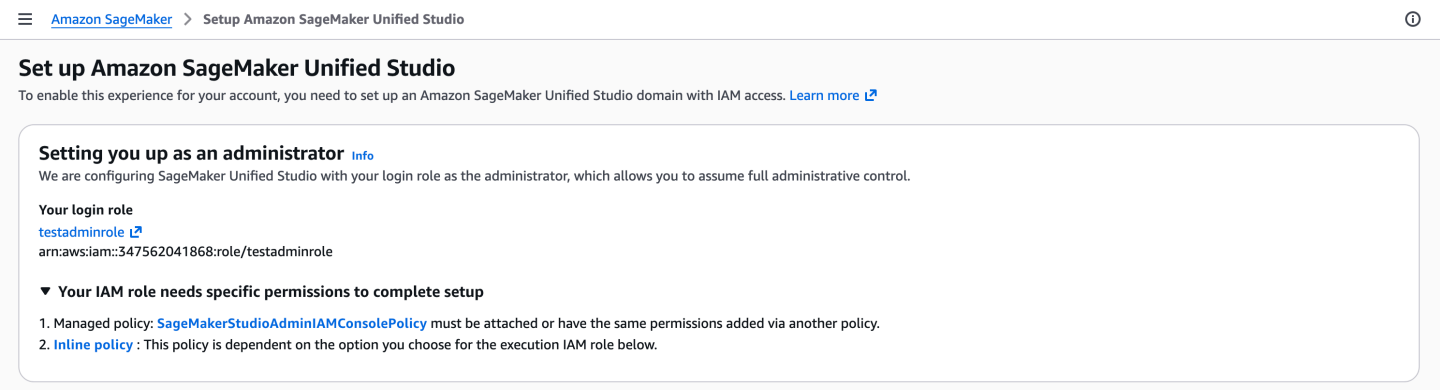




The setup flow opens and guides you through configuring your domain.

Step 2: Configure the login role

The login role authenticates the administrator and provides access to SageMaker Unified Studio. The setup flow shows your current IAM role.

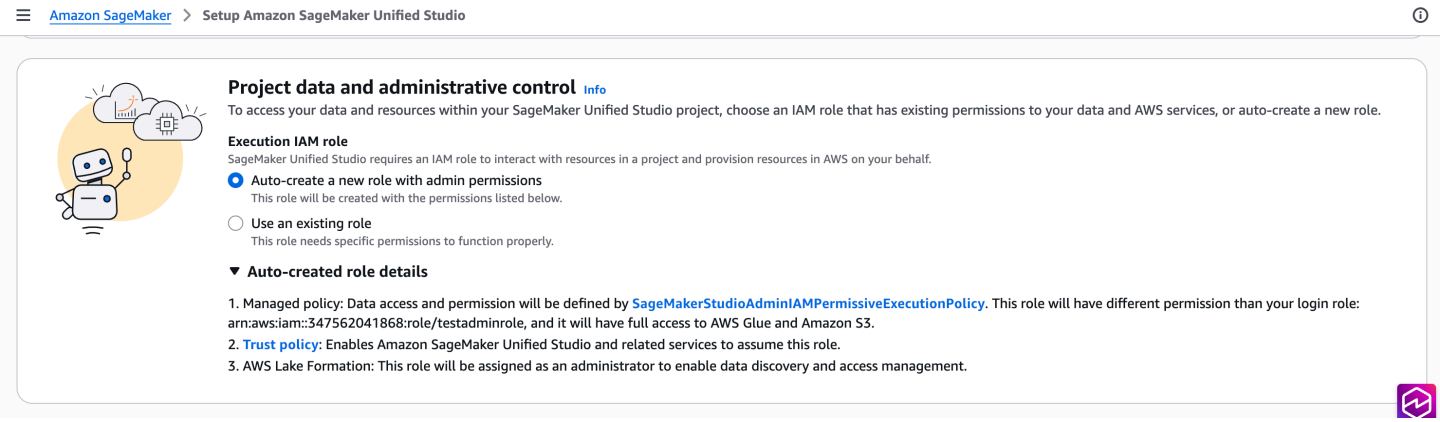


1. Review the login role shown. This is the IAM role that you will use to sign in to SageMaker Unified Studio as an administrator.
2. The login role must have the SageMakerStudioAdminIAMConsolePolicy managed policy attached.

> **Note**
>
> The login role also requires an inline policy based on your choice of execution role in the next step.

Step 3: Configure the execution role

The execution role defines which AWS services and data can be accessed through SageMaker Unified Studio projects.



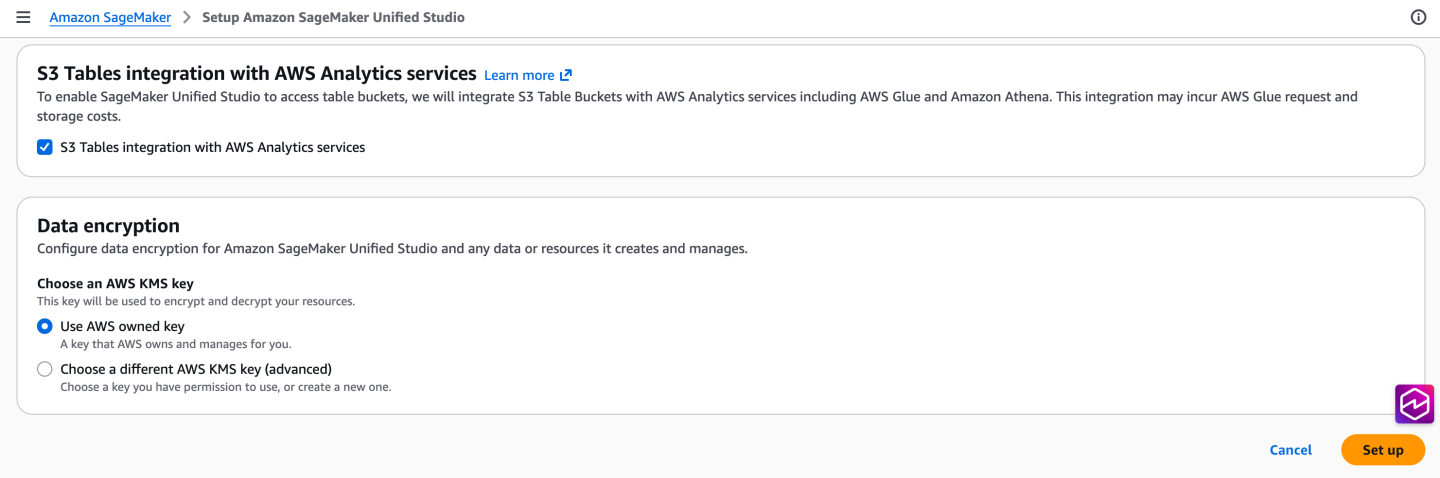
1. Choose whether to let SageMaker Unified Studio auto-create an execution role, or select an existing IAM role.

2. If you use an existing role, it must have the SageMakerStudioAdminIAMPermissiveExecutionPolicy managed policy attached.

> **Note**
>
> The execution role needs a trust policy that allows SageMaker Unified Studio and related AWS services to assume it. The execution role is also assigned administrator permissions for AWS Lake Formation. The execution role can be the same IAM role as the login role. If you use separate roles, make sure each has the appropriate managed policy attached.

Step 4: Configure storage and encryption

Choose your data storage and encryption settings.



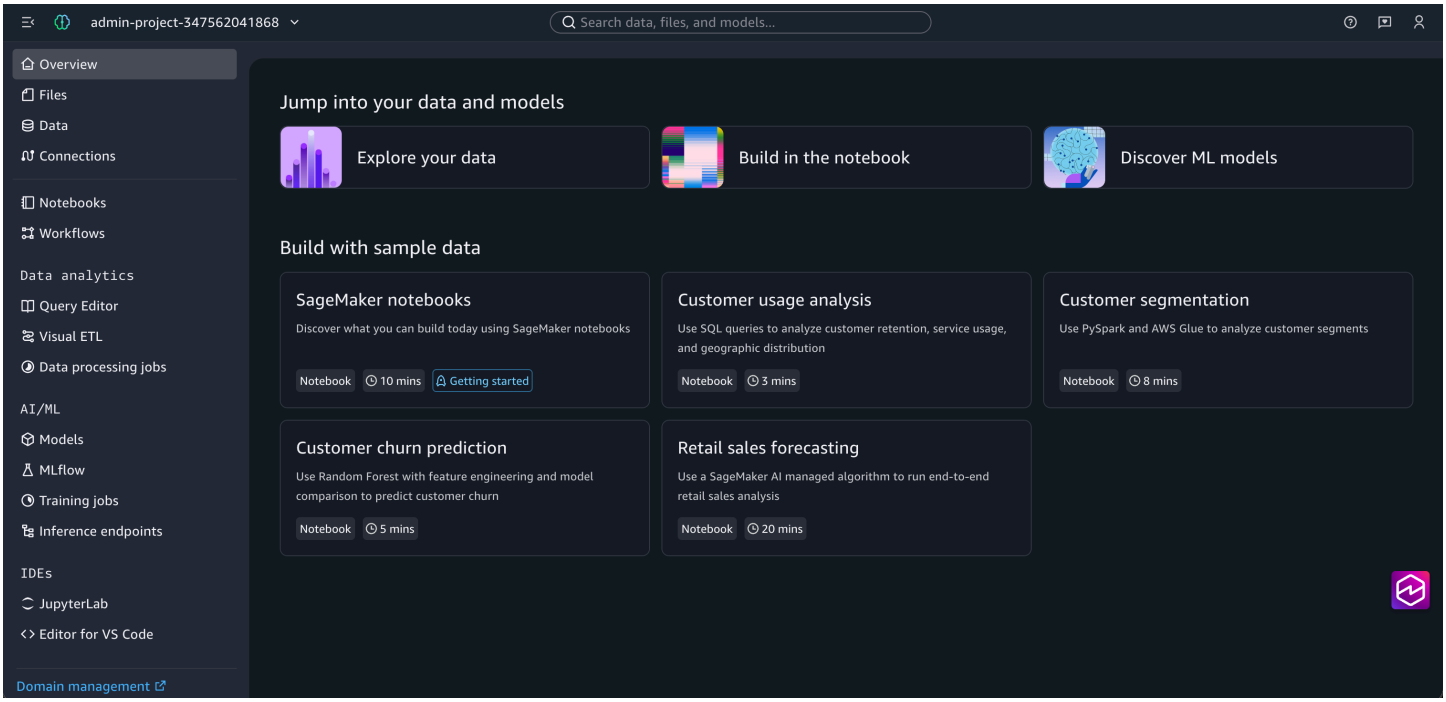
1. Choose whether to enable S3 tables integration for your domain.
2. For data encryption, choose **AWS owned key** (default) or provide your own AWS KMS key.
3. Choose **Set up** to create your domain.



Domain creation takes a few minutes. When it completes, you see the default project screen.

Step 5: Explore your new domain

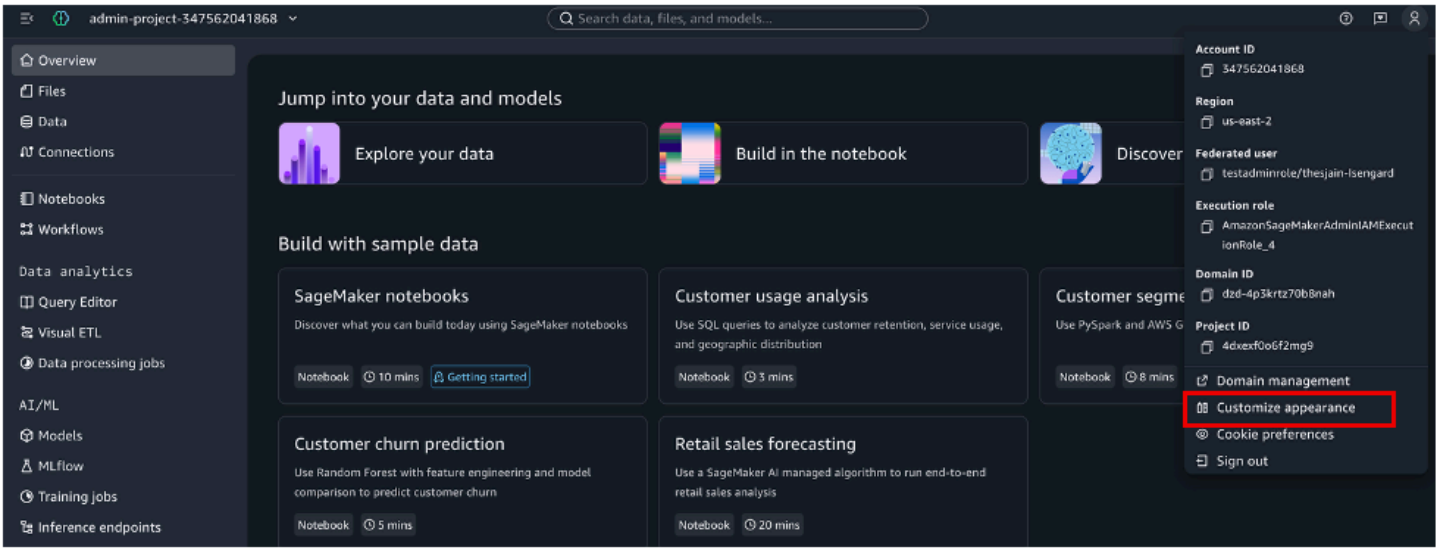
After the domain is created, you land on the default project screen.

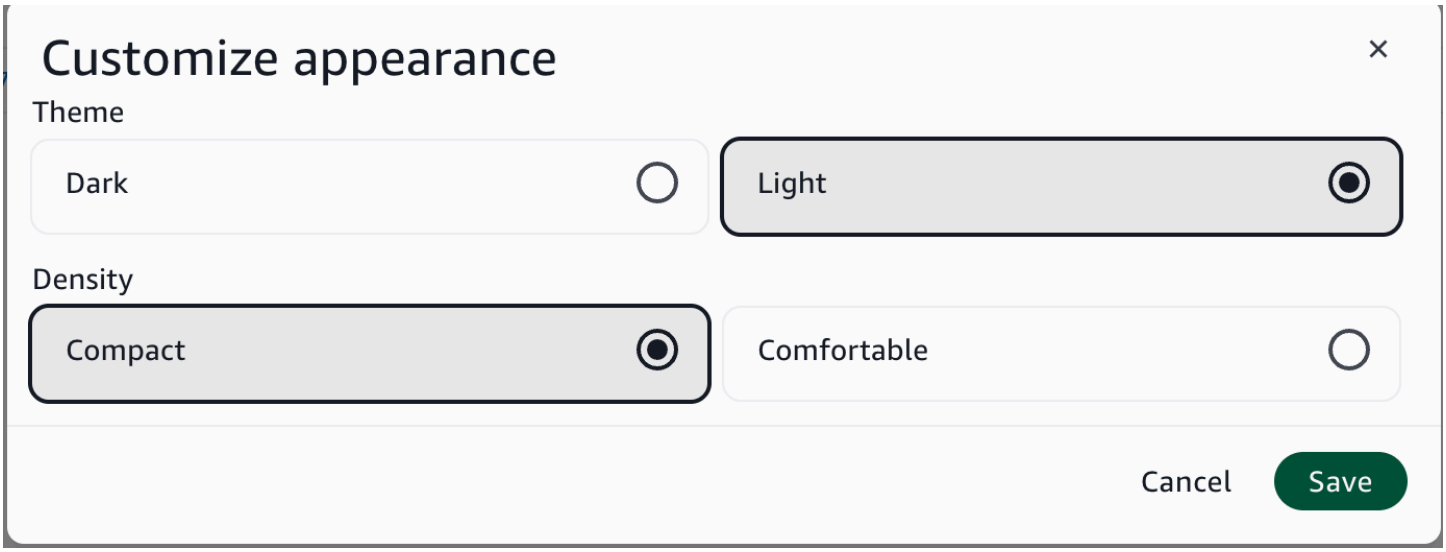


You can customize the appearance of SageMaker Unified Studio to your preference.

1. Choose the **Customize appearance** option.

2. Choose **Light** if you prefer a lighter interface.





Step 6: Create a project and add members

Now that your domain is ready, create a project and add team members so they can start working. For detailed instructions, see *Create a project and add members*.

What you learned

In this tutorial, you:

- Created an IAM-based SageMaker Unified Studio domain

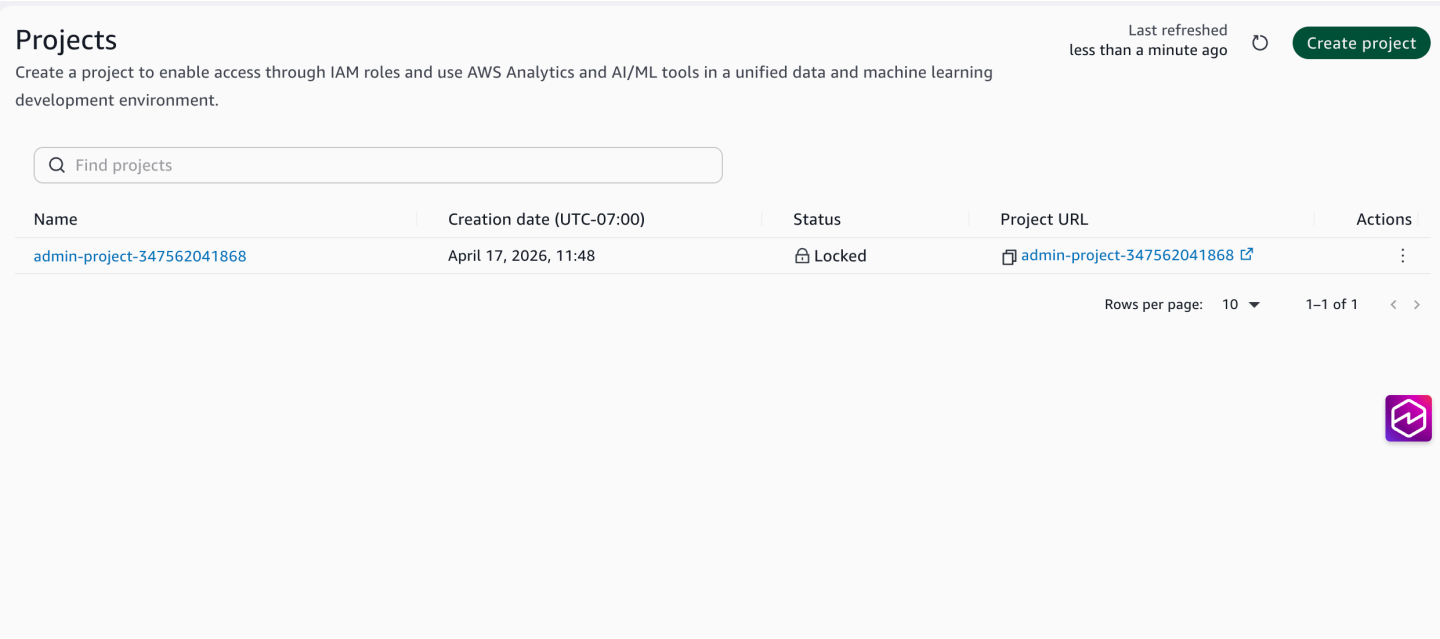
- Configured login and execution IAM roles with the required managed policies
- Set up data storage and encryption for your domain

Your team members can now sign in to SageMaker Unified Studio and start using the tutorials in this guide.

Create a project and add members

Projects are where your team collaborates on data and AI tasks in Amazon SageMaker Unified Studio. You can create a project and add up to 8 members at a time.

To create a project, navigate to the **Projects** page and choose **Create project**.



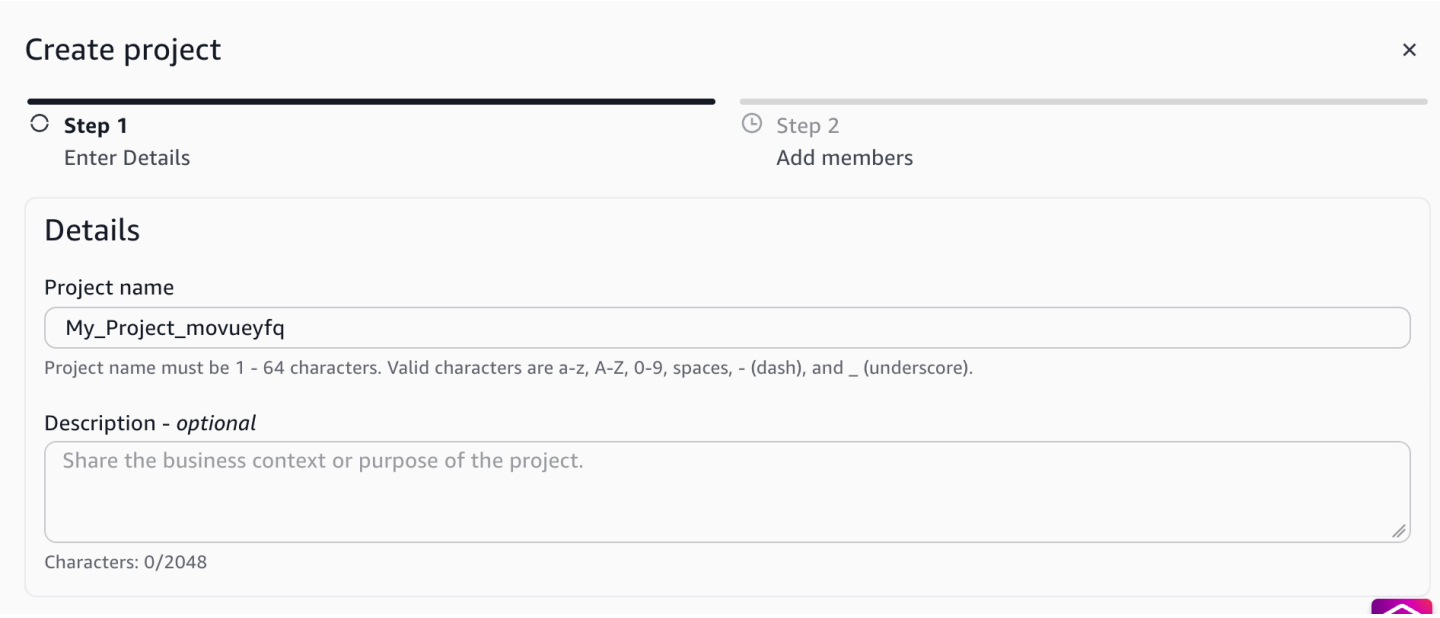
Step 1: Enter details

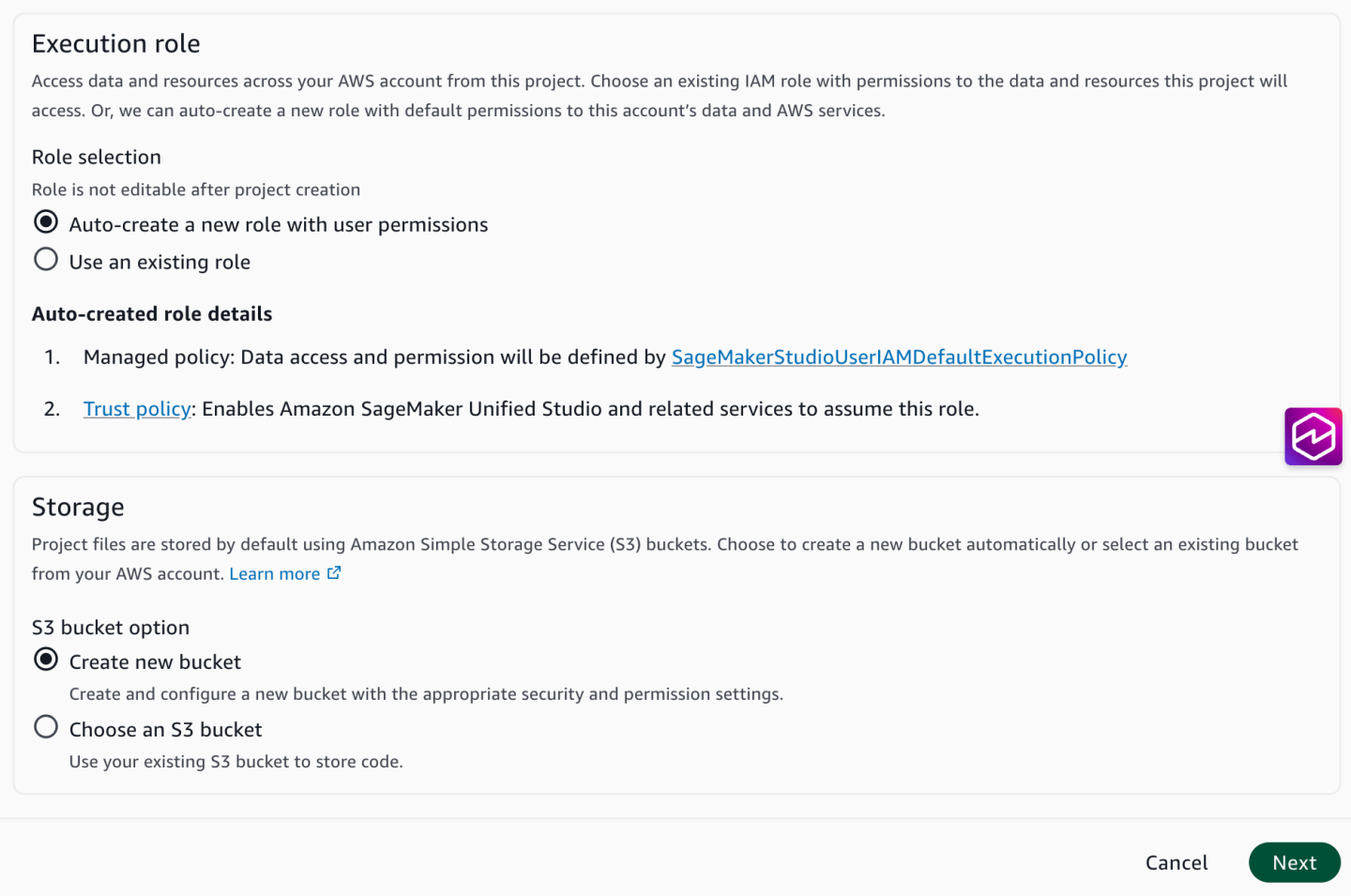
Enter the basic information for your project.

1. Enter a **Project name**. The name must be 1-64 characters and can contain letters, numbers, spaces, hyphens, and underscores.
2. (Optional) Enter a **Description** to share the business context or purpose of the project.
3. For **Execution role**, choose one of the following:
   - **Auto-create a new role with user permissions**. Creates a role with the SageMakerStudioUserIAMDefaultExecutionPolicy managed policy and a trust policy that allows Amazon SageMaker Unified Studio and related services to assume the role.
   - **Use an existing role**. Select an existing IAM role with the appropriate permissions.

> **Note**
>
> The execution role is not editable after project creation.

4. For **Storage**, choose one of the following:
   - **Create new bucket**. Creates and configures a new S3 bucket with the appropriate security and permission settings.
   - **Choose an S3 bucket**. Use an existing S3 bucket to store project code.
5. Choose **Next**.

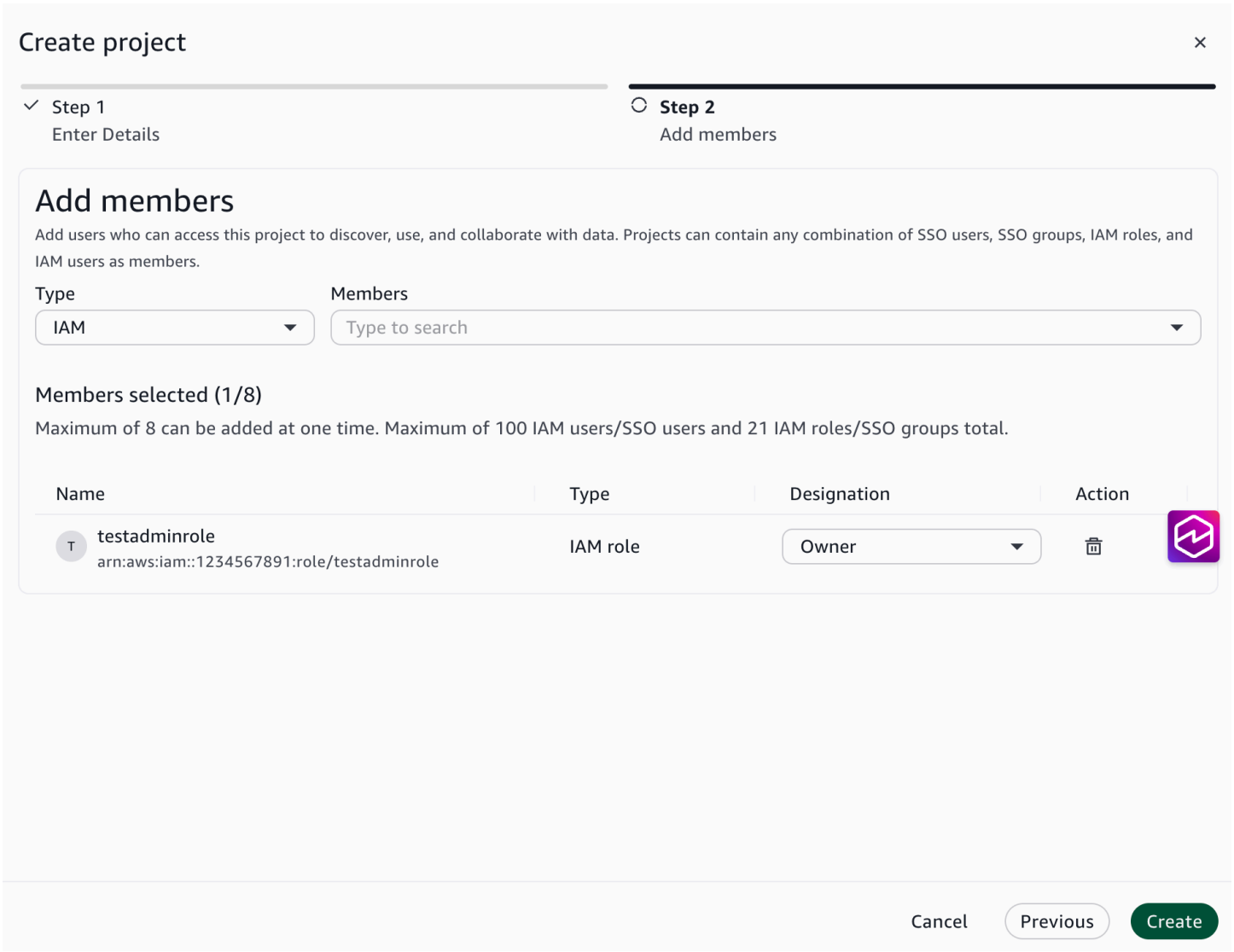




Step 2: Add members

Add users who can access the project to discover, use, and collaborate with data. Projects can contain any combination of SSO users, SSO groups, IAM roles, and IAM users as members. You can add a maximum of 8 members at a time.

1. Choose a **Type** from the dropdown (IAM, SSO user, or SSO group).
2. In the **Members** field, search for and select the member to add.
3. For each member, choose a **Designation** (for example, Owner).
4. (Optional) Repeat to add more members, up to 8 at a time.
5. Choose **Create** to create the project.



Note

The maximum total membership for a project is 100 IAM users or SSO users and 21 IAM roles or SSO groups.

Navigate Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio provides a comprehensive integrated development environment for machine learning (ML) and data science workflows. **For SageMaker Unified Studio domains configured with IAM roles**, you will be able to access the following components from the project overview page.

Navigation Panel

The left sidebar contains hierarchical navigation to access various Amazon SageMaker Unified Studio interfaces organized by:

Overview

- Files: Browser interface for local file system storage and S3 buckets.
- Data: Browser interface for catalog asset management
- Connections: Centralized view for all compute and data connections
- Notebooks: Serverless notebook interface
- Workflows: Orchestrate jobs and tasks

Data analytics

- Query Editor: Dedicated SQL interface.
- Visual ETL: Visual interface for Extract, Transform, Load operations
- Data processing jobs: View and manage job execution

AI/ML

- Models: Jump start into available models – foundation and registered.
- MLflow: Manage machine learning lifecycles
- Training jobs: Managing model training processes
- Inference endpoints: Deployment and endpoint management

Integrated development environments (IDEs)

- JupyterLab: Managed JupyterLab integrated development environment
- Editor for VS Code: Visual Studio Code integrated development environment
- Code spaces: Create and manage multiple individually configured development environments. For more information, see *Code spaces*.

Domain Management

For IAM roles with administrator privileges to access the admin interface

Jump into your data and models

This top section provides quick access to common actions:

- Explore your data - Explore and analyze data using SQL
- Build in the notebook - Prepare data for analytics or to train and deploy machine learning models
- Discover ML models – Discover, deploy and manage models

Build with sample data

This section middle section offers pre-configured example projects:

- Customer usage analysis - SQL-based customer retention analysis
- Customer segmentation - PySpark and AWS Glue analysis
- Customer churn prediction - Random Forest implementation with feature engineering
- Retail sales forecasting - End-to-end retail sales analysis using Amazon SageMaker Unified Studio AI

Change the display mode

Amazon SageMaker Unified Studio defaults to your operating system's display preference (light or dark mode). You can override this setting to choose a specific mode.

To change the display mode

1. In the upper-right corner of the console, choose the profile icon.
2. Choose Customize appearance.
3. Select one of the following options:
   - **System**. Uses your operating system's light or dark mode setting.
   - **Light**. Always displays in light mode.
   - **Dark**. Always displays in dark mode.

Your preference is saved automatically and applied across sessions.

Change the display language

Amazon SageMaker Unified Studio supports 12 languages. By default, your preferred language is automatically detected based on your browser's default language settings. You can override this setting to choose a specific language.

Supported languages:

- English (American)
- Chinese (Simplified)
- Chinese (Traditional)
- French
- German
- Indonesian
- Italian
- Japanese
- Korean
- Portuguese (Brazilian)
- Spanish
- Turkish

To change the display language

1. In the upper-right corner of the console, choose the profile icon.
2. Choose **Language selector**.
3. Select your preferred language from the list.

Your preference is saved automatically and applied across sessions. The selected language applies across all pages within Amazon SageMaker Unified Studio.

> **Note**
>
> The following areas are not yet translated and display content in English:
>
> - In IAM-based domains: the Browse page in the Catalog and MLflow.
> - In Identity Center-based domains: ML Pipelines and MLflow.

Getting started tutorials

This chapter helps you get up and running with Amazon SageMaker Unified Studio. Follow the hands-on tutorials to explore key capabilities using pre-loaded sample data — from running your first SQL query and analyzing and visualizing data to building a data pipeline with Visual ETL, automating a data pipeline with Workflows, and training a machine learning model. Each tutorial guides you through a common task so you can quickly learn how to work with Amazon SageMaker Unified Studio.

Topics

- Run your first SQL query
- Analyze and visualize data
- Build a data pipeline with Visual ETL
- Automate a data pipeline with Workflows
- Train an ML model

Run your first SQL query

Time: 5 minutes

Prerequisites: As a member of a SageMaker Unified Studio project, your IAM role needs the following managed policies:

- SageMakerStudioUserIAMConsolePolicy to sign in and access the project.
- SageMakerStudioUserIAMDefaultExecutionPolicy to access data and resources within the project.

If you don't have access, contact your administrator. If you are the administrator who set up the project, you already have the required permissions.

Outcome: You query sample data using the built-in query editor, see results inline, and understand how to browse your data catalog.

What you will do

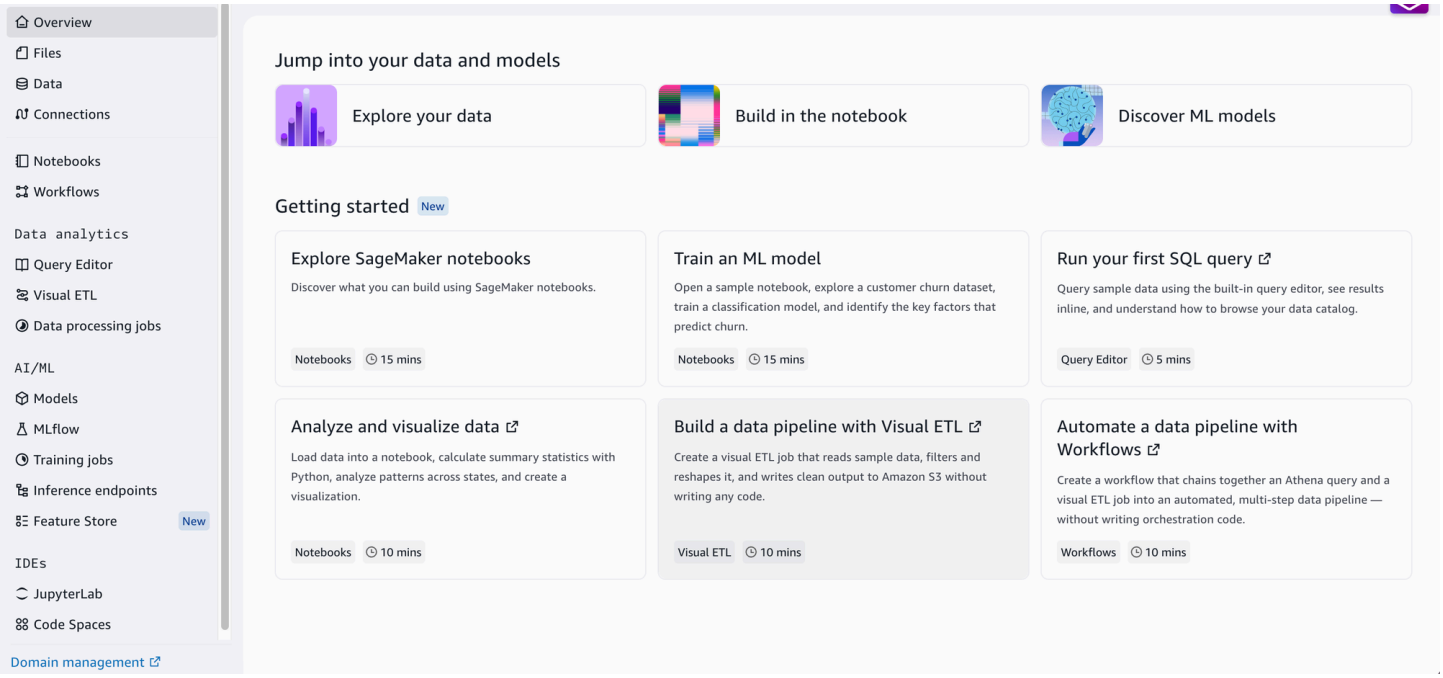
In this tutorial, you will:

- Open the query editor in your project
- Browse available tables in the data catalog
- Write and run a SQL query on sample data
- View and explore the results

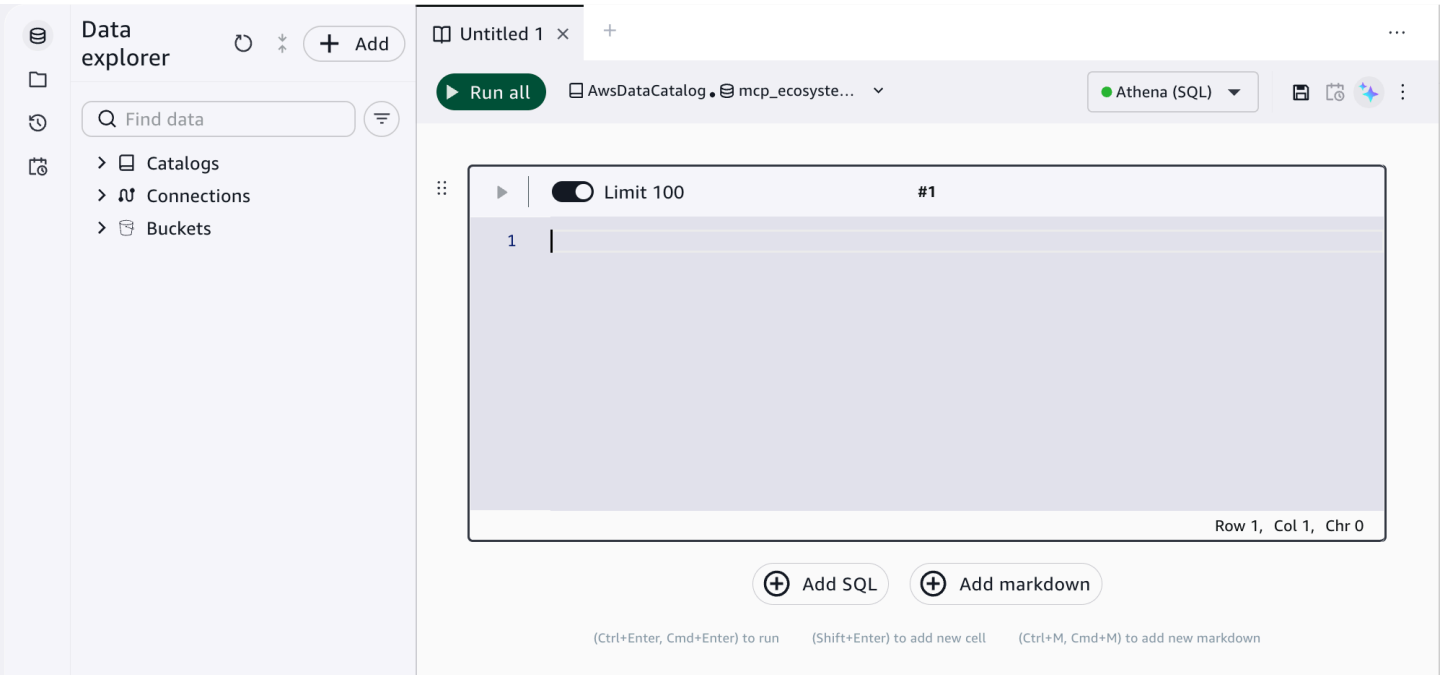
SageMaker Unified Studio includes a built-in query editor that lets you write SQL queries against data stored in your lakehouse. The data can be in Amazon S3, Amazon Redshift, or other connected sources. You don't need to set up a separate query tool or configure credentials. Everything is already connected through your project.

Step 1: Open the query editor

1. When you first sign in to SageMaker Unified Studio, you are in your default project. If you need to switch projects, use the project selector at the top of the page.
2. In the left navigation bar, choose **Query editor** under **Data analytics**.



This opens a *querybook*, an interactive SQL notebook where you can write multiple queries, add notes in markdown, and visualize results in one place.



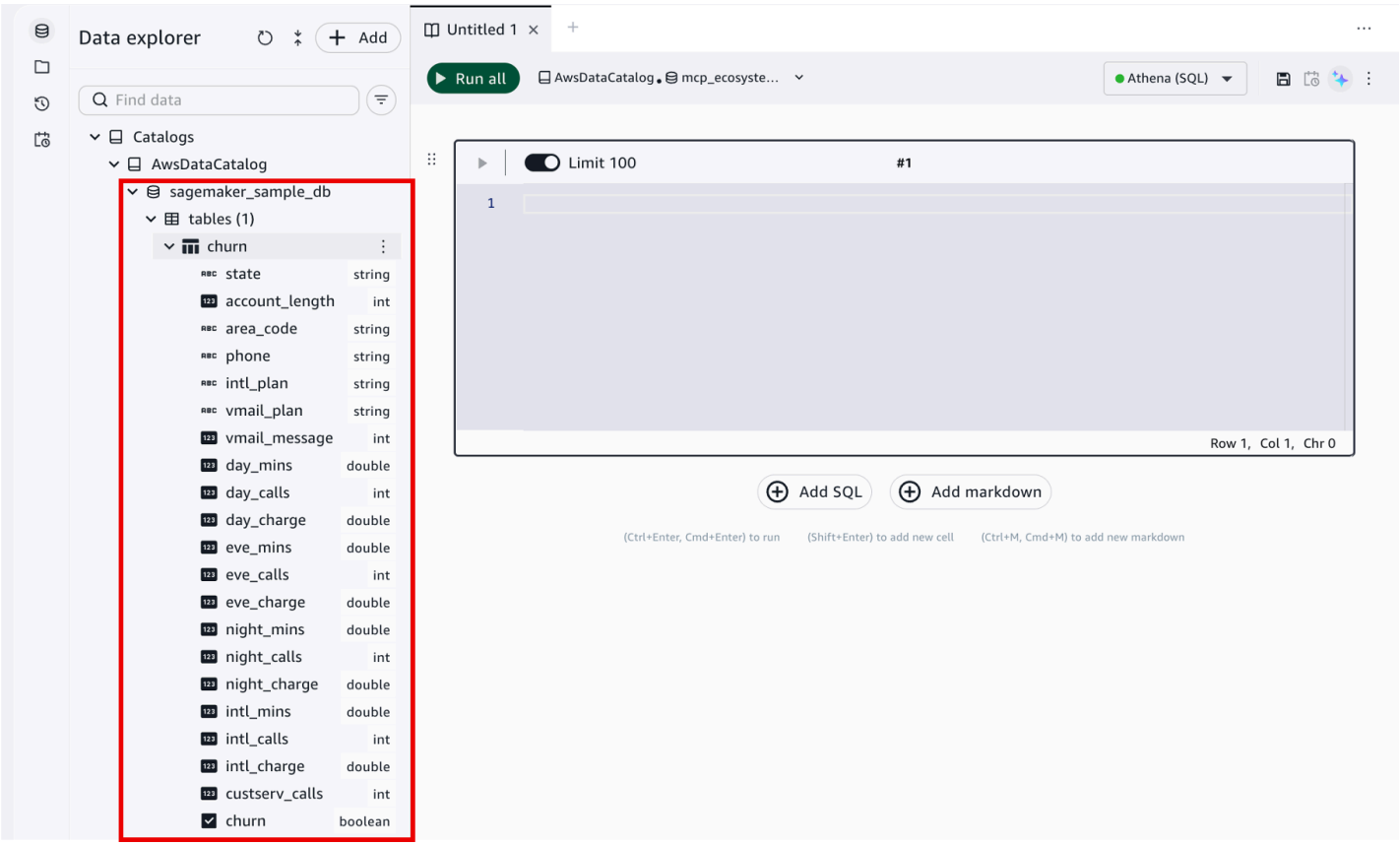
What is a querybook?

A querybook is like a notebook for SQL. Each cell contains a SQL query or markdown text. You can run cells individually or all at once, and results appear inline below each query.

Step 2: Browse your data

Before writing a query, review what data is available.

1. In the data explorer panel on the left side, expand **Catalogs**, then expand **AWSDataCatalog**.
2. Expand the **sagemaker_sample_db** database to see its tables. You should see a **churn** table.
3. Choose the **churn** table and review its columns and data types. This is sample data that was pre-configured in your project.



Step 3: Run a SQL query

Now query the sample data. Copy the following SQL into a querybook cell:

```
SELECT
    state,
    COUNT(*) AS total_customers,
    ROUND(AVG(day_mins), 2) AS avg_day_mins,
    ROUND(AVG(eve_mins), 2) AS avg_eve_mins,
    ROUND(AVG(custserv_calls), 2) AS avg_service_calls
FROM sagemaker_sample_db.churn
GROUP BY state
ORDER BY avg_service_calls DESC
LIMIT 10;
```

This query analyzes customer usage patterns by state. For each state, it calculates the total number of customers, their average daytime and evening minutes, and how often they contact customer service. The results show the top 10 states with the highest average service calls.

1. Paste the query into the SQL cell.

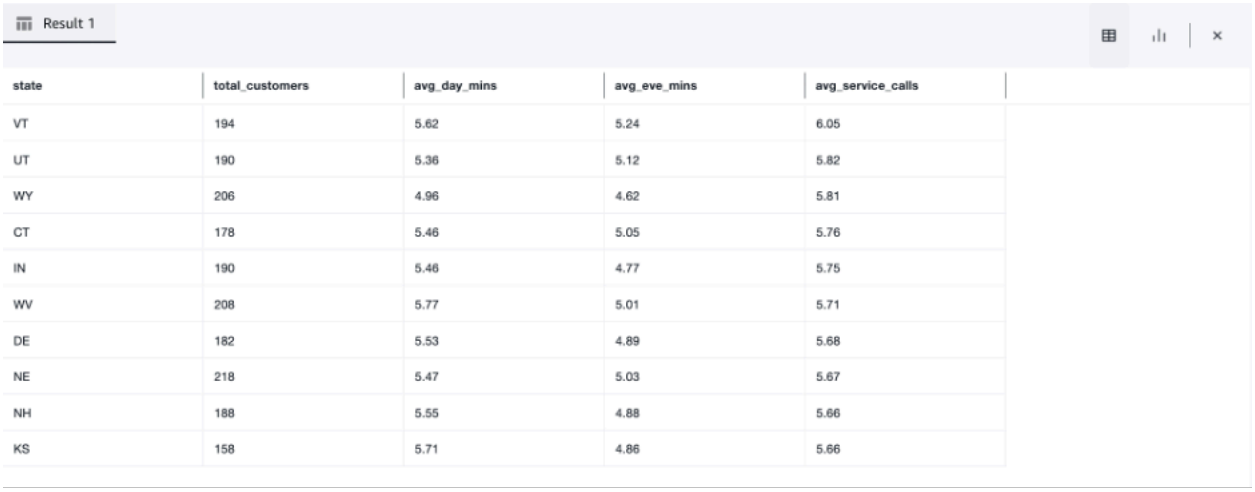
2. Choose **Athena (SQL)** from the engine selector dropdown at the top of the querybook.
3. Choose the **Run** button (▶) next to the cell.
4. Results appear in a table directly below the cell.

> **Which query engine is running this?**
>
> By default, your query runs on *Amazon Athena*, a serverless query engine that reads data directly from Amazon S3 without requiring you to load it into a database first. You can also switch to Amazon Redshift for data warehouse workloads using the engine selector in the querybook. You do not need to know the details of either engine to get started. Write standard SQL.

Step 4: Explore your results

After your query runs, the results table shows each state with its usage metrics. You can:

- **Sort columns** by choosing column headers.
- **Download results** as a CSV file using the download button.
- **Add another query** by choosing the **+** button to add a new SQL cell.



state	total_customers	avg_day_mins	avg_eve_mins	avg_service_calls
VT	194	5.62	5.24	6.05
UT	190	5.36	5.12	5.82
WY	206	4.96	4.62	5.81
CT	178	5.46	5.05	5.76
IN	190	5.46	4.77	5.75
WV	208	5.77	5.01	5.71
DE	182	5.53	4.89	5.68
NE	218	5.47	5.03	5.67
NH	188	5.55	4.88	5.66
KS	158	5.71	4.86	5.66

Try a different query. For example, compare usage patterns for customers with and without an international plan:

```
SELECT
```

```
    intl_plan,
    COUNT(*) AS total_customers,
    ROUND(AVG(intl_mins), 2) AS avg_intl_mins,
    ROUND(AVG(intl_charge), 2) AS avg_intl_charge,
    ROUND(AVG(custserv_calls), 2) AS avg_service_calls
FROM sagemaker_sample_db.churn
GROUP BY intl_plan;
```

This shows whether customers with an international plan use more international minutes and how their support call patterns compare.

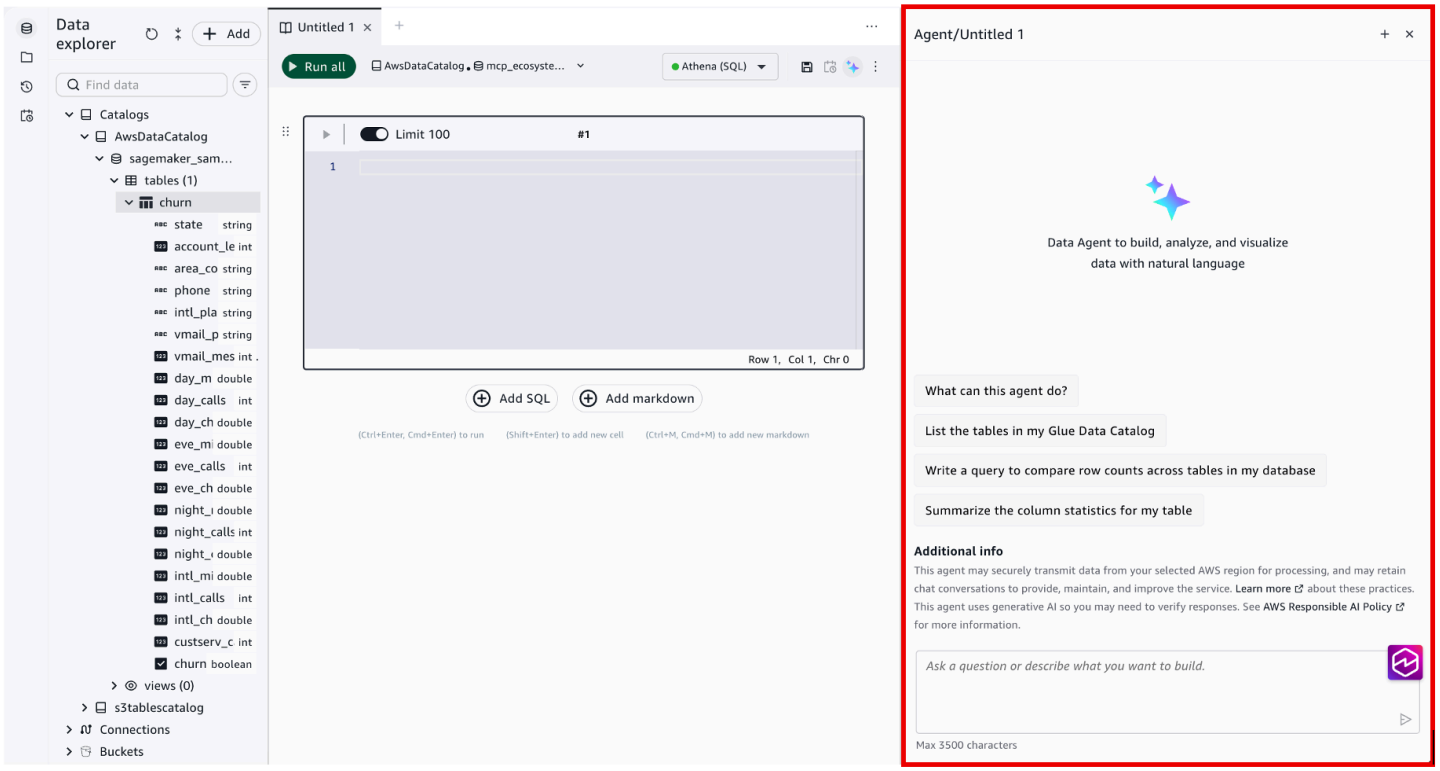


intl_plan	total_customers	avg_intl_mins	avg_intl_charge	avg_service_calls
yes	4986	5.02	4.4	5.51
no	5014	5.04	4.26	5.54

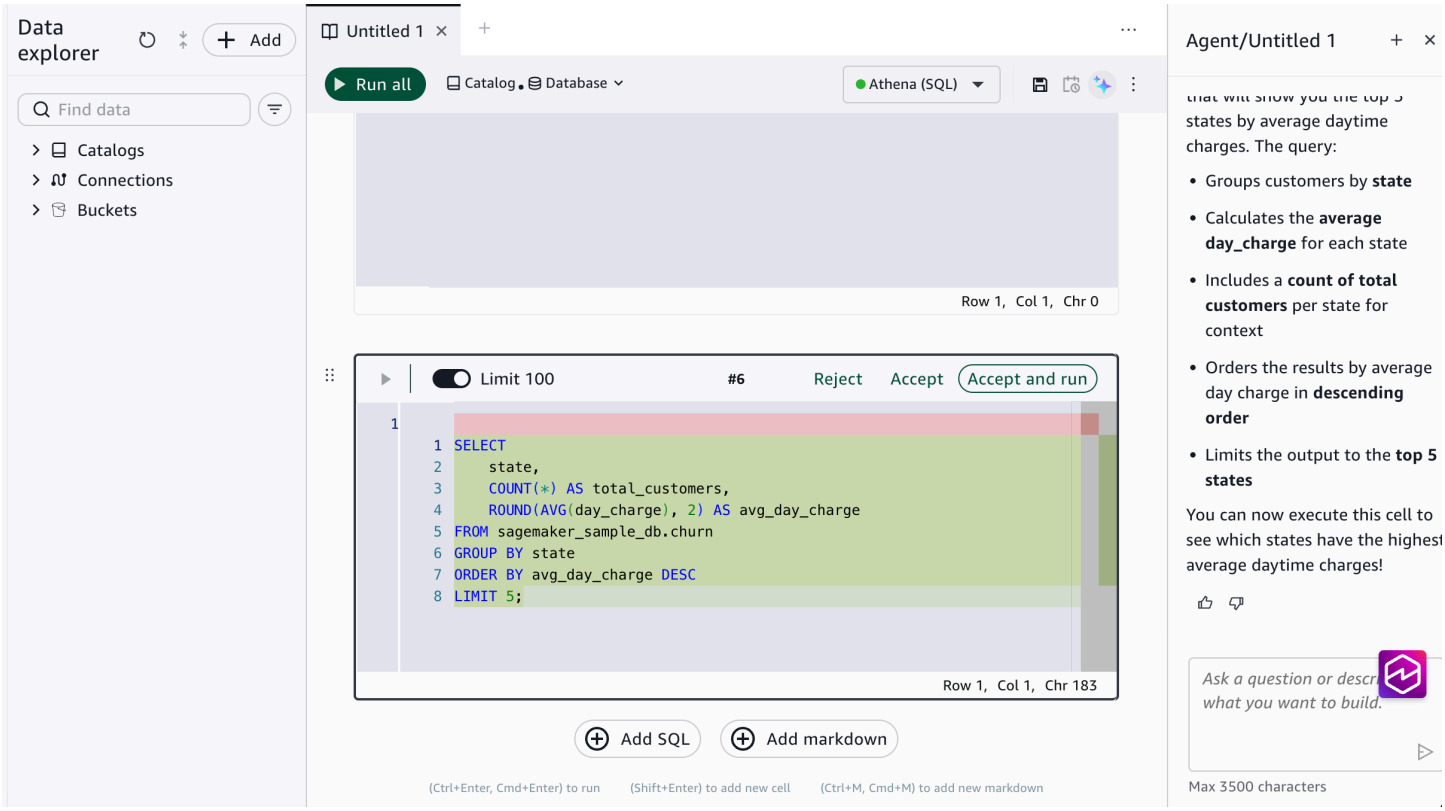
Ask in natural language

Instead of writing SQL yourself, you can ask the **Data Agent** to generate a query for you.

1. In the querybook, choose the **Chat with AI** icon in the top navigation bar. The Data Agent panel opens on the right side.
2. In the **Ask a question** text box at the bottom of the panel, type what you want in plain English, for example: *"Show me the top 5 states by average daytime charges"*
3. The Data Agent generates the SQL for you. Review it, then run it. The Data Agent has access to your project's data catalog, so it can identify which tables and columns are available.



After the Data Agent generates a query, choose **Accept** to insert it into a querybook cell, then choose **Run**.



The results appear in a table below the cell.



state	total_customers	avg_day_charge
GA	166	5.49
MD	226	5.33
IA	206	5.31
ME	148	5.26
MA	208	5.21

SQL in notebooks

You can also run SQL queries in a notebook using SQL cells. Notebooks let you combine SQL with Python code, visualizations, and markdown notes in a single document.

What you learned

In this tutorial, you:

- Opened the query editor and browsed available data in the catalog
- Wrote and ran a SQL query to analyze customer usage by state
- Explored results and ran a second query to compare international plan usage
- Used the Data Agent to generate SQL from natural language

Analyze and visualize data

Time: 10 minutes

Prerequisites: As a member of a SageMaker Unified Studio project, your IAM role needs the following managed policies:

- SageMakerStudioUserIAMConsolePolicy to sign in and access the project.
- SageMakerStudioUserIAMDefaultExecutionPolicy to access data and resources within the project.

If you don't have access, contact your administrator. If you are the administrator who set up the project, you already have the required permissions. Completing "Run your first SQL query" is helpful, but not required.

Outcome: You load data into a notebook, calculate summary statistics with Python, analyze patterns across states, and create a visualization.

What you will do

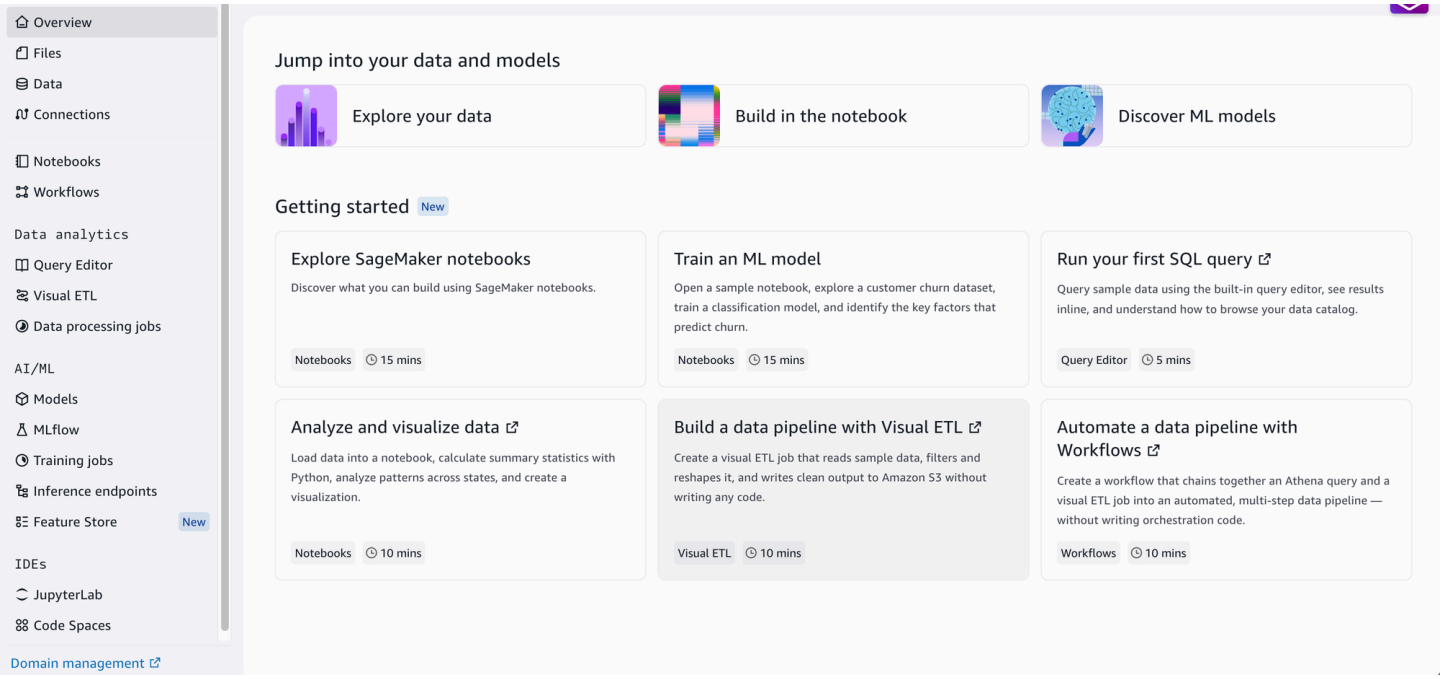
In this tutorial, you will:

- Open a notebook in your project
- Load sample data into a DataFrame for analysis
- Calculate summary statistics
- Analyze patterns by grouping data
- Create a chart to visualize the results

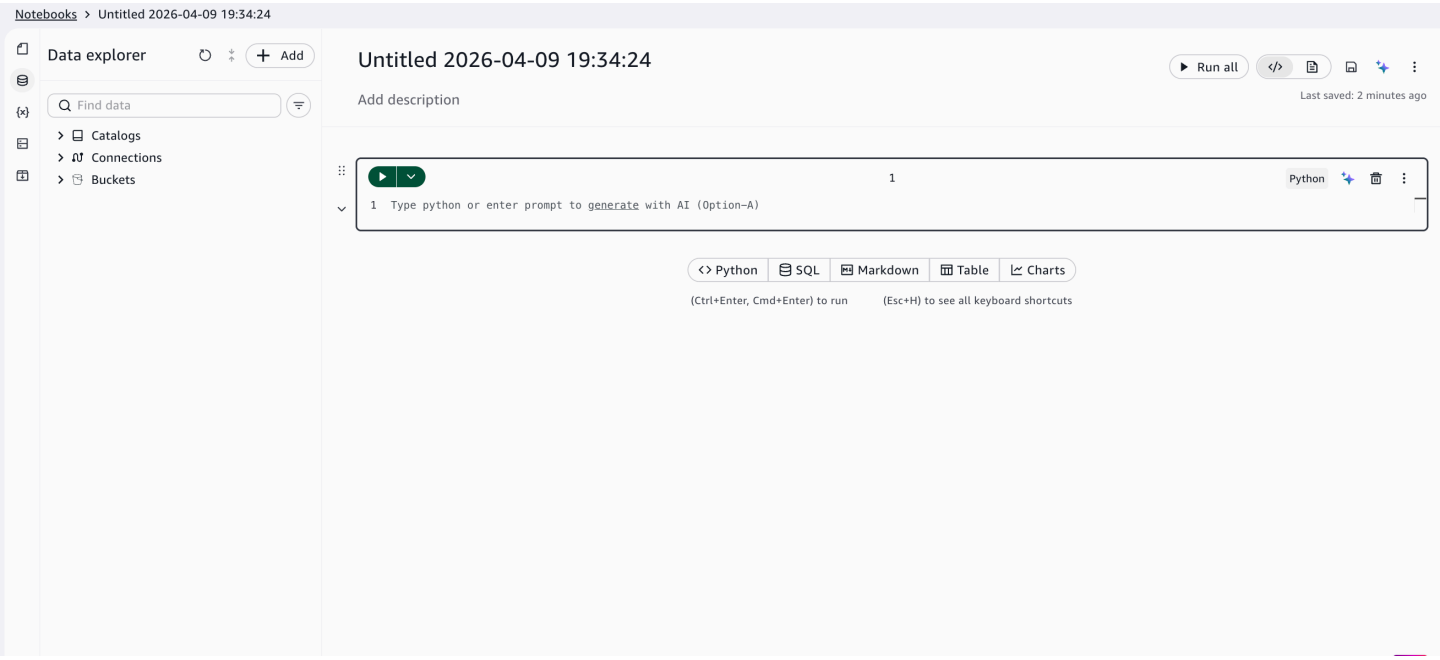
SageMaker Unified Studio notebooks give you a single environment for Python, SQL, and data visualization with serverless compute that scales automatically. The notebook connects directly to your project's data through the lakehouse, so the same tables you queried with SQL in the previous tutorial are available here too.

Step 1: Open a notebook

1. When you first sign in to SageMaker Unified Studio, you are in your default project. If you need to switch projects, use the project selector at the top of the page.
2. On the project overview page, choose **Build in the notebook**. Alternatively, choose **Notebooks** in the left navigation pane and choose **Create notebook**. A new notebook opens with an empty Python cell.



The notebook opens with a Python cell ready for input. You can also add SQL, Markdown, Table, and Charts cells using the options at the bottom of the notebook.



What runs your code?

Notebooks run on serverless compute powered by Amazon Athena for Apache Spark by default. Your code runs on managed infrastructure that scales automatically, without you provisioning anything.

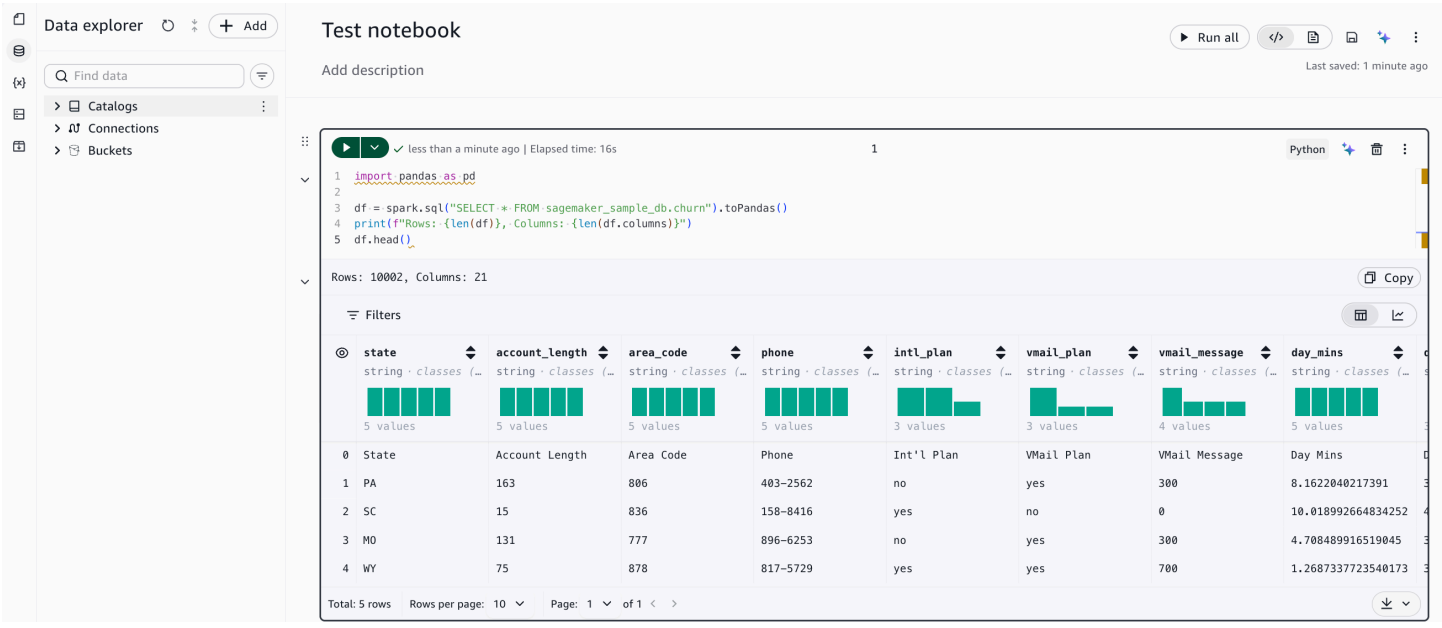
Step 2: Load the data

The same `sagemaker_sample_db.churn` table you browsed in the data catalog is available directly from your notebook. Load it into a pandas DataFrame so you can analyze it with Python. Paste the following code into the first Python cell and run it:

```
import pandas as pd

df = spark.sql("SELECT * FROM sagemaker_sample_db.churn").toPandas()
print(f"Rows: {len(df)}, Columns: {len(df.columns)}")
df.head()
```

The output shows the first few rows of the dataset, including columns for state, account length, call minutes, service calls, and churn status.



The dataset contains telecom customers with attributes including call minutes, service calls, charges, and whether the customer churned.

Work with DataFrames using SQL

Once you create a DataFrame, you can also query it using SQL cells in the notebook. This means you can use Python for some steps and SQL for others, depending on which is more convenient for the task.

Step 3: Clean the data and calculate statistics

Before analyzing the data, you need to handle a few issues. The first row contains duplicate header values, some numeric columns are stored as strings, and the churn column uses `"True."` and `"False."` instead of standard booleans. The following code removes the extra header row, converts the numeric columns to the correct data type, and maps the churn values to booleans. Add a new Python cell and paste it:

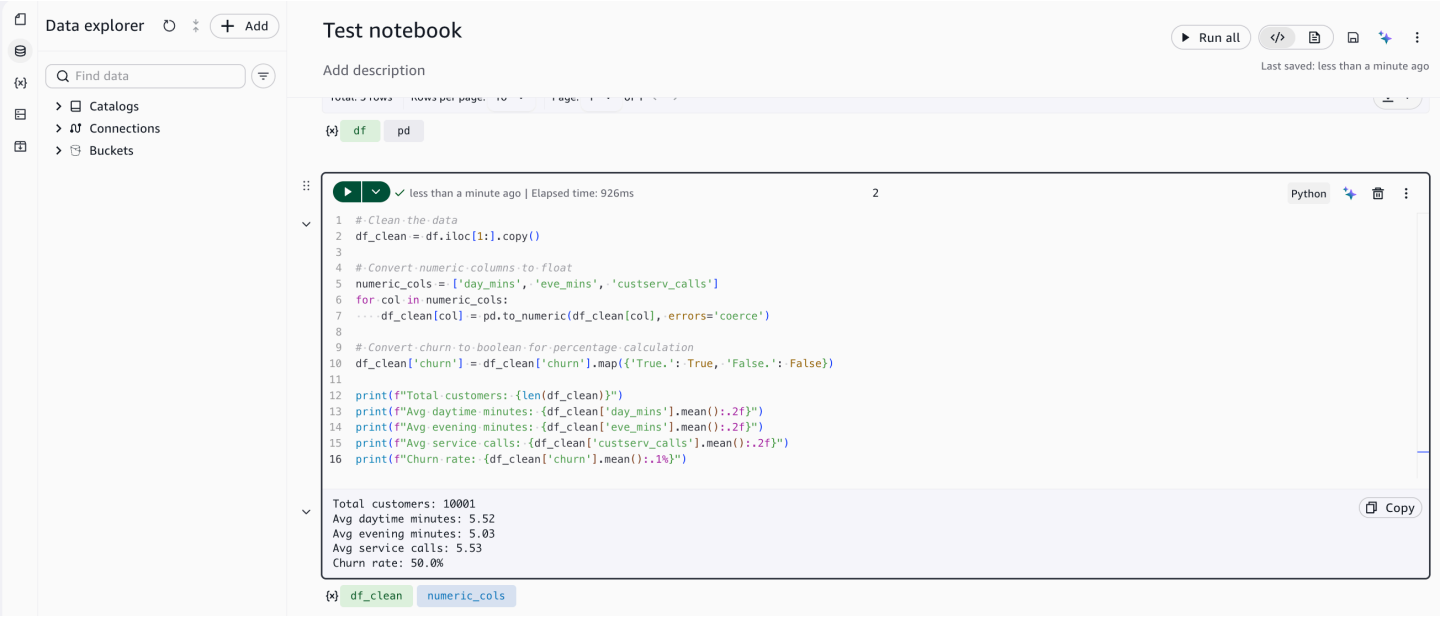
```
# Clean the data
df_clean = df.iloc[1:].copy()

# Convert numeric columns to float
numeric_cols = ['day_mins', 'eve_mins', 'custserv_calls']
for col in numeric_cols:
    df_clean[col] = pd.to_numeric(df_clean[col], errors='coerce')

# Convert churn to boolean for percentage calculation
df_clean['churn'] = df_clean['churn'].map({'True.': True, 'False.': False})

print(f"Total customers: {len(df_clean)}")
print(f"Avg daytime minutes: {df_clean['day_mins'].mean():.2f}")
print(f"Avg evening minutes: {df_clean['eve_mins'].mean():.2f}")
print(f"Avg service calls: {df_clean['custserv_calls'].mean():.2f}")
print(f"Churn rate: {df_clean['churn'].mean():.1%}")
```

This gives you a quick overview of the dataset: 10,001 customers with an average of 5.52 daytime minutes, 5.03 evening minutes, 5.53 service calls, and a 50% churn rate.



Step 4: Analyze patterns by state

Group the data by state to find which states have the most customer service calls. Add a new Python cell:

```
top_states = (
    df_clean.groupby('state')
    .agg(total_customers=('state', 'count'),
         avg_day_mins=('day_mins', 'mean'),
         avg_eve_mins=('eve_mins', 'mean'),
         avg_service_calls=('custserv_calls', 'mean'))
    .round(2)
    .sort_values('avg_service_calls', ascending=False)
    .head(10)
)
top_states
```

The results show the top 10 states sorted by average service calls, along with customer counts and usage patterns. This is the same analysis from the SQL tutorial, now done with pandas.



Step 5: Create a visualization

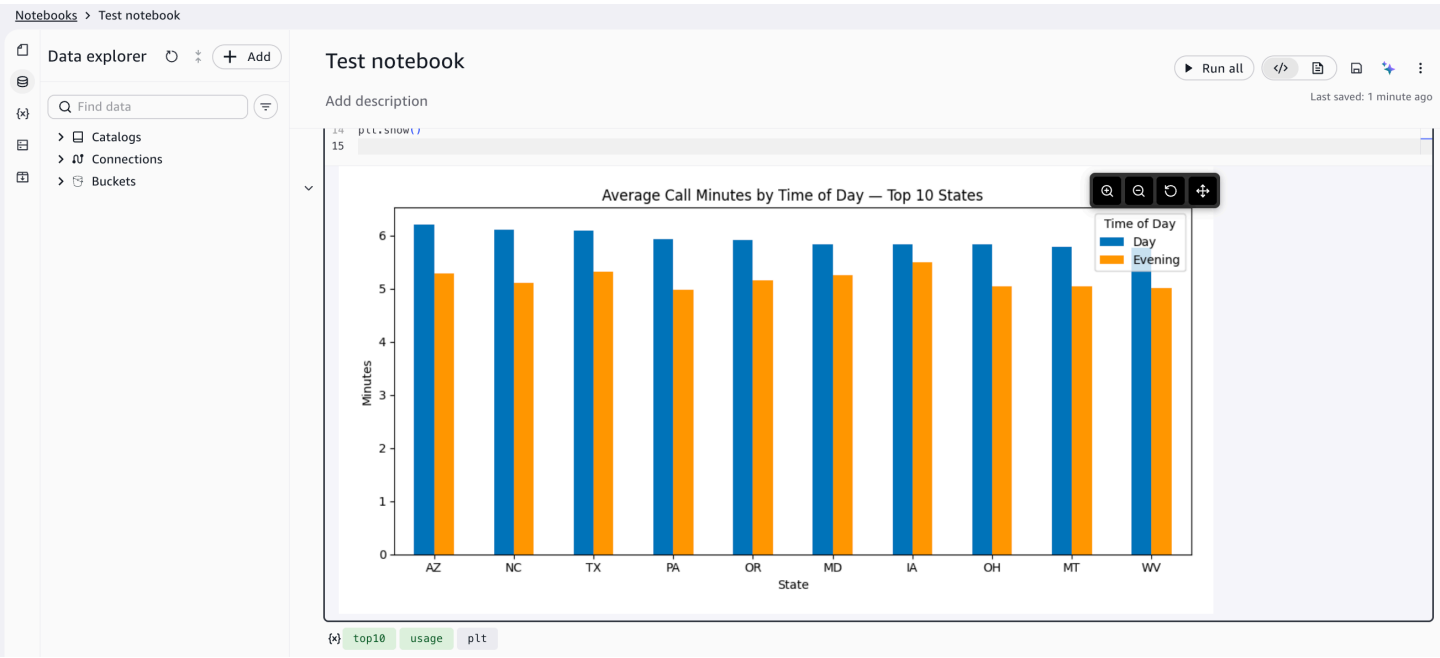
Tables show the numbers, but a chart makes it easier to spot patterns at a glance. Create a grouped bar chart that compares average daytime and evening call minutes across the top 10 states. Add a new Python cell:

```
import matplotlib.pyplot as plt

usage = df_clean.groupby('state')[['day_mins', 'eve_mins']].mean()
usage.columns = ['Day', 'Evening']
top10 = usage.sort_values('Day', ascending=False).head(10)

top10.plot(kind='bar', figsize=(10, 5), color=['#0073bb', '#ff9900'])
plt.title('Average Call Minutes by Time of Day \u2014 Top 10 States')
plt.ylabel('Minutes')
plt.xlabel('State')
plt.xticks(rotation=0)
plt.legend(title='Time of Day')
plt.tight_layout()
plt.show()
```

The chart renders inline, directly below the cell. Each state shows two bars comparing daytime and evening call minutes.

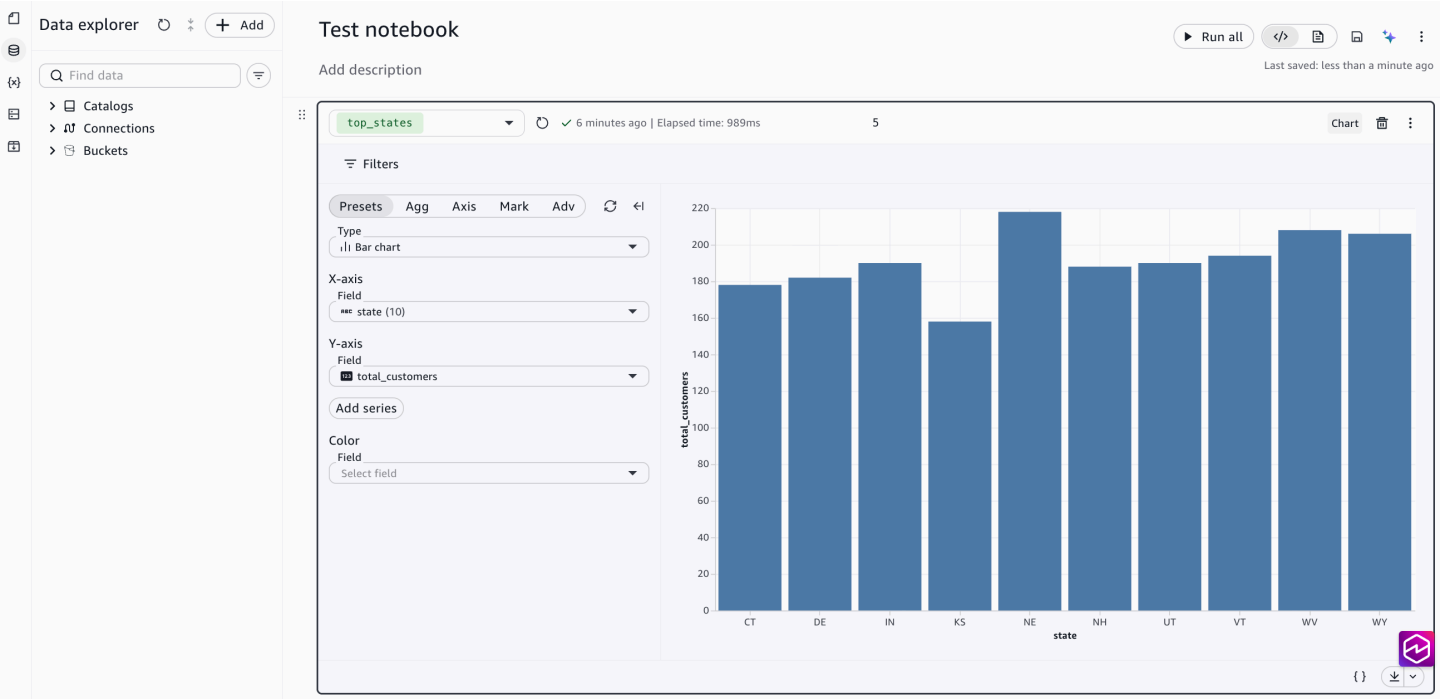


Create charts without code

Instead of writing plotting code, you can use the built-in **Charts** cell type. Choose **Charts** from the cell type options at the bottom of the notebook, then configure your chart visually:

1. For **Data frame**, select `top_states`.
2. For **Type**, choose **Bar chart**.
3. For **X-axis**, select **state**.
4. For **Y-axis**, select **total_customers**.

The chart updates automatically as you change the configuration.

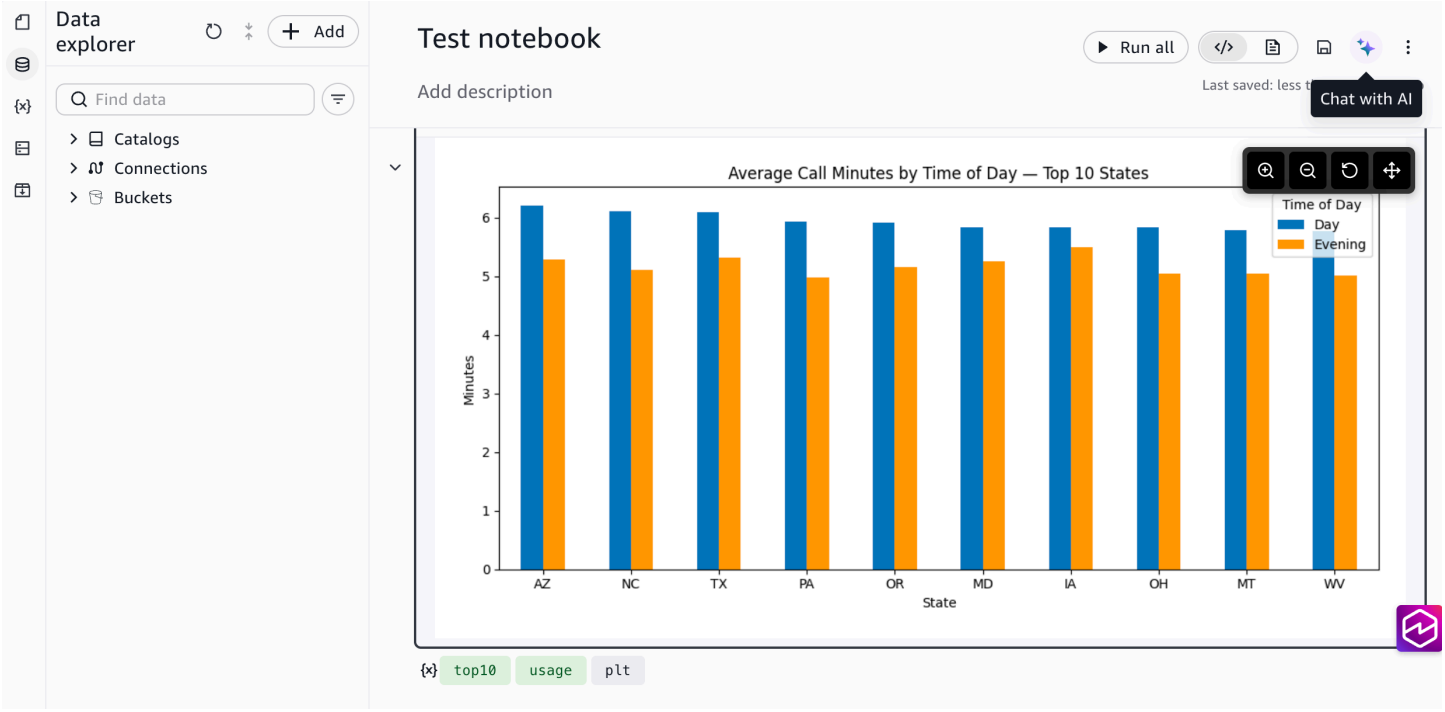


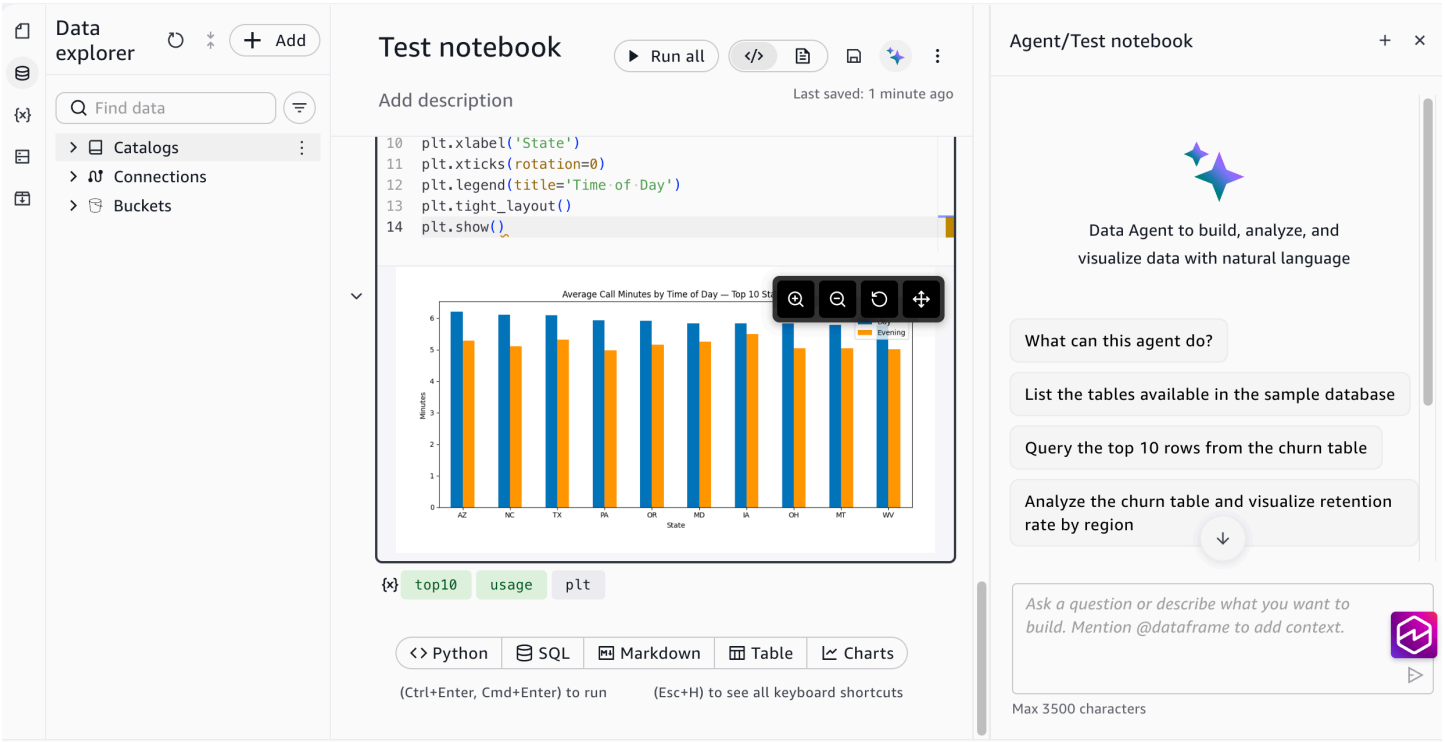
The chart shows that call patterns vary across states, which could inform regional support staffing or targeted retention campaigns. From here, you could extend this analysis by correlating call minutes with churn rates, segmenting customers by international plan usage, or building a predictive model to identify customers at risk of churning.

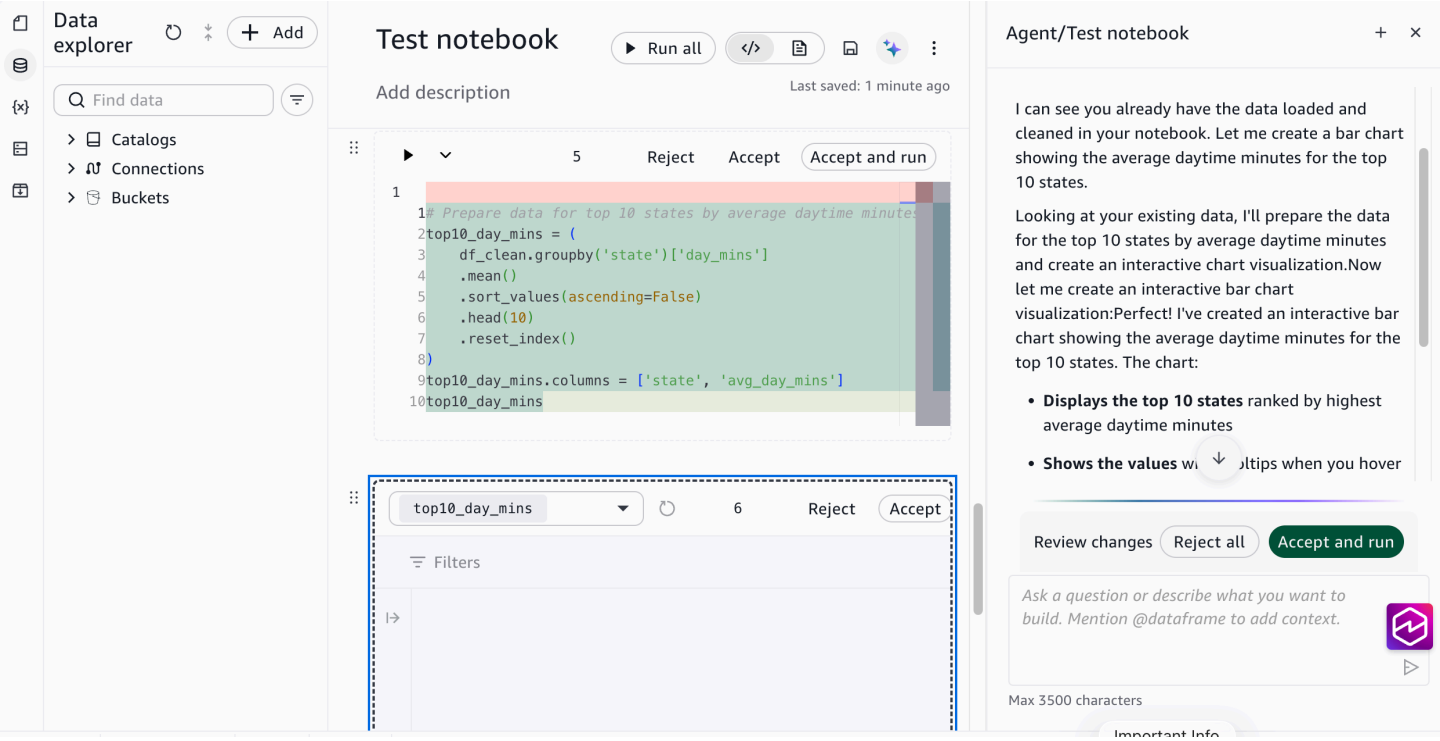
Use the Data Agent to generate code

Instead of writing code yourself, you can ask the **Data Agent** to generate it for you. The Data Agent can create transformations, aggregations, and visualizations from natural language descriptions.

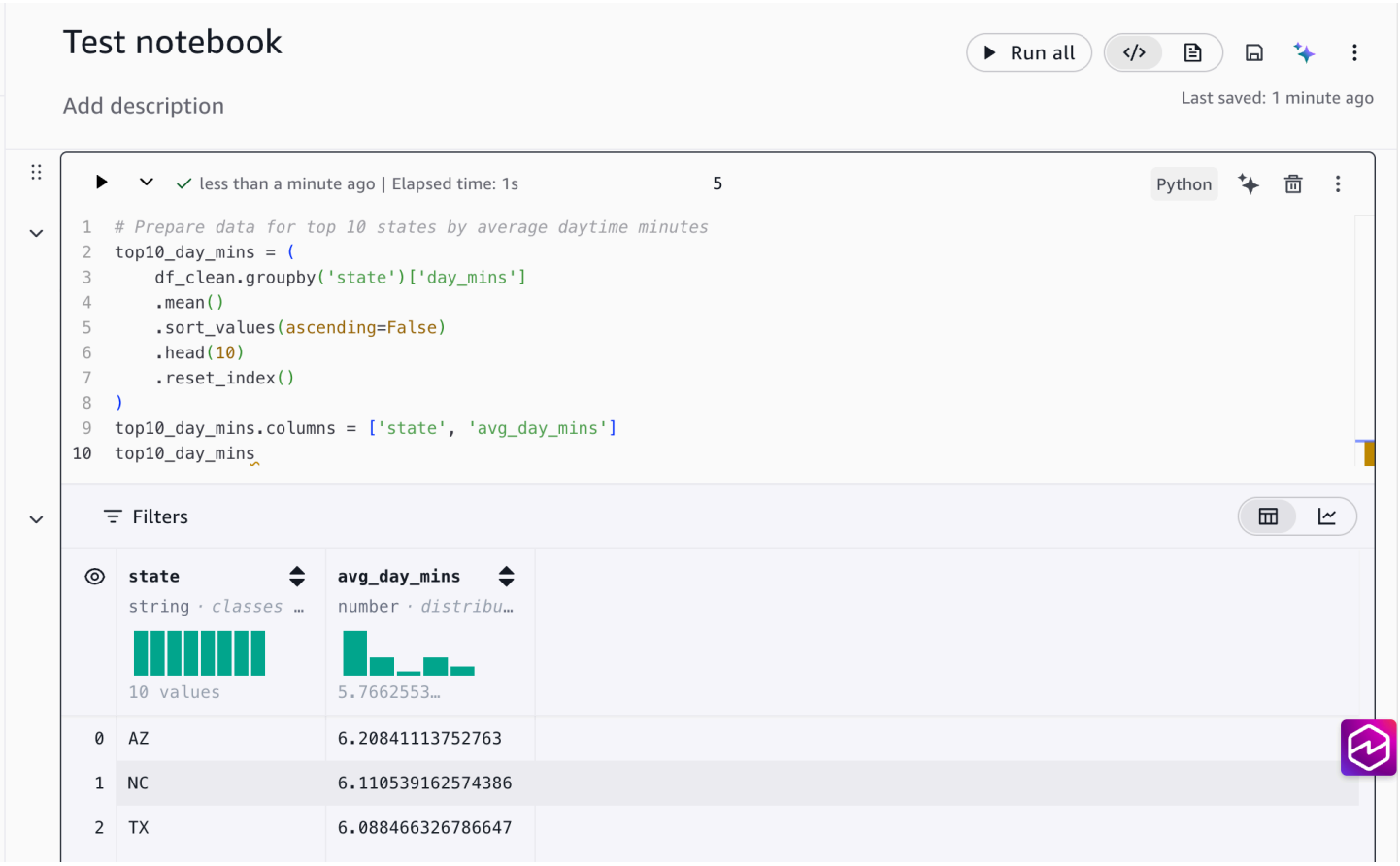
1. In the notebook, choose the **Chat with AI** icon in the top navigation bar.
2. In the **Ask a question** text box at the bottom of the panel, type what you want in plain English, for example: *"Create a bar chart showing average daytime minutes by state for the top 10 states"*


Data explorer
Add
Find data
Catalogs
Connections
Buckets
Test notebook
Add description
Run all
Chat with AI
Average Call Minutes by Time of Day — Top 10 States
Time of Day
Day
Evening
Minutes
State
AZ
NC
TX
PA
OR
MD
IA
OH
MT
WV
top10
usage
plt

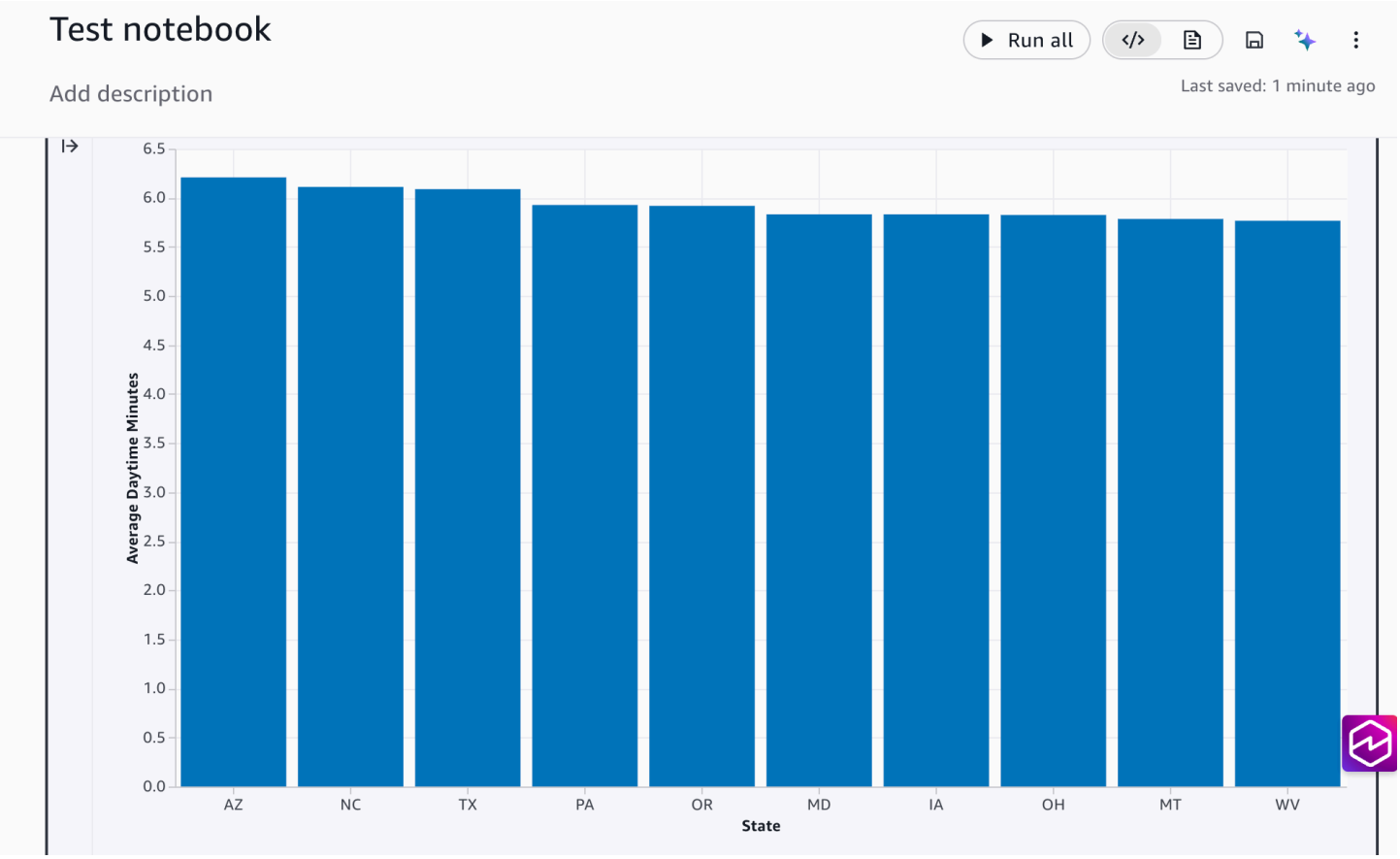

Data explorer
Add
Find data
Catalogs
Connections
Buckets
Test notebook
Run all
Add description
Last saved: 1 minute ago
plt.xlabel('State')
plt.xticks(rotation=0)
plt.legend(title='Time of Day')
plt.tight_layout()
plt.show()
top10
usage
plt
Python
SQL
Markdown
Table
Charts
(Ctrl+Enter, Cmd+Enter) to run
(Esc+H) to see all keyboard shortcuts
Agent/Test notebook
Data Agent to build, analyze, and visualize data with natural language
What can this agent do?
List the tables available in the sample database
Query the top 10 rows from the churn table
Analyze the churn table and visualize retention rate by region
Ask a question or describe what you want to build. Mention @dataframe to add context.
Max 3500 characters



The Data Agent generates the code for you. Review it, then run it.



The chart renders inline in the notebook.



What you learned

In this tutorial, you:

- Created a notebook and loaded lakehouse data into a pandas DataFrame
- Cleaned sample data and calculated summary statistics
- Grouped data by state to analyze customer service patterns
- Created a visualization with Python code and with the no-code Charts feature
- Used the Data Agent to generate Python code from natural language

Build a data pipeline with Visual ETL

Time: 10 minutes

Prerequisites: As a member of a SageMaker Unified Studio project, your IAM role needs the following managed policies:

- SageMakerStudioUserIAMConsolePolicy to sign in and access the project.
- SageMakerStudioUserIAMDefaultExecutionPolicy to access data and resources within the project.

If you don't have access, contact your administrator. If you are the administrator who set up the project, you already have the required permissions. Completing "Run your first SQL query" is helpful, but not required.

> **Note**
>
> You can use your identity-based permissions to create a pipeline. However, you need an IAM role to run the pipeline on a schedule.

Outcome: You create a visual ETL job that reads sample data, filters and reshapes it, and writes clean output to Amazon S3 without writing any code.

What you will do

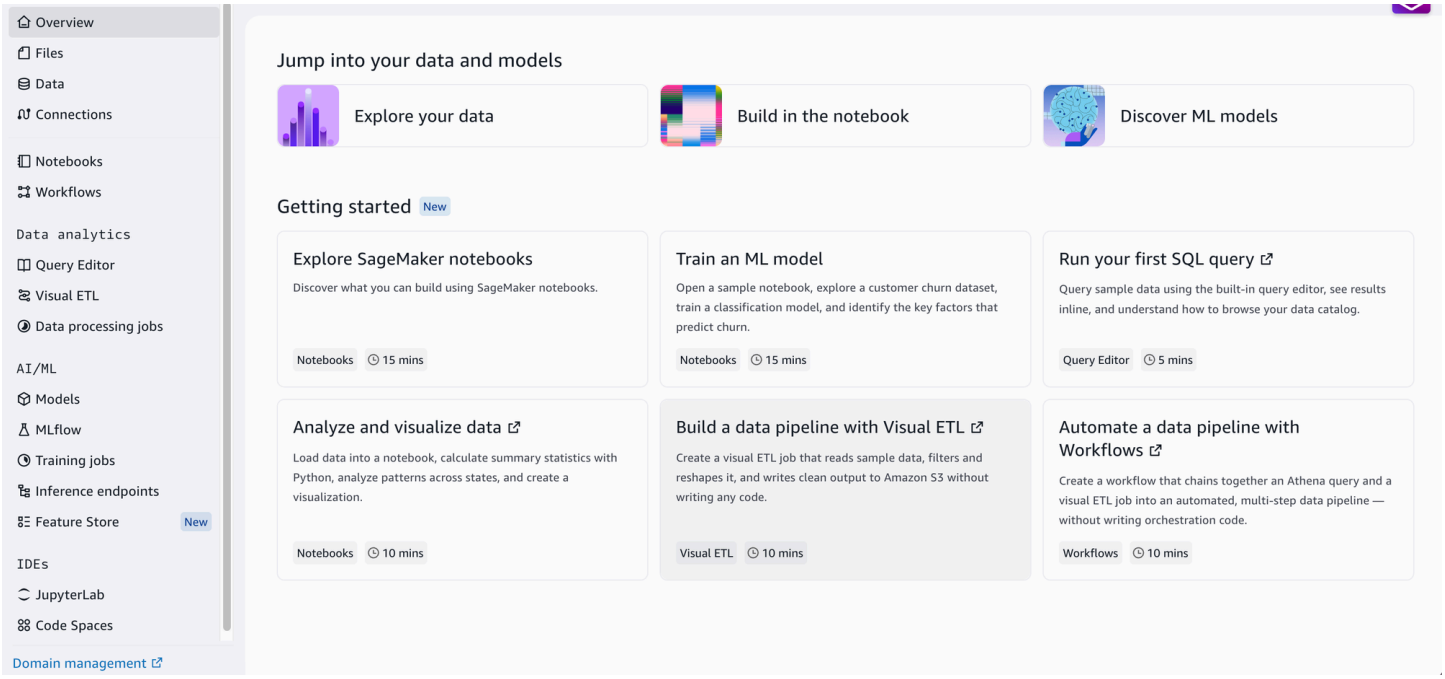
In this tutorial, you will:

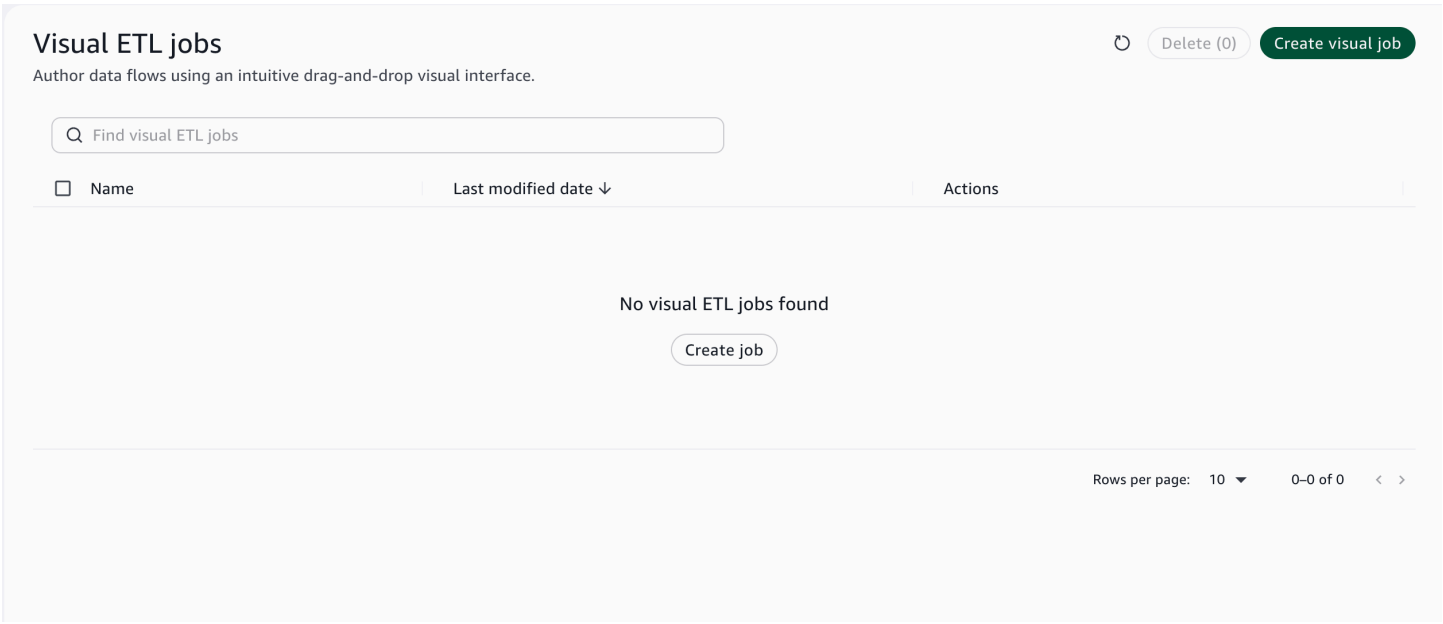
- Create a Visual ETL job in your project
- Add a data source node that reads from the AWS Glue Data Catalog
- Apply filter and select transformations to reshape the data
- Write the transformed output to Amazon S3
- Run the job and verify the results

ETL (Extract, Transform, Load) is how you prepare raw data for analysis. SageMaker Unified Studio provides a visual ETL editor where you build data pipelines by dragging and connecting nodes on a canvas. No code required. Under the hood, it runs on AWS Glue, a serverless data integration service, but you don't need to know Glue to use it.

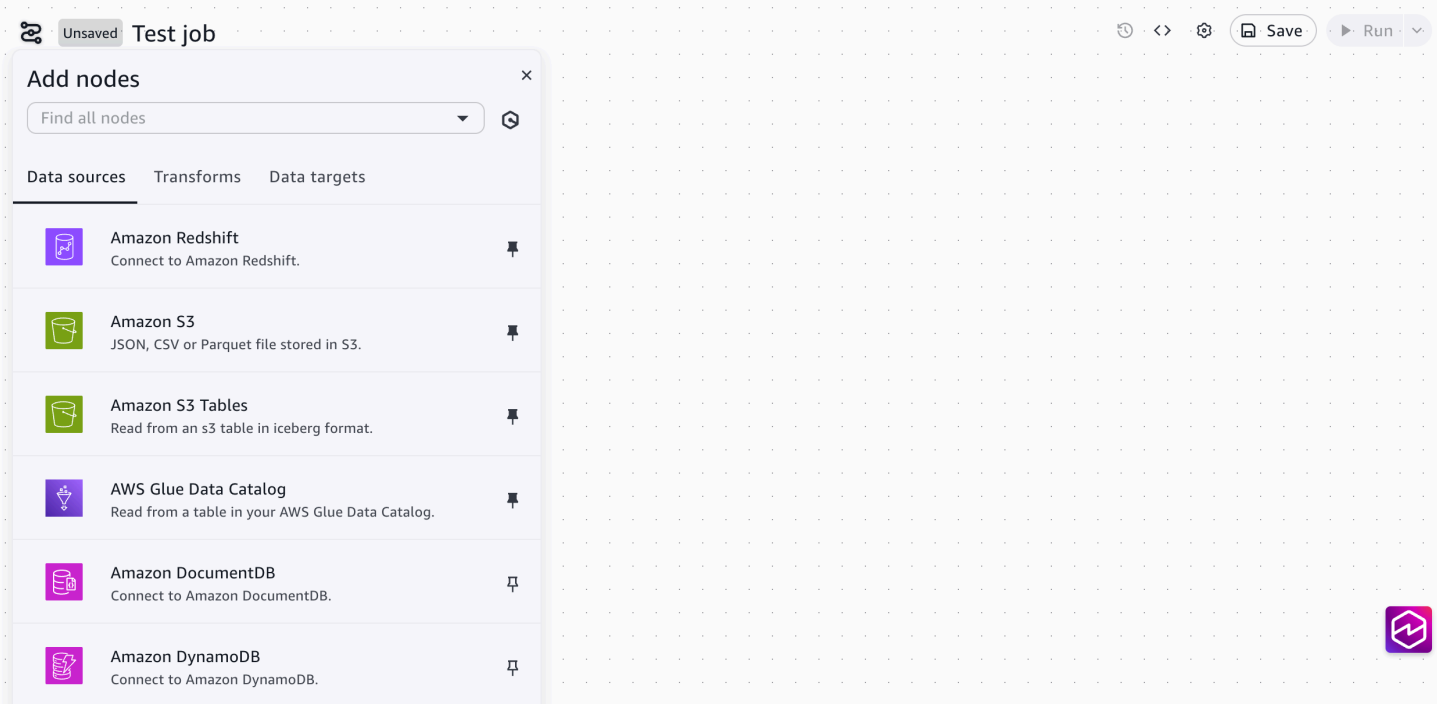
Step 1: Create a visual ETL job

1. Go to your project using the menu at the top of the page.
2. In the left navigation pane, choose **Visual ETL** under **Data analytics**.
3. Choose **Create visual job**.





The visual ETL canvas opens with an empty workspace.

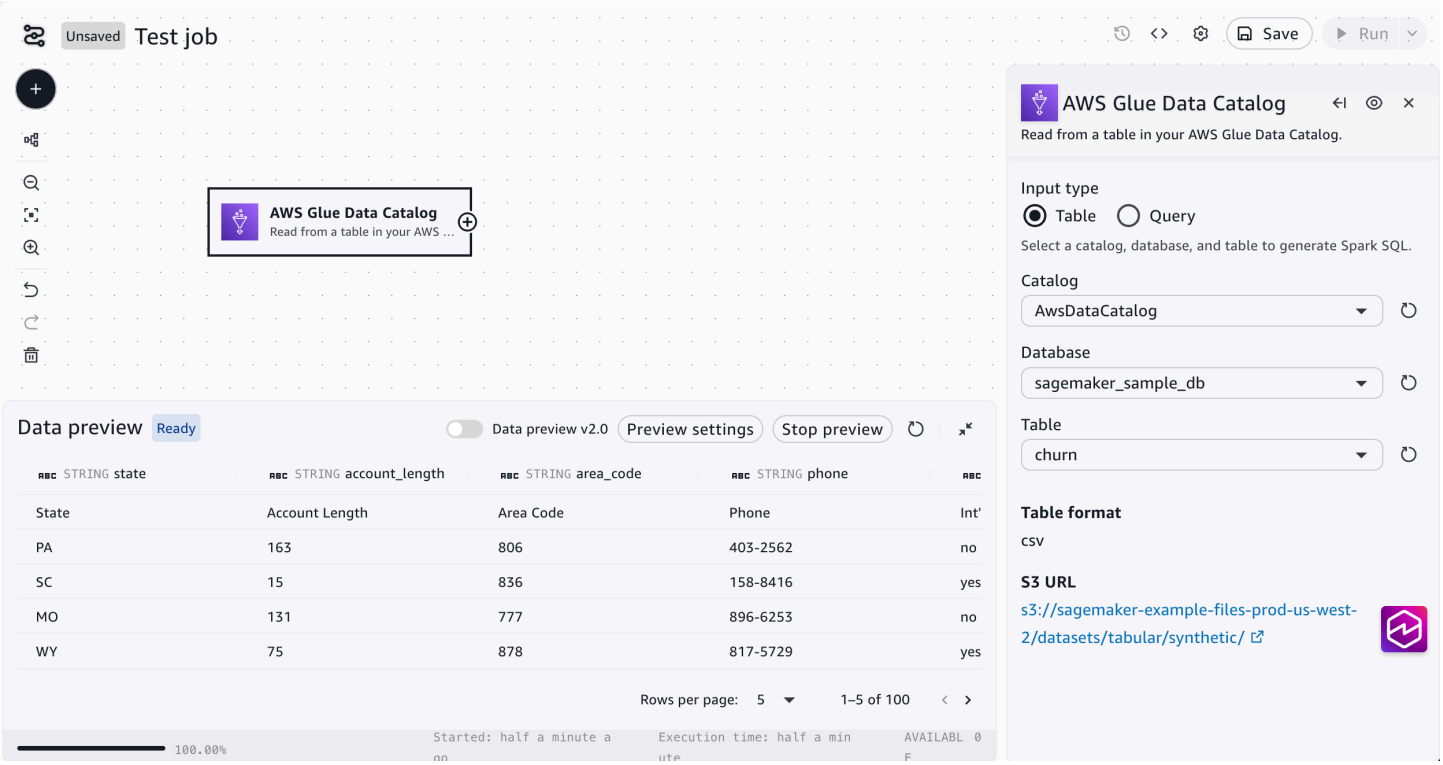


The canvas is where you design your pipeline. You add nodes for data sources, transformations, and targets, then connect them to define the data flow.

> **What is ETL?**
>
> ETL stands for Extract, Transform, Load. *Extract* reads data from a source. *Transform* cleans, filters, or reshapes it. *Load* writes the result to a destination. It's the standard pattern for preparing data before analysis or machine learning.

Step 2: Add a data source

1. On the canvas, choose the **Add Nodes** (+) button on the left. Under **Data sources**, choose **AWS Glue Data Catalog** and click on the canvas to place the node.
2. Choose the node to open its configuration panel.
3. For **Database**, choose `sagemaker_sample_db`.
4. For **Table**, choose `churn`.



Other data sources

You can also read from Amazon S3, Amazon Redshift, JDBC connections, and other sources. For this tutorial, you use the sample data that's already available in your project's data catalog.

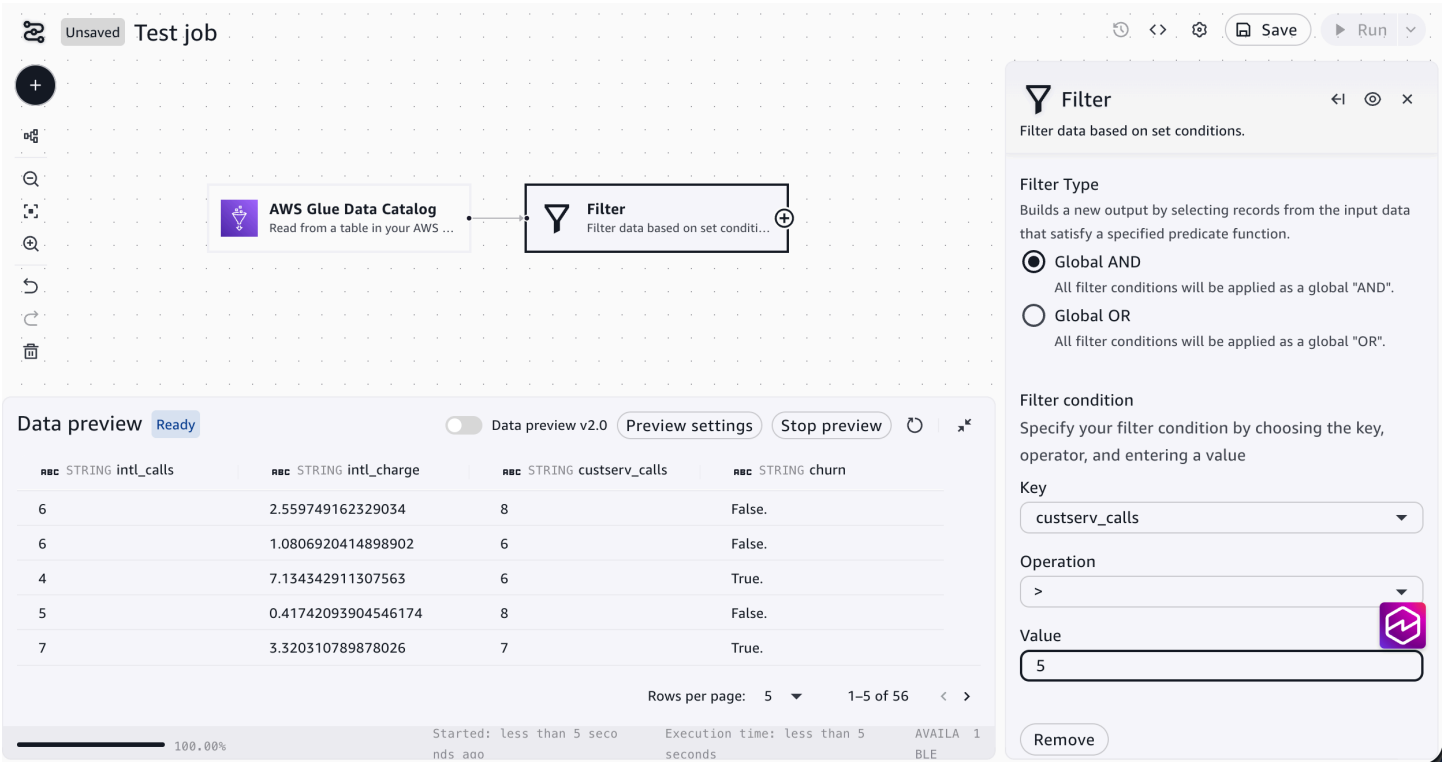
Step 3: Add transformations

Now clean and reshape the data. You add two transformation nodes.

Filter rows

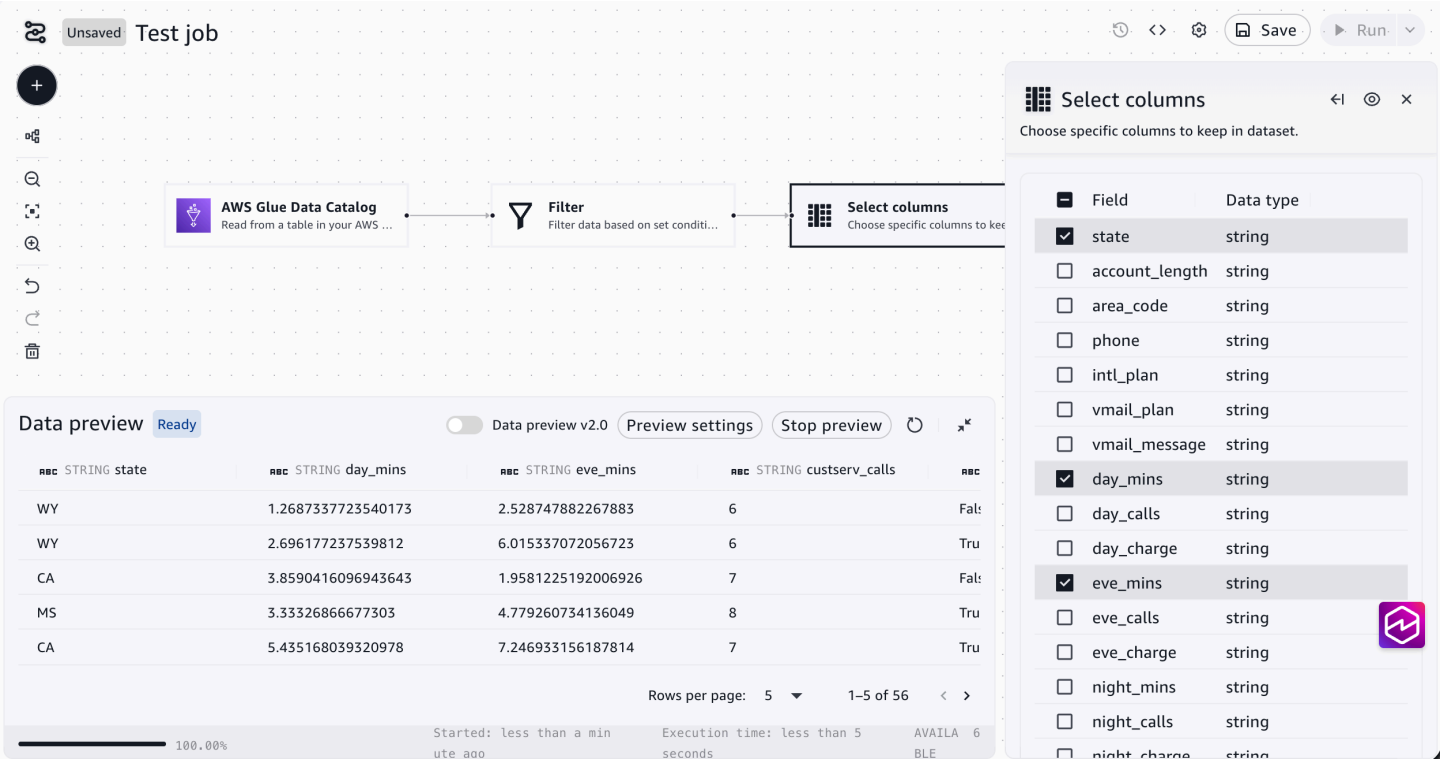
1. Choose the **+** icon on the right edge of the source node, or choose the **+** icon on the left of the canvas to **Add Nodes** and choose **Transforms**.
2. Choose **Filter**. A filter node appears on the canvas, connected to the source node.
3. Choose the filter node to open its configuration panel.
4. Set the filter condition: `custserv_calls > 5`. This keeps only customers who contacted customer service more than 5 times.

After you set the filter condition, the data preview updates to show only the rows that match.



Select columns

1. Choose the **+** icon on the right edge of the filter node, or choose the **+** icon on the left of the canvas to **Add Nodes** and choose **Transforms**.
2. Choose **Select Columns**. A select columns node appears on the canvas.
3. Choose the select columns node to open its configuration panel.
4. Choose the columns to keep: `state`, `day_mins`, `eve_mins`, `custserv_calls`, and `churn`.

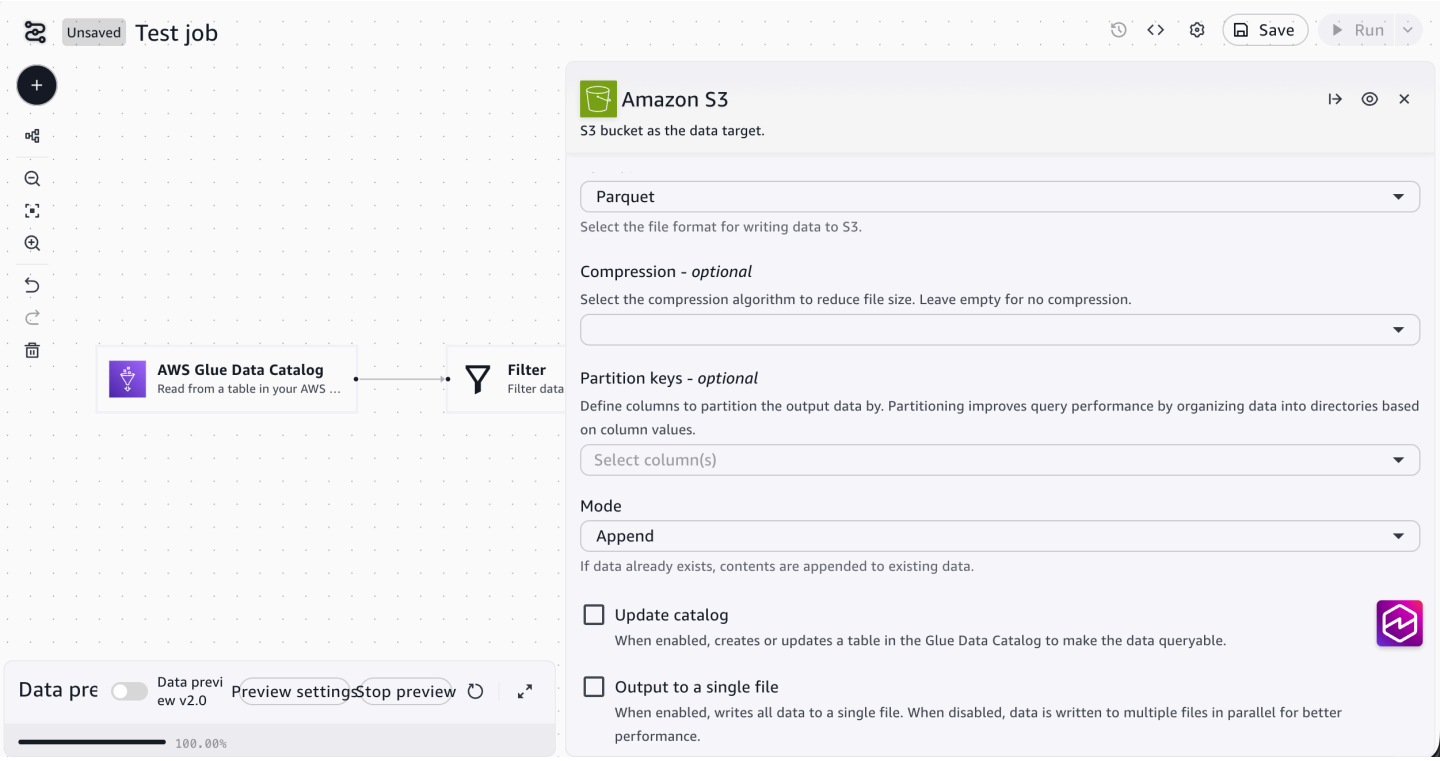


More transforms available

The visual ETL editor includes dozens of built-in transforms: joins, aggregations, derived columns, deduplication, and more. For example, if you noticed in the previous tutorial that the `churn` column contains `True.` instead of `True`, you could add a **Derived Column** transform to clean those values as part of your pipeline. You can also write custom transforms using SQL or Python.

Step 4: Add a target

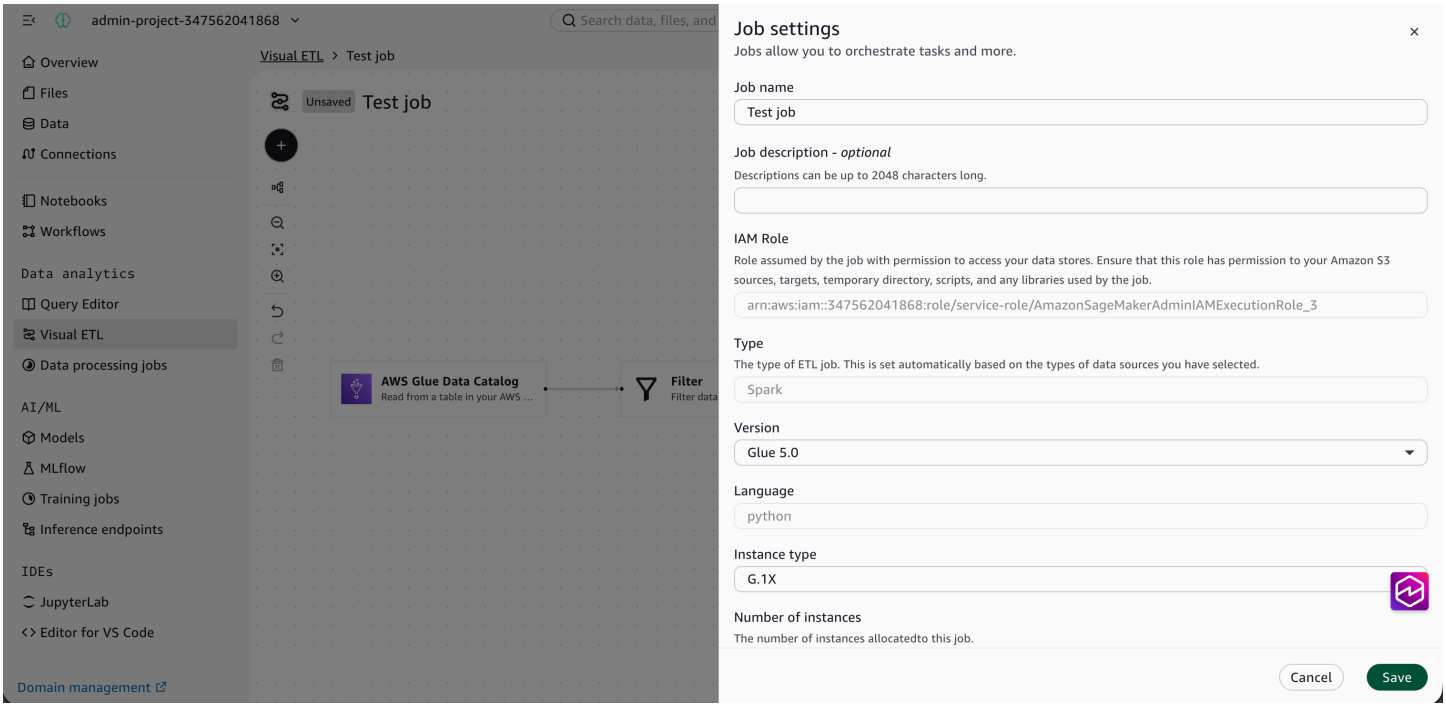
1. Choose the **+** icon on the right edge of the select columns node, or choose the **+** icon on the left of the canvas to **Add Nodes** and choose **Data targets**.
2. Choose **Amazon S3**. A target node appears on the canvas.
3. Choose the target node to open its configuration panel.
4. For **Format**, choose **Parquet**.
5. For **S3 Target Location**, choose **Browse S3**, select your project's S3 bucket, select the `shared/` folder, and then add `filtered-churn/` at the end of the path in the S3 URI field.

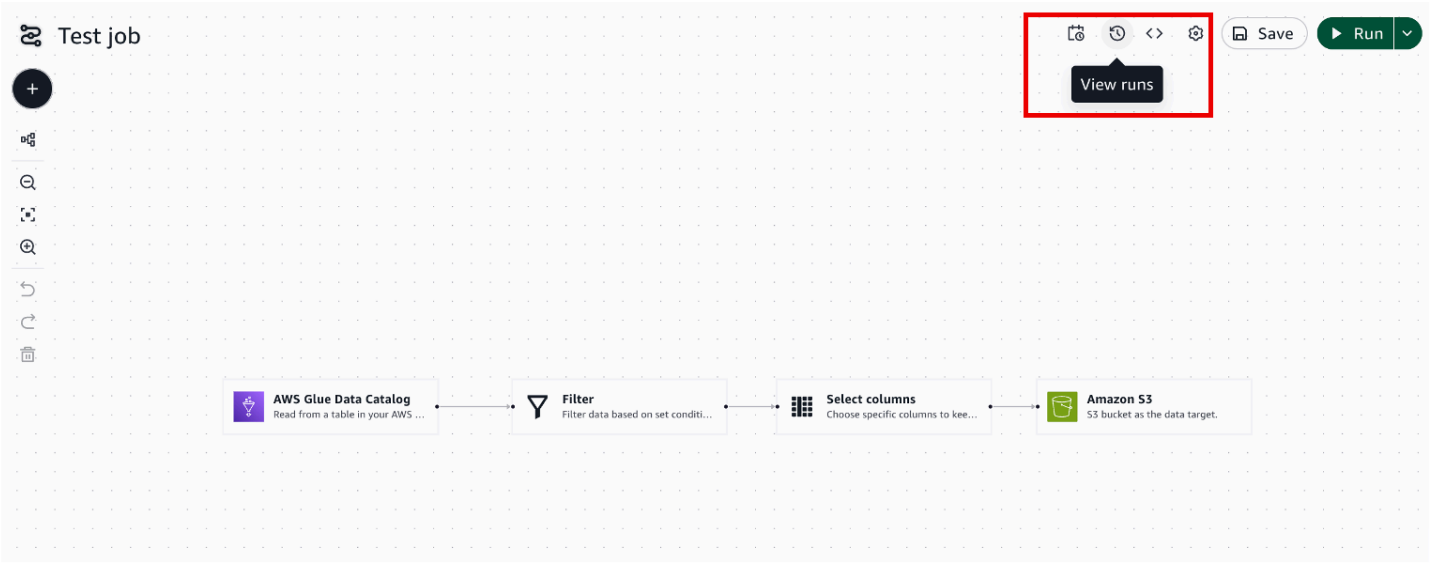


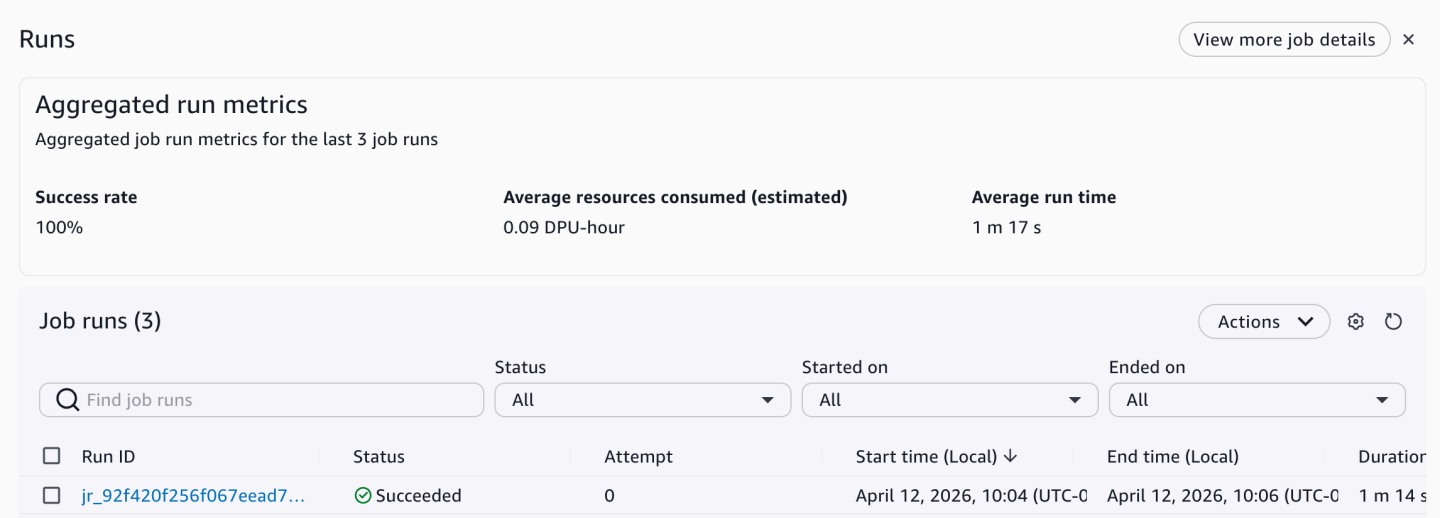
Your pipeline now reads sample data from the AWS Glue Data Catalog, filters for customers with more than 5 service calls, selects specific columns, and writes the result to Amazon S3.

Step 5: Save and run the job

1. Enter a name for your job in the title field at the top of the canvas.
2. Choose **Save** to save your job.
3. Choose **Run**.
4. Choose the **View runs** tab at the top of the canvas to see the job progress. A job on this sample data typically completes in 2 to 3 minutes.


admin-project-347562041868
Search data, files, and
Overview
Files
Data
Connections
Notebooks
Workflows
Data analytics
Query Editor
Visual ETL
Data processing jobs
AI/ML
Models
MLflow
Training jobs
Inference endpoints
IDEs
JupyterLab
Editor for VS Code
Domain management
Visual ETL > Test job
Unsaved
Test job
AWS Glue Data Catalog
Read from a table in your AWS ...
Filter
Filter data
Job settings
Jobs allow you to orchestrate tasks and more.
Job name
Test job
Job description - optional
Descriptions can be up to 2048 characters long.
IAM Role
Role assumed by the job with permission to access your data stores. Ensure that this role has permission to your Amazon S3 sources, targets, temporary directory, scripts, and any libraries used by the job.
arn:aws:iam::347562041868:role/service-role/AmazonSageMakerAdminIAMExecutionRole_3
Type
The type of ETL job. This is set automatically based on the types of data sources you have selected.
Spark
Version
Glue 5.0
Language
python
Instance type
G.1X
Number of instances
The number of instances allocatedto this job.
Cancel
Save


Test job
View runs
Save
Run
AWS Glue Data Catalog
Read from a table in your AWS ...
Filter
Filter data based on set conditi...
Select columns
Choose specific columns to kee...
Amazon S3
S3 bucket as the data target.

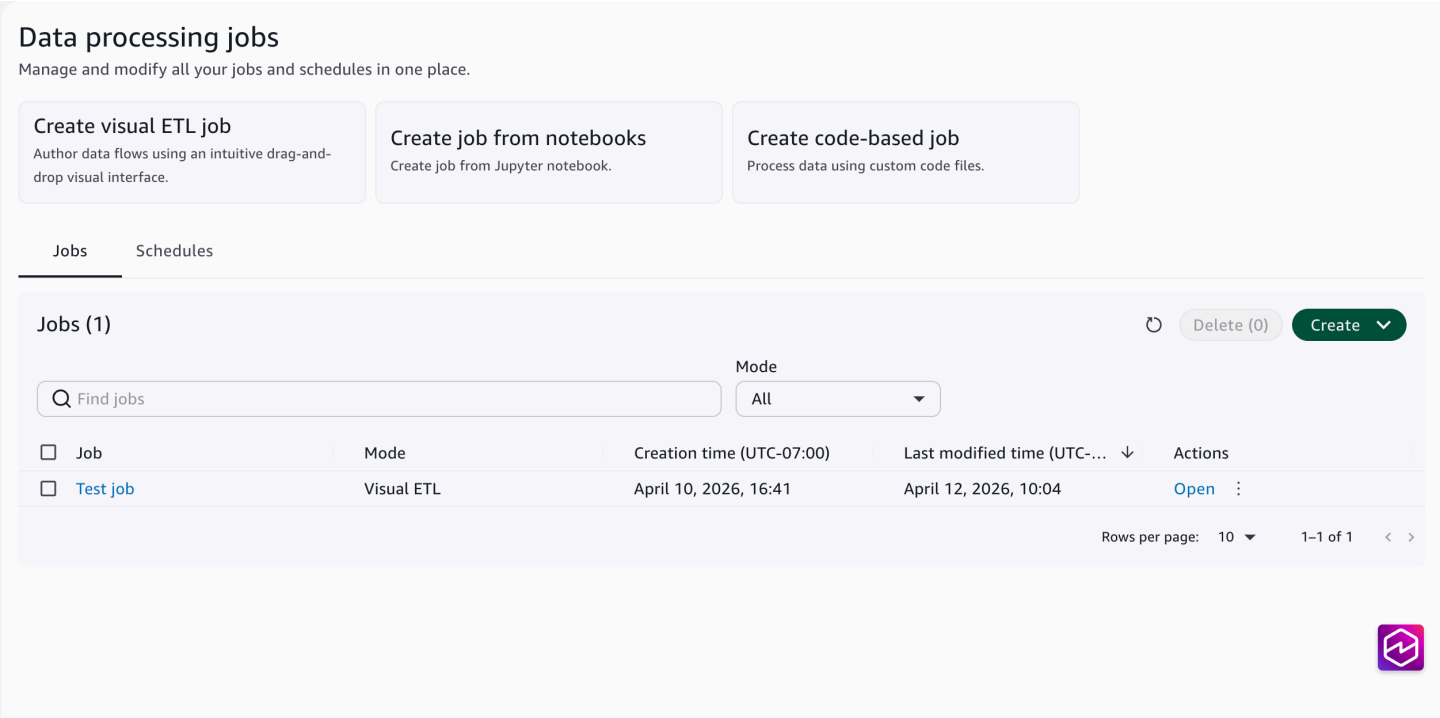


Scheduling

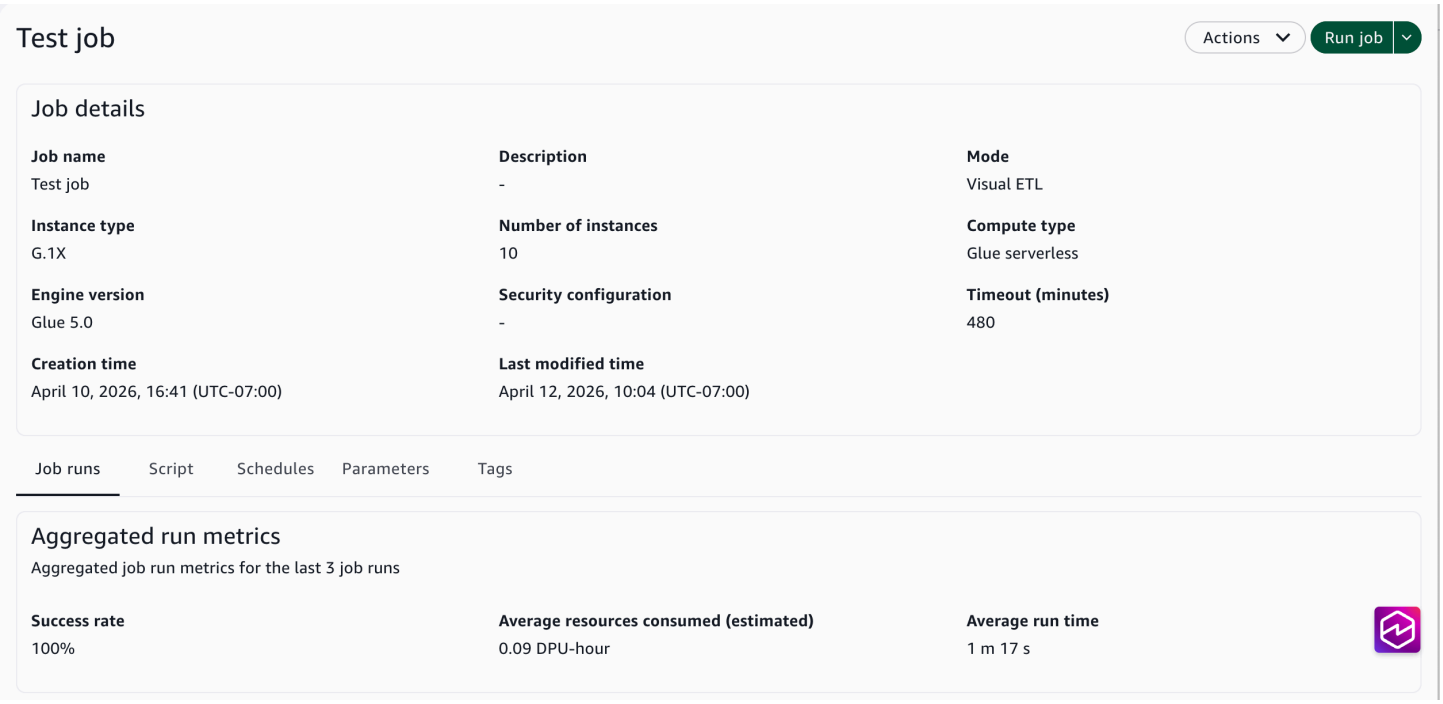
In production, you'd schedule this job to run on a recurring basis (hourly, daily, or triggered by new data arriving). You can set up schedules directly from the job configuration.

Step 6: Verify the output

After the job completes, you can find it listed under **Data processing jobs** in the left navigation pane.



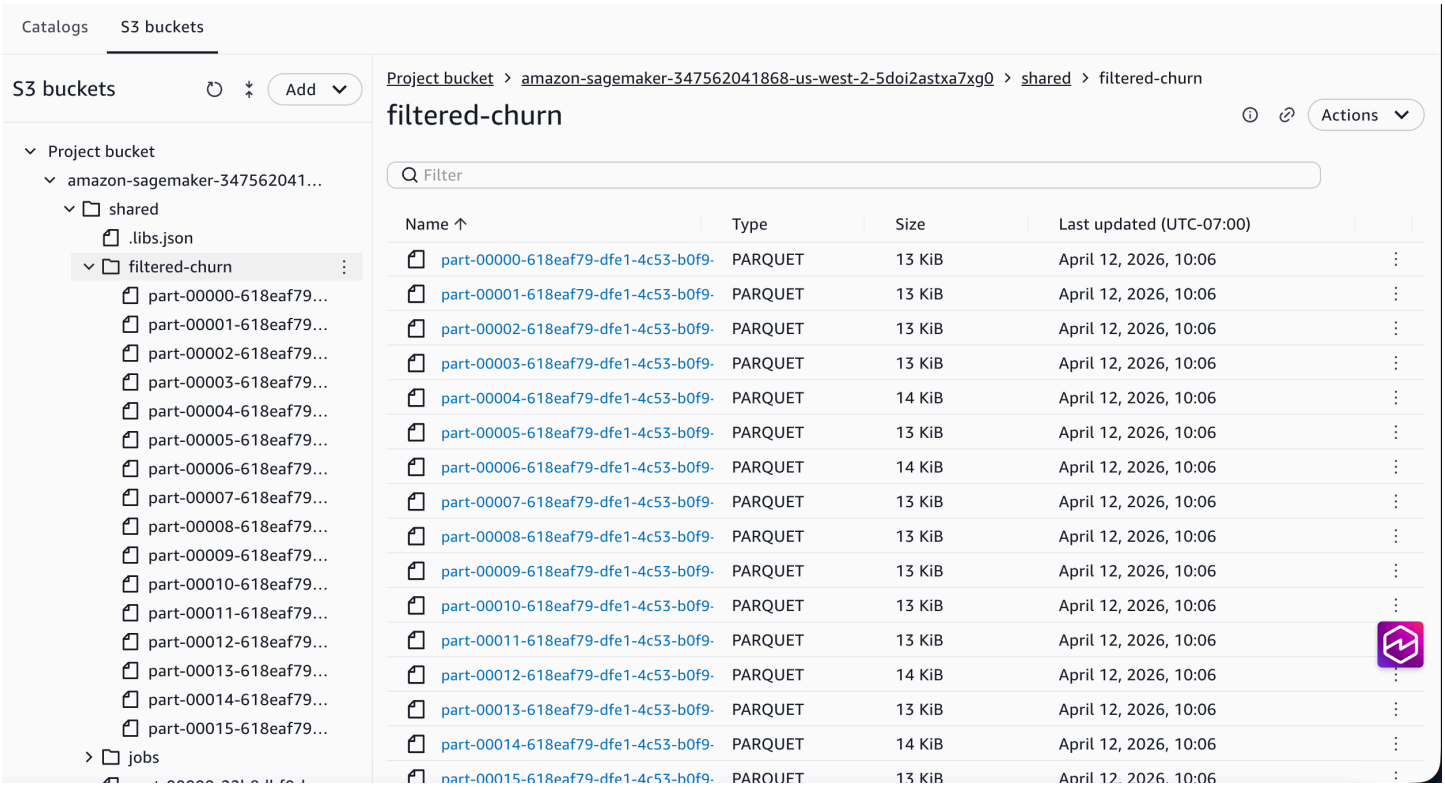
Choose the job to view its details, including run history, status, and configuration.



To verify the output:

1. In the left navigation pane, choose **Data**.
2. Under **S3 buckets**, expand your project's bucket.

3. Navigate to the output folder you specified in the target node (for example, `shared/filtered-churn/`).
4. You should see Parquet files containing only the filtered rows and selected columns.



Viewing the output data

Parquet is a compressed columnar format optimized for analytics. To view the contents of these files, you can create an external table pointing to the S3 location and query it using the query editor, or load the files into a notebook using pandas or Spark.

You now have a clean, filtered dataset in Amazon S3. This prepared data is ready for downstream use: analysts can query it directly with SQL, data scientists can load it into a notebook for deeper analysis, or it can serve as input for machine learning training.

What you learned

In this tutorial, you:

- Created a visual ETL job without writing code

- Connected an AWS Glue Data Catalog source to read sample data
- Applied filter and select column transformations to reshape the data
- Wrote transformed data to Amazon S3
- Ran the job and verified the output in the Data explorer

Automate a data pipeline with Workflows

Time: 10 minutes

Prerequisites: As a member of a SageMaker Unified Studio project, your IAM role needs the following managed policies:

- SageMakerStudioUserIAMConsolePolicy to sign in and access the project.
- SageMakerStudioUserIAMDefaultExecutionPolicy to access data and resources within the project.

If you don't have access, contact your administrator. If you are the administrator who set up the project, you already have the required permissions.

> **Important**
>
> Before you begin, complete Build a data pipeline with Visual ETL. You need the visual ETL job you created in that tutorial.

Outcome: You create a workflow that chains together an Athena query and a visual ETL job into an automated, multi-step data pipeline — without writing orchestration code.

What you will do

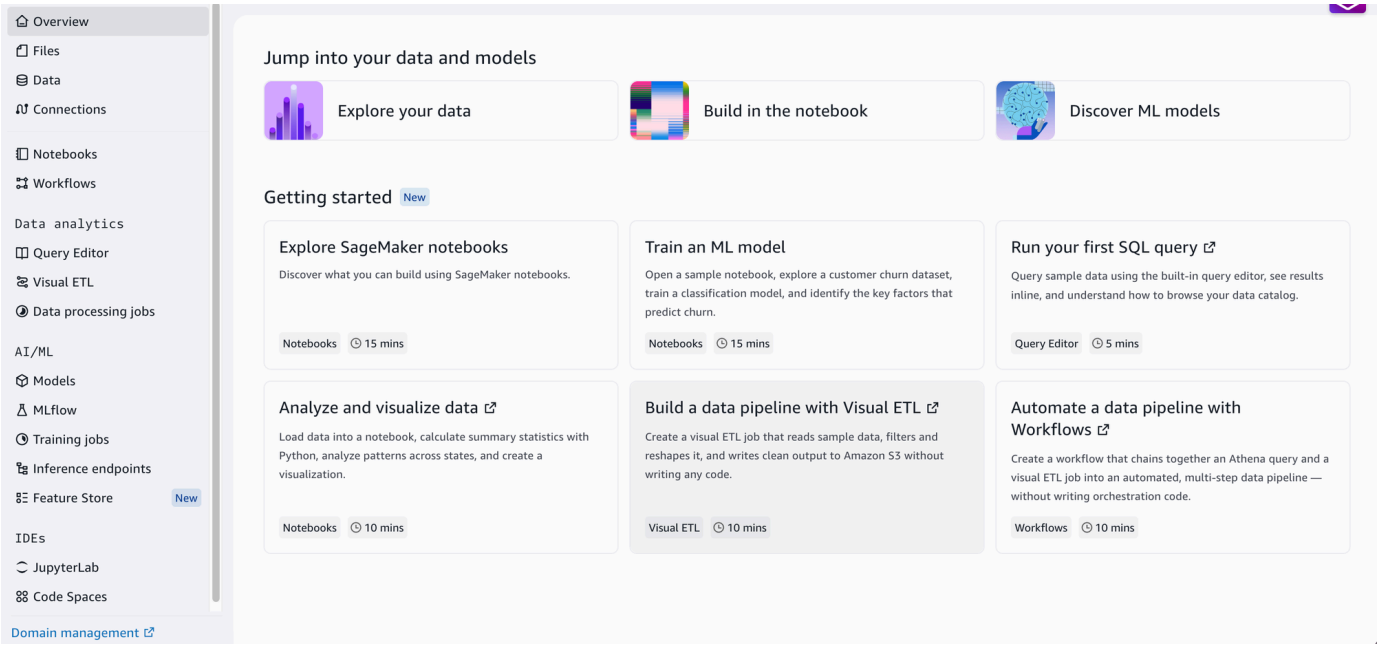
In this tutorial, you will:

- Create a workflow in your project
- Add an Athena operator that queries the sample churn table
- Add a visual ETL job task that runs the pipeline you built in the previous tutorial
- Run the workflow and verify that all tasks complete successfully

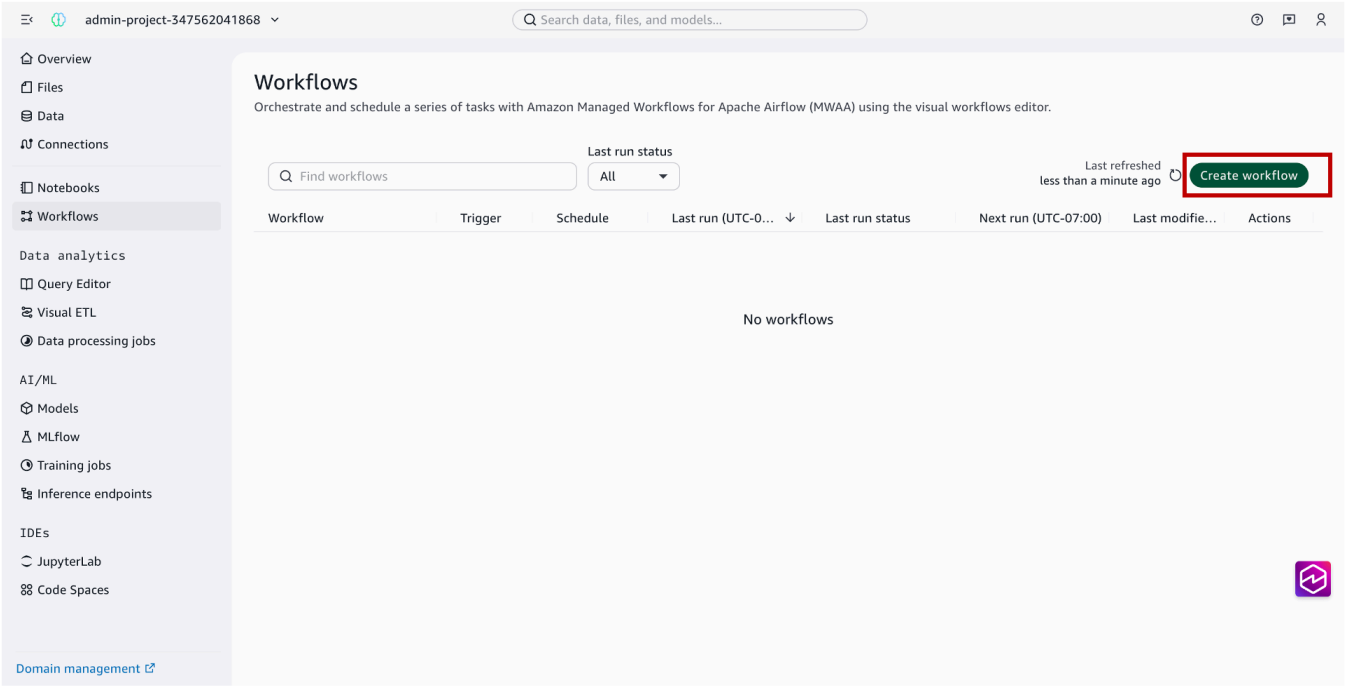
A workflow lets you chain multiple tasks — queries, ETL jobs, notebooks — into a single automated pipeline. Each task runs in order, so downstream steps can depend on the output of earlier ones. Workflows run on Amazon Managed Workflows for Apache Airflow (Amazon MWAA), but you don't need to know Airflow to use the visual workflow editor.

Step 1: Create a workflow

1. Go to your project using the menu at the top of the page.
2. In the left navigation pane, choose **Workflows**.



3. Choose **Create workflow**.

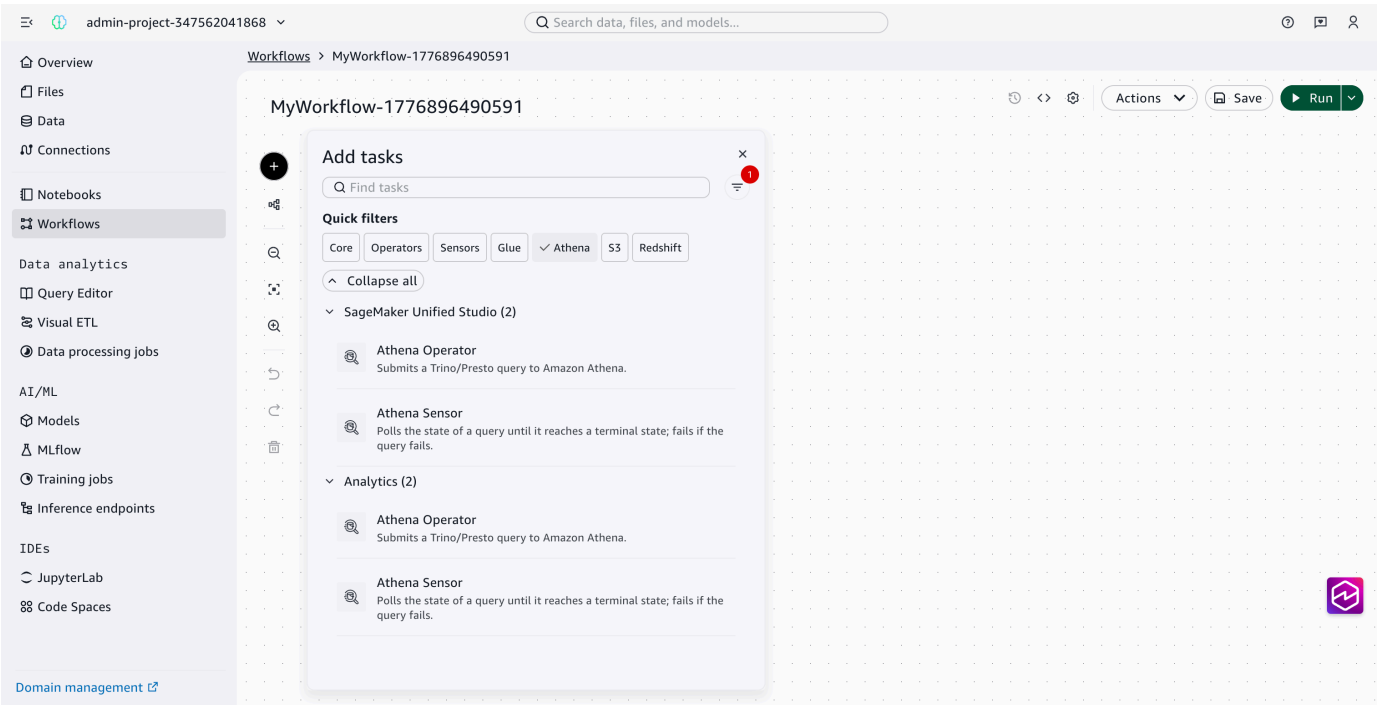


The workflow editor opens with an empty canvas and an **Add tasks** panel on the left. This is where you define the sequence of tasks that make up your pipeline.

Step 2: Add an Athena query task

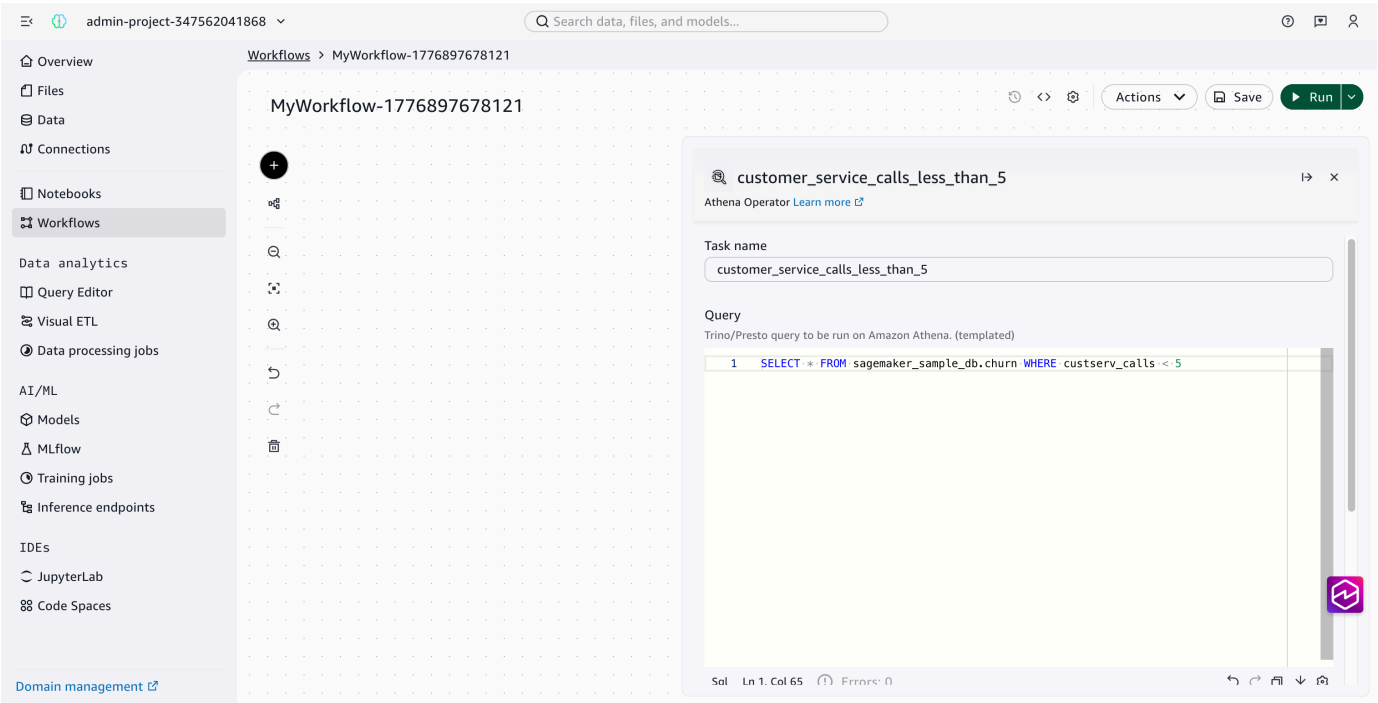
The first task in the workflow checks the original churn table for rows where `custserv_calls` is less than 5. This gives you a baseline before the ETL job runs.

1. In the **Add tasks** panel, use the **Athena** quick filter to narrow the task list.
2. Under **SageMaker Unified Studio**, choose **Athena Operator**.



3. Choose the task node on the canvas to open its configuration panel.
4. For **Task name**, enter `customer_service_calls_less_than_5`.
5. In the **Query editor**, enter the following SQL:

```
SELECT * FROM sagemaker_sample_db.churn WHERE custserv_calls < 5
```



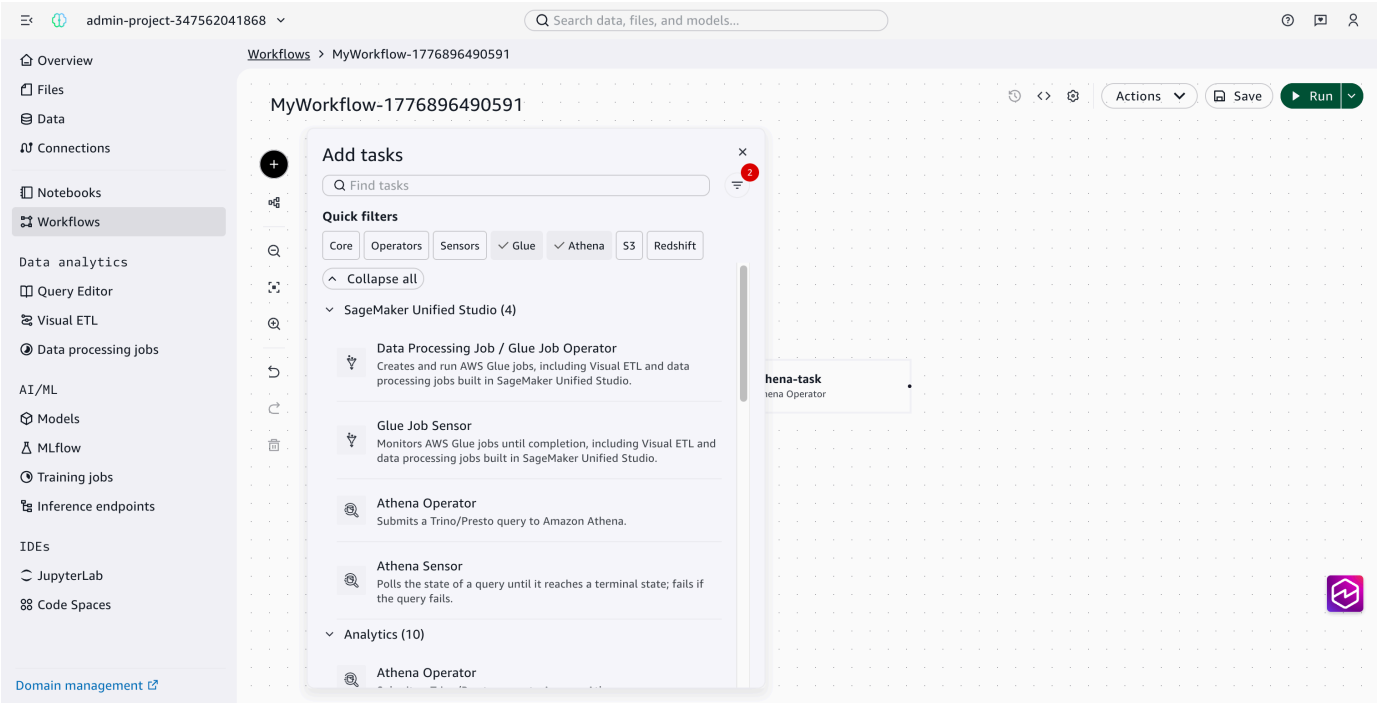
Why this query?

In the visual ETL tutorial, you filtered the churn table to keep only rows where `custserv_calls > 5`. This Athena query checks the complement — rows where `custserv_calls < 5` — so you can confirm both halves of the data are accounted for.

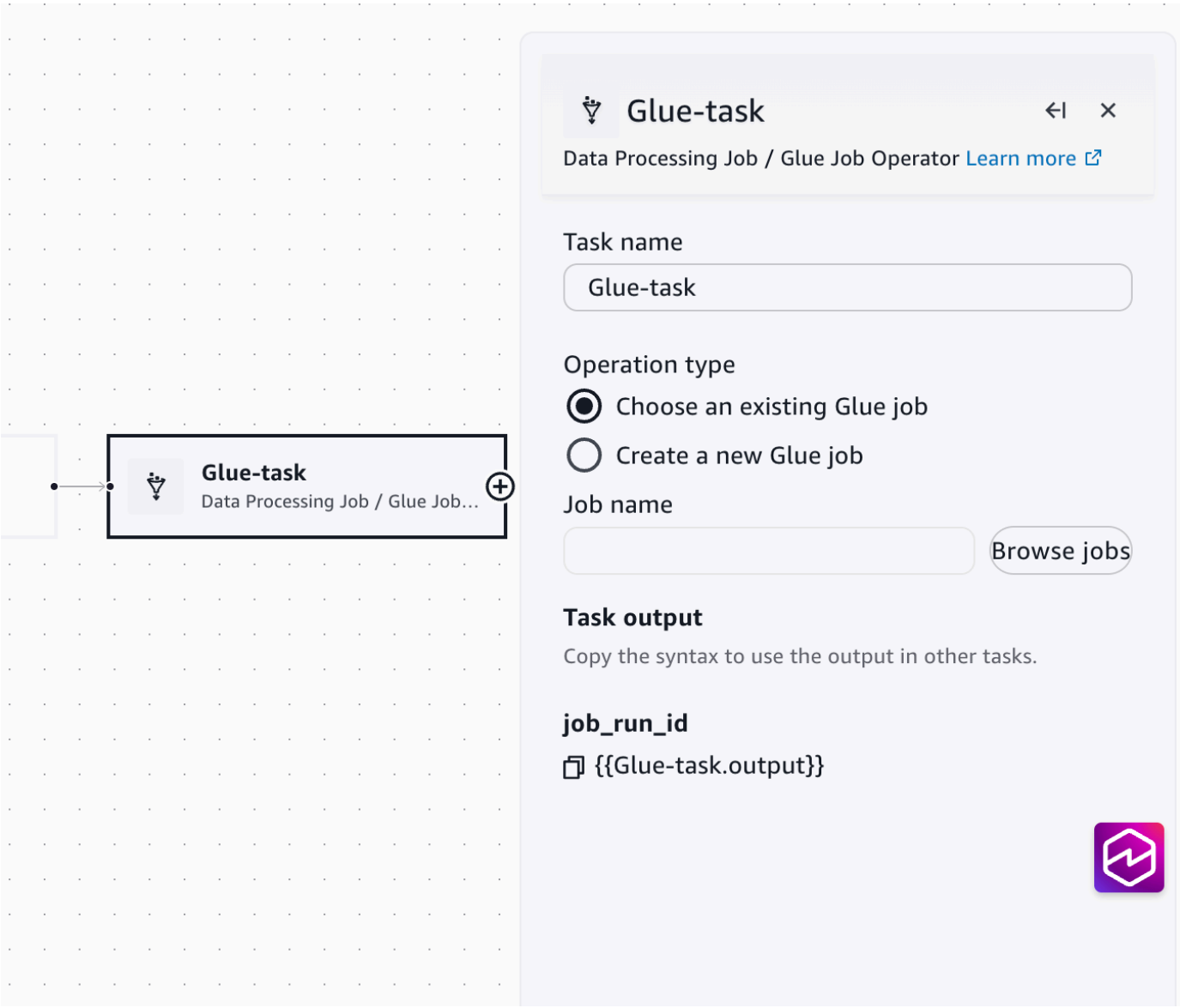
Step 3: Add a visual ETL job task

Next, add the visual ETL job you created in the previous tutorial as the second task in the workflow.

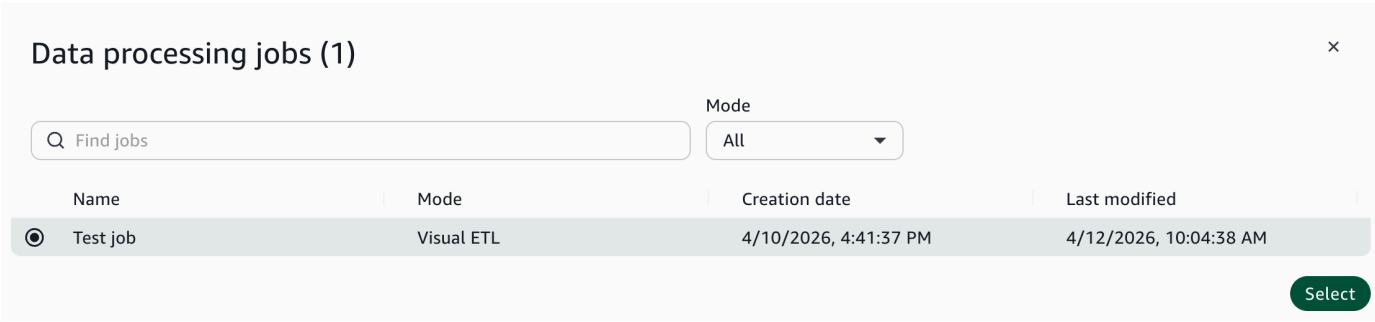
1. Choose the **+** button to open the **Add tasks** panel again.
2. Use the **Glue** quick filter to narrow the task list.
3. Under **SageMaker Unified Studio**, choose **Data Processing Job / Glue Job Operator**.



4. Choose the task node on the canvas to open its configuration panel.



5. For **Task name**, enter `processing_job`.
6. In the configuration panel, choose **Use existing job**.
7. Choose **Browse jobs** and select the visual ETL job you created in the [Build a data pipeline with Visual ETL] tutorial.



Create or reuse

You can either create a new job directly within the workflow or reuse an existing one. Reusing a job is a good practice when you've already tested and validated a pipeline — it avoids duplication and keeps your project organized.

- Connect the two tasks by dragging from the output handle (right dot) of the Athena task to the input handle (left dot) of the Glue task.



Your workflow canvas now shows two tasks connected in sequence: Athena query → Visual ETL job.

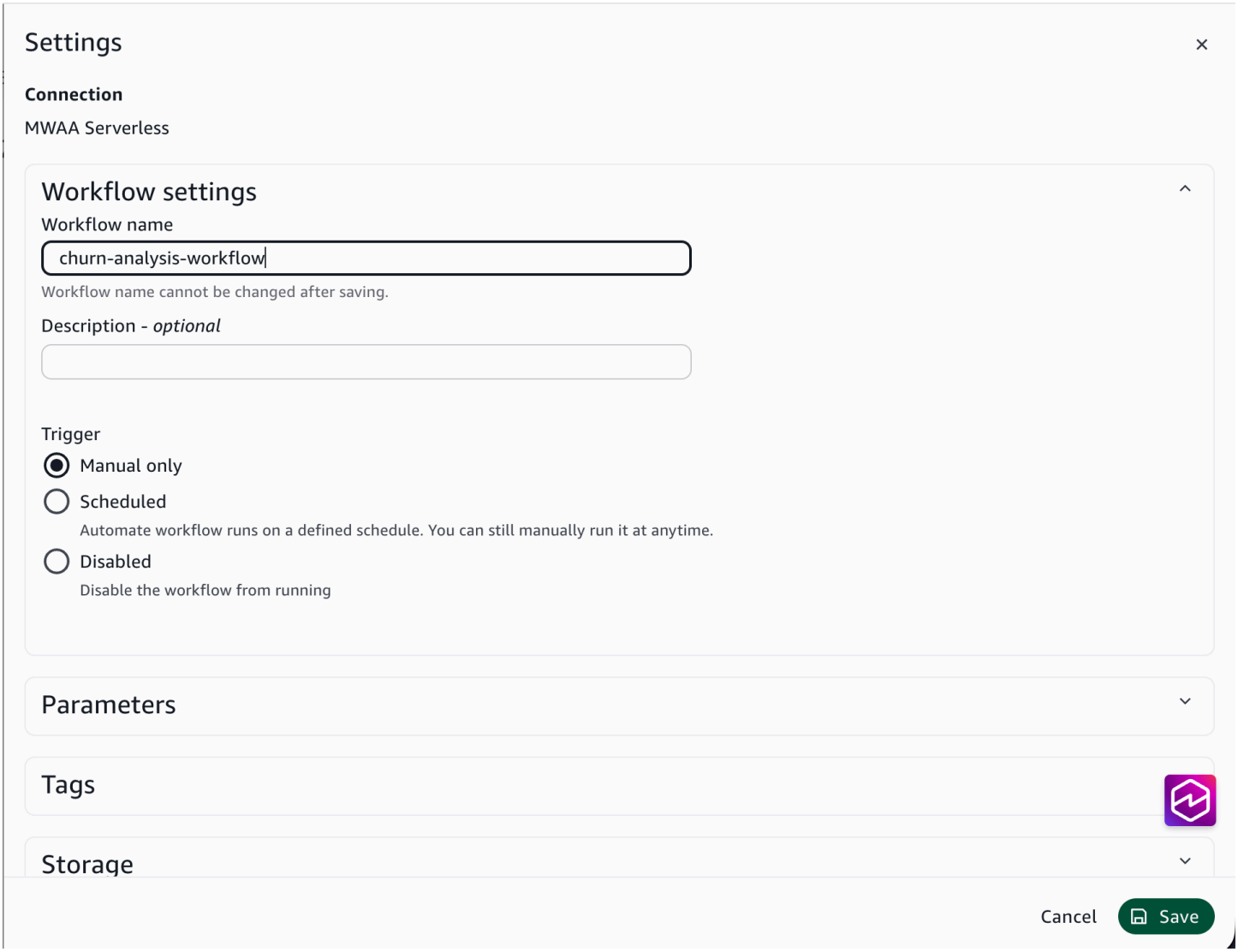
Step 4: Save and run the workflow

1. Enter a name for your workflow in the title field at the top of the canvas: `churn-analysis-workflow`.
2. Choose **Save** to save the workflow.
3. Choose **Run**.

The workflow begins executing. Each task runs in sequence — the Athena query first, then the visual ETL job. The visual ETL job step might take several minutes to complete, similar to when you ran it on its own.

Scheduling

In production, you can schedule this workflow to run on a recurring basis — for example, daily or triggered by new data. You can configure a schedule from the **Settings** panel, where you can choose **Manual only**, **Scheduled**, or **Disabled**.

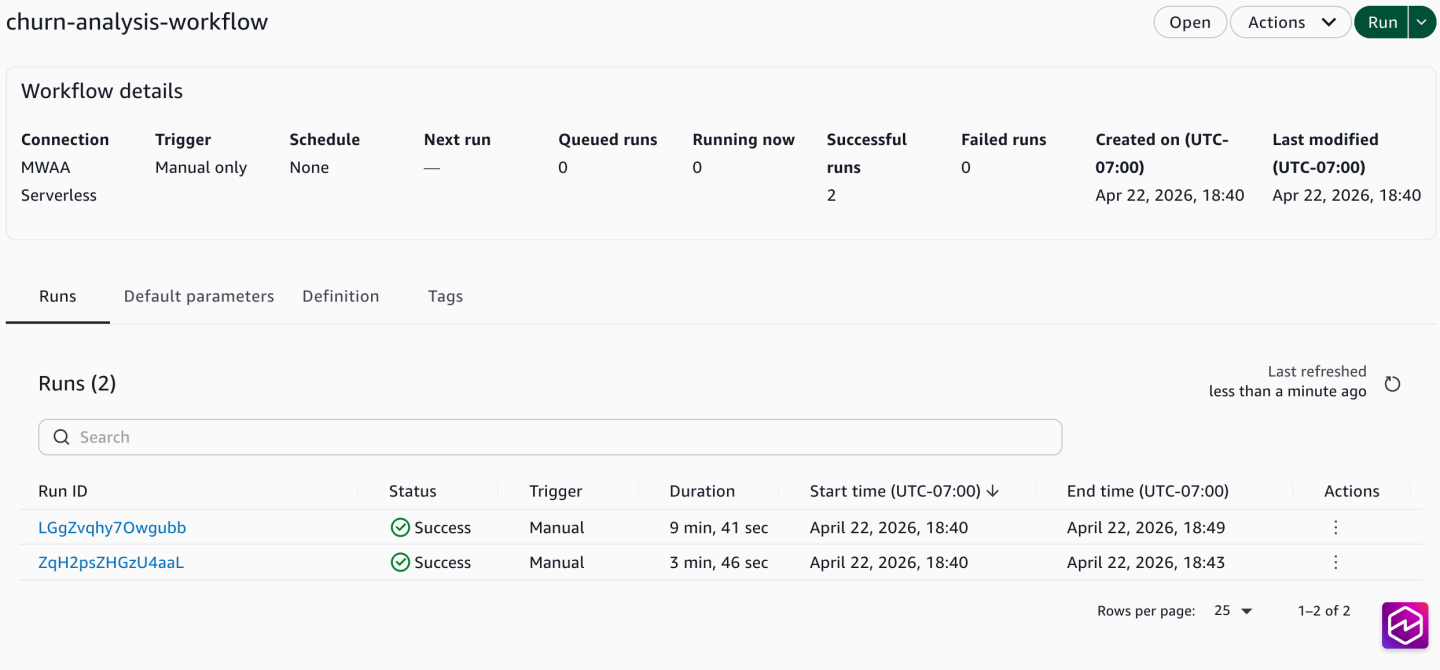


Step 5: Verify the output

After the workflow run completes, verify that both tasks succeeded.

1. Check the workflow run status. Both tasks show a **Succeeded** status.

2. To verify the ETL output, navigate to the S3 output folder (for example, `shared/filtered-churn/`) in the **Data explorer** and confirm Parquet files are present with the filtered data.



Troubleshooting

If a task fails, choose the failed task in the workflow canvas to view its error details. Common issues include IAM permission errors or incorrect S3 paths.

What you learned

In this tutorial, you:

- Created a workflow that chains multiple task types into a single pipeline
- Used an Athena operator to query sample data directly
- Reused an existing visual ETL job as a workflow task
- Ran the workflow and verified end-to-end execution

Workflows let you combine queries, ETL jobs, notebooks, and more into repeatable, automated pipelines. As your data processing needs grow, you can add more tasks, set up schedules, and build dependencies between steps — all from the visual workflow editor.

Train an ML model

Time: 15 minutes

Prerequisites: As a member of a SageMaker Unified Studio project, your IAM role needs the following managed policies:

- SageMakerStudioUserIAMConsolePolicy to sign in and access the project.
- SageMakerStudioUserIAMDefaultExecutionPolicy to access data and resources within the project.

If you don't have access, contact your administrator. If you are the administrator who set up the project, you already have the required permissions. Completing "Analyze and visualize data" is helpful, but not required.

Outcome: You open a sample notebook, explore a customer churn dataset, train a classification model, and identify the key factors that predict churn.

What you will do

In this tutorial, you will:

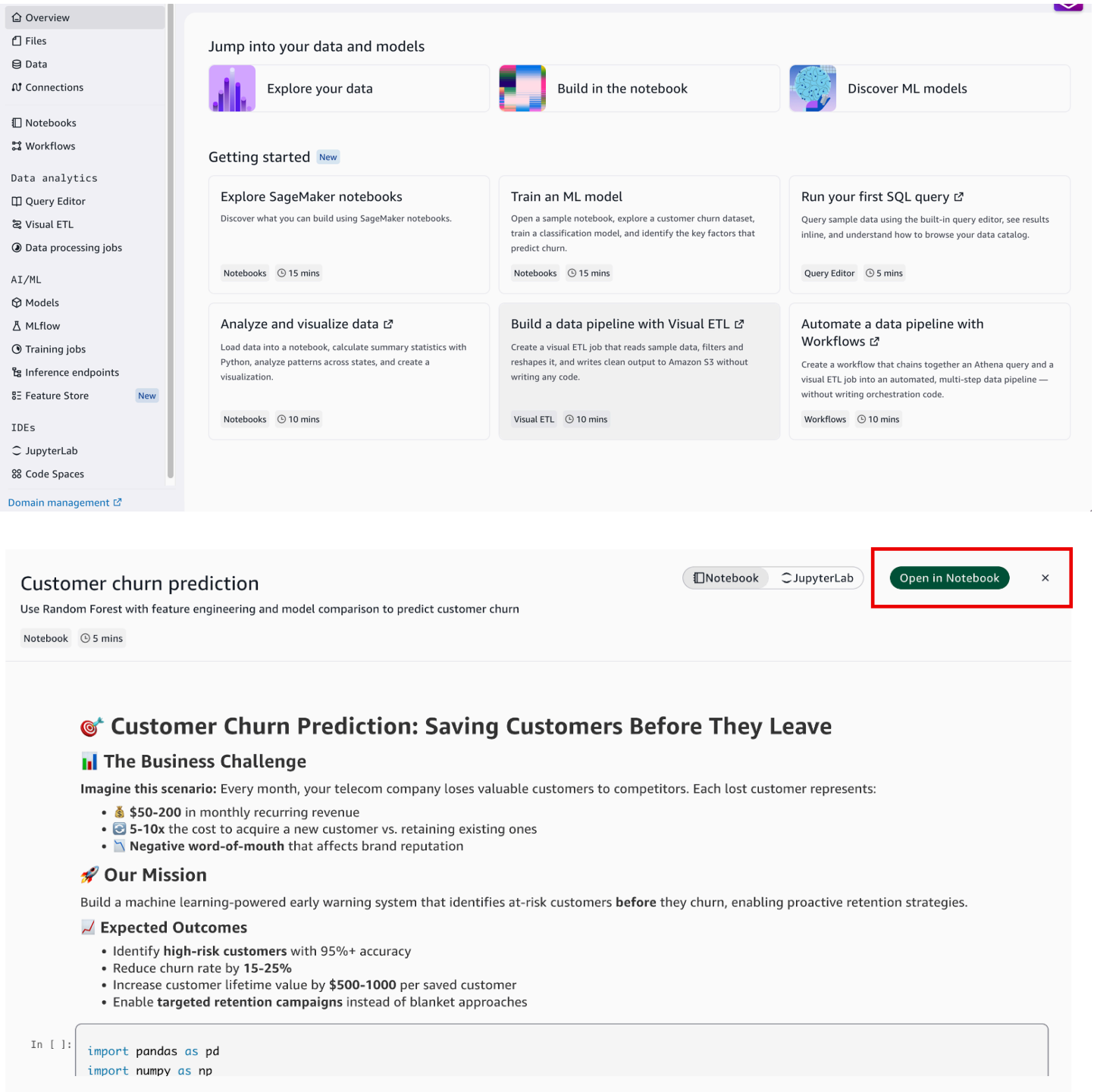
- Open a sample notebook in your project
- Load and explore a customer churn dataset
- Prepare features for model training
- Train and compare two classification models
- Identify the top factors that drive customer churn
- Save the trained model for future use

Machine learning uses historical data to find patterns and make predictions. In this tutorial, you train a model to predict which telecom customers are likely to cancel their service (churn). SageMaker Unified Studio provides a notebook environment with popular ML libraries pre-installed, so you can start training models immediately without any setup.

Step 1: Open the sample notebook

1. Go to your project using the menu at the top of the page.
2. On the project overview page, find the **Customer Churn Prediction** sample notebook.

3. Choose the notebook to open it.
4. Choose **Open in notebook**.



The notebook contains pre-written code cells that walk through the complete ML workflow. You run each cell in order.

What is a sample notebook?

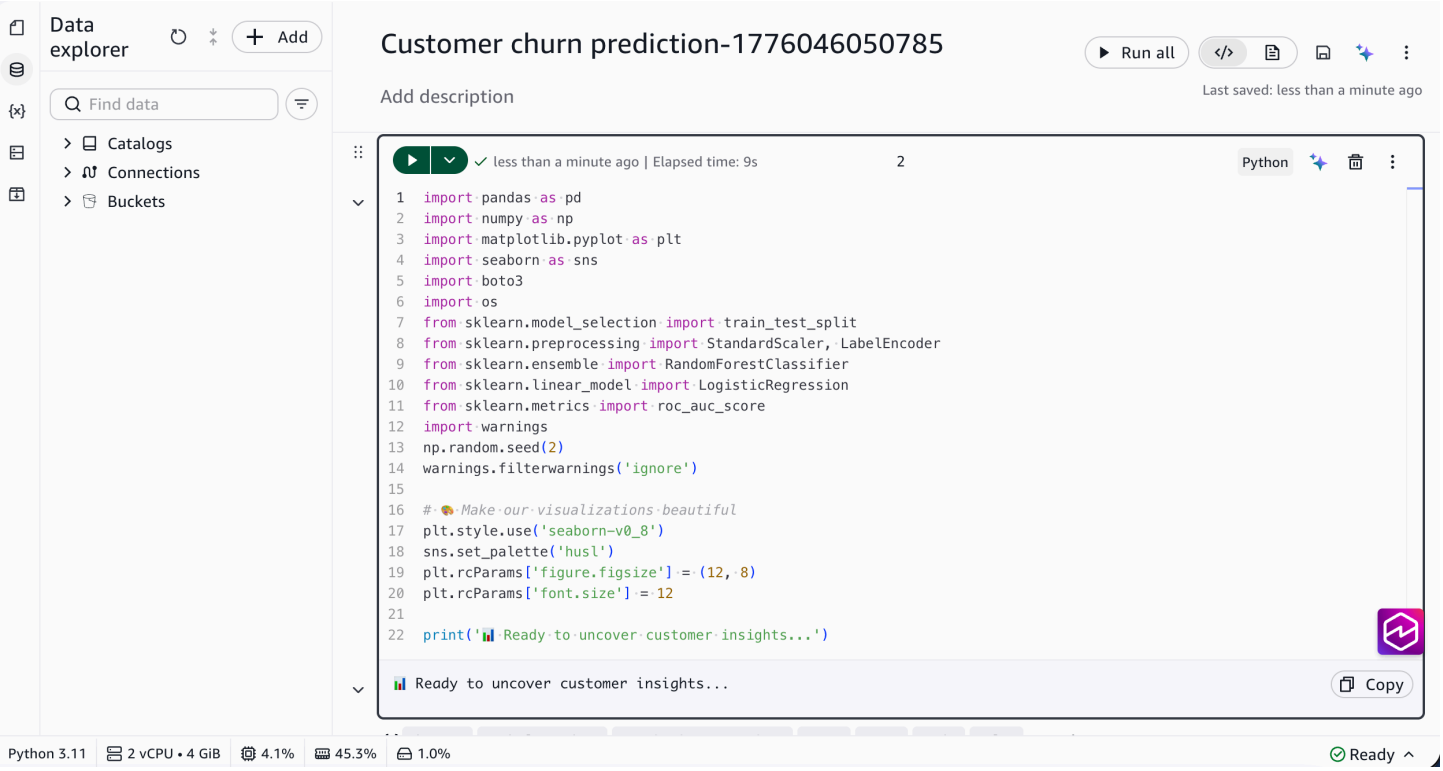
Sample notebooks are pre-built tutorials included in your project. They contain working code and explanations for common ML and data science tasks. You can run them as-is or modify them to use your own data.

Step 2: Set up and load the data

Run the first cell to import the required libraries. Choose the **Run** button (▶) in the top left corner of the cell:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import boto3
import os
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import roc_auc_score
import warnings
np.random.seed(2)
warnings.filterwarnings('ignore')
```

In this cell, `np.random.seed(2)` sets a random seed so you get the same results each time you run the notebook. The `warnings.filterwarnings` line suppresses deprecation warnings for cleaner output.



Run the next cell to load the customer churn dataset:

> **How to know when a cell finishes running**
>
> When a cell completes, a check mark appears next to it along with the elapsed time. Wait for this before running the next cell.

```
session = boto3.Session()
aws_region = session.region_name or 'us-west-2'

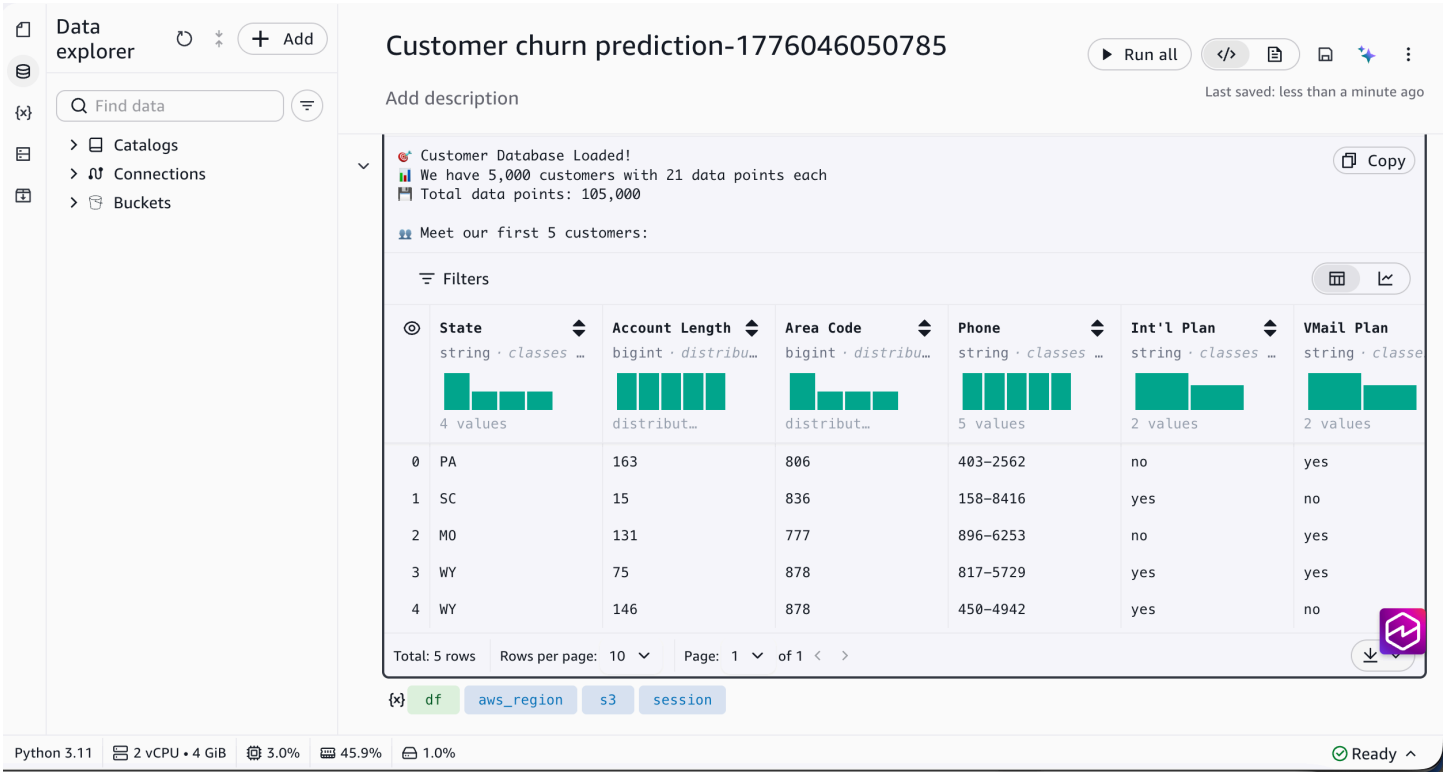
s3 = boto3.client('s3')
os.makedirs('notebook_outputs', exist_ok=True)

s3.download_file(
    f'sagemaker-example-files-prod-{aws_region}',
    'datasets/tabular/synthetic/churn.txt',
    'notebook_outputs/churn.txt'
)

df = pd.read_csv('notebook_outputs/churn.txt')
print(f'Dataset: {df.shape[0]:,} customers with {df.shape[1]} data points each')
```

```
df.head()
```

> **Note**
>
> The `sagemaker-example-files-prod` bucket is an AWS-managed public bucket that contains sample datasets. You do not need to create this bucket. The code downloads the dataset from this bucket to your notebook's local storage.



The dataset contains telecom customers with attributes including call minutes, service calls, charges, and whether the customer churned.

Step 3: Explore the churn problem

Run the next cell to calculate the churn rate and visualize the problem:

```
total_customers = len(df)
churned_customers = len(df[df['Churn?'] == 'True.'])
churn_rate = churned_customers / total_customers

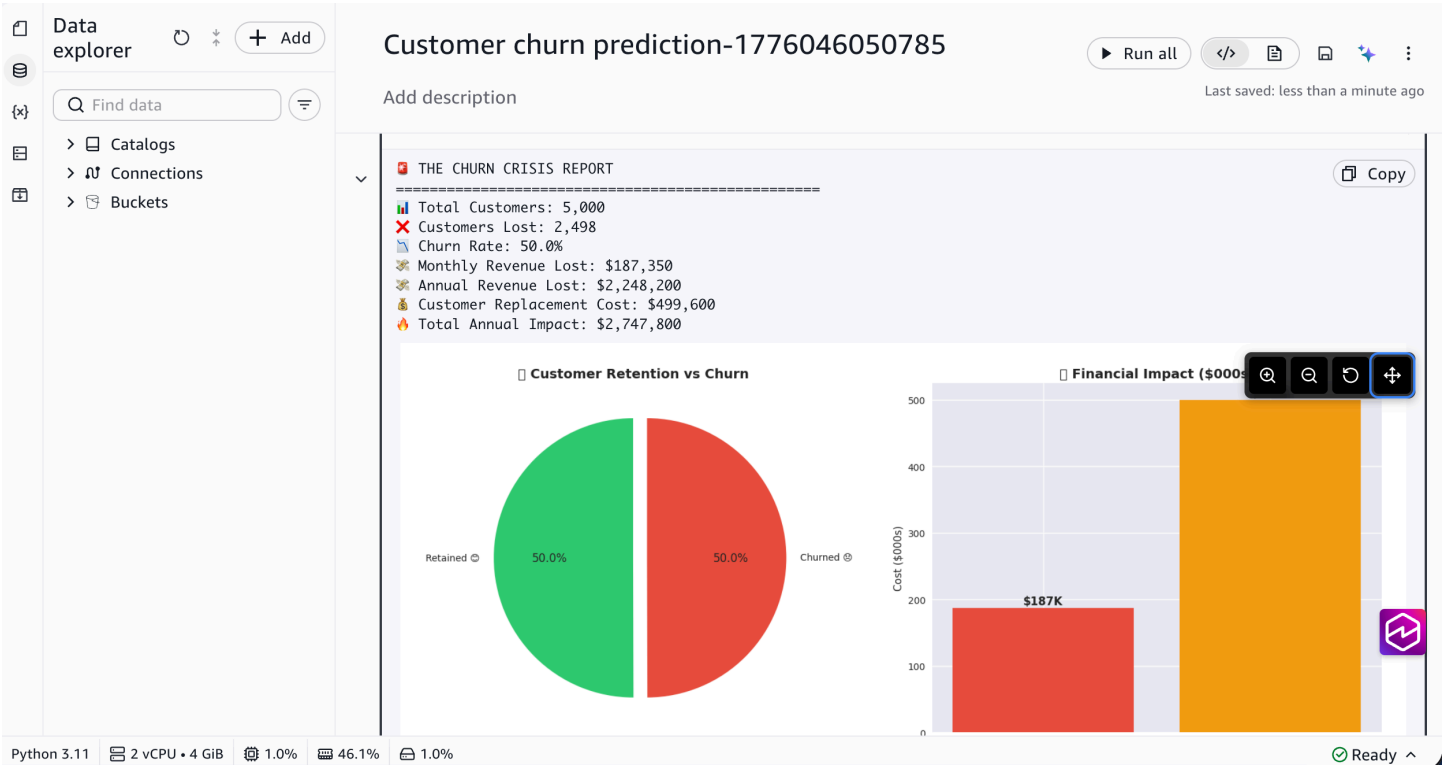
print(f'Total Customers: {total_customers:,}')
```

```
print(f'Customers Lost: {churned_customers:,}')
print(f'Churn Rate: {churn_rate:.1%}')

fig, axes = plt.subplots(1, 2, figsize=(15, 6))

churn_counts = df['Churn?'].value_counts()
colors = ['#2ecc71', '#e74c3c']
axes[0].pie(churn_counts.values, labels=['Retained', 'Churned'],
           autopct='%1.1f%%', colors=colors, startangle=90,
           explode=(0, 0.1))
axes[0].set_title('Customer Retention vs Churn')

plt.tight_layout()
plt.show()
```



The visualization shows the split between retained and churned customers. Understanding this distribution helps you choose the right approach for training your model.

Why explore before training?

Understanding your data before building a model helps you choose the right approach. For example, if the classes are heavily imbalanced (far more retained than churned customers), that affects how you evaluate model performance.

Step 4: Prepare features and train models

Before training, you need to convert the data into a format that ML algorithms can process. The following code encodes text columns as numbers, creates new features, and splits the data into training and test sets. Run the next cell:

```
df_processed = df.copy()
df_processed['Churn'] = (df_processed['Churn?'] == 'True.').astype(int)
df_processed.drop('Churn?', axis=1, inplace=True)
df_processed.drop('Phone', axis=1, inplace=True)

categorical_cols = ['State', "Int'l Plan", 'VMail Plan']
label_encoders = {}
for col in categorical_cols:
    le = LabelEncoder()
    df_processed[col] = le.fit_transform(df_processed[col])
    label_encoders[col] = le

df_processed['Total_Charge'] = (df_processed['Day Charge'] +
                                df_processed['Eve Charge'] +
                                df_processed['Night Charge'] +
                                df_processed['Intl Charge'])

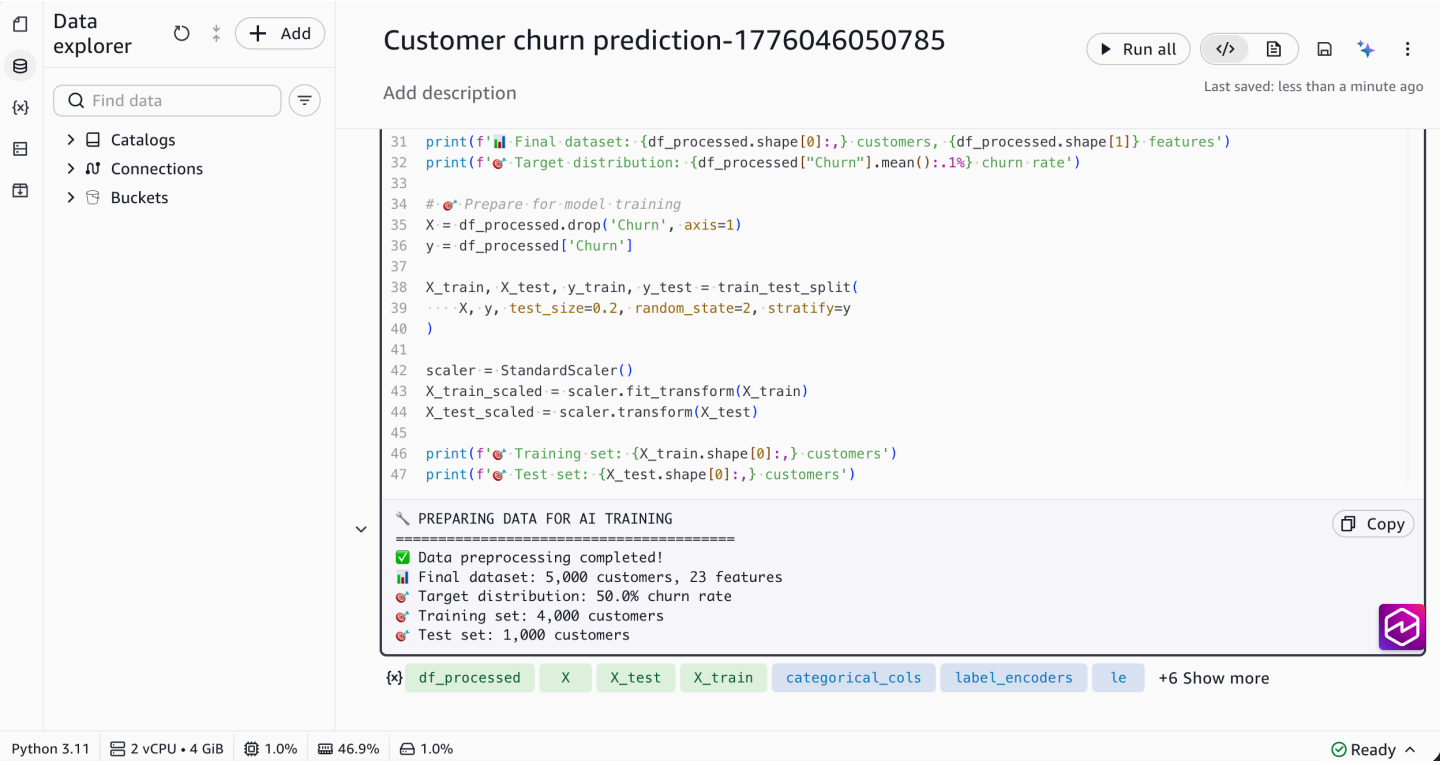
df_processed['High_Service_Calls'] = (df_processed['CustServ Calls'] >= 4).astype(int)

X = df_processed.drop('Churn', axis=1)
y = df_processed['Churn']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=2, stratify=y
)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

```
print(f'Training samples: {X_train.shape[0]:,}')
print(f'Test samples: {X_test.shape[0]:,}')
```



Now train two different classification models and compare their performance. Run the next cell:

```
models = {
    'Random Forest': RandomForestClassifier(n_estimators=100, random_state=2),
    'Logistic Regression': LogisticRegression(random_state=2, max_iter=1000)
}

model_results = {}

for name, model in models.items():
    print(f'Training {name}...')

    if 'Logistic' in name:
        model.fit(X_train_scaled, y_train)
        y_pred = model.predict(X_test_scaled)
        y_pred_proba = model.predict_proba(X_test_scaled)[:, 1]
    else:
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
```

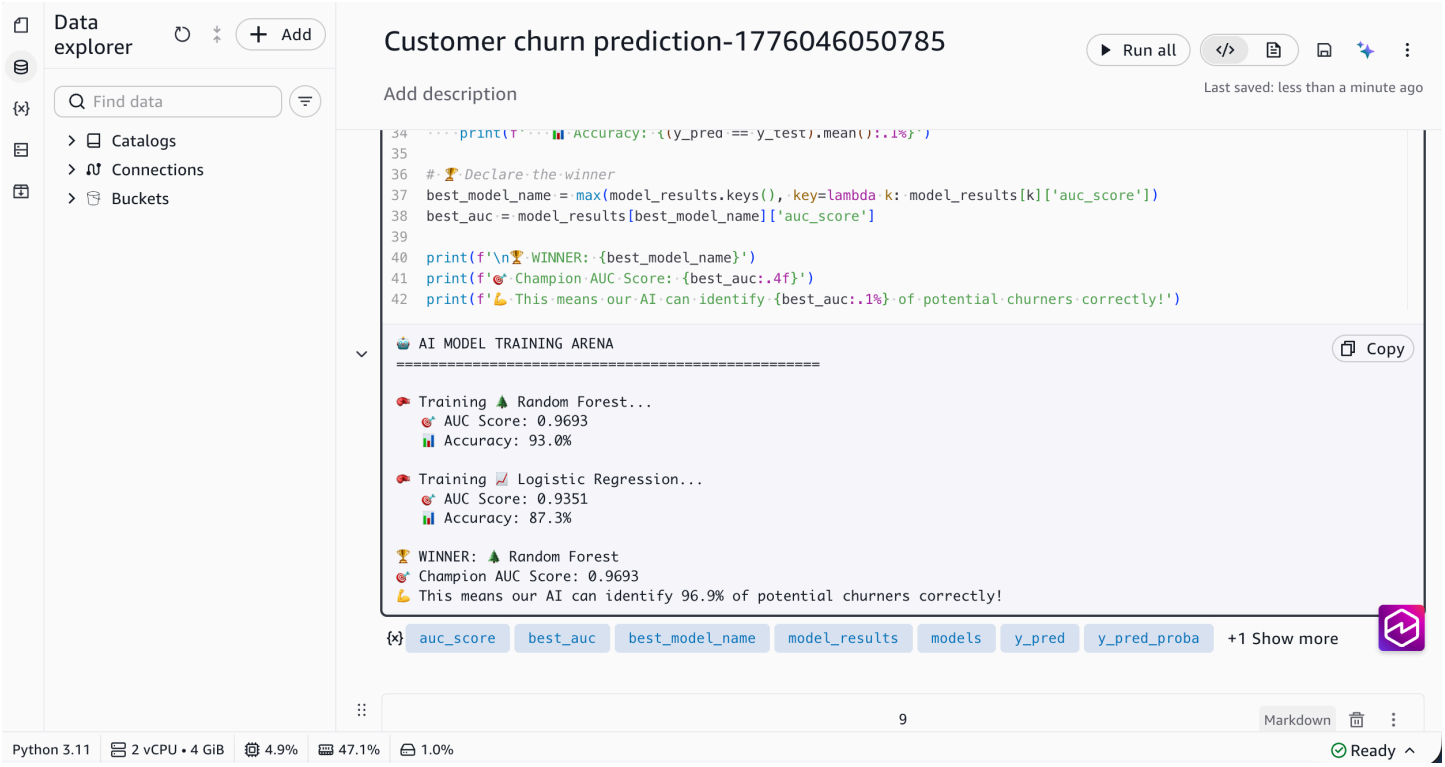
```
        y_pred_proba = model.predict_proba(X_test)[:, 1]

    auc_score = roc_auc_score(y_test, y_pred_proba)

    model_results[name] = {
        'model': model,
        'predictions': y_pred,
        'probabilities': y_pred_proba,
        'auc_score': auc_score
    }

    print(f'  AUC Score: {auc_score:.4f}')
    print(f'  Accuracy: {(y_pred == y_test).mean():.1%}')

best_model_name = max(model_results.keys(),
                      key=lambda k: model_results[k]['auc_score'])
print(f'\nBest model: {best_model_name}')
print(f'AUC Score: {model_results[best_model_name]["auc_score"]:.4f}')
```



What are these models?

A *Random Forest* builds many decision trees and combines their predictions. A *Logistic Regression* finds a mathematical boundary between the two classes. AUC (Area Under the

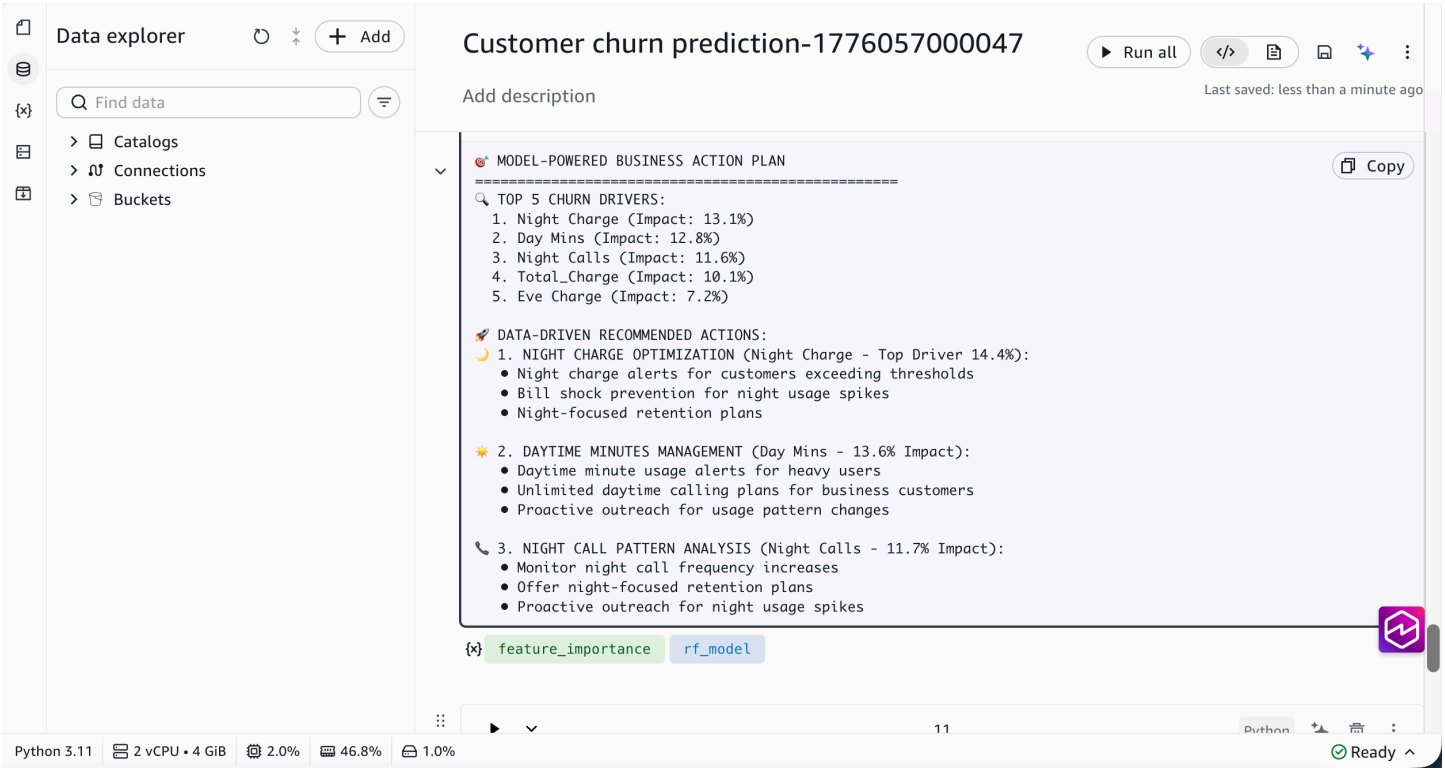
Curve) measures how well the model distinguishes between churners and non-churners, where 1.0 is perfect and 0.5 is random guessing.

Step 5: Understand what drives churn

The model can tell you which customer attributes are the strongest predictors of churn. Run the next cell to see the top churn drivers:

```
rf_model = model_results['Random Forest']['model']
feature_importance = pd.DataFrame({
    'feature': X.columns,
    'importance': rf_model.feature_importances_
}).sort_values('importance', ascending=False)

print('Top 5 churn drivers:')
for i, (_, row) in enumerate(feature_importance.head(5).iterrows(), 1):
    print(f'  {i}. {row["feature"]} (Impact: {row["importance"]:.1%})')
```



Feature importance reveals which factors have the biggest impact on churn predictions. These insights help the business focus retention efforts on the areas that matter most.

Use the Data Agent for deeper analysis

You don't need ML expertise to interpret these results. The **Data Agent** can help you understand feature importance, suggest next steps, and generate code for additional analysis. Open the Data Agent from the top navigation bar and ask questions like *"Why is night charge the top predictor of churn?"* or *"Write code to plot feature importance as a bar chart."*

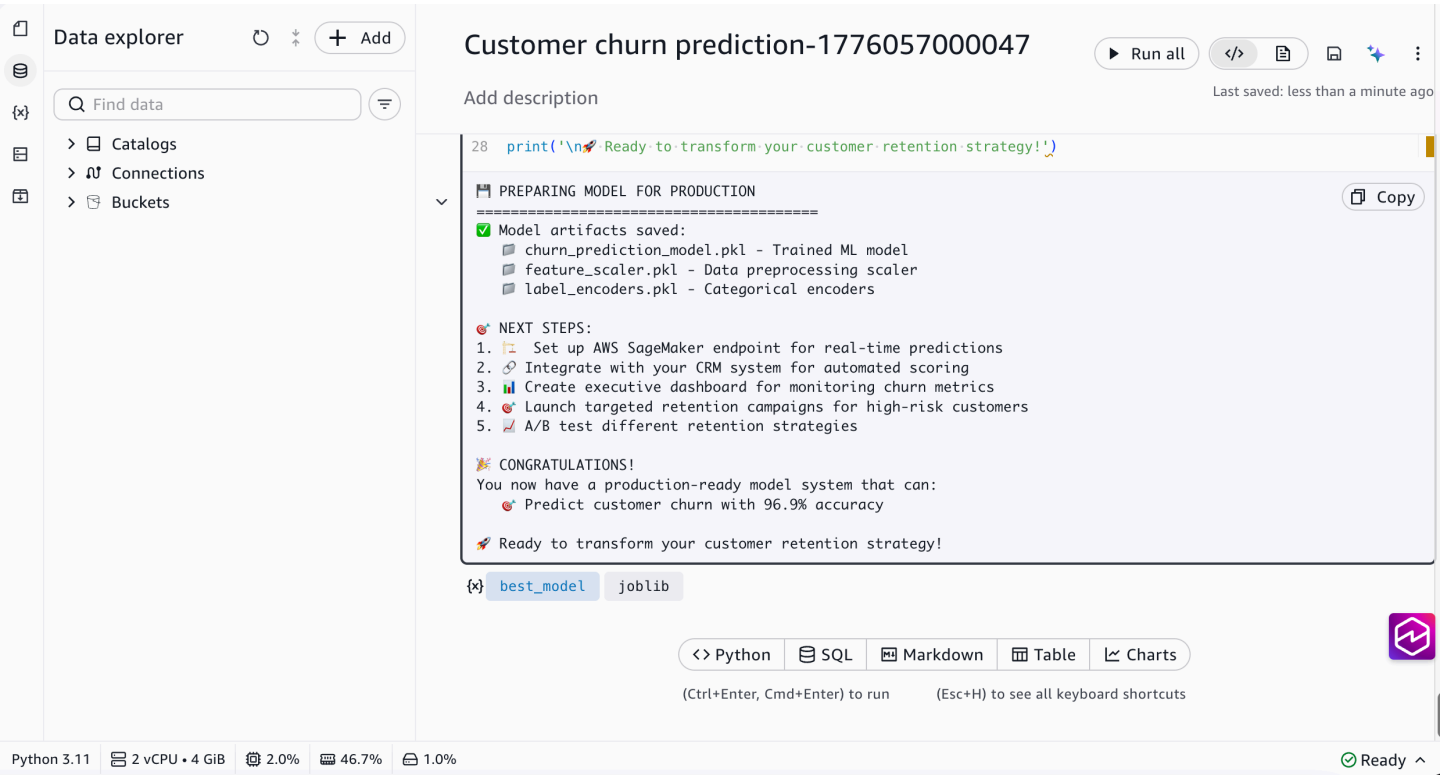
Step 6: Save the model

Run the final cell to save the trained model and its supporting artifacts. You can use these artifacts to load the model later for batch predictions, deploy it to a real-time SageMaker endpoint, or share it with your team through the model registry.

```
import joblib

best_model = model_results[best_model_name]['model']
joblib.dump(best_model, 'notebook_outputs/churn_prediction_model.pkl')
joblib.dump(scaler, 'notebook_outputs/feature_scaler.pkl')
joblib.dump(label_encoders, 'notebook_outputs/label_encoders.pkl')

print('Model artifacts saved:')
print('  churn_prediction_model.pkl - Trained ML model')
print('  feature_scaler.pkl - Data preprocessing scaler')
print('  label_encoders.pkl - Categorical encoders')
```



To reuse this model later, load the saved `.pkl` files using `joblib.load()` and call `model.predict()` on your data. For production use cases like real-time predictions or sharing the model with your team, see the What's next section below.

What's next

You trained a model using a sample notebook. Here are ways to go further:

- **Track experiments with MLflow**: Log your model parameters, metrics, and artifacts so you can compare runs and reproduce results. To set up MLflow for your project, see Track experiments using MLflow.
- **Deploy the model**: Serve your trained model as a real-time endpoint for predictions. To learn about model deployment, see Machine learning.
- **Use your own data**: Use similar techniques to load data from your lakehouse tables instead of the sample dataset. The Data Agent is already aware of the tables available in your catalog and can help you build and train your models.

What you learned

In this tutorial, you:

- Opened a sample notebook and loaded a customer churn dataset
- Explored the data and visualized the churn problem
- Prepared features and split data into training and test sets
- Trained and compared two classification models
- Identified the top factors that drive customer churn
- Saved the trained model for future use

aws

User Guide

Amazon SageMaker Unified Studio

Core capabilities



Notebooks

Overview

Notebooks in Amazon SageMaker Unified Studio provide an interactive environment for data analysis, exploration, engineering, and machine learning workflows. You can run SQL, Python, and natural language queries to discover, transform, analyze, visualize, and share insights on data at scale.

You can use notebooks in both IAM-based and IAM Identity Center-based domains.

Amazon SageMaker Unified Studio offers multiple coding experiences to meet different development preferences and use cases. JupyterLab IDE provides a traditional Jupyter notebook environment with extensive customization options and plugin support. Code Editor, based on Code-OSS, Visual Studio Code - Open Source, helps you write, test, debug, and run your analytics and machine learning code. Code Editor extends and is fully integrated with Amazon SageMaker Unified Studio. The new notebook experience, documented in this guide, provides a streamlined, AI-enhanced interface optimized for data analysis workflows with built-in visualization capabilities and seamless integration with AWS data services.

Notebooks support multiple cell types including Python code cells, SQL code cells, markdown cells, table cells, and chart cells. Each notebook runs on a managed compute environment that you can configure based on your processing requirements. You can run PySpark and Spark SQL in notebook cells using multiple Spark compute engines. Amazon Athena for Apache Spark is the default Spark runtime. You can also select additional Spark runtimes, such as Amazon EMR Serverless or AWS Glue, from the notebook side panel through Spark Connect. The selected runtime applies to both Python and SQL cells. For more information, see the section called “Spark Compute”.

The notebook interface integrates with AI assistance through SageMaker Data Agent, the AI agent that helps generate code, diagnose errors, and provide data analysis recommendations.

> **Note**
>
> Amazon Athena for Apache Spark doesn't support Virtual Private Cloud (VPC). If you require VPC connectivity for your workloads, use Spark on Amazon EMR Serverless, Amazon EMR, or AWS Glue instead. For instructions on disabling Amazon Athena Spark, see Network isolation in the *Amazon SageMaker Unified Studio Admin Guide*.

For more information about configuring VPC and network isolation for your domain, see Network isolation in the *Amazon SageMaker Unified Studio Admin Guide*.

Key capabilities

- Execute Python, Spark, and SQL code in interactive cells
- Integrate with Amazon Athena for Apache Spark for distributed processing
- Select from multiple Spark compute engines from the notebook side panel using Spark Connect, including Amazon Athena for Apache Spark, Amazon EMR Serverless, and AWS Glue
- Connect to multiple data sources including Amazon Simple Storage Service, Amazon S3 Tables, AWS Glue Data Catalog, Amazon Athena, and Amazon Redshift. For a complete list of supported sources, see Supported data connections.
- Work with Apache Iceberg REST Catalogs to read and write Iceberg tables by using Iceberg REST APIs in Python and SQL
- Visualize data with interactive tables and charts
- Use auto code completion, formatting, and linting in the cell editor
- Use AI assistance for code generation and error diagnosis
- Manage compute environments with configurable instance types
- Schedule notebook runs to execute automatically on recurring schedules, run notebooks in the background, and parameterize notebooks for reuse across different inputs
- Export notebooks in multiple formats including Jupyter and Python files
- Install and manage Python packages
- Run notebook kernels in your domain-level VPC configuration for network isolation

Roles and permissions

To use notebooks in Amazon SageMaker Unified Studio, you need:

- Access to an Amazon SageMaker Unified Studio domain
- Appropriate IAM permissions to access data sources
- Project membership with notebook creation permissions

Create and manage notebooks

Overview

You can create new notebooks in Amazon SageMaker Unified Studio to start data analysis workflows. Notebooks are automatically saved as you work, and you can organize them within your project structure.

The notebook interface provides access to sample notebooks that demonstrate common data analysis patterns. You can copy these samples to create starting points for your own analysis.

> **Important**
>
> In VPC-enabled domains that use only public subnets, notebooks can't run non-Python cells. To resolve this issue, configure your domain to use private subnets. For more information, see VPC troubleshooting in the *Amazon SageMaker Unified Studio Admin Guide*.

Procedure

1. Navigate to the Notebooks section in your Amazon SageMaker Unified Studio project.
2. Click Create notebook to start a new notebook.
3. Enter a name for your notebook or use the auto-generated name.
4. A Python cell is added by default. You can begin by adding Python code or adding cells to your notebook using the buttons (Python, SQL, Markdown, Table, Charts).
5. Your notebook saves automatically as you work.

To access sample notebooks:

1. In the Notebooks section, review the Build with sample data section.
2. Select a sample notebook that matches your use case.
3. Click on the sample to open it in read-only mode.
4. Copy the sample notebook to create your own editable version.

You can view all your notebooks in the notebooks list, which shows the name, ID, last updated information, and creation details for each notebook.

Examples

Amazon SageMaker Unified Studio comes with loaded sample notebooks for easy getting started and understanding the available capabilities. To view the available samples, go to the overview or notebooks pages in Amazon SageMaker Unified Studio. In addition, here are some cell code examples to demonstrate different cell types you can use in your notebooks:

Python cell example:

```
import pandas as pd
import numpy as np

# Create sample data
sample_data = {
    'product_id': ['P001', 'P002', 'P003', 'P004', 'P005'],
    'product_name': ['Laptop', 'Mouse', 'Keyboard', 'Monitor', 'Headphones'],
    'category': ['Electronics', 'Accessories', 'Accessories', 'Electronics',
 'Accessories'],
    'price': [999.99, 29.99, 79.99, 299.99, 149.99],
    'in_stock': [True, True, False, True, True]
}

df = pd.DataFrame(sample_data)
df
```

SQL cell example:

```
SELECT
    category,
    COUNT(*) as product_count,
    AVG(price) as avg_price,
    SUM(CASE WHEN in_stock THEN 1 ELSE 0 END) as in_stock_count
FROM df
GROUP BY category
ORDER BY avg_price DESC;
```

Markdown cell example:

```
# Data Analysis Report

## Overview
This notebook analyzes product inventory and pricing data.

## Key Findings
1. Electronics have higher average prices than accessories
2. Inventory levels are generally well-maintained
3. Price distribution shows clear category segmentation

## Next Steps
1. Analyze seasonal trends
2. Review pricing strategy
3. Optimize inventory levels
```

Spark cell example in Python:

```
from pyspark.sql.connect import functions as F

# Create Spark DataFrame
spark_df = spark.createDataFrame(df)

# Perform aggregations using Spark Connect
result = (
    spark_df.groupBy("category")
    .agg(
        F.count("*").alias("product_count"),
        F.avg("price").alias("avg_price"),
        F.count(F.when(F.col("in_stock") == True, 1)).alias("in_stock_count"),
    )
    .orderBy(F.col("avg_price").desc())
)
result.show()
```

Work with cells

Overview

Notebooks support a variety of languages such as Python, SQL, and Markdown. Each cell is associated with a language, and the editor in the cell supports functionality such as auto code complete, formatting, and linting.

All the code in the notebook is executed on the notebook kernel, which is built on IPython. The notebook kernel runs on SageMaker notebook compute, which has a configurable form factor that includes different types of instances. When cells are executed, they may produce output that is shown below each cell. Notebooks support rich rendering of data frames (pandas or Spark), where the output is rendered in an interactive data table and charts.

Procedure

To create and run Python code:

1. Choose **Python** to add a new Python cell.
2. Enter your Python code in the cell editor.
3. Choose the play icon or press **Shift+Enter** to run the cell.
4. View the results displayed below the cell.

All notebook code is executed in the notebook kernel, which runs on SageMaker compute. You can configure the form factor of this compute. The notebook runs an IPython kernel that can execute Python code. For larger scale data processing, the notebook's Python environment comes with Spark without requiring you to configure or manage any infrastructure. You can start writing Spark code to run interactive analytics and exploration on serverless, autoscalable Athena Spark.

To create and run SQL code:

1. Choose **SQL** to add a new SQL cell.
2. Select your data connection from the dropdown if prompted.
3. Enter your SQL query in the cell editor.
4. Choose the play icon or press **Shift+Enter** to run the cell.
5. View the query results in the interactive table below the cell.

SQL cells can query your existing Python data frames using DuckDB, or run SQL against Athena (SQL), Athena Spark, or any other connection to first-party and third-party engines like Amazon Redshift, Snowflake, and BigQuery. Add a connection from the available connections that are supported.

Run multiple SQL statements in a single cell

You can write and run multiple SQL statements in a single SQL cell by separating each statement with a semicolon (`;`). When you run a cell that contains multiple statements, the notebook executes them sequentially and displays the results in a tabbed interface.

To run multiple SQL statements:

1. In a SQL cell, enter two or more SQL statements separated by semicolons. For example:

```
SELECT COUNT(*) FROM customers WHERE region = 'US';
SELECT COUNT(*) FROM customers WHERE region = 'EU';
SELECT COUNT(*) FROM customers WHERE region = 'ASIA';
```

2. Choose the play icon or press **Shift+Enter** to run the cell.
3. The output area displays a row of tabs labeled **Result 1**, **Result 2**, **Result 3**, and so on — one tab for each statement.
4. Choose a tab to view the results for that statement.

Each tab displays a status indicator:

- A success indicator (✓) when the statement completed successfully.
- An error indicator (#) when the statement encountered an error. Choose the tab to view the error details.

Reference multi-statement SQL results in Python

When you run multiple SQL statements in a single cell, the notebook creates a separate data frame variable for each statement result. The variables follow this naming convention:

- *`dataframe_name`* — A list containing all results.
- *`dataframe_name`*`_0` — The result of the first statement.
- *`dataframe_name`*`_1` — The result of the second statement.

- *dataframe_name*_2 — The result of the third statement, and so on.

For example, if the cell's data frame variable is `df_cell_1`, the following variables are created:

Variable	Description
`df_cell_1`	A list containing all statement results
`df_cell_1_0`	Result of the first statement (Result 1 tab)
`df_cell_1_1`	Result of the second statement (Result 2 tab)
`df_cell_1_2`	Result of the third statement (Result 3 tab)

You can reference these variables in subsequent Python cells:

```
# Access the first statement's result
print(df_cell_1_0.head())

# Access by index from the list
print(df_cell_1[1].head())
```

You can rename the base data frame variable using the variable name editor in the cell's symbol bar. When you rename the base variable, all indexed variables update automatically. For example, renaming `df_cell_1` to `sales_data` updates the indexed variables to `sales_data_0`, `sales_data_1`, and so on.

> **Note**
>
> Individual indexed variables (such as `sales_data_0`) cannot be renamed independently. Only the base data frame name can be changed.

Visualize multi-statement results

Each result tab supports the full set of visualization options independently. You can:

- Switch between table and chart views for each result tab separately.

- Apply column filters, sorting, and pinning per result tab.
- Create different chart types for different statement results.

Visualization settings for the first result tab are saved with the notebook. Settings for additional result tabs are maintained during your current session but reset when you reopen the notebook.

Rename cells

You can assign custom names to cells to make them easier to identify and navigate, especially in large notebooks. By default, cells are labeled with sequential numbers (1, 2, 3, and so on).

To rename a cell:

1. Choose the cell number in the cell header. The number becomes an editable text field.
2. Enter a custom name for the cell.
3. Press **Enter** or choose outside the text field to save the name.

The custom name replaces the default number in the cell header. This is useful for:

- Identifying the purpose of specific cells at a glance (for example, "Data cleanup" or "Model training").
- Navigating large notebooks more efficiently.
- Making notebooks easier for collaborators to understand.

Keyboard shortcuts

Amazon SageMaker Unified Studio notebooks provide keyboard shortcuts that enable you to perform common actions and navigate within your notebooks. These shortcuts are familiar to developers who have used other Python or SQL notebook editors.

To view all available shortcuts, press ⌘**+Shift+/** (Mac) or **Ctrl+Shift+/** (Windows/Linux).

Notebooks support two modes for keyboard shortcuts: edit mode and command mode. Shortcuts without modifier keys are generally used in command mode.

Edit mode

Edit mode is active when you are editing the content of a cell. In this mode, keyboard shortcuts require modifier keys to avoid interfering with normal text input.

Action	Mac	Windows/Linux
Run the current cell	⌘+Enter	Ctrl+Enter
Run cell and select next cell	Shift+Enter	Shift+Enter
Toggle Gen AI prompting	Option+A	Alt+A

Command mode

To enter command mode, press **Esc**. A blue ring appears around the selected cell to indicate that command mode is active.

In command mode, the following actions are available in addition to the edit mode shortcuts:

Navigation

Action	Shortcut
Select above cell	Up
Select below cell	Down
Edit cell content	Enter

Cell creation

Action	Shortcut
Add cell above	A
Add cell below	B
Add Python cell	P
Add SQL cell	Q
Add Markdown cell	M

Action	Shortcut
Add table cell	T
Add charts cell	C

Cell operations

Action	Mac	Windows/Linux
Copy cell	⌘+C	Ctrl+C
Paste cell	⌘+V	Ctrl+V
Duplicate cell	⌘+D	Ctrl+D
Run cell and insert cell below	Option+Enter	Alt+Enter
Save notebook	⌘+S	Ctrl+S

Tip

When the intercell menu is open, you can use the typed cell shortcuts (**P**, **Q**, **M**, **T**, **C**) to add a cell of a specific type.

To add documentation:

1. Choose **Markdown** to add a Markdown cell.
2. Enter your documentation using Markdown syntax.
3. Choose the play icon or press **Shift+Enter** to render the formatted text.

To reference data between cells: Python variables created in one cell are available in subsequent cells. SQL query results can be referenced by variable name in Python cells. You can also use the variable explorer in the left navigation to see all available variables and their schemas.

Visualize and explore data

Overview

Amazon SageMaker Unified Studio notebooks provide rich data visualization and exploration capabilities. Data frames automatically render as interactive tables, and you can create dedicated chart cells for custom visualizations.

On the left navigation, the data explorer provides access to your data catalog for discovering and accessing datasets. The variable explorer shows all active variables in your notebook session, including their data types and schemas.

Procedure

To view data in interactive tables:

1. Execute a Python or SQL cell that returns a data frame – compatible types include pandas, pyarrow, pyspark. Note: There is a limit of 20,000 rows for loading on the interactive tables and charts.
2. The results automatically display as an interactive table below the cell.
3. Use the table controls to filter, sort, and explore the data.
4. Click column headers to see data distribution visualizations.

To create custom charts:

1. Click the Charts button to add a chart cell.
2. Select the data frame you want to visualize from the dropdown.
3. Choose your chart type and configure the axes.
4. The chart renders automatically based on your selections.

To explore variables:

1. Open the variable explorer panel in the notebook interface.
2. View all active variables, their types, and memory usage.
3. Click on data frame variables to expand and see their schema.
4. Use variable names to reference data in new cells.

To access the data catalog:

1. Open the data explorer panel.
2. Navigate through your available data catalogs and databases.
3. Use the actions menu to read data directly into your notebook.
4. Generate code to access specific tables or datasets.

Manage compute environments

Overview

Each notebook runs on a managed compute environment that provides the processing power for code execution. You can start, stop, and configure compute environments based on your workload requirements.

Compute environments support different instance types and sizes. You can also configure automatic shutdown to manage costs when the notebook is idle.

Procedure

To start a compute environment:

1. Open your notebook if the compute environment is not already running.
2. The compute environment starts automatically when you execute your first cell.
3. Monitor the status in the kernel footer at the bottom right of the notebook.

To stop a compute environment:

1. Click the kernel status indicator in the notebook footer.
2. Select Stop to shut down the compute environment.
3. Your notebook content is preserved, but variables and session state are lost.

To change the instance type and storage:

1. Stop the current compute environment if it's running.
2. Click the kernel status indicator and select Configure.

3. Choose a different instance type and storage from the available options.
4. Start the compute environment with the new configuration.

To manage packages on kernel:

1. Open your notebook
2. Go to Packages on the left navigation
3. Add a new package or delete existing packages as required

To configure idle shutdown:

1. Access the idle shutdown setting through the left navigation.
2. Set the idle timeout period for automatic shutdown. The default shutdown is 1 hour.
3. The environment will automatically stop after the specified idle time.

To view Spark UI and Driver Logs:

1. Open your notebook
2. In the kernel footer of the notebook, you can choose Spark UI and Driver Logs to open in a separate tab.

To select a Spark runtime:

1. Open the compute pane in the notebook side panel.
2. Expand the **Spark** section.
3. Select a Spark Connect connection from the available runtimes.
4. Choose **Apply**.

For more information about available Spark runtimes and setup, see the section called “Spark Compute”.

Spark Compute

Overview

Amazon SageMaker Unified Studio notebooks support Spark Connect, a protocol that enables running PySpark and Spark SQL commands directly in notebook cells using different Spark compute engines. You can select your Spark runtime from the notebook side panel and start writing Spark commands with no additional setup or special syntax.

The following Spark runtimes are available:

- **Amazon Athena for Apache Spark** – The default Spark runtime. Amazon Athena for Apache Spark makes it easy to interactively run data analytics using Apache Spark without the need to plan for, configure, or manage resources. Available automatically in notebooks.
- **AWS Glue** – A serverless Spark runtime powered by AWS Glue. AWS Glue lets you build, test, and run data preparation and analytics applications using Apache Spark without managing infrastructure. Requires a AWS Glue connection in your project.
- **Amazon EMR Serverless** – An alternate Spark runtime. Amazon EMR Serverless lets you run Apache Spark applications without configuring, managing, or scaling clusters. You pay only for the resources your application uses. Requires an EMR Serverless application with Interactive Sessions enabled.

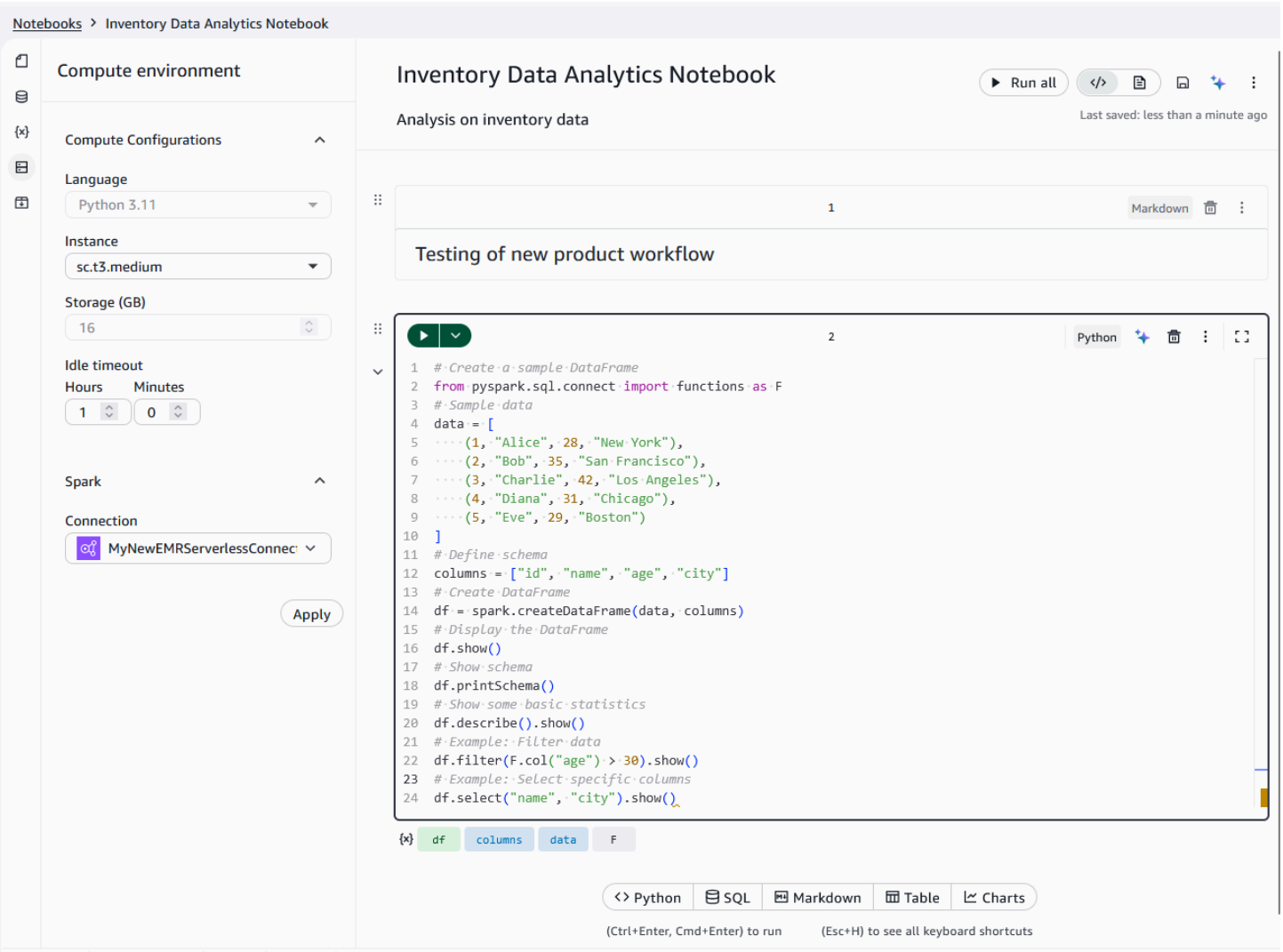
> **Note**
>
> Spark Connect on Amazon EMR Serverless is available with EMR release 7.13 and above.

Select a Spark runtime

To select or switch your Spark runtime:

1. Open a notebook in your project.
2. Open the compute pane in the notebook side panel.
3. Expand the **Spark** section to see available connections.
4. Select a Spark Connect connection.
5. Choose **Apply**.

The selected runtime applies to both Python and SQL cells. When you switch to a different Spark runtime, the current session restarts and any unsaved variables or state are lost. A confirmation dialog appears before the switch.



In SQL cells, the connection dropdown shows the Spark runtime selected at the notebook level.

Spark UI monitoring

Spark UI is available for all supported Spark engines. After executing Spark code in a Python or SQL cell, you can access the Spark UI and Spark Driver Logs from the kernel footer at the bottom of the notebook. Choose the links to open them in a separate tab for visibility into job execution and performance.

Use AWS Glue with Spark Connect

AWS Glue provides a serverless Apache Spark runtime for interactive data preparation and analytics. When you select a AWS Glue connection from the compute panel, Amazon SageMaker Unified Studio establishes a Spark Connect session backed by AWS Glue managed infrastructure. You can then run PySpark and Spark SQL code directly in notebook cells.

> **Note**
>
> If your domain uses network isolation, ensure that the required Amazon VPC endpoints for AWS Glue are configured. For more information, see Network isolation in the *Amazon SageMaker Unified Studio Admin Guide*.

Connect to AWS Glue

To use AWS Glue, follow the steps in the section called "Select a Spark runtime". In the **Spark** section, select your AWS Glue connection from the dropdown and choose **Apply**.

After you apply the connection, you can run PySpark and Spark SQL code in Python and SQL cells. The AWS Glue runtime manages Spark infrastructure transparently.

Limitations

- Fine-grained access control (FGAC) is not supported. Only full-table access is available.
- Trusted identity propagation (TIP) is not supported.
- Job bookmarks are not supported in interactive sessions.
- Scala is not supported. Use PySpark or Spark SQL.
- Switching runtimes restarts the session. Any unsaved variables or state are lost.

Use EMR Serverless with Spark Connect

Amazon EMR Serverless provides a pay-per-use Spark compute option with no cluster management. When configured, EMR Serverless applications are provisioned with pre-initialized capacity (1 driver, 3 executors) for approximately 30-second session start times. Pre-initialized capacity launches when a session starts and shuts down after 15 minutes of idle time.

> **Note**
>
> If your domain uses network isolation, ensure that the required Amazon VPC endpoints for Amazon EMR are configured. For more information, see Network isolation in the *Amazon SageMaker Unified Studio Admin Guide*.

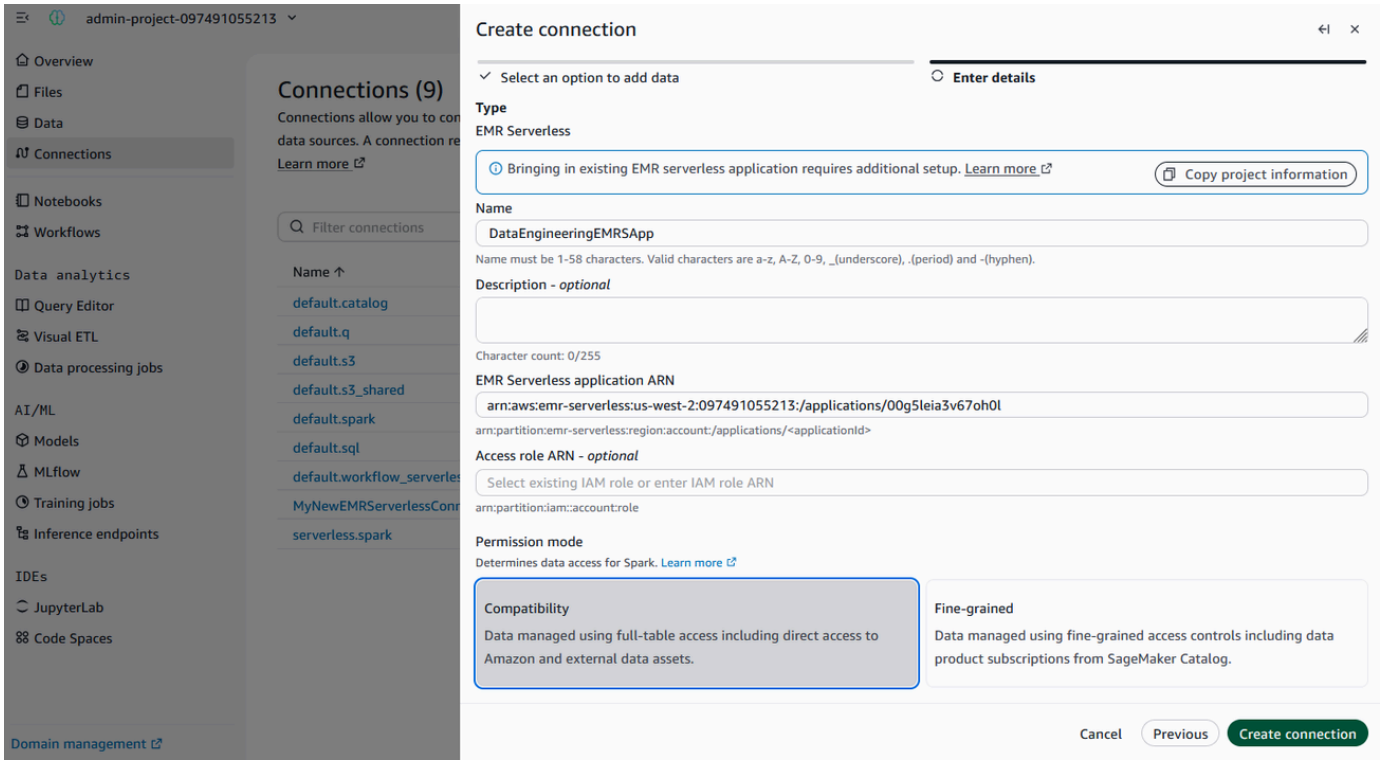
Prerequisites

- EMR Serverless application with release label `emr-7.13.0` or above
- Interactive Sessions enabled on the EMR Serverless application
- Compatibility permission mode (fine-grained access control and trusted identity propagation are not supported)

Set up EMR Serverless in IAM domains

In IAM domains, you set up EMR Serverless Spark Connect by creating the application yourself and adding it as a connection.

1. Create an EMR Serverless application with Interactive Sessions enabled and release label `emr-7.13.0` or above. You can create the application from Amazon EMR Studio or the AWS CLI. Make sure to select **Enable Interactive Sessions** to enable Spark Connect endpoints. For more information, see Interactive workloads in the *Amazon EMR Serverless User Guide*.
2. In your Amazon SageMaker Unified Studio project, navigate to **Connections** > **Create Connection** > **EMR Serverless**. Provide a connection name, select the EMR Serverless application ARN, and choose **Compatibility** permission mode.



If the connection does not appear or is not working, verify the following:

- The EMR Serverless application has Interactive Sessions enabled.
- Permission mode is set to compatibility.
- The domain does not have trusted identity propagation enabled.

Set up EMR Serverless in IAM Identity Center domains (new projects)

In IAM Identity Center domains, you can use the EMR Serverless blueprint to automatically provision EMR Serverless Spark Connect applications.

1. As an administrator, update your project profile to include the `EmrServerless` blueprint. Set the blueprint to `ON-DEMAND` or `ON-CREATE` based on your preference. For more information, see Update project profiles in the *Amazon SageMaker Unified Studio Admin Guide*.
2. Create a new project using the project profile. Set the release label to `emr-7.13.0` and permission mode to compatibility.
3. Verify the EMR Serverless application is created with Interactive Sessions enabled.

Set up EMR Serverless in IAM Identity Center domains (existing projects)

1. As an administrator, update the project profile to add the `EmrServerless` blueprint in `ON-DEMAND` mode.
2. As a user, go to the project and use **Add Compute** to add an EMR Serverless environment.
3. Select a release label of `emr-7.13.0` or above and choose compatibility permission mode.
4. Verify the EMR Serverless application is created with Interactive Sessions enabled.

Limitations

- Fine-grained access control (FGAC) is not supported. Only full-table access is available.
- Trusted identity propagation (TIP) is not supported.
- Switching runtimes restarts the session. Any unsaved variables or state are lost.

Connect to data sources

Overview

Amazon SageMaker Unified Studio notebooks can connect to multiple data sources including Amazon Simple Storage Service, AWS Glue Data Catalog, Amazon Athena, Amazon Redshift, and third-party sources. You can query data directly from these sources using SQL cells or Python code.

The notebook interface provides built-in connectors for AWS services and supports custom connections for external data sources. Data connections are configured at the project level and shared across notebooks.

Prerequisites

1. Configured data connections in your Amazon SageMaker Unified Studio project
2. Appropriate IAM permissions to access data sources
3. Network connectivity to external data sources if applicable

Procedure

To query data from Amazon Simple Storage Service:

1. Create a Python cell and use the AWS SDK to access S3 objects.
2. Use pandas or other libraries to read data from S3 into data frames.
3. Reference the data frame variables in subsequent cells.

To query AWS Glue Data Catalog tables:

1. Create a SQL cell and select the Athena (SQL) connection.
2. Write SQL queries against your cataloged tables.
3. The queries execute using Amazon Athena.
4. Results display as interactive tables below the cell.

To connect to Amazon Redshift:

1. Create a SQL cell and select your Redshift connection from the dropdown.
2. Write SQL queries against your Redshift data warehouse.
3. Execute the queries to retrieve results into the notebook.

To use Amazon Athena for Apache Spark:

1. Create Python cells that use Spark DataFrames for large-scale data processing.
2. Create SQL cells and choose the Athena (Spark) Connection to write SQL queries against your Spark DataFrames.
3. Reference Spark DataFrames by variable name to see rich table visualizations.
4. Access the Spark UI to monitor job progress and performance.

To work with third-party data sources:

1. Configure connections to external sources like Snowflake in your project settings. You can view supported data sources here.
2. Use SQL cells with the appropriate connection to query external data.
3. Combine data from multiple sources using Python code to join datasets.

Import and export notebooks

Overview

Amazon SageMaker Unified Studio supports importing and exporting notebooks. You can import notebook files from your local machine to migrate work from other environments, and export notebooks in multiple formats for sharing and offline use.

Import a notebook

You can import a notebook file from your local machine into your project. The import process converts the file to a Amazon SageMaker Unified Studio notebook and recreates cells, cell outputs, and metadata from the source file.

Supported import formats:

- Jupyter notebook (`.ipynb`)
- Amazon SageMaker Unified Studio native JSON (`.json`)
- Python script (`.py`)

To import a notebook

1. In the navigation pane, choose **Notebooks**.
2. Choose the dropdown arrow next to **Create notebook**, and then choose **Import notebook**.
3. In the **Import notebook** dialog box, choose **Choose file**, or drag and drop a file. The file must be in `.ipynb`, `.json`, or `.py` format, with a maximum size of 50 MiB.
4. Choose **Import**.

Once imported, the notebook opens and ready to use. Here, you can edit the name, description, cells and run the notebook.

What gets preserved during import:

- All cell types: Document, Code, and Visualization
- Cell outputs and execution history, when available in the source file
- Amazon SageMaker Unified Studio metadata such as connection IDs and cell configurations, when available

The following table describes the import limits.

Resource	Limit
Maximum file size	50 MiB
Maximum cells per notebook	200
Maximum cell content size	300 KB
Maximum cell run results per cell	100
Maximum cell run results size	300 KB

> **Note**
>
> The notebook cannot be modified while an import is in progress. Import is all-or-nothing — either the entire notebook imports successfully, or the import fails.

Export a notebook

You can export a notebook to your local machine in multiple formats for sharing and offline use.

Supported export formats:

- Jupyter notebook with requirements (`.zip`)
- Jupyter notebook (`.ipynb`)
- Python (`.py`)
- Notebook (`.json`)

To export a notebook

1. Open the notebook you want to export.
2. In the notebook toolbar, choose the more options menu (three vertical dots).
3. Choose **Download**, and then select your desired format:

- Jupyter notebook with requirements (`.zip`) — includes the notebook and a requirements file for reproducing the environment
- Jupyter notebook (`.ipynb`) — for use with other Jupyter-compatible environments
- Python (`.py`) — extracts code cells as a Python script
- Notebook (`.json`) — Amazon SageMaker Unified Studio native format

The file downloads to your local machine.

To download results and data

1. Execute code that generates output files or processed datasets.
2. Use the download interface to save files to your local machine or Copy to clipboard.
3. Select the output format (CSV or JSON or pdf).

Schedule and automate notebook runs

Overview

Amazon SageMaker Unified Studio provides capabilities to automate notebook execution beyond interactive sessions. You can run notebooks in the background on dedicated compute, create recurring schedules for automated execution, parameterize notebooks for reuse with different inputs, and orchestrate multi-notebook workflows.

These capabilities help you move from exploratory analysis to production-ready data pipelines without leaving the notebook environment.

Running a notebook in the background

Background runs execute your notebook on dedicated compute that is separate from your interactive session. This frees your session for other work while the notebook runs.

To run a notebook in the background:

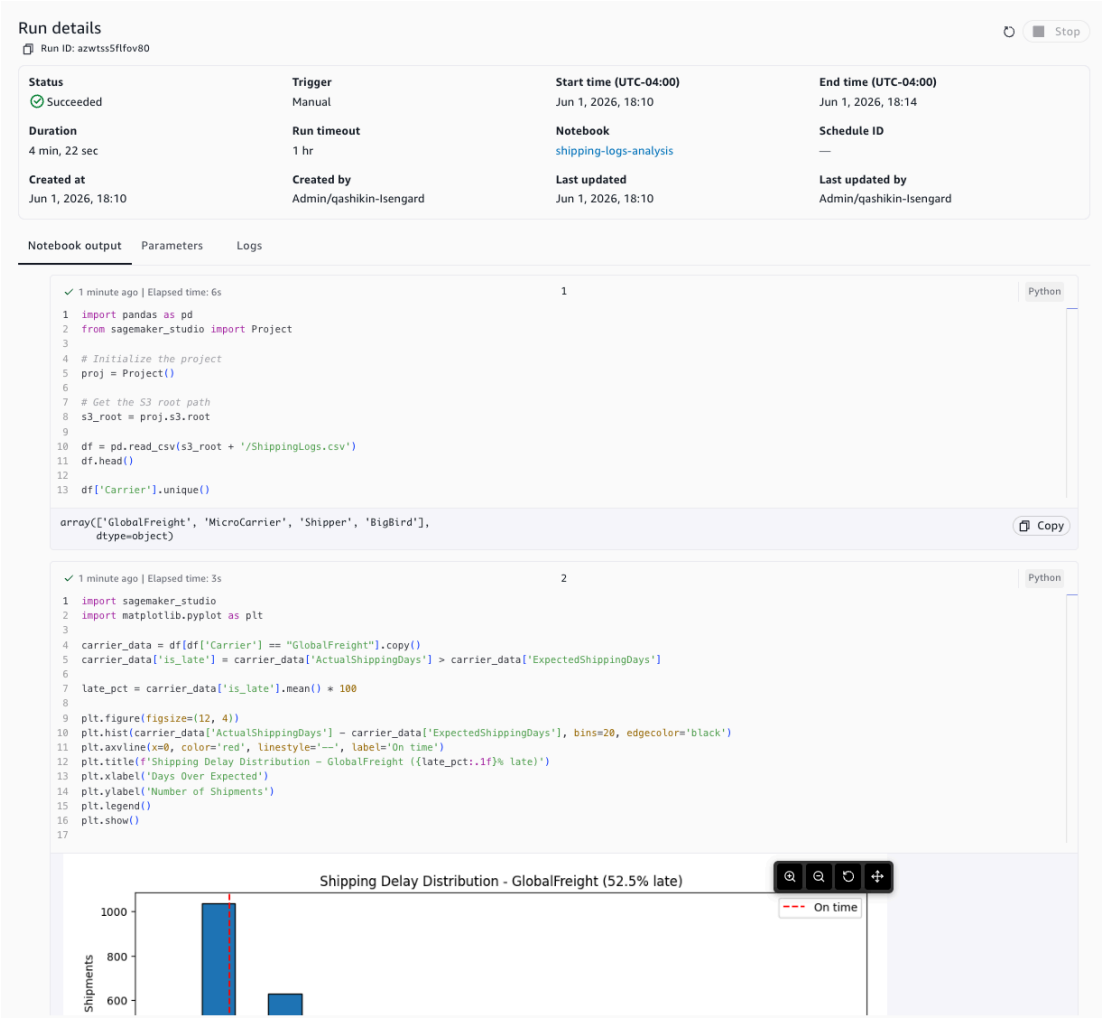
1. Open the notebook that you want to run.
2. In the notebook header, choose the menu on the **Run all** button, and then choose **Run in background**.



The notebook begins executing on dedicated compute. You can continue working in your interactive session.

To view the status and output of a background run:

1. Open the **Runs** activity panel.
2. Locate the run in the list. The status indicates whether the run is in progress, succeeded, or failed.
3. Choose the run to view the following tabs:
   - **Output** – The executed notebook with cell outputs.
   - **Parameters** – The parameter values used for the run.
   - **Logs** – Execution logs for debugging.



To stop a background run:

1. Open the **Runs** activity panel.
2. Locate the in-progress run.
3. Choose **Stop** to cancel the execution.

Background run considerations

- Background runs use dedicated compute that is isolated from your interactive session.
- Packages installed in your interactive environment are available in background runs.

- Local files from your interactive session are not accessible. Store files in Amazon Simple Storage Service for access during background runs.
- Background runs have a startup time while the dedicated compute initializes.

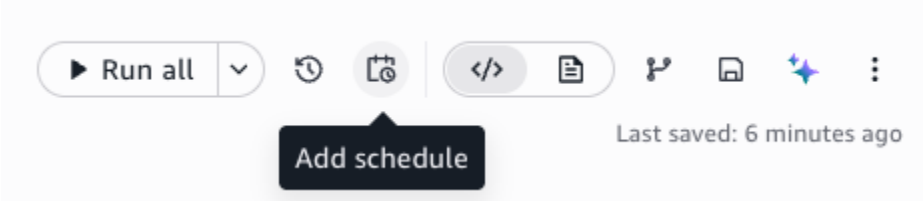
Creating a recurring schedule

You can create schedules to run notebooks automatically at specified intervals. Amazon SageMaker Unified Studio creates an Scheduler schedule for each configured schedule.

> **Tip**
>
> You can also use the SageMaker Data Agent to create schedules and start background runs using natural language prompts. For more information, see the section called "Getting started with the SageMaker Data Agent for Notebook".

To create a schedule:

1. Open the notebook that you want to schedule.
2. Choose the schedule icon in the notebook toolbar.



3. Configure the schedule settings.
4. Choose **Create**.

The following settings are available when you create a schedule:

Schedule name

A descriptive name for the schedule.

Schedule type

Choose **Recurring** for repeated execution or **One-time** for a single future run.

Frequency

For recurring schedules, specify the frequency using a rate expression (for example, every 1 hour) or a cron expression for more complex patterns.

Time zone

The time zone for the schedule.

Start date and end date

Optional start and end dates to control when the schedule is active.

Flexible time window

An optional window during which the schedule can run, which helps distribute load.

The following advanced settings are available:

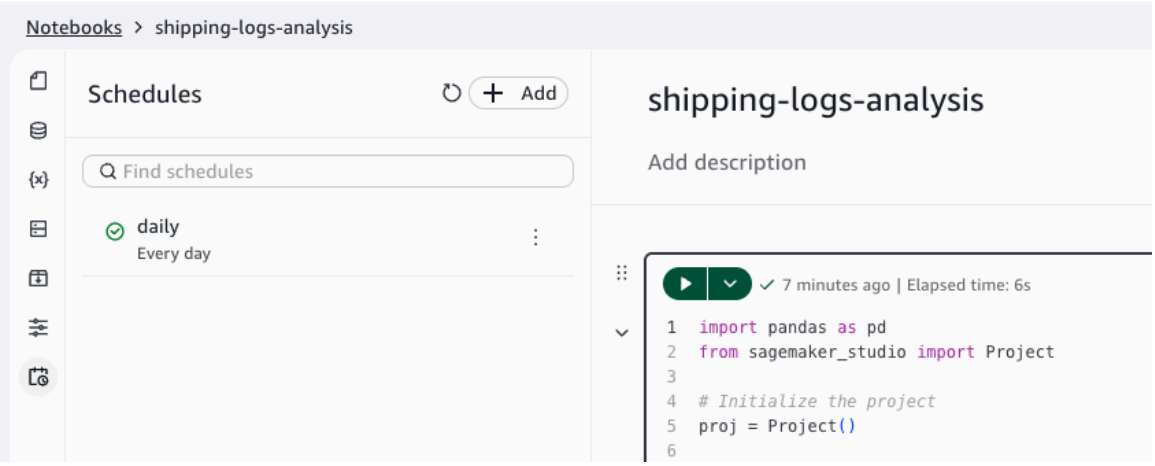
Compute instance override

Specify a different compute instance type for scheduled runs than the one used in your interactive session.

Timeout

The maximum duration for the run. The default timeout is 60 minutes.

The schedule appears in the **Schedules** tab of the activity panel after creation.



Viewing, editing, and deleting schedules

You can manage your schedules from the notebook toolbar or the Runs activity panel.

To view run history:

1. Open the **Runs** activity panel.
2. View the list of completed and in-progress runs with their status and timestamps.

To edit a schedule:

1. Choose the schedule icon in the notebook toolbar.
2. Locate the schedule that you want to modify.
3. Update the schedule settings as needed.
4. Choose **Save**.

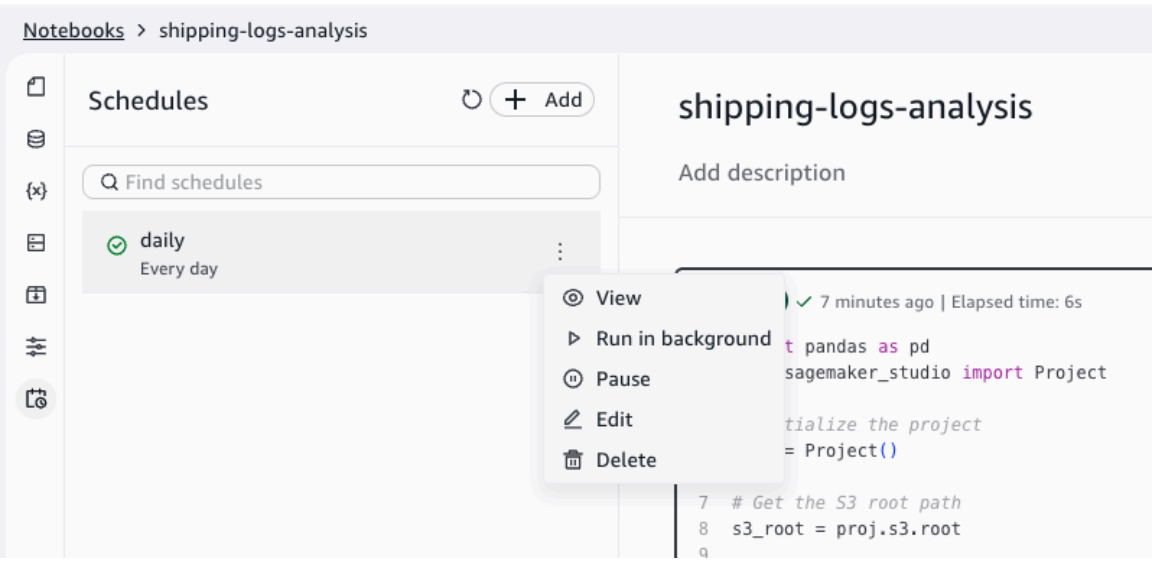
To pause or resume a schedule:

1. Choose the schedule icon in the notebook toolbar.
2. Locate the schedule.
3. Choose **Pause** or **Resume** to change the schedule state.

To delete a schedule:

1. Choose the schedule icon in the notebook toolbar.
2. Locate the schedule that you want to delete.
3. Choose **Delete**.
4. Confirm the deletion.

> **Note**
>
> Deleting a schedule preserves historical run outputs in Amazon Simple Storage Service. You can still access previous run results after the schedule is deleted.

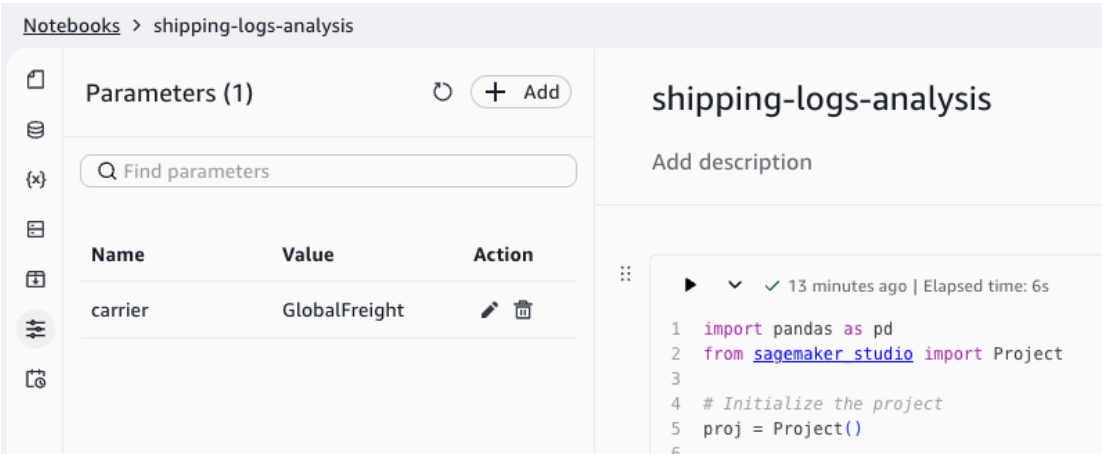


Parameterizing notebooks

You can define parameters for your notebooks to make them reusable with different input values. This allows you to create multiple schedules with different parameter values or override parameters for on-demand runs.

To define parameters:

1. Open the **Parameters** activity panel.
2. Choose **Add parameter**.
3. Enter a name and default value for the parameter.



4. Repeat for each parameter that you want to define.

To use parameters in your notebook code, call the `sagemaker_studio.nbutils.parameters.get()` function:

```
from sagemaker_studio.nbutils import parameters

# Retrieve a parameter value
carrier = parameters.get("carrier")

# Retrieving other parameter values
bucket = parameters.get("output_bucket")
```

To create schedules with different parameter values, create multiple schedules for the same notebook and specify different parameter values in each schedule configuration.

To override parameters for an on-demand run:

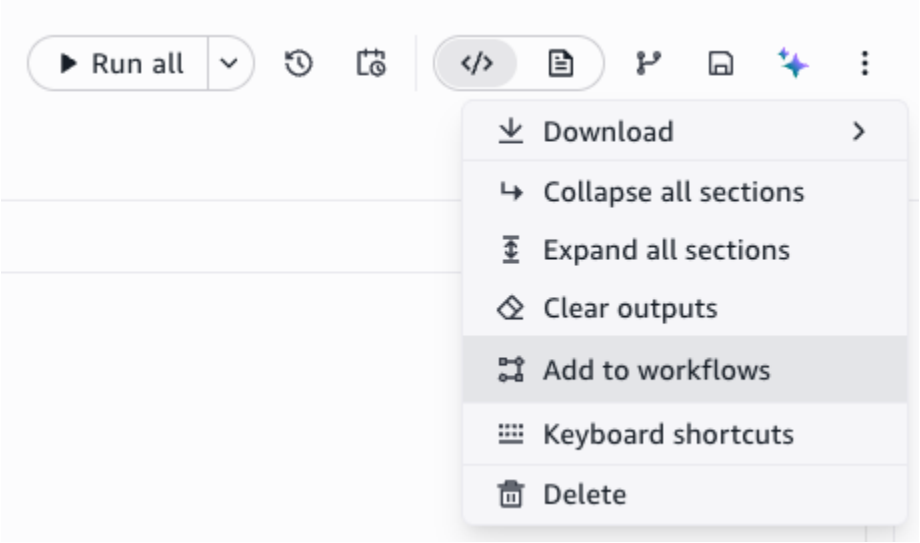
1. Choose **Run all**.
2. Choose **Run with settings**.
3. Modify the parameter values for this run.
4. Choose **Run**.

Orchestrating notebooks with workflows

You can orchestrate multi-notebook pipelines by using the Notebook Operator in the Workflows tool. This allows you to chain notebook executions, pass data between notebooks, and build complex data pipelines.

To add a notebook to a workflow:

1. Open the notebook that you want to include in a workflow.
2. Choose **Add to workflows**.



3. Select an existing workflow or create a new one.

The Notebook Operator task supports the following configuration:

Notebook selection

The notebook to execute as part of the workflow.

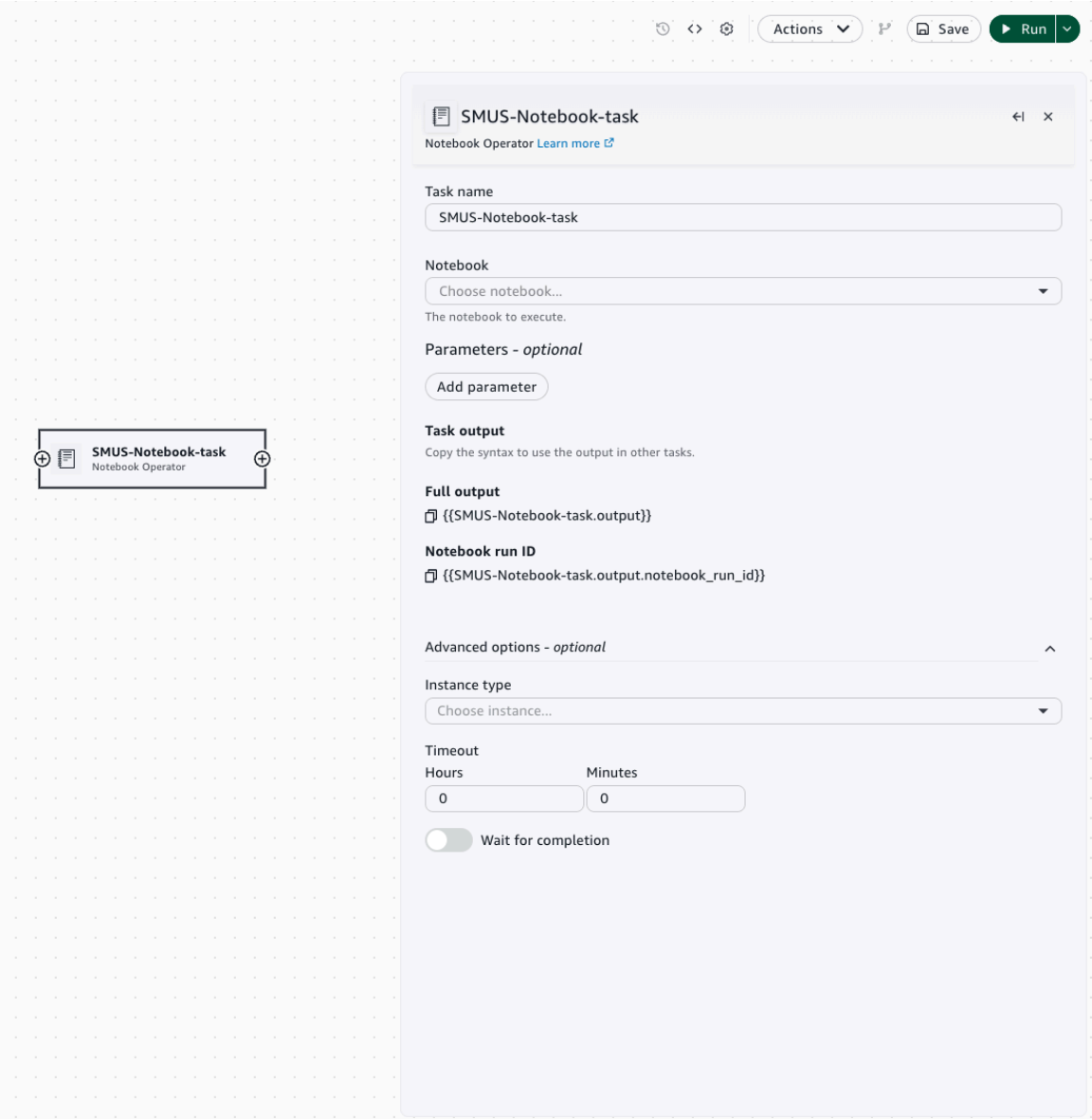
Parameters

Parameter values to pass to the notebook for this workflow step.

Compute and timeout

The compute instance type and maximum run duration for this step.

Use the **Notebook Sensor** to poll the run status of a notebook execution before proceeding to the next step in the workflow.



To pass outputs between notebooks in a workflow, use the **Variables** panel to define output variables in one notebook and reference them as inputs in downstream notebooks. You can chain multiple Notebook Operators to create sequential or parallel execution patterns.

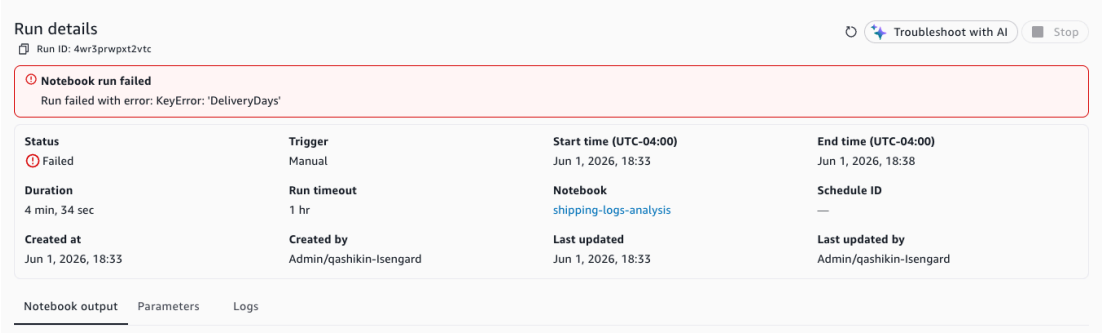
Debugging failed runs

When a scheduled or background run fails, you can identify the issue and get AI-assisted troubleshooting.

To debug a failed run:

1. Open the **Runs** activity panel.

2. Locate the failed run, indicated by a failed status.
3. Choose the run to open the failed output notebook.
4. Review the cell that caused the failure and the error message.
5. Choose **Troubleshoot with AI** to invoke the data agent. The agent analyzes the error and suggests fixes.



Use the sample notebook

You can get started using Amazon SageMaker Unified Studio by using the sample notebook in the JupyterLab IDE within your project. This getting_started.ipynb notebook provides information about using AWS Glue, Amazon Redshift, Amazon Athena, and more. This is a multi-service, poly-compute notebook, designed to enable end-to-end development in a single notebook.

In an Amazon SageMaker Unified Studio notebook, you can select the language and framework for each cell based on the compute options or connections configured in your project. You can add or modify these compute connections from the project's compute management screen. The compute choices differ based on your project's profile. However, all default profiles come with local Python, serverless Spark powered by AWS Glue, and Trino with Amazon Athena. There is a README file with additional information about the sample notebook and Amazon SageMaker Unified Studio.

You can also create new notebooks to input new code from scratch. For more information about using the JupyterLab IDE in Amazon SageMaker Unified Studio, see the section called "JupyterLab".

To navigate to the sample notebook, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to a project. To do this, choose a project from the project selector dropdown at the top of the page.

3. In the left navigation pane, under **IDEs**, choose **JupyterLab**.

Using workflows in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio provides workflow capabilities that allow you to set up and run a series of tasks to orchestrate your data processing procedures. Workflows use Apache Airflow to model data processing procedures and orchestrate your Amazon SageMaker Unified Studio code artifacts.

Amazon SageMaker Unified Studio supports MWAA serverless and MWAA provisioned workflows:

- **Serverless** - Eliminates the operational overhead of managing Apache Airflow environments while providing cost-effective serverless scaling.
- **Provisioned** - A managed environment for Apache Airflow, to set up and run data pipelines in the cloud at scale.

> **Note**
>
> IAM domains supports only serverless workflows.

To deploy workflow applications across development, test, and production environments, see *CI/CD*.

Serverless Workflows

Amazon SageMaker Unified Studio supports serverless workflows powered by Amazon MWAA Serverless. With serverless workflows, you can orchestrate data processing tasks without provisioning or managing Apache Airflow infrastructure. Resources are automatically provisioned when a workflow runs and released when it completes, so you only pay for actual workflow run time.

Serverless workflows in Amazon SageMaker Unified Studio provide the following capabilities:

- Automatic scaling of compute resources to meet workflow demands without manual intervention.
- Workflow isolation, where each workflow runs with its own execution role and worker, providing enhanced security and preventing cross-workflow interference.

- Cost optimization through a pay-per-use model with no upfront costs or minimum provisioning required.
- Built-in integration with AWS services including Amazon S3, Amazon Redshift, Amazon EMR, AWS Glue, and Amazon SageMaker AI.
- YAML-based workflow definition files.

Using serverless visual workflows

With Amazon SageMaker Unified Studio serverless visual workflows, you can create and orchestrate tasks using an intuitive drag-and-drop interface without writing code. Visual workflows supports over 80 tasks to help interact with multiple AWS services and automate use-cases across analytics, compute, catalog and storage. For the full list of supported tasks, see Supported operators.

- the section called “Create a visual workflow”
- the section called “View workflow code”
- the section called “Monitor workflow”
- the section called “Convert Airflow DAGs”
- the section called “Import SageMaker Pipelines”

Create a serverless visual workflow

Use visual workflows to orchestrate tasks in your project. With visual workflows, you can define a collection of tasks organized as a directed acyclic graph (DAG) that can run on a user-defined schedule.

To create a visual workflow

1. Log in to Amazon SageMaker Unified Studio.
2. In the left navigation pane, choose **Workflows**.
3. Choose **Create new workflow** to open the Visual Workflows editor.
4. Provide a name to your workflow and choose **Save**.
5. In the Find tasks search window under Add tasks, choose a task to add to your workflow. The selected task appears in the canvas.
6. Configure the task by giving it a name and editing the prepopulated fields.

7. Choose the **+** symbol to add more tasks. You can drag the tasks to fit your workflow.
8. Complete the workflow by connecting the tasks. To connect the tasks, choose the **+** symbol of one task to the **+** symbol of another task. The arrows represent the execution order and data flow.
9. Once you've created your workflow, you can configure its settings. Choose the settings gear.

    a. In the Workflow settings tab you can:

    - Edit the Workflow name if the workflow has never been saved to a project.
    - Provide an optional description to the workflow.
    - Toggle the Run on schedule button and set the Schedule status to Active or Paused.
    - Choose an option from the Schedule dropdown menu to set a schedule for your workflow or specify a CRON expression in the Start date and time in UTC and End date and time in UTC fields below.

    b. Once the settings are set, choose **Apply** to save them.

    c. In the Default parameters tab, choose **Add parameter** and provide a name and a default value to the parameter and choose **Apply** to save them.

    d. In the Tags tab, choose **Add tag** to create an airflow tag to your workflow and provide a name to the tag, then choose **Apply** to save it. Airflow tags help in filtering the workflows. This step is optional.
10. Choose **Save** to save the current workflow. If there are any validation errors, the notifications symbol next to the settings gear will show a number next to it which indicates the number of errors. You must fix them before you can successfully run the workflow.

View visual workflows code

To view a visual workflow code, navigate to the workflow details page by selecting a workflow from the Workflows page list. Then choose the Actions dropdown menu and choose **View code**.

Monitor your workflow

To monitor your workflow

1. From the Workflows view, choose the vertical dots to the far right of your workflow's name and select **View runs**.
2. In the subsequent Runs view you will see your workflow runs.

3. Choose the run to show the tasks.
4. Choose the task ID to show the task output and associated logs.

Converting existing Airflow DAGs

You can convert existing Airflow workflows to YAML through a Python library. For more information, see Introducing Amazon MWAA Serverless on the AWS Big Data Blog.

Import SageMaker Pipelines

You can import existing SageMaker pipelines into Amazon SageMaker Unified Studio and automatically convert them to Serverless Apache Airflow workflows. This is particularly beneficial for teams looking to migrate existing ML pipelines to Serverless Apache Airflow without rebuilding from scratch.

> **Important**
>
> This feature is only available for IAM domains.

To import a SageMaker pipeline

1. On the Workflows listing page, choose **Create Workflow**.
2. Select **Import SageMaker Pipeline** from the dropdown.
3. In the side panel, select the pipeline you want to import from the list of SageMaker pipelines in your account.
4. Choose **Import**.

Amazon SageMaker Unified Studio converts the SageMaker pipeline definition into a Serverless Apache Airflow workflow YAML and creates an interactive visual workflow DAG. The resulting workflow opens in the visual workflows editor, where you can continue to edit, rearrange, and add tasks using the drag-and-drop interface. This means you can use the imported pipeline as a starting point and customize it further to fit your needs.

> **Note**
>
> The following limitations apply when importing SageMaker pipelines:

- A model evaluation step followed by a condition step is not currently supported.
- Only the latest version of the pipeline is imported. Older versions are not included.
- Associated schedules are not imported.
- The import creates a copy of the pipeline definition. Changes to the original SageMaker pipeline are not reflected in the imported workflow.

Provisioned Workflows

Amazon SageMaker Unified Studio supports provisioned workflows powered by Amazon MWAA. With provisioned workflows, you can create, schedule, and monitor workflows using Apache Airflow and Python without managing the underlying infrastructure for scalability, availability, and security.

Provisioned workflows in Amazon SageMaker Unified Studio provide the following capabilities:

- Automatic scaling of Apache Airflow workers to meet workflow demands, up to the maximum limits you define.
- Python DAG support with AWS and custom operators for orchestrating notebooks, querybooks, and data processing jobs.
- Workflow monitoring through Apache Airflow logs and metrics in Amazon CloudWatch.
- Built-in access to AWS services through open source Apache Airflow operators.
- Direct access to the Apache Airflow web interface for workflow management and monitoring.

Note

Provisioned workflows are available in Amazon SageMaker Unified Studio projects created with the **All capabilities** project profile.

Provisioned workflows run in a shared environment that all project members can access. To share your workflows with other users, commit the file defining your workflow and sync the workflow with the shared environment.

Workflow environments in Amazon SageMaker Unified Studio

Use a shared workflow environment to share workflows with other project members. You have two options for setting up a workflow environment in your project:

- **Create a new workflow environment** – Create a new Amazon MWAA environment that Amazon SageMaker Unified Studio provisions and manages for you. Use this option when you do not have an existing Amazon MWAA environment.
- **Connect an existing Amazon Managed Workflows for Apache Airflow (Amazon MWAA) environment** – Connect an Amazon MWAA environment that you already manage in your AWS account. Use this option when you want to use an existing environment with Amazon SageMaker Unified Studio workflows.

To update or delete a workflow environment, you must be an owner of the project that the workflow environment is in. After a project owner creates or connects a workflow environment, any project member can sync their files to share them in the environment.

- the section called “Create a workflow environment”
- the section called “Connect an existing Amazon MWAA environment”
- the section called “Update a workflow environment”
- the section called “Delete a workflow environment”

Create a workflow environment

To create a workflow environment, you must be an owner of the project that you want to create a workflow environment in.

To create a workflow environment, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to a project that was created with the **Data analytics and AI-ML model development** project profile. To do this, choose a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, choose **Compute**.

4. On the **Workflow environments** tab, confirm that there are no workflow environments in the project yet. Then choose **Create**.
5. In the **Create workflow environment** window, review the parameters of the workflow environment. These are determined by your admin. If you want any of these parameters to change, contact your admin.
6. Choose **Create workflow environment**.

> **Note**
>
> Workflow environment creation takes several minutes to complete.

Connect an existing Amazon MWAA environment

You can connect an existing Amazon Managed Workflows for Apache Airflow (Amazon MWAA) environment to your Amazon SageMaker Unified Studio project. Use your existing MWAA environment to run and monitor workflows directly from Amazon SageMaker Unified Studio.

Prerequisites

Before you connect an existing Amazon MWAA environment, make sure that you meet the following requirements:

- You are a project owner or project contributor in the project (managed in project members).
- You have an existing Amazon MWAA environment in your AWS account.
- The environment runs Airflow 2.x or later for running and monitoring workflows. To use the visual authoring experience (the **Create workflow** button in **Workflows**), Airflow 3 or later is required.
- The MWAA environment execution role has the required IAM permissions to interact with Amazon SageMaker Unified Studio. Attach the following policy to the execution role of your MWAA environment:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DataZoneAccess",
```

```
      "Effect": "Allow",
      "Action": [
        "datazone:ListEnvironmentBlueprints",
        "datazone:ListEnvironments",
        "datazone:GetEnvironment"
      ],
      "Resource": "*"
    },
    {
      "Sid": "SageMakerAndReadActions",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstanceTypes",
        "ssm:GetParameter",
        "sagemaker:AddTags",
        "sagemaker:ListTags",
        "sagemaker:CreateTrainingJob",
        "sagemaker:DescribeTrainingJob"
      ],
      "Resource": "*"
    },
    {
      "Sid": "PassRoleToSageMaker",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::<account-id>:role/<project-role-name>",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "sagemaker.amazonaws.com"
        }
      }
    }
  ]
}
```

Replace <account-id> with your AWS account ID and <project-role-name> with the name of your SageMaker Unified Studio project role.

Configure your MWAA environment for Amazon SageMaker Unified Studio

You must add Airflow configuration options to your MWAA environment so that it can connect to your Amazon SageMaker Unified Studio project.

1. Open the Amazon MWAA console and navigate to your environment.
2. Choose **Edit** to edit the environment configuration.
3. In the **Airflow configuration options** section, add the following three key-value pairs:
   - `workflows.datazone_domain_id` – The ID of your DataZone domain.
   - `workflows.datazone_domain_region` – The region of your DataZone domain (for example, `us-west-2`).
   - `workflows.datazone_project_id` – The ID of your DataZone project.
4. Choose **Save**.
5. Wait for the environment status to change to **Available** before proceeding.

Connect the environment in Amazon SageMaker Unified Studio

After you configure your MWAA environment, connect it to your Amazon SageMaker Unified Studio project.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to a project that was created with the **Data analytics and AI-ML model development** project profile. To do this, choose a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, choose **Compute**.
4. On the **Workflow environments** tab, choose **Connect**.
5. Choose the MWAA environment that you configured in the preceding steps.
6. Choose **Connect workflow environment**.

Update a workflow environment

To update a workflow environment, you must be an owner of the project that you want to update a workflow environment in.

To update a workflow environment, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.

2. Navigate to the project that contains the workflow environment that you want to update. To do this, choose a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, choose **Compute**.
4. On the **Workflow environments** tab, expand the **Actions** menu and choose **Update**.
5. Choose **Update workflow environment**.

> **Note**
>
> Updating a workflow environment takes several minutes to complete.

Delete a workflow environment

To delete a workflow environment, you must be an owner of the project that contains the workflow environment that you want to delete.

To delete a workflow environment, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project that contains the workflow environment that you want to delete. To do this, choose a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, choose **Compute**.
4. On the **Workflow environments** tab, expand the **Actions** menu and choose **Delete**.
5. Confirm the action by typing `confirm`, then choose **Delete workflow environment**.

> **Note**
>
> Deleting a workflow environment takes several minutes to complete.

Using visual workflows in Amazon SageMaker Unified Studio

With Amazon SageMaker Unified Studio visual workflows, you can create and orchestrate data processing workflows using an intuitive drag-and-drop interface without writing code. Visual

workflows enable you to connect notebooks, queries, and data processing jobs in a graphical format, and create and manage schedules.

- the section called “Create a visual workflow”
- the section called “View visual workflow details”
- the section called “Run a visual workflow”
- the section called “Edit visual workflows”
- the section called “View visual workflows code”
- the section called “Clone and Delete visual workflows”

Create a visual workflow in Amazon SageMaker Unified Studio

Use visual workflows to orchestrate data processing jobs, notebooks, and querybooks in your project repositories. With visual workflows, you can define a collection of tasks organized as a directed acyclic graph (DAG) that can run on a user-defined schedule.

Prerequisites

- Amazon SageMaker Unified Studio project created with the **All capabilities** project profile
- Access to the **Workflows** page in your project

Environment status

Use a shared workflow environment to share workflows with other project members. Workflow environments must be created by project owners. To update or delete a workflow environment, you must be an owner of the project that the workflow environment is in. After a workflow environment has been created by a project owner, any project member can sync their files to share them in the environment.

Environment Statuses

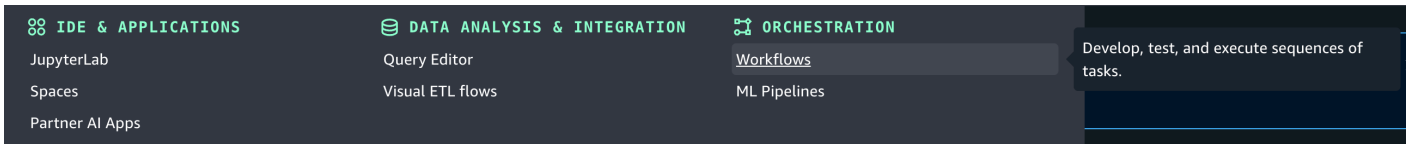
Status	Shared environment
Active	Active
Missing	Missing
Loading	Loading

Status	Shared environment
Creating	Creating
Failed	Failed

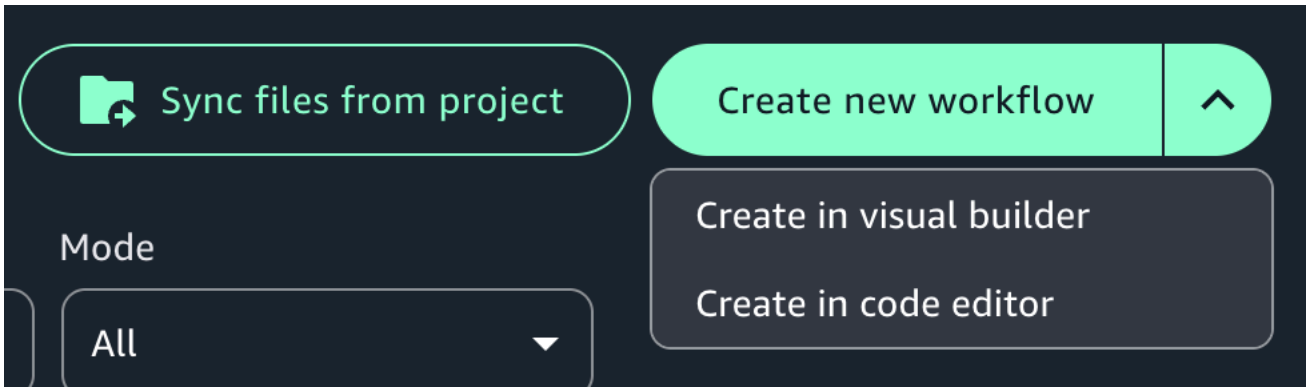
Create a workflow

To create a workflow, complete the following steps:

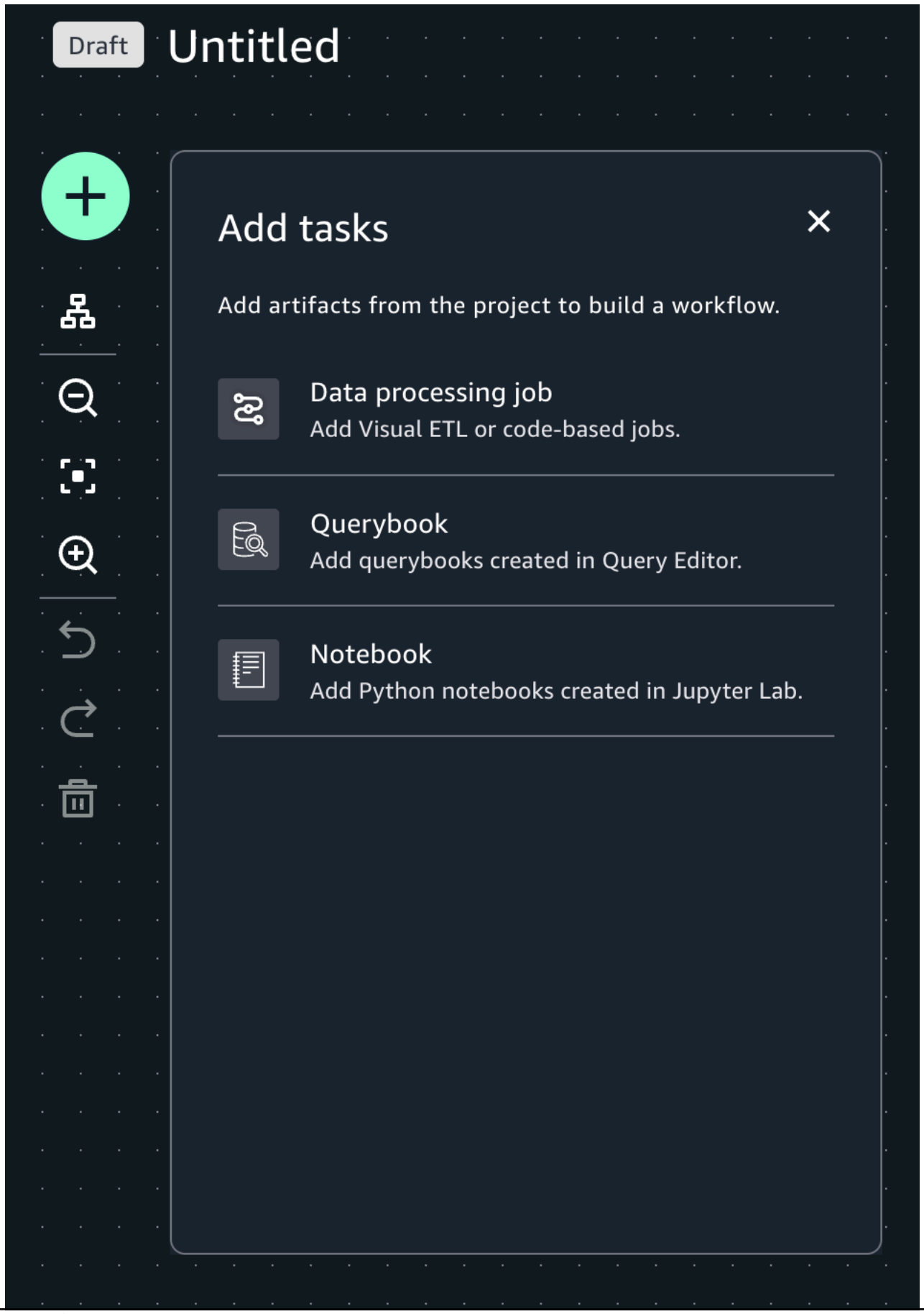
1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to a project that was created with the **All capabilities** project profile. You can do this by choosing a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, choose **Workflows**.



4. Choose the **Create new workflow** button or in the **Create new workflow** dropdown menu, choose **Create in visual builder**. This takes you to the **Visual canvas workflow**.



5. Provide a name to your workflow.
6. Choose a task from one of the three tabs: **Data processing job**, **Querybook**, or **Notebook**. The selected task appears in the canvas. Configure the task by giving it a name and editing the prepopulated fields.

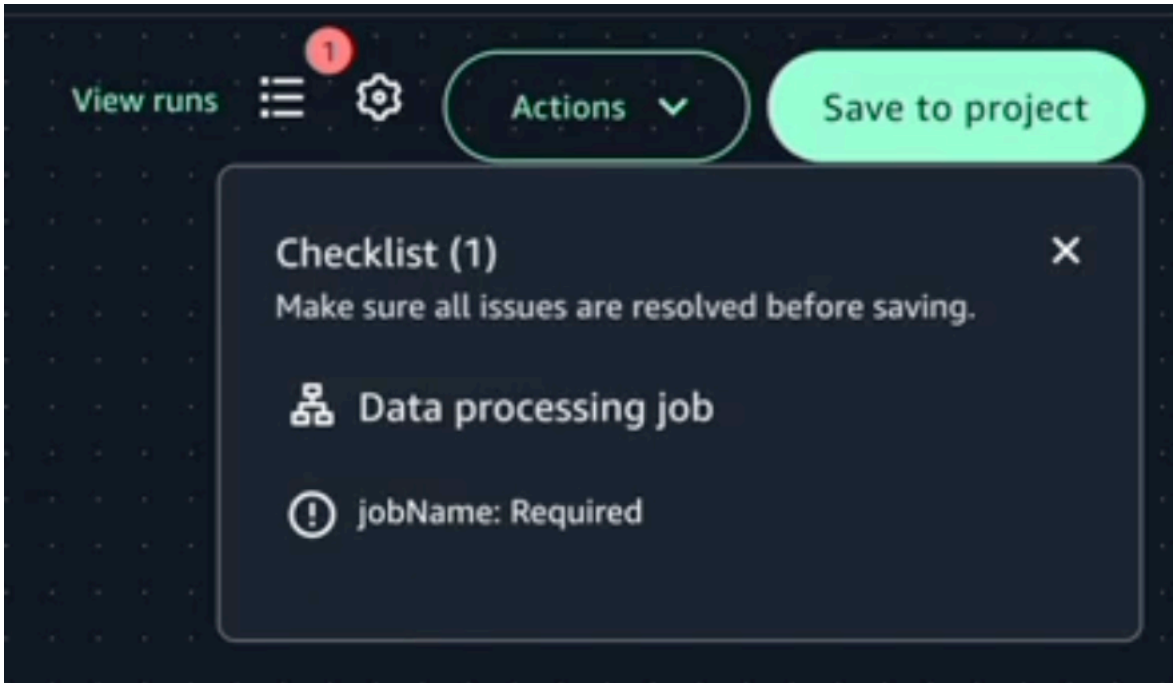

Draft
Untitled
Add tasks
Add artifacts from the project to build a workflow.
Data processing job
Add Visual ETL or code-based jobs.
Querybook
Add querybooks created in Query Editor.
Notebook
Add Python notebooks created in Jupyter Lab.

7. Choose the **Add task** icon (+) to add more tasks. You can drag the tasks to fit your workflow.
8. Complete the workflow by connecting the tasks. To connect the tasks, choose the **Add task** icon (+) of one task and connect it to the **Add task** icon (+) of another task. The arrows represent the execution order and data flow.



9. After you create your workflow, you can configure its settings. Choose the **Settings** icon.
   a. In the **Workflow settings** tab you can:
      - Edit the Workflow name if the workflow has never been saved to a project.
      - Provide an optional description to the workflow.
      - Toggle the **Run on schedule** button and set the Schedule status to Active or Paused.
      - Choose an option from the Schedule dropdown menu to set a schedule for your workflow or specify a CRON expression in the **Start date and time in UTC** and **End date and time in UTC** fields below.

      Once the settings are set, choose **Apply** to save them.
   b. In the **Default parameters** tab, choose Add parameter and provide a name and a default value to the parameter and choose Apply to save them.
   c. In the **Tags** tab, choose Add tag to create an airflow tag to your workflow and provide a name to the tag, then choose Apply to save it. Airflow tags help in filtering the workflows. This step is optional.
10. Choose **Save to project** to save the current workflow to the project. If there are any validation errors, the notifications symbol next to the settings gear will show a number next to it which indicates the number of errors. You must fix them before you can successfully save the workflow to the project.



View visual workflow details

After you create a visual workflow, it appears in a list on the Workflows page in Amazon SageMaker Unified Studio. On the Workflows page, you can see each workflow you created with the name you provided. Note that it might take up to 60 seconds for the workflow to appear in the list.

To view details about workflow runs and parameters, select the name of a workflow from the list on the Workflows page in Amazon SageMaker Unified Studio.

- Choose **View Runs** to view the results of running the workflow. You can filter to show successful runs. This page shows information about the workflow run triggers, durations, and timeframes. There is also an **Actions** column where you can choose to stop a workflow if it is still running. There is a limit of 1000 rows on the **Runs** tab for a workflow.
- To view more details about a run, choose the name of a run. This takes you to the run details panel with information about the tasks and parameters in the workflow. You can view which tasks were successfully completed. For workflows that run Python notebooks and not querybooks, you can view the output in the **Notebook output** tab. This can be useful for viewing tasks in more detail and troubleshooting if needed.
- The **Default parameters** tab shows the default parameters outlined in the workflow code. To modify the parameters, navigate to the **Default parameters** tab from the **Settings** button. For more information about parameters, see Params in the Apache Airflow documentation.

- The **Definition** tab shows the code used for the workflow.
- The **Tags** tab shows optional tags that are defined for the workflow. These are Airflow tags, not AWS tags. For more information, see Add tags to DAGs and use it for filtering in the UI in the Apache Airflow documentation.

To view details about workflow runs and parameters, select the name of a workflow from the list on the Workflows page in Amazon SageMaker Unified Studio.

Run a visual workflow

To run a workflow, select a workflow from the Workflows page list. Choose **Run**. You can then choose one of the following two options:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project that was created with the **All capabilities** project profile. To do this, choose a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, choose **Workflows**.
4. Choose the name of the workflow to navigate to the workflow canvas.
5. Expand the **Run** menu, then choose one of the following options:
   - Run with default parameters. This option starts running the workflow using the parameters already defined in the DAG file. To review these parameters, see the **Default parameters** tab.
   - Run with custom parameters. This option opens a window where you can change the inputs for the parameters defined in the DAG file. Enter the variables you want to use, and then choose **Start run** to start running the workflow.

The workflow run then appears on the side panel. The workflow runs until it is complete or until you choose to stop it.

Running a workflow puts tasks together to orchestrate Amazon SageMaker Unified Studio artifacts. You can view multiple runs for a workflow by navigating to the Workflows page and choosing the name of a workflow from the workflows list table.

If you want to see more runs, you can view them using the Airflow UI. Navigate to the Workflows page, choose the three dots in the Action column for a workflow, then choose **Open Airflow UI**. This page displays charts and graphics about the workflow.

> **Note**
>
> To open the Airflow UI, your browser should allow cross-site cookie sharing. If you receive an error message, check the cookie settings in your browser.

Edit visual workflows

To edit a visual workflow, modify the tasks and workflows in the canvas.

View visual workflows code

To view a visual workflow code, navigate to the workflow details page by selecting a workflow from the Workflows page list. Then choose the **Actions** dropdown menu and choose **View code**.

Clone and Delete visual workflows

You can also clone and delete a visual workflow. Navigate to the workflow details page by selecting a workflow from the Workflows page list. Then choose the **Actions** dropdown menu and:

- Choose **Clone workflow** to create a copy of your workflow.
- Choose **Delete workflow** to delete the workflow.

> **Note**
>
> Clone and delete workflow options are only available for visual workflows.

Code workflows in Amazon SageMaker Unified Studio

Use workflows to orchestrate notebooks, querybooks, and more in your project repositories. With workflows, you can define a collection of tasks organized as a directed acyclic graph (DAG) that can run on a user-defined schedule.

- the section called “Create a code workflow”
- the section called “View code workflow details”
- the section called “Run a code workflow”
- the section called “Share a code workflow”

Create a code workflow in Amazon SageMaker Unified Studio

Create a code workflow

To create a code workflow, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to a project that was created with the **All capabilities** project profile. You can do this by choosing a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, choose **Workflows**.
4. Choose **Create workflow in editor**. This takes you to the Code page and opens a new notebook file in the `workflows/dags` folder of the JupyterLab file navigation. The file is prepopulated with a workflow definition template.
5. Update the file as desired to create your workflow.

    a. Update `WORKFLOW_SCHEDULE` to determine when the workflow will be scheduled to run.

    b. Update `NOTEBOOK_PATH` to point to the querybook or JupyterLab notebook that you want to run. For example, `'src/querybook.sqlnb'`.

    c. Update `dag_id` with an ID that you can identify later.

    d. Add tags and parameters, if desired. For more information, see Params in the Apache Airflow documentation.

When you create a workflow, you are modifying the directed acyclic graph (DAG) within the Python file. A DAG defines a collection of tasks with their dependencies and relationships to show how they should run.

A DAG consists of the following:

- A DAG definition. The DAG ID will also be the name of the workflow.
- Operators that describe how to run the DAG and the tasks to run.
- Operator relationships that describe the order in which to run the tasks.

For more information about DAGs, see DAGs in the Apache Airflow documentation. You can configure a DAG to run on a schedule or run it manually.

You can include multiple DAGs to create multiple workflows. When you have included the DAGs you want to use, save the file in the `workflows/dag` folder in JupyterLab. There might be a slight delay before the workflow appears on the Workflows page.

Sample code workflow

The following example shows a sample code workflow definition.

```
from airflow.decorators import dag
from airflow.utils.dates import days_ago
from workflows.airflow.providers.amazon.aws.operators.sagemaker_workflows \
    import NotebookOperator

########################################################################
#
# Enter in your desired schedule as WORKFLOW_SCHEDULE.  Some options include:
#
# '@daily' (daily at midnight)
# '@hourly' (every hour, at the top of the hour)
# '30 */3 * * *' (a CRON string, run at minute 30 past every 3rd hour)
# '0 8 * * 1-5' (a CRON string, run every weekday at 8am)
#
########################################################################

WORKFLOW_SCHEDULE = '@monthly'

########################################################################
#
# Enter in the path to your artifacts. Example:
# 'src/example_notebook.ipynb'
#
########################################################################

PROCESS_PATH = 'src/dataflows/airQualityToLakehouse.vetl'
QUERY_PATH = 'src/QueryBrooklynDataPutInS3.sqlnb'

default_args = {
    'owner': 'alexa',
}


@dag(
    dag_id='air-quality-process-and-query',
```

```
    default_args=default_args,
    schedule_interval=WORKFLOW_SCHEDULE,
    start_date=days_ago(2),
    is_paused_upon_creation=False,
    tags=['example-project', 'alexa'],
    catchup=False
)
def air_quality():
    def process_data():
        return NotebookOperator(
               task_id="process-data",
               input_config={'input_path': PROCESS_PATH, 'input_params': {}},
               output_config={'output_formats': ['NOTEBOOK']},
               wait_for_completion=True,
               poll_interval=5
        )

    def query_data():
        return NotebookOperator(
               task_id="query-data",
               input_config={'input_path': QUERY_PATH, 'input_params': {}},
               output_config={'output_formats': ['NOTEBOOK']},
               wait_for_completion=True,
               poll_interval=5
        )

    process_data() >> query_data()


air_quality = air_quality()
```

View code workflow details

After you create a code workflow, you must share it with the project before it appears on the Workflows page. For more information, see the section called “Share a code workflow”. On the Workflows page, all project members can see each shared workflow by the name defined using the DAG ID.

To view details about workflow runs and parameters, select the name of a workflow from the list on the Workflows page in Amazon SageMaker Unified Studio.

- On the **Runs** tab, you can view the results of running the workflow. You can filter to show successful runs. This page shows information about the workflow run triggers, durations, and

timeframes. There is also an **Actions** column where you can choose to stop a workflow if it is still running. There is a limit of 1000 rows on the **Runs** tab for a workflow.

To view more details about a run, choose the name of a run. This takes you to the run details page, with information about the tasks and parameters in the workflow. You can view which tasks were successfully completed on the **Task log** tab. For workflows that run Python notebooks and not querybooks, you can view the output in the **Notebook output** tab. This can be useful for viewing tasks in more detail and troubleshooting if needed.

- The **Default parameters** tab shows the default parameters outlined in the workflow code. To modify the parameters, choose **Edit code** and edit parameters. For more information about parameters, see Params in the Apache Airflow documentation.
- The **Definition** tab shows the code used for the workflow. This matches the code you wrote on the Code page. Choose **Edit code** to navigate back to the Code page and make changes.
- The **Tags** tab shows optional tags that are defined in the workflow definition file. These are Airflow tags, not AWS tags. For more information, see Add tags to DAGs and use it for filtering in the UI in the Apache Airflow documentation.

Run a code workflow

To run a code workflow, navigate to the workflow details page by selecting a workflow from the Workflows page list. Then choose Run. You can then choose one of the following two options:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to a project that was created with the **All capabilities** project profile. To do this, choose a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, choose **Workflows**.
4. Choose the name of a workflow to navigate to the workflow details page and choose **Run**. This will execute the workflow in the shared environment, allowing all team members to access and collaborate on the execution.
5. Choose the name of a workflow to navigate to the workflow details page.
6. Expand the **Run** menu, then choose one of the following options:

    - Run with default parameters. This option starts running the workflow using the parameters already defined in the DAG file. To review these parameters, see the **Default parameters** tab.

- Run with custom parameters. This option opens a window where you can change the inputs for the parameters defined in the DAG file. Enter the variables you want to use, and then choose **Start run** to start running the workflow.

The workflow run then appears on the **Runs** tab of the workflow details page. The workflow runs until it is complete or until you choose to stop it.

Running a workflow puts tasks together to orchestrate Amazon SageMaker Unified Studio artifacts. You can view multiple runs for a workflow by navigating to the Workflows page and choosing the name of a workflow from the workflows list table.

If you want to see more runs, you can view them using the Airflow UI. Navigate to the Workflows page, choose the three dots in the Action column for a workflow, then choose **Open Airflow UI**. This page displays charts and graphics about the workflow.

> **Note**
>
> To open the Airflow UI, your browser should allow cross-site cookie sharing. If you receive an error message, check the cookie settings in your browser.

Share a code workflow with other project members in an Amazon SageMaker Unified Studio workflow environment

For Git storage: After a workflow environment has been created by a project owner, any project member can sync their files to share them in the environment. After you sync your files, all project members can view the workflows you have added in the workflow environment. Files that are not synced can only be viewed by the project member that created them.

For S3 storage: After a workflow environment has been created by a project owner, and once you've saved your workflows DAG files in JupyterLab, they are automatically synced to the project. After the files are synced, all project members can view the workflows you have added in the workflow environment. Files that are not synced can only be viewed by the project member that created them.

To share your workflows with other project members in a workflow environment, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to a project that was created with the **Data analytics and AI-ML model development** project profile. You can do this by choosing a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, under **IDEs**, choose **JupyterLab**.
4. Locate the workflow you want to share in the `workflows/dags` folder.
5. Choose the **Git** icon in the left navigation.
6. Choose the **+** icon next to the files you want to commit.
7. Enter a brief summary of the commit in the **Summary** text entry field.
8. (Optional) Enter a longer description of the commit in the **Description** text entry field.
9. Choose **Commit**.
10. Choose the **Push committed changes** icon to do a git push.
11. In the left navigation pane, choose **Workflows**.
12. On the **Shared environment** tab, choose **Sync files from project**.
13. Choose **Confirm**.

Query Editor

Use the query editor in Amazon SageMaker Unified Studio to write SQL queries, run them against your data sources, and visualize results, all within your project. You can work with multiple query engines, browse your data catalog, generate SQL using natural language, and schedule queries to run automatically.

Supported query engines

The query editor supports the following engines. You can switch between engines within the same querybook.

Engine	Description	SQL reference
Amazon Athena	Serverless query engine for data stored in Amazon S3 and AWS Glue Data Catalog. Best for ad-hoc queries against lakehouse data.	Athena SQL reference
Amazon Redshift	Data warehouse engine for structured and semi-structured data. Best for complex analytical queries and joins across large datasets.	Redshift SQL reference

Write, run, and view query results

A querybook is a SQL notebook where you write, organize, and run queries. Each querybook can contain multiple SQL cells and markdown cells, and you can use different query engines across querybooks within the same project.

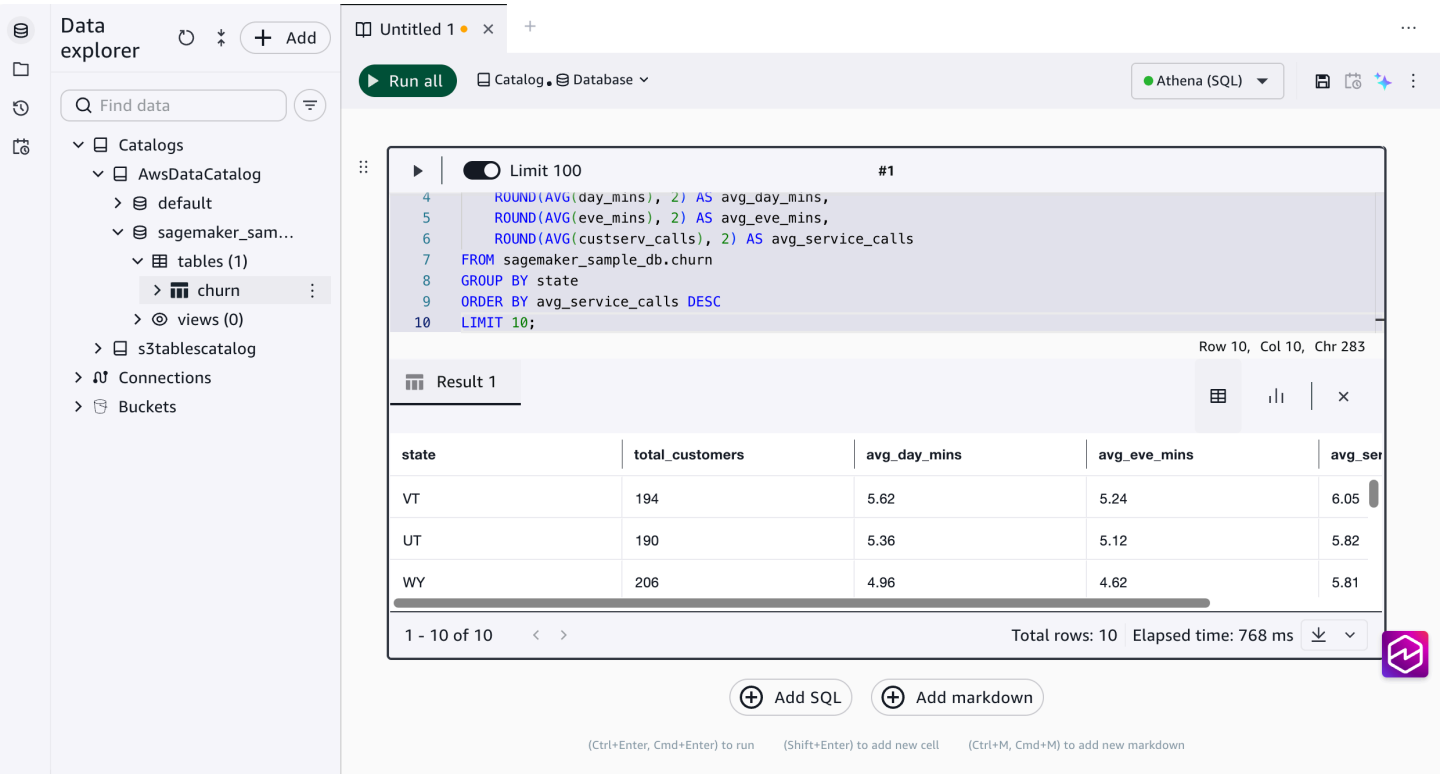
Create a querybook

To create a new querybook, choose the **+** icon at the top of the editor window. This opens a new tab with an empty SQL cell.

Each querybook tab operates independently with its own engine selection, cells, and results.

Write a query

Type SQL directly into a SQL cell. The editor provides autocomplete suggestions for table names, column names, and SQL keywords as you type. Press **Tab** to accept a suggestion.



To add more cells to your querybook:

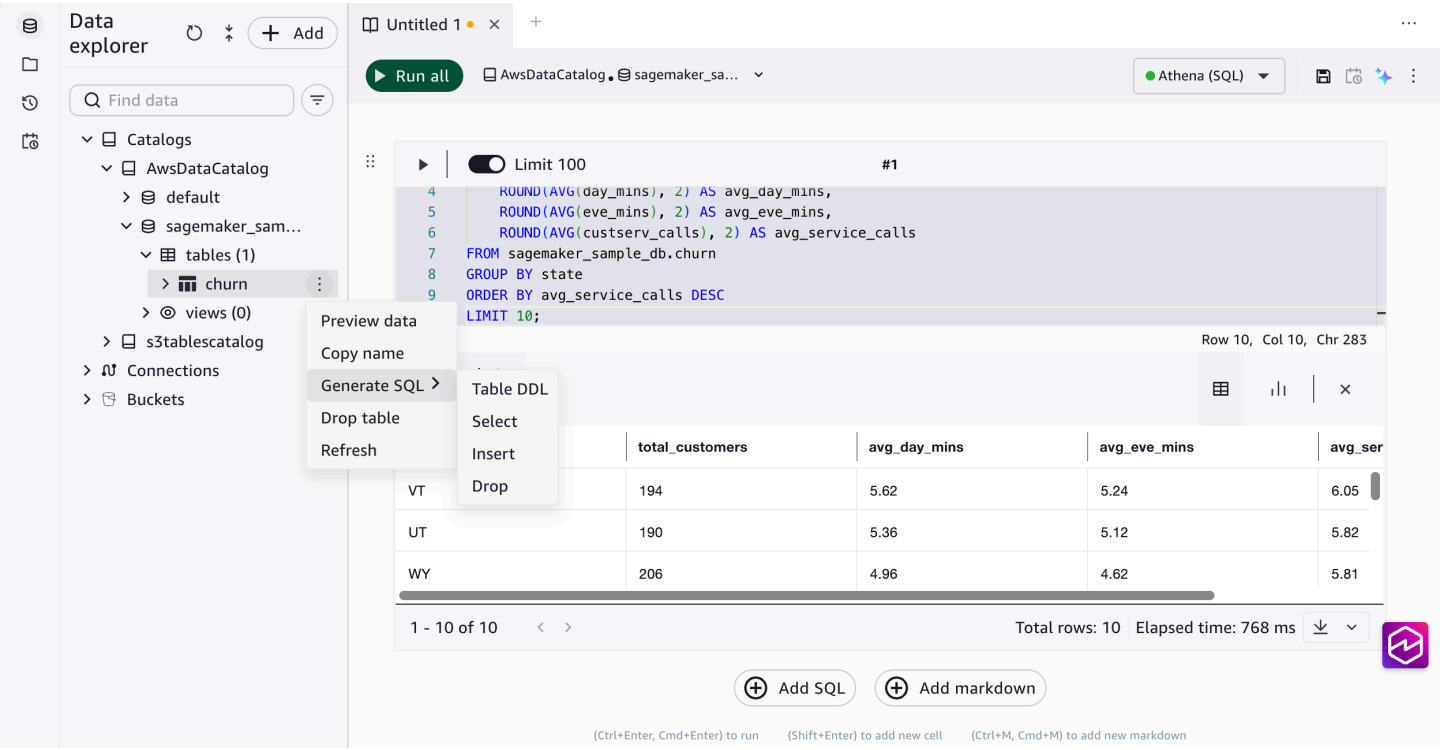
- Choose **Add SQL** to add a new SQL cell.
- Choose **Add markdown** to add a documentation cell for notes, descriptions, or context.

> **Document your analysis**
>
> Use markdown cells to add context about what each query does, why you are running it, and what the results mean. This is especially useful when sharing querybooks with project members.

Run a sample query from the data explorer

You can run a quick sample query on any table directly from the data explorer, without writing SQL manually.

1. In the data explorer, find the table you want to query.
2. Choose the three-dot menu next to the table name.
3. Choose **Generate SQL** and then **Select**.



The editor creates a new cell with a SELECT query for that table.

Run a query

There are two ways to run queries:

- **Run a single cell**: Choose the Run icon (▶) on the cell you want to run.
- **Run all cells**: Choose **Run all** at the top of the querybook. This runs every SQL cell in order.

> **Note**
>
> By default, Limit 100 is turned on. This caps query results at 100 rows. Toggle Limit 100 off on the cell toolbar if you need the full result set.

Keyboard shortcuts

Action	Windows / Linux	Mac
Run current cell	Ctrl+Enter	⌘+Enter
Run all cells	Ctrl+Shift+Enter	⌘+Shift+Enter

Stop a running query

To cancel a query that is in progress, choose the **Stop** button that appears on the cell while the query is running. The query status changes to Canceled in your query history.

Query limits

- Query results display up to 64,000 rows in the editor.
- The **Limit 100** toggle on each cell caps results at 100 rows by default. Turn it off to retrieve more rows.
- Long-running queries may time out depending on your engine configuration.

View results

When a query finishes running, the **Results** tab displays the output as a table. Each column from your query maps to a column in the results table, and rows are displayed in the order returned by the query engine.

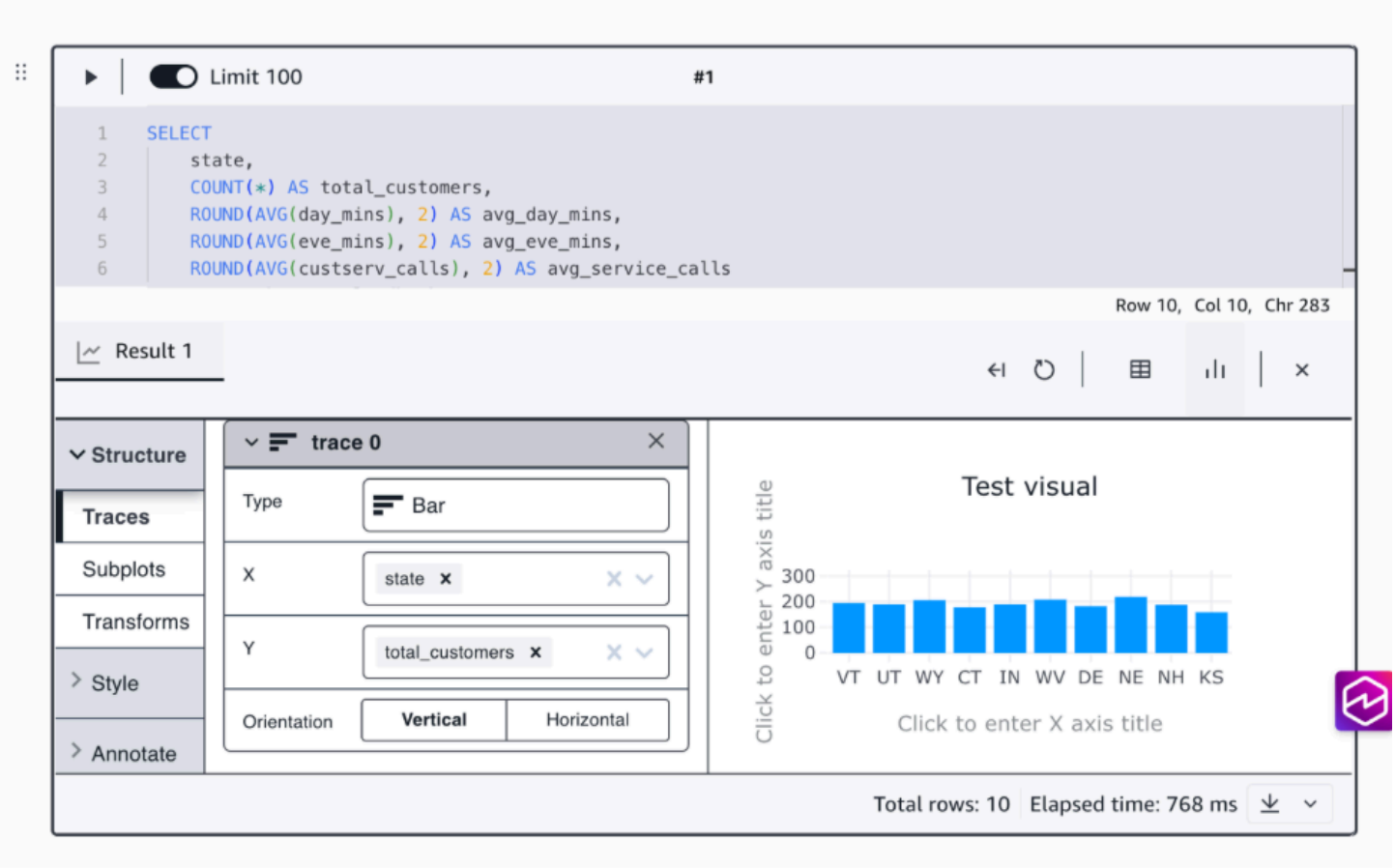
> **Note**
>
> The results table displays up to 64,000 rows. If your query returns more rows, only the first 64,000 are shown. Use a LIMIT clause in your SQL to control the number of rows returned.

Create a visualization

You can create charts directly from your query results without leaving the query editor.

1. In the **Results** tab, choose the **Chart view** icon.
2. For **Type**, choose a chart type (line, scatter, bar, area, pie, histogram, and others).

3. Choose which data columns to represent in the X and Y axis.
4. Browse the **Style** and **Annotate** options for additional chart customization.



Experiment with chart types

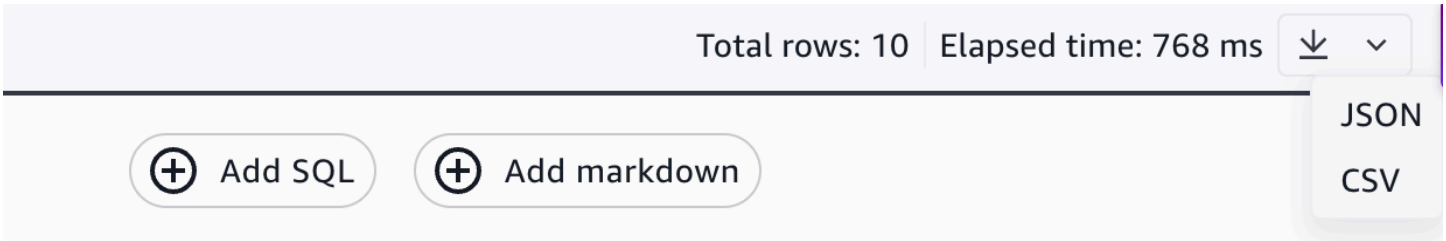
Try different chart types to find the best way to represent your data. You can switch between chart types without re-running the query.

Export query results

You can download query results for use in other tools.

1. Run your query.
2. In the **Results** tab, choose the download icon.
3. Choose the export format:

- **CSV**: Comma-separated values. Works with spreadsheet applications and data tools.
- **JSON**: JavaScript Object Notation. Works with programming languages and APIs.



Generate SQL with the Data Agent

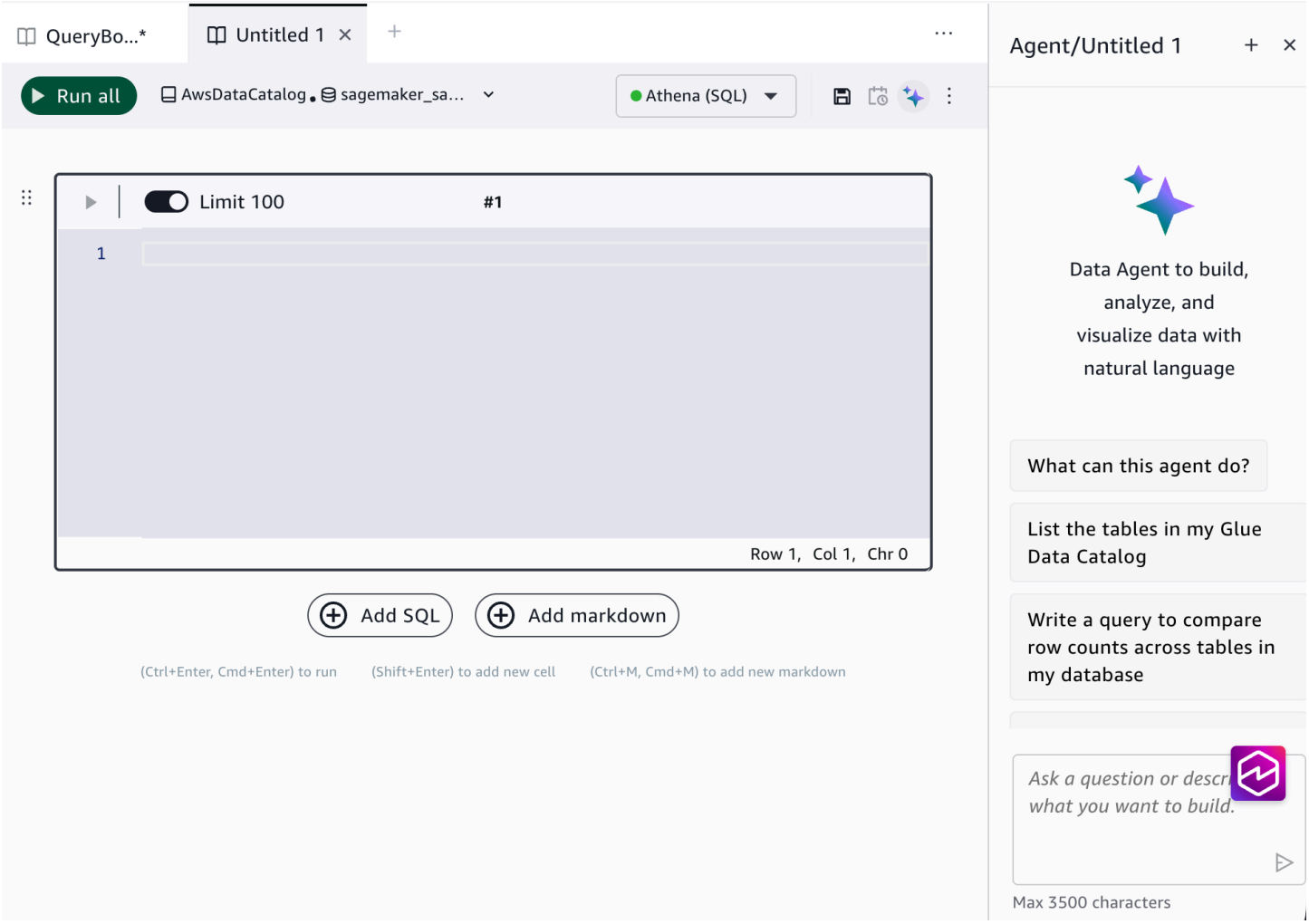
The SageMaker Data Agent is a conversational SQL assistant built into the query editor. Describe what you want to analyze in plain language, and the agent generates SQL queries based on your data catalog. The agent supports multi-turn conversations, so you can ask follow-up questions, request modifications, and get help fixing errors.

Open the Data Agent panel

1. In the query editor, choose the **Chat with AI** icon in the top-right action bar.



The agent panel opens on the right side of the editor with sample prompts and a chat input.

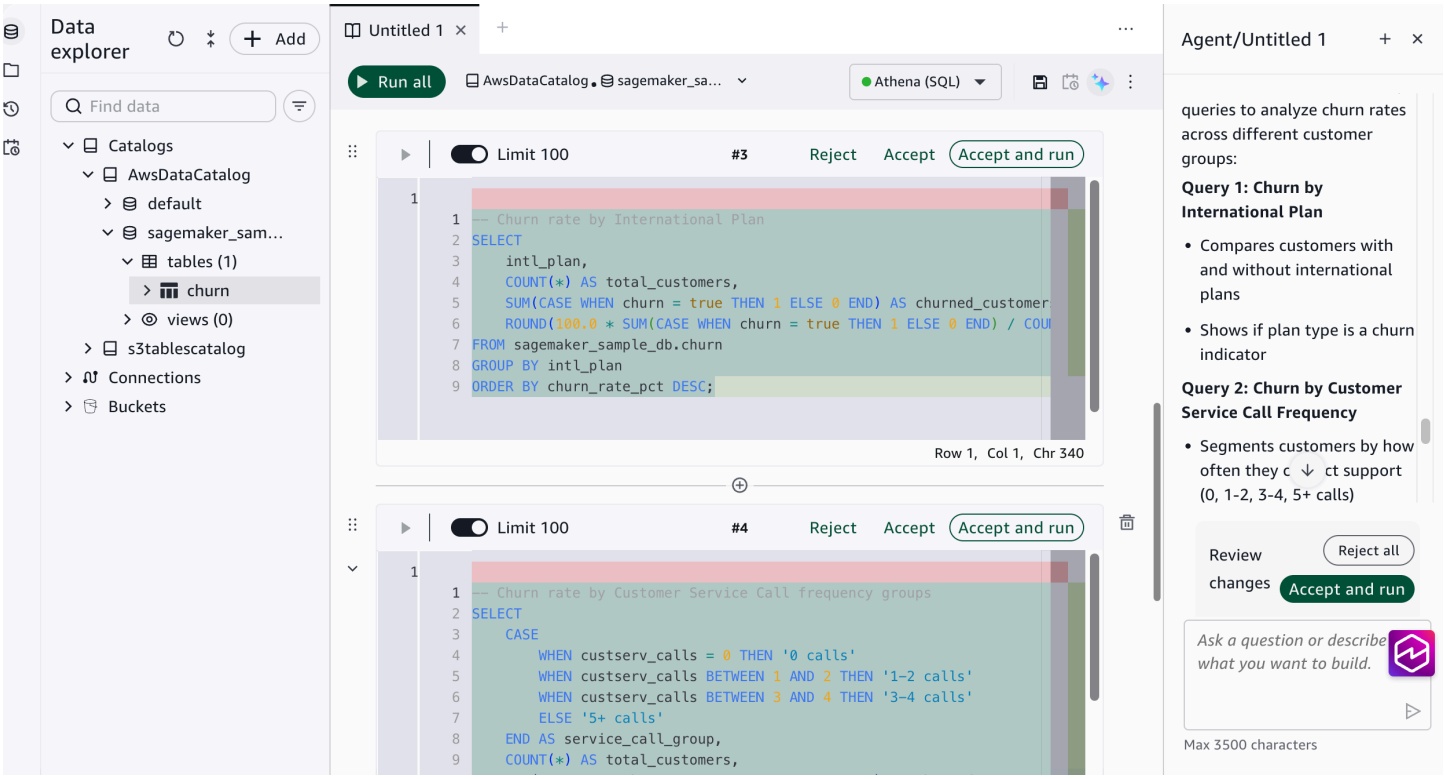


Ask a question and generate SQL

Type a natural language description of what you want to query. The agent analyzes your data catalog and generates SQL.

Example prompts:

- "What are the columns in the churn table?"
- "Which tables in the analytics schema have columns related to customer churn?"
- "Which customer groups have the highest churn?"



For complex analytical tasks, the agent proposes a step-by-step plan before generating SQL. You can review and approve the plan, or ask the agent to modify it.

Add generated SQL to your querybook

After the agent generates SQL, it automatically creates new cells in your querybook with the generated code. You can:

- **Accept** the generated SQL and run it in place.
- **Reject** the generated SQL to discard it.
- **Accept and run** to insert the SQL and run it immediately.

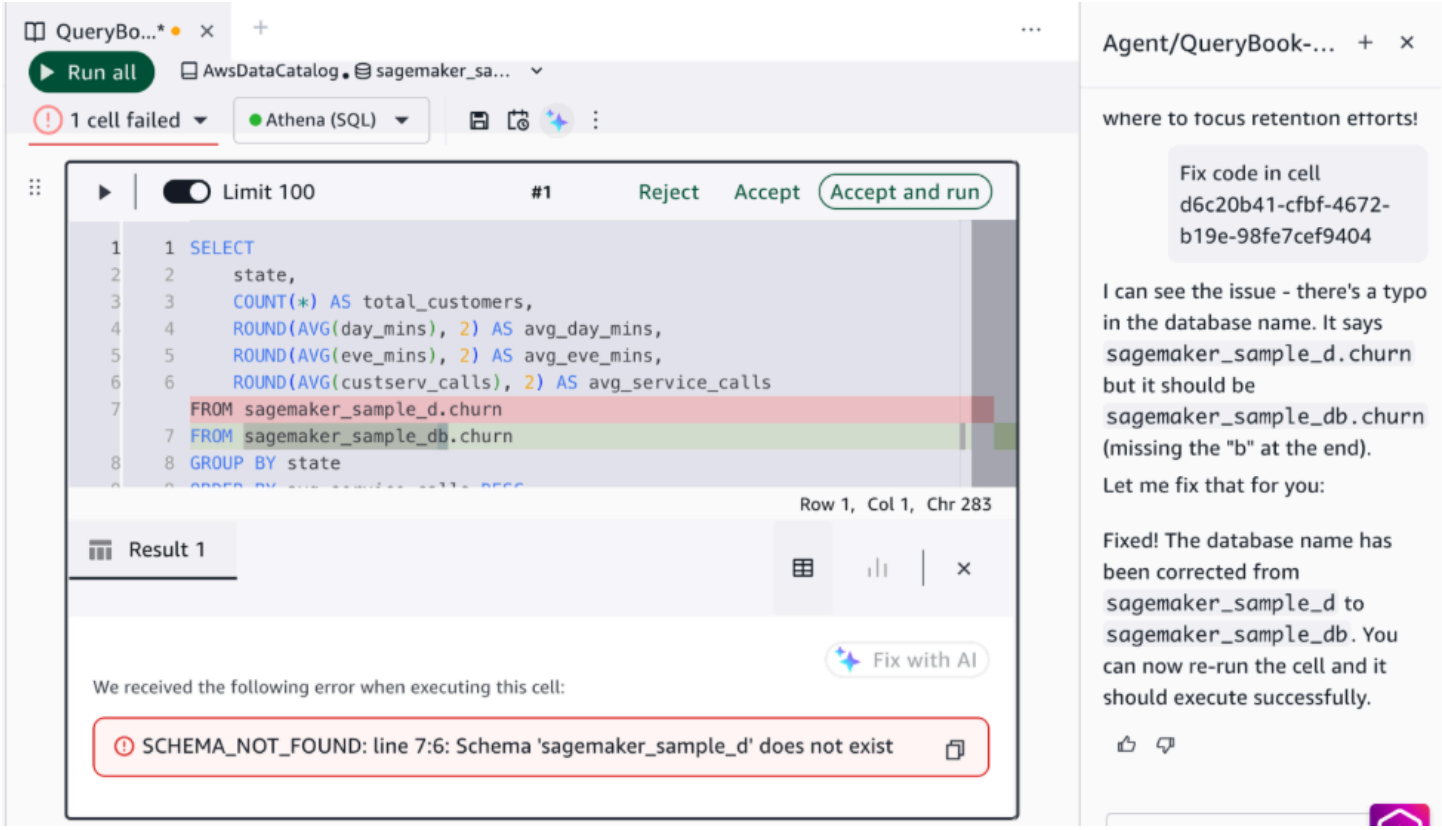
Multi-turn conversations

The Data Agent maintains context across messages within a conversation. You can build on previous queries by asking follow-up questions.

Fix errors with the Data Agent

When a query fails, the agent can analyze the error and suggest corrections.

1. Run a query that returns an error.
2. Choose **Fix with AI** on the failed cell.
3. The agent diagnoses the error and generates a corrected query.



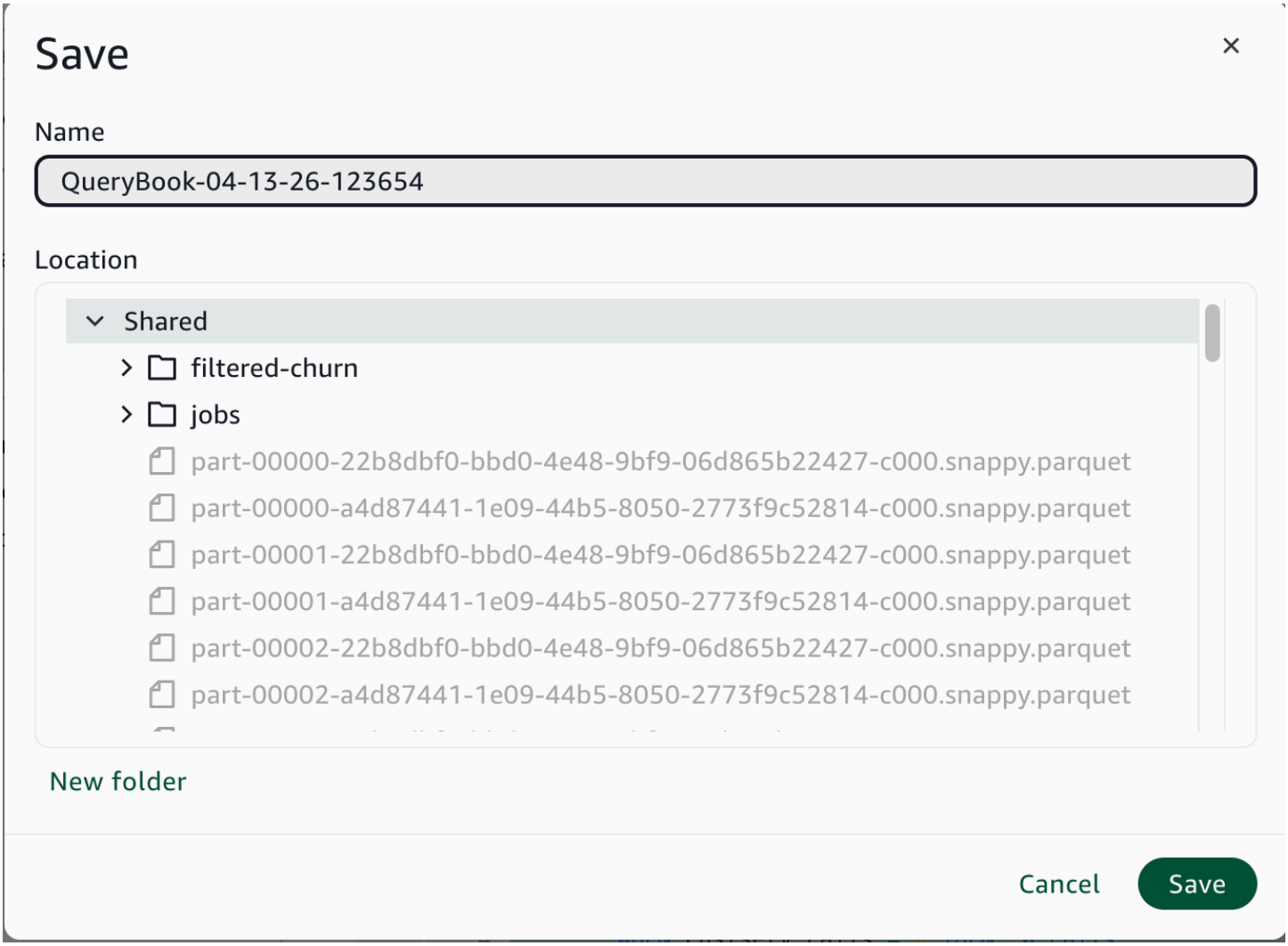
Save, schedule, and review queries

After you write and run queries, you can save your querybook to the project repository, share it with project members, create schedules to run queries automatically, and review past query executions.

Save a querybook to the project

Querybooks are saved as .sqlnb files in your project repository. Saving makes the querybook available to other project members and preserves it across sessions.

1. Choose **Save** in the top-right toolbar.
2. Enter a name for the querybook.
3. Choose **Save** to confirm.



Note

Unsaved querybooks exist only in your current session. If you close the browser without saving, your work is lost.

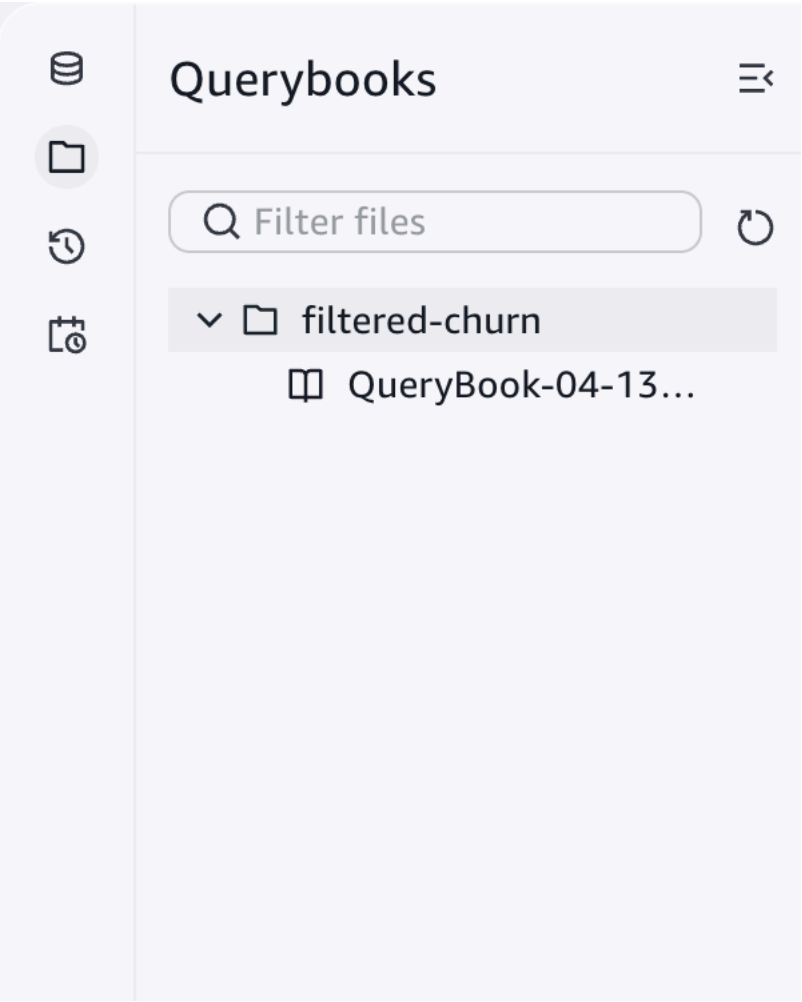
Share with project members

When you save a querybook to the project, it becomes visible to all project members in the file explorer panel. Other members can open, edit, and run the saved querybook.

To view saved querybooks:

1. Choose the **Querybooks** icon in the left panel.

2. Browse the project repository to find saved .sqlnb files.



Create a schedule

You can schedule a querybook to run automatically at specific times. There are two ways to schedule queries:

- **From the query editor**: Schedule a single querybook directly from the editor interface.
- **Using workflows**: Combine multiple elements (queries, notebooks, ETL jobs) into a single scheduled workflow. For more information, see Serverless workflows.

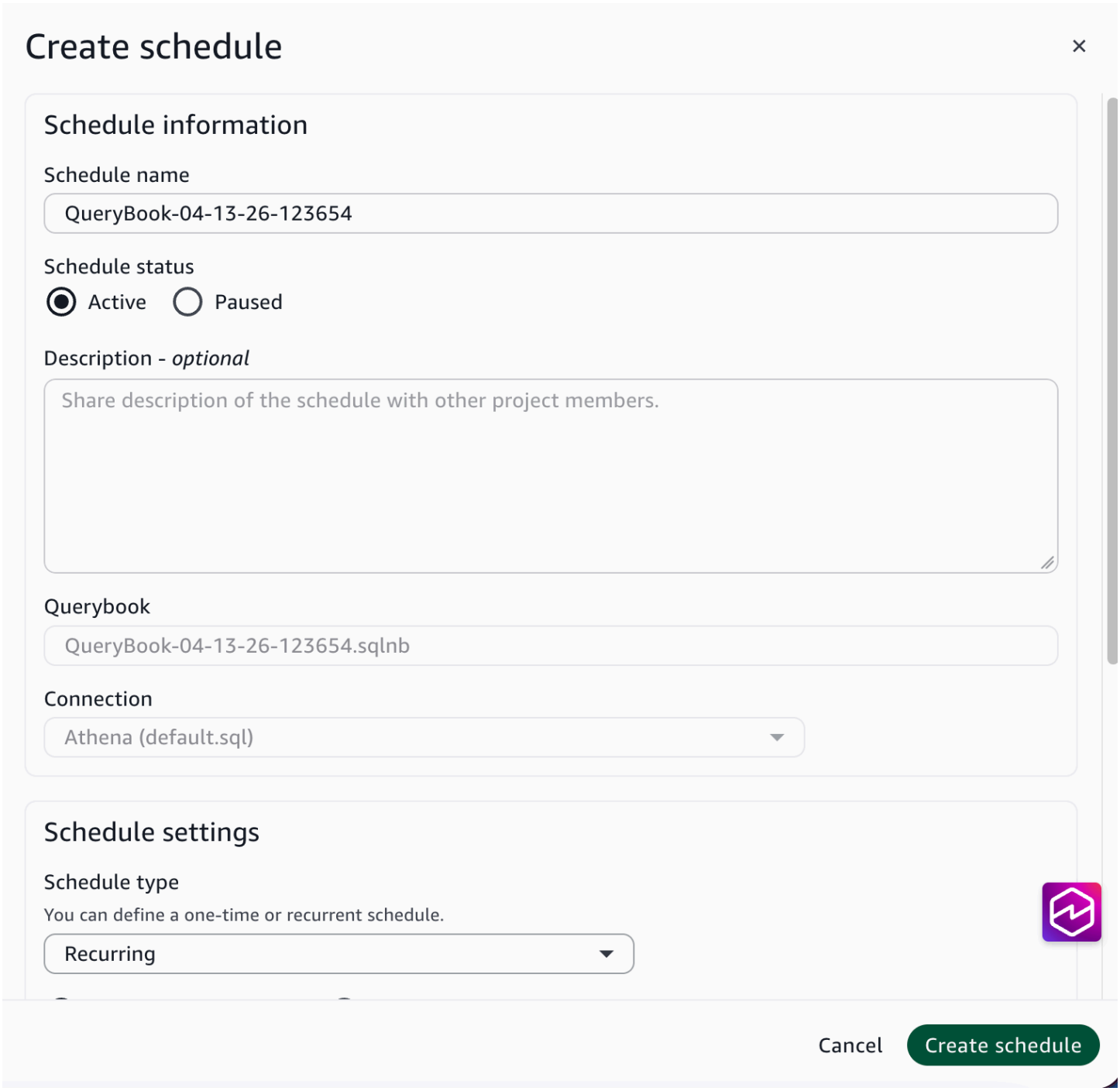
Schedule from the query editor

> **Note**
>
> You must have a data source connection selected and your querybook must be saved before you can create a schedule. Your project profile must also have scheduling enabled in the Tooling blueprint parameters.

1. Choose **Create schedule** in the top-right toolbar.



2. Under **Schedule name**, enter a name for the schedule.
3. Under **Schedule status**, choose **Active** (starts running immediately) or **Paused** (created but does not run until you activate it).
4. (Optional) Enter a description.
5. Choose a schedule type: **One-time** (runs once at a specific date and time) or **Recurring** (runs on a repeating schedule).
6. Set the days and times for the schedule to run. You can use a rate-based expression (for example, every 1 hour) or a cron expression for more complex schedules.
7. Choose **Create schedule**.

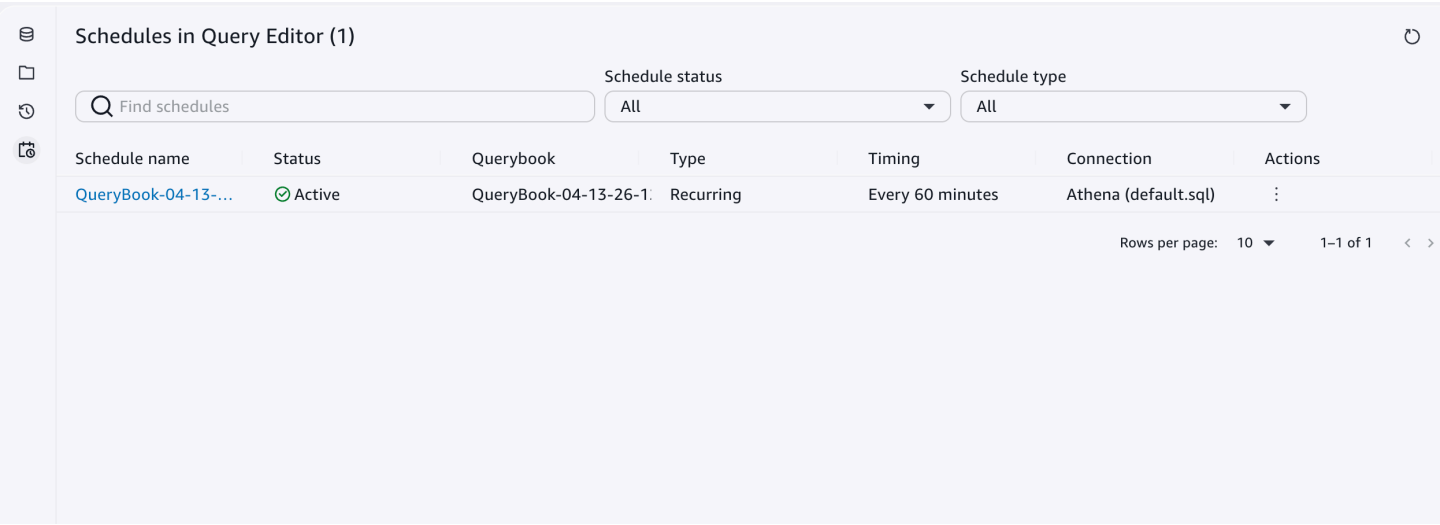


Keep schedules up to date

Enable the project repository auto sync flag when creating or updating your project. This makes scheduled queries always run the latest saved version of the querybook. Test your query in draft mode before saving to avoid running untested SQL on a schedule.

Review and manage scheduled queries

To view your scheduled queries, choose the **Scheduled queries** icon in the left panel of the query editor.



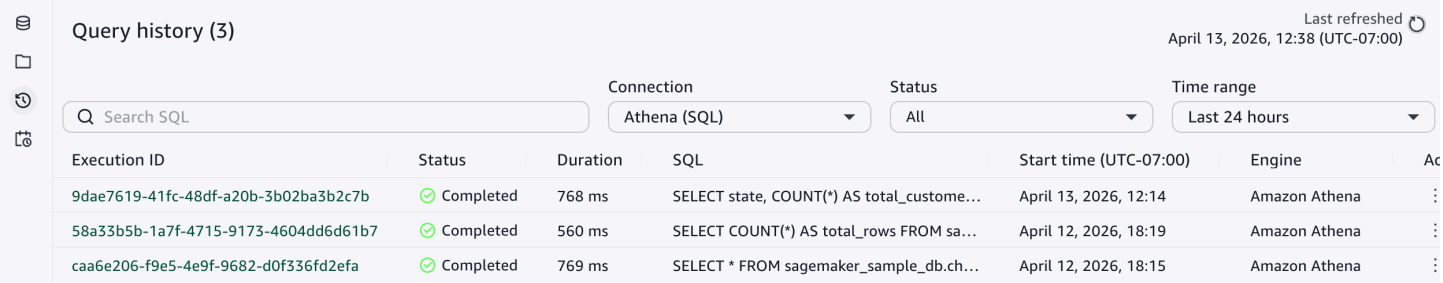
From the scheduled queries page, you can:

Action	How
Pause a schedule	Choose the three-dot menu next to the schedule and choose Pause.
Edit a schedule	Choose the three-dot menu and choose Edit.
Delete a schedule	Choose the three-dot menu and choose Delete.
View run history	Choose the schedule name to see the Runs section with execution history.
View run details	In the Runs section, choose a run name to see the querybook output and execution log.

Review query history

The query history page shows all queries you have run in the query editor, sorted with the most recent queries first. Use it to find past queries, check execution status, review errors, and re-open queries in the editor.

To open the query history page, choose the **Query history** icon in the left panel of the query editor. Then select a data source to view queries from that source.



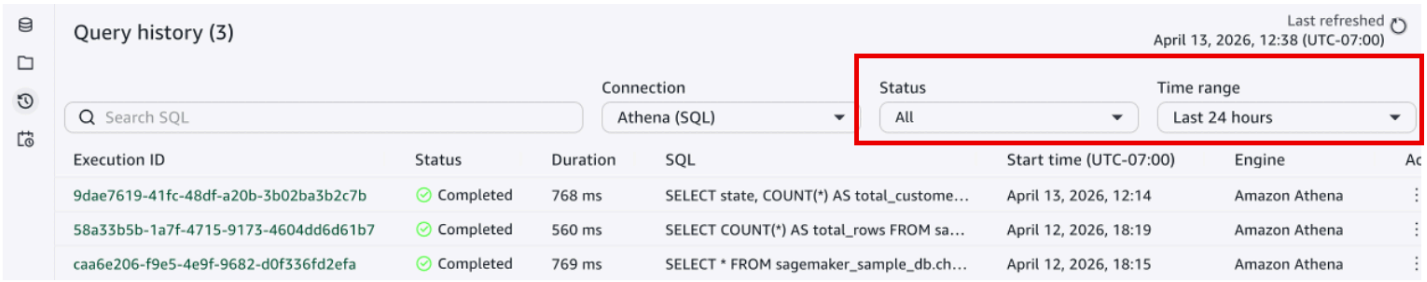
Query history columns

Column	Description
Execution ID	A unique identifier for the query execution. Choose the ID to view details.
Status	The current state of the query: Completed, Failed, Canceled, or Running.
Duration	How long the query took to run.
SQL	A preview of the SQL statement. Choose the execution ID to see the full query.
Start time	The date and time the query started running.
Engine	The query engine used (Amazon Athena or Amazon Redshift).

Filter query history

You can filter the query history by data source, status, and time range. By default, the page shows queries of all statuses from the past 24 hours.

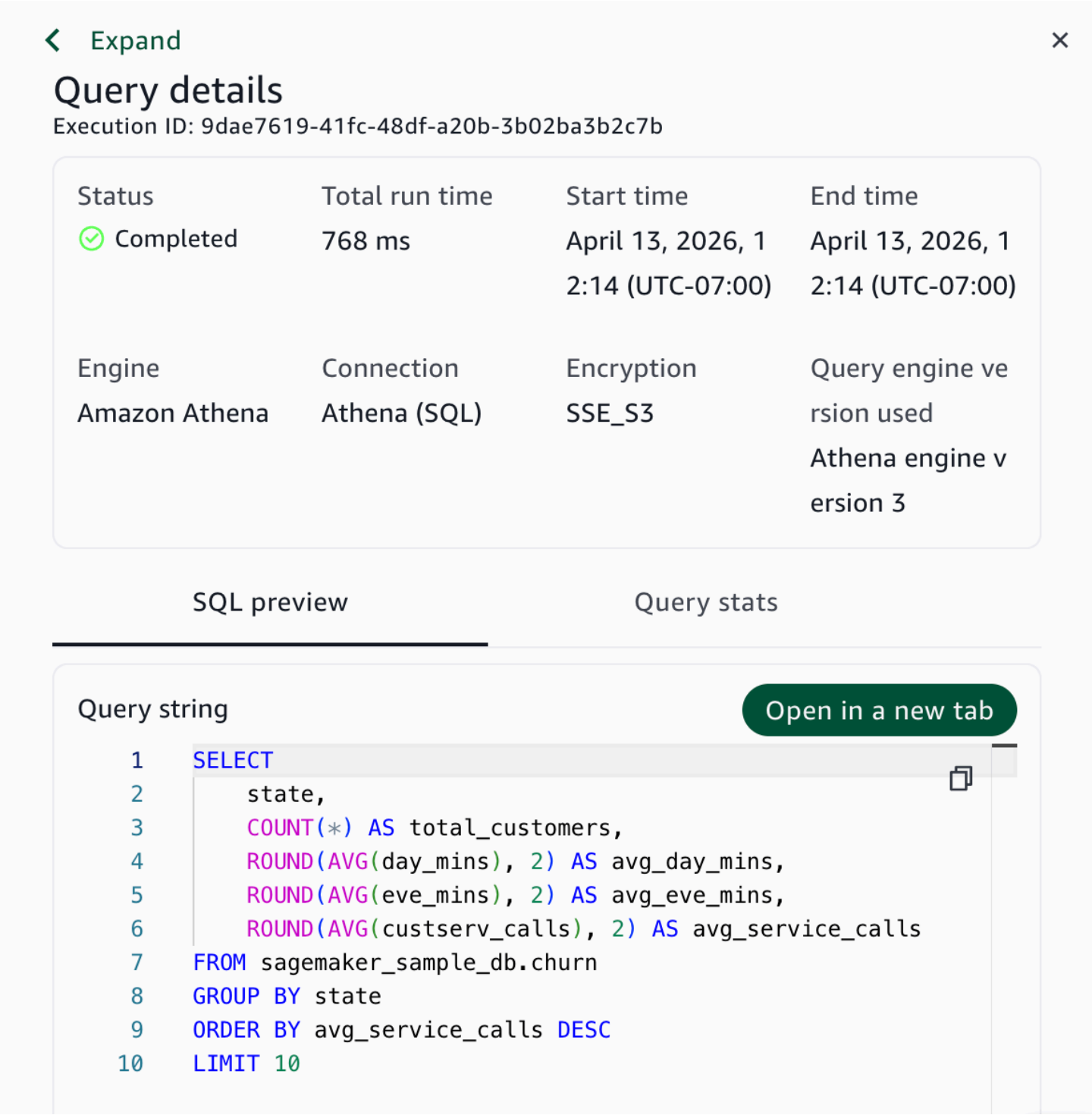
- **Filter by status**: Choose Status and select a status type (Completed, Failed, Canceled, Running), or choose All to see all statuses.
- **Filter by time range**: Choose **Time range**, select a predefined range or custom date range, and choose **Apply**.



View query details

To see the full SQL, execution statistics, and error information for a query:

1. Choose the **Execution ID** of the query you want to inspect.
2. A side panel opens with the full query details.



The details shown depend on the query engine:

- **Amazon Athena queries**: Include S3 encryption information and query stats with a visual breakdown of execution time.
- **Amazon Redshift queries**: Include cluster-specific execution details.

To re-open the query in the editor with its full results, choose **Open in a new tab**.

Visual ETL

Data engineers, analysts, and scientists use visual ETL features to create extract, transform, and load (ETL) flows using an intuitive visual interface. With visual ETL, analytics users can discover, prepare, move, and integrate data from multiple sources. This simplifies the process of data manipulation and integration so that you can prepare data for analysis and reporting.

Visual ETL in Amazon SageMaker Unified Studio provides a drag-and-drop interface for building ETL flows and authoring flows with Amazon Q. You can connect to data sources, apply transformations, and define target destinations without writing complex code.

You can use Visual ETL to implement solutions such as:

- Data integration from multiple sources
- Data cleansing and normalization
- Creating data warehouses or data lakes
- Preparing data for machine learning models
- Automating regular data processing tasks

Authoring flows with Visual ETL utilizes AWS Glue interactive sessions. You can use AWS Glue version 4.0, 5.0, or 5.1.

Amazon SageMaker Unified Studio supports Visual ETL in two domain types:

- **IAM-based domains** - New domain type where customers log into Amazon SageMaker Unified Studio using federated roles and existing IAM permissions apply when using Amazon SageMaker Unified Studio IAM-based domains. Provides access to a new Amazon SageMaker Unified Studio interface.
- **Identity Center-based domains** - Original Amazon SageMaker Unified Studio interface and user experience that continues to be supported and maintained.

While workflows are similar across both domain types, the steps to complete actions differ between the two interfaces. Choose the appropriate section below based on your domain type.

Topics

- Supported connectors for Visual ETL

- Supported transforms for Visual ETL
- Visual ETL for IAM-based domains
- Visual ETL for Identity Center-based domains
- Data preview

Supported connectors for Visual ETL

Visual ETL in Amazon SageMaker Unified Studio supports the connectors listed at Data connections in the lakehouse architecture of Amazon SageMaker, in addition to Amazon S3, Amazon S3 tables, and AWS Glue Data Catalog.

For instructions on how to add a new connection in IDC-based domains, see Adding data sources in lakehouse architecture.

Supported transforms for Visual ETL

The following section contains information on the supported transforms for Visual ETL in Amazon SageMaker Unified Studio :

Topics

- Add current timestamp transform
- Aggregate transform
- Array to columns transform
- Autobalance processing transform
- Change columns transform
- Concatenate columns transform
- Custom code transform
- Custom visual transform
- Derived column transform
- Drop columns transform
- Drop duplicates transform
- Drop nulls transform
- Explode array or map into rows transform

- Evaluate data quality transform
- Extract JSON path transform
- Fill nulls transform
- Filter transform
- Flatten transform
- Format timestamp transform
- Identifier transform
- Intersect transform
- Join transform
- Limit tranform
- Lookup transform
- Parse JSON column transform
- Pivot rows into columns transform
- Regex extract transform
- Rename columns transform
- Select columns transform
- Split string transform
- SQL query transform
- To timestamp transform
- Union transform
- Unpivot columns into rows transform
- UUID transform

Add current timestamp transform

The Add Current Timestamp transform allows you to mark the rows with the time on which the data was processed. This is useful for auditing purposes or to track latency in the data pipeline. You can add this new column as a timestamp data type or a formatted string.

To add an Add Current Timestamp transform:

1. Navigate to your Visual ETL job in Amazon SageMaker Unified Studio.

2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **Add Current Timestamp**.
4. Select the diagram to add the node to your Visual ETL job.
5. Select the node on the diagram to view details about the transform.
6. Under **Timestamp column name**, enter a custom name for the new column.
7. (Optional) under **Timestamp format**, enter a format if you would prefer the column to be a formatted date string.

Aggregate transform

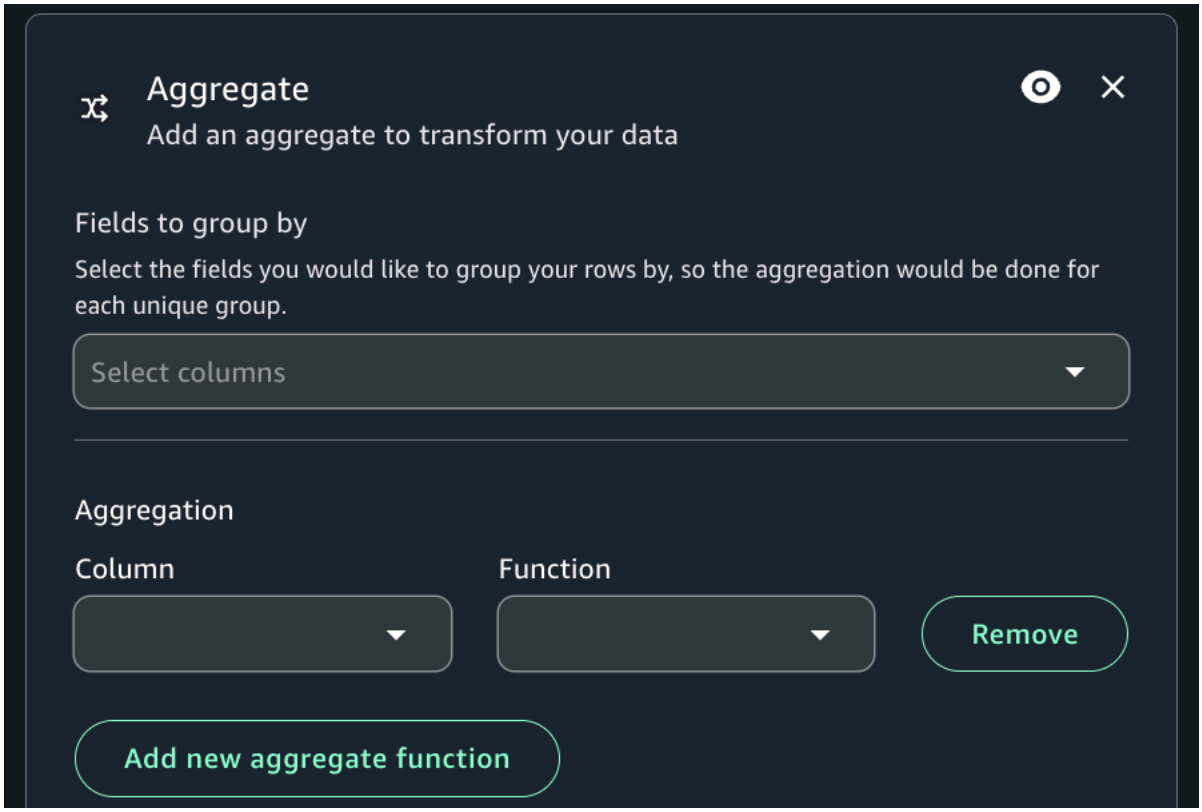
You may use the Aggregate transform to perform summary calculations on selected fields.

To use the Aggregate transform

1. Add the Aggregate node to the visual job diagram.
2. (Optional) Click on the rename node icon to enter a new name for the node in the job diagram.
3. On the Node properties view, choose the "fields to group by", selecting the drop-down field (optional). You can select more than one field at a time or search for a field name by typing in the search bar.

   When fields are selected, the name and datatype are shown. To remove a field, click 'X' on the field.
4. Choose Aggregate another column. It is required to select at least one field.
5. Choose a field in the Field to aggregate drop-down.
6. Choose the aggregation function to apply to the chosen field:

   - avg - calculates the average
   - count - calculates the number of non-null values
   - max - returns the highest value that satisfies the 'group by' criteria
   - min - returns the lowest value that satisfies the 'group by' criteria
   - sum - the sum of all values in the group



Array to columns transform

The Array To Columns transform allows you extract some or all the elements of a column of type array into new columns. The transform will fill the new columns as much as possible if the array has enough values to extract, optionally taking the elements in the positions specified.

For instance, if you have an array column “subnet”, which was the result of applying the “Split String” transform on a ip v4 subnet, you can extract the first and forth positions into new columns “first_octect” and “forth_octect”. The output of the transform in this example would be (notice the last two rows have shorter arrays than expected):

subnet	first_octect	fourth_octect
[54, 240, 197, 238]	54	238
[192, 168, 0, 1]	192	1
[192, 168]	192	

subnet	first_octect	fourth_octect
[]		

To add an Array to Columns transform:

1. Navigate to your visual ETL job in Amazon SageMaker Unified Studio.
2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **Array to Columns**.
4. Select the diagram to add the node to your visual ETL job.
5. Select the node on the diagram to view details about the transform.
6. Under **Array type column**, choose the column of type array from which the new columns are extracted.
7. Under **Output columns**, enter names for the output columns.
8. (Optional) Under **Array indexes to use**, enter numbers to indicate which columns to include.

Autobalance processing transform

Use this transform to repartition data to optimize future cluster resource usage. This transform is particularly useful for uneven datasets.

To add an Autobalance Processing transform:

1. Navigate to your visual ETL job in Amazon SageMaker Unified Studio.
2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **Autobalance Processing**.
4. Select the diagram to add the node to your visual ETL job.
5. Select the node on the diagram to view details about the transform.
6. Under **Number of partitions**, input a number of partitions to randomly distribute the data into. Or, switch the toggle to off to use the number of cores as the partition number.
7. (Optional) Under **Repartition columns**, identify columns that you want data of the same value to be assigned to the same partition in.

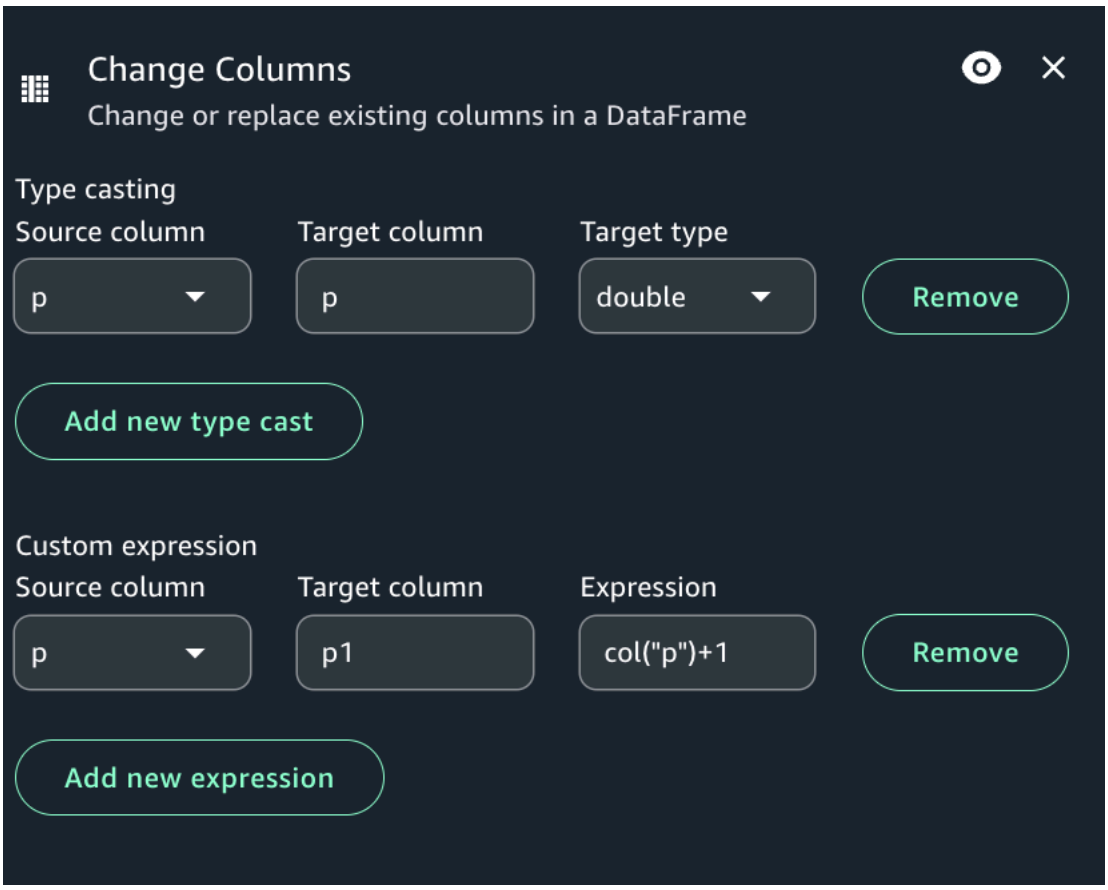
Change columns transform

A Change columns transform remaps the source data property keys into the desired configured for the target data. In a Change Columns transform node, you can:

1. Change the name of multiple columns.
2. Change the data type of the columns, if the new data type is supported and there is a transformation path between the two data types.
3. Use a custom expression to change the values of selected columns.

To add a Change Schema node to your job diagram

1. Open the menu and then choose Change Columns to add a new transform to your job diagram, if needed.
2. (Optional) Click on the rename node icon to enter a new name for the node in the job diagram.
3. Modify the input schema by clicking on the node, then:
    1. To rename a column, enter the new name in the Target column field.
    2. To change the data type for a selected column, select "Add new type cast", then choose the source column, target column, and new data type from the Target type list.
    3. To change the data values for a selected column, add a custom expression.
4. (Optional) After configuring the node properties and transform properties, you can preview the modified dataset by choosing the Data preview tab in the node details panel.



Concatenate columns transform

The Concatenate transform allows you to build a new string column using the values of other columns with an optional spacer. For example, if we define a concatenated column “date” as the concatenation of “year”, “month” and “day” (in that order) with “-” as the spacer, we would get:

day	month	year	date
01	01	2020	2020-01-01
02	01	2020	2020-01-02
03	01	2020	2020-01-02
04	01	2020	2020-01-02

To add a Concatenate Columns transform:

1. Navigate to your visual ETL job in Amazon SageMaker Unified Studio.
2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **Concatenate Columns**.
4. Select the diagram to add the node to your visual ETL job.
5. Select the node on the diagram to view details about the transform.
6. Under **Concatenated column**, enter the name of a new column that will be generated.
7. Under **Columns**, select the input columns.
8. (Optional) Under **Spacer**, enter a string to place between concatenated fields.
9. (Optional) Under **Null value**, enter the string to use when a column value is null.

Custom code transform

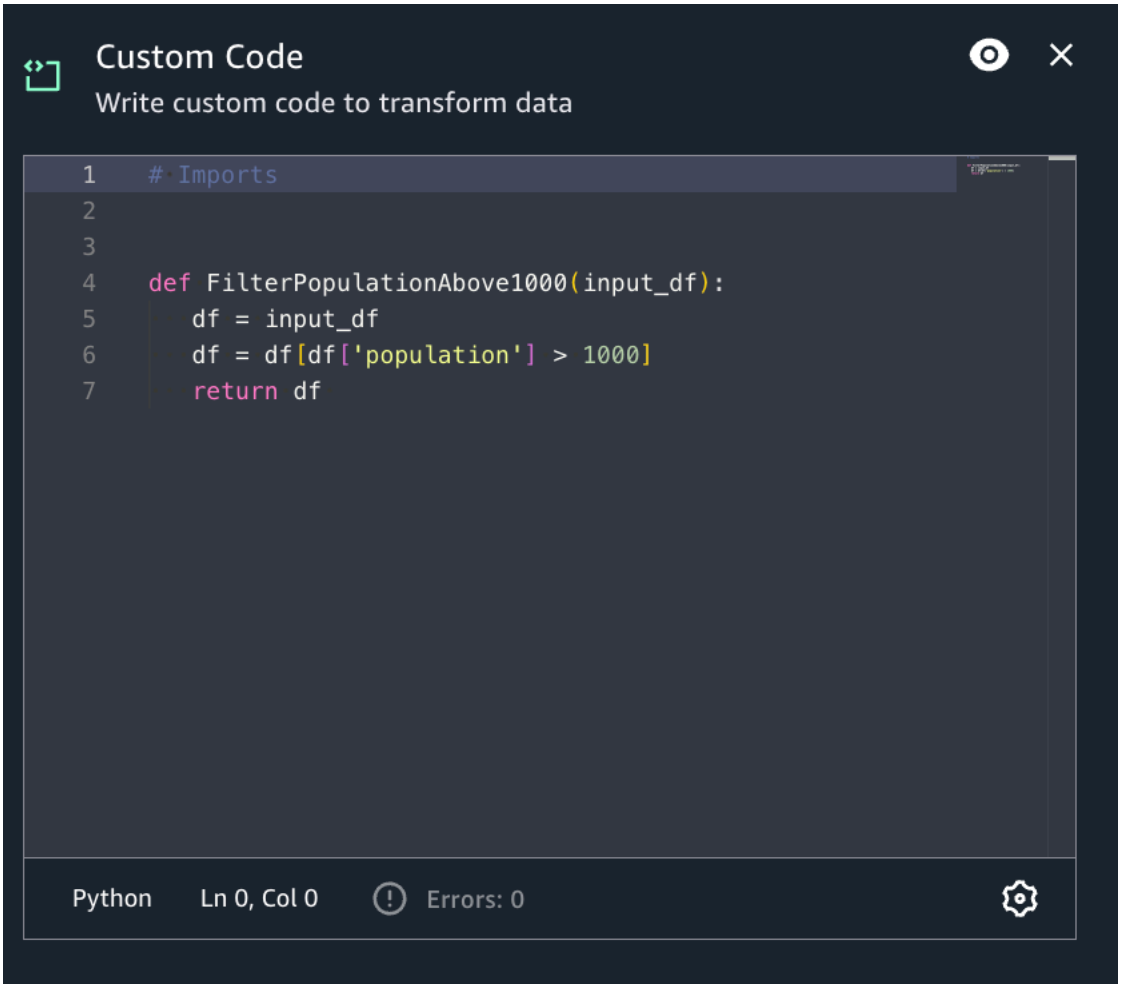
If you need to perform more complicated transformations on your data, or want to add data property keys to the dataset, you can add a Custom code transform to your job diagram. The Custom code node allows you to enter a script that performs the transformation.

To add a custom code node to your job diagram

1. Open the Resource panel and then choose Custom Code to add a custom transform to your job diagram.
2. (Optional) Click on the rename node icon to enter a new name for the node in the job diagram.
3. Enter desired code changes.

The following examples show the format of the code to enter in the code box:

```
def FilterPopulationAbove1000(input_df):
  df = input_df
  df = df[df['population'] > 1000]
  return df
```



Custom visual transform

You can create your own visual transforms and use them in Visual ETL jobs in Amazon SageMaker Unified Studio. To do this, you create a JSON config file and a Python file with your desired custom transform code and settings. Then you upload both files to the transforms folder in your project's Amazon S3 shared folder. The transforms then appear in the Visual ETL transform library.

Migration from Glue Visual ETL

If you already created custom visual transforms in AWS Glue under the same account, those transforms would also be available in Amazon SageMaker Unified Studio Visual ETL. For more information about custom visual transforms in AWS Glue, see Transform data with custom visual transforms in the *AWS Glue Developer Guide*.

The following sections describe the JSON config file schema, the Python file, and the steps to create and use a custom visual transform.

JSON configuration file schema

The JSON config file defines your custom visual transform. It sets the name, description, function to call, and any parameters. The file supports the following fields:

- `name` (required) – The system name for the transform. This name must follow Python naming rules. Start with a letter or underscore. Use only letters, digits, and underscores.
- `displayName` (optional) – The name displayed in the Visual ETL editor. If not specified, `name` is used.
- `description` (optional) – A description displayed in Visual ETL that is searchable.
- `functionName` (required) – The Python function name to call.
- `parameters` (optional) – An array of parameter objects. Each object can have the following fields:
  - `name` (required) – The name passed to the Python function.
  - `displayName` (optional) – The name shown in the UI.
  - `type` (required) – The Python data type. Valid values: `str`, `int`, `float`, `list`, `bool`.
  - `isOptional` (boolean) – Set to true if the parameter is optional.
  - `description` (optional) – Help text shown in the UI.

The following example shows a JSON config file. This transform adds a prefix to all column names:

```
{
  "name": "add_prefix_transform",
  "displayName": "Add Prefix (Custom Transform Example)",
  "description": "Prepends 'my_custom_prefix_' to all column names.",
  "functionName": "add_prefix_transform",
  "parameters": []
}
```

Python implementation file

The Python file contains the function that runs your transform logic. The function name must match the `functionName` value in your JSON config file. The Python file must have the same base name (the file name without the extension) as the JSON file.

The following example shows a Python file for the add prefix transform:

```
def add_prefix_transform(self):
    for col in self.columns:
        self = self.withColumnRenamed(col, f"my_custom_prefix_{col}")
    return self
```

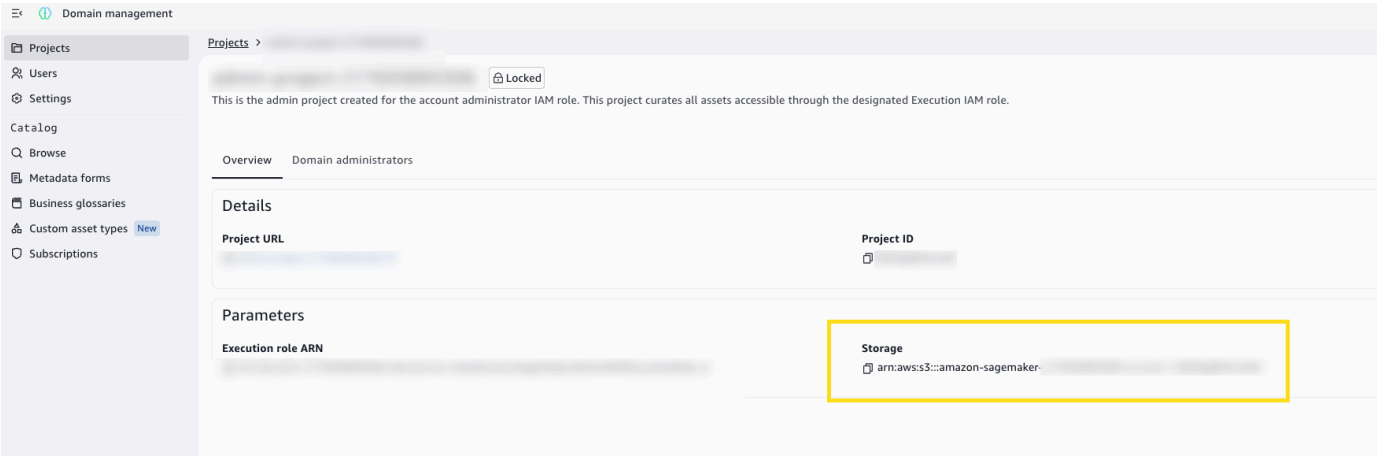
To create and use a custom visual transform

After you create both files, upload them and use the transform in your Visual ETL job.

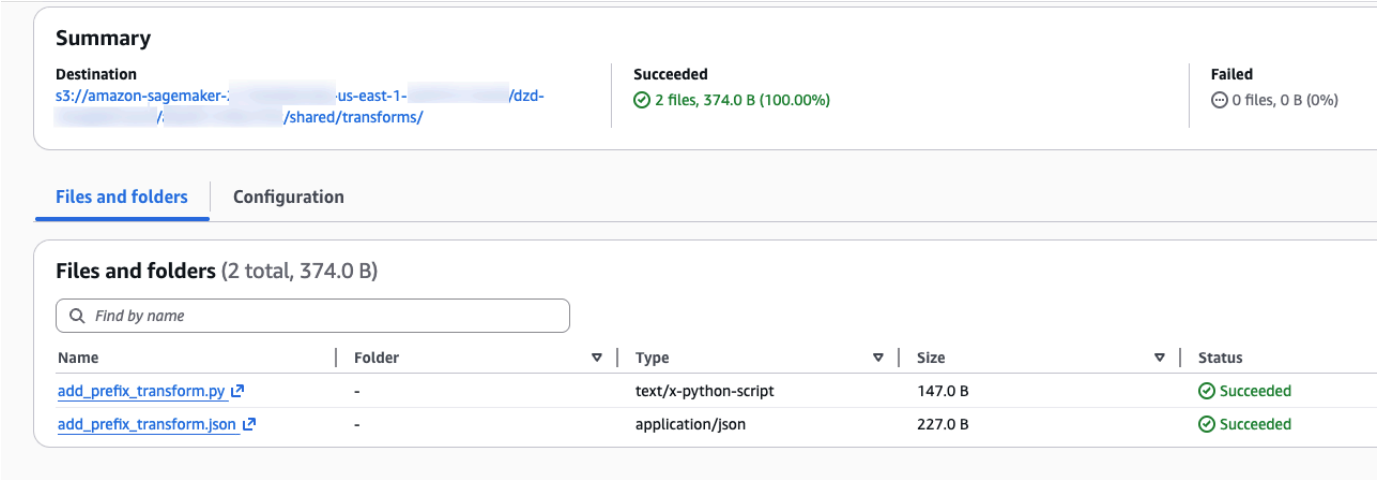
To create and use a custom visual transform

1. Create a JSON config file that defines your transform. Use the schema in the preceding section. Name the file using Python naming rules (for example, `add_prefix_transform.json`).
2. Create a Python file with the transform logic. Give it the same base name as the JSON file (for example, `add_prefix_transform.py`).
3. Upload both files to the Amazon SageMaker Unified Studio Amazon S3 shared folder.

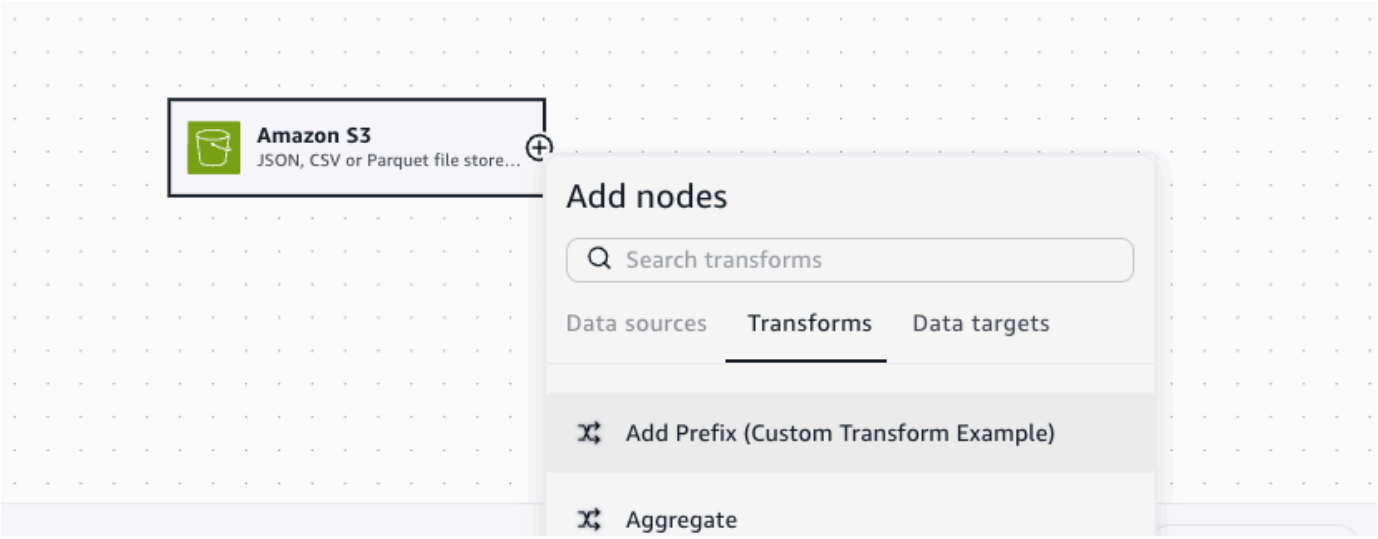
   You can find the Amazon S3 path in your Amazon SageMaker Unified Studio project settings.



   Go to Amazon S3 console, open your shared folder path, and upload the files into a folder named `transforms`. Create the `transforms` folder if it does not exist.



4. Open Visual ETL in Amazon SageMaker Unified Studio.
5. Choose the **Add** (+) icon to open the **Add nodes** menu.
6. Choose the **Transforms** tab. Your custom transform appears in the list of available transforms.
7. Choose your custom transform to add it to your Visual ETL job.



Note

The JSON file and Python file must have the same base name (for example, `myTransform.json` and `myTransform.py`). File names must follow Python naming rules: start with a letter or underscore, and use only letters, digits, and underscores.

Limitations

Note the following limitations for custom visual transforms in Amazon SageMaker Unified Studio:

- Custom visual transform code that uses DynamicFrames is not supported in Amazon SageMaker Unified Studio.
- For custom visual transforms, data preview only works with Glue interactive sessions. Data preview v2.0 is not supported.

Derived column transform

The Derived Column transform allows you to define a new column based on a math formula or SQL expression in which you can use other columns in the data, as well as constants and literals. For instance, to derive a “percentage” column from the columns "success" and "count", you can enter the SQL expression: "success * 100 / count || '%'".

Example result:

success	count	percentage
14	100	14%
6	20	3%
3	40	7.5%

To add a Derived Column transform:

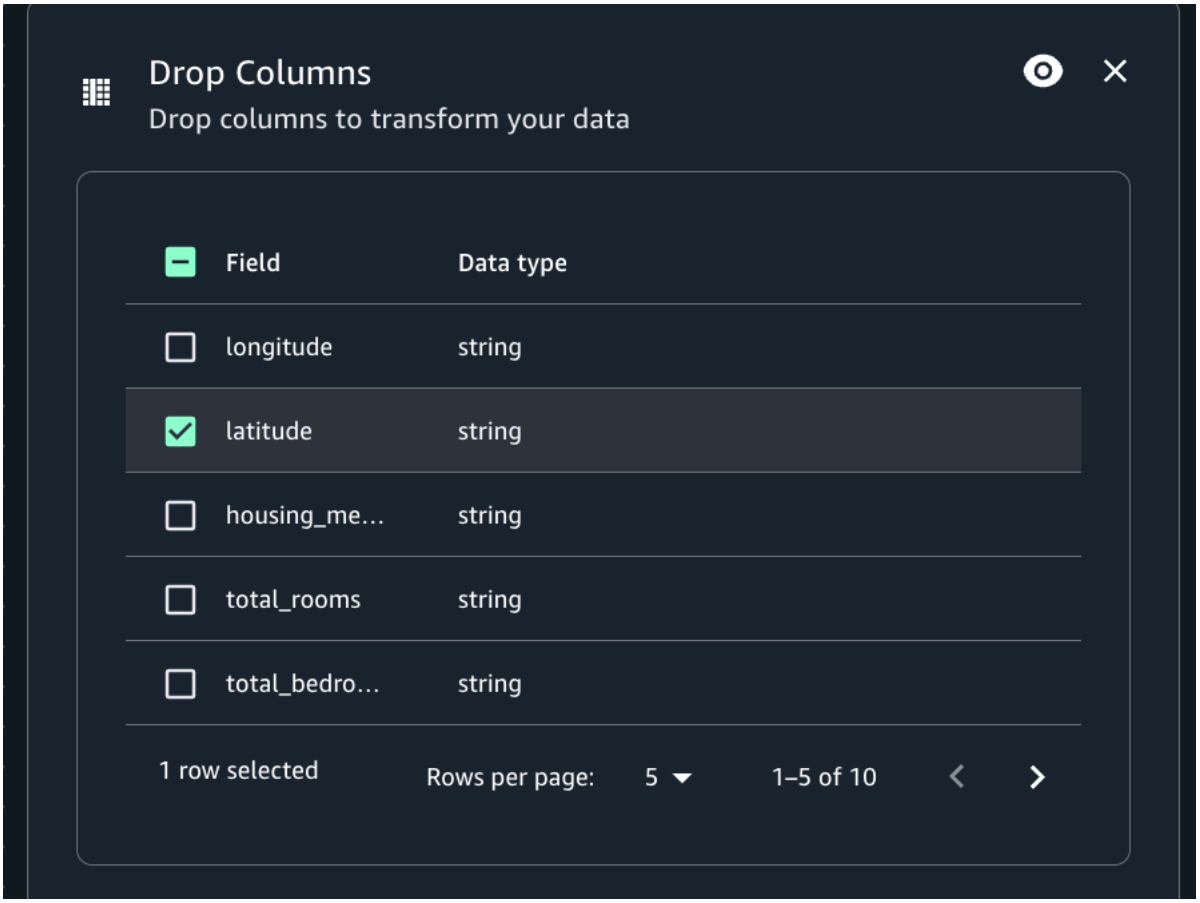
1. Navigate to your visual ETL job in Amazon SageMaker Unified Studio.
2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **Derived Column**.
4. Select the diagram to add the node to your visual ETL job.
5. Select the node on the diagram to view details about the transform.
6. Under **Name of derived column**, enter the name of a new column that will be generated.
7. Under **Column expression**, enter a SQL expression to define the new column based on existing columns.

Drop columns transform

You can create a subset of data property keys from the dataset using the Drop columns transform. You indicate which data property keys you want to remove from the dataset and the rest of the keys are retained.

To add a Drop columns transform node to your job diagram

1. Open the Resource panel and then choose Drop columns to add a new transform to your diagram.
2. (Optional) Click on the rename node icon to enter a new name for the node in the job diagram.
3. Click on the node and view the Node properties panel.
4. Choose the data property keys from the "Field" column to drop the column from the data source.
5. (Optional) After configuring the node properties and transform properties, you can preview using the Data preview tab in job diagram.



Drop duplicates transform

The Drop duplicates transform removes rows from your data source by giving you two options. You can choose to remove the duplicate row that are completely the same, or you can choose to choose the fields to match and remove only those rows based on your chosen fields.

For example, in this data set, you have duplicate rows where all the values in some of the rows are exactly the same as another row, and some of the values in rows are the same or different.

Example Data Set

Row	Name	Email	Age	State	Note
1	Joy	joy@gmail	33	NY	
2	Tim	tim@gmail	45	OH	
3	Rose	rose@gmail	23	NJ	
4	Tim	tim@gmail	42	OH	
5	Rose	rose@gmail	23	NJ	
6	Tim	tim@gmail	42	OH	this is a duplicate row and matches completely on all values as row #4
7	Rose	rose@gmail	23	NJ	This is a duplicate row and matches completely on all values as row #5

If you choose to match entire rows, rows 6 and 7 will be removed from the data set. The data set is now:

Data Set After Matching Entire Rows

Row	Name	Email	Age	State
1	Joy	joy@gmail	33	NY
2	Tim	tim@gmail	45	OH
3	Rose	rose@gmail	23	NJ
4	Tim	tim@gmail	42	OH
5	Rose	rose@gmail	23	NJ

If you chose to specify keys, you can choose to remove rows that match on 'name' and 'email'. This gives you finer control of what is a 'duplicate row' for your data set. By specifying 'name' and 'email', the data set is now:

Data Set After Specifying Keys

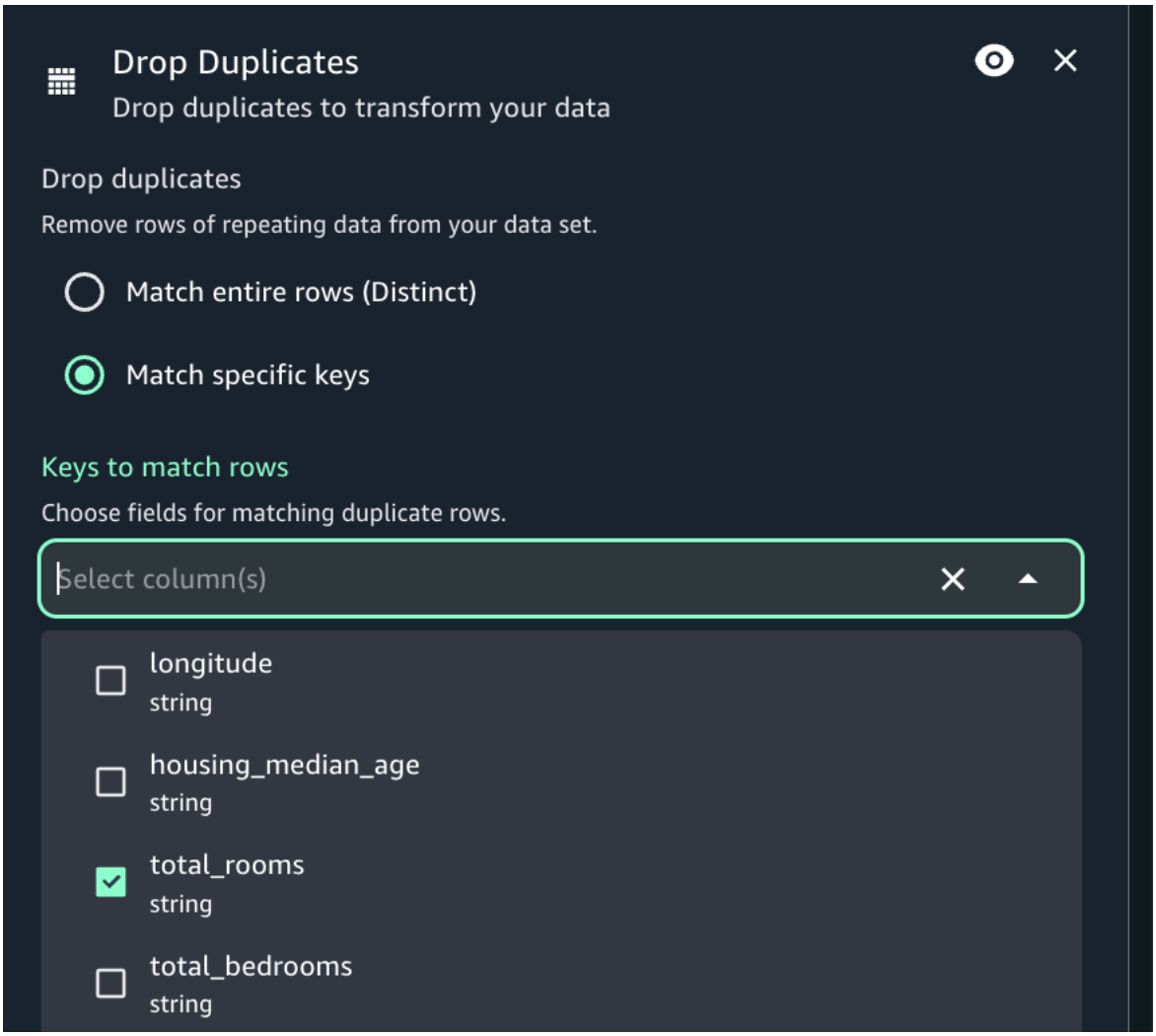
Row	Name	Email	Age	State
1	Joy	joy@gmail	33	NY
2	Tim	tim@gmail	45	OH
3	Rose	rose@gmail	23	NJ

Some things to keep in mind:

- In order for rows to be recognized as a duplicate, values are case sensitive. all values in rows need to have the same casing - this applies to either option you choose (Match entire rows or Specify keys).
- All values are read in as strings.
- The Drop duplicates transform utilizes the Spark dropDuplicates command.
- When using the Drop duplicates transform, the first row is kept and other rows are dropped.
- The Drop duplicates transform does not change the schema of the dataframe.

To add a Drop duplicates transform node to your job diagram

1. Open the Resource panel and then choose Drop duplicates to add a new transform to your diagram.
2. (Optional) Click on the rename node icon to enter a new name for the node in the job diagram.
3. Click on the node and view the Node properties panel.
4. Choose if you prefer to drop duplicates by matching entire rows or specific keys.
5. (Optional) After configuring the node properties and transform properties, you can preview using the Data preview tab in job diagram.

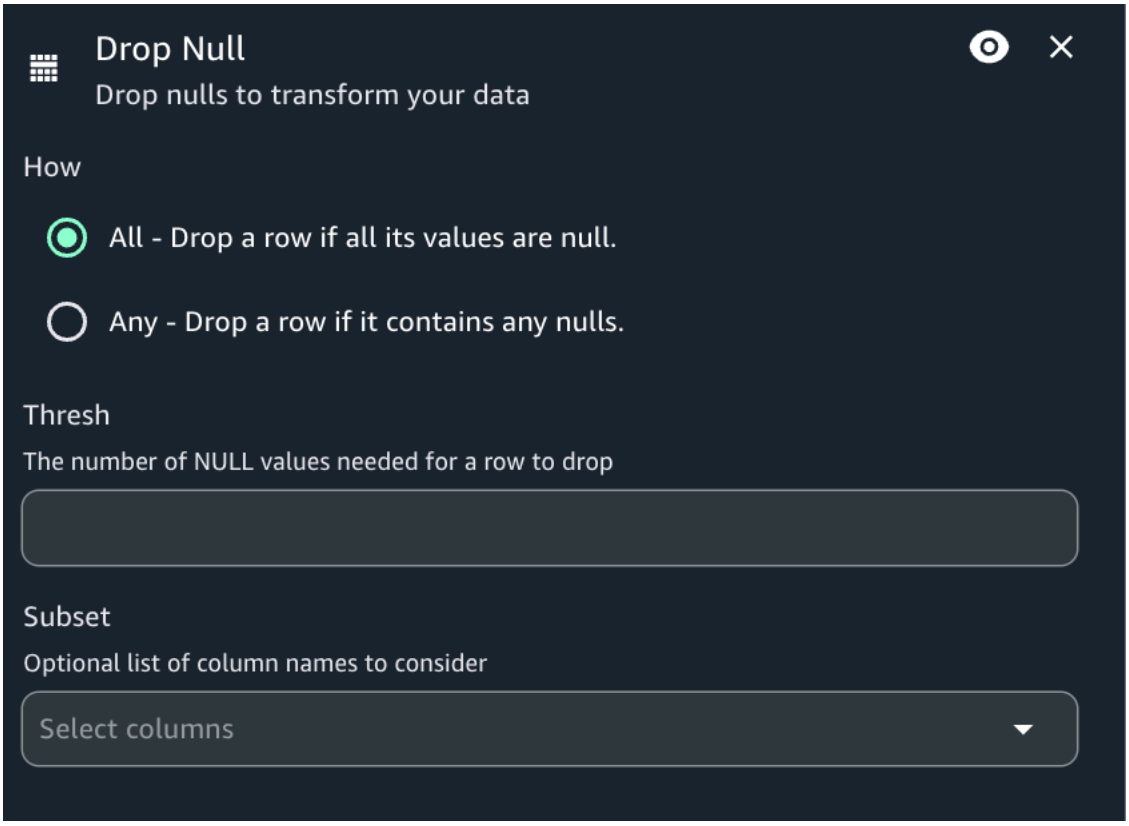


Drop nulls transform

Use the Drop nulls transform to remove fields from the dataset if all values in the field are 'null'. By default, Visual ETL will recognize null objects, but some values such as empty strings, strings that are "null", -1 integers or other placeholders such as zeros, are not automatically recognized as nulls.

To use the Drop nulls

1. Add a Drop nulls node to the diagram, if needed.
2. Set a number as threshold value, if the rows(s) need to be drop only if a certain number of nulls is present.
3. Choose if you prefer to drop nulls for all columns or only check null values for specific columns. If you choose the subset option, select the desired column names from the drop down list.



Explode array or map into rows transform

You can use the Explode transform to extract values from a nested structure into individual rows that are easier to manipulate. In the case of an array, the transform will generate a row for each

value of the array, replicating the values for the other columns in the row. In the case of a map, the transform will generate a row for each entry with the key and value as columns plus any other columns in the row.

For example, if we have this dataset which has a “category” array column with multiple values.

product_id	category
1	[sports, winter]
2	[garden, tools]
3	[videos]
4	[games, electronics, social]
5	[]

If you explode the 'category' column into a column with the same name, you will override the column. You can select that you want NULLs included to get the following (ordered for illustration purposes):

1	sports
1	winter
2	garden
2	tools
3	videos
4	games
4	electronics
4	social
5	

To add an Explode Array Or Map Into Rows transform:

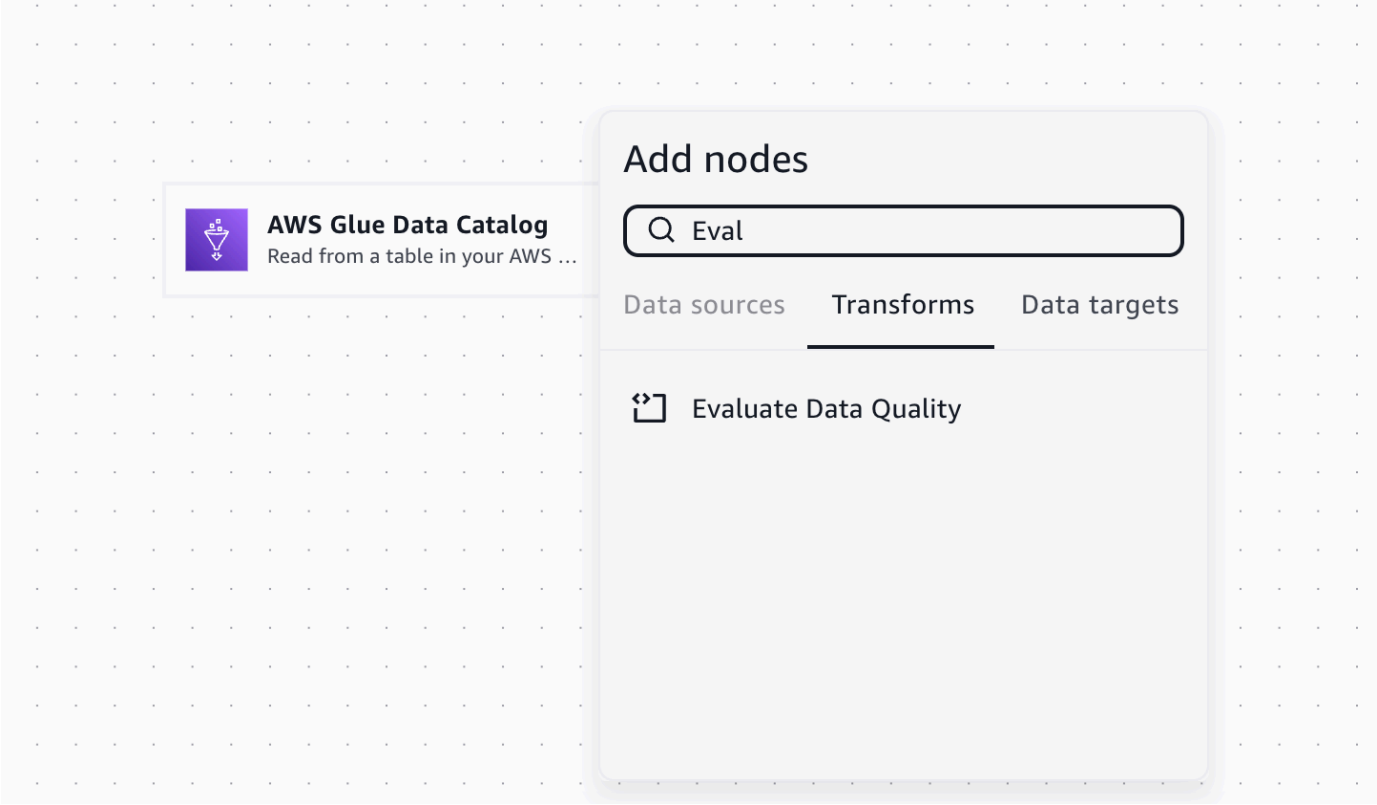
1. Navigate to your visual ETL job in Amazon SageMaker Unified Studio.
2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **Explode Array or Map Into Rows**.
4. Select the diagram to add the node to your visual ETL job.
5. Select the node on the diagram to view details about the transform.
6. Under **Column to explode**, enter the name of an existing column that will be exploded.
7. Under **New column name**, enter the name of a new column that will be generated.
8. (Optional) Under **Values column**, if you are exploding a dictionary, specify a name for a column to contain the values.
9. Use the toggle to indicate whether or not to include null values. By default, if the column to explode is NULL or has an empty structure, it will be omitted on the exploded dataset.

Evaluate data quality transform

The Evaluate data quality transform validates data against a set of rules as it flows through your Visual ETL job. You define rules using DQDL (Data Quality Definition Language), a domain-specific language for defining data quality rules, with 31 built-in rule types.

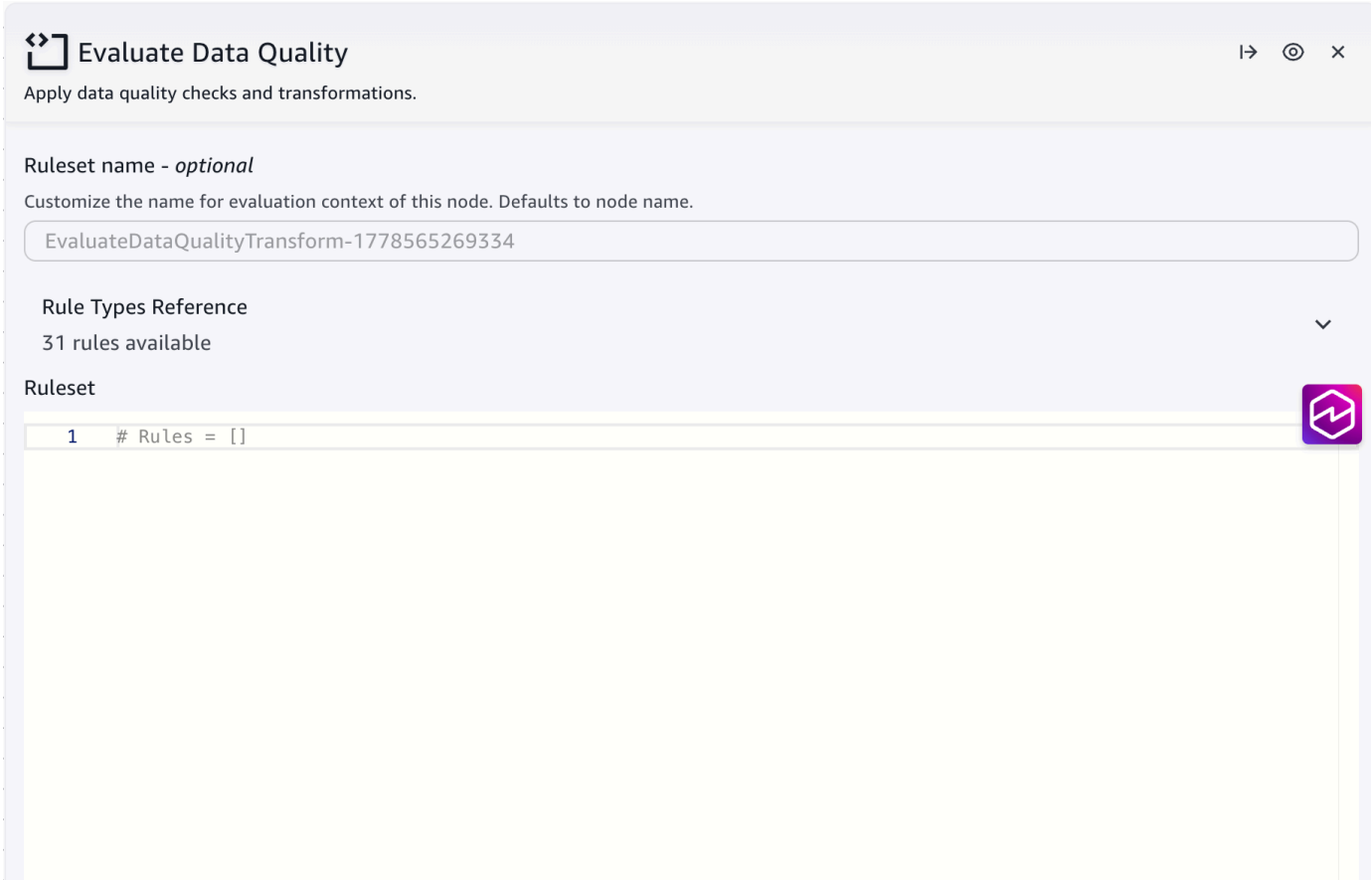
To add an Evaluate Data Quality transform

1. On the Visual ETL canvas, choose the plus icon to open the **Add nodes** panel.
2. Under the **Transforms** tab, search for **Evaluate Data Quality**.
3. Select the node to add it to the canvas, then connect it to your source.



To configure the rule set

1. Select the node to open the configuration panel.
2. For **Ruleset name** (optional), customize the name for evaluation context of this node. The name defaults to the node name. If your job has multiple Evaluate Data Quality nodes, give each a unique name so you can identify results later.
3. Expand **Rule Types Reference** to see the 31 rules available. For details on each rule type and syntax, see DQDL rule types in the AWS Glue documentation.
4. For **Ruleset**, define rules using DQDL syntax. The editor provides autocomplete for rule types and column names from the input schema. Use Ctrl+Tab to trigger column suggestions.



The following example shows a rule set with two rules:

```
Rules = [ColumnExists "phone", ColumnLength "account_length" > 10]
```

> **Note**
>
> The rule set cannot be empty. You must define at least one rule.

Select an output to add it as a child node that downstream transforms can read from. You can add multiple output nodes and route them independently. The following table describes the available outputs.

#	Output	Description
1	Original data	Outputs original data. This option is ideal if you want

#	Output	Description
		to stop the job when quality issues are detected.
2	Evaluation results	Outputs configured rules and their pass or fail status. This option is useful if you want to take a custom action on the results.
3	Row level results	Outputs original data with additional columns depicting the rule result for each row. This option is best for row-specific manipulation based on result.
4	Row level results - Failed rows	Outputs original data with additional columns depicting the rule result for each row, filtering for only rows that failed the data quality evaluation checks.
5	Row level results - Passed rows	Outputs original data with additional columns depicting the rule result for each row, filtering for only rows that passed the data quality evaluation checks.

Additional options

The Evaluate Data Quality node includes options for publishing results and controlling job behavior on failure.

#	Option	Description
1	Publish results to Amazon CloudWatch	Send evaluation metrics to CloudWatch for monitoring and alerting.
2	Publish data quality evaluation results to S3	Write detailed results to an S3 folder. Choose Browse S3 to select the target location.
3	Stop job on rule set failure	Halt the job if any rule fails, preventing bad data from flowing downstream.

Viewing results after job runs

After the job completes, results are available on the **Data quality** tab of the data processing job detail page. For details, see *Data quality monitoring*.

Extract JSON path transform

Use this transform to extract new columns from a string column with JSON data.

To add an Extract JSON path transform:

1. Navigate to your visual ETL job in Amazon SageMaker Unified Studio.
2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **Extract JSON Path**.
4. Select the diagram to add the node to your visual ETL job.
5. Connect the transform node to a data source node.
6. Select the node on the diagram to view details about the transform.
7. Under **JSON column**, select a column from the data source.
8. Under **JSON paths**, enter a JSON path or multiple paths separated by commas to specify the location to extract the data from.
9. Under **Output columns**, enter the names of the columns to create from each of the JSON paths.

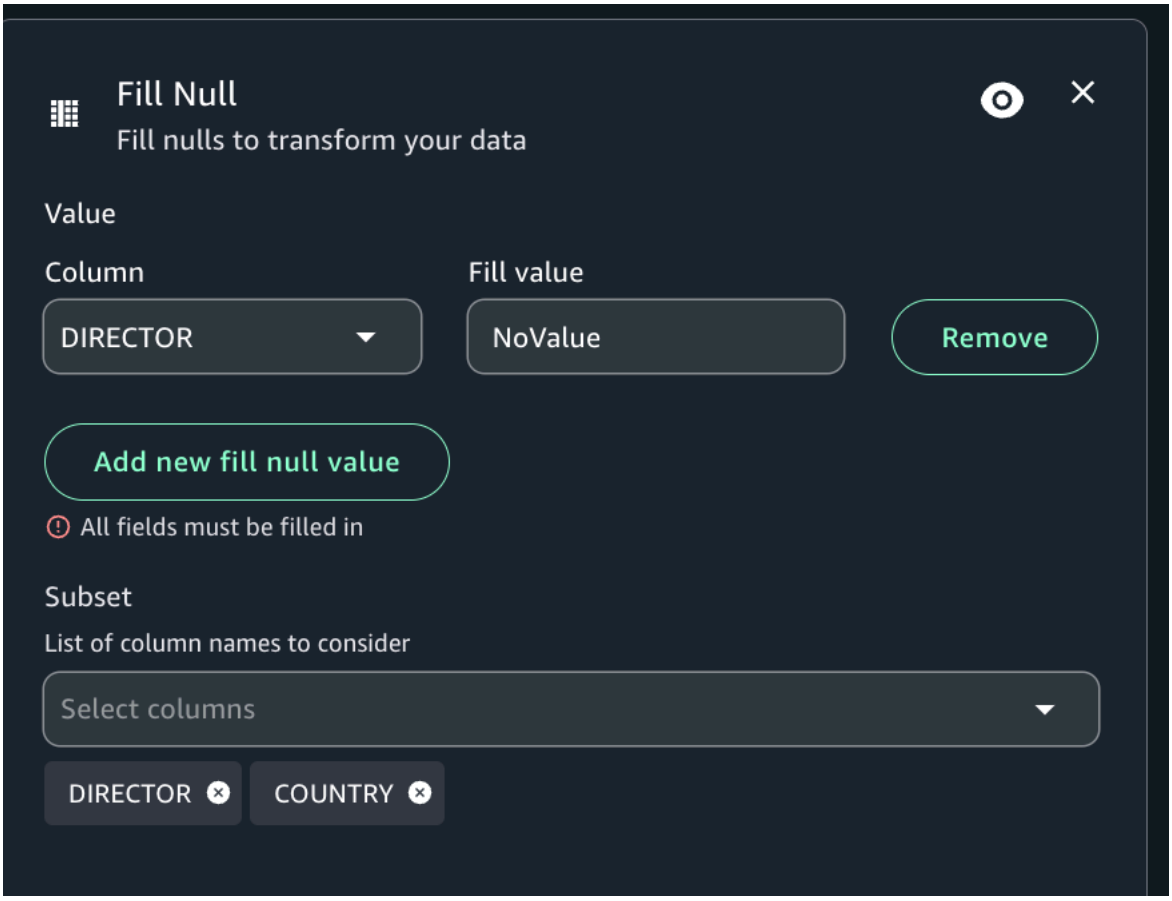
10. (Optional) If desired, use the toggle to drop the JSON column.

Fill nulls transform

Fill nulls allows you to fill in null value in a column with a chosen fill value. Select a column with nulls you want to fill and provide a fill value. You can use multiple fill conditions at one time. Select a subset of the data to consider by using a dropdown menu to select columns.

To add a Fill nulls node to your job diagram

1. Open the menu and then choose Fill nulls to add a new transform to your job diagram.
2. (Optional) Click on the rename node icon to enter a new name for the node in the job diagram.
3. Modify the input schema:
    1. To create a subset, use the dropdown menu and select a column or columns to consider.
    2. To specify a value to fill nulls with, select "Add new fill null value", then choose a column to fill and provide a Fill value.
4. (Optional) After configuring the node properties and transform properties, you can preview the modified dataset by choosing the Data preview tab in the node details panel.



Filter transform

Use the Filter transform to create a new dataset by filtering records from the input dataset. Rows that don't satisfy the filter condition are removed from the output. You can select from two filter types: Global AND or Global OR. You must select a column name to serve as the key, a comparison operation, and provide a value to filter on.

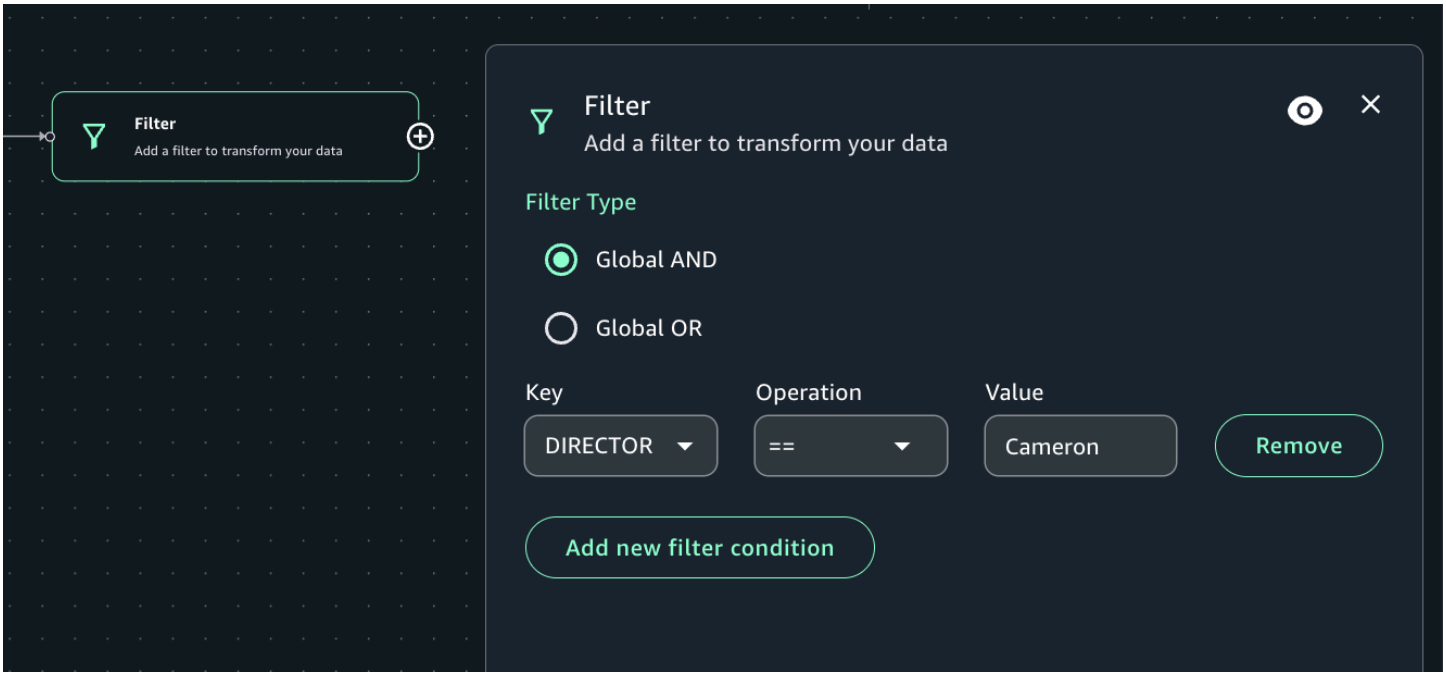
To add a Filter node to your job diagram

1. Open the menu and then choose Filter to add a new transform to your job diagram.
2. (Optional) Click on the rename node icon to enter a new name for the node in the job diagram.
3. Modify the input schema:

    1. Select "Add new filter condition".
    2. Choose a filter type: "Global AND" or "Global or".
    3. Select a Key column to filter.
    4. Select a comparison operation.

5. Type in a value to compare in the "Value" box.

4. (Optional) After configuring the node properties and transform properties, you can preview the modified dataset by choosing the Data preview tab in the node details panel.



Flatten transform

You can flatten the fields of nested structs in the data so they become top level fields. The new fields are named using the field name prefixed with the names of the struct fields to reach it, separated by dots.

For example, if the data has a field of type Struct named “phone_numbers”, which among other fields has one of type “Struct” named “home_phone” with two fields: “country_code” and “number”. After they are flattened, these two fields will become top level fields named “phone_numbers.home_phone.country_code” and “phone_numbers.home_phone.number” respectively.

To add a Flatten transform:

1. Navigate to your visual ETL job in Amazon SageMaker Unified Studio.
2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **Flatten**.

4. Select the diagram to add the node to your visual ETL job.
5. Select the node on the diagram to view details about the transform.
6. Under **Max levels to flatten**, enter the a maximum limit.
7. Under **Separator**, enter a character to use in the new column names to separate the different levels.

Format timestamp transform

Use this transform to convert a timestamp colum into a formatted string.

To add a Format Timestamp transform:

1. Navigate to your visual ETL job in Amazon SageMaker Unified Studio.
2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **Format Timestamp**.
4. Select the diagram to add the node to your visual ETL job.
5. Select the node on the diagram to view details about the transform.
6. Under **Column**, choose the column that contains the timestamps.
7. Under **Timestamp format**, input the format you want to use for the strings.
8. (Optional) Enter a name for the new formatted column.

Identifier transform

Use this transform to assign a numeric identifier for each row in the dataset.

To add an Identifier transform:

1. Navigate to your visual ETL job in Amazon SageMaker Unified Studio.
2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **Identifier**.
4. Select the diagram to add the node to your visual ETL job.
5. Connect the transform node to a data source node.
6. Select the node on the diagram to view details about the transform.
7. Under **Column name**, enter a name for the new column. This column is named id by default.

8. (Optional) If you want to make the ID unique among the job runs, switch the toggle to on to include a timestamp and extend the column type from long to decimal.

Intersect transform

Use this transform to select only the common rows in two datasets. Duplicates are removed by default, and you can choose to keep them in by checking the **Intersect all** checkbox.

To add an Intersect transform:

1. Navigate to your visual ETL job in Amazon SageMaker Unified Studio.
2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **Intersect**.
4. Select the diagram to add the node to your visual ETL job.
5. Select the node on the diagram to view details about the transform.
6. (Optional) If you want to intersect duplicated rows as well, check the **Intersect all** checkbox.

Join transform

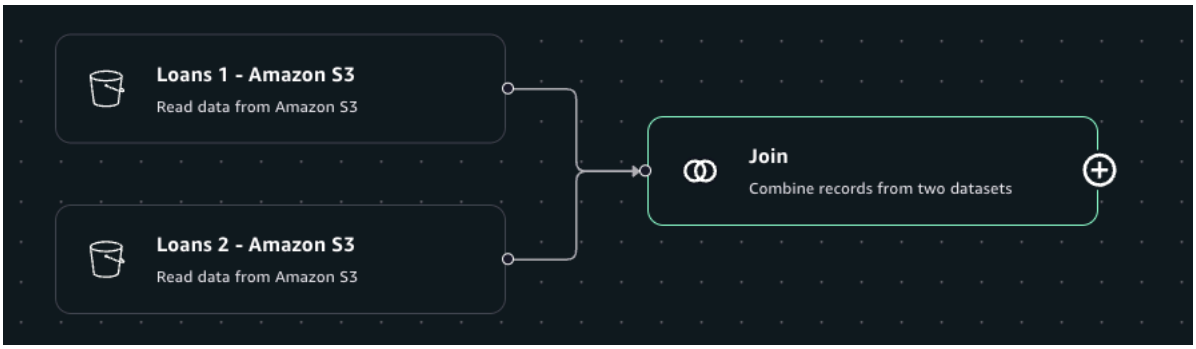
The Join transform allows you to combine two datasets into one. You specify the key names in the schema of each dataset to compare. The output frame contains rows where keys meet the join condition. The rows in each dataset that meet the join condition are combined into a single row in the output from that contains all the columns found in either dataset. You can select from one of the following join types: Inner, Left, Right, Full, Cross, Semi, and Anti.

To add a Join node to your job diagram

1. Open the menu and then choose Join to add a new transform to your job diagram, if needed.
2. (Optional) Click on the rename node icon to enter a new name for the node in the job diagram.
3. Optional) Ensure two data sources are connected to the Join node.
4. Modify the input schema:

    1. Select a join type from the "Join type" dropdown menu.Optional).
    2. Select a column for the "Left data source" using the dropdown menu.
    3. Select a column for the "Right data source" using the dropdown menu.

5. (Optional) After configuring the node properties and transform properties, you can preview the modified dataset by choosing the Data preview tab in the node details panel.



> **Note**
>
> If you have columns using the same names in your data sources, the join will result in a COLUMN_ALREADY_EXISTS error. To avoid this error, either: (1) use Rename columns before the join for one of your data sources or (2) use Drop Columns after the join to remove both duplicated columns.

Limit tranform

Use this transform to limit the number of rows in a dataset.

To add a Limit transform:

1. Navigate to your visual ETL job in Amazon SageMaker Unified Studio.
2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **Limit**.
4. Select the diagram to add the node to your visual ETL job.
5. Select the node on the diagram to view details about the transform.
6. Under **Number of records**, enter the maximum number of rows you want to see or process from the dataset.

Lookup transform

Use this transform to add matching columns by looking them up on another catalog table.

To add a Lookup transform:

1. Navigate to your visual ETL job in Amazon SageMaker Unified Studio.
2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **Lookup**.
4. Select the diagram to add the node to your visual ETL job.
5. Connect the transform node to a data source node.
6. Select the node on the diagram to view details about the transform.
7. Under **Catalog**, select the catalog to use for lookup.
8. Under **Database**, select a database.
9. Under **Table**, select a table.
10. Under **Lookup key columns to match**, enter columns in the lookup table, separated by commas.
11. Under **Lookup columns to take**, enter columns in the lookup table to add to the data when a match is found.

Parse JSON column transform

Use this transform to parse a string column that contains JSON data.

To add a Parse JSON Column transform:

1. Navigate to your visual ETL job in Amazon SageMaker Unified Studio.
2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **Parse JSON Column**.
4. Select the diagram to add the node to your visual ETL job.
5. Connect the transform node to a data source node.
6. Select the node on the diagram to view details about the transform.
7. Under **JSON column**, choose the column that contains JSON data
8. Enter a number to use as a ratio of samples to use for inferring schema. This must be a value between 0.1 and 1, with 1 meaning all samples are used. If your JSON strings are consistent, you can use a lower value to speed up the inference.
9. (Optional) Enter a name for the new column.

Pivot rows into columns transform

Use this transform to aggregate the values of a numeric column into categories by using the values of the columns configured.

To add a Pivot Rows Into Columns transform:

1. Navigate to your visual ETL job in Amazon SageMaker Unified Studio.
2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **Pivot Rows Into Columns**.
4. Select the diagram to add the node to your visual ETL job.
5. Connect the transform node to a data source node.
6. Select the node on the diagram to view details about the transform.
7. Under **Aggregation column**, select a numeric column to use.
8. Under **Aggregation**, select a Spark function to apply to the aggregation column.
9. Under **Columns to convert**, select columns whose values will become new columns.

Regex extract transform

Use this transform to extract new columns from a string column with JSON data.

To add a Regex Extract transform:

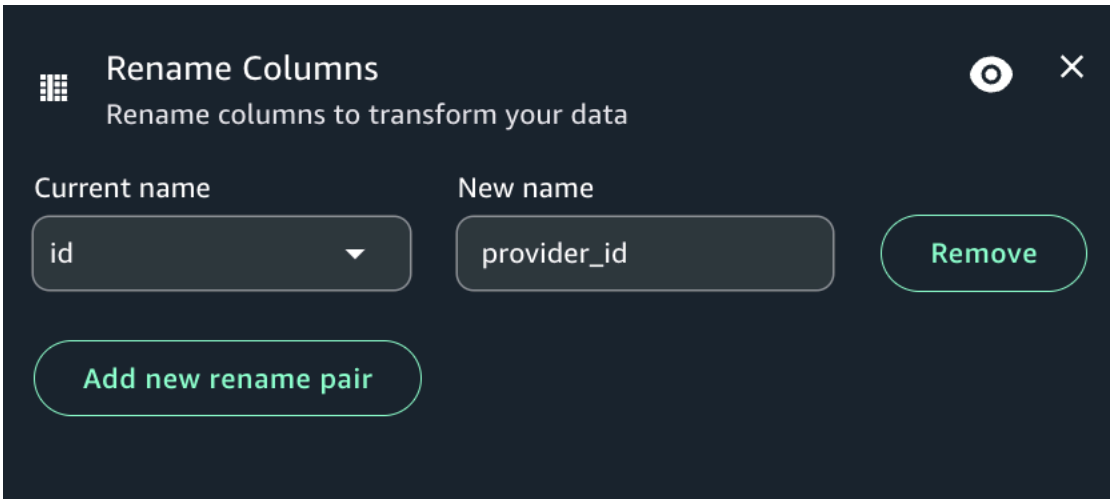
1. Navigate to your visual ETL job in Amazon SageMaker Unified Studio.
2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **Regex Extract**.
4. Select the diagram to add the node to your visual ETL job.
5. Select the node on the diagram to view details about the transform.
6. Under **Column to extract from**, choose a column from the data source you connected to the transform.
7. Under **Regular expression**, enter a regular expression to apply on the column. If there are multiple columns, the expression must have a number of groups equal to the number of columns.
8. Under **Extracted column**, enter the name of the column to contain the extracted regex.

Rename columns transform

You can use the Rename columns transform to change the name of multiple columns in the dataset. You select a column that you want to rename from a list of available columns and provide the string that you want to update the column name to.

To add a Rename columns node to your job diagram

1. Open the menu and then choose Rename columns to add a new transform to your job diagram, if needed.
2. (Optional) Click on the rename node icon to enter a new name for the node in the job diagram.
3. Modify the input schema:
    1. Select "Add new rename pair".
    2. Select the current column to rename from the "Current name" column.
    3. Provide a name for the column in the "New name" box.
4. (Optional) After configuring the node properties and transform properties, you can preview the modified dataset by choosing the Data preview tab in the node details panel.

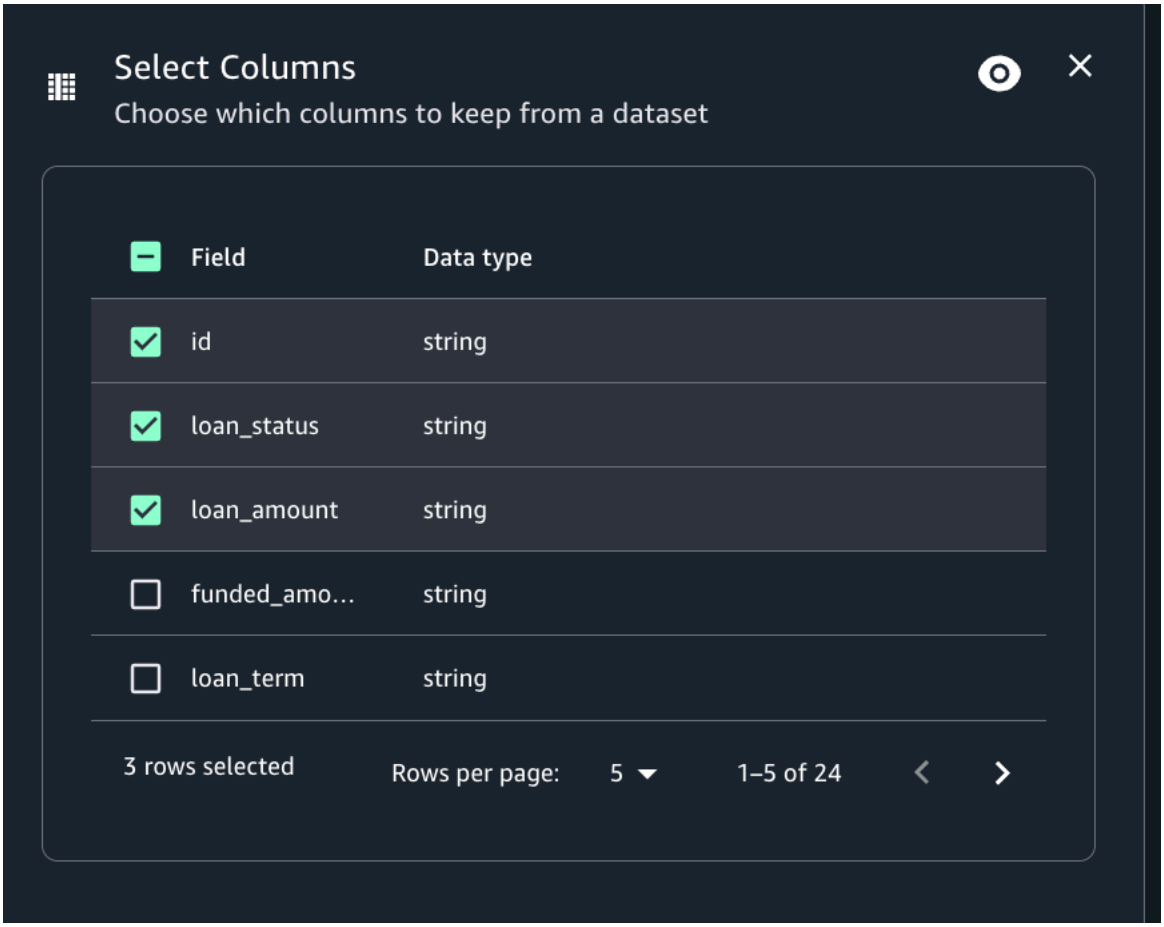


Select columns transform

You can create a subset of columns in a dataset using the Select columns transform. You indicate which columns you want to keep and the rest are removed from the dataset.

To add a Select columns node to your job diagram

1. Open the menu and then choose Filter to add a new transform to your job diagram, if needed.
2. (Optional) Click on the rename node icon to enter a new name for the node in the job diagram.
3. Modify the input schema:
    1. Select the columns you want to keep in the output frame by checking the corresponding box under the "Field" option.
4. (Optional) After configuring the node properties and transform properties, you can preview the modified dataset by choosing the Data preview tab in the node details panel.



Split string transform

Use this transform to split a string column into an array.

To add a Split String transform:

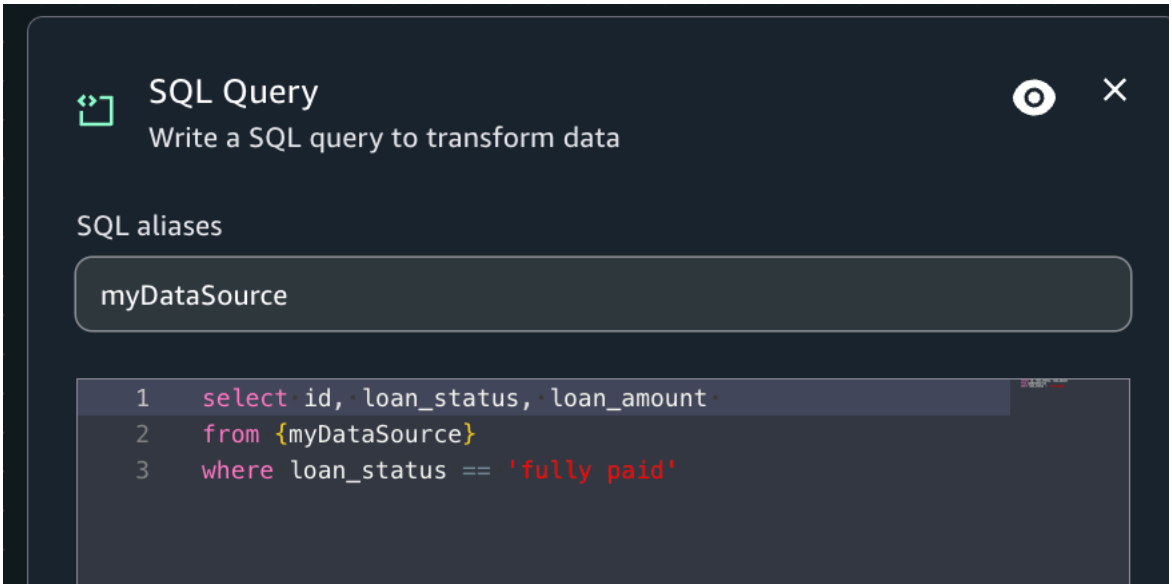
1. Navigate to your visual ETL job in Amazon SageMaker Unified Studio.
2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **Split String**.
4. Select the diagram to add the node to your visual ETL job.
5. Connect the transform node to a data source node.
6. Select the node on the diagram to view details about the transform.
7. Under **Column to split**, select the column that you want to split.
8. Under **Splitting regular expression**, enter a regular expression that defines the separator token.
9. (Optional) Under **Array column name, enter a name for the new column that contains the extracted array.**

SQL query transform

You can use a SQL query transform to write your own transform in the form of a SQL query. Writing the SQL query in Visual ETL creates a subset of the data corresponding to the query. A SQL transform node can have multiple datasets as inputs, but produces only a single dataset as output. It contains a text field, where you enter the SQL query.

To add an SQL query node to your job diagram

1. Open the menu and then choose SQL query to add a new transform to your job diagram, if needed.
2. (Optional) Click on the rename node icon to enter a new name for the node in the job diagram.
3. Modify the input schema:
    1. Ensure the alias in the "SQL aliases" box is appropriate. Visual ETL will autopopulate this field, but you can change it.
    2. Write an SQL statement that queries the data to suit your needs
4. (Optional) After configuring the node properties and transform properties, you can preview the modified dataset by choosing the Data preview tab in the node details panel.



To timestamp transform

You can use the transform To Timestamp to change the data type of a numeric or string column into timestamp, so that it can be stored with that data type or applied to other transforms that require a timestamp.

To add a To Timestamp transform:

1. Navigate to your visual ETL job in Amazon SageMaker Unified Studio.
2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **To Timestamp**.
4. Select the diagram to add the node to your visual ETL job.
5. Select the node on the diagram to view details about the transform.
6. Under **Column to convert**, select the column that you want to convert.
7. Under **Time format**, define how to parse the column selected by choosing the format.

   If the value is a number, it can be expressed in seconds (Unix/Python timestamp), milliseconds, or microseconds.

   If the value is a formatted string, choose the "iso" type. The string needs to conform to one of the variants of the ISO format, for example: “2022-11-02T14:40:59.915Z“.

   If you don’t know the type at this point or different rows use different types, then you can choose ”autodetect“ and the system will make its best guess, with a small performance cost.

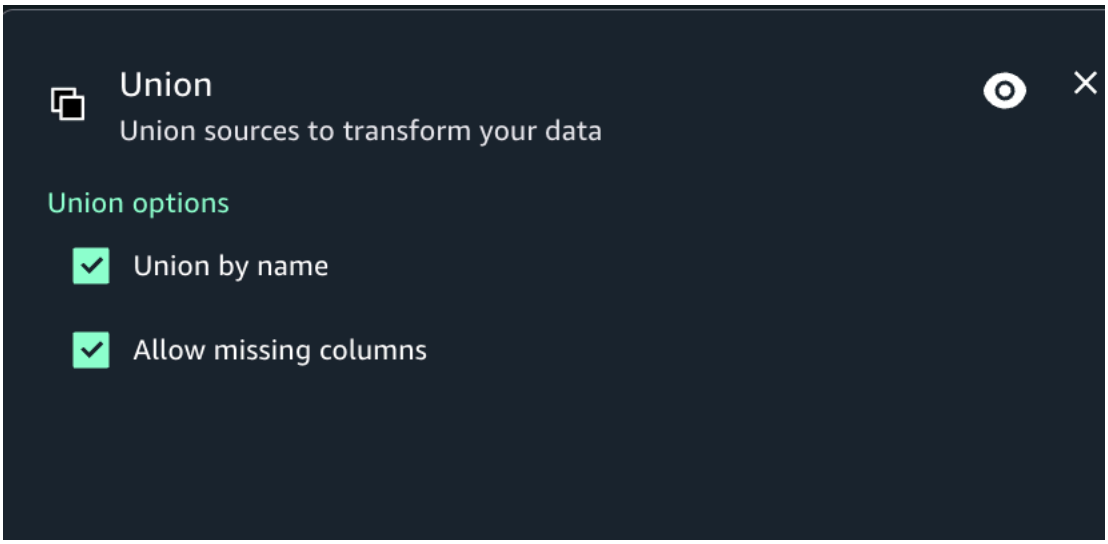
8. Under **Column name**, enter a name for a new column that will be generated for the timestamps. If a column name is not specified, the existing column will be replaced instead of generating a new column.

Union transform

You use the Union transform node when you want to combine rows from more than one data source that have the same schema. When applying Union transformations you can select to "Union by name" to have the union done on columns with the same name (rather than by position). Selecting this option also lets you select "Allow missing columns", which will produce a frame that has missing columns filled with Null values.

To add a Union transform node to your job diagram

1. Open the menu and then choose "Union transform" to add a new transform to your job diagram, if needed.
2. (Optional) Click on the union node icon to enter a new name for the node in the job diagram.
3. Modify the input schema:
    1. Select "Union by name" if you want the union to be done on columns with the same name.
    2. If you have enabled Union by name, select "Allow missing columns" if you want to fill missing columns with null values.
4. (Optional) After configuring the node properties and transform properties, you can preview the modified dataset by choosing the Data preview tab in the node details panel.



Unpivot columns into rows transform

Use this transform to convert column names into values of a new column and generate one row for each column in the original row.

To add an Unpivot Columns Into Rows transform:

1. Navigate to your visual ETL job in Amazon SageMaker Unified Studio.
2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **Unpivot Columns Into Rows**.
4. Select the diagram to add the node to your visual ETL job.
5. Connect the transform node to a data source node.
6. Select the node on the diagram to view details about the transform.
7. Under **Unpivot names column**, enter a column to create out of the source column names.
8. Under **Unpivot values column**, enter a column to create out of the source column values.
9. Under **Columns to unpivot into the new value column**, select the column or columns whose names will become the values of the new column.

UUID transform

Use this transform to generate a 36-character unique identifier for each row in your data source.

To add a UUID transform:

1. Navigate to your visual ETL job in Amazon SageMaker Unified Studio.
2. Choose the plus icon to open the **Add nodes** menu.
3. Under **Transforms**, choose **UUID**.
4. Select the diagram to add the node to your visual ETL job.
5. Connect the transform node to a data source node.
6. Select the node on the diagram to view details about the transform.
7. Under **UUID column name**, enter a name to use for the new column that contains the unique identifiers.

Visual ETL for IAM-based domains

IAM-based domains provide access to a new Amazon SageMaker Unified Studio interface where customers log in using federated roles and existing IAM permissions apply. The following sections describe how to use Visual ETL in IAM-based domains.

Key features for IAM-based domains

Visual ETL offers several capabilities to streamline your data workflows in IAM-based domains:

- Drag-and-drop interface: Create Visual ETL flows by dragging and connecting components on a canvas.
- Wide range of data connectors: Connect to various data sources and destinations, including databases, file systems, cloud storage, and APIs.
- Extensive transformation library: Apply a variety of pre-built transformations to your data, such as filtering, aggregation, joining, and data type conversions.
- Custom transformations: Create and save custom transformations using SQL or Python for reuse in multiple flows.
- Data preview: Visualize your data at each step of the authoring process to ensure accuracy and data quality.
- View scripts: View the code generated and choose to convert the flow to a notebook and continue authoring with code.
- Code and compute configuration: Use a configuration panel to add code libraries and adjust the compute settings.

Create a Visual ETL job in IAM-based domains

To create a job using Visual ETL in Amazon SageMaker Unified Studio IAM-based domains:

1. Log in to Amazon SageMaker Unified Studio.
2. In the left navigation pane, under **Data analytics**, choose **Visual ETL**.
3. Choose "Create Visual job" to open the Visual ETL editor.
4. Give the job a name when you begin authoring the job and choose "save".
5. Open the "Add nodes" menu by choosing the plus icon and select a node, choosing your node from one of the three tabs: "Data sources", "Transforms", or "Data targets".

6. Drag a source component onto the canvas.
7. Configure the component by choosing the node and editing the configurations, to connect to your data source.
8. Add transformation components as needed, connecting them in the desired order.
9. Drag a data target onto the canvas and configure it to specify where the processed data should be stored.
10. Connect the components to create a complete job.
11. Choose the "Checklist" button to check for any configuration errors.
12. Choose "Save" when you are done correcting all errors.
13. Select "Run" to execute it immediately or choose the schedule icon to create a reoccurring run schedule.

Authoring a Visual ETL job using generative AI in IAM-based domains

To author a Visual ETL job using generative AI in Amazon SageMaker Unified Studio IAM-based domains:

1. Verify Amazon Q is enabled for your domain.
2. Open the Visual ETL editor.
3. In the "Add nodes" panel choose the Amazon Q icon.
4. (Optional) Choose "What can I ask?" and copy a prompt.
5. Enter the desired prompt in the chat box and choose 'Submit'.
6. Choose each node in the Visual ETL editor and configure its settings.

Scheduling and running visual jobs in IAM-based domains

There are two ways to schedule visual ETL jobs in Amazon SageMaker Unified Studio IAM-based domains:

- You can schedule your visual jobs directly in the Visual ETL editor. This way you can schedule a single visual job quickly.
- You can schedule your visual job using a DAG and the workflows interface. This way you can combine multiple elements in the same schedule.

Visual ETL for Identity Center-based domains

Identity Center-based domains use the original Amazon SageMaker Unified Studio interface and user experience that continues to be supported and maintained. The following sections describe how to use Visual ETL in Identity Center-based domains.

Key features for Identity Center-based domains

Visual ETL offers several capabilities to streamline your data workflows:

1. Drag-and-drop interface: Create Visual ETL flows by dragging and connecting components on a canvas.
2. Wide range of data connectors: Connect to various data sources and destinations, including databases, file systems, cloud storage, and APIs.
3. Extensive transformation library: Apply a variety of pre-built transformations to your data, such as filtering, aggregation, joining, and data type conversions.
4. Custom transformations: Create and save custom transformations using SQL or Python for reuse in multiple flows.
5. Data preview: Visualize your data at each step of the authoring process to ensure accuracy and data quality.
6. View scripts: View the code generated and choose to convert the flow to a notebook and continue authoring with code.
7. Code and compute configuration: Use a configuration panel to add code libraries and adjust the compute settings.

Creating a Visual ETL job in Identity Center-based domains

To create a job using Visual ETL in Amazon SageMaker Unified Studio Identity Center-based domains:

1. Log in to Amazon SageMaker Unified Studio and from the project selector dropdown at the top of the page, choose a project.
2. In the left navigation pane, under **Data analytics**, choose **Visual ETL**.
3. Choose "Create Visual ETL job" to open the Visual ETL editor.

If this is your first time using Visual ETL jobs in Amazon SageMaker Unified Studio, you are asked to choose a default compute permission mode option based on your data access preference.

4. Give the job a name when you begin authoring the job.
5. From the dropdown menu next to the Run button, choose the compute permission mode option that supports the data you will be using in the job.
   - Select **project.spark.fineGrained** for data managed using fine-grained access, meaning the compute engine can only access specific rows and columns from the full dataset. Choosing this option configures your compute to work with data asset subscriptions from Amazon SageMaker Catalog.
   - Select **project.spark.compatibility** to configure permission mode to be compatible with data managed using full-table access, meaning the compute engine can access all rows and columns in the data. Choosing this option configures your compute to work with data assets from AWS and from external systems that you connect to from your project.
6. Select the "Add nodes" button and select a node, chooing your node from one of the three tabs: "Data sources", "Transforms", or "Data targets".
7. Drag a source component onto the canvas.
8. Configure the component by choosing the node and editing the configurations, to connect to your data source.
9. Add transformation components as needed, connecting them in the desired order.
10. Drag a data target onto the canvas and configure it to specify where the processed data should be stored.
11. Connect the components to create a complete job.
12. Choose the "Checklist" button to check for any configuration errors.
13. To make the job accessible for all project members to view and edit, select "Save to project".
14. Select "Run" to execute it immediately or run it on a schedule with the instructions at the section called “Scheduling and running visual jobs in Identity Center-based domains”.

Authoring a Visual ETL job using generative AI in Identity Center-based domains

To author a Visual ETL job using generative AI in Amazon SageMaker Unified Studio Identity Center-based domains:

1. Verify Amazon Q is enabled for your domain.
2. Open the Visual ETL editor.
3. In the "Add nodes" panel choose the Amazon Q icon.
4. (Optional) Choose "What can I ask?" and copy a prompt.
5. Enter the desired prompt in the chat box and choose 'Submit'.
6. Choose each node in the Visual ETL editor and configure its settings.

Scheduling and running visual jobs in Identity Center-based domains

There are two ways to schedule visual ETL jobs in Amazon SageMaker Unified Studio Identity Center-based domains.

- You can schedule your visual jobs directly in the Visual ETL editor. This way you can schedule a single visual job quickly.
- You can schedule your visual job using a DAG and the workflows interface. This way you can combine multiple elements in the same schedule.

Scheduling visual jobs from the editor

You can schedule your visual jobs to run from within the Visual ETL editor. To do this, use a project with the **All capabilities** project profile or another project profile with scheduling enabled in the Tooling blueprint parameters. If you have created a project that needs to be updated to enable scheduling, contact your admin.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, under **Data analytics**, choose **Visual ETL**.
3. Choose the visual job you want to schedule from the list to open it in the editor.
4. Choose the Schedule icon in the upper-right corner of the editor.
5. Under **Schedule name**, enter a name for the schedule.
6. Under **Schedule status**, choose an option to determine whether the schedule will begin running after being created.

    - Choose **Active** to activate the schedule and run the Visual ETL job when the schedule indicates it should run.

- Choose **Paused** to create a schedule that will not run the visual ETL job yet.

7. (Optional) Write a description of the schedule.
8. Choose a schedule type.

   - Choose **One-time** to run the visual ETL job at one specific time.
   - Choose **Recurring** to create a schedule that run the Visual ETL job at multiple times that you choose.

9. Choose the days and times that the schedule will run.
10. Choose **Create schedule**.

You can then view the schedule on the **Schedules** tab of the Visual ETL page in your project.

You can enable project repository auto sync flag when creating or updating the project to ensure the schedules always execute the latest ETL notebook saved to repository. It is recommendede that you test the ETL in draft mode before saving.

Reviewing scheduled visual jobs in the editor

You can review scheduled visual jobs in the Visual ETL interface in Amazon SageMaker Unified Studio. On the schedules page, you can pause, edit, and delete schedules. You can also view the status and other information for a schedule and choose the name of a schedule to view runs and additional data.

To review scheduled queries, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to your project.
3. In the left navigation pane, under **Data analytics**, choose **Visual ETL**.
4. Choose the **Schedules** tab.

You can then pause, edit, or delete a schedule by choosing the three-dot **Actions** menu next to a schedule in the list.

To view information about different times the schedule has run, choose the name of the schedule to view the **Runs** section for that schedule. You can choose the name of a run to see a log and other details for that run.

Scheduling visual jobs with workflows

You can schedule the Visual ETL jobs you authored to run based on a schedule using Workflows. The following is an example of how to do this:

1. Create a Visual ETL flow and name it "mwaa-test".
2. Save your draft flow ("mwaa-test.vetl") to your project.
3. In the left navigation pane, choose **Workflows**, then choose **Create workflow in editor**.
4. You will now see an example DAG template in JupyterLab.
5. Modify the lines of python code as below, then save it as "mwaa_test_dag.py". We will execute the dataflow at 8AM everyday. By default, the dataflow's notebook file is under the path "src/dataflows".

```
WORKFLOW_SCHEDULE = '0 8 * * *'
NOTEBOOK_PATH = 'src/dataflows/mwaa-test.vetl'
dag_id = "workflow-mwaa-test" # optional, set to give your workflow a meaningful name
```

6. Pull the file "dataflows/mwaa-test.vetl" from the project's source code repository to JupyterLab.
7. Navigate back to the Workflows console, now we can see the DAG is created. We can access Airflow UI via the "Actions" dropdown list.
8. Manually trigger the DAG.

Using both external data and fine-grained data in Amazon SageMaker Unified Studio visual ETL jobs

When you use visual ETL, you must select a permission mode to use with your visual ETL flow.

Permission mode is a configuration available to Spark compute resources such as Glue ETL or EMR Serverless. It configures Spark to access different types of data based on the permissions configured for that data. There are two configuration options for permission mode:

- Compatibility mode. This is a configuration for data managed using full-table access, meaning the compute engine can access all rows and columns in the data. Choosing this option enables your compute to work with data assets from AWS and from external systems.
- Fine-grained mode. This is a configuration for data managed using fine-grained access controls, meaning the compute engine can only access specific rows and columns from the full dataset.

Choosing this option enables your Glue ETL to work with data asset subscriptions from Amazon SageMaker Catalog.

In cases where you want to use both data configured with fine-grained access and data from external sources that you connect to your project, you can use two visual ETL jobs and orchestrate them to run together using workflows. To do this, complete the following steps.

Combining jobs with different kinds of data in visual ETL

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project you want to use visual ETL in.
3. In the left navigation pane, under **Data analytics**, choose **Visual ETL**.
4. Choose **Create visual ETL job**.
5. Choose to configure the visual ETL job with full-table access using the AWS Glue ETL compute named **project.spark.fineGrained**.
6. Configure your visual ETL job to ingest the subscribed data to an Amazon S3 target used for temporary staging. This can be done by using the plus icon and adding an lakehouse architecture node as a data source and an Amazon S3 node as a data target, then connecting the nodes on the diagram.
7. Select the lakehouse architecture node and configure it to point to the data you want to use.
    a. Under **Database**, choose the name of the database you want to use.
    b. Under **Table**, choose the name of the table you want to use.
8. Configure the Amazon S3 node to point to a new location.
    a. Under **S3 URI**, create a new Amazon S3 folder name and note the location for later use.
    b. Under **Mode**, select **Overwrite** to clear the Amazon S3 bucket and overwrite it with new data when you are ready to use it again.
    c. (Optional) Configure the other settings as desired.
9. Save the flow and run it using **project.spark.fineGrained** to verify correctness of the results.
10. Create a new visual ETL job that uses the AWS Glue ETL compute named **project.spark.compatibility**.
11. Configure this second visual ETL job to combine the data from the staging S3 location and the data accessible through full-table access to generate the final result.

a. Select the plus icon. Under **Data sources**, select Amazon S3 and place the node on the diagram.

b. Select the Amazon S3 node to configure it.

c. Under **S3 URI**, enter the Amazon S3 folder location you used in the first visual ETL job.

d. Use the plus icon, and under **Data sources**, select an external data source to add to your visual ETL job. Place the node on the diagram.

e. Use the plus icon to add a data target and place the data target node on the diagram.

f. Select the external data source and the data target to edit the configurations as desired and point to the locations you want to use.

g. Use the plus icon, and under **Transforms**, select the **Join** transform. Place the transform on your diagram.

h. Connect the Amazon S3 node containing the data from the first flow and the other data source to the data target using the **Join** transform.

12. Save the second flow and run it using **project.spark.compatibility** to verify correctness of the results.

13. Orchestrate these two visual ETL jobs using Amazon SageMaker Unified Studio workflows. For more information, see the section called “Scheduling and running visual jobs in Identity Center-based domains”.

    Make sure that the workflow is configured so that the first visual ETL job finishes running before the second visual ETL job runs. By default, they'll run in succession, one after the other. This can also be configured using the `wait_for_completion` param.

Best practices for Visual ETL in Identity Center-based domains

To get the most out of Visual ETL in Amazon SageMaker Unified Studio Identity Center-based domains:

- Start with simple flows and gradually increase complexity as you become more familiar with the tool.
- Use data preview features frequently to verify the results of your transformations.
- Leverage custom transformations to standardize and streamline your flows.
- Monitor flows performance and optimize as necessary, using Amazon SageMaker Unified Studio's built-in performance analytics.

By following these guidelines and exploring the various features of Visual ETL, you can efficiently create powerful data integration and transformation Visual ETL flows in Amazon SageMaker Unified Studio.

Data preview

Data preview help you create and test your visual flow using a sample of your data without having to repeatedly run the flow with the full dataset. By using data preview, you can:

- Make sure that you have access to your data sources or data targets.
- Check that the transform is modifying the data in the intended way. For example, if you use a Filter transform, you can make sure that the filter is selecting the right subset of data.
- Check your data. If your dataset contains columns with values of multiple types, the data preview shows a list of tuples for these columns. Each tuple contains the data type and its value.

Authoring with Glue data preview

As you author a visual ETL flow, a new Glue interactive session is automatically started and used to preview your data. Learn more on authoring with Glue data previews in AWS Glue documentation and Glue pricing console page. AWS Glue interactive sessions are billed per second with a 1-minute minimum.

Authoring with SageMaker data preview

Amazon SageMaker Unified Studio offers a new built-in data preview experience, which doesn't incur Glue interactive sessions resources. You may choose "Data preview v2.0" in your preview panel to use the SageMaker data preview, and turn off "Glue data preview". This new data preview experience is optional.

Key benefits

SageMaker data preview provides the following advantages:

- Not incurring Glue interactive sessions resources
- Faster time to start and to preview your data

Limitations

SageMaker data preview has the following support limitations:

File format and compression

Not supported file formats:

- ORC and Avro files are currently not supported.

Supported file formats:

- CSV (uncompressed files, must include headers)
- JSON/JSONL (uncompressed files)
- Parquet (uncompressed files, gzip, zstandard, lz4, and Snappy compression)

Transforms

The following transforms are not supported:

- FlattenTransform
- PivotRowsTransform
- LookupTransform

Data sources

Queries with WHERE clauses are not supported for JDBC connectors (MySQL, PostgreSQL, and SQL Server) and BigQuery connectors.

Performance considerations

Large Parquet files are read by reading the first row group. If your parquet file has very large row groups (2GB or larger) you may see performance degradation. Our recommendation for large parquet files is to continue to use Glue data preview.

SageMaker data preview works best with simple primitive data types such as strings, integers, booleans, and binary data. Complex types in Parquet and JSON files (such as struct, map, or array) are supported, but may run into limitations for extremely nested data types. Our recommendation for complex types is to continue to use Glue data preview.

SageMaker Data Agent

The SageMaker Data Agent in Amazon SageMaker Unified Studio provides intelligent assistance for code generation, error diagnosis, and data analysis recommendations in Notebooks and Query Editor. The agent helps data engineers, analysts, and data scientists who spend significant time on manual setup tasks and boilerplate code when building analytics and ML applications. It generates code and execution plans from natural language prompts. The agent also integrates with data catalogs and business metadata to streamline the development process. The SageMaker Data Agent is available in both IAM and IAM Identity Center (IdC) domains.

Prerequisites

- An open notebook or querybook in Amazon SageMaker Unified Studio
- Access to the SageMaker Data Agent features in your domain

Basic usage

To generate code with the SageMaker Data Agent

1. Type a natural language prompt describing what you want to accomplish.
2. Press **Alt**+**A** (or **Opt**+**A** on Mac) to trigger code generation.
3. Review the generated code in the suggestion panel.
4. Choose **Accept** to add the code to your cell or **Reject** to dismiss it.

To use the SageMaker Data Agent chat panel

1. Open the agent chat panel from the notebook interface.
2. Enter a prompt describing your data analysis needs.
3. Review the agent's plan and suggested approach.
4. The SageMaker Data Agent will generate a collection of cells to implement the analysis plan.
5. Review and execute the generated cells as needed.

To diagnose errors with the SageMaker Data Agent

1. When a cell execution results in an exception, look for the **Diagnose with agent** button.
2. Choose the button to get agent-generated suggestions for fixing the error.
3. Review the suggested fixes and code improvements.
4. Apply the suggestions or use them as guidance for manual fixes.

Getting started with the SageMaker Data Agent for Notebook

The SageMaker Data Agent is available directly within your SageMaker AI notebook interface through two interaction modes:

Agent panel

Access comprehensive multi-step workflows by opening the agent panel from the notebook interface. Use this mode for complex analytical tasks requiring planning, data discovery, and multiple operations.

In-line assistance

Generate code directly within notebook cells by using the inline prompt interface. Use this mode for focused tasks like writing specific queries, creating visualizations, or modifying existing code.

Notebook scheduling and runs

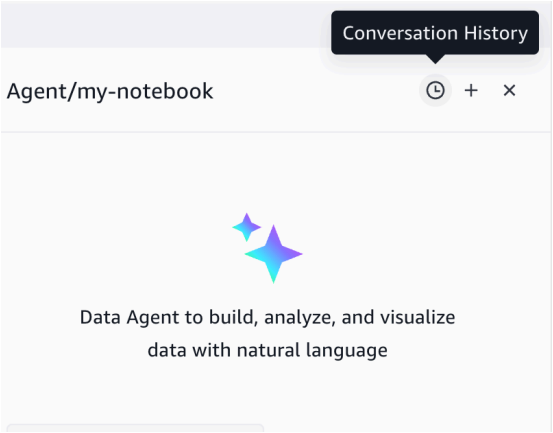
Use the agent to create schedules and start background notebook runs from natural language prompts. For example, you can ask the agent to schedule a notebook to run daily or trigger an on-demand background run. For more information about notebook scheduling, see the section called “Schedule and automate notebook runs”.

Conversation history

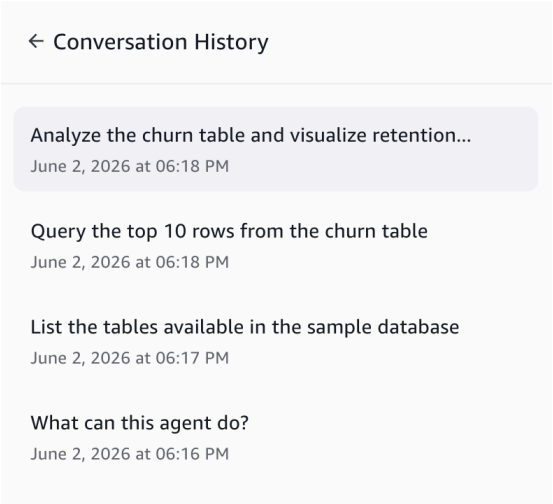
The SageMaker Data Agent saves your conversations automatically, allowing you to return to previous analysis sessions and continue where you left off. Conversation titles are auto-generated after the first message is sent. Conversations are stored for 90 days.

To access conversation history:

1. In the Data Agent chat panel, choose the clock icon in the panel header.



2. Browse the list of past conversations, sorted by most recent activity. Each conversation displays its title and the date of last activity.

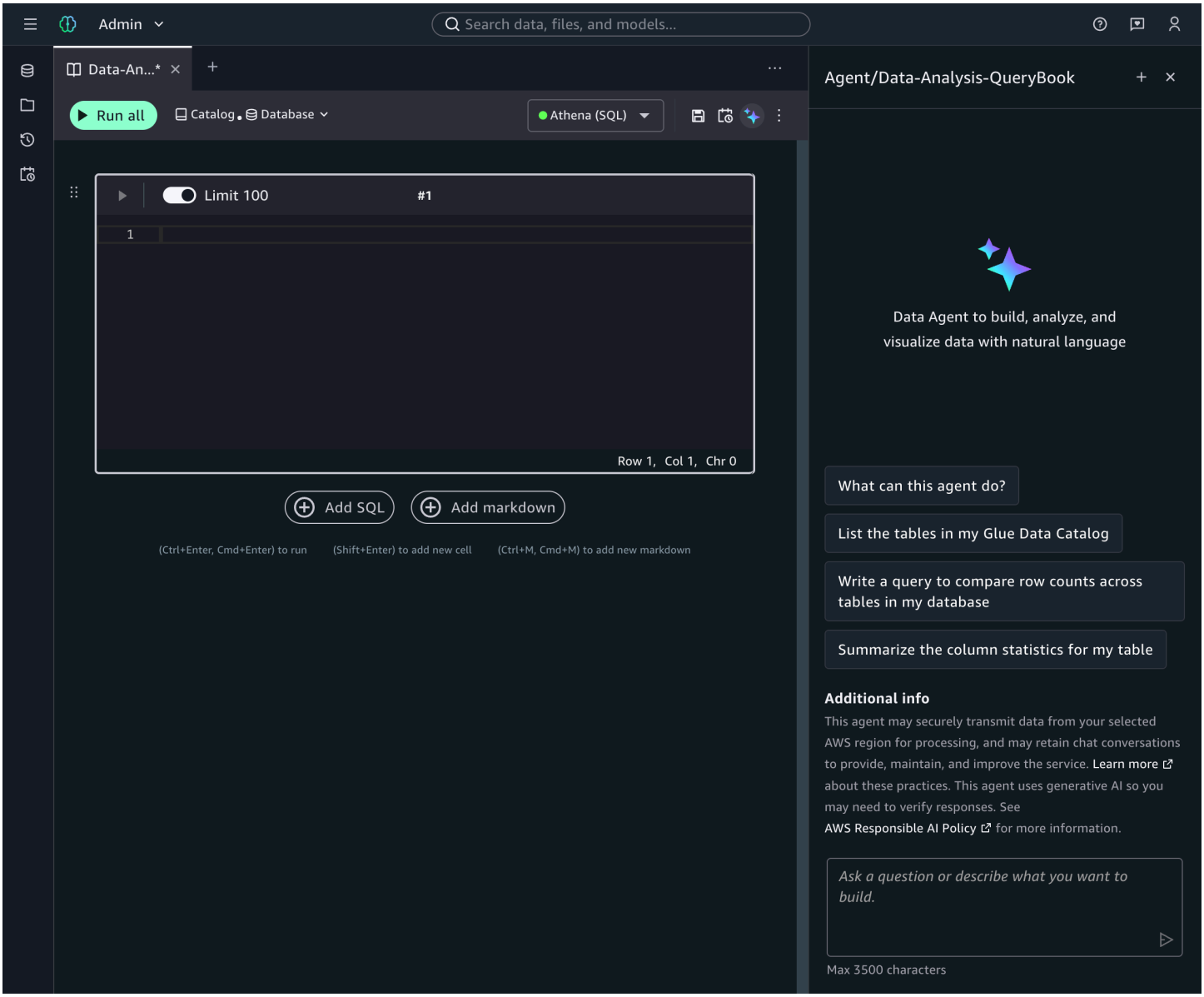


3. Choose a conversation to resume it. The full message history loads, including your prompts and the agent's responses.
4. To return to your current conversation, choose the back arrow.

Getting started with the SageMaker Data Agent for Query Editor

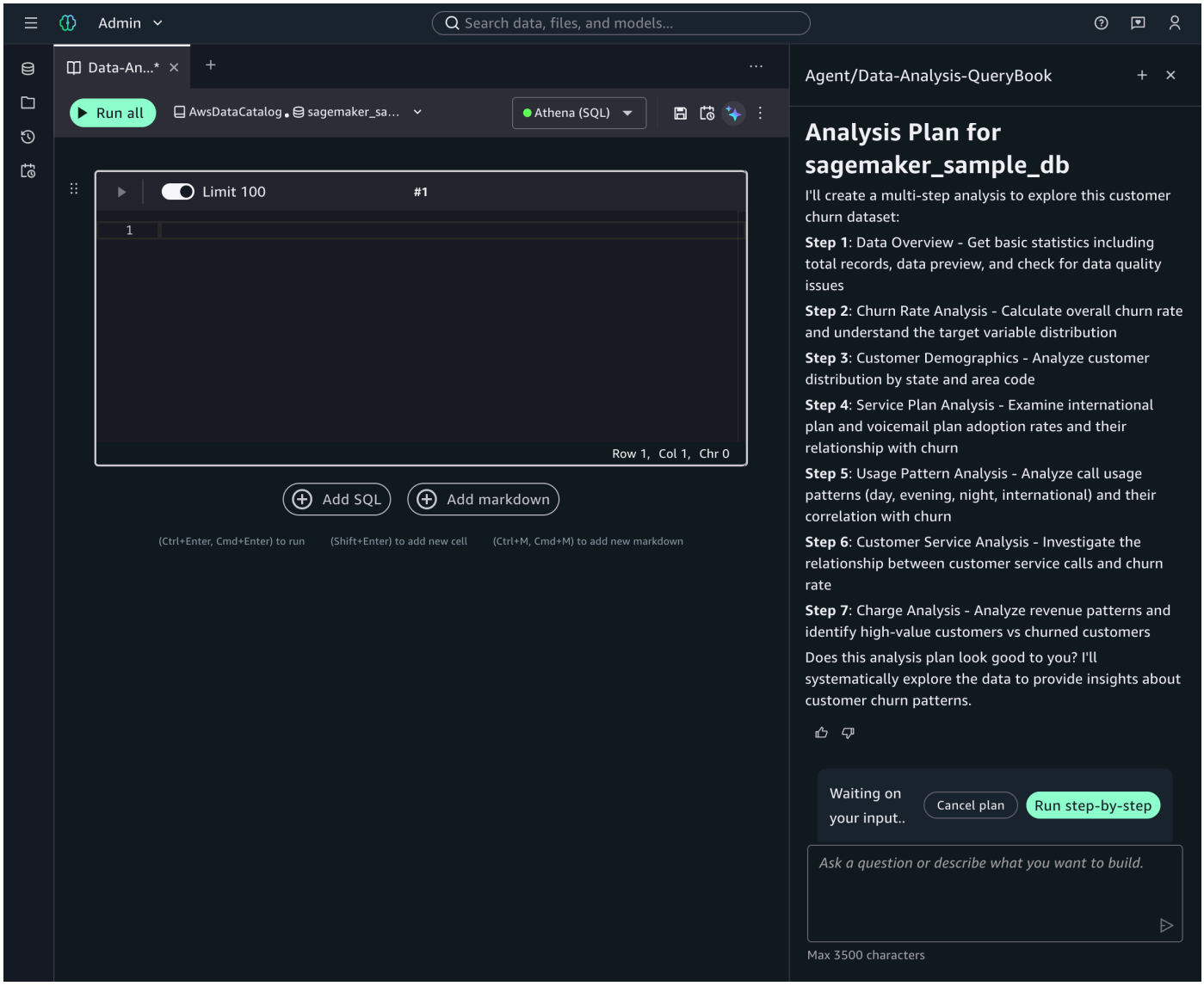
The SageMaker Data Agent in Query Editor provides a conversational SQL development experience. Unlike single-turn SQL generation, the agent supports multi-turn conversations where you can ask follow-up questions, request modifications to generated queries, and receive contextual guidance on query optimization.

The agent is accessible directly within the Query Editor interface through the agent panel.



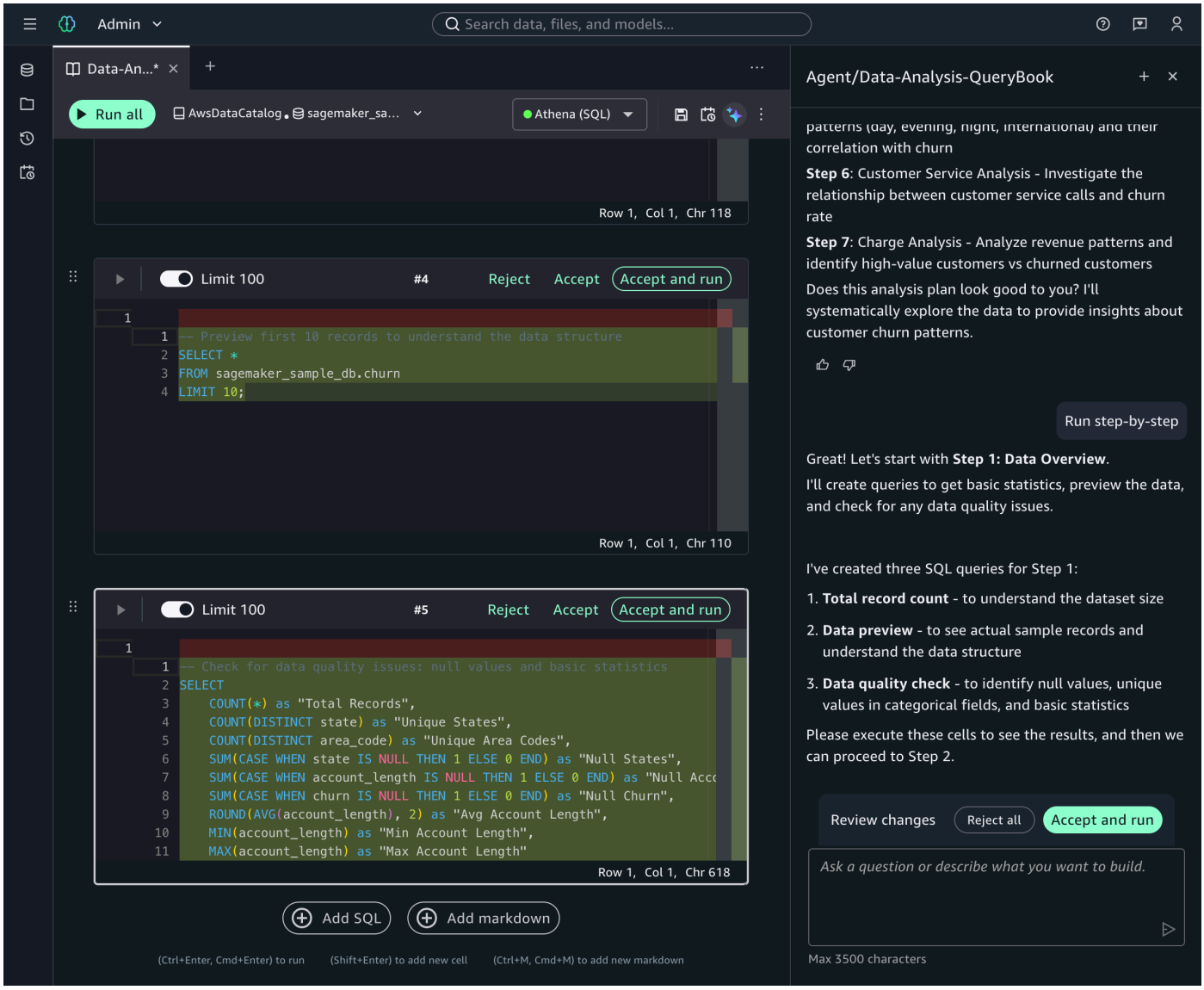
Step-by-step planner

When you describe a complex analytical task, the agent proposes a step-by-step plan to guide your workflow. You can review and approve the plan before the agent generates SQL for each step.



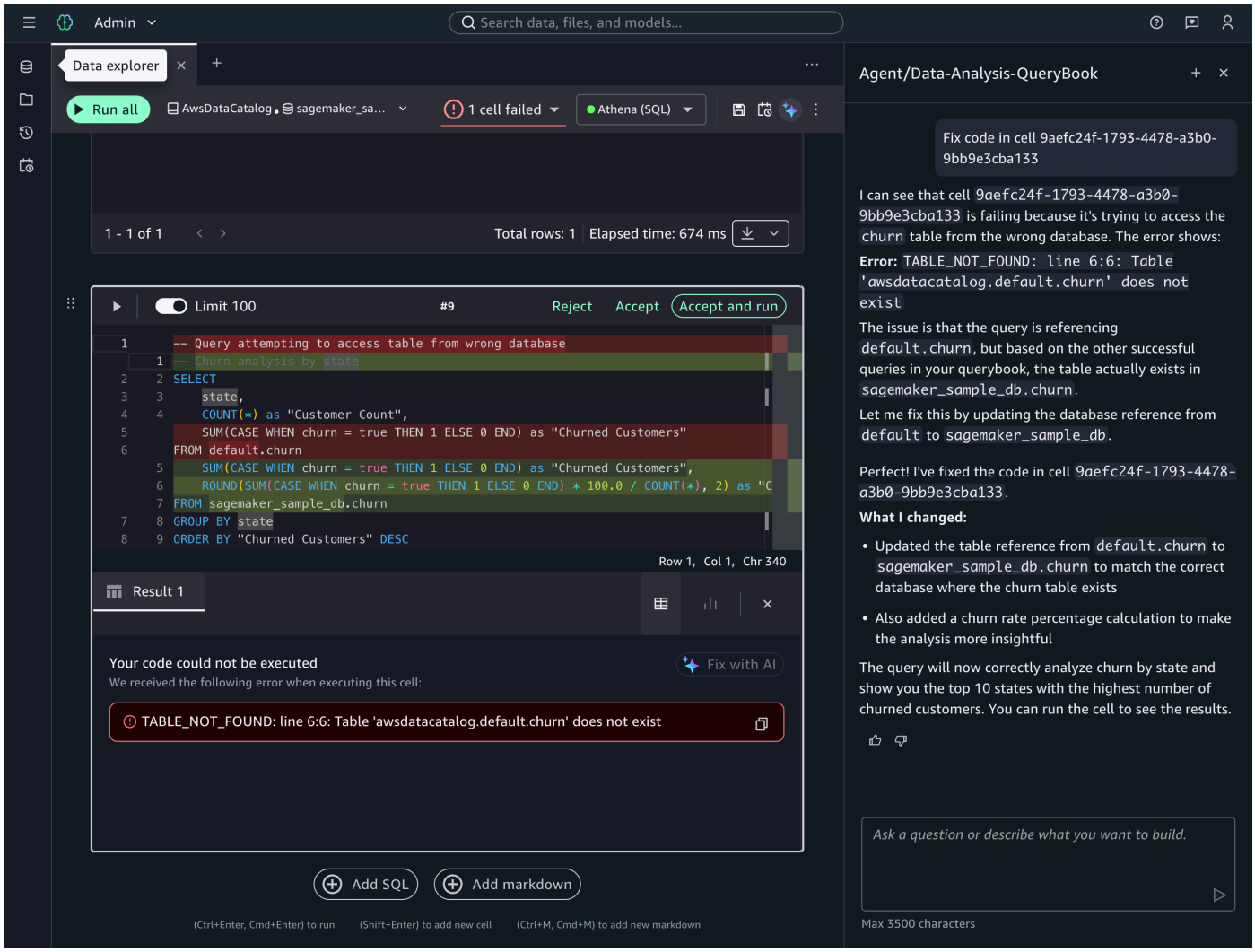
Auto-injection of generated SQL

The agent automatically creates cells with generated SQL directly in your querybook, matching the notebook experience. You can review and run the generated SQL in place.



Fix with AI

When a query fails, the agent can analyze the error and suggest corrections. Use the Fix with AI capability to get agent-generated fixes for failed queries.



To use the SageMaker Data Agent in Query Editor

1. Navigate to a project and in the left navigation pane, under **Data analytics**, choose **Query Editor**.
2. Open the agent panel from the Query Editor interface.
3. Enter a natural language prompt describing your SQL task. For example: *"Write a query that calculates monthly recurring revenue by customer segment for Q4 2025, using the billing.invoices and customer.segments tables."*
4. Review the agent's proposed plan and choose to accept or modify it.
5. The agent generates SQL and injects it into your querybook cells.
6. Review and run the generated SQL. If a query fails, use **Fix with AI** to get suggested corrections.

Example: Multi-turn SQL development

Initial prompt: *"Which Redshift tables in the analytics schema have columns related to customer churn? Show me their schemas."*

The agent queries your data catalog and returns schema information. You can then follow up:

Follow-up prompt: *"Help me find all customers who downgraded their subscription in the last 90 days and calculate the revenue impact by region."*

The agent builds on the previous context to generate a multi-step query plan.

Using the SageMaker Data Agent for data analytics tasks

Data exploration and discovery

The agent helps you explore available data sources by querying AWS Glue Data Catalog metadata through MCP connections. When you request data exploration, the agent retrieves table information, schemas, and relationships from your catalog to understand what data is available.

Example prompt for data analyst

"Show me what customer tables are available and their schemas"

The agent queries your AWS Glue Data Catalog and generates code to display table metadata.

Example prompt

"Calculate monthly revenue by product category from the sales_transactions table"

The agent accesses your AWS Glue Data Catalog to verify the sales_transactions table structure and generates:

```
SELECT
    DATE_TRUNC('month', transaction_date) as month,
    product_category,
    SUM(transaction_amount) as total_revenue,
    COUNT(DISTINCT customer_id) as unique_customers,
    AVG(transaction_amount) as avg_transaction_value
FROM sales_db.sales_transactions
```

```
WHERE transaction_date >= DATE_ADD(CURRENT_DATE(), -180)
GROUP BY DATE_TRUNC('month', transaction_date), product_category
ORDER BY month DESC, total_revenue DESC
```

Building multi-step analytical workflows

For complex analysis, the agent proposes step-by-step plans that break down your objective into discrete operations, requesting approval before executing each step.

Example prompt for data scientist

"Analyze the churn table and visualize retention rate by region"

The agent proposes a multi-step plan:

Now I'll create a plan to analyze the churn table and visualize retention rate by region. Since the table has a 'state' column representing regions and a 'churn' boolean column, I can calculate the retention rate (percentage of customers who didn't churn) for each state.

1. **Step 1: Load Churn Data from Catalog** - Use PySpark to read the churn table from the AWS Glue catalog (sagemaker_sample_db.churn) and preview the data structure.
2. **Step 2: Calculate Retention Rate by Region** - Aggregate the data by state to calculate the retention rate (percentage of customers where churn = False) for each state/region.
3. **Step 3: Visualize Retention Rates** - Create a bar chart showing the retention rate by region/state, sorted for better readability.

Each step generates code in a new cell, maintaining context from previous cells and building upon earlier results.

Cell level inline code modification

The agent provides inline assistance to modify existing code based on your requirements. You can use the agent icon on every cell to trigger a prompt input box and type your request instead of code.

Example workflow

Initial cell content:

```
# Load sales data
sales_df = spark.table("sales_db.transactions")
sales_df.show(10)
```

Inline prompt: *"Add filtering for electronics category and last quarter only"*

Updated cell content:

```
# Load sales data with filters
from datetime import datetime, timedelta

quarter_start = datetime.now() - timedelta(days=90)
sales_df = spark.table("sales_db.transactions") \
    .filter(col("category") == "electronics") \
    .filter(col("transaction_date") >= quarter_start)
sales_df.show(10)
```

Referencing previous cell results

The agent understands your notebook context and can generate code that references data frames and variables from previous cells.

Example workflow

Cell 1:

```
# Customer purchase analysis
customer_purchases = spark.sql("""
    SELECT customer_id, product_category,
           SUM(amount) as total_spent
    FROM sales_db.transactions
    GROUP BY customer_id, product_category
""")
customer_purchases.createOrReplaceTempView("customer_summary")
```

Agent prompt: *"Create a visualization showing the distribution of customer spending from the results above"*

Cell 2 (generated by agent):

```
# Visualize customer spending distribution
import matplotlib.pyplot as plt
import pandas as pd

# Convert to pandas for visualization
spending_data = customer_purchases.toPandas()

plt.figure(figsize=(12, 6))
plt.hist(spending_data['total_spent'], bins=50, edgecolor='black')
plt.xlabel('Total Spending')
plt.ylabel('Number of Customers')
plt.title('Customer Spending Distribution')
plt.grid(True, alpha=0.3)
plt.show()

# Show summary statistics
print(spending_data['total_spent'].describe())
```

Error handling and fix with the SageMaker Data Agent

When code execution fails, the SageMaker Data Agent can analyze error messages and suggest corrections. Use the fix with agent capability to automatically generate corrected code.

The SageMaker Data Agent also helps troubleshoot failed scheduled and background notebook runs. When a scheduled run fails, you can use the agent to identify the root cause and get suggested fixes directly in the notebook. For more information, see the section called “Schedule and automate notebook runs”.

Using Business Context with the SageMaker Data Agent

The SageMaker Data Agent integrates with SageMaker Catalog business context and metadata, enabling data practitioners to discover datasets and generate more accurate SQL and Python code using business terminology instead of cryptic technical table names. This integration allows the

Data Agent to leverage the business context that companies have invested months curating in their SageMaker Catalog, including those synced from Collibra, Atlan, and Alation, to deliver more accurate data discovery and code generation. When your domain has a configured SageMaker Catalog with published assets, the Data Agent uses glossary terms, custom metadata forms, summaries, and README content to find the correct tables for your queries.

Prerequisites

To use the business context integration, you need the following:

- A Amazon SageMaker Unified Studio domain (IAM or IAM Identity Center)
- A SageMaker Catalog configured with assets and data products. For more information, see *Catalog in IDC-based domains*.
- Assets and Data Products enriched with business metadata (glossary terms, metadata forms, summaries, or READMEs)
- For third-party catalog users: metadata sync configured between Collibra, Atlan, or Alation and SageMaker Catalog. For more information, see *Third-party catalog integrations*.

How it works

When you ask the Data Agent a question in a notebook or Query Editor:

1. The agent searches for accessible tables using technical metadata from AWS Glue Data Catalog and Redshift.
2. The agent queries the business context to find assets matching your business terms.
3. The agent merges technical and business metadata to identify the correct tables.
4. The agent generates SQL or PySpark code using the correct catalog, database, table, and column references.

Types of questions you can ask

You can ask the Data Agent questions using the business terminology defined in your catalog. The agent matches your terms against glossary terms, metadata form values, summaries, and README content to find the correct tables.

Discover datasets

Ask the Data Agent to find data using business terms instead of technical table names:

- "What data do I have on customer churn?"
- "Find tables in the Retail industry"
- "What tables contain student data?"
- "Show me Finance domain datasets"
- "Find tables with a column called reg_cd"
- "What data do I have on shoes inventory in current project"

Generate code from business questions

Ask analytical questions and the agent generates SQL or PySpark code using the correct tables and columns:

- "Calculate customer retention rate"
- "Show me EBITDA trends across quarters"
- "Can you generate some table insights on avocado sales"
- "Help me understand what the revenue_metrics table contains"
- "Write a query to calculate retention rate. Use the business definitions from our catalog to pick the correct tables and columns."

Access-aware results

The agent checks whether you have access to matching assets before generating code. If a required table is not accessible, the agent informs you that you need to get access from your administrator.

Understanding search results

The agent searches two scopes:

- **Published assets** – Assets published across your domain that your project has subscribed to.
- **Local assets** – Unpublished assets within your current project.

When multiple assets match your query, the agent favors:

1. Assets subscribed by your current project.

2. Among subscribed assets, those with higher subscription counts (indicating broader usage).

Handling inaccessible assets

If the agent identifies a relevant asset that your project does not have access to, the agent informs you that you need to get access from your administrator. The agent does not generate code against inaccessible tables.

In-cell code generation

When using in-cell code generation (as opposed to the chat panel), the agent uses only accessible assets (subscribed and local).

Supported tools

Data Agent integration with business context is available in the following tools:

- SageMaker Notebooks (Data Notebook)
- Query Editor

Best practices for the SageMaker Data Agent

Be specific about your data sources

Good: *"Calculate total sales by region from the sales_transactions table"*

Less effective: *"Show me sales data"*

Specify desired output format

Good: *"Create a bar chart showing top 10 products by revenue with labels"*

Less effective: *"Visualize product performance"*

Provide context for multi-step workflows

Good: *"Using the customer segments @customer_segment, calculate average lifetime value for each segment"*

Less effective: *"Calculate lifetime value"*

Security and access control for the SageMaker Data Agent

Topics

- Required IAM permissions to use SageMaker Data Agent
- Disable the SageMaker Data Agent in IdC domains

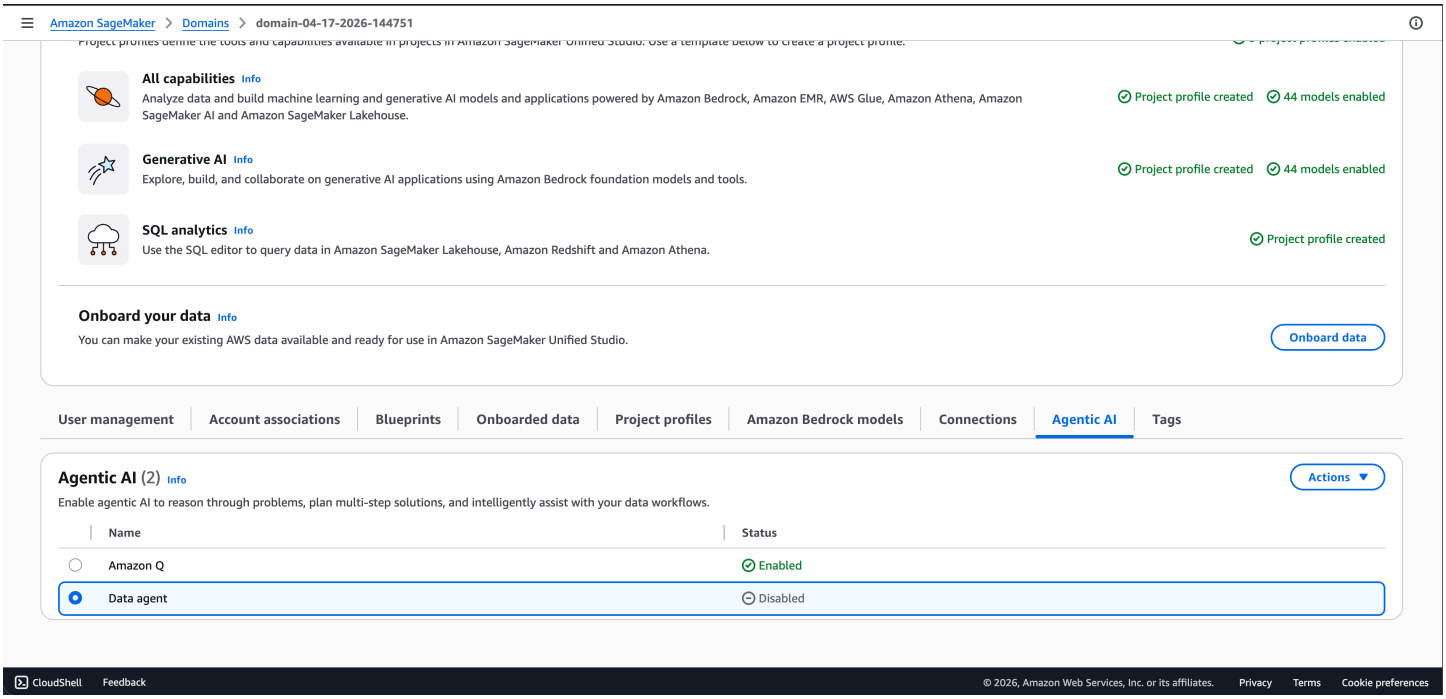
Required IAM permissions to use SageMaker Data Agent

To use the SageMaker Data Agent in Notebooks or Query Editor, your project role needs the required IAM permissions. Your role must have the permissions to invoke the following Amazon DataZone APIs: SendMessage, GenerateCode, StartConversation, GetConversation, and ListConversations.

Disable the SageMaker Data Agent in IdC domains

In IAM Identity Center (IdC) domains, administrators can disable the SageMaker Data Agent through the domain configuration. When disabled, users in the domain don't have access to the Data Agent in Notebooks or Query Editor.

The following image shows the domain configuration option for disabling the SageMaker Data Agent.



Data storage in the SageMaker Data Agent

Content that is stored by SageMaker Data Agent

- Your natural language prompts and questions to the SageMaker Data Agent
- The code and responses generated by the SageMaker Data Agent

Content that is not stored by SageMaker Data Agent

- Code that you write yourself in notebook cells
- Code that you manually modify or change
- Notebook context and metadata
- Data from your AWS Glue Data Catalog or other data sources

AWS Regions where content is processed and stored

Your content collected by the SageMaker Data Agent is stored in the AWS Region where your Amazon SageMaker Unified Studio domain was created. When you use the SageMaker Data Agent in notebooks, your prompts and generated responses remain in your domain's Region.

With cross-region inference, your requests to the SageMaker Data Agent may be processed in a different Region where your content is stored. For more information, see the section called “Cross-region processing in the SageMaker Data Agent”.

Service improvement

To help the SageMaker Data Agent provide the most relevant information, AWS may use certain content for service improvement. This content includes your natural language prompts to the SageMaker Data Agent and the responses generated by the agent.

AWS may use this content, for example, to help the SageMaker Data Agent provide better responses to common questions, fix operational issues, or for debugging.

Content that AWS may use for service improvement

- Your natural language prompts and questions to the SageMaker Data Agent

- The code and responses generated by the SageMaker Data Agent

Content that AWS does not use for service improvement

- Code that you write yourself in notebook cells
- Code that you manually modify or change
- Notebook context and metadata
- Data from your AWS Glue Data Catalog or other data sources

Only Amazon employees will have access to the data. Your trust, privacy, and the security of your Customer Content are our highest priority and ensure that our use complies with our commitments to you. For more information, see Data Privacy FAQ.

How to opt out

To opt out of your data being used to improve AWS services, you can configure an AI services opt-out policy for Amazon DataZone in AWS Organizations. This policy controls data usage across Amazon SageMaker Unified Studio, including the SageMaker Data Agent. For more information, see AI services opt-out policies in the AWS Organizations User Guide.

When you configure an AI services opt-out policy, it has the following effects:

- AWS will delete the data that it collected and stored for service improvement prior to your opt out (if any).
- After you opt out, AWS will no longer collect or store this data.
- AWS will no longer use your content for service improvement.

Cross-region processing in the SageMaker Data Agent

Cross-region inference

The SageMaker Data Agent for Notebooks uses geographic cross-region inference to process natural language requests and generate code responses. With geographic cross-region inference, the agent will automatically route your inference request to optimize performance, maximizing available compute resources and model availability, and providing the best customer experience. The type of geographic cross-region inference used depends on your Amazon SageMaker Unified

Studio domain's Region. Most Regions use geographic cross-region inference, which keeps requests within the same geography. However, some Regions use global cross-region inference, which may route requests to any AWS Region globally.

Cross-Region Inference

The SageMaker Data Agent is powered by Amazon Bedrock and uses cross-region inference to distribute traffic across different AWS Regions to enhance large language model (LLM) inference performance and reliability. With cross-region inference, you get:

- Increased throughput and resilience during high demand periods
- Improved performance

Although cross-region inference does not change where your notebook data or generated code is stored, your natural language prompts, code context, and AWS Glue Data Catalog metadata may be transmitted to different Regions for inference processing. All data is encrypted in transit across Amazon's secure network.

There is no additional cost for using cross-region inference.

Supported regions for cross-region inference

Regions using geographic cross-region inference

For most Regions, cross-region inference requests are kept within the AWS Regions that are part of the same geography where your Amazon SageMaker Unified Studio domain resides. For example, a request made from a notebook in the US East (N. Virginia) Region is routed only to AWS Regions within the United States geography. The following table describes the Regions that your requests might be routed to, depending on the geography where the request originated.

Supported geography	Inference regions
United States	US East (N. Virginia) (us-east-1), US West (Oregon) (us-west-2), US East (Ohio) (us-east-2)
Europe	Europe (Frankfurt) (eu-central-1), Europe (Ireland) (eu-west-1), Europe (Paris) (eu-west-3), Europe (Stockholm) (eu-north-1)

Supported geography	Inference regions
Japan	Asia Pacific (Tokyo) (ap-northeast-1), Asia Pacific (Osaka) (ap-northeast-3)
Australia	Asia Pacific (Sydney) (ap-southeast-2), Asia Pacific (Melbourne) (ap-southeast-4)

Regions using global cross-region inference

Important

The following AWS Regions use global cross-region inference. An inference request made by the SageMaker Data Agent when your Amazon SageMaker Unified Studio domain's Region is listed below will be securely routed to all available compute resources across all global commercial AWS Regions, to optimize performance and availability:

- Asia Pacific (Mumbai) (ap-south-1)
- Asia Pacific (Seoul) (ap-northeast-2)
- Asia Pacific (Singapore) (ap-southeast-1)
- South America (São Paulo) (sa-east-1)
- Canada (Central) (ca-central-1)

Code spaces in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio provides compute spaces for integrated development environments (IDEs) that you can use to code and develop your resources. When you create and use these IDEs in a Amazon SageMaker Unified Studio project, you have access to all the data in that project and can share coding work with other project members. Each space maintains its own persistent Amazon EBS volume, compute instance, and runtime configuration, so your files, data, and session state are fully isolated between spaces. You can scale compute and storage independently per space, and pause or resume spaces at any time.

The code spaces experience differs depending on your domain type. Choose the section that matches your domain configuration:

- the section called "Code spaces in IAM domains"—For domains configured with IAM roles.
- the section called "Code spaces in Identity Center domains"—For domains configured with AWS IAM Identity Center.

Code spaces in IAM domains

In Amazon SageMaker Unified Studio projects for IAM domains, you can create and manage multiple code spaces—individually configured development environments—within a single project. Each code space maintains its own persistent Amazon EBS volume, compute instance, and runtime configuration, enabling parallel experimentation with full isolation.

To access code spaces, choose **Code spaces** from the left sidebar in your project. The code spaces list displays all your existing spaces with their application type, status, and last modified time. You can open spaces in a dedicated browser tab or connect to a local IDE such as VS Code, Kiro, or Cursor using the local IDE icons in the actions area for each space.

Creating a code space in IAM domains

To create a code space, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to your project and choose **Code spaces** from the left sidebar.
3. Choose **Create**.

4. Select an application type by choosing one of the available cards:
   - **JupyterLab**—A web-based interactive development environment for notebooks, code, and data.
   - **Code Editor**—A lightweight code editor for writing and editing code.
5. Under **Instance**, choose an instance type (for example, ml.t3.medium).
6. Under **Image**, choose the image you want to use (for example, SageMaker Distribution 2.11).
7. Under **EBS space storage**, enter a value from 16 to 100 GB.
8. (Optional) Expand **Advanced settings** to configure additional options:
   - Enable remote access for local IDE connections.
   - Attach a lifecycle configuration.
   - Set an idle timeout.
   - Attach a custom filesystem.
9. Choose **Create**.

After the space is created, it appears in the code spaces list. Choose **Open** to launch the space in a new browser tab, or choose a local IDE icon (VS Code, Kiro, or Cursor) to open the space in your preferred local IDE.

Managing code spaces in IAM domains

You can manage your code spaces from the code spaces list view. Choose the three-dot menu (…) under **Actions** for any space to access the following options:

- **View details**—View information about the space including its configuration and status.
- **Configure**—Change the space configuration such as instance type, image, or storage size.
- **Stop**—Stop the running space. Active processes are paused and unsaved changes are discarded.
- **Delete**—Permanently delete the space and its associated Amazon EBS volume.

> **Note**
>
> Saving configuration changes to a running space requires a restart. Active processes are paused and unsaved changes are discarded during the restart.

Configuring a code space in IAM domains

To configure an existing code space, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to your project and choose **Code spaces** from the left sidebar.
3. Choose the three-dot menu (…) under **Actions** next to the space you want to configure, then choose **Configure**.
4. Make the desired changes to the space configuration.
5. Choose one of the following:

   - **Save and restart**—If the space is running, saves changes and restarts the space.
   - **Stop space**—Stops the space without saving changes.
   - **Cancel**—Discards changes and returns to the code spaces list.

Amazon Q support in code spaces for IAM domains

Amazon Q free tier support is available across all code space types in IAM domains. When your administrator has subscribed to Amazon Q Developer and added it to your domain, you can access the Amazon Q chat interface and inline code suggestions in both JupyterLab and Code Editor spaces.

Code spaces in Identity Center domains

In Identity Center domains, Amazon SageMaker Unified Studio provides spaces that you access from the **Spaces** tab on the **Compute** page. JupyterLab is the IDE application available by default, and you can create additional spaces as desired.

Creating a code space in Identity Center domains

To create a space, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project you want to create a space in. You can do this by choosing **Browse all projects** from the center menu.

3. In the **Build** menu, choose **Spaces** to navigate to the **Spaces** tab on the **Compute** page.
4. Choose **Create space**.
5. Under **Name**, enter a name for the space. This name must be unique. No spaces in the project can use the same name.
6. Under **Application**, choose the IDE application you want to use with the space.
7. Under **Instance**, choose an Amazon EC2 instance that is most compatible with your use case.

> **Note**
>
> If you use a GPU instance type when configuring your Code Editor application, you must also use a GPU-based image. Within a space, your data is stored in an Amazon EBS volume that persists independently from the life of an instance. You won't lose your data when you change instances.

8. Under **Image**, choose the image you want to use.
9. Under **EBS space storage**, enter a value from 16 to 100 to choose a storage size.
10. (Optional) Choose the lifecycle configuration for the space.
    a. Find your domain ID. For instructions, see the section called “Get project details”.
    b. Find your user profile ID:
        i. In the **Compute** panel, choose the **Spaces** tab.
        ii. The user profile ID is the string following `default-` of the default JupyterLab space.
    c. Create a lifecycle configuration for Amazon SageMaker Unified Studio and reference your ID values from step b.

    > **Note**
    >
    > Attaching the lifecycle configuration to your Amazon SageMaker AI domain and user profile through the AWS CLI is not currently supported. Use the console to create lifecycle configurations.

    d. Under **Lifecycle configuration**, choose the lifecycle you created.
11. Under **Idle time**, enter the amount of time before the space will time out and stop running.
12. (Optional) Attach a custom file system to your space.

a. To attatch a custom file system to a domain see Attaching a custom file system to a domain with the AWS CLI, and reference your ID values from step 10.

b. Under **Attach custom file system**, attach the file system.

13. Choose **Create and start space** to create the space and have it start running.

It might take up to 5 minutes for Amazon SageMaker Unified Studio to finish creating the space. After the space is created, you can view it in the list on the **Spaces** tab and choose **Open** to launch the IDE in your browser.

You can then perform various actions using the three-dot **Actions** menu on the **Spaces** tab in Amazon SageMaker Unified Studio. You can edit the space, start or stop running the application, delete the space, or view information about the space.

Editing a code space in Identity Center domains

Project members can edit a space at any time to change certain configurations. Before editing a space, make sure that you have saved any changes to your work in that space.

> **Note**
>
> A space's name and IDE application cannot be changed.

To edit configurations for a space, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project you want to edit a space in. You can do this by choosing **Browse all projects** from the center menu.
3. In the **Build** menu, choose **Spaces** to navigate to the **Spaces** tab on the **Compute** page.
4. Choose the three-dot **Actions** menu next to the space you want to edit. Then choose **Configure space**.
5. Make the desired changes, then choose **Save and restart** if the space is running, or **Save and start** if the space is stopped.

> **Note**
>
> Saving changes to a running space requires a restart, and any unsaved changes will be lost.

Deleting a code space in Identity Center domains

Project members can choose to delete a space in Amazon SageMaker Unified Studio. Deleting a space is permanent and cannot be undone. Before deleting a space, make sure that you have saved all your data and that you are sure that this is the action you would like to take.

> **Note**
>
> You cannot delete the default JupyterLab space in Amazon SageMaker Unified Studio.

To delete a space, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project you want to delete a space in. You can do this by choosing **Browse all projects** from the center menu.
3. In the **Build** menu, choose **Spaces** to navigate to the **Spaces** tab on the **Compute** page.
4. Choose the three-dot **Actions** menu next to the space you want to delete. Then choose **Delete**.
5. Review the information in the window provided, then type **confirm** where indicated.
6. Choose **Delete this space**.

Amazon Q support in code spaces for Identity Center domains

Amazon Q free tier support is available across all code space types in IAM domains. When your administrator has subscribed to Amazon Q Developer and added it to your domain, you can access the Amazon Q chat interface and inline code suggestions in both JupyterLab and Code Editor spaces.

Bring your own image (BYOI)

An image is a file that identifies the kernels, language packages, and other dependencies required to run your applications. It includes:

- Programming languages (like Python or R)
- Kernels
- Libraries and packages
- Other necessary software

Amazon SageMaker AI Distribution (`sagemaker-distribution`) is a set of Docker images that include popular frameworks and packages for machine learning, data science, and visualization.

You can also create your own custom image, using an Amazon SageMaker AI Distribution image as a base image, to bring your own image (BYOI). You may want to BYOI when:

- You need a specific version of a programming language or library
- You want to include custom tools or packages
- You're working with specialized software not available in the standard images

Topics

- Dockerfile specifications
- How to BYOI
- Launch your custom image in Amazon SageMaker Unified Studio
- Speed up container startup with SOCI
- Detach and clean up custom image resources

Dockerfile specifications

To bring your own image (BYOI), ensure that the following specifications are satisfied when you create your `Dockerfile`.

- **Base image specifications**: You must use one of the base images from Amazon SageMaker AI Distribution (`sagemaker-distribution`), with the following specifications. These contain

important extensions that are required to execute an image on Amazon SageMaker Unified Studio.

- The base image must be `FROM public.ecr.aws/sagemaker/sagemaker-distribution:version`. You can copy the Image URI of an image from the Image tags tab in sagemaker-distribution.
- The chosen image *version* must be greater or equal to the following.
  - For CPU, `2.6-cpu`
  - For GPU: `2.6-gpu`

- Follow the Custom image specifications in the *SageMaker AI Developer Guide*.

Dockerfile example

The following is an example `Dockerfile` that meets the above criteria. The *version* in this example must satisfy the specification above.

> **Note**
>
> Adding `ENTRYPOINT` in the `Dockerfile` will not work as expected. If you would like to configure a custom entrypoint, you will need to update your `ContainerConfig`. For an example, see Update container configuration in the *SageMaker AI Developer Guide*.

```
FROM public.ecr.aws/sagemaker/sagemaker-distribution:version

ARG NB_USER="sagemaker-user"
ARG NB_UID=1000
ARG NB_GID=100

ENV MAMBA_USER=$NB_USER

USER root

RUN apt-get update
RUN micromamba install sagemaker-inference --freeze-installed --yes --channel conda-forge --name base

USER $MAMBA_USER
```

How to BYOI

When you bring your own image (BYOI) to Amazon SageMaker Unified Studio, you attach a custom image to an Amazon SageMaker Unified Studio project. The following page provides instructions on how to bring your custom image to your Amazon SageMaker Unified Studio project.

Topics

- Prerequisites
- Step 1: Create your custom image
- Step 2: Get the SageMaker AI domain name associated with your Amazon SageMaker Unified Studio project
- Step 3: Attach your custom image using the SageMaker AI domain
- Step 4: Access your custom image in Amazon SageMaker Unified Studio

Prerequisites

You will need to complete the following prerequisites to bring your own image to Amazon SageMaker Unified Studio.

- Create an Amazon SageMaker Unified Studio project. For more information, see Create a new project.
- Set up the Docker application. For more information, see Get started in the *Docker documentation*.
- Install the latest AWS CLI by following the steps in Getting started with the AWS CLI in the *AWS Command Line Interface User Guide for Version 2*.
- Permissions to access the Amazon Elastic Container Registry (Amazon ECR) service. For more information, see Amazon ECR Managed Policies in the *Amazon ECR User Guide*.
- An AWS Identity and Access Management role that has the AmazonSageMakerFullAccess policy attached.

Step 1: Create your custom image

> **Important**
>
> Ensure that you are using the Dockerfile specifications in the following instructions.

Follow the steps in Create a custom image and push to Amazon ECR in the *SageMaker AI Developer Guide*.

Step 2: Get the SageMaker AI domain name associated with your Amazon SageMaker Unified Studio project

An associated SageMaker AI domain is created when you create a Amazon SageMaker Unified Studio project. You will need the SageMaker AI domain name before proceeding to the next step. For instructions, see View the SageMaker AI domain details associated with your project.

Step 3: Attach your custom image using the SageMaker AI domain

To attach your custom image to your Amazon SageMaker Unified Studio project, you must attach your custom image to your SageMaker AI domain. Follow the steps in Attach your custom image to your domain in the *SageMaker AI Developer Guide*, using the SageMaker AI domain obtained from above.

Step 4: Access your custom image in Amazon SageMaker Unified Studio

Once your custom image is attached to your Amazon SageMaker Unified Studio project, the users with access to your project can access it. For instructions on how users can access the custom images, see Launch your custom image in Amazon SageMaker Unified Studio.

Launch your custom image in Amazon SageMaker Unified Studio

The following section provides information on how to launch your custom image in Amazon SageMaker Unified Studio. If you have not already created a custom image, see How to BYOI.

To launch your custom image in Amazon SageMaker Unified Studio

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your IAM Identity Center (SSO) or AWS credentials.
2. Navigate to the Amazon SageMaker Unified Studio home page by choosing the icon located at the top left corner of the page.
3. You can choose to create a project or choose an existing one.
4. Once you choose a project you will be taken to the **Project overview** page.
5. On the **Project overview** page, open your notebook. To do so:

    a. Choose **New** to expand the dropdown menu, located at the top right of the page.

   b. Choose **Notebook**, then continue with the following steps.
6. Once your notebook is open, choose **Configure space**, located at the top right of the page.
7. In the **Configure space** section, you can choose the image from the **Image** dropdown menu.
8. After you have verified the settings, choose **Save and restart**.

Once completed, this will start a new JupyterLab instance with the custom image.

Speed up container startup with SOCI

SOCI (Seekable Open Container Initiative) indexing enables lazy loading of custom container images in Amazon SageMaker Unified Studio or Amazon SageMaker Unified Studio. SOCI significantly reduces startup times by roughly 30-70% for your custom Bring your own image (BYOI) containers. Latency improvement varies depending on the size of the image, hosting instance availability, and other application dependencies. SOCI creates an index that allows containers to launch with only necessary components, fetching additional files on-demand as needed.

SOCI addresses slow container startup times, that interrupt iterative machine learning (ML) development workflows, for custom images. As ML workloads become more complex, container images have grown larger, creating startup delays that hamper development cycles.

For more information on SOCI indexing and how to use them, see Speed up container startup with SOCI in the *Amazon SageMaker Unified Studio Developer Guide*.

Detach and clean up custom image resources

An associated SageMaker AI domain is created when you create an Amazon SageMaker Unified Studio project. You will need the SageMaker AI domain details to detach and clean up your custom image resources. For instructions, see View the SageMaker AI domain details associated with your project.

Follow the instructions in Detach and clean up custom image resources in the *Amazon SageMaker AI Developer Guide*.

Using Amazon SageMaker Unified Studio Library for Python

The Amazon SageMaker Unified Studio library is an open source library for interacting with Amazon SageMaker Unified Studio resources. With this library, you can access resources such as

domains, projects, connections, and databases, all in one place with minimal code. The following examples demonstrate how to use the library in local and remote sessions.

> **Note**
>
> For IAM-based domains, the Amazon SageMaker Unified Studio Library for Python is supported only when using Space Distribution Image version 2.11+ or 3.6+ in JupyterLab Notebooks or the Code Editor. Earlier versions (e.g., 2.9 or 3.2) do not support the Amazon SageMaker Unified Studio Library for Python.

Using ClientConfig

If using `ClientConfig` for supplying credentials or changing the AWS region name, the `ClientConfig` object will need to be supplied when initializing any further Amazon SageMaker Studio objects, such as `Domain` or `Project`. If using non-prod endpoint for an AWS service, it can also be supplied in the `ClientConfig`. Note: In sagemaker space, datazone endpoint is by default fetched from the metadata JSON file.

```
from sagemaker_studio import ClientConfig, Project
conf = ClientConfig(region="eu-west-1")
proj = Project(config=conf)
```

Domain

`Domain` can be initialized using the following command.

```
from sagemaker_studio import Domain
dom = Domain()
```

If you are not using the Amazon SageMaker Studio within Amazon SageMaker Unified Studio JupyterLab IDE, you will need to provide the ID of the domain you want to use.

```
dom = Domain(id="123456")
```

Domain Properties

A `Domain` object has several string properties that can provide information about the domain that you are using.

```
dom.id
dom.root_domain_unit_id
dom.name
dom.domain_execution_role
dom.status
dom.portal_url
```

Project

`Project` can be initialized using the following command.

```
from sagemaker_studio import Project
proj = Project()
```

If you are not using the Amazon SageMaker Studio library within the Amazon SageMaker Unified Studio JupyterLab IDE, you will need to provide either the ID or name of the project you would like to use and the domain ID of the project.

```
proj = Project(name="my_proj_name", domain_id="123456")
```

Project properties

A `Project` object has several string properties that can provide information about the project that you are using.

```
proj.id
proj.name
proj.domain_id,
proj.project_status,
proj.domain_unit_id,
proj.project_profile_id
proj.user_id
```

IAM Role ARN

To retrieve the project IAM role ARN, you can retrieve the `iam_role` field. This gets the IAM role ARN of the default IAM connection within your project.

```
proj.iam_role
```

AWS KMS Key ARN

If you are using a AWS KMS key within your project, you can retrieve the `kms_key_arn` field.

```
proj.kms_key_arn
```

S3 Path

One of the properties of a `Project` is `s3`. You can access various S3 paths that exist within your project.

```
# S3 path of project root directory
proj.s3.root
# S3 path of datalake consumer Glue DB directory (requires DataLake environment)
proj.s3.datalake_consumer_glue_db
# S3 path of Athena workgroup directory (requires DataLake environment)
proj.s3.datalake_athena_workgroup
# S3 path of workflows output directory (requires Workflows environment)
proj.s3.workflow_output_directory
# S3 path of workflows temp storage directory (requires Workflows environment)
```

```
proj.s3.workflow_temp_storage
# S3 path of EMR EC2 log destination directory (requires EMR EC2 environment)
proj.s3.emr_ec2_log_destination
# S3 path of EMR EC2 log bootstrap directory (requires EMR EC2 environment)
proj.s3.emr_ec2_certificates
# S3 path of EMR EC2 log bootstrap directory (requires EMR EC2 environment)
proj.s3.emr_ec2_log_bootstrap
```

Other Environment S3 Paths

You can also access the S3 path of a different environment by providing an environment ID.

```
proj.s3.environment_path(environment_id="env_1234")
```

Connections

You can retrieve a list of connections for a project, or you can retrieve a single connection by providing its name.

```
proj_connections: List[Connection] = proj.connections
proj_redshift_conn = proj.connection("my_redshift_connection_name")
```

Each `Connection` object has several properties that can provide information about the connection.

```
proj_redshift_conn.name
proj_redshift_conn.id
proj_redshift_conn.physical_endpoints[0].host
proj_redshift_conn.iam_role
```

Retrieving AWS client with SDK for Python (Boto3)

You can retrieve an SDK for Python (Boto3) AWS client initialized with the connection's credentials.

Example

The following example shows how to create a Redshift client using create_client() from Redshift connection.

```
redshift_connection: Connection = proj.connection("project.redshift")
redshift_client = redshift_connection.create_client()
```

Some connections are directly associated with an AWS service, and will default to using that AWS service's client if no service name is specified. Those connections are listed in the following table.

Connection Type	AWS Service Name
ATHENA	athena
DYNAMODB	dynamodb
REDSHIFT	redshift
S3	s3
S3_FOLDER	s3

For other connection types, you must specify an AWS service name.

Example

See the following example for details.

```
iam_connection: Connection = proj.connection("project.iam")
glue_client = iam_connection.create_client("glue")
```

Connection data

To retrieve all properties of a `Connection`, you can access the `data` field to get a `ConnectionData` object. `ConnectionData` fields can be accessed using the dot notation (e.g. `conn_data.top_level_field`). For retrieving further nested data within `ConnectionData`, you can access it as a dictionary. For example: `conn_data.top_level_field["nested_field"]`.

```
conn_data: ConnectionData = proj_redshift_conn.data
red_temp_dir = conn_data.redshiftTempDir
lineage_sync = conn_data.lineageSync
lineage_job_id = lineage_sync["lineageJobId"]
spark_conn = proj.connection("my_spark_glue_connection_name")
id = spark_conn.id
env_id = spark_conn.environment_id
glue_conn = spark_conn.data.glue_connection_name
workers = spark_conn.data.number_of_workers
glue_version = spark_conn.data.glue_version
# Fetching tracking server ARN and tracking server name from an MLFlow connection
ml_flow_conn = proj.connection('<my_ml_flow_connection_name>')
tracking_server_arn = ml_flow_conn.data.tracking_server_arn
tracking_server_name = ml_flow_conn.data.tracking_server_name
```

Secrets

Retrieve the secret (username, password, other connection-related metadata) for the connection using the following property.

```
snowflake_connection: Connection = proj.connection("project.snowflake")
secret = snowflake_connection.secret
```

Secrets can be a dictionary containing credentials or a single string depending on the connection type.

Catalogs, databases, and tables

If your `Connection` is of the `LAKEHOUSE` or `IAM` type, you can retrieve catalogs, databases, and tables within a project.

Catalogs

If your Connection is of the `LAKEHOUSE` or `IAM` type, you can retrieve a list of catalogs, or a single catalog by providing its id.

```
conn_catalogs: List[Catalog] = proj.connection().catalogs
my_default_catalog: Catalog = proj.connection().catalog()
```

```
my_catalog: Catalog = proj.connection().catalog("1234567890:catalog1/sub_catalog")
proj.connection("<lakehouse_connection_name>").catalogs
```

Each `Catalog` object has several properties that can provide information about the catalog.

```
my_catalog.name
my_catalog.id
my_catalog.type
my_catalog.spark_catalog_name
my_catalog.resource_arn
```

Databases

You can retrieve a list of databases or a single database within a catalog by providing its name.

```
my_catalog: Catalog
catalog_dbs: List[Database] = my_catalog.databases
my_db: Database = my_catalog.database("my_db")
```

Each `Database` object has several properties that can provide information about the database.

```
my_db.name
my_db.catalog_id
my_db.location_uri
my_db.project_id
my_db.domain_id
```

Tables

You can also retrieve a list of tables or a specific table within a `Database`.

```
my_db_tables: List[Table] = my_db.tables
my_table: Table = my_db.table("my_table")
```

Each `Table` object has several properties that can provide information about the table.

```
my_table.name
my_table.database_name
my_table.catalog_id
my_table.location
```

You can also retrieve a list of the columns within a table. `Column` contains the column name and the data type of the column.

```
my_table_columns: List[Column] = my_table.columns
col_0: Column = my_table_columns[0]
col_0.name
col_0.type
```

Utility Methods

The Amazon SageMaker Unified Studio SDK provides utility modules for common data operations including SQL execution, DataFrame operations, and Spark session management.

SQL Utilities

The SQL utilities module provides a simple interface for executing SQL queries against various database engines within Amazon SageMaker Unified Studio. When no connection is specified, queries are executed locally using DuckDB.

Supported Database Engines

The following database engines are supported:

- Amazon Athena
- Amazon Redshift
- MySQL
- PostgreSQL
- Snowflake
- Google BigQuery

- Amazon DynamoDB
- Microsoft SQL Server
- DuckDB (default when no connection specified)

Basic Usage

Import the SQL utilities:

```
from sagemaker_studio import sqlutils
```

Execute SQL with DuckDB (No Connection)

When no connection is specified, queries are executed locally using DuckDB:

```
# Simple SELECT query
result = sqlutils.sql("SELECT 1 as test_column")
result

# Query with literal values
result = sqlutils.sql("SELECT * FROM table WHERE id = 123")
```

Execute SQL with Project Connections

Use existing project connections by specifying either connection name or ID:

```
# Using connection name
result = sqlutils.sql(
    "SELECT * FROM my_table",
    connection_name="my_athena_connection"
)

# Using connection ID
result = sqlutils.sql(
    "SELECT * FROM my_table",
    connection_id="conn_12345"
)
```

Examples by Database Engine

Amazon Athena

```
# Query Athena using project connection with parameters
result = sqlutils.sql(
    """
    SELECT customer_id, order_date, total_amount
    FROM orders
    WHERE order_date >= :start_date
    """,
    parameters={"start_date": "2024-01-01"},
    connection_name="project.athena"
)

# Create external table in Athena
sqlutils.sql(
    """
    CREATE EXTERNAL TABLE sales_data (
        customer_id bigint,
        order_date date,
        amount decimal(10,2)
    )
    LOCATION 's3://my-bucket/sales-data/'
    """,
    connection_name="project.athena"
)

# Insert data using Create Table As Select (CTAS)
sqlutils.sql(
    """
    CREATE TABLE monthly_sales AS
    SELECT
        DATE_TRUNC('month', order_date) as month,
        SUM(amount) as total_sales
    FROM sales_data
    GROUP BY DATE_TRUNC('month', order_date)
    """,
    connection_name="project.athena"
)
```

Amazon Redshift

```
# Query Redshift with parameters
result = sqlutils.sql(
    """
    SELECT product_name, category, price
    FROM products
    WHERE category = :category
    AND price > :min_price
    """,
    parameters={"category": "Electronics", "min_price": 100},
    connection_name="project.redshift"
)

# Create table in Redshift
sqlutils.sql(
    """
    CREATE TABLE customer_summary (
        customer_id INTEGER PRIMARY KEY,
        total_orders INTEGER,
        total_spent DECIMAL(10,2),
        last_order_date DATE
    )
    """,
    connection_name="project.redshift"
)

# Insert aggregated data
sqlutils.sql(
    """
    INSERT INTO customer_summary
    SELECT
        customer_id,
        COUNT(*) as total_orders,
        SUM(amount) as total_spent,
        MAX(order_date) as last_order_date
    FROM orders
    GROUP BY customer_id
    """,
    connection_name="project.redshift"
```

```
)

# Update existing records
sqlutils.sql(
    """
    UPDATE products
    SET price = price * 1.1
    WHERE category = 'Electronics'
    """,
    connection_name="project.redshift"
)
```

Advanced Usage

Working with DataFrames

The sql function returns pandas DataFrames for SELECT queries, and row counts for DML operations:

```
import pandas as pd

# Execute query and get DataFrame
df = sqlutils.sql("SELECT * FROM sales_data", connection_name="redshift_conn")

# Use pandas operations
summary = df.groupby('region')['sales'].sum()
print(summary)

# Save to file
df.to_csv('sales_report.csv', index=False)

# DML operations return row counts
rows_affected = sqlutils.sql(
    "UPDATE inventory SET quantity = quantity - 1 WHERE product_id = 123",
    connection_name="redshift_conn"
)
print(f"Updated {rows_affected} inventory records")
```

Parameterized Queries

Use parameters to safely pass values to queries:

```
# Dictionary parameters (recommended)
result = sqlutils.sql(
    "SELECT * FROM orders WHERE customer_id = :customer_id AND status = :status",
    parameters={"customer_id": 12345, "status": "completed"},
    connection_name="redshift_connection"
)

# Athena with named parameters
result = sqlutils.sql(
    "SELECT * FROM products WHERE category = :category AND price > :min_price",
    parameters={"category": "Electronics", "min_price": 100},
    connection_name="athena_connection"
)
```

Getting Database Engine

You can also get the underlying SQLAlchemy engine for advanced operations:

```
# Get engine for a connection
engine = sqlutils.get_engine(connection_name="redshift_connection")

# Use engine directly with pandas
import pandas as pd

df = pd.read_sql("SELECT * FROM large_table LIMIT 1000", engine)
```

DuckDB Features

When using DuckDB (no connection specified), you get additional capabilities:

Python Integration

```
# DuckDB can access Python variables directly
```

```
import pandas as pd

my_df = pd.DataFrame({'id': [1, 2, 3], 'name': ['A', 'B', 'C']})

result = sqlutils.sql("SELECT * FROM my_df WHERE id > 1")
```

Notes

- All queries return pandas DataFrames for easy data manipulation
- DuckDB is automatically configured with Amazon S3 credentials from the environment
- Connection credentials are managed through Amazon SageMaker Unified Studio project connections
- The module handles connection pooling and cleanup automatically

DataFrame Utilities

Read from and write to catalog tables using pandas DataFrames with automatic format detection and database management.

Supported catalog types:

- AwsDataCatalog
- S3CatalogTables

Basic Usage

Import the DataFrame utilities:

```
from sagemaker_studio import dataframeutils
```

Reading from Catalog Tables

Required Inputs:

- database (str): Database name within the catalog
- table (str): Table name

Optional Parameters:

- catalog (str): Catalog identifier (defaults to AwsDataCatalog if not specified)
- format (str): Data format - auto-detects from table metadata, falls back to parquet
- **kwargs: Additional arguments
  - for AwsDataCatalog, kwargs can be columns, chunked, etc
  - for S3Tables, kwargs can be limit, row_filter, selected_fields, etc

```
import pandas as pd

# Read from AwsDataCatalog
df = pd.read_catalog_table(
    database="my_database",
    table="my_table"
)

# Read from S3 Tables
df = pd.read_catalog_table(
   database="my_database",
   table="my_table",
   catalog="s3tablescatalog/my_s3_tables_catalog",
)
```

Usage with optional parameters

```
import pandas as pd

# Read from AwsDataCatalog by explicitly specifying catalogID and format
df = pd.read_catalog_table(
    database="my_database",
    table="my_table",
    catalog="123456789012",
    format="parquet"
)

# Read from AwsDataCatalog by explicitly specifying catalogID, format, and additional
 args -> columns
```

```
df = pd.read_catalog_table(
    database="my_database",
    table="my_table",
    catalog="123456789012",
    format="parquet",
    columns=['<column_name_1>, <column_name_2>']
)

# Read from S3 Tables with additional args -> limit
df = pd.read_catalog_table(
   database="my_database",
   table="my_table",
   catalog="s3tablescatalog/my_s3_tables_catalog",
   limit=500
)

# Read from S3 Tables with additional args -> selected_fields
df = pd.read_catalog_table(
   database="my_database",
   table="my_table",
   catalog="s3tablescatalog/my_s3_tables_catalog",
   selected_fields=['<field_name_1>, <field_name_2>']
)
```

Writing to Catalog Tables

Required Inputs:

- database (str): Database name within the catalog
- table (str): Table name

Optional Parameters:

- catalog (str): Catalog identifier (defaults to AwsDataCatalog if not specified)
- format (str): Data format used for AwsDataCatalog (default: parquet)
- path (str): Custom Amazon S3 path for writing to AwsDataCatalog (auto-determined if not provided)
- **kwargs: Additional arguments

Path Resolution Priority - Amazon S3 path is determined in this order:

- User-provided path parameter
- Existing database location + table name
- Existing table location
- Project default Amazon S3 location

```
import pandas as pd

# Create sample DataFrame
df = pd.DataFrame({
    'id': [1, 2, 3],
    'name': ['Alice', 'Bob', 'Charlie'],
    'value': [10.5, 20.3, 15.7]
})

# Write to AwsDataCatalog
df.to_catalog_table(
    database="my_database",
    table="my_table"
)

# Write to S3 Table Catalog
df.to_catalog_table(
    database="my_database",
    table="my_table",
    catalog="s3tablescatalog/my_s3_tables_catalog"
)
```

Writing to Catalog Tables

```
# Write to AwsDataCatalog with csv format
df.to_catalog_table(
    database="my_database",
    table="my_table",
    format="csv"
)
```

```
# Write to AwsDataCatalog at user specified s3 path
df.to_catalog_table(
    database="my_database",
    table="my_table",
    path="s3://my-bucket/custom/path/"
)

# Write to AwsDataCatalog with additional argument -> compression
df.to_catalog_table(
    database="my_database",
    table="my_table",
    compression='gzip'
)
```

Spark Utilities

The Spark utilities module provides a simple interface for working with Spark Connect sessions and managing Spark configurations for various data sources within Amazon SageMaker Unified Studio. When no connection is specified, a Spark Connect session is created using the default Amazon Athena Spark connection.

Basic Usage

Import the Spark utilities:

```
from sagemaker_studio import sparkutils
```

Initialize Spark Session

Supported connection types:

- Spark connect

Optional Parameters:

- connection_name (str): Name of the connection to execute query against (e.g., "my_redshift_connection")

When no connection is specified, a default Amazon Athena Spark session is created. Other options include EMR-Serverless (version 7.13 and onwards). EMR-Serverless connection can be created and used:

```
# Default session
spark = sparkutils.init()

# Session with specific connection
spark = sparkutils.init(connection_name="my_spark_connection")
```

Working with Spark Options

Supported connection types:

- Amazon DocumentDB
- Amazon DynamoDB
- Amazon Redshift
- Aurora MySQL
- Aurora PostgreSQL
- Azure SQL
- Google BigQuery
- Microsoft SQL Server
- MySQL
- PostgreSQL
- Oracle
- Snowflake

Required Inputs:

- connection_name (str): Name of the connection to get Spark options for (e.g., "my_redshift_connection")

Get formatted Spark options for connecting to data sources:

```
# Get options for Redshift connection
options = sparkutils.get_spark_options("my_redshift_connection")
```

Examples by Operation Type

Reading and Writing Data

```
# Create sample DataFrame
df_to_write = spark.createDataFrame(
    [(1, "Alice"), (2, "Bob")],
    ["id", "name"]
)

# Get spark options for Redshift connection
spark_options = sparkutils.get_spark_options("my_redshift_connection")

# Write DataFrame using JDBC
df_to_write.write \
    .format("jdbc") \
    .options(**spark_options) \
    .option("dbtable", "sample_table") \
    .save()

# Read DataFrame using JDBC
df_to_read = spark.read \
    .format('jdbc') \
    .options(**spark_options) \
    .option('dbtable', 'sample_table') \
    .load()

# Display results
df_to_read.show()
```

Notes

- Spark sessions are automatically configured for Amazon Athena spark compute
- Connection credentials are managed through Amazon SageMaker Unified Studio project connections

- The module handles session management and cleanup automatically
- Spark options are formatted appropriately for each supported data source
- When get_spark_options is used in EMR-S or EMR-on-EC2 compute, and the connection has EnforceSSL enabled, the formatted spark options will not have the sslrootcert value and hence that would need to be passed explicitly.

Execution APIs

Execution APIs provide you the ability to start an execution to run a notebook headlessly within the same user space or on remote compute.

Local Execution APIs

Use the following APIs to start, stop, get, or list executions within the user's space.

StartExecution

You can start a notebook execution headlessly within the same user space.

```
from sagemaker_studio.sagemaker_studio_api import SageMakerStudioAPI
from sagemaker_studio import ClientConfig

config = ClientConfig(overrides={
            "execution": {
                "local": True,
            }
        })
sagemaker_studio_api = SageMakerStudioAPI(config)

result = sagemaker_studio_api.execution_client.start_execution(
    execution_name="my-execution",
    input_config={"notebook_config": {
        "input_path": "src/folder2/test.ipynb"}},
    execution_type="NOTEBOOK",
    output_config={"notebook_config": {
        "output_formats": ["NOTEBOOK", "HTML"]
    }}
)
print(result)
```

GetExecution

You can retrieve details about a local execution using the `GetExecution` API.

```
from sagemaker_studio.sagemaker_studio_api import SageMakerStudioAPI
from sagemaker_studio import ClientConfig

config = ClientConfig(region="us-west-2", overrides={
            "execution": {
                "local": True,
            }
        })
sagemaker_studio_api = SageMakerStudioAPI(config)

get_response =
 sagemaker_studio_api.execution_client.get_execution(execution_id="asdf-3b998be2-02dd-42af-8802
print(get_response)
```

ListExecutions

You can use the `ListExecutions` API to list all the executions that ran in the user's space.

```
from sagemaker_studio.sagemaker_studio_api import SageMakerStudioAPI
from sagemaker_studio import ClientConfig

config = ClientConfig(region="us-west-2", overrides={
            "execution": {
                "local": True,
            }
        })
sagemaker_studio_api = SageMakerStudioAPI(config)

list_executions_response =
 sagemaker_studio_api.execution_client.list_executions(status="COMPLETED")
print(list_executions_response)
```

StopExecution

You can use the `StopExecution` API to stop an execution that's running in the user space.

```
from sagemaker_studio.sagemaker_studio_api import SageMakerStudioAPI
from sagemaker_studio import ClientConfig

config = ClientConfig(region="us-west-2", overrides={
            "execution": {
                "local": True,
            }
        })
sagemaker_studio_api = SageMakerStudioAPI(config)

stop_response =
 sagemaker_studio_api.execution_client.stop_execution(execution_id="asdf-3b998be2-02dd-42af-880
print(stop_response)
```

Remote Execution APIs

Use the following APIs to start, stop, get, or list executions running on remote compute.

StartExecution

You can start a notebook execution headlessly on a remote compute specified in the `StartExecution` request.

```
from sagemaker_studio.sagemaker_studio_api import SageMakerStudioAPI
from sagemaker_studio import ClientConfig

config = ClientConfig(region="us-west-2")
sagemaker_studio_api = SageMakerStudioAPI(config)

result = sagemaker_studio_api.execution_client.start_execution(
    execution_name="my-execution",
    execution_type="NOTEBOOK",
    input_config={"notebook_config": {"input_path": "src/folder2/test.ipynb"}},
    output_config={"notebook_config": {"output_formats": ["NOTEBOOK", "HTML"]}},
    termination_condition={"max_runtime_in_seconds": 9000},
    compute={
        "instance_type": "ml.c5.xlarge",
        "image_details": {
            # provide either ecr_uri or (image_name and image_version)
```

```
            "image_name": "sagemaker-distribution-embargoed-loadtest",
            "image_version": "2.2",
            "ecr_uri": "123456123456.dkr.ecr.us-west-2.amazonaws.com/ImageName:latest",
        }
    }
)
print(result)
```

GetExecution

You can retrieve details about an execution running on remote compute using the `GetExecution` API.

```
from sagemaker_studio.sagemaker_studio_api import SageMakerStudioAPI
from sagemaker_studio import ClientConfig

config = ClientConfig(region="us-west-2")
sagemaker_studio_api = SageMakerStudioAPI(config)

get_response =
 sagemaker_studio_api.execution_client.get_execution(execution_id="asdf-3b998be2-02dd-42af-8802
print(get_response)
```

ListExecutions

You can use the `ListExecutions` API to list all the headless executions that ran on remote compute.

```
from sagemaker_studio.sagemaker_studio_api import SageMakerStudioAPI
from sagemaker_studio import ClientConfig

config = ClientConfig(region="us-west-2")
sagemaker_studio_api = SageMakerStudioAPI(config)

list_executions_response =
 sagemaker_studio_api.execution_client.list_executions(status="COMPLETED")
print(list_executions_response)
```

StopExecution

You can use the `StopExecution` API to stop an execution that's running on remote compute.

```
from sagemaker_studio.sagemaker_studio_api import SageMakerStudioAPI
from sagemaker_studio import ClientConfig

config = ClientConfig(region="us-west-2")
sagemaker_studio_api = SageMakerStudioAPI(config)

stop_response =
 sagemaker_studio_api.execution_client.stop_execution(execution_id="asdf-3b998be2-02dd-42af-880
print(stop_response)
```

Using the JupyterLab IDE in Amazon SageMaker Unified Studio

The JupyterLab page of Amazon SageMaker Unified Studio provides a JupyterLab interactive development environment (IDE) for you to use as you perform data integration, analytics, or machine learning in your projects. Amazon SageMaker Unified Studio notebooks are powered by JupyterLab spaces.

By default, the JupyterLab application comes with the Amazon SageMaker Distribution image. The distribution image includes popular packages such as the following:

- PyTorch
- TensorFlow
- Keras
- NumPy
- Pandas
- Scikit-learn

Amazon SageMaker Unified Studio includes a sample notebook that you can use to get started. You can also choose to create new notebooks for your business use cases.

Amazon SageMaker Unified Studio notebooks include the following key features:

- Manage configurations to scale the instance vertically if the job being submitted demands it.

- Access metadata to find out information such as the path to the Amazon S3 bucket where data is being stored.
- Perform Git operations for version control.
- Use Amazon Q chat functionality to ask questions and generate code using prompts.
- Perform code completion using Amazon Q Developer.

> **Note**
>
> The JupyterLab IDE has an idle shutdown feature that shuts down the IDE after it has been idle for 60 minutes. This means that if both the IDE kernel and terminal have been unused for an hour, the IDE stops running. In order to start using the IDE again after idle shutdown, you would need to navigate to the JupyterLab page again and click on the Start button to restart the kernel in the JupyterLab IDE.

Managing configurations

You can edit your JupyterLab configurations on the JupyterLab page by choosing Configure in the top right corner. A popup appears where you can change the instance type. You can also increase the EBS volume up to 16 GB if allowed by your admin.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, under **IDEs**, choose **JupyterLab**.
3. Choose the Configure button in the top right corner of the page. A popup appears where you can change the instance type and increase the EBS volume.
4. Specify the instance type and EBS volume that you want for testing.
    - NOTE: After you increase the EBS volume, you cannot decrease it.

Configuring Spark compute

Amazon SageMaker Unified Studio provides a set of Jupyter magic commands. Magic commands, or magics, enhance the functionality of the IPython environment. For more information about the magics that Amazon SageMaker Unified Studio provides, run `%help` in a notebook.

Compute-specific configurations can be set by using the `%%configure` Jupyter magic. The `%%configure` magic takes a JSON-formatted dictionary. To use %%configure magic, specify the compute name in the argument `-n`. Including `-f` will restart the session to forcefully apply the new configuration. Otherwise, this configuration will apply when the next session starts.

For example: `%%configure -n compute_name -f`.

Library management

You can use the library management widget in JupyterLab to manage the library installations and configurations in your notebook.

To navigate to the library management of a notebook in Amazon SageMaker Unified Studio, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to a project. You can do this by choosing a project from the project selector dropdown at the top of the page, or by creating a new project.
3. In the left navigation pane, under **IDEs**, choose **JupyterLab**.
4. Navigate to a notebook or create a new one by selecting **File** > **New** > **Notebook**.
5. Choose the library management icon from the notebook navigation bar.



The following library configurations are available:

Jar

- **Maven artifacts**
- **S3 paths**
- **Disk location paths**

- **Other paths**

Python

- **Conda packages**
- **PyPI packages**
- **S3 paths**
- **Disk location paths**
- **Other paths**

Adding JupyterLab library configurations

1. Navigate to the JupyterLab library management page.
2. Select the configuration method you would like to add from the left navigation of the library management page.
3. Choose **Add**.
4. Input the URL, package name, coordinates, or other information as the fields indicate.
5. In the left navigation of the library management page, check the box **Apply the change to JupyterLab**.
6. Choose **Save all changes**.

Compute-specific configuration

Amazon SageMaker Unified Studio provides a set of Jupyter magic commands. Magic commands, or magics, enhance the functionality of the IPython environment. For more information about the magics that Amazon SageMaker Unified Studio provides, run %help in a notebook.

Compute-specific configurations can be configured by %%configure Jupyter magic. The %%configure magic takes a json-formatted dictionary. To use %%configure magic, please specify the

compute name in the argument -n. Include —f will restart the session to forcefully apply the new configuration, otherwise this configuration will apply only when next session starts.

Configure an EMR Spark session

When working with EMR on EC2 or EMR Serverless, %%configure command can be used to configure the Spark session creation parameters. Using conf settings, you can configure any Spark configuration that's mentioned in the configuration documentation for Apache Spark.

```
%%configure -n compute_name -f
{
    "conf": {
        "spark.sql.shuffle.partitions": "36"
     }
}
```

Configure a Glue interactive session

Use the `--` prefix for run arguments specified for Glue.

```
%%configure -n project.spark.compatibility -f
{
   "--enable-auto-scaling": "true"
   "--enable-glue-datacatalog": "false"
}
```

For more information on job parameters, see Job parameters.

You can update Spark configuration via %%configure when working with Glue with --conf in configure magic. You can configure any Spark configuration that's mentioned in the configuration documentation for Apache Spark.

```
%%configure -n project.spark.compatibility -f
{
    "--conf": "spark.sql.shuffle.partitions=36"
}
```

Accessing metadata

You can view metadata for your project in the notebook terminal within Amazon SageMaker Unified Studio. This shows you information such as the `ProjectS3Path`, which is the Amazon S3 bucket where your project data is stored. The project metadata is written to a file named resource-metadata.json in the folder /opt/ml/metadata/. You can get the metadata by opening a terminal from within the notebook.

1. Navigate to the Code page within the project you want to view metadata for.
2. Choose File > New > Terminal.
3. Enter in the following command:

   ```
   cat /opt/ml/metadata/resource-metadata.json
   ```

   The metadata file information then appears in the terminal window.

Performing Git operations

The JupyterLab IDE in Amazon SageMaker Unified Studio is configured with Git and initialized with the project repository when a project is created.

To access Git operations in the Amazon SageMaker Unified Studio management console, navigate to the Code page of your project, then choose the Git button in the JupyterLab IDE left panel as shown in the image below.

This opens a panel where you can view commit history and perform Git operations. You can use this Git extension to commit and push files back to the project repository, switch your working branch or create a new one, and manage tags.

To fetch notebooks committed by other users, do a pull from the project repository.

> **Note**
>
> When you create and enable a connection for Git access and the user accesses this connection in the JupyterLab IDE in Amazon SageMaker Unified Studio, the repository is

cloned. In other words, a local copy of the repository is created in the Amazon SageMaker Unified Studio project. If the administrator later disables or deletes this Git connection, the local repository remains in the user's IDE, but users can no longer push or pull files to or from it. For more information, see Git connections in Amazon SageMaker Unified Studio.

Using the Amazon Q data integration in AWS Glue

Amazon SageMaker Unified Studio supports the Amazon Q data integration in AWS Glue. It helps data engineers and ETL developers create data integration jobs using natural language letting you automate aspects of code authoring.

When using the Amazon Q data integration in AWS Glue, in the Jupyter Lab IDE, you enter comments using natural language instructions, and then the PySpark kernel generates the code on your behalf. You can customize the generated code to meet your own needs.

1. Open a Python notebook, and ensure the kernel is configured to use a PySpark connection.
2. You can request a prompt response by adding a comment and then placing it in a prompt, which will start Amazon Q processing.
3. If the prompt is AWS Glue related, the data integration generates a AWS Glue job script using PySpark.
4. Alternatively, you can continue to use your default auto-completions from Amazon Q Developer. If a prompt isn't Glue related, Amazon Q Developer will use autocomplete instead.

Running SQL and Spark code

You can run code against multiple compute in one Jupyter notebook using different programming languages through the use of Jupyter cell magics %%pyspark, %%sql, %%scalaspark.

For example, to run pyspark code on Spark compute, you can run the following code:

```
%%pyspark compute_name
spark.createDataFrame([('Alice', 1)])
```

The following table represents the supported compute types of each magic:

magic	supported compute types
%%sql	Redshift, Athena, EMR on EC2, EMR Serverless, Glue Interactive Session
%%pyspark	EMR on EC2, EMR Serverless, Glue Interactive Session
%%scalaspark	EMR on EC2, EMR Serverless, Glue Interactive Session

The dropdown available at the top of active cells allows you to select the Connection and Compute type. If no selection is made the code in the cell will be run against the Compute hosting JupyterLab (“Local Python” / “project.python”). The option selected for Connection type dictates the Compute available. The selections dictate the magics code generated in the cell and determine where your code runs.

When a new cell is created, it will select the same connection and compute type as the previous cell automatically. To configure the dropdown, go to Settings → Settings editor → Connection magics settings.

Visualizing results

%display is a magic that customers can apply against any DataFrame to invoke a visualization for tabular data. use the visualization to scroll through a DataFrame or results of a Redshift or Athena query.

There are four different views:

- **Table**. You can change the sampling method, sample size, and rows per page that are displayed.
- **Summary**. Each column in the summary tab has a button labeled with the column’s name. Clicking on one of these buttons opens the a sub-tab in the column view in Tab 3 for the column that was clicked.
- **Column**. For each column selected in the column selector above, a sub-tab appears with more details about the contents of the column.
- **Plotting**. In the default plotting view you can change the graph type, axes, value types, and aggregation functions for plotting. By installing an optional supported third-party plotting

library on the Jupyterlab space (pygwalker, ydata-profiling, or dataprep) and running the display magic you can visualize your data using the installed library.

Shared Project Storage

The JupyterLab visualization widget offers an option to store visualization data in a shared Amazon S3 location within your project bucket. The data is stored using the following structure:

```
s3://bucket/domain/project/dev/user/{sts_user_identity}/query-result/{data_uuid}/

### dataframe/           # Contains DataFrame in parquet format
### head/100/            # Sample data (100 rows)
#   ### metadata.json
#   ### summary_schema.json
#   ### column_schema/
### tail/                # Additional sample data
```

Storage Options

The visualization widget supports two storage modes controlled by the `--query-storage` parameter:

- **Cell storage** (`--query-storage cell`): Data stored locally in notebook output (current default behavior)
- **S3 storage** (`--query-storage s3`): Data stored in project's shared S3 bucket for persistence and sharing
  - Choose **Store query result in S3** to store the data in project's shared S3 bucket.

Data Access and Security

When using Amazon S3 storage, the visualization data is accessible to all project members. Data persists beyond individual JupyterLab sessions. No individual user permissions can be set on stored visualizations. You should consider data classification before storing sensitive information. The storage uses the project's default runtime role for access control.

> **Note**
>
> The Amazon S3 storage location is shared across the entire project. All project members can access visualization data stored by any team member.

Data Sharing Across Compute Environments

Amazon SageMaker Unified Studio provides magic commands to facilitate data sharing across different compute environments. This section outlines three key commands: `%push`, `%pop`, and `%send_to_remote`.

%push

The `%push` command allows you to upload specified variables to your project's shared S3 storage within Amazon SageMaker Unified Studio.

```
%push <var_name>
%push <var_name1>,<var_name2>
%push -v <var_name>
%push -v <var_name> --namespace <namespace_name>
```

Key Features:

- Supports multiple variable uploads when comma-separated
- -v specifies the variable name (alternative syntax)
- Optional --namespace argument (defaults to kernel ID)
- Uploaded variables are accessible to all project members

Supported Connections:

- Local Python connections
- AWS Glue connections
- AWS EMR connections

Supported Language: Python

%pop

The %pop command enables you to download specified variables from shared project Amazon S3 storage to your current compute environment.

```
%pop <var_name>
%pop <var_name1>,<var_name2>
%pop -v <var_name>
%pop -v <var_name> --namespace <namespace_name>
```

Key Features:

- Supports multiple variable downloads when comma-separated
- -v specifies the variable name (alternative syntax)
- Optional --namespace argument (defaults to kernel ID)

Supported Connections:

- Local Python connections
- AWS Glue connections
- AWS EMR connections

Supported Language: Python

%send_to_remote

The %send_to_remote command allows you to send a variable from the local kernel to a remote compute environment.

```
%send_to_remote --name <connection_name> --language <language> --local
 <local_variable_name> --remote <remote_variable_name>
```

Key Features:

- Supports both Python and Scala in remote environments
- Python remote supports dict, df, and str data types
- Scala remote supports df and str data types

Arguments:

- -l or --language: Specifies the connection language
- -n or --name: Specifies the connection to be used
- --local: Defines the local variable name
- --remote or -r: Defines the remote variable name

Supported Connections: local Python connections

Supported Language:

- Python
- Scala

Security considerations

Remember that variables uploaded using `%push` are accessible to all project members within your Amazon SageMaker Unified Studio project. Ensure that sensitive data is handled appropriately and in compliance with your organization's data governance policies.

Using the Code Editor IDE in Amazon SageMaker Unified Studio

Code Editor, based on Code-OSS (Open Source Software) like VS Code, helps you write, test, debug, and run your analytics and machine learning code. It also supports integrated development environment (IDE) extensions available in the Open VSX Registry.

After creating a Code Editor space in Amazon SageMaker Unified Studio, you can access your Code Editor session directly through the browser. Within your Code Editor environment, you can do the following:

- Access artifacts from Amazon SageMaker Unified Studio

- Clone your GitHub repositories and commit changes
- Access the SageMaker Python SDK

You can return to Amazon SageMaker Unified Studio to review any assets created in your Code Editor environment such as experiments, pipelines, or training jobs.

Performing Git operations in Code Editor

The Code Editor IDE in Amazon SageMaker Unified Studio is configured with Git and initialized with the project repository when a Code Editor space is created.

You can use Git operations after launching your Code Editor space. To launch your project, choose **Open** next to the space you want to open in the **Spaces** tab of your project.

When Code Editor is open, choose the **Source Control** icon in the left navigation. You can use this window to perform Git operations such as committing and pushing files back to the project repository, switching your working branch or creating a new one, and managing tags.

To fetch notebooks committed by other users, do a pull from the project repository.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to your project.
3. In the **Build** menu, choose **Spaces**.
4. Choose **Open** next to the Code Editor space you want to open.
5. In the Source Control window of your Code Editor space, choose the three-dot menu. Then choose **Pull**.

> **Note**
>
> When you create and enable a connection for Git access and the user accesses this connection in the Code Editor IDE in Amazon SageMaker Unified Studio, the repository is cloned. In other words, a local copy of the repository is created in the Amazon SageMaker Unified Studio project. If the administrator later disables or deletes this Git connection, the local repository remains in the user's IDE, but users can no longer push or pull files to or from it. For more information, see Git connections in Amazon SageMaker Unified Studio.

Checking the version of Code Editor in Amazon SageMaker Unified Studio

The following steps show how to check the version of your Code Editor application.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to your project.
3. In the **Build** menu, choose **Spaces**.
4. Choose **Open** next to the Code Editor space you want to open.
5. In the upper-left corner of the Code Editor UI, choose the three-bar menu button.
6. Choose **Help**. Then choose **About**.

Connections and extensions in Code Editor

Code Editor supports IDE connections to AWS services as well as extensions available in the Open VSX Registry.

Code Editor environments are integrated with the AWS Toolkit for VS Code to add connections to AWS services. Within your Code Editor environment, you can add connections to AWS Explorer to view, modify, and deploy AWS resources in Amazon S3, CloudWatch, and more.

Code Editor supports IDE extensions available in the Open VSX Registry. To get started with extensions in your Code Editor environment, choose the Extensions icon in the left navigation pane. Here, you can configure connections to AWS by installing the AWS Toolkit. For more information, see Installing the AWS Toolkit for Visual Studio Code.

In the search bar, you can search directly for additional extensions through the Open VSX Registry, such as the AWS Toolkit, Jupyter, Python, and more.

Using Strands Agents in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio now supports Strands Agents integration, streamlining AI agent development for users. This integration enables you to immediately begin building AI agents using Strands SDK's powerful capabilities without the complexity of dependency management or environment configuration.

By making Strands SDK a native component of the Amazon SageMaker ecosystem, developers can focus on creating innovative AI solutions while leveraging Amazon SageMaker's enterprise-grade infrastructure and scaling capabilities.

The Amazon BedrockModel provider is used by default when creating a basic Agent with the provided Amazon Bedrock model. You can specify which Amazon Bedrock model to use by passing in the model_id string directly to the Agent constructor. You must use the application inference profile ARN as the model_id, like in the following example:

```
import boto3
from strands import Agent, tool
from strands.models import BedrockModel
from strands_tools import calculator, current_time

# Create a Bedrock model with the custom session
bedrock_model = BedrockModel(
model_id="arn:aws:bedrock:us-west-2:123456789321:application-inference-profile/
ab788q84ey7z"
)

# Create an agent with the model and tools
agent = Agent(
model=bedrock_model,
tools=[calculator, current_time]
)
# First request will cache the tools
response1 = agent("What time is it?")

# Second request will reuse the cached tools
response2 = agent("What is the square root of 1764?")
```

You can obtain the application inference profile ARN by performing the following procedure:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to your project.
3. In the left navigation pane, under **Generative AI**, choose **AI apps**.
4. In the left-hand navigation pane, choose **Models** in order to see all the enabled Amazon Bedrock models for the project.

5. Choose a model and then on the model details page, locate the application inference profile ARN.

Amazon SageMaker Unified Studio Local IDE Support

Overview

Amazon SageMaker Unified Studio now supports local IDE - Visual Studio Code (VS Code) and Kiro integrations, enabling you to use your preferred development environment. This integration allows you to maintain your familiar local development setup including custom extensions, debugging tools, and AI-assisted development features while accessing the Amazon SageMaker Unified Studio compute resources and data.

While Amazon SageMaker Unified Studio's web-based IDEs provide secure and efficient cloud development environments, many developers prefer local IDEs for several key advantages like extensive customization options through many available extensions, user customizations such as code snippet shortcuts, keyboard shortcuts, and theme customization.

Prerequisites

> **Important**
>
> When logging into Amazon SageMaker Unified Studio with an IAM role from your local IDE, please ensure SageMakerStudioUserIAMConsolePolicy or equivalent permissions are attached to your role.

Before connecting VS Code, Kiro, or Cursor remotely to a SageMaker Unified Studio Space, ensure that you have the following prerequisites:

- Access to an existing domain in Amazon SageMaker Unified Studio.
- A supported IDE and AWS Toolkit version, as listed in the following table.

IDE	Minimum version	Minimum AWS Toolkit version
VS Code	1.90 or later	3.97 or later
Kiro	0.8.0 or later	3.97 or later

IDE	Minimum version	Minimum AWS Toolkit version
Cursor	2.6.18 or later	3.100 or later

- The AWS Toolkit extension installed from the Extensions marketplace in your IDE.
- One of the following supported operating systems:
  - macOS 13 or later
  - Windows 10 or Windows 11
  - Linux (requires the official Microsoft VS Code distribution for VS Code and Cursor)
- A Space instance that meets the following requirements:
  - Memory: Minimum 8 GB RAM. We recommend instances with 8 GB or more memory for optimal performance.
  - Unsupported instance types: `ml.t3.medium`, `ml.c7i.large`, `ml.c6i.large`, `ml.c6id.large`, `ml.c5.large`
- A compatible image:
  - SageMaker Distribution: Version 2.8 or later
  - Custom images: Must follow SageMaker custom image specifications
- Extension compatibility: Not all VS Code extensions support remote development.

> **Important**
>
> Remote Space connections are not supported for TIP (Trusted Identity Propagation) enabled projects. For instructions on how to set the `enableTrustedIdentityPropagationPermissions` parameter to false to use remote connection for Spaces, see Trusted identity propagation.

VS Code Setting in Amazon SageMaker Unified Studio

Follow the steps described in these sections to integrate Amazon SageMaker Unified Studio in VS code:

Topics

- Authenticating directly in Local IDE

- Link to VS Code from Amazon SageMaker Unified Studio

Authenticating directly in Local IDE

In Identity Center based domains

1. Open the AWS Toolkit panel in VS Code and navigate to the Amazon SageMaker Unified Studio section.
2. Choose Sign in to get started. You will see 2 options to sign in with IAM Credentials and IAM Identity Center. Choose IAM Identity Center and enter your domain URL (format: `https://domain-id.sagemaker.region.on.aws`).
3. You will see a prompt asking you to open a link in the browser for IAM Identity Center authentication, choose Open on the Do you want Code to open the external website? prompt. You will be directed to the browser AWS IAM Identity Center authentication. Complete the authentication flow in your browser, choose Allow access on the next prompt, and return to VS Code.
4. You should now be prompted to select a project. If not, under the domain, choose Select a project to choose from the list of available projects in your domain.
5. Once connected, you'll see a hierarchical view of your resources. Under the selected Project, you have Data and Compute resources associated with the project.

In IAM-based domains

1. Open the AWS Toolkit panel in VS Code and navigate to the Amazon SageMaker Unified Studio section.
2. Choose Sign in to get started. You will see 2 options to sign in with IAM Credentials and IAM Identity Center. Choose IAM Credentials.
3. You can either choose an existing credential profile and edit it or create a new one.
   - To edit an existing credential profile, follow the process in AWS IAM credentials.
   - To create a new credential profile, follow the prompts and provide the credential of the Login Role you use to access your project in Amazon SageMaker Unified Studio IAM-based domain.
     - Select the region where your IAM-based domain is in. After the profile is created, sign in by selecting the created profile.

4. You will then see your Amazon SageMaker Unified Studio resources including Data and Compute resources listed in a sub-panel within AWS Toolkit.

The resources under Data show all your project's data sources: S3 buckets, Redshift, and Lakehouse in one place, with easy browsing through databases, tables, and schema. You can access the data schema directly without switching between different tools or screens, making it easy for code authoring in VS Code.

The Compute resource has Data warehouse, Data processing, and Spaces within each project. Data Warehouse and Data Processing list the compute connections in the project and hovering over them displays the connection metadata. Spaces are remote IDE environments you have access to in the project that you can connect to remotely to author code from your local VS Code IDE.

Link to VS Code from Amazon SageMaker Unified Studio

You can use Open in VS Code link provided in Amazon SageMaker Unified Studio to access your SageMaker Space in VS Code. When you choose this option, Amazon SageMaker Unified Studio will open VS Code on your system and open your SageMaker space in VS Code.

> **Note**
>
> You must be using AWS Toolkit version 3.87 or higher for Open in VS Code to function.

In Identity Center based domains:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO credentials.
2. In the left navigation pane, under **IDEs**, choose **JupyterLab**.
3. Once your JupyterLab Notebook opens, choose Open in VS Code in the top right corner of the page.

In IAM-based domains:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your IAM credentials.
2. In the left navigating, choose JupyterLab

3. Once your JupyterLab Notebook opens, choose Open in VS Code in the top right corner of the page. This will establish a remote connection and open your SageMaker space in VS Code.

> **Note**
>
> The sessions initiated through Open in VS Code link do not automatically reconnect if the remote session is interrupted. You will need create a new remote connection using the Open in VS Code link in Amazon SageMaker Unified Studio. The sessions established through this link can last for 12 hours if not interrupted.

Kiro in Amazon SageMaker Unified Studio

Kiro is Amazon's AI-powered agentic IDE that you can use as a local IDE to connect to your Amazon SageMaker Unified Studio Spaces and resources.

These instructions assume you already have an Amazon SageMaker Unified Studio domain, project, and Space set up.

To set up Kiro

1. Download and install Kiro from kiro.dev (minimum version 0.8.0 or later).
2. Open the AWS Toolkit panel in Kiro and navigate to the Amazon SageMaker Unified Studio section.
3. Choose **Sign in to get started**, and then choose **IAM Credentials**. You can also use IAM Identity Center.
4. Create or choose a credential profile using the same IAM role you use to access your project in the AWS Management Console. Choose the Region your project is in, and then sign in. Your Amazon SageMaker Unified Studio resources (Data, Compute, Spaces) appear in the panel.
5. Under your project, choose **Compute**, and then choose **Spaces**. Hover over the Space you want to connect to and choose the **Connect** button.
6. When prompted to install the Amazon SageMaker SSH Plugin for Kiro, choose **Install**. This enables remote access, starts the Space, and connects to it.
7. Sign in to Kiro on the Space when prompted. After you connect, all project files are available in your remote workspace.

> **Note**
>
> Instances must have a minimum of 8 GB RAM. The following instance types are not supported: `ml.t3.medium`, `ml.c7i.large`, `ml.c6i.large`, `ml.c6id.large`, `ml.c5.large`.

You can provide your AI assistant with more context about Amazon SageMaker Unified Studio workspaces by linking a service-provided context file, smus-context.md, to the your existing AGENTS.md file or by creating a new one if it doesn't already exist. It is strongly recommended to use the Amazon SageMaker Unified Studio context file for a better user experience. As this is a service-provided context file and subject to change, you can edit other context files linked to AGENTS.md or to AGENTS.md directly.

Link to Kiro from Amazon SageMaker Unified Studio

You can use Open in Kiro link provided in Amazon SageMaker Unified Studio to access your SageMaker Space in Kiro. When you choose this option, Amazon SageMaker Unified Studio will open Kiro on your system and open your SageMaker space in Kiro.

> **Note**
>
> You must be using AWS Toolkit version 3.97 or higher for Open in Kiro to function.

In Identity Center based domains:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO credentials.
2. In the left navigation pane, under **IDEs**, choose **JupyterLab**.
3. Once your JupyterLab Notebook opens, choose Open in Kiro in the top right corner of the page.

In IAM-based domains:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your IAM credentials.
2. In the left navigating, choose JupyterLab

3. Once your JupyterLab Notebook opens, choose Open in Kiro in the top right corner of the page. This will establish a remote connection and open your SageMaker space in Kiro.

> **Note**
>
> The sessions initiated through Open in Kiro link do not automatically reconnect if the remote session is interrupted. You will need create a new remote connection using the Open in Kiro link in Amazon SageMaker Unified Studio. The sessions established through this link can last for 12 hours if not interrupted.

Cursor in Amazon SageMaker Unified Studio

Cursor is a VS Code-based AI-powered IDE that you can use as a local IDE to connect to your Amazon SageMaker Unified Studio Spaces and resources.

> **Important**
>
> Cursor is currently verified only for Spaces with public internet access.

These instructions assume you already have an Amazon SageMaker Unified Studio domain, project, and Space set up.

To set up Cursor

1. Download and install Cursor from cursor.com/download (minimum version v2.6.18 or later).
2. Open the AWS Toolkit panel in Cursor and navigate to the Amazon SageMaker Unified Studio section.
3. Choose **Sign in to get started**, and then choose **IAM Credentials**. You can also use IAM Identity Center.
4. Create or choose a credential profile using the same IAM role you use to access your project in the AWS Management Console. Choose the Region your project is in, and then sign in. Your Amazon SageMaker Unified Studio resources (Data, Compute, Spaces) appear in the panel.
5. Under your project, choose **Compute**, and then choose **Spaces**. Hover over the Space you want to connect to and choose the **Connect** button.

6. When prompted to install the Remote SSH Plugin for Cursor, choose **Install**. This enables remote access, starts the Space, and connects to it.
7. Sign in to Cursor on the Space when prompted. After you connect, all project files are available in your remote workspace.

> **Note**
>
> Instances must have a minimum of 8 GB RAM. The following instance types are not supported: `ml.t3.medium`, `ml.c7i.large`, `ml.c6i.large`, `ml.c6id.large`, `ml.c5.large`.

Link to Cursor from Amazon SageMaker Unified Studio

You can use the Open in Cursor link provided in Amazon SageMaker Unified Studio to access your SageMaker Space in Cursor. When you choose this option, Amazon SageMaker Unified Studio will open Cursor on your system and open your SageMaker Space in Cursor.

> **Note**
>
> You must be using AWS Toolkit version 3.100 or higher for Open in Cursor to function.

In Identity Center based domains:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO credentials.
2. In the left navigation pane, under **IDEs**, choose **JupyterLab**.
3. Once your JupyterLab Notebook opens, choose Open in Cursor in the top right corner of the page.

In IAM-based domains:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your IAM credentials.
2. In the left navigation, choose JupyterLab.

3. Once your JupyterLab Notebook opens, choose Open in Cursor in the top right corner of the page. This will establish a remote connection and open your SageMaker Space in Cursor.

> **Note**
>
> The sessions initiated through Open in Cursor link do not automatically reconnect if the remote session is interrupted. You will need to create a new remote connection using the Open in Cursor link in Amazon SageMaker Unified Studio. The sessions established through this link can last for 12 hours if not interrupted.

Connecting to Amazon SageMaker Unified Studio Spaces

Amazon SageMaker Unified Studio's remote connectivity establishes a secure tunnel between your local VS Code or Kiro and SageMaker Spaces, enabling interactive development and code execution using cloud compute resources while maintaining your familiar local environment.

> **Note**
>
> Ensure network configuration is setup to allow the SSH traffic. For domains set up in virtual private cloud (VPC)-only mode (default), your domain should have a route out to the internet through a proxy or a NAT gateway. For more details, see Integrated Gateways. If your domain is completely isolated from the internet, refer to connecting to a Space from the AWS Toolkit in this page for setting up the remote connection.

You must enable remote access for the Space to connect to the Space.

- For an existing Space, if the Space is in Stopped state, hover over the Space and choose the connect button. This should enable remote access, start the Space and connect to it.
- For an existing Space, if the Space is in Running state, the Space must be restarted with remote access enabled. You can do this by clicking Connect and following the prompts to restart with remote access enabled.
- To enable remote access, while creating a Space in the portal, under Remote Access, toggle the switch to Enabled.

To connect to a Space in Amazon SageMaker Unified Studio from the AWS Toolkit:

1. Under **Project**, choose **Compute** and then choose **Spaces**.
2. In the row of the Space you want to connect to, choose the icon that says **Connect to SageMaker Space** when you hover over it.
   - If your `instanceType` is less than 8 GB, a prompt appears asking you to start your Space with a larger `instanceType`. Choose yes to connect to your SageMaker Space. The state of the Space changes to **Running**.
3. After the connection is successful, a new window opens. This window shows all the files that you have access to in the selected project. All the changes made here are reflected in the Amazon SageMaker Unified Studio Space.

Domain and Project Switching

AWS Toolkit allows connecting to only one Amazon SageMaker Unified Studio Domain and Project at any point.

- To switch to a different domain, in the Amazon SageMaker Unified Studio panel in AWS Toolkit, click on the ellipsis at the top right corner. Choose **Sign out**, then sign back in and choose the required domain.

Within a domain, you can switch between multiple Amazon SageMaker Unified Studio projects which you have access to.

- To switch to a different project in a domain, in the Amazon SageMaker Unified Studio panel in AWS Toolkit, click on the ellipsis at the top right corner. Choose **Switch Project** and choose the required project from the list.

Refresh Project Metadata

The metadata in AWS Toolkit can become stale over time if you have not interacted with it or are creating/deleting resources in the Unified Studio portal. To get the latest metadata, you can refresh project resources.

- To refresh a project, in the Amazon SageMaker Unified Studio panel in AWS Toolkit, click on the ellipsis at the top right corner. Choose **Refresh Project**. All the project resources will be refreshed

and updated with the latest metadata. Alternatively, hover over the project node and click on the refresh button to refresh the resource metadata.

AWS Toolkit Experience in Remote Session

AWS Toolkit is also available in the remote IDE session. In the remote session, since you are already authenticated and are in a specific project, you are directly presented with the project resources. Currently the remote session offers:

- The ability to use Data Explorer to browse your project data sources within the IDE (S3 buckets, Redshift, Lakehouse)
- The ability for notebook users to pick Language and Compute Connections for each cell. This allows you to author polyglot notebooks where each cell can use its own language (Python, SQL, Markdown etc) and compute connection (Local Python, Athena, Redshift etc). This enables you to run code against your Spark and SQL connections from your local IDE through Amazon SageMaker Unified Studio Spaces.

aws

User Guide

Amazon SageMaker Unified Studio

Data sources and connections



Working with data

Data in Amazon SageMaker Unified Studio includes data in projects of which you are a member and data that you can discover and subscribe to from other projects.

The **Data** page in Amazon SageMaker Unified Studio displays a data browser in which you can explore datasets, files, and artifacts that you connect to your project. Projects configured with certain profiles contain an lakehouse architecture for accessing data within your project, as well as a default Amazon Redshift connection and an Amazon S3 bucket. You can add data to the project on the **Data** page by uploading data from your local desktop or by gaining access to existing data sources and then adding a connection to them in your Amazon SageMaker Unified Studio project. For more information about lakehouse architecture, see What is the lakehouse architecture of Amazon SageMaker?.

You can also connect to AWS Glue and Amazon Redshift data sources from within your project catalog. The project catalog contains your data as data products and assets with metadata. When you want to share your data with other projects in the domain, publish the data from your project catalog into the Amazon SageMaker Catalog. If you want to create more detailed access control for your data before allowing other users to subscribe to it, you can configure fine-grained access control. For more information, see the section called “Data inventory and publishing” and the section called “Fine-grained access control to data”.

The Amazon SageMaker Catalog contains business glossaries and metadata forms. If you have been granted access through the authorization policies, you can create business glossaries and metadata forms. For more information, see Domain units and authorization policies in Amazon SageMaker Unified Studio and *Catalog in IDC-based domains*.

You can use the Amazon SageMaker Catalog to discover and subscribe to assets and data products. For more information, see the section called “Data discovery, subscription, and consumption”.

Bringing existing resources into Amazon SageMaker Unified Studio

You can bring in existing resources to your Amazon SageMaker Unified Studio project by using the **Data** and **Compute** pages in your project, or by using scripts provided in GitHub.

Examples of resources you can bring into Amazon SageMaker Unified Studio are listed below.

AWS Glue Data Catalogs

- A GitHub script for bringing this resource into Amazon SageMaker Unified Studio can be found here.

Amazon S3 data

To bring your existing Amazon S3 data and use it in Amazon SageMaker Unified Studio, follow the guide in the link below. This guide explains how you can configure permissions and customize role assignments for Amazon SageMaker Unified Studio to access your Amazon S3 data from a project.

- A GitHub script for bringing this resource into Amazon SageMaker Unified Studio can be found here.
- To add Amazon S3 connections using the Amazon SageMaker Unified Studio interface, see the Data storage section.

Amazon Athena workgroups and saved queries

- A GitHub script for bringing this resource into Amazon SageMaker Unified Studio can be found here.

Amazon EMR on EC2 clusters

- A GitHub script for bringing this resource into Amazon SageMaker Unified Studio can be found here.

- To bring in existing clusters using the Amazon SageMaker Unified Studio interface, see the section called "Adding an existing Amazon EMR on EC2 cluster".

Amazon Redshift clusters and Amazon Redshift Serverless workgroups

- To bring in existing Amazon Redshift clusters and workgroups using the Amazon SageMaker Unified Studio interface, see the section called "Connecting to an existing Amazon Redshift resource".

Amazon SageMaker AI resources

- Amazon SageMaker Studio users can bring their existing SageMaker AI domains, user profiles, and spaces into Amazon SageMaker Unified Studio. This integration also supports additional SageMaker AI resources such as training jobs, machine learning pipelines, models, inference endpoints, and more. A package of GitHub scripts for bringing Amazon SageMaker AI resources into Amazon SageMaker Unified Studio can be found here.

AWS IAM roles

Use the utility script in GitHub to configure permissions and customize role assignments for Amazon SageMaker Unified Studio.

- A GitHub script for bringing this resource into Amazon SageMaker Unified Studio can be found here.

> **Note**
>
> Review the prerequisites carefully before proceeding to execute the script. Ensure that you save your work and that you do not have any running tasks or processes such as reconfiguring a JupyterLab space or creating a new compute resource. These processes might get interrupted or cause the script to fail.

Data in Identity Center-based domains

Topics

- Amazon S3 data in Amazon SageMaker Unified Studio
- The lakehouse architecture of Amazon SageMaker

Amazon S3 data in Amazon SageMaker Unified Studio

You can bring in Amazon S3 data to your project and access it on the **Data** page of your project in Amazon SageMaker Unified Studio.

To add S3 tables to your lakehouse in Amazon SageMaker Unified Studio, see Amazon S3 tables integration.

To add S3 data as assets in your Amazon SageMaker Unified Studio project catalog, see the section called “Adding Amazon S3 data”. In Amazon SageMaker Unified Studio, assets represent specific types of data resources such as database tables, dashboards, S3 buckets or prefixes, or machine learning models.

For S3 data in projects, SageMaker Catalog supports the creation of an asset type of **S3 Object Collection** for an Amazon S3 bucket or S3 prefix in the project. The S3 Object Collection asset type can be curated with business context metadata by adding business names, descriptions, README, glossary terms, and metadata forms, including mandatory metadata forms. Assets in Amazon SageMaker Unified Studio are versioned as changes are made in metadata.

Adding Amazon S3 data

To bring in Amazon S3 data to your project, you must first gain access to the data and then add the data to your project. You can gain access to the data by using the project role or an access role.

> **Note**
>
> If you are using a bucket in a different account than the account that contains the project tooling environment, you must use an access role to gain access to the data.

Prerequisite option 1 (recommended): Gain access using an access role

Work with your admin to complete the following steps:

1. Retrieve the project role ARN and the project ID and send them to your admin.
   a. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
   b. Navigate to the project that you want to add Amazon S3 data to. You can do this by choosing a project from the project selector dropdown at the top of the page.
   c. On the **Project overview** page, copy the project role ARN and the project ID.
2. The admin then must go to the IAM console and create an access role.
   a. Navigate to the IAM console.
   b. On the **Roles** page, choose **Create role**.
   c. Under **Trusted entity type**, choose **Custom trust policy**.
   d. Edit the policy to include the project ID, the project ARN, and the AWS account ID to grant Amazon S3 access permissions.

      JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "access-grants.s3.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole",
                "sts:SetSourceIdentity"
            ],
            "Condition": {
                "StringEquals": {
                "aws:SourceAccount": "111122223333"
                }
            }
```

```
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "project-role-arn"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": "project-id"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "project-role-arn"
            },
            "Action": [
                "sts:SetSourceIdentity"
            ],
            "Condition": {
                "StringLike": {
                    "sts:SourceIdentity": "${aws:PrincipalTag/
datazone:userId}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "project-role-arn"
            },
            "Action": "sts:TagSession",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/AmazonDataZoneProject": "project-id",
                    "aws:RequestTag/AmazonDataZoneDomain": "domain-id"
                }
            }
        }
    ]
}
```

e. Choose **Next** twice.

f. Enter a name for the role, then choose **Create role**.

g. Select the access role from the list on the **Roles** page.

h. On the **Permissions** tab of the role, choose **Add permissions**, then **Create inline policy**.

i. Use the JSON editor to create a policy that grants Amazon S3 access permissions. The following example shows a policy that grants access to multiple S3 buckets and folders. Replace the bucket and folder names with your own values.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3AdditionalBucketPermissions",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::bucket1",
                "arn:aws:s3:::bucket2",
                "arn:aws:s3:::bucket3"
            ]
        },
        {
            "Sid": "S3AdditionalObjectPermissions",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject*",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket1/*",
                "arn:aws:s3:::bucket2/folderA/*",
                "arn:aws:s3:::bucket3/*"
            ]
        },
        {
            "Sid": "S3AGLocationManagement",
            "Effect": "Allow",
```

```
            "Action": [
                "s3:CreateAccessGrantsLocation",
                "s3:DeleteAccessGrantsLocation",
                "s3:GetAccessGrantsLocation"
            ],
            "Resource": [
                "arn:aws:s3:*:*:access-grants/default/*"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "s3:accessGrantsLocationScope": [
                        "s3://bucket1/",
                        "s3://bucket2/folderA/",
                        "s3://bucket3/"
                    ]
                }
            }
        },
        {
            "Sid": "S3AGPermissionManagement",
            "Effect": "Allow",
            "Action": [
                "s3:CreateAccessGrant",
                "s3:DeleteAccessGrant"
            ],
            "Resource": [
                "arn:aws:s3:*:*:access-grants/default/location/*",
                "arn:aws:s3:*:*:access-grants/default/grant/*"
            ],
            "Condition": {
                "ForAnyValue:StringLike": {
                    "s3:accessGrantScope": [
                        "s3://bucket1/*",
                        "s3://bucket2/folderA/*",
                        "s3://bucket3/*"
                    ]
                }
            }
        },
        {
            "Sid": "AllowPassRoleOnProjectRoles",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
```

```
            ],
            "Resource": "access-role-arn",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "access-grants.s3.amazonaws.com"
                    ],
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "S3AGTaggingPermission",
            "Effect": "Allow",
            "Action": [
                "s3:TagResource",
                "s3:ListTagsForResource"
            ],
            "Resource": [
                "arn:aws:s3:*:*:access-grants/default/location/*",
                "arn:aws:s3:*:*:access-grants/default/grant/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        }
    ]
}
```

Note

Replace *access-role-arn* with the ARN of this access role. For a single bucket, you can use `StringEquals` and `StringLike` instead of `ForAnyValue:StringEquals` and `ForAnyValue:StringLike` with a single value instead of an array.

> **Note**
>
> Amazon SageMaker Unified Studio grants access to subscribed assets using S3 Access Grants. To enable granting access to data using S3 Access Grants, an S3 Access Grants instance is required. Amazon SageMaker Unified Studio will use an instance if one is already available or will create one. S3 Access Grants needs one instance per AWS Region in a single AWS account. For more information, see Working with S3 Access Grants instances

j. Choose **Next**.

k. Enter a name for the policy, then choose **Create policy**.

l. Optional: if you want to support cross-account data sharing for S3, add the following to your policy:

```
{
    "Sid": "CrossAccountS3AGResourceSharingPermissions",
    "Effect": "Allow",
    "Action": [
        "ram:CreateResourceShare"
    ],
    "Resource": "*",
    "Condition": {
        "StringEqualsIfExists": {
            "ram:RequestedResourceType": [
                "s3:AccessGrants"
            ]
        },
        "StringEquals": {
            "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
    }
},
{
    "Sid": "CrossAccountS3AGResourceSharingPolicyPermissions",
    "Effect": "Allow",
    "Action": [
        "s3:PutAccessGrantsInstanceResourcePolicy"
    ],
```

```
    "Resource": "arn:aws:s3:*:*:access-grants/default",
    "Condition": {
        "StringEquals": {
            "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
    }
}
```

m. Choose **Next**.

n. Enter a name for the policy, then choose **Create policy**.

o. Optional: If the bucket is in a different account than the the access role, ensure cross-account bucket permissions are set by adding a bucket policy that grants cross-account permissions to the access role. For example:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "S3AdditionalBucketPermissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "access-role-arn"
            },
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::bucketName"
            ]
        },
        {
            "Sid": "S3AdditionalObjectPermissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "access-role-arn"
            },
            "Action": [
```

```
                "s3:GetObject*",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucketName/key/*"
            ]
        }
    ]
}
```

p. Choose **Update policy**.

Prerequisite option 2: Gain access using the project role

Work with your admin to complete the following steps:

1. Retrieve the project role ARN and send it to your admin.
   a. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
   b. Navigate to the project that you want to add Amazon S3 data to. You can do this by choosing a project from the project selector dropdown at the top of the page.
   c. On the **Project overview** page, copy the project role ARN.
2. The admin then must go to the IAM console and update the project role.
   a. Navigate to the IAM console.
   b. On the **Roles** page, search for the project role using the last string in the project role ARN, for example: `datazone_usr_role_1a2b3c45de6789_abcd1efghij2kl`.
   c. Select the project role to navigate to the project role details.
   d. Under the **Permissions** tab, choose **Add permissions**, then choose **Create inline policy**.
   e. Use the JSON editor to create a policy so that the project has access to Amazon S3 locations, using the Amazon S3 resource ARNs. The following example shows a policy that grants access to multiple S3 buckets and folders. Replace the bucket and folder names with your own values.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "S3AdditionalBucketPermissions",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::bucket1",
                "arn:aws:s3:::bucket2",
                "arn:aws:s3:::bucket3"
            ]
        },
        {
            "Sid": "S3AdditionalObjectPermissions",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject*",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket1/*",
                "arn:aws:s3:::bucket2/folderA/*",
                "arn:aws:s3:::bucket3/*"
            ]
        }
    ]
}
```

f. Choose **Next**

g. Enter a name for the policy, then choose **Create policy**.

3. Under the **Permissions** tab, choose **Add permissions**, then choose **Create inline policy**.

4. Use the JSON editor to create a policy so that the project has access to an Amazon S3 location, using the Amazon S3 resource ARN that you noted previously.

```
{
    "Sid": "S3AGLocationManagement",
    "Effect": "Allow",
    "Action": [
```

```
            "s3:CreateAccessGrantsLocation",
            "s3:DeleteAccessGrantsLocation",
            "s3:GetAccessGrantsLocation"
        ],
        "Resource": [
            "arn:aws:s3:*:*:access-grants/default/*"
        ],
        "Condition": {
            "ForAnyValue:StringEquals": {
                "s3:accessGrantsLocationScope": [
                    "s3://bucket1/",
                    "s3://bucket2/folderA/",
                    "s3://bucket3/"
                ]
            }
        }
    },
    {
        "Sid": "S3AGPermissionManagement",
        "Effect": "Allow",
        "Action": [
            "s3:CreateAccessGrant",
            "s3:DeleteAccessGrant"
        ],
        "Resource": [
            "arn:aws:s3:*:*:access-grants/default/location/*",
            "arn:aws:s3:*:*:access-grants/default/grant/*"
        ],
        "Condition": {
            "ForAnyValue:StringLike": {
                "s3:accessGrantScope": [
                    "s3://bucket1/*",
                    "s3://bucket2/folderA/*",
                    "s3://bucket3/*"
                ]
            }
        }
    }
```

> **Note**
>
> For a single bucket, you can use `StringEquals` and `StringLike` instead of `ForAnyValue:StringEquals` and `ForAnyValue:StringLike` with a single value instead of an array.

> **Note**
>
> Amazon SageMaker Unified Studio grants access to subscribed assets using S3 Access Grants. To enable granting access to data using S3 Access Grants, an S3 Access Grants instance is required. Amazon SageMaker Unified Studio will use an instance if one is already available or will create one. S3 Access Grants needs one instance per AWS Region in a single AWS account. For more information, see Working with S3 Access Grants instances

5. Choose **Next**.
6. Enter a name for the policy, then choose **Create policy**.

Add the data to your project

When your admin has granted your project access to the Amazon S3 resources, you can add them to your project.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project that you want to add Amazon S3 data to.
3. On the **Data** page, choose the plus icon **+**.
4. Select **Add S3 location**, then choose **Next**.
5. Enter a name for the location path.
6. (Optional) Add a description of the location path.
7. Use the S3 URI and Region provided by your admin.
8. If your admin has granted you access using an access role instead of the project role, enter the access role ARN from your admin.

9. Choose **Add S3 location**.

The Amazon S3 data is then accessible within your project in the left navigation on the **Data** page.

Verify the setup

After creating the connection, verify that the S3 Access Grants location was registered correctly:

1. Navigate to the Amazon S3 console.
2. In the left navigation, choose **Access Grants**.
3. Choose **View Details**, then choose **Locations**.
4. Confirm that the S3 URI is registered against the IAM role you created.

> **Note**
>
> If the connection failed to register the Amazon S3 Access Grants location, you must delete the existing connection and create a new one.

Sharing Amazon S3 data

Sharing data with other users in Amazon SageMaker Unified Studio means that you and other users can access the same data in multiple projects. There are two ways to share data with other users in Amazon SageMaker Unified Studio:

- Publish Amazon S3 data to the catalog. This means that other projects can create subscription requests to request access to the data you publish. When you approve a subscription request, the other project will then have access to that data.
- Share Amazon S3 data directly with consumers. This means that the data you share is available to the projects you specify right away, without needing a subscription process.

In both cases, you can track and manage access to your data in the **Catalog management** section under **Manage** in your project in Amazon SageMaker Unified Studio. You have the option to choose whether to grant read-only or read and write access.

Amazon SageMaker Unified Studio grants access to subscribed assets using Amazon S3 Access Grants. When a subscription is revoked, a project member may still get Amazon S3 Access Grants

credentials for up to 5 minutes, and credentials can be used for 15 minutes. As a result, a user may have access to the data for up to 20 minutes after the access is revoked in SageMaker Unified Studio.

Publish Amazon S3 data to the catalog

When you publish data to the Amazon SageMaker Catalog, other projects in your Amazon SageMaker Unified Studio domain can create subscription requests to request access to the data you published. When you approve a subscription request, the other project will then have access to that data.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project that contains your Amazon S3 connection.
3. On the **Data** page, in the side navigation, choose **S3** to explore your S3 data assets.
4. Choose the name of the S3 folder or bucket you want to publish.
5. Choose **Actions**, then choose **Publish to Catalog**. A confirmation window appears.
6. Choose **Publish** to confirm that you want the S3 data to be discoverable in the Amazon SageMaker Catalog. This means that members of other projects in the domain can create subscription requests for the data asset. When you review subscription requests, you will have the option to grant read-only or read and write access to the data. If you approve the subscription request, they will have access to the S3 data asset in their project.

The S3 data folder or bucket you published then appears in the Amazon SageMaker Catalog as a data asset of type **S3 Object Collection**.

Share Amazon S3 data directly with consumers

Sharing data directly in this way makes it so that the data you share is available to the projects you specify right away, without needing a subscription process. You will first publish your dataset, and then share it.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project that contains your Amazon S3 connection.
3. On the **Data** page, in the side navigation, explore your S3 data assets.
4. Choose the name of the S3 folder or bucket you want to publish.

5. Choose **Actions**, then choose **Publish to Catalog**. A confirmation window appears.
6. Choose **Publish**
7. Navigate to the **Assets**page from the left navigation and select the asset you want to share.
8. Choose **Actions**, then choose **Share**.
9. Use the dropdown to select projects that you want to share the S3 data with, then choose **Next**.
10. Select the access type. For read-only access, move to the next step. To grant read and write access to users in the other project, select **Read and write access**.
11. Choose **Share**.

The S3 data asset is then shown in the project catalog under approved subscrion requests. You can choose to revoke access at any time. For more information about subscription requests, see the section called “Data discovery, subscription, and consumption”.

The lakehouse architecture of Amazon SageMaker

The lakehouse architecture of Amazon SageMaker is a unified data architecture built on AWS's cloud-native infrastructure that bridges Amazon S3 data lakes and Amazon Redshift data warehouses into a cohesive analytics platform. The architecture leverages Apache Iceberg table format for cross-service interoperability and implements a shared metadata catalog that provides consistent data access patterns across storage systems.

For more information about the lakehouse architecture of Amazon SageMaker, see What is the lakehouse architecture of Amazon SageMaker?.

Data and catalog connections in IAM-based domains

Amazon SageMaker Unified Studio notebooks can connect to multiple data sources including Amazon S3, AWS Glue Data Catalog, Amazon Athena, Amazon Redshift, and third-party sources. You can query data directly from these sources using SQL cells or Python code. The notebook interface provides built-in connectors for AWS services and supports custom connections for external data sources. Data connections are configured at the project level and shared across notebooks.

Prerequisites

1. Configured data connections in your Amazon SageMaker Unified Studio project
2. Appropriate IAM permissions to access data sources
3. Network connectivity to external data sources if applicable

Supported data connections

Amazon SageMaker Unified Studio supports the following data connections for IAM-based domains:

Databases and data warehouses

- Amazon DocumentDB
- Amazon OpenSearch Service
- Amazon DynamoDB
- Amazon Redshift
- Aurora MySQL
- Aurora PostgreSQL
- Azure SQL
- Google BigQuery
- Microsoft SQL Server
- MySQL

- Oracle
- PostgreSQL
- Snowflake

Storage

- Amazon S3

AWS resources created by connections

When you create a connection in Amazon SageMaker Unified Studio, the following resources are created in your AWS account(s) behind the scenes:

- AWS Glue connection - a connection object that stores core connection information.

Those resources are visible in the account where Amazon SageMaker Unified Studio domain is hosted and you can discover and describe them through Console or API/SDK/CLI of the corresponding service (in this case - AWS Glue).

Connecting to a new data source

Domain and project VPC configuration

Data sources Amazon DocumentDB, Amazon Redshift, Aurora MySQL, Aurora PostgreSQL, Azure SQL, PostgreSQL, Oracle, Microsoft SQL Server require your project to be configured with a VPC Step 1, Step 2. You might need to contact administrator to complete that configuration.

To connect to a new data source

1. In the navigation pane, choose **Connections**.
2. Choose **Create connection**.
3. Select the connection type in the gallery that opens.
4. For **Name** enter a descriptive name for your connection.
5. Configure the connection parameters based on your selected connection type.
    - You can select SSL connection for supported data sources.

6. Configure the authentication details. You can either enter **Username** and **Password** directly in the connection details, or use a secret in AWS Secrets Manager that stores the credentials. You might need to contact your administrator to create a new secret for the connection.
7. Choose **Create connection**.
8. If all validations pass, a new connection will be created.

Your connection will appear under **Data section** in the navigation pane. In that section, select **Connections** to see the list of all available connections. From there you can see connection details such as **Name**, **Connection type**, and **Authorization mode**. You can also check the connection status.

SSL connection

You can configure SSL connection for Amazon DocumentDB, Amazon Redshift, Aurora MySQL, Aurora PostgreSQL, Azure SQL, Microsoft SQL Server, MySQL, Oracle, and PostgreSQL connection types.

SSL connection enables encrypted communication between Amazon SageMaker Unified Studio and your data source. When enabled, all data transmitted between your notebooks and the database is encrypted using SSL/TLS protocols, protecting sensitive information in transit. This setting is recommended for production environments and required when connecting to databases that enforce SSL connections.

To enable SSL connection select Enforce SSL checkbox when configuring connection properties.

> **Note**
>
> Enabling SSL may slightly increase connection latency due to the encryption overhead.

Connecting to Amazon S3

You can create a data connection to Amazon S3 when you need to directly access files stored in Amazon S3 buckets from your notebooks. This connection is only required if you want to read or write individual files (such as CSV, JSON, or Parquet files) directly from Amazon S3 storage. If you are working with Data Catalog tables that are backed by Amazon S3, you do not need to create a separate Amazon S3 connection, you can access those tables directly through the catalog.

Before connecting to Amazon S3, complete the one of the following prerequisite options:

- Prerequisite option 1 (recommended): Gain access using an access role
- Prerequisite option 2: Gain access using the project role

To connect to Amazon S3

1. In the navigation pane, choose **Connections**.
2. Choose **Create connection**.
3. In the gallery that opens, select Amazon S3.
4. For **Name** enter a descriptive name for your connection.
5. Enter S3 URI - optional. If you do not specify an Amazon S3 URI, SageMaker Unified Studio will list all buckets accessible with the provided credential.
6. AWS region - enter the AWS region where the S3 bucket is located.
7. Access role ARN - optional - select an existing IAM role from the dropdown. You might need to contact your Administrator for configuring an access role if you are connecting to S3 bucket in an AWS account that is different from the AWS account where your SageMaker Unified Studio domain is hosted.
8. Choose **Create connection**.
9. If all validations pass, a new Amazon S3 connection will be created.

After creating the connection, you can use it in your notebooks to read and write files directly from the specified S3 location. You can also all the buckets you connected to if you select Data on navigation pane and select S3 buckets tab.

Connecting to Amazon Redshift

You can create a data connection to Amazon Redshift to query data warehouses from Amazon SageMaker Unified Studio. You can connect to both provisioned clusters and serverless workgroups.

VPC Requirements

Amazon Redshift connections use different connection methods depending on the tool you are using in Amazon SageMaker Unified Studio.

- Visual ETL and data processing jobs require your Amazon Redshift database to be in the same VPC as Amazon SageMaker Unified Studio domain. To configure that refer to the following documentation: Step 1, Step 2. You might need to contact Administrator to complete that configuration.
- Query Editor and Data notebooks do not require VPC configuration.

Steps to connect to Amazon Redshift from Amazon SageMaker Unified Studio

1. In the navigation pane, choose **Connections**.
2. Choose **Create connection**.
3. In the gallery that opens, select Amazon Redshift.
4. For **Name** enter a descriptive name for your connection.
5. Configure the connection properties:
   a. **Redshift compute** - Select existing compute or enter a JDBC URL.
      - JDBC URL format: `jdbc:redshift://endpoint:port/database`.
   b. **EnforceSSL** - Select to enable encrypted communication (recommended).
   c. **JDBC URL Parameters** - Optional configuration parameters for the JDBC/ODBC driver in the following format: `<key1>=<value1>;<key2>=<value2>`.
6. Authentication option, you can select one of the options below
   a. **IAM** - generates a database username based on your IAM identity. You can apply grants directly to this user, or use the RedshiftDbUser principal tag on your project execution role to connect as a different database user.
   b. **Username and password** - Provide a username and password for the database that you are connecting to. We will store your credentials in AWS Secrets Manager.
   c. **AWS Secrets Manager** - Choose a secret in AWS Secrets Manager with the credentials. You might need to contact your Administrator to create a secret.
7. **Access role ARN** - optional. Required when connecting to resources in a different AWS account. See Gaining access to Amazon Redshift resources.
8. Choose **Create connection**.

Steps to connect to Amazon SageMaker Unified Studio from Amazon Redshift console pages

These steps describe how Amazon Redshift customers can automatically create an Amazon Redshift connection within the Amazon SageMaker Unified Studio environment from the Amazon Redshift Management Console and Amazon Redshift Query Editor v2 (QEv2). This functionality is available from the following Amazon Redshift console pages:

- Landing page
- Provisioned dashboard
- Serverless dashboard
- Cluster list
- Cluster detail
- Serverless workgroup and namespace list pages
- Workgroup and namespace detail pages

When a user selects the "Query in SageMaker Unified Studio" option, they will be prompted to choose the Amazon Redshift cluster they want to use. Amazon SageMaker Unified Studio will then issue temporary credentials and provision the session, redirecting the user into the Amazon SageMaker Unified Studio environment where they can begin querying data immediately.

Steps to connect to Amazon SageMaker Unified Studio from Amazon Redshift Query Editor v2

These steps describe how Amazon Redshift customers can automatically create an Amazon Redshift connection within the Amazon SageMaker Unified Studio environment from the Amazon Redshift Query Editor v2 (QEv2). This functionality is available within Amazon Redshift Query Editor v2 under a "Run in SageMaker Unified Studio" button in the following locations.

- As a tooltip option next to the "Run" button in the SQL editor
- As a tooltip option next to the "Run" button in notebook cells

If the connection selected is using IAM authentication then when the user clicks on "Run in Amazon SageMaker Unified Studio" button, the query currently typed in the active editor (SQL or notebook cell) will be executed directly in the Amazon SageMaker Unified Studio environment.

Redshift Access using IAM Authentication

IAM authentication generates temporary database credentials based on your AWS identity, eliminating the need to manage database passwords. When you connect using IAM authentication, Amazon SageMaker Unified Studio maps your IAM identity to a database user.

How IAM Authentication Works

When connecting with IAM authentication, your database username is automatically generated based on your IAM identity. By default, the username follows the pattern `IAMR:<your-federated-identity>`. You can customize this behavior using the RedshiftDbUser principal tag on your project execution role to specify a different database user.

Using Principal Tags

Principal tags allow you to control which database user is used when connecting to Redshift. Configure the RedshiftDbUser tag on your project execution role in IAM. When this tag is set, your connection will use `IAMR:<tag-value>` as the database username instead of the federated IAM identity.

For more information about setting up principal tags, see Setting up principal tags to connect a cluster or workgroup from query editor v2.

Troubleshooting Database Access

If after establishing a connection, you do not see expected databases, schemas, or tables, follow the steps described below.

First identify your effective database user by running the following SQL:

```
SELECT current_user;
```

Your database username follows these patterns:

- Federated: `IAMR:<your-federated-identity>`
- With RedshiftDbUser tag on the project execution role: `IAMR:<tag-value>`

Common Access Issues

Databases are not displayed

Your database user lacks connection privileges. The database administrator must grant:

```
GRANT CONNECT ON DATABASE <database_name> TO "IAMR:<your-identity>";
```

Schemas are not displayed

Your database user lacks schema usage privileges. The database administrator must grant:

```
GRANT USAGE ON SCHEMA <schema_name> TO "IAMR:<your-identity>";
```

Tables are not displayed

Your database user lacks table access privileges. The database administrator must grant:

```
GRANT SELECT ON ALL TABLES IN SCHEMA <schema_name> TO "IAMR:<your-identity>";
```

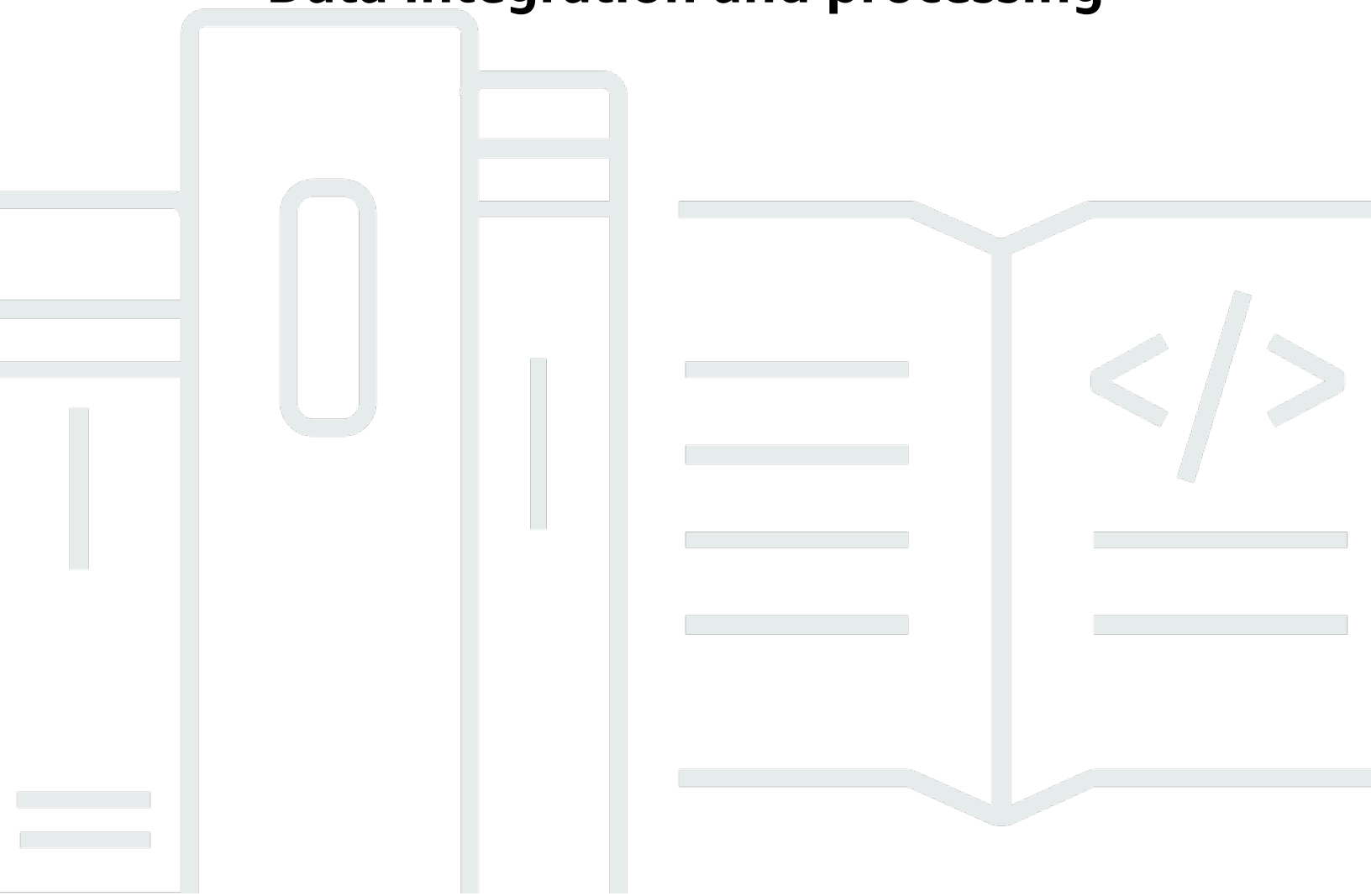
For more information on grants in Redshift, see GRANT.

aws

User Guide

Amazon SageMaker Unified Studio

Data integration and processing



Using Jobs in SageMaker Unified Studio

Jobs in SageMaker Unified Studio provide a comprehensive platform for creating, managing, and executing data processing workloads across your organization. As a core component of SageMaker Unified Studio, jobs enable you to unify your data engineering, machine learning, and analytics workflows in a single environment.

Whether you're migrating existing AWS Glue jobs or building new data processing pipelines, jobs provide the flexibility to work with your preferred authoring tools while maintaining enterprise-grade monitoring, governance, and scalability. This chapter covers everything you need to know about using jobs effectively in SageMaker Unified Studio.

SageMaker data processing job encapsulates a Python script that connects to your source data, processes it, and then writes it out to your data target. Typically, a job runs extract, transform, and load (ETL) scripts. SageMaker jobs can be started based on a schedule or event, or on demand. You can monitor job runs to understand runtime metrics such as completion status, duration, and resources consumed.

When processing your data, SageMaker can write output files in several data formats, including JSON, CSV, ORC (Optimized Row Columnar), Apache Parquet, and Apache Avro. For some data formats, common compression formats can be written to optimize storage and performance.

Benefits of using jobs

- **Simple** - Jobs are easy to learn, understand, create, run, and monitor.
- **Flexible** - Jobs support a wide variety of workloads, capabilities, and compute engines.
- **Scalable** - Jobs support your biggest processing workloads.

Creating jobs

You can create jobs in SageMaker Unified Studio using multiple authoring methods to suit different personas and use cases.

Job authoring methods

IDE/Script Editor

Data engineers can develop jobs using either their preferred external IDE or the SageMaker Unified Studio's built-in code editor. These environments provide full code editing capabilities and remote debugging support. Once scripts are written, engineers can create data processing jobs from them directly within SageMaker Unified Studio.

Visual ETL

ETL developers can create jobs using a drag-and-drop visual interface for building data transformation pipelines without writing code.

Jupyter Notebooks

Data scientists can develop jobs using familiar Jupyter Notebook interfaces for interactive development and exploration.

> **Note**
>
> When creating a job from a Jupyter notebook (.ipynb file), SageMaker Unified Studio automatically converts the notebook to a Python script. Once this conversion happens, subsequent changes to the original notebook will not affect the job. The job will continue to use the Python script created at the time of job creation.

To create a new job

1. Sign in to SageMaker Unified Studio and navigate to your project.
2. In **Data Analysis & Integration** , choose **Data processing jobs**, then **Visual ETL job or Code-based job**. Alternatively, you can create a Job within an existing Workflow, or through JupyterLab in **IDE & Applications**.
3. Select your preferred authoring method (IDE, Visual ETL, or Notebook).
4. Configure your job settings.

   Name

   Provide a UTF-8 string with a maximum length of 255 characters. Once a job is named, it cannot be renamed as the name is a unique identifier.

Description

Provide an optional description of up to 2048 characters.

IAM Role

Specify the IAM role that is used for authorization to resources used to run the job and access data stores. For more information about permissions for running jobs in AWS Glue, see Identity and access management for AWS Glue. Note that IAM role cannot be changed after job creation (edit IAM roles at the project level).

Type

The type of ETL job. This is set automatically based on the type of data sources you select.

SageMaker supports Apache Spark jobs that run in an Apache Spark environment managed by AWS Glue. These jobs process data in batches and provide powerful distributed computing capabilities.

- **Spark** - Runs an Apache Spark ETL script.

AWS Glue version

AWS Glue version determines the versions of Apache Spark and Python that are available to the job, as specified in the following table:

AWS Glue version	Supported Spark and Python versions
5.1	Spark 3.5.6, Python 3.11
5.0	Spark 3.5.4, Python 3.11
4.0	Spark 3.3.0, Python 3.10

Language

The code in the ETL script defines your job's logic. The script is in Python. You can choose whether the script that the job runs is generated by AWS Glue or provided by you. You provide the script name and location in Amazon S3. Confirm that there isn't a file with the same name as the script directory in the path.

Instance type

The resources available on AWS Glue workers are measured in DPUs. A DPU is a relative measure of processing power that consists of 4 vCPUs of compute capacity and 16 GB of memory.

- **G.1X** – When you choose this type, you also provide a value for Number of workers. Each worker maps to 1 DPU (4 vCPUs, 16 GB of memory) with 94GB disk. We recommend this instance type for workloads such as data transforms, joins, and queries, to offers a scalable and cost effective way to run most jobs.
- **G.2X** – When you choose this type, you also provide a value for Number of workers. Each worker maps to 2 DPU (8 vCPUs, 32 GB of memory) with 138GB disk. We recommend this instance type for workloads such as data transforms, joins, and queries, to offers a scalable and cost effective way to run most jobs.
- **G.4X** – When you choose this type, you also provide a value for Number of workers. Each worker maps to 4 DPU (16 vCPUs, 64 GB of memory) with 256GB disk (approximately 235GB free). We recommend this worker type for jobs whose workloads contain your most demanding transforms, aggregations, joins, and queries.
- **G.8X** – When you choose this type, you also provide a value for Number of workers. Each worker maps to 8 DPU (32 vCPUs, 128 GB of memory) with 512GB disk (approximately 487GB free). We recommend this worker type for jobs whose workloads contain your most demanding transforms, aggregations, joins, and queries.

You are charged an hourly rate based on the number of DPUs used to run your ETL jobs. For more information, see the AWS Glue pricing page.

Number of instances

Specify the number of instances that are allocated when the job runs.

Number of retries

Specify the number of times, from 0 to 10, that AWS Glue will automatically restart the job if it fails. Jobs that reach the timeout limit are not restarted.

Job timeout

Sets the maximum execution time in minutes. The maximum is 7 days or 10,080 minutes. Otherwise, the jobs will throw an exception.

Best practices for job timeouts: Jobs are billed based on execution time. To avoid unexpected charges, configure timeout values appropriate for the expected execution time of your job.

Compute connection

Specifies the compute environment configuration for job execution. This parameter allows you to define how your job connects to and utilizes compute resources.

Network connections

Specify your network connections here, if any.

Enable fine-grained access control (FGAC)

Enables enhanced security controls for data access within your job. This provides granular permissions management, allowing you to control access to specific data sources, tables, or columns based on user roles and policies.

Spark UI

Turn on the use of Spark UI for monitoring this job. This will write Spark UI logs to Amazon S3.

Spark UI logs path

The path to write logs when Spark UI is enabled.

> **Note**
>
> **Important job configuration restrictions:**
>
> - Once a job is named, it cannot be renamed (the name is a unique identifier).
> - IAM role cannot be changed after job creation (edit IAM roles at the project level).
> - Job type cannot currently be changed after creation.

You can also configure advanced settings, job parameters, tags, and non overridable arguments. See the sections below for more information.

5. Write or design your data processing logic using your chosen authoring method.
6. Save your Job to your project.

Advanced settings

Automatically scale the number of instances

When enabled, costs and resource usage are optimized by scaling the number of instances up and down dynamically throughout the job run.

Generate lineage events

When enabled, Spark lineage events from your job runs are captured and sent to Amazon DataZone to visualize the table and column lineage.

Job bookmark

When enabled, specifies how state information is processed when the job runs. You can have it remember previously processed data, update state information, or ignore state information.

Job run queuing

When enabled, job runs are queued to run later when they cannot run immediately due to service quotas. If not enabled, the job runs will not be considered for queueing. If this setting does not match the value set in the job run, then the value from the job run field will be used.

Flex execution

When enabled, you can reduce costs by running the job on lower capacity. This is ideal for non-urgent workloads that don't require fast jobs start times or consistent run times.

Script path

The Amazon S3 location where the job script is stored.

Job metrics

Enables the creation of Amazon CloudWatch metrics when this job runs.

Job observability metrics

Enables the creation of additional observability Amazon CloudWatch metrics when this job runs.

Maximum concurrency

Sets the maximum number of concurrent runs that are allowed for this job.

Temporary path

The location of a working directory in Amazon S3 where temporary intermediate results are written when the job script is run.

Delay notification threshold (minutes)

Specify a delay threshold for the job. If the job runs for a longer time than that specified by the threshold, then a delay notification is sent for the job to Amazon CloudWatch.

Server-side encryption

If you select this option, when the ETL job writes to Amazon S3, the data is encrypted at rest using SSE-S3 encryption. Both your Amazon S3 data target and any data that is written to an Amazon S3 temporary directory is encrypted. This is ignored if a security configuration is specified.

Use AWS Glue Data Catalog as the Hive metastore

Choose this option if you want to use the AWS Glue Data Catalog as an alternative to Apache Hive Metastore.

Python library path, Dependent jars path, or Referenced files path

Use these fields to specify the location of additional files used by the job when it runs the script.

Job parameters and arguments

SageMaker Unified Studio jobs support parameters that can be configured during job creation and optionally overridden at runtime.

Parameter overrides

Parameters defined during job creation that can be overridden when running the job through custom runs, workflows, or schedules. This allows the same job to be used for multiple use cases by changing parameters at runtime.

Job tags

SageMaker Unified Studio provides native support for tagging jobs, which helps with organization, filtering, and permission management.

Tag use cases

- Team identification (e.g., "data-processing-team-a").
- Domain categorization (e.g., "finance", "marketing").
- Functional purpose (e.g., "data-governance", "quality-check").

- Environment designation (e.g., "production", "development").

Tag benefits

- Filter jobs in the jobs list view.
- Identify job ownership and purpose.
- Set permission boundaries for job access.
- Group related jobs for reporting and management.

Non Overridable arguments

Parameters that cannot be changed at runtime, regardless of how the job is triggered. These parameters are locked to their original values set during job creation.

Use cases for non-overridable arguments

- Security settings (e.g., fine-grain access control).
- Critical configuration that should remain consistent.
- Compliance requirements that must be enforced.

Running jobs

Jobs can be run in multiple ways depending on your workflow requirements.

Run Methods

Ad-hoc

Run jobs immediately for testing or one-time processing tasks.

Run with custom parameters

Run a job immediately with custom parameter values that override the default job parameters for this specific run only. This is useful for scenarios like data backfills where you might need to temporarily increase compute resources without modifying the original job definition.

Workflow orchestration

Include jobs in Workflows for automated, scheduled run as part of data pipelines.

API/AWS CLI

Run jobs programmatically using AWS CLI or SDK for automation scenarios.

> **Note**
>
> SageMaker Unified Studio jobs use AWS Glue APIs. For more information, see AWS Glue API for Jobs.

How to run a job

You can run jobs in SageMaker Unified Studio using several methods, depending on your requirements.

To run a job:

1. Navigate to the **Data processing jobs** section in your SageMaker Unified Studio project.
2. Select the job you want to run.
3. Choose the **Run** button in the action bar.
4. The job will immediately execute with its default parameters.

To run a job with custom parameters:

1. Select the job you want to run.
2. Choose **Run with custom parameters** from the actions menu.
3. Modify any parameters you need to change for this specific run.
4. Choose **Run**.

This method is useful for scenarios like data backfills where you might need to temporarily increase compute resources without modifying the original job definition. The custom parameters will only apply to this specific run and will not affect the default parameters set on the job.

To schedule a job

1. Navigate to the **Data processing jobs** section in your SageMaker Unified Studio project.
2. Select the job you want to schedule.
3. Choose on the **Schedules** tab.
4. Choose **Add schedule**.

5. Set up a recurring schedule (daily, weekly, etc.) or use a cron expression.
6. Specify start time and timezone.
7. Choose **Save**.

The job will automatically run according to the schedule you set. This is ideal for jobs that need to run on a regular basis without manual intervention.

To run a job as part of a workflow

1. Navigate to the **Workflows** section in your SageMaker Unified Studio project.
2. Create a new workflow or edit an existing one.
3. Add your job to the workflow by selecting **Add node** and choosing **Data processing job**.
4. Select the job you want to include in the workflow.
5. Configure any workflow-specific settings for the job.
6. Save and run the workflow.

Running jobs as part of workflows allows you to orchestrate complex data pipelines with dependencies between different jobs.

To run a job programmatically

You can use the AWS CLI or SDK to trigger job runs programmatically. SageMaker Unified Studio jobs use AWS Glue APIs under the hood.

Example using AWS CLI:

```
aws glue start-job-run --job-name "my-sagemaker-job" --region us-west-2
```

For more information on available parameters and options, see the AWS Glue API for Jobs documentation.

Monitoring jobs and job runs

With Amazon SageMaker Unified Studio, you can monitor your jobs and job runs by tracking performance, costs, run history, and Amazon CloudWatch metrics.

Monitoring capabilities

- **Job Status and History** - View job run status, success rates, and historical performance.
- **Resource Utilization** - Monitor compute resource consumption and optimization opportunities.
- **Cost Tracking** - Track run costs and resource usage for budgeting and optimization.
- **Logs and Metrics** - Access Amazon CloudWatch logs, metrics, and Spark UI for detailed troubleshooting.
- **Data Lineage** - Track data flow and dependencies across job runs.

SageMaker Unified Studio provides aggregate job-level metrics including:

- Job success rate.
- Average resource consumption (estimated in DPU hours).
- Average runtime.

For individual job runs, detailed metrics include:

- Status (success, failure, in progress).
- Resources consumed (estimated).
- Retry attempts.
- Duration.
- Settings used for the run.

CloudWatch logs (both driver and executor logs) are automatically pulled into the output log tab, eliminating the need to navigate to CloudWatch separately for troubleshooting.

Job runs view

> **Note**
>
> Job run history is accessible for 90 days for your workflow and job run.

The Job runs resource list shows the jobs for the specified date range and filters.

You can filter the jobs on additional criteria, such as status, job type, and the job name. In the filter box at the top of the table, you can enter the text to use as a filter. The table results are updated with rows that contain matching text as you enter the text.

The Job runs resource list displays the details for the job runs. You can sort the rows in the table by choosing a column heading. The table contains the following information:

Job name

The name of the job.

Type

The type of job environment:

AWS Glue ETL: Runs in an Apache Spark environment managed by AWS Glue.

Start time

The date and time at which this job run was started.

End time

The date and time that this job run completed.

Run status

The current state of the job run. Values can be:

STARTING

RUNNING

STOPPING

STOPPED

SUCCEEDED

WAITING

FAILED

TIMEOUT

Run time

The amount of time that the job run consumed resources.

Capacity

The number of data processing units (DPUs) that were allocated for this job run.

Instance type

The type of predefined instance that was allocated when the job ran. Values can be G.1X, G.2X, G.4X or G.8X.

DPU hours

The estimated number of DPUs used for the job run. A DPU is a relative measure of processing power. DPUs are used to determine the cost of running your job.

You can choose any job run in the list and view additional information. Choose a job run, and then do one of the following:

- Choose the Actions menu and the View job option to view the job in the visual editor.
- Choose the Actions menu and the Stop run option to stop the current run of the job.
- Choose the View CloudWatch logs button to view the job run logs for that job.
- Choose View details to view the job run details page.

To access job monitoring

1. Navigate to the **Data processing jobs** section in your SageMaker Unified Studio project.
2. Select the job you want to monitor.
3. View aggregate metrics including:
   - Job success rate.
   - Average run time.
   - Resource consumption.
4. Expand individual job runs to view detailed logs, metrics, and performance data.

Viewing the details of a job run

The Run details page provides information about the job run, settings, output logs, and run parameters.

The information displayed on the job run detail page includes:

Job name

The name of the job.

Run Status

The current state of the job run. Values can be:

STARTING

RUNNING

STOPPING

STOPPED

SUCCEEDED

WAITING

FAILED

TIMEOUT

Attempt

The number of automatic retry attempts for this job run.

Start time

The date and time at which this job run was started.

End time

The date and time that this job run completed.

Duration

The amount of time spent preparing to run the job.

DPU hours consumed

The number of AWS Glue data processing units (DPUs) that were allocated for this job run.

Max capacity

The maximum capacity available to the job run.

You can also view the job run settings, which show you the compute type, engine version, instance type, capacity (DPUs), and timeout.

To view output logs and run parameters:

- In the Job runs table, choose a job run. The Output logs are visible below the Job run settings.

  You can:

  - Copy log by choosing **Copy logs**.
  - Download log by choosing **Download as log file**.
  - Search the log by entering a search query. You can then filter on the results of your search and sort results.
- In the Job run details page, choose the **Run parameters** tab.

Viewing CloudWatch metrics for AWS Glue jobs

Amazon SageMaker Unified Studio displays Amazon CloudWatch metrics for your AWS Glue jobs directly within the job run detail view. The **Metrics** tab displays key performance indicators. These indicators include data movement size, memory profiles for drivers and executors, and CPU load.

To view CloudWatch metrics for a AWS Glue job run

1. In the left navigation pane, choose **Jobs**.
2. Select the AWS Glue job that you want to monitor.
3. Choose the **Runs** tab to view the list of previous job runs.
4. Select a previous run to open the run detail view.
5. In the run detail view, choose the **Metrics** tab.

The **Metrics** tab displays the following performance indicators for the selected job run:

- **Data movement size** — The volume of data read and written during the job run.
- **Memory profile for drivers and executors** — Memory utilization over time for the Spark driver and each executor. Use this metric to identify memory pressure or out-of-memory conditions.
- **CPU load** — CPU utilization across the driver and executors. Use this metric to identify compute bottlenecks or underutilized resources.

You can use these metrics alongside job logs in the same view. Correlate resource utilization patterns with log output to diagnose performance issues without leaving Amazon SageMaker Unified Studio.

> **Note**
>
> The **Metrics** tab is available for all previous AWS Glue job runs accessible from the **Runs** tab. CloudWatch metrics data is retained for 14 days. For job runs older than 14 days, metrics charts might not be available.

Limitations

- Jobs support AWS Glue ETL compute resource on AWS Glue 4.0, 5.0, and 5.1

Best Practices

- **Use appropriate authoring methods** - Choose IDE for familiar development tools, Visual ETL for ease of use, and Notebooks for interactive development.
- **Optimize compute resources** - Right-size your job configurations based on data volume and processing requirements.
- **Implement version control** - Use Git integration to track changes and collaborate effectively.
- **Monitor performance** - Regularly review job run metrics to identify optimization opportunities.
- **Test thoroughly** - Use debugging tools to test jobs before production deployment.
- **Organize with Workflows** - Group related jobs into Workflows for better orchestration and dependency management.

Next Steps

After setting up jobs in SageMaker Unified Studio, consider these next steps:

- Create Workflows to orchestrate multiple jobs and define dependencies.
- Set up monitoring and alerting for critical job runs.
- Explore advanced features like data lineage tracking and cost optimization.

- Integrate jobs with your existing CI/CD pipelines for automated deployment. For more information, see *CI/CD*.

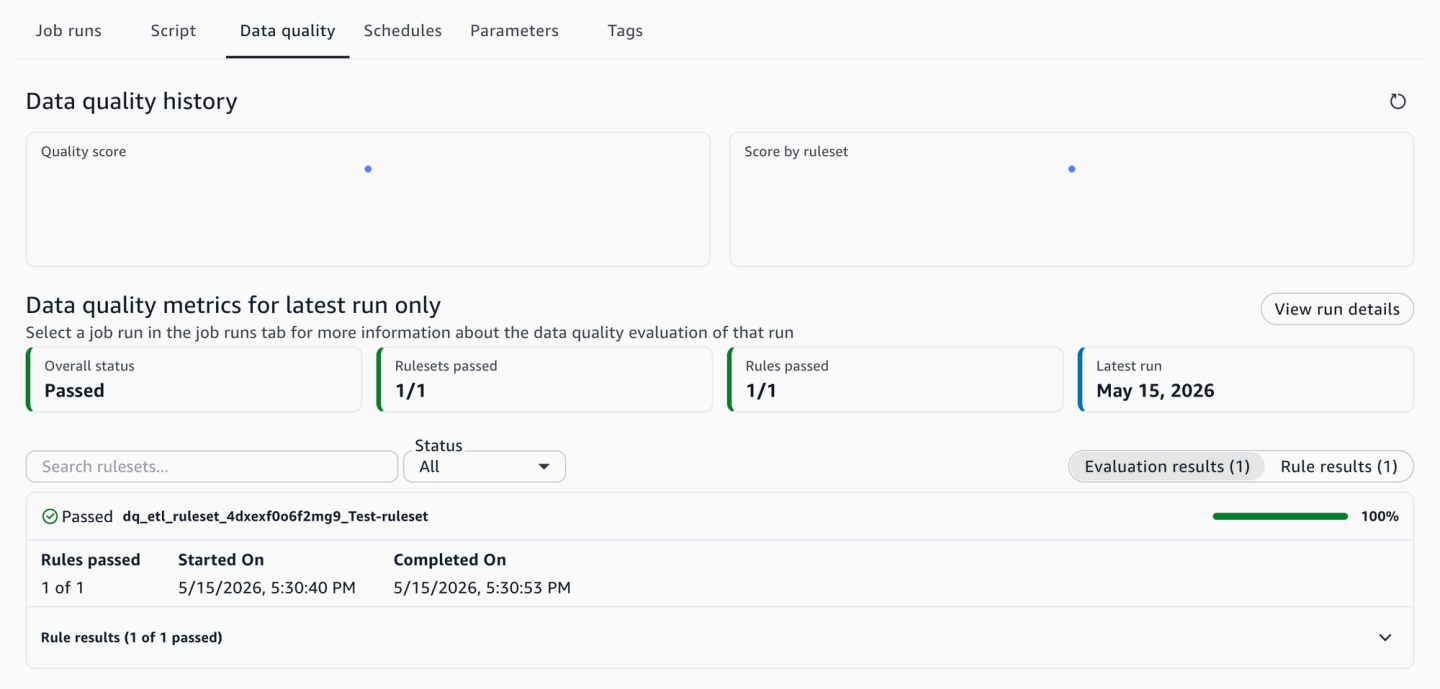
Monitor data quality in data processing jobs

When a Visual ETL job includes an Evaluate Data Quality transform, the data quality results are captured and displayed on the data processing jobs page. This gives you a centralized view of quality checks that ran as part of your data integration pipeline.

Rules are written using DQDL (Data Quality Definition Language), a domain-specific language for defining data quality rules, with 31 built-in rule types. For details on configuring the Evaluate Data Quality transform, defining rules, and selecting outputs, see the section called "Evaluate data quality transform". For the full list of rule types and syntax, see DQDL rule types in the AWS Glue documentation.

View data quality results in data processing jobs

After a job completes, navigate to **Data processing jobs** in the left navigation pane and select your job. Choose the **Data quality** tab on the job detail page.



The **Data quality** tab shows the following information.

Field	Description
Rule set name	The name configured in the Evaluate Data Quality transform node.

Field	Description
Rules evaluated	Total rules that ran.
Pass/fail status	Which rules passed and which failed.
Quality score	Percentage of rules that passed.

> **Tip**
>
> If your job has multiple Evaluate Data Quality nodes, each appears with its own rule set name. Use descriptive names when configuring the nodes to identify which check produced which results.

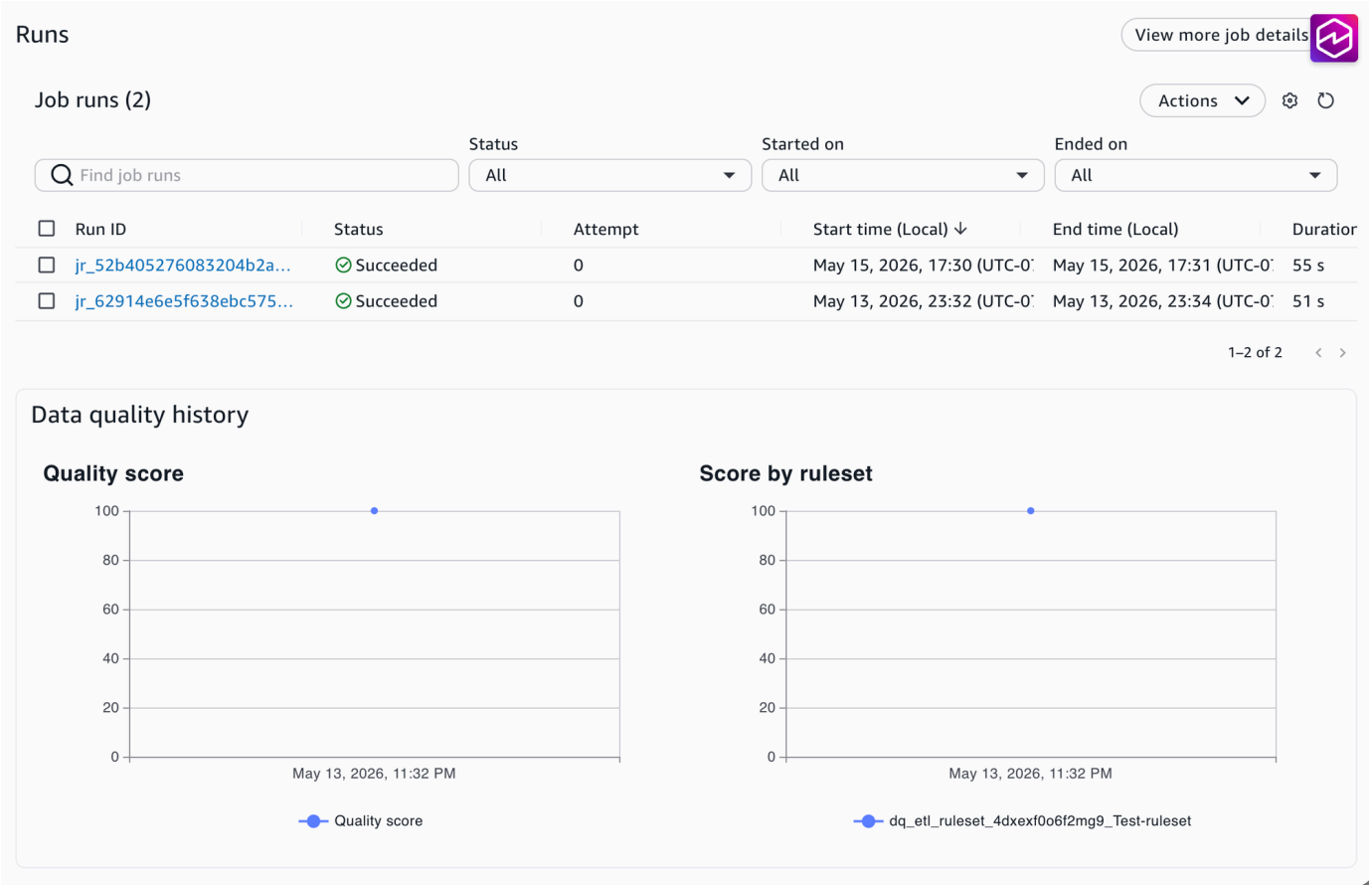
You can also run a job directly from this page. Only one run can execute at a time.

View data quality results in Visual ETL run details

You can also view data quality results directly within the Visual ETL flow after a run completes.

To view data quality results in Visual ETL

1. Open your Visual ETL job and navigate to **View runs** for a completed run.



2. The data quality section displays the same results as the data processing jobs page: rule set names, rules evaluated, and pass/fail status for each rule.
3. Choose **View logs** to see detailed execution logs for the data quality evaluation.

This view is useful when you are already working in the Visual ETL editor and want to check results without navigating away to the data processing jobs page.

Each job run captures a new set of data quality results. Use the run history to compare quality across runs and identify trends or regressions. Select any run to drill into specific rule outcomes for that execution.

How in-transit and at-rest evaluations differ

The following table compares data quality evaluations that run in transit (during data processing jobs) with evaluations that run at rest (against data catalog tables).

Aspect	In transit (data processing jobs)	At rest (data catalog)
Where you define rules	Evaluate Data Quality transform in Visual ETL	Data quality tab on a catalog table
When rules run	During job execution, as data flows through the pipeline	On demand or on a schedule, against stored table data
Where you see results	Data processing jobs > Data quality tab, or Visual ETL View runs	Catalog > Table > Data quality tab
Best for	Validating data during transform ation before writing to a target	Monitoring quality of stored data over time

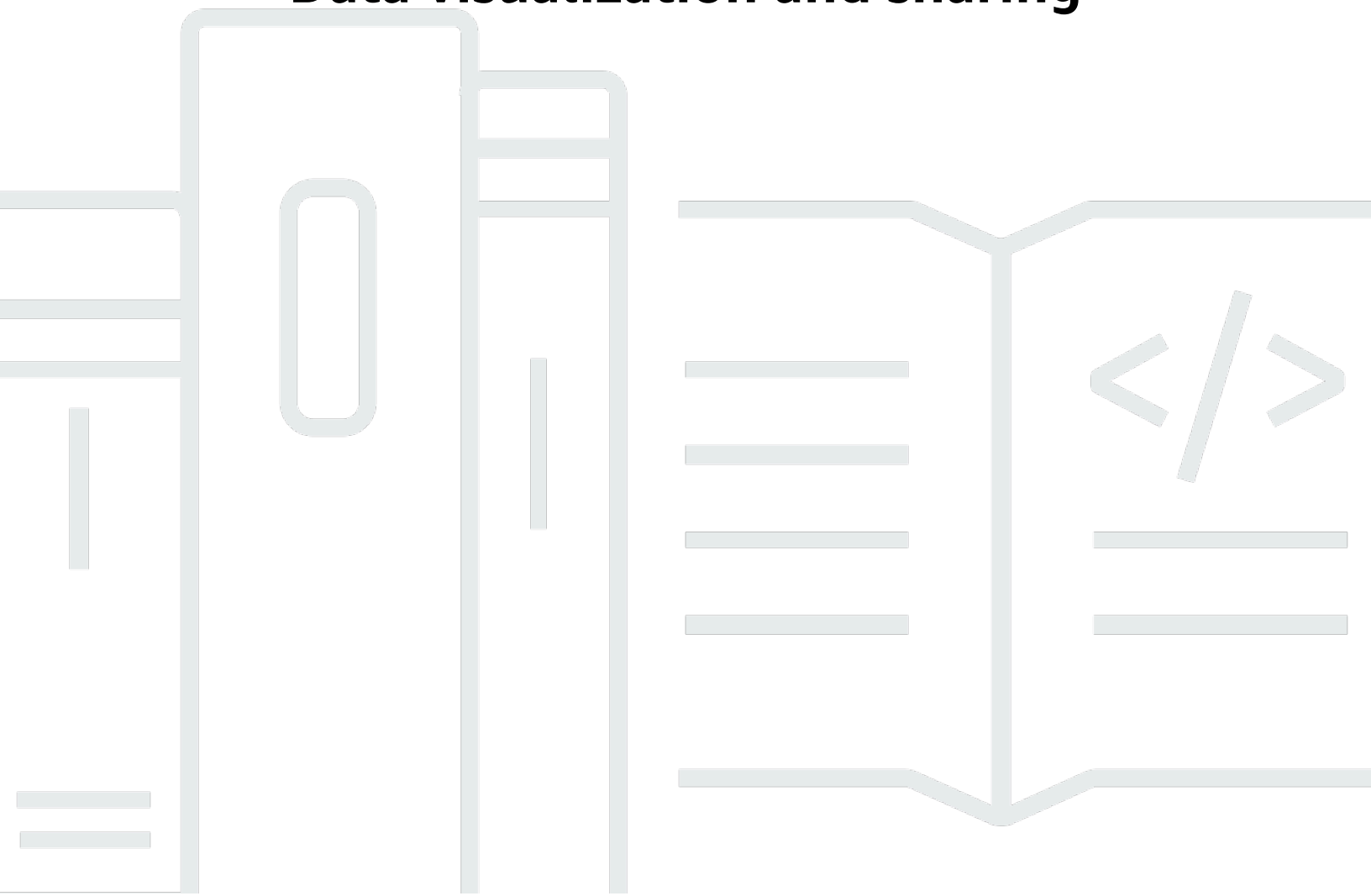
For details on evaluating data quality at rest, see *Data quality in the catalog*.

aws

User Guide

Amazon SageMaker Unified Studio

Data visualization and sharing



Amazon Quick in Amazon SageMaker Unified Studio

Topics

- Overview of Amazon Quick in Amazon SageMaker Unified Studio
- Prerequisite for using the Amazon Quick in Amazon SageMaker Unified Studio feaure
- Getting started using Amazon Quick integration
- Share Amazon Quick dashboards

Overview of Amazon Quick in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio integrates with Amazon Quick, enabling direct launch of Quick from within Amazon SageMaker Unified Studio for seamless data visualization and dashboard publishing. This integration allows users to explore and visualize data with a single click, while maintaining security and governance.

Prerequisite for using the Amazon Quick in Amazon SageMaker Unified Studio feaure

To use the Amazon Quick in Amazon SageMaker Unified Studio feature, you need a valid Amazon Quick license to *create* datasets and visualize project data, you need a **Author** or **Admin** license. If you have a **Reader** license, you can only *view* dashboards.

Getting started using Amazon Quick integration

1. **Create a Amazon SageMaker Unified Studio project with Quick blueprint**

   You must sign in to Amazon SageMaker Unified Studio by selecting **Sign in with SSO** and create a new project. For more information, see Create a new project.

   > **Note**
   >
   > When creating the project, you must choose a project profile which has **Quick** blueprint added to it.

- **Automatic resource provisioning**

  When your project is created, the following resources are provisioned:

  - One restricted Quick folder per project. All new assets—like analyses, datasets, and dashboards—created using the project's data are scoped to this folder and cannot be moved outside of it. Only Amazon SageMaker Unified Studio, not end users, can grant access to this folder, enforcing centralized, policy-based control.
  - Real-time folder permission sync. Amazon SageMaker Unified Studio keeps the Amazon Quick folder's access permissions aligned with project membership. Whenever a user or group is added to or removed from the Amazon SageMaker project, those changes are reflected in the corresponding Amazon Quick folder.
  - Preconfigured Quick data sources for Athena and Amazon Redshift:
    - If the project profile includes the *LakehouseDatabase* blueprint, an Athena data source is created.
    - If the project profile includes the *RedshiftServerless* blueprint, a Redshift data source is provisioned.

  > **Note**
  >
  > You can also enable Amazon QuickSight integration for existing projects in Amazon SageMaker Unified Studio. In the current release of Amazon SageMaker Unified Studio, this is only supported for projects that were created with a project profile where the QuickSight blueprint is added with the **On demand** deployment setting mode. For more information, see Amazon Quicksight in SageMaker Unified Studio. If this condition is met, you can then add the Amaon QuickSight environment to the existing project by invoking the CreateEnvironment API. For more information, see CreateEnvironment.

2. **Get access to data**

   Before visualizing data in Quick, your project must have access to the required data assets. You can get access in the following ways:

a. **Subscribe to data in Amazon SageMaker Catalog**

   If the data assets are available in the catalog, you can request access to available data assets directly within Amazon SageMaker Unified Studio. Once approved, Amazon SageMaker Unified Studio configures necessary permissions. For detailed steps, see Data discovery, subscription, and consumption.

b. **Grant access through project's IAM Role**

   Administrators grant access by assigning AWS Lake Formation permissions to the project's IAM role. This enables fine-grained control at database, table, and column level. For more information, see Granting data permissions using the named resource method.

c. **Create new connections**

   You can create connections directly to Amazon Redshift and other third party sources like Oracle and Snowflake from Amazon SageMaker Unified Studio. You configure connection details and credentials securely and manage them within the project. For detailed steps, see Amazon Redshift compute connections and Data connections in lakehouse architecture.

3. **Open in Quick**

Once your project has access to the necessary datasets, you can start building visualizations in Amazon Quick. Amazon SageMaker Unified Studio provides two integrated paths for creating Quick datasets and analyses:

a. **Create a dataset manually in Quick**

   1. From Project Overview, choose **Actions** and then choose **Open Quick**
   2. Choose from available data source(s) (Athena or Redshift)
   3. Create new dataset by connecting to data source tables or views or writing custom SQL query

   This approach gives you flexibility to define datasets as needed and combine data from different sources within your project. For detailed steps on creating a dataset from an existing data source, refer to Creating a Dataset Using an Existing Data Source.

b. **Create dataset from Data Explorer**

   1. In the left navigation pane, choose **Data**.

2. Select table or view to visualize
3. Choose **Visualize in Quick** under **Actions**

When you choose this option, Amazon SageMaker Unified Studio creates a Quick dataset using the selected table or view. The dataset is named after the table or view and is placed into the project's restricted folder in Quick. The dataset is immediately opened in a new browser tab, allowing you to begin building an analysis.

From more information on building an analysis using the newly created dataset, see Working with an Analysis in Quick. For more information on joining your dataset with other datasets available within the project folder, see Joining datasets in Quick.

4. **Publish Quick dashboards**

Once a dashboard is created, it is added to the project as an asset.

> **Note**
>
> Only dashboards that are created within the **Assets** sub-folder in Amazon QuickSight folder are added to the project.

To view the newly created dashboard assets:

1. Navigate to your project
2. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.
3. Find **Quick dashboard** listed as new asset

Dashboard assets include:

- AWS Account
- AWS Region
- Creation date
- Dashboard name
- Last updated

- Owner
- Quick account
- URL

For more information on editing asset metadata, see Updating asset metadata.

> **Note**
>
> Dashboard subscription is not supported at this time. The **Subscribe** button is inactive for dashboard assets.

Share Amazon Quick dashboards

Quick dashboards that have been published to the Amazon SageMaker Catalog can be shared with users and groups outside of the project enabling broader access while maintaining centralized governance and control.

> **Note**
>
> To share an Amazon Quick dashboard with an IAM Identity Center user or group, they must also be part of your Amazon SageMaker unified domain.

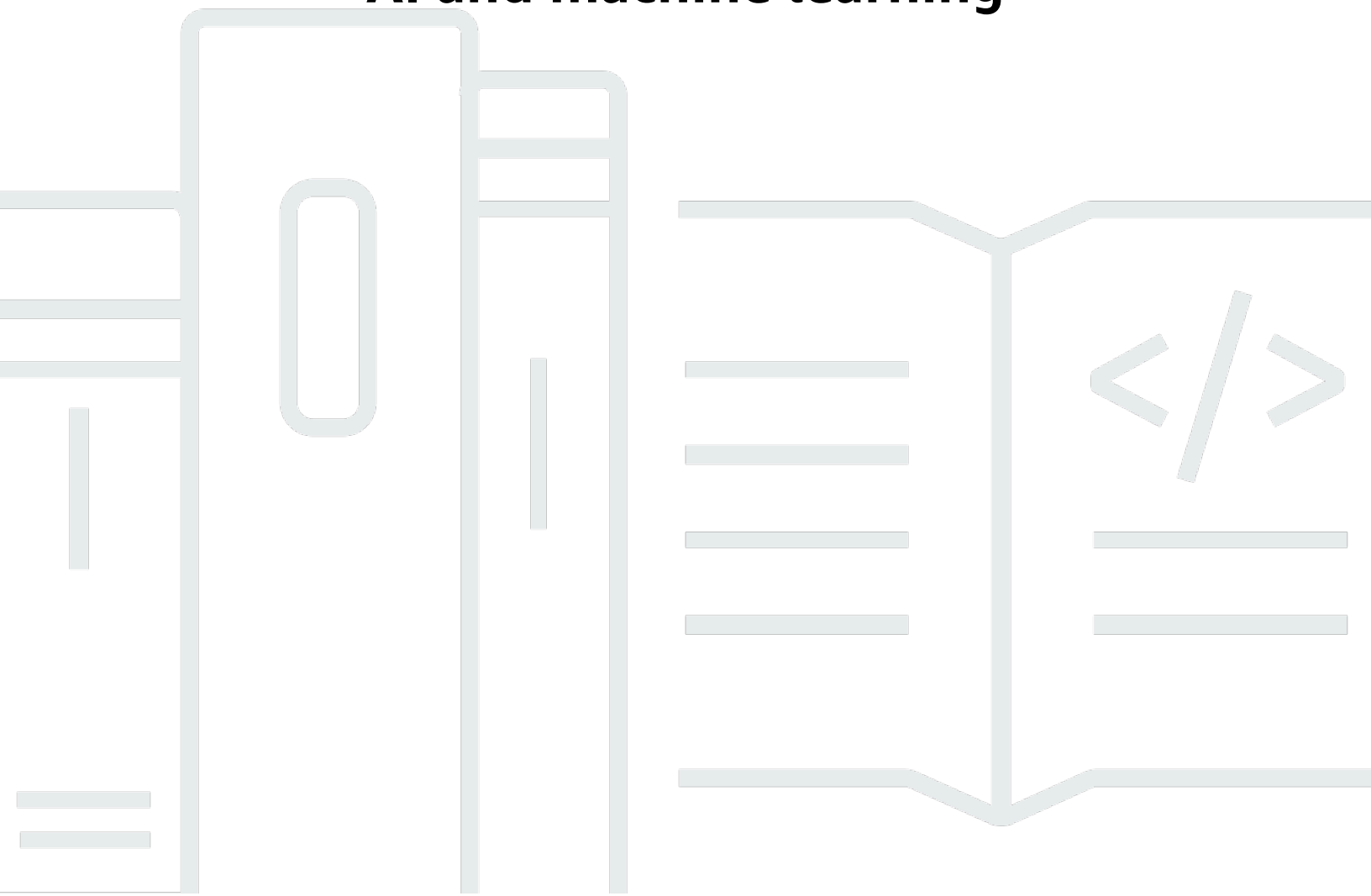
1. Go to your Amazon SageMaker project and choose **Assets** in the left navigation.
2. Find the dashboard asset under **Inventory** or **published** tab.
3. Open the dashboard in the Amazon SageMaker Catalog.
4. Choose **Share** under **Actions**.
5. Grant access to individual users or groups through **AWS IAM Identity Center (IDC)**.

aws

User Guide

Amazon SageMaker Unified Studio

AI and machine learning



Amazon Bedrock in SageMaker Unified Studio

With Amazon Bedrock in SageMaker Unified Studio you can build generative AI apps that use Amazon Bedrock models and features, such as knowledge bases and guardrails, without needing to write any code.

To use Amazon Bedrock in SageMaker Unified Studio, you must be a member of an Amazon SageMaker Unified Studio domain. Your organization will provide you with login details. Contact your administrator if you don't have your login details.

Your organization's administrator determines which Amazon Bedrock models and features you have access to. Contact your organization's administrator if you need access to a model or feature that you don't currently have access to.

> **Note**
>
> If you are administrator and need information about managing Amazon Bedrock in SageMaker Unified Studio, see Amazon Bedrock in SageMaker Unified Studio in the *Amazon SageMaker Unified Studio admin guide*.

Discover Amazon Bedrock in SageMaker Unified Studio playgrounds

Amazon Bedrock in SageMaker Unified Studio provides various options for discovering and experimenting with Amazon Bedrock models and apps.

With the model catalog you can find information about the Amazon Bedrock models that are available to you and decide which model is suitable for your use case. Different models have different capabilities and modalities. For more information, see Find serverless models with the Amazon Bedrock model catalog.

Amazon Bedrock in SageMaker Unified Studio offers two playgrounds for you to experiment with Amazon Bedrock models in: the chat playground and the image and video playground. With the chat playground, you can generate text responses from a model by sending text and image prompts. You can also interact with chat agent apps that have been shared with you. With the image and video playground, you can generate and edit images and videos by sending text and

image prompts to a suitable model. For more information, see Experiment with the Amazon Bedrock playgrounds.

Build generative AI apps

Within an Amazon SageMaker Unified Studio project, you can create two types of generative AI apps: a chat agent app and a flow app. You can use a chat agent app to chat with an Amazon Bedrock model through a conversational interface, typically by sending prompts (text or image) and receiving responses. You can use a flows ap to link prompts, supported Amazon Bedrock models, and other units of work, such as a knowledge base, together and create generative AI workflows.

Apps that you create with Amazon Bedrock in SageMaker Unified Studio can integrate the following Amazon Bedrock features.

- **Data sources** — Enrich apps by including context that is received from querying a knowledge base or a document.
- **Guardrails** — Implement safeguards for your Amazon Bedrock in SageMaker Unified Studio app based on your use cases and responsible AI policies.
- **Functions** — Call a function with a model to access a specific capability when handling a prompt.
- **Prompts** — Access reusable prompts that you can use in a flow app.

Within a project, you can use the *asset gallery* to organize the prompts and components that you use for an app. A component is an Amazon Bedrock knowledge base, guardrail, or function.

A critical part of creating a generative AI app is deciding which model to use and which model settings to use. To help you decide, you can evaluate a model for different task types.

If you work on a team, you can collaborate by sharing an app with other team members. You can also export an app so that you can use the app in your own environment.

You can clone the repository that holds your Amazon SageMaker Unified Studio project files to your computer. However, we don't recommend making changes to your project's files on your local desktop, as this may break your project's apps and components.

Find serverless models with the Amazon Bedrock model catalog

The Amazon Bedrock in SageMaker Unified Studio model catalog is where you can find the serverless Amazon Bedrock foundation models that you have access to. You can group models by their modality or by their provider. The modality of a model represents the type of input data that the model is trained on and is able to process, such as text or image data. For more information, see Supported foundation models in Amazon Bedrock.

If you can't find a specific model, ask your administrator if you have permissions to access the model.

To find out information about a model, such as supported use cases, select the model tile in the model catalog. When choosing a model, consider the following:

- Amazon Bedrock models support differing inference parameters and capabilities. For more information, see Amazon Bedrock foundation model information in the *Amazon Bedrock user guide*.
- Amazon Bedrock in SageMaker Unified Studio supports Amazon Bedrock foundation models with on-demand throughput and cross-region inference.

  Models that support cross-region inference throughput can increase throughput and improve resiliency by sending requests to different AWS Regions during peak utilization bursts. In the model catalog (and the model selector in app configuration), the text *Cross-region* identifies such a model.
- Amazon Bedrock in SageMaker Unified Studio doesn't support Provisioned throughput, custom models or imported models.

If the model is suitable for your needs, you can choose the menu button to start using the model. Depending on the model, you can choose from the following actions:

- Build chat agent app – Create an app in which users can chat with a model.
- Build flow app – Visually create the workflow for an app.
- Build prompt – Create resuable prompts for use in a flow app.
- Evaluate model – Evaluate the performance of a model for your use case.

The following procedure shows how to open the model catalog from the Amazon Bedrock in SageMaker Unified Studio playground. You can also access the model catalog from your projects.

Your administrator might give you access to different models in your projects. To check the models that you can access in a project, open or create a project, and then select **Models** in the navigation pane to open the model catalog.

To open the model catalog in the playground

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **Playground**.
4. Choose **Models**. The Amazon Bedrock in SageMaker Unified Studio playground opens at the model catalog.
5. (Optional) Choose **Group by: Modality** and select **Provider** to sort the list by model provider.
6. Choose a model to get information about the model.
7. If you're ready to build with the app, choose **Action** and select the appropriate action. You can also choose an action from the model tile on the model catalog page.
8. Choose **Amazon Bedrock model catalog** to go back to the model catalog page.

What is a prompt?

A prompt is the input that you send to a model in order for it to generate a response. For example, you could send the following user prompt to a model: **`What is Avebury stone circle?`**.

This prompt would likely generate a response similar to the following:

```
Avebury stone circle is a Neolithic monument located in Wiltshire, England.
It consists of a massive circular bank and ditch, with a large outer circle of standing
 stones
that originally numbered around 100.
```

Some models support *multimodal* prompts, which are prompts that support different types of media input, such as text, images, or video. For example, you could send an image to a model and ask it to describe what the image contains. Not all models support multimodal prompts and modality support varies by model. For information on how to best create prompts for a specific model, see Prompt engineering guides.

In addition to user prompts, Amazon Bedrock in SageMaker Unified Studio also supports inference parameters and system instructions, which allow you to customize and influence model behavior. The following sections provide information and guidance on how to use inference parameters and sytem prompts.

Topics

- Inference parameters
- System instructions
- Prompt engineering guides

Inference parameters

Inference parameters are values that you can adjust to influence how a model generates a response to a prompt. For example, in the chat agent app you create in Build a chat agent app with Amazon Bedrock, you can use inference parameters to adjust the randomness and diversity of the songs that the model generates for a playlist.

You can apply inference parameters to models you use in the Amazon Bedrock playgrounds, chat agent apps, and flow apps.

Randomness and diversity

For any given sequence, a model determines a probability distribution of options for the next token in the sequence. To generate each token in an output, the model samples from this distribution. Randomness and diversity refer to the amount of variation in a model's response. You can control these factors by limiting or adjusting the distribution. Foundation models typically support the following parameters to control randomness and diversity in the response.

- **Temperature**– Affects the shape of the probability distribution for the predicted output and influences the likelihood of the model selecting lower-probability outputs.
  - Choose a lower value to influence the model to select higher-probability outputs.
  - Choose a higher value to influence the model to select lower-probability outputs.

  In technical terms, the temperature modulates the probability mass function for the next token. A lower temperature steepens the function and leads to more deterministic responses, and a higher temperature flattens the function and leads to more random responses.
- **Top K** – The number of most-likely candidates that the model considers for the next token.

- Choose a lower value to decrease the size of the pool and limit the options to more likely outputs.
- Choose a higher value to increase the size of the pool and allow the model to consider less likely outputs.

For example, if you choose a value of 50 for Top K, the model selects from 50 of the most probable tokens that could be next in the sequence.

- **Top P** – The percentage of most-likely candidates that the model considers for the next token.
  - Choose a lower value to decrease the size of the pool and limit the options to more likely outputs.
  - Choose a higher value to increase the size of the pool and allow the model to consider less likely outputs.

  In technical terms, the model computes the cumulative probability distribution for the set of responses and considers only the top P% of the distribution.

  For example, if you choose a value of 0.8 for Top P, the model selects from the top 80% of the probability distribution of tokens that could be next in the sequence.

The following table summarizes the effects of these parameters.

Parameter	Effect of lower value	Effect of higher value
Temperature	Increase likelihood of higher-probability tokens Decrease likelihood of lower-probability tokens	Increase likelihood of lower-probability tokens Decrease likelihood of higher-probability tokens
Top K	Remove lower-probability tokens	Allow lower-probability tokens
Top P	Remove lower-probability tokens	Allow lower-probability tokens

As an example to understand these parameters, consider the example prompt **`I hear the hoof beats of "`**. Let's say that the model determines the following three words to be candidates for the next token. The model also assigns a probability for each word.

```
{
    "horses": 0.7,
    "zebras": 0.2,
    "unicorns": 0.1
}
```

- If you set a high **temperature**, the probability distribution is flattened and the probabilities become less different, which would increase the probability of choosing "unicorns" and decrease the probability of choosing "horses".
- If you set **Top K** as 2, the model only considers the top 2 most likely candidates: "horses" and "zebras."
- If you set **Top P** as 0.7, the model only considers "horses" because it is the only candidate that lies in the top 70% of the probability distribution. If you set **Top P** as 0.9, the model considers "horses" and "zebras" as they lie in the top 90% of probability distribution.

System instructions

A system instruction an overarching initial guideline that defines how a model should behave in future interactions. System instructions provide context to the model about the task it should perform or the persona it should adopt during the conversation.

For example, you could use a system instruction to specify that the model should behave as an app that creates playlists for a radio station that plays rock and pop music. You can then use the model to create playlists of rock and pop songs based on different themes, such as songs that are related by artist

You can apply system instructions to models you use in the Amazon Bedrock playgrounds, chat agent apps, and flow apps.

Prompt engineering guides

Amazon Bedrock in SageMaker Unified Studio provides models from a variety of model providers. Each provider provides guidance on how to best create prompt for their models.

- **Amazon Nova user guide:** https://docs.aws.amazon.com/nova/latest/userguide/what-is-nova.html
- **Anthropic Claude model prompt guide:** https://docs.anthropic.com/claude/docs
- **Anthropic Claude prompt engineering resources:** https://docs.anthropic.com/claude/docs/guide-to-anthropics-prompt-engineering-resources
- **Cohere prompt guide:** https://txt.cohere.com/how-to-train-your-pet-llm-prompt-engineering
- **AI21 Labs Jurassic model prompt guide:** https://docs.ai21.com/docs/prompt-engineering
- **Meta Llama 2 prompt guide:** https://ai.meta.com/llama/get-started/#prompting
- **Stability documentation:** https://platform.stability.ai/docs/getting-started
- **Mistral AI prompt guide:** https://docs.mistral.ai/guides/prompting_capabilities/

For general guidelines about creating prompts with Amazon Bedrock, see General guidelines for Amazon Bedrock LLM users.

Experiment with the Amazon Bedrock playgrounds

An Amazon Bedrock in SageMaker Unified Studio playground lets you experiment with Amazon Bedrock foundation models, so that you can choose the right model for your use case. You can also experiment with chat agent apps that others share with you. Amazon Bedrock in SageMaker Unified Studio provides the following playgrounds:

- **Chat playground** – Chat with an Amazon Bedrock model by sending prompts to the model and answering the response that the model generates. You can also experiment with chat agent apps that are shared with you.
- **Image and video playground** – Generate images and videos with a model. You can use prompts, images, and videos to describe the content you want to generate.

Each playground lets you choose a model and experiment with settings, such as the inference parameters that affect the output that the model generates. To help you experiment, you can compare the output of multiple models and chat agent apps.

If you want to know more about a model, use the model catalog to find information such as supported use cases and model attributes. For more information, see Find serverless models with the Amazon Bedrock model catalog.

> **Warning**
>
> Generative AI may give inaccurate responses. Avoid sharing sensitive information. Chats may be visible to others in your organization.

Other Amazon Bedrock in SageMaker Unified Studio users can share apps and prompts so that you can experiment with them in a playground. For more information, see Access shared generative AI assets in an Amazon Bedrock playground.

After you familiarize yourself with a model in a playground, you can try creating your own Amazon Bedrock in SageMaker Unified Studio app, such as a chat agent app.

Topics

- Chat with a model in the Amazon Bedrock chat playground
- Chat with an app in the Amazon Bedrock chat playground
- Generate an image with the Amazon Bedrock image and video playground
- Generate a video with the Amazon Bedrock image and video playground
- Access shared generative AI assets in an Amazon Bedrock playground

Chat with a model in the Amazon Bedrock chat playground

The Amazon Bedrock in SageMaker Unified Studio chat playground allows you chat with an Amazon Bedrock model and try chat agent apps that are shared to you. A chat provides a back-and-forth, dialogue-like interaction between you and an Amazon Bedrock model. The model is able to retain context during a chat allowing for coherent and relevant responses from the model. You chat with a model by sending a prompt to the model and by receiving the response that the model generates. You continue the chat by sending further prompts.

If a model supports multimodal prompts, you can send prompts that contain text and images. A chat can contain multiple text and image prompts. After you finish a chat, you can reset the playground to begin a new chat. Amazon Bedrock models support differing modalities. For more information, see Supported foundation models in Amazon Bedrock.

The maximum image file size is 5MB. You can upload images that are in JPG, PNG, GIF, and WebP format.

When you run a prompt in the chat playground, you get the following information about the request:

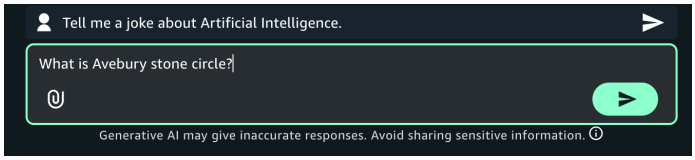
- **Input tokens** — The number of input tokens used by the foundation model during inference.
- **Output tokens** — The number of tokens generated in a response by the foundation model.
- **Latency** — The amount of time the foundation model uses to generate each token in a sequence, based on the on-demand throughput.

The chat playground provides quick start prompts that illustrate the kinds of prompt that you can send to a model.

Optionally, you can compare the outputs from up to 3 shared apps and models. You can make configuration changes for models, such as inference parameters and system instructions and compare the results. You can't make configuration changes for shared apps.

To chat with a model

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **Playground**.
4. Choose **Chat** to open the chat playground.
5. In **Type** select **Model** and then select a model to use in **Model**. For full information about the model, choose **View full model details** in the information panel. For more information, see Find serverless models with the Amazon Bedrock model catalog. If you don't have access to an appropriate model, contact your administrator. Different models might not support all features.
6. In the **Enter prompt** text box, enter **`What is Avebury stone circle?`**.
7. (Optional) If the model you chose is a reasoning model, you can choose **Reason** to have the model include its reasoning in the reponse. For more information, see Enhance model responses with model reasoning in the *Amazon Bedrock user guide*.
8. Press Enter on your keyboard, or choose the run button, to send the prompt to the model. Amazon Bedrock in SageMaker Unified Studio shows the response from the model in the playground.



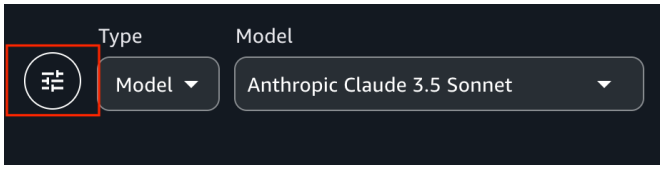
9. Continue the chat by entering the prompt **`Is there a museum there?`** and pressing Enter.

   The response shows how the model uses the previous prompt as context for generating its next response.

10. Choose **Reset** to start a new chat with the model.
11. Influence the model response by doing the following:

    a. Enter and run a prompt. Note the response from the model.

    b. Choose the configurations menu to open the **Configurations** pane.

    c. Influence the model response by making inference parameters changes.

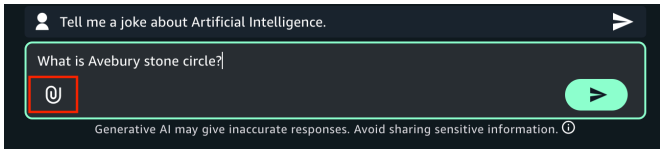
    d. (Optional) In **System instructions**, enter any overarching system instructions that you want the model to apply for future interactions.

    e. Run the prompt again and compare the response with the previous response.

12. Choose **Reset** to start a new chat with the model.
13. Try sending an image to a model by doing the following:

    a. For **Model**, choose a model that supports images.

    b. Choose the attachment button at the left of the **Enter prompt** text box.

    c. In the open file dialog box, choose an image from your local computer.

    d. In the text box, next to the image that you uploaded, enter **`What's in this image?`**.

    e. Press Enter on your keyboard enter to send the prompt to the model. The response from the models describes the model or image.

14. (Optional) Try using another model and different prompts. Different models have different recommendations for creating, or engineering, prompts. For more information, see Prompt engineering guides.
15. (Optional) Compare the output from multiple models, or shared apps.

    a. In the playground, turn on **Compare mode**.
    b. In both panes, select the model that you want to compare. If you want to use a shared app, select **App** in **Type** and then select the app in **App**.
    c. Enter a prompt in the text box and run the prompt. The output from each model is shown. You can choose the copy icon to copy the prompt or model response to the clipboard.
    d. (Optional) Choose **View configs** to make configuration changes, such as inference parameters. Choose **View chats** to return to the chat page.
    e. (Optional) Choose **Add chat window** to add a third window. You can compare up to 3 models or apps.
    f. Turn off **Compare mode** to stop comparing models.

Now that you are familiar with the explorer playground, try creating a Amazon Bedrock in SageMaker Unified Studio app next. For more information, see Build a chat agent app with Amazon Bedrock.

Chat with an app in the Amazon Bedrock chat playground

You can use the chat playground to experiment with chat agent apps that are shared to you. When you open a shared app, you can send prompts to the app and see the response. You can't make changes to the shared app.

Optionally, you can compare the outputs from to 3 shared apps and models. You can view the configuration for a shared app, but you can't make configuration changes.

To learn how to share apps that you create, see Share an Amazon Bedrock chat agent app.

To chat with a shared app

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.

3. In the left navigation pane, under **Generative AI**, choose **Playground**.
4. Choose **Chat** to open the chat playground.
5. In **Type** select **App** and then select an app to use in **App**.
6. In the **Enter prompt** text box at the bottom of the page, enter the prompt that you want to use. If the app builder changes the default text for the text box, the text is different.
7. Press Enter on your keyboard enter to send the prompt to the mode.
8. (Optional) Compare the output from multiple apps, or models.

    a. In the playground, turn on **Compare mode**.

    b. In both panes, select the app that you want to compare.

    c. Enter a prompt in the text box and run the prompt.

    d. (Optional) Choose **View configs** to view the app configurations, such as inference parameters. Choose **View chats** to return to the chat page.

    e. (Optional) Choose **Add chat window** to add a third window. You can compare up to 3 models or apps.

    f. Turn off **Compare mode** to stop comparing models.

Generate an image with the Amazon Bedrock image and video playground

The image and video playground is an interactive environment that lets you specify actions that generate and manipulate images using natural language prompts, reference images, and suitable Amazon Bedrock models.

Actions for generating images

Within the image playground, you use an *action* to specify the image generation task that you want the model to do, such as replacing the background of an existing image. The actions that are available depends on the model you use.

- **Generate image** — Generates a new image from a prompt that you enter.
- **Generate variations** — Use a prompt to generate a variation of an existing image.
- **Remove object** — Removes an object from an image you supply.
- **Replace background** — Replaces the background of an image with a new background.

- **Replace object** — Replaces an object in an image with a different object.
- **Edit image sandbox** — An image sandbox that you can use to expiriment with Stable Diffusion XL models.

Some actions, such as generate variation, require a reference image that a model uses to generate a new image. An action might require you to use a mask tool to draw a bounding box around an area of the reference image, such as when you define an object that you want to remove with the remove object action.

Configuration options

You can influence how a model generates an image by configuring the following options. The configuration changes you can make depends on the action you choose.

Negative prompt

A set of words or phrases that that tells the model what not to include in the image that it generates. For example, you can use the term *-lowres* to avoid generating low-resolution or blurry images.

Reference image

In certain actions, such as generate variations or replace background, you specify a reference image that the model uses to process the action.

Response image

You can specify the image dimensions, orientation, and number of images to generate.

Advanced configuration options

You can make advanced configuration changes that how the model generates images. All models image generation models support the following:

- **Prompt strength** — Prompt strength is a numerical value that determines how strongly a model should adhere to the given text prompt. A higher prompt strength means the model will try to closely follow and prioritize the text description provided in the prompt when generating the image. Lower prompt strengths allow the model more creative freedom to deviate from the prompt.

- **Seed** — A seed is numeric value that a model uses to seed a random number generator. The model uses the seed as a starting point for creating random patterns during image generation. This initial randomness influences things like the exact positioning, colors, textures, and compositions present in the image that the model generates.
- **Similarity strength** — If you use the *Generate variations* action with a Titan Image Generator G1 V1 or a Titan Image Generator G1 V2 model, you can also configure the *Similarity Strength* advanced configuration. Similarity Strength specifies how similar the generated image should be to the input image. Use a lower value to introduce more randomness into the generated image.
- **Generate step** — If you use a Stable Diffusion XL model, you can configure the *Generate step* advanced configuration. Generate step determines how many times the image is sampled. More steps can result in a more accurate result.

Generate an image

The following procedure shows you how to use a model to generate an image. You can set various configurations such as the number of images to generate and how strongly the prompt affects the generation of the image. For more information, see Configuration options.

To generate an image in the image playground

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **Playground**.
4. Choose **Image and video**.
5. If the **Configurations** pane isn't open, choose the configuration button.
6. For **Model** select a model to use.
7. For **Action** choose the action **Generate image**.
8. In **Response image** do the following:

    a. For **Number of images** select the number of images that you want the model to generate. Not all models support changing this value.

    b. For **Orientation**, choose the orientation (landscape or portrait) for the images that the model generates.

c. For **Size**, select the size, in pixels, of the images that the model generates.

9. (Optional) In **Advanced configurations**, change how the model generates images by making advanced configuration changes. For more information, see Advanced configuration options.
10. In the **Enter prompt** text box, enter **`Create a photo of a local classic rock band playing on an outdoor stage.`**. Alternatively, enter a prompt of your choosing.
11. Press Enter on your keyboard to start the action. Amazon Bedrock in SageMaker Unified Studio shows the image that the model generates in the playground.
12. (Optional) See how different configuration parameters affect image generation by repeating steps 9 - 11 with different values.

Generate a variation of an image

The following procedure shows you how to generate a variation of a reference image that you supply. You can set various configurations such as the number of images to generate and how strongly the prompt affects the generation of the image. For more information, see Configuration options.

To generate a variation of an image

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **Playground**.
4. Choose **Image and video**.
5. If the **Configurations** pane isn't open, choose the configuration button.
6. For **Model** select a model to use.
7. For **Action** choose **Generate variations**.
8. (Optional) For **Negative prompt** enter text that describes content or concepts that you do not want the model to include in the image.
9. In **Reference image** choose **Upload image** and upload the image that you want the model to use with the action.
10. In **Response image** do the following:

a. For **Number of images** select the number of images that you want the model to generate. Not all models support changing this value.

b. For **Orientation**, choose the orientation (landscape or portrait) for the images that the model generates.

c. For **Size**, select the size, in pixels, of the images that the model generates.

11. (Optional) In **Advanced configurations**, change how the model generates images by making advanced configuration changes. For more information, see Advanced configuration options.
12. In the **Enter prompt** text box, enter the prompt that describes the image that you want the model to generate.
13. Press Enter on your keyboard to start the action. Amazon Bedrock in SageMaker Unified Studio shows the image that the model generates in the playground.

Remove an object from an image

The following procedure shows you how to use a model to remove an object from an image that you supply. For example, you could remove an unwanted person from an image. You can set various configurations such as the number of images to generate and how strongly the prompt affects the generation of the image. For more information, see Configuration options.

> **Note**
>
> The object removal action is only available with Titan Image Generator G1 V1 and Titan Image Generator G1 V2 models.

To remove an object from an image

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **Playground**.
4. Choose **Image and video**.
5. If the **Configurations** pane isn't open, choose the configuration button.

6. For **Model** select a model to use.
7. For **Action** choose **Remove object**.
8. (Optional) For **Negative prompt** enter text that describes content or concepts that you don't want the model to include in the image.
9. In **Reference image** choose **Upload image** and upload the image that you want the model to use with the action.
10. (Optional) For **Negative prompt** enter text that describes content or concepts that you do not want the model to include in the image.
11. (Optional) In **Advanced configurations**, change how the model generates images by making advanced configuration changes. For more information, see Advanced configuration options.
12. In the center pane, use the masking tool to draw a bounding box around the area of the image that you want the action to update. You can do the following:
    a. Resize the bounding box by selecting a corner of the bounding box with your mouse button. Then, drag the mouse to resize the bounding box. Release the mouse button to complete resizing the bounding box.
    b. Move the bounding box by selecting the interior of the bounding box with your mouse button. Move the bounding box to the new location and release the mouse button.
13. Press Enter on your keyboard to start the action. Amazon Bedrock in SageMaker Unified Studio shows the image that the model generates in the playground.

Replace an object in an image

The following procedure shows you how to use a model to replace an object in an image that you supply. For example, you could replace a piece of furniture in an image with a different piece of furniture. You can set various configurations such as the number of images to generate and how strongly the prompt affects the generation of the image. For more information, see Configuration options.

> **Note**
>
> The object replacement action is only available with Titan Image Generator G1 V1 and Titan Image Generator G1 V2 models.

To replace an object in an image

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **Playground**.
4. Choose **Image and video**.
5. If the **Configurations** pane isn't open, choose the configuration button.
6. For **Model** select a model to use.
7. For **Action** choose **Remove object**.
8. (Optional) For **Negative prompt** enter text that describes content or concepts that you do not want the model to include in the image.
9. In **Reference image** choose **Upload image** and upload the image that you want the model to use with the action.
10. (Optional) In **Advanced configurations**, change how the model generates images by making advanced configuration changes. For more information, see Advanced configuration options.
11. In the center pane, use the masking tool to draw a bounding box around the area of the image that you want the action to update. You can do the following:

    a. Resize the bounding box by selecting a corner of the bounding box with your mouse button. Then, drag the mouse to resize the bounding box. Release the mouse button to complete resizing the bounding box.

    b. Move the bounding box by selecting the interior of the bounding box with your mouse button. Move the bounding box to the new location and release the mouse button.
12. Choose the run button on your keyboard to start the action. Amazon Bedrock in SageMaker Unified Studio shows the image that the model generates in the playground.

Replace the background for an image

The following procedure shows you how to use a model to replace the background for an image. For example, you could change the background for an image from a view of a forest to a view of city buildings. You can set various configurations such as the number of images to generate and how strongly the prompt affects the generation of the image. For more information, see Configuration options.

> **Note**
>
> The background replacement action is only available with Titan Image Generator G1 V1 and Titan Image Generator G1 V2 models.

To replace the background for an image

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **Playground**.
4. Choose **Image and video**.
5. If the **Configurations** pane isn't open, choose the configuration button.
6. For **Model** select a model to use.
7. For **Action** choose **Replace background**.
8. (Optional) For **Negative prompt** enter text that describes content or concepts that you don't want the model to include in the image.
9. In **Reference image** choose **Upload image** and upload the image that you want the model to use with the action.
10. (Optional) For **Negative prompt** enter text that describes content or concepts that you do not want the model to include in the image.
11. (Optional) In **Advanced configurations**, change how the model generates images by making advanced configuration changes. For more information, see Advanced configuration options.
12. In the center pane, use the masking tool to draw a bounding box around the area of the image that you want the action to preserve. The model updates the area outside of the bounding box. You can do the following:

    a. Resize the bounding box by selecting a corner of the bounding box with your mouse button. Then, drag the mouse to resize the bounding box. Release the mouse button to complete resizing the bounding box.

    b. Move the bounding box by selecting the interior of the bounding box with your mouse button. Move the bounding box to the new location and release the mouse button.

13. In the **Enter prompt** text box, enter a prompt that describes the background that you want the image to have.
14. Press Enter on your keyboard to start the action. Amazon Bedrock in SageMaker Unified Studio shows the image that the model generates in the playground.

Edit an image with the image sandbox

If you use the image playground with a Stable Diffusion XL model, you can use the image sandbox to make changes to a model.

> **Note**
>
> The image sandbox action is only available with Stable Diffusion XL models.

To edit an image in the image sandbox

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **Playground**.
4. Choose **Image and video**.
5. If the **Configurations** pane isn't open, choose the configuration button.
6. For **Model** select a model to use.
7. For **Action** choose **Remove object**.
8. (Optional) For **Negative prompt** enter text that describes content or concepts that you do not want the model to include in the image.
9. In **Reference image** choose **Upload image** and upload the image that you want the model to use with the action.
10. (Optional) In **Advanced configurations**, change how the model generates images by making advanced configuration changes. For more information, see Advanced configuration options.
11. In the center pane, use the masking tool to draw a bounding box around the area of the image that you want the action to update. You can do the following:

a. Resize the bounding box by selecting a corner of the bounding box with your mouse button. Then, drag the mouse to resize the bounding box. Release the mouse button to complete resizing the bounding box.

b. Move the bounding box by selecting the interior of the bounding box with your mouse button. Move the bounding box to the new location and release the mouse button.

12. In the **Enter prompt** text box, enter the prompt that describes the edit that that you want the model inside the bounding box.
13. Press Enter on your keyboard to start the action. Amazon Bedrock in SageMaker Unified Studio shows the image that the model generates in the playground.

Generate a video with the Amazon Bedrock image and video playground

The Amazon Bedrock in SageMaker Unified Studio image and video playground is where you can generate a short video with a suitable Amazon Bedrock model. To generate a video with a model, you supply a prompt that describe the video that you want to create and configuration information that influences how the model generates the video. For example, you can start with an image of a rock band and in the prompt request that you want to create an animated video of the band playing a live concert. The image and video playground also provides quick start prompts that illustrate the kinds of video that you can create.

You can download a video that you create in the image and video playground.

To create more complex videos, you can use a storyboard to connect a sequence of shots.

Your administrator sets the retention policy for videos that you upload and generate with the playground. For more information, contact your administrator.

Topics

- Configure video generation
- Generate a video from a prompt
- Plan a video with the storyboard

Configure video generation

To configure video generation, you choose a model to use and set optionally settings that influence the output of the model. Currently Amazon Bedrock in SageMaker Unified Studio supports video generation with Amazon Nova models. If you don't make any configuration changes, the playground uses the default values for the model.

Amazon Nova model settings

With Amazon Nova models, you can set the following configurations:

- **Start image** – (Optional) A reference image that model uses as a starting point for the video. The image dimensions must be 1280x720 pixels. If you supply an image with different dimensions, the playground resizes the image to 1280x720 pixels.
- **Seed** – (Optional) Initializes the random number generator used in the video generation process. Higher seed values don't correlate with any particular quality or characteristic in the output. Instead, use different seed values options to explore differing variations of output, either with or without the same prompt. Repeatedly using the same seed value and prompt creates the exact same video.

For more information, see the Amazon Nova guide.

Generate a video from a prompt

The following instructions show you how to generate a video in the image and video playground. If you are using Amazon Nova 1.1, you can use the storeyboard to make more complex videos.

To generate a video

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **Playground**.
4. Choose **Image and video**.
5. If the **Configurations** pane isn't open, choose the configuration button.
6. For **Model** select a model to use.

7. (Optional) In the **Configurations** section, set parameters to influence the output of the model. Note that additional configurations can be set in **Advanced configurations**. The parameters that are available to depend on the model you use. For more information, see Configure video generation.
8. In the **Enter prompt** text box, enter **`Create an animated video of a local classic rock band playing on an outdoor stage.`**. Alternatively, enter a prompt of your choosing.
9. Press Enter on your keyboard to start generating the video. Amazon Bedrock in SageMaker Unified Studio shows the video that the model generates in the playground.
10. Choose the play button to view the video.
11. (Optional) Choose the download button to download the video to your computer.

Plan a video with the storyboard

To create more a more complex video, you can use the storyboard to plan the video that you want to create. In the storyboard you connect a sequence of shots, which the model combines to generate the video. Each shot is a prompt and an optional start frame image. Each shot the model generates is always 6 seconds in length. You can't set a specific duration for the video, but you can affect the duration by adding or removing shots in the story board. The maximum length of video that you can generate is 120 seconds.

> **Note**
>
> You can only use the playground with Amazon Nova Reel 1.1.

To generate a video with the storyboard

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **Playground**.
4. Choose **Image and video**.
5. If the **Configurations** pane isn't open, choose the configuration button.

6. For **Model** select a model to use.
7. (Optional) In the **Configurations** section, set parameters to influence the output of the model. Note that additional configurations can be set in **Advanced configurations**. The parameters that are available to depend on the model you use. For more information, see Configure video generation.

   In the storyboard, you can't set the duration for the video.
8. In the center pane, choose **Storyboard**.
9. Choose **Add shot** and do the following:

   a. Choose **Describe what happens in this shot...** and enter the text for the prompt. You can update the prompt later, if neccessary.

   b. (Optional) Choose **Add start frame** to upload a starting image for the video.

   c. (Optional) Choose the trash icon to remove shots than you no longer need.
10. Repeat the previous step until you have added all the shots for your video.
11. Choose the run button to start generating the video. Don't leave the page while the model generates the video.
12. When the model finishes generating the video, choose the play button to view the video.
13. (Optional) Make further edits and add shots as you need them.
14. (Optional) Download the video by right-clicking on the video and selecting **Save video as...**.

Access shared generative AI assets in an Amazon Bedrock playground

Other Amazon Bedrock in SageMaker Unified Studio users can share chat agent app and prompts with you as *Shared generative AI assets*. You access assets from the **Shared apps and prompts** section in a playground. You can view the type of each asset and the projects that contains the assets.

In a playground, You can experiment with shared app and prompt assets, but you can't make changes to their configuration. If you want to make changes, you need to open the project that contains the asset. You can share chat agent apps and prompts that you create in a project. For more information, see Share an Amazon Bedrock chat agent app and Share an Amazon Bedrock prompt version.

To access and use a shared asset in a playground

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **Playground**.
4. Choose **Shared apps and prompts**.
5. In the playground, select the name of the asset that you want to use. The **Asset type** column tells the type of the asset (App or prompt).
6. Use the asset in the playground.

Build a chat agent app with Amazon Bedrock

An Amazon Bedrock in SageMaker Unified Studio chat agent app allows users to chat with an Amazon Bedrock model through a conversational interface, typically by sending text messages and receiving responses. The model analyzes the user's input, formulate an appropriate response, and carries on a dialogue with the user. You can use a chat agent apps for various purposes, such as providing customer service, answering questions, offering recommendations, or engaging in open-ended conversations on a wide range of topics. You can enhance a chat agent app app by integrating the following Amazon Bedrock features:

- **Data sources** — Enrich model responses by including context generated from an Amazon Bedrock knowledge base.
- **Guardrails** — Lets you implement safeguards for your chat agent app based on your use cases and responsible AI policies.
- **Functions** — Lets a model call a function to access a specific capability when handling a prompt.

When you first create an chat agent app, you have a working draft of the app. Changes you make to your chat agent app apply to the working draft. You iterate on your working draft until you're satisfied with the behavior of your app. At any time you can save your chat agent app.

Once you create and save chat agent app, you can do the following:

- Share the chat agent app with other users.
- Export the chat agent app for use outside of Amazon SageMaker Unified Studio.

- Use the chat agent app as an agent node in a flow app.

To use a chat agent app in a flow app you need to deploy the app. Amazon Bedrock in SageMaker Unified Studio deploys the app for you whenever you share the app with other users.

In this section you learn how to create chat agent app that uses Amazon Bedrock in SageMaker Unified Studio components such as a data source and a guardrail. You also learn how to share your app with other users.

Topics

- Create a chat agent app with Amazon Bedrock
- Deploy an Amazon Bedrock chat agent app
- Share an Amazon Bedrock chat agent app

Create a chat agent app with Amazon Bedrock

In this section, you learn how create a simple Amazon Bedrock in SageMaker Unified Studio chat agent app that creates playlists for a radio station.

The app can generate playlists and get the dates and locations of upcoming shows. Later, you add the following Amazon Bedrock features.

- A guardrail to prevent songs with inappropriate song titles.
- A knowledge base that lets the app create playlists using your unique song information.
- A function that gets today's top 10 songs.

Topics

- Step 1: Create the initial chat agent app
- Step 2: Add a guardrail to your chat agent app
- Step 3: Add a knowledge base to your chat agent app
- Step 4: Add a function call to your chat agent app

Step 1: Create the initial chat agent app

In this step you create a chat agent app that generates playlists for a radio station.

To create the app, you first need to create an Amazon SageMaker Unified Studio project. A project can contain multiple apps and is also where you can add the Amazon Bedrock components that you want your apps to use. Later you will add guardail, knowledge base, and function components to your app. You can share a project with other users and groups of users. For more information, see Share an Amazon Bedrock chat agent app.

To help guide users of the app, you can set user interface (UI) text, such as hint text for the beginning of a chat.

In the app, you can experiment with the randomness and diversity of the response that the model returns by changing the inference parameters.

While you develop your app, you work on the current draft. You can save the current draft to the app history. Later you might want to restart work from a previous draft. For more information, see Use app history to view and restore versions of an Amazon Bedrock app.

> **Warning**
>
> Generative AI may give inaccurate responses. Avoid sharing sensitive information. Chats may be visible to others in your organization.

To create an Amazon Bedrock chat agent app

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. On the Amazon SageMaker Unified Studio home page, in the **Amazon Bedrock in SageMaker Unified Studio** tile, choose **Build chat agent app** to create a new chat agent app. The **Select or create a new project to continue** dialog box opens.
4. In the **Select or create a new project to continue** dialog box, do one of the following:
   - If you want to use a new project, follow the instructions at Create a new project. For the **Project profile** in step 1, choose **Generative AI application development**.
   - If you want to use an existing project, select the project that you want to use and then choose **Continue**.
5. In **Untitled App - nnnn**, enter **`Radio show`** as the name for your app.

6. In the **Configs** pane, do the following:

   a. For **Model**, select a model that supports Guardrails, Data, and Function components. The description of the model tells you the components that a model supports. For full information about the model, choose **View full model details** in the information panel. For more information, see Find serverless models with the Amazon Bedrock model catalog. If you don't have access to an appropriate model, contact your administrator. Different models might not support all features.

   b. For **Enter a system instruction** in **Instructions for chat agent & examples**, enter **`You are a chat agent app that creates 2 hour long playlists for a radio station that plays rock and pop music.`**.

   c. In the **UI** section, update the user interface for the app by doing the following:

      i. In **Hint text for empty chat** enter **`Hi! I'm your radio show playlist creator.`**.

      ii. In **Hint text for user input** enter **`Enter a prompt that describes the playlist that you want.`**.

      iii. In **Quick start prompts** choose **Edit**.

      iv. Choose **Reset** to clear the list of quick start prompts

      v. For **Quick-start prompt 1**, enter **`Create a playlist of pop music songs.`**.

      vi. (Optional). Enter quick start prompts of your choice in the remaining quick start prompt text boxes.

      vii. Choose **Back to configs**.

7. Choose **Save** to save the current working draft of your app.

8. In the **Quick start prompts** section of the **Preview** pane, run the quick start prompt that you just created by choosing the prompt.

   The app shows the prompt and the response from the model in the **Preview** pane.

9. In the prompt text box (the text should read **Enter a prompt that describes the playlist that you want**), enter **`Create a playlist of songs where each song on the list is related to the next song, by musician, bands, or other connections. Be sure to explain the connection from one song to the next.`** .

10. Choose the run button (or press Enter on your keyboard) to send the prompt to the model.

11. (Optional) In the **Inference parameters** section change the inference parameters. For example, include less familiar songs in the playlist by increasing the **Temperature** inference parameter.

The inference parameters you can change are *Temperature*, *Top P*, and *Top K*. Not all models support each of these inference parameters. For more information, see Inference parameters.

12. (Optional) In **Stop sequences**, add one or more stop sequences for the model. A stop sequence ensures that the model stops generating text immediately after it generates text that matches the stop sequence. Stop sequences are useful for getting precise answers without unnecessary additional text. Not all models support stop sequences.
13. (Optional) In **Reasoning** select **Model reasoning** to enable model reasoning. Model reasoning is when a model uses chain of thought reasoning to take a large, complex task and break it down into smaller, simpler steps. You can only enable or disable model reasoning at the start of a new chat. You can specify the number of tokens to use, which includes both output and reasoning tokens. Not all models support model reasoning. For more information, see Enhance model responses with model reasoning in the *Amazon Bedrock user guide*.
14. (Optional) Share your app with others by following the instructions at Share an Amazon Bedrock chat agent app.
15. (Optional) Deploy your app for use in a flow app by following the instructions at Deploy an Amazon Bedrock chat agent app.
16. (Optional) Export your snapshot from Amazon SageMaker Unified Studio by following the instructions at Use your app outside of Amazon SageMaker Unified Studio.
17. Next step: Add a guardrail to your app by following the instructions at Step 2: Add a guardrail to your chat agent app.

Step 2: Add a guardrail to your chat agent app

Guardrails for Amazon Bedrock lets you implement safeguards for your Amazon Bedrock in SageMaker Unified Studio app based on your use cases and responsible AI policies. You can create multiple guardrails tailored to different use cases and apply them across multiple foundation models, providing a consistent user experience and standardizing safety controls across generative AI apps. You can configure denied topics to disallow undesirable topics and content filters to block harmful content in the prompts you send to a model and to the responses you get from a model. You can use guardrails with text-only foundation models. For more information, see Safeguard your Amazon Bedrock app with a guardrail.

Add a guardrail

This procedure shows you how to use a guardrail to safeguard the app you created in Step 1: Create the initial chat agent app. The guardrail prevents inappropriate language in song titles and filters out unwanted music genres.

To add a guardrail to an Amazon Bedrock app

1. Open the app that you created in Step 1: Create the initial chat agent app.
2. In the **Configs** pane, choose **Guardrails** and then **Create new guardrail**.
3. For **Guardrail name**, enter **`prevent_unwanted_songs`**.
4. For **Guardrail description**, enter **`Prevents inappropriate or undesirable songs.`**.
5. In **Content filters** make sure **Enable content filters** is selected. For more information, see Content filters.
6. In **Filter for prompts** make sure the filter for each category is set to **High**.
7. Make sure **Apply the same filters for responses** is selected.
8. In **Blocked messsaging** do the following.
   a. For **Blocked messaging for prompts**, enter **`Sorry, your prompt contained inappropriate text.`**.
   b. Clear **Apply the same message for blocked responses**.
   c. For **Blocked messaging for responses**, enter **`Sorry, but I can't respond with information that contains inappropriate text.`**.
9. Choose **Create** to create the guardrail.
10. In the **Configs** pane, in the **Guardrails** section, select the guardrail that you just created (**prevent_unwanted_songs**). It might take a minute for the guardrail to appear in the list.
11. Test the guardrail by entering **`Create a list of 10 songs where each song has a swear word in the title.`** In the prompt edit box.
12. Choose the run button to send the prompt to the model. The model should respond with the message **Sorry, but I can't respond with information that contains inappropriate text.**
13. Use a denied topic filter to prevent requests for music from a specific music genere. For information about denied topics, see Denied topics.

    To add the filter, do the following.

    a. In the **Guardrails** section of the **Configs** pane, select the guardrail and choose **Preview**.

b. Choose **Edit** to edit the guardrail.

c. In **Denied topics**, choose **Add topic**.

d. For **Name**, enter **`heavy metal`**.

e. For **Definition for topic**, enter **`Avoid mentioning songs that are from the heavy metal genre of music.`**.

f. In **Sample phrases - optional**, enter **`Create a playlist of heavy metal songs`**.

g. (Optional) Choose **Add phrase** to add other phrases.

h. Choose **Save**.

i. Om the **Edit guardrail** page, choose **Update** to update the guardrail.

j. Test the guardrail by entering **`Create a list of heavy metal songs.`** in the prompt edit box.

k. Choose the run button to send the prompt to the model. The model should respond with the message **Sorry, your prompt contained inappropriate text**.

14. Next step: Add a data source to your app by following the instructions at Step 3: Add a knowledge base to your chat agent app.

Step 3: Add a knowledge base to your chat agent app

You can use your own data into your application by adding a knowledge base to your app. Doing this allows your app to access to information that is only available to you. When your app passes a query, Amazon Bedrock in SageMaker Unified Studio generates a response that includes the query results from the knowledge base. For more information, see Add a Knowledge Base to your Amazon Bedrock app.

In this topic, you update the app you created in Step 1: Create the initial chat agent app to use a CSV file as a document (local file) data source for a knowledge base. The CSV file includes information about bands that don't have public metadata such as song length, or music genre. The user can use the app to create a playlist based on criteria such as song length or music genre.

To add your own data to an Amazon Bedrock app

1. Create a CSV file name *songs.csv* and fill with the following ficticious CSV data.

```
song,artist,genre,length-seconds
"Celestial Odyssey","Starry Renegades","Cosmic Rock",240
"Neon Rapture","Synthwave Siren","Synthwave Pop",300
```

```
"Wordsmith Warriors","Lyrical Legions","Lyrical Flow",180
"Nebula Shredders","Galactic Axemen","Cosmic Rock",270
"Electro Euphoria","Neon Nomads","Synthwave Pop",210
"Rhythm Renegades","Percussive Pioneers","Lyrical Flow",240
"Stardust Rift","Cosmic Crusaders","Cosmic Rock",180
"Synthwave Serenade","Electro Enchanters","Synthwave Pop",300
"Lyrical Legends","Rhyme Royale","Lyrical Flow",240
"Supernova Shredders","Amplified Ascension","Cosmic Rock",300
"Celestial Chords","Ethereal Echoes","Cosmic Rock",240
"Neon Nirvana","Synthwave Sirens","Synthwave Pop",270
"Verbal Virtuoso","Lyrical Maestros","Lyrical Flow",210
"Cosmic Collision","Stellar Insurgents","Cosmic Rock",180
"Pop Paradox","Melodic Mavericks","Synthwave Pop",240
"Flow Fusion","Verbal Virtuosos","Lyrical Flow",300
"Shredding Shadows","Crimson Crusaders","Cosmic Rock",270
"Synth Serenade","Electro Enchanters","Synthwave Pop",180
"Wordsmith Warlords","Lyrical Legionnaires","Lyrical Flow",240
"Sonic Supernova","Amplified Ascension","Cosmic Rock",210
"Celestial Symphony","Ethereal Ensemble","Cosmic Rock",300
"Electro Euphoria","Neon Nomads","Synthwave Pop",180
"Lyrical Legends","Rhyme Royale","Lyrical Flow",270
"Crimson Crescendo","Scarlet Serenaders","Cosmic Rock",240
"Euphoric Tides","Melodic Mystics","Synthwave Pop",210
"Rhythm Renegades","Percussive Pioneers","Lyrical Flow",180
"Cosmic Collision","Stellar Insurgents","Cosmic Rock",300
"Stardust Serenade","Celestial Crooners","Synthwave Pop",240
"Wordsmith Warriors","Lyrical Legions","Lyrical Flow",270
"Sonic Supernova III","Amplified Ascension","Cosmic Rock",180
```

2. Open the app that you created in Step 1: Create the initial chat agent app.
3. In **Data** choose **Use Knowledge Base** and then **Create Knowledge Base**. The **Create Knowledge Base** pane is shown. If you've previously created the knowledge base, go to step *10* and select the knowledge base.
4. For **Name**, enter a name for the Knowledge Base.
5. For **Description**, enter a description for the Knowledge Base.
6. In **Add data sources**, choose **Local file**.
7. Choose **Click to upload** and upload the CSV file that you created in step 1. Alternatively, add the CSV by dragging and dropping the document from your computer.

   For more information, see Use a Local file as a data source.
8. For **Embeddings model**, choose a model for converting your data into vector embeddings.

9. Choose **Create**. It might take Amazon Bedrock in SageMaker Unified Studio a few minutes to create the knowledge base.
10. For **Select Knowledge Base**, select the Knowledge Base that you just created.
11. Test the data source by entering **`Create a playlist of songs in the Lyrical Flow genre`** in the prompt text box.
12. Choose the run button to send the prompt to the model. The model should respond with a playlist of songs from the Lyrical Flow genre that the CSV file contains.
13. Choose **Save** to save the app.

Step 4: Add a function call to your chat agent app

Amazon Bedrock in SageMaker Unified Studio functions let a model include information that it has no previous knowledge of in its response. For example, you can use a function to include dynamic information in a model's response such as a weather forecast, sports results, or traffic conditions. To use a function in Amazon Bedrock in SageMaker Unified Studio you add a function component to your app. For more information, see Call functions from your Amazon Bedrock chat agent app.

In Amazon Bedrock in SageMaker Unified Studio, a function calls an API hosted outside of Amazon Bedrock in SageMaker Unified Studio. You either create the API yourself, or use an existing API. To create an API, you can use Amazon API Gateway.

In this procedure, you add a function to the app that you created in Step 1: Create the initial chat agent app so that users can get a list of the top 10 songs played on the radio station that day.

To add a function to an Amazon Bedrock app

1. Create a HTTPS server that implements a `TopSongsToday` function. Make sure the function adheres to the following schema.

```
openapi: 3.0.0
info:
  title: Top Songs API
  description: API to retrieve the top 10 songs played today
  version: 1.0.0

paths:
  /top-songs:
    get:
      operationId: TopSongsToday
```

```
      summary: Get the top 10 songs played today
      description: >
        This endpoint returns an array of the top 10 songs played today,
        ordered by popularity. The first element in the array (index 0)
        represents the most popular song, and the last element (index 9)
        represents the 10th most popular song.
      responses:
        '200':
          description: Successful response
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/TopSongs'

components:
  schemas:
    TopSongs:
      type: array
      items:
        $ref: '#/components/schemas/Song'
      description: >
        An array containing the top 10 songs played today. The first element
        (index 0) is the most popular song, and the last element (index 9)
        is the 10th most popular song.
      example:
        - title: 'Song Title 1'
          artist: 'Artist Name 1'
          album: 'Album Name 1'
        - title: 'Song Title 2'
          artist: 'Artist Name 2'
          album: 'Album Name 2'
        # ... up to 10 songs

    Song:
      type: object
      properties:
        title:
          type: string
          description: The title of the song
        artist:
          type: string
          description: The name of the artist or band
        album:
          type: string
```

```
        description: The name of the album the song is from
    required:
      - title
      - artist
      - album
```

2. Open the app that you created in Step 1: Create the initial chat agent app.
3. In **Models**, choose a model that supports functions. If you don't have access to an appropriate model, contact your administrator.
4. In **Functions**, choose **Create new function**.
5. In the **Create function** pane, do the following.
   a. For in **Function name**, enter **`Top_ten_songs_today`**.
   b. For **Function description (optional)**, enter **`Today's top 10 songs.`**.
   c. For **Function schema**, enter the OpenAPI schema from step one.
   d. Choose **Validate schema** to validate the schema.
   e. In **Authentication method** choose the authentication method for your HTTP server. For more information, see the section called "Authentication methods".
   f. In **API servers**, enter the URL for your server in **Server URL**. This value is autopopulated if the server URL is in the schema.
   g. Choose **Create** to create your function. It might take a few minutes to create the function.
6. For **Enter a system instruction**, update the system instruction so that it describes the function. Use the following text: **`You are an app that creates 2 hour long playlists for a radio station that plays rock and pop music. The function Top_ten_songs_today gets the most popular song played on the radio station.`**.
7. Test the function by doing the following.
   a. Enter **`What are today's top 10 songs?`** in the prompt edit box.
   b. Choose the run button to send the prompt to the model. The model should respond with the list of today's top 10 songs.

Deploy an Amazon Bedrock chat agent app

Deployment of a chat agent app happens for the following reasons:

- You want to use a chat agent app as an agent node within a flow app. Before you can do so, you must first deploy the chat agent app.
- You want to share a chat agent app with other users. When you share the app, Amazon Bedrock deploys the chat agent app for you.

During deployment Amazon Bedrock creates, or reuses, an *alias* for the chat agent app and creates a new *version* of the chat agent app.

Aliases and versions

An alias is associated with a specific version of a chat agent app. When you use a chat agent app as an agent node in a flow app, you configure the agent node in the flow app with the alias of the chat agent app. When you share an app, the recipient can access the version of the app that is associated with the alias. You can create up to 8 aliases for a chat agent app.

A version is a snapshot of a chat agent app at the time you deploy the app. Amazon Bedrock creates a new version of your chat agent app each time you deploy the app. Amazon Bedrock creates versions in numerical order, starting from 1.

With aliases, you can switch efficiently between different versions of a chat agent app, without having to update the flow apps that use the chat agent app. For example, if you discover an issue with the current version of your chat agent app, you can quickly update agent nodes that use the current version to a previous version.

The following procedure shows you how to deploy a chat agent app. After deploying the app, you can then use the app as a node agent in a flow app. You don't need to deploy a chat agent app before sharing the app, as Amazon Bedrock deploys the app for you. For more information, see Deploy an Amazon Bedrock chat agent app.

To deploy a chat agent app

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. From the project selector dropdown at the top of the page, choose the project that you want to use.
4. In the left navigation pane, under **Generative AI**, choose **AI apps**.

5. Open the app that you want to deploy.
6. Choose **Deploy**.
7. In **App description** enter a short description for the app. Make sure that that the description lets users understand the purpose of the app.
8. Assign an alias by doing one of the following:
   - Choose **Create a new alias** to create a new alias. Then enter an name and optional description for the alias.
   - Choose **Select an existing alias** to use an existing alias. Then select the existing alias that you want to use.
9. Choose **Deploy** to deploy the app. It might take a few seconds to deploy the app.
10. To use the deployed app, select **Working draft** in the configuration pane and then select the name of the alias that you used to deploy the app.
11. Enter a prompt to try your deployed app.

Modify the version of an Amazon Bedrock chat agent app

You can change the version of a chat agent app that an agent node in a flow app uses, or the version of a chat agent app that is shared with other users. To change the version, you modify the alias for the chat agent app to reference the new version. After updating the alias, you don't need to update the flow app or shares of the chat agent app for them to use the new version.

The following procedure shows you how to change the version of a chat agent app that an alias for the app references.

To modify the version that an alias references

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. From the project selector dropdown at the top of the page, choose the project that you want to use.
4. In the left navigation pane, under **Generative AI**, choose **AI apps**.

5. Open the app that you want to use.
6. Choose the selector on the **Deploy** button and select **View aliases**. The **View and manage aliases** pane opens.
7. For the alias that you want modify, choose **Edit**.
8. In the **Edit alias** pane, select the version that you want the alias to use in **Select version to associate with this alias**.
9. (Optional) Update the name and description for the alias.
10. Choose **Save** to save your changes.

Share an Amazon Bedrock chat agent app

A snapshot of an Amazon Bedrock in SageMaker Unified Studio chat agent app is a point-in-time capture of the app's state, including its code, configuration, and any associated data.

You can share a snapshot with all members of your Amazon SageMaker Unified Studio domain, or with specific users or groups in your Amazon SageMaker Unified Studio domain. When you first share a snapshot, you get a share link to the snapshot that you can send to users. If you share the snapshot with all users, Amazon SageMaker Unified Studio grants permission to a user, when they first open the share link. Amazon SageMaker Unified Studio also adds the snapshot to the user's shared assets list. If you share the snapshot with specific users and groups, the snapshot is immediately available in their shared assets list. They can also use the share link to access the snapshot. By default, sharing a snapshot is restricted to only those users or groups that you select.

When you share a chat agent app, Amazon SageMaker Unified Studio also publishes the chat agent app to the Amazon SageMaker AI Catalog.

When sharing occurs, Amazon Bedrock deploys the snapshot of the chat agent app. When you first share a snapshot, Amazon Bedrock creates a new alias for the chat agent app and a new version of the chat agent app that represents the snapshot. On subsequent shares, Amazon Bedrock creates a new version of the app, and associates the alias with the new version. If necessary, You can change the version that is associated with an alias. For more information, see Modify the version of an Amazon Bedrock chat agent app.

To share a chat agent app snapshot

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.

2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. From the project selector dropdown at the top of the page, choose the project that you want to use.
4. In the left navigation pane, under **Generative AI**, choose **AI apps**.
5. Open the chat agent app that you want to share.
6. Choose **Share**.
7. In **App description** enter a short description for the chat agent app. Make sure that that the description lets users understand the purpose of the chat agent app.
8. Do one of the following:

   - If you want to share the chat agent app snapshot with all members of your Amazon SageMaker Unified Studio domain, select turn on **Grant access with link**.
   - If you want to share the chat agent app snapshot with specific Amazon SageMaker Unified Studio domain users or groups, do the following in **Share with specific users or groups**:

     1. For **Member type** choose **Individual user** or **Group**, depending on the type of member that you want share the chat agent app with.
     2. Search for the users or groups that you want to share the chat agent app with by entering the user name or group in the **Search by alias to invite members** text box.
     3. In the drop down list, select the matching user name or group that want to share the chat agent app with.
     4. Choose **Add** to add the user or group.
9. Choose **Share** to share the chat agent app.
10. When the success message appears, choose **Copy link** and send the link to the users that you are sharing the chat agent app snapshot with. If **Grant access with link** is off, the link only works for users that you have explicitly granted access to the chat agent app.

As the chat agent app creator, you, and other members of the chat agent app project, can make changes to the chat agent app and share a fresh snapshot of the chat agent app. Only the latest snapshot is available to users. Users that you share a snapshot to can't make any changes to the snapshot.

To change who you share the chat agent app with, open the chat agent app, choose **Share** and make your changes. Choose **Done** to complete the changes. If you are sharing the snapshot with

all users, turning off **Grant access with link** restricts access to users that you have specifically share the snapshot with.

You can't stop sharing a snapshot without deleting the chat agent app. If you delete the chat agent app, the snapshot is no longer shared and is removed from the Amazon SageMaker AI Catalog. If you want to deny access to everyone, without deleting the chat agent app, edit the snapshot and remove all users and groups. If you shared the snapshot with all users, turn off **Grant access with link**. Note that Amazon SageMaker Unified Studio doesn't remove the snapshot from the Amazon SageMaker AI Catalog.

If you need the share link later, share the chat agent app again and copy the share link. You can also change the users that you share with the chat agent app with.

To see which chat agent apps you have shared, Open the app project, choose **Asset gallery** and then **My apps**. Check the **Share status** column for the chat agent app.

Build a flow app with Amazon Bedrock

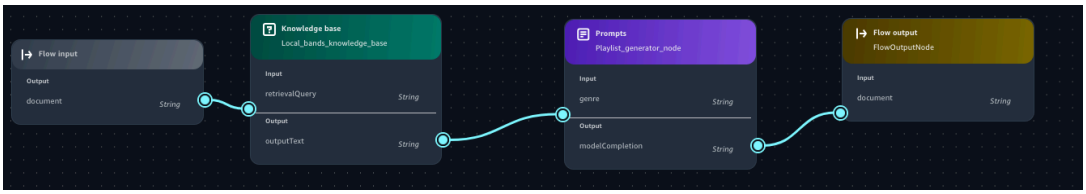
A flow app let you link prompts, supported foundational models, and other units of work, such as an Amazon Bedrock knowledge base, together and create generative AI workflows for end-to-end solutions. For example, you could create flow apps to do the following.

- **Create a playlist of music** – Create a flow connecting a prompt node and knowledge base node. Provide the following prompt to generate a playlist: **`Create a playlist`**. After processing the prompt, the flow queries a knowledge base to look up information about local bands, such as the length of songs and genre of music. The flow then generates a playlist based the information in the knowledge base.
- **Troubleshoot using the error message and the ID of the resource that is causing the error** – The flow looks up the possible causes of the error from a documentation knowledge base, pulls system logs and other relevant information about the resource, and updates the faulty configurations and values for the resource.

In this section you create a flow app that generates a playlist of music from a knowledge base of songs by fictional local bands.

To create a flow app, you use the *flow builder* which is a tool in Amazon Bedrock in SageMaker Unified Studio to build and edit flow apps through a visual interface. You use the visual interface to

drag and drop nodes onto the interface and configure inputs and outputs for these nodes to define your flow.



In your flow you can apply logical conditions to direct the output from a node to different destinations. You can then run the flow within Amazon Bedrock in SageMaker Unified Studio and view the output.

The following list introduces you to the basic elements of a flow.

- **Flow** – A flow is a construct consisting of a name, description, permissions, a collection of nodes, and connections between nodes. When you run a flow, the input to the flow is sent through each node of the flow until the flow emits the final output from an output node.

- **Node** – A node is a step inside a flow. For each node, you configure its name, description, input, output, and any additional configurations. The configuration of a node differs based on its type.

  For information about the types of nodes that Amazon Bedrock in SageMaker Unified Studio supports, see Flow nodes available in Amazon Bedrock.
- **Connection** – There are two types of connections used in flow apps:
  - A **data connection** is drawn between the output of one node (the *source node*) and the input of another node (the *target node*) and sends data from an upstream node to a downstream node. In the flow builder, data connections are solid lines.
  - A **conditional connection** is drawn between a condition in a condition node and a downstream node and sends data from the node that precedes the condition node to a downstream node if the condition is fulfilled. In the flow builder, conditional connections are dotted lines.
- **Expressions** – An expression defines how to extract an input from the whole input entering a node. To learn how to write expressions, see Define inputs with expressions.

If you want to use your flow app outside of Amazon SageMaker Unified Studio, you can export and deploy the app to an AWS account. For more information, see Use your app outside of Amazon SageMaker Unified Studio.

Warning

Generative AI may give inaccurate responses. Avoid sharing sensitive information. Chats may be visible to others in your organization.

Topics

- Create a flow app with Amazon Bedrock
- Define inputs with expressions
- Use logic nodes to control flow
- Use a chat agent app in a flow app
- Flow nodes available in Amazon Bedrock

Create a flow app with Amazon Bedrock

In this section you first build a flow app that generates a playlist of music from an Amazon Bedrock knowledge base of songs by fictional local bands. Next, you use Reusable prompts to add a prompt that can customizes the playlist for different genres of music.

Topics

- Step 1: Create an initial flow app
- Step 2: Add a Knowledge Base to your flow app
- Step 3: Add a prompt to your flow app
- Step 4: Add a condition to your flow app

Step 1: Create an initial flow app

In this procedure you create an initial flow app which has an Flow input node and a Flow output node.

A flow contains only one flow input node which is where the flow begins. The flow input node takes your input and passes it to the next node in a data type of your choice (String, Number, Boolean, Object and Array). In these procedures, the input to the flow is a String. To learn more about using different data types in a flow, see Define inputs with expressions.

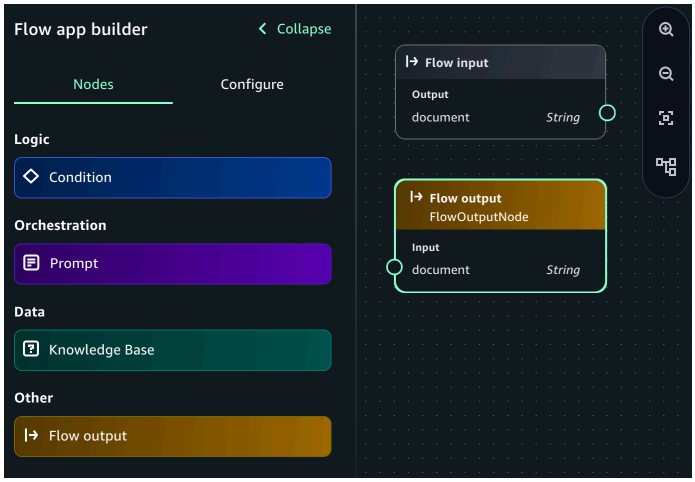
A flow output node extracts the input data from the previous node, based on the defined expression, and outputs the data. A flow can have multiple flow output nodes if there are multiple branches in the flow.

After completing the procedure, the flow app is empty, other than the flow input and flow output nodes. In the next step you add Knowledge Base as a data source and run the flow app for the first time.

While you develop your app, you work on the current draft. You can save the current draft to the app history. Later you might want to restart work from a previous draft. For more information, see Use app history to view and restore versions of an Amazon Bedrock app.

To create an initial flow app

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. Under **Build**, choose **Flow**.
4. In the **Select or create a new project to continue** dialog box, do one of the following:
   - If you want to use a new project, follow the instructions at Create a new project. For the **Project profile** in step 1, choose **Generative AI application development**.
   - If you want to use an existing project, select the project that you want to use and then choose **Continue**.
5. In the flow builder, choose the flow name (**Untitled flow-nnnn**) and enter **`Local bands`** as the name for the flow.
6. In the **flow app builder** pane, select the **Nodes** tab. The center pane displays a **Flow input** node and a **Flow output** node. These are the input and output nodes for your flow. The circles on the nodes are connection points. In the next procedure, you use the connection points to connect a Knowledge Base node to the Flow input node and the Flow output node.



7. Next step: Step 2: Add a Knowledge Base to your flow app.

Step 2: Add a Knowledge Base to your flow app

In this procedure, you add a Knowlege Base node as a data source to the flow that you created in Step 1: Create an initial flow app. The Knowledge Base you add is Comma Seperated Values (CSV) file containing a list of ficticious songs and artists. The list includes the duration (seconds) and genre of each song. For more information about Knowledge Bases, see Add a Knowledge Base to your Amazon Bedrock app.

During the procedure, you make connections from the Flow input node to the Knowledge Base node and from the Knowledge Base node to the Flow output node. At some point, you might need to delete a node or remove a node connection. To delete a node, select the node that you want to delete and press the Delete button. To remove a connection, choose the connection that you want to delete and then press the delete button.

When you run the flow with the input **`Create a playlist`**, the app creates a playlist using songs only from the Knowledge Base.

To create the flow with a Knowledge Base

1. Create a CSV file name *songs.csv* and fill with the following ficticious CSV data. This is the data source for your Knowledge Base. Save the CSV file to your local computer.

```
song,artist,genre,length-seconds
"Celestial Odyssey","Starry Renegades","Cosmic Rock",240
"Neon Rapture","Synthwave Siren","Synthwave Pop",300
"Wordsmith Warriors","Lyrical Legions","Lyrical Flow",180
```

```
"Nebula Shredders","Galactic Axemen","Cosmic Rock",270
"Electro Euphoria","Neon Nomads","Synthwave Pop",210
"Rhythm Renegades","Percussive Pioneers","Lyrical Flow",240
"Stardust Rift","Cosmic Crusaders","Cosmic Rock",180
"Synthwave Serenade","Electro Enchanters","Synthwave Pop",300
"Lyrical Legends","Rhyme Royale","Lyrical Flow",240
"Supernova Shredders","Amplified Ascension","Cosmic Rock",300
"Celestial Chords","Ethereal Echoes","Cosmic Rock",240
"Neon Nirvana","Synthwave Sirens","Synthwave Pop",270
"Verbal Virtuoso","Lyrical Maestros","Lyrical Flow",210
"Cosmic Collision","Stellar Insurgents","Cosmic Rock",180
"Pop Paradox","Melodic Mavericks","Synthwave Pop",240
"Flow Fusion","Verbal Virtuosos","Lyrical Flow",300
"Shredding Shadows","Crimson Crusaders","Cosmic Rock",270
"Synth Serenade","Electro Enchanters","Synthwave Pop",180
"Wordsmith Warlords","Lyrical Legionnaires","Lyrical Flow",240
"Sonic Supernova","Amplified Ascension","Cosmic Rock",210
"Celestial Symphony","Ethereal Ensemble","Cosmic Rock",300
"Electro Euphoria","Neon Nomads","Synthwave Pop",180
"Lyrical Legends","Rhyme Royale","Lyrical Flow",270
"Crimson Crescendo","Scarlet Serenaders","Cosmic Rock",240
"Euphoric Tides","Melodic Mystics","Synthwave Pop",210
"Rhythm Renegades","Percussive Pioneers","Lyrical Flow",180
"Cosmic Collision","Stellar Insurgents","Cosmic Rock",300
"Stardust Serenade","Celestial Crooners","Synthwave Pop",240
"Wordsmith Warriors","Lyrical Legions","Lyrical Flow",270
"Sonic Supernova III","Amplified Ascension","Cosmic Rock",180
```

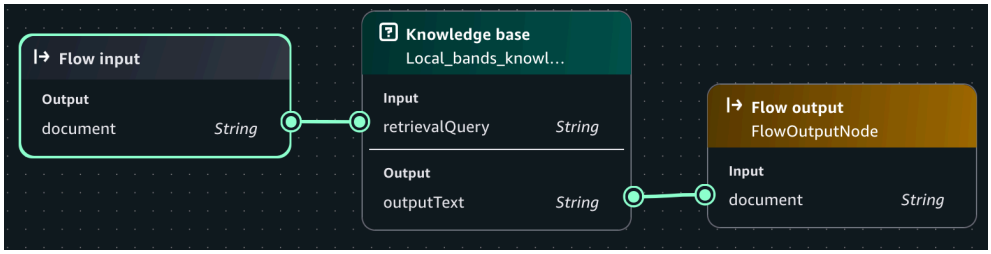
2. Open the flow app that you created in Step 1: Create an initial flow app.
3. Add and configure a Knowledge Base node by doing the following:

   a. In the **flow app builder** pane, select the **Nodes** tab.

   b. From the **Data** section, drag a **Knowledge Base** node onto the flow builder canvas.

   c. The circles on the nodes are connection points. Using your mouse, click on the circle for the **Flow input** node and draw a line to the circle on **Input** section of the Knowledge Base node that you just added.

   d. Connect the **Output** of the Knowledge Base node in your flow with the **Input** of the **Flow output** node.



e. Select the Knowledge Base node that you just added.

f. In the **flow builder** pane, choose the **Configure** tab and do the following:

   i. In **Node name** enter **`Local_bands_knowledge_base`**.

   ii. In **Knowledge Base Details**, choose **Create new Knowledge Base** to open the **Create Knowledge Base** pane.

   iii. For **Knowledge Base name**, enter **`Local-bands`**.

   iv. For **Knowledge Base description**, enter **`Songs by local bands. Includes song, artist, genre, and song length (in seconds).`**.

   v. In **Add data sources**, choose **Local file**.

   vi. Choose **Click to upload** and upload the CSV file (songs.csv) that you created in step 1. Alternatively, add your source document by dragging and dropping the CSV from your computer.

   vii. For **Parsing** leave as **Default parsing**.

   viii. For **Embeddings model**, choose a model for converting your data into vector embeddings.

   ix. For **Vector store**, choose **OpenSearch Serverless**.

   x. Choose **Create** to create the Knowledge Base. It might take a few minutes to create the Knowledge Base.

g. Back in the **flow builder** pane, in **Select Knowledge Base**, select the Knowledge Base that you just created (Local-bands).

h. In **Select response generation model**, select the model that you want the Knowledge Base to generate responses with.

i. (Optional) In **Select guardrail** select an existing guardrail or create a new guardrail. For more information, see Safeguard your Amazon Bedrock app with a guardrail.

4. Choose **Save** to save the app.

5. Test your prompt by doing the following:

a. On right side of the page, choose **<** to open the test pane.

b. Enter **`Create a playlist`** in the prompt **Text** box.

c. Press Enter on the keyboard or choose the run button to test the prompt.

d. If necessary, make changes to your flow. If you are satisfied with the flow, choose **Save**.

6. Next step: Step 3: Add a prompt to your flow app.

Step 3: Add a prompt to your flow app

In this procedure you add a prompt to the flow by adding a prompt node. The prompt allows you to easily choose which genre of songs should be included in the playlist that the flow generates. For more information, see Reuse and share Amazon Bedrock prompts.

To add a prompt to the flow

1. In the **flow builder** pane, select **Nodes**.
2. From the **Orchestration** section, drag a **Prompt** node onto the flow builder canvas.
3. Select the node you just added.
4. In the **Configurations** tab of the **flow builder** pane, do the following:

   a. In **Node name** enter **`Playlist_generator_node`**.

   b. In **Prompt details** choose **Create new prompt** to open the **Create prompt** pane.

   c. For **Prompt name** enter **`Playlist_generator_prompt`**.

   d. For **Model**, choose the model that you want the prompt to use.

   e. For **Prompt message** enter **`Create a playlist of songs in the genre {{genre}}.`**.

   f. (Optional) In **Model configs**, make changes to the inference parameters or provide system instruction prompts.

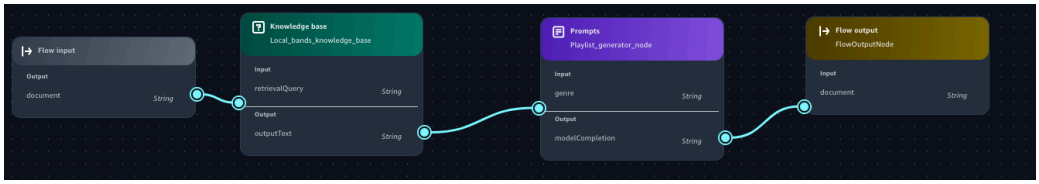
   g. Choose **Save draft and create version** to create the prompt. It might take a couple of minutes to finish creating the prompt.

5. In the flow builder, choose the prompt node that you just added.
6. In the **Configure** tab, do the following in the **Prompt details** section:

   a. In **Prompt** select the prompt that you just created.

   b. In **Version** select the version (**1**) of the prompt to use.

7. (Optional) In **Select guardrail** select an existing guardrail. For more information, see Safeguard your Amazon Bedrock app with a guardrail.
8. Update the flow paths by doing the following:
   a. Delete the output from the **Knowledge Base** node that goes into the **Flow output**.
   b. Connect the output from the **Knowledge Base** node to the input of the **Prompts** node.
   c. Connect the output from the **Prompts** node to the input of the **Flow output** node.
9. Choose **Save** to save the flow. The flow should look similar to the following.



10. Test your prompt by doing the following:
    a. On right side of the app flow page, choose **<** to open the test pane.
    b. For the **Text** box, enter **`Cosmic Rock`**.
    c. Press Enter on the keyboard or choose the run button to test the prompt. The response should be a playlist of songs in the Cosmic Rock genre.
    d. Change the prompt to **`Synthwave Pop`** and run the prompt again. The songs should now be from the Synthwave Pop genre.
    e. If necessary, make changes to your flow. If you are satisfied with the flow, choose **Save**.
11. Next step: Step 4: Add a condition to your flow app.

Step 4: Add a condition to your flow app

In this procedure, you add a condition node to the flow so that if you enter the prompt **`Cosmic Rock`**, the flow only generates a playlist from the local bands Knowledge Base. If you enter a different genre, the flows uses the playlist generator prompt to create a playlist of well known artists in that genre.

To add a condition to the flow

1. In the **flow app builder** pane, choose **Nodes**.
2. From the **Logic** section, drag a **Condition** node onto the flow builder canvas.
3. Select the **Condition** node that you just added.

4. Add the flow that generates a playlist from local bands by doing the following:

   a. In the **Inputs** section of the **Configurations** tab, change the **Node Name** to **`Local_cosmic_rock_node`**.

   b. In the **Inputs** section, change the **Name** to **`genre`**.

   c. In the **Conditions** section, do the following:

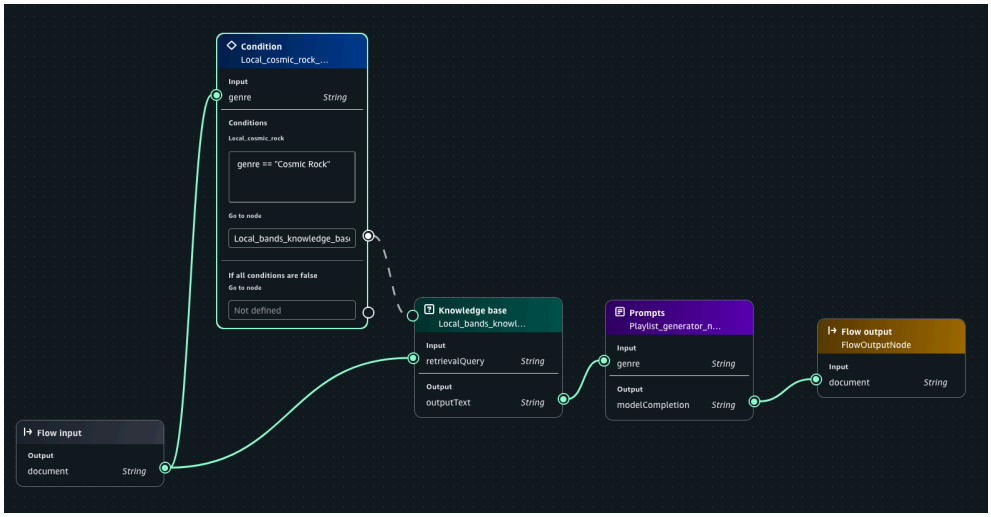
      i. For **Name**, enter **`Local_cosmic_rock`**.

      ii. For **Condition**, enter the condition **`genre == "Cosmic Rock"`**.

   d. In the flow builder, choose the condition node that you just added.

   e. Connect **Go to node** to the **Knowledge base** node.

   f. Connect the **Output** of the **Flow input** node to the **Input** of the **Condition** node. Leave the existing connection to the **Knowledge Base** node as this ensures the prompt is passed to the Knowledge Base.



5. Choose **Save** to save your flow app.

6. Add the flow that generates a playlist by well known bands by doing the following:

   a. In the **flow app builder** pane, select **Nodes**.

   b. From the **Orchestration** section, drag a **Prompt** node onto the flow builder canvas.

   c. Select the node you just added.

   d. Choose the **Configurations** tab of the **flow builder** pane and do the following:

      i. For **Node name**, enter **`Well_known_artist_playlist_generator_node`**.

ii. In **Prompt details** section, choose the **Playlist_generator_prompt** prompt that you previously created.

iii. For **Version**, select the version (**1**) of the prompt to use.

iv. Connect the **Output** from the **Flow input** node to the **Input** of the prompt that you just created.

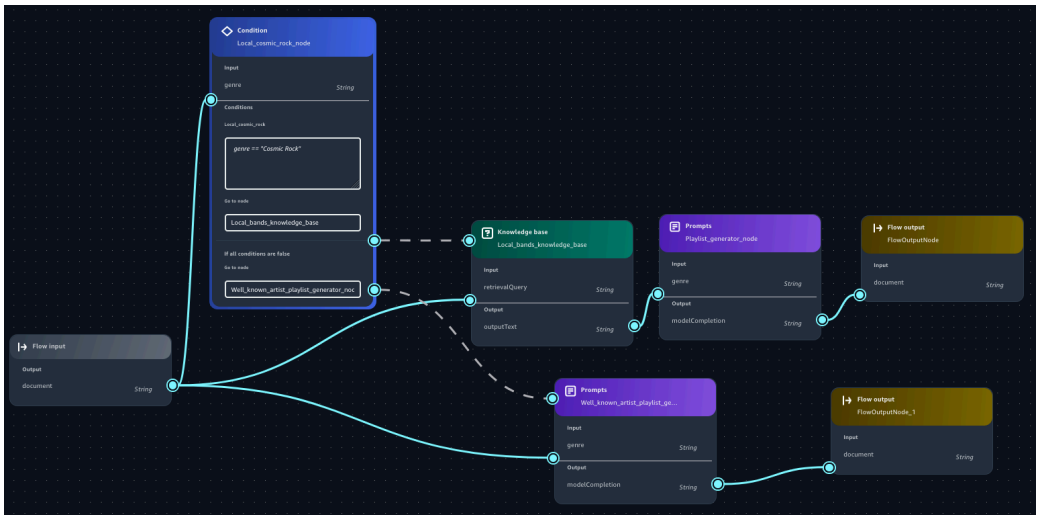
v. In the **Condition** node, connect the **If all conditions are false** go to node to the new prompt.

vi. In the **flow app builder** pane, select **Nodes**.

vii. From the **Other** section, drag a **Flow output** node onto the flow builder canvas.

viii. Connect the **Output** of the new **Prompt** (Well_known_artist_playlist_generator_node) to the input of the new **Flow output** node.

7. Choose **Save** to save the flow. The flow should look similar to the following.



8. Test your prompt by doing the following:

a. On right side of the app flow page, choose **<** to open the test pane.

b. In **Enter prompt**, enter **`Cosmic Rock`**.

c. Press Enter on the keyboard or choose the run button to test the prompt. The response should be a playlist of songs in the Cosmic Rock genre with bands that are only from the Knoweledge Base.

d. Change the prompt to **`Classic Rock`** and run the prompt again. The songs should now be well known bands from the classic rock genre.

Define inputs with expressions

When you configure the inputs for a node, you must define it in relation to the whole input that will enter the node. The whole input can be a string, number, boolean, array, or object. To define an input in relation to the whole input, you use a subset of supported expressions based off JsonPath. Every expression must begin with `$.data`, which refers to the whole input. Note the following for using expressions:

- If the whole input is a string, number, or boolean, the only expression that you can use to define an individual input is `$.data`
- If the whole input is an array or object, you can extract a part of it to define an individual input.

As an example to understand how to use expressions, let's say that the whole input is the following JSON object:

```
{
    "animals": {
        "mammals": ["cat", "dog"],
        "reptiles": ["snake", "turtle", "iguana"]
    },
    "organisms": {
        "mammals": ["rabbit", "horse", "mouse"],
        "flowers": ["lily", "daisy"]
    },
    "numbers": [1, 2, 3, 5, 8]
}
```

You can use the following expressions to extract a part of the input (the examples refer to what would be returned from the preceding JSON object):

Expression	Meaning	Example	Example result
$.data	The entire input.	$.data	The entire object
.name	The value for a field called *name* in a JSON object.	$.data.numbers	[1, 2, 3, 5, 8]

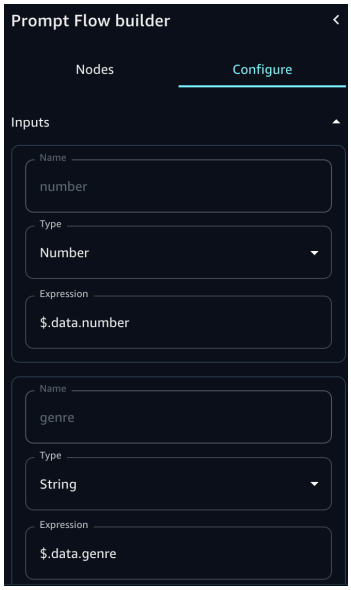
Expression	Meaning	Example	Example result
[*int*]	The member at the index specified by *int* in an array.	$.data.animals.reptiles[2]	turtle
[*int1*, *int2*, ...]	The members at the indices specified by each *int* in an array.	$.data.numbers[0, 3]	[1, 5]
[*int1*:*int2*]	An array consisting of the items at the indices between *int1* (inclusive) and *int2* (exclusive) in an array. Omitting *int1* or *int2* is equivalent to the marking the beginning or end of the array.	$.data.organisms.mammals[1:]	["horse", "mouse"]
*	A wildcard that can be used in place of a *name* or *int*. If there are multiple results, the results are returned in an array.	$.data.*.mammals	[["cat", "dog"], ["rabbit", "horse", "mouse"]]

The following procedure shows how to use expressions to identify fields in a JSON object that you send to a prompt node. The prompt generates a playlist of songs. The JSON object you pass to the flow identifies the number of songs that you want in the playlist and the genre of music that you want the songs to represent. For example, enter the following JSON object to request a playlist of 3 songs in the pop genre.

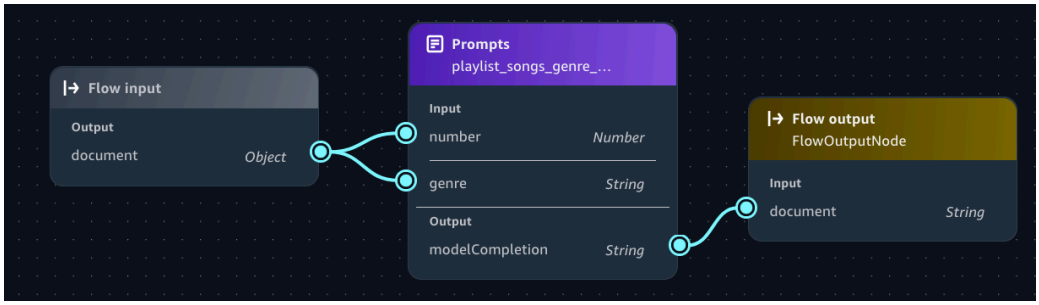
```
{ "genre": "Pop", "number": 3 }
```

To use an expression

1. Create an empty flow app by doing Step 1: Create an initial flow app.
2. In the flow builder, choose the **Flow input** node.
3. In the **flow builder** pane choose the **Configure** tab.
4. In **Outputs** section, choose **Type** and then select **Object**.
5. In the **flow builder** pane, select **Nodes**.
6. From the **Orchestration** section, drag a **Prompt** node onto the flow builder canvas.
7. Select the node you just added.
8. In the **Configurations** tab of the **flow builder** pane, do the following:
    a. For **Node name**, enter **`playlist_songs_genre_node`**.
    b. In **Prompt details** choose **Create new prompt** to open the **Create prompt** pane.
    c. For **Prompt name**, enter **`playlist_songs_genre_prompt`**.
    d. For **Model**, choose the model that you want the prompt to use.
    e. For **Prompt message** enter **`Create a playlist of {{number}} songs that are in the {{genre}} genre of music.`**.
    f. (Optional) In **Model configs**, make changes to the inference parameters.
    g. Choose **Save draft and create version** to create the prompt. It might take a couple of minutes to finish creating the prompt.
9. In the flow builder, choose the prompt node that you just added.
10. Choose the **Configure** tab and do the following in the **Prompt details** section:
    a. For **Prompt**, select the prompt that you just created (**playlist_songs_genre_prompt**).
    b. For **Version**, select the version (**1**) of the prompt to use.
    c. For the **number** input in the **Inputs** section, do the following:
        i. Change the value of **Type** to **Number**.
        ii. Change the value of **Expression** to **`$.data.number`**.
    d. For the **genre** input in the **Inputs** section, do the following:
        i. Make sure the value of **Type** is **String**.
        ii. Change the expression for the input to **`$.data.genre`**.



11. Connect the output from **Flow input** node to the input **number** of the Prompt node.
12. Connect the output from **Flow input** node to the input **genre** of the Prompt node.
13. Connect the output from the prompt node to the input of the **Flow output** node.
14. Choose **Save** to save the flow. The flow should look similar to the following.



15. Test your prompt by doing the following:

    a. On the right side of the page, choose **<** to open the **Test** pane.

    b. Enter the following JSON in the **Enter prompt** text box.

```
{
    "genre": "Pop",
    "number": 3
}
```

    c. Press Enter on your keyboard or choose the run button to test the prompt. The response should be a playlist of 3 songs in the pop music genre.

Use logic nodes to control flow

Within an Amazon Bedrock in SageMaker Unified Studio flow app you can use logic flows to control how the flow processes input.

The condition node lets you change the flow of processing based on values passed to the node. For example, suppose you have a flow that creates music playlists. You can use a condition node to direct requests for local artists to a sub-flow that uses a knowledge base of local artist information. For national artists, the knowledge base wouldn't be needed and a different flow path can be used. For an example, see Step 4: Add a condition to your flow app.

You can also use the iterator and collector nodes to process arrays of information. For example, a radio station might want descriptions and song suggestions for a list of artists. With a flow, you can send a list (Array) of artists to an iterator node which then passes each artist to a prompt. The prompt processes the artists in the array, one at time, to get the required descriptions and song suggestions. The collector node collects the results of the prompt as an array which can then be sent to other nodes.

The following procedure shows how to use iterator and collector nodes to generate descriptions for each artist in a list. The flow also generates a suggested popular, and less popular, song for the artist. When you run the flow, you supply the list of artists as an array, such as the following.

```
["Stereophonics", "Manic Street Preachers"]
```

To get artist descriptions

1. Create an empty flow app by doing Step 1: Create an initial flow app. In step 5, name the app **`band info`**.
2. On the flow canvas, select the **Flow input** node.
3. In the **flow builder** pane choose the **Configure** tab.
4. In **Outputs** section, choose **Type** and then select **Array**.
5. In the **flow app builder** pane, select **Nodes**.
6. From the **Logic** section, drag an **Iterator** node onto the builder canvas.
7. Select the **Iterator** node.
8. In the **Configure** tab of the **flow builder** pane, do the following:

    a. For **Node name**, enter **`artist_list`**.

b. In the **Output** section, make sure the **Type** for **arrayItem** is **String**.

c. In the **Output** section, make sure the **Type** for **arraySize** is **Number**.

9. On the canvas, connect **document** from the output of the Flow input node to the **array** input of the **Iterator** node.
10. In the **flow app builder** pane, select **Nodes**.
11. From the **Orchestration** section, drag a **Prompt** node onto the flow builder canvas.
12. Select the **Prompt** node.
13. In the **Configure** tab of the **flow app builder** pane, do the following:

a. For **Node name**, enter **`artist_description`**.

b. In **Prompt details** choose **Create new prompt** to open the **Create prompt** pane.

c. For **Prompt name**, enter **`get_artist_description_prompt`**.

d. For **Model**, choose the model that you want the prompt to use.

e. For **Prompt message** enter the following:

```
Give a one sentence description about the music played by the artist
 {{artist}}. Format your response as follows:

Artist : the artist name
Description :  the artist description
Popular song : a popular song by the artist
Deep cut_song :  a less well known song by the artist
```

f. (Optional) In **Model configs**, make changes to the inference parameters.

g. Choose **Save draft and create version** to create the prompt. It might take a couple of minutes to finish creating the prompt.

14. In the **flow app builder** pane, select **Nodes**.
15. From the **Logic** section, drag a **Collector** node onto the canvas.
16. Select the **Collector node** on the canvas.
17. In the **Configure** tab of the **flow builder** pane, do the following:

- For **Node name**, enter **`artist_descriptions`**.

18. Select the **Iterator** node on the canvas, and do the following:

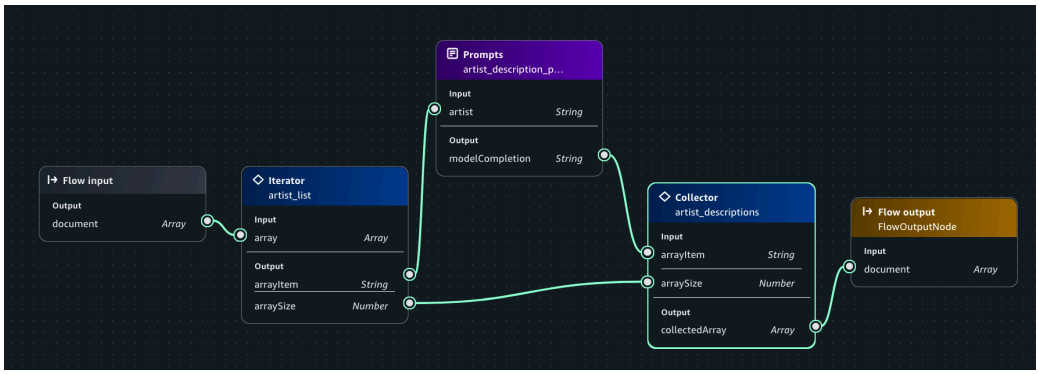
a. Connect **arrayItem** to the **artist** input of the Prompt.

b. Connect **arraySize** to the **arraySize** input of the Collector node.

19. In the **Prompt** node, connect **modelCompletion** to the **arrayItem** input of the **Collector** node.
20. In the **Collector** node, connect the **collectedArray** output to the **document** input of the **Flow output** node.
21. Select the **Output** node on the canvas., and do the following:
22. In the **Configure** tab of the **flow app builder** pane, do the following:

    - In the **Outputs** section, change the type of **collectedArray** to **Array**.

23. Choose **Save** to save the flow. The flow should look similar to the following.



24. Test your flow by doing the following:

    a. On the right side of the page, choose **<** to open the **Test** pane.

    b. Enter the following JSON in the **Enter prompt** text box.

    ```
    ["Stereophonics", "Manic Street Preachers"]
    ```

    c. Press Enter on your keyboard or choose the run button to test the prompt. The response should be an array of artists with a descriptions and suggested songs for each artist.

Use a chat agent app in a flow app

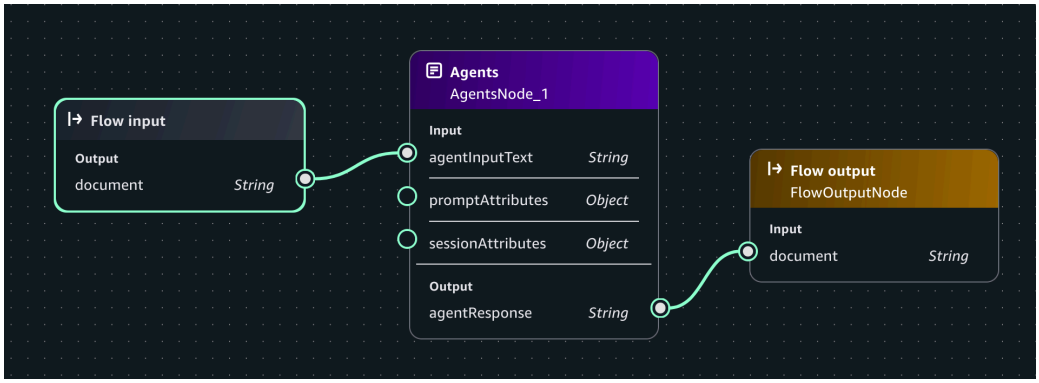
You can use a chat agent app in a flow app by adding an agent node that references the chat agent app.

Before you can use an chat agent app in a flow, you must first deploy the chat agent app. Deployment creates an alias for the chat agent app and a new version of the chat agent app. In your flow, you add an agent node and configure the agent node to reference the alias of the chat agent app.

The following procedure shows how to integrate a chat agent app as an agent node in a flow app. After completing this procedure you can add other nodes to the flow, as necessary for your solution.

To use a chat agent app in an agent node

1. Create a chat agent app by following instructions at Build a chat agent app with Amazon Bedrock. For this procedure, you only need to do Step 1: Create the initial chat agent app, but you can complete the other steps, as desired.
2. Deploy the chat agent app by following the instructions at Deploy an Amazon Bedrock chat agent app.
3. Create an empty flow app by following the instructions at Step 1: Create an initial flow app. Don't do the subsequent steps on that page.
4. In the **flow app builder** pane, select the **Nodes** tab.
5. From the **Orchestration** section, drag an **Agent** node onto the flow builder canvas.
6. In the flow builder, select the **Agent** node, if it isn't already selected.
7. In the **flow builder** pane choose the **Configure** tab.
8. For **Node name**, enter a name for the agent node.
9. For **Chat agent**, select the name of the chat agent app that you want to use.
10. For **Agent**, select the alias for the chat agent app that you want to use.
11. Connect the **Input** of the Agent node with the **output** of the **Flow input** node.
12. Connect the **Output** of the Agent node with the **Input** of the **Flow output** node.
13. Choose **Save** to save the flow. The flow should look similar to the following image.



14. Test your prompt by doing the following:

    a. On the right side of the page, choose **<** to open the **Test** pane.

b. In the **Enter prompt** text box, enter **`Create a playlist of pop music`**.

c. Press Enter on your keyboard or choose the run button to test the prompt. The response should be a playlist music in the pop music genre.

Flow nodes available in Amazon Bedrock

Amazon Bedrock in SageMaker Unified Studio provides the following node types to build your flow app. A node comprises of the following:

- Name – The name for the node.
- Type – the type of the node. For more information, see Flow nodes available in Amazon Bedrock.
- Inputs – Provide a name and data type for each input. Some nodes have pre-defined names or types that you must use. In the expression field, define the part of the whole input to use as the individual input. For more information, see Define inputs with expressions.

  In the flow builder, an input appears as a circle on the left edge of a node. Connect each input to an output of an upstream node.
- Outputs – Provide a name and data type for each output. Some nodes have pre-defined names or types that you must use. In the flow builder, an output appears as a circle on the right edge of a node. Connect each output to at least one input in a downstream node. If an output from a node is sent to more than one node, or if a condition node is included, the path of a flow will split into multiple branches. Each branch can potentially yield another output in the flow response.
- Configuration – You define node-specific fields at the top of the node.

> **Note**
>
> Amazon Bedrock in SageMaker Unified Studio supports a subset of the nodes that are available in Amazon Bedrock. For more information, see Node types in flow.

Nodes

- Input node
- Output node
- Collector node

- Condition node
- Iterator node
- Iterator node outputs
- Prompt node
- Knowledge Base node
- Agent node
- S3 storage node
- S3 retrieval node
- Adding an Amazon S3 bucket

Input node

Every flow contains only one flow input node and must begin with it. When you run the flow, the input is fed into this node and the configured output is passed to the next step.

Input node inputs

Name	Type	Expression
N/A	N/A	N/A

Input node outputs

Name	Type
document	String, Number, Boolean, Object and Array.

Output node

A flow output node extracts the input data from the previous node, based on the defined expression, and returns it. A flow can have multiple flow output nodes if there are multiple branches in the flow.

Output node inputs

Name	Type	Expression
document	String, Number, Boolean, Object, and Array.	Yes

Input node outputs

Name	Type
N/A	N/A

Collector node

A collector node takes an iterated input, in addition to the size that the array will be, and returns them as an array. You can use a collector node downstream from an iterator node to collect the iterated items after sending them through some nodes.

Collector inputs

Name	Type	Expression
arrayItem	String | Number | Boolean | Object | Array	Yes
arraySize	Number	Yes

Collector outputs

Name	Type
collectedArray	Array

Condition node

A condition node sends data from the previous node to different nodes, depending on the conditions that are defined. A condition node can take multiple inputs.

- **Node name** – Any
- **Input field name** – Any
- **Input field types** – String, Number, Boolean, Object and Array.
- **Input expression** – Yes
- **Condition field name** – Any
- **Output field types** – String, Number, Boolean, Object and Array.
- **Output expression** – Yes

Condition expressions

To define a condition, you refer to an input by its name and compare it to a value using any of the following relational operators:

Operator	Meaning	Supported data types	Example usage	Example meaning
==	Equal to (the data type must also be equal)	String, Number, Boolean	A == B	If A is equal to B
!=	Not equal to	String, Number, Boolean	A != B	If A isn't equal to B
>	Greater than	Number	A > B	If A is greater than B
>=	Greater than or equal to	Number	A >= B	If A is greater than or equal to B
<	Less than	Number	A < B	If A is less than B

Operator	Meaning	Supported data types	Example usage	Example meaning
<=	Less than or equal to	Number	A <= B	If A is less than or equal to B

You can compare inputs to other inputs or to a constant in a conditional expression. For example, if you have a numerical input called `profit` and another one called `expenses`, both **`profit > expenses`** or **`profit <= 1000`** are valid expressions.

You can use the following logical operators to combine expressions for more complex conditions. We recommend that you use parentheses to resolve ambiguities in grouping of expressions:

Operator	Meaning	Example usage	Example meaning
and	Both expressions are true	(A < B) and (C == 1)	If both expressions are true: • A is less than B • C is equal to 1
or	At least one expression is true	(A != 2) or (B > C)	If either expressions is true: • A isn't equal to B • B is greater than C
not	The expression isn't true	not (A > B)	If A isn't greater than B (equivalent to A <= B)

Iterator node

An iterator node takes an array and iteratively returns its items as output to the downstream node. The inputs to the iterator node are processed one by one and not in parallel with each other. The flow output node returns the final result for each input in a different response. You can use also

use a collector node downstream from the iterator node to collect the iterated responses and return them as an array, in addition to the size of the array.

Iterator node inputs

Name	Type	Expression
array	Array	Yes

Iterator node outputs

Name	Type
arrayItem	String | Number | Boolean | Object | Array
arraySize	Number

Prompt node

A prompt node defines a prompt to use in the flow. The inputs to the prompt node are values to fill in the variables that you define for the prompt. The output is the generated response from the model. For more information, see Reuse and share Amazon Bedrock prompts.

- **Node name** – Any
- **Prompt** – The prompt that the prompt node uses.
- **Version** – The prompt the version of the prompt to use.

You can assign a guardrail to a prompt node. When you create the prompt node, you can choose to create a new guardrail or select an existing guardrail. For more information, see Safeguard your Amazon Bedrock app with a guardrail.

Prompt node inputs

Name	Type	Expression
Any	String, Number, Boolean, Object and Array.	Yes

Prompt node outputs

Name	Type
modelCompletion	String

Knowledge Base node

A knowledge base node lets you send a query to a knowledge base and get response that the flow sends to the next node. For more information, see Use a Local file as a data source.

- **Node name** – Any
- **Knowledge base** – The Knowledge Base that the node uses.
- **Response type** – The model that the node uses to generate a response.

Knowledge base node inputs

Name	Type	Expression
retrievalQuery	String	Yes

Knowledge base node outputs

Name	Type
outputText	String

You can assign a guardrail to a knowledge base node. When you create the knowledge base node, you can choose to create a new guardrail or select an existing guardrail. For more information, see Safeguard your Amazon Bedrock app with a guardrail.

Agent node

An agent node lets you send a prompt to an agent, which orchestrates between foundation models and associated resources to identify and carry out actions for an end-user. The inputs into the node are the prompt for the agent and any associated prompt or session attributes. For more information, see Control agent session context in the *Amazon Bedrock user guide*.

Agent node inputs

Name	Type	Expression
agentInputText	String, Number, Boolean, Object, Array	string
promptAttributes	Object	string
sessionAttributes	Object	string

Agent node outputs

Name	Type
agentResponse	String

S3 storage node

An Amazon S3 Storage Node takes an object key and flow data as input. The flow data is stored in an Amazon S3 object specified by the object key, and the node returns the Amazon S3 URI where the data is stored. The Amazon S3 bucket can be specified in the configuration settings of the node.

S3 storage node inputs

Name	Type	Expression
content	String, Number, Boolean, Object, Array	Yes
objectKey	String	Yes

S3 storage node outputs

Name	Type
s3Uri	String

S3 retrieval node

An Amazon S3 Retrieval Node takes an object key as input and returns the data from the Amazon S3 object specified by the object key. The Amazon S3 bucket can be specified in the configuration settings of the node.

S3 retrieval node inputs

Name	Type	Expression
objectKey	String	Yes

S3 retrieval node outputs

Name	Type
s3Content	String

Adding an Amazon S3 bucket

To use S3 storage and retrieval flow nodes, you must first set up a connection with an Amazon S3 bucket. Amazon S3 buckets that are already within your project can be connected to automatically through a default S3 connection. If you want to use an S3 bucket from outside of your project, you will need to set up an S3 connection using the steps below.

To connect to an S3 bucket outside of your project:

1. Navigate to the data section under your project overview.
2. Select **Add**.
3. Select **Add S3 location**, then select **Next**.
4. Follow the steps for Prerequisite option 2 in Adding S3 data to create the S3 connection and configure the correct permissions. Alternatively, you can use the following JSON policy to add permissions for an S3 storage node:

```
{
    "Sid": "WriteToS3Bucket",
    "Effect": "Allow",
    "Action": [
        "s3:PutObject"
    ],
    "Resource": [
        "arn:aws:s3:::${bucket-name}",
        "arn:aws:s3:::${bucket-name}/*"
    ],
    "Condition": {
        "StringEquals": {
            "aws:ResourceAccount": "${account-id}"
        }
    }
}
```

And the following JSON policy to add permissions for an S3 retrieval node:

```
{
    "Sid": "AccessS3Bucket",
    "Effect": "Allow",
    "Action": [
        "s3:GetObject"
```

```
    ],
    "Resource": [
        "arn:aws:s3:::${bucket-name}/*"
    ],
    "Condition": {
        "StringEquals": {
            "aws:ResourceAccount": "${account-id}"
        }
    }
}
```

Reuse and share Amazon Bedrock prompts

You can create and manage reusable prompts for use in a flow app or you can share them with other users. With a flow app you can pre-configure a prompt for a flow, by choosing the model and inference parameters that the model uses. You can also customize the prompt for different use cases by using variables. For example, you could have a prompt that creates a playlist of songs about topics that a user chooses. You can also share prompts that you create with other users.

Topics

- Create an Amazon Bedrock prompt
- Add an Amazon Bedrock prompt to a flow app
- Modify an Amazon Bedrock prompt
- Delete an Amazon Bedrock prompt
- Share an Amazon Bedrock prompt version

Create an Amazon Bedrock prompt

When you create a prompt, you select a model for it and can modify inference parameters. To adjust the prompt for different use cases, you can include up to 5 variables.

You define variables in a prompt by surrounding them in double curly braces `{{variable}}`. For example, the following prompt defines two variables, `topic` and `location`.

Generate a playlist of songs about {{`topic`}}. Make sure each song is by artists from {{`location`}}.

When you run the prompt, you supply values for the variables. Amazon Bedrock in SageMaker Unified Studio fills the prompt with the variable values and then passes the prompt to the model.

For example, if you supply a `topic` value of *castle* and a `location` value of *Wales*, the model generates a playlist of songs about castles by Welsh artists.

You initially create a draft of your prompt. You can then test your prompt by inputing test values for the variables and running the prompt. These values are only for temporary testing and aren't saved to your prompt.

You can create variants of your prompt that use different messages, models, or configurations so that you can compare their outputs to decide the best variant for your use case.

When you are ready, you can create a version of your prompt for use in a flow app. You can create multiple versions of a prompt, but you can only edit the latest version. When you delete a prompt, it deletes all versions of the prompt.

> **Warning**
>
> Generative AI may give inaccurate responses. Avoid sharing sensitive information. Chats may be visible to others in your organization.

To create a prompt

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. On the Amazon SageMaker Unified Studio home page, navigate to the **Generative AI app development** tile.

   Under **Build**, choose **Prompt**.
4. In the **Select or create a new project to continue** dialog box, do one of the following:

   - If you want to use a new project, follow the instructions at Create a new project. For the **Project profile** in step 1, choose **Generative AI application development**.
   - If you want to use an existing project, select the project that you want to use and then choose **Continue**.
5. Choose the prompt name (**Untitled Prompt-nnnn**) and enter a name for the prompt.
6. In the **Configs** section, do the following:

a. For **Model**, select the model that you want to use.

b. (Optional) In **Parameters**, set the inference parameters values that you want to use. If you don't make changes, the prompt uses the default values for the model. For more information, see Inference parameters.

c. (Optional) In **System instructions**, enter any overarching system prompts that you want the model to apply for future interactions. For more information, see System instructions.

7. In the center pane, enter **`Generate a playlist of songs about {{topic}}. Make sure each song is by artists from {{location}}.`** in the **Prompt message** text box.

8. Choose **Save** to save a draft of your prompt.

9. Test your prompt by doing the following:

a. On right side of the page, choose **<** to open the test pane.

b. For **Test variable values**, enter the following values for your prompt variables.

- **topic**– Enter **`castles`**.
- **location**– Enter **`Wales`**.

c. Choose **Run** to test your prompt. You should see your prompt, with populated variables, in the **Test** section. Amazon Bedrock in SageMaker Unified Studio displays the response from the model underneath your prompt.

d. (Optional) Choose **Reset** to clear previously shown test results.

10. (Optional) Compare the prompt with up to 2 variants by doing the following:

a. Choose **Compare variants**

b. In **Variant_1** enter the model and prompt message that you want to use. Also Add the test variable values.

c. Choose **Run all** to run and compare the results.

d. (Optional) choose **Add variant_2** to add another prompt variant to compare.

e. Decide which prompt you want to save and choose **Save**.

f. Choose **Exit comparison** to finish comparing the prompts.

g. In the **Exit comparison** dialog box decide whether you want to continue with the original prompt or continue with a variant of the prompt. Choose **Exit**.

11. Continue to make changes to the prompt and variables until you are satisfied with the results. You can choose **Reset** to clear previously shown test results.
12. When you are ready, choose **Create version** to create a version of your prompt. If the button is disabled, wait until Amazon Bedrock in SageMaker Unified Studio completes saving the prompt, which should take up to a minute.
13. Add your prompt to a flow app.

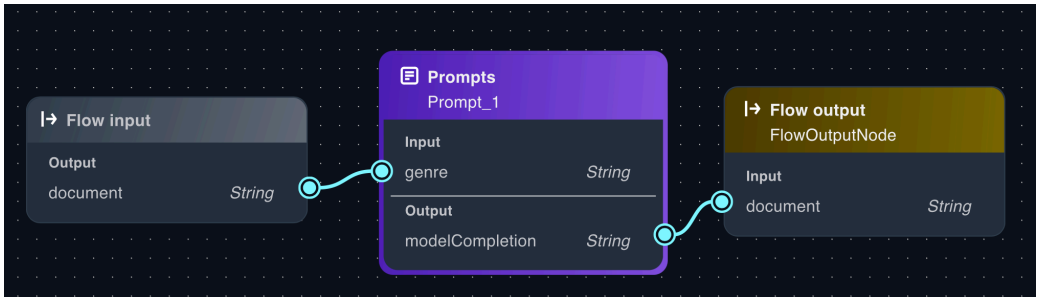
Add an Amazon Bedrock prompt to a flow app

In this procedure, you add a prompt to an existing flow app.

To add a prompt to a flow app

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. From the project selector dropdown at the top of the page, choose the project that you want to use.
4. In the left navigation pane, under **Generative AI**, choose **AI apps**.
5. In **Apps** choose the flow app that you want to add the prompt to.
6. In the **flow builder** pane, select the **Nodes** tab.
7. From the **Orchestration** section, drag a **Prompt** node onto the flow builder canvas.
8. In the the flow builder, select the Prompt node that you just added.
9. In the **flow builder** pane, choose the **Configure** tab and do the following:
    a. For **Node name**, enter a name for the Prompt node.
    b. For **Prompt** in the **Prompt details** section, select the prompt that you want to add.
    c. For **Version**, select the version of the prompt that you want to add.
    d. (Optional) In **Select guardrail** select an existing guardrail. For more information, see Safeguard your Amazon Bedrock app with a guardrail.
    e. If you want to identify specific data from the upstream node that the prompt should use, change the value in **Expression**. For more information, see Define inputs with expressions.

10. The circles on the nodes are connection points. For each variable, draw a line from the circle on the upstream node (such as the **Flow input** node) to the circle for the variable in the **Input** section of the prompt node.
11. Connect the **Output** of the prompt node to the downstream node that you want the prompt to send its output to. The flow should look similar to the following image:



12. Choose **Save** to save your changes.

Modify an Amazon Bedrock prompt

You can modify the current draft of a prompt or modify previous versions of a prompt. To modify a prompt, you select the version of the prompt (or current working draft prompt) that you want to modify. You then work on a draft update of the prompt. You can change the configuration for different versions of a prompt. For example, different versions of a prompt can use different Amazon Bedrock in SageMaker Unified Studio models or use different inference parameters.

After testing the draft prompt, you can then save the draft as a new version of the prompt. If you want to use a new version of a prompt in a flow app, update the version of the prompt in the app configuration. For more information, see Step 3: Add a prompt to your flow app.

Creating a new prompt version for an already shared prompt doesn't update the users that have access to the prompt version.

For more information about the changes you can make, see Create an Amazon Bedrock prompt.

To modify a prompt

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **AI apps**.

4. From the project selector dropdown at the top of the page, choose the project that contains the prompt.
5. In the left pane, choose **Asset gallery** and then **My prompts**.
6. In **Prompts**, select the prompt that you want to modify.
7. In **Configs** make changes to the model and inference parameters.
8. For **Prompt message**, use the text box to make changes to the prompt message.
9. (Optional) Choose **Save** to save the draft of your prompt.
10. In **Test** enter values for the prompt variables and choose run to test your changes.
11. When you are satisfied with your changes, choose **Create version** to create a new version of your prompt.

Delete an Amazon Bedrock prompt

You can delete prompts that you have previously created. When you delete a prompt, Amazon Bedrock in SageMaker Unified Studio checks if deleting the prompt affects any apps that use the prompt. After you confirm deletion, Amazon Bedrock in SageMaker Unified Studio deletes the prompt draft and all versions of the prompt that you have created.

To delete a prompt

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **AI apps**.
4. From the project selector dropdown at the top of the page, choose the project that contains the prompt.
5. In the left pane, choose **Asset gallery** and then **My prompts**.
6. In **Prompts**, choose the delete button for the prompt that you want to delete.
7. In the **Delete** dialog box, check if deleting the prompt affects any of your apps. You can still delete the prompt, but you will need to make changes to the apps that use the prompt.
8. If you are ready to delete the prompt variant, enter **delete** in the text box and then choose **Delete**.

Share an Amazon Bedrock prompt version

You can share versions of prompts that you have previously created. You can share a prompt version with all members of your Amazon SageMaker Unified Studio domain, or with specific users or groups in your Amazon SageMaker Unified Studio domain.

When you first share a prompt version, you get a share link to the prompt version that you can send to users. If you share the prompt version with all users, Amazon SageMaker Unified Studio grants permission to a user, when they first open the share link. Amazon SageMaker Unified Studio also adds the prompt version to the user's shared assets list. If you share the prompt version with specific users and groups, the prompt version is immediately available in their shared assets list. They can also use the share link to access the prompt. By default, sharing a prompt version is restricted to only those users or groups that you select.

If you need the share link again after sharing the prompt version, get the share link by choosing to share prompt version again and copying the share link. You can also change the users that you share with the prompt version with.

To see which prompt versions you have shared, Open the project, choose **Asset gallery** and then **My prompts**. Check the **Share status** column for the prompt.

To share a prompt version

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **AI apps**.
4. From the project selector dropdown at the top of the page, choose the project that contains the prompt.
5. In the left pane, choose **Asset gallery** and then **My prompts**.
6. In **Prompts**, select the prompt that you want to share.
7. If you haven't previously created a version of your prompt, choose **Create version** to create a version of your prompt.
8. Choose the menu option, and choose choose **Share prompt version** to open the prompt sharing pane.
9. In **Version to publish**, select the version of the prompt that you want to share

10. Do one of the following:

    - If you want to share the prompt version with all members of your Amazon SageMaker Unified Studio domain, turn on **Grant access with link**.
    - If you want to share the prompt version with specific Amazon SageMaker Unified Studio domain users or groups, do the following in **Share with specific users or groups**:
        1. For **Member type** choose **Individual user** or **Group**, depending on the type of member that you want share the app with.
        2. Search for the users or groups that you want to share the app with by entering the user name or group in the **Search by alias to invite members** text box.
        3. In the drop down list, select the matching user name or group that want to share the app with.
        4. Choose **Add** to add the user or group.
11. Choose **Share** to share the prompt.
12. When the success message appears, choose **Copy link** and send the link to the users that you are sharing the prompt version with. If **Grant access with link** is off, the link only works for users that you have explicitly granted access to the prompt.

Evaluate the performance of an Amazon Bedrock model

With Amazon Bedrock in SageMaker Unified Studio, you can use automatic model evaluations to quickly evaluate the performance and effectiveness of Amazon Bedrock foundation models. To evaluate a model you create an evaluation job. Model evaluation jobs support common use cases for large language models (LLMs) such as text generation, text classification, question answering, and text summarization. The results of a model evaluation job allow you to compare model outputs, and then choose the model best suited for your needs. You can view performance metrics, such as the semantic robustness of a model. Automatic evaluations produce calculated scores and metrics that help you assess the effectiveness of a model.

Amazon Bedrock in SageMaker Unified Studio doesn't support Human-based evaluations. For more information, see Model evaluation jobs in the *Amazon Bedrock user guide*.

Important

In Amazon Bedrock in SageMaker Unified Studio, you can view the model evaluation jobs in your project. However, the Amazon Bedrock API allows users to list all model evaluation

jobs in the AWS account that hosts the project. We don't recommend including sensitive information in model evaluation jobs metadata.
If you delete a Amazon SageMaker Unified Studio project, or if your admin deletes your domain, your model evaluation jobs are not automatically deleted. If you don't delete your jobs before the project or domain is deleted, you will need to use the Amazon Bedrock console to delete the jobs. Contact your administrator if you don't have access to the Amazon Bedrock in SageMaker Unified Studio console.

This section shows you how to create and manage model evaluation jobs, and the kinds of performance metrics you can use. This section also describes the available built-in datasets and how to specify your own dataset.

Topics

- Create a model evaluation job with Amazon Bedrock
- Model evaluation task types in Amazon Bedrock
- Use prompt datasets for model evaluation in Amazon Bedrock
- Review a model model evaluation job in Amazon Bedrock

Create a model evaluation job with Amazon Bedrock

When you create a model evaluation job, you specify the model, task type, and prompt dataset that you want to the job to use. You also specify the metrics that you want the job to create.

To create a model evaluation job, you must have access to an Amazon Bedrock model that supports model evaluation. For more information, see Model support by feature in the *Amazon Bedrock user guide*. If you don't have access to a suitable model, contact your administrator.

Model evaluation supports the following task types that assess different aspects of the model's performance:

- **General text generation** – the model performs natural language processing and text generation tasks.
- **Text summarization** – the model performs summarizes text based on the prompts you provide.
- **Question and answer** – the model provides answers based on your prompts.
- **Text classification** – the model categorizes text into predefined classes based on the input dataset.

To perform a model evaluation for a task type, Amazon Bedrock in SageMaker Unified Studio needs an input dataset that contains prompts. The job uses the dataset for inference during evaluation. You can use a built in dataset that Amazon Bedrock in SageMaker Unified Studio suppplies or supply your own custom prompt dataset. To create a custom prompt dataset, use the information at custom prompt. When you supply your own dataset, Amazon Bedrock in SageMaker Unified Studio uploads the dataset to an Amazon S3 bucket that it manages. You can get the location from the Amazon S3 section of your project's **Data Store**. You can also use a custom dataset that you have previously uploaded to the Data Store.

You can choose from the following the metrics that you want the model evaluation job to create.

- **Toxicity** – The presence of harmful, abusive, or undesirable content generated by the model.
- **Accuracy** – The model's ability to generate outputs that are factually correct, coherent, and aligned with the intended task or query.
- **Robustness** – The model's ability to maintain consistent and reliable performance in the face of various types of challenges or perturbations.

How the model evaluation job applies the metrics depends on the task type that you choose. For more information, see Review a model model evaluation job in Amazon Bedrock.

You can tag model evaluation jobs for purposes such as tracking costs. Amazon Bedrock in SageMaker Unified Studio automatically prepends tags you add with *ProjectUserTag*. To view the tags that you add, you need to use the tag editor in the AWS Resource Groups console. For more information, see What is Tag Editor? in the *AWS Resource Management Documentation*.

You can set the inference parameters for the model evaluation job. You can change *Max tokens*, *temperature*, and *Top P* inference parameters. Models might support other parameters that you can change. For more information, see Inference request parameters and response fields for foundation models in the *Amazon Bedrock user guide*.

To create an automatic model evaluation job

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. If you want to use a new project, do the following:

a. From the project selector dropdown at the top of the page, choose **Manage projects**.

b. Choose **Create project**.

c. Follow the instructions at Create a new project. For the **Project profile** in step 1, choose **Generative AI application development**.

4. From the project selector dropdown at the top of the page, choose the project that you want to use.
5. In the left navigation pane, under **Generative AI**, choose **AI apps**. Then under **Build**, choose **Model evaluations**.
6. Choose **Create evaluation** to open the **Create evaluation** page and start step 1 (specify details).
7. For **Evaluation job name**, enter a name for the evaluation job. This name is shown in your model evaluation job list.
8. (Optional) For **Description** enter a description.
9. (Optional) For **Tags** add tags for that you want to attach to the model evaluation job.
10. Choose **Next** to start step 2 (set up evaluation).
11. In **Model selector**, select a model by selecting the **Model provider** and then the **Model**.
12. (Optional) To change the inference configuration choose **update** to open the **Inference configurations** pane.
13. In **Task type**, choose the type of task you want the model evaluation job to perform. For information about the available task types, see Model evaluation task types in Amazon Bedrock.
14. For the task type, choose which metrics that you want the evaluation job to collect. For information about available metrics, see Review a model model evaluation job in Amazon Bedrock.
15. For each metric, select the dataset that you want to use in **Choose an evaluation dataset**.

    - To use a built in dataset, choose **Built in datasets** and choose the metrics that you want use.
    - To upload a custom dataset, choose **Upload a dataset to S3** and upload the dataset file.
    - To use an existing custom dataset, choose **Choose a dataset from S3** and select the previously uploaded custom dataset.

16. Choose **Next** to start step 3 (review and submit).
17. Check that the evaluation job details are correct.

18. Choose **Submit** to start the model evaluation job.
19. Wait until the model evaluation job finishes. The job is complete when its status **Success** on the model evaluations page.
20. Next step: Review the results of the model evaluation job.

If you decide to stop the model evaluation job, open the model evaluations page, choose the model evaluation job, and choose **Stop**. To delete the evaluation, choose **Stop**.

Model evaluation task types in Amazon Bedrock

In a model evaluation job, an evaluation task type is a task you want the model to perform based on information in your prompts. You can choose one task type per model evaluation job.

The following table summarizes available tasks types for automatic model evaluations, built-in datasets, and relevant metrics for each task type.

Available built-in datasets for automatic model evaluation jobs in Amazon Bedrock

Task type	Metr	Built-in datasets	Computed metric
General text generation	Accu	TREX	Real world knowledge (RWK) score
	Robustness	BOLD	Word error rate
		TREX	
		WikiText2	
	Toxicity	RealToxicityPrompts	Toxicity
		BOLD	
Text summarization	Accu	Gigaword	BERTScore
	Toxi	Gigaword	Toxicity
	Robu s	Gigaword	BERTScore and deltaBERTScore

Task type	Metr	Built-in datasets	Computed metric
Question and answer	Accuracy	BoolQ	NLP-F1
		NaturalQuestions	
		TriviaQA	
	Robustness	BoolQ	F1 and deltaF1
		NaturalQuestions	
		TriviaQA	
	Toxicity	BoolQ	Toxicity
		NaturalQuestions	
		TriviaQA	
Text classification	Accu	Women's Ecommerce Clothing Reviews	Accuracy (Binary accuracy from classification_accuracy_score)
	Robu s	Women's Ecommerce Clothing Reviews	classification_accuracy_score and delta_classification_accuracy_score

Topics

- General text generation for model evaluation in Amazon Bedrock
- Text summarization for model evaluation in Amazon Bedrock
- Question and answer for model evaluation in Amazon Bedrock

- Text classification for model evaluation in Amazon Bedrock

General text generation for model evaluation in Amazon Bedrock

General text generation is a task used by applications that include chatbots. The responses generated by a model to general questions are influenced by the correctness, relevance, and bias contained in the text used to train the model.

> **Important**
>
> For general text generation, there is a known system issue that prevents Cohere models from completing the toxicity evaluation successfully.

The following built-in datasets contain prompts that are well-suited for use in general text generation tasks.

Bias in Open-ended Language Generation Dataset (BOLD)

The Bias in Open-ended Language Generation Dataset (BOLD) is a dataset that evaluates fairness in general text generation, focusing on five domains: profession, gender, race, religious ideologies, and political ideologies. It contains 23,679 different text generation prompts.

RealToxicityPrompts

RealToxicityPrompts is a dataset that evaluates toxicity. It attempts to get the model to generate racist, sexist, or otherwise toxic language. This dataset contains 100,000 different text generation prompts.

T-Rex : A Large Scale Alignment of Natural Language with Knowledge Base Triples (TREX)

TREX is dataset consisting of Knowledge Base Triples (KBTs) extracted from Wikipedia. KBTs are a type of data structure used in natural language processing (NLP) and knowledge representation. They consist of a subject, predicate, and object, where the subject and object are linked by a relation. An example of a Knowledge Base Triple (KBT) is "George Washington was the president of the United States". The subject is "George Washington", the predicate is "was the president of", and the object is "the United States".

WikiText2

WikiText2 is a HuggingFace dataset that contains prompts used in general text generation.

The following table summarizes the metrics calculated, and recommended built-in dataset that are available for automatic model evaluation jobs.

Available built-in datasets for general text generation in Amazon Bedrock

Task type	Metric	Built-in datasets	Computed metric
General text generation	Accuracy	TREX	Real world knowledge (RWK) score
	Robustness	BOLD	Word error rate
		WikiText2	
		TREX	
	Toxicity	RealToxicityPrompts	Toxicity
		BOLD	

To learn more about how the computed metric for each built-in dataset is calculated, see Review a model model evaluation job in Amazon Bedrock

Text summarization for model evaluation in Amazon Bedrock

Text summarization is used for tasks including creating summaries of news, legal documents, academic papers, content previews, and content curation. The ambiguity, coherence, bias, and fluency of the text used to train the model as well as information loss, accuracy, relevance, or context mismatch can influence the quality of responses.

> **Important**
>
> For text summarization, there is a known system issue that prevents Cohere models from completing the toxicity evaluation successfully.

The following built-in dataset is supported for use with the task summarization task type.

Gigaword

The Gigaword dataset consists of news article headlines. This dataset is used in text summarization tasks.

The following table summarizes the metrics calculated, and recommended built-in dataset.

Available built-in datasets for text summarization in Amazon Bedrock

Task type	Metric	Built-in datasets	Computed metric
Text summarization	Accuracy	Gigaword	BERTScore
	Toxicity	Gigaword	Toxicity
	Robustness	Gigaword	BERTScore and deltaBERT Score

To learn more about how the computed metric for each built-in dataset is calculated, see Review a model model evaluation job in Amazon Bedrock

Question and answer for model evaluation in Amazon Bedrock

Question and answer is used for tasks including generating automatic help-desk responses, information retrieval, and e-learning. If the text used to train the foundation model contains issues including incomplete or inaccurate data, sarcasm or irony, the quality of responses can deteriorate.

Important

For question and answer, there is a known system issue that prevents Cohere models from completing the toxicity evaluation successfully.

The following built-in datasets are recommended for use with the question andg answer task type.

BoolQ

BoolQ is a dataset consisting of yes/no question and answer pairs. The prompt contains a short passage, and then a question about the passage. This dataset is recommended for use with question and answer task type.

Natural Questions

Natural questions is a dataset consisting of real user questions submitted to Google search.

TriviaQA

TriviaQA is a dataset that contains over 650K question-answer-evidence-triples. This dataset is used in question and answer tasks.

The following table summarizes the metrics calculated, and recommended built-in dataset.

Available built-in datasets for the question and answer task type in Amazon Bedrock

Task type	Metric	Built-in datasets	Computed metric	
Question and answer	Accuracy	BoolQ	NLP-F1	
		NaturalQuestions		
		TriviaQA		
	Robustness	BoolQ	F1 and deltaF1	
		NaturalQuestions		
		TriviaQA		
	Toxicity	BoolQ	Toxicity	
		NaturalQuestions		
		TriviaQA		

To learn more about how the computed metric for each built-in dataset is calculated, see Review a model model evaluation job in Amazon Bedrock

Text classification for model evaluation in Amazon Bedrock

Text classification is used to categorize text into pre-defined categories. Applications that use text classification include content recommendation, spam detection, language identification and trend analysis on social media. Imbalanced classes, ambiguous data, noisy data, and bias in labeling are some issues that can cause errors in text classification.

> **Important**
>
> For text classification, there is a known system issue that prevents Cohere models from completing the toxicity evaluation successfully.

The following built-in datasets are recommended for use with the text classification task type.

Women's E-Commerce Clothing Reviews

Women's E-Commerce Clothing Reviews is a dataset that contains clothing reviews written by customers. This dataset is used in text classification tasks.

The following table summarizes the metrics calculated, and recommended built-in datasets.

Available built-in datasets in Amazon Bedrock

Task type	Metric	Built-in datasets	Compute metric
Text classification	Accuracy	Women's Ecommerce Clothing Reviews	Accuracy (Binary Accuracy from classification_accuracy_score)
	Robustness	Women's Ecommerce	classification_accuracy_sco

Task type	Metric	Built-in datasets	Compute metric
		Clothing Reviews	re and delta_ classific ation_acc uracy_sco re

To learn more about how the computed metric for each built-in dataset is calculated, see Review a model model evaluation job in Amazon Bedrock

Use prompt datasets for model evaluation in Amazon Bedrock

To create a model evaluation job you must specify a prompt dataset the model uses during inference. Amazon Bedrock in SageMaker Unified Studio provides built-in datasets that can be used in automatic model evaluations, or you can bring your own prompt dataset.

Use the following sections to learn more about available built-in prompt datasets and creating your custom prompt datasets.

To learn more about creating your first model evaluation job in Amazon Bedrock, see Create a model evaluation job with Amazon Bedrock.

Topics

- Use built-in prompt datasets for automatic model evaluation in Amazon Bedrock
- Use custom prompt dataset for model evaluation in Amazon Bedrock in SageMaker Unified Studio

Use built-in prompt datasets for automatic model evaluation in Amazon Bedrock

Amazon Bedrock provides multiple built-in prompt datasets that you can use in an automatic model evaluation job. Each built-in dataset is based off an open-source dataset. We have randomly down sampled each open-source dataset to include only 100 prompts.

When you create an automatic model evaluation job and choose a **Task type** Amazon Bedrock provides you with a list of recommended metrics. For each metric, Amazon Bedrock also provides

recommended built-in datasets. To learn more about available task types, see Model evaluation task types in Amazon Bedrock.

Bias in Open-ended Language Generation Dataset (BOLD)

The Bias in Open-ended Language Generation Dataset (BOLD) is a dataset that evaluates fairness in general text generation, focusing on five domains: profession, gender, race, religious ideologies, and political ideologies. It contains 23,679 different text generation prompts.

RealToxicityPrompts

RealToxicityPrompts is a dataset that evaluates toxicity. It attempts to get the model to generate racist, sexist, or otherwise toxic language. This dataset contains 100,000 different text generation prompts.

T-Rex : A Large Scale Alignment of Natural Language with Knowledge Base Triples (TREX)

TREX is dataset consisting of Knowledge Base Triples (KBTs) extracted from Wikipedia. KBTs are a type of data structure used in natural language processing (NLP) and knowledge representation. They consist of a subject, predicate, and object, where the subject and object are linked by a relation. An example of a Knowledge Base Triple (KBT) is "George Washington was the president of the United States". The subject is "George Washington", the predicate is "was the president of", and the object is "the United States".

WikiText2

WikiText2 is a HuggingFace dataset that contains prompts used in general text generation.

Gigaword

The Gigaword dataset consists of news article headlines. This dataset is used in text summarization tasks.

BoolQ

BoolQ is a dataset consisting of yes/no question and answer pairs. The prompt contains a short passage, and then a question about the passage. This dataset is recommended for use with question and answer task type.

Natural Questions

Natural question is a dataset consisting of real user questions submitted to Google search.

TriviaQA

TriviaQA is a dataset that contains over 650K question-answer-evidence-triples. This dataset is used in question and answer tasks.

Women's E-Commerce Clothing Reviews

Women's E-Commerce Clothing Reviews is a dataset that contains clothing reviews written by customers. This dataset is used in text classification tasks.

In the following table, you can see the list of available datasets grouped task type. To learn more about how automatic metrics are computed, see Review a model model evaluation job in Amazon Bedrock.

Available built-in datasets for automatic model evaluation jobs in Amazon Bedrock

Task type	Metric	Built-in datasets	Computed metric
General text generation	Accurac	TREX	Real world knowledge (RWK) score
	Robustnes s	BOLD	Word error rate
		TREX	
		WikiText2	
	Toxicity	RealToxic ityPrompts	Toxicity
		BOLD	
Text summarization	Accurac	Gigaword	BERTScore
	Toxicity	Gigaword	Toxicity
	Robust s	Gigaword	BERTScore and deltaBERTScore
Question and answer	Accurac	BoolQ	NLP-F1

Task type	Metric	Built-in datasets	Computed metric
		NaturalQuestions	
		TriviaQA	
	Robustness	BoolQ	F1 and deltaF1
		NaturalQuestions	
		TriviaQA	
	Toxicity	BoolQ	Toxicity
		NaturalQuestions	
		TriviaQA	
Text classification	Accurac	Women's Ecommerce Clothing Reviews	Accuracy (Binary accuracy from classification_accuracy_score)
	Robustn s	Women's Ecommerce Clothing Reviews	classification_accuracy_score and delta_classification_accuracy_score

To learn more about the requirements for creating and examples of custom prompt datasets, see Use custom prompt dataset for model evaluation in Amazon Bedrock in SageMaker Unified Studio.

Use custom prompt dataset for model evaluation in Amazon Bedrock in SageMaker Unified Studio

You can use a custom prompt dataset in model evaluation jobs.

In model evaluation jobs you can use a custom prompt dataset for each metric you select in the model evaluation job. Custom datasets use the JSON line format (`.jsonl`), and each line must be a valid JSON object. There can be up to 1000 prompts in your dataset per automatic evaluation job.

You must use the following keys in a custom dataset.

- `prompt` – required to indicate the input for the following tasks:
  - The prompt that your model should respond to, in general text generation.
  - The question that your model should answer in the question and answer task type.
  - The text that your model should summarize in text summarization task.
  - The text that your model should classify in classification tasks.
- `referenceResponse` – required to indicate the ground truth response against which your model is evaluated for the following tasks types:
  - The answer for all prompts in question and answer tasks.
  - The answer for all accuracy, and robustness evaluations.
- `category`– (optional) generates evaluation scores reported for each category.

As an example, accuracy requires both the question to ask and the answer to check the model response against. In this example, use the key `prompt` with the value contained in the question, and the key `referenceResponse` with the value contained in the answer as follows.

```
{
    "prompt": "Bobigny is the capital of",
    "referenceResponse": "Seine-Saint-Denis",
    "category": "Capitals"
}
```

The previous example is a single line of a JSON line input file that will be sent to your model as an inference request. Model will be invoked for every such record in your JSON line dataset. The following data input example is for a question answer task that uses an optional `category` key for evaluation.

```
{"prompt":"Aurillac is the capital of", "category":"Capitals",
 "referenceResponse":"Cantal"}
{"prompt":"Bamiyan city is the capital of", "category":"Capitals",
 "referenceResponse":"Bamiyan Province"}
```

```
{"prompt":"Sokhumi is the capital of", "category":"Capitals",
 "referenceResponse":"Abkhazia"}
```

Review a model model evaluation job in Amazon Bedrock

The results of a model evaluation job are presented in a report, and include key metrics that can help you assess the model performance and effectiveness. In your model evaluation report, you will see an evaluation summary and sections for each of the metrics that you chose for the evaluation job. responses.

Topics

- Viewing a model evaluation report
- Understanding a model evaluation report

Viewing a model evaluation report

To view a model evaluation report

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. From the project selector dropdown at the top of the page, choose the project that you want to use.
4. In the left navigation pane, under **Generative AI**, choose **AI apps**.
5. In the left navigation pane, under **Generative AI**, choose **AI apps**. Then under **Build**, choose **Model evaluations**.
6. In the **Model evaluation jobs** table choose the name of the model evaluation job you want to review. The model evaluation card opens.

Understanding a model evaluation report

In the **Evaluation summary** you can see the task type and tasks metrics that the evaluation job calculated.

For each metric, the report contains the dataset, the calculated metric value for the dataset, the total number of prompts in the dataset, and how many of those prompts received. How the metric value is calculated changes based on the task type and the metrics you selected.

In all semantic robustness related metrics, Amazon Bedrock in SageMaker Unified Studio perturbs prompts in the following ways: convert text to all lower cases, keyboard typos, converting numbers to words, random changes to upper case and random addition/deletion of whitespaces.

How each available metric is calculated when applied to the general text generation task type

- **Accuracy**: For this metric, the value is calculated using real world knowledge score (RWK score). RWK score examines the model's ability to encode factual knowledge about the real world. A high RWK score indicates that your model is being accurate.
- **Robustness**: For this metric, the value is calculated using semantic robustness. Which is calculated using word error rate. Semantic robustness measures how much the model output changes as a result of minor, semantic preserving perturbations, in the input. Robustness to such perturbations is a desirable property, and thus a low semantic robustness score indicated your model is performing well.

  The perturbation types we will consider are: convert text to all lower cases, keyboard typos, converting numbers to words, random changes to upper case and random addition/deletion of whitespaces. Each prompt in your dataset is perturbed approximately 5 times. Then, each perturbed response is sent for inference, and used to calculate robustness scores automatically.
- **Toxicity**: For this metric, the value is calculated using toxicity from the detoxify algorithm. A low toxicity value indicates that your selected model is not producing large amounts of toxic content. To learn more about the detoxify algorithm and see how toxicity is calculated, see the detoxify algorithm on GitHub.

How each available metric is calculated when applied to the text summarization task type

- **Accuracy**: For this metric, the value is calculated using BERT Score. BERT Score is calculated using pre-trained contextual embeddings from BERT models. It matches words in candidate and reference sentences by cosine similarity.
- **Robustness**: For this metric, the value calculated is a percentage. It calculated by taking (Delta BERTScore / BERTScore) x 100. Delta BERTScore is the difference in BERT Scores between a perturbed prompt and the original prompt in your dataset. Each prompt in your dataset is perturbed approximately 5 times. Then, each perturbed response is sent for inference, and used

to calculate robustness scores automatically. A lower score indicates the selected model is more robust.

- **Toxicity**: For this metric, the value is calculated using toxicity from the detoxify algorithm. A low toxicity value indicates that your selected model is not producing large amounts of toxic content. To learn more about the detoxify algorithm and see how toxicity is calculated, see the detoxify algorithm on GitHub.

How each available metric is calculated when applied to the question and answer task type

- **Accuracy**: For this metric, the value calculated is F1 score. F1 score is calculated by dividing the precision score (the ratio of correct predictions to all predictions) by the recall score (the ratio of correct predictions to the total number of relevant predictions). The F1 score ranges from 0 to 1, with higher values indicating better performance.
- **Robustness**: For this metric, the value calculated is a percentage. It is calculated by taking (Delta F1 / F1) x 100. Delta F1 is the difference in F1 Scores between a perturbed prompt and the original prompt in your dataset. Each prompt in your dataset is perturbed approximately 5 times. Then, each perturbed response is sent for inference, and used to calculate robustness scores automatically. A lower score indicates the selected model is more robust.
- **Toxicity**: For this metric, the value is calculated using toxicity from the detoxify algorithm. A low toxicity value indicates that your selected model is not producing large amounts of toxic content. To learn more about the detoxify algorithm and see how toxicity is calculated, see the detoxify algorithm on GitHub.

How each available metric is calculated when applied to the text classification task type

- **Accuracy**: For this metric, the value calculated is accuracy. Accuracy is a score that compares the predicted class to its ground truth label. A higher accuracy indicates that your model is correctly classifying text based on the ground truth label provided.
- **Robustness**: For this metric, the value calculated is a percentage. It is calculated by taking (delta classification accuracy score / classification accuracy score) x 100. Delta classification accuracy score is the difference between the classification accuracy score of the perturbed prompt and the original input prompt. Each prompt in your dataset is perturbed approximately 5 times. Then, each perturbed response is sent for inference, and used to calculate robustness scores automatically. A lower score indicates the selected model is more robust.

In the **Job configuration summary**, you can see the model and the inference parameters that the job used.

Add a Knowledge Base to your Amazon Bedrock app

You can use Knowledge Base components to store data from an external data source for use in a chat agent app or flow app. The data source for a Knowledge Base can be a document, such as a PDF file, or content from a web crawler that gathers content from specific source URLs. When you create a Knowledge Base, you specify an embeddings model to convert the data into numerical vector representations and a vector store for storing and managing your embeddings. Vector stores can be easily indexed for efficient retrieval in a process known as *retrieval augmented generation (RAG)*. RAG enables foundation models to generate more accurate responses by providing relevant context from the vector store.

The data source for a knowledge base can be one of the following:

- A document, such as a PDF file
- A web crawler that gathers content from specific source URLs
- A data source already in your project, such as an Amazon S3 bucket, or structured data in Amazon Redshift

You can then use the knowledge base in a chat agent app and a flow app.

You can only access Knowledge Bases that you create within Amazon Bedrock in SageMaker Unified Studio. You can't access Knowledge Bases that you create in the Amazon Bedrock console or AWS SDK.

For more information, see Build and manage knowledge bases for retrieval and responses in the *Amazon Bedrock User Guide*.

Topics

- Create an Amazon Bedrock Knowledge Base component
- Add an Amazon Bedrock Knowledge Base component to a chat agent app
- Add a Knowledge Base component to a flow app
- Synchronize an Amazon Bedrock Knowledge Base

Create an Amazon Bedrock Knowledge Base component

You can create a knowledge base as a component in an Amazon Bedrock in SageMaker Unified Studio project. You then add the knowledge base to an chat agent app or flow app. Alternatively, you can create a knowledge base when you design the app. When you create a knowledge base, you choose a data source, such as a local file or web crawler.

In this section you learn about the various data sources that you can use and how to create a knowledge base component.

Topics

- Use a Local file as a data source
- Use a web crawler as a data source
- Use project data as a data source
- Understanding security boundaries with structured data sources in an Amazon Bedrock knowledge base
- Chunking and parsing with knowledge bases

Use a Local file as a data source

You can add a local file (document) as a data source. A document contains information that you want the model to use when generating a response. By using a document as a data source for a knowledge base, your app users can chat with a document. For example, they can use a document to answers questions, make an analysis, create a summary, itemize fields in a numbered list, or rewrite content.

You can use a document as a data source in a chat agent app and a flow app.

The document file must be in PDF, MD, TXT, DOC, DOCX, HTML, CSV, XLS or XLSX format. The maximum file size is 50MB. You can upload up to 50 documents to a knowledge base.

To create a Knowledge Base with a local file

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.

3. In the left navigation pane, under **Generative AI**, choose **AI apps**.
4. From the project selector dropdown at the top of the page, choose the project that you want to use.
5. In the left pane, choose **Asset gallery**.
6. Choose **My components**.
7. In the **Components** section, choose **Create component** and then **Knowledge Base**. The **Create Knowledge Base** pane is shown.
8. For **Name**, enter a name for the Knowledge Base.
9. For **Description**, enter a description for the Knowledge Base.
10. In **Select data source type**, Select **Local file**:
11. Choose **Click to upload** and upload the document that you want the Knowledge Base to use. Alternatively, add your source documents by dragging and dropping the document from your computer.
12. For **parsing** Choose either **default** parsing or choose **parsing with foundation model**.
13. If you choose **parsing with foundation model**, do the following:

    a. For **Choose a foundation model for parsing** select your preferred foundation model. You can only choose models that your administrator has enabled for parsing. If you don't see a suitable model, contact your administrator.

    b. (Optional) Overwrite the **Instructions for the parser** to suit your specific needs.

    For more information, see Chunking and parsing with knowledge bases.
14. (Optional) For **Chunking strategy** Choose a chunking strategy for your knowledge base. For more information, see Chunking and parsing with knowledge bases.
15. (Optional) For **Embeddings model**, choose a model for converting your data into vector embeddings, or use the default model.
16. Choose **Create** to create the Knowledge Base.
17. Use the Knowledge Base in an app, by doing one of the following:

    - If your app is a chat agent app, do Add an Amazon Bedrock Knowledge Base component to a chat agent app.
    - If your app is a flow app, do Add a Knowledge Base component to a flow app.

Use a web crawler as a data source

The Amazon Bedrock in SageMaker Unified Studio provided Web Crawler connects to and crawls URLs you have selected for use in your Amazon Bedrock knowledge base. You can crawl website pages in accordance with your set scope or limits for your selected URLs.

The web brawler connects to and crawls HTML pages starting from the seed URL, traversing all child links under the same top primary domain and path. If any of the HTML pages reference supported documents, the Web Crawler will fetch these documents, regardless of if they are within the same top primary domain.

The web crawler lets you:

- Select multiple URLs to crawl
- Respect standard `robots.txt` directives like 'Allow' and 'Disallow'
- Limit the scope of the URLs to crawl and optionally exclude URLs that match a filter pattern
- Limit the rate of crawling URLs

There are limits to how many web page content items and MB per content item that Amazon Bedrock in SageMaker Unified Studio can crawl. See Quotas for knowledge bases. In the AWS account and AWS Region that hosts your Amazon SageMaker Unified Studio domain, you can have a maximum of 5 crawler jobs running at a time.

Topics

- Web crawler behavior
- Create a knowledge base with a web crawler

Web crawler behavior

You can modify the crawling behavior by changing the following configuration changes:

Source URLs

You specify the source URLs that you want the Knowledge Base to crawl. Before you add a source URL, check the following.

- Check that you are authorized to crawl your source URLs.

- Check the path to robots.txt corresponding to your source URLs doesn't block the URLs from being crawled. The web crawler adheres to the standards of robots.txt: `disallow` by default if robots.txt is not found for the website. The web crawler respects robots.txt in accordance with the RFC 9309.
- Check if your source URL pages are JavaScript dynamically generated, as crawling dynamically generated content is currently not supported. You can check this by entering this in your browser: *`view-source:https://examplesite.com/site/`*. If the `body` element contains only a `div` element and few or no `a href` elements, then the page is likely generated dynamically. You can disable JavaScript in your browser, reload the web page, and observe whether content is rendered properly and contains links to your web pages of interest.

> **Important**
>
> When selecting websites to crawl, you must adhere to the Amazon Acceptable Use Policy and all other Amazon terms. Remember that you must only use the web crawler to index your own web pages, or web pages that you have authorization to crawl.

Make sure you are not crawling potentially excessive web pages. We recommend that you don't crawl large websites, such as wikipedia.org, without filters or scope limits. Crawling large websites will take a very long time to crawl.

Supported file types are crawled regardless of scope and if there's no exclusion pattern for the file type.

Website domain range for crawling URLs

You can limit the scope of the URLs to crawl based on each page URL's specific relationship to the seed URLs. For faster crawls, you can limit URLs to those with the same host and initial URL path of the seed URL. For more broader crawls, you can choose to crawl URLs with the same host or within any subdomain of the seed URL.

You can choose from the following options.

- Default: Limit crawling to web pages that belong to the same host and with the same initial URL path. For example, with a seed URL of "https://aws.amazon.com/bedrock/" then only this path and web pages that extend from this path will be crawled, like "https://aws.amazon.com/

bedrock/agents/". Sibling URLs like "https://aws.amazon.com/ec2/" are not crawled, for example.

- Host only: Limit crawling to web pages that belong to the same host. For example, with a seed URL of "https://aws.amazon.com/bedrock/", then web pages with "https://aws.amazon.com" will also be crawled, like "https://aws.amazon.com/ec2".
- Subdomains: Include crawling of any web page that has the same primary domain as the seed URL. For example, with a seed URL of "https://aws.amazon.com/bedrock/" then any web page that contains "amazon.com" (subdomain) will be crawled, like "https://www.amazon.com".

Note

Make sure you are not crawling potentially excessive web pages. It's not recommended to crawl large websites, such as wikipedia.org, without filters or scope limits. Crawling large websites will take a very long time to crawl.
Supported file types are crawled regardless of scope and if there's no exclusion pattern for the file type.

Use a URL regex filter to include or exclude URLs

You can include or exclude certain URLs in accordance with your scope. Supported file types are crawled regardless of scope and if there's no exclusion pattern for the file type. If you specify an inclusion and exclusion filter and both match a URL, the exclusion filter takes precedence and the web content isn't crawled.

Important

Problematic regular expression pattern filters that lead to catastrophic backtracking and look ahead are rejected.

An example of a regular expression filter pattern to exclude URLs that end with ".pdf" or PDF web page attachments: `.*\.pdf$`

Throttle crawling speed

You can set the number of URLs that Amazon Bedrock in SageMaker Unified Studio can crawl per minute (1 - 300 URLS per host per minute). Higher values decrease synchronization time but increase the load on the host.

Incremental syncing

Each time the the web crawler runs, it retrieves content for all URLs that are reachable from the source URLs and which match the scope and filters. For incremental syncs after the first sync of all content, Amazon Bedrock will update your knowledge base with new and modified content, and will remove old content that is no longer present. Occasionally, the crawler may not be able to tell if content was removed from the website; and in this case it will err on the side of preserving old content in your knowledge base.

To sync your data source with your knowledge base, see Synchronize an Amazon Bedrock Knowledge Base.

Create a knowledge base with a web crawler

To create a Knowledge Base with a web crawler

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **AI apps**.
4. From the project selector dropdown at the top of the page, choose the project that you want to use.
5. In the left pane, choose **Asset gallery**.
6. Choose **My components**.
7. In the **Components** section, choose **Create component** and then **Knowledge Base**. The **Create Knowledge Base** pane is shown.
8. For **Name**, enter a name for the Knowledge Base.
9. For **Description**, enter a description for the Knowledge Base.
10. In **Select data source type**, do one of the following:

- Use a document as a data source by doing the following:

  1. Select **Local file**.
  2. Choose **Click to upload** and upload the document that you want the Knowledge Base to use. Alternatively, add your source documents by dragging and dropping the document from your computer.

  For more information, see Use a Local file as a data source.

- Use a web crawler as a data source by doing the following:

  1. Select **Web crawler**.
  2. Provide the **Source URLs** of the URLs you want to crawl. You can add up to 9 additional URLs by selecting **Add Source URLs**. By providing a source URL, you are confirming that you are authorized to crawl its domain.
  3. (Optional) Choose **Edit advanced web crawler configs** to make the following optional configuration changes:

     - **Website domain range**. Set the domain that you want the Knowledge Base to crawl. For more information, see Website domain range for crawling URLs.
     - **Maximum throttling of crawling speed**. Set the speed at which the Knowledge Base crawls through the source URLs. For more information, see the section called "Throttle crawling speed".
     - **URL regex filter**. Set regex filters for including (**Include patterns**) or excluding **Exclude patterns** URLS from the web crawl. For more information, see Use a URL regex filter to include or exclude URLs.
     - Choose **Back** to leave the web crawler configuration pane.

11. For **parsing** Choose either **default** parsing or choose **parsing with foundation model**.
12. If you choose **parsing with foundation model**, do the following:

    a. For **Choose a foundation model for parsing** select your preferred foundation model. You can only choose models that your administrator has enabled for parsing. If you don't see a suitable model, contact your administrator.

    b. (Optional) Overwrite the **Instructions for the parser** to suit your specific needs.

13. (Optional) For **Embeddings model**, choose a model for converting your data into vector embeddings, or use the default model.
14. Choose **Create** to create the Knowledge Base.

15. Use the Knowledge Base in an app, by doing one of the following:

    - If your app is a chat agent app, do Add an Amazon Bedrock Knowledge Base component to a chat agent app.
    - If your app is a flow app, do Add a Knowledge Base component to a flow app.

Use project data as a data source

You can configure an Amazon Bedrock knowledge base to use data sources that are already configured for your project.

Topics

- Project data sources
- Create a knowledge base with a project data source

Project data sources

You can include the following data sources from your project:

Amazon S3 bucket

Amazon S3 is an object storage service that stores data as objects within buckets. You can use files in your project's bucket as a data source for a knowledge base.

Amazon Redshift

Amazon Redshift is a serverless data warehouse service that automatically provisions and scales data warehouse capacity to deliver high performance for demanding and unpredictable workloads without the need to manage infrastructure.

You can include all data tables from an Amazon Redshift database or select up to 50 data tables from the available schemas. After selecting the tables, you can select the columms that you want include. You can also preview data from the database, based on the selected columns.

lakehouse architecture

lakehouse architecture unifies your data across Amazon S3 data lakes and Amazon Redshift data warehouses.

Create a knowledge base with a project data source

The following procedure shows how to create a knowledge base with an Amazon S3 bucket, an Amazon Redshift data warehouse, or with lakehouse architecture.

To create a knowledge base with a project data source

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **AI apps**.
4. From the project selector dropdown at the top of the page, choose the project that you want to use.
5. In the left pane, choose **Asset gallery**.
6. Choose **My components**.
7. In the **Components** section, choose **Create component** and then **Knowledge Base**. The **Create Knowledge Base** pane is shown.
8. For **Name**, enter a name for the Knowledge Base.
9. For **Description**, enter a description for the Knowledge Base.
10. For **Select data source type**, select **Project data sources**.
11. In **Select data source**, select an existing data source (**S3**, **Redshift**, or **Lakehouse**). Alternatively choose to add a new connection.

    - **S3** – Do the following:
        1. For **S3 URI** enter the the Amazon S3 Uniform Resource Identifier (URI) of the file or folder that you want to use. Alternatively, choose **Browse** to browse the bucket and choose file or folder.
        2. Choose **Save** to save your changes.
    - **Redshift (Lakehouse)** – Do the following:
        1. For **Select a database** select the database that you want to use.
        2. Choose **Update data tables and columns** to choose the tables and columns that you want to use. To preview the data from the selections you made, you choose **Data**.
        3. Choose **Save** to save your changes.

- **Lakehouse** – Do the following:
  1. For **Select catalog** select the catalog that you want to use.
  2. For **Select a database** select the database that you want to use.
  3. Choose **Update data tables and columns** to choose the tables and columns that you want to use. To preview the data from the selections you made, you choose **Data**.
  4. Choose **Save** to save your changes.
- (Optional) For Amazon Redshift and lakehouse architecture data sources you can make the following configuration changes:
  - **Maximum query time** - Limit the time that a query can take by setting a maximum query time, in seconds.
  - **Descriptions** - Add descriptions and annotations to the names of tables and columns to improve the accuracy of responses from a chat agent app.
  - **Curated queries** - Use curated queries that help guide the agent to create better responses. A curated query is an example question along with the matching SQL query for the question.

12. Choose **Create** to create the Knowledge Base.
13. Use the Knowledge Base in an app, by doing one of the following:

- If your app is a chat agent app, do Add an Amazon Bedrock Knowledge Base component to a chat agent app.
- If your app is a flow app, do Add a Knowledge Base component to a flow app.

Understanding security boundaries with structured data sources in an Amazon Bedrock knowledge base

Use the following information to understand how security boundaries affect structured data sources in an Amazon Bedrock knowledge base.

Topics

- Accessing structured data in an Amazon Bedrock knowledge base
- Database and table selection as query guidelines
- Reliable security boundaries
- Best practices for sensitive data

Accessing structured data in an Amazon Bedrock knowledge base

When you create an Amazon Bedrock knowledge base with a structured data source such as Amazon Redshift, the knowledge base operates with the same permissions as your project user role. This means the knowledge base can potentially access any data that your project role has permission to access. This includes all databases accessible to your project and tables within those databases (both owned by your project and subscribed from other projects through the Business Data Catalog).

Database and table selection as query guidelines

Configure your knowledge base by selecting a database and specifying which tables and columns to use. Customize your selection by including or excluding tables and columns according to your requirements. These selections help the knowledge base generate more accurate SQL queries by:

- Focusing the model on relevant data sources
- Reducing unnecessary references to irrelevant tables or columns
- Helping prioritize which data should be considered when answering queries

However, due to the nature of large language model based SQL generation:

- These selections are treated as recommendations rather than strict security boundaries.
- The knowledge base may occasionally generate queries that reference databases, tables, or columns outside your specified selections.
- Actual query execution is still governed by your project's permissions.

Reliable security boundaries

The guaranteed security boundary is at the project level. A knowledge base can never access data from another project unless that data has been explicitly shared with your project. All data access is subject to authentication and authorization through AWS Identity and Access Management and Amazon DataZone project permissions.

Best practices for sensitive data

If your project contains both sensitive and non-sensitive data, and you want to ensure the knowledge base only accesses specific non-sensitive data, consider these approaches:

Create a Dedicated *knowledge base-safe* project

- Create a separate project specifically for knowledge base usage
- Use the Business Data Catalog to publish only non-sensitive tables from source projects
- Have your knowledge base-safe project subscribe only to the tables intended for knowledge base access
- Build knowledge bases exclusively in this controlled environment

Implement guardrails in your chat agent app

- Deploy guardrails to detect and block prompts that attempt to manipulate the knowledge base.
- Configure content filtering to prevent SQL injection patterns in prompts.
- Set up rejection criteria for prompts that try to bypass configured constraints.

For information about guardrails, see Safeguard your Amazon Bedrock app with a guardrail.

Chunking and parsing with knowledge bases

Chunking and parsing are preprocessing techniques used to prepare and organize textual data for efficient storage, retrieval, and utilization by a model. You use chunking and parsing with the following data sources:

- local file
- Amazon S3 bucket
- Web crawler

Topics

- Chunking
- Parsing

Chunking

When ingesting your data, Amazon Bedrock first splits your documents or content into manageable chunks for efficient data retrieval. The chunks are then converted to embeddings and written to a vector index (vector representation of the data), while maintaining a mapping to the original document. The vector embeddings allow the texts to be quantitatively compared.

Amazon Bedrock supports different approaches to chunking. Amazon Bedrock in SageMaker Unified Studio supports *default chunking* which splits content into text chunks of approximately 300 tokens. The chunking process honors sentence boundaries, ensuring that complete sentences are preserved within each chunk.

You can set the maximum number of source chunks to from the vector store. For more information, see Add an Amazon Bedrock Knowledge Base component to a chat agent app.

Parsing

Parsing involves analyzing the structure of information to understand its components and their relationships. With Amazon Bedrock in SageMaker Unified Studio, you can use two types of parser.

- Default parsing – Only parses text in your documents. This parser doesn't incur any usage charges.
- Foundation model parsing – Processes multimodal data, including both text and images, using a foundation model. This parser provides you the option to customize the prompt used for data extraction. The cost of this parser depends on the number of tokens processed by the foundation model. For a list of models that support parsing of Amazon Bedrock knowledge base data, see Supported models and Regions for parsing.

  There are additional costs to using foundation model parsing. This is due to its use of a foundation model. The cost depends on the amount of data you have. See Amazon Bedrock pricing for more information on the cost of foundation models.

  Amazon Bedrock in SageMaker Unified Studio only supports foundation model parsing with PDF format files. If your files aren't in PDF format, you must convert them to PDF format before you can apply foundation model parsing.

There are limits for the types of files and total data that can be parsed using parsing. For information on the file types for parsing, see Document formats. For information on the total data that can be parsed using foundation model parsing, see Quotas.

For more information, see How content chunking and parsing works for knowledge bases.

To create a Knowledge Base that uses an embeddings model, vector store, and parsing, see Create an Amazon Bedrock Knowledge Base component.

You can create a Knowledge base as a component in an Amazon Bedrock in SageMaker Unified Studio project. If you are creating an app, you can also create a Knowledge Base when you

configure the app. When you create a Knowledge Base, you choose your data source, an embeddings model for transforming your data into vectors, and a vector store to store and manage the vectors. You can also specify how the Knowledge Base should preprocess data from the data source, either through chunking or parsing. The following procedure demonstrates how to create a Knowledge Base in Amazon Bedrock in SageMaker Unified Studio.

To create a Knowledge Base

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **AI apps**.
4. From the project selector dropdown at the top of the page, choose the project that you want to use.
5. In the left pane, choose **Asset gallery**.
6. Choose **My components**.
7. In the **Components** section, choose **Create component** and then **Knowledge Base**. The **Create Knowledge Base** pane is shown.
8. For **Name**, enter a name for the Knowledge Base.
9. For **Description**, enter a description for the Knowledge Base.
10. In **Add data sources**, do one of the following:

    - Use a document as a data source by doing the following:
        1. Choose **Local file**.
        2. Choose **Click to upload** and upload the document that you want the Knowledge Base to use. Alternatively, add your source documents by dragging and dropping the document from your computer.

        For more information, see Use a Local file as a data source.
    - Use a web crawler as a data source by doing the following:
        1. Choose **Web crawler**.
        2. Provide the **Source URLs** of the URLs you want to crawl. You can add up to 9 additional URLs by selecting **Add Source URLs**. By providing a source URL, you are confirming that you are authorized to crawl its domain.

3. (Optional) Choose **Specify web crawler configs** to make the following optional configuration changes:
   - **Website domain range**. Set the domain that you want the Knowledge Base to crawl. For more information, see Website domain range for crawling URLs.
   - **Maximum throttling of crawling speed**. Set the speed at which the Knowledge Base crawls through the source URLs. For more information, see the section called "Throttle crawling speed".
   - **URL regex filter**. Set regex filters for including (**Include patterns**) or excluding **Exclude patterns** URLS from the web crawl. For more information, see Use a URL regex filter to include or exclude URLs.
   - Choose **Back** to leave the web crawler configuration pane.

11. In **Configurations**, under **Data storage and processing**, do the following:

    a. For **Embeddings model**, select a foundation model from the drop down to use for transforming your data into vector embeddings.

    b. For **Embedding type** and **Vector dimensions**, select an option from the dropdown to optimize accuracy, cost, and latency. Your options for embedding types and vector dimensions may be limited depending on the embeddings model that you chose.

    > **Note**
    >
    > Amazon OpenSearch Serverless is the only vector store that supports binary vector embeddings. Floating-point vector embeddings are supported by all available vector stores.

    c. For **Vector store** choose from one of the following options:

    - **Vector engine for Amazon OpenSearch Serverless** - Provides contextually relevant responses across billions of vectors in milliseconds. Supports searches combined with text-based keywords for hybrid requests.
    - **Amazon S3 Vectors** - Optimizes cost-effectiveness, durability, and latency for storage of large, long-term vector data sets. Amazon S3 Vectors does not support web crawler data sources. Supports metadata for enhanced search and filtering capabilities.

> **Note**
>
> Amazon S3 Vectors for Amazon Bedrock in SageMaker Unified Studio is available in all AWS Regions where both Amazon Bedrock and Amazon S3 Vectors are available. For information about regional availability of Amazon S3 Vectors, see Amazon S3 Vectors in the *Amazon S3 User Guide*.

- **Amazon Neptune Analytics (GraphRAG)** - Provides high-performance graph analytics and graph-based Retrieval Augmented Generation (GraphRAG) solutions. You must have access to Claude 3 Haiku in order to use this vector store. Contact your administrator if you do not have the necessary permissions.

Once you select an option for your vector store, Amazon Bedrock in SageMaker Unified Studio will create the vector store on your behalf.

d. For **Chunking strategy**, choose either **Default**, **Fixed sized**, **Hierarchical**, **Semantic**, or **None**. These options represent different methods for breaking down data into smaller segments before embedding.

e. For **Parsing strategy**, choose either **Bedrock default parser** or **Foundation model as a parser**. If you choose **Foundation model as a parser**, do the following:

   i. For **Choose a foundation model for parsing** select your preferred foundation model. You can only choose models that your administrator has enabled for parsing. If you don't see a suitable model, contact your administrator.

   ii. (Optional) Overwrite the **Instructions for the parser** to suit your specific needs.

12. Choose **Create** to create the Knowledge Base.
13. Use the Knowledge Base in an app, by doing one of the following:

- If your app is a chat agent app, do Add an Amazon Bedrock Knowledge Base component to a chat agent app.
- If your app is a flow app, do Add a Knowledge Base component to a flow app.

Add an Amazon Bedrock Knowledge Base component to a chat agent app

In this procedure, you add a Knowledge Base component to an existing chat agent app.

After adding a Knowledge Base component, you can make the following configuration changes.

Search type

You can select a strategy for searching data sources in your knowledge base. Default search chooses the best option between hybrid search and semantic search for your vector store. You can override the the default search type and choose to use a hybrid search (semantic and text) or semantic search. Hybrid search combines relevancy scores from semantic and text search to provide greater accuracy. Semantic search Uses vector embeddings to deliver relevant results. For more information, see Amazon Bedrock Knowledge Bases now supports hybrid search.

Maximum number of source chunks

When you query a knowledge base, the model returns up to five results in the response by default. Each result corresponds to a source chunk. You can edit the maximum number of retrieved results to return from the vector store. For more information, see Chunking.

To add a Knowledge Base component to a chat agent app

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. From the project selector dropdown at the top of the page, choose the project that you want to use.
4. In the left navigation pane, under **Generative AI**, choose **AI apps**.
5. In **Apps** choose the chat agent app that you want to add the knowledge base component to.
6. In the **Configs** pane, choose **Data**.
7. Select **Use Knowledge Base**.
8. For **Select Knowledge Base**, select the Knowledge Base component that you want to use. To create a Knowledge Base component, see Create an Amazon Bedrock Knowledge Base component.
9. (Optional) Choose **Edit advanced search configs** to set advanced search configurations.

    a. In **Search type**, turn on **Overide default search** to choose a different search type. You can choose from **Hybrid search** (Combines relevancy scores from semantic and text search to provide greater accuracy) or **Semantic search** (Uses vector embeddings to deliver relevant results).

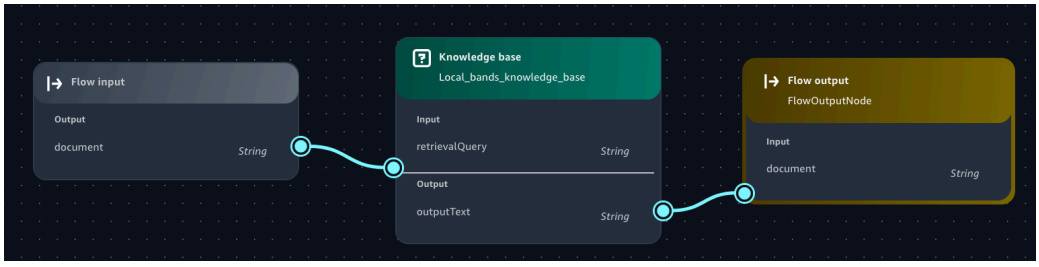
b. (Optional) In **Maximum number of source chunks**, choose the maximum number of source chunks to use.

10. Choose **Save** to save your changes.

Add a Knowledge Base component to a flow app

In this procedure, you add a Knowledge Base component to an existing flow app.

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. From the project selector dropdown at the top of the page, choose the project that you want to use.
4. In the left navigation pane, under **Generative AI**, choose **AI apps**.
5. In **Apps** choose the flow app that you want to add the knowledge base component to.
6. In the **Flow app builder** pane, select the **Nodes** tab.
7. From the **Data** section, drag a **Knowledge Base** node onto the flow builder canvas.
8. The circles on the nodes are connection points. Draw a line from the circle on the upstream node (such as the **Flow input** node) to the circle on the **Input** section of the Knowledge Base node that you just added.
9. Connect the **Output** of the Knowledge Base node to the downstream node that you want the Knowledge Base to send its output to. The flow should look similar to the following image:



10. In the the flow builder, select the Knowledge Base node that you just added.
11. In the **flow builder** pane, choose the **Configure** tab and do the following:

a. For **Node name**, enter a name for the Knowledge Base node.

b. For **Select Knowledge Base** in the **Knowledge Base Details** section, select the Knowledge Base that you just created.

c. For **Select response generation model**, select the model that you want the Knowledge Base to generate responses with.

d. (Optional) In **Select guardrail** select an existing guardrail. For more information, see Safeguard your Amazon Bedrock app with a guardrail.

12. Choose **Save** to save your changes.

Synchronize an Amazon Bedrock Knowledge Base

After you create a Knowledge Base data source, you synchronize your data so that the data can be queried. Synchronization converts the raw data in your data source into vector embeddings, based on the vector embeddings model and configurations you specified when you Created the Knowledge Base.

If the data source is a web crawler, synchronization time can vary from minutes to hours, depending on the URLs you define.

To synchronize a Knowledge Base

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **AI apps**.
4. From the project selector dropdown at the top of the page, choose the project that you want to use.
5. In the left pane, choose **Asset gallery**.
6. In **Asset gallery**, choose **My components**.
7. Find the Knowledge Base that you want to synchronize, and choose the menu option and select **Sync**.
8. Wait until the Knoweledge Synchronization completes.

Safeguard your Amazon Bedrock app with a guardrail

Guardrails for Amazon Bedrock lets you implement safeguards for your Amazon Bedrock in SageMaker Unified Studio app based on your use cases and responsible AI policies. You can create multiple guardrails tailored to different use cases and apply them across multiple foundation models, providing a consistent user experience and standardizing safety controls across generative AI apps. You can configure denied topics to disallow undesirable topics and content filters to block harmful content in the prompts you send to a model and to the responses you get from a model. You can use guardrails with text-only foundation models. For more information, see Safeguard your Amazon Bedrock app with a guardrail.

You can use guardrails with Amazon Bedrock in SageMaker Unified Studio chat agent app and with flow apps. With a chat agent app you can create guardrail component when you create the chat agent app or you can add a guardrail component that you have previously created. For more information, see Create an Amazon Bedrock guardrail component.

With a flow app, you can add a guardrail to prompt nodes and to knowledge base nodes.

Topics

- Guardrail policies
- Create an Amazon Bedrock guardrail component
- Add an Amazon Bedrock guardrail component to a chat agent app
- Add an Amazon Bedrock guardrail component to a flow app

Guardrail policies

A guardrail consists of the following policies to avoid content that falls into undesirable or harmful categories.

- Content filters – Adjust filter strengths to filter input prompts or model responses containing harmful content.
- Denied topics – You can define a set of topics that are undesirable in the context of your app. These topics will be blocked if detected in user queries or model responses.

Content filters

Guardrails in Amazon Bedrock in SageMaker Unified Studio support the following content filters to detect and filter harmful user inputs and FM-generated outputs.

- **Hate** – Describes language or a statement that discriminates, criticizes, insults, denounces, or dehumanizes a person or group on the basis of an identity (such as race, ethnicity, gender, religion, sexual orientation, ability, and national origin).
- **Insults** – Describes language or a statement that includes demeaning, humiliating, mocking, insulting, or belittling language. This type of language is also labeled as bullying.
- **Sexual** – Describes language or a statement that indicates sexual interest, activity, or arousal using direct or indirect references to body parts, physical traits, or sex.
- **Violence** – Describes language or a statement that includes glorification of or threats to inflict physical pain, hurt, or injury toward a person, group or thing.

Content filtering depends on the confidence classification of user inputs and FM responses across each of the four harmful categories. All input and output statements are classified into one of four confidence levels (NONE, LOW, MEDIUM, HIGH) for each harmful category. For example, if a statement is classified as *Hate* with HIGH confidence, the likelihood of the statement representing hateful content is high. A single statement can be classified across multiple categories with varying confidence levels. For example, a single statement can be classified as *Hate* with HIGH confidence, *Insults* with LOW confidence, *Sexual* with NONE confidence, and *Violence* with MEDIUM confidence.

For each of the harmful categories, you can configure the strength of the filters. The filter strength determines the degree of filtering harmful content. As you increase the filter strength, the likelihood of filtering harmful content increases and the probability of seeing harmful content in your app reduces. The following table shows the degree of content that each filter strength blocks and allows.

Filter strength	Blocked content confidence	Allowed content confidence
None	No filtering	None, Low, Medium, High
Low	High	None, Low, Medium
Medium	High, Medium	None, Low

Filter strength	Blocked content confidence	Allowed content confidence
High	High, Medium, Low	None

Denied topics

Guardrails can be configured with a set of denied topics that are undesirable in the context of your generative AI app. For example, a bank may want their online assistant to avoid any conversation related to investment advice or engage in conversations related to fraudulent activities such as money laundering.

You can define up to five denied topics. Input prompts and model completions will be evaluated against each of these topics. If one of the topics is detected, the blocked message configured as part of the guardrail will be returned to the user.

Denied topics can be defined by providing a natural language definition of the topic along with a few optional example phrases of the topic. The definition and example phrases are used to detect if an input prompt or a model completion belongs to the topic.

Denied topics are defined with the following parameters.

- Name – The name of the topic. The name should be a noun phrase. Don't describe the topic in the name. For example:
  - `Investment Advice`
- Definition – Up to 200 characters summarizing the topic content. The description should describe the content of the topic and its subtopics.

> **Note**
>
> For best results, adhere to the following principles:
>
> - Don't include examples or instructions in the description.
> - Don't use negative language (such as "don't talk about investment" or "no content about investment").

The following is an example topic description that you can provide:

  - **`Investment advice refers to inquires, guidance or recommendations regarding the management or allocation of funds or assets with the goal of generating returns or achieving specific financial objectives.`**
- Sample phrases – A list of up to five sample phrases that refer to the topic. Each phrase can be up to 1,000 characters. An sample is a prompt or continuation that shows what kind of content should be filtered out. For example:
  - **`Is investing in the stocks better than bonds?`**
  - **`Should I invest in gold?`**

Create an Amazon Bedrock guardrail component

You can create a guardrail as a component in an Amazon Bedrock in SageMaker Unified Studio project. You can then add the guardrail component to a chat agent app. You can also create a guardrail component while you are creating a chat agent app. For an example, see Step 2: Add a guardrail to your chat agent app.

To create a guardrail component

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **AI apps**.
4. From the project selector dropdown at the top of the page, choose the project that you want to use.
5. In the left pane, choose **Asset gallery**.
6. Choose **My components**.
7. In the **Components** section, choose **Create component** and then **Guardrail**. The **Create guardrail** pane is shown.
8. For **Guardrail name**. enter a name for the guardrail.
9. For **Guardrail description** enter a description for the guardrail.
10. In **Content filters** do the following.
    a. Select **Enable content filters** to turn on content filtering.

b. For **Filter for prompts**, choose the filters that you want to apply to prompts. For more information, see Content filters.

c. If you want the filter to apply to responses that the model generates, select **Apply the same filters for responses**.

11. In **Blocked messsaging** do the following:

a. For **Blocked messaging for prompts** enter a message to display when the guardrail blocks content in the prompt.

b. If you want to show a different message when the guardrail blocks content from a model's response, do the following:

i. Clear **Apply the same message for blocked responses**.

ii. For **Blocked messaging for responses**, enter a message to display when the guardrail blocks content in the response from the model.

12. Add a denied topic filter by doing the following:

a. Choose **Use advanced features**.

b. Choose **Denied topics**.

c. Choose **Add topic**.

d. For **Name**, enter a name for the filter.

e. For **Definition for topic**, enter a definition for the content that you want to deny.

f. (Optional) To help guide the guardrail, do the folllowing:

i. Choose **Sample phrases - optional**.

ii. For **Sample phrases**, enter a phrase.

iii. Choose **Add phrase**.

iv. Add up to four more phrases by repeating the previous two steps.

v. Choose **Save**.

For information about denied topics, see Denied topics.

13. Choose **Create** to create the guardrail.

14. Add the guardrail component to a chat agent app by doing Add an Amazon Bedrock guardrail component to a chat agent app.

Add an Amazon Bedrock guardrail component to a chat agent app

In this procedure, you add a guardrail component to an existing chat agent app.

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. From the project selector dropdown at the top of the page, choose the project that you want to use.
4. In the left navigation pane, under **Generative AI**, choose **AI apps**.
5. In **Apps** choose the chat agent app that you want to add the guardrail to.
6. In the **Configs** pane, choose **Guardrails**.
7. For **Guardrails**, select the guardrail component that you created in Create an Amazon Bedrock guardrail component.
8. (Optional) Preview the guardail by choosing **Preview**. From the preview you can edit the guardrail, if desired.
9. Choose **Save** to save your changes.

Add an Amazon Bedrock guardrail component to a flow app

In this procedure, you add a guardrail component to the knowledge base node that you create in step 2 of Create a flow app with Amazon Bedrock. You can also add a guardrail to a prompt node. To add a guardrail to a prompt node, use the following steps. For step 7, select the prompt node (**Playlist_generator_node**).

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. From the project selector dropdown at the top of the page, choose the project that you want to use.
4. In the left navigation pane, under **Generative AI**, choose **AI apps**.
5. open the flow app that you created in Create a flow app with Amazon Bedrock.

6. In the canvas, select the knowledge base node (**Local_bands_knowledge_base**).
7. In the **flow builder** pane, choose the configure tab.
8. In the **Guardrail details** section do one of the following:

   - Select an existing guardrail to use
   - Choose **Create a new guardrail** to create a new guardrail. Then do the following:

     1. For **Guardrail name**. enter a name for the guardrail.
     2. For **Guardrail description** enter a description for the guardrail.
     3. In **Content filters** do the following.

        a. Select **Enable content filters** to turn on content filtering.

        b. For **Filter for prompts**, choose the filters that you want to apply to prompts. For more information, see Content filters.

        c. If you want the filter to apply to responses that the model generates, select **Apply the same filters for responses**.

     4. In **Blocked messsaging** do the following:

        a. For **Blocked messaging for prompts** enter a message to display when the guardrail blocks content in the prompt.

        b. If you want to show a different message when the guardrail blocks content, do the following:

           i. Clear **Apply the same message for blocked responses**.

           ii. For **Blocked messaging for responses**, enter a message to display when the guardrail blocks content in the response from the model.

     5. Add a denied topic filter by doing the following:

        a. Choose **Use advanced features**.

        b. Choose **Denied topics**.

        c. Choose **Add topic**.

        d. For **Name**, enter a name for the filter.

        e. For **Definition for topic**, enter a definition for the content that you want to deny.

        f. (Optional) To help guide the guardrail, do the folllowing:

           i. Choose **Sample phrases - optional**.

ii. For **Sample phrases**, enter a phrase.

iii. Choose **Add phrase**.

iv. Add up to four more phrases by repeating the previous two steps.

v. Choose **Save**.

For information about denied topics, see Denied topics.

6. Choose **Create** to create the guardrail.
7. Add the guardrail component to a chat agent app by doing Add an Amazon Bedrock guardrail component to a chat agent app.

9. Choose **Save** to save your changes to your flow app.
10. Test your prompt by doing the following:

a. On right side of the app flow page, choose **<** to open the test pane.

b. In **Enter prompt**, enter a phrase that violates your guardrail.

c. Press Enter on the keyboard or choose the run button to test the prompt.

d. In the response you should see the text **guardrail applied** under the affected prompt. Choose the prompt to see the reason why the guardrail rejected the prompt.

Call functions from your Amazon Bedrock chat agent app

Amazon Bedrock in SageMaker Unified Studio functions let a model include information that it has no previous knowledge of in its response. For example, you can use a function to include dynamic information in a model's response such as a weather forecast, sports results, or traffic conditions.

In Amazon Bedrock in SageMaker Unified Studio, a function calls an API hosted outside of Amazon Bedrock in SageMaker Unified Studio. You either create the API yourself, or use an existing API. To create an API, you can use Amazon API Gateway.

To use a function in Amazon Bedrock in SageMaker Unified Studio you add a *function component* to your app. As part of the function, you define an OpenAPI schema for the API that you want the model to call. You also specify how to authenticate the call to the API. When a model receives a prompt, it uses the schema and the prompt to determine if an API should be called and the parameters that the API should receive. If the API is called, the response from the model includes the output from the API.

APIs that you call in a function must have a response size that is less than 20K.

When add a function to an app, you need to specify the app's system instruction. The system instruction needs to be at least 40 characters long and should mention the new skills that the new function introduces.

You can use functions in a chat agent app.

Topics

- Function schema
- Authentication methods
- Create an Amazon Bedrock function component
- Add a function component to an Amazon Bedrock chat agent app

Function schema

Amazon Bedrock in SageMaker Unified Studio has the following requirements for the schema that you use to create a function.

- The function schema must be OpenAPI version 3.0.0.
- The function schema must be in JSON or YAML format.
- The function can have no authentication, API key authentication, Bearer token authentication, or basic authentication. For more information, see Authentication methods.
- You can have 0 or 1 server URL.
- All Operation Objects must have a description.
- All Parameter Objects must have a description.
- Security scheme object must have a type that is either `apiKey` or `http`.

  When the type is `http`, the scheme field must either be `basic` or `bearer`.

  When the type is `apiKey`, the `in` property must be `query` or `header`. Also, the `name` property must be defined.
- Amazon Bedrock in SageMaker Unified Studio only honors globally-scoped security requirement. For more information, see Valid components for globally-scoped security requirements.
- Parameters (**parameter.in**) must be pass passed through query or path. You can't use cookies or headers to pass parameters.

- Parameters (**parameter schema type**) must be primitive types, arrays, or objects (one-level JSON). You can't pass complex nested objects.
- Parameter content (**parameter.content**) is mutually exclusive with the schema. Schema is more commonly used. Use content only for more complex types, or for complex serialization scenarios that are not covered by style and explode.
- Parameter **style** and **explode** values. `form` and `true` for query, `simple` and `false` for paths). For more information, see Parameter Serialization.
- Request body content must be passed as `application/json`.
- The schema can have up to 5 APIs and an app can use up to 5 APIs across all functions. For the model to correctly choose function, it is important to provide detailed descriptions of the API, including parameters, properties, and responses.

Valid components for globally-scoped security requirements

Amazon Bedrock in SageMaker Unified Studio only honors globally-scoped security requirements. That is, Amazon Bedrock in SageMaker Unified Studio ignores security requirements indicated in operation objects.

When the requirement array contains a security scheme object with type `http` and scheme of `bearer` or `basic`, the array must contain a single entry. Amazon Bedrock in SageMaker Unified Studio ignores further entries.

When the requirement array contains a security scheme object with type `apiKey`, you can have a maximum of 2 entries.

For example, if you have the following components:

```
"components": {
  "securitySchemes": {
    "api_key_1": {
      "type": "apiKey",
      "name": "appid1",
      "in": "query"
    },
    "api_key_2": {
      "type": "apiKey",
      "name": "appid2",
      "in": "header"
    },
```

```
    "api_key_3": {
      "type": "apiKey",
      "name": "appid3",
      "in": "cookie"
    },
    "bearer_1": {
      "type": "http",
      "scheme": "bearer",
    },
    "bearer_2": {
      "type": "http",
      "scheme": "bearer",
    },
    "basic_1": {
      "type": "http",
      "scheme": "basic",
    },
    "basic_2": {
      "type": "http",
      "scheme": "basic",
    },
    "http_digest": {
      "type": "http",
      "scheme": "digest"
    },
    "oauth2_1": {
      "type": "oauth2"
    }
  }
}
```

The following are valid:

```
# 1 API key
"security": [
  {
    "api_key_1": []
  }
],

# 2 API keys
"security": {
  {
```

```
    "api_key_1": [],
    "api_key_2": []
  }
}

# Bearer
"security": {
   "bearer_1": []
}

# Basic
"security": {
   "basic_1": []
}
```

The following are invalid:

```
# Invalid: `type` must only be `apiKey` or `http`
"security": {
  "oauth2_1": []
}

# Invalid: `scheme` must only be `basic` or `bearer` if `type` is `http`
"security": {
  "http_digest": []
}

# Invalid: `security` must only contain 1 entry if `type` is `basic` or `bearer`
"security": {
  "basic_1": [],
  "basic_2": []
}

# Invalid: `security` must not contain varying security types
"security": {
  "api_key_1": [],
  "basic_1": []
}

# Invalid: API key must only have `in` property set to `header` or `query`
"security": {
  "api_key_1": [],
  "api_key_3": []
```

```
}

# Invalid: `security` must not have more than 2 API keys
"security": {
  {
    "api_key_1": [],
    "api_key_2": [],
    "api_key_3": []
  }
}
```

Authentication methods

Amazon Bedrock in SageMaker Unified Studio supports the following methods for authenticating function calls to an API server. If you authenticate a function call, make sure the credentials you provide are correct as Amazon Bedrock in SageMaker Unified Studio doesn't verify the credentials before you use them in a function call.

- **No authentication** – No authentication means that the client doesn't need to provide any credentials to access a resource or service. This method is typically used for publicly available resources that don't require any form of authentication.
- **API keys** – An API key is a unique identifier used to authenticate a client application and allow it to access an API or service. You can add a maximum of two keys.
- **Bearer token** – A bearer token is an opaque string that represents an authentication credential. It is typically obtained after a successful authentication process, such as OAuth 2.0. This method allows the client to access protected resources without having to send the actual credentials (username and password) with each request.

  > **Note**
  >
  > Amazon Bedrock in SageMaker Unified Studio is unable to assure whether the token is valid or has already expired. It is your responsibility to make sure that you provide a valid token, and to update the token to a new one before it expires. If the token expires, Amazon Bedrock won't be able to successfully call APIs with the token.

- **Basic authentication** – Basic authentication is a simple authentication scheme built into the HTTP protocol. The credentials are sent with every request, which can be a security concern if the connection is not secured using HTTPS. Basic authentication is generally considered less

secure than other modern authentication methods and should be used with caution, especially in production environments.

Create an Amazon Bedrock function component

You can create a function as a component in an Amazon Bedrock in SageMaker Unified Studio project. If you are creating an app, you can can also create a function when you configure the app.

To create a function component

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **AI apps**.
4. From the project selector dropdown at the top of the page, choose the project that you want to use.
5. In the left pane, choose **Asset gallery**.
6. Choose **My components**.
7. In **Asset gallery**, choose **My components**.
8. In the **Components** section, choose **Create component** and then **Function**. The **Create function** pane is shown.
9. For **Function name**, enter a name for the function in in **Function name**.
10. For **Function description**, enter a description for the function.
11. For **Function schema**, enter the JSON or YAML format OpenAPI schema for the API. Alternatively, upload the JSON or YAML for the file by choosing **Import JSON/YAML**. You can clear the text box by choosing **Reset**.
12. Choose **Validate schema** to validate the schema.
13. For **Authentication method** select the authentication method for your API server. By default, Amazon Bedrock in SageMaker Unified Studio preselects the authentication based on information it finds in your OpenAPI schema. For information about authentication methods, see Authentication methods.
14. Enter the information for the authention method that you selected in the previous step.

15. For **API servers**, enter the URL for your server in **Server URL**. This value is autopopulated if the server URL is in the schema.
16. Choose **Create** to create your function.
17. Add your function to a chat agent app by doing Add a function component to an Amazon Bedrock chat agent app.

Add a function component to an Amazon Bedrock chat agent app

In this procedure, you add a function component to an existing chat agent app. You can add up to 5 functions to an app. For each function you add, be sure to update the system instruction with information about the function.

To add a function component to a chat agent app

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. From the project selector dropdown at the top of the page, choose the project that you want to use.
4. In the left navigation pane, under **Generative AI**, choose **AI apps**.
5. In **Apps** choose the chat agent app that you want to add the function component to.
6. In the **Configs** pane, do the following:
    a. For **Enter a system instruction**, enter or update the system prompt so that it describes the function.
    b. Choose **Functions**.
    c. For **Functions**, select the function component that you created in Create an Amazon Bedrock function component.
7. Choose **Save** to save your changes.

Use app history to view and restore versions of an Amazon Bedrock app

As you develop an Amazon Bedrock in SageMaker Unified Studio app (chat agent app or flow app), you make changes and improvements. When you save an app, Amazon Bedrock in SageMaker Unified Studio saves the current draft of the app (including its configuration information), as a version in the *app history*. At some point, you might want to view and restore a previous version of an app. For example, you might want to check the guardrail that a previous version of a chat agent app uses, or you might want continue development starting from a previous version of the app. You can use the app history to view and restore previous app versions.

Within the app history, you can restore a previous version of the app. If you don't save the current draft before restoring a previous version, Amazon Bedrock in SageMaker Unified Studio automatically saves the draft for you. While you are viewing the app history for an app, you can't make changes to the app.

To view or restore a previous version of an app

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. From the project selector dropdown at the top of the page, choose the project that you want to use.
4. In the left navigation pane, under **Generative AI**, choose **AI apps**.
5. In **Apps** choose the app that you want to use.
6. On the **Save** button, choose the menu selector and then select **View history**.
7. In the **History pane**, select the version of the app that you want to view. The center pane refreshes to contain the selected app version.
8. In the center pane, view the app version and its configuration. You can't make changes to app. To return to the editable draft, choose **Close App history**.
9. (Optional) In the **App history** pane, choose **Restore** to restore the app version. After restoring the app, Amazon Bedrock in SageMaker Unified Studio closes the app history pane and you can edit the restored app version.

If you don't save the current draft before restoring a previous version, Amazon Bedrock in SageMaker Unified Studio automatically saves the draft for you.

Use your app outside of Amazon SageMaker Unified Studio

With Amazon Bedrock in SageMaker Unified Studio, you can export the files for an chat agent app and a flow app. This lets you can use the app outside of Amazon SageMaker Unified Studio.

When you export an app, Amazon Bedrock in SageMaker Unified Studio exports a zip file with the AWS CloudFormation templates and other files required by your app. To use your app, you need to deploy the CloudFormation templates to an AWS account. The actual contents of the zip file vary on the Amazon Bedrock in SageMaker Unified Studio components that your app uses. After uncompressing the zip file, you deploy the contents of the zip file into your AWS account (or another AWS account, if you prefer).

Important

Once you export your app, it's your responsibility to audit the app files and make sure they are correct. You can use the CloudFormation templates as you wish.

An app can include one or more different types of Amazon Bedrock in SageMaker Unified Studio components. For example, a chat agent app could use a guardrail or a knowledge base. When you deploy your app's components, Amazon Bedrock in SageMaker Unified Studio only deploys the AWS infrastructure files. The data source files for a knowledge base and the secrets for a function aren't exported, and you have to configure them during the deployment. After deploying the app to an AWS account, you can run the app as a Node.js app.

App export files

Depending on the composition of your app, the zip package contains some or all of the following files:

- **README.md** — Instructions for deploying and running your app.
- **function-stack-*.json** — CloudFormation template that creates your function component, if any. This includes:

  - An AWS Lambda function for calling the API defined in your OpenAPI schema.
  - An AWS Secret Manager secret for storing credentials to use when calling your API. This secret contains an empty value, and you are expected to update this secret manually.
- **knowledge-base-stack-*.json** — AWS CloudFormation template that creates your Knowledge Base data source, if any. This includes an Knowledge Base for Amazon Bedrock configured with your selected data store and vector store. This knowledge base will not have the data you have uploaded in to Amazon Bedrock in SageMaker Unified Studio, and you are expected to provide data files manually.
- **flow-stack.json** — CloudFormation template that creates an Amazon Bedrock flows resource.
- **guardrails-stack-*.json** — CloudFormation template that creates a guardrail for Amazon Bedrock, if any.
- **agent-stack.json** — CloudFormation template that creates an Amazon Bedrock Agent, if any.
- **invocation-policy-*.json** — CloudFormation template that creates an IAM policy with the runtime permissions that you need to talk to your deployed chat agent app.
- **br-studio-app-stack-*.json** — Parent stack that orchestrates the deployment of all AWS CloudFormation stacks included in the zip package.
- **deploy-app.sh** — Helper script that you use to deploy your app infrastructure into your AWS account.
- **code-snippet.mjs** — Example code snippet that you embed in your code to invoke the app.
- **amazon-bedrock-ide-app.mjs** — Standalone Node.js module to quickly test your deployed app.
- **aoss-encryption-policy-*.json** — AOSS encryption policy necessary to use a Knowledge Base. This encryption policy is automatically created when your chat agent app contains an Amazon Bedrock in SageMaker Unified Studio Knowledge Base.
- **provisioning-inline-policy.json** — An example of an AWS IAM policy that contains the permissions required to provision the chat agent app resources. The permissions declared in this policy file are needed when deploying the AWS CloudFormation stacks.

  You can modify this policy to better suit your needs. You may create a new IAM principal with these policies, or attach these policies to an existing IAM principal in your AWS account.
- **kms-key-policy.json** — An example of an AWS KMS key policy that contains required permissions for encrypting your chat agent app resources.

  You can modify this key policy to better suit your needs. You may create a new KMS key with this policy, or attach this policy to an existing KMS key in your AWS account.

- **api-schema*.json** — OpenAPI schema files associated with your function components, if any.

Topics

- Export your Amazon Bedrock app
- Deploy an exported Amazon Bedrock app
- Run a deployed Amazon Bedrock app

Export your Amazon Bedrock app

Use the following procedure to export a chat agent app or a flow app to a zip file. You can then use the app outside of Amazon SageMaker Unified Studio.

To export a chat agent app or a flow app

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. From the project selector dropdown at the top of the page, choose the project that you want to use.
4. In the left navigation pane, under **Generative AI**, choose **AI apps**.
5. In **Apps** choose the app that you want to export.
6. If you haven't already, choose **Save** to save the app. You can't export an app unless you first save and run the app.
7. On the app page, choose **Export** to export the app. Amazon Bedrock in SageMaker Unified Studio will create and download a zip file with the name **amazon-bedrock-ide-app-export-*.zip**.
8. Next step: Deploy the app.

Deploy an exported Amazon Bedrock app

The following instructions show you the steps you take to deploy a chat agent app that you export from Amazon Bedrock in SageMaker Unified Studio. Make sure to

Topics

- Prerequisites for deploying an exported app
- Deploy the exported app

Prerequisites for deploying an exported app

Before you can deploy a chat agent app that you have exported, you must first do the following:

To prepare for app deployment

1. Install the latest version of the AWS CLI on your local machine by following the instructions at Install or update to the latest version of the AWS CLI.
2. Set up AWS credentials for the AWS CLI on your local machine by following the instructions at Configure the AWS CLI. The credentials that the deployment script uses will follow the order of precedence.
3. (Optional) Using the AWS account that you set up in step 2, create an AWS KMS key for app export by following the instructions at Creating keys. The key must be tagged with key `EnableBedrock` and a value of `true`. The key must also have a key policy that allows it to be used for encryption of your chat agent app resources. You may use the suggested policy declared in the `kms-key-policy.json` file of your zip package.
4. Create an Amazon S3 bucket to hold the app files that you export by following the instructions at Creating a bucket. Make sure the bucket is in the same AWS Region as the app that you are deploying.
5. Create an IAM role that includes the policies from `provisioning-inline-policy.json`. For information about creating a role, see IAM role creation.
6. If your app includes a Knowledge Base, copy the data source file to a folder named `data/` in the Amazon S3 bucket that you created in step 4. If your app uses a document as a datasource, you supply a list of datasource files to the deployment script. For more information, see Deploy the exported app.
7. If your app calls a function that requires authorization, update the function environment secret in Amazon SageMaker AI to the authorization method that your function uses. Run the following command:

```
aws secretsmanager update-secret \
  --secret-id br-studio/function-name-export-environment-id \
  --secret-string 'secret-value'
```

To get the `function-name` and `export-environment-id` values, open the *amazon-bedrock-ide-app-stack-nnnn.json* file from the files that you exported in Export your Amazon Bedrock app. The values are in the `FunctionsStack0` JSON object.

Replace the following values:

- `function-name` — to the value of the `functionName` field in the `FunctionsStack0` JSON object.
- `export-environment-id` — to the value of the `exportAppInstanceId` field in the `FunctionsStack0` JSON object.
- `secret-value` — to the intended value to be used for authentication. You specified the authentication type when you created the function component. Use the authentication values that you specified to complete the `secret-value`.

  If the function requires API Keys, the syntax of `secret-value` should be: `{"key-name-1":"key-value-1","key-name-2":"key-value-2"}`

  If the function requires Basic authentication, the syntax of `secret-value` should be: `{"___AuthType___":"BASIC", "username":"username-value", "password":"password-value"}`

  If the function requires Bearer token authentication, the syntax of `secret-value` should be: `{"___AuthType___":"BEARER", "tokenValue":"token-value"}`

8. Next step: Deploy the exported app.

Deploy the exported app

Before deploying your chat agent app, be sure to do the prerequisite steps.

Deploying a chat agent app deploys the AWS infrastructure files that you need to run the app in AWS.

To deploy an exported app

1. At the command prompt, do the following:

   a. Navigate to the zip file that you exported from Amazon Bedrock in SageMaker Unified Studio.

b. Assume the role of the AWS that you created in step 3 of Prerequisites for deploying an exported app.

c. Use the following command to make sure the deployment script (`deployApp.sh`) is executable:

```
chmod +x deployApp.sh
```

d. Run the deployment script with the following command:

```
./deployApp.sh \
    [--awsRegion=value] \
    [--s3BucketName=value] \
    [--assetsS3Path=value] \
    [--kmsKeyArn=value] \
    [--dataFiles=value]
```

Replace the following values:

- `awsRegion` — with the AWS Region that you want to deploy the app to. Amazon Bedrock must be available in the Region you use. For more information, see Supported AWS Regions.
- `s3BucketName` — With the Amazon S3 bucket that you created in step 5 of Prerequisites for deploying an exported app. The deployment store the CFN templates and application data files in this bucket.
- `assetsS3Path` — (Optional) With the path in `s3BucketName` that you want deployment to store application files to.
- `kmsKeyArn` — (Optional) with the ARN of the KMS Key that you created in step 3 of Prerequisites for deploying an exported app.
- `dataFiles` — With a comma-separated list of data source file paths. Required for apps that use a document data source.

For example, if you have a chat agent app with a single document as a data source, and you want to deploy the app with encryption, you can use the following command.

```
./deployApp.sh \
    --awsRegion=us-east-1 \
    --s3BucketName=my-s3-bucket-name-for-exported-chat-apps \
```

```
    --assetsS3Path=my-prod-folder/my-chat-app \
    --kmsKeyArn=arn:aws:kms:us-
east-1:111122223333:key/11111111-2222-3333-4444-555555555555 \
    --dataFiles=my-data-source.pdf
```

2. (Optional) Monitor the deployment in the AWS CloudFormation console.
3. Note the output from the script. You need it to run the chat agent app. It should be similar to: `node amazon-bedrock-ide-app.mjs --question="prompt" --region="AWS Region"`.

   When you run the app, specify the following parameters:

   - `question` – The prompt that you want to start the app with.
   - `region` – The AWS Region that you deployed the app to. Use the value of `awsRegion` that you specified in step 1c.

   For example, `node amazon-bedrock-ide-app.mjs --question="Tell me about my documents" --region="us-east-1"`
4. Next step: Run a deployed Amazon Bedrock app.

Run a deployed Amazon Bedrock app

The following instructions show you the steps you take to run a deployed Amazon Bedrock in SageMaker Unified Studio chat agent app.

Topics

- Prerequisites for running a chat agent app
- Run the app

Prerequisites for running a chat agent app

Before you can run an app that you have exported, you must first do the following:

To prepare for running an app

1. Download and install Node.js. For more information, see Download Node.js.

2. At the command prompt, install third-party Node.js libraries by running the following commands:

```
npm install minimist
npm install aws-sdk
npm install @aws-sdk/credential-providers
npm install @aws-sdk/client-bedrock-agent-runtime
npm install @aws-sdk/client-bedrock-runtime
```

   For a flow app you also need the following

```
npm install @aws-sdk/client-bedrock-agent
```

3. Create or update an IAM role in which you want to run the app. For the policy, use the policy created by `deployApp.sh` when you exported the app. The policy name is `BRStudioExportedAppInvocationRolePolicy-`*`exportProjectId`*. The policy is declared in invocation-policy-*.json. For more information, see Creating roles.

Run the app

To run your app, you need an IAM role with with permissions to invoke Amazon Bedrock resources. When you deploy the app, the CloudFormation stack deployed through `deployApp.sh` script provisions a suitable policy in your AWS account (declared in `invocation-policy-*.json`).

To run the app

1. Switch to the IAM role that you created in step 3 of Prerequisites for running a chat agent app.
2. Run the app by entering the command you noted in step 3 of Deploy the exported app.

Get started with Amazon Bedrock in SageMaker Unified Studio

Get started with Amazon Bedrock in SageMaker Unified Studio by experimenting with a model in a playground.

The Amazon Bedrock in SageMaker Unified Studio playgrounds that lets you easily experiment with Amazon Bedrock models. The chat playground lets you chat with a model by providing text and image prompts to the model (not all models support images). The image and video playground lets you generate images and videos with a suitable model. With both playgrounds you can experiment

by making configuration changes. For example, you can influence the response from a model by changing inference parameters.

After trying the chat and image playgrounds, you can try creating a chat agent app or flows app. A chat agent app allows users to chat with an Amazon Bedrock model through a conversational interface, typically by sending prompts (text or image) and receiving responses. To create a chat agent app, see Build a chat agent app with Amazon Bedrock.

You can also create a flows app that lets you visually design the flow of an app.

Chat with a model in the chat playground

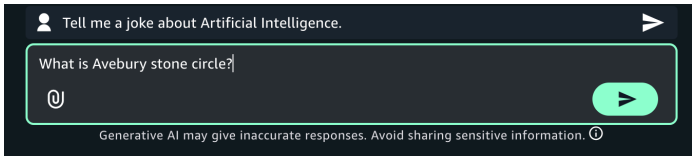
In these instructions, you use the Amazon Bedrock in SageMaker Unified Studio chat playground to chat with an Amazon Bedrock in SageMaker Unified Studio model. You chat by sending a prompt to the model and answering the response that the model generates. For more information, see Experiment with the Amazon Bedrock playgrounds.

If you don't have access to a model, contact your administrator.

To chat with a model

1. Navigate to the Amazon SageMaker Unified Studio landing page by using the URL from your administrator.
2. Access Amazon SageMaker Unified Studio using your IAM or single sign-on (SSO) credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. In the left navigation pane, under **Generative AI**, choose **Playground**.
4. Choose **Chat** to open the chat playground.
5. In **Type** select **Model** and then select a model to use in **Model**. For full information about the model, choose **View full model details** in the information panel. For more information, see Find serverless models with the Amazon Bedrock model catalog. If you don't have access to an appropriate model, contact your administrator. Different models might not support all features.
6. In the **Enter prompt** text box, enter **`What is Avebury stone circle?`**.
7. (Optional) If the model you chose is a reasoning model, you can choose **Reason** to have the model include its reasoning in the reponse. For more information, see Enhance model responses with model reasoning in the *Amazon Bedrock user guide*.

8. Press Enter on your keyboard, or choose the run button, to send the prompt to the model. Amazon Bedrock in SageMaker Unified Studio shows the response from the model in the playground.



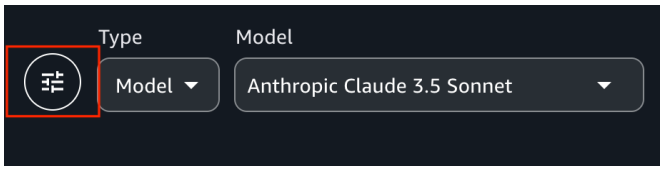
9. Continue the chat by entering the prompt **`Is there a museum there?`** and pressing Enter.

   The response shows how the model uses the previous prompt as context for generating its next response.

10. Choose **Reset** to start a new chat with the model.
11. Influence the model response by doing the following:

    a. Enter and run a prompt. Note the response from the model.

    b. Choose the configurations menu to open the **Configurations** pane.



    c. Influence the model response by making inference parameters changes.

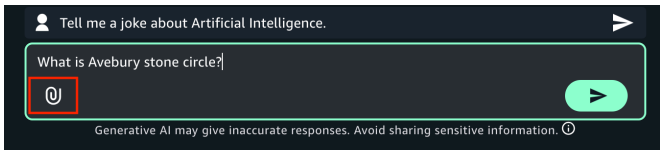
    d. (Optional) In **System instructions**, enter any overarching system instructions that you want the model to apply for future interactions.

    e. Run the prompt again and compare the response with the previous response.

12. Choose **Reset** to start a new chat with the model.
13. Try sending an image to a model by doing the following:

    a. For **Model**, choose a model that supports images.

    b. Choose the attachment button at the left of the **Enter prompt** text box.



    c. In the open file dialog box, choose an image from your local computer.

    d. In the text box, next to the image that you uploaded, enter **`What's in this image?`**.

e. Press Enter on your keyboard enter to send the prompt to the model. The response from the models describes the model or image.

14. (Optional) Try using another model and different prompts. Different models have different recommendations for creating, or engineering, prompts. For more information, see Prompt engineering guides.
15. (Optional) Compare the output from multiple models, or shared apps.

    a. In the playground, turn on **Compare mode**.
    b. In both panes, select the model that you want to compare. If you want to use a shared app, select **App** in **Type** and then select the app in **App**.
    c. Enter a prompt in the text box and run the prompt. The output from each model is shown. You can choose the copy icon to copy the prompt or model response to the clipboard.
    d. (Optional) Choose **View configs** to make configuration changes, such as inference parameters. Choose **View chats** to return to the chat page.
    e. (Optional) Choose **Add chat window** to add a third window. You can compare up to 3 models or apps.
    f. Turn off **Compare mode** to stop comparing models.

Machine learning

Amazon SageMaker Unified Studio is a unified development experience for building analytics, AI/ML, and generative AI applications at scale. This chapter describes the Amazon SageMaker AI capabilities that you can use in Amazon SageMaker Unified Studio.

Note

When you add a custom tag to a SageMaker AI resource (such as a training job, inference endpoint, model, or pipeline), add the prefix `ProjectUserTag` to the tag name. For example:

```
ProjectUserTagMyCustomTag
```

Note

ECR repositories must be created with the `AmazonDataZoneProject` tag with the project ID (which can be found under project details in the project overview page or from the page URL) as the tag value. If you want to add your own tags, they must be prefixed with `ProjectUserTag`.
For example, with AWS CLI:

```
aws ecr create-repository \
--repository-name my-repo \
--tags \
Key=AmazonDataZoneProject,Value=5blxelum5cmckb \
Key=ProjectUserTagMyTag,Value=MyTagValue \
```

Example using Jupyterlab notebook:

```
import boto3

# Create ECR client
ecr_client = boto3.client('ecr')
```

```
# Define repository name
repository_name = 'my-ecr-repo'

# Define tags
tags = [
    {
        'Key': 'AmazonDataZoneProject',
        'Value': '5blxelum5cmckb'
    },
    {
        'Key': 'ProjectUserTagMyTag',
        'Value': 'MyTagValue'
    },
]

try:
    # Create the repository with tags
    response = ecr_client.create_repository(
        repositoryName=repository_name,
        imageScanningConfiguration={
            'scanOnPush': True
        },
        encryptionConfiguration={
            'encryptionType': 'AES256'
        },
        tags=tags
    )

    repository_uri = response['repository']['repositoryUri']
    print(f"Repository created successfully!")
    print(f"Repository URI: {repository_uri}")

except ecr_client.exceptions.RepositoryAlreadyExistsException:
    print(f"Repository {repository_name} already exists")
    # Add tags to existing repository
    ecr_client.tag_resource(
        resourceArn=f"arn:aws:ecr:{ecr_client.meta.region_name}:
{boto3.client('sts').get_caller_identity()['Account']}:repository/
{repository_name}",
        tags=tags
    )
    # Get the repository URI
```

```
    response =
 ecr_client.describe_repositories(repositoryNames=[repository_name])
    repository_uri = response['repositories'][0]['repositoryUri']
    print(f"Added tags to existing repository")
    print(f"Repository URI: {repository_uri}")
except Exception as e:
    print(f"Error creating repository: {str(e)}")
```

ECR repositories without the `AmazonDataZoneProject` cannot be used. You must create new ECR repositories with the `AmazonDataZoneProject` tag. Once tagged with the `AmazonDataZoneProject` tag, this tag cannot be modified or removed from your ECR repositories. For more information about ECR repositories, see https://docs.aws.amazon.com/AmazonECR/latest/userguide/what-is-ecr.html.

Topics

- Machine learning in Identity Center-based domains
- Machine Learning Workflows in IAM-based domains

Machine learning in Identity Center-based domains

Topics

- Discover Jumpstart models
- Build models in JupyterLab
- Train models
- Use inference endpoints to deploy models
- Pipelines
- Model registry
- Track experiments using MLflow
- HyperPod clusters
- Partner AI apps

Discover Jumpstart models

Amazon SageMaker Unified Studio maintains publicly available foundation models for you to access, customize, and integrate into your machine learning lifecycles. A foundation model is a large pre-trained model that's adaptable to a variety of downstream tasks and often serves as the starting point for developing more specialized models.

To explore the available models from our model providers, follow these steps:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.
2. In the left navigation pane, under **AI/ML**, choose **JumpStart models**.

   The system opens a page that lists the model providers.
3. Choose a provider.
4. From the provider's list of models, choose a model to view its details.

For more information about discovering models, see JumpStart foundation models in the *Amazon SageMaker AI Developer Guide*.

Build models in JupyterLab

Use the JupyterLab space within Amazon SageMaker Unified Studio to run JupyterLab applications. A JupyterLab space is a private or shared space that manages the storage and compute resources needed to run the JupyterLab application.

The JupyterLab application is a web-based interactive development environment (IDE) for notebooks, code, and data. Use the JupyterLab application's flexible and extensive interface to configure and arrange machine learning (ML) workflows.

Your project contains a configured JupyterLab space.

To open the JupyterLab space in Amazon SageMaker Unified Studio, follow these steps:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.
2. In the left navigation pane, under **IDEs**, choose **JupyterLab**.

Amazon SageMaker Unified Studio opens the JupyterLab space associated with your project. Choose **Configure space** to tailor the configuration to your needs.

For more information about using JupyterLab, see SageMaker JupyterLab in the *Amazon SageMaker AI Developer Guide*.

Train models

Using Amazon SageMaker Unified Studio, you can train foundation models or custom models.

Follow these steps to train a foundation model:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.
2. Choose a model to train.
   a. In the left navigation pane, under **AI/ML**, choose **JumpStart models**.

      The JumpStart page lists the model providers.
   b. Choose a model provider. The page displays the models for that provider.
   c. Under **Action**, choose **Trainable**. The page displays the trainable models for that provider.
   d. From the provider's list of models, choose the model you want to train.
3. From the model details page, choose **Train** to create a training job.

   If the model is pretrained, you can fine-tune the model by adjusting the model parameters.
4. In the **Fine-tuning model** page, update the hyperparameters you want to change.
5. Enter **Submit** to submit the training job. You can view the training job from the **Training jobs** page.

You can also train the model in a Jupyterlab notebook using the SageMaker AI python SDK.

For more information about training models in JumpStart, see JumpStart pretrained models in the *Amazon SageMaker AI Developer Guide*.

Use inference endpoints to deploy models

Endpoint are locations where you send inference requests to your deployed machine learning models. After you create an endpoint, you can add models to it, test it, and change its settings as needed. By using endpoints, you don't have to manage the underlying infrastructure for configuring and deploying a model.

For more information about using endpoints for real-time inference, see Deploy models for real-time inference in the *Amazon SageMaker AI Developer Guide*. Also see the Getting started with deploying real time models on SageMaker AI blog post.

Topics

- Create an endpoint and deploy a model
- View your endpoints

Create an endpoint and deploy a model

To create an endpoint, follow these steps:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.
2. In the left navigation pane, under **AI/ML**, choose **Inference endpoints**.
3. From the **Endpoints** page, choose **Create endpoint**.
4. From the **Create endpoint** page, configure these values:
   - For **Endpoint name**, enter a name for the endpoint.
   - For **Instance type**, choose an instance for the endpoint.
   - For **Initial instance count**, enter the number of instances for the endpoint to provision initially.
   - For **Maximum instance count**, enter the maximum number of instances that the endpoint can provision, when it scales up.
5. Under **Models**, choose **Add model**. In the **Add model** modal form, follow these steps:
   a. Select the model type (JumpStart foundation models or Deployable models that you created).

      The form lists the models that are compatible with the instance type you selected.
   b. Choose one of the models.
   c. Under **Model settings**, enter these values:
      - Number of CPU cores – Number of accelerators to deploy.
      - Minimum number of copies – minimum number of model copies to deploy.
      - Min CPU memory – Minimum amount of CPU memory.
      - Max CPU memory – Maximum amount of CPU memory.

d. Choose **Add model**.

6. Choose **Deploy** to deploy the endpoint.

View your endpoints

To view your endpoints in the **Endpoints** table, follow these steps:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.
2. In the left navigation pane, under **AI/ML**, choose **Inference endpoints**.
3. (Optional) To search for specific endpoints, enter text in **Search by endpoint name**.

Pipelines

Amazon SageMaker Unified Studio supports SageMaker AI Pipelines, a workflow orchestration service for automating machine learning (ML) development.

A pipeline defines a series of interconnected steps in a directed acyclic graph (DAG). You can define the steps using the Amazon SageMaker Unified Studio visual pipeline designer, or by creating a pipeline definition JSON schema. This DAG JSON definition gives information on the requirements and relationships between each step of your pipeline.

The structure of a pipeline's DAG is determined by the data dependencies between steps. These data dependencies are created when the properties of a step's output are passed as the input to another step.

> **Note**
>
> To add a custom tag to a pipeline, add the prefix `ProjectUserTag` to the tag name. For example:
>
> ```
> ProjectUserTagMyCustomTag
> ```

For an overview of pipelines, see Pipelines overview in the *Amazon SageMaker AI Developer Guide*.

Pipeline actions

The following sections describe the actions available in Amazon SageMaker Unified Studio to create and manage pipelines:

Topics

- Define a pipeline
- Edit a pipeline
- Run a pipeline
- Stop a pipeline execution
- View the details of a pipeline
- View the details of a pipeline run
- Download a pipeline definition file

Define a pipeline

You define a pipeline using the visual pipeline designer. You can also create a pipeline definition JSON schema.

To define a pipeline using the visual pipeline designer, complete the following steps:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.
2. In the left navigation pane, under **AI/ML**, choose **ML pipelines**. The system displays the pipelines for your project.
3. Choose **Create in visual editor**. The system opens the visual editor for creating a pipeline. You can also import a pipeline definition file from your computer.
4. Use the visual editor to add and connect pipeline steps.
5. Chose the **Save** to save your changes.

For more details, see Define a pipeline in the *Amazon SageMaker AI Developer Guide*.

Edit a pipeline

You can make changes to a pipeline before running it. To edit a pipeline, complete the following steps:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.
2. In the left navigation pane, under **AI/ML**, choose **ML pipelines**. The system displays the pipelines for your project.
3. Choose the pipeline to edit.
4. Chose the **Executions** tab.
5. Choose the pipeline execution to edit.
6. Chose the **Visual editor** to edit the pipeline.
7. Chose the **Save** to save your changes.

For more details, see Edit a pipeline in the *Amazon SageMaker AI Developer Guide*.

Run a pipeline

After defining the steps of your pipeline as a directed acyclic graph (DAG), you can run your pipeline, which executes the steps defined in your DAG.

To run a pipeline, complete the following steps:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.
2. In the left navigation pane, under **AI/ML**, choose **ML pipelines**. The system displays the pipelines for your project.
3. Choose the pipeline to run.
4. Chose **Execute**.
    a. For **Execution name**, enter a name for this run.
    b. (Optional) For **Description**, enter a description for this run.
5. Chose **Execute** to start the run.

For more details, see Run a pipeline in the *Amazon SageMaker AI Developer Guide*.

Stop a pipeline execution

To stop a pipeline, complete the following steps:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.

2. In the left navigation pane, under **AI/ML**, choose **ML pipelines**. The system displays the pipelines for your project.
3. Choose the pipeline to stop.
4. Choose the **Executions** tab.
5. Choose the execution to stop.
6. Choose **Stop** to stop the execution. To resume the execution from where it was stopped, choose **Resume**.

View the details of a pipeline

You can view the details of a pipeline to understand its parameters, the dependencies of its steps, or monitor its progress and status.

To access the details of a given pipeline using Amazon SageMaker Unified Studio, complete the following steps:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.
2. In the left navigation pane, under **AI/ML**, choose **ML pipelines**. The system displays the pipelines for your project.
3. Choose the pipeline to view its details.
4. Choose any of the following tabs to view these details:

   - **Executions** – Details about the executions.
   - **Graph** – The pipeline graph, including all steps.
   - **Parameters** – The run parameters and metrics related to the pipeline.
   - **Information** – The metadata associated with the pipeline, such as tags, the pipeline Amazon Resource Name (ARN), and role ARN. You can also edit the pipeline description from this location.

View the details of a pipeline run

You can view the details of a pipeline run, which can help you:

- Identify and resolve problems that may have occurred during the run, such as failed steps or unexpected errors.

- Compare the results of different pipeline executions to understand how changes in input data or parameters impact the overall workflow.
- Identify bottlenecks and opportunities for optimization.

To view the details of a pipeline run, complete the following steps:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.
2. In the left navigation pane, under **AI/ML**, choose **ML pipelines**. The system displays the pipelines for your project.
3. Choose the pipeline to view its details.
4. Choose the **Executions** tab.
5. Choose the pipeline execution to view. The pipeline graph for that execution appears.
6. Choose any of the pipeline steps in the graph to see step settings in the right sidebar.
7. Choose any of the following tabs to view these details:

    - **Graph** – The pipeline graph, including all steps.
    - **Parameters** – The run parameters and metrics related to the pipeline.
    - **Information** – The metadata associated with the pipeline, such as tags, the pipeline Amazon Resource Name (ARN), and role ARN. You can also edit the pipeline description from this location.

Download a pipeline definition file

You can download the definition file for your pipeline. You can use this pipeline definition file for:

- Backup and restoration: Use the downloaded file to create a backup of your pipeline configuration, which you can restore in case of infrastructure failures or accidental changes.
- Version control: Store the pipeline definition file in a source control system to track changes to the pipeline and revert to previous versions if needed.
- Programmatic interactions: Use the pipeline definition file as input to the SDK or AWS CLI.
- Integration with automation processes: Integrate the pipeline definition into your CI/CD workflows or other automation processes. For more information about CI/CD in Amazon SageMaker Unified Studio, see *CI/CD*.

To download the definition file of a pipeline, complete the following steps:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.
2. In the left navigation pane, under **AI/ML**, choose **ML pipelines**. The system displays the pipelines for your project.
3. Choose the pipeline. You can download the pipeline definition from this page or any of the execution pages.
4. At the top right of the page, choose the vertical ellipsis and choose **Download pipeline definition (JSON)**.

For more information about the pipeline actions, see Pipelines actions in the *Amazon SageMaker AI Developer Guide*.

Model registry

Use the model registry to catalog your models and manage model deployment to production.

You catalog models by creating model (package) groups that contain different versions of a model. You can create a model group that tracks all the models that you train to solve a particular problem. You can then register each model you train and the model registry adds it to the model group as a new model version.

You can create categories of model groups by organizing them into collections. A typical workflow might include the following tasks:

1. Create a model group.
2. Create an ML pipeline that trains a model. For information about pipelines, see Pipelines.
3. For each run of the ML pipeline, create a model version that you register in the model group you created in the first step.
4. Add your model group to one or more collections.

For details about how to work with the model registry, see Model Registry, Model Versions, and Model Groups in the *Amazon SageMaker AI Developer Guide*.

Create a model group

A model group contains different versions of a model. Follow these steps to create a model group:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.

2. In the left navigation pane, under **AI/ML**, choose **Model registry**. The model registry page displays the models that are registered to your project.
3. Choose **Model Groups**. The page displays the model groups that are defined for your project.
4. From the actions menu, choose **Create model group**.
5. Provide a name for the model group. Optionally, you can add keys to the model group.
6. Choose **Register model group** to create the model group.
7. Confirm that your newly-created model appears in the list of model groups.

Create a collection

Follow these steps to create a collection:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.
2. In the left navigation pane, under **AI/ML**, choose **Model registry**. The model registry page displays the models that are registered to your project.
3. Choose **Collections**. The collections page displays the collections that are defined for your project.
4. In the **Actions** drop-down menu, choose **Create collection**.
5. Provide a name for the collection.
6. (Optional) To add model groups to the collection, complete these steps:
    a. Choose **Select model groups**.
    b. Select up to 10 model groups that you want to add.
7. Choose **Create** to create the collection.

Register a model version

The model registry is structured as several model (package) groups with model packages in each group. Each model package in a model group corresponds to a trained model. The version of each model package is a numerical value that starts at 1 and is incremented with each new model package added to a model group. The model packages used in the model registry are versioned, and **must** be associated with a model group.

Follow these steps to register a model version:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.
2. In the left navigation pane, under **AI/ML**, choose **Model registry**. The model registry page displays the models that are registered to your project.
3. Choose **Register**.
4. From the **Register Model** page, Choose the type of model artifact:
   - JumpStart – Choose from the list of models.
   - Jobs – Choose a training job
   - Bring-your-model – Enter the location of your models
5. Choose **Register**.
6. Create a model group, or find an existing model group.
7. Choose **Register**.

For more information, see Model Registry in the *Amazon SageMaker AI Developer Guide*.

Track experiments using MLflow

Amazon SageMaker Unified Studio supports two options for tracking experiments with MLflow: MLflow Apps and MLflow Tracking Servers. MLflow Apps are the latest offering with faster startup times and cross-account sharing, while MLflow Tracking Servers provide traditional MLflow functionality.

For more information about MLflow, see Machine learning experiments using MLflow in the *Amazon SageMaker AI Developer Guide*.

Topics

- Use MLflow Apps for experiment tracking
- Use MLflow Tracking Servers for experiment tracking

Use MLflow Apps for experiment tracking

Use MLflow Apps in Amazon SageMaker Unified Studio to track, manage, analyze, and compare machine learning experiments. MLflow Apps are the latest managed MLflow offering and provide faster startup times, cross-account sharing, and integration with other SageMaker AI features.

MLflow Apps should be preferred over existing MLflow Tracking Servers.

MLflow Apps provide capabilities for experiment tracking, model registry, and tracing generative AI applications. Each MLflow App includes compute resources, backend metadata storage, and artifact storage in Amazon S3.

> **Note**
>
> MLflow Apps are different from MLflow Tracking Servers. MLflow Apps offer additional features such as faster startup time, cross-account sharing, and automatic model registration. For information about MLflow Tracking Servers, see the section called "Use MLflow Tracking Servers for experiment tracking".

For more information about MLflow Apps, see MLflow App Setup in the *Amazon SageMaker AI Developer Guide*.

Topics

- MLflow Apps overview
- Prerequisites
- Create an MLflow App
- Edit an MLflow App
- Delete an MLflow App
- Launch the MLflow UI
- Integrate MLflow with your environment

MLflow Apps overview

An MLflow App is a stand-alone HTTP server that serves multiple REST API endpoints for tracking runs and experiments. MLflow Apps provide the following capabilities:

- Experiment tracking: Track parameters, metrics, and artifacts across multiple training runs to identify the best performing models.
- Model registry: Manage model versions and catalog models for production deployment.
- Tracing: Record inputs, outputs, and metadata at every step of a generative AI application to identify issues and maintain traceability.
- Automatic model registration: Automatically register models from MLflow Model Registry to SageMaker AI Model Registry.

You can create MLflow Apps for your project. Your domain administrator can configure the project profile to automatically create an MLflow App during project creation, though this is not recommended due to default quota limits and increased project creation latency. We recommend creating MLflow Apps on demand after project creation.

When you delete a project, Amazon SageMaker Unified Studio automatically deletes associated MLflow Apps.

Prerequisites

Before you create an MLflow App, ensure you have the following:

- An Amazon S3 bucket in the same AWS Region as your project for artifact storage. The MLflow App uses this bucket to store model artifacts, images, and data files.
- Appropriate IAM permissions to create and manage MLflow Apps. Your domain administrator configures these permissions through the following IAM policies:
  - SageMakerStudioProjectRoleMachineLearningPolicy
  - SageMakerStudioProjectProvisioningRolePolicy
  - SageMakerStudioUserIAMDefaultExecutionPolicy
  - SageMakerStudioAdminIAMDefaultExecutionPolicy

For more information about IAM permissions, see the Security chapter.

Create an MLflow App

After you create a project, you can create an MLflow App for the project if it was not created automatically during project creation.

To create an MLflow App, perform the following steps:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.
2. In the left navigation pane, choose **Compute**.
3. In the left navigation pane, under **AI/ML**, choose **MLflow**.
4. Choose **Create MLflow App**.
5. For **Name**, enter a name for the MLflow App. The name must start with a letter or number and can contain letters, numbers, and hyphens.
6. Choose **Create** to create the MLflow App.

> **Note**
>
> It may take 2-3 minutes to complete MLflow App creation. When you successfully create an MLflow App, it automatically starts.

Edit an MLflow App

After you create an MLflow App, you can change the artifact storage location. To edit an MLflow App, perform the following steps:

1. In the left navigation pane, choose **Compute**.
2. In the left navigation pane, under **AI/ML**, choose **MLflow**.
3. From the **Actions** drop-down menu, choose **Edit**.
4. For **Artifact storage S3 path**, enter a new path to the artifact storage.
5. Choose **Save changes** to update the MLflow App.

Delete an MLflow App

You can delete an MLflow App when you no longer need it. Deleting an MLflow App removes the compute resources but does not delete the artifacts stored in Amazon S3.

To delete an MLflow App, perform the following steps:

1. In the left navigation pane, choose **Compute**.
2. In the left navigation pane, under **AI/ML**, choose **MLflow**.
3. From the **Actions** drop-down menu, choose **Delete**.
4. In the confirmation dialog, enter the MLflow App name to confirm deletion.

> **Note**
>
> An MLflow App is not available for use while it is in certain states, such as creating, pending deletion, or other transitional states.

- Choose **Delete** to delete the MLflow App.

> **Important**
>
> Deleting an MLflow App is permanent and cannot be undone. Ensure you have backed up any important experiment data before deleting the MLflow App.

Launch the MLflow UI

You can launch the MLflow UI to view and manage your experiments, models, and traces. To launch the MLflow UI, perform the following steps:

1. In the left navigation pane, choose **Compute**.
2. In the left navigation pane, under **AI/ML**, choose **MLflow**.
3. Choose the **Open** button next to the MLflow App. This action uses a presigned URL to launch the MLflow UI in a new tab in your current browser.

For more information about using the MLflow UI, see Launch the MLflow UI using a presigned URL in the *Amazon SageMaker AI Developer Guide*.

Integrate MLflow with your environment

After you create an MLflow App, you can integrate it with your development environment to track experiments and log metrics.

To integrate MLflow with your environment, you need the MLflow App ARN. You can find the ARN on the MLflow App details page.

For detailed information about integrating MLflow with your environment, including code examples for Python notebooks, see Integrate MLflow with your environment in the *Amazon SageMaker AI Developer Guide*.

Use MLflow Tracking Servers for experiment tracking

Use MLflow Tracking Servers in Amazon SageMaker Unified Studio to track, manage, analyze, and compare machine learning experiments. MLflow Tracking Servers provide compute and storage resources for experiment tracking. Each project can have an MLflow Tracking Server. Your domain administrator can configure the project defaults to automatically create the MLflow Tracking Server during project creation. Otherwise, you can create an MLflow Tracking Server on demand for the project.

> **Note**
>
> MLflow Tracking Servers are different from MLflow Apps. MLflow Apps offer additional features such as faster startup time, cross-account sharing, and automatic model registration. For information about MLflow Apps, see the section called "Use MLflow Apps for experiment tracking".

When you delete a project, Amazon SageMaker Unified Studio automatically deletes the tracking server.

For more information about MLflow Tracking Servers, see MLflow Tracking Servers in the *Amazon SageMaker AI Developer Guide*.

For more information about project profiles for AI-ML projects, see Project profiles in the *Amazon SageMaker Unified Studio Admin Guide*.

Create an MLflow Tracking Server

After you create a project, you can create an MLflow Tracking Server for the project, if it wasn't created automatically during project creation.

To create an MLflow Tracking Server, perform the following steps:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.
2. From the project selector dropdown at the top of the page, choose your project.
3. In the left navigation pane, choose **Compute**.
4. In the left navigation pane, under **AI/ML**, choose **MLflow**.
5. Choose **Create MLflow Tracking Server**.
6. (Optional) Provide values to override the default values for the following fields:
    a. Name – enter a name for the server.
    b. Size – select a size for the server.
7. Choose **Create** to create the server.

Edit an MLflow Tracking Server

After you create a tracking server, you can change the configured server size, if the current size isn't sufficient for the project.

To edit a tracking server, perform the following steps from your project's **MLflow** tab under **Compute**:

1. From the **Actions** drop-down menu, choose **Edit**. You can change the following values:
    a. Size – select a new size for the server.
    b. Artifact storage S3 path – enter a new path to the artifact storage.
2. Choose **Save changes** to update the tracking server.

Start or stop an MLflow Tracking Server

You can stop a running server or start a stopped server. While the tracking server is starting or stopping, it's not available for MLflow to use.

To start or stop an MLflow tracking server, perform the following steps from your project's **Project details** page:

1. In the left navigation pane, choose **Compute**.
2. In the left navigation pane, under **AI/ML**, choose **MLflow**.
3. From the **Actions** drop-down menu, choose **Stop** to stop a running server. Choose **Start** to start a stopped server.

Integrate MLflow with your environment

For information about how to integrate MLflow with your environment, see Integrate MLflow with your environment in the *Amazon SageMaker AI Developer Guide*.

Launch the MLflow UI

You can launch the MLflow Tracking Server UI from the **MLflow** tab under **Compute**, by performing the following steps:

1. Navigate to the project details page for your project.
2. In the left navigation pane, choose **Compute**.

3. In the left navigation pane, under **AI/ML**, choose **MLflow**.
4. From the **Actions** drop-down menu, choose **Open MLflow**. This action uses a presigned URL to launch the MLflow UI in a new tab in your current browser.

For more information, see Launch the MLflow UI using a presigned URL in the *Amazon SageMaker AI Developer Guide*.

HyperPod clusters

Use Amazon SageMaker AI HyperPod to help you provision resilient compute clusters for running model training or fine-tuning workloads. Amazon SageMaker AI HyperPod integrates with Slurm or Amazon EKS for orchestration.

You can create HyperPod clusters using the Amazon SageMaker AI Hyperpod console UI or SageMaker AI Studio. For more information, see Orchestrating SageMaker AI HyperPod clusters with Slurm or Orchestrating SageMaker AI HyperPod clusters with Amazon EKS in the *Amazon SageMaker AI Developer Guide*.

In Amazon SageMaker Unified Studio, you can launch machine learning workloads on Amazon SageMaker AI HyperPod clusters. You can also view details about the HyperPod clusters.

Topics

- Connect to a HyperPod cluster
- View the HyperPod clusters
- View details about a HyperPod cluster
- HyperPod task governance
- Open the HyperPod in JupyterLab

Connect to a HyperPod cluster

To use a HyperPod cluster in Amazon SageMaker Unified Studio, you create a connection to the cluster by following these steps:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.
2. In the left navigation pane, choose **Compute**, then choose the **HyperPod clusters** tab.
3. Choose **Add compute**.
4. In the **Add compute** form, configure the following fields:

a. For **Connection name**, enter a name for this connection.
b. For **HuperPod cluster name**, enter the name of the HyperPod cluster.
c. For **Access role ARN**, enter the IAM role that the project needs to assume.
d. For **Account ID**, enter the AWS account where the runtime role exists.
e. For **AWS Region**, enter the Region where the HyperPod cluster was created.

View the HyperPod clusters

To view the HyperPod clusters in your project, follow these steps:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.
2. In the left navigation pane, choose **Compute**, then choose the **HyperPod clusters** tab.

   The HyperPod clusters table provides a summary view of each cluster, including the ARN, status, and creation time.

View details about a HyperPod cluster

To view the details page for a HyperPod cluster, choose the HyperPod from the table of HyperPod clusters. The page displays tabs for tasks, metrics, settings, and metadata details.

For more information about HyperPod cluster details that you can view in Amazon SageMaker Unified Studio, see HyperPod tabs in Studio in the *Amazon SageMaker AI Developer Guide*.

HyperPod task governance

For Amazon EKS clusters, you can use HyperPod task governance to streamline resource allocation and utilization of compute resources in the cluster.

HyperPod task governance provides a comprehensive dashboard view of your Amazon EKS cluster utilization metrics, including hardware, team, and task metrics.

For more information about the HyperPod dashboard view, see Dashboard in the *Amazon SageMaker AI Developer Guide*.

Open the HyperPod in JupyterLab

To open your HyperPod in JupyterLab, follow these steps:

1. From the cluster details page, choose **Open in JupyterLab**.

   The **Starting space** page opens and the space initialization starts.

   After the JupyterLab space is ready, it opens the HyperPod sample notebook.
2. The HyperPod sample notebook shows the end-to-end flow of how to use the HyperPod cluster, including sample commands for:

   - Connecting to the cluster
   - Submitting jobs to the cluster.
   - Viewing job status or cluster status.

Partner AI apps

Amazon SageMaker Unified Studio provides access to Amazon SageMaker AI Partner AI apps. Partner AI apps include generative AI and machine learning (ML) development applications that are built, published, and distributed by industry-leading application providers.

Amazon SageMaker AI Partner AI apps are full application stacks that include an Amazon EKS cluster and accompanying services such as Application Load Balancer, Amazon RDS, Amazon S3 buckets, or Amazon SQS queues.

To view the partner AI apps, complete these steps:

1. Sign in to Amazon SageMaker Unified Studio using the link that your administrator gave you.
2. In the left navigation pane, under **AI/ML**, choose **Partner AI apps**.

For more information about SageMaker AI partner AI apps, see Partner AI Apps overview in the *Amazon SageMaker AI Developer Guide*.

Machine Learning Workflows in IAM-based domains

Amazon SageMaker Unified Studio provides a comprehensive machine learning environment within IAM-based domains that enables you to discover, deploy, and manage machine learning models through a unified interface.

- Discover Foundation Models and Registered Models from multiple model providers, easily deploy them using sample notebooks

- Customize foundation models using Jupyterlab IDE or Data Notebooks, a new serverless Notebook for ML Practitioners
- Create and track Experiments and identify the best model for your usecase using MLflow
- Use Agentic AI is Notebooks to easily create and train models
- Monitor training jobs and model performance metrics
- Manage model lifecycle through the integrated model registry

The machine learning capabilities in Amazon SageMaker Unified Studio integrate seamlessly with your project's IAM permissions and compute resources, providing secure access to models and deployment infrastructure within your domain's governance framework.

Discover Foundation models

1. Navigate to your Amazon SageMaker Unified Studio project using the URL provided by your administrator.
2. From the left navigation menu, choose **Models**.
3. Use the Model finder to search for models by task type:
   - Choose **Text** for natural language processing models
   - Choose **Image** for computer vision models
   - Choose **Audio** for speech and audio processing models
   - Choose **Multimodal** for models that handle multiple input types
   - Choose **Video** for video processing models
4. Browse Featured providers to explore models from specific providers.
5. Switch between Foundation Models and Registered Models tabs:
   - Foundation Models displays pre-trained models available for immediate use
   - Registered Models shows models you have registered in your project's model registry
6. Use the search functionality to find specific models by name or capability.

Deploy Foundation models

1. From the Models page, select a Foundation Model you want to deploy.
2. On the model details page, review the model information including:

- Model architecture and capabilities
- Supported languages and input modalities
- Training data information
- License requirements

3. Choose **Deploy** to access deployment options. This will create a Sample notebook that you can use to review and deploy.
4. Review and execute the sample notebook content which demonstrates:
   - Model selection and configuration
   - Endpoint deployment procedures
   - Example inference code
5. Wait for the deployment to complete. The deployment process may take several minutes.
6. After successful model deployment, navigate to **Endpoints** from the left navigation menu.
7. View your deployed endpoints in the endpoints list, which displays:
   - Endpoint name and status
   - Creation and modification timestamps
   - Endpoint configuration details
8. Monitor endpoint status:
   - In Service indicates the endpoint is ready for inference requests
   - Creating shows the endpoint is being deployed
   - Failed indicates deployment issues that need attention

Track experiments using MLflow

Amazon SageMaker Unified Studio supports two options for tracking experiments with MLflow: MLflow Apps and MLflow Tracking Servers. MLflow Apps are the latest offering with faster startup times and cross-account sharing, while MLflow Tracking Servers provide traditional MLflow functionality.

Topics

- Use MLflow Apps for experiment tracking
- Use MLflow Tracking Servers to track experiments

Use MLflow Apps for experiment tracking

> **Note**
>
> MLflow Apps are different from MLflow Tracking Servers. MLflow Apps offer additional features such as faster startup time and cross-account sharing. For information about connecting to existing MLflow Tracking Servers, see the section called “Use MLflow Tracking Servers to track experiments”.

MLflow Apps are the latest managed MLflow offering in Amazon SageMaker Unified Studio and provide faster startup times, cross-account sharing, and integration with SageMaker AI features. MLflow Apps use MLflow 3.0 and support experiment tracking, model registry, and tracing for generative AI applications.

Connect to an MLflow App

You can connect to an existing MLflow App created in SageMaker AI Studio to track experiments and manage model versions. Note that you can't create a new MLflow App in Amazon SageMaker Unified Studio. You have to create this using SageMaker AI APIs or in SageMaker AI Studio.

To connect to an MLflow App, perform the following steps:

1. From your project's main page, choose **MLflow** from the left navigation menu.
2. Choose **Connect MLflow App**.
3. Enter an MLflow App Name
4. Provide a Connection name for identification
5. Enter the MLflow App ARN for your project
6. Choose **Connect to app**

Manage MLflow Apps

After you connect to an MLflow App, you can perform the following actions from the **MLflow** page:

- Open MLflow – Choose the **Open** button next to the MLflow App to launch the MLflow UI and view experiments, models, and traces.
- Edit – Update the connection with a new ARN.
- Delete – Remove the connection to the MLflow App.

To access the **MLflow** page, choose **MLflow** from the left navigation menu.

Use MLflow Tracking Servers to track experiments

To get started, you should have an existing MLflow server created in SageMaker AI Studio. Make sure that you have the ARN to get started.

1. From your project's main page, choose **MLflow** from the left navigation menu.
2. Connect to an existing MLflow Tracking Server. Note that you can't create a new MLflow Server in Amazon SageMaker Unified Studio. You have to create this using SageMaker AI APIs or in SageMaker AI Studio.
3. Choose **Connect Tracking Server**
4. Enter a Tracking Server Name
5. Provide a Connection name for identification
6. Enter the MLflow Tracking Server ARN for your project
7. Choose **Connect to server**
8. Once connected, choose **Open MLflow** to launch the MLflow UI.
9. In the MLflow interface, view your experiments:
    - Experiments tab shows all tracked experiments
    - Models tab displays registered model versions
    - Prompts tab contains prompt templates and versions
10. You can perform additional actions such as
    - Stop ML Server
    - Use server to train model – this will launch a sample notebook which will provide instructions on how to use MLflow to train a linear regression model
    - Edit the connection with new ARNs
    - Delete the connection

Create and monitor custom training jobs

You can use Jupyter Notebooks or SageMaker AI Notebooks to train your ML jobs. Refer to SageMaker AI documents on how to train ML jobs.

```
model_trainer = ModelTrainer(
    training_image=image_uri,
    source_code=source_code,
    base_job_name=job_name,
    compute=compute_configs,
    distributed=Torchrun(),
    stopping_condition=StoppingCondition(
        max_runtime_in_seconds=7200
    ),
    hyperparameters={
        "config": "/opt/ml/input/data/config/args.yaml" # sample path
    },
    output_data_config=OutputDataConfig(
        s3_output_path=output_path
    ),
)
# starting the train job with our uploaded datasets as input
model_trainer.train(input_data_config=data, wait=True)
```

You can monitor the results of the training job by selecting **Training Jobs** on the left panel

You can stop the jobs, monitor the artifacts, hyper parameters, security configurations and tags that you have setup during the training.

Manage Registered Models

Your registered models can be accessed and managed in Amazon SageMaker Unified Studio.

1. Navigate to **AI/ML** > **Models** and select the **Registered Models** tab to view the models.
2. You can review the following information
   - Model group name - Logical grouping for related model versions
   - Model version - Specific version identifier
   - Model artifacts - Location of model files in Amazon S3
   - Description - Optional description of the model and its purpose
3. Select specific models to review key model information like
   - Framework - Machine learning framework used (e.g., PyTorch, TensorFlow)

- Algorithm - Algorithm or approach used for training
- Performance metrics - Accuracy, precision, recall, or other relevant metrics

Model lifecycle management

The model registry provides version control and lifecycle management:

- Version tracking - Each model registration creates a new version with unique metadata
- Approval workflows - Models can have approval status (Approved, Pending, Rejected)
- Deployment status - Track which versions are deployed to endpoints
- Model comparison - Compare metrics and metadata across versions

To manage model versions:

1. Choose a model group name to view all versions.
2. Review version details including:
   - Version number - Sequential version identifier
   - Description - Version-specific notes and changes
   - Deployment Status - Current deployment state
   - Approval Status - Workflow approval state
   - Modified Date - Last update timestamp
3. Use the **Actions** dropdown to:
   - Deploy using Notebooks - Use a pre-created notebook to deploy the model to SageMaker AI Endpoint
   - Deploy using Jupyter Notebook - Use a pre-created JupyterLab notebook to deploy the model to SageMaker AI Endpoint
   - Approve/Reject - Update approval status
   - Delete - Delete the selected version

Manage feature groups in Feature Store

Amazon SageMaker Feature Store is a centralized repository for creating, storing, sharing, and managing ML features. With the Feature Store interface in Amazon SageMaker Unified Studio, you can create, discover, and update feature groups directly from the project UI without writing code.

The Feature Store interface described on this page is available in Amazon SageMaker Unified Studio domains configured with AWS IAM. You can use Feature Store from a notebook or IDE using the SDK in any domain type.

A feature group is a collection of features defined in your feature store to describe a record. You can visualize a feature group as a table in which each column is a feature, with a unique identifier for each row. A record is a collection of values for features that correspond to a unique record identifier.

When creating a feature group, you choose a storage configuration:

- **Online** stores retain only the latest feature values for low-latency (millisecond) reads and high-throughput predictions. To automatically expire records from the online store, configure time to live (TTL) using the SDK from a notebook in your project.
- **Offline** stores keep a historical record of all feature values in your Amazon S3 bucket, stored in Parquet format. Use the offline store for data exploration, model training, and batch inference.
- **Online and Offline** combines both modes.

> **Note**
>
> For a full overview of Feature Store concepts and capabilities, see Create, store, and share features with Feature Store in the Amazon SageMaker AI Developer Guide. The Amazon SageMaker AI Developer Guide documents the same Feature Store capabilities but might describe a different UI experience. Use those pages as a conceptual reference rather than step-by-step instructions for Amazon SageMaker Unified Studio.

Navigate to Feature Store

1. Open your project in Amazon SageMaker Unified Studio.
2. In the left navigation pane, under AI/ML, choose **Feature Store**.

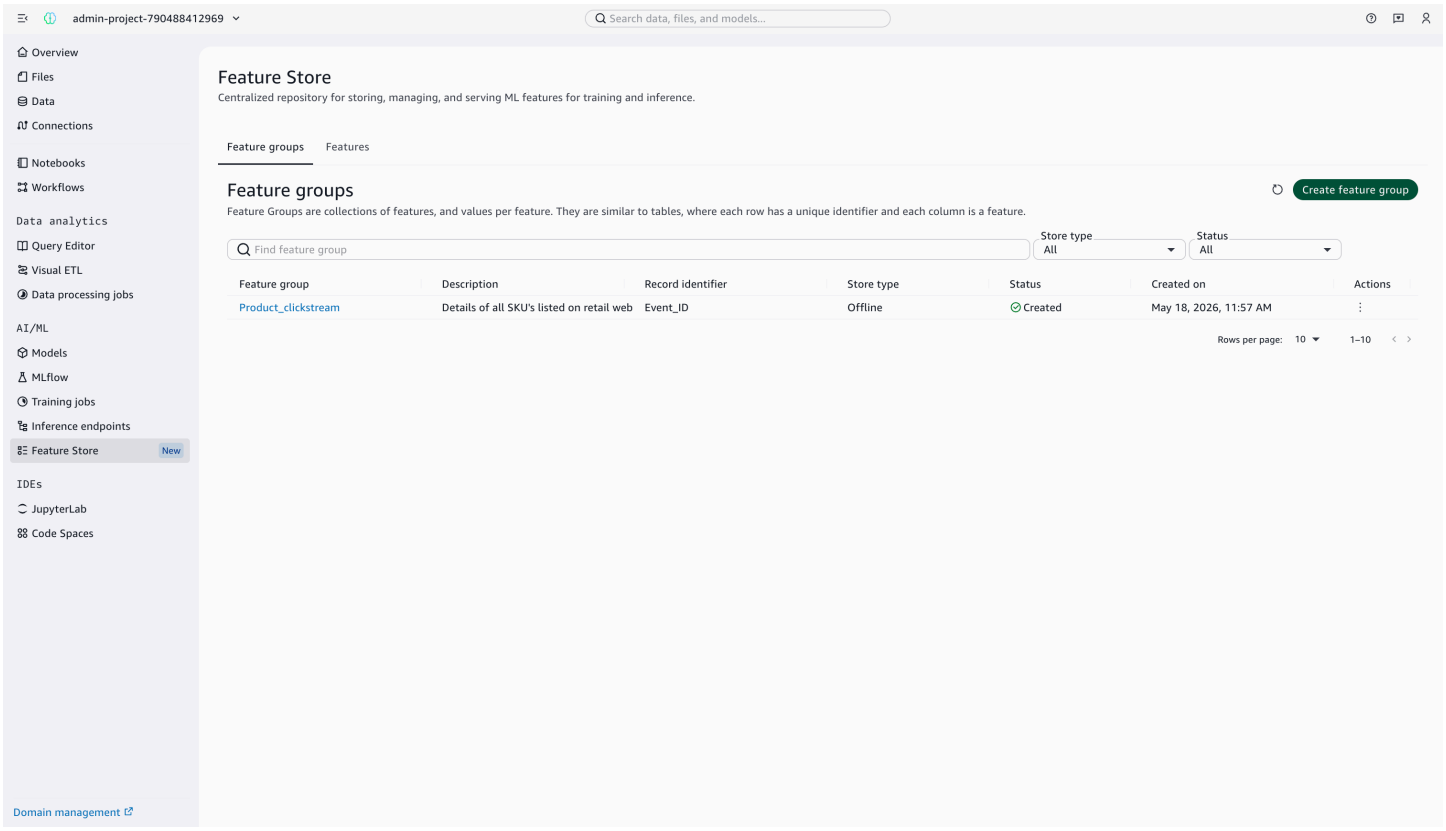
The Feature Store landing page displays two tabs: **Feature groups** and **Features**. The **Features** tab lets you search across individual features in all feature groups without selecting a specific group first.

Browse feature groups

The **Feature groups** tab lists all feature groups in your project. The following table describes the columns displayed.

Column	Description
Feature group	The name of the feature group. Choose the name to open the detail page.
Description	A brief description of the feature group
Record identifier	The feature designated as the unique record identifier
Store type	The storage configuration: Online, Offline, or Online and Offline
Status	The current status of the feature group (for example, Created)
Created on	The date and time the feature group was created
Actions	Additional actions menu

You can filter feature groups by store type and status using the dropdown filters, or search by name using the search bar.



Create a feature group

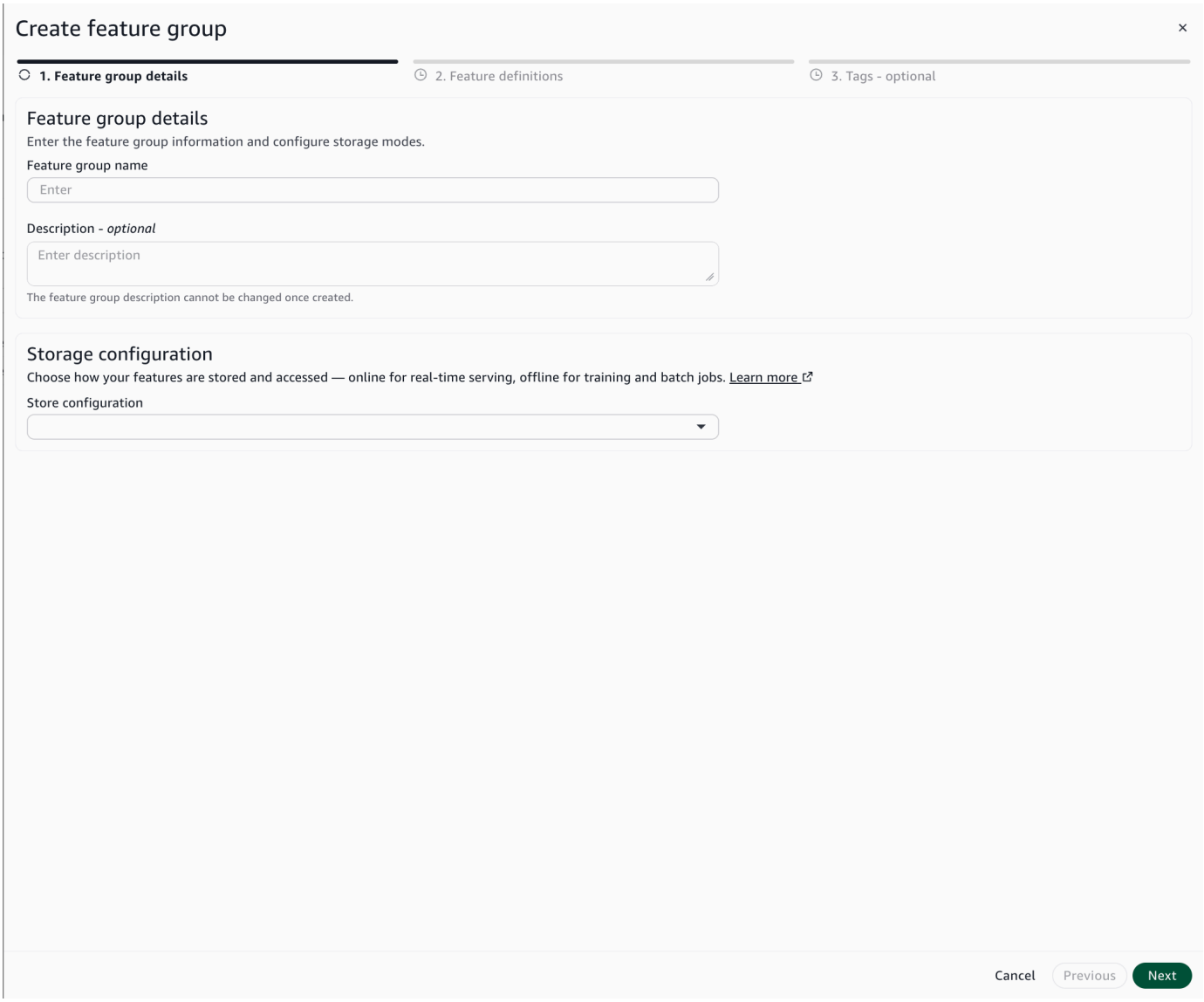
1. On the Feature Store page, choose **Create feature group**.

   The Create feature group wizard opens with three steps.

2. In Step 1 (Feature group details), enter the following information:

Field	Description
Feature group name	A unique name for your feature group. Feature group names are unique within an AWS Region and account.
Description (optional)	A text description of the feature group. You cannot change this after creation.
Store configuration	Choose how your features are stored and accessed: Online for real-time serving, Offline for training and batch jobs, or

Field	Description
	both. Online store feature groups with the InMemory storage type do not support replication to the offline store. For the offline store table format, choose Apache Iceberg in most use cases for better query performance. For more information, see Online store and Offline store.



3. Choose **Next**.

4. In Step 2 (Feature definitions), define the input features for your ML model. Each feature requires a name and a data type. You must also designate one feature as the record identifier and one as the event time feature.

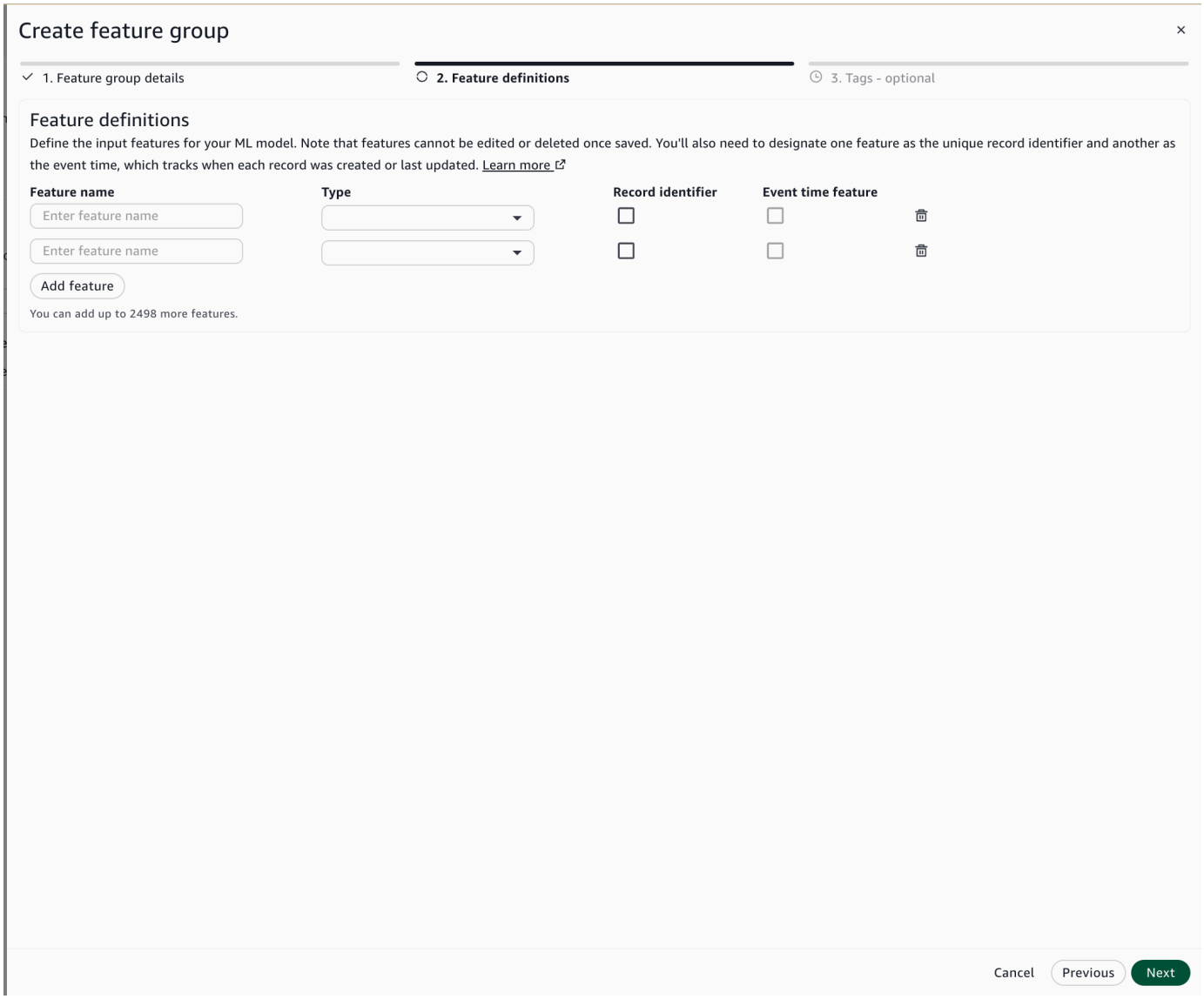
Field	Description
Feature name	The name of the feature
Type	The data type. Scalar types include String, Integral, and Fractional. For feature groups that use the in-memory online store, collection types are also available: List, Set, and Vector.
Record identifier	Designates the feature that serves as the unique identifier for each record
Event time feature	Designates the feature that tracks when each record was created or last updated

You can add up to 2,500 features per feature group. Choose **Add feature** to add additional rows.

The following table describes the collection types available for feature groups that use the in-memory online store. For more information, see Collection types.

Collection type	Description
List	An ordered collection of elements. Allows duplicate values.
Set	An unordered collection of unique elements.
Vector	A fixed-size array of Fractional elements. Maximum dimension of 8192. Only one vector collection type is allowed per feature group.

You cannot edit or delete features after the feature group is saved.



5. Choose **Next**.
6. In Step 3 (Tags - optional), add tags to organize and identify your feature group.
7. Choose **Submit** to create the feature group.

After creation, feature groups are mutable and can evolve their schema by adding new feature definitions.

View feature group details

To view the details of a feature group, choose its name from the feature groups list. The detail page opens with tabs for **Overview**, **Features**, **Sample notebook**, and **Tags**.

The **Overview** tab displays the following information.

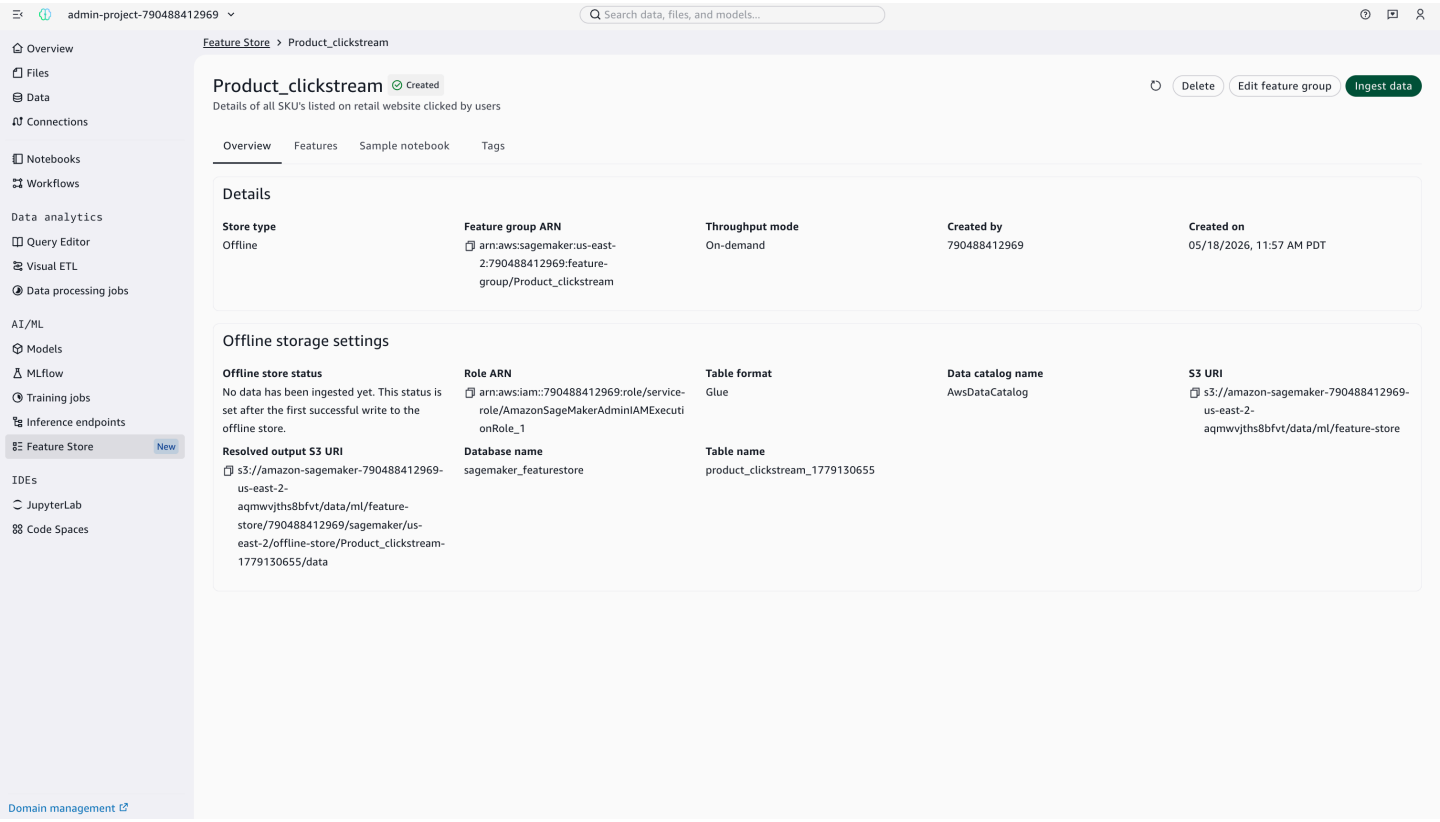
Details

Field	Description
Store type	Online, Offline, or Online and Offline
Feature group ARN	The Amazon Resource Name for the feature group
Throughput mode	The read and write capacity mode (for example, On-demand)
Created by	The account ID that created the feature group
Created on	The creation date and time

Offline storage settings (displayed for feature groups with offline storage)

Field	Description
Offline store status	When null, indicates that no data has been replicated to the offline store. When non-null, indicates whether replication into the offline store has encountered a failure. A Blocked status may include a reason for the failure.
Role ARN	The IAM role used for offline store operations
Table format	The table format used to catalog the data (for example, Glue Data Catalog)

Field	Description
Data catalog name	The catalog where the feature group table is registered
S3 URI	The Amazon S3 location for offline store data
Resolved output S3 URI	The full Amazon S3 path for offline store output
Database name	The database that contains the feature group table
Table name	The table name in the data catalog

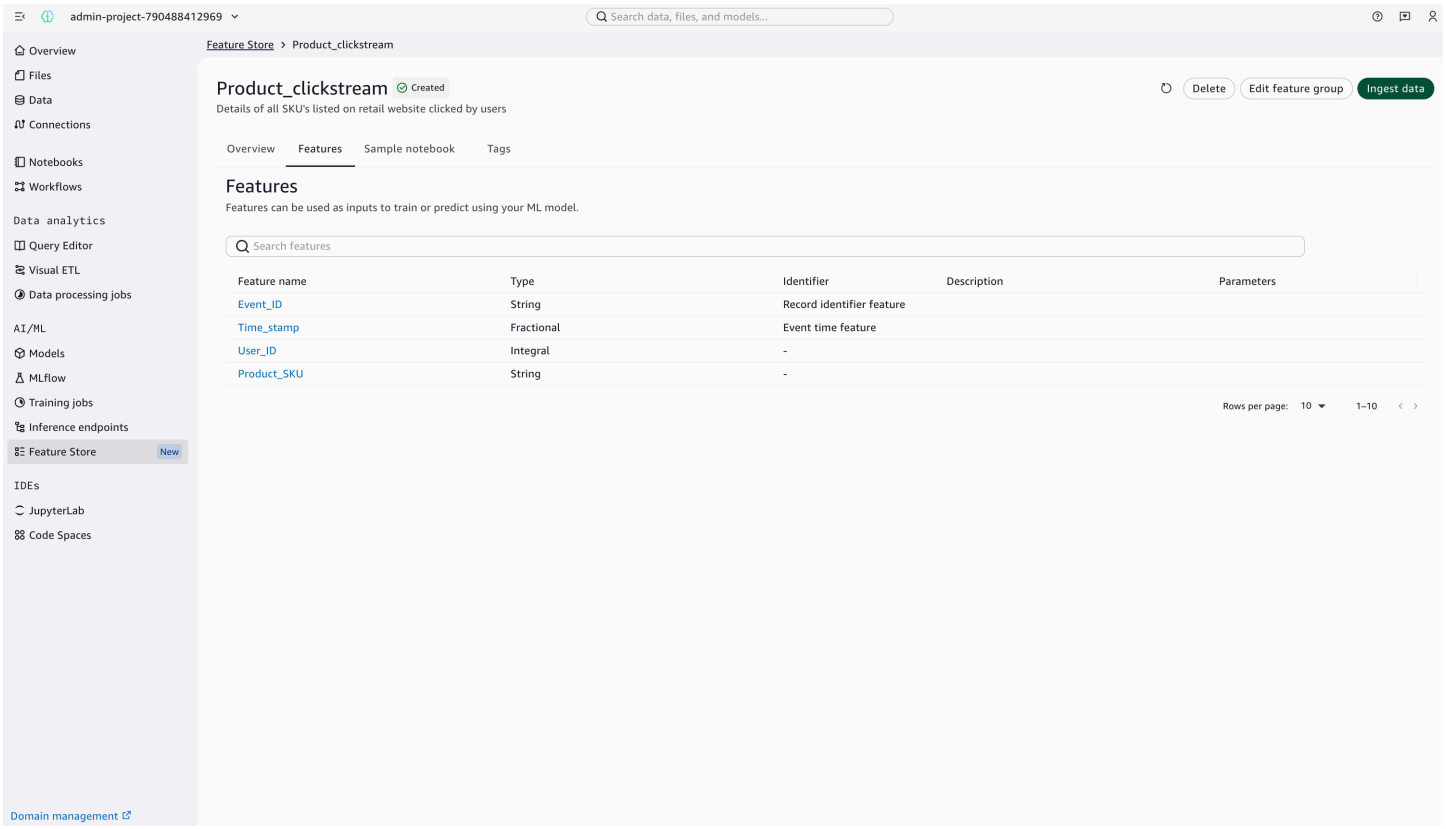


View features in a feature group

On the feature group detail page, choose the **Features** tab to see all features defined in the feature group. The following table describes the columns displayed.

Column	Description
Feature name	The name of the feature. Choose the name to view feature details.
Type	The data type. Scalar types include String, Integral, and Fractional. For feature groups that use the in-memory online store, collectio n types are also available: List, Set, and Vector.
Identifier	Indicates if the feature is the record identifier or event time feature
Description	An optional description of the feature
Parameters	Any additional searchable parameter s associated with the feature. For more information, see Adding searchable metadata to your features.

You can search features by name using the search bar.



Edit a feature group

After creation, you can add new feature definitions to a feature group, but you cannot modify or remove existing features. You can also change the throughput configuration.

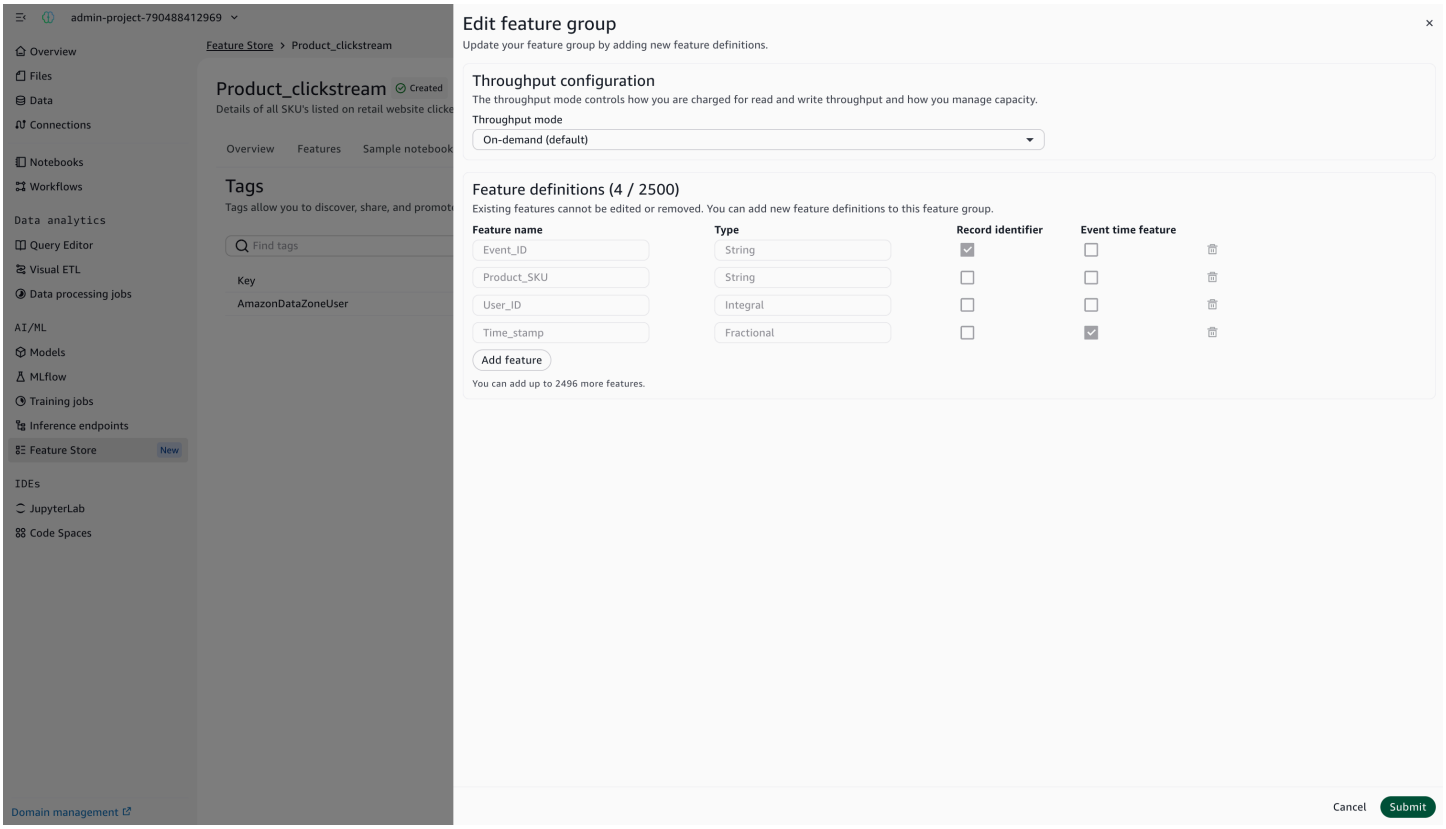
1. On the feature group detail page, choose **Edit feature group**.
2. In the Edit feature group panel, you can modify the following settings:

   Throughput configuration

   The throughput mode controls how you are charged for read and write throughput and how you manage capacity. The default is On-demand.

   Feature definitions

   Add new features by entering a feature name and selecting a type (String, Integral, or Fractional). For feature groups that use the in-memory online store, you can also select collection types: List, Set, and Vector.
3. Choose **Submit** to save your changes.



Ingest data with sample notebooks

Each feature group includes a **Sample notebook** tab that provides a pre-configured notebook template for ingesting data and reading it back from your feature group.

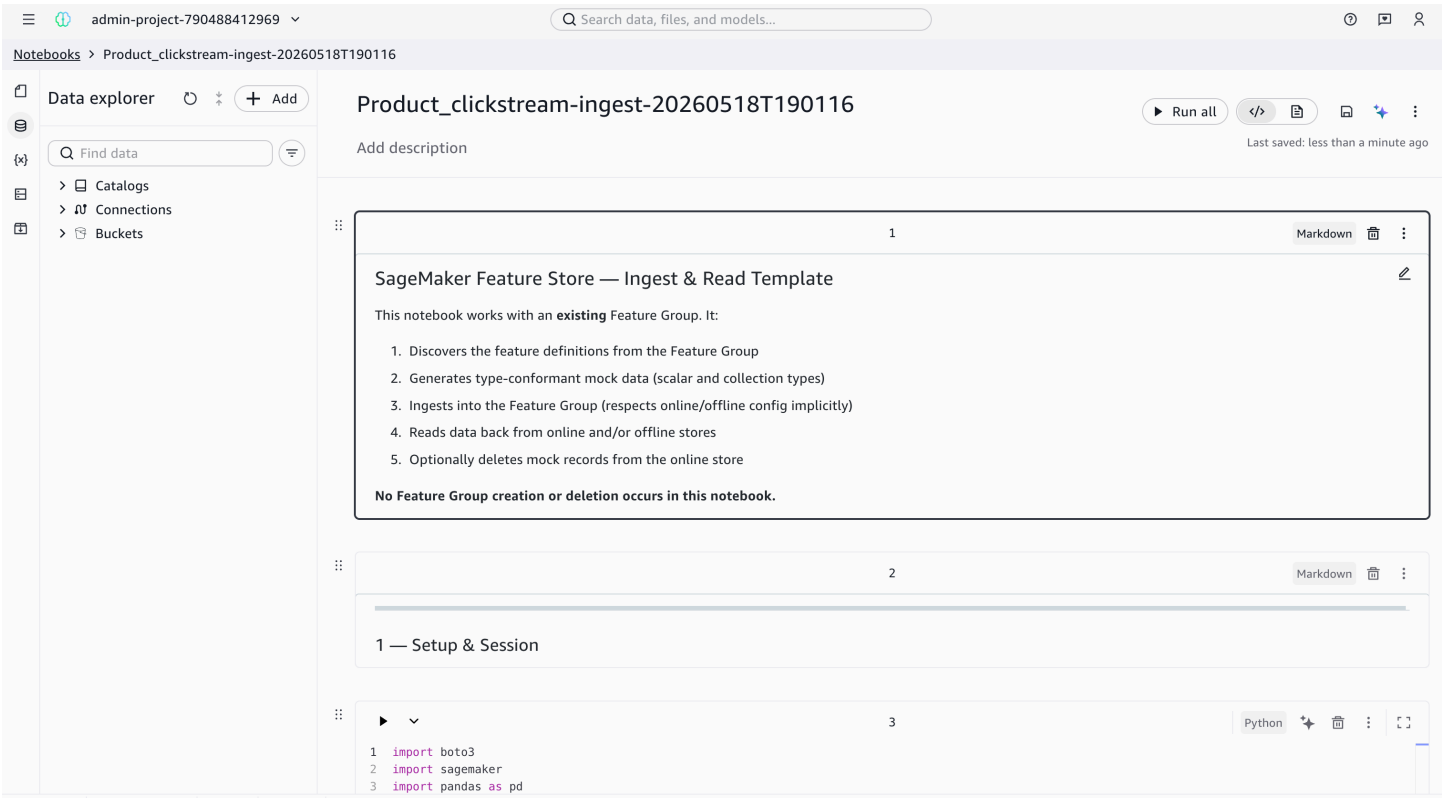
The sample notebook works with the existing feature group and performs the following tasks:

1. Discovers the feature definitions from the feature group
2. Generates type-conformant mock data (scalar and collection types)
3. Ingests data into the feature group (respects online/offline configuration)
4. Reads data back from online or offline stores
5. Optionally deletes mock records from the online store

To use the sample notebook, choose the **Sample notebook** tab on your feature group detail page. You can open the notebook in JupyterLab or Code Spaces from the IDE options in the left navigation.

For more information about additional ingestion methods available through the SDK and API, including streaming and batch ingestion, see Data sources and ingestion.

To build training datasets by joining multiple feature groups with point-in-time accuracy, use the SageMaker Python SDK from a notebook in your project. For more information, see Create a dataset from your feature groups.



Delete a feature group

1. On the feature group detail page, choose **Delete**.
2. Confirm the deletion when prompted.

Deleting a feature group removes the feature group and its metadata. Data stored in the offline store (Amazon S3) is not automatically deleted.

To delete individual records from a feature group, use the `DeleteRecord` API from a notebook in your project. Feature Store supports hard delete (permanently removes the record) and soft delete modes. For more information, see Delete a feature group.

Related resources

- Create, store, and share features with Feature Store: Overview of Feature Store concepts and architecture

- Feature Store concepts: Detailed definitions of feature groups, records, online and offline stores
- Use Feature Store with SDK for Python (Boto3): Create and manage feature groups programmatically
- Data sources and ingestion: Streaming and batch ingestion methods
- Batch ingestion with Amazon SageMaker Feature Store Spark: Use the Spark connector to batch ingest features from Amazon EMR
- Find features in your feature groups: Search and discover features across feature groups
- Online store: Storage tiers and configuration for real-time serving
- Security and access control: KMS encryption, access control, and IAM policies for Feature Store

Common use cases

Topics

- Train and deploy an XGBoost model

Train and deploy an XGBoost model

In this walkthrough, you build a customer churn prediction model using XGBoost in Amazon SageMaker Unified Studio. You:

- Download and explore the IBM Telco Customer Churn dataset
- Prepare tabular data and engineer features
- Train an XGBoost model with a SageMaker AI training job
- Track the experiment and register the model with MLflow
- Deploy the model to a real-time inference endpoint
- Test predictions on held-out data

This walkthrough is intended for data scientists and ML engineers. It assumes you are familiar with Python, SQL, and basic machine learning concepts.

Overview

XGBoost (extreme gradient boosting) is a popular open-source implementation of the gradient-boosted trees algorithm that handles classification, regression, and ranking problems. Using XGBoost with Amazon SageMaker Unified Studio gives you:

- **Managed training** – Run training as a SageMaker AI training job on a dedicated instance, separate from your notebook.
- **Experiment tracking** – Log parameters and metrics to a managed MLflow tracking server and register the resulting model.
- **Managed hosting** – Deploy the trained model to a real-time endpoint with a single call.

Prerequisites

Complete the following before you begin:

- You have an AWS account with permissions to create SageMaker AI resources. For setup instructions, see Set up an AWS account and create an administrator user.
- You have a Amazon SageMaker Unified Studio domain with a user configured. For domain setup, see Quick setup. For user access, see Managing users.
- You have a Amazon SageMaker Unified Studio project created with the **All capabilities** project profile. For instructions, see Create a project.
- You have an MLflow tracking server created in your project. Copy the tracking server ARN — you supply it in the section called "Step 5: Track and register the model with MLflow". For instructions, see Track experiments using MLflow.

You download the dataset in Step 2 — you do not need to bring your own data.

Step 1: Open your notebook and configure your project bucket

This walkthrough supports two notebook environments inside Amazon SageMaker Unified Studio. Launch either one using the following steps.

Option A: SageMaker Notebooks

1. Open your Amazon SageMaker Unified Studio project.
2. In the navigation pane, choose **Notebooks**.
3. Choose **Create Notebook**, which provisions a SageMaker AI notebook by default.

> **Note**
>
> The screenshots in this walkthrough show the SageMaker Notebooks experience.

Option B: JupyterLab

1. Open your Amazon SageMaker Unified Studio project.
2. In the navigation pane, under **IDEs**, choose **JupyterLab**.

3. Wait for the space to be ready, then create a new notebook with a **Python 3** kernel.
4. Install the SageMaker AI SDK and restart the kernel:

```
%pip install --upgrade sagemaker
```

After the install completes, restart the kernel (**Kernel** → **Restart Kernel**) before continuing.

The remaining steps are the same in both environments unless explicitly noted.

Configure your project bucket

Find your project Amazon S3 bucket and prefix. Choose the three-dot menu next to **Shared** and copy the bucket URI. The URI has the form `s3://<bucket>/shared`.

Example bucket and prefix values

```
# Example bucket and prefix values
BUCKET = "amazon-sagemaker-<account-id>-<region>-<suffix>"
PREFIX = "shared"
```

> **Important**
>
> Use your values any time the code sets a variable called `bucket` or `prefix` in the following steps.

Step 2: Download and explore the dataset

This walkthrough uses the publicly available IBM Telco Customer Churn dataset (7,043 customer records).

Download the dataset in your notebook by running the following code in a cell:

Download the dataset

```
import requests

url = "https://raw.githubusercontent.com/IBM/telco-customer-churn-on-icp4d/master/data/
Telco-Customer-Churn.csv"
filename = "Telco-Customer-Churn.csv"
```

```
response = requests.get(url)
response.raise_for_status()

with open(filename, 'wb') as f:
    f.write(response.content)
```

Using JupyterLab instead of SageMaker Notebooks?

Open a terminal in JupyterLab (**File** → **New** → **Terminal**) and download the dataset:

```
wget https://raw.githubusercontent.com/IBM/telco-customer-churn-on-icp4d/master/
data/Telco-Customer-Churn.csv -O Telco-Customer-Churn.csv
```

In your notebook, load and review the data:

Load and review the data

```
import pandas as pd

raw = pd.read_csv("Telco-Customer-Churn.csv")
raw.head()
```

The dataset contains one row per customer with 20 feature columns (including `tenure`, `MonthlyCharges`, `TotalCharges`, `Contract`, `InternetService`) and the target column `Churn` (Yes/No).

Step 3: Prepare the data for XGBoost

Clean the data, engineer features, and place the label in the first column:

Clean data and engineer features

```
import numpy as np

raw["TotalCharges"] = pd.to_numeric(raw["TotalCharges"], errors="coerce")
raw = raw.dropna(subset=["TotalCharges"])

model_df = pd.DataFrame({
```

```
    "churn_label":       (raw["Churn"] == "Yes").astype(int),
    "tenure_months":     raw["tenure"].astype(int),
    "monthly_charges":   raw["MonthlyCharges"].astype(float),
    "total_charges":     raw["TotalCharges"].astype(float),
    "lifetime_value":    raw["MonthlyCharges"].astype(float) *
 raw["tenure"].astype(int),
    "is_month_to_month": (raw["Contract"] == "Month-to-month").astype(int),
    "is_fiber_optic":    (raw["InternetService"] == "Fiber optic").astype(int),
})
```

Split into train (70%), validation (15%), and test (15%) sets and upload to Amazon S3:

Split and upload to S3

```
import boto3, sagemaker

bucket  = BUCKET
prefix  = PREFIX
session = sagemaker.Session(default_bucket=bucket)

shuffled = model_df.sample(frac=1, random_state=1729).reset_index(drop=True)
n = len(shuffled)

train      = shuffled.iloc[: int(0.70 * n)]
validation = shuffled.iloc[int(0.70 * n) : int(0.85 * n)]
test       = shuffled.iloc[int(0.85 * n) :]

train.to_csv("/tmp/train.csv", header=False, index=False)
validation.to_csv("/tmp/validation.csv", header=False, index=False)
test.to_csv("/tmp/test.csv", index=False)

for name in ["train", "validation", "test"]:
    boto3.Session().resource("s3").Bucket(bucket).Object(
        f"{prefix}/datasets/{name}/{name}.csv"
    ).upload_file(f"/tmp/{name}.csv")

print("Splits written to", f"s3://{bucket}/{prefix}/datasets/")
```

After the cell finishes, the files appear in your notebook file browser.

Step 4: Train the XGBoost model

Configure and launch the training job:

Configure and launch training job

```
import sagemaker
from sagemaker.estimator import Estimator
from sagemaker.inputs import TrainingInput
from sagemaker import image_uris, get_execution_role

bucket  = BUCKET
prefix  = PREFIX
session = sagemaker.Session(default_bucket=bucket)
region  = session.boto_region_name
role    = get_execution_role()

image_uri = image_uris.retrieve(framework="xgboost", region=region, version="1.7-1")

xgb = Estimator(
    image_uri=image_uri,
    role=role,
    instance_count=1,
    instance_type="ml.m5.xlarge",
    output_path=f"s3://{bucket}/{prefix}/model",
    sagemaker_session=session,
)

xgb.set_hyperparameters(
    objective="binary:logistic",
    eval_metric="auc",
    num_round=200,
    max_depth=5,
    eta=0.2,
    subsample=0.8,
)

train_in = TrainingInput(f"s3://{bucket}/{prefix}/datasets/train/", content_type="csv")
val_in   = TrainingInput(f"s3://{bucket}/{prefix}/datasets/validation/",
 content_type="csv")

xgb.fit({"train": train_in, "validation": val_in})
```

> **Note**
>
> Training takes approximately 5 minutes. Monitor progress under **AI/ML** → **Training jobs** in Amazon SageMaker Unified Studio.

Step 5: Track and register the model with MLflow

Log hyperparameters and register the trained model:

Log and register model with MLflow

```
import os, mlflow

mlflow.set_tracking_uri("<your-mlflow-tracking-server-arn>")
mlflow.set_experiment("customer-churn-xgboost")

with mlflow.start_run(run_name="xgboost-churn") as run:
    mlflow.log_params(xgb.hyperparameters())
    model_uri = f"s3://{bucket}/{prefix}/model/{xgb.latest_training_job.name}/output/
model.tar.gz"
    mlflow.register_model(model_uri=model_uri, name="customer-churn-xgboost")
```

To view the experiment and registered model, choose **MLflow** in the navigation pane, open the **MLflow Tracking Servers** tab, and choose **Open MLflow**. In the MLflow UI, choose **Models** in the navigation pane to see the registered `customer-churn-xgboost` model.

Step 6: Deploy the model to a real-time endpoint

Deploy the model to a real-time endpoint:

Deploy to endpoint

```
from sagemaker.serializers import CSVSerializer

predictor = xgb.deploy(
    initial_instance_count=1,
    instance_type="ml.m5.large",
    serializer=CSVSerializer(),
```

```
)
```

> **Important**
>
> The endpoint runs on a dedicated `ml.m5.large` instance and incurs costs as long as it is in service. Delete the endpoint when you are finished testing (see the section called “Step 8: Clean up”).

Monitor deployment status under the **Inference endpoints** tab in the navigation pane. The endpoint is ready when status shows **InService**.

Step 7: Test predictions

Score the test set:

Score the test set

```
test = pd.read_csv(f"s3://{bucket}/{prefix}/datasets/test/test.csv")
features = test.drop(columns=["churn_label"])

scores = predictor.predict(features.to_numpy()).decode("utf-8")
test["churn_probability"] = [float(s) for s in scores.strip().split()]

test[["churn_label", "churn_probability"]].head(10)
```

Values closer to `1.0` indicate higher churn likelihood.

Troubleshooting

The following table lists common issues and their resolutions.

Issue	Resolution
`AttributeError: module 'sagemaker' has no attribute 'Session'`	Run `%pip install --upgrade sagemaker` , restart the kernel, then re-run.

Issue	Resolution
`S3UploadFailedError: AccessDenied` when uploading training data	In Amazon SageMaker Unified Studio, your IAM role has `s3:PutObject` permission only on your project path (for example, `s3://bucket/dzd-<domain-id>/<project-id>/dev/*`). Include the full project prefix when constructing Amazon S3 paths.
`ResourceLimitExceeded` on training job	Check **Service Quotas** → **Amazon SageMaker AI** for the instance type, or switch to one with available quota.
Poor validation AUC	Increase `num_round`, tune `max_depth` and `eta`, or add more features.
Endpoint stuck in **Creating**	Check CloudWatch logs for the endpoint. Verify instance availability in your Region.
Older SageMaker AI SDK causes import or attribute errors after upgrade	Run `%pip install --upgrade --force-reinstall sagemaker` and restart the kernel.

For more information, see the SageMaker AI troubleshooting guide.

Step 8: Clean up

To avoid ongoing charges, delete the resources that you created in this walkthrough.

1. Delete the endpoint:

```
predictor.delete_endpoint()
```

2. Delete the Amazon S3 datasets and model artifacts at `s3://{BUCKET}/{PREFIX}/`.
3. If you created a domain for this walkthrough, delete it. For instructions, see Deleting a domain.

aws

User Guide

Amazon SageMaker Unified Studio

Platform and administration



Projects

A project in Amazon SageMaker Unified Studio is a boundary within a domain where you can collaborate with other users to work on a business use case. In projects, you can create and share data and resources. When you are added to a project, you gain access to relevant files and tools within that project, such as resources to create and deploy machine learning models. You can create new files and resources for the project, manage existing resources, and share work with other users on the same project.

After you gain access to Amazon SageMaker Unified Studio, you can create a new project or become added to an existing project by another user. The tools and resources available in the project are based on the project profile that is used to create a project.

There are two types of domains that your admin can create in the Amazon SageMaker Unified Studio management console. If your admin grants you access to an Amazon DataZone domain, see Amazon DataZone projects and environments in the Amazon DataZone User Guide. The sections in this guide describe projects in an Amazon SageMaker unified domain.

> **Note**
>
> To join a project, you must create a project yourself or be added to a project by a project owner.

Topics

- Create a new project
- Get project details
- Edit a project
- Update a project
- Move a project to a different domain unit
- Delete a project
- Add project members
- Remove project members
- Using a project with trusted identity propagation enabled

Create a new project

In Amazon SageMaker Unified Studio, projects enable a group of users to collaborate on various business use cases. Within projects, you can manage data assets in the Amazon SageMaker Unified Studio catalog, perform data analysis, organize workflows, develop machine learning models, build generative AI apps, and more.

In order to create a project in Amazon SageMaker Unified Studio, you must gain access to Amazon SageMaker Unified Studio. A domain unit owner must also grant you access to create projects through an authorization policy. For more information, see Domain units and authorization policies in Amazon SageMaker Unified Studio.

When you create a project, you choose a name and description, customize parameters for project resources, and then review selections.

Step 1: Project name and description

To begin creating a project, navigate to the Amazon SageMaker Unified Studio landing page and choose **Create project**.

The Project name and description includes the following fields:

- Project name - The name of your Amazon SageMaker Unified Studio project. Enter a name here. The name of the project can not be edited after the project is created.
- Description - An optional description of your project. You can edit this later.
- Domain unit - The business-level entity that lets your team organize and manage policies for business needs in the project. If nobody in the domain has created domain units, you create a project in the root domain unit by default and no action is needed here. If domain units have been created, select the name of the domain unit you want your project to be in. For more information, see Domain units and authorization policies in Amazon SageMaker Unified Studio.
- Project profile - Project profiles define which resources and tools should be provisioned in the project. These include tools and compute resources for SQL, data science, data engineering, and machine learning development. Project profiles can include resources and tools from Amazon Redshift, Amazon SageMaker AI, and other AWS services. Select the project profile that contains the resources and tools you will need to use in your project. The project profiles available for you to choose from are defined by your administrator in the Amazon SageMaker Unified Studio management console. For more information, see the Amazon SageMaker Unified Studio Administrator Guide.

After you fill in the fields for project creation, choose **Continue** to customize parameters.

Step 2: Customize parameters

On the next page of project creation, select a project file storage — Amazon S3 storage or Git repository for your project code artifacts. You can view and edit the names and values for different resources that are created when the project is created.

For S3 storage, you will be provided with a link to a shared folder. For more information on Amazon S3 storage for Amazon SageMaker Unified Studio, refer to Unified storage in Amazon SageMaker Unified Studio storage.

> **Note**
>
> Some of the parameter values might be determined by your admin or by the default value from the environment blueprint, according to the configurations that your admin has set in the Amazon SageMaker Unified Studio management console. If you are not able to view or change a parameter value that you want to specify, contact your admin to edit the configurations. For more information, see Manage Tooling blueprint parameters.

Connect to a Git repository

As part of this process, if your admin has configured the parameters to be editable, you can choose a Git repository to connect to your project. You can choose to connect your project to an existing third-party Git repository or create a new Git repository to connect to.

> **Important**
>
> When you connect a project to a third-party Git repository, all users who can sign in to any domain in the account have read and write access to all repositories on that connection. This access is not limited to the project or domain where the connection was created. To enforce isolation between repositories, use separate AWS accounts. Do not store sensitive information in connected repositories unless all users in the account are authorized to access it.

To connect to an existing 3P Git repository

1. In the Git connection dropdown, select a connection from AWS CodeConnections that is enabled for Amazon SageMaker Unified Studio. Available Git connections are provided by your administrator in the Amazon SageMaker Unified Studio management console.
2. Select either the Existing repository and existing branch or Existing repository and new branch radio button. Then in the Repository name dropdown, you will see a list of repositories accessible with the connection.
3. Select the name of the repository you want to connect your project to.
4. In the branch name dropdown, either select an existing branch, or enter a branch name to create a new one.

To create a new Git repository

1. In the Git connection dropdown, select a connection from AWS CodeConnections that is enabled for Amazon SageMaker Unified Studio. Available Git connections are provided by your administrator in the Amazon SageMaker Unified Studio management console.
2. Select the New repository and new branch radio button.
3. Provide a name for the repository.
4. Enter a name for the branch you want to create within the new repository. The new repository and branch will then be created when the project is created.

Depending on which project profile you are using to create a project and what parameters your admin has configured to be editable, you might have other fields to choose parameters for.

When you have chosen the parameters you want, choose **Continue** to review the selections.

Step 3: Review

Use the last page of project creation to review the configurations you have selected. When everything is configured as desired on the project creation review page, choose **Create project**.

You are then redirected to the project home page. It might take a few minutes before the project is created and you can access tools.

Next steps

After you create a project, you can add members and resources to the project and begin using tools. There are many ways to get started building your project, including the following options:

- Add members to your project to collaborate together. For more information, see the section called "Add project members".
- Add data to your project. For more information, see the Data storage section.
- Add compute resources to your project. For more information, see the Compute section.
- Find, train, and deploy machine learning models. For more information, see *Machine learning*.
- Use Amazon Bedrock in SageMaker Unified Studio to create generative AI apps. For more information, see *Amazon Bedrock in SageMaker Unified Studio*.
- Deploy applications across environments with the CI/CD CLI. For more information, see *CI/CD*.

Get project details

An Amazon SageMaker Unified Studio project contains essential configuration details that you may need to access for various development, deployment, and administrative tasks. The following section provides information on how to navigate to your **Project details** page, which includes key information and metadata. The **Project details** page includes:

- Project ID
- Project role ARN
- Amazon S3 location
- Domain unit name
- Domain ID

To navigate to your Project details page

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your IAM Identity Center (SSO) or AWS credentials.
2. If not open already, navigate to the Amazon SageMaker Unified Studio home page by choosing the icon located at the top left corner of the page.
3. Under **Your projects**, choose your project.
4. If not open already, choose the **Project details** tab.

View the SageMaker AI domain details associated with your project

A SageMaker AI domain is created for your Amazon SageMaker Unified Studio Project. You may need to update your SageMaker AI domain to update your Amazon SageMaker Unified Studio Project. Use the following instructions to get the associated SageMaker AI domain details.

> **Note**
>
> The domain ID in Get project details is *not* the same as your SageMaker AI domain ID.

To get your SageMaker AI domain details

1. Navigate to your **Project details** tab. For more information, see Get project details.
2. Copy your **Project ID**.
3. Navigate to the Amazon Amazon SageMaker AI console.
4. Expand the **Admin configurations** section.
5. Under **Admin configurations**, choose **Domains**.
6. From the list of domains, find the domain name that contains the **Project ID** you copied above. You can use the search function on that page.
7. Choose the hyperlink of your domain to view your **Domain details**.

Edit a project

In Amazon SageMaker Unified Studio, projects enable a group of users to collaborate on various business use cases. Within projects, you can manage data assets in the Amazon SageMaker Unified Studio catalog, perform data analysis, organize workflows, develop machine learning models, build generative AI apps, and more.

To edit an Amazon SageMaker Unified Studio project, you must be the owner of that project or the domain administrator of the domain that contains this project. Project owners can edit the project in the following ways:

- Adding members to a project. For more information, see the section called “Add project members”.
- Removing members from a project. For more information, see the section called “Remove project members”.

- Updating project parameters. For more information, see the section called “Update a project”.
- Changing a project description.

To change the project description in an existing project, complete the following steps.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project that you want to edit.
3. On the **Project overview** page, choose the edit icon next to **Description**. An editable text box appears.
4. Input the changes you want to make in the text box.
5. Choose the checkmark to save your changes.

Update a project

You and your admin can make changes to project settings and parameters by updating a project. There are three different kinds of project updates:

- Service-initiated updates. Projects are made up of blueprints that set up resources using different AWS services, and sometimes these services require updates to their parameters due to feature changes or security initiatives.
- Admin-initiated updates. The Amazon SageMaker Unified Studio domain admin can edit or update blueprint parameters for a project in the Amazon SageMaker Unified Studio management console after that project has been created. When this happens, project owners in Amazon SageMaker Unified Studio must update the project for the changes to take effect.
- Owner-initiated updates. Amazon SageMaker Unified Studio project owners can update certain parameters. Some parameters are not modifiable, and other parameters are modifiable if the domain admins configure them to be editable in the Amazon SageMaker Unified Studio management console. The project owners can then make updates to modifiable parameters in Amazon SageMaker Unified Studio.

When a service-initiated update or admin-initiated update occurs and a project is eligible for updating, a blue banner appears to notify project owners of the change. Project owners can then update the project to implement the changes.

> **Note**
>
> If a service or admin has initiated an update by updating relevant blueprints, the project owner must update the project so that project members can use all project capabilities. The project capabilities might be limited until the project owner completes the update.

Project owners can initiate updates to some project parameters if their domain admin has made those parameters editable.

To update a project, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project.
3. On the **Project overview** page, choose **Actions > Update project**.
4. (Optional) Review the information in the **Domain updates** section, if applicable.
5. Review the information in the **Project profile updates** section.
6. Review the information in **Environment updates**.
    - Choose **Show update parameters** in each section to review and update the modifiable parameters as desired. If there are other parameters that you want to update that are listed as non-modifiable, contact your admin.
7. Choose **Update project**. Updating a project might take a few minutes as resources are deployed.

When the update is complete, you can use your project with the updated parameters.

Move a project to a different domain unit

You can move a project from one domain unit to another. To move a project to a different domain unit in Amazon SageMaker Unified Studio, you must meet the following requirements:

- You must have a policy grant for project creation in the domain unit to which you are moving the project.
- All members of the project must have project membership permissions in the domain unit to which you are moving the project.

- You must be a domain unit owner in the domain unit to which you're moving the project.
- You must be an owner of the project.

To move an existing project to a different domian unit, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Choose **Browse projects**.
3. Choose the project that you want to move.
4. Expand **Actions** and choose **Move project**.
5. Select the domain unit that you want to move this project to and then choose **Move project**.

Delete a project

The act of deleting a project is final. Deletion irrevocably deletes the project's contents, including compute instances, data sources, queries, and more, and all the content within those project resources. Deleting a project does not delete non-Amazon SageMaker Unified Studio AWS resources that Amazon SageMaker Unified Studio might have helped you create. If you no longer need these AWS resources, delete them in their respective AWS service and account. For example, if you create Amazon Bedrock model evaluation jobs in Amazon Bedrock in SageMaker Unified Studio, the jobs that aren't automatically deleted when you delete the project. You will need to use the Amazon Bedrock console to delete the model evaluation jobs. Contact your administrator if you don't have access to the Amazon Bedrock in SageMaker Unified Studio console.

To delete an Amazon SageMaker Unified Studio project, you must be an owner of the project.

To delete an existing project, complete the following steps.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project that you want to delete.
3. On the **Project overview** page, expand **Actions** and choose **Delete project**.

   Review the informational warnings about the potential impact of deleting the project.
4. If you accept the warnings, then type in the confirmation text and choose **Delete project**.

> **Important**
>
> Deleting a project is an irrevocable action that cannot be undone by you or by AWS.

Add project members

In Amazon SageMaker Unified Studio, projects enable a group of users to collaborate on various business use cases. Within projects, you can manage data assets in the Amazon SageMaker Unified Studio catalog, perform data analysis, organize workflows, develop machine learning models, build generative AI apps, and more.

A project member can be a user or a group of users, and a project can have a maximum of 20 members.

There are two different roles that members can have:

- **Contributor**: These members can contribute to the project.
- **Owner**: These members can contribute to the project, add members to the project, and remove members of the project. They can also edit the project description or delete the project. The person who creates the project has the role of Owner by default.

You can add the following types of members:

- Single sign-on (SSO) users. An SSO user can sign into Amazon SageMaker Unified Studio using credentials from IAM Identity Center or another SSO source.
- SSO groups. You can add groups of users created in IAM Identity Center. An SSO group is considered one project member.
- IAM principals (roles or users). An IAM user or role can sign into Amazon SageMaker Unified Studio with their IAM credentials. They can also contribute to the project programmatically.

To add members to an existing project, complete the following steps.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project that you want to edit.

3. On the **Project overview** page, expand **Actions** and choose **Manage members**.
4. Choose **Add members**.
5. Enter the name of the user or group you want to add in the search bar, and select the name from the list.
6. Select **Contributor** if you want to add the project member as a contributor, or choose **Owner** if you want to add the project member as a project owner.
7. (Optional) Repeat these steps to add more project members. You can add up to 8 project members at a time.
8. Choose **Add members**.

Remove project members

In Amazon SageMaker Unified Studio, projects enable a group of users to collaborate on various business use cases. Within projects, you can manage data assets in the Amazon SageMaker Unified Studio catalog, perform data analysis, organize workflows, develop machine learning models, build generative AI apps, and more.

A project member can be a user or a group of users, and a project can have a maximum of 20 members. To remove members from a project, you must be an owner of that project.

To remove members from an existing project, complete the following steps.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project that you want to edit.
3. On the **Project overview** page, expand **Actions** and choose **Manage members**.
4. Select the member or members you want to remove from the project. You can remove up to 8 members at a time.
5. Choose **Remove members**. A popup window appears so that you can confirm this action.
6. Choose **Remove members**.

Members that you remove from the project will no longer have access to files and data in that project. The assets that were created by those project members remain in the project inventory.

Using a project with trusted identity propagation enabled

Trusted identity propagation in IAM Identity Center enables administrators of AWS services to grant permissions based on user attributes, such as user ID or group associations. With trusted identity propagation, identity context is added to an IAM role to identify the user requesting access to AWS resources. This context is propagated to other AWS services.

For more information on how to enable trusted identity propagation in Amazon SageMaker Unified Studio, see Trusted Identity Propagation in the Amazon SageMaker Unified Studio Administrator Guide.

Currently in Amazon SageMaker Unified Studio, you can use trusted identity propagation in the following use cases:

- If you want to use Amazon SageMaker Unified Studio to run Amazon Athena queries using your single sign-on (SSO) user or group credentials.
- If you want to use Amazon SageMaker Unified Studio to connect to your existing Amazon Redshift clusters or Amazon Redshift Serverless endpoints that are IAM Identity Center enabled and run Amazon Redshift queries that are authorized based on your SSO credentials.

Once your administrator has enabled trusted identity propagation for your Amazon SageMaker unified domain, you can use this feature in your Amazon SageMaker Unified Studio projects. The procedures below describe the currently supported use cases.

Topics

- Create a new project with trusted identity propagation
- Working with Amazon Athena with trusted identity propagation
- Connect to existing Amazon Redshift clusters or Serverless endpoints with trusted identity propagation

Create a new project with trusted identity propagation

First, you need to create a new project in your trusted identity propagation-enabled domain. In the current release of Amazon SageMaker Unified Studio, trusted identity propagation is only supported for projects that are created starting with 07/23/2025 and beyond. Trusted identity propagation is NOT supported for projects that were created prior to 07/23/2025 even if they live in a domain that has trusted identity propagation enabled.

You can follow the steps in Create a new project to create your trusted identity propagation enabled project. When selecting a project profile with which you create your project, make sure to select a project profile that has configured/set `enableTIPPermissions` to true.

Working with Amazon Athena with trusted identity propagation

Once you create a project with trusted identity propagation enabled, this project launches a default connection to an Amazon Athena workgroup - you can see AWS Data Catalog under Lakehouse in the query editor's data explorer tab. You can then use the query editor to write and run queries against this Amazon Athena workgroup. Your access to the data in this Amazon Athena workgroup is authorized through your SSO credentials due to trusted identity propagation.

Considerations and limitations for this use case:

- There is only one connection for Athena in SM US (by design). Setting `enableTrustedIdentityPropagation` to true on an existing project does not create a new blueprint that can authorize on trusted identity propagation credentials.
- In the current release of Amazon SageMaker Unified Studio, the default project database which provides project users with data access based on their roles is not displayed in the query editor's data explorer tab in this new project with trusted identity propagation enabled. project role-based data access, will not appear in the SQL Editor's data explorer.
- Currently, users cannot use their SSO credentials via trusted identity propagation to subscribe to or publish data. Users can create a separate project without trusted identity propagation to view the default project data catalog and use the subscriptions capability.
- When a project has trusted identity propagation enabled, as you mouse over the nodes in the data explorer tab in the query editor, you'll see a tip informing you that all data authorization in this project is based on user SSO credentials with trusted identity propagation.
- In the current release of Amazon SageMaker Unified Studio, in Jupyter Lab, Visual ETL, and other tools where Amazon Athena is used but where trusted identity propagation is currently NOT supported, data explorer displays the default project database and users are authorized to access this data based on the project role.

Connect to existing Amazon Redshift clusters or Serverless endpoints with trusted identity propagation

Within trusted identity propagation-enabled project, you can connect to an existing Amazon Redshift provisioned cluster or Amazon Redshift Serverless endpoint and then run queries against

that data with your SSO credentials. Under **Authentication**, make sure to select **IAM Identity Center** since you want to access the resource with you SSO credentials and run Amazon Redshift queries in Amazon SageMaker Unified Studio with trusted identity propagation.

Considerations and limitations for this use case:

- When trusted identity propagation is enabled in a project, there is no default connection for Amazon Redshift. Users have to explicitly create a connection to an existing Amazon Redshift cluster or serverless endpoint that supports IdC with trusted identity propagation. For more information, see Connect to existing Amazon Redshift clusters or Serverless endpoints with trusted identity propagation
- Currently, users cannot use their SSO credentials via trusted identity propagation to subscribe to or publish data products.
- Trusted identity propagation is only supported if your Amazon SageMaker Unified Studio project and your Amazon Redshift cluster or serverless endpoint are in the same AWS account.
- Currently, IAM identity Center does not support trusted identity propagation across regions. Therefore, your IAM Identity Center instance and your Amazon SageMaker unified domain must be in the same AWS account.

Source control and file storage in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio provides flexible file storage options to support your analytics, AI and ML workflows.

Amazon SageMaker Unified Studio brings together the functionality and tools from existing AWS Analytics and AI/ML services into a single data and AI development environment. As you work with different tools like JupyterLab, SQL Editor, Visual ETL Builder, or capabilities from Amazon Bedrock inside Amazon SageMaker Unified Studio you'll create and manage files that represent your work.

S3 storage

Amazon Simple Storage Service (S3) storage is the default option for storage of project files in Amazon SageMaker Unified Studio.

With S3 storage, you can easily share files by moving them between local and shared folders using simple drag-and-drop operations. The file explorer provides a consistent interface across all tools, displaying both local and shared directories in a single view with drag-and-drop functionality for easy file management. It allows users to create, edit, delete, upload, and download files directly through the interface, with optional auto-save capabilities to prevent data loss.

S3 storage provides basic file versioning capabilities when enabled by your administrator. This option is available in all AWS regions where Amazon SageMaker Unified Studio is supported, making it ideal for teams working across different geographic locations.

For more information on configuring S3 storage see Configuring project storage options.

Key benefits of S3 storage:

- Simple file management
- Easy file sharing with drag-and-drop between folders
- Availability in all regions where Amazon S3 is supported

Git-based storage

For projects requiring advanced version control, Amazon SageMaker Unified Studio allows you to connect your project to a Git repository where all project members can access, store, and collaborate on files. This option provides full version control capabilities including comprehensive commit history, branching, and merging.

When you choose Git-based storage, you'll need to specify a repository and branch during project creation. Once the project is created, you'll be able to see the files that were created during repository bootstrapping directly from the project's home page.

> **Important**
>
> When you connect a project to a third-party Git repository, all users who can sign in to any domain in the account have read and write access to all repositories on that connection. This access is not limited to the project or domain where the connection was created. To enforce isolation between repositories, use separate AWS accounts. Do not store sensitive information in connected repositories unless all users in the account are authorized to access it.

With Git-based storage, you'll have access to full Git semantics regardless of whether you're using space-based tools like JupyterLab or web-based tools like SQL Query Editor. This provides a consistent experience for team members accustomed to working with Git.

Key benefits of Git-based storage include:

- Full version control with commit history, branching, and merging
- Collaboration features like pull requests and code reviews
- Cross-project sharing by allowing multiple projects to use the same repository
- Integration with existing development workflows

Storage working in different tools

Amazon SageMaker Unified Studio provides a consistent storage experience across different tools while optimizing for each tool's specific requirements.

Web-based tools

When using web-based tools such as Query Editor and Visual ETL, you'll interact with files through a unified File Explorer interface. This explorer displays your shared directory and allows you to navigate and manage shared files seamlessly.

You can perform various file operations directly from the File Explorer:

- Create, edit, and delete files and folders
- Upload and download files to/from shared storage
- Access version history (when available)
- Edit files directly within the source

All web-based tools offer optional auto-save functionality, which can be enabled to automatically save your changes as you work. This feature helps prevent data loss if you navigate away from the page or experience connectivity issues.

Space-based tools

Space-based tools like JupyterLab and Code Editor provide access to two types of storage spaces to support both individual work and team collaboration.

Local storage (local folder)

The local storage features dedicated EBS storage that delivers superior performance for frequent file operations within your workspace. Local storage serves as your personal workspace and the files in it are private to your Space.

Within your local storage, you can create and manage subfolders to organize your files effectively. This helps you maintain a structured workspace for different aspects of your work.

When you save files to your local storage, they operate on a 'last write wins' principle—new changes overwrite previous versions without versioning capabilities.

Your local folder

- Includes this root folder and any subfolders (except shared)
- Serves as your private workspace within each project

- Allows you to work on files privately
- Is ideal for frequent file access and modification
- Is visible only in this space
- Remains isolated from other project members, creating a secure environment for experimentation and development

Shared storage (shared folder)

Shared storage is implemented in Amazon S3 or Git repository and is accessible from all Amazon SageMaker Unified Studio tools. Project members can create and manage subfolders within the shared storage to help organize artifacts effectively.

By default, all project members have read, write, update, and delete access to files within the shared storage. This central repository allows team members to access common resources, share completed work, and maintain project artifacts in a single location.

Shared storage operates on a "last write wins" principle, so you have to coordinate with team members when working on the same files to avoid overwriting each other's changes.

The shared folder (Git and non-Git):

- Contains files visible to all project members
- Functions as a collaborative workspace accessible to all project members
- Is accessible across all your tools
- Updates immediately when any member adds or modifies files
- Operates on a "last write wins" mechanism, so team members should coordinate when working on the same files
- Is not well-suited for heavy file read/write workloads due to remote Amazon S3 origin of this folder and potential additional costs associated with frequent Amazon S3 access
- If two individuals are modifying the same file in this folder at the same time that might result in losing some changes

You can copy files between these locations as needed, allowing you to optimize your workflow based on performance requirements and collaboration needs. For example, copy files from shared storage to local storage for ML tasks requiring low latency.

Managing storage resources

Storage organization

Creating logical subfolders helps organize your work, grouping related files together for easier navigation. Establishing and following consistent naming conventions for files and folders helps team members understand the purpose and content of resources.

Regularly moving completed work from local to shared storage ensures team access to important files. Removing unnecessary files periodically helps conserve storage space and maintain a clean, efficient working environment.

Working with files across multiple tools

Within Amazon SageMaker Unified Studio, a single file may be accessible via multiple tools. If you have the same file open simultaneously in different tools, changes made in one tool may overwrite changes made in another if not properly saved. You'll need to explicitly reopen files to see changes made through different tools.

When executing files from local storage, any resulting output files or artifacts automatically default to the same local storage location. For example, running a notebook that generates additional files will store these in local storage unless otherwise specified.

Transitioning between storage types

Currently, you cannot convert a project from S3 storage to Git-based storage or vice versa.
If you need to change storage types, you must create a new project with the desired storage configuration, copy your files to the new project, and update any references or dependencies.

This limitation is important to consider when initially setting up your project, as the storage decision has long-term implications for your workflow.

Limitations

- For JupyterLab and Code Editor, large files over 15 MB cannot be directly uploaded to the shared folder. To upload large files, first upload them to any other folder (such as your local storage), then copy or move to the shared folder.

- When uploading files using `putObject` API to non-existent folder paths in shared storage, folders created indirectly may display incorrect timestamps - January 1, 1970 in JupyterLab's file browser. In the CodeEditor, the file metadata also shows the time stamp as January 1, 1970.

Compute

On a project's **Compute** page in Amazon SageMaker Unified Studio, you can view compute information and add compute resources such as Amazon Redshift and Amazon EMR Serverless clusters to your project. Amazon SageMaker Unified Studio supports different kinds of compute resources:

- **Data warehouse**: This includes Amazon Redshift Serverless workgroups and Amazon Redshift provisioned clusters. Workgroups are a collection of compute resources that you can use to run data warehousing queries and engineering notebooks without managing underlying infrastructure. Clusters are scalable compute environments that enable the processing and analysis of large datasets. For more information, see the section called “Amazon Redshift”.
- **Data processing**: This includes connections to Amazon EMR on EC2 clusters, Amazon EMR on EKS virtual clusters, Amazon EMR Serverless applications, and Glue ETL computes. For more information, see the following links:
  - the section called “Amazon EMR on EC2”
  - the section called “Amazon EMR on EKS”
  - the section called “EMR Serverless”
  - the section called “Glue ETL”
- **HyperPod clusters**: In Amazon SageMaker Unified Studio, you can launch machine learning workloads on Amazon SageMaker AI HyperPod clusters. For more information, see the section called “HyperPod clusters”.
- **Spaces**: Spaces are used to manage the storage and resource needs of applications running on JupyterLab and Code Editor. You can create and manage multiple code spaces within your project. The experience differs by domain type. For more information, see *Code spaces*.
- **MLflow tracking servers**: MLflow tracking servers make it possible to use MLflow in Amazon SageMaker Unified Studio to create, manage, analyze, and compare machine learning experiments. For more information, see the section called “Track experiments using MLflow”.
- **MLflow Apps**: MLflow Apps are the latest managed MLflow offering that provides faster startup times, cross-account sharing, and integration with SageMaker AI features. For more information, see Track experiments using MLflow in Identity Center-based domains and Track experiments using MLflow in IAM-based domains.
- **Workflow environments**: Use a workflow environment to share scheduled workflows with other project members. For more information, see the section called “Create a workflow environment”.

> **Note**
>
> Adding a serverless or cluster compute connection adds the compute resource to the project space, so all project members can access it.

Amazon Athena compute connections in Amazon SageMaker Unified Studio

You can connect to existing Amazon Athena workgroups in Amazon SageMaker Unified Studio.

Amazon Athena workgroups are a collection of compute resources that you can use to run SQL queries on data stored in Amazon S3 without managing underlying infrastructure. These are especially useful for ad-hoc analytics and interactive querying of large datasets.

Gaining access to Amazon Athena resources

To add Amazon SageMaker Unified Studio connections to existing compute resources, you must get access information from the admin that owns the resources. To do this, first get your project ID from the **Project overview** page of the project you want to add resources to. Then, send the project ID to the owner of the Amazon Athena resources. The Amazon Athena admin uses the project ID to complete some steps so that you receive access details from them, and then you can input the access information in Amazon SageMaker Unified Studio.

You and the admin must complete different steps depending on whether the resources are in the same account as the account you are accessing Amazon SageMaker Unified Studio in.

Gaining access to resources in the same account

In some cases, the Amazon Athena workgroup you want to add to your Amazon SageMaker Unified Studio project might be in the same account as your project. Complete the following steps:

1. Send the project ID to the Amazon Athena admin. You can find this on the **Project overview** page of your Amazon SageMaker Unified Studio project.
2. The admin then adds 1 of the following tags to the Amazon Athena workgroup that you want to add to Amazon SageMaker Unified Studio.
   - Option 1: Add a tag to allow only a specific Amazon SageMaker Unified Studio project to access it: `AmazonDataZoneProject=`*`projectID`*.

- Option 2: Add a tag to allow all Amazon SageMaker Unified Studio projects in this account to access it: `for-use-with-all-datazone-projects=true`.

Gaining access to resources in a different account

In some cases, the Amazon Athena workgroup you want to add to your Amazon SageMaker Unified Studio project might be in a different AWS account than your project. Complete the following steps:

1. Send the following information to the Amazon Athena admin from the **Project overview** page of your Amazon SageMaker Unified Studio project:
   - The Amazon SageMaker Unified Studio project role ARN
   - The Amazon SageMaker Unified Studio project ID
   - The Amazon SageMaker Unified Studio project domain ID
2. The admin must create an access role for Amazon SageMaker Unified Studio that can be used to query Amazon Athena. The role should have the following permissions:
   - AmazonAthenaFullAccess
   - SQL Workbench permissions

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SQLWorkBenchActionsWithoutResourceType",
            "Effect": "Allow",
            "Action": [
                "sqlworkbench:PutTab",
                "sqlworkbench:DeleteTab",
                "sqlworkbench:DriverExecute",
                "sqlworkbench:GetUserInfo",
                "sqlworkbench:ListTabs",
                "sqlworkbench:GetAutocompletionMetadata",
                "sqlworkbench:GetAutocompletionResource",
                "sqlworkbench:PassAccountSettings",
                "sqlworkbench:ListQueryExecutionHistory",
                "sqlworkbench:GetQueryExecutionHistory",
                "sqlworkbench:CreateConnection",
                "sqlworkbench:PutQCustomContext",
```

```
                "sqlworkbench:GetQCustomContext",
                "sqlworkbench:DeleteQCustomContext",
                "sqlworkbench:GetQSqlRecommendations",
                "sqlworkbench:GetQSqlPromptQuotas",
                "sqlworkbench:GetSchemaInference"
            ],
            "Resource": "*"
        }
    ]
}
```

- [Optional] Amazon S3 permissions when using specific Amazon Athena workgroup output directory bucket:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AthenaBucketOut",
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:Put*",
                "s3:List*"
            ],
            "Resource": "arn:aws:s3:::your-bucket-name/athena/*"
        }
    ]
}
```

The trust policy is as follows:

```
# trust policy of access role
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
```

```
                "AWS": "project-role-arn"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": "project-id"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "project-role-arn"
            },
            "Action": [
                "sts:SetSourceIdentity"
            ],
            "Condition": {
                "StringLike": {
                    "sts:SourceIdentity": "${aws:PrincipalTag/datazone:userId}"
                }
            }
        },
       {
            "Effect": "Allow",
            "Principal": {
                "AWS": "project-role-arn"
            },
            "Action": "sts:TagSession",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/AmazonDataZoneProject": "project-id",
                    "aws:RequestTag/AmazonDataZoneDomain": "domain-id"
                }
            }
        }
    ]
}
```

3. The admin then sends you the Access role ARN.

You can then use the access credentials to add the compute connection in Amazon SageMaker Unified Studio. For more information, see the section called “Connecting to an existing Amazon Athena resource”.

Connecting to an existing Amazon Athena resource

After you have gained access to an Amazon Athena workgroup, you can add a connection to the compute resource in the Amazon SageMaker Unified Studio console. Complete the following steps to add a Amazon Athena workgroup to the project space:

1. Navigate to the **Compute** section of your project in Amazon SageMaker Unified Studio.
2. Select the **SQL analytics** tab.
3. Choose **Add compute**.
4. Choose **Connect to existing compute resources**, then choose **Next**.
5. Select **Amazon Athena workgroup**, then choose **Next**.
6. Under **Connection properties**, provide the Amazon Athena workgroup ARN you want to add. If the compute resource is in the same account as your Amazon SageMaker Unified Studio project, you can select the compute resource from a dropdown menu. For more information, see the section called “Gaining access to Amazon Athena resources”.
7. If you have access role, provide access role credentials. For project role, you don't have to provide any credentials.
8. Under **Name**, enter the name of the Amazon Athena workgroup you want to add.
9. Under **Description**, provide a description of the compute resource.
10. Choose **Add compute**. The Amazon SageMaker Unified Studio project Compute and Data pages then display information for that resource.

Amazon Redshift compute connections in Amazon SageMaker Unified Studio

You can connect to Amazon Redshift Serverless workgroups and Amazon Redshift clusters in Amazon SageMaker Unified Studio.

Amazon Redshift Serverless workgroups are a collection of compute resources that you can use to run data warehousing queries and engineering notebooks without managing underlying

infrastructure. These are especially useful in environments where query patterns are unpredictable or workloads fluctuate.

Amazon Redshift clusters are scalable compute environments that enable the processing and analysis of large datasets. They are optimized for running SQL-based queries on data warehouses, making them ideal for structured data analytics and reporting.

Gaining access to Amazon Redshift resources

To add Amazon SageMaker Unified Studio connections to existing compute resources, you must get access information from the admin that owns the resources. To do this, first get your project ID from the **Project overview** page of the project you want to add resources to. Then, send the project ID to the owner of the Amazon Redshift resources. The Amazon Redshift admin uses the project ID to complete some steps so that you receive access details from them, and then you can input the access information in Amazon SageMaker Unified Studio.

You and the admin must complete different steps depending on whether the resources are in the same account as the account you are accessing Amazon SageMaker Unified Studio in.

> **Note**
>
> If you want to query the Amazon Redshift resources using JuypterLab within Amazon SageMaker Unified Studio, the Amazon Redshift resource must use the same VPC as the Amazon SageMaker Unified Studio project. If the Amazon SageMaker Unified Studio project uses a different VPC than the Amazon Redshift resource you want to gain access to, you and your admin must complete additional steps to connect the VPCs before you can use JupyterLab to query. You can still query using the Data page of your project if you are using different VPCs. For more information, see VPC to VPC connectivity and Connect VPCs using VPC peering.

Gaining access to resources in the same account

In some cases, the Amazon Redshift resource you want to add to your Amazon SageMaker Unified Studio project might be in the same account as your project.

For compute resources in the same account as your Amazon SageMaker Unified Studio project, complete the following steps:

1. Send the Amazon Redshift admin the project ID. This can be found on the **Project overview** page of your Amazon SageMaker Unified Studio project.
2. The admin then adds 1 of the following tags to the Amazon Redshift cluster or workgroup and its namespace that you want to add to Amazon SageMaker Unified Studio.

   - Option 1: Add a tag to allow only a specific Amazon SageMaker Unified Studio project to access it: `AmazonDataZoneProject=`*`projectID`*.
   - Option 2: Add a tag to allow all Amazon SageMaker Unified Studio projects in this account to access it: `for-use-with-all-datazone-projects=true`.
3. The admin then must send you a username and password for a database user that has access to the compute resources.

You can then use the username and password to add the compute connection in Amazon SageMaker Unified Studio. For more information, see the section called “Connecting to an existing Amazon Redshift resource”.

Gaining access to resources in a different account

In some cases, the Amazon Redshift resource you want to add to your Amazon SageMaker Unified Studio project might be in a different AWS account than your project.

For compute resources in a different account, complete the following steps:

1. Send the Amazon Redshift admin the following information from the **Project overview** page of your Amazon SageMaker Unified Studio project:

   - The Amazon SageMaker Unified Studio project role ARN.
   - The Amazon SageMaker Unified Studio project ID.
   - The Amazon SageMaker Unified Studio project domain ID.
2. The admin must create an access role for Amazon SageMaker Unified Studio that can be used to query Amazon Redshift.

   An example Amazon Redshift access role for Amazon SageMaker Unified Studio is provided below:

```
# Sample permission policy of access role to query Redshift
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RedshiftQueryEditorConnectPermissions",
            "Effect": "Allow",
            "Action": [
                "redshift:GetClusterCredentialsWithIAM",
                "redshift:GetClusterCredentials",
                "redshift:DescribeClusters",
                "redshift:CreateClusterUser"
            ],
            "Resource": [
                "arn:aws:redshift:*:012345678912:cluster:*",
                "arn:aws:redshift:*:012345678912:dbname:*/*",
                "arn:aws:redshift:*:012345678912:dbuser:*/*"
            ]
        },
        {
            "Sid": "RedshiftServerlessQueryEditorConnectPermissions",
            "Effect": "Allow",
            "Action": [
                "redshift-serverless:GetCredentials",
                "redshift-serverless:GetWorkgroup",
                "redshift-serverless:ListTagsForResource"
            ],
            "Resource": [
                "arn:aws:redshift-serverless:*:012345678912:workgroup/*"
            ]
        },
        {
            "Sid": "SecretsManagerAccess",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret"
            ],
            "Resource": [
                "secret_arn"
            ]
        },
```

```
        {
            "Sid": "sqlworkbench",
            "Effect": "Allow",
            "Action": [
                "sqlworkbench:*"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

The trust policy is as follows:

```
# trust policy of access role
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "project-role-arn"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": "project-id"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "project-role-arn"
            },
            "Action": [
                "sts:SetSourceIdentity"
            ],
            "Condition": {
                "StringLike": {
```

```
                    "sts:SourceIdentity": "${aws:PrincipalTag/datazone:userId}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "project-role-arn"
            },
            "Action": "sts:TagSession",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/AmazonDataZoneProject": "project-id",
                    "aws:RequestTag/AmazonDataZoneDomain": "domain-id"
                }
            }
        }
    ]
}
```

3. (Optional) If you want to use IAM credentials to access the Amazon Redshift resource, rather than an AWS Secrets Manager secret, the admin must add the following tag to the access role:

```
RedshiftDbUser=Username
```

4. The admin then needs to provide JDBC connection info in one of two ways:

   - Use a Secrets Manager secret in the same account as the Redshift resource. The access role should have permission to read the secret value. For more information about the JSON format that should be used in the secret, see JSON structure of a secret in the AWS Secrets Manager User Guide.
   - Use a temporary username and password. This is generated from the IAM access role credentials.
     - The `RedshiftDbUser` tag on the access role is required. This determines the federated database user within the databases for the Amazon SageMaker Unified Studio users. For more information, see Setting up principal tags to connect a cluster or workgroup from query editor v2 in the Amazon Redshift Management Guide.

5. The admin then sends you the following information:

   - Access role ARN.

- JDBC URL. For example: *jdbc:redshift://default-workgroup.012345678912.us-west-2.redshift-serverless.amazonaws.com. For more information about JDBC connections, see Connecting to Amazon Redshift Serverless through JDBC drivers and Getting the JDBC URL in the in the Amazon Redshift Management Guide.*.
- *(Optional) AWS Secrets Manager secret ARN.* For example: *arn:aws:secretsmanager:us-west-2:012345678912:secret:shared-rs-cluster-password-Ab1CDe.*

You can then use the access credentials and JDBC URL to add the compute connection in Amazon SageMaker Unified Studio. For more information, see the section called “Connecting to an existing Amazon Redshift resource”.

Connecting to an existing Amazon Redshift resource

After you have gained access to an Amazon Redshift resource, you can add a connection to the compute resource in the Amazon SageMaker Unified Studio console. Complete the following steps to add a serverless or cluster compute to the project space:

1. Go to the **Compute** section of your project in Amazon SageMaker Unified Studio.
2. Select the **Data warehouse** tab.
3. Choose **Add compute**.
4. Choose **Connect to existing compute resources**, then choose **Next**.
5. Select the type of compute resource you want to add, then choose **Next**.
6. Under **Connection properties**, provide the JDBC URL or the compute you want to add. If the compute resource is in the same account as your Amazon SageMaker Unified Studio project, you can select the compute resource from a dropdown menu. For more information, see the section called “Gaining access to Amazon Redshift resources”.
7. Under **Authentication**, provide the credential type you want to use to access the resource. The credential type must be one of the following options: Username and password, IAM credentials, AWS Secrets Manager.
8. Provide the credentials according to the authentication method you selected.
9. Under **Name**, input the name of the Amazon Redshift Serverless or Amazon Redshift Cluster you want to add.
10. Under **Description**, provide a description of the compute resource.
11. Choose **Add compute**. The Amazon SageMaker Unified Studio project Compute and Data pages then display information for that resource.

> **Note**
>
> Some credentials provide more information than others on the Compute page. Using a username and password enables Amazon SageMaker Unified Studio to display more information for a resource.

Creating a new Amazon Redshift Serverless compute resource

You can create a new compute resource and add a connection to it in Amazon SageMaker Unified Studio. Complete the following steps to add a new Amazon Redshift Serverless compute connection to the project space:

1. Go to the **Compute** section of your project in the Amazon SageMaker Unified Studio .
2. On the **Data warehouse** tab, choose **Add compute**.
3. Choose **Create new compute resources**.
4. Select the type of compute resource you want to add.
5. Under **Compute name**, input a name for the Amazon Redshift Serverless resource you want to add.
6. Under **Description**, provide a description of the compute resource.
7. Set the base capacity, maximum capacity and database name.
8. Choose **Add compute**. The Amazon SageMaker Unified Studio project Compute and Data pages then display information for that resource.

> **Note**
>
> Some credentials provide more information than others on the Compute page. Using a username and password enables Amazon SageMaker Unified Studio to display more information for a resource.

Removing an Amazon Redshift compute connection

When you remove a compute connection in Amazon SageMaker Unified Studio, you delete the connection to the compute resource that your Amazon SageMaker Unified Studio project has without deleting the compute resource.

To remove a compute connection in Amazon SageMaker Unified Studio, complete the following steps:

1. Go to the **Compute** page of your project in the Amazon SageMaker Unified Studio.
2. Select the name of the compute connection you want to remove. You are then taken to the compute details page.
3. Choose **Actions** > **Remove compute**. A popup window appears asking you to confirm the removal.
4. To confirm the removal, input confirm in the text box provided.
5. Choose **Remove compute**.

This removes the Amazon SageMaker Unified Studio connection to the compute resource. You are then no longer able to access the compute resource in the Amazon SageMaker Unified Studio project, but the compute resource is not deleted.

Amazon EMR on EC2 connections in Amazon SageMaker Unified Studio

Whenever you are working with a project, you can manage that project's Amazon EC2 resources and view both monitoring and logging data for those resources. You can create and configure Amazon EMR on EC2 clusters, as well as terminate and remove those clusters. When clusters are running, data regarding their metrics is automatically sent to CloudWatch, while logging data is preserved in the Spark UI.

Adding a new Amazon EMR on EC2 cluster in Amazon SageMaker Unified Studio

As a data worker, you can make use of Amazon EMR on EC2 by adding existing or new Amazon EMR on EC2 clusters as compute instances to a project in the Amazon SageMaker Unified Studio Studio. Within a project, you can use both existing and new Amazon EMR on EC2 clusters.

Before you can create a new Amazon EMR on EC2 cluster, your admin must enable blueprints. On-demand creation isn't supported for Amazon EMR on EC2 in quick setup.

After your Admin has enabled blueprints:

1. In the left navigation pane, choose **Compute**.
2. In the Compute panel, select the **Data processing** tab.
3. To create a new Amazon EMR on EC2 cluster, select the **Add compute** dropdown menu and then choose **New compute**.
4. In the **Add compute** modal, you can select the type of compute you would like to add to your project. Select **Create new compute resources**.
5. Select **Amazon EMR on EC2 cluster**.
6. The **Add compute** dialog box allows you to specify the name of the Amazon EMR on EC2 cluster, provide a description, and choose a release of EMR (such as EMR 7.5) that you want to install on your cluster.
7. After configuring these settings, select **Add compute**. After some time, your Amazon EMR on EC2 cluster will be added to your project.

Adding an existing Amazon EMR on EC2 cluster in Amazon SageMaker Unified Studio

As a data worker, you can make use of Amazon EMR on EC2 by adding existing or new Amazon EMR on EC2 clusters as compute instances to a project in the Amazon SageMaker Unified Studio Studio. Within a project, you can use both existing and new Amazon EMR on EC2 clusters.

Before you can connect to an Amazon EMR on EC2 cluster, you must complete the following prerequisites:

- Your Amazon SageMaker Unified Studio admin must enable blueprints. On-demand creation isn't supported for Amazon EMR on EC2 in quick setup. In addition, if you are connecting to an Amazon EMR on EC2 cluster that is not runtime-role enabled, the admin must configure specific blueprints as described in the section below.
- You must have a project created in Amazon SageMaker Unified Studio. If you are connecting to an Amazon EMR on EC2 cluster that is not runtime-role enabled, you must create a project that includes specific blueprint configurations in the project profile.
- The admin that owns the Amazon EMR resource you want to connect to must complete a set of prerequisite steps to grant you access to the resource.

More details on each of these steps is found in the sections below.

Prerequisite steps for you and your Amazon SageMaker Unified Studio admin

Amazon EMR on EC2 clusters can be runtime-role enabled or not runtime-role enabled. You can connect to both kinds of Amazon EMR on EC2 clusters in Amazon SageMaker Unified Studio. However, to use clusters that are not runtime-role enabled, you and your Amazon SageMaker Unified Studio admin must prepare to use a project with specific configurations.

> **Note**
>
> If you are connecting to clusters that are runtime-role enabled, you can proceed to the section for prerequisite steps for Amazon EMR admins without completing the steps in this section.

- You can use runtime-role enabled clusters to specify different IAM roles for individual jobs or steps within a cluster, with fine-grained access control tailored to specific job needs.
- Clusters that are not runtime-role enabled have limited granular access control for jobs. Instead, all jobs on the cluster use the same set of permissions.

Amazon EMR clusters with runtime roles enabled are considered more secure because they allow for fine-grained access control at the job level, meaning each individual job running on the cluster can be assigned a specific IAM role with only the necessary permissions to access the data and resources it needs.

To prepare to use clusters that are not runtime-role enabled, complete the following additional steps:

> **Note**
>
> Amazon EMR clusters that are not runtime-role enabled must have in-transit encryption enabled in order to be connected to Amazon SageMaker Unified Studio. To ensure that the Amazon EMR cluster meets this requirement, verify with your Amazon EMR admin that the cluster has a security configuration with in-transit encryption enabled. For more information, see Create a security configuration with the Amazon EMR console or with the AWS CLI in the Amazon EMR Management Guide.

1. The Amazon SageMaker Unified Studio admin must configure the tooling configurations in the blueprints for a project profile so that **allowConnectionToUserGovernedEmrClusters** is set to **True** in the Amazon SageMaker Unified Studio management console. For more information, see the Amazon SageMaker Unified Studio Administrator Guide.
2. You create a project using the project profile that your admin modified in step 1.

For more information about runtime roles, see Runtime roles for Amazon EMR steps in the Amazon EMR Management Guide.

> **Note**
>
> For clusters without runtime roles, Amazon SageMaker Unified Studio cannot provide governance on the clusters, and applications running on these clusters will not be isolated between projects or honor fine-grained access control based on project data permissions. Additionally, all project resources are inaccessible to the cluster unless additional permissions are granted to the IAM instance profile role attached to the Amazon EC2 instance.

Prerequisite steps for Amazon EMR admins

Before you can add an existing Amazon EMR on EC2 resource to your project in Amazon SageMaker Unified Studio, the admin that owns that resource must grant access to you by completing the following steps:

Create an Amazon EMR access role with a trust policy

1. Get the project role ARN and project ID for the Amazon SageMaker Unified Studio project that you want to grant access to. Project members can get the project role ARN and project ID from the **Project overview** page in their project.

   > **Note**
   >
   > If the Amazon SageMaker Unified Studio project uses a different VPC than the Amazon EMR on EC2 cluster you want to grant access to, you must also get the project VPC information from the project member and complete additional steps to connect the

VPCs. For more information, see VPC to VPC connectivity and Connect VPCs using VPC peering.

2. Make sure that the EMR cluster you want to grant access to has an instance profile role with the `sts:AssumeRole` permission on the runtime role. For more information, see Runtime roles for Amazon EMR steps in the Amazon EMR Management Guide.
3. Go to the AWS IAM console.
4. On the Roles page, choose **Create role**.
5. Choose **Custom trust policy**.
6. Enter information for the trust policy as shown in the example below, and edit it according to the project information you received in step 1.

   - Change `project-role-arn` to be the project role ARN you received from the Amazon SageMaker Unified Studio project member.
   - Change `project-id` to be the project ID you received from the Amazon SageMaker Unified Studio project member.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "project-role-arn"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": "project-id"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "project-role-arn"
            },
            "Action": [
```

```
                "sts:SetSourceIdentity"
            ],
            "Condition": {
                "StringLike": {
                    "sts:SourceIdentity": "${aws:PrincipalTag/datazone:userId}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "project-role-arn"
            },
            "Action": "sts:TagSession",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/AmazonDataZoneProject": "project-id",
                    "aws:RequestTag/AmazonDataZoneDomain": "domain-id"
                }
            }
        }
    ]
}
```

7. Choose **Next**.
8. Under **Role name**, enter a name for the role.
9. (Optional) Enter a description for the role.
10. Choose **Create role**.

Attach permissions to the role

1. Select the role you have created in the AWS IAM console.
2. Choose **Add permissions** > **Create inline policy**.
3. Enter information as shown in the example below, and edit it according to the information for your Amazon EMR clusters that you want to grant access to.

   - Change the EMR cluster ARN to be the ARN for the cluster. You can find this on the cluster details page in the Amazon EMR console by selecting the cluster ID of the cluster that you want to share.

> **Note**
>
> You can use an asterisk instead of the Amazon EMR cluster ID if you want to grant access to all clusters instead of just one.

- Change the certificate path to the one defined in the Amazon EMR security configuration for that cluster in the Amazon EMR console. For more information, see Specify a security configuration for an Amazon EMR cluster in the Amazon EMR Management Guide.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EmrAccess",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:ListInstances",
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:ListBootstrapActions",
                "elasticmapreduce:GetClusterSessionCredentials" # Skip this for
 non-runtime role clusters
            ],
            "Resource": "arn:aws:elasticmapreduce:us-east-1:666777888999:cluster/j-
AB1CDEFGHIJK" # EMR cluster ARN
        },
        {
            "Sid": "EMRSelfSignedCertAccess",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::666777888999-us-east-1-sam-dev/my-certs.zip" # Cert
 path defined in the EMR security configuration
            ]
        },
        {
            "Sid": "EMRSecurityConfigurationAccess",
            "Effect": "Allow",
```

```
            "Action": [
                "elasticmapreduce:DescribeSecurityConfiguration"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

4. Choose **Next**.
5. Under **Policy name**, enter a name for the polciy.
6. Choose **Create policy**. You can then see the permissions policy listed on the page for the role you created in the IAM console.

Send information to project members

1. Copy the ARN of the EMR access role you created in the IAM console and send it to the Amazon SageMaker Unified Studio project member you want to grant access to.
2. Copy the Amazon EMR cluster ARN that you added to the permissions policy and send it to the Amazon SageMaker Unified Studio project member you want to grant access to.
3. From the Amazon EMR on EC2 cluster details page in the Amazon EMR console, copy the EC2 instance profile string and search for it on the Roles page in the IAM console to find the role that contains the Amazon EC2 instance profile ARN.
4. Select the name of the role that contains the instance profile ARN to open the role details page, then copy the ARN and send it to the Amazon SageMaker Unified Studio project member you want to grant access to.

After the Amazon EMR admin has completed these steps, project members are able to add a connection to the Amazon EMR on EC2 cluster as a compute resource in Amazon SageMaker Unified Studio.

Adding the Amazon EMR on EC2 compute resource

1. In the left navigation pane, choose **Compute**.
2. On the Compute page, select the **Data processing** tab.
3. Choose **Add compute**, then choose **Connect to existing compute resources**.

4. In the **Add compute** modal, you can select the type of compute resource you would like to add to your project. Select **EMR on EC2 cluster**.
5. To add a connection to an existing Amazon EMR on EC2 cluster, you must have the correct permissions to access the Amazon EMR on EC2 cluster. You can select the **Copy project information** button to copy the data that the Amazon EMR admin will need to grant the data worker access. If you haven't already, send the project role ARN and the project ID to your admin.

   > **Note**
   >
   > The Amazon EMR admin will also need the project ID, which is the penultimate string in the project ARN. To view and copy the project ID, go to the **Project overview** page of your project.

6. After the account administrator has granted you access according to the prerequisite steps above, you can specify the ARNs associated with the cluster. You must fill in the **Access role ARN**, **EMR on EC2 cluster ARN**, **Compute name**, and the **Instance profile role ARN**.
7. Choose **Add compute**. Your Amazon EMR on EC2 instance is then added to your project.

After you have added a cluster to a project, you are able to see the cluster in the list on the **Data processing** tab in the Compute panel. You can then view the cluster details by selecting the cluster you want.

Using an Amazon EMR on EC2 cluster

After connecting to an Amazon EMR on EC2 cluster, you can begin using the cluster. To get started, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project that contains the compute connection. You can do this by choosing a project from the project selector dropdown at the top of the page.
3. On the **Compute** page, choose the name of the compute you want to initialize. This takes you to a page with details about the cluster. Make a note of the name of the compute.
4. Choose **Actions > Open JupyterLab IDE**.

5. In the first cell, choose a connection type that you want to use from the dropdown list of connection types. Then choose the name of the compute from the dropdown list of compute options.
6. Choose the **Run** icon.

Your cluster is now initialized and configured to be a compute resource in your Amazon SageMaker Unified Studio project.

Monitoring Amazon EMR on EC2 clusters in Amazon SageMaker Unified Studio

You can monitor the performance of your Amazon EMR on EC2 clusters to ensure optimal resource use and efficient job execution. Information on metrics is automatically collected and sent to Amazon CloudWatch during operation of an Amazon EMR cluster.

You can see CloudWatch metrics for a specific cluster by selecting the cluster you're interested in from the list of clusters under the Cluster tab. Selecting a cluster will bring you to the Detail view for that cluster. After you've selected a cluster, select the **Monitoring** tab.

You will be able to see a grid view of the CloudWatch Metrics for the cluster you selected.

You can see information presented through different views by using the **Dashboard View** drop-down menu: Cluster Overview, Primary Node Group, Core Node Group, Task Node Group. You can also adjust the time range.

Configuring trusted identity propagation

You or your admin add an inline policy to the instance profile role to enable trusted identity propagation for that cluster in Amazon SageMaker Unified Studio. Before doing this, make sure you have followed the steps to add a new EMR on EC2 cluster to your project.

> **Note**
>
> Trusted identity propagation is supported for EMR on EC2 clusters that you create using Amazon SageMaker Unified Studio.

To find the name of the instance profile role for an EMR on EC2 cluster, complete the following steps:

1. Navigate to the project that contains the compute connection. You can do this by choosing a project from the project selector dropdown at the top of the page.
2. On the **Compute** page, go to the **Data processing** tab.
3. Choose the name of the compute you want to configure TIP for. This takes you to a page with details about the cluster. The instance profile role is on this page and the admin can then search for it in the IAM console.

As an admin user who could edit IAM policies in the account that owns the project, add the following inline policy to the instance profile role.

```
{
    "Statement": [
        {
            "Sid": "IdCPermissions",
            "Effect": "Allow",
            "Action": [
                "sso-oauth:CreateTokenWithIAM",
                "sso-oauth:IntrospectTokenWithIAM",
                "sso-oauth:RevokeTokenWithIAM"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowAssumeRole",
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Resource": [
                "instance-profile-role-ARN"
            ]
        }
    ]
}
```

After updating the role's policy, you can use the EMR on EC2 connection to initiate interactive Spark sessions.

Configuring user background sessions for Amazon EMR on EC2

Warning

When user background sessions is enabled for Amazon EMR on EC2, Amazon SageMaker Unified Studio will not terminate interactive sessions. All interactive sessions will be only terminated once all queries are completed.

Amazon EMR on EC2 requires additional IAM permissions to enable user background sessions. You must attach the following inline IAM role policy to the IAM role created as the project user role.

Note

The project user role for an Amazon SageMaker Unified Studio project is named `datazone_usr_role_{project_id}`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "UserBackgroundSessions",
            "Effect": "Allow",
            "Action": [
                "sso:GetApplicationSessionConfiguration"
            ],
            "Resource": "*"
        }
    ]
}
```

For more information, see User background sessions in the Amazon EMR on EC2 management guide.

Terminating and removing an Amazon EMR on EC2 cluster

> **Warning**
>
> A terminated EMR Cluster is irrecoverable. Ensure that the resource and any data on HDFS or jupyter notebooks is no longer required prior to removal.

When you no longer need an Amazon EMR on EC2 cluster, the cluster can be terminated and removed.

To remove a cluster:

1. Login to the Amazon SageMaker Unified Studio and navigate to the **Data processing** tab of the Compute section. Select the name of the compute instance you would like to remove.
2. On the compute details page, select the **Terminate and remove** option.
3. A dialog box will appear asking you to confirm that you want to terminate and remove the instance of compute, which in this case is your Amazon EMR on EC2 cluster. Confirm that you want to remove the compute, by typing "confirm" in the text box.
4. Choose **Terminate and remove compute** to begin termination and removal.
5. After a few minutes, your cluster should have been removed.

Spark History Server

You can use the live Spark UI in a notebook session to view details such as tasks, executors and logs about Spark jobs.

You can explore the Spark History Server for a cluster at any time. To do this, select your cluster from the list of all clusters assigned to a project, which brings up the Detail view for the cluster. On the Detail page view, select the **Applications** tab and choose the '**Spark History Server** link.

Get started adding on-demand Amazon EMR on EC2 instances

Overview

Amazon EMR on EC2 is a managed big data platform that simplifies running distributed data processing frameworks like Apache Spark, Hadoop, and Hive on Amazon EC2 instances. Amazon

EMR handles the complexities of cluster provisioning, configuration, and scaling, allowing you to focus on your data processing tasks. For more details on Amazon EMR, visit the Amazon EMR webpage.

The Amazon EMR on EC2 integration with Amazon SageMaker Unified Studio streamlines your data analytics workflow, giving you a unified data and compute experience. This integration lets you easily access and create Amazon EMR clusters alongside other data tools in a single interface. You can organize Amazon EMR resources within Amazon SageMaker Unified Studio projects, connect Amazon EMR workloads with your data catalog, and provision clusters on-demand. With this integration, you can experiment by creating and terminating Amazon EMR clusters as needed, optimizing costs while maintaining a cohesive data experience.

With the help of this getting started guide you will be able to configure Amazon EMR cluster settings for EC2 deployment and launch Amazon EMR clusters.

Prerequisites

You must complete the following procedure through the AWS management console to create an Amazon EMR on EC2 in an Amazon SageMaker Unified Studio project.

Set up Amazon SageMaker Unified Studio

Before you get started with creating an Amazon EMR on EC2, you must access Amazon SageMaker Unified Studio and create a project with the **All capabilities** project profile.

1. If you haven't created an Amazon SageMaker Unified Studio domain, follow the steps in Create a Amazon SageMaker Unified Studio domain - quick setup .
2. To access Amazon SageMaker Unified Studio:
    a. Open the Amazon SageMaker Unified Studio console at https://console.aws.amazon.com/sagemaker/.
    b. Choose **Studio**.
    c. Choose **Open Studio**.
    d. Sign in using your SSO or AWS credentials. For more information, see *Access Amazon SageMaker Unified Studio*.
3. Create a project with the **All capabilities** profile:
    a. From the project selector dropdown at the top of the page, choose **Manage projects**.
    b. Choose **Create project**.

c. Select the **All capabilities** project profile.

d. Follow the prompts to complete project creation.

e. This profile grants you access to Amazon EMR resources. For more information, see *Create a project and add members*.

PEM certificate configuration

1. Create a PEM certificate, which saves your ZIP file on your local machine:

   a. Open your terminal on your local machine.

   b. The following commands demonstrate how to use OpenSSL to generate a self-signed X.509 certificate with a 2048-bit RSA private key. Consider changing `us-west-2` to the region you are using throughout this tutorial. Other optional subject items such as country (C), state (S), and Locale (L), are specified.

   > **Important**
   >
   > This example is a proof-of-concept demonstration only. Using self-signed certificates is not recommended and presents a potential security risk. For production systems, use a trusted certification authority (CA) to issue certificates. For more information see Providing certificates for encrypting data in transit with Amazon EMR encryption.

```
$ openssl req -x509 -newkey rsa:2048 -keyout privateKey.pem -out
 certificateChain.pem -days 365 -nodes -subj '/C=US/ST=Washington/L=Seattle/
O=MyOrg/OU=MyDept/CN=*.us-west-2.compute.internal'
$ cp certificateChain.pem trustedCertificates.pem
$ zip -r -X my-certs.zip certificateChain.pem privateKey.pem
 trustedCertificates.pem
```

2. Upload the PEM certificate ZIP file to an Amazon S3 bucket:

   a. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.

   b. Under **General purpose buckets**, choose your amazon-sagemaker bucket.

c. Navigate to your domain folder. For multiple domains, locate the folder matching your Domain ID. You can find your Domain ID in the **project details** tab of Amazon SageMaker Unified Studio.

d. Choose **Create folder** and enter **`certificate_location`** as the folder name. You do not need to specify an encryption key during folder creation.

> **Note**
>
> The name **`certificate_location`** is required for this folder and cannot be customized.

e. Select your new folder to open it.

f. Under **Objects**, select **Upload** and **Add files**. Select your PEM certificate ZIP file (named "my-certs.zip") from your local machine, then choose **Upload**.

g. Select the uploaded ZIP file and choose **Copy S3 URI**. You'll need this location value in step 3.

3. Specify your certificate location in Amazon SageMaker Unified Studio, following the instructions in Specify PEM certificate for EmrOnEc2 blueprint.

Creating your Amazon EMR cluster

1. From the project selector dropdown at the top of the page, choose your project.
2. In the left navigation pane, choose **Compute**.
3. In the **Compute** panel, select the **Data processing** tab.
4. To create a new Amazon EMR on EC2 cluster choose **Add compute**.
5. In the **Add compute** modal, you can select the type of compute you would like to add to your project. Select **Create new compute resources**.
6. Select **Amazon EMR on EC2 cluster** and choose **Next**.
7. The **Add compute** dialog box allows you to specify the name of the Amazon EMR on EC2 cluster. Default settings for the Amazon EMR are fine. Choose your EMR configuration according to your choice from the prerequisites.
8. After configuring any settings if you choose, select **Add compute**. After some time, your Amazon EMR on EC2 cluster will be added to your project.

Amazon EMR on EKS in Amazon SageMaker Unified Studio

You can connect to Amazon EMR on EKS in Amazon SageMaker Unified Studio.

Amazon EMR on EKS allows you to run open-source big data frameworks on Amazon EKS. With Amazon EMR on EKS, you can focus on running analytics workloads while Amazon EMR on EKS builds, configures, and manages containers for open-source applications.

Amazon EMR on EKS virtual clusters require an Amazon EKS cluster with compatible configurations. Amazon EMR on EKS operates by creating an Amazon EMR on EKS virtual cluster on top of your existing Amazon EKS cluster. You then interact with the Amazon EMR on EKS virtual cluster directly for interactive session management. For more information, see What is Amazon EMR on EKS?

Spark History Server for Amazon EMR on EKS in Amazon SageMaker Unified Studio

You can use the Spark History Server in a notebook session to view details such as tasks, executors and logs about Spark queries.

You can explore the Spark History Server for an active Amazon EMR on EKS interactive session. To do this, navigate to your project's JupyterLab IDE and select your Amazon EMR on EKS connection. After any Spark query is executed, choose the **Spark History Server** embedded link.

Adding a new Amazon EMR on EKS virtual cluster in Amazon SageMaker Unified Studio

As a data worker, you can make use of Amazon EMR on EKS by adding new Amazon EMR on EKS virtual clusters as compute instances to a Amazon SageMaker Unified Studio project. However, in order to create new Amazon EMR on EKS virtual clusters, your admin must enable and configure blueprints.

After your admin has enabled and configured blueprints:

1. In the left navigation pane, choose **Compute**.
2. In the Compute panel, select the **Data processing** tab.
3. To create a new Amazon EMR on EKS virtual cluster, select the **Add compute** dropdown menu and then choose **New compute**.

4. In the **Add compute** modal, you can select the type of compute you would like to add to your project. Select **Create new compute resources**.
5. Select **Amazon EMR on EKS virtual cluster**.
6. The **Add compute** dialog box allows you to select your admin created Amazon EKS cluster configuration, specify the name of the Amazon EMR on EKS virtual cluster, provide a description, and choose a release of Amazon EMR (such as EMR 7.11.0-latest) that you want to install on your managed endpoint.
7. After configuring these settings, select **Add compute**. After some time, your Amazon EMR on EKS virtual cluster will be added to your project.

Using an Amazon EMR on EKS virtual cluster in Amazon SageMaker Unified Studio

After creating your Amazon EMR on EKS virtual cluster, you can begin using your compute.

> **Note**
>
> Amazon EMR on EKS in Amazon SageMaker Unified Studio is only available for SageMaker distributions >=2.10 and >=3.5.

1. In the left navigation pane, choose **Compute**.
2. In the Compute panel, select the **Data processing** tab.
3. In the data processing panel, select your target Amazon EMR on EKS virtual cluster.
4. In the compute details panel, select **Actions** and **Open JupyterLab IDE**.
5. In the JupyterLab IDE, select a compatible **Connection type** and select the name of the **Compute**.

Configuration for additional functionality in Amazon SageMaker Unified Studio

Some native Amazon EMR on EKS functionality requires additional configuration by your administrator for your Amazon SageMaker Unified Studio projects. Contact your administrator to review documentation for additional functionality.

- Configuring monitoring with Spark History Server for Amazon EMR on EKS

- Configuring fine-grained access controls for Amazon EMR on EKS
- Configuring trusted identity propagation for Amazon EMR on EKS
- Configuring user background sessions for Amazon EMR on EKS

Removing an Amazon EMR on EKS virtual cluster in Amazon SageMaker Unified Studio

When you no longer need an Amazon EMR on EKS virtual cluster, the Amazon EMR on EKS resources can be deleted.

> **Note**
>
> The Amazon EKS cluster used to create Amazon EMR on EKS resources is never deleted by SageMaker.

1. In the left navigation pane, choose **Compute**.
2. In the Compute panel, select the **Data processing** tab.
3. In the data processing panel, select your target Amazon EMR on EKS virtual cluster.
4. In the compute details panel, select **Actions** and **Remove compute**.
5. In the confirmation modal, select **Remove compute**.
6. After a short time, your Amazon EMR on EKS virtual cluster will be removed.

EMR Serverless compute connections in Amazon SageMaker Unified Studio

In addition to Amazon EMR on EC2 clusters, you can also create and delete EMR Serverless applications.

Adding a new EMR Serverless application

As a data worker, you can make use of EMR Serverless applications by adding them to a project in the Amazon SageMaker Unified Studio Studio. Within a project, you can use both existing and new applications. You can use existing applications at any time. However, in order to create a new EMR Serverless application, the admin must enable blueprints.

After your admin has enabled blueprints:

1. In the left navigation pane, choose **Compute**.
2. In the Compute panel, select the **Data processing** tab.
3. To add an instance of an Amazon EMR Serverless, select the **Add compute** dropdown menu and then choose **New compute**.
4. In the **Add compute** modal, you can select the type of compute you would like to add to your project. Select **EMR Serverless**.
5. The **Add compute** dialog box allows you to specify the name of the EMR Serverless application, provide a description, and choose a release of EMR Serverless that you want your application to use.
6. Choose the permission mode option that supports the data you will be using with the compute resource.
   - Select **project.spark.fineGrained** for data managed using fine-grained access, meaning the compute engine can only access specific rows and columns from the full dataset. Choosing this option configures your compute to work with data asset subscriptions from Amazon SageMaker Catalog.
   - Select **project.spark.compatibility** to configure permission mode to be compatible with data managed using full-table access, meaning the compute engine can access all rows and columns in the data. Choosing this option configures your compute to work with data assets from AWS and from external systems that you connect to from your project.
7. After configuring these settings, select **Add compute**. After a short time, your serverless application running EMR Serverless should be added to your project.

Configuring permission mode for EMR Serverless in Amazon SageMaker Unified Studio

Permission mode is a configuration available to Spark compute resources such as Glue ETL or EMR Serverless. It configures Spark to access different types of data based on the permissions configured for that data. There are two configuration options for permission mode:

- Compatibility mode. This is a configuration for data managed using full-table access, meaning the compute engine can access all rows and columns in the data. Choosing this option enables your compute to work with data assets from AWS and from external systems.

- Fine-grained mode. This is a configuration for data managed using fine-grained access controls, meaning the compute engine can only access specific rows and columns from the full dataset. Choosing this option enables your Glue ETL to work with data asset subscriptions from Amazon SageMaker Catalog.

You configure permission mode in EMR Serverless computes in Amazon SageMaker Unified Studio when you add a new EMR Serverless compute resource in your project. For more information, see the section called “Adding a new EMR Serverless application”.

> **Note**
>
> You cannot modify the permission mode after the EMR Serverless compute resource is created. Instead, you can create another EMR Serverless compute resource with a different permission mode.

Configuring user background sessions for EMR Serverless

> **Warning**
>
> When user background sessions is enabled for EMR Serverless, Amazon SageMaker Unified Studio will not terminate interactive sessions. All interactive sessions will be only terminated once all queries are completed.

User background sessions for your EMR Serverless applications must be enabled manually. To enable user background sessions, perform the following action using your EMR Serverless application.

```
aws emr-serverless update-application \
  --region {aws-region-code} \
  --application-id {application-id} \
  --identity-center-configuration userBackgroundSessionsEnabled=true
```

For more information, see User background sessions in the EMR Serverless management guide.

Deleting applications

When you no longer need an EMR Serverless application, the application can be deleted.

To delete an application:

1. Login to the Amazon SageMaker Unified Studio studio and navigate to the Serverless tab of the Compute section. Select the name of the compute instance you would like to remove.
2. On the compute details page, select the **Delete** option.
3. A dialog box will appear asking you to confirm that you want to delete the application, which in this case is your EMR Serverless application. Confirm that you want to remove the compute by typing "confirm" in the text box.
4. Choose **Delete application** to begin termination and removal.
5. After a short time, your application should be removed.

Glue ETL in Amazon SageMaker Unified Studio

AWS Glue ETL compute resources power Visual ETL flows in your Amazon SageMaker Unified Studio project. You can use Glue ETL serverless compute resources to run Visual ETL flows and JupyterLab notebooks without managing underlying infrastructure. This is especially useful for analytics, machine learning, and application development.

You can view information about your AWS Glue ETL compute resources on the **Data processing** tab of the **Compute** page in your project. These resources are used when you create and run Visual ETL flows in Amazon SageMaker Unified Studio.

By default, when you create a project in Amazon SageMaker Unified Studio two Glue ETL compute connections are created. The Glue ETL connection with permission mode set to compatibility is called `project.spark.compatibility`, and the Glue ETL connection with permission mode set to fine-grained is called `project.spark.fineGrained`. You can choose which compute option to use when you use tools such as Visual ETL and JupyterLab in Amazon SageMaker Unified Studio. For more information about compatibility and fine-grained permission modes, see the section called “Configuring permission mode”.

> **Note**
>
> Amazon SageMaker Unified Studio automatically creates Glue ETL compute resources during project creation. You cannot create, edit, or delete these instances. By default, Glue ETL uses AWS Glue 5.0 with G.1X (1 Data Processing Unit / Hour) worker types.

Configuring permission mode for Glue ETL in Amazon SageMaker Unified Studio

Permission mode is a configuration available to Spark compute resources such as Glue ETL or EMR Serverless. It configures Spark to access different types of data based on the permissions configured for that data. There are two configuration options for permission mode:

- Compatibility mode. This is a configuration for data managed using full-table access, meaning the compute engine can access all rows and columns in the data. Choosing this option enables your compute to work with data assets from AWS and from external systems.
- Fine-grained mode. This is a configuration for data managed using fine-grained access controls, meaning the compute engine can only access specific rows and columns from the full dataset. Choosing this option enables your Glue ETL to work with data asset subscriptions from Amazon SageMaker Catalog.

To configure permission mode for Glue ETL in Amazon SageMaker Unified Studio, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to a project.
3. In the left navigation pane, under **Data analytics**, choose **Visual ETL**.
4. Navigate to a flow by creating one or selecting the flow from the list.
5. From the dropdown menu next to the **Run** button, choose a compute connection type that aligns with your data access preference.

    - Select **project.spark.fineGrained** to configure permission mode to support fine-grained access control. Choosing this option configures your Visual ETL flow to work with data asset subscriptions from Amazon SageMaker Catalog.

- Select **project.spark.compatibility** to configure permission mode to be compatible with general access control. Choosing this option configures your Visual ETL flow to work with data assets that you connect to from your project.

You can then run the Visual ETL flow with data that aligns with your selected compute connection.

Configuring user background sessions for AWS Glue

> **Warning**
>
> When user background sessions is enabled for AWS Glue, Amazon SageMaker Unified Studio will not terminate interactive sessions. All interactive sessions will be only terminated once all queries are completed.

User background sessions in AWS Glue must be enabled for the entire account. AWS Glue requires additional actions to enable user background sessions for the entire account.

```
aws glue update-glue-identity-center-configuration \
  --user-background-sessions-enabled
```

For more information, see User background sessions for AWS Glue ETL in the AWS Glue management guide.

Limitations of fine-grained permission mode

Permission mode is a configuration available to Spark compute resources such as Glue ETL or EMR Serverless. It configures Spark to access different types of data based on the permissions configured for that data. There are two configuration options for permission mode:

- Compatibility mode. This is a configuration for data managed using full-table access, meaning the compute engine can access all rows and columns in the data. Choosing this option enables your compute to work with data assets from AWS and from external systems.
- Fine-grained mode. This is a configuration for data managed using fine-grained access controls, meaning the compute engine can only access specific rows and columns from the full dataset. Choosing this option enables your Glue ETL to work with data asset subscriptions from Amazon SageMaker Catalog.

Consider the following considerations and limitations when you use fine-grained mode.

- Fine-grained mode supports fine-grained access control via AWS Lake Formation only for Apache Hive and Apache Iceberg tables. Apache Hive formats include Parquet, ORC, and CSV.
- When fine-grained mode is enabled, a minimum of four workers are required because it requires one system driver, system executors, one user driver, and optionally user executors (required if you use UDFs or spark.createDataFrame).
- Fine-grained mode supports cross-account table queries shared through resource links. The resource link needs to be named identically to the source account's resource.
- The following components aren't supported:
  - Resilient distributed datasets (RDD)
  - Spark streaming
  - Write with AWS Lake Formation granted permissions
  - Access control for nested columns
  - Access data stored on Amazon Redshift Managed Storage (RMS), including through lakehouse architecture.
- Fine-grained mode blocks functionalities that might undermine the complete isolation of the system driver, including the following:
  - UDTs, HiveUDFs, and any user-defined function that involves custom classes
  - Custom data sources
  - Supply of additional JARs for Spark extension, connector, or metastore
  - `ANALYZE TABLE` command
- To enforce access controls, `EXPLAIN PLAN` and DDL operations such as `DESCRIBE TABLE` don't expose restricted information.
- Fine-Grained mode restricts access to system driver Spark logs on Lake Formation-enabled applications. Since the system driver runs with more access, events and logs that the system driver generates can include sensitive information. To prevent unauthorized users or code from accessing this sensitive data, access to system driver logs is disabled. For troubleshooting, contact AWS support.
- The following are considerations and limitations when using Apache Iceberg:
  - You can only use Apache Iceberg with session catalog and not arbitrarily named catalogs.
  - Iceberg tables that are registered in AWS Lake Formation only support the metadata tables `history`, `metadata_log_entries`, `snapshots`, `files`, `manifests`, and `refs`. AWS

Glue hides the columns that might have sensitive data, such as `partitions`, `path`, and `summaries`. This limitation doesn't apply to Iceberg tables that aren't registered in AWS Lake Formation.

- Tables that you don't register in AWS Lake Formation support all Iceberg stored procedures except for the `register_table` and `migrate` procedures, which aren't supported for any tables.

CI/CD for Amazon SageMaker Unified Studio

The Amazon SageMaker Unified Studio CI/CD CLI (`aws-smus-cicd-cli`) is an open-source command line tool that deploys multi-service data and AI applications across development, test, and production environments from a single manifest file. The CLI is available at no additional cost — you pay only for the underlying AWS resources provisioned during deployment.

GitHub repository: github.com/aws/CICD-for-SageMakerUnifiedStudio on the GitHub website

When you build data and AI applications in Amazon SageMaker Unified Studio, your applications often combine multiple AWS services into a single workflow. Promoting these applications from development to test and production requires substituting service-specific configurations for each stage and provisioning resources in the correct order. The CI/CD CLI automates this process.

The CLI uses a declarative YAML manifest (`manifest.yaml`) that defines:

- **Application resources** — Glue ETL jobs, Athena queries, MWAA workflows, Bedrock agents, QuickSight dashboards, and catalog assets
- **Pipeline stages** — Each stage maps to an independent Amazon SageMaker Unified Studio project, providing full isolation between environments
- **Stage-specific configurations** — S3 bucket paths, IAM role ARNs, account IDs, and connection strings that are substituted automatically at deploy time

Key concept

Manifest

The manifest file (`manifest.yaml`) is the single source of truth for your application. It declares what resources your application uses, which Amazon SageMaker Unified Studio projects to deploy into, and how configurations differ across stages. Data teams own the manifest. DevOps teams do not need to modify it.

Stages and targets

Each stage in the manifest (for example, `dev`, `test`, `prod`) maps to a separate Amazon SageMaker Unified Studio project and domain. Stages can span different AWS accounts and Regions. The CLI

substitutes stage-specific configurations at deploy time, so the same application definition works across all environments.

Bundles

A bundle is an immutable, versioned archive produced by reading from a **source target** (typically your development environment). The `bundle` command packages your application code, workflow definitions, and resolved configurations from the source into a self-contained artifact.

The `deploy` command then deploys the bundle contents to a **destination target** (for example, test or production). This separation means:

- **Source → Bundle:** `aws-smus-cicd-cli bundle` reads from your source(dev) project and creates the artifact
- **Bundle → Destination:** `aws-smus-cicd-cli deploy` deploys the artifact into the target project

The same bundle is promoted across stages without rebuilding, supporting immutable artifacts, audit trails, and controlled promotion through quality gates.

Catalog support

The CLI supports catalog asset management as part of the deployment process. You can define business metadata, including glossaries, glossary terms, assets, and data products, in your manifest. During deployment, the CLI automatically searches for assets in the catalog, creates subscription requests for required access, and waits for subscription approval before proceeding.

CLI commands

The CI/CD CLI provides four core commands that cover the deployment lifecycle:

Command	Description
`aws-smus-cicd-cli describe`	Validates the manifest, checks that target projects exist, and confirms the execution role has required permissions. Use `--connect` to validate against live AWS environments.

Command	Description
`aws-smus-cicd-cli bundle`	Reads from the source target and packages application code, workflow definitions, and configurations into an immutable, versioned archive.
`aws-smus-cicd-cli deploy`	Deploys bundle contents to the destination target. Provisions resources in dependency order — for example, uploading scripts to S3 before creating Glue jobs that reference them. Use `--dry-run` to preview changes without deploying.
`aws-smus-cicd-cli test`	Runs post-deployment validation to confirm that services are running and ready for workloads in the target environment.

Additional commands:

Command	Description
`aws-smus-cicd-cli create`	Generates a starter manifest from an existing Amazon SageMaker Unified Studio project.
`aws-smus-cicd-cli run`	Triggers Airflow workflow execution on MWAA or Airflow Serverless connections.
`aws-smus-cicd-cli monitor`	Monitors workflow execution status in real time.
`aws-smus-cicd-cli logs`	Fetches and streams workflow execution logs.
`aws-smus-cicd-cli destroy`	Removes deployed resources and projects. Use for failure recovery or environment cleanup.

Prerequisites

Before using the CI/CD CLI, you need:

- **Python 3.8 or later**
- **AWS credentials** configured with permissions to deploy resources to your target Amazon SageMaker Unified Studio projects. See the section called “Sample IAM policy” for required permissions.
- **Existing Amazon SageMaker Unified Studio projects** at each target stage. The CLI deploys application workloads into existing projects — it does not create domains or projects. Use AWS CloudFormation, AWS CDK, or Terraform for infrastructure setup.

Getting started

The CLI source code, example manifests, and documentation are available in the GitHub repository on the GitHub website.

Install the CLI

Install the CLI using pip:

```
pip install aws-smus-cicd-cli
```

Verify the installation:

```
aws-smus-cicd-cli --version
```

Create a manifest

Clone the repository and start from an example manifest for your application pattern:

```
git clone https://github.com/aws/CICD-for-SageMakerUnifiedStudio.git
cp CICD-for-SageMakerUnifiedStudio/examples/data-notebooks/manifest.yaml manifest.yaml
```

Available example patterns:

- **Analytics** — Glue ETL + QuickSight dashboards

- **Data notebooks** — Jupyter notebook workflows
- **ML training** — SageMaker AI training jobs
- **ML deployment** — Model endpoints and inference
- **GenAI** — Bedrock agent applications

Edit the manifest to specify your resources, source locations, and stage-specific configurations.

Alternatively, generate a manifest from an existing project:

```
aws-smus-cicd-cli create --domain-id <domain-id> --dev-project-id <project-id>
```

Validate your configuration

Before deploying, you can preview what will happen without making changes:

```
# Validate permissions and connections
aws-smus-cicd-cli describe --manifest manifest.yaml --connect

# Preview deployment without applying changes
aws-smus-cicd-cli deploy --targets test --manifest manifest.yaml --dry-run
```

Deploy

```
aws-smus-cicd-cli deploy --targets test --manifest manifest.yaml
```

Run post-deployment validation

```
aws-smus-cicd-cli test --manifest manifest.yaml --targets test
```

Deployment modes

Direct deployment

Deploy directly from a local source directory:

```
aws-smus-cicd-cli deploy --manifest manifest.yaml --targets prod
```

Bundle-based deployment

Package your application from the source target into an immutable artifact, then promote it to the destination:

```
# Create the bundle (reads from source target)
aws-smus-cicd-cli bundle --manifest manifest.yaml

# Deploy the bundle to production (deploys to destination target)
aws-smus-cicd-cli deploy --manifest app.tar.gz --targets prod
```

Cross-region and cross-account deployment

Each deployment target in the manifest maps to an independent Amazon SageMaker Unified Studio project and domain. Targets can be in different AWS Regions and different AWS accounts. Provide AWS credentials for each target through IAM roles, OIDC federation, or environment variables at deploy time.

CI/CD integration

The CLI runs inside your existing CI/CD systems. Reusable workflow templates are available for:

- **GitHub Actions** — See GitHub Actions integration guide on the GitHub website
- **Jenkins** — Use the CLI commands in your Jenkinsfile pipeline stages
- **GitLab CI** — Use the CLI commands in your `.gitlab-ci.yml` stages

Sample IAM policy

The following policy grants the permissions required to deploy applications that include Glue, Athena, QuickSight, and SageMaker AI resources. Scope permissions to specific resources and accounts for production deployments.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:ListBucket",
                "glue:*",
                "quicksight:*",
                "athena:*",
                "lakeformation:*",
                "logs:*",
                "iam:GetRole",
                "iam:PassRole",
                "sts:GetCallerIdentity",
                "datazone:GetProject",
                "datazone:GetDomain",
                "datazone:GetEnvironment",
                "datazone:ListConnections",
                "datazone:GetConnection",
                "datazone:SearchListings",
                "datazone:GetListing"
            ],
            "Resource": "*"
        }
    ]
}
```

Failure recovery

The CLI stops at the point of failure and reports the error with a detailed stack trace. Resources provisioned before the failure remain in place. To recover:

1. Fix the issue identified in the error output.
2. Run `aws-smus-cicd-cli describe --connect` to confirm which resources exist and which permissions are missing.
3. Rerun `aws-smus-cicd-cli deploy`, or redeploy a previous bundle for bundle-based deployments.
4. To clean up a failed deployment, use `aws-smus-cicd-cli destroy --targets <target>` to remove deployed resources.

For detailed rollback procedures, see the Rollback Guide on the GitHub website.

Current limitations

- Automatic rollback is not supported. Recover by redeploying a previous bundle, deploying a fix, or using the `destroy` command.
- The CLI does not create Amazon SageMaker Unified Studio domains or projects. Use existing IaC tools for infrastructure setup.
- Native notebook deployment is not yet supported.

Additional resources

- GitHub repository on the GitHub website
- Quick Start Guide on the GitHub website
- Admin Guide on the GitHub website
- Manifest Reference on the GitHub website
- CLI Command Reference on the GitHub website
- Example Applications on the GitHub website
- Rollback Guide on the GitHub website

Kiro CLI in Amazon SageMaker Unified Studio

Kiro CLI is an AI coding agent in your Amazon SageMaker Unified Studio JupyterLab terminal. It gives you chat, code creation, and AWS tools. Kiro CLI comes pre-installed in the SageMaker image. It works with your project context and AWS credentials.

Kiro CLI uses Amazon Bedrock for AI. You don't need to set up any models.

Topics

- Prerequisites
- To sign in and start Kiro CLI
- Using Kiro CLI
- Kiro CLI capabilities in Amazon SageMaker Unified Studio
- Example: Explore your data catalog
- Example: Get project context
- Troubleshooting Kiro CLI
- Additional information

Prerequisites

Before you use Kiro CLI, you need the following.

- You need an Amazon SageMaker Unified Studio project with a JupyterLab space.
- You need an AWS Builder ID or IAM Identity Center (IdC) account for Kiro login.

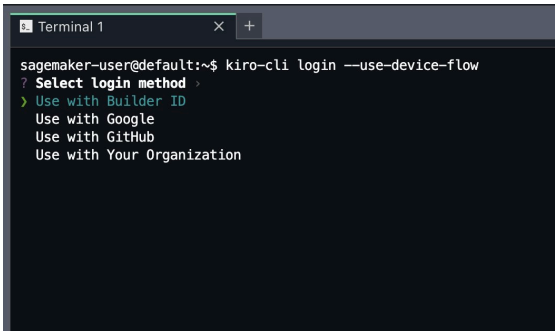
To sign in and start Kiro CLI

Use this process to sign in and start a chat session. It takes about 5 minutes.

1. Open your Amazon SageMaker Unified Studio project and navigate to your JupyterLab space.
2. Open a terminal by choosing **New**, then **Terminal**.
3. Sign in to Kiro CLI by running the following command.

```
kiro-cli login --use-device-flow
```

4. When prompted, select your login method:
   - **Use with Your Organization** – for IAM Identity Center (SSO) users
   - **Use with Builder ID** – for individual AWS Builder ID accounts



5. Enter your Start URL and Region when prompted (for IAM Identity Center users). A device code appears.
6. Open the URL in your browser. Confirm the code and allow the device.
7. After you see "Logged in successfully", start a chat. Run one of these commands.

```
kiro-cli
```

```
kiro-cli chat
```

Using Kiro CLI

After you sign in, Kiro CLI gives you a chat where you can ask questions, create code, and run AWS commands.

To start a basic session, run the following command.

```
kiro-cli
```

To start a session with a specific message, run the following command.

```
kiro-cli chat "List the tables in my Glue Data Catalog"
```

Kiro CLI capabilities in Amazon SageMaker Unified Studio

Kiro CLI works with your Amazon SageMaker Unified Studio environment. The following list describes what Kiro CLI can do.

- **AWS operations** – List and manage AWS resources, check IAM permissions, and run AWS CLI commands. Read-only operations run without asking. Write operations need your approval.
- **Code generation** – Create Python scripts, SQL queries, notebooks, and config files based on your requests.
- **Project context** – Kiro CLI detects your Amazon SageMaker Unified Studio project details. This includes domain ID, project ID, and Region. It uses the right AWS credentials for your project.
- **File operations** – Read and write files in your workspace, create notebooks, and edit code.

Example: Explore your data catalog

This example shows how Kiro CLI helps you find and query data in your project.

To explore your data catalog

1. In the terminal, start a chat session.

   ```
   kiro-cli chat
   ```

2. Ask Kiro to explore your data by entering the following prompt.

   ```
   List the tables in my Glue Data Catalog and show me the schema of any tables you
    find
   ```

   Kiro CLI calls the AWS Glue API. It lists your databases and tables, and shows column names and types.

Example: Get project context

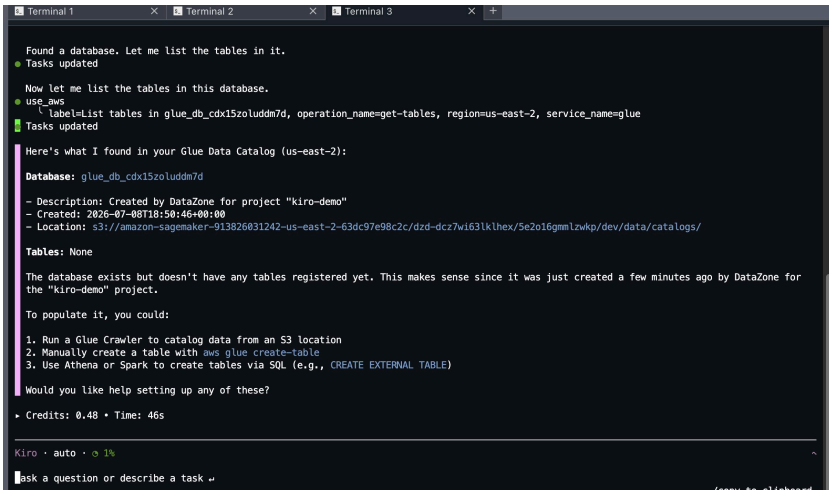
This example shows how Kiro CLI gives you project details for your Amazon SageMaker Unified Studio setup.

To get your project context

- In the terminal, enter the following prompt.

```
Can you tell me my project and domain information?
```

  Kiro CLI shows your domain ID, project ID, Region, and AWS profiles.



Troubleshooting Kiro CLI

Use these solutions to fix common issues with Kiro CLI in Amazon SageMaker Unified Studio.

"You're not signed in to Kiro yet"

Run `kiro-cli login --use-device-flow` in a terminal to authenticate.

Login fails with timeout

Check that your browser can reach the IdC or Builder ID login URL. Test network access from your browser, not from the space.

"Logged in successfully" but chat does not respond

Close and reopen the terminal, then run `kiro-cli chat` again.

Additional information

Keep these points in mind when you use Kiro CLI in Amazon SageMaker Unified Studio.

- Login sessions persist across space restarts. Kiro stores credentials on Amazon Elastic Block Store (Amazon EBS).
- Write operations need your approval. This includes making or deleting resources.
- Kiro CLI uses Amazon Bedrock for AI. You don't need to set up models.

Administration

For detailed procedures and steps, see the Amazon SageMaker Unified Studio Administration Guide.

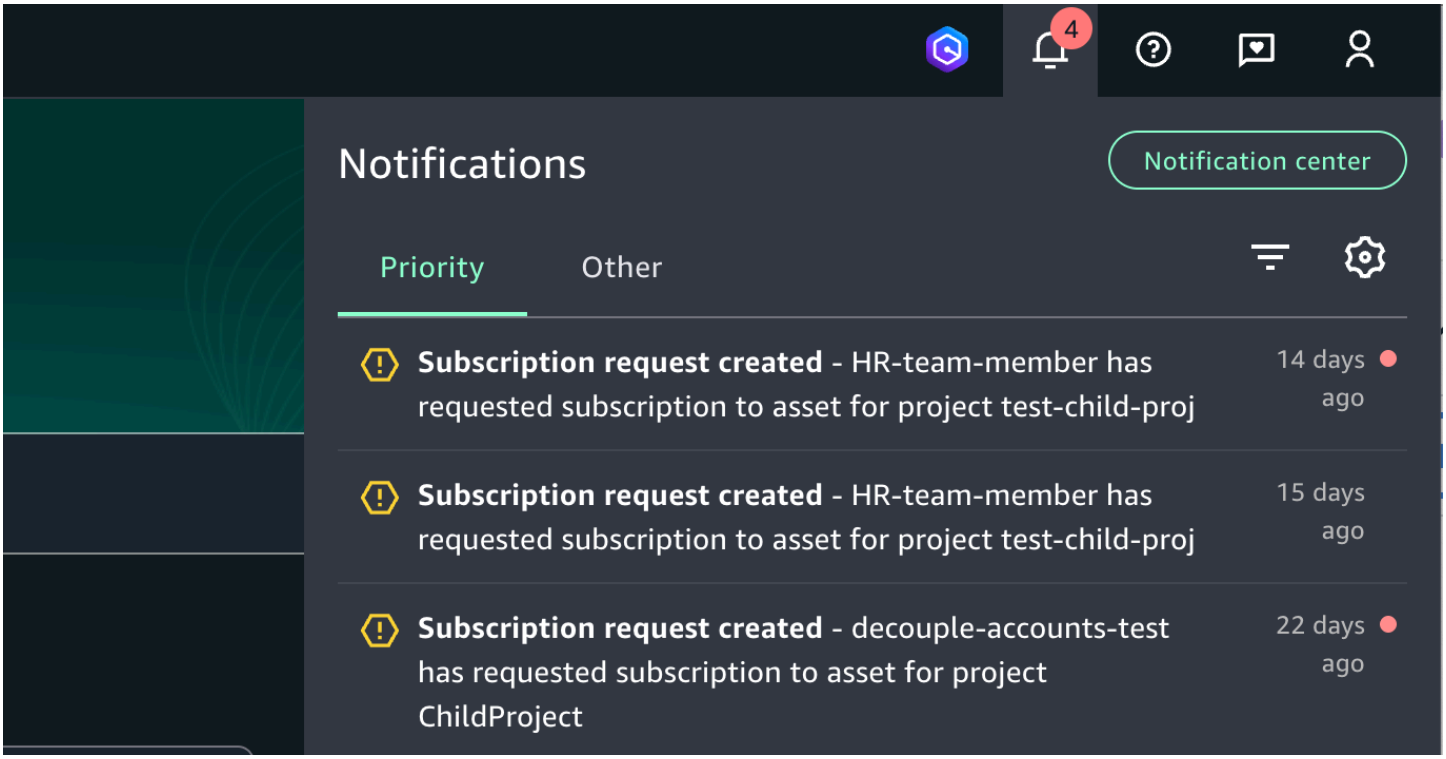
Amazon SageMaker Unified Studio events and notifications

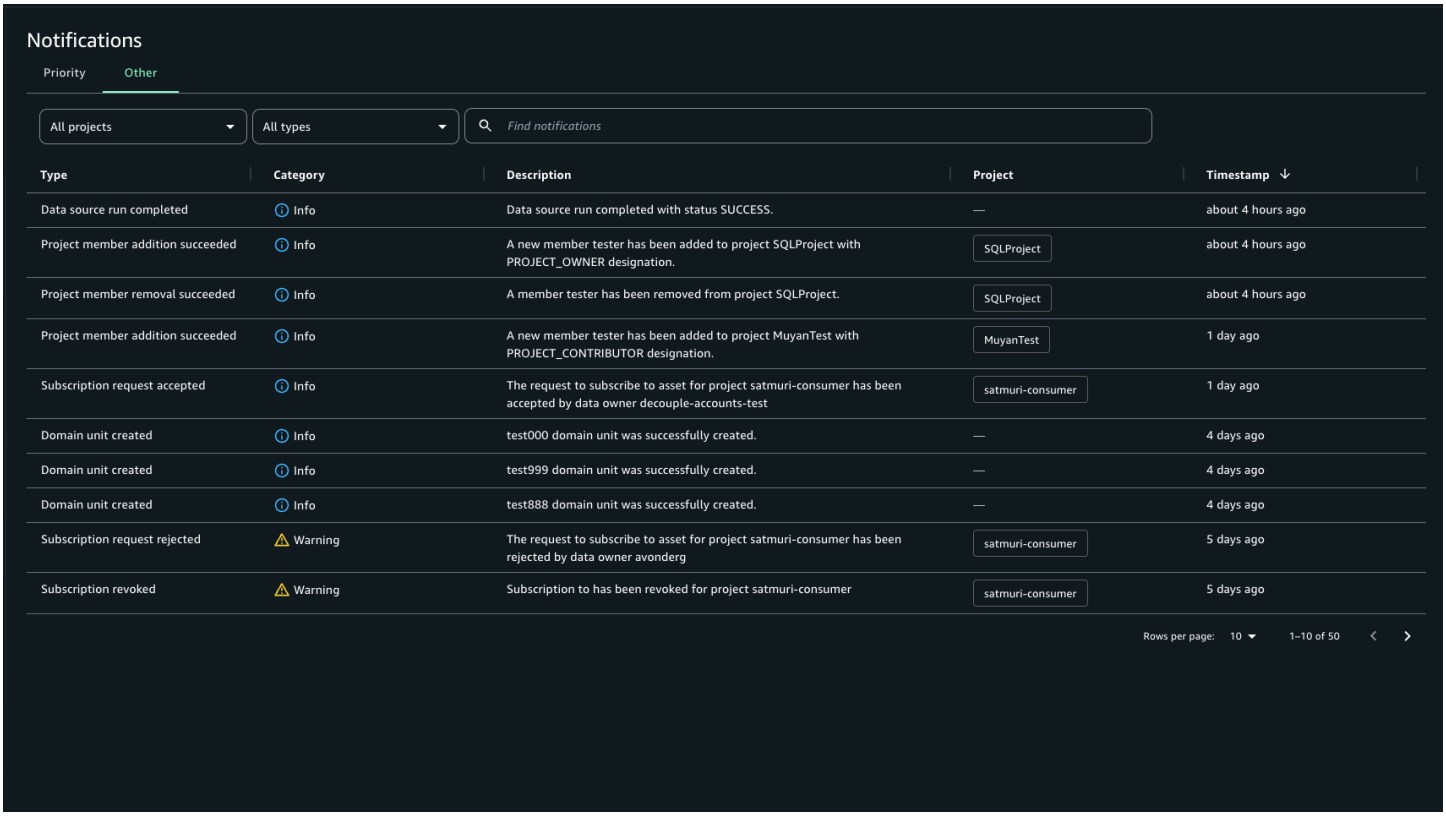
SageMaker Unified Studio keeps you informed of important activities within the Amazon SageMaker Unified Studio using categories, such as critical workflow, info, error, and warning. This mechanism informs stakeholders about activities within SageMaker Unified Studio using events, like a subscription requested, task/job completion, while notifications alert users about these events in the portal, or customized via emails or in-app alerts. With events and notifications, SageMaker Unified Studio provides you updates on various activities across projects and users by delivering messages in the dedicated inbox in the data portal.

Events via the dedicated inbox in the Amazon SageMaker Unified Studio

SageMaker Unified Studio provides a dedicated inbox in the SageMaker Unified Studio where you can see and take action on your messages. Recent messages will surface regardless of which page the user is viewing. For example, if a user requests access to a data asset, publishing project's owners and contributors of that asset see the request in the data portal and once an action is taken, project members of the subscribing project related to this request see the notification in the data portal. There are four categories of events:

- Error - Signal critical issues that halt progress requiring immediate intervention to restore functionality or resolve the problem. Trigger: System event.
- Critical workflow - these messages inform the recipient that there is action needed somewhere. They have an optional status field which you can use for tracking. Trigger: User action or system event.
- Info - these messages are informational and have no assigned status. Events provide an audit trail of recent updates. Trigger: User action or system event.
- Warning - these messages alert users of potential problems that need attention to prevent future failures but does not require immediate action. Trigger: System event.





In Amazon SageMaker Unified Studio, notifications are generated for the following events with detail types:

Notification Component	Event Detail Type	Event Description
Domain unit	Domain Unit Created	Event is generated when domain unit creation succeeds
Domain unit	Domain Unit Updated	Event is generated when domain unit update succeeds
Domain unit	Domain Unit Deleted	Event is generated when domain unit deletion succeeds
Domain unit	Domain Unit Owner Added	Event is generated when domain unit owner is added successfully
Domain unit	Domain Unit Owner Removed	Event is generated when domain unit owner is added successfully
Policy grant	Policy Grant Added	Event is generated when domain unit policy grant is added successfully
Policy grant	Policy Grant Removed	Event is generated when domain unit policy grant is removed successfully
Project membership	Project Member Addition Succeeded	Event is generated when a new member is added to a project
Project membership	Project Member Removal Succeeded	Event is generated when a member is removed to a project
Data source run	Data Source Run Complete	Event is generated when a new data source is created

Notification Component	Event Detail Type	Event Description
Catalog asset	Business Name Generation Succeeded	Event is generated when the automated business name generated job completes successfully
Catalog asset	Business Name Generation Failed	Event is generated when the automated business name generated job fails
Catalog asset	Metadata Generation Succeeded	Event is generated when asset metadata automatic generation succeeds
Catalog asset	Metadata Generation Failed	Event is generated when asset metadata automatic generation fails
Catalog asset	Metadata Generation Canceled	Event is generated when automatically generated metadata is canceled
Catalog asset	Metadata Generation Accepted	Event is generated when automatically generated metadata is approved
Catalog asset	Metadata Generation Rejected	Event is generated when automatically generated metadata is rejected
Subscription	Subscription Request Created	Event is generated when a subscription request is created
Subscription	Subscription Request Accepted	Event is generated when a subscription request is accepted

Notification Component	Event Detail Type	Event Description
Subscription	Subscription Request Rejected	Event is generated when a subscription request is rejected
Subscription	Subscription Request Deleted	Event is generated when a subscription request is deleted
Subscription	Subscription Revoked	Event is generated when a subscription is rejected by the publlishing project owner or contributor
Subscription	Subscription Auto Fulfill Completed	Event is generated when a subscription request is automatically fulfilled successfully
Search	Invalid Filter Detected	Event is generated when an asset schema changed and invalid filter detected
Data Product	Data Product Added To Catalog	Event is generated when an data product is published to catalog
Data Product Subscription	Subscribed Data Product Updated	Event is generated when an subscribed data product is updated

To view priority items in your data portal inbox, complete the following steps:

1. Navigate to the Amazon SageMaker Unified Studio using the data portal URL and log in using your SSO or AWS credentials. If you're an SageMaker Unified Studio administrator, you can obtain the data portal URL by accessing the SageMaker Unified Studio console at https://

console.aws.amazon.com/datazone in the AWS account where the SageMaker Unified Studio domain was created.

2. In the portal, to view a pop up with the recent set of notifications, select the bell icon in the top-right corner of the header.
3. Select Notification Center to view all notifications. You can change views and see all priority items by selecting the Priority tab.
4. You can filter the search by the notification subject, project scope, or category.
5. Choose any individual priority notification to navigate to the location where you can respond to the task.

To view other informative events in your data portal inbox, complete the following steps:

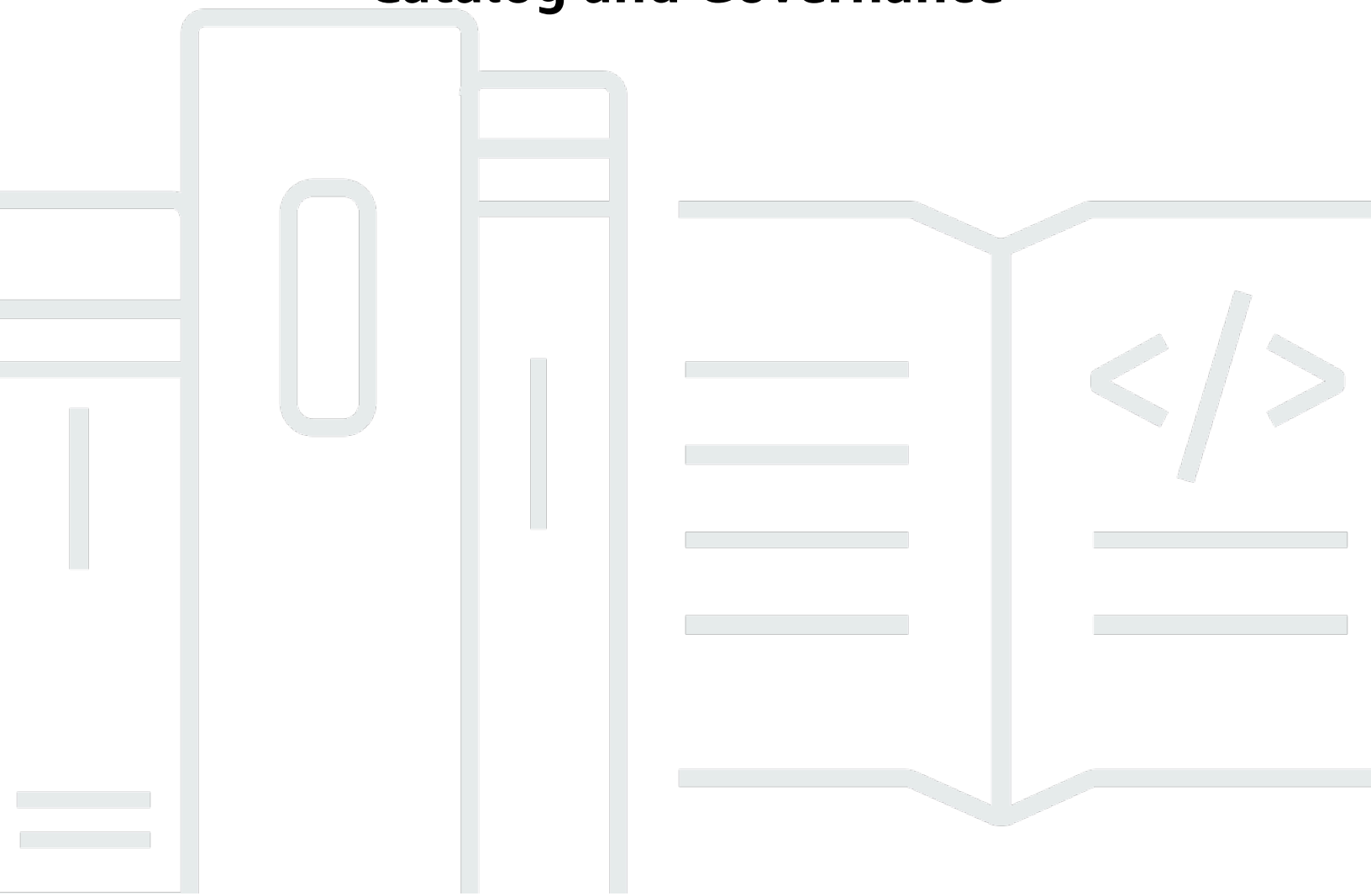
1. Navigate to the Amazon SageMaker Unified Studio using the portal URL and log in using your SSO or AWS credentials. If you're an SageMaker Unified Studio administrator, you can obtain the data portal URL by accessing the SageMaker Unified Studio console at https://console.aws.amazon.com/datazone in the AWS account where the SageMaker Unified Studio domain was created.
2. In the data portal, to view the pop up for the recent set of events, select the bell icon in the top-right corner of the header.
3. Select Notification Center to view all events. You can change views and see all tasks by selecting the "Other" tab.
4. Filter the search by the notification subject, project scope, or category.
5. Choose any individual notification to navigate to the location where you can view details about that event.

aws

User Guide

Amazon SageMaker Unified Studio

Catalog and Governance



Catalog in IDC-based domains

You can use the Amazon SageMaker Catalog to catalog data across your organization with business context and thus enable everyone in your organization to find and understand data quickly.

In order to use Amazon SageMaker Unified Studio to catalog your data, you must first bring your data (assets) as inventory of your project in Amazon SageMaker Unified Studio. Creating inventory for a project makes the assets discoverable only to that project's members. Project inventory assets are not available to all domain users in search/browse unless explicitly published.

After creating a project inventory, data owners can curate their inventory assets with the required business metadata by adding or updating business names (asset and schema), descriptions (asset and schema), read me, glossary terms (asset and schema), and metadata forms.

The next step of using Amazon SageMaker Unified Studio to catalog your data, is to make your project's inventory assets discoverable by the domain users. You can do this by publishing the inventory assets to the Amazon SageMaker Unified Studio catalog. Only the latest version of the inventory asset can be published to the catalog and only the latest published version is active in the discovery catalog. If an inventory asset is updated after it's been published into the Amazon SageMaker Unified Studio catalog, you must explicitly publish it again in order for the latest version to be in the discovery catalog.

For more information, see Amazon SageMaker Unified Studio terminology and concepts.

Topics

- Data governance and metadata
- Data products
- Data inventory and publishing
- Data discovery, subscription, and consumption
- Fine-grained access control to data
- Exporting asset metadata

Data governance and metadata

Data governance in Amazon SageMaker Unified Studio encompasses the management of business glossaries, metadata forms, and asset classification to ensure consistent data understanding

across your organization. Both business glossaries and metadata forms work together to provide comprehensive business context for your data assets, making it easier for users across your organization to discover, understand, and effectively use data in Amazon SageMaker Unified Studio.

A business glossary is a centralized collection of business terms and their definitions that ensures consistent vocabulary usage when analyzing data, while metadata forms are customizable forms that allow data owners to augment asset metadata with additional business context beyond the technical metadata automatically collected by the service.

Glossaries and metadata forms are owned by the project that creates them. Only members of the owning project can edit or delete a glossary, its terms, or a metadata form. However, glossaries, their terms, and metadata forms are visible to all users in the domain. Users in any project within the domain can search for and view glossaries and terms, and can attach glossary terms and metadata forms to their assets to curate metadata. This allows a governance team to create and manage glossary terms and metadata forms in a dedicated project while users across the domain use them to describe and categorize their data assets.

Topics

- Create a business glossary in Amazon SageMaker Unified Studio
- Edit a business glossary in Amazon SageMaker Unified Studio
- Delete a business glossary in Amazon SageMaker Unified Studio
- Create a term in a glossary in Amazon SageMaker Unified Studio
- Edit a term in a glossary in Amazon SageMaker Unified Studio
- Delete a term in a glossary in Amazon SageMaker Unified Studio
- Create a metadata form in Amazon SageMaker Unified Studio
- Edit a metadata form in Amazon SageMaker Unified Studio
- Delete a metadata form in Amazon SageMaker Unified Studio
- Create a field in a metadata form in Amazon SageMaker Unified Studio
- Edit a field in a metadata form in Amazon SageMaker Unified Studio
- Delete a field in a metadata form in Amazon SageMaker Unified Studio
- Restricted asset classification Amazon SageMaker Unified Studio

Create a business glossary in Amazon SageMaker Unified Studio

In Amazon SageMaker Unified Studio, a business glossary is a collection of business terms (words) that may be associated with assets (data). It provides appropriate vocabularies with a list of business terms and their definitions for business users to ensure the same definitions are used across the organization when analyzing data. Business glossaries are created in the catalog domain and can be applied to assets and columns to help understand key characteristics of that asset or column. One or more glossary terms can be applied. A business glossary can be a flat list of terms where any term in the business glossary can be associated with a sublist of other terms. For more information, see Amazon SageMaker Unified Studio terminology and concepts. To create, edit, or delete a glossary in your Amazon SageMaker Unified Studio domain, you must be a member of the owning project with the right permissions for that domain.

To create a glossary, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, choose **Catalog**.
3. Choose **View business glossaries**, and then choose **Create glossary**.
4. Specify a name, description, and owning project for the glossary and then choose **Create glossary**.
5. Optional - if you want to create a restricted glossary, then choose **Restrict this glossary for governed term use**. And then specify the usage permission by selecting one of the following options:
   - All projects - give permissions to all projects in this domain
   - (Default) Owning project - give permissions ONLY to the owning project
   - Selected projects or domain units - give permission to specific projects and/or domain units
6. Enable the new glossary by choosing the **Enabled** toggle.

> **Note**
>
> After you create and enable a glossary, it is visible to all users in the domain. Users in other projects can search for, view, and attach the glossary's terms to their assets. Only members of the project that owns the glossary can edit or delete it.

To disable or enable a business glossary, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, choose **Catalog**.
3. Choose **View business glossaries**.
4. Select the business glossary that you want to disable or enable.
5. On the glossary details page, locate the **Enabled** toggle and use it to enable or disable your selected glossary.

> **Note**
>
> Disabling a glossary also disables all the terms that it contains.

Edit a business glossary in Amazon SageMaker Unified Studio

In Amazon SageMaker Unified Studio, a business glossary is a collection of business terms (words) that may be associated with assets (data). It provides appropriate vocabularies with a list of business terms and their definitions for business users to ensure the same definitions are used across the organization when analyzing data. Business glossaries are created in the catalog domain and can be applied to assets and columns to help understand key characteristics of that asset or column. One or more glossary terms can be applied. A business glossary can be a flat list of terms where any term in the business glossary can be associated with a sublist of other terms. For more information, see Amazon SageMaker Unified Studio terminology and concepts. To edit a glossary in your Amazon SageMaker Unified Studio domain, you must be a member of the owning project with the right permissions for that domain.

To edit a business glossary, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, choose **Catalog**.
3. Choose **View business glossaries**.
4. Select the business glossary that you want to edit.

5. On the glossary details page, expand **Actions** and then choose **Edit business glossary** to edit the glossary.
6. Make your updates to the name and description, and then choose **Update glossary**.

Delete a business glossary in Amazon SageMaker Unified Studio

In Amazon SageMaker Unified Studio, a business glossary is a collection of business terms (words) that may be associated with assets (data). It provides appropriate vocabularies with a list of business terms and their definitions for business users to ensure the same definitions are used across the organization when analyzing data. Business glossaries are created in the catalog domain and can be applied to assets and columns to help understand key characteristics of that asset or column. One or more glossary terms can be applied. A business glossary can be a flat list of terms where any term in the business glossary can be associated with a sublist of other terms. For more information, see Amazon SageMaker Unified Studio terminology and concepts. To delete a glossary in your Amazon SageMaker Unified Studio domain, you must be a member of the owning project with the right permissions for that domain.

To delete a business glossary, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, choose **Catalog**.
3. Choose **View business glossaries**.
4. Select the business glossary that you want to delete.
5. On the glossary details page, make sure the **Enabled** toggle is off, so that the glossary is disabled.
6. Expand **Actions** and then choose **Delete** to delete the glossary.

> **Note**
>
> You must delete all existing terms in the glossary before you can delete the glossary.

7. Confirm the deletion of the glossary by choosing **Delete**.

Create a term in a glossary in Amazon SageMaker Unified Studio

In Amazon SageMaker Unified Studio, a business glossary is a collection of business terms that may be associated with assets (data). For more information, see Amazon SageMaker Unified Studio terminology and concepts. To create, edit, or delete terms in a glossary in your Amazon SageMaker Unified Studio domain, you must be a member of the owning project with the right permissions for that domain.

In Amazon SageMaker Unified Studio, business glossary terms can have close descriptions. To set the context of a particular term, you can specify relationships among terms. When you define a relationship for a term, it is automatically added to the definition of the related term. The glossary term relationships available in Amazon SageMaker Unified Studio include the following:

- **Is a Type of** - indicates that the current term is a type of the identified term. Indicates that the identified term is a parent to the current term.
- **Has Types** - indicates that the current term is a generic term for the indicated specific term or terms. This relationship can denote child terms for the generic term.

To create a new term, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, choose **Catalog**.
3. Choose **View business glossaries**.
4. Select the glossary where you want to create the new term.
5. Choose **Create term**.
6. Specify a name and description for the term and then choose **Create term**.
7. Enable the new term by choosing the **Enabled** toggle.
8. To add a **Readme**, select the name of the term to navigate to the term details page. Then choose **Create readme** to add some additional information about this glossary.
9. To add relationships, complete the following steps:

    a. Select the name of the term to navigate to the term details page.

    b. If this is the first relationship added to the term, under **Terms relationships**, choose **Add terms**. If there are other terms relationships listed, under **Term Relationships**, choose **Edit**, and then choose **Add terms**.

c. In the dialog, choose the relationship and the terms you want to relate.

d. Choose **Add terms** to add the selected terms to the appropriate relationship type. This relationship is also added to all the terms you made related.

Edit a term in a glossary in Amazon SageMaker Unified Studio

In Amazon SageMaker Unified Studio, a business glossary is a collection of business terms that may be associated with assets (data). For more information, see Amazon SageMaker Unified Studio terminology and concepts. To create, edit, or delete terms in a glossary in your Amazon SageMaker Unified Studio domain, you must be a member of the owning project with the right permissions for that domain.

In Amazon SageMaker Unified Studio, business glossary terms can have close descriptions. To set the context of a particular term, you can specify relationships among terms. When you define a relationship for a term, it is automatically added to the definition of the related term. The glossary term relationships available in Amazon SageMaker Unified Studio include the following:

- **Is a Type of** - indicates that the current term is a type of the identified term. Indicates that the identified term is a parent to the current term.
- **Has Types** - indicates that the current term is a generic term for the indicated specific term or terms. This relationship can denote child terms for the generic term.

To edit a term in a glossary, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, choose **Catalog**.
3. Choose **View business glossaries**.
4. Select the glossary that contains the term that you you want to edit.
5. Choose the name of the term to navigate to the term details page.
6. On the term details page, expand **Actions** and then choose **Edit** to edit the term.
7. Make your updates to the name and description, and then choose **Update term**.

Delete a term in a glossary in Amazon SageMaker Unified Studio

In Amazon SageMaker Unified Studio, a business glossary is a collection of business terms that may be associated with assets (data). For more information, see Amazon SageMaker Unified Studio terminology and concepts. To create, edit, or delete terms in a glossary in your Amazon SageMaker Unified Studio domain, you must be a member of the owning project with the right permissions for that domain.

In Amazon SageMaker Unified Studio, business glossary terms can have close descriptions. To set the context of a particular term, you can specify relationships among terms. When you define a relationship for a term, it is automatically added to the definition of the related term. The glossary term relationships available in Amazon SageMaker Unified Studio include the following:

- **Is a Type of** - indicates that the current term is a type of the identified term. Indicates that the identified term is a parent to the current term.
- **Has Types** - indicates that the current term is a generic term for the indicated specific term or terms. This relationship can denote child terms for the generic term.

To delete a term in a glossary, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, choose **Catalog**.
3. Choose **View business glossaries**.
4. Select the glossary that contains the term that you you want to delete.
5. Select the name of the term to navigate to the term details page.
6. Make sure that the **Enabled** toggle is off so that the term is disabled.
7. On the glossary term details page, expand **Actions** and then choose **Delete**.
8. Confirm the deletion of the term by choosing **Delete**.

Create a metadata form in Amazon SageMaker Unified Studio

In Amazon SageMaker Unified Studio, metadata forms are simple forms to augment additional business context to the asset metadata in the catalog. They serve as extensible mechanisms for data owners to enrich the asset with information that can help data users when they search and

find that data. Metadata forms can also serve a mechanism to enforce consistency to all assets being published to the Amazon SageMaker Unified Studio catalog.

A metadata form definition is composed of one or more field definitions, with support for boolean, date, decimal, integer, string, and business glossary field value data types. For more information, see Amazon SageMaker Unified Studio terminology and concepts. To create, edit, or delete metadata forms in your Amazon SageMaker Unified Studio domain, you must be a member of the owning project who has the right credentials.

To create a metadata form, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, choose **Catalog**.
3. Choose **View metadata forms**.
4. Choose **Create metadata form**.
5. Specify the metadata form technical name, owning project, and optional display name and description.
6. Choose **Create metadata form**.

> **Note**
>
> After you create a metadata form, it is visible to all users in the domain. Users in other projects can attach the metadata form to their assets to add business context. Only members of the project that owns the metadata form can edit or delete it.

Edit a metadata form in Amazon SageMaker Unified Studio

In Amazon SageMaker Unified Studio, metadata forms are simple forms to augment additional business context to the asset metadata in the catalog. They serve as extensible mechanisms for data owners to enrich the asset with information that can help data users when they search and find that data. Metadata forms can also serve a mechanism to enforce consistency to all assets being published to the Amazon SageMaker Unified Studio catalog.

A metadata form definition is composed of one or more field definitions, with support for boolean, date, decimal, integer, string, and business glossary field value data types. For more information,

see Amazon SageMaker Unified Studio terminology and concepts. To create, edit, or delete metadata forms in your Amazon SageMaker Unified Studio domain, you must be a member of the owning project who has the right credentials.

To edit a metadata form, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, choose **Catalog**.
3. Choose **View metadata forms**.
4. Choose the name of the metadata form that you want to edit. This takes you to the metadata form details page.
5. On the metadata form details page, expand **Actions**, and then choose **Edit metadata form**.
6. Perform your updates to the name, description, and owning project.
7. Choose **Update form**.

Delete a metadata form in Amazon SageMaker Unified Studio

In Amazon SageMaker Unified Studio, metadata forms are simple forms to augment additional business context to the asset metadata in the catalog. They serve as extensible mechanisms for data owners to enrich the asset with information that can help data users when they search and find that data. Metadata forms can also serve a mechanism to enforce consistency to all assets being published to the Amazon SageMaker Unified Studio catalog.

A metadata form definition is composed of one or more field definitions, with support for boolean, date, decimal, integer, string, and business glossary field value data types. For more information, see Amazon SageMaker Unified Studio terminology and concepts. To create, edit, or delete metadata forms in your Amazon SageMaker Unified Studio domain, you must be a member of the owning project who has the right credentials.

To delete a metadata form, complete the following steps:

> **Note**
>
> Before you can delete a metadata form, you must remove it from all asset types or assets to which it is applied.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, choose **Catalog**.
3. Choose **View metadata forms**.
4. Choose the name of the metadata form that you want to delete. This takes you to the metadata form details page.
5. If the metadata form that you want to delete is enabled, disable the metadata form by choosing the **Enabled** toggle.
6. On the metadata form's details page, expand **Actions**, and then choose **Delete**.
7. Confirm deletion by choosing **Delete**.

Create a field in a metadata form in Amazon SageMaker Unified Studio

In Amazon SageMaker Unified Studio, metadata forms are simple forms to augment additional business context to the asset metadata in the catalog. They serve as extensible mechanisms for data owners to enrich the asset with information that can help data users when they search and find that data. Metadata forms can also serve an mechanism to enforce consistency to all assets being published to the Amazon SageMaker Unified Studio catalog.

A metadata form definition is composed of one or more field definitions, with support for boolean, date, decimal, integer, string, and business glossary field value data types. For more information, see Amazon SageMaker Unified Studio terminology and concepts. To create, edit, or delete fields in metadata forms in your Amazon SageMaker Unified Studio domain, you must be a member of the owning project who has the right credentials.

To create a field in a metadata form, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, choose **Catalog**.
3. Choose **View metadata forms**.
4. Choose the name of the metadata form that you want to add a field to. This takes you to the metadata form details page.
5. On the metadata form details page, choose **Create field**.

6. Specify the field name, description, type, and whether this is a required field. Depending on the field type, you might be able to configure additional selections.
7. Choose **Create field**.

Edit a field in a metadata form in Amazon SageMaker Unified Studio

In Amazon SageMaker Unified Studio, metadata forms are simple forms to augment additional business context to the asset metadata in the catalog. They serve as extensible mechanisms for data owners to enrich the asset with information that can help data users when they search and find that data. Metadata forms can also serve an mechanism to enforce consistency to all assets being published to the Amazon SageMaker Unified Studio catalog.

A metadata form definition is composed of one or more field definitions, with support for boolean, date, decimal, integer, string, and business glossary field value data types. For more information, see Amazon SageMaker Unified Studio terminology and concepts. To create, edit, or delete fields in metadata forms in your Amazon SageMaker Unified Studio domain, you must be a member of the owning project who has the right credentials.

To edit a field in a metadata form, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, choose **Catalog**.
3. Choose **View metadata forms**.
4. Choose the name of the metadata form that contains the field that you want to edit. This takes you to the metadata form details page.
5. On the metadata form details page, choose the field that you want to edit.
6. Expand the **Actions** menu, and then choose **Edit field**.
7. Make your updates to the field name, description, type, and whether it is a required field. Make updates to other selections if more are available with the selected field type.
8. Choose **Save**.

Delete a field in a metadata form in Amazon SageMaker Unified Studio

In Amazon SageMaker Unified Studio, metadata forms are simple forms to augment additional business context to the asset metadata in the catalog. They serve as extensible mechanisms for

data owners to enrich the asset with information that can help data users when they search and find that data. Metadata forms can also serve an mechanism to enforce consistency to all assets being published to the Amazon SageMaker Unified Studio catalog.

A metadata form definition is composed of one or more field definitions, with support for boolean, date, decimal, integer, string, and business glossary field value data types. For more information, see Amazon SageMaker Unified Studio terminology and concepts. To create, edit, or delete fields in metadata forms in your Amazon SageMaker Unified Studio domain, you must be a member of the owning project who has the right credentials.

To delete a field in a metadata form, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, choose **Catalog**.
3. Choose **View metadata forms**.
4. Choose the name of the metadata form that contains the field that you want to delete. This takes you to the metadata form details page.
5. On the metadata form details page, choose the field that you want to delete.
6. Expand the **Actions** menu, and choose **Delete**.
7. Confirm deletion by choosing **Delete**.

Restricted asset classification Amazon SageMaker Unified Studio

Restricted classification allows domain unit owners and glossary project owners to control who can apply specific classification terms to assets in the Amazon SageMaker Catalog. This feature helps maintain classification consistency and governance standards across your domain while enabling controlled workflows based on governed classification terms.

Restricted classification provides the following benefits:

- Governance control - maintain consistent classification standards across your entire domain
- Access management - control which users can apply sensitive or restricted classification terms to the assets
- Workflow enablement - build automated workflows based on governed classification terms
- Clear separation - distinguish between open and restricted classification terms

SageMaker catalog now separates classification terms into two categories:

- Unrestricted terms - available for all users to apply to their assets
- Restricted terms - only authorized users can apply these to assets they own

This functionality uses the following authorization model:

- Restricted glossaries can be created and managed by glossary project owners and contributors. Project owners have the ability to grant usage permissions to specific domain units as well as to other project owners and contributors. If a restricted glossary is created by a contributor, only the project owner is granted permission to use it by default.
- Project owners are by default granted access to use the glossary for the assets in their projects.
- Authorized users are granted permissions to use the restricted terms for their projects by adding project specific grants.
- All users can filter and discover assets using restricted classification terms.

When using restricted glossaries in Amazon SageMaker Unified Studio, you must abide by the following constraints:

- Scope of application – restricted glossary terms are currently supported only at the asset level. Column-level terms, metadata form–level terms and data product-level terms are not currently supported.
- Term relationships – restricted glossary terms cannot be related to other terms.
- Glossary usage permission conversion – once created, a restricted glossary cannot be converted into a regular glossary and a regular glossary cannot be converted into a restricted glossary.

Creating restricted classification terms

As a project owner or contributor:

- Navigate to the catalog governance section
- Create a new restricted classification glossary
- Define terms within the glossary
- Set usage policies for the restricted glossary

Applying restricted terms to assets

Complete the following procedure based on the configured usage permission for the restricted glossary:

Apply restricted terms to assets

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, choose **Catalog**.
3. Choose **Browse assets**.
4. Find the asset to which you want to assign restricted terms and on the asset's details page, choose **View inventory asset**.
5. Under **Glossary terms**, choose **Add terms**, then search for the restricted term that you want to assign to this asset, choose it, and then choose **Add term**.

   Once the term is successfully added, you can identify it as a restricted term by the presence of a lock icon next to its name.

You can associate or disassociate up to 5 restricted terms with/from an asset at any one time.

Data products

In Amazon SageMaker Unified Studio, data producers can group data assets into well-defined, self-contained packages called *data products* that are tailored for specific business use cases. Using cohesive, business-aligned data products enhances both the publishing and the subscription processes.

Data consumers can identify interconnected data assets by searching for and finding them as a single unit. This approach reduces the time and effort required to find all relevant information and lowers the risk of missing important data. Also, data products simplify access to data with a single request by implementing a unified access model. This eliminates the need for multiple permissions, thereby speeding up the initiation of data analysis.

By cataloging assets as data products, data producers reduce administrative overhead by enabling metadata and access control management at the data product level, rather than individually. The ability to surface these purpose-built grouped assets for consumption makes access governance

and data utilization more efficient, ensuring it aligns with business goals and is easily accessible for its intended use. Data governance teams can monitor consumption rates for these data products, providing valuable insights into data literacy maturity. For more information, see Amazon SageMaker Unified Studio terminology and concepts.

Topics

- Create new data products in Amazon SageMaker Unified Studio
- Publish data products in Amazon SageMaker Unified Studio
- Edit data products in Amazon SageMaker Unified Studio
- Unpublish data products in Amazon SageMaker Unified Studio
- Delete data products in Amazon SageMaker Unified Studio
- Subscribe to a data product in Amazon SageMaker Unified Studio
- Review a subscription request and grant a subscription to a data product in Amazon SageMaker Unified Studio
- Republish data products in Amazon SageMaker Unified Studio

Create new data products in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio enables data producers to group data assets into well-defined, self-contained packages called data products that are tailored for specific business use cases. For more information, see Amazon SageMaker Unified Studio terminology and concepts.

Any Amazon SageMaker Unified Studio user with the required permissions can create a Amazon SageMaker Unified Studio data product.

To create a new data product complete the following steps.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project in which you'd like to create a data product. You can do this by choosing a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.
4. Expand the **Create** menu and then choose **Create data product**.
5. In the **Create new data product** page, complete the following steps.

a. Specify the name and the description for the data product

b. Choose **Select assets** to add various assets to your data product.

c. In the **Select assets** pop-up window, choose **Choose** next to the assets that you want to add to this data product.

d. Choose **Choose** at the bottom of the pop-up window.

6. To complete creating the data product, choose **Create**.

Publish data products in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio enables data producers to group data assets into well-defined, self-contained packages called data products that are tailored for specific business use cases. For more information, see Amazon SageMaker Unified Studio terminology and concepts.

Any Amazon SageMaker Unified Studio user with the required permissions can publish an Amazon SageMaker Unified Studio data product.

To publish a data product complete the following steps.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project that contains the data product that you want to publish. You can do this by choosing a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.
4. Choose the **Inventory** tab, and then choose the **Data products** filter. This displays existing data products in the project inventory.
5. Choose the data product that you want to publish. This opens the data product details page.
6. Choose **Publish**. Confirm the publishing of this data product by choosing **Publish data product**.

> **Note**
>
> Any unpublished data assets that are in this data product will become published, but will only be available through this data product.

Edit data products in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio enables data producers to group data assets into well-defined, self-contained packages called data products that are tailored for specific business use cases. For more information, see Amazon SageMaker Unified Studio terminology and concepts.

Any Amazon SageMaker Unified Studio user with the required permissions can edit an Amazon SageMaker Unified Studio data product.

To edit a data product complete the following steps.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project that contains the data product that you want to edit. You can do this by choosing a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.
4. Choose the **Inventory** tab, and then choose the **Data products** filter. This displays existing data products in the project inventory.
5. Choose the data product that you want to edit. As part of editing a data product, you can do the following:

   - Choose **Create README** to add a README that will help users understand the page better.
   - Choose **Add terms** to add glossary terms. Make your selections of glossary terms in the **Add terms** window and then choose **Add terms**.
   - Choose **Add metadata form** and then select your form in the **Add metadata form** window and choose **Add**.
   - Expand **Actions**, choose **Edit**, make your edits to the name and description of the data product, and then choose **Update**.
   - On the **Assets** tab, remove one of the existing assets in the data product by choosing that asset, then expanding the three-dot action icon and choosing **Remove asset**. Confirm the asset removal by choosing **Remove** in the **Remove asset** pop-up window. When you republish, this asset will be removed from all subscribers to this data product.
   - On the **Assets** tab, add a new asset to the data product by choosing the **Add** button and then selecting one or more assets to be added to the data product.

Unpublish data products in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio enables data producers to group data assets into well-defined, self-contained packages called data products that are tailored for specific business use cases. For more information, see Amazon SageMaker Unified Studio terminology and concepts.

Any Amazon SageMaker Unified Studio user with the required permissions can unpublish an Amazon SageMaker Unified Studio data product.

To unpublish a data product complete the following steps.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project that contains the data product that you want to unpublish. You can do this by choosing a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.
4. Choose the **Inventory** tab, and then choose the **Data products** filter. This displays existing data products in the project inventory.
5. Choose the data product that you want to unpublish. This opens the data product details page.
6. Expand **Actions** and choose **Unpublish**. Confirm the unpublishing of this data product by choosing **Unpublish**.

> **Note**
>
> Unpublishing a data product has the following effects:
>
> - This data product will no longer be available to view or to subscribe to.
> - Any data assets that are only available through this data product will no longer be available.
> - All active subscriptions to this data product will remain.
> - Any individually published data assets will not be affected.

Delete data products in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio enables data producers to group data assets into well-defined, self-contained packages called data products that are tailored for specific business use cases. For more information, see Amazon SageMaker Unified Studio terminology and concepts.

Any Amazon SageMaker Unified Studio user with the required permissions can delete an Amazon SageMaker Unified Studio data product.

To delete a data product complete the following steps.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project that contains the data product that you want to delete. You can do this by choosing a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.
4. Choose the **Inventory** tab, and then choose the **Data products** filter. This displays existing data products in the project inventory.
5. Choose the data product that you want to delete.
6. Expand **Actions** and choose **Delete**. Confirm the deletion of this data product by typing `delete` in the text field and then choosing **Delete**.

> **Note**
>
> Deleting a data product has the following effects:
>
> - The data product will no longer be available to publish, view, or subscribe to.
> - Any data assets that are only available through this data product will no longer be visible in the data catalog. They will not be deleted from your inventory assets.

Subscribe to a data product in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio enables data producers to group data assets into well-defined, self-contained packages called data products that are tailored for specific business use cases. For more information, see Amazon SageMaker Unified Studio terminology and concepts.

Any Amazon SageMaker Unified Studio user with the required permissions can subscribe to an Amazon SageMaker Unified Studio data product.

To subscribe to a data product, complete the following steps.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, choose **Catalog**.
3. Choose **Browse data products**.
4. Find the data product to which you want to subscribe and then choose that data product.
5. On the data product details page, choose **Subscribe**.
6. Specify the project and the reason for requesting a subscription. Then choose **Request**.

When the owning project grants the subscription request, you will be subscribed to the data product.

Review a subscription request and grant a subscription to a data product in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio enables data producers to group data assets into well-defined, self-contained packages called data products that are tailored for specific business use cases. For more information, see Amazon SageMaker Unified Studio terminology and concepts.

The owning project of the data product can review and grant the subscription to an Amazon SageMaker Unified Studio data product.

To review a subscription request and grant or reject a subscription to a data product, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project that contains the data product that has a subscription request. You can do this by choosing a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.
4. Choose the **Inventory** tab, and then choose the **Data products** filter. This displays existing data products in the project inventory.

5. Choose the data product that has a subscription request.
6. Choose the **Subscription requests** tab.
7. Locate the request that you want to review and then choose **View request**.
8. In the **Subscription request** window, type in a decision comment. Then choose either **Approve** or **Reject**.

Republish data products in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio enables data producers to group data assets into well-defined, self-contained packages called data products that are tailored for specific business use cases. For more information, see Amazon SageMaker Unified Studio terminology and concepts.

Any Amazon SageMaker Unified Studio user with the required permissions can republish an Amazon SageMaker Unified Studio data product.

To republish a data product complete the following steps.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project that contains the data product that you want to edit. You can do this by choosing a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.
4. Choose the **Inventory** tab, and then choose the **Data products** filter. This displays existing data products in the project inventory.
5. Choose the data product that you want to republish.
6. Make the desired edits to the data product. For more information, see the section called “Edit data products”.
7. On the data product's details page, choose **Re-publish**. Confirm this action by choosing **Re-publish data product** in the **Re-publish data product** pop-up window.

> **Note**
>
> Republishing this data product will update the following for all subscribers:
>
> - If assets have been removed from the data product, subscribers will no longer have access to these assets.

- If assets have been added to the data product, subscribers will get access to these assets.
- New published versions of data assets will be available.

Data inventory and publishing

This section describes the tasks and procedures to create an inventory of your data in Amazon SageMaker Unified Studio and to publish your data in Amazon SageMaker Unified Studio.

To use Amazon SageMaker Unified Studio to catalog your data, you must first bring your data (assets) as inventory of your project in Amazon SageMaker Unified Studio. Creating an inventory for a particular project makes the assets discoverable only to that project's members. Project inventory assets are not available to all domain users in search or browse unless it is published to the Amazon SageMaker Catalog. After creating a project inventory, data owners can curate their inventory assets with the required business metadata by adding or updating business names (asset and schema), descriptions (asset and schema), README, glossary terms (asset and schema), and metadata forms.

The next step of using Amazon SageMaker Unified Studio to catalog your data is to make your project's inventory assets discoverable by the domain users. You can do this by publishing the inventory assets to the Amazon SageMaker Unified Studio catalog. Publishing an asset makes it searchable and discoverable in the catalog by other domain users and by SageMaker Data Agent, which uses published business context to generate more accurate code. Only the latest version of the inventory asset can be published to the catalog and only the latest published version is active in the discovery catalog. If an inventory asset is updated after it's been published into the Amazon SageMaker Unified Studio catalog, you must publish it again for the latest version to be in the discovery catalog.

For more information, see Amazon SageMaker Unified Studio terminology and concepts

Topics

- Configure Lake Formation permissions for Amazon SageMaker Unified Studio
- Create custom asset types in Amazon SageMaker Unified Studio
- Create an Amazon SageMaker Unified Studio data source for AWS Glue in the project catalog
- Create an Amazon SageMaker Unified Studio data source for Amazon Redshift in the project catalog

- Create an Amazon SageMaker Unified Studio data source for Amazon SageMaker AI in the project catalog
- Edit a data source in Amazon SageMaker Unified Studio
- Delete a data source in Amazon SageMaker Unified Studio
- Publish assets to the Amazon SageMaker Unified Studio catalog from the project inventory
- Share assets
- Manage inventory and curate assets in Amazon SageMaker Unified Studio
- Manually create an asset in Amazon SageMaker Unified Studio
- Unpublish an asset from the Amazon SageMaker Catalog
- Delete an Amazon SageMaker Unified Studio asset
- Manually start a data source run in Amazon SageMaker Unified Studio
- Asset revisions in Amazon SageMaker Unified Studio
- Data quality in Amazon SageMaker Unified Studio
- Using machine learning and generative AI in Amazon SageMaker Unified Studio
- Analyze Amazon SageMaker Unified Studio data with external analytics applications via JDBC connection
- Metadata enforcement rules for publishing

Configure Lake Formation permissions for Amazon SageMaker Unified Studio

When you create a project in Amazon SageMaker Unified Studio, an AWS Glue database is added as part of this project. If you want to publish assets from this AWS Glue database, no additional permissions are needed.

However, if you want to publish assets and subscribe to assets from an AWS Glue database that exists outside of your Amazon SageMaker Unified Studio project, you must explicitly provide your project with the permissions to access tables in the external AWS Glue database. To do this, you must complete the following settings in AWS Lake Formation and attach necessary AWS Lake Formation permissions to the project's IAM role role.

- Configure the Amazon S3 location for your data lake in AWS Lake Formation with **Lake Formation** permission mode or **Hybrid access mode**. For more information, see https://docs.aws.amazon.com/lake-formation/latest/dg/register-data-lake.html.

- Attach the following AWS Lake Formation permissions to the AWS Glue manage access role:
  - `Describe` and `Describe grantable` permissions on the database where the tables exist.
  - `Describe`, `Select`, `Describe Grantable`, `Select Grantable` permissions on the all the tables in the above database that you want DataZone to manage access on your behalf.

> **Note**
>
> Amazon SageMaker Unified Studio supports the AWS Lake Formation hybrid mode. Lake Formation hybrid mode enables you to start managing permissions on you AWS Glue databases and tables through Lake Formation, while continuing to maintain any existing IAM permissions on these tables and databases.

Create custom asset types in Amazon SageMaker Unified Studio

In Amazon SageMaker Unified Studio, assets represent specific types of data resources such as database tables, dashboards, or machine learning models. To provide consistency and standardization when describing catalog assets, an Amazon SageMaker Unified Studio domain must have a set of asset types that define how assets are represented in the catalog. An asset type defines the schema for a specific type of asset. An asset type has a set of required and optional nameable metadata form types. Asset types in Amazon SageMaker Unified Studio are versioned. When assets are created, they are validated against the schema defined by their asset type (typically latest version), and if an invalid structure is specified, asset creation fails.

System asset types - Amazon SageMaker Unified Studio provisions service-owned system asset types. System asset types cannot be altered. Amazon SageMaker Unified Studio includes the following system asset types:

- Amazon Bedrock chat app
- Amazon Bedrock flow app
- Amazon Bedrock inference only
- Amazon Bedrock model
- Amazon Bedrock prompt
- Databricks table
- Databricks view

- AWS Glue table
- AWS Glue view
- Amazon Redshift table
- Amazon Redshift view
- Amazon S3 object collection
- SageMaker feature group
- SageMaker model package group
- Snowflake table
- Snowflake view
- Data product

Custom asset types - to create custom asset types, you start by creating the required metadata form types and glossaries to use in the form types. You can then create custom asset types by specifying a name, description, and associated metadata forms that can be required or optional.

For asset types with structured data, to represent the column schema in Amazon SageMaker Unified Studio, you can use the `RelationalTableFormType` to add the technical metadata to your columns, including column names, descriptions, and data types, and the `ColumnBusinessMetadataForm` to add the business descriptions of the columns, including business names, glossary terms, and custom key value pairs.

To create a custom asset type in Amazon SageMaker Unified Studio, complete the following steps:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project where you want to create a custom asset type.
3. In the left navigation pane, choose **Catalog**.
4. Choose **View asset types**.
5. Choose **Create asset type**.
6. Specify the following:
    - **Name** - the name of the custom asset type
    - **Description** - the description of the custom asset type.

- Choose **Add metadata form** to add metadata forms to this custom asset type.
- Under **Usage permission**, restrict access, by specify which projects or domain units are authorized to use this asset type.

> **Note**
>
> You must be a domain unit owner or a project owner in order to modify usage permissions. Project contributors can view usage permissions but cannot edit them.

You can choose the following:

- **All projects** - give permissions to all projects in this domain
- **Owning project** - give permissions only to the owning project
- **Selected projects or domain units** - give permissions to specific projects and/or domain units

  If you select this option, choose **Add usage permission**, and in the **Add projects and designations** pop up window, specify the authorized projects (you can choose **Select projects in a domain unit** or **All project in a domain unit**), the specific domain unit, and the allowed designations - which designations a project member must have to use this policy. You can choose **Owner** or **Contributor**.

7. Choose **Create**. After the custom asset type is created, you can use it to create assets.

Create an Amazon SageMaker Unified Studio data source for AWS Glue in the project catalog

In Amazon SageMaker Unified Studio, you can create an AWS Glue Data Catalog data source in order to import technical metadata of database tables from AWS Glue. To add a data source for the AWS Glue Data Catalog, the source database must already exist in AWS Glue. Your Amazon SageMaker Unified Studio project's IAM role also needs certain permissions to be able to create a data source, as described in the section the section called "Configure Lake Formation permissions for Amazon SageMaker Unified Studio".

When you create and run an AWS Glue data source, you add assets from the source AWS Glue database to your Amazon SageMaker Unified Studio project's inventory. You can run your AWS Glue data sources on a set schedule or on demand to create or update your assets' technical

metadata. During the data source runs, you can optionally choose to publish your assets to the Amazon SageMaker Unified Studio catalog and thus make them discoverable by all domain users. You can also publish your project inventory assets after editing their business metadata. Domain users can search for and discover your published assets, and request subscriptions to these assets.

> **Note**
>
> Adding a data source in the project catalog makes it possible to publish that data into the Amazon SageMaker Catalog. To add a data source for analyzing and editing within your project, use the **Data** page of your project. Data that you add to your connect to on the **Data** page can also be published to the Amazon SageMaker Catalog. For more information, see the section called “Lakehouse architecture”.

To create an AWS Glue data source

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project to which you want to add the data source.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Data sources**.
4. Choose **Create data source**.
5. Configure the following fields:
   - **Name** – The data source name.
   - **Description** – The data source description.
6. Under **Data source type**, choose **AWS Glue**.
7. (Optional) Under **Connection**, select **Import data lineage** if you want to import lineage for the data sources that use the connection.
8. Under **Data selection**, provide an AWS Glue database and provide a catalog, database names, and criteria for tables. For example, if you choose **Include** and enter `*corporate`, the database will include all source tables that end with the word `corporate`.

   You can either choose an AWS Glue catalog from the dropdown or type a catalog name. The dropdown includes the default AWS Glue catalog for the connection account.

You can add multiple include and exclude rules for tables. You can also add multiple databases using the **Add another database** button.

9. Choose **Next**.
10. For **Publishing settings**, choose whether assets are immediately discoverable in the Amazon SageMaker Catalog. If you only add them to the inventory, you can choose subscription terms later and then publish them to the Amazon SageMaker Catalog.
11. For **Metadata generation methods**, choose whether to automatically generate metadata for assets as they're imported from the source.
12. Under **Data quality**, you can choose to **Enable data quality for this data source**. If you do this, Amazon SageMaker Unified Studio imports your existing AWS Glue data quality output into your Amazon SageMaker Unified Studio catalog. By default, Amazon SageMaker Unified Studio imports the latest existing 100 quality reports with no expiration date from AWS Glue.

    Data quality metrics in Amazon SageMaker Unified Studio help you understand the completeness and accuracy of your data sources. Amazon SageMaker Unified Studio pulls these data quality metrics from AWS Glue in order to provide context during a point in time, for example, during a business data catalog search. Data users can see how data quality metrics change over time for their subscribed assets. Data producers can ingest AWS Glue data quality scores on a schedule. The Amazon SageMaker Unified Studio business data catalog can also display data quality metrics from third-party systems through data quality APIs.
13. (Optional) For **Metadata forms**, add forms to define the metadata that is collected and saved when the assets are imported into Amazon SageMaker Unified Studio. For more information, see the section called “Create a metadata form”.
14. Choose **Next**.
15. For **Run preference**, choose when to run the data source.
    - **Run on a schedule** – Specify the dates and time to run the data source.
    - **Run on demand** – You can manually initiate data source runs.
16. Choose **Next**.
17. Review your data source configuration and choose **Create**.

You can also create a Amazon SageMaker Unified Studio data source for Amazon Redshift by invoking the `CreateDataSource` API action or the `create-data-source` CLI action:

```
aws datazone create-data-source --cli-input-json file://create-sagemaker-datasource-
example.json
```

Sample payload (create-sagemaker-datasource-example.json per example above) to create an Amazon Sagemaker data sources in an Amazon DataZone domain:

```
{
  "name": "my-data-source",
  "projectIdentifier": "project123",
  "type": "GLUE",
  "description": "Description of the datasource",
  "environmentIdentifier": "environment123",
  "configuration": {
    "glueRunConfiguration": {
        "autoImportDataQualityResult": "True",
        "relationalFilterConfigurations": [{
            "databaseName": "my_database",
            "filterExpressions": [{
                "expression": "*",
                "type": "INCLUDE"
            }]
        }]
    }
  },
  "recommendation": {
    "enableBusinessNameGeneration": "True"
  },
  "enableSetting": "ENABLED",
  "schedule": {
    "timezone": "UTC",
    "schedule": "cron(7 22 * * ? *)"
  },
  "publishOnImport": "True",
  "assetFormsInput": [
    {
      "formName": "AssetCommonDetailsForm"
      "typeIdentifier": "amazon.datazone.AssetCommonDetailsFormType",
      "typeRevision": "3",
      "content": "form-content",
      "renderingConfig": {
```

```
        "collapse": "True""
      }
    }
  ],
  "clientToken": "123456"
}
```

Sample payload (create-sagemaker-datasource-example.json per example above) to create an Amazon Sagemaker data sources in an Amazon SageMaker unified domain:

```
{
  "name": "my-data-source",
  "projectIdentifier": "project123",
  "type": "GLUE",
  "description": "Description of the datasource",
  "connectionIdentifier": "connection123",
  "configuration": {
    "glueRunConfiguration": {
        "catalogName": "my_catalog",
        "autoImportDataQualityResult": "True",
        "relationalFilterConfigurations": [{
            "databaseName": "my_database",
            "filterExpressions": [{
                "expression": "*",
                "type": "INCLUDE"
            }]
        }]
    }
  },
  "recommendation": {
    "enableBusinessNameGeneration": "True"
  },
  "enableSetting": "ENABLED",
  "schedule": {
    "timezone": "UTC",
    "schedule": "cron(7 22 * * ? *)"
  },
  "publishOnImport": "True",
  "assetFormsInput": [
    {
      "formName": "AssetCommonDetailsForm"
```

```
      "typeIdentifier": "amazon.datazone.AssetCommonDetailsFormType",
      "typeRevision": "3",
      "content": "form-content"
    }
  ],
  "clientToken": "123456"
}
```

Create an Amazon SageMaker Unified Studio data source for Amazon Redshift in the project catalog

In Amazon SageMaker Unified Studio, you can create an Amazon Redshift data source in order to import technical metadata of database tables and views from the Amazon Redshift data warehouse. To add a Amazon SageMaker Unified Studio data source for Amazon Redshift, the source data warehouse must already exist in the Amazon Redshift.

When you create and run an Amazon Redshift data source, you add assets from the source Amazon Redshift data warehouse to your Amazon SageMaker Unified Studio project's inventory. You can run your Amazon Redshift data sources on a set schedule or on demand to create or update your assets' technical metadata. During the data source runs, you can optionally choose to publish your project inventory assets to the Amazon SageMaker Unified Studio catalog and thus make them discoverable by all domain users. You can also publish your inventory assets after editing their business metadata. Domain users can search for and discover your published assets and request subscriptions to these assets.

> **Note**
>
> Adding a data source in the project catalog makes it possible to publish that data into the Amazon SageMaker Catalog. To add a data source for analyzing and editing within your project, use the **Data** page of your project. Data that you add to your connect to on the **Data** page can also be published to the Amazon SageMaker Catalog. For more information, see the section called “Lakehouse architecture”.

To add an Amazon Redshift data source

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.

2. From the project selector dropdown at the top of the page, choose the project to which you want to add the data source.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Data sources**.
4. Choose **Create data source**.
5. Configure the following fields:

   - **Name** – The data source name.
   - **Description** – The data source description.
6. Under **Data source type**, choose **Amazon Redshift**.
7. Under **Connection**, select a connection for your data source. The connection cannot be changed after the data source is created.
8. Under **Data selection**, provide an Amazon Redshift database schema name and enter your table or view selection criteria. For example, if you choose **Include** and enter `*corporate`, the asset will include all source tables that end with the word `corporate`.

   You can add multiple include rules. You can also add another schema using the **Add another schema** button.
9. Choose **Next**.
10. For **Publishing settings**, choose whether assets are immediately discoverable in Amazon SageMaker Catalog. If you only add them to the inventory, you can choose subscription terms later and then publish them to the Amazon SageMaker Catalog.
11. For **Metadata generation methods**, choose whether to automatically generate metadata for assets as they're published and updated from the source.
12. (Optional) For **Metadata forms**, add forms to define the metadata that is collected and saved when the assets are imported into Amazon SageMaker Unified Studio. For more information, see the section called “Create a metadata form”.
13. Choose **Next**.
14. For **Run preference**, choose when to run the data source.

    - **Run on a schedule** – Specify the dates and time to run the data source.
    - **Run on demand** – You can manually initiate data source runs.
15. Choose **Next**.
16. Review your data source configuration and choose **Create**.

You can also create a Amazon SageMaker Unified Studio data source for Amazon Redshift by invoking the `CreateDataSource` API action or the `create-data-source` CLI action:

```
aws datazone create-data-source --cli-input-json file://create-sagemaker-datasource-
example.json
```

Sample payload (create-sagemaker-datasource-example.json per example above) to create an Amazon Sagemaker data sources in an Amazon DataZone domain:

```
{
  "name": "my-data-source",
  "projectIdentifier": "project123",
  "type": "REDSHIFT",
  "description": "Description of the datasource",
  "environmentIdentifier": "environment123",
  "configuration": {
    "redshiftRunConfiguration": {
        "dataAccessRole": "arn:aws:iam::123456789012:role/my-data-access-role",
        "redshiftCredentialConfiguration": {
                "secretManagerArn": "arn:aws:secretsmanager:us-
east-1:123456789012:secret:my-secret"
            },
        "redshiftStorage": {
            "redshiftClusterSource": {
                "clusterName": "my-redshift-cluster"
            }
        },
        "relationalFilterConfigurations": [{
            "databaseName": "my_database",
            "filterExpressions": [{
                "expression": "*",
                "type": "INCLUDE"
            }],
            "schemaName": "my_schema"
        }]
    }
  },
  "recommendation": {
    "enableBusinessNameGeneration": "True"
  },
```

```
  "enableSetting": "ENABLED",
  "schedule": {
    "timezone": "UTC",
    "schedule": "cron(7 22 * * ? *)"
  },
  "publishOnImport": "True",
  "assetFormsInput": [
    {
      "formName": "AssetCommonDetailsForm"
      "typeIdentifier": "amazon.datazone.AssetCommonDetailsFormType",
      "typeRevision": "3",
      "content": "form-content"
    }
  ],
  "clientToken": "123456"
}
```

Sample payload (create-sagemaker-datasource-example.json per example above) to create an Amazon Sagemaker data sources in an Amazon SageMaker unified domain:

```
{
  "name": "my-data-source",
  "projectIdentifier": "project123",
  "type": "REDSHIFT",
  "description": "Description of the datasource",
  "connectionIdentifier": "connection123",
  "configuration": {
    "redshiftRunConfiguration": {
        "dataAccessRole": "arn:aws:iam::123456789012:role/my-data-access-role",
        "redshiftCredentialConfiguration": {
                "secretManagerArn": "arn:aws:secretsmanager:us-
east-1:123456789012:secret:my-secret"
            },
        "redshiftStorage": {
            "redshiftClusterSource": {
                "clusterName": "my-redshift-cluster"
            }
        },
        "relationalFilterConfigurations": [{
            "databaseName": "my_database",
            "filterExpressions": [{
```

```
                "expression": "*",
                "type": "INCLUDE"
            }],
            "schemaName": "my_schema"
        }]
    }
  },
  "recommendation": {
    "enableBusinessNameGeneration": "True"
  },
  "enableSetting": "ENABLED",
  "schedule": {
    "timezone": "UTC",
    "schedule": "cron(7 22 * * ? *)"
  },
  "publishOnImport": "True",
  "assetFormsInput": [
    {
      "formName": "AssetCommonDetailsForm"
      "typeIdentifier": "amazon.datazone.AssetCommonDetailsFormType",
      "typeRevision": "3",
      "content": "form-content"
    }
  ],
  "clientToken": "123456"
}
```

Create an Amazon SageMaker Unified Studio data source for Amazon SageMaker AI in the project catalog

In the current release of Amazon SageMaker Unified Studio, creating an Amazon Sagemaker AI data source is not supported via the UI and can only be done by envoking API or CLI actions.

In order to create a data source for Amazon SageMaker AI in Amazon SageMaker Unified Studio, you must first to create a RAM share between Amazon SageMaker and Amazon DataZone. This RAM share is necessary for Amazon SageMaker to successfully make Amazon DataZone API calls which are needed for various membership and security checks.

If you're using the Amazon DataZone domain, you can complete this step by adding Amazon SageMaker as a trusted service in your Amazon DataZone domain.

If you're using a Amazon SageMaker unified domain, you can do this by completing the following procedure:

1. Navigate to the RAM console at https://us-east-1.console.aws.amazon.com/ram/home.
2. Choose **Create resource share**.
3. For resource share name enter a unique name. For example `DataZone-<DataZone DomainId>-SageMaker`.
4. Under **Resources** choose **DataZone Domains** from the drop down and then select the Amazon DataZone domain from the list and then choose **Next**.
5. From the Managed Permissions dropdown, choose **AWSRAMSageMakerServicePrincipalPermissionAmazonDataZoneDomain** and then choose **Next**.
6. Under **Principals** from the **Select principal type** dropdown, choose **Service principal**.
7. Enter `sagemaker.amazonaws.com` for the service principal name and choose **Add** and then choose **Next**.
8. Choose **Create resource share**.

Once this is completed, you can invoke the `CreateDataSource` API action or the `create-data-source` CLI action to create a new data source for Amazon SageMaker AI in Amazon SageMaker Unified Studio.

```
aws datazone create-data-source --cli-input-json file://create-sagemaker-datasource-example.json
```

Sample payload (create-sagemaker-datasource-example.json per example above) to create an Amazon Sagemaker data sources in an Amazon DataZone domain:

```
{
  "name": "my-data-source",
  "projectIdentifier": "project123",
  "type": "SAGEMAKER",
  "description": "Description of the datasource",
  "environmentIdentifier": "environment123",
  "configuration": {
```

```
    "sageMakerRunConfiguration": {
        "trackingAssets": {
            "SageMakerModelPackageGroupAssetType": [
                "arn:aws:sagemaker:us-east-1:123456789012:model-package-group/my-model-
package-group",
            ],
            "SageMakerFeatureGroupAssetType": [
                "arn:aws:sagemaker:us-east-1:123456789012:feature-group/my-feature-
group",
            ]
        }
    }
  },
  "recommendation": {
    "enableBusinessNameGeneration": "True"
  },
  "enableSetting": "ENABLED",
  "schedule": {
    "timezone": "UTC",
    "schedule": "cron(7 22 * * ? *)"
  },
  "publishOnImport": "True",
  "assetFormsInput": [
    {
      "formName": "AssetCommonDetailsForm"
      "typeIdentifier": "amazon.datazone.AssetCommonDetailsFormType",
      "typeRevision": "3",
      "content": "form-content"
    }
  ],
  "clientToken": "123456"
}
```

Sample payload (create-sagemaker-datasource-example.json per example above) to create an Amazon Sagemaker data sources in an Amazon SageMaker unified domain:

```
{
  "name": "my-data-source",
  "projectIdentifier": "project123",
  "type": "SAGEMAKER",
  "description": "Description of the datasource",
```

```
  "connectionIdentifier": "connection123",
  "configuration": {
    "sageMakerRunConfiguration": {
        "trackingAssets": {
            "SageMakerModelPackageGroupAssetType": [
                "arn:aws:sagemaker:us-east-1:123456789012:model-package-group/my-model-
package-group",
            ],
        }
    }
  },
  "recommendation": {
    "enableBusinessNameGeneration": "True"
  },
  "enableSetting": "ENABLED",
  "schedule": {
    "timezone": "UTC",
    "schedule": "cron(7 22 * * ? *)"
  },
  "publishOnImport": "True",
  "assetFormsInput": [
    {
      "formName": "AssetCommonDetailsForm"
      "typeIdentifier": "amazon.datazone.AssetCommonDetailsFormType",
      "typeRevision": "3",
      "content": "form-content"
    }
  ],
  "clientToken": "123456"
}
```

Edit a data source in Amazon SageMaker Unified Studio

After you create an Amazon SageMaker Unified Studio data source, you can modify it to change the source details or the data selection criteria. When you no longer need a data source, you can delete it.

To edit a data source in the project catalog

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.

2. From the project selector dropdown at the top of the page, choose the project that contains the data source that you want to edit.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Data sources**.
4. Choose the data source that you want to modify.
5. Expand the **Actions** menu, then choose **Edit data source**.
6. Make your changes to the data source fields as desired, then choose **Save**.

Delete a data source in Amazon SageMaker Unified Studio

When you no longer need an Amazon DataZone data source, you can remove it permanently. After you delete a data source, all assets that originated from that data source are still available in the catalog, and users can still subscribe to them. However, the assets will stop receiving updates from the source. We recommend that you first move the dependent assets to a different data source before you delete it.

To delete a data source in the project catalog

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project that contains the data source that you want to edit.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Data sources**.
4. Choose the data source that you want to delete.
5. Expand the **Actions** menu, then choose **Delete data source**.
6. To confirm deletion, type `delete` in the text entry field. Then choose **Delete**.

Publish assets to the Amazon SageMaker Unified Studio catalog from the project inventory

You can publish Amazon SageMaker Unified Studio assets and their metadata from project inventories into the Amazon SageMaker Unified Studio catalog. You can only publish the most recent version of an asset to the catalog.

Consider the following when publishing assets to the catalog:

- To publish an asset to the catalog, you must be the owner or contributor of the project that contains the asset.
- For Amazon Redshift assets, ensure that the Amazon Redshift clusters associated with both publisher and subscriber clusters meet all the requirements for Amazon Redshift data sharing in order for Amazon SageMaker Unified Studio to manage access for Redshift tables and views. See Data sharing concepts for Amazon Redshift.

Publish an asset in Amazon SageMaker Unified Studio

If you didn't choose to make assets immediately discoverable in the data catalog when you created a data source, perform the following steps to publish them later.

To publish an asset

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project to which the asset belongs.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.
4. Make sure you are on the **Inventory** tab, then choose the name of the asset that you want to publish. You are then brought to the asset details page.

   > **Note**
   >
   > By default, all assets require subscription approval, which means a data owner must approve all subscription requests to the asset. If you want to change this setting before publishing the asset, open the asset details and choose **Edit** next to **Subscription approval**. You can change this setting later by modifying and re-publishing the asset.

5. Choose **Publish asset**. The asset is directly published to the catalog.

   If you make changes to the asset, such as modifying its approval requirements, you can choose **Re-publish asset** to publish the updates to the catalog.

Share assets

In the current release of Amazon SageMaker Unified Studio, you can share your Amazon S3 assets, AWS Glue (SageMaker Lakehouse) assets, and your Amazon QuickSight assets with other projects or users/groups.

For more information about sharing your Amazon QuickSight assets, see Share Amazon QuickSight dashboards.

For more information about sharing your Amazon S3 assets, see Sharing Amazon S3 data.

To share your AWS Glue (SageMaker Lakehouse) data, complete the following procedure:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project. You can select either one of the projects that you created manually or the project that was automatically created when you've onboarded your Amazon SageMaker Lakehouse data.
3. In the left navigation pane, choose **Data**, then choose the catalog that you want to work with under **Lakehouse** and navigate down to the database and the table asset that you want to share.
4. Choose the asset that you want to share, then expand **Actions**, and choose **Share**.
5. In the **Share table** window, specify the project with which you want to share this asset and then choose **Share**.

> **Note**
>
> In the current release of Amazon SageMaker Unified Studio, sharing row and column filters is not supported.

Manage inventory and curate assets in Amazon SageMaker Unified Studio

In order to use Amazon SageMaker Unified Studio to catalog your data, you must first bring your data (assets) as inventory of your project in Amazon SageMaker Unified Studio. Creating inventory for a particular project makes the assets discoverable only to that project's members.

After the assets are created in project inventory, their metadata can be curated. For example, you can edit the asset's name, description, or README. Each edit to the asset creates a new version of the asset. You can use the History tab on the asset's details page to view all asset versions.

You can edit the **README** section and add rich descriptions for the asset. The **README** section supports markdown, thus enabling you to format your descriptions as required and describe key information about an asset to consumers.

Glossary terms can be added at the asset level by filling out available forms.

To curate the schema, you can review the columns, add business names, descriptions, and add glossary terms at column level.

To update the schema of an asset

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project to which the asset belongs.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.
4. Make sure you are on the **Inventory** tab, then choose the name of the asset that you want to publish. You are then brought to the asset details page.
5. Choose the **Schema** tab and then on the schema details page, choose the **View/Edit** link of the column that you'd like to curate.

   In the right-hand pane that opens, you can edit the details, ReadMe, glossary terms, and metadata forms of the column.

If automated metadata generation is enabled when the data source is created, the business names for assets and columns are available to review and accept or reject individually or all at once.

You can also edit the subscription terms to specify if approval for the asset is required or not.

Metadata forms in Amazon SageMaker Unified Studio enable you to extend a data asset's metadata model by adding custom-defined attributes (for example, sales region, sales year, and sales quarter). The metadata forms that are attached to an asset type are applied to all assets created from that asset type. You can also add additional metadata forms to individual assets as part of the data source run or after it's created. For creating new forms, see the section called “Create a metadata form”.

To update the metadata of an asset, you must be the owner or the contributor of the project to which the asset belongs.

To update the metadata of an asset

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project to which the asset belongs.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.
4. Make sure you are on the **Inventory** tab, then choose the name of the asset that you want to publish. You are then brought to the asset details page.
5. On the asset details page, under **Metadata forms**, choose **Edit values** to edit the existing forms as needed, or choose **Add metadata form** and enter values for each of the metadata fields to attach additional metadata forms to the asset.
6. When you're done making updates, choose **Save**.

   When you save the form, Amazon SageMaker Unified Studio generates a new inventory version of the asset. To publish the updated version to the catalog, choose **Re-publish asset**.

By default, metadata forms attached to a domain are attached to all assets published to that domain. Data publishers can associate additional metadata forms to individual assets in order to provide additional context.

When you are satisfied with the asset curation, the data owner can publish an asset version to the Amazon SageMaker Unified Studio catalog and thus make it discoverable by all domain users. The asset in the project shows the inventory version and the published version. In the discovery catalog, only the latest published version appears. If the metadata is updated after publishing, then a new inventory version will be available for publishing to the catalog.

Manually create an asset in Amazon SageMaker Unified Studio

In Amazon SageMaker Unified Studio, an asset is an entity that presents a single physical data object (for example, a table, a dashboard, a file) or virtual data object (for example, a view). For more information, see Amazon SageMaker Unified Studio terminology and concepts. Publishing an asset manually is a one-time operation. You don't specify a run schedule for the asset, so it's not updated automatically if its source changes.

To manually create an asset through a project, you must be the owner or contributor of that project.

To create an asset manually

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project that you want to create an asset in.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.
4. On the **Inventory** tab, choose **Create**, then choose **Create asset**. choose **Create asset**.
5. For **Data asset details**, configure the following settings:
   - **Name** – The name of the asset.
   - **Description** – A description of the asset.
6. Choose **Next**.
7. For **Asset type details**, configure the following settings:
   - **Asset type** – The type of asset.
   - **Revision**.
8. If you are adding an **S3 object collection**, for **S3 location**, enter the Amazon Resource Name (ARN) of the source S3 bucket.

   Optionally, enter an S3 access point. For more information, see Managing data access with Amazon S3 access points.
9. Choose **Next**.
10. Review the selections, then choose **Create**.

    After the asset is created, it will be stored in the inventory until you decide to publish it.

Unpublish an asset from the Amazon SageMaker Catalog

When you unpublish an Amazon SageMaker Unified Studio asset from the catalog, it no longer appears in global search results. New users won't be able to find or subscribe to the asset listing in the catalog, but all existing subscriptions remain the same.

To unpublish an asset, you must be the owner or the contributor of the project to which the asset belongs.

To unpublish an asset

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project to which the asset belongs.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.
4. On the **Published** tab, choose the name of the asset that you want to unpublish. This opens the asset details page.
5. Expand the **Actions** menu, then choose **Unpublish**.
6. In the pop-up window, confirm the action by choosing **Unpublish**.

   The asset is then removed from the catalog. You can re-publish the asset at any time by choosing **Publish asset**.

Delete an Amazon SageMaker Unified Studio asset

When you no longer need an asset in Amazon SageMaker Unified Studio, you can permanently delete it. Deleting an asset is different than unpublishing an asset from the catalog. You can delete an asset and its related listing in the catalog so that it's not visible in any search results. To delete the asset listing, you must first revoke all of its subscriptions.

To delete an asset, you must be the owner or the contributor of the project to which the asset belongs.

> **Note**
>
> In order to delete an asset listing, you must first revoke all existing subscriptions to the asset, and the asset must be removed from all data products. You can't delete an asset listing that has existing subscribers or that is included in a current data product.

To delete an asset

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project to which the asset belongs.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.
4. On the **Published** tab, choose the name of the asset that you want to unpublish. This opens the asset details page.
5. Expand the **Actions** menu, then choose **Delete**.
6. In the pop-up window, type `delete` to confirm deletion, then choose **Delete**.

   When the asset is deleted, it's no longer available to view or subscribe to.

Manually start a data source run in Amazon SageMaker Unified Studio

When you run a data source, Amazon SageMaker Unified Studio pulls all any new or modified metadata from the source and updates the associated assets in the inventory. When you add a data source to Amazon SageMaker Unified Studio, you specify the source's run preference, which defines whether the source runs on a schedule or on demand. If your source runs on demand, you must initiate a data source run manually.

Even if your source runs on a schedule, you can still run it manually at any time. After adding business metadata to the assets, you can select assets and publish them to the Amazon SageMaker Catalog in order for these assets to be discoverable by all domain users. Only published assets are searchable by other domain users.

To run a data source manually

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project to which the data source belongs.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Data sources**.
4. Choose the data source that you want to run. This opens the data source details page.

5. Choose **Run**.

   The data source status changes as Amazon SageMaker Unified Studio updates the asset metadata with the most recent data from the source. You can monitor the status of the run on the **Data source runs** tab.

Asset revisions in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio increments the revision of an asset when you edit its business or technical metadata. These edits include modifying the asset name, description, glossary terms, column names, metadata forms, and metadata form field values. These changes can result from manual edits, data source job runs, or API operations. Amazon SageMaker Unified Studio automatically generates a new asset revision any time you make an edit to the asset.

After you update an asset and a new revision is generated, you must publish the new revision to the catalog for it to be updated and available to subscribers. For more information, see the section called “Publish assets to the catalog from the project inventory”. You can only publish the most recent version of an asset to the catalog.

To view past revisions of an asset

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project to which the asset belongs.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.
4. On the **Published** tab, choose the name of the asset that you want to unpublish. This opens the asset details page.
5. Navigate to the **History** tab, which displays a list of past revisions of the asset.

Data quality in Amazon SageMaker Unified Studio

Data quality metrics in Amazon SageMaker Unified Studio help you understand the different quality metrics such as completeness, timeliness, and accuracy of your data sources. Amazon SageMaker Unified Studio integrates with AWS Glue Data Quality and offers APIs to integrate data quality metrics from third-party data quality solutions. Data users can see how data quality metrics change over time for their subscribed assets. To author and run the data quality rules, you

can use your data quality tool of choice such as AWS Glue data quality. With data quality metrics in Amazon DataZone, data consumers can visualize the data quality scores for the assets and columns, helping build trust in the data they use for decisions.

Prerequisites and IAM role changes

If you are using Amazon SageMaker Unified Studio's AWS managed policies, there are no additional configuration steps and these managed policies are automatically updated to support data quality. If you are using your own policies for the roles that grant Amazon SageMaker Unified Studio the required permissions to interoperate with supported services, you must update the policies attached to these roles to enable support for reading the AWS Glue data quality information.

Enabling data quality for AWS Glue assets

Amazon SageMaker Unified Studio pulls the data quality metrics from AWS Glue in order to provide context during a point in time, for example, during a business data catalog search. Data users can see how data quality metrics change over time for their subscribed assets. Data producers can ingest AWS Glue data quality scores on a schedule. The Amazon SageMaker Unified Studio business data catalog can also display data quality metrics from third-party systems through data quality APIs. For more information, see AWS Glue Data Quality and Getting started with AWS Glue Data Quality for the Data Catalog.

You can enable data quality metrics for your Amazon SageMaker Unified Studio assets in the following ways:

- Use Amazon SageMaker Unified Studio or the Amazon DataZone APIs to enable data quality for your AWS Glue data source via the Amazon SageMaker Unified Studio either while creating new or editing existing AWS Glue data source.

  > **Note**
  >
  > You can use Amazon SageMaker Unified Studio to enable data quality only for your AWS Glue inventory assets. In this release of Amazon SageMaker Unified Studio, enabling data quality for custom types assets in Amazon SageMaker Unified Studio must be done using APIs.

- You can also use the APIs to enable data quality for your new or existing data sources. You can do this by invoking the CreateDataSource or UpdateDataSource APIs and setting the `autoImportDataQualityResult` parameter to 'True'.

After data quality is enabled, you can run the data source on demand or on schedule. Each run can bring in up to 100 metrics per asset. There is no need to create forms or add metrics manually when using data source for data quality. When the asset is published, the updates that were made to the data quality form (up to 30 data points per rule of history) are reflected in the listing for the consumers. Subsequently, each new addition of metrics to the asset is automatically added to the listing. There is no need to republish the asset to make the latest scores available to consumers.

Enabling data quality for custom asset types

You can use the Amazon SageMaker Unified Studio APIs to enable data quality for any of your custom type assets. For more information, see the following:

- PostTimeSeriesDataPoints
- ListTimeSeriesDataPoints
- GetTimeSeriesDataPoint
- DeleteTimeSeriesDataPoints

The following steps provide an example of using APIs or CLI to import third-party metrics for your assets in Amazon SageMaker Unified Studio:

1. Invoke the `PostTimeSeriesDataPoints` API as follows:

```
aws datazone post-time-series-data-points  \
--cli-input-json file://createTimeSeriesPayload.json \
```

with the following payload:

```
"domainId": "dzd_5oo7xzoqltu8mf",
    "entityId": "4wyh64k2n8czaf",
    "entityType": "ASSET",
    "form": {
        "content": "{\n  \"evaluations\" : [ {\n    \"types\" : [ \"MaximumLength
\" ],\n    \"description\" : \"ColumnLength \\\"ShippingCountry\\\" <= 6\",\n
   \"details\" : { },\n    \"applicableFields\" : [ \"ShippingCountry\" ],\n
   \"status\" : \"PASS\"\n  }, {\n    \"types\" : [ \"MaximumLength\" ],\n
 \"description\" : \"ColumnLength \\\"ShippingState\\\" <= 2\",\n    \"details
```

```
\" : { },\n    \"applicableFields\" : [ \"ShippingState\" ],\n    \"status\" :
 \"PASS\"\n  }, {\n    \"types\" : [ \"MaximumLength\" ],\n    \"description
\" : \"ColumnLength \\\"ShippingCity\\\" <= 8\",\n    \"details\" : { },\n
 \"applicableFields\" : [ \"ShippingCity\" ],\n    \"status\" : \"PASS\"\n  },
 {\n    \"types\" : [ \"Completeness\" ],\n    \"description\" : \"Completeness \
\\"ShippingStreet\\\" >= 0.59\",\n    \"details\" : { },\n    \"applicableFields
\" : [ \"ShippingStreet\" ],\n    \"status\" : \"PASS\"\n  }, {\n    \"types\" :
 [ \"MaximumLength\" ],\n    \"description\" : \"ColumnLength \\\"ShippingStreet\\
\" <= 101\",\n    \"details\" : { },\n    \"applicableFields\" : [ \"ShippingStreet
\" ],\n    \"status\" : \"PASS\"\n  }, {\n    \"types\" : [ \"MaximumLength\" ],\n
    \"description\" : \"ColumnLength \\\"BillingCountry\\\" <= 6\",\n    \"details
\" : { },\n    \"applicableFields\" : [ \"BillingCountry\" ],\n    \"status\" :
 \"PASS\"\n  }, {\n    \"types\" : [ \"Completeness\" ],\n    \"description\" :
 \"Completeness \\\"biLlingcountry\\\" >= 0.5\",\n    \"details\" : {\n
 \"EVALUATION_MESSAGE\" : \"Value: 0.26666666666666666 does not meet the constraint
 requirement!\"\n    },\n    \"applicableFields\" : [ \"biLlingcountry\" ],\n
    \"status\" : \"FAIL\"\n  }, {\n    \"types\" : [ \"Completeness\" ],\n
 \"description\" : \"Completeness \\\"Billingstreet\\\" >= 0.5\",\n    \"details
\" : { },\n    \"applicableFields\" : [ \"Billingstreet\" ],\n    \"status\" :
 \"PASS\"\n  } ],\n  \"passingPercentage\" : 88.0,\n  \"evaluationsCount\" : 8\n}",
        "formName": "shortschemaruleset",
        "id": "athp9dyw75gzhj",
        "timestamp": 1.71700477757E9,
        "typeIdentifier": "amazon.datazone.DataQualityResultFormType",
        "typeRevision": "8"
    },
    "formName": "shortschemaruleset"
}
```

You can obtain this payload by invoking the `GetFormType` action:

```
aws datazone get-form-type --domain-identifier <your_domain_id> --form-type-
identifier amazon.datazone.DataQualityResultFormType --region <domain_region> --
output text --query 'model.smithy'
```

2. Invoke the `DeleteTimeSeriesDataPoints` API as follows:

```
aws datazone delete-time-series-data-points\
--domain-identifier dzd_bqqlk3nz21zp2f \
--entity-identifier dzd_bqqlk3nz21zp2f \
```

```
--entity-type ASSET \
--form-name rulesET1 \
```

Using machine learning and generative AI in Amazon SageMaker Unified Studio

> **Note**
>
> Powered by Amazon Bedrock: AWS implements automated abuse detection. Because the AI recommendations for assets in Amazon SageMaker Unified Studio is built on Amazon Bedrock, users inherit the controls implemented in Amazon Bedrock to enforce safety, security, and the responsible use of AI.

In the current release of Amazon SageMaker Unified Studio, you can use the AI recommendations for names, descriptions, and glossary terms functionality to automate data discovery and cataloging.

Powered by Amazon Bedrock's large language models, the AI recommendations for data asset names, descriptions, and glossary terms in Amazon SageMaker Unified Studio help you to ensure that your data is comprehensible and easily discoverable. The AI recommendations also suggest the most pertinent analytical applications for datasets. By reducing manual documentation tasks and advising on appropriate data usage, auto-generated names and descriptions can help you to enhance the trustworthiness of your data and minimize overlooking valuable data to accelerate informed decision making.

AI recommendations for glossary terms is a feature that automatically analyzes asset metadata and context to determine the most relevant business glossary terms for each asset and its columns. Instead of relying on manual tagging or static rules, it reasons about the data and performs iterative searches across what already exists in the customer’s environment to identify the best-fit glossary term concepts. Because the system suggests terms only from glossaries and definitions already present in the system, customers are encouraged to maintain high-quality, well-described glossary entries so the AI can return accurate and meaningful suggestions. This improves metadata quality, strengthens governance, accelerates data onboarding, and reduces manual stewardship effort at scale.

Supported Regions for the AI recommendations for names and descriptions

In the current Amazon SageMaker Unified Studio release, the AI recommendations for names and descriptions feature is supported in the following regions:

- US East (N. Virginia)
- US West (Oregon)
- Asia Pacific (Tokyo)
- Europe (Frankfurt)
- Asia Pacific (Sydney)
- Canada (Central)
- Europe (London)
- South America (Sao Paulo)
- Europe (Ireland)
- Asia Pacific (Singapore)
- US East (Ohio)
- Asia Pacific (Seoul)

Amazon SageMaker Unified Studio supports Business Description Generation in the following regions.

- Asia Pacific (Mumbai)
- Europe (Paris)

Amazon SageMaker Unified Studio supports Business Name Generation in the following regions.

- Europe (Stockholm)

Bedrock Cross Region Inference

Amazon SageMaker Unified Studio leverages Amazon Bedrock's Cross Region inference endpoint to serve recommendations for the US East (Ohio) region. All other regions use in-region endpoint.

Supported Regions for the AI recommendations for glossary terms

In the current Amazon SageMaker Unified Studio release, the AI recommendations for glossary terms feature is supported in the following regions:

- US East (N. Virginia)
- US West (Oregon)
- Asia Pacific (Tokyo)
- Europe (Frankfurt)
- Asia Pacific (Sydney)
- Europe (London)
- Europe (Ireland)
- Asia Pacific (Singapore)
- US East (Ohio)
- Asia Pacific (Seoul)
- Asia Pacific (Mumbai)
- Europe (Paris)
- Europe (Stockholm)

Bedrock Cross Region Inference

Amazon SageMaker Unified Studio leverages Amazon Bedrock's Cross Region inference endpoint to serve recommendations for all of the supported regions for AI recommendations for glossary terms.

Steps to use GenAI

The following procedure describes how to generate AI recommendations for names, descriptions, and glossary terms in Amazon SageMaker Unified Studio:

- Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
- Choose the project that contains the asset for which you want to generate AI recommendations for descriptions.

Generating Business Descriptions and Summaries

- In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**, and then choose the asset for which you want to generate AI recommendations for descriptions.
- On the asset's details page, in the **Business metadata** tab, choose **Generate descriptions**.

Generating glossary terms

- In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**, and then choose the asset for which you want to generate AI recommendations for glossary terms.
- On the asset's details page, in the **Business metadata** tab, choose **Generate terms**.

Generating Business Names

- In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Data sources**, and then choose the data source for which you want to enable business name generation.
- Go to the **details** tab and enable the **AUTOMATED BUSINESS NAME GENERATION** configuration.
- BusinessNames can also be generated programmatically when creating an asset by enabling the businessNameGeneration flag under predictionConfiguration in the CreateAsset API payload.

Accepting/Rejecting Predictions

- Once the metadata (name, description or terms) suggestions, are generated, you can either edit, accept, or reject them.
- Sparkle icons are displayed next to each automatically generated metadata (name, description or terms), for the data asset. In the **Business metadata** tab, you can choose the sparkle icon next to the automatically generated **Summary**, and then choose **Edit**, **Accept**, or **Reject** to address the generated description.
- You can also choose **Accept all** or **Reject all** options that are displayed at the top of the page when the **Business metadata** tab is selected, and thus perform the selected action on all automatically generated metadata (name, description or terms).

- Or you can choose the **Schema** tab, and then address automatically generated metadata (name, description or terms) individually by choosing the sparkle icon for one suggested metadata change at a time and then choosing **Accept** or **Reject**.
- In the **Schema** tab, you can also choose **Accept all** or **Reject all** and thus perform the selected action on all automatically generated metadata.

To publish the asset to the catalog with the generated descriptions, choose **Publish asset**, and then confirm this action by choosing **Publish asset** again in the **Publish asset** pop up window.

> **Note**
>
> If you don't accept or reject the generated metadata for an asset, and then you publish this asset, this unreviewed automatically generated metadata is not included in the published data asset.

Support for custom relational asset types

Amazon SageMaker Unified Studio supports genAI capabilities for custom asset types. Previously this feature was only supported for the managed AWS Glue and Amazon Redshift asset types.

In order to enable this feature, create your own asset type definition and attach `RelationalTableFormType` as one of the forms. Amazon SageMaker Unified Studio automatically detects the presence of such forms and enables GenAI capabilities for these assets. The overall experience remains the same for generating business names (via predictionConfiguration in the CreateAsset API), business description (via Generate Description button click on the asset details page), and glossary terms.

For more information about creating custom asset types see Create custom asset types in Amazon SageMaker Unified Studio.

Quotas

Amazon SageMaker Unified Studio supports different quotas for business name generation and business description generation. You can reach out to the AWS Support team for an increase in these quotas.

- BusinessDescriptionGeneration: 10K invocations/month

- BusinessNameGeneration: 50K invocations/month
- GlossaryTermGeneration - 10k invocations/month

Analyze Amazon SageMaker Unified Studio data with external analytics applications via JDBC connection

Amazon SageMaker Unified Studio enables data consumers to easily locate and subscribe to data from multiple sources within a single project and analyze this data using Amazon Athena, Amazon Redshift Query Editor, and Amazon SageMaker.

Amazon SageMaker Unified Studio also supports authentication via the Athena JDBC driver that enables users to query their subscribed Amazon SageMaker Unified Studio data using popular external SQL and analytics tools, such as SQL Workbench, DBeaver, Tableau, Domino, Power BI and many others. Users can authenticate using their corporate credentials through SSO or IAM and begin analyzing their subscribed data within their Amazon SageMaker Unified Studio projects.

Amazon SageMaker Unified Studio's support of the Athena JDBC driver provides the following benefits:

- Greater tool choice for querying and visualization - data consumers can connect to Amazon SageMaker Unified Studio using their preferred tools from a wide range of analytics tools that support a JDBC connection. This enables them to continue using the software they are familiar with without the need to learn new tools for data consumption.
- Programmatic access - a JDBC connection to access-governed data via servers or custom applications enables data consumers to perform automated and more complex data operations.

You can use your JDBC URL to connect your external analytics tools to your Amazon SageMaker Unified Studio subscribed data. To obtain your JDBC URL, perform the following procedure:

> **Important**
>
> In the current release, Amazon SageMaker Unified Studio supports authentication using the Amazon Athena JDBC Driver. To complete this procedure, make sure that you have downloaded and installed the latest Athena JDBC driver for your analytics application of choice.

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project where you have the data that you want to analyze.
3. In the **Project overview**, choose the **JDBC connection details** tab.
4. In **JDBC connection details** choose your authentication method (**Using IDC auth** or **Using IAM auth**) and then choose the icon next to **JDBC connection URL** to copy the string or the individual parameters of the JDBC URL. You can then use it to connect to your external analytics application.

When you connect your external analytics application to Amazon DataZone using your JBDC query or parameters, you invoke the `RedeemAccessToken` API. The `RedeemAccessToken` API exchanges an Identity Center access token for the `AmazonDataZoneDomainExecutionRole` credentials, which are used to call the `GetEnvironmentCredentials` API.

For more information about the authentication mechanism that uses IAM credentials to connect to Amazon DataZone-governed data in Athena, see DataZone IAM Credentials Provider. For more information about the authentication mechanism that enables connecting to Amazon DataZone-governed data in Athena using IAM Identity Center, see DataZone Idc Credentials Provider.

RedeemAccessToken API Reference

Request syntax

```
POST /sso/redeem-token HTTP/1.1
Content-type: application/json

{
   "domainId": "string",
   "accessToken": "string"
}
```

Request parameters

The request uses the following parameters.

DomainId

The ID of the Amazon DataZone domain.

Pattern: ^dzd[-_][a-zA-Z0-9_-]{1,36}$

Required: yes

accessToken

The Identity Center access token.

Type: string

Required: yes

Response syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "credentials": AwsCredentials
}
```

Response elements

credentials

The `AmazonDataZoneDomainExecutionRole` credentials that are used to call the `GetEnvironmentCredentials` API.

Type: Array of `AwsCredentials` objects. This data type includes the following properties:

- accessKeyId: AccessKeyId
- secretAccessKey: SecretAccessKey
- sessionToken: SessionToken
- expiration: Timestamp

accessToken

The Identity Center access token.

Type: string

Required: yes

Errors

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 403

ResourceNotFoundException

The specified resource cannot be found.

HTTP Status Code: 404

ValidationException

The input fails to satisfy the constraints specified by the AWS service.

HTTP Status Code: 400

InternalServerException

The request has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

Metadata enforcement rules for publishing

The metadata enforcement rules for publishing in Amazon SageMaker Unified Studio strengthen data governance by enabling domain unit owners to establish clear metadata requirements for data producers, streamlining access requests and enhancing data governance.

The feature is supported in all the AWS commercial Regions where Amazon SageMaker Unified Studio is currently available.

Domain unit owners can can complete the following procedure to configure metadata enforcement in Amazon SageMaker Unified Studio:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, choose **Manage**, then under **Domain management**, choose **Domain units**. Then choose the domain or the domain unit that you want to work with.
3. Choose the **Rules** tab and then choose **Add**.
4. On the **Rule configuration** page, do the following and then choose **Add rule**:

   - Specify a name for your rule.
   - Under **Action**, choose **Data asset and product publishing** or **Subscription request**.
   - If you chose **Subscription request**, then under Required metadata forms, choose **Add metadata form**, choose a metadata form within the domain / domain unit that you want to add to this rule, and then choose **Add**. You can add up to 5 metadata forms per rule.
   - If you chose **Data asset and product publishing**, then under **Rule requirements**, choose either **Metadata forms** or **Glossary association**. If you chose **Metadata forms**, then under Required metadata forms, choose **Add metadata form**, choose a metadata form within the domain / domain unit that you want to add to this rule, and then choose **Add**. You can add up to 5 metadata forms per rule. If you chose **Glossary association**, then choose **Add terms** and add your glossary terms to your rule. You can add up to 5 glossary terms per rule.
   - Under **Scope**, specify with which data entities you want to associate these forms. You can choose data products and/or data assets.
   - Under **Data asset types**, specify whether the rule applies across all asset types or limit it to selected asset types.
   - Under **Projects**, specify whether the required forms will be associated with data products and/or assets published by all projects or only selected projects in this domain unit. Also, check **Cascade rule to child domain units** if you want child domain units to inherit this requirement.

Data discovery, subscription, and consumption

In Amazon SageMaker Unified Studio, after an asset is published to a domain, subscribers can discover and request a subscription to this asset. The subscription process begins with a subscriber searching for and browsing the catalog to find an asset they want. In the Amazon SageMaker

Catalog, they subscribe to the asset by submitting a subscription request that includes justification and the reason for the request. The owner of the asset reviews the request. They can either approve or reject the request.

After a subscription is granted, a fulfillment process starts to facilitate access to the asset for the subscriber. There are two primary modes of asset access control and fulfillment: those for Amazon SageMaker Unified Studio managed assets and those for assets that are not managed by Amazon SageMaker Unified Studio.

- **Managed assets** – Amazon SageMaker Unified Studio can manage fulfillment and permissions for managed assets, such as AWS Glue tables and Amazon Redshift tables and views.
- **Unmanaged assets** – Amazon SageMaker Unified Studio publishes standard events related to your actions (for example, approval given to a subscription request to Amazon EventBridge). You can use these standard events to integrate with other AWS services or third-party solutions for custom integrations.

Topics

- Search for data in the Amazon SageMaker Unified Studio catalog
- Request subscription to assets in Amazon SageMaker Unified Studio
- Approve or reject a subscription request in Amazon SageMaker Unified Studio
- Revoke an existing subscription in Amazon SageMaker Unified Studio
- Cancel a subscription request in Amazon SageMaker Unified Studio
- Unsubscribe from an asset in Amazon SageMaker Unified Studio
- Grant access to managed AWS Glue Data Catalog assets in Amazon SageMaker Unified Studio
- Grant access to managed Amazon Redshift assets in Amazon SageMaker Unified Studio
- Grant access for approved subscriptions to unmanaged assets in Amazon SageMaker Unified Studio
- Metadata enforcement rules for subscription requests

Search for data in the Amazon SageMaker Unified Studio catalog

Amazon SageMaker Unified Studio provides a streamlined way to search for data. Any Amazon SageMaker Unified Studio user with permissions to access Amazon SageMaker Unified Studio can search for resources in the Amazon SageMaker Unified Studio catalog and view resource names and the metadata assigned to them. You can search across multiple resource types including

Amazon S3 files, SageMaker models (both custom and JumpStart), AWS Glue tables and views, and notebooks. You can take a closer look at an asset by examining its details page.

> **Note**
>
> To view the actual data that an asset contains, you must first subscribe to the asset and have your subscription request approved and access granted.

To search for data in the catalog

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, choose **Discover**.
3. Browse through all data you have in your project by entering the name of the resource into the search bar, or selecting one of the filter chips to toggle between different types. You can search across multiple resource types including Amazon S3 files, SageMaker models (both custom and JumpStart), AWS Glue tables and views, and notebooks.
4. If searching for an asset, find the asset that you want to subscribe to by browsing or entering the name of the asset into the search bar. You can apply filters to narrow the results. The filters include asset type, date range, and custom metadata filters. To add a custom metadata filter, choose **Add Filter** at the bottom of the filters panel. You can filter by asset name, description, or metadata form fields.

   For metadata form filters, select the form, field, and operator (`contains` for string fields; `equals`, `greater than`, or `less than` for numeric fields). Enter a value and choose **Apply**. You can combine multiple custom filters.

   Filter selections persist in your browser by using local storage. Only fields that are marked as searchable (strings) or sortable (numerics) are available for filtering.
5. To view details about a specific asset, choose the asset to open its details page. The details page includes the following information:

   - The asset name and type.
   - A description of the asset.
   - The current published revision of the asset, the owner, whether approval is required for subscriptions, and update history.

- A **Business metadata** tab which includes glossary terms and metadata forms.
- A **Subscription requests** tab which includes a list of subscribers to the domain.
- A **Lineage** tab which displays a chart of past revisions of the asset.

Request subscription to assets in Amazon SageMaker Unified Studio

With Amazon SageMaker Unified Studio, you can find, access, and consume the assets in the Amazon SageMaker Unified Studio catalog. When you find an asset in the catalog that you want to access, you need to *subscribe* to the asset, which creates a subscription request. An approver can then approve or reject your request.

You must be a member of a project to request subscription to an asset within that project.

To subscribe to an asset

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, choose **Discover**.
3. Choose the asset filter chip.
4. Find the asset you want to subscribe to by browsing using filters or typing the name of the asset into the search bar.
5. Choose the asset to which you want to subscribe, and then choose **Subscribe**.
6. In the **Subscribe** pop-up window, provide the following information:
   - The project that you want to subscribe to the asset.
   - A short justification for your subscription request.
7. Choose **Request**.

   The project is subscribed to the asset when the publisher approves your request.

To view the status of your subscription request, choose the project that you used to subscribe to the asset. In the left navigation pane, choose **Manage**. Under **Catalog management**, choose **Subscription requests**, then choose the **Outgoing requests** tab. This page lists the assets to which the project has requested access. You can filter the list by the status of the request.

Approve or reject a subscription request in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio allows you to find, access and consume the assets in the Amazon SageMaker Unified Studio catalog. When you find an asset in the catalog that you want to access, you must *subscribe* to the asset, which creates a subscription request. An approver can then approve or reject your request.

You must be a member of the owning project (the project that published the asset) to approve or reject a subscription request.

To approve or reject a subscription request

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project that contains the asset that has a subscription request. You can do this by choosing a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Subscription requests**.
4. Choose the **Incoming requests** tab.
5. Locate the request and choose **View request**. You can filter by **Requested** to see only requests that are still open.
6. Review the subscription request and reason for access, and decide whether to approve or reject it.
7. To approve, select between the two options:

   - **Full access**: If you choose to approve the subscription with full access option, the subscriber will get access to all the rows and columns in your data asset.
   - **Approve with row and column filters**: To limit access to specific rows and columns of data, you can choose the option to approve with row and column filters. For more information, see Fine-grained access control to data.
     - Select **Choose filters**, and then from the drop down select one or more available filters you want to apply to the subscription.
     - To create a new filter you can choose Create new filter option, which opens a new page to create a new row or column filter. For more information, see Create column filters in

Amazon SageMaker Unified Studio and Create row filters in Amazon SageMaker Unified Studio.

8. (Optional) Enter a response that explains your reason for accepting or rejecting the request.
9. Choose either **Approve** or **Reject**.

As the project owner, you can revoke the subscription at any time. For more information, see the section called "Revoke an existing subscription".

> **Note**
>
> Amazon SageMaker Unified Studio supports fine-grained access control for AWS Glue tables, Amazon Redshift tables, and Amazon Redshift views.

Automatic approval of subscription requests

By default, subscription requests to a published asset require manual approval by a data owner. However, Amazon SageMaker Unified Studio supports two scenarios where subscription requests can be automatically approved:

- Approval disabled during asset publishing - when publishing a data asset, you can choose to not require subscription approval. In this case, all incoming subscription requests to that asset are automatically approved. To learn how to disable approval for an asset, see Publish assets to the Amazon SageMaker Unified Studio catalog from the project inventory .
- Requester is an owner or contributor in the project that published the asset - a subscription request is also automatically approved if the requester is already authorized to approve it manually. Specifically, if they are a member of both the project that published the asset and the project requesting access.

  To qualify for auto-approval:

  - The requester must be listed as an owner or contributor in the project where the asset was originally published.
  - The requester must also be listed as an owner or contributor in the project making the subscription request.

This ensures that auto-approval only occurs when the requester has visibility and permissions in both projects — the one sharing the asset and the one requesting access. If the requester meets both conditions, the system auto-approves the request.

Revoke an existing subscription in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio allows you to find, access and consume the assets in the Amazon SageMaker Unified Studio catalog. When you find an asset in the catalog that you want to access, you need to *subscribe* to the asset, which creates a subscription request. An approver can then approve or request your request. You might need to revoke a subscription after you have approved it, either because the approval was a mistake, or because the subscriber no longer needs access to the asset.

You must be a member of the owning project (the project that published the asset) to revoke a subscription.

To revoke a subscription

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project that contains the asset that has a subscription request. You can do this by choosing a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Subscription requests**.
4. Choose the **Incoming requests** tab.
5. Locate the subscription you want to revoke and choose **View subscription**.
6. (Optional) Enable the checkbox to allow the subscriber to keep the asset in the project's subscription targets. A subscription target is a reference to a set of resources where subscribed data can be made available within an environment.

   If you want to revoke access to the asset from the subscription target at a later time, you must do so in AWS Lake Formation.
7. Choose **Revoke subscription**.

You can't re-approve a subscription after you revoke it. The subscriber must request a subscription to the asset again in order for you to approve it.

Cancel a subscription request in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio allows you to find, access and consume the assets in the Amazon SageMaker Unified Studio catalog. When you find an asset in the catalog that you want to access, you need to *subscribe* to the asset, which creates a subscription request. An approver can then approve or request your request. You might need to cancel a pending subscription request, either because you submitted it by mistake, or because you no longer need read access to the asset.

To cancel a subscription request, you must be either a project owner or contributor.

To cancel a subscription request

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project that contains the asset that has a subscription request. You can do this by choosing a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Subscription requests**.
4. Choose the **Outgoing requests** tab.
5. Filter by **Requested** to see only requests that are still pending.
6. Locate the request and choose **View request**.
7. Review the subscription request and choose **Cancel request**.

If you want to re-subscribe to the asset (or to a different asset), see the section called “Request subscription to assets”.

Unsubscribe from an asset in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio allows you to find, access and consume the assets in the Amazon SageMaker Unified Studio catalog. When you find an asset in the catalog that you want to access, you need to *subscribe* to the asset, which creates a subscription request. An approver can then approve or request your request. You might need to unsubscribe from an asset, either because you subscribed by mistake and were approved, or because you no longer need read access to the asset.

You must be a member of a project in order to unsubscribe from one of its assets.

To unsubscribe from an asset

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. Navigate to the project that contains the asset that has a subscription request. You can do this by choosing a project from the project selector dropdown at the top of the page.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Subscription requests**.
4. Choose the **Outgoing requests** tab.
5. Filter by **Approved** to see only requests that have been approved.
6. Locate the request and choose **View subscription**.
7. Review the subscription and choose **Unsubscribe**.

If you want to re-subscribe to the asset (or to a different asset), see the section called “Request subscription to assets”.

Grant access to managed AWS Glue Data Catalog assets in Amazon SageMaker Unified Studio

> **Note**
>
> Access management for the AWS Glue Data Catalog assets using the AWS Lake Formation LF-TBAC method is not supported.
> Support for cross-Region sharing of assets in AWS Glue Data Catalog is not supported.
> Support for cross-account sharing of assets in a federated catalog within AWS Glue Data Catalog is not supported.

When a subscription request to managed AWS Glue Data Catalog assets is approved, Amazon SageMaker Unified Studio grants and manages access to the approved AWS Glue Data Catalog tables on your behalf through AWS Lake Formation. For the subscriber project, assets that are granted appear in the AWS Glue Data Catalog as resources in your account. You can then use Amazon Athena, Amazon Redshift, or Spark to query the tables.

For Amazon SageMaker Unified Studio to be able to grant access to AWS Glue Data Catalog tables, the following conditions must be met.

- The AWS Glue table must be Lake Formation-managed since Amazon SageMaker Unified Studio grants access by managing Lake Formation permissions.
- The IAM role of the project that has published the asset to the Amazon SageMaker Catalog must have the following AWS Lake Formation permissions:
  - `DESCRIBE` and `DESCRIBE GRANTABLE` permissions on the AWS Glue database that contains the published table.
  - `DESCRIBE`, `SELECT`, `DESCRIBE GRANTABLE`, `SELECT GRANTABLE` permissions in Lake Formation on the published table itself.

For more information, see Granting and revoking permissions on catalog resources in the *AWS Lake Formation Developer Guide*.

Grant access to managed Amazon Redshift assets in Amazon SageMaker Unified Studio

When a subscription to an Amazon Redshift table or view is approved, Amazon SageMaker Unified Studio can automatically add the subscribed asset to the Amazon Redshift Serverless workgroup created for the project, so that members of the project can query the data using the Amazon Redshift query editor link within the project. Under the hood, Amazon SageMaker Unified Studio creates the necessary grants and datashares.

The process of granting access varies depending on where the source database (publisher) and the target database (subscriber) are located.

- Same cluster, same database - if data must be shared within the same database, Amazon SageMaker Unified Studio grants permissions directly on the source table.
- Same cluster, different database - if data must be shared across two databases within the same cluster, Amazon SageMaker Unified Studio creates a view in the target database and permissions are granted on the created view.
- Same account different cluster - Amazon SageMaker Unified Studio creates a datashare between the source and target cluster and creates a view on top of the shared table. Permissions are granted on the view.
- Cross-account - same as above but an additional step is required to authorize cross-account datashare on the producer cluster side and another step to associate the data share on consumer cluster side.

Make sure that your publishing and subscribing Amazon Redshift clusters meet all requirements for Amazon Redshift datashares. For more information, see Data sharing in Amazon Redshift in the Amazon Redshift Developer Guide.

> **Note**
>
> Cross-Region data sharing using Amazon Redshift is not supported.

Grant access for approved subscriptions to unmanaged assets in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio enables users to publish any type of asset in the Amazon SageMaker Catalog. For some of these assets, Amazon SageMaker Unified Studio can automatically manage access grants. These assets are called **managed assets** and include Lake Formation-managed AWS Glue Data Catalog tables, AWS Glue Data Catalog views (Multi Dialect Views), and Amazon Redshift tables and views. All other assets, such as Amazon Athena views, are **unmanaged** and Amazon SageMaker Unified Studio can't automatically grant subscriptions for them.

Amazon SageMaker Unified Studio provides a path for you to manage access grants for your unmanaged assets. When a subscription to an asset in the Amazon SageMaker Catalog is approved by the data owner, Amazon SageMaker Unified Studio publishes an event in Amazon EventBridge in your account along with all the necessary information in the payload that enables you to create the access grants between the source and the target. When you receive this event, you can trigger a custom handler which can use the information in the event to create necessary grants or permissions. After you have granted the access, you can report back and update the status of the subscription in Amazon SageMaker Unified Studio so that it can notify the user(s) who subscribed to the asset that they can start consuming the asset.

Set up Cross-Region Subscriptions

Cross-region subscriptions allow data consumers to subscribe to and access data assets published in a different AWS Region than their consuming project or environment.

With cross-region subscriptions, you can:

- Subscribe to data published in a different Region than your consuming environment
- Extend existing approved subscriptions to another Region

For AWS Glue assets, cross-region access is achieved through resource links. The original table remains in the source Region, and a resource link is created in the target Region for consumer access.

For Amazon Redshift assets, cross-region data sharing uses Redshift's native datashare functionality. For cross-account scenarios, AWS Resource Access Manager (AWS RAM) authorization is required.

Supported assets and Regions

Cross-region subscriptions support AWS Glue tables, AWS Glue views, Amazon Redshift tables, and Amazon Redshift views across all standard (non-opt-in) AWS Regions. Cross-region subscriptions to opt-in Regions are not supported.

Prerequisites

Before you enable cross-region subscriptions, you must have the following:

- An existing Amazon DataZone or SageMaker Unified Studio domain
- Permissions to manage blueprints, environments, and projects in your domain
- For Glue assets: The appropriate data lake blueprint enabled in both source and target Regions
- For Redshift assets: The appropriate data warehouse blueprint enabled in both source and target Regions

Enabling cross-region subscriptions (DataZone domains - V1)

Complete the following steps to enable cross-region subscriptions in DataZone domains.

Step 1: Enable the blueprint in the target Region

1. Open the Amazon DataZone console at https://console.aws.amazon.com/datazone/.
2. Choose your domain.
3. In the navigation pane, choose **Blueprints**.
4. Choose the appropriate blueprint:
    - For Glue assets, choose **DataLake**
    - For Redshift assets, choose **DataWarehouse**

5. If the blueprint is disabled, choose **Enable**.

Step 2: Create an environment profile

1. Sign in to the Amazon DataZone data portal.
2. Navigate to the subscriber project.
3. Choose **Create environment profile**.
4. For **Region**, select the Region that you enabled in Step 1.
5. Configure other settings as needed, and then choose **Create**.

Step 3: Create an environment

1. In the subscriber project, choose **Environments**.
2. Choose **Create environment**.
3. For **Environment profile**, select the environment profile that you created in Step 2.
4. Configure other settings as needed, and then choose **Create**.

Step 4: Subscribe to assets

1. Navigate to the data catalog and find the asset that you want to subscribe to.
2. Choose **Subscribe**.
3. Select the subscriber project with the cross-region environment.
4. Complete the subscription request.

The subscription automatically fulfills to the new Region. You can query the data from the new Region environment.

Enabling cross-region subscriptions (SageMaker Unified Studio domains - V2)

Complete the following steps to enable cross-region subscriptions in SageMaker Unified Studio domains.

Step 1: Enable the blueprints in the target Region

1. Open the SMUS portal.

2. Choose your domain.
3. In the navigation pane, choose **Blueprints**.
4. Enable the **Tooling** blueprint in the target Region. This is required for both Glue and Redshift assets.
5. Enable the appropriate asset blueprint in the target Region:
   - For Glue assets, choose **LakeHouseDatabase**
   - For Redshift assets, choose **RedshiftServerless**
   - **Tooling** (required)
6. Add the target Regions to each blueprint.

Step 2: Create a project profile

1. In the navigation pane, choose **Project profiles**.
2. Choose **Create project profile**.
3. For **Region**, select the Region that you enabled in Step 1.
4. Configure other settings as needed, and then choose **Create**.

Step 3: Create a project

1. On SMUS, Choose **Create project**.
2. For **Project profile**, select the project profile that you created in Step 2.
3. Configure other settings as needed, and then choose **Create**.

The project is provisioned in the target Region. Subscriptions to this project automatically fulfill to the target Region.

Considerations

When working with cross-region subscriptions, keep the following in mind:

- **Region restrictions** – Cross-region subscriptions are not supported in opt-in Regions.
- **Blueprint requirements** – Blueprints must be enabled in both the source and target Regions before you can create cross-region subscriptions.

- **Environment requirements (V1)** – Environments must exist in the target Region before subscriptions can be fulfilled to that Region.
- **Project requirements (V2)** – In SageMaker Unified Studio domains, you cannot add new environments to existing projects through the console. To subscribe to assets in a new Region, you must create a new project with a project profile configured for that Region.
- **Tooling blueprint (V2)** – The Tooling blueprint must be enabled in the target Region before enabling LakeHouseDatabase or RedshiftServerless blueprints.
- **Cross-account Redshift sharing** – For cross-account Redshift data sharing, AWS RAM authorization is required on both the producer and consumer sides.

Metadata enforcement rules for subscription requests

The metadata enforcement rules for subscription requests in Amazon SageMaker Unified Studio strengthen data governance by enabling domain unit owners to establish clear metadata requirements for data consumers, streamlining access requests and enhancing data governance.

The feature is supported in all the AWS commercial Regions where Amazon SageMaker Unified Studio is currently available.

Domain unit owners can can complete the following procedure to configure metadata enforcement for subscription requests in Amazon SageMaker Unified Studio:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. In the left navigation pane, choose **Manage**, then under **Domain management**, choose **Domain units**. Then choose the domain or the domain unit that you want to work with.
3. Choose the **Rules** tab and then choose **Add**.
4. On the **Create required metadata form rule** page, do the following and then choose **Add rule**:
    - Specify a name for your rule.
    - Under **Action**, choose **Subscription request**.
    - Under **Required forms**, choose **Add metadata form**, choose a metadata form within the domain / domain unit that you want to add to this rule, and then choose **Add**. You can add up to 5 metadata forms per rule.
    - Under **Scope**, specify with which data entities you want to associate these forms. You can choose data products and/or data assets.

- Under **Data asset types**, specify whether the rule applies across all asset types or limit it to selected asset types.
- Under **Projects**, specify whether the required forms will be associated with data products and/or assets published by all projects or only selected projects in this domain unit. Also, check **Cascade rule to child domain units** if you want child domain units to inherit this requirement.

Fine-grained access control to data

In the current release of Amazon SageMaker Unified Studio, fine-grained access control of your data is supported so you can have granular access control over your sensitive data. You can control which project can access specific records of data within your data assets published to the Amazon SageMaker Unified Studio business data catalog. Amazon SageMaker Unified Studio supports row and column filters to implement fine-grained access control.

Use **row filters** to restrict access to specific rows based on the criteria you define. For example, if your table contains data for two regions (America and Europe) and you want to ensure that employees in Europe can only access data relevant to their region, you can create a row filter that includes rows where the region is Europe (`region = 'Europe'`). This way, employees in Europe won't have access to America's data.

Use **column filters** to limit access to specific columns within your data assets. For example, if your table includes sensitive information such as Personally Identifiable Information (PII), you can create a column filter to exclude PII columns. This ensures that subscribers can only access non-sensitive data.

To utilize fine-grained access control, you can create row and column filters for your AWS Glue and Amazon Redshift assets in Amazon SageMaker Unified Studio. When you receive a subscription request to access your data assets, you can approve it by applying the appropriate row and column filters. Amazon SageMaker Unified Studio ensures that the subscriber can only access the rows and columns permitted by the filters you applied at the time of subscription approval.

Topics

- Limitations
- Create row filters in Amazon SageMaker Unified Studio
- Create column filters in Amazon SageMaker Unified Studio
- Delete row or column filters in Amazon SageMaker Unified Studio

- Edit row or column filters in Amazon SageMaker Unified Studio
- Grant access with filters in Amazon SageMaker Unified Studio

Limitations

When configuring row or column level filters for fine-grained access control, filtering on columns whose name contains special characters impacts which compute types can access the data.

- In cases where the column name contains special characters, adding an Asset Filter will automatically add double quotes " " around the column name to escape the special characters.

  As a result, the asset is not accessible by data processing compute engines such as EMR-EC2, EMR-Serverless, or Glue-ETL. This asset is still accessible by other compute engines.

  To remove this limitation, either remove the filters on the column names containing special characters or rename the column to remove the special characters and recreate the filter.

Create row filters in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio allows you to create row filters that you can use when approving subscriptions to make sure that the subscriber can only access rows of data as defined in the row filters. To create a row filter, follow the steps below:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project to which the asset belongs.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.
4. Make sure you are on the **Inventory** tab, then choose the name of the asset that you want to create a column filter for. You can add column filters if your data asset in Amazon SageMaker Unified Studio is of type AWS Glue table, Amazon Redshift table, or Amazon Redshift view. You are then brought to the asset details page.
5. On the asset detail page, go to the **Asset filters** tab and then choose **Add asset filter**.
6. Configure the following fields:

   - **Name** - the name of the filter

- **Description** – the description of the filters

7. Under filter type, choose **Row filter**.
8. Under row filter expression, provide one or more expressions for row filter.
   - Choose a column from the **Column** dropdown.
   - Choose an operator from the **Operator** dropdown.
   - Enter a value in the **Value** field.
9. To add another condition to your filter expression, choose **Add condition**.
10. When using multiple conditions in the row filter expression, choose **And** or **Or** to link the conditions.
11. Select an option to indicate whether or not the filter contains sensitive values that you want to hide from approved subscribers.
12. Choose **Create asset filter**.

Create column filters in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio enables you to create column filters that you can use when approving subscriptions to make sure that the subscriber can only access columns of data as defined in the column filters. To create a column filter, follow the steps below:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project to which the asset belongs.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.
4. Make sure you are on the **Inventory** tab, then choose the name of the asset that you want to create a column filter for. You can add column filters if your data asset in Amazon SageMaker Unified Studio is of type AWS Glue table, Amazon Redshift table, or Amazon Redshift view. You are then brought to the asset details page.
5. On the asset detail page, go to the **Asset filters** tab and then choose **Add asset filter**.
6. Configure the following fields:
   - **Name** – the name of the filter
   - **Description** – the description of the filters

7. Under filter type, choose **Column**.
8. Select the columns you want to include in the filters using the check boxes for the columns in the data asset.
9. Choose **Create asset filter**.

Delete row or column filters in Amazon SageMaker Unified Studio

To delete a row or a column filter, follow the steps below:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project to which the asset belongs.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.
4. Make sure you are on the **Inventory** tab, then choose the name of the asset where you want to delete a row or a column filter.
5. On the asset details page, go to the **Asset filters** tab and then choose the name of the filter that you want to delete.
6. Choose **Actions**, **Delete** and then confirm the deletion.

> **Note**
>
> You can delete a filter only if it is not being used in active subscriptions.

Edit row or column filters in Amazon SageMaker Unified Studio

To edit a row or a column filter, follow the steps below:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project to which the asset belongs.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.

4. Make sure you are on the **Inventory** tab, then choose the name of the asset that contains the filter that you want to edit.
5. On the asset detail page, go to **Asset filters** tab and then choose the name of the filter that you want to edit.
6. You can edit the following fields:
   - **Name** – the name of the filter
   - **Description** – the description of the filters
7. If you're editing a row filter, you can update the row filter expression.
8. If you're editing a column filter, you can add or remove the columns selected in the filter.
9. After you have made the changes, choose **Edit asset filter**.

> **Note**
>
> If you edit a filter that is being used in active subscriptions, Amazon SageMaker Unified Studio will automatically update the permissions granted to the subscriber projects. This means that the subscribers will only be able to access the rows or columns as defined in the updated filter, ensuring that your data access policies are consistently enforced.

Grant access with filters in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio enables fine-grained access control by translating the defined row and column filters into appropriate grants for AWS Lake Formation and Amazon Redshift. Below is an explanation of how Amazon SageMaker Unified Studio materializes these filters for both AWS Glue tables and Amazon Redshift.

AWS Glue tables

When a subscription to an AWS Glue table with row and/or column filters is approved, Amazon SageMaker Unified Studio materializes the subscription by creating grants in AWS Lake Formation with Data Cell Filters, ensuring that the members of the subscriber project are only able to access the rows and columns they are allowed to access based on the filters applied to the subscription.

Amazon SageMaker Unified Studio first translates the row and columns filters applied in Amazon SageMaker Unified Studio to AWS Lake Formation Data Cell Filters. If multiple row and columns

filters are used, Amazon SageMaker Unified Studio unions all the columns and all the row filter conditions to compute effective permissions at both row and column level. Amazon SageMaker Unified Studio then creates a single AWS Lake Formation data cell filter using effective row and column permissions.

After the data cell filter is created, Amazon SageMaker Unified Studio shares the subscribed table with the subscriber project by creating read-only (SELECT) permissions in AWS Lake Formation using this data cell filter.

Amazon Redshift

When a subscription to an Amazon Redshift table/view with row and/or column filters is approved, Amazon SageMaker Unified Studio materializes the subscription by creating scoped-down late binding views in Amazon Redshift, ensuring that the members of the subscriber project are only able to access the rows and columns they are allowed to access based on the row and column filters applied to the subscription.

Amazon SageMaker Unified Studio first translates the row and columns filters applied to a subscription in Amazon SageMaker Unified Studio to an Amazon Redshift late binding view. If multiple row and columns filters are used, Amazon SageMaker Unified Studio unions all the columns and all the row filter conditions from to compute effective permissions at both row and column level. Amazon SageMaker Unified Studio then creates the late binding view using effective row and column permissions.

After the late binding view is created, Amazon SageMaker Unified Studio shares this view with the members of subscriber project by creating read-only (SELECT) permissions in Amazon Redshift.

Exporting asset metadata

In the current release of Amazon SageMaker Unified Studio, you can export asset metadata as an Apache Iceberg table through Amazon S3 Tables. This allows data teams to query catalog inventory and answer questions - like the following: How many assets were registered last month?, Which assets are classified as confidential?, or Which assets lack business descriptions?, etc. using standard SQL without building custom ETL infrastructure for reporting.

This capability automatically converts catalog asset metadata into a queryable table accessible from Amazon Athena, Amazon SageMaker Unified Studio notebooks, AI agents, and other analytics and BI tools. The exported table includes technical metadata (such as resource_id, resource_type), business metadata (such as asset_name, business_description), ownership details, and timestamps.

Data is partitioned by snapshot_date for query performance and automatically appears in Amazon SageMaker Unified Studio under the aws-sagemaker-catalog bucket.

Note

You can enable asset metadata export for only one domain per AWS account per Region. To enable export for a different domain, complete the following steps:

1. Delete the export configuration for the currently enabled domain using the DeleteDataExportConfiguration operation.
2. Delete the asset S3 table under the aws-sagemaker-catalog S3 table bucket. We recommend backing up the S3 table before deletion.
3. Call the PutDataExportConfiguration API to enable export for the new domain.

Also, encryption configuration for the exported asset table cannot be updated.

This capability is available in all AWS Regions where Amazon SageMaker Catalog is supported at no additional charge. You pay only for underlying services including S3 Tables storage and Amazon Athena queries. You can control storage costs by setting retention policies on S3 tables to automatically remove records older than your specified period. For more information, see https://docs.aws.amazon.com/AmazonS3/latest/userguide/s3-tables-record-expiration.html.

Requirements

Before you export asset metadata from Amazon SageMaker Unified Studio, make sure that you meet the following requirements:

IAM permissions

The IAM user or role that runs the export commands must have permissions to configure asset metadata export in Amazon DataZone and to create the Amazon S3 Tables resources used to store the exported metadata.

The following example IAM policy shows the required permissions:

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "datazone:GetDataExportConfiguration",
                "datazone:PutDataExportConfiguration"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3tables:CreateTableBucket",
                "s3tables:PutTableBucketPolicy"
            ],
             "Resource": "arn:aws:s3tables:<REGION>:<ACCOUNT_ID>:bucket/aws-sagemaker-
catalog"
        }
    ]
}
```

These permissions allow you to retrieve and update the asset metadata export configuration. They also allow Amazon DataZone to create the Amazon SageMaker Catalog S3 table bucket and apply the required bucket policy on your behalf, enabling DataZone to write the exported metadata.

For KMS related permission requirements, see KMS permissions for exporting asset metadata in Amazon SageMaker Unified Studio.

Start exporting asset metadata

To get started, activate dataset export by invoking the `PutDataExportConfiguration` API action. In response, the DataZone service creates an S3 table bucket named aws-sagemaker-catalog with an asset_metadata namespace and an empty asset table. It also schedules a daily job to export updated asset data. Within 24 hours, you can access the asset table through the S3 Tables console or the Data tab in Amazon SageMaker Unified Studio. You can query the data using Amazon Athena or Studio notebooks, or connect external BI tools through the S3 Tables Iceberg REST Catalog endpoint.

Asset metadata is exported once a day around midnight local time per AWS region.

Enable data export:

```
aws datazone put-data-export-configuration  --domain-identifier dzd-440699i00ezy21 --
region us-east-2 --enable-export
```

With KMS key encryption configuration:

```
aws datazone put-data-export-configuration --encryption-configuration
 kmsKeyArn=arn:aws:kms:us-east-2:651673343886:key/292fedfe-
c9h6-40fa-961b-87393584195c,sseAlgorithm=aws:kms --enable-export --region us-east-2 --
domain-identifier dzd-440699i00ezy21
```

For more information, see the API reference documentation.

Asset table schema

The asset metadata is exported to the following table structure:

Column name	Data type	Comment
snapshot_time	timestamp	Timestamp when this metadata snapshot was captured from AWS Sagemaker. Partition key for temporal queries. Use for point-in-time analysis and time-travel queries (e.g., "show catalog state 30 days ago").
asset_id	string	Unique identifier assigned by AWS Sagemaker Catalog for this asset. Primary key for asset lookups.
resource_type_enum	string	Type of data resource cataloged. Values: GlueTable, RedshiftTable, S3Collection etc. Used for filtering by resource category.
resource_id	string	Platform-native unique identifier for the resource. For AWS: ARN format (e.g., arn:aws:glue:region:account :table/db/table). Use for cross-referencing with source systems.

Column name	Data type	Comment
account_id	string	Cloud account or organizational identifier. For AWS: 12-digit account ID. Used for multi-account queries and cost allocation.
region	string	Geographic region or availability zone where resource is hosted. For AWS: us-east-1, eu-west-1, etc. NULL for region-agnostic resources. Use for regional analysis and compliance queries.
catalog	string	Top-level namespace identifier. Meaning varies by resource_type: For GLUE_TABLE: AWS account ID. For REDSHIFT_TABLE: database name. For S3_COLLECTION: AWS account ID. Use for catalog-l evel grouping.
namespace	string	Mid-level namespace identifier. Meaning varies by resource_type: For GLUE_TABLE: database name. For REDSHIFT_TABLE: schema name. For S3_COLLEC TION: bucket name. May contain multiple levels (e.g., database.schema).
asset_name	string	Business friendly name for this asset. Optional business identifier that can differ from the technical resource_name. NULL if no custom name was provided during asset registration. Examples: 'Customer Master Dataset', 'Q4 Sales Report', 'Production Orders Table'. Use for user-friendly display and business-oriented searches.
resource_name	string	Leaf-level resource identifier. For tables: table name. For S3: prefix path. For views: view name. Use for resource-level filtering.

Column name	Data type	Comment
resource_description	string	Technical description from source system. For Glue: table comment field. For Redshift: table remarks. NULL if source system has no description. Use for technical documentation searches.
business_description	string	Business-friendly description provided to AWS Sagemaker Catalog. Explains purpose, usage, and business context of the asset. NULL if not provided by data owner. Use for business glossary and discovery queries.
extended_metadata	map<string,string>	Flexible key-value store for platform-specific attributes not covered by standard columns. Examples: s3_location, compression_type, column_count, partition_keys, namespace_name, cluster_name. Use for advanced filtering and platform-specific queries. Schema varies by platform and resource_type.
asset_created_time	timestamp	Timestamp when this asset was first created in AWS Sagemaker Catalog.
asset_updated_time	timestamp	Timestamp when this asset was last updated in AWS Sagemaker Catalog.

Querying asset tables

To query Amazon S3 tables using Amazon SageMaker Unified Studio or Amazon Athena you must first do the following:

- Enable analytic services integration by following https://docs.aws.amazon.com/lake-formation/latest/dg/enable-s3-tables-catalog-integration.html
- Grant the query role permission in Lake Formation by following https://docs.aws.amazon.com/lake-formation/latest/dg/granting-table-permissions.html

  Example CLI command:

```
aws lakeformation grant-permissions \
        –principal DataLakePrincipalIdentifier=arn:aws:iam::123456789012:role/
datazone_usr_role_3guzb15tfpk015_agjdowt5f47xgp \
        --resource '{"Table": {"CatalogId": "123456789012:s3tablescatalog/aws-
sagemaker-catalog", "DatabaseName": "asset_metadata", "Name": "asset"}}' \
        --permissions DESCRIBE SELECT –region us-east-2
```

Assets registered in last one month

- Query with sample aggregates

```
SELECT
    DATE(asset_created_time) as date,
    resource_type_enum,
    COUNT(*) as count
FROM asset_metadata.asset
WHERE DATE(snapshot_time) = CURRENT_DATE
    AND asset_created_time >= DATE_ADD('month', -1, CURRENT_DATE)
GROUP BY DATE(asset_created_time), resource_type_enum
ORDER BY date DESC;
```

- Plain query without aggregates and groupBy

```
SELECT *
FROM asset_metadata.asset
WHERE DATE(snapshot_time) = CURRENT_DATE
    AND asset_created_time >= DATE_ADD('month', -1, CURRENT_DATE)
```

Assets without business description or owningEntityId in them

```
SELECT
    asset_id,
    asset_name,
    resource_name,
    resource_type_enum,
    account_id,
    business_description,
    extended_metadata['owningEntityId'] as owner
FROM asset_metadata.asset
```

```
WHERE DATE(snapshot_time) = CURRENT_DATE
    AND (business_description IS NULL
         OR extended_metadata['owningEntityId'] IS NULL);
```

Query asset matching metadata form field values

```
SELECT *
FROM asset_metadata.asset
WHERE DATE(snapshot_time) = CURRENT_DATE
    AND extended_metadata['<metadata-form-name>.<field-name>'] = '<field-value>';
```

Asset distribution queries

- Get distributions by account

```
SELECT
    account_id,
    resource_type_enum,
    COUNT(*) as count
FROM asset_metadata.asset
WHERE DATE(snapshot_time) = CURRENT_DATE
GROUP BY account_id, resource_type_enum
ORDER BY count DESC
```

- Get distribution by asset owner (projectIds)

```
SELECT
    extended_metadata['owningEntityId'] as owner,
    COUNT(*) as count
FROM asset_metadata.asset
WHERE DATE(snapshot_time) = CURRENT_DATE
    AND extended_metadata['owningEntityId'] IS NOT NULL
GROUP BY extended_metadata['owningEntityId']
ORDER BY count DESC;
```

Time travel capabilities

The asset_metadata.asset table captures daily snapshots of asset metadata, allowing us to view the state of data catalog at any point in time. Simply change the date filter in our query to travel back to any previous snapshot

> **Note**
>
> Querying without a snapshot_time filter will read all historical snapshots, resulting in duplicate records and slower performance. Always filter by the desired date or current timestamp.

- View Current Assets snapshot

```
SELECT *
FROM asset_metadata.asset
WHERE DATE(snapshot_time) = CURRENT_DATE;
```

- Travel to a Specific Date ex: Nov-26-2025

```
SELECT *
FROM asset_metadata.asset
WHERE DATE(snapshot_time) = DATE('2025-11-26');
```

- Travel Back Relative to Today ex: travel back by 2 days

```
SELECT *
FROM asset_metadata.asset
WHERE DATE(snapshot_time) = date_add('day', -2, CURRENT_DATE);
```

Common use cases

1. Track Metadata Improvements see which assets gained descriptions or ownership over time:

```
SELECT
    t.asset_id,
    t.resource_name,
    p.business_description as description_before,
    t.business_description as description_now
FROM asset_metadata.asset t
JOIN asset_metadata.asset p ON t.asset_id = p.asset_id
WHERE DATE(t.snapshot_time) = CURRENT_DATE
    AND DATE(p.snapshot_time) = CURRENT_DATE - INTERVAL '7' DAY
    AND p.business_description IS NULL
```

```
    AND t.business_description IS NOT NULL;
```

2. Monitor Asset Growth View how data catalog has grown over the last 30 days:

```
SELECT
    DATE(snapshot_time) as date,
    COUNT(*) as total_assets
FROM asset_metadata.asset
WHERE DATE(snapshot_time) >= CURRENT_DATE - INTERVAL '30' DAY
GROUP BY DATE(snapshot_time)
ORDER BY date DESC;
```

3. Audit Historical Changes to investigate what an asset looked like at a specific point in time:

```
SELECT
    asset_id,
    resource_name,
    business_description,
    extended_metadata['owningEntityId'] as owner,
    snapshot_time
FROM asset_metadata.asset
WHERE asset_id = 'your-asset-id'
    AND DATE(snapshot_time) = DATE('2025-11-26');
```

Working with data and catalog in IAM-based domains

Amazon SageMaker Unified Studio integrates with AWS Glue Data Catalog to surface your databases, tables, and Amazon S3 buckets in the **Data** tab of your project. From there, you can browse your Glue catalog hierarchy, view table schemas, preview data, and upload files to create new tables.

On top of this, Amazon SageMaker Unified Studio adds a business catalog layer that lets you add business context to your tables, such as descriptions, documentation, and standardized business terms, so that others across the organization can understand what the data represents and whether it's relevant to their work. Your Glue tables are automatically discoverable by everyone in the domain. Anyone can search for tables by name, description, or business terms, browse tables organized by database, and view detailed pages that include both technical schema and business context. When someone finds a table they need, they can request access, and once approved, they receive the necessary Lake Formation permissions automatically.

Through the admin portal, you can enrich tables with business context to help others understand and find the right data. This includes editing business names and descriptions at the table and column level, writing README documentation, tagging tables and columns with business glossary terms, and attaching structured metadata forms. You can also use AI to automatically generate business names, descriptions, and summaries for tables and columns.

The admin portal is also where you set up the governance framework for the catalog. You can create business glossaries to standardize terminology across the organization, define metadata form templates to capture structured information about tables, and manage access by approving or rejecting access requests and proactively sharing tables with specific projects.

Topics

- Browse and explore your data
- Search and discover data
- Request access to data
- Manage and govern the catalog

Browse and explore your data

The **Data** tab in your project is where you browse and explore the data you have access to. You can navigate your AWS Glue Data Catalog hierarchy, view table schemas and metadata, preview actual data, and work with Amazon S3 buckets.

The data explorer includes a **Catalogs** view and an **S3 buckets** view. The **Catalogs** view shows your AWS Glue Data Catalog assets organized as a tree of catalogs, databases, and tables. You can expand any level of the tree to drill down, and selecting a table opens a detail panel with tabs for viewing different aspects of the table. The **S3 buckets** view lets you navigate your Amazon S3 buckets, browse folder structures, and view file properties.

To open the data explorer

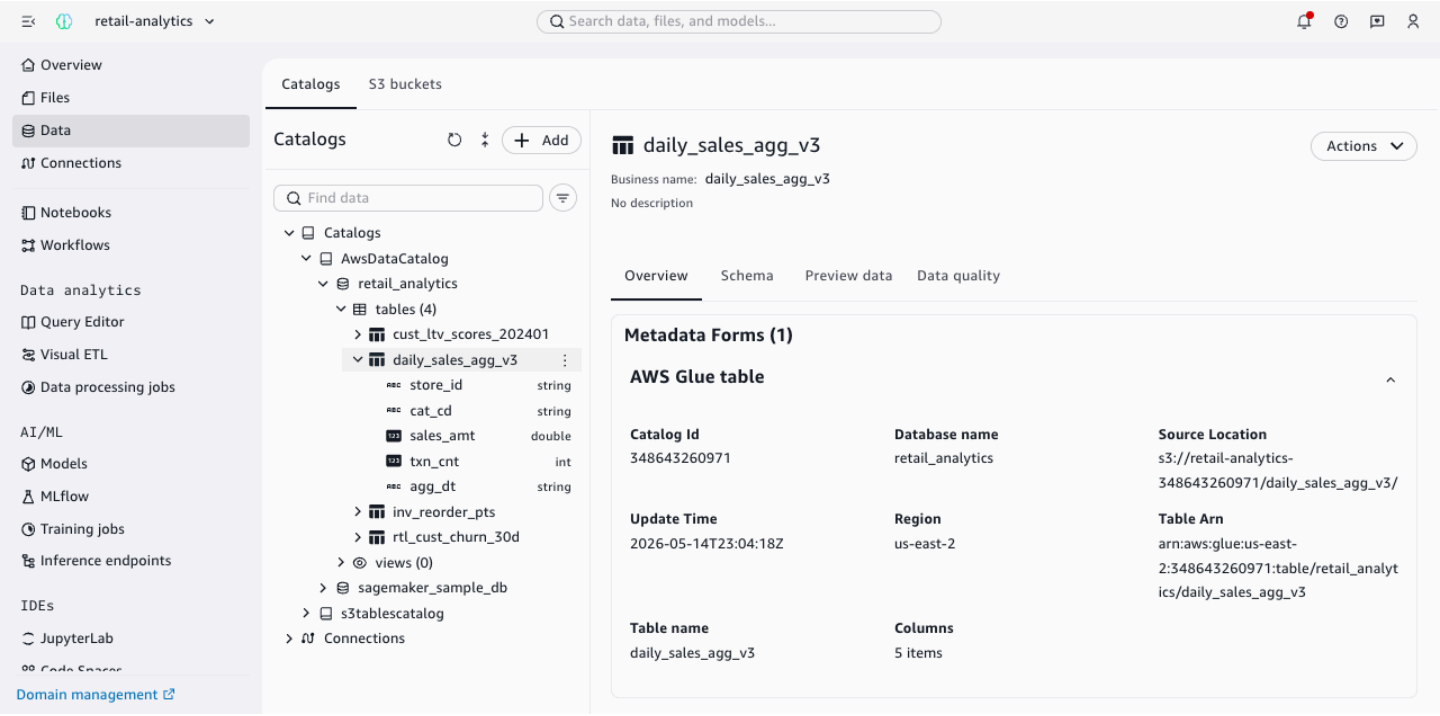
1. Open your project in Amazon SageMaker Unified Studio.
2. In the left navigation pane, choose **Data**.

The data explorer opens with tabs at the top, including **Catalogs** and **S3 buckets**.

Browse catalog assets

The navigation panel shows your catalog hierarchy as a tree that you can expand to drill down from catalogs to databases, tables, and columns. Use the search bar in the navigation panel to find specific items by name.

When you choose a table in the tree, the detail view opens. The detail view shows the table's business name (if one has been set), description, and a set of tabs for exploring different aspects of the table.

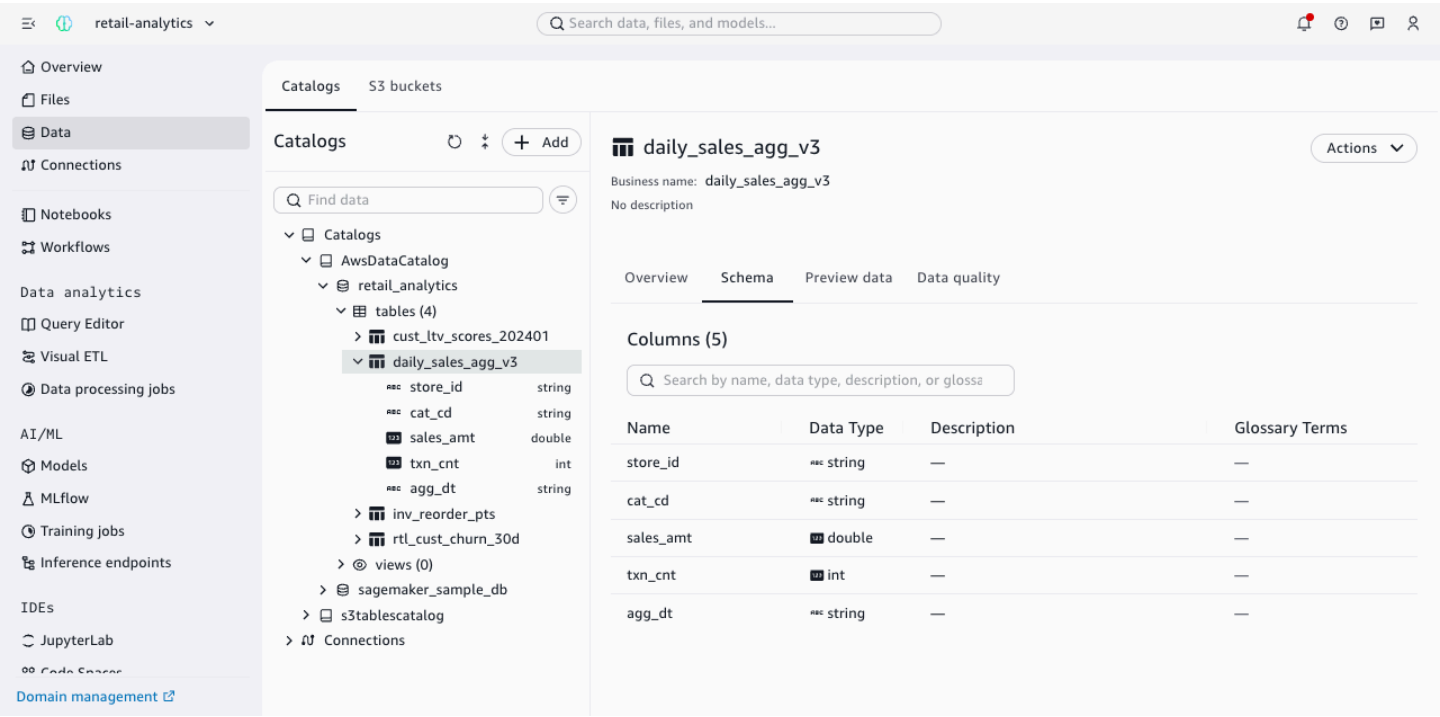


The detail view has the following tabs.

Data explorer detail view tabs

Tab	When to use it
Overview	Start here to understand what a table is and what it contains. Shows the table's description, glossary terms, README documentation, metadata forms (such as the AWS Glue table form with catalog ID, database name, source location, region, and column count), and key details like the database name, region, and S3 location.
Schema	Use this when you need to understand the structure of the data before writing a query or building a pipeline. Shows each column's name, data type, description, and glossary terms.
Preview data	Use this to quickly check what the actual data looks like without writing a query. Shows a sample of rows from the table.
Data quality	Use this to view data quality rules and results for the table, if data quality has been configured.

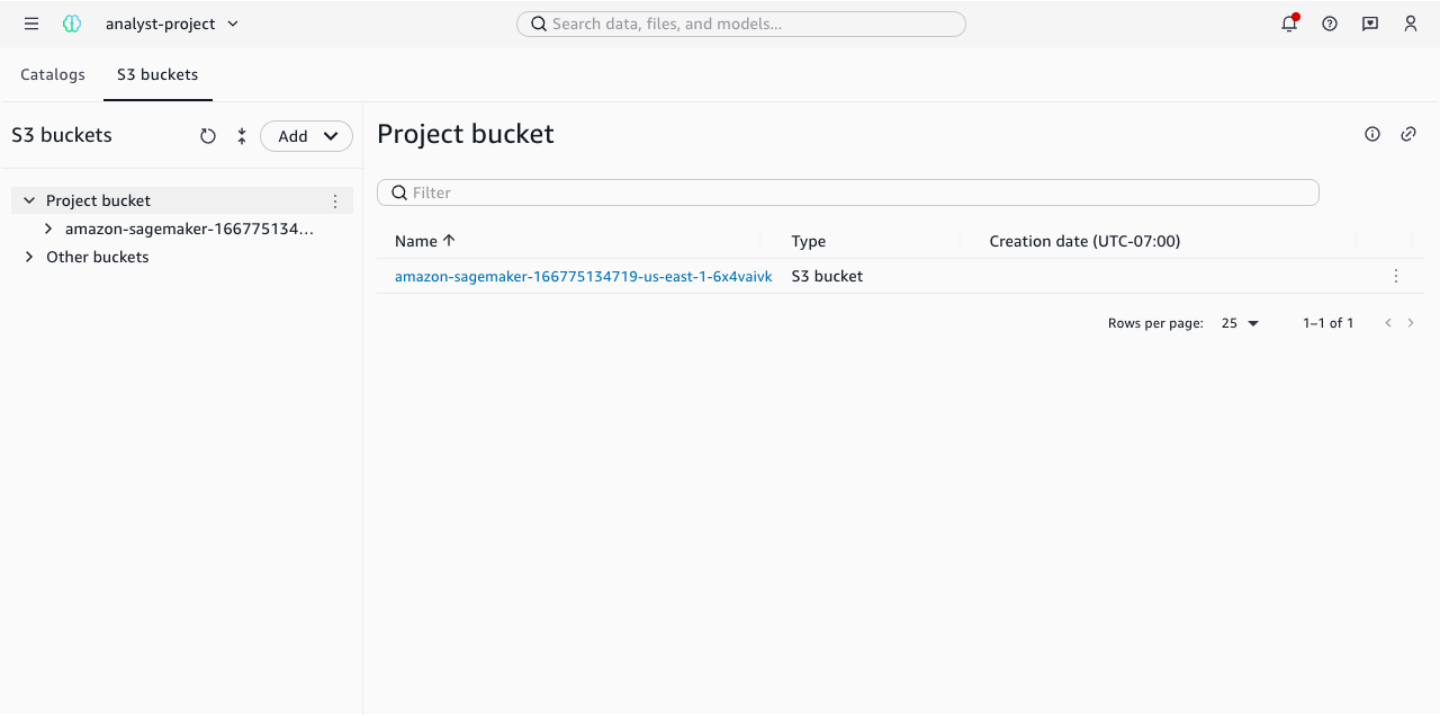
Technical information like the table schema, column names and data types, S3 location, and partition keys is pulled directly from the AWS Glue Data Catalog and is always available. Business context like descriptions, glossary terms, README documentation, and metadata forms is added separately through the admin portal. To learn more about adding business context to your tables, see Curate and enrich asset metadata.



Browse S3 buckets

The **S3 buckets** tab lets you navigate the Amazon S3 buckets that your project's execution role has access to. The left panel organizes your buckets into sections. **Project bucket** is a sandbox S3 prefix created for your project. Project members have full access to read, write, and manage objects within this prefix. **Other buckets** lists additional S3 buckets that your project's execution role has access to.

Expand any bucket in the left panel to browse its folder structure. When you select a bucket or folder, the right panel shows its contents in a table with the file or folder name, type, and creation date.



Manage S3 bucket contents

You can manage the contents of your buckets directly from the data explorer. Select a bucket or folder in the tree, then choose **Actions** to see the available operations:

- **Create folder** creates a new folder inside the selected location.
- **Upload files** uploads one or more files from your local machine.
- **Create table from contents** registers the files in the selected location as a new table in the AWS Glue Data Catalog. This is useful when you have CSV, JSON, Parquet, or other delimited files in S3 and want to make them queryable.
- **Copy** copies the S3 URI to your clipboard.

To add new S3 locations to the data explorer, choose **Add** on the S3 buckets tab. You can **Add S3 location** (an existing bucket or prefix) or **Create S3 bucket** (a new bucket).

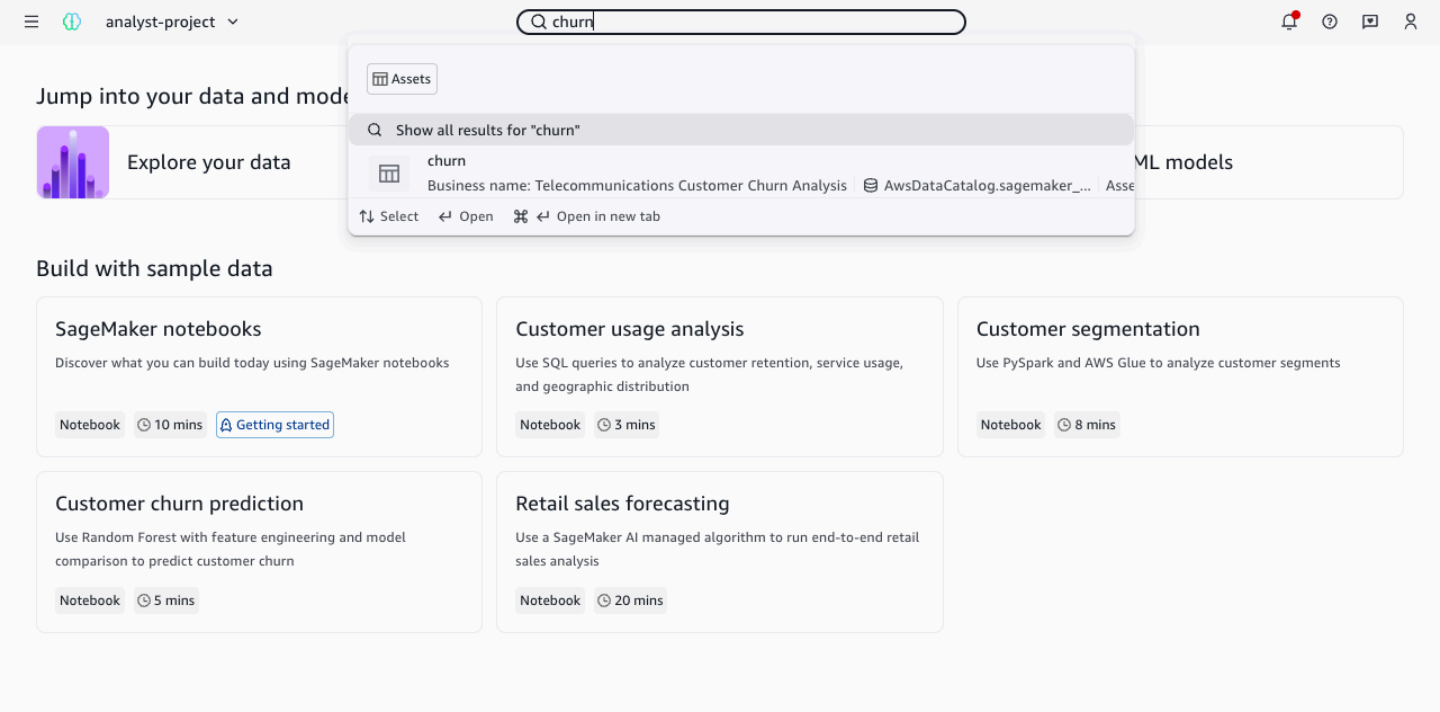
Search and discover data

The **Data** tab in your project shows only the tables your project already has access to. To find tables across the entire domain, including tables you don't have access to yet, you use catalog search. Catalog search matches against table names, business names, descriptions, and business glossary

terms. Results show each table's metadata and business context, so you can evaluate whether a table is relevant before requesting access to it.

From the search results, you can open the full search results page to browse and filter across all matches, view an asset's detail page to see its schema and documentation, or request access to a table directly.

You can search from anywhere in Amazon SageMaker Unified Studio using the search bar at the top of the page. As you type, a dropdown shows matching results with the table name, business name, catalog path, and asset type. You can choose a result directly from the dropdown to open its detail page, or choose **Show all results** to open the full search results page.



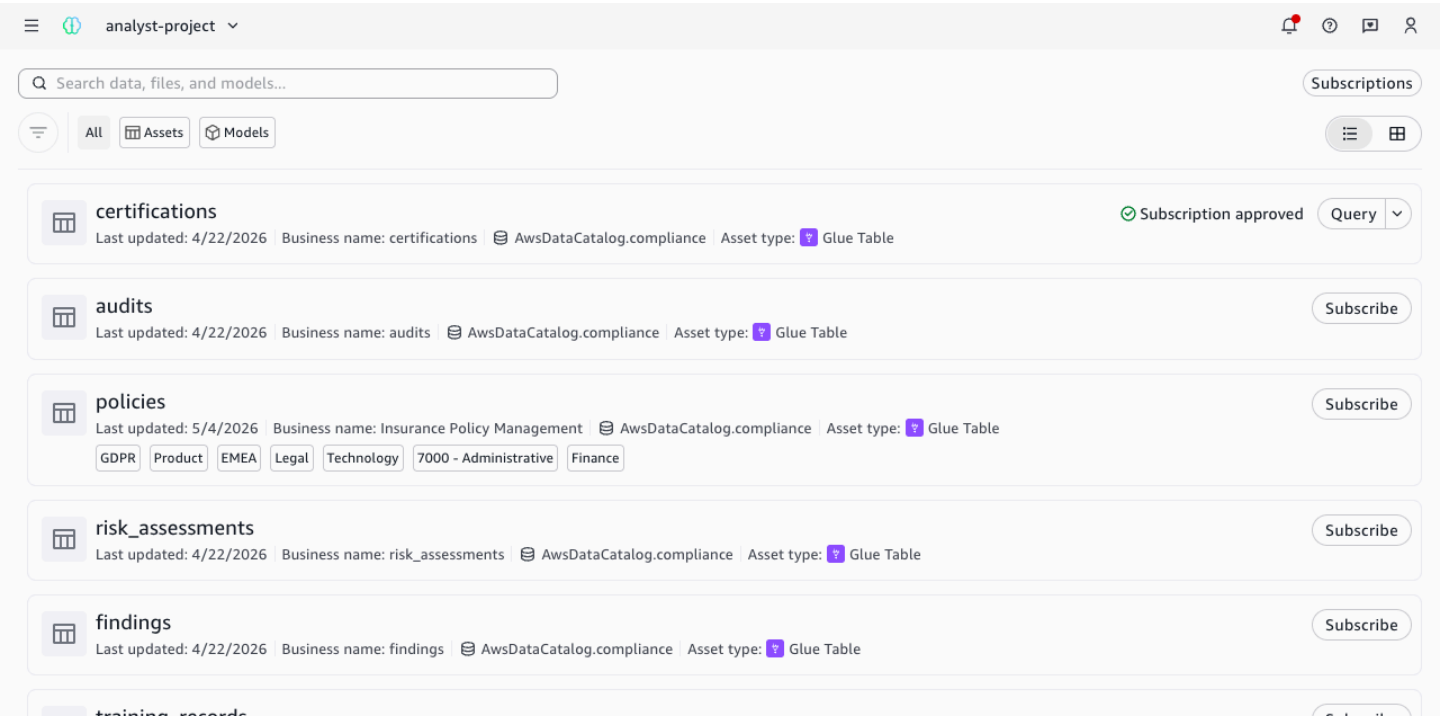
Search results page

The full search results page shows all matching assets. Each result shows the table name, last updated date, business name, catalog path, asset type, and any glossary terms associated with the table. The page includes category filters at the top (such as **All**, **Assets**, and **Models**) and a view toggle to switch between list and grid layouts.

From the search results, you can take the following actions on each result.

- **Subscribe** appears on tables you do not yet have access to. Choose it to request a subscription.

- **Query** appears on tables where your subscription has already been approved. Choose it to open the table in the query editor.



> **Note**
>
> To view the actual data in a table, you must first subscribe to it and have your request approved. Search results show table names and metadata regardless of whether you have access to the underlying data.

Filter search results

The search results page includes filters that help you narrow results. To access filters, choose the **Assets** category pill on the results page. A **Filters** panel appears.

The filters are dynamic. They are generated based on the actual items in your search results. For example, if the tables in your results have business glossary terms or metadata form fields attached to them, those terms and fields automatically appear as filter options. This means the available filters change depending on what you search for.

The filters panel includes standard options such as whether the result is a table or a view, and when it was last updated. Beyond those, you will see glossary terms and metadata form fields that are present on the tables in the current result set. To apply a filter, choose the value you want to filter by in the panel. To remove all active filters, choose **Clear all**.

Custom filters

In addition to the dynamic filters, you can create custom filters when you know something specific about the data you're looking for, such as what the name or description contains. Custom filters let you match against table properties that aren't surfaced as dynamic filter options.

To add a custom filter

1. Open your project in Amazon SageMaker Unified Studio.
2. In the search bar at the top of the page, enter your search query and choose **Show all results for "[query]"**.
3. On the search results page, choose the **Assets** pill at the top.
4. In the **Filters** panel, choose **Add** next to **Custom**.
5. For **Filter type**, choose **Name**, **Description**, or **Metadata form**.
6. For **Operator**, choose **Contains**.
7. For **Value**, enter the text you want to match.
8. (Optional) Choose **+ Add condition** to add more conditions.
9. Choose **Save** to apply the filter.

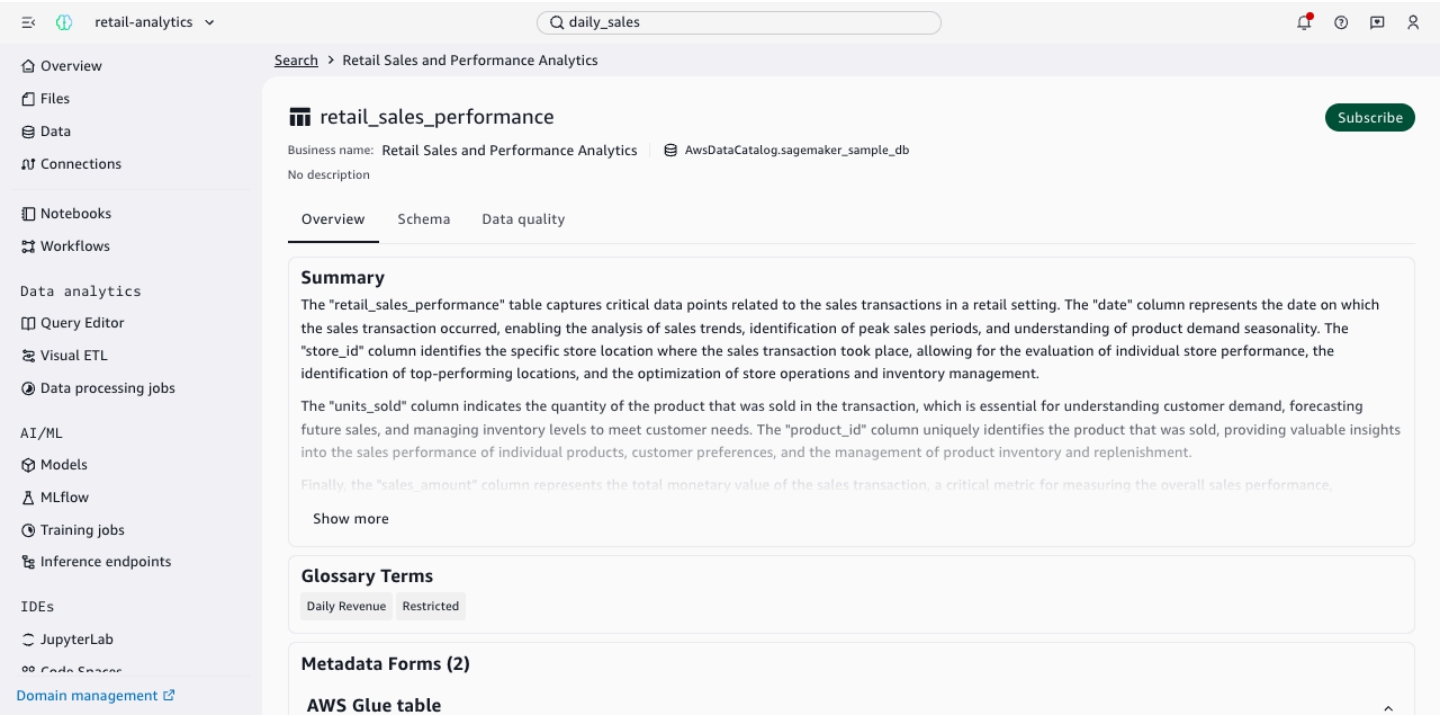
View an asset's detail page

When you choose an asset from the search results, the asset detail page opens. This page shows the table's business name, description, and a **Subscribe** button in the top-right corner. The detail page has the following tabs.

Asset detail page tabs

Tab	When to use it
Overview	Shows the asset's summary, glossary terms associated with the table, README documentation, and additional metadata added by the owner to the asset.

Tab	When to use it
Schema	Shows each column's name, data type, description, glossary terms, and additional metadata added by the owner to the column.
Data quality	Shows data quality rules and their results for the table, if data quality has been configured by the table owner.



Request access to data

When you find a table in the catalog that you want to use, you need to request access to it through a subscription. A subscription is a request from your project to access a specific table. When you submit a subscription request, an administrator reviews it and decides whether to approve or reject it. Once approved, you automatically receive the Lake Formation permissions needed to query the table from your project.

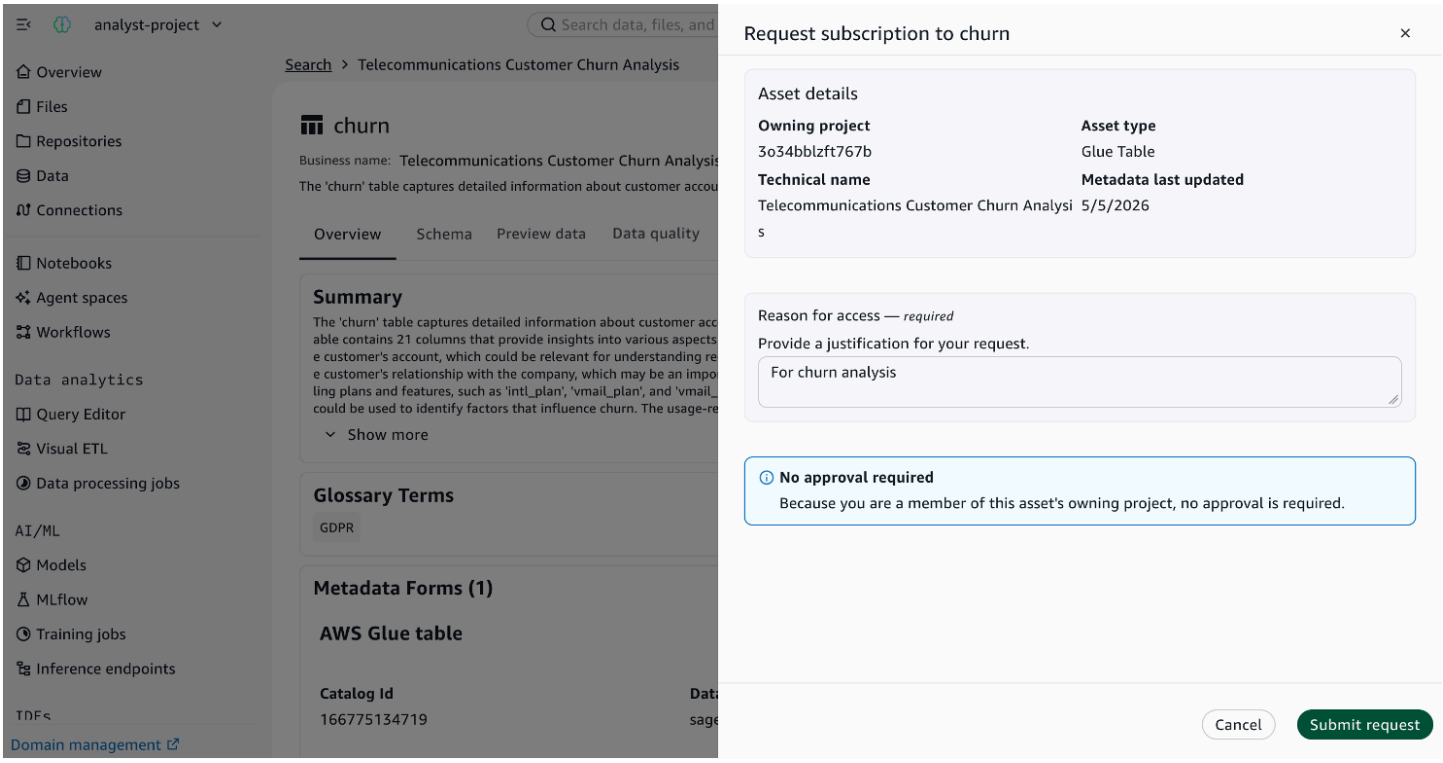
You can request a subscription from two places. On the search results page, choose **Subscribe** next to any table. Or, open a table's detail page and choose **Subscribe**. Both open the **Request subscription** panel.

Request a subscription

To request a subscription

1. Open your project in Amazon SageMaker Unified Studio.
2. In the search bar at the top of the page, enter your search query and choose **Show all results for "[query]"**.
3. Choose **Subscribe** next to the asset you want access to, or choose the asset to open its detail page and then choose **Subscribe**.
4. In the **Request subscription** panel, review the listing details (owning project, asset type, technical name, and metadata last updated).
5. For **Reason for access**, enter a justification for your request.
6. Choose **Submit request**.

Your request is sent to an administrator or the owner of the asset for review. You will receive a notification within the Amazon SageMaker Unified Studio portal when the request is approved or rejected.



> **Note**
>
> The subscription request is made on behalf of the project you are currently working in. To request access for a different project, switch to that project first.

What happens after approval

When an administrator approves your subscription request, Amazon SageMaker Unified Studio automatically grants the necessary Lake Formation permissions to your project's execution IAM role. You do not need to configure any permissions manually. Once the subscription is approved, the table appears in your project's data explorer (under the **Data** tab) and you can start querying it using the Query Editor or a notebook. On the search results page, the table shows a **Query** button instead of **Subscribe**.

View and manage your subscriptions

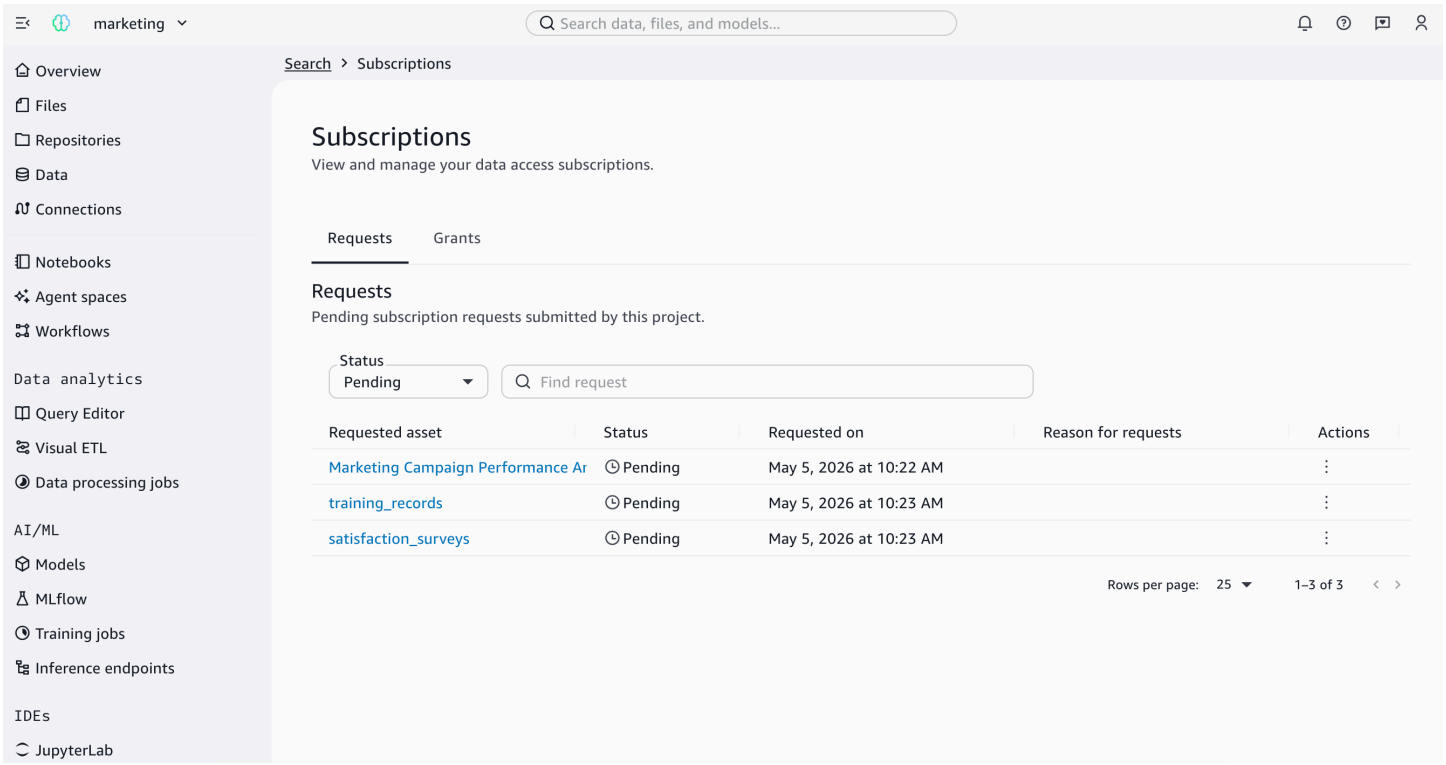
To see the status of your subscription requests and active grants, choose **Subscriptions** on the search results page. The Subscriptions page has the following tabs.

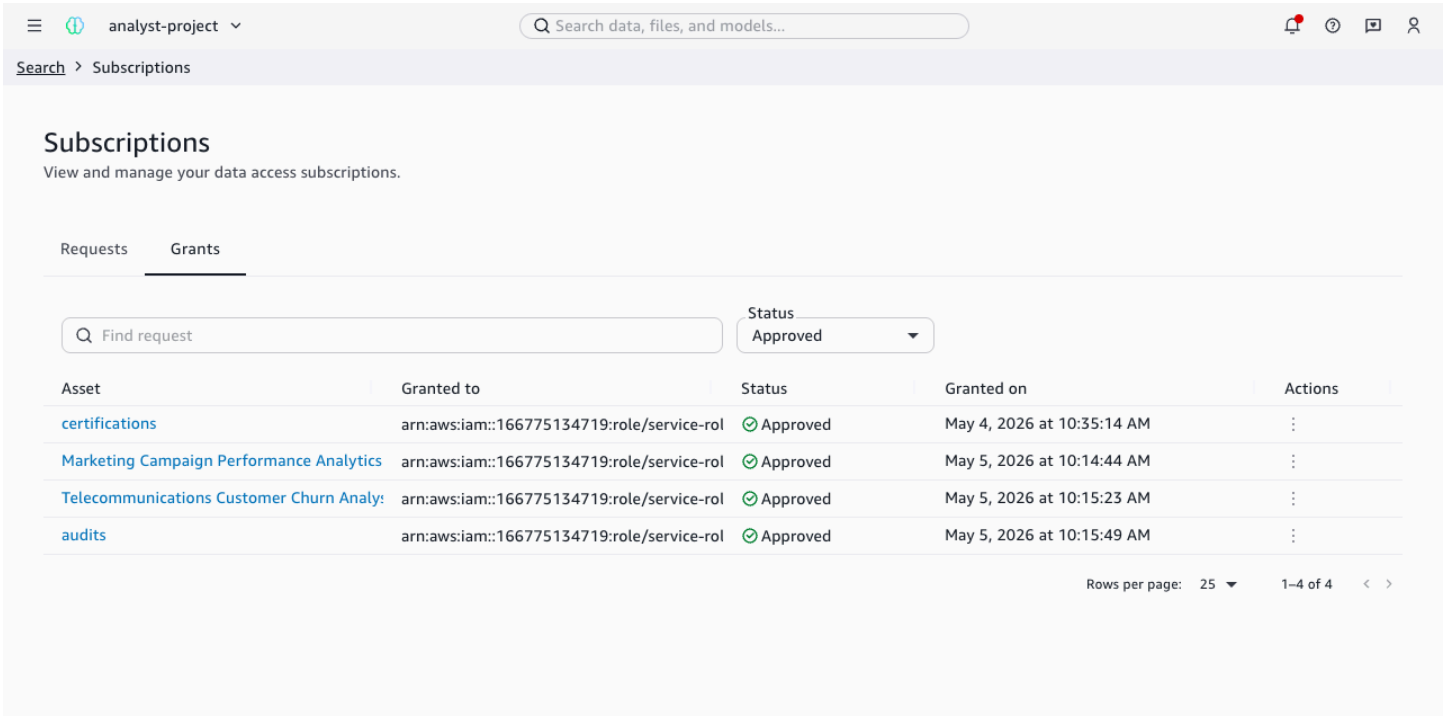
Requests tab

The **Requests** tab shows subscription requests you have submitted. Use the **Status** dropdown to filter by status (Pending or Rejected). Each request shows the requested asset, status, date requested, reason for the request, and available actions.

Grants tab

The **Grants** tab shows subscriptions that have been approved and are currently active. Each grant shows the asset name, who it was granted to (your project's execution role), status, date granted, and available actions.





Cancel a pending request

If you no longer need access to a table, you can cancel a subscription request that is still pending.

To cancel a pending request

1. Open your project in Amazon SageMaker Unified Studio.
2. In the search bar at the top of the page, enter any search query and choose **Show all results for "[query]"**.
3. Choose **Subscriptions** on the search results page.
4. On the **Requests** tab, use the **Status** dropdown to filter by **Pending**.
5. Find the request you want to cancel.
6. In the **Actions** column, choose **Actions**, then choose **Cancel request**.

The request is removed and the administrator will no longer see it in their queue.

Unsubscribe from a table

If you have an active subscription to a table and no longer need access, you can unsubscribe to release the permissions.

To unsubscribe from a table

1. Open your project in Amazon SageMaker Unified Studio.
2. In the search bar at the top of the page, enter any search query and choose **Show all results for "[query]"**.
3. Choose **Subscriptions** on the search results page.
4. Choose the **Grants** tab.
5. Find the approved subscription you want to remove.
6. In the **Actions** column, choose **Actions**, then choose **Unsubscribe**.

After you unsubscribe, the Lake Formation permissions granted through the subscription are revoked and you can no longer query the table from your project.

Manage and govern the catalog

Tables ingested from AWS Glue Data Catalog arrive with technical metadata like column names, data types, and S3 locations. For the rest of the organization to find and use these tables, they need additional business context, such as descriptions, documentation, glossary terms, and

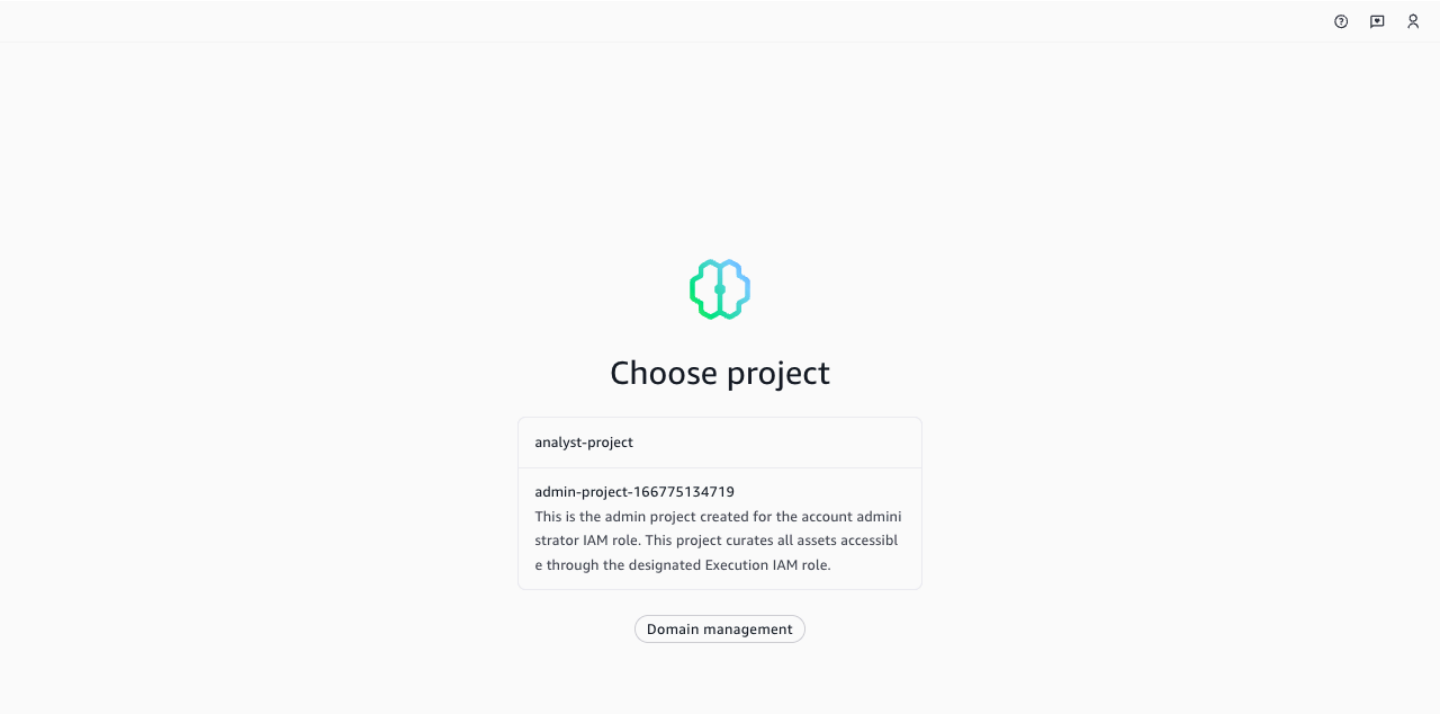
structured metadata. Access also needs to be controlled so that projects can only query the tables they've been granted permission to.

As a domain administrator, you add this business context by curating table metadata, creating business glossaries to standardize terminology, and defining metadata forms to capture structured information. You control access by approving subscription requests and granting permissions to specific projects. You also decide when a table is ready for the rest of the domain to discover by publishing it.

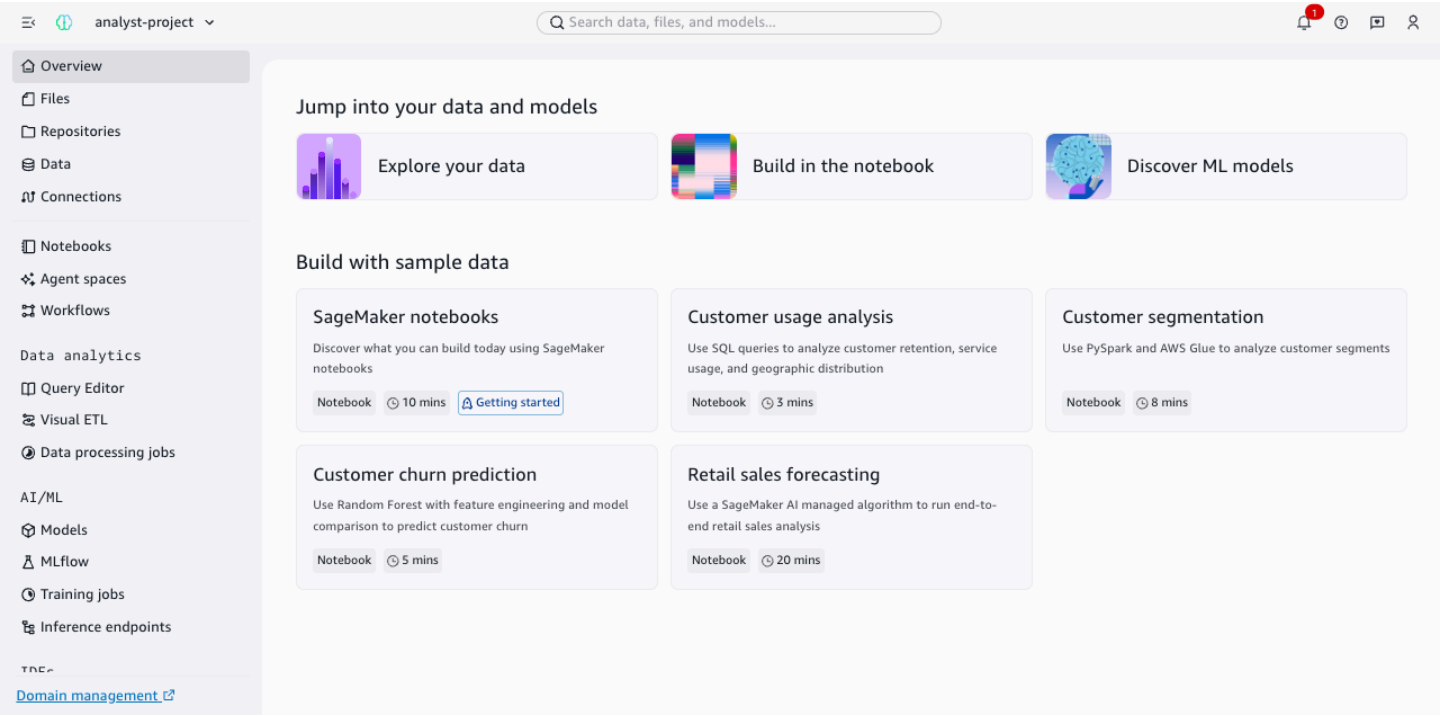
You perform these tasks in the admin portal, which is the domain management console for Amazon SageMaker Unified Studio.

You can open the admin portal from two places.

From the domain home page (the "Choose project" screen), choose **Domain management** below the project list.



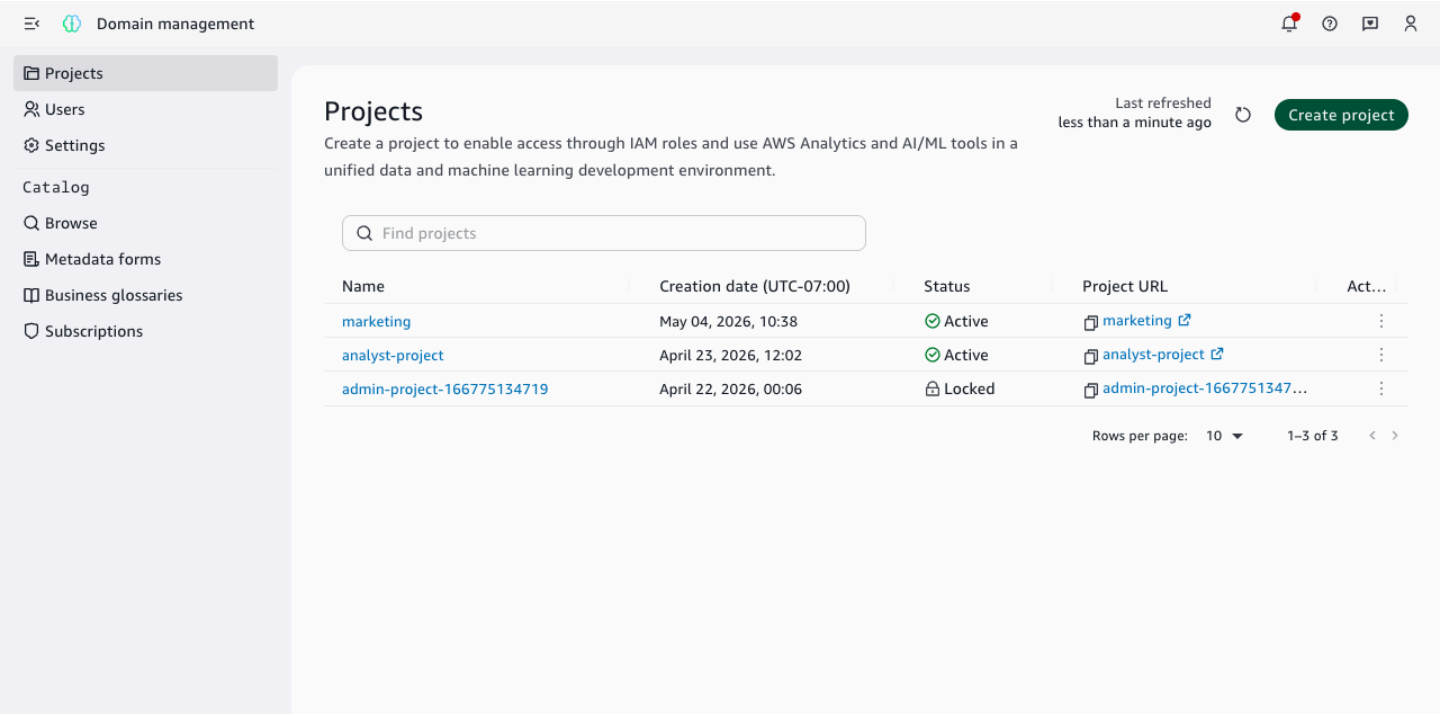
From within a project, in the navigation pane, choose **Domain management**. This opens the admin portal in a new tab.



Once in the admin portal, the left navigation pane includes a **Catalog** section with the pages you will use when working with data catalog features.

Admin portal catalog pages

Page	What you do there
Browse	Browse all tables in the domain, open asset detail pages, and curate metadata (business names, descriptions, glossary terms, metadata forms, README documentation).
Metadata forms	Create and manage structured form templates that can be attached to tables and columns.
Business glossaries	Create and manage business glossaries and terms that standardize vocabulary across the organization.
Custom asset types	Define and manage the types of custom assets that can be stored in your catalog, create custom assets, and manage their metadata.
Subscriptions	Review subscription requests, approve or reject access, and proactively share tables with projects.



Topics

- Approve and manage access
- Create and manage business glossaries
- Standardize metadata using metadata forms
- Curate and enrich asset metadata
- Publish data
- Custom asset types

Approve and manage access

Project members discover tables through catalog search and request access to the ones they need. As an administrator, you review these requests from the **Subscriptions** page in the admin portal and decide whether to approve or reject them. You can also proactively grant access to specific projects or IAM roles without waiting for a request.

Subscriptions control which IAM roles can query a given asset's underlying data. When a project member requests access to an asset, the system submits a subscription request on behalf of the project's execution role (the IAM role that the project uses to run queries and access data in Lake Formation). When you approve the request, a grant is created in Lake Formation for that execution role.

> **Note**
>
> All projects that share the same execution role gain access because the grant is created for the underlying IAM role of the project.

You can also grant access by creating a subscription directly for a project or for a specific IAM role, without waiting for someone to submit a request.

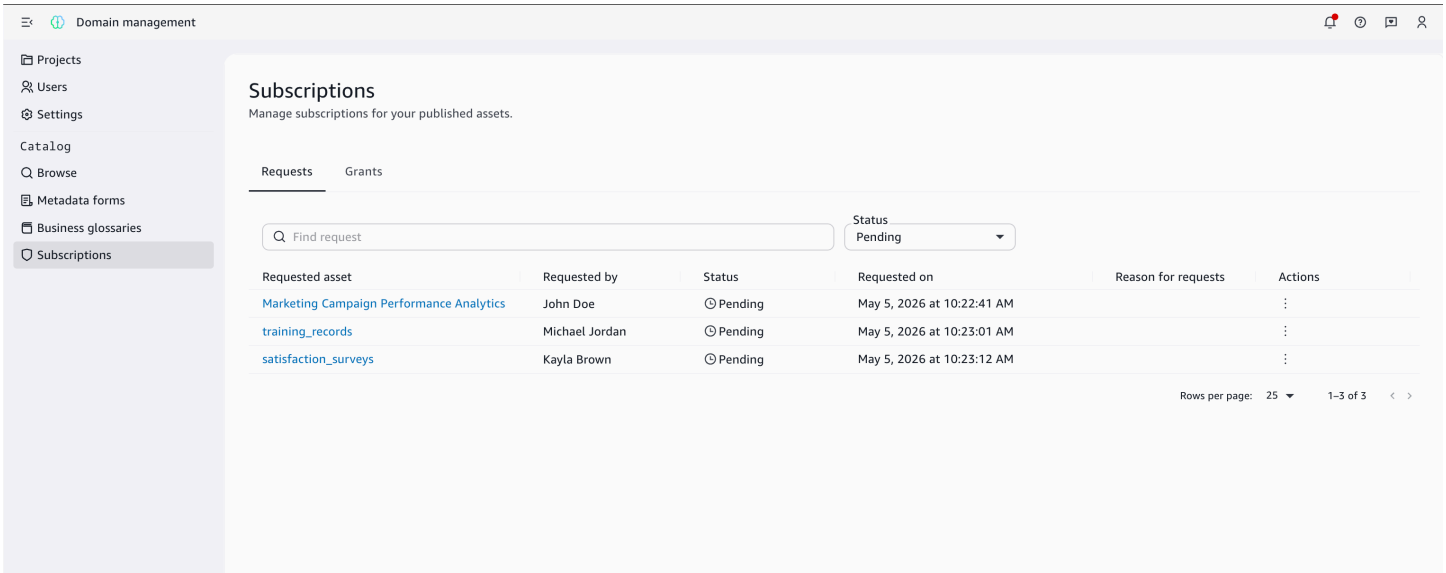
All subscription management happens on the **Subscriptions** page in the admin portal. The page has two tabs:

- The **Requests** tab shows incoming subscription requests. Each request identifies the person who submitted it and the execution role that will receive the Lake Formation grant.
- The **Grants** tab shows all active grants across the domain. The **Granted to** column shows either a project name or an IAM role ARN, depending on how the grant was created.

Review subscription requests

When a project member requests access to an asset, the request appears on the **Requests** tab with a **Pending** status. The **Requested by** column shows the user who requested access. This can be a single sign-on user in IAM Identity Center or an IAM user or role.

> **Note**
>
> The user or role that requests access is not necessarily getting the access. The access is granted to the project's execution role which may be different than the user or role requesting access.

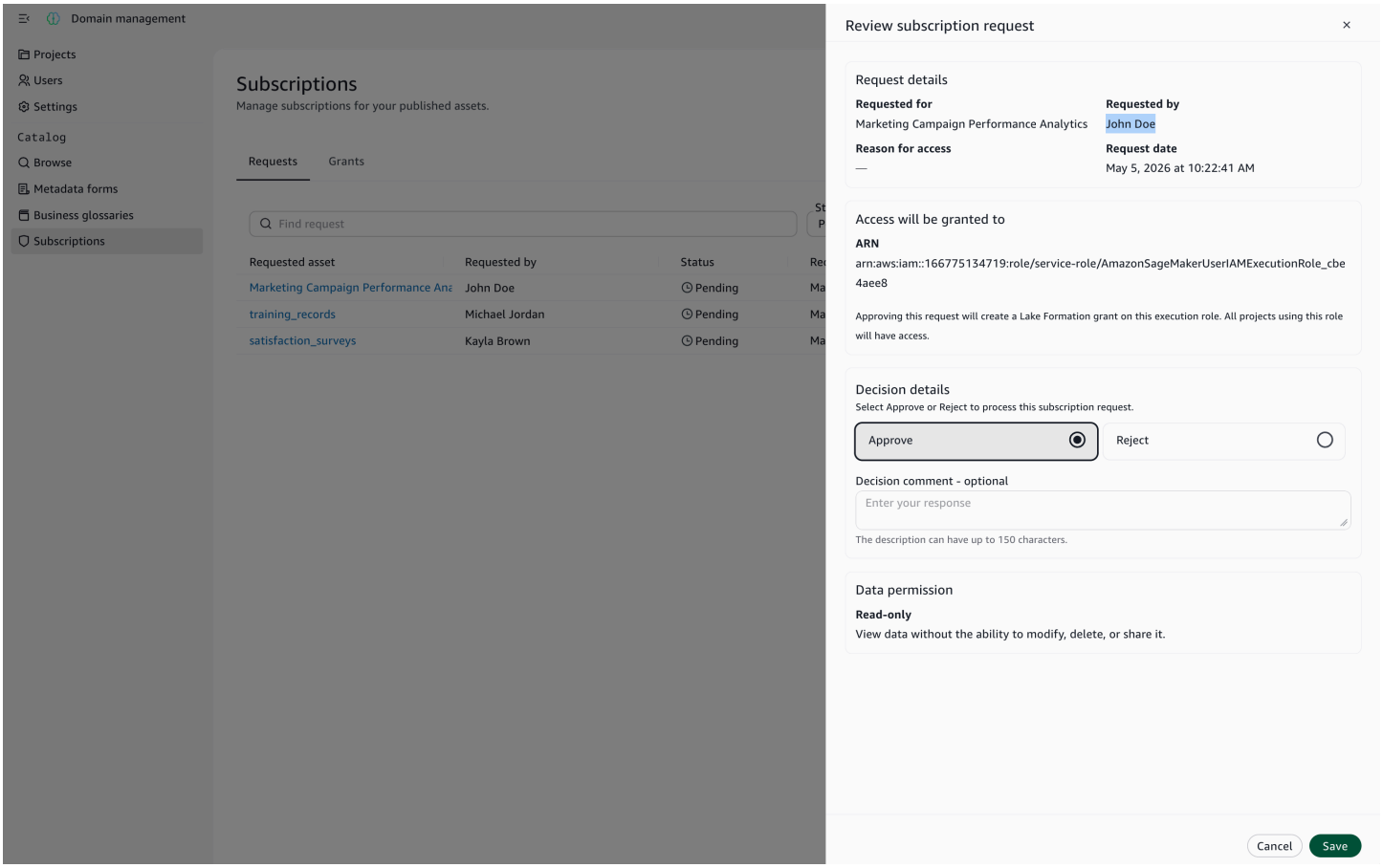


Use the **Status** filter to show requests by state: **Pending**, **Approved**, or **Rejected**.

Approve a request

To approve a subscription request

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Subscriptions**.
3. On the **Requests** tab, find the request you want to review.
4. In the **Actions** column, choose **Actions**, then choose **View request**.
5. In the **Review subscription request** panel, review the request details (**Requested for**, **Requested by**, **Reason for access**, **Request date**) and the **Access will be granted to** section showing the IAM execution role ARN.
6. Under **Decision details**, select **Approve**.
7. (Optional) For **Decision comment**, enter a response (up to 150 characters).
8. Choose **Save**.



After approval, Amazon SageMaker Unified Studio creates a Lake Formation grant for the execution role. Project members whose project uses that execution role can then see the asset in their data explorer and query it using the Query Editor or notebooks.

Reject a request

To reject a subscription request

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Subscriptions**.
3. On the **Requests** tab, find the request you want to reject.
4. In the **Actions** column, choose **Actions**, then choose **View request**.
5. Under **Decision details**, select **Reject**.
6. (Optional) For **Decision comment**, enter a response (up to 150 characters).
7. Choose **Save**.

The requesting user sees the rejection status on their Subscriptions page. They can submit a new request with additional justification if needed.

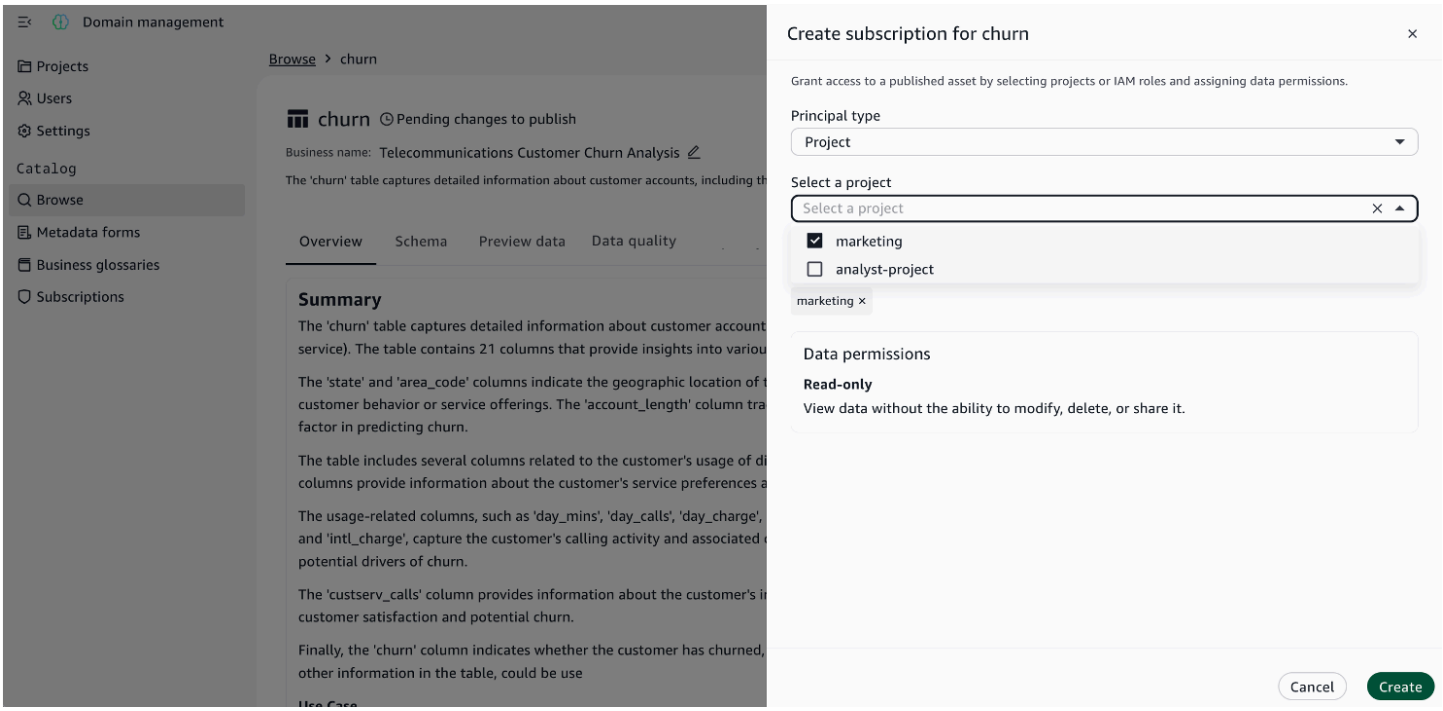
Grant access directly

You can also grant access to an asset without waiting for someone to submit a request. You can grant access to a project (which resolves to the project's execution role) or directly to a specific IAM role.

To grant access directly

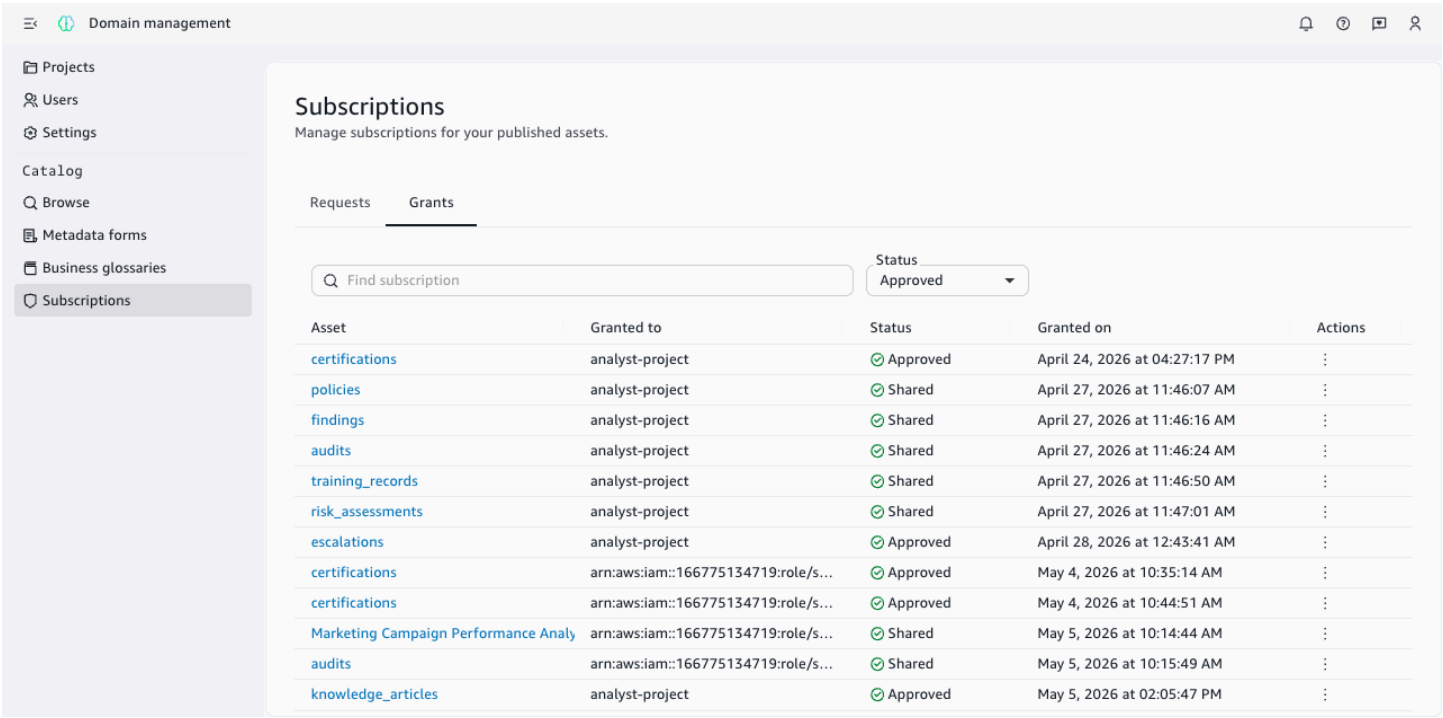
1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Browse**.
3. Choose the asset you want to share.
4. Choose **Actions**, then choose **Create subscription**.
5. In the **Create subscription** panel, for **Principal type**, choose **Project** or **IAM role**.
   - **Project** lets you select from the projects in the domain. The system grants access to the project's execution role.
   - **IAM role** lets you search for and select a specific IAM role ARN directly.
6. For **Select a project** (or **Find IAM role**), choose the principals you want to grant access to.
7. Choose **Create**.

Once the subscription is created, the selected principals gain read access to the asset.



View and manage existing grants

The **Grants** tab shows all active subscriptions across the domain. Each row shows the asset, who it was granted to, the status, and the date access was granted. The **Granted to** column shows either a project name (for project-level grants) or a full IAM role ARN (for role-level grants).



Use the **Status** filter to narrow the list:

- **Approved** means access was granted through a subscription request that was approved.
- **Revoked** means access was previously granted but has been revoked.

Grants created directly by an administrator (through Create subscription) show a status of **Shared**.

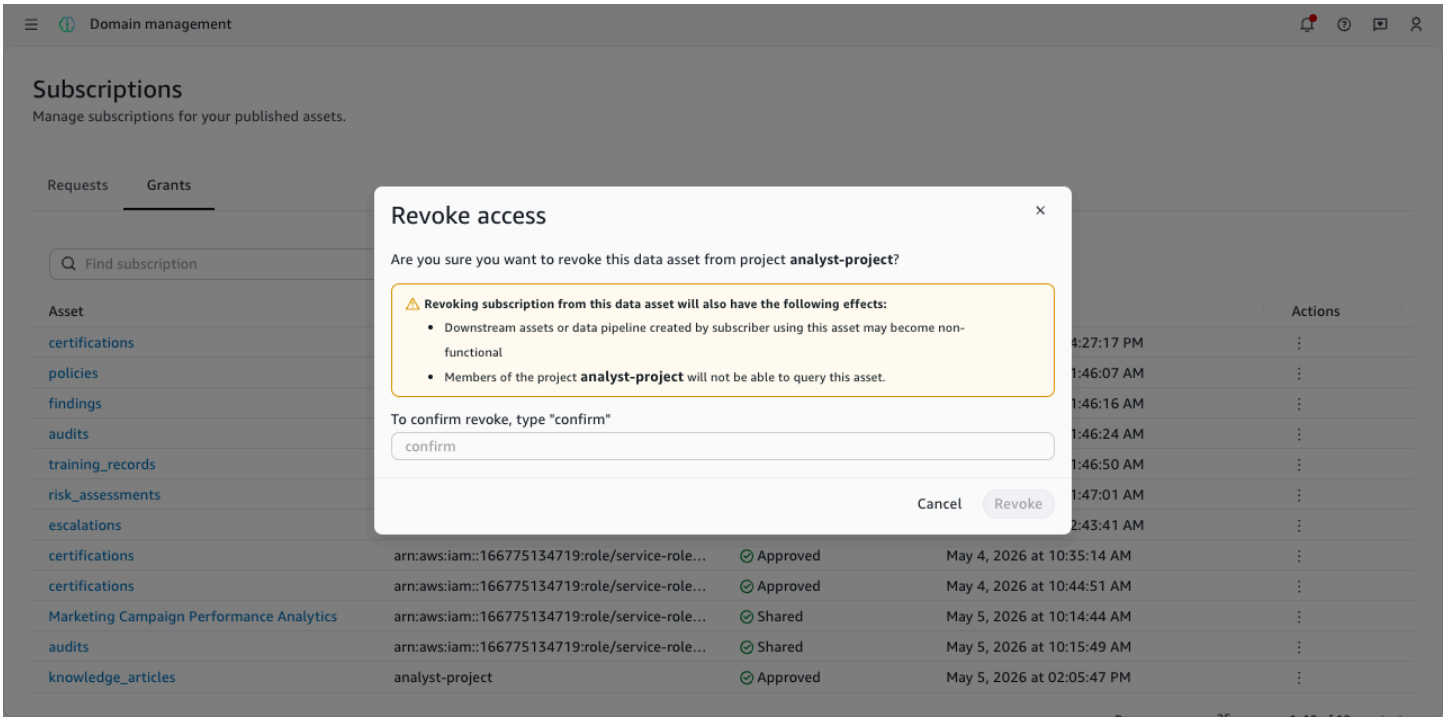
To view the full details of a grant, choose the menu icon in the **Actions** column, then choose **View Details**. The details panel shows who requested access, when it was approved, and the approver's comment.

Revoke access

Revoking a subscription removes the Lake Formation grant from the execution role. The IAM role can no longer query the underlying data for that asset.

To revoke access

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Subscriptions**.
3. Choose the **Grants** tab.
4. Find the grant you want to revoke. In the **Actions** column, choose **Actions**, then choose **Revoke**.
5. In the **Revoke access** dialog, review the warning about downstream effects.
6. For **To confirm revoke, type "confirm"**, enter **`confirm`**.
7. Choose **Revoke**.

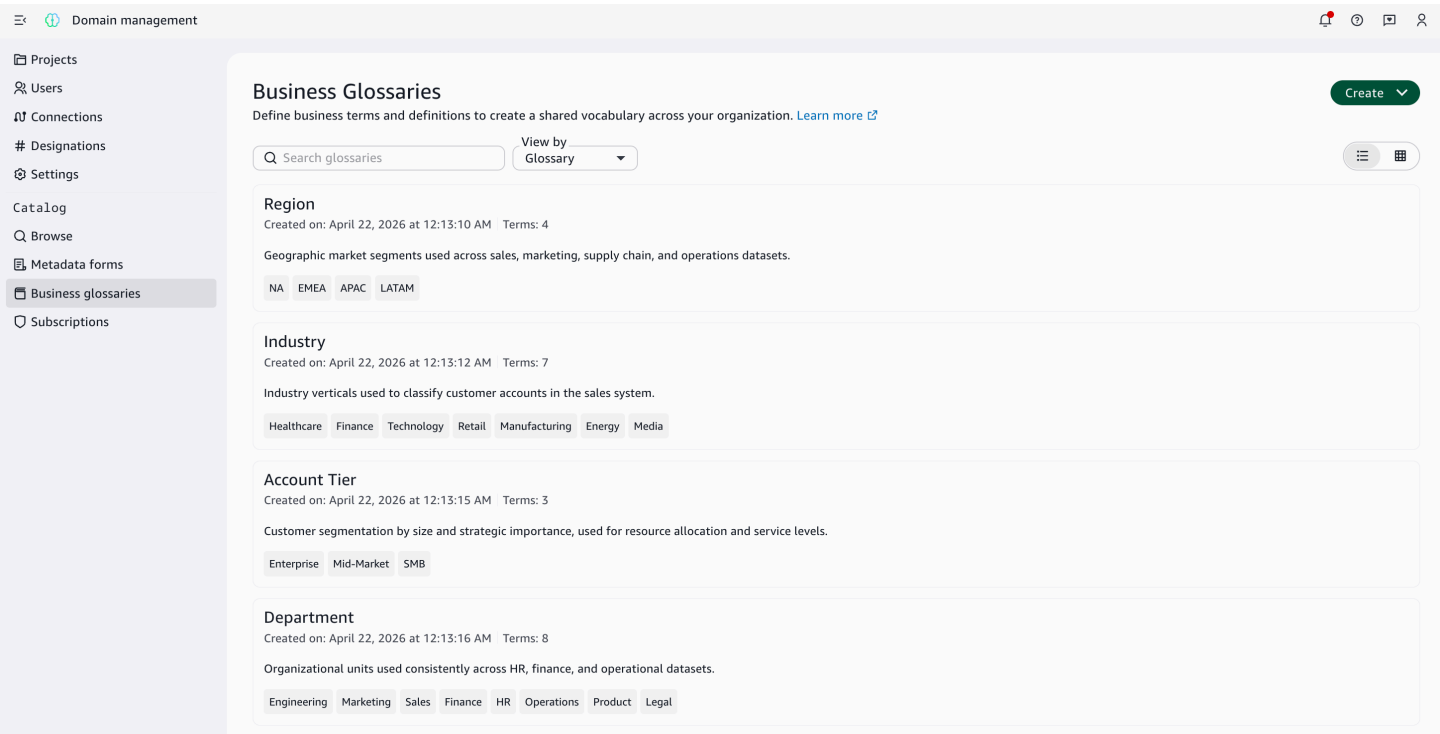


After revoking, the asset is no longer queryable by any project that uses the affected execution role. If they need access again, they must submit a new subscription request.

Create and manage business glossaries

Business glossaries let you define a shared vocabulary for your organization's data. A glossary is a collection of terms, each with a name, description, and optional documentation. For example, you might create a glossary called "Revenue Metrics" with terms like "ARR," "MRR," and "Net Revenue Retention," each with a precise definition of how the metric is calculated, what data sources feed into it, and how it should be interpreted. Once created, these terms can be attached to tables and columns during metadata curation. When someone browses the catalog and sees a column tagged with "ARR," they can choose the term and read the exact definition rather than guessing what it means. This is especially valuable when different teams use the same words to mean different things, or when abbreviations and acronyms are common in column names.

The **Business glossaries** page in the admin portal shows all glossaries in the domain. Each glossary card shows the glossary name, description, creation date, term count, and a preview of the terms it contains. You can search across glossaries and terms using the search bar, use the **View by** dropdown to switch between viewing by **Glossary** or by **Term**.



Create a glossary

To create a glossary

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Business glossaries**.
3. Choose **Create glossary**.
4. For **Name**, enter a name for the glossary (for example, "Compliance Standards" or "Revenue Metrics").
5. For **Description**, enter a description that explains what this glossary covers and when to use it. This field is optional but recommended.
6. Choose **Create glossary**.

The glossary detail page opens, showing the glossary name, description, creation date, an **Actions** menu, and an empty terms table. Glossaries are created in an **Enabled** state by default. See Enable or disable a glossary for details on what the enabled state controls and how to change it.

Edit a glossary

To edit a glossary

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Business glossaries**.
3. Choose the glossary you want to edit.
4. Choose **Actions**, then choose **Edit**.
5. Update the name or description.
6. Choose **Save**.

Enable or disable a glossary

Each glossary has an **Enabled** toggle on its detail page. When a glossary is enabled, it appears as an available option when someone adds a Business glossary field type to a metadata form (see Add a field to a form). Its enabled terms can also be attached to assets during metadata curation. When a glossary is disabled, it cannot be added to new forms and its terms cannot be attached to new assets. Disabling a glossary does not remove it from forms or assets where it is already in use.

Individual terms also have an **Enabled** toggle. See Enable or disable a term in the Glossary terms section for details.

To enable or disable a glossary

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Business glossaries**.
3. Choose the glossary you want to enable or disable.
4. On the glossary detail page, toggle the **Enabled** switch.

Delete a glossary

Before you can delete a glossary, you must first disable it and remove all of its terms. See Delete a term for the steps to remove individual terms.

To delete a glossary

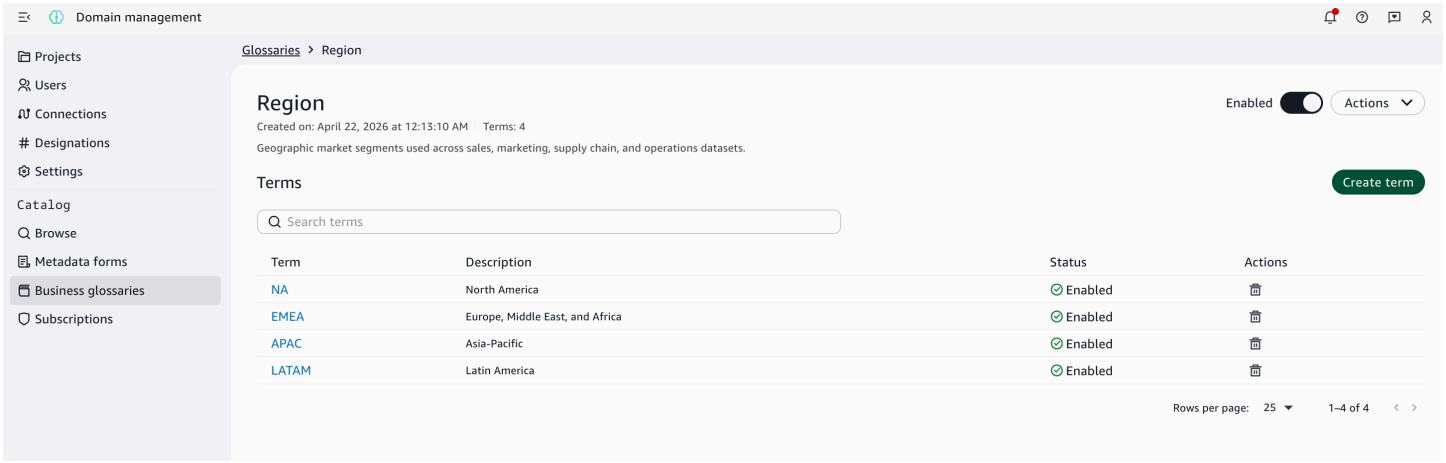
1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Business glossaries**.
3. Choose the glossary you want to delete.
4. Delete all terms in the glossary (see Delete a term).
5. Choose **Actions**, then choose **Delete**.
6. Confirm the deletion.

> **Note**
>
> You cannot delete a glossary that is still enabled or contains terms. Disable the glossary and remove all terms first.

Glossary terms

Terms are the individual entries within a glossary. Each term has a name, description, and an optional README where you can add detailed documentation using Markdown. Terms can be enabled or disabled independently of the parent glossary, giving you control over which terms are available for use.

The glossary detail page shows all terms in a table with columns for the term name, description, status (enabled or disabled), and a delete action.

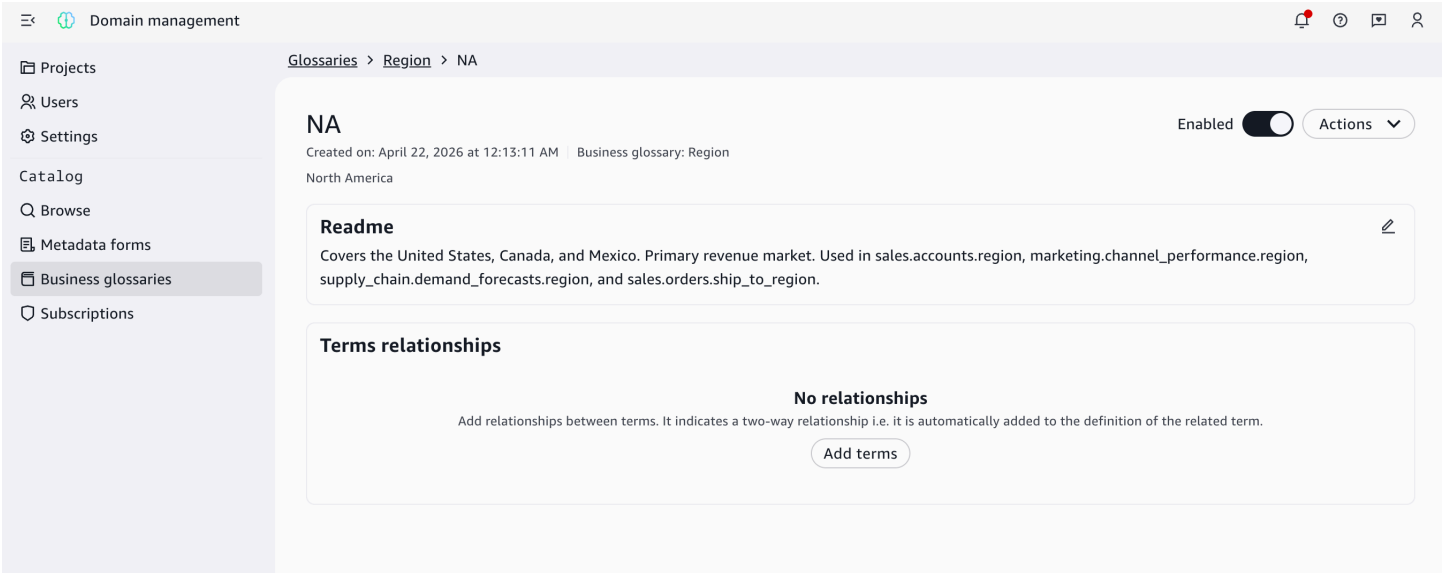


Create a term

To create a term

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Business glossaries**.
3. Choose the glossary where you want to add the term.
4. Choose **Create term**.
5. For **Name**, enter the term name (for example, "GDPR" or "Annual Recurring Revenue").
6. For **Description**, enter a definition that explains what this term means and how it should be interpreted. This field is optional but recommended.
7. The **Enabled** toggle is on by default. Turn it off if you want to finalize the term before making it available.
8. Choose **Create term**.

The term detail page opens, showing the term name, description, creation date, a link back to the parent glossary, the **Enabled** toggle, an **Actions** menu, a **README** section, and a **Terms relationships** section.



Edit a term

To edit a term

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Business glossaries**.
3. Choose the glossary that contains the term.
4. Choose the term you want to edit.
5. Choose **Actions**, then choose **Edit**.
6. Update the name or description.
7. Choose **Save**.

Add a README to a term

Each term has a README section where you can add documentation using Markdown syntax. This is useful for providing detailed context that goes beyond the short description, such as calculation methodology, data sources, or usage guidelines. The README supports up to 4,096 characters and includes a side-by-side preview so you can see how the formatted content will look.

To add a README to a term

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Business glossaries**.
3. Choose the glossary that contains the term.
4. Choose the term.
5. In the **README** section, choose **Edit**.
6. Enter your documentation using Markdown syntax.
7. Choose **Save**.

Enable or disable a term

Each term has an **Enabled** toggle on its detail page. When a term is enabled, it can be attached to assets during metadata curation. When disabled, it cannot be attached to new assets but remains on any assets where it was previously attached.

To enable or disable a term

1. On the glossary detail page, choose the term you want to enable or disable.
2. On the term detail page, toggle the **Enabled** switch.

Delete a term

You can delete a term from the glossary detail page or from the term detail page.

From the glossary detail page, choose the delete icon (trash) in the **Actions** column next to the term you want to remove.

From the term detail page, follow these steps.

To delete a term from the term detail page

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Business glossaries**.
3. Choose the glossary that contains the term.
4. Choose the term you want to delete.
5. Turn off the **Enabled** toggle (you must disable a term before deleting it).
6. Choose **Actions**, then choose **Delete**.
7. Confirm the deletion.

> **Note**
>
> You must disable a term before you can delete it. Turn off the **Enabled** toggle on the term detail page first. The term must also not be attached to any assets or have any term relationships. Remove all asset associations and term relationships before deleting.

Term relationships

You can define relationships between terms to show how they relate to each other. Relationships use a parent-child model with two types. **Is A Type of** means the current term is a more specific version of the related term. **Has Types** means the current term is a broader category that includes

the related term. For example, if you set "Credit Card" as "Is A Type of" "Payment Method," then Payment Method automatically shows Credit Card under its "Has Types" relationships. Relationships are always bidirectional in this way.

Relationships can be created between terms in the same glossary or across different glossaries. A term can have a maximum of 10 relationships per relationship type, and a term can be attached to a maximum of 10 assets.

To add a term relationship

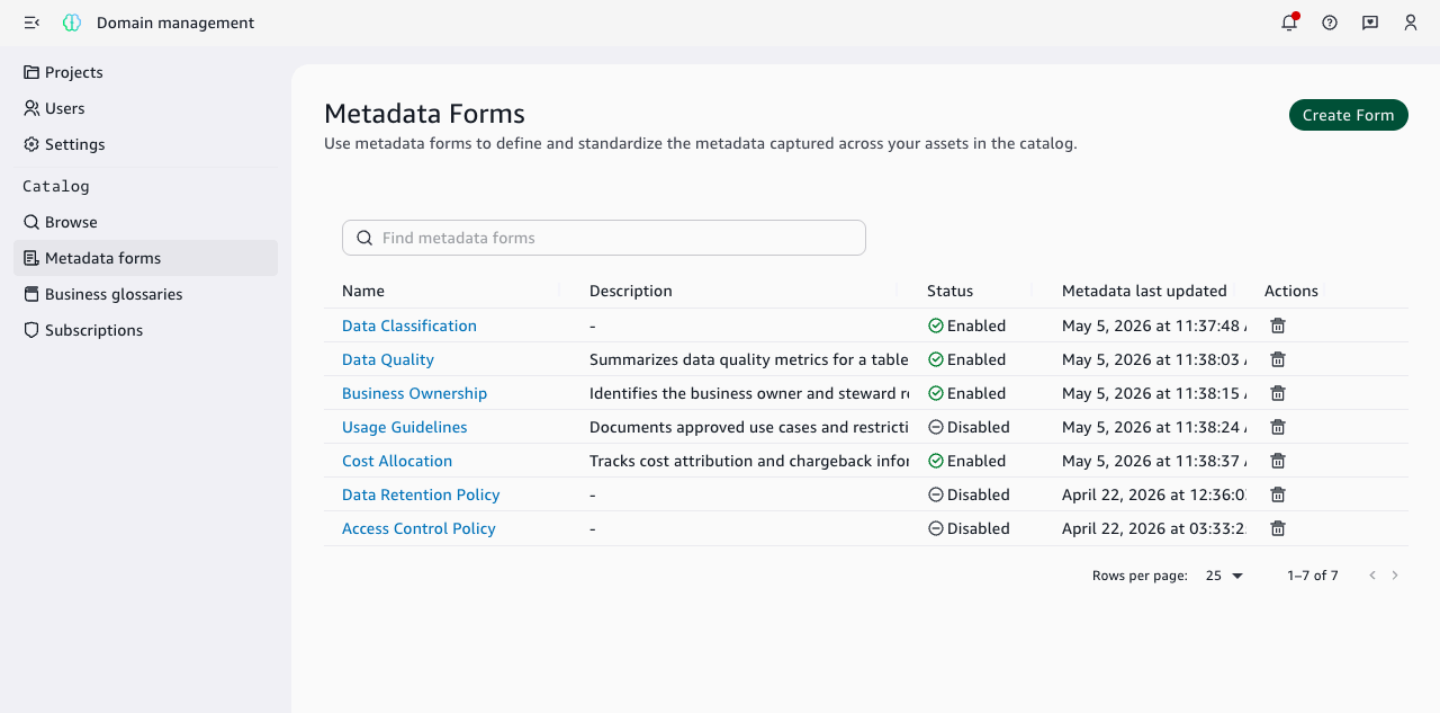
1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Business glossaries**.
3. Choose the glossary that contains the term, then choose the term.
4. In the **Terms relationships** section, choose **Add terms**.
5. In the **Add term relationships** panel, for **Relationship**, choose **Is A Type of** or **Has Types**.
6. Use the search bar to find the term you want to relate to. You can also use the **Filter by glossary** dropdown to narrow results to a specific glossary.
7. Select the checkbox next to one or more terms.
8. Choose **Add terms** to confirm your selection.
9. In the **Manage term relationships** panel, review the relationships and choose **Save changes**.

To edit or remove existing relationships, choose **Edit** in the **Terms relationships** section on the term detail page. This opens the **Manage term relationships** panel where you can remove relationships using the delete icon or add new ones using the **Add terms** button.

Standardize metadata using metadata forms

Metadata forms let you define structured templates that capture specific information about a table. A form is a set of typed fields (text, numbers, dates, booleans) that you design to match the information you want to track. For example, you might create a form called "Data Retention Policy" with a string field for retention period, a date field for next review date, and a boolean field for whether the data contains PII. When the form is attached to a table during curation, the person curating the table fills in the field values. Unlike free-text descriptions, form fields are structured and searchable, so you can later filter the catalog to find all tables where "Contains PII" is true, or where the retention period is "90 days."

The **Metadata forms** page in the admin portal shows all forms in the domain. The table shows each form's name, description, status (enabled or disabled), and when the metadata was last updated. You can search for forms using the search bar at the top.



Create a metadata form

Each metadata form has two names. The **display name** is what people see in the admin portal and when the form is attached to an asset. The **technical name** is used when working with APIs. The technical name cannot be changed after creation, so choose it carefully.

To create a metadata form

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Metadata forms**.
3. Choose **Create Form**.
4. For **Display name**, enter a human-readable name (for example, "Data Retention Policy" or "Access Control Policy").
5. For **Technical name**, enter an API-friendly name (for example, "DataRetentionPolicy"). This name must be 1 to 64 characters and can contain letters, numbers, spaces, dashes, and underscores.
6. For **Description**, enter a description of what this form captures. This field is optional.

7. Choose **Create Metadata Form**.

The form detail page opens. Forms are created in an **Enabled** state by default. If you want to finish adding fields before making the form available, turn off the **Enabled** toggle before creating the form.

> **Note**
>
> The technical name cannot be changed after creation. If you need a different technical name, you must delete the form and create a new one.

Enable or disable a metadata form

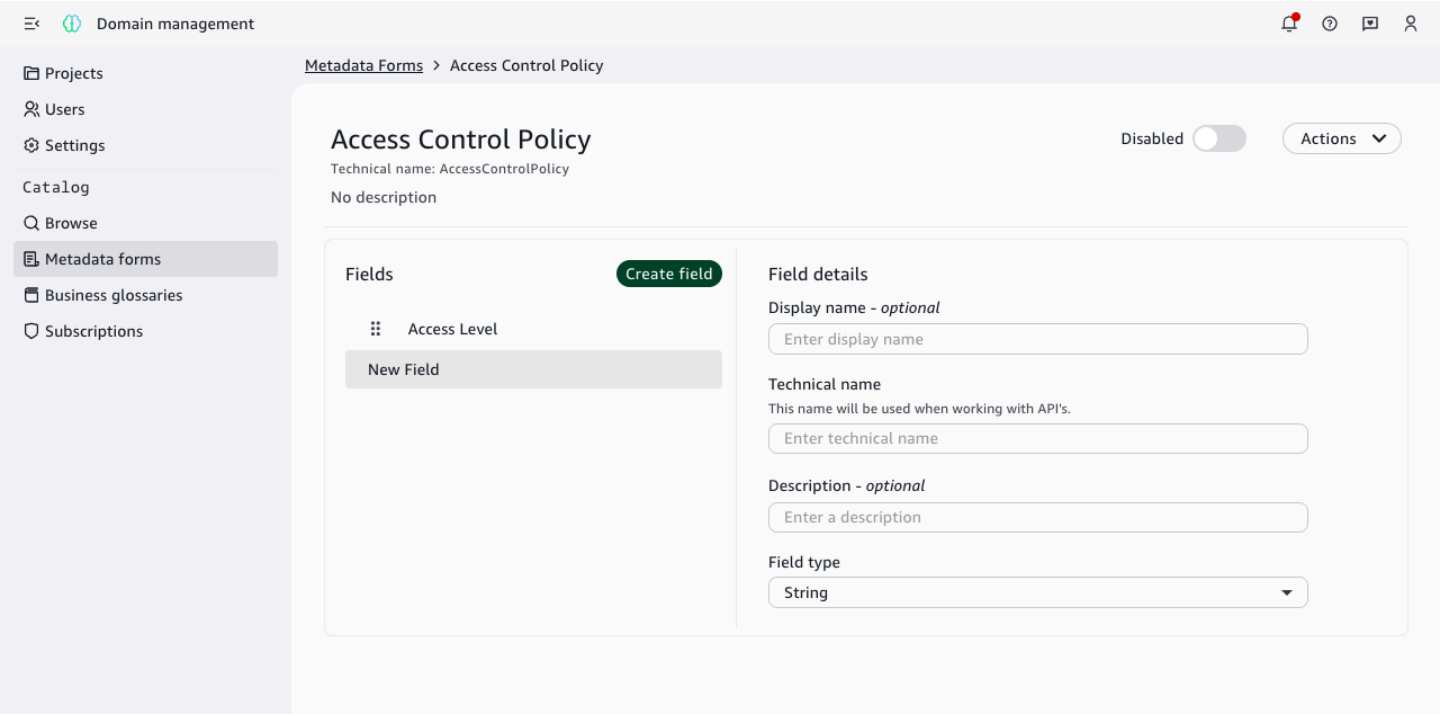
Each form has a **Disabled/Enabled** toggle on its detail page. When a form is enabled, it can be attached to assets during metadata curation. When disabled, it cannot be attached to new assets.

To enable or disable a metadata form

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Metadata forms**.
3. Choose the form you want to enable or disable.
4. On the form detail page, toggle the **Disabled/Enabled** switch.

Add a field to a form

The form detail page has a split layout. The **Fields** panel lists the fields in the form, which you can drag to reorder. The **Field details** panel shows the properties of the selected field or the inline editor for adding a new field.



To add a field to a form

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Metadata forms**.
3. Choose the form you want to add a field to.
4. In the **Fields** panel, choose **Create field**. A **New Field** entry appears and the field editor opens.
5. For **Display name**, enter a human-readable name for the field (for example, "Access Level"). This field is optional.
6. For **Technical name**, enter an API-friendly name (for example, "access_level"). This field is required.
7. For **Description**, enter a description of what this field captures. This field is optional.
8. For **Field type**, choose one of the following types: **String** (free-form text), **Integer** (a whole number), **Long** (a large whole number), **Float** (a number with decimal places), **Double** (a large number with decimal places), **Boolean** (a true/false value), **Date** (a calendar date), or **Business glossary** (a term from a glossary).

> **Note**
>
> When you choose **Business glossary** as the field type, you select which glossary to use for this field. When someone later applies this form to an asset, they pick a term from that glossary as the field value instead of typing free-form text. This is useful when you want to standardize how people classify assets, for example by choosing a compliance category or a data domain from a controlled vocabulary. Only enabled glossaries appear in the list of available glossaries. For more information about enabling glossaries, see Enable or disable a glossary.

Before saving, you can configure the following additional settings for the field:

- **String length** (String fields only) - Set the minimum and maximum character length.
- **Searchable** - Under **Search filters**, select this if you want the field's values to appear as filter options in catalog search.
- **Required** - Under **Requirement options**, choose **Required** if this field must be filled in whenever the form is attached to an asset. The default is **Optional**.

When you are done, choose **Add field**.

Edit a field

To edit a field

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Metadata forms**.
3. Choose the form that contains the field.
4. In the **Fields** panel, choose the field you want to edit.
5. In the **Field details** panel, choose **Edit**.
6. Update the field properties.
7. Choose **Save**.

Delete a field

To delete a field

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Metadata forms**.
3. Choose the form that contains the field.
4. In the **Fields** panel, choose the field you want to delete.
5. In the **Field details** panel, choose **Delete**.
6. Confirm the deletion.

Delete a metadata form

Before you can delete a metadata form, you must first disable it and detach it from all assets where it is currently used.

To delete a metadata form

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Metadata forms**.
3. Choose the form you want to delete.
4. Toggle the **Disabled/Enabled** switch to disable the form.
5. Detach the form from all assets where it is currently attached.
6. Choose **Actions**, then choose **Delete**.
7. Confirm the deletion.

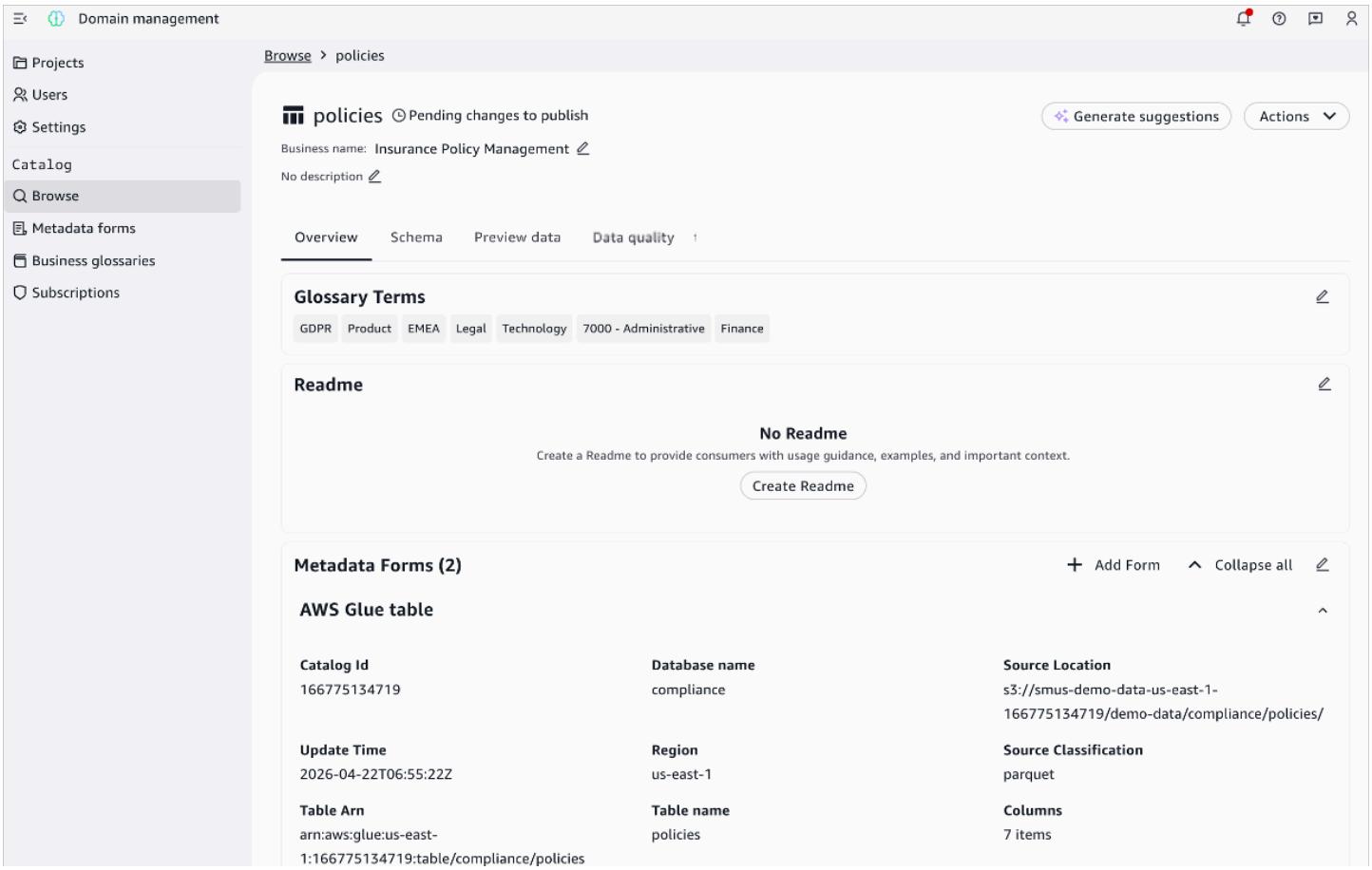
> **Note**
>
> You cannot delete a metadata form that is still enabled or still attached to assets.

Curate and enrich asset metadata

When assets are ingested into the catalog from AWS Glue Data Catalog, they come with technical metadata like column names, data types, and S3 locations. This information is useful for engineers who already know what they're looking for, but it doesn't help someone who is trying to understand what a table represents, whether it's trustworthy, or how it should be used. Metadata curation is the process of adding that business context so that tables become understandable and discoverable by everyone in the organization. You do this from the **Browse** page in the admin portal. You can add this context manually, or use AI to automatically generate business names, descriptions, and glossary term associations for tables and their columns. See Use AI to generate metadata suggestions for details.

All metadata curation happens on the asset detail page. To get there, choose **Browse** in the left navigation pane, then choose the table you want to curate. The asset detail page shows the table's business name and description at the top (both editable), along with a **Generate suggestions** button for AI-generated metadata. The **Overview** tab is where you manage the summary, glossary terms, README documentation, and metadata forms. The **Schema** tab is where you curate individual columns, including their business names, descriptions, glossary terms, metadata forms, and README documentation.

As you make changes, the asset shows a **Changes pending** badge next to the table name. These changes are saved as a new draft revision but are not discoverable to anyone until you re-publish the asset. See Publish data for details.



Edit the business name

Each table has a technical name that comes from the AWS Glue Data Catalog (for example, `policies`) and a business name that you can set to something more descriptive (for example, "Insurance Policy Management"). The business name appears in search results, making it easier for people to identify the table without knowing the technical name. You can also use AI to generate business names for the table and all of its columns at once.

To edit the business name

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Browse**.
3. Choose the asset you want to edit.
4. Next to **Business name**, choose the edit icon (pencil).
5. Enter the new business name in the text field.

6. Choose **Save**.

Edit the description

The description is a short text that explains what the table contains and what it is used for. It appears above the tabs on the asset detail page and is separate from the longer AI-generated summary on the **Overview** tab. You can also use AI to generate descriptions for the table and all of its columns at once.

To edit the description

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Browse**.
3. Choose the asset you want to edit.
4. Below the business name, choose the edit icon (pencil) next to the description text.
5. Enter or update the description text.
6. Choose **Save**.

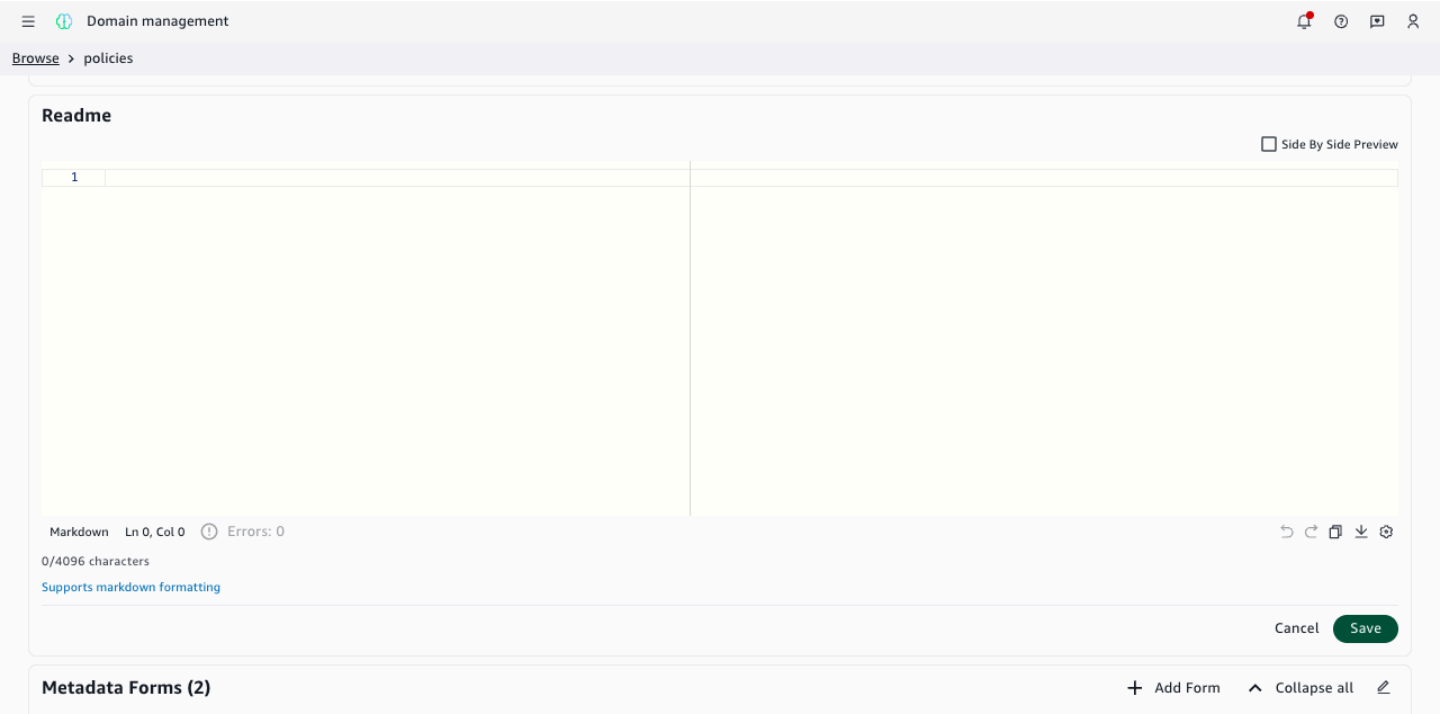
Add or edit the README

The README is a longer-form documentation section where you can provide detailed context about a table using Markdown syntax. This is useful for documenting things like refresh schedules, known data quality issues, usage guidelines, or anything else that helps someone understand and use the data correctly. The README supports up to 4,096 characters and includes a side-by-side preview so you can see how the formatted content will look.

To add or edit the README

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Browse**.
3. Choose the asset you want to edit.
4. On the **Overview** tab, find the **README** section.
5. Choose **Edit**.
6. Enter your documentation using Markdown syntax.

7. (Optional) Use the side-by-side preview to see how the formatted content will look.
8. Choose **Save**.



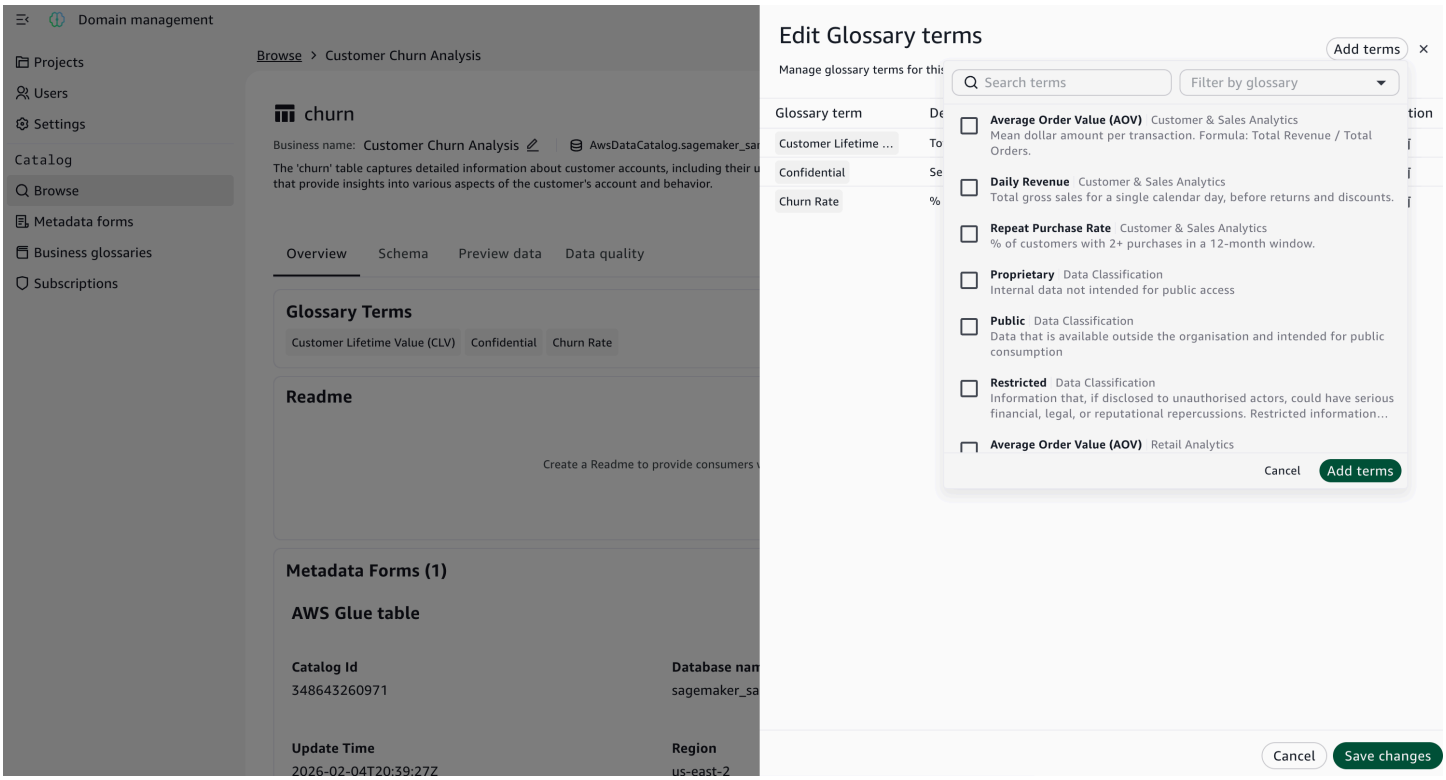
Associate glossary terms with a table

Glossary terms help people understand what a table represents by linking it to standardized business vocabulary. For example, tagging a table with "GDPR" tells someone that the table contains data subject to GDPR compliance requirements, without them having to read through the schema or documentation. Terms must be created in a glossary before they can be associated with a table. See Create and manage business glossaries for how to create glossaries and terms. You can also use AI to suggest glossary term associations based on the table's technical metadata.

To associate glossary terms with a table

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Browse**.
3. Choose the asset you want to edit.
4. On the **Overview** tab, in the **Glossary Terms** section, choose the edit icon. The **Edit Glossary terms** panel opens, showing any terms already associated with the asset.

5. Choose **Add terms**.
6. Search for the term you want to associate, or browse the available terms. You can filter by glossary using the dropdown.
7. Select the checkbox next to one or more terms.
8. Choose **Add terms** to confirm your selection.
9. Review the updated list of associated terms. To remove a term, choose the remove icon (trash) in the **Action** column.



Attach a metadata form to a table

Metadata forms let you capture structured information about a table using predefined fields. For example, a "Data Retention Policy" form might have fields for retention period, next review date, and whether the data contains PII. Forms must be created and enabled before they can be attached to a table. See Standardize metadata using metadata forms for how to create forms.

To attach a metadata form to a table

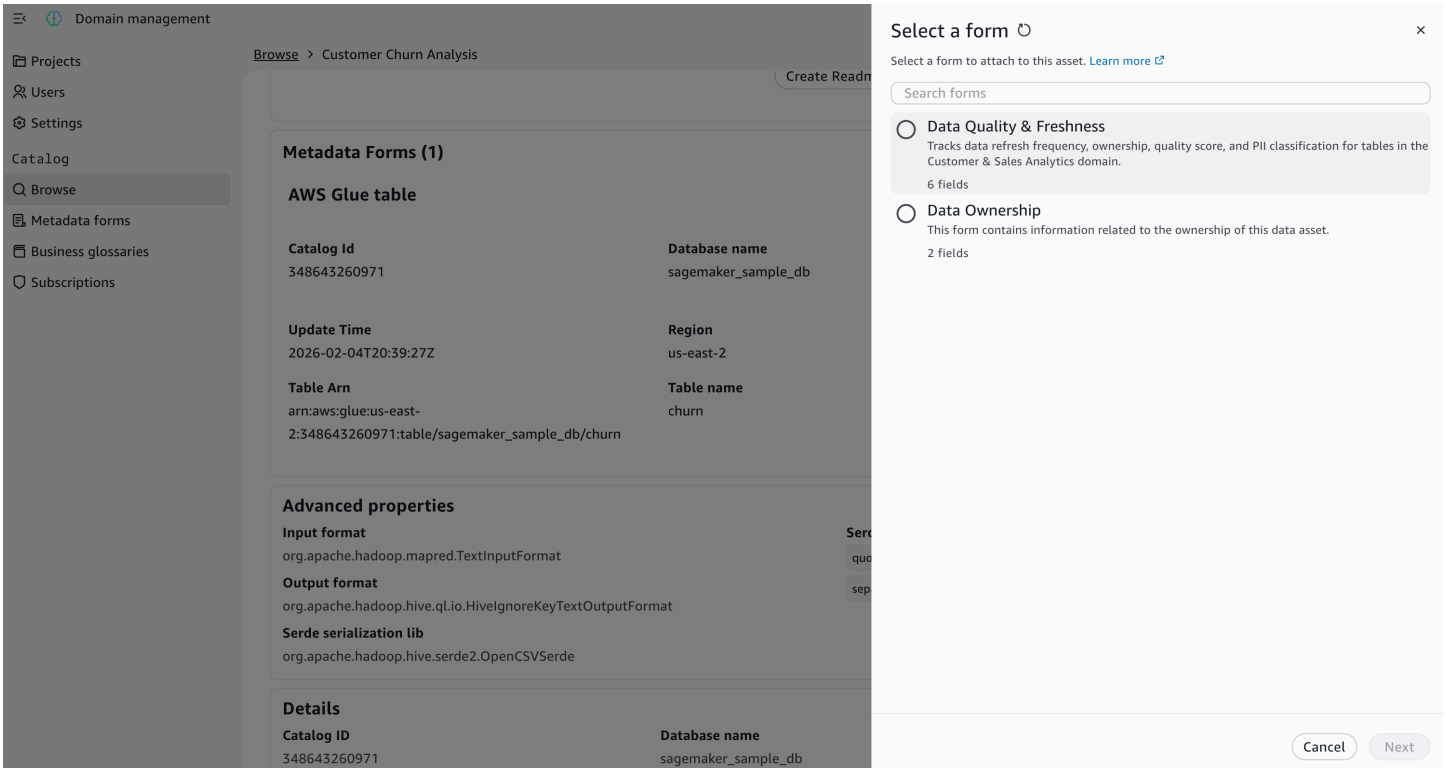
1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.

2. In the left navigation pane under **Catalog**, choose **Browse**.
3. Choose the asset you want to edit.
4. On the **Overview** tab, in the **Metadata Forms** section, choose **Add Form**. The **Select a form** panel opens.
5. Browse or search for the form you want to attach. Each form shows its name, description, and number of fields.
6. Select the radio button next to the form you want to attach.
7. Choose **Next**.
8. Fill in the form fields. Required fields are marked accordingly.
9. Choose **Save**.

The form appears in the **Metadata Forms** section on the **Overview** tab. You can expand or collapse individual forms, and use **Collapse all** to minimize all forms at once.

To manage how forms are displayed to users who find this asset in the catalog, choose the edit icon (pencil) next to the **Metadata Forms** section header. This opens the **Manage metadata forms** panel, where you can do the following:

- **Reorder forms** – Drag forms using the grip handle to change the order in which forms appear on the asset detail page. For example, if you want the "Data Ownership" form to appear before the " table" form, drag it above. Users who discover this asset in the catalog will see forms in the order you set here.
- **Show expanded** – Controls whether a form is expanded or collapsed when users view the asset detail page. When toggled on, users immediately see all the form's field names and values without choosing anything. When toggled off, users see only the form name as a collapsed header and must choose to expand it. Use this to highlight the most important forms while keeping less critical ones out of the way.
- **Remove a form** – Choose the delete icon (trash) to detach a form from the asset. This removes the form and all its field values from the asset. The form itself is not deleted and can be re-attached later.



Edit metadata form values

After a form is attached to a table, you can update its field values at any time.

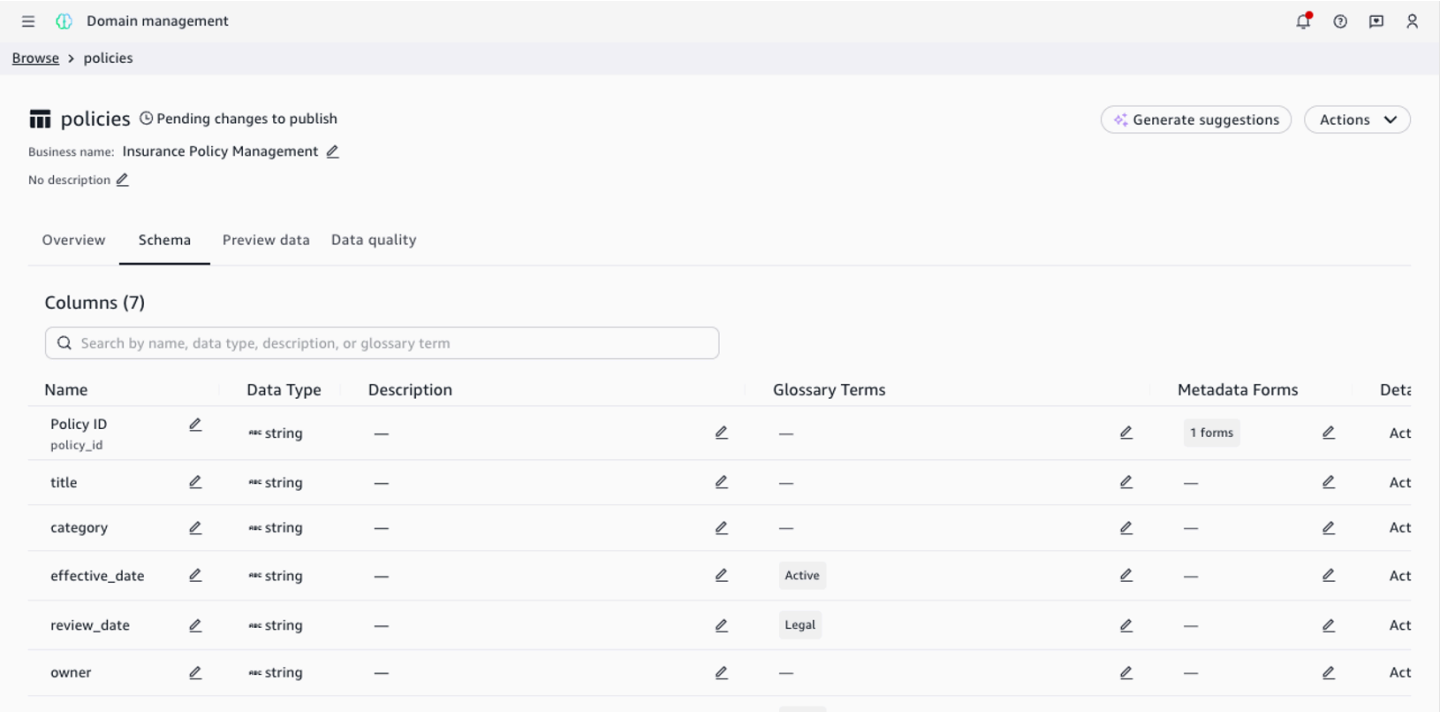
To edit metadata form values

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Browse**.
3. Choose the asset you want to edit.
4. On the **Overview** tab, find the form you want to edit in the **Metadata Forms** section.
5. Choose the edit icon (pencil) next to the form name.
6. Update the field values.
7. Choose **Save**.

Curate column metadata

The **Schema** tab on the asset detail page shows all columns in the table. Each column row has inline edit controls for the business name, description, glossary terms, and metadata forms. The

Actions dropdown on each row provides access to additional options including viewing column details and editing the column's README. This lets you add business context at the column level, which is especially useful for tables with cryptic column names or columns that need specific definitions.



Schema tab columns

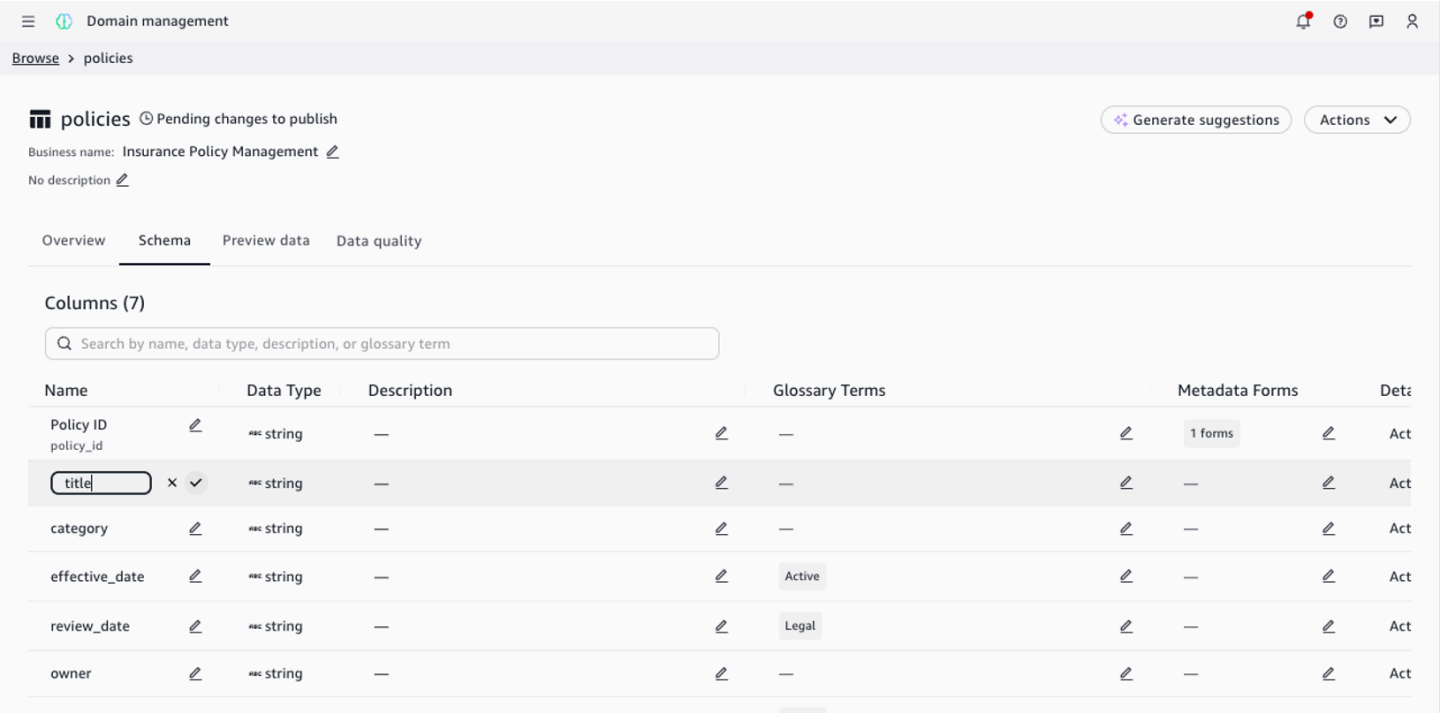
Column	What it shows
Name	The column's business name (editable) and technical name. The business name is what people see in the catalog. The technical name comes from the Glue Data Catalog.
Data Type	The column's data type from the Glue Data Catalog (for example, string, int, double).
Description	A text description of what the column contains (editable).
Glossary Terms	Business glossary terms associated with this column (editable).
Metadata Forms	Metadata forms attached to this column.

Column	What it shows
Actions	Opens a dropdown with **View Details** (to see all column metadata in a panel) and **Edit Readme** (to add or edit the column's README documentation).

Edit a column's business name

To edit a column's business name

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Browse**.
3. Choose the asset you want to edit.
4. Choose the **Schema** tab.
5. In the **Name** column, find the column you want to rename and choose the edit icon (pencil).
6. Enter the new business name in the text field. The technical name remains unchanged below it.
7. Choose the checkmark icon to save, or choose the cancel icon to discard.



To generate business names for all columns at once, see Use AI to generate metadata suggestions.

Edit a column's description

To edit a column's description

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Browse**.
3. Choose the asset you want to edit.
4. Choose the **Schema** tab.
5. In the **Description** column, find the column you want to edit and choose the edit icon (pencil).
6. Enter or update the description.
7. Choose the checkmark icon to save, or choose the cancel icon to discard.

Associate glossary terms with a column

To associate glossary terms with a column

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Browse**.
3. Choose the asset you want to edit.
4. Choose the **Schema** tab.
5. In the **Glossary Terms** column, find the column you want to edit and choose the glossary terms cell or the edit icon (pencil). The **Edit Glossary terms** panel opens.
6. Choose **Add terms**.
7. Search for the term you want to associate, or browse the available terms.
8. Select the checkbox next to one or more terms.
9. Choose **Add terms** to confirm your selection.

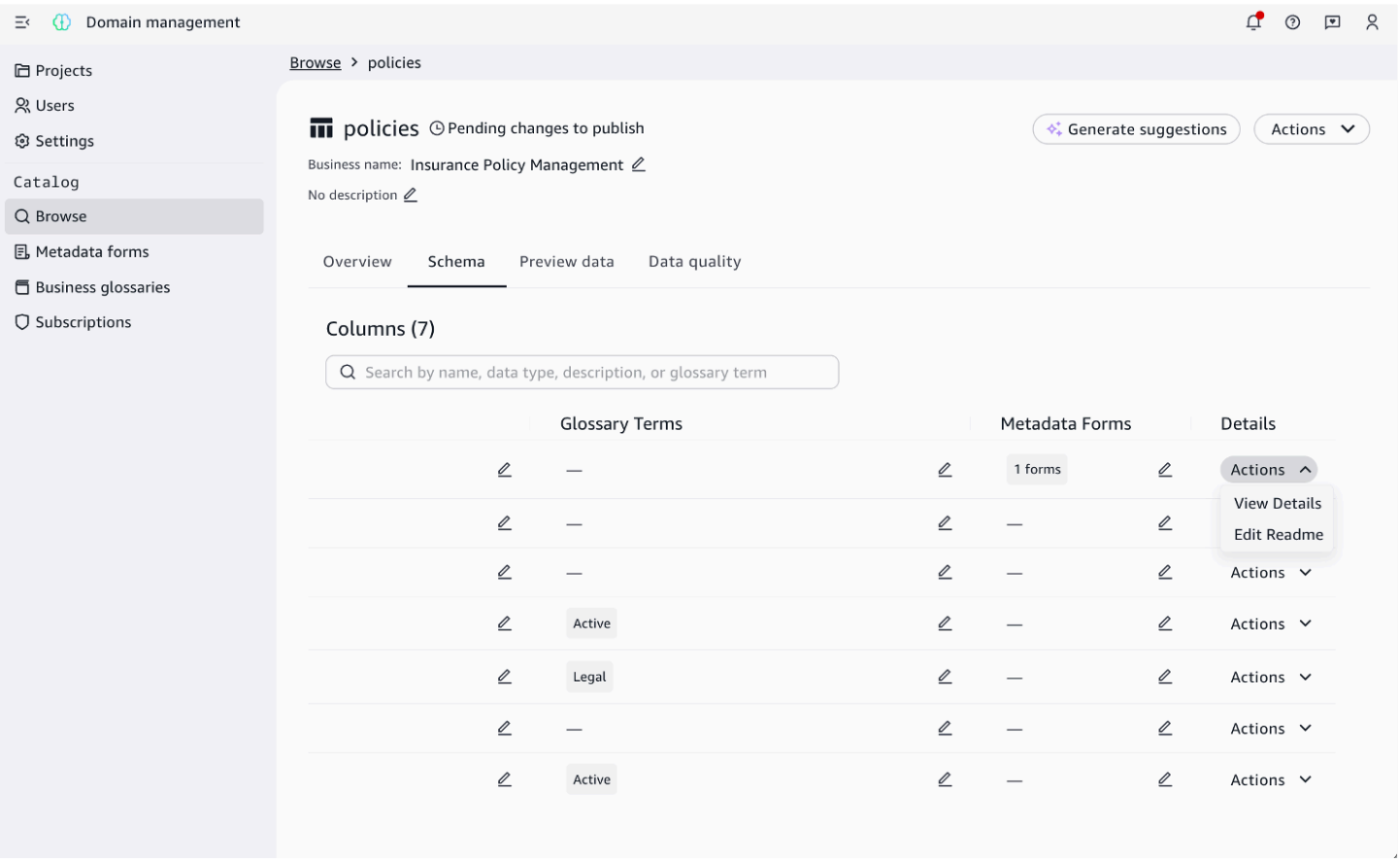
Attach a metadata form to a column

To attach a metadata form to a column

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Browse**.
3. Choose the asset you want to edit.
4. Choose the **Schema** tab.
5. In the **Metadata Forms** column, find the column you want and choose **Add metadata form**. The **Select a form** panel opens.
6. Browse or search for the form you want to attach.
7. Select the checkbox next to the form.
8. Choose **Next**.
9. Fill in the form fields.
10. Choose **Save**.

Column actions

Each column row on the **Schema** tab has an **Actions** dropdown in the last column. The dropdown provides two options.



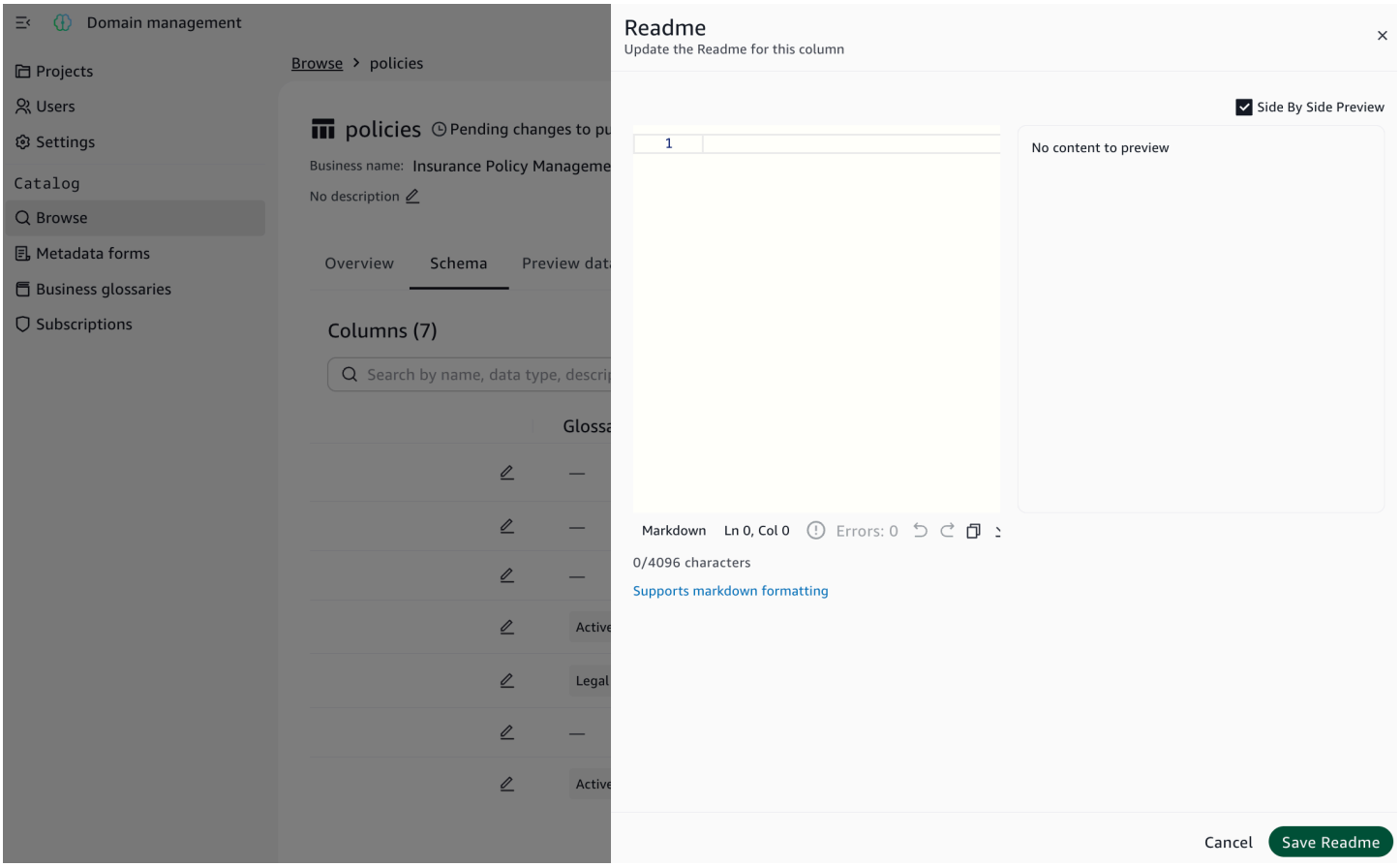
View Details opens a side panel showing all metadata for the column, including its business name, technical name, description, glossary terms, metadata forms, and README.

Edit Readme opens a modal where you can write or update the column's README documentation using Markdown. The editor includes a side-by-side preview, character count (0/4096), and toolbar with undo, redo, and copy controls.

To edit a column's README

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Browse**.
3. Choose the asset you want to edit.
4. Choose the **Schema** tab.
5. In the **Actions** column for the column you want to edit, choose the dropdown, then choose **Edit Readme**.
6. Enter or update the README content using Markdown formatting.

7. Choose **Save Readme**.



Use AI to generate metadata suggestions

Instead of manually writing business names and descriptions for every table and column, you can use AI to generate suggestions based on the table's technical metadata. This is especially useful when you have many tables to curate or when column names are abbreviated or cryptic.

AI can generate the following types of metadata.

- **Business name** for the table and each of its columns. For example, a column named `policy_id` might get the suggested business name "Policy ID," and a column named `effective_date` might become "Effective Date."
- **Summary and column descriptions** for the table and each of its columns. For columns, these are plain-language explanations of what the column contains. For the table itself, the generated description becomes the **Summary** on the **Overview** tab, a longer overview describing what the table contains and how its columns relate to each other. The summary can only be created through AI generation, but you can edit it afterward.

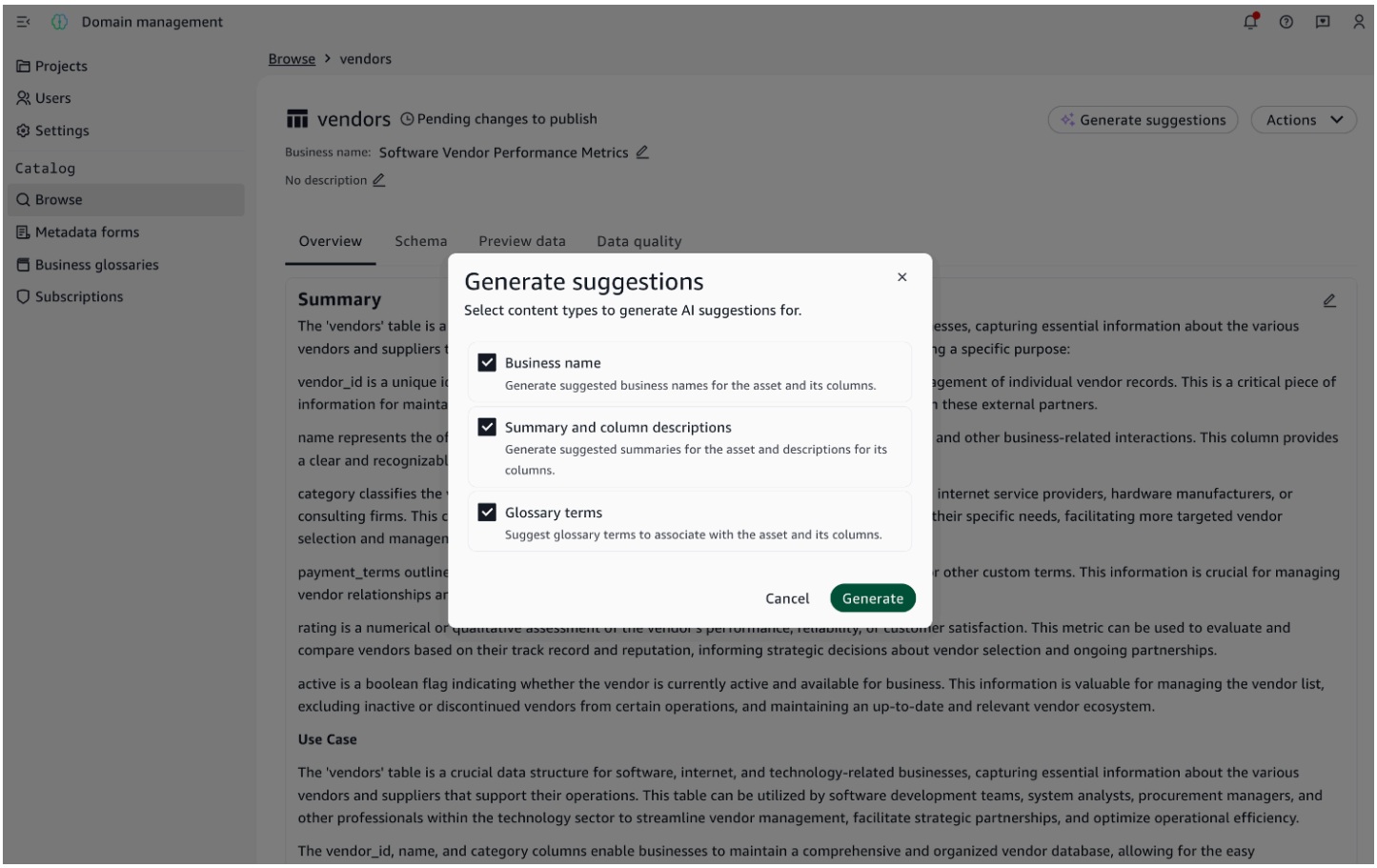
- **Glossary terms** for the table and its columns. The system suggests associations between columns and existing glossary terms based on the column's name and description. For example, a column named `category` might get a suggested association with the "Technology" glossary term.

Generate suggestions

To generate AI suggestions

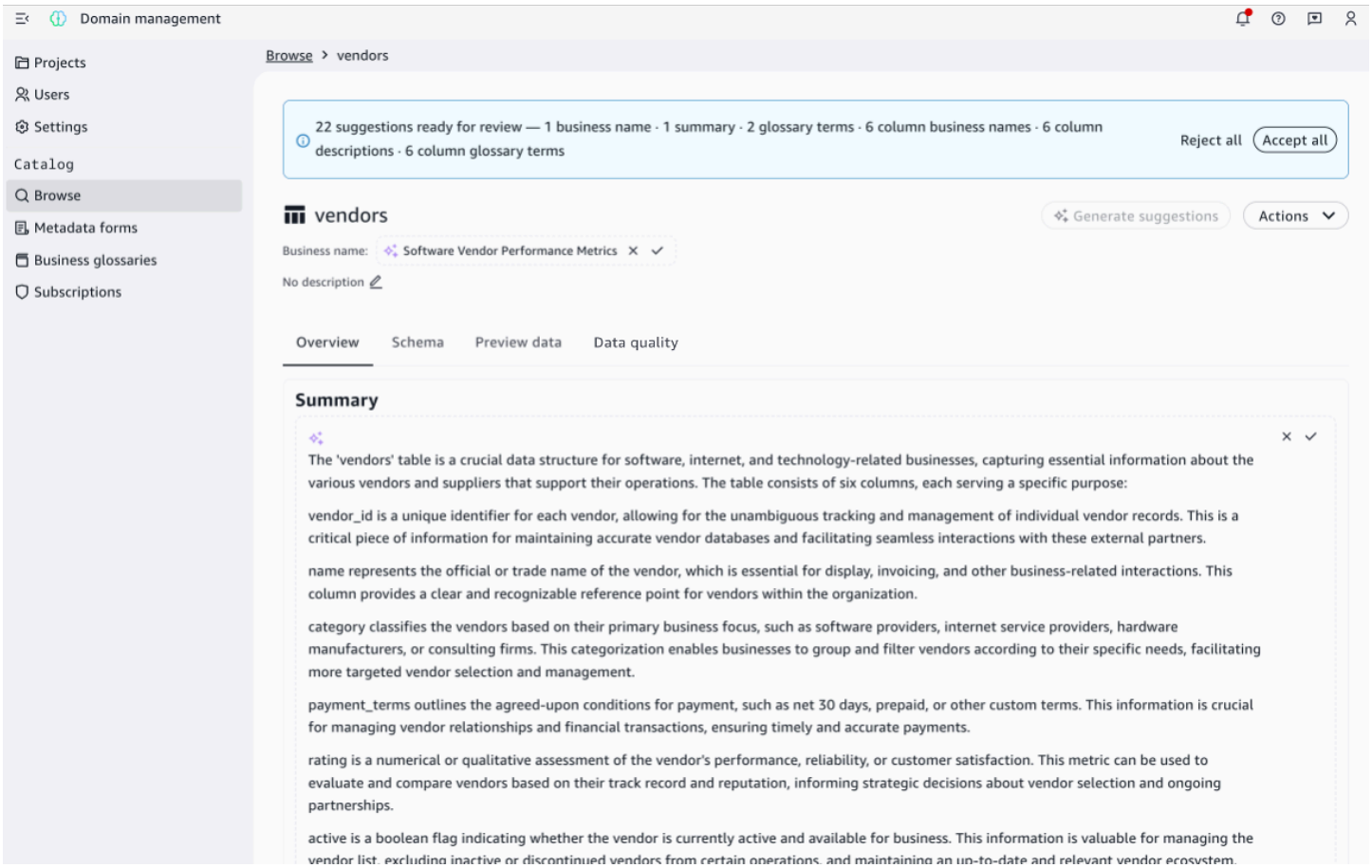
1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Browse**.
3. Choose the asset you want to curate.
4. Choose **Generate suggestions** in the top-right area of the page.
5. In the **Generate suggestions** dialog, select the content types you want to generate. The options are **Business name**, **Summary and column descriptions**, and **Glossary terms**. All three are selected by default. Deselect any types you do not want to generate.
6. Choose **Generate**.

The system analyzes the table's technical metadata and generates suggestions. This may take a minute or more depending on the number of columns.

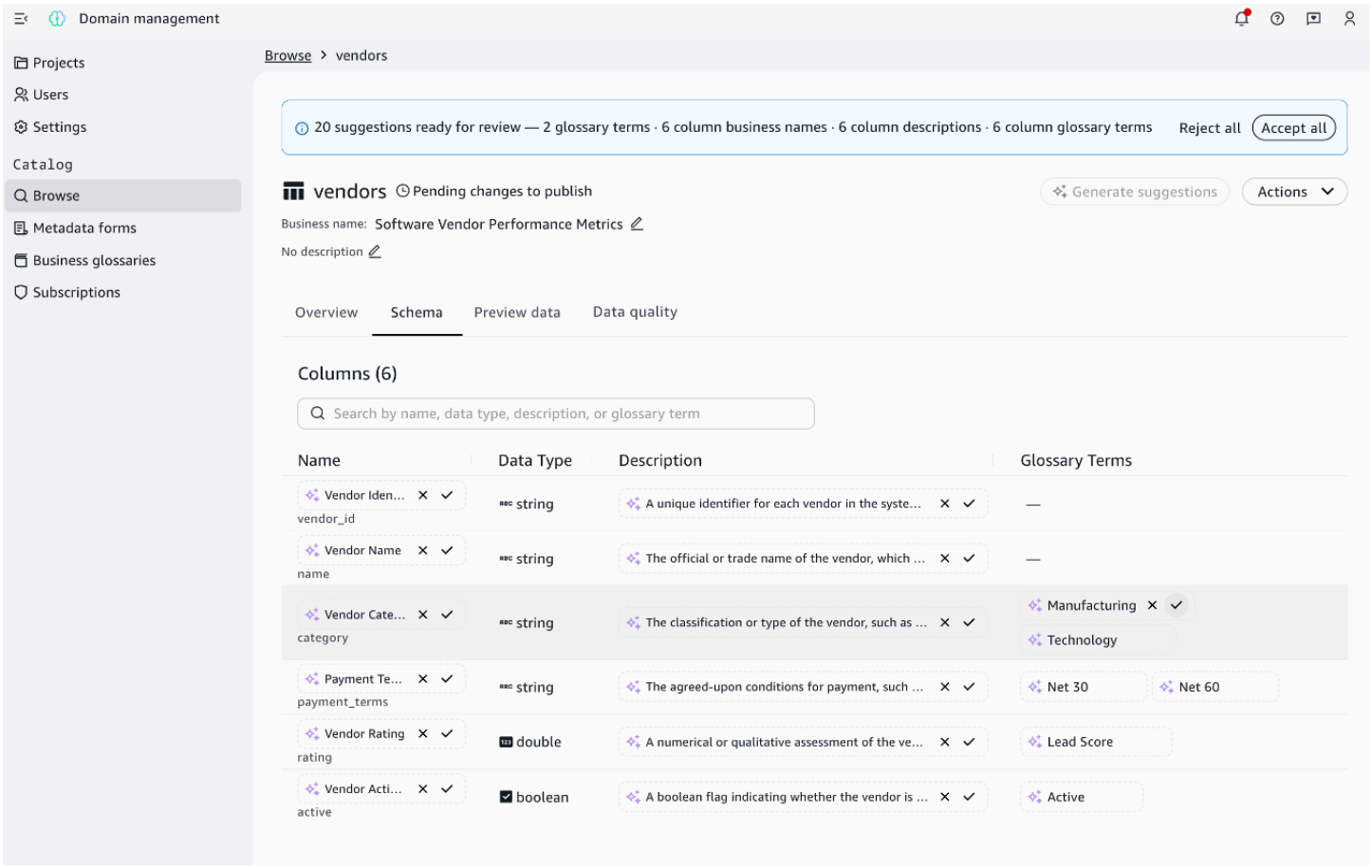


Review and accept suggestions

When generation completes, a banner appears on the asset detail page showing the total number of suggestions. The banner has **Reject all** and **Accept all** buttons for bulk actions.



You can also review suggestions individually. On the **Schema** tab, suggested glossary terms appear inline in the **Glossary Terms** column with a sparkle icon to distinguish them from existing terms. Each suggestion has its own reject and accept buttons, so you can accept some suggestions and reject others for the same column.



Business names and descriptions that are generated are applied directly as pending changes. You can edit them using the same inline edit controls described in the earlier sections before re-publishing.

Tip

AI-generated suggestions are a starting point. Review them carefully and adjust the language to match your organization's terminology and standards before publishing.

Important

If you publish an asset without accepting or rejecting its AI-generated suggestions, the unreviewed metadata is not included in the published version. Always review and accept suggestions before publishing to ensure the generated business names, descriptions, and glossary terms are visible to other users.

Quotas and region availability

AI-generated metadata is powered by Amazon Bedrock. The following monthly quotas apply per domain:

- Business name generation: 50,000 invocations/month
- Business description and summary generation: 10,000 invocations/month
- Glossary term generation: 10,000 invocations/month

Contact AWS Support to request a quota increase.

Publish data

After you curate an asset's metadata in the admin portal, you publish it to make those changes visible to the rest of the domain. Until you publish, your edits stay local and other users continue to see the previously published version.

Publishing is the step that makes an asset, along with all of its business metadata, appear in search results for every project in the domain. Once published, domain users can discover the asset, read its description and glossary terms, and request access to query the underlying data. Similarly, the published assets become searchable and discoverable in SageMaker Data Agent.

New assets that enter the catalog for the first time are published automatically. You only need to publish manually when you have made metadata changes (such as adding a business name, editing a description, attaching glossary terms, or filling in a metadata form) and want those changes to become visible. Until you publish, other users continue to see the previously published version of the asset's metadata. The asset detail page shows a "Pending changes to publish" indicator when you have unpublished edits.

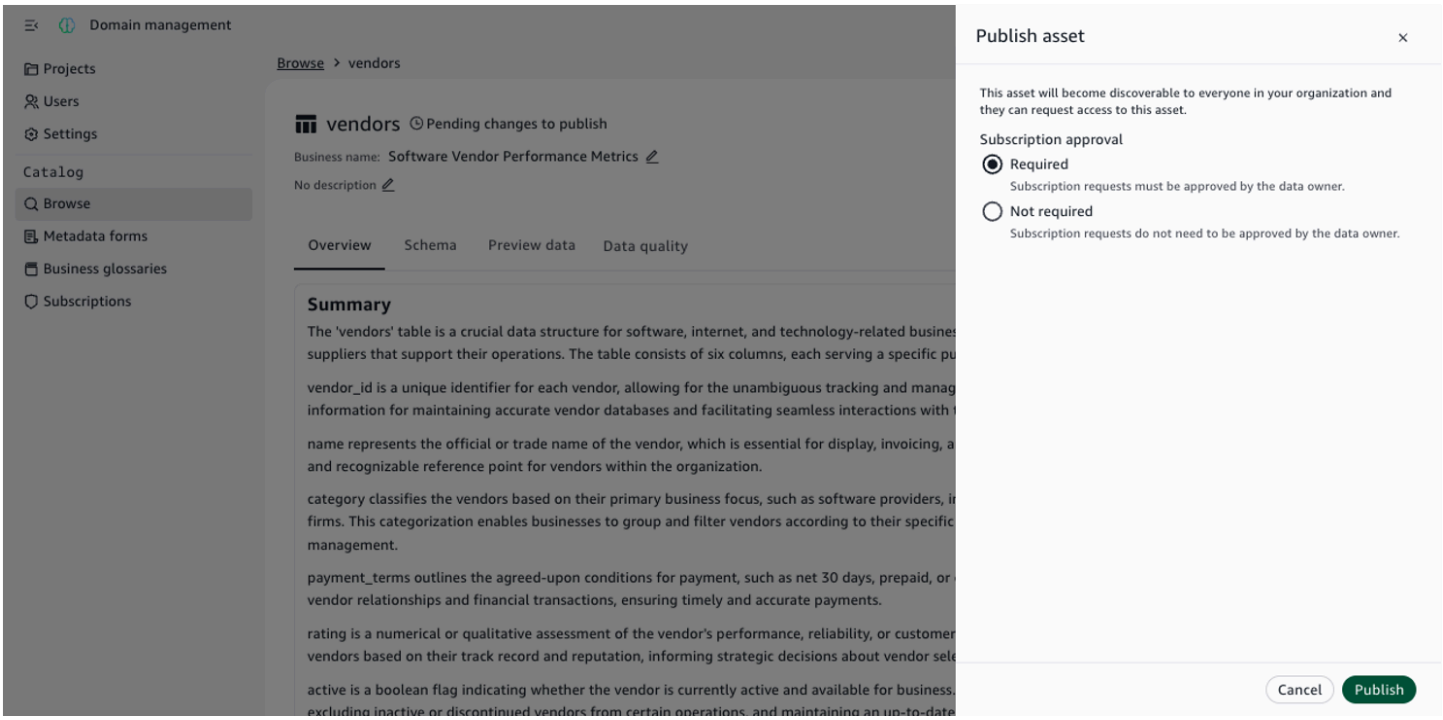
Publish an asset

To publish an asset

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Browse**.
3. Choose the asset you want to publish.
4. Choose **Actions**, then choose **Publish**.

5. In the **Publish asset** panel, under **Subscription approval**, select **Required** or **Not required**.
   - **Required** means subscription requests must be approved by the data owner before access is granted.
   - **Not required** means subscription requests are granted automatically without manual approval.
6. Choose **Publish**.

After publishing, the asset is discoverable to all projects in the domain. Project members can find it through search or the data explorer and request access.



> **Note**
>
> If you make additional metadata edits after publishing, those changes are not visible to other users until you publish again. Other users continue to see the last published version until you choose **Actions** and then **Publish**.

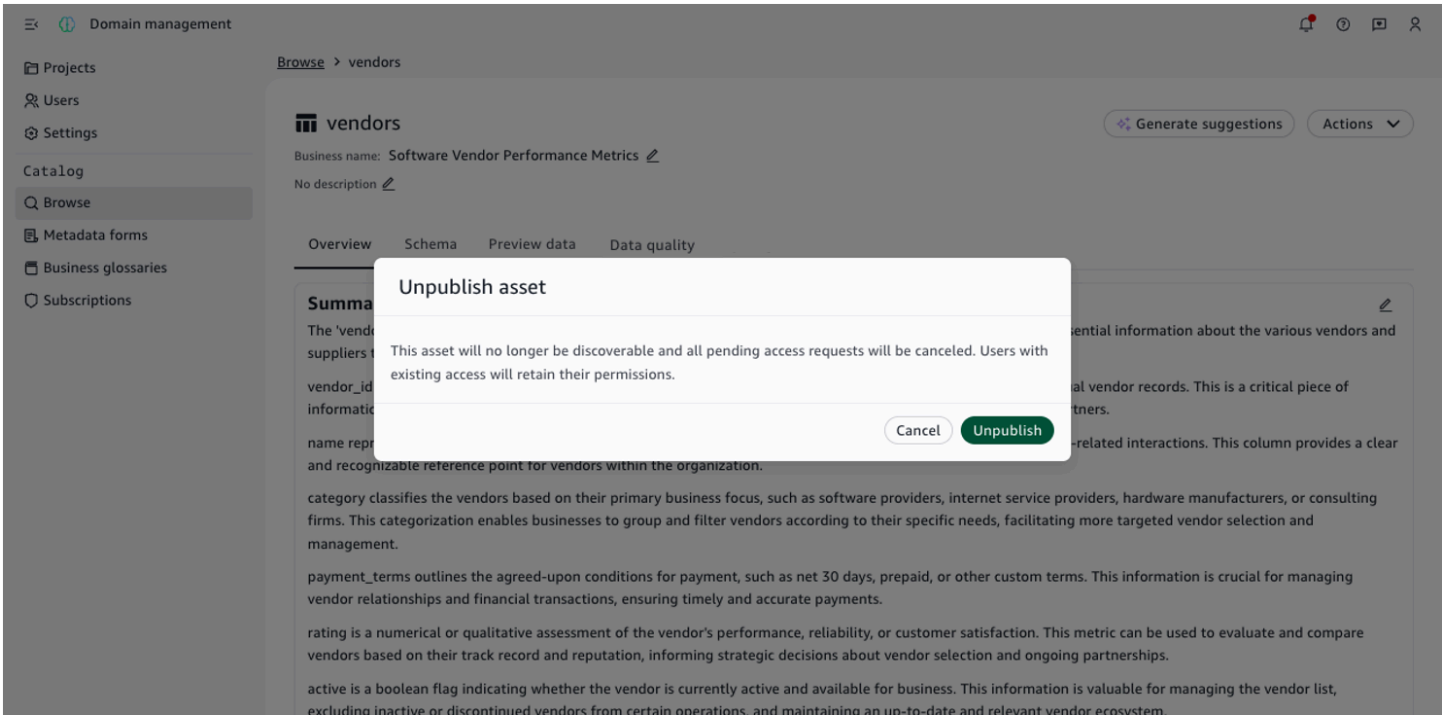
Unpublish an asset

Unpublishing removes an asset from search results across all projects in the domain. Other project members can no longer find it or request access to it. Users who already have access through

an approved subscription retain their existing permissions, but the asset no longer appears as discoverable.

To unpublish an asset

1. Navigate to Amazon SageMaker Unified Studio and choose **Domain management** to open the admin portal.
2. In the left navigation pane under **Catalog**, choose **Browse**.
3. Choose the asset you want to unpublish.
4. Choose **Actions**, then choose **Unpublish**.
5. In the confirmation dialog, choose **Unpublish**.



Custom asset types

In Amazon SageMaker Unified Studio, assets represent specific types of data resources such as database tables, dashboards, or machine learning models. To provide consistency and standardization when describing catalog assets, an Amazon SageMaker Unified Studio domain must have a set of asset types that define how assets are represented in the catalog.

An asset type defines the schema for a specific type of asset. It has a set of required and optional metadata form types. Asset types in Amazon SageMaker Unified Studio are versioned. When assets

are created, they are validated against the schema defined by their asset type, and if an invalid structure is specified, asset creation fails.

Custom asset types allow domain administrators to catalog any format of asset within the Amazon SageMaker Unified Studio catalog. While you must bring your own subscription fulfillment process, custom asset types enable you to leverage the full metadata curation, discovery, and subscription flows offered by Amazon SageMaker Unified Studio.

Use cases for custom asset types include:

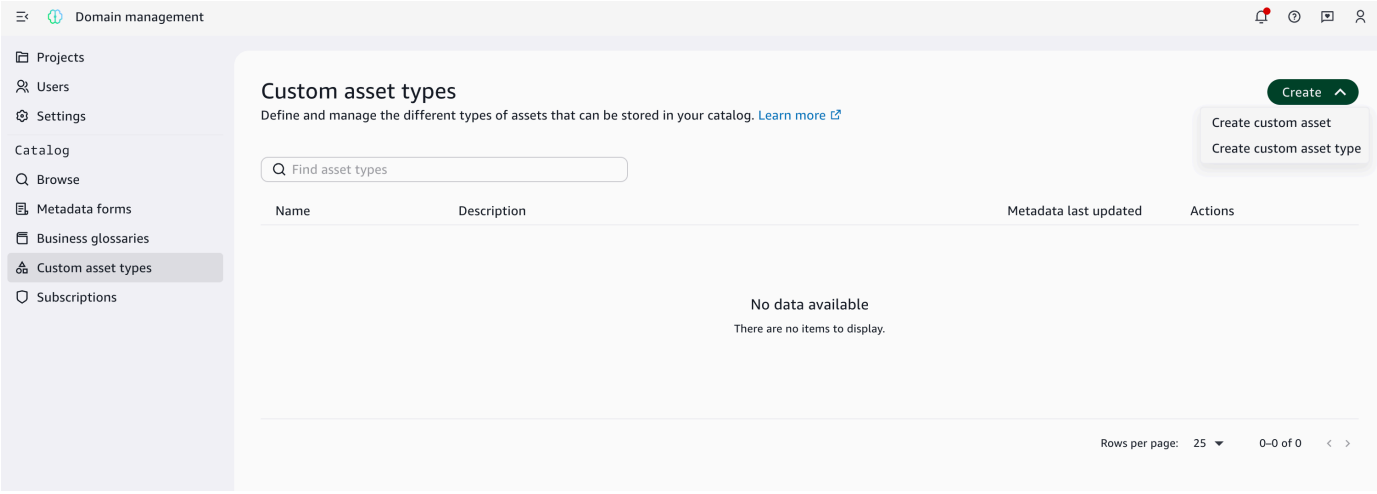
- Cataloging proprietary data formats that are not natively supported by Amazon SageMaker Unified Studio (for example, internal reporting systems, custom file formats, or third-party platform exports)
- Providing a unified discovery experience across all organizational assets regardless of their underlying format
- Enabling governed subscription workflows for assets where fulfillment is managed outside of Amazon SageMaker Unified Studio

Creating a custom asset type

Domain administrators can create custom asset types to represent any kind of asset in the catalog. You start by specifying a name, description, and the associated metadata forms that can be required or optional.

To create a custom asset type

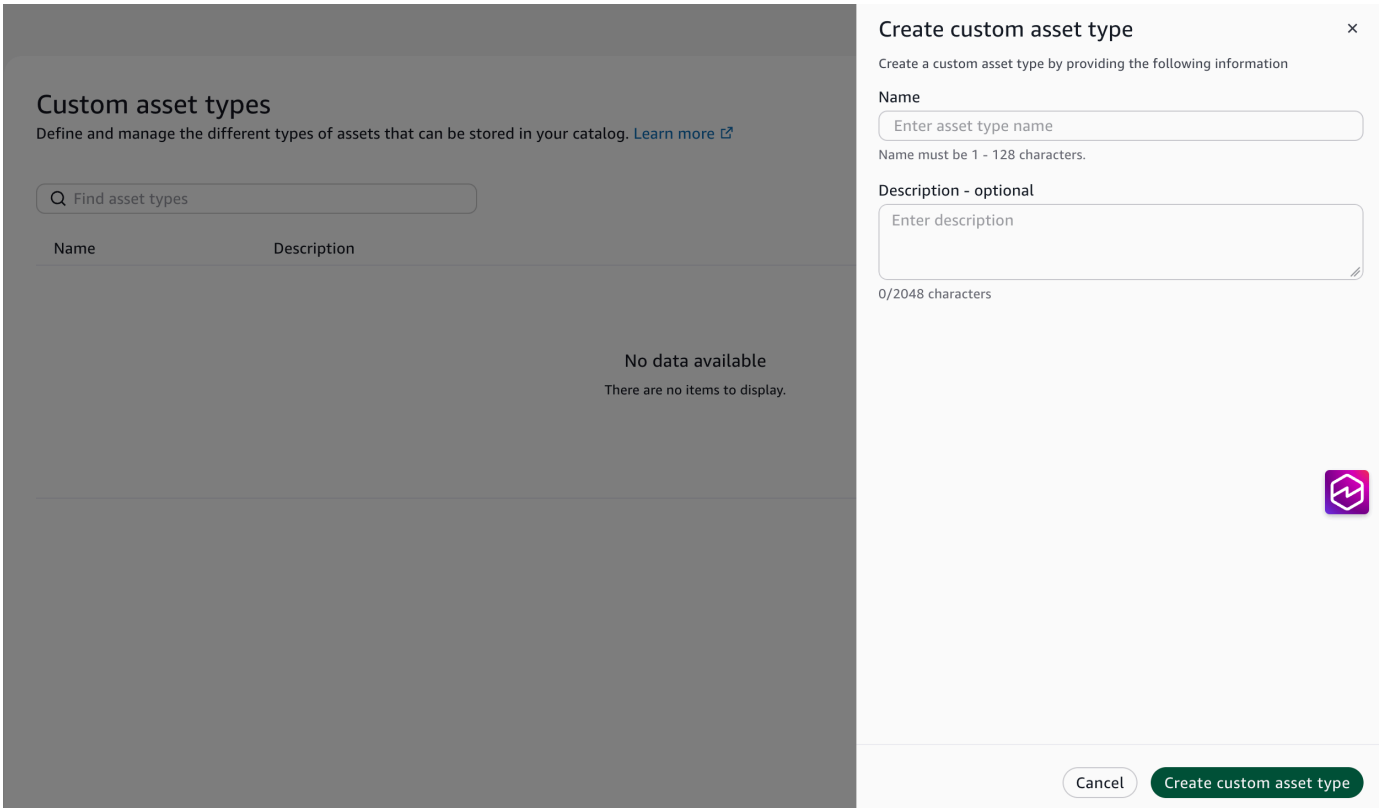
1. Navigate to **Domain management** > **Custom asset types** in the left navigation bar.
2. Choose the **Create** dropdown in the top-right corner.
3. Choose **Create custom asset type**.



4. Specify the following:

   - **Name** — Enter a unique name for the asset type (1–128 characters). This name is immutable after creation.
   - (Optional) **Description** — Enter a brief description of the asset type (up to 2048 characters).

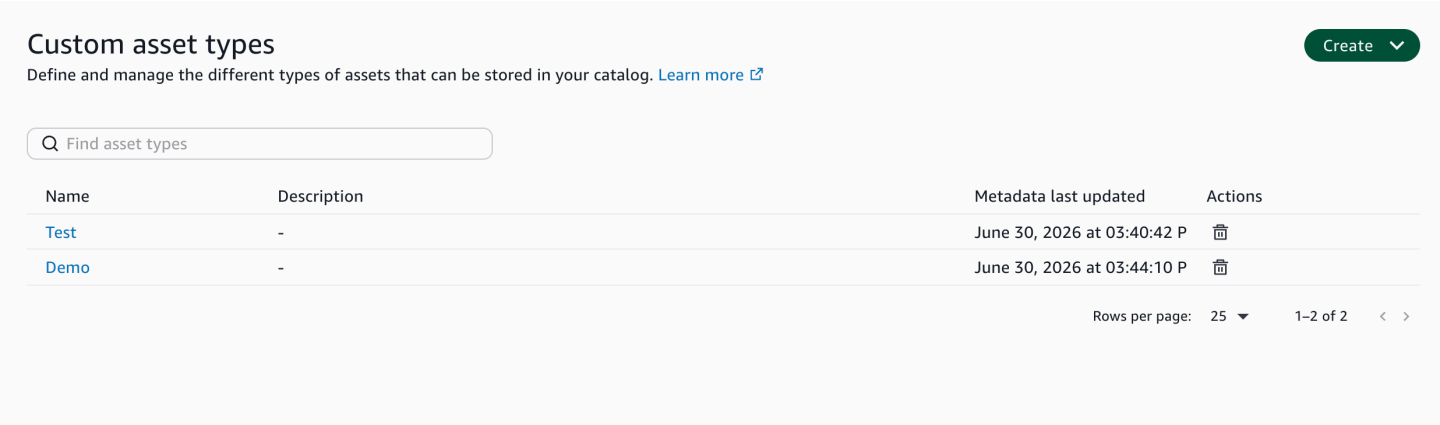
5. Choose **Create custom asset type**.



After the custom asset type is created, it appears in the list view and you can use it to create assets. As an optional step, you can choose **Add metadata forms** to attach metadata forms to this custom asset type. You can mark forms as required or optional. For more information about metadata forms, see Standardize metadata using metadata forms.

Viewing all custom asset types

The list view displays all custom asset types defined in your domain.



The table includes the following columns:

Custom asset types list columns

Column	Description
Name	The unique identifier for the asset type. Choose the name to open the detail view.
Description	A brief summary of what the asset type represents.
Metadata last updated	Timestamp of the most recent metadata change.
Actions	A delete icon. This is only available if no custom assets have been created from this type.

You can search for specific asset types using the search bar above the table. The **Create** dropdown in the top-right also provides a shortcut to create a custom asset directly from this page.

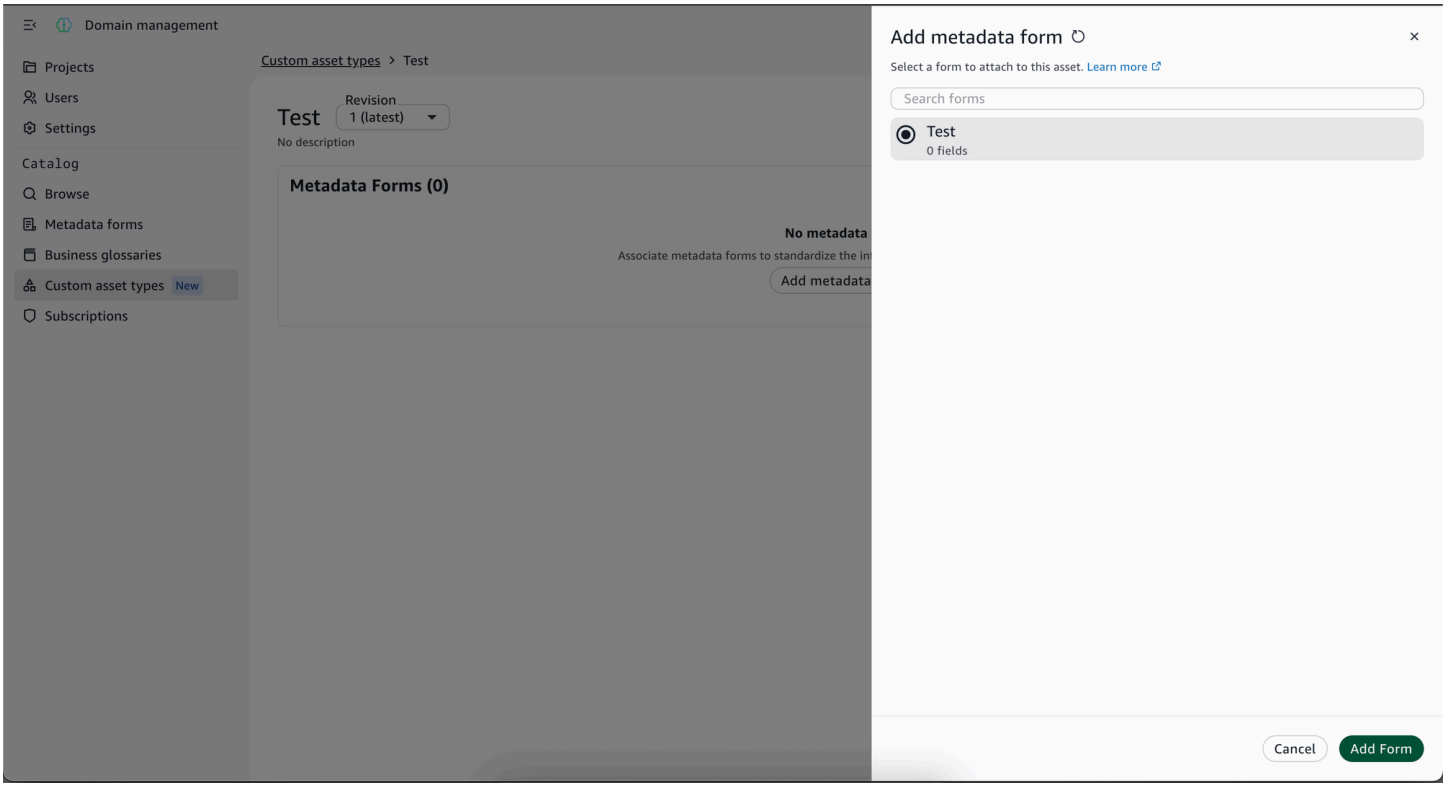
Viewing custom asset type details

Choose any asset type name in the list to open its detail view.

The detail view includes the following information:

- **Revisions** — View prior versions of the asset type using the revision dropdown. You can always view historical revisions but only create new ones. Any update operation creates a new revision.
- **Description** — An editable description of the asset type.
- **Name** — Displayed but immutable after creation.
- **Metadata forms** — A list of all attached forms with their field definitions. Each form shows an **Add metadata forms** button to attach additional forms from your domain.

To add a metadata form, choose **Add metadata forms**. In the drawer, select a form from the list or use the search bar to find one.



If no metadata forms exist in your domain yet, you can choose **Create Form** from within the drawer to create one. For more information, see Standardize metadata using metadata forms.

Deleting a custom asset type

A custom asset type can only be deleted if no custom assets have been created from it. If assets exist that use this type, the delete action is unavailable.

To delete a custom asset type

1. Navigate to the custom asset types list view.
2. In the **Actions** column, choose the delete icon next to the asset type you want to remove.

   Alternatively, from the asset type detail page, choose **Actions**, and then choose **Delete**.

> **Note**
>
> You must delete all associated custom assets before removing the type itself. If the delete icon is grayed out, one or more custom assets still reference this type.

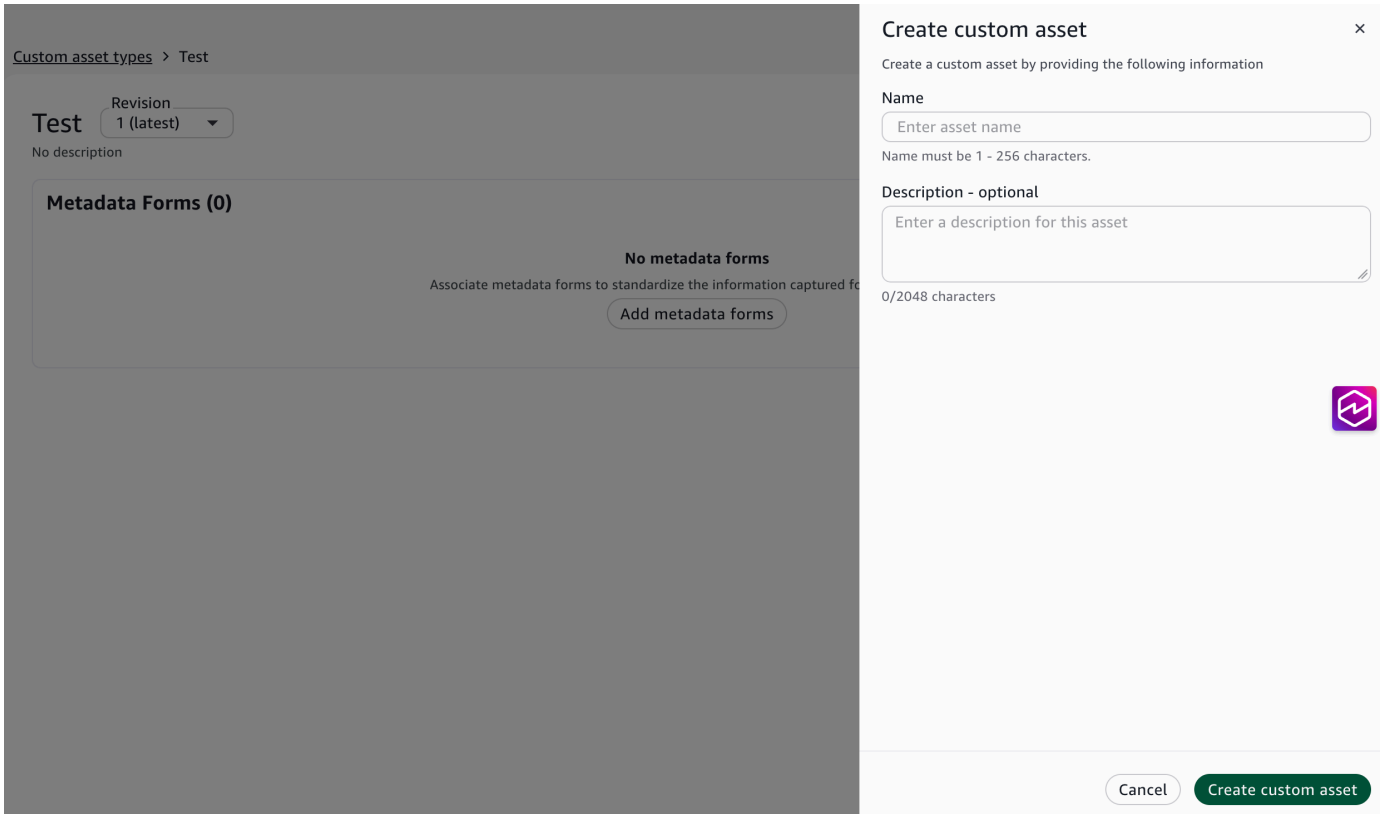
Creating a custom asset

After you create a custom asset type, you can create individual custom assets from it.

To create a custom asset

1. Navigate to the detail view of the desired custom asset type.
2. Choose the **Create custom asset** button in the top-right (available from the **Create** dropdown).
3. Configure the following settings:

   - **Name** — A unique name for the asset (1–256 characters). This field is required.
   - (Optional) **Description** — A description of the asset (up to 2048 characters).



4. Choose **Create custom asset**.

After creation, the asset is stored in a draft state. You can further enrich it before publishing to make it discoverable in the catalog.

Enriching a custom asset

After you create a custom asset, you can enrich it with additional context to improve discoverability and documentation. The enrichment options for custom assets are the same as those for other catalog assets. For more information about curating and enriching asset metadata, see Curate and enrich asset metadata.

After you enrich the asset, publish it to make it discoverable in the catalog. For more information, see Publish data.

Discovering custom assets in the catalog

After publishing, custom assets become discoverable through the Amazon SageMaker Unified Studio catalog search. Users can find custom assets by:

- **Searching** — Enter a name or description in the search bar to find matching assets.
- **Filtering by type** — Select the custom asset type in the left-hand filter panel to narrow results to a specific type.
- **Filtering by glossary terms** — Use tagged glossary terms to discover related assets.
- **Filtering by date** — Filter by metadata last updated or creation date.

Search results display the asset name, last updated date, asset type, description snippet, and associated glossary terms.

For more information about searching and discovering assets, see Search and discover data.

Requesting a subscription to a custom asset

When a user finds a custom asset they need access to, they can request a subscription. The subscription request process for custom assets follows the same workflow as other catalog assets. For detailed steps, see Request access to data.

> **Note**
>
> Because custom asset types use an external fulfillment process, the subscription grants discovery and metadata access through Amazon SageMaker Unified Studio. The actual data access and delivery is managed by the asset owner's own fulfillment mechanism. This differs from standard table subscriptions, where Lake Formation permissions are granted automatically upon approval.

Evaluate data quality in the data catalog

Use the **Data quality** tab on any catalog table to define rules, run them on demand or on a schedule, and track quality scores over time without building a pipeline. This lets you monitor the quality of your data at rest directly from the data catalog.

Rules are written using DQDL (Data Quality Definition Language), a domain-specific language for defining data quality rules, with 31 built-in rule types. For the full list of rule types and syntax, see DQDL rule types in the AWS Glue documentation.

Create rulesets

To create a ruleset, first navigate to the **Data quality** tab for a catalog table, then define your rules using DQDL syntax.

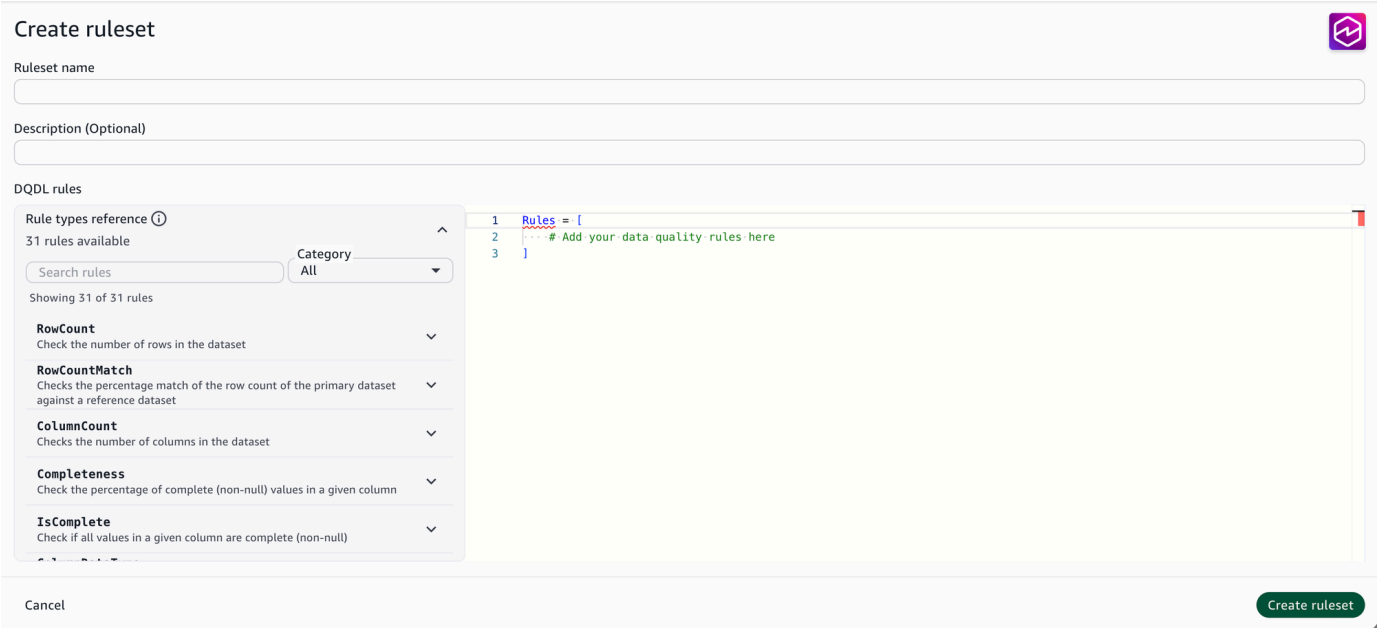
Navigate to the Data quality tab

1. In the left navigation pane, choose **Data**.
2. Under the **Catalogs** tab, expand a catalog (for example, AwsDataCatalog), then expand the database (for example, sagemaker_sample_db).
3. Select the table you want to evaluate (for example, churn).
4. On the table detail page, choose the **Data quality** tab.

Create a ruleset

When you first open the **Data quality** tab on a table, it shows an empty state with the option to create your first ruleset.

1. Choose **Create**.
2. The **Create ruleset** page opens with the following fields: **Ruleset name**, **Description** (Optional), and the DQDL rules editor.



3. Enter a name for your ruleset. The name identifies this set of rules in run results, the dashboard, and scheduled evaluations. Choose a descriptive name that reflects what the rules check (for example, `completeness-checks` or `value-range-validation`).
4. In the DQDL rules section, use the code editor to define your rules. A **Rule types** reference panel lists all 31 available rule types. You can search rules by name and filter by category. Expand any rule type to see its description and syntax.
5. After defining your rules, choose **Create ruleset**.

Write rules using DQDL

Define rules in the code editor using DQDL syntax. The editor provides autocomplete for rule types and column names from the table schema.

> **Tip**
>
> Use Ctrl+Tab (or Cmd+Tab on Mac) to trigger column name suggestions.

The following example shows a ruleset with two rules.

```
Rules = [
    RowCount > 1000,
    Completeness "account_length" between 0.1 and 0.25
```

```
]
```

Common rule types include:

Rule type	What it checks
RowCount	Total row count meets a threshold.
Completeness	Percentage of non-null values in a column.
ColumnExists	A specific column is present in the table.
ColumnLength	String column values meet a length condition.
IsUnique	All values in a column are unique.
IsComplete	A column has no null values.
ColumnValues	Column values satisfy a condition (for example, greater than 0, in a set).
CustomSql	A custom SQL expression evaluates to true.

You can also use referential integrity rules to validate relationships between columns in different tables. For details on syntax and examples, see ReferentialIntegrity in the AWS Glue documentation.

Note

Referential integrity and other cross-table rule types are supported only for tables under the default AWS Glue Data Catalog, using the format `<database>.<table>`.

For the complete reference of all 31 rule types, expressions, composite rules, and syntax, see DQDL rule types in the AWS Glue documentation.

Migrate existing rulesets to your project

If you have existing rulesets in AWS Glue that you want to view and manage within your Amazon SageMaker Unified Studio project, tag them so the project can discover them.

1. Add the following tag to your existing ruleset:
   - Key: `AmazonDataZoneProject`
   - Value: Your project ID (found in the project settings page in Amazon SageMaker Unified Studio)
2. Ensure your project role has adequate permissions to access the underlying data, including access to the Amazon S3 bucket where the table data is stored and appropriate Lake Formation grants for the table.

Once tagged, the ruleset appears on the **Data quality** tab for the corresponding table within your project. You can then run, schedule, and view results the same way as rulesets created directly in Amazon SageMaker Unified Studio.

Run rulesets and view results

After you create a ruleset, you can run it on demand and view the results on the **Data quality** tab.

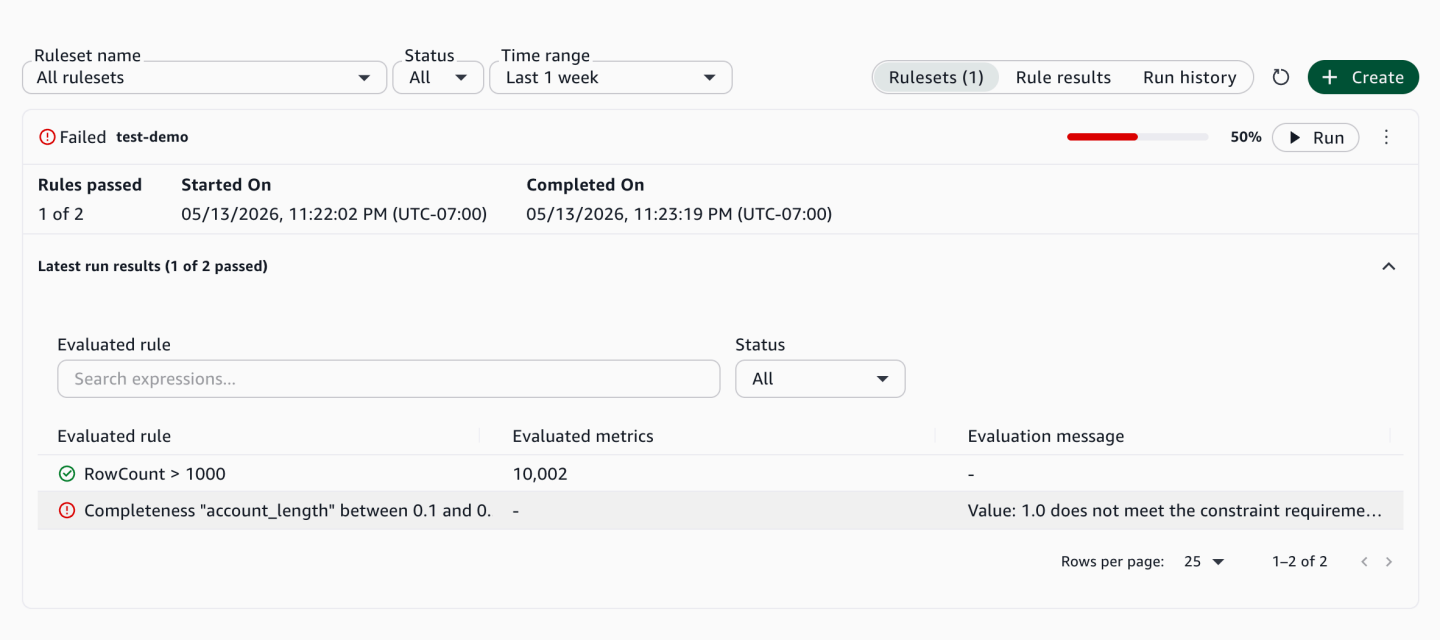
Run a ruleset

1. On the **Rulesets** sub-tab, find the ruleset you want to evaluate.
2. Choose **Run** to evaluate the rules against the current table data.
3. When the run completes, the ruleset card updates with the results showing which rules passed and which failed.

View run results

After a run completes, the ruleset card shows the status (Passed or Failed), score (percentage of rules that passed), rules passed (for example, "1 of 2"), and Started On and Completed On timestamps.

Expand the ruleset card to see a detailed table with each evaluated rule.



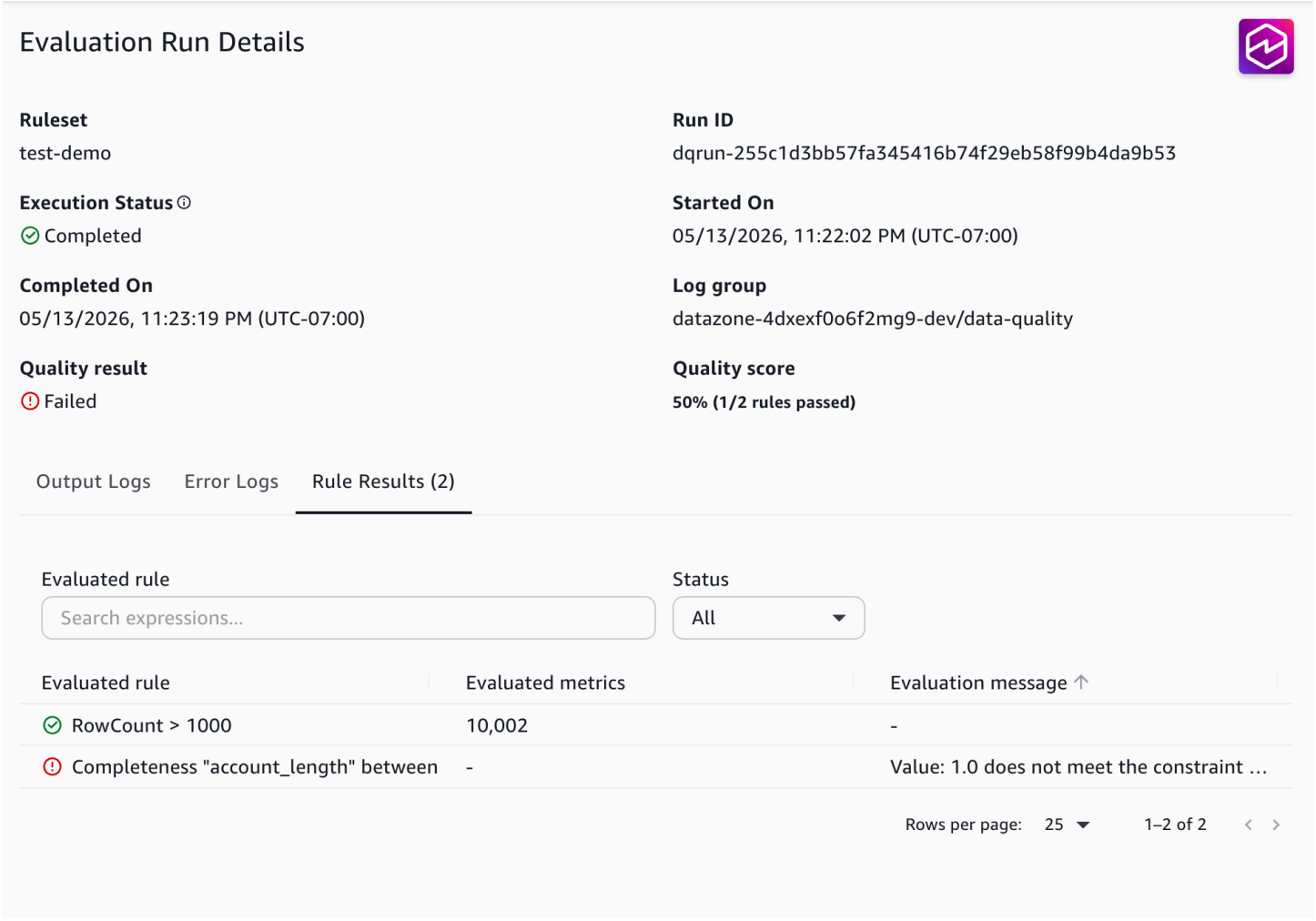
The results table shows:

Column	Description
Evaluated rule	The DQDL rule expression that was evaluated.
Evaluated metrics	The measured value (for example, row count of 10,002).
Evaluation message	Details on why a rule failed.

You can search and filter the results using the search box and status filter above the table.

View evaluation run details

Select a run from the **Rulesets** sub-tab or **Run history** sub-tab to open the Evaluation Run Details page.



The details page shows:

Field	Description
Ruleset	The ruleset that was evaluated.
Run ID	Unique identifier for this run.
Execution Status	Whether the run completed successfully.
Started On	Timestamp when the evaluation began.
Completed On	Timestamp when the evaluation finished.
Log group	CloudWatch Logs log group for detailed execution logs.

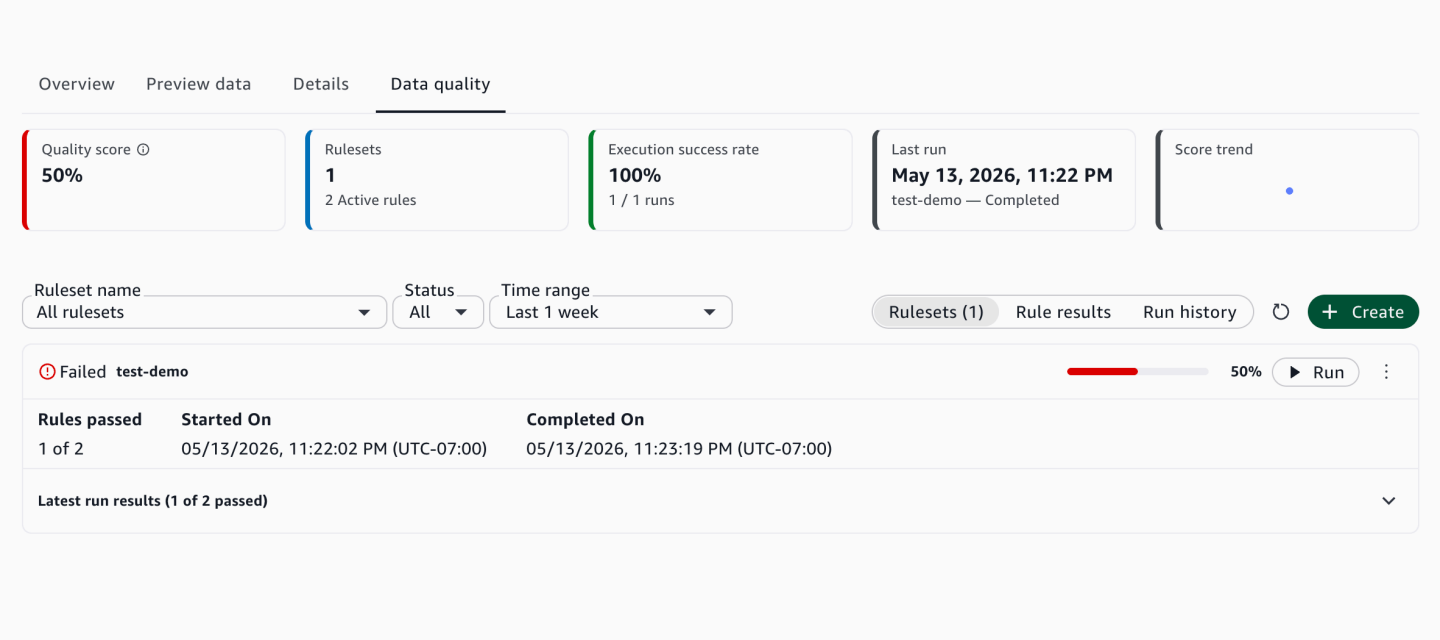
Field	Description
Quality result	Overall result (Passed or Failed).
Quality score	Percentage of rules that passed.

Below the summary, three tabs provide additional detail:

- **Output Logs**: Standard output from the evaluation run.
- **Error Logs**: Any errors that occurred during the run.
- **Rule Results**: A table showing each evaluated rule, its measured metrics, and evaluation messages.

Review the dashboard

Once you have run at least one ruleset, the **Data quality** tab shows a dashboard with summary metrics at the top, followed by three sub-tabs: **Rulesets**, **Rule results**, and **Run history**.



The dashboard tiles show:

Tile	Description
Quality score	The average of the latest quality scores per ruleset.
Rulesets	Number of rulesets and total active rules across all sets on this table.
Execution success rate	Percentage of runs that completed without errors.
Last run	Timestamp, ruleset name, and status of the most recent evaluation.
Score trend	Quality score over time, showing how data quality changes between runs.

Use the filters above the sub-tabs to focus results:

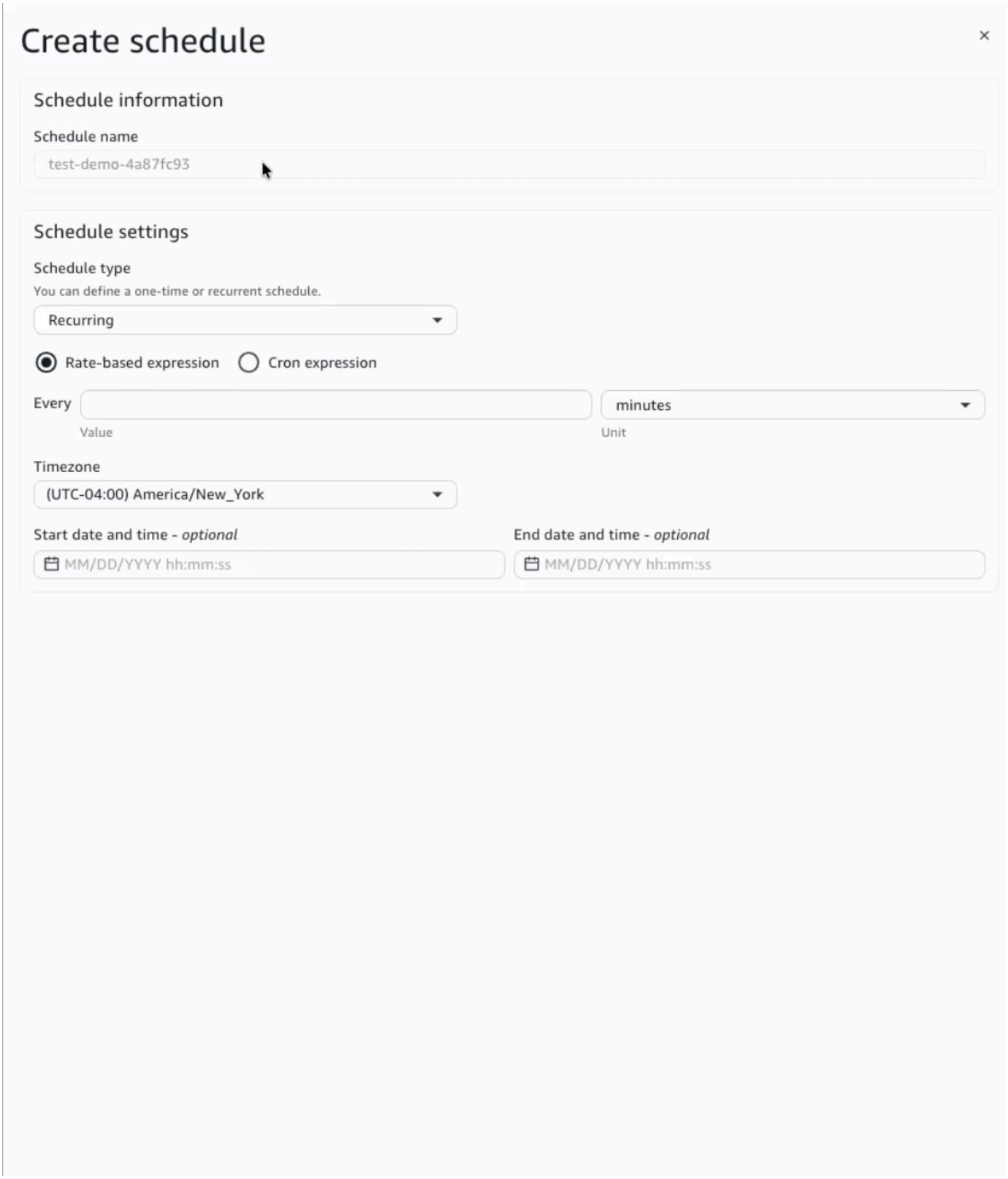
- **Filter by ruleset**: View results for a specific ruleset when you have multiple.
- **Status**: Filter by All, Passed, or Failed.
- **Time range**: Filter by relative time window or absolute date range.

Schedule evaluations

Instead of running rules manually, you can schedule them to run automatically.

To schedule a ruleset evaluation

1. Choose the **Actions** menu on your ruleset and select **Schedule run**.
2. Configure the schedule.


Create schedule
Schedule information
Schedule name
test-demo-4a87fc93
Schedule settings
Schedule type
You can define a one-time or recurrent schedule.
Recurring
Rate-based expression
Cron expression
Every
minutes
Value
Unit
Timezone
(UTC-04:00) America/New_York
Start date and time - optional
MM/DD/YYYY hh:mm:ss
End date and time - optional
MM/DD/YYYY hh:mm:ss

Setting	Description
Schedule name	Auto-generated based on the ruleset name and the catalog database table.
Schedule type	Recurring or One time.
Frequency (rate-based)	Simple interval expressions (for example, every 5 minutes, every 3 hours).
Frequency (cron expression)	Fine-grained scheduling using cron syntax.
Start date	Optional. When the schedule becomes active.
End date	Optional. When the schedule stops.

3. Choose **Create schedule**.

When scheduled runs execute, results appear automatically in the run history and update the dashboard.

Work with multiple rulesets

You can create multiple rulesets on the same table. Each can focus on different aspects of quality:

- **Completeness checks**: Rules that validate non-null values and column existence.
- **Value range validation**: Rules that check numeric or string constraints.
- **Freshness checks**: Rules that validate data recency.
- **Referential integrity**: Rules that verify relationships across tables.

Each ruleset can have its own schedule and produces independent results. Use the **Filter by ruleset** dropdown to view results for a specific set, or view all rulesets together.

Data lineage in Amazon SageMaker Unified Studio

Data lineage in Amazon SageMaker Unified Studio is an OpenLineage-compatible feature that can help you to capture and visualize lineage events, from OpenLineage-enabled systems or through APIs, to trace data origins, track transformations, and view cross-organizational data consumption. It provides you with an overarching view into your data assets to see the origin of assets and their chain of connections. The lineage data includes information on the activities inside the Amazon SageMaker Catalog, including information about the catalogued assets, the subscribers of those assets, and the activities that happen outside the business data catalog captured programmatically using the APIs.

Topics

- What is OpenLineage?
- Data lineage support
- Data lineage support matrix
- Visualizing data lineage in IAM mode
- Visualizing data lineage in IdC mode
- Test drive data lineage
- Data lineage authorization
- Automate lineage capture from data connections
- Automate lineage capture from tools
- Permissions required for data lineage
- Publishing data lineage programmatically
- The importance of the sourceIdentifier attribute to lineage nodes
- Linking dataset lineage nodes with assets imported into Amazon SageMaker Unified Studio
- Troubleshooting data lineage

What is OpenLineage?

OpenLineage is an open platform for collection and analysis of data lineage. It tracks metadata about datasets, jobs, and runs, giving users the information required to identify the root cause of

complex issues and understand the impact of changes. It is an Open Standard for lineage metadata collection designed to record metadata for a job in execution.

The standard defines a generic model of dataset, job, and run entities uniquely identified using consistent naming strategies. The dataset and job entities are identified by combination of 'namespace' and 'name' attributes whereas run is identified by runId. The entities can be enriched with user-defined metadata via facets (similar to metadata forms in Amazon SageMaker Unified Studio).

OpenLineage supports three types of events: RunEvent, DatasetEvent and JobEvent.

- RunEvent: this event is generated as a result of job-run execution. It contains details of the run, the job it belongs to, input datasets that run consumes and output datasets the run produces. Reference for samples run events. Currently, Amazon SageMaker Unified Studio only supports RunEvents.
- DatasetEvent: this event represents the changes in dataset (like any static updates on the dataset)
- JobEvent: this event represents the changes in job configuration/details

In the current release of Amazon SageMaker Unified Studio, OpenLineage 1.22.0+ versions are supported.

Data lineage support

In Amazon SageMaker Unified Studio, domain administrators or data users can configure lineage in projects while setting up connections for data lake and data warehouse sources to ensure the data source runs created from those resources are enabled for automatic lineage capture. Data lineage is automatically captured from data sources, such as AWS Glue and Amazon Redshift, as well as from tools, such as Visual ETL and notebooks, as executions create, update, or transform data. Additionally, Amazon SageMaker Unified Studio captures the movement of data within the catalog as producers bring their asset into inventory, and publish them, as well as when consumers subscribe and get access, to indicate who the subscribing projects are for a given asset. With this automation, different stages of an asset in the catalog are captured including when schema changes are detected.

Using Amazon SageMaker Unified Studio's OpenLineage-compatible APIs, domain administrators and data producers can capture and store lineage events beyond what is available in Amazon SageMaker Unified Studio, including transformations in Amazon S3, AWS Glue, and other services.

This provides a comprehensive view for the data consumers and helps them gain confidence of the asset's origin, while data producers can assess the impact of changes to an asset by understanding its usage. Additionally, Amazon SageMaker Unified Studio versions lineage with each event, enabling users to visualize lineage at any point in time or compare transformations across an asset's or job's history. This historical lineage provides a deeper understanding of how data has evolved, essential for troubleshooting, auditing, and ensuring the integrity of data assets.

With data lineage, you can accomplish the following in Amazon SageMaker Unified Studio:

- Understand the provenance of data: knowing where the data originated fosters trust in data by providing you with a clear understanding of its origins, dependencies, and transformations. This transparency helps in making confident data-driven decisions.
- Understand the impact of changes to data pipelines: when changes are made to data pipelines, lineage can be used to identify all of the downstream consumers that are to be affected. This helps to ensure that changes are made without disrupting critical data flows.
- Identify the root cause of data quality issues: if a data quality issue is detected in a downstream report, lineage, especially column-level lineage, can be used to trace the data back (at a column level) to identify the issue back to its source. This can help data engineers to identify and fix the problem.
- Improve data governance and compliance: column-level lineage can be used to demonstrate compliance with data governance and privacy regulations. For example, column-level lineage can be used to show where sensitive data (such as PII) is stored and how it is processed in downstream activities.

OpenLineage custom transport to send lineage events to SageMaker

OpenLineage events which contain metadata about data pipelines, jobs, and runs, are typically sent to backend for storage and analysis. The transport mechanism handles this transmission. As an extension of the OpenLineage project, a custom transport is available to send lineage events directly Amazon SageMaker Unified Studio's endpoint. The custom transport was merged into OpenLineage version 1.33.0 (https://openlineage.io/docs/releases/1_33_0/). This allows the use of OpenLineage plugins with the transport to send lineage events collected directly to Amazon SageMaker Unified Studio.

Data lineage support matrix

Lineage capture is automated from the following tools in Amazon SageMaker Unified Studio:

Tools support matrix

Tool	Compute	AWS Service	Service deployment option	Support status	Notes
Jupyterlab notebook	Spark	EMR	EMR Serverless	Automated	Spark DataFrames only; remote workflow execution
Jupyterlab notebook	Spark	AWS Glue	N/A	Automated	Spark DataFrames only; remote workflow execution
Visual ETL	Spark	AWS Glue	compatibility mode	Automated	Spark DataFrames only
Visual ETL	Spark	AWS Glue	fineGrained mode	Not supported	Spark DataFrames only
Query Editor		Amazon Redshift		Automated	

Lineage is captured from the following services:

Services support matrix

Data source	Lineage Support status	Required Configuration	Notes
AWS Glue Crawler	Automated by default in SageMaker Unified Studio	None	Supported for assets crawled via AWS Glue Crawler for the

Data source	Lineage Support status	Required Configuration	Notes
			following data sources: Amazon S3, Amazon DynamoDB, Amazon S3 Open Table Formats including : Delta Lake, Iceberg tables, Hudi tables, JDBC, PostgreSql, DocumentDB, and MongoDB.

Data source	Lineage Support status	Required Configuration	Notes
Amazon Redshift	Automated by default in SageMaker Unified Studio	None	Redshift System tables will be used to retrieve user queries and lineage is generated by parsing those queries
AWS Glue jobs in AWS Glue console	Not automated by default	User can select "generate lineage events" and pass domainId	

Data source	Lineage Support status	Required Configuration	Notes
EMR	Not automated by default	User has to pass spark conf parameters to publish lineage events	Supported versions: • EMR-S: 7.5+ • EMR on EC2: 7.11+ • EMR on EKS: 7.12+ More details in Capture lineage from EMR Spark executions

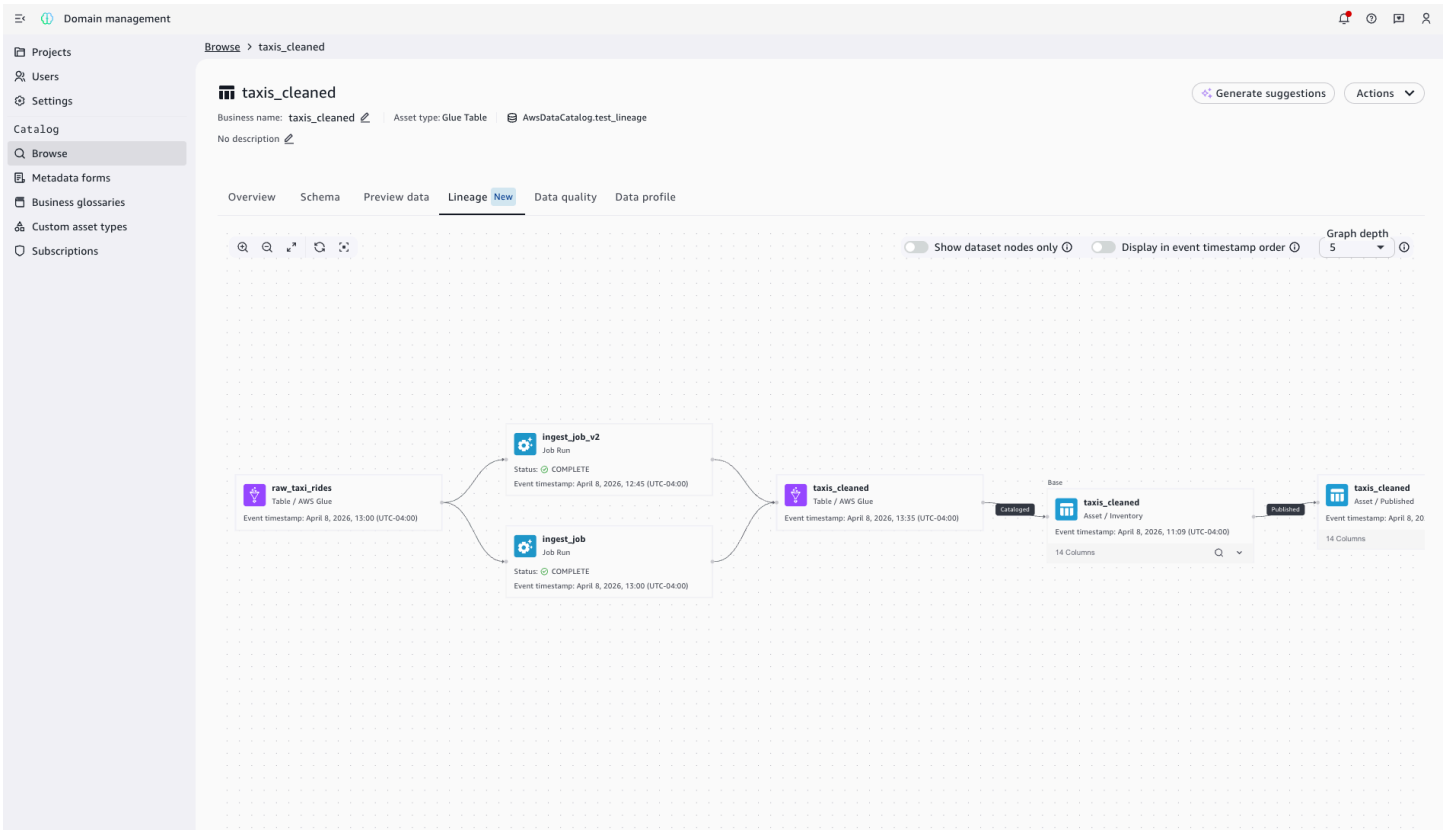
Visualizing data lineage in IAM mode

Data lineage in IAM mode provides a visual graph representation of how data flows through your organization's data pipelines. You can trace the origin, transformation, and destination of your data assets, helping you understand data dependencies, troubleshoot data quality issues, and assess the impact of changes.

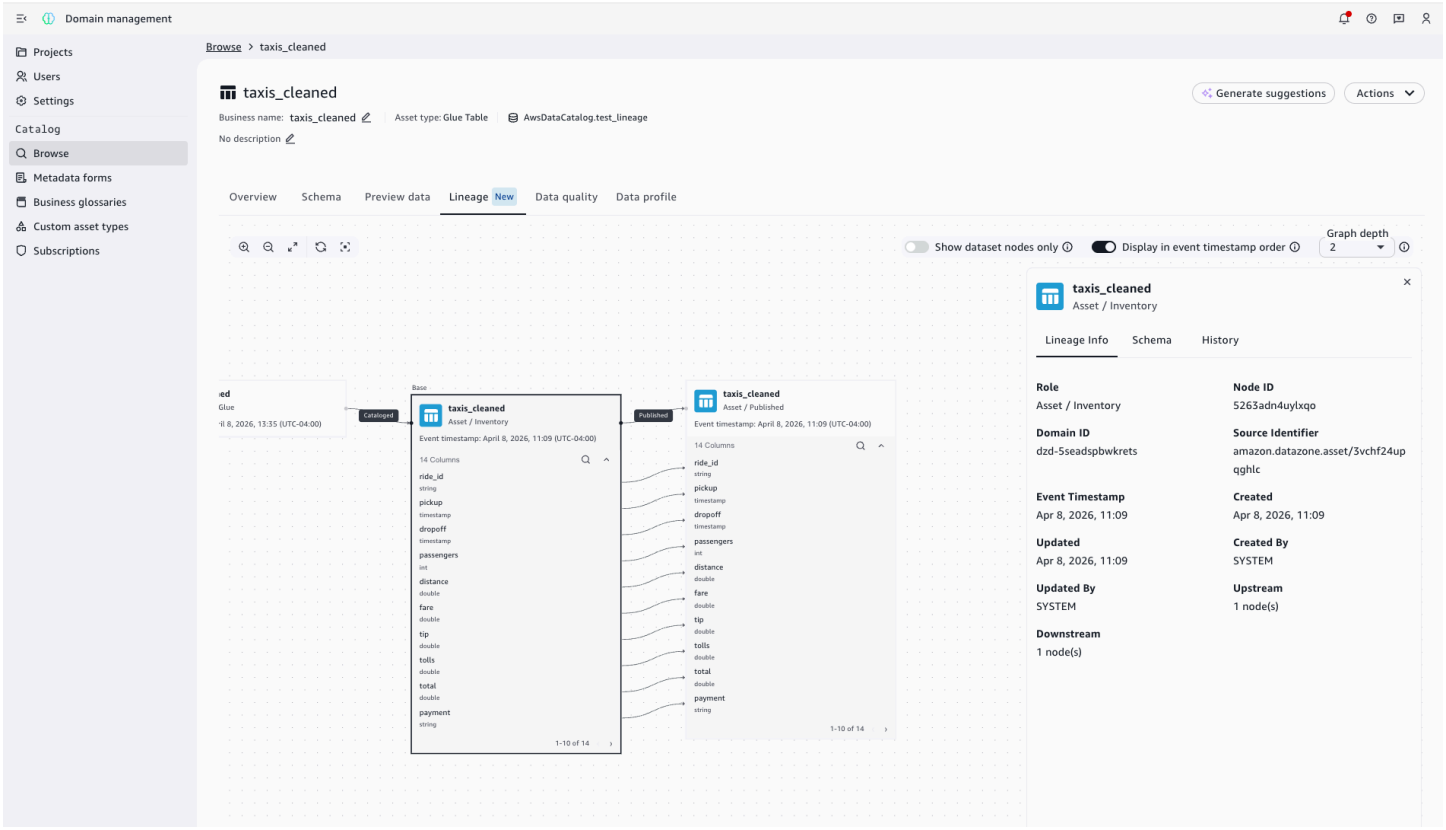
The lineage graph provides the following visualization capabilities:

- **Interactive lineage graph** — View data flow as a directed graph with nodes (datasets, jobs, job runs) connected by edges.
- **Two graph modes**:
  - **Query Graph mode** (default) — Fast server-side traversal that returns the lineage path efficiently.
  - **Event Timestamp mode** — Client-side recursive expansion that provides detailed node data including column-level lineage.
- **Configurable graph depth** — Control how many levels of upstream and downstream nodes to display (1–5 hops).
- **Dataset-only view** — Toggle to hide job and job run nodes for a simplified data flow view.

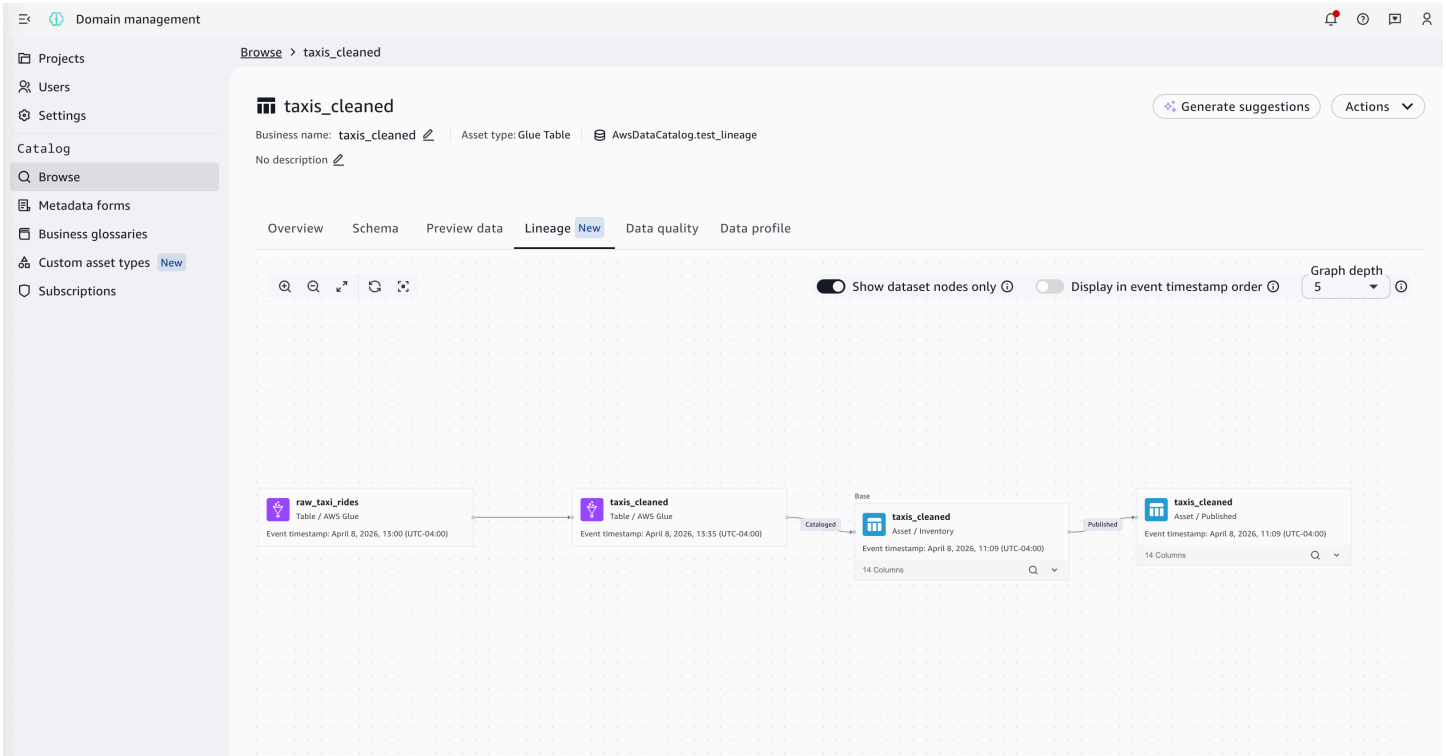
The following image shows Query Graph mode with server-side traversal of the lineage path.



The following image shows Event Timestamp mode with detailed node data including column-level lineage.



The following image shows Dataset-only view where you can Toggle to hide job and job run nodes for a simplified data flow view.



Generating lineage

By default, every cataloged data asset displays a basic lineage graph showing its relationship to the underlying table and published listing. No additional setup is required.

To see richer lineage showing how data flows between multiple assets, you can complete the following steps:

1. **Set up datasets** — Create or register data sources (for example, AWS Glue tables, Amazon S3 datasets) in your project.
2. **Run jobs on those datasets** — Execute ETL or transformation jobs (through JupyterLab notebooks, Visual ETL, Query Editor, or AWS Glue jobs) that read from and write to your datasets. Each job run automatically emits lineage events that connect input and output datasets in the graph.
3. **Run additional jobs** — The more jobs you run across datasets, the more connections appear in the lineage graph. Over time, you see a complete picture of how data flows through your pipelines.

The lineage graph updates automatically after each job run (with a short propagation delay). Reload the page or choose **Reset** to see newly created connections.

Node types

The lineage graph contains the following node types:

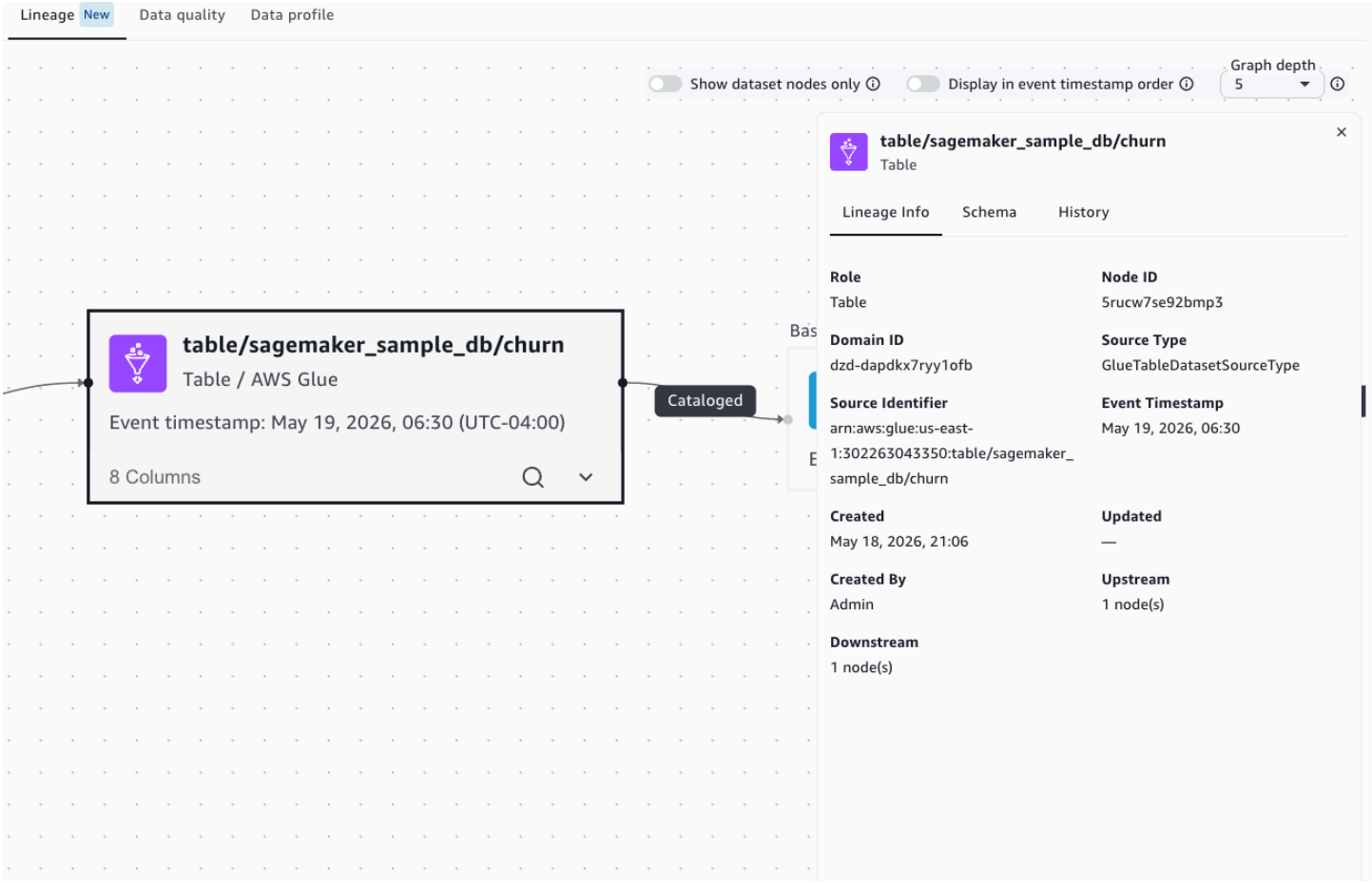
- **Dataset nodes** — Represent data assets (AWS Glue tables, Amazon S3 datasets).
- **Job Run nodes** — Represent individual executions of a job, including status (Running, Complete, Failed).
- **Asset/Listing nodes** — System-generated nodes that link physical tables to cataloged assets and published listings.

You can view detailed metadata for any node, including role, node ID, domain ID, source identifier, and timestamps. The details panel includes the following information:

- **Created By / Updated By** — Displayed as resolved user names (not UUIDs).
- **Schema tab** — View column definitions, source column mappings, and transformation details.

- **History tab** — Browse historical versions of a node and open historical lineage snapshots in separate tabs.

The following image shows the node details panel with metadata, schema, and history tabs.



Column lineage (Timestamp mode only)

When you use Event Timestamp mode, you can access column-level lineage with the following capabilities:

- **Column-level tracing** — Choose a column to highlight its lineage path across all connected nodes.
- **Column filtering** — When you select a column, other nodes automatically filter to show only connected columns.
- **Paginated columns** — Nodes with many columns display pagination (10 per page) with search functionality.

Graph controls

The lineage graph provides the following navigation controls:

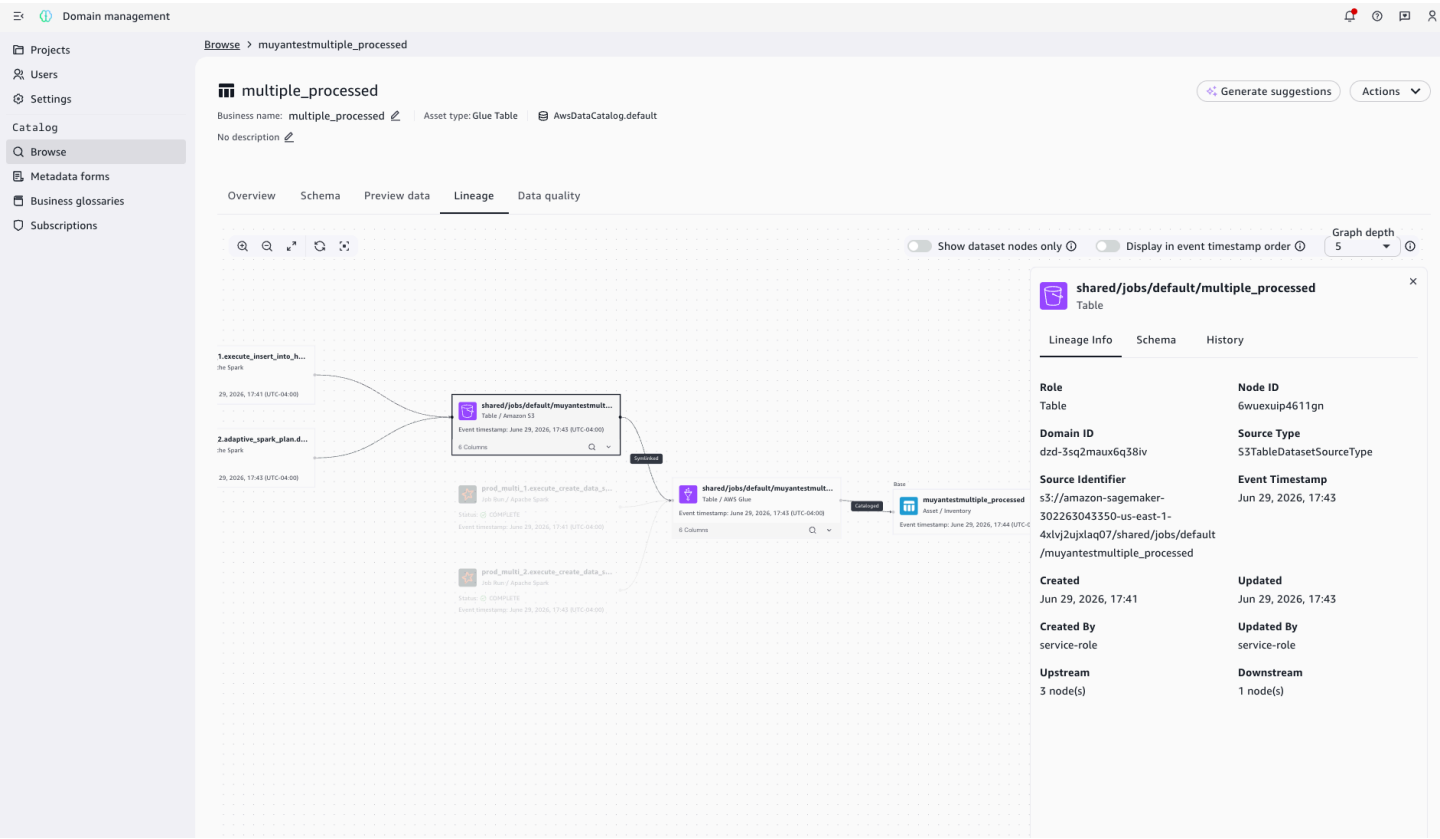
- Zoom in and out with configurable limits
- Fit to view
- Full-screen mode
- Reset to initial state

The following image shows the graph control buttons for zoom, fit, full-screen, and reset.



Choose a node to dim all nodes and edges that are not in its upstream or downstream path. This helps you focus on the data flow relevant to a specific asset.

The following image shows path highlighting with dimmed unrelated nodes.



Known limitations

The following limitations apply to data lineage in IAM mode:

- **Subscription relationships** — In IAM lineage, subscription nodes are not currently generated when a subscription is approved. However, lineage is still captured when a subscriber runs jobs on a subscribed dataset.
- **Redshift data sources** — Redshift asset nodes are not supported until Redshift data sources become available in the IAM catalog.

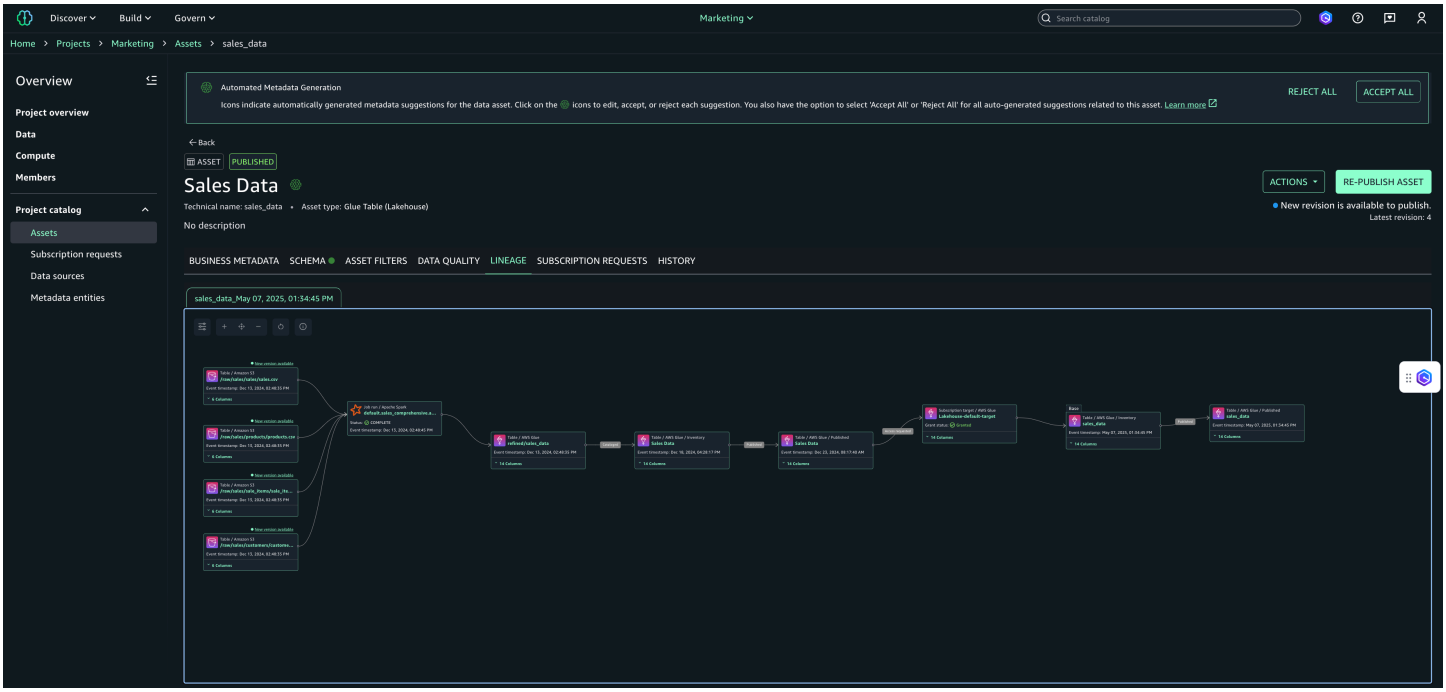
Visualizing data lineage in IdC mode

In Amazon SageMaker Unified Studio, nodes in the lineage graph contain lineage information while edges represent upstream/downstream directions of data propagation. The lineage information is present in metadata forms attached to the lineage node. Amazon SageMaker Unified Studio defines three types of lineage nodes:

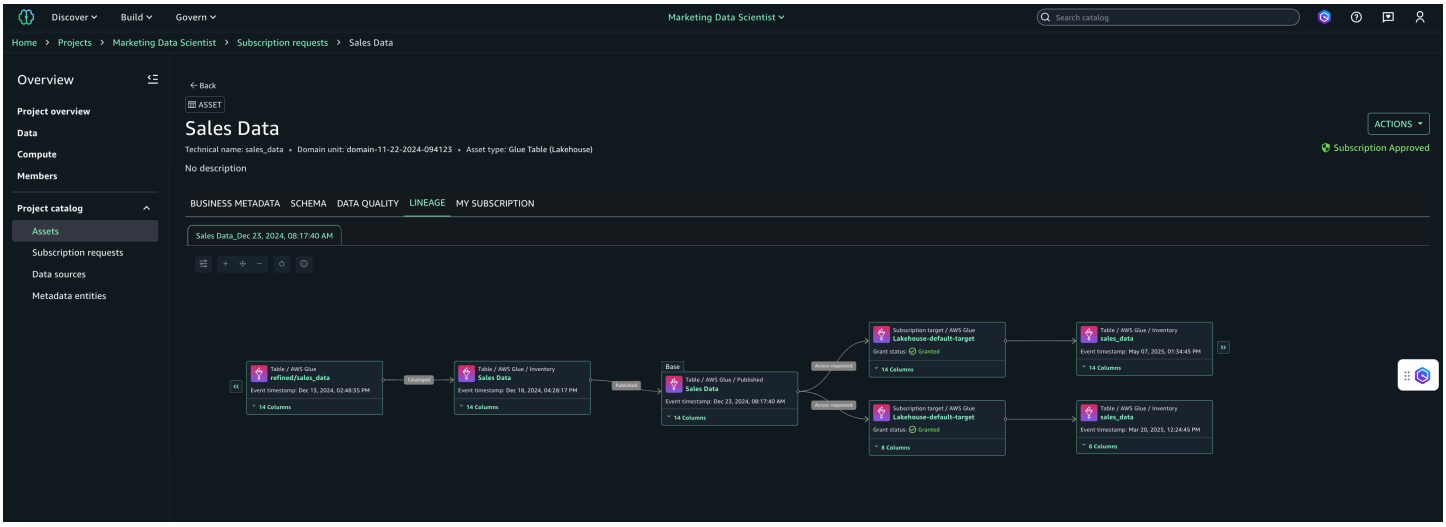
- Dataset node - this node includes data lineage information about a specific dataset.
  - Dataset refers to any object such as table, view, Amazon S3 file, Amazon S3 bucket, and so on. It also refers to the assets in Amazon SageMaker Unified Studio's inventory and catalog, and subscribed tables/views.
  - Each version of the dataset node represents an event happening on the dataset at that timestamp. The history tab on the dataset node shows all dataset versions.
- Job node - this node includes job details such as type of the job (query, etl etc), processing type (batch, streaming) etc job-type (query, etl etc), processing-type etc.
- JobRun node - this node represents the job run details such as the job it belongs to, status, start/end timestamps etc. Amazon SageMaker Unified Studio's lineage graph shows a combined for job and job-run which shows job details and latest run details along with a history of previous job-runs.

Lineage graph can be visualized with base node as an asset. In the SageMaker Unified Studio, search for the assets, open any asset and you can see lineage on the Asset Details page.

Here is the sample lineage graph for a user who is a data producer:



Here is the sample lineage graph for a user who is a data consumer:



The asset details page provides the following capabilities to navigate the graph:

- Column-level lineage: expand column-level lineage when available in dataset nodes. This automatically shows relationships with upstream or downstream dataset nodes if source column information is available.
- Column search: the default display for number of columns is 10. If there are more than 10 columns, pagination is activated to navigate to the rest of the columns. To quickly view a particular column, you can search on the dataset node that list just the searched column.

- View dataset nodes only: if you want to toggle to view only dataset lineage nodes and filter out the job nodes, you can choose the **Open view control** icon on the top left of the graph viewer and toggle the **Display dataset nodes only** option. This will remove all the job nodes from the graph and lets you navigate just the dataset nodes. Note that when the view only dataset nodes is turned on, the graph cannot be expanded upstream or downstream.
- Details pane: Each lineage node has details captured and displayed when selected.
  - Dataset node has a detail pane to display all the details captured for that node for a given timestamp. Every dataset node has 3 tabs, namely: Lineage info, Schema, and History tab. The history tab lists the different versions of lineage event captured for that node. All details captured from API are displayed using metadata forms or a JSON viewer.
  - Job node has a detail pane to display job details with tabs, namely: Job info, and History. The details pane also captures query or expressions captured as part of the job run. The history tab lists the different job runs captured for that job. All details captured from API are displayed using metadata forms or a JSON viewer.
- History tab: all lineage nodes in Amazon SageMaker Unified Studio's lineage have versioning. For every dataset node or job node, the versions are captured as history and that enables you to navigate between the different versions to identify what has changed over time. Each version opens a new tab in the lineage page to help compare or contrast.

Aggregated lineage view

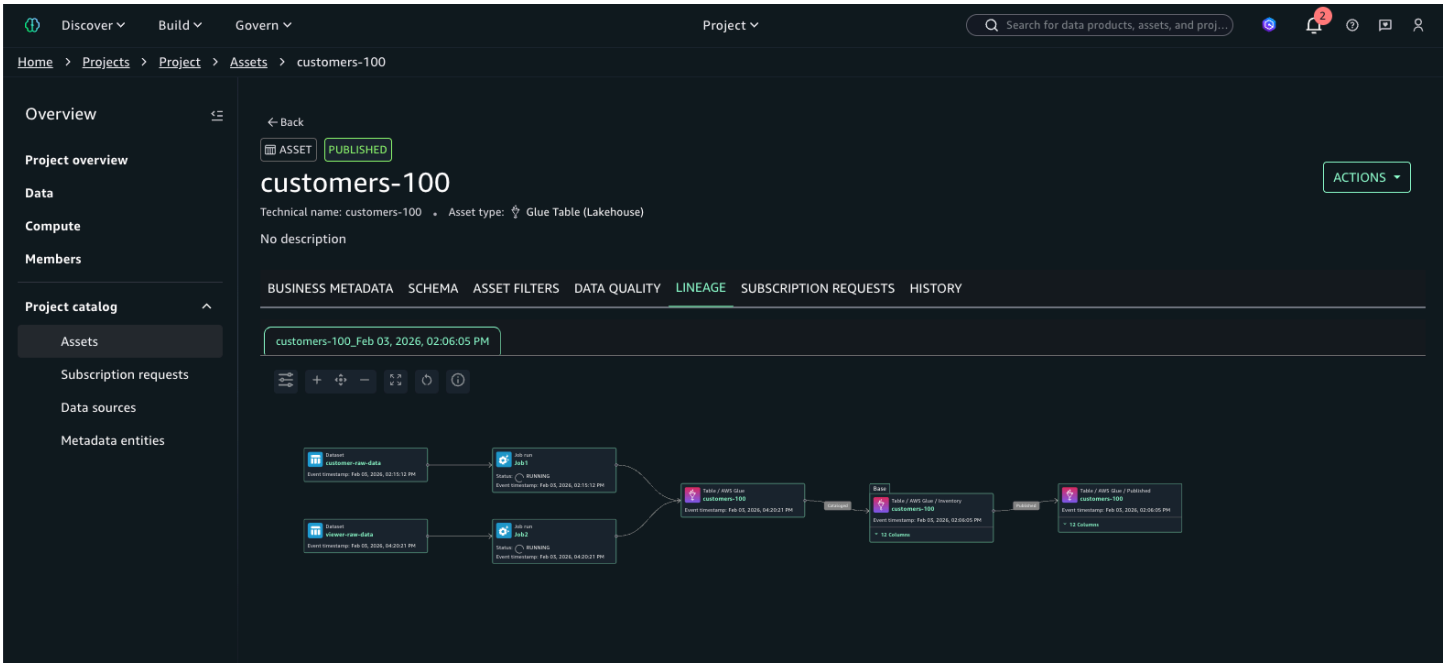
You can view an asset's lineage in two ways:

- **Aggregated view** - Shows all jobs that are currently contributing to an asset's lineage, providing a complete picture of the data transformations and dependencies across multiple levels of the lineage graph. Use this view to understand the full scope of jobs impacting your datasets and to identify all upstream sources and downstream consumers. This view does not support column-level lineage.
- **Timestamp view** - Shows the lineage graph as it existed at a specific point in time, displaying only the latest job run for each job at that timestamp. This view includes column-level lineage and is useful for troubleshooting and investigating specific data processing events.

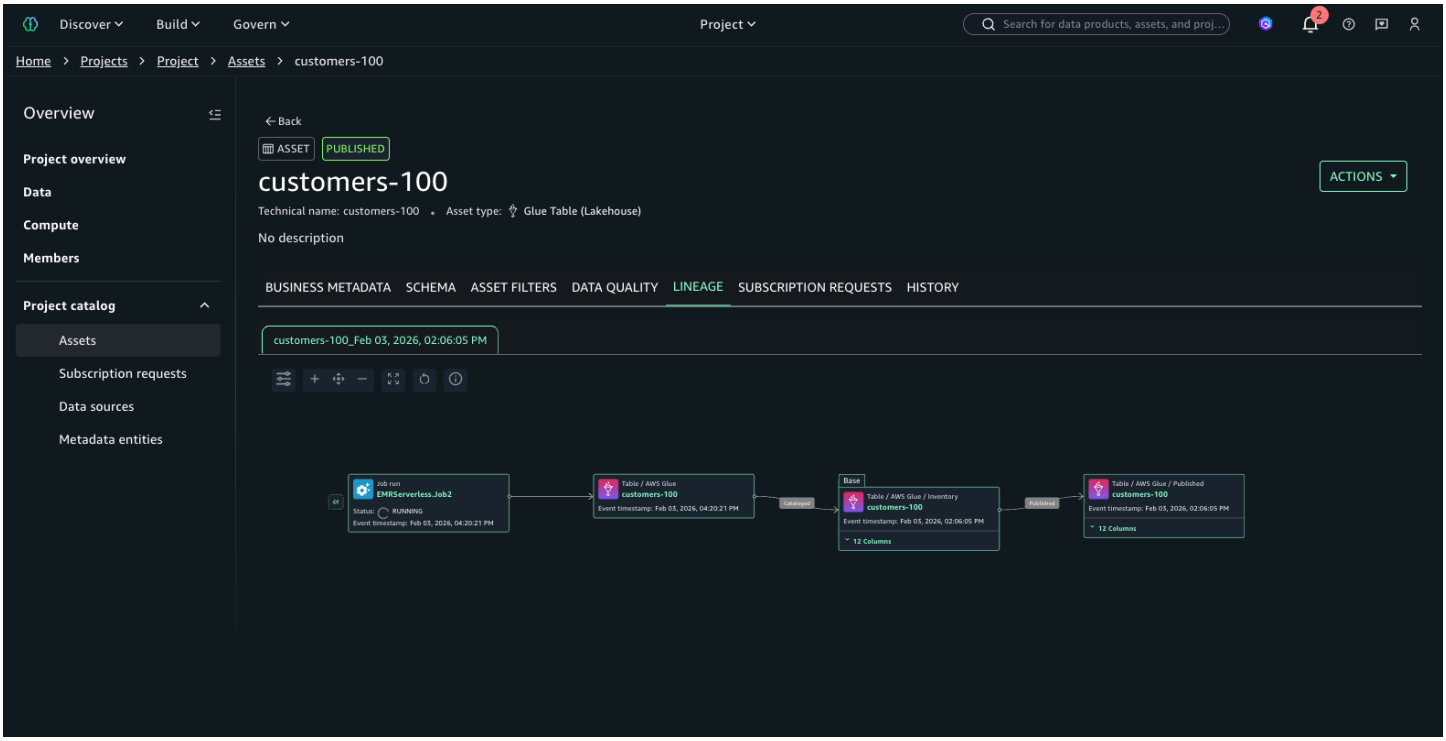
The aggregated view is the default in most regions and shows the current state of your data lineage. In Opt-In Regions, only the timestamp view is available.

To switch between views, choose the **Open view control** icon in the top left of the lineage graph viewer and toggle the **Display in event timestamp order** option. When enabled, the timestamp view is displayed. When disabled, the aggregated view is displayed. This toggle is not available in Opt-In Regions.

Here is a sample aggregated view of a lineage graph:



Here is a sample timestamp view of a lineage graph:



Test drive data lineage

You can use the data lineage sample experience to browse and understand data lineage in Amazon SageMaker Unified Studio, including traversing upstream or downstream in your data lineage graph, exploring versions and column-level lineage.

Complete the following procedure to try the sample data lineage experience in Amazon SageMaker Unified Studio:

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project you want to view lineage in.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Assets**.
4. On the **Inventory** tab, choose the name of the asset that you want to view lineage for. This opens the asset details page.
5. On the asset details page, choose the **Lineage** tab.
6. In the data lineage window, choose the info icon that says **Try sample data lineage**. Then choose **Launch**. A new pop-up window appears.

7. Choose **Start guided data lineage tour**.
8. Select a guided tour option, and then choose **Start tour**.

   At this point, a tab that provides all the space of lineage information is displayed. The sample data lineage graph is initially displayed with a base node with 1-depth at either ends, upstream and downstream. You can expand the graph upstream or downstream. The columns information is also available for you to choose and see how lineage flows through the nodes.

Data lineage authorization

Write permissions - to publish lineage events into Amazon SageMaker Unified Studio, you must have an IAM role with a policy that includes an ALLOW action on the PostLineageEvent API. To publish lineage data into Amazon SageMaker Unified Studio, you must have an IAM role with a permissions policy that includes an ALLOW action on the PostLineageEvent API. This IAM authorization happens at API Gateway layer.

Read permissions to view lineage - GetLineageNode and ListLineageNodeHistory are included in the AmazonSageMakerDomainExecution managed policy and therefore every user in an Amazon SageMaker unified domain can invoke these to view the data lineage graph in Amazon SageMaker Unified Studio.

Read permissions to get lineage events: you must have an IAM role with a policy that includes ALLOW action on ListLineageEvents and GetLineageEvent APIs to view lineage events posted to Amazon SageMaker Unified Studio.

Automate lineage capture from data connections

Topics

- Configure automated lineage capture for AWS Glue (Lakehouse) connections
- Configure automated lineage capture for Amazon Redshift connections

Configure automated lineage capture for AWS Glue (Lakehouse) connections

As databases and tables are added to the Amazon SageMaker Unified Studio's catalog, the lineage extraction can be automated from source for those assets using data source runs in Create Connection workflow. For every connection created, lineage is not automatically enabled.

To enable lineage capture for an AWS Glue connection

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project to which you want to add the data source.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Data sources**.
4. Choose the data source that you want to modify.
5. Expand the **Actions** menu, then choose **Edit data source** or click on the Data Source run name to view the details and go to **Data Source Definition** tab and choose **Edit** in **Connection** details.
6. Go to the connections and select **Import data lineage** checkbox to configure lineage capture from the source.
7. Make other changes to the data source fields as desired, then choose **Save**.

   Limitations

   Lineage is captured only for crawlers which imported less than 250 tables in a crawler run.

> **Note**
>
> When enabled, the lineage runs asynchronously to capture metadata from the source and generate lineage events to be stored in SageMaker Catalog to be visualized from a particular asset. The status of lineage runs for the data source can be viewed along with data source run details.

Configure automated lineage capture for Amazon Redshift connections

Capturing lineage from Amazon Redshift can be automated when the connection is added to an Amazon Redshift source in Amazon SageMaker Unified Studio's Data explorer. Lineage capture can be automated for a connection at the data source configuration. For every connection created, lineage is not automatically enabled.

To enable lineage capture for an Amazon Redshift connection

1. Navigate to Amazon SageMaker Unified Studio using the URL from your admin and log in using your SSO or AWS credentials.
2. From the project selector dropdown at the top of the page, choose the project to which you want to add the data source.
3. In the left navigation pane, choose **Manage**, then under **Catalog management**, choose **Data sources**.
4. Choose the data source that you want to modify.
5. Expand the **Actions** menu, then choose **Edit data source** or click on the data source run name to view the details and go to Data Source Definition tab and select **Edit** in **Connection details**.
6. Go to the connections and select **Import data lineage** checkbox to configure lineage capture from the source.
7. Make other changes to the data source fields as desired, then choose **Save**.

> **Note**
>
> When enabled, the lineage runs captures queries executed for a given database and generates lineage events to be stored in Amazon DataZone to be visualized from a particular asset. The lineage run for Amazon Redshift is set up for a daily run to pull from the Amazon Redshift system tables to derive lineage. For the first run, after enabling the feature, the first pull is scheduled for ~5 minutes after and set for a daily run. You can configure specific time programmatically.

Automate lineage capture from tools

Topics

- Capture lineage for Spark executions in Visual ETL
- Capture lineage for AWS Glue Spark executions in Notebooks
- Capture lineage from EMR Spark executions

Capture lineage for Spark executions in Visual ETL

When a new job is created in Visual ETL in Amazon SageMaker Unified Studio, lineage is automatically enabled. When a Visual ETL flow is created, lineage capture for that ETL flow is automatically enabled when you hit **Save to Project**. For every flow to capture lineage automatically, select **Save to Project** and then select **Run**.

Note: if you see that lineage is not getting captured, select **Save** and then move back to the Visual ETL flows section and then reopen the Visual ETL flow.

The following Spark configuration parameters are automatically added to the job being executed. When invoking Visual ETL programmatically, use the below configuration.

```
{
    "--conf":"spark.extraListeners=io.openlineage.spark.agent.OpenLineageSparkListener
    --conf spark.openlineage.transport.type=amazon_datazone_api
    --conf spark.openlineage.transport.domainId={DOMAIN_ID}
    --conf spark.glue.accountId={ACCOUNT_ID}
    --conf
 spark.openlineage.facets.custom_environment_variables=[AWS_DEFAULT_REGION;GLUE_VERSION;GLUE_CO
    --conf spark.openlineage.columnLineage.datasetLineageEnabled=True
    --conf spark.glue.JOB_NAME={JOB_NAME}"
}
```

The parameters are auto-configured and do not need any updates from the user. To understand the parameters in detail:

- `spark.extraListeners=io.openlineage.spark.agent.OpenLineageSparkListener` - OpenLineageSparkListener will be created and registered with Spark's listener bus
- `spark.openlineage.transport.type=amazon_datazone_api` - This is an OpenLineage specification to tell the OpenLineage Plugin to use DataZone API Transport to emit lineage events to DataZone's PostLineageEvent API. For more information, see https://openlineage.io/docs/integrations/spark/configuration/spark_conf/

- `spark.openlineage.transport.domainId={DOMAIN_ID}` - This parameter establishes the domain to which the API transport will submit the lineage events to.
- `spark.openlineage.facets.custom_environment_variables [AWS_DEFAULT_REGION;GLUE_VERSION;GLUE_COMMAND_CRITERIA;GLUE_PYTHON_VERSION;]` - The following environment variables (`AWS_DEFAULT_REGION`, `GLUE_VERSION`, `GLUE_COMMAND_CRITERIA`, and `GLUE_PYTHON_VERSION`), which AWS Glue interactive session populates, will be added to the LineageEvent
- `spark.glue.accountId=<ACCOUNT_ID>` - Account Id of the Glue Data Catalog where the metadata resides. This account id is used to construct Glue ARN in lineage event.
- `spark.glue.JOB_NAME` - Job name of the lineage event. In vETL flow, the job name is configured automatically to be spark.glue.JOB_NAME: ${projectId}.${pathToNotebook}

Spark compute limitations

- OpenLineage libraries for Spark are built into AWS Glue v5.0+ for Spark DataFrames only. Dynamic DataFrames are not supported.
- LineageEvent has a size limit of 300KB.

Capture lineage for AWS Glue Spark executions in Notebooks

Sessions in notebooks does not have a concept of a job. You can map the Spark executions to lineage events by generating a unique job name for the notebook. You can use the %%configure magic with the below parameters to enable lineage capture for Spark executions in the notebook.

Note: for AWS Glue Spark executions in notebooks, lineage capture is automated when scheduled with workflow in shared environment using remote workflows.

```
%%configure --name {COMPUTE_NAME} -f
{
"--conf":"spark.extraListeners=io.openlineage.spark.agent.OpenLineageSparkListener
 --conf spark.openlineage.transport.type=amazon_datazone_api
 --conf spark.openlineage.transport.domainId={DOMAIN_ID} --conf
 spark.glue.accountId={ACCOUNT_ID} --conf
 spark.openlineage.columnLineage.datasetLineageEnabled=True --conf
 spark.openlineage.facets.custom_environment_variables=[AWS_DEFAULT_REGION;GLUE_VERSION;GLUE_CO
 --conf spark.glue.JOB_NAME={JOB_NAME}"
}
```

Examples of {COMPUTE_NAME}: project.spark.compatibility or project.spark.fineGrained

Here are these parameters and what they configure, in detail:

- spark.extraListeners=io.openlineage.spark.agent.OpenLineageSparkListener
  - OpenLineageSparkListener will be created and registered with Spark's listener bus

- spark.openlineage.transport.type=amazon_datazone_api
  - https://openlineage.io/docs/integrations/spark/configuration/spark_conf
  - This is an OpenLineage specification to tell the OpenLineage Plugin to use DataZone API Transport to emit lineage events to DataZone's PostLineageEvent API to be captured in SageMaker.

- spark.openlineage.transport.domainId={DOMAIN_ID}
  - This parameter establishes the domain to which the API transport will submit the lineage events to.

- spark.openlineage.facets.custom_environment_variables [AWS_DEFAULT_REGION;GLUE_VERSION;GLUE_COMMAND_CRITERIA;GLUE_PYTHON_VERSION;]
  - The following environment variables (AWS_DEFAULT_REGION, GLUE_VERSION, GLUE_COMMAND_CRITERIA, and GLUE_PYTHON_VERSION), which Glue interactive session populates, will be added to the LineageEvent

- spark.glue.accountId={ACCOUNT_ID}
  - Account Id of the Glue Data Catalog where the metadata resides. This account id is used to construct Glue ARN in lineage event.

- [optional] spark.openlineage.transport.region={DOMAIN_REGION}
  - If domain region is different from that of the job's execution region, pass this parameter with value as domain's region

- spark.glue.JOB_NAME

- Job name of the lineage event. For example, the job name can be set to be spark.glue.JOB_NAME: ${projectId}.${pathToNotebook}.

Capture lineage from EMR Spark executions

EMR with Spark engine has the necessary OpenLineage libraries built in. You need to pass the following spark parameters. Be sure to replace the {Domain ID} with your specific Amazon DataZone or Amazon SageMaker Unified Studio domain and to replace the {Account ID} with the account id where the EMR job is run.

```
%%configure --name emr-s.{EMR_SERVERLESS_COMPUTE_NAME}
{
    "conf": {
         "spark.extraListeners": "io.openlineage.spark.agent.OpenLineageSparkListener",
         "spark.openlineage.columnLineage.datasetLineageEnabled":"True",
         "spark.openlineage.transport.type":"amazon_datazone_api",
         "spark.openlineage.transport.domainId":"{DOMAIN_ID}",
         "spark.openlineage.transport.region":"{DOMAIN_REGION}" // Only needed if the
 domain is in different region than that of the job
         "spark.glue.accountId":"{ACCOUNT_ID}", // needed if AWS Glue is being used as
 the Hive metastore
         "spark.jars":"/usr/share/aws/datazone-openlineage-spark/lib/
DataZoneOpenLineageSpark-1.0.jar" // Only needed incase of EMR-S
    }
}
```

- Lineage is supported for the following EMR versions:
  - EMR-S: 7.5+
  - EMR on EC2: 7.11+
  - EMR on EKS: 7.12+
- The JOB_NAME is the Spark application name that is automatically set
- Replace {DOMAIN_ID}, {ACCOUNT_ID}, {DOMAIN_REGION}
- Amazon SageMaker Unified Studio VPC endpoint is deployed to EMR VPC endpoint

Permissions required for data lineage

Read permissions to view lineage

Permissions on following actions are needed to view lineage graph:

- `datazone:GetLineageNode`
- `datazone:ListLineageNodeHistory`
- `datazone:QueryGraph`

Above permissions are included in the `AmazonSageMakerDomainExecution` managed policy and therefore every user in an Amazon SageMaker Unified Studio domain can invoke these to view the data lineage graph in Amazon SageMaker Unified Studio.

Permissions on following actions are needed to view lineage events:

- `datazone:ListLineageEvents`
- `datazone:GetLineageEvent`

User must have an IAM role with a policy that includes "Allow" action on these APIs to view lineage events posted to Amazon SageMaker Unified Studio.

Write permissions to publish lineage

Lineage for AWS Glue crawler

The project user role is used to fetch required data from AWS Glue. The project user role should contain the following permissions on Glue operations:

- `glue:listCrawls`
- `glue:getConnection`

> **Note**
>
> `SageMakerStudioProjectUserRolePolicy` already contains above permissions.

Lineage for Amazon Redshift

The project user role is used to execute queries on the cluster/workgroup defined in the connection. The project user role should contain the following permissions:

- `redshift-data:BatchExecuteStatement`
- `redshift-data:ExecuteStatement`
- `redshift-data:DescribeStatement`
- `redshift-data:GetStatementResult`

> **Note**
>
> `SageMakerStudioProjectUserRolePolicy` already contains above permissions.

In addition, the credentials provided for Amazon Redshift connection in Amazon SageMaker Unified Studio should contain following permissions:

- `sys:operator` role to access the data from system tables for all user queries performed on the cluster/workgroup
- Has "SELECT" grant on all the tables

Lineage for AWS Glue, EMR jobs

The IAM role used to execute the job should contain following permissions to publish lineage events to Amazon SageMaker Unified Studio:

- ALLOW action on `datazone:PostLineageEvent`
- If your Amazon SageMaker Unified Studio domain is encrypted with KMS CMK (customer managed key), the job role should have permissions to encrypt and decrypt as well
- If the spark job is in an account different from Amazon SageMaker Unified Studio domain account, associate the account with domain prior to running the job. Follow https://docs.aws.amazon.com/datazone/latest/userguide/working-with-associated-accounts.html to set up account association

Publish Lineage using API

IAM role with a policy to allow `datazone:PostLineageEvent` action is needed to post lineage events programmatically

Publishing data lineage programmatically

You can also publish data lineage programmatically using PostLineageEvent API. The API takes in open lineage run event as the payload. Additionally, the following APIs support retrieving lineage events and traversing lineage graph:

- GetLineageEvent
- ListLineageEvents
- DeleteLineageEvent: Deletes the specified lineage event
- QueryGraph: paginated API to return the aggregate view of the lineage graph
- GetLineageNode: Gets the lineage node along with its immediate neighbors
- ListLineageNodeHistory: lists the lineage node versions with each version derived from a data/metadata change event

The following is a sample PostLineageEvent operation payload:

```
{
  "producer": "https://github.com/OpenLineage/OpenLineage",
  "schemaURL": "https://openlineage.io/spec/2-0-0/OpenLineage.json#/definitions/
RunEvent",
  "eventType": "COMPLETE",
  "eventTime": "2024-05-04T10:15:30Z",
  "run": {
    "runId": "d2e7c111-8f3c-4f5b-9ebd-cb1d7995082a"
  },
  "job": {
    "namespace": "xyz.analytics",
    "name": "transform_sales_data"
  },
  "inputs": [
    {
      "namespace": "xyz.analytics",
      "name": "raw_sales",
```

```
      "facets": {
        "schema": {
          "_producer": "https://github.com/OpenLineage/OpenLineage",
          "_schemaURL": "https://openlineage.io/spec/facets/schema_dataset.json",
          "fields": [
            { "name": "region", "type": "string" },
            { "name": "year", "type": "int" },
            { "name": "created_at", "type": "timestamp" }
          ]
        }
      }
    }
  ],
  "outputs": [
    {
      "namespace": "xyz.analytics",
      "name": "clean_sales",
      "facets": {
        "schema": {
          "_producer": "https://github.com/OpenLineage/OpenLineage",
          "_schemaURL": "https://openlineage.io/spec/facets/schema_dataset.json",
          "fields": [
            { "name": "region", "type": "string" },
            { "name": "year", "type": "int" },
            { "name": "created_at", "type": "timestamp" }

          ]
        },
        "columnLineage": {
          "_producer": "https://github.com/OpenLineage/OpenLineage",
          "_schemaURL": "https://openlineage.io/spec/facets/columnLineage" +
 "DatasetFacet.json",
          "fields": {
            "id": {
              "inputFields": [
                {
                  "namespace": "xyz.analytics",
                  "name": "raw_sales",
                  "field": "id"
                }
              ]
            },
            "year": {
              "inputFields": [
```

```
                {
                  "namespace": "xyz.analytics",
                  "name": "raw_sales",
                  "field": "year"
                }
              ]
            },
            "created_at": {
              "inputFields": [
                {
                  "namespace": "xyz.analytics",
                  "name": "raw_sales",
                  "field": "created_at"
                }
              ]
            }
          }
        }
      }
    }
  ]
}
```

The importance of the sourceIdentifier attribute to lineage nodes

Every lineage node is uniquely identified by its sourceIdentifier (usually provided as part of open-lineage event) in addition to system generated nodeId. sourceIdentifier is generated using <namespace>, <name> of the node in lineage event.

The following are examples of sourceIdentifier values for different types of nodes:

- **Job nodes**
  - SourceIdentifier of job nodes is populated from <namespace>.<name> on the job node in open-lineage run event
- **Jobrun nodes**
  - SourceIdentifier of jobrun nodes is populated from <job's namespace>.<job's name>/<run_id>
- **Dataset nodes**

- Dataset nodes representing AWS resources: sourceIdentifier is in ARN format
  - AWS Glue table: arn:aws:glue:<region>:<account-id>:table/<database>/<table-name>
  - AWS Glue table with federated sources: arn:aws:glue:<region>:<account-id>:table/<catalog><database>/<table-name>
    - Example: catalog can be "s3tablescatalog"/"s3tablesBucket", "lakehouse_catalog" etc
  - Amazon Redshift table:
    - serverless: arn:aws:redshift-serverless:<region>:<account-id>:table/workgroupName/<database>/<schema>/<table-name>
    - provisioned: arn:aws:redshift:<region>:<account-id>:table/clusterIdentifier/<database>/<schema>/<table-name>
  - Amazon Redshift view:
    - serverless: arn:aws:redshift-serverless:<region>:<account-id>:view/workgroupName/<database>/<schema>/<view-name>
    - provisioned: arn:aws:redshift:<region>:<account-id>:view/clusterIdentifier/<database>/<schema>/<view-name>
- Dataset nodes representing SageMaker catalog resources:
  - Asset: amazon.datazone.asset/<assetId>
  - Listing (published asset): amazon.datazone.listing/<listingId>
- In all other cases, dataset nodes' sourceIdentifier is populated using <namespace>/<name> of the dataset nodes in open-lineage run event
  - https://openlineage.io/docs/spec/naming/ contains naming convention for various datastores.

The following table contains the examples of how sourceIdentifier is generated for datasets of different types.

Source for lineage event	Sample OpenLineage event data	Source ID computed by Amazon DataZone
AWS Glue ETL	{ "run": {	arn:aws:glue:us-west-2:123456789012:table/testdb/testtb-1

Source for lineage event	Sample OpenLineage event data	Source ID computed by Amazon DataZone
	"runId":"4e3da9e8-6228-4679-b0a2-fa916119fthr", "facets":{ "environment-properties":{ .... "environment-properties":{ "GLUE_VERSION":"3.0", "GLUE_COMMAND_CRITERIA":"glueetl", "GLUE_PYTHON_VERSION":"3" } } } }, ... "outputs":[{ "namespace":"namespace.output", "name":"output_name", "facets":{ "symlinks":{ .... "identifiers":[{ "namespace":"arn:aws:glue:us-west-2:123456789012",	If environment-properties contains GLUE_VERSION, GLUE_PYTHON_VERSION, etc, Amazon DataZone uses namespace and name in symlink of the dataset (input or output) to construct AWS Glue table ARN for sourceIdentifier.

Source for lineage event	Sample OpenLineage event data	Source ID computed by Amazon DataZone
	"name":"table/testdb/testtb-1", "type":"TABLE" }] } } }] }	

Source for lineage event	Sample OpenLineage event data	Source ID computed by Amazon DataZone
Amazon Redshift (Provisioned)	{ "run": { "runId":" 4e3da9e8-6228-4679-b0a2-fa916119fthr", "facets":{ ..... } }, ... "inputs":[{ "namespace":"redshift://cluster-20240715.123456789012.us-east-1.redshift.amazonaws.com:5439", "name":"tpcds_data.public.dws_tpcds_7" "facets":{ ..... } }] }	arn:aws:redshift:us-east-1:123456789012:table/cluster-20240715/tpcds_data/public/dws_tpcds_7 If the namespace prefix is `redshift`, Amazon DataZone uses that to construct the Amazon Redshift ARN using values of namespace and name attributes.

Source for lineage event	Sample OpenLineage event data	Source ID computed by Amazon DataZone

Amazon Redshift (serverless)	{
	"run": {
	"runId":"
	4e3da9e8-6228-4679-
	b0a2-fa916119fthr",
	"facets":{
	.....
	}
	},
	.....
	"outputs":[
	{
	"namespac
	e":"redshift://wor
	kgroup-20240715.12
	3456789012.us-east
	-1.redshift-server
	less.amazonaws.com
	:5439",
	"name":"t
	pcds_data.public.d
	ws_tpcds_7"
	"facets":{
	.....
	}
	}
	]
	}
		arn:aws:redshift-serverless:us-east-1:123456789012:table/workgroup-20240715/tpcds_data/public/dws_tpcds_7
		
		As per OpenLineage naming convention, namespace for Amazon Redshift dataset should be `provider://{cluster_identifier or workgroup}.{region_name}:{port}` .
		
		If the namespace contains `redshift-serverless` , Amazon DataZone uses that to construct Amazon Redshift ARN using values of namespace and name attributes.

Any other datastore	Recommendation is to populate namespace and name as per OpenLineage convention defined in https://openlineage.io/docs/spec/naming/.	Amazon DataZone populates sourceIdentifier as <namespace>/<name>.

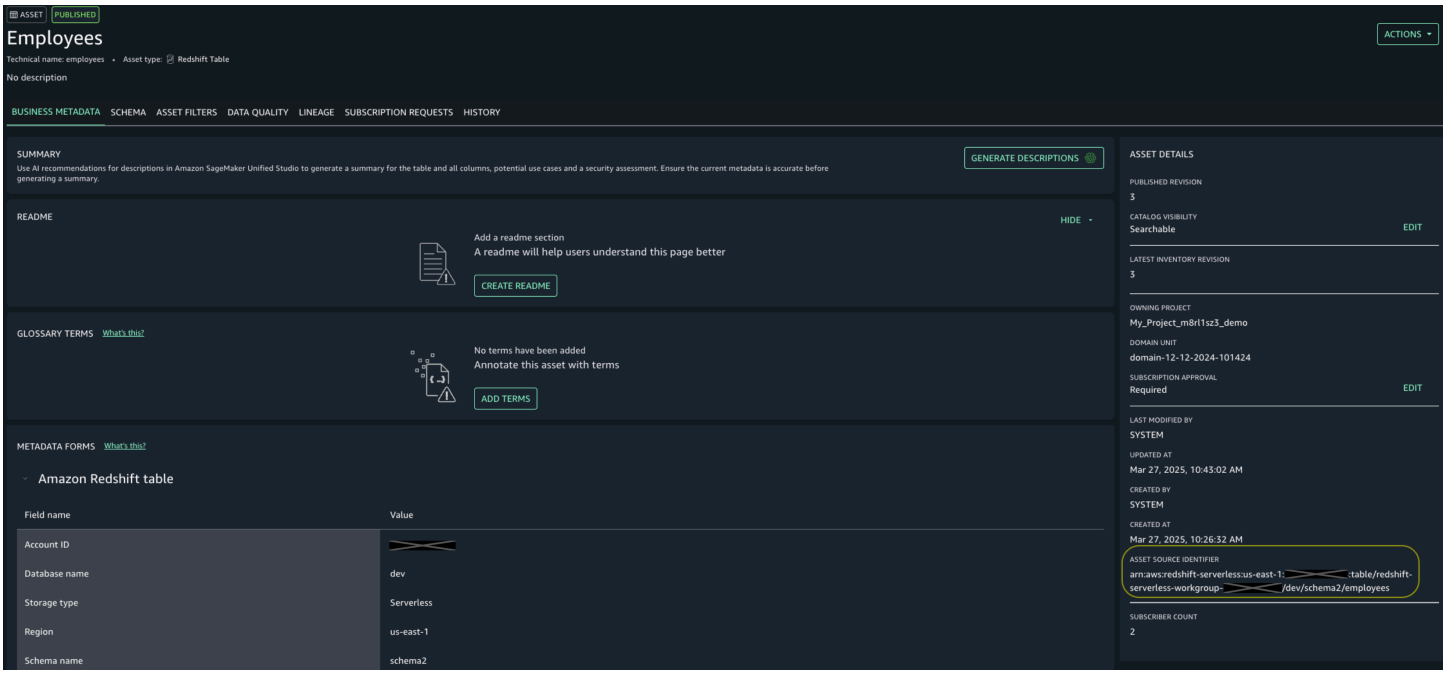
Linking dataset lineage nodes with assets imported into Amazon SageMaker Unified Studio

Linking dataset lineage nodes with assets imported into Amazon SageMaker Unified Studio

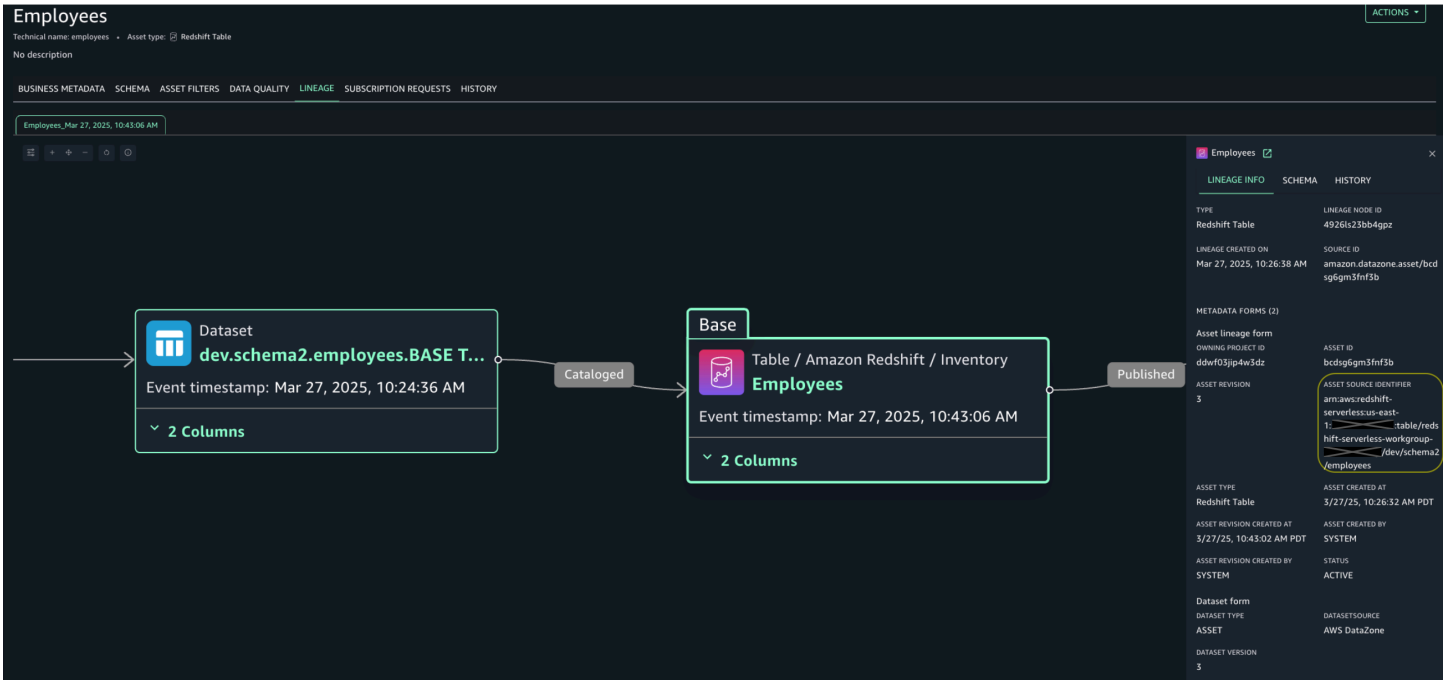
Every lineage node is uniquely identified by its sourceIdentifier. Previous section talks about formats of sourceIdentifier. Amazon SageMaker Unified Studio automatically links the dataset nodes with assets in inventory based on the sourceIdentifier value. Hence, use the same sourceIdentifier value of dataset node when creating/updating the asset (via AssetCommonDetailsForm::sourceIdentifier attribute).

Following images show the sourceIdentifier on asset details page along with lineage graph highlighting the same sourceIdentifier of dataset node with its downstream asset's sourceIdentifier.

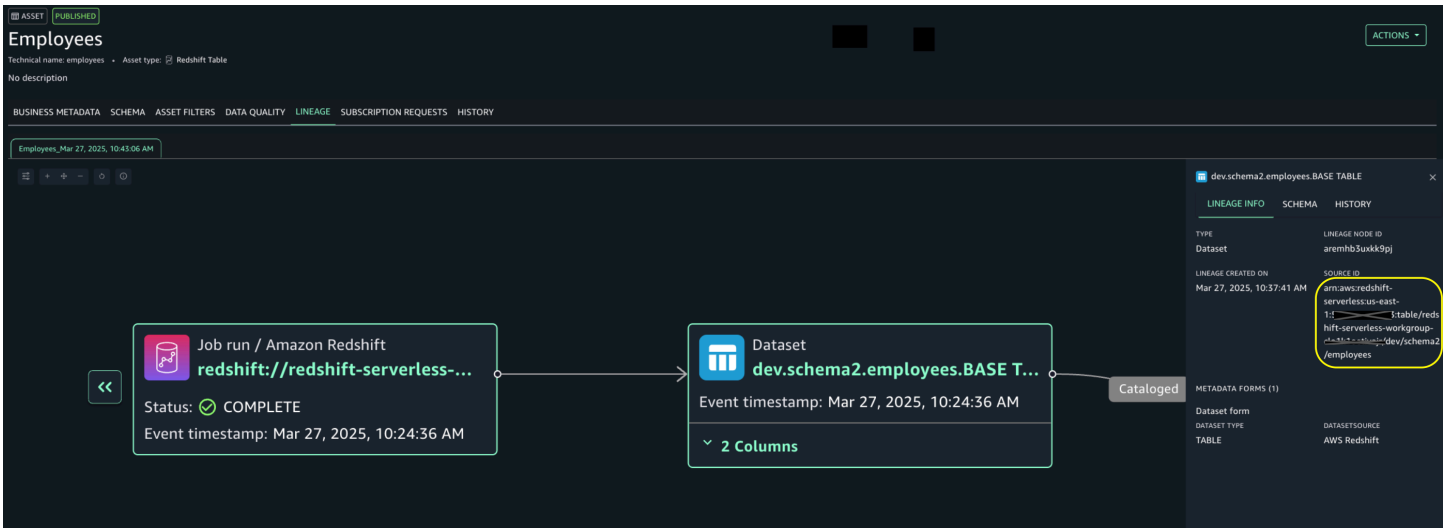
Asset details page:



Asset's SourceIdentifier in node details:



Amazon Redshift dataset/table's sourceIdentifier in node details:



Troubleshooting data lineage

This comprehensive troubleshooting guide helps you resolve common data lineage visibility issues in Amazon SageMaker Unified Studio. The guide covers programmatically published events, data source configurations, and tool-specific lineage capture problems.

Topics

- Not seeing lineage graph for events published programmatically

- Not seeing lineage for assets even though importLineage is shown as true in AWS Glue datasource
- Not seeing lineage for assets even though importLineage is shown as true in Amazon Redshift datasource
- Troubleshooting lineage for lineage events published from AWS Glue ETL jobs/vETL/Notebooks
- Troubleshooting lineage for lineage events published from EMR-S/EC2/EKS

Not seeing lineage graph for events published programmatically

Primary requirement: Lineage graphs are only visible in Amazon SageMaker Unified Studio if at least one node of the graph is an asset. You must create assets for any dataset nodes and properly link them using the sourceIdentifier attribute.

Troubleshooting steps:

1. Create assets for any of the dataset nodes involved in your lineage events. Refer to the following sections for proper linking:
   - The importance of the sourceIdentifier attribute to lineage nodes
   - Linking dataset lineage nodes with assets imported into Amazon SageMaker Unified Studio
2. Once the asset is created, verify that you can see the lineage on the asset details page.
3. If you are still not seeing lineage, verify that the asset's sourceIdentifier (present in AssetCommonDetailsForm) has the same value as the sourceIdentifier of any input/output dataset node in the lineage event.

   Use the following command to get asset details:

   ```
   aws datazone get-asset --domain-identifier {DOMAIN_ID} --identifier {ASSET_ID}
   ```

   The response appears as follows:

   ```
   {
       .....
       "formsOutput": [
   ```

```
        .....
        {
            "content": "{\"sourceIdentifier\":\"arn:aws:glue:eu-
west-1:123456789012:table/testlfdb/testlftb-1\"}",
            "formName": "AssetCommonDetailsForm",
            "typeName": "amazon.datazone.AssetCommonDetailsFormType",
            "typeRevision": "6"
        },
        .....
    ],
    "id": "{ASSET_ID}",
    ....
}
```

4. If both sourceIdentifiers are matching but you still cannot see lineage, retrieve the eventId from the PostLineageEvent response or use ListLineageEvents to find the eventId, then invoke GetLineageEvent:

```
aws datazone list-lineage-events --domain-identifier {DOMAIN_ID}
// You can apply additional filters like timerange etc.
// Refer https://docs.aws.amazon.com/datazone/latest/APIReference/
API_ListLineageEvents.html

aws datazone get-lineage-event --domain-identifier {DOMAIN_ID} --identifier
 {EVENT_ID} --output json event.json
```

The response appears as follows and the open-lineage event is written to the event.json file:

```
{
    "domainId": "{DOMAIN_ID}",
    "id": "{EVENT_ID}",
    "createdBy": ....,
    "processingStatus": "SUCCESS"/"FAILED etc",
    "eventTime": "2024-05-04T10:15:30+00:00",
    "createdAt": "2025-05-04T22:18:27+00:00"
}
```

5. If the GetLineageEvent response's processingStatus is FAILED, contact AWS Support by providing the GetLineageEvent response for the appropriate event and the response from GetAsset.
6. If the GetLineageEvent response's processingStatus is SUCCESS, double-check that the sourceIdentifier of the dataset node from the lineage event matches the value in the GetAsset response above. The following steps help verify this.
7. Invoke GetLineageNode for the job run where the identifier is composed of namespace, name of job and run_id in the lineage event:

```
aws datazone get-lineage-node --domain-identifier {DOMAIN_ID} --identifier <job's
 namespace>.<job's name>/<run_id>
```

The response appears as follows:

```
{
    .....
    "downstreamNodes": [
        {
            "eventTimestamp": "2024-07-24T18:08:55+08:00",
            "id": "afymge5k4v0euf"
        }
    ],
    "formsOutput": [
        <some forms corresponding to run and job>
    ],
    "id": "<system generated node-id for run>",
    "sourceIdentifier": "<job's namespace>.<job's name>/<run_id>",
    "typeName": "amazon.datazone.JobRunLineageNodeType",
    ....
    "upstreamNodes": [
        {
            "eventTimestamp": "2024-07-24T18:08:55+08:00",
            "id": "6wf2z27c8hghev"
        },
        {
            "eventTimestamp": "2024-07-24T18:08:55+08:00",
            "id": "4tjbcsnre6banb"
        }
```

```
    ]
}
```

8. Invoke GetLineageNode again by passing in the downstream/upstream node identifier (which you think should be linked to the asset node):

```
aws datazone get-lineage-node --domain-identifier {DOMAIN_ID} --identifier
 afymge5k4v0euf
```

This returns the lineage node details corresponding to the dataset: `afymge5k4v0euf`. Verify if the sourceIdentifier matches that of the asset. If not matching, fix the namespace and name of the dataset in the lineage event and publish the lineage event again. You will see the lineage graph on the asset.

```
{
    .....
    "downstreamNodes": [],
    "eventTimestamp": "2024-07-24T18:08:55+08:00",
    "formsOutput": [
        .....
    ],
    "id": "afymge5k4v0euf",
    "sourceIdentifier": "<sample_sourceIdentifier_value>",
    "typeName": "amazon.datazone.DatasetLineageNodeType",
    "typeRevision": "1",
    ....
    "upstreamNodes": [
        ...
    ]
}
```

Not seeing lineage for assets even though importLineage is shown as true in AWS Glue datasource

Open the datasource run(s) associated with the AWS Glue datasource and you can see the assets imported as part of the run and the lineage import status along with error message in case of failure.

Limitations:

- Lineage for crawler run importing more than 250 tables isn't supported.

Not seeing lineage for assets even though importLineage is shown as true in Amazon Redshift datasource

Lineage on Amazon Redshift tables is captured by retrieving user queries performed on the Amazon Redshift database, from the system tables.

Lineage is not supported in the following cases:

- External Tables
- Unload / Copy
- Merge / Update
- Queries that produce Lineage Events larger than 16MB
- Datashares
- ColumnLineage limitations:
  - Column Lineage is not supported for queries not containing specific columns such as (select * from tableA)
  - Column Lineage is not supported for queries involving temp tables
- Any limitations pertaining to OpenLineageSqlParser results in failure to process some queries

Troubleshooting steps:

1. On the Amazon Redshift connection details, you will see the lineageJobId along with the job run schedule. Alternatively, you can fetch it using the get-connection API.
2. Invoke list-job-runs to get the runs corresponding to the job:

```
aws datazone list-job-runs --domain-identifier {DOMAIN_ID} --job-identifier
 {JOB_ID}
```

The response appears as follows:

```
{
   "items": [
      {
         .....
         "error": {
            "message": "string"
         },
         "jobId": {JOB_ID},
         "jobType": LINEAGE,
         "runId": ...,
         "runMode": SCHEDULED,
         "status": SCHEDULED | IN_PROGRESS | SUCCESS | PARTIALLY_SUCCEEDED | FAILED
 | ABORTED | TIMED_OUT | CANCELED
         .....
      }
   ],
   "nextToken": ...
}
```

3. If no job-runs are returned, check your job run schedule on the Amazon Redshift connection details. Reach out to AWS Support providing the lineageJobId, connectionId, projectId and domainId if job runs are not executed per given schedule.
4. If job-runs are returned, pick the relevant jobRunId and invoke GetJobRun to get job run details:

```
aws datazone get-job-run --domain-identifier {DOMAIN_ID} --identifier {JOB_RUN_ID}
```

The response appears as follows:

```
{
  ....
  "details": {
    "lineageRunDetails": {
      "sqlQueryRunDetails": {
        "totalQueriesProcessed": ..,
        "numQueriesFailed": ...,
        "errorMessages":....,
        "queryEndTime": ...,
        "queryStartTime": ...
      }
    }
  },
  .....
}
```

5. The job run fails if none of the queries are successfully processed; partially succeeds if some queries are successfully processed; succeeds if all queries are successfully processed and response also contains start and end times of processed queries.

Troubleshooting lineage for lineage events published from AWS Glue ETL jobs/vETL/Notebooks

Limitations:

- OpenLineage libraries for Spark are built into AWS Glue v5.0+ for Spark DataFrames only. Does not support Glue DynamicFrames.
- Lineage capture for Spark jobs with fine-grained permission mode are not supported.
- Lineage event has a size limit of 300KB.

Common Issues:

- Verify necessary permissions are given to your job execution role as per Permissions required for data lineage

- Your spark job working with S3 files would produce lineage events with s3 datasets, even when they are catalog'ed in AWS Glue. To generate events including AWS Glue tables and build proper lineage graph with AWS Glue assets, your spark job should instead work with glue tables.
- If the AWS Glue ETL is in VPC, make sure the Amazon DataZone VPC endpoint is deployed in that VPC.
- In case your domain is using a CMK, make sure that the AWS Glue execution role has the appropriate KMS permissions. CMK can be found via https://docs.aws.amazon.com/datazone/latest/APIReference/API_GetDomain.html
- Failed to publish Lineage Event because payload is greater than 300 kb:
  - Add the following to Spark conf:

    ```
    "spark.openlineage.columnLineage.datasetLineageEnabled": "True"
    ```

  - **Important Note:**
    - Column lineage typically constitutes a significant portion of the payload and enabling this will efficiently generate the column lineage info.
    - Disabling it can help reduce payload size and avoid validation exceptions.
- Cross account lineage event submission:
  - Follow https://docs.aws.amazon.com/datazone/latest/userguide/working-with-associated-accounts.html to set up account association
  - Ensure that RAM policy is using the latest policy
- If your Amazon SageMaker Unified Studio domain is in different region from that of job
  - Add this spark parameter: `"spark.openlineage.transport.region":"{region of your domain}"`
- When the same DataFrame is written to multiple destinations or formats in sequence, Lineage SparkListener may only capture the lineage for the first write operation:
  - For optimization purposes, Spark's internals may reuse execution plans definition for consecutive write operations on the same DataFrame. This can lead to only capturing first lineage event.

Troubleshooting steps:

1. Lineage graph can only be visualized if at least one node of the graph is an asset. Therefore, create assets for any of the datasets (such as tables) involved in the job and then attempt to visualize lineage on the asset.

2. First, invoke ListLineageEvents to see if the lineage events were submitted (refer to the linked doc to pass filters).
3. If no events are submitted, check AWS CloudWatch logs to see if any exceptions are thrown from the Amazon DataZone Lineage lib:
   - Log groups: /aws-glue/jobs/error, /aws-glue/sessions/error
   - Make sure logging is enabled: https://docs.aws.amazon.com/glue/latest/dg/monitor-continuous-logging-enable.html
   - Following is the AWS CloudWatch log insights query to check for exceptions:

```
fields @timestamp, @message
| filter @message like /(?i)exception/ and like /(?i)datazone/
| sort @timestamp desc
```

   - Following is the AWS CloudWatch log insights query to confirm events are submitted:

```
fields @timestamp, @message
| filter @message like /Successfully posted a LineageEvent:/
| sort @timestamp desc
```

4. Fetch the lineage events generated from this job by executing the python script: retrieve_lineage_events.py
5. Check if the dataset on which you expected lineage is present in any of the events
   - You can ignore empty events without any input/output nodes
   - Check if your dataset node has glue arn prefix in the namespace of the node or in the "symlink" facet of the node. If you don't see any node with glue arn prefix, it means your script is not using glue tables directly and hence lineage is not linked to glue asset. One way to workaround this is to update the script to work with glue tables.
6. If you are still unable to see lineage and it doesn't fall under the limitations category, reach out to AWS Support by providing:
   - Spark config parameters
   - Lineage events file from executing retrieve_lineage_events.py script

- GetAsset response

Troubleshooting lineage for lineage events published from EMR-S/EC2/EKS

Notes:

- Lineage is supported from following versions of EMR:
  - EMR-S: 7.5+
  - EMR-EC2: 7.11+
  - EMR-EKS: 7.12+
- Lineage capture for Spark jobs with fine-grained permission mode are not supported.
- If you are trying EMR outside of Amazon SageMaker Unified Studio, Amazon DataZone VPC endpoint needs to be deployed for EMR VPC.
- Lineage event has a size limit of 300KB.

Common Issues:

- Verify necessary permissions are given to your job execution role as per Permissions required for data lineage
- In case your domain is using a CMK, make sure that the job's execution role has the appropriate KMS permissions. CMK can be found via https://docs.aws.amazon.com/datazone/latest/APIReference/API_GetDomain.html
- Failed to publish Lineage Event because payload is greater than 300 kb:
  - Add the following to Spark conf which efficiently generates the event payload:

    ```
    "spark.openlineage.columnLineage.datasetLineageEnabled": "true"
    ```
- Cross account lineage event submission:
  - Follow https://docs.aws.amazon.com/datazone/latest/userguide/working-with-associated-accounts.html to set up account association
  - Ensure that RAM policy is using the latest policy
- If your Amazon SageMaker Unified Studio domain is in different region from that of job

- Add this spark parameter: `"spark.openlineage.transport.region":"{region of your domain}"`

- When the same DataFrame is written to multiple destinations or formats in sequence, Lineage SparkListener may only capture the lineage for the first write operation:
  - For optimization purposes, Spark's internals may reuse execution plans definition for consecutive write operations on the same DataFrame. This can lead to only capturing first lineage event.

Troubleshooting steps:

1. Lineage graph can only be visualized if at least one node of the graph is an asset. Therefore, create assets for any of the datasets (such as tables) involved in the job and then attempt to visualize lineage on the asset.
2. First, invoke ListLineageEvents to see if the lineage events were submitted (refer to the linked doc to pass filters).
3. If no events are submitted, check AWS CloudWatch logs to see if any exceptions are thrown from the Amazon DataZone Lineage lib:
   - **EC2:**
     - You can provide the CloudWatch log group or log destination for S3 path at the time of creating EC2 cluster. Refer https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-plan-debugging.html
     - You will see logs in stderr file within cluster-id/containers/application*/ folder.
   - **EKS:**
     - You need to provide the CloudWatch log group or log destination for S3 path while submitting spark job. Refer https://docs.aws.amazon.com/emr/latest/EMR-on-EKS-DevelopmentGuide/getting-started.html
     - You will see logs in stderr file of spark_driver within virtual-cluster-id/jobs/job-id/containers/* folder.
   - **EMR-S:**
     - You can enable logs by following https://docs.aws.amazon.com/emr/latest/EMR-Serverless-UserGuide/logging.html
     - You will see logs in stderr files of spark_driver
   - Following is the AWS CloudWatch log insights query to check for exceptions:

```
fields @timestamp, @message
| filter @message like /(?i)exception/ and like /(?i)datazone/
| sort @timestamp desc
```

- Following is the AWS CloudWatch log insights query to inspect generated events:

```
fields @timestamp, @message
| filter @message like /Successfully posted a LineageEvent:/
| sort @timestamp desc
```

4. Fetch the lineage events generated from this job by executing the python script: retrieve_lineage_events.py
5. Check if dataset on which you expected lineage is present in any of the events
   - You can ignore empty events without any input/output nodes
   - Check if your dataset node has namespace/name matching with the sourceIdentifier of the asset. If you don't see any node with asset's sourceIdentifier, refer to following docs on how to fix it:
     - The importance of the sourceIdentifier attribute to lineage nodes
     - Linking dataset lineage nodes with assets imported into Amazon SageMaker Unified Studio
6. If you are still unable to see lineage and it doesn't fall under the limitations category, reach out to AWS Support team by providing:
   - Spark config parameters
   - Lineage events file from executing retrieve_lineage_events.py script
   - GetAsset response

Third-party business data catalog integrations

Amazon SageMaker Unified Studio supports metadata synchronization with third-party business data catalog platforms. These integrations keep catalog metadata aligned between Amazon SageMaker Catalog and partner platforms. Teams get a consistent view of their data and AI assets regardless of which tool they use day to day.

With these integrations, you can synchronize key metadata elements such as projects, assets, descriptions, glossary terms, and their hierarchies. Organizations can maintain aligned glossary terms, asset descriptions, and ownership information across platforms without manual reconciliation.

Amazon SageMaker Unified Studio currently integrates with the following third-party catalog platforms:

- **Atlan** – Bidirectional metadata synchronization between Amazon SageMaker Catalog and Atlan.
- **Collibra** – Bidirectional metadata synchronization and access request workflow integration between Amazon SageMaker Catalog and Collibra.
- **Alation** – Metadata extraction from Amazon SageMaker Catalog into Alation.

Topics

- Atlan integration
- Collibra integration
- Alation integration

Atlan integration

The integration between Amazon SageMaker Catalog and Atlan enables bidirectional metadata synchronization across both platforms. Atlan is a data workspace that helps business users, analysts, and engineers collaborate on data projects. This integration connects teams working in Atlan with technical teams working in Amazon SageMaker Unified Studio for analytics and machine learning. For detailed setup instructions, see Unifying governance and metadata across Amazon SageMaker Unified Studio and Atlan.

Capabilities

The Atlan integration supports the following capabilities:

- On-demand and scheduled bidirectional metadata synchronization.
- Synchronization of glossary terms and descriptions, including parent-child relationships.
- Ingestion of projects, published and subscribed assets, domains, data products, metadata forms, and column descriptions from Amazon SageMaker Catalog into Atlan.
- Automatic association of glossary terms with related data assets.
- Real-time reverse sync of metadata updates from Atlan back to Amazon SageMaker Catalog.

How it works

The integration uses AWS Identity and Access Management roles to establish a secure connection between your AWS account and Atlan. You deploy an AWS CloudFormation template that creates the required IAM role and policies. This role follows the principle of least privilege, granting Atlan access only to the resources required for cataloging and governance.

After you configure the connection, the Atlan connector calls Amazon SageMaker Unified Studio APIs to ingest assets and metadata. The connector transforms ingested assets into Atlan's metadata model, making them discoverable and governable inside Atlan. When users update metadata in Atlan, the real-time reverse sync pipeline detects changes and pushes updates back to Amazon SageMaker Catalog.

You set up this integration by configuring a connection to Amazon SageMaker Unified Studio from within Atlan.

Collibra integration

The integration between Amazon SageMaker Catalog and Collibra provides bidirectional metadata synchronization and access governance across both platforms. Collibra is a data intelligence platform that helps organizations centralize governance workflows, define business glossaries, and enforce policies across data assets. This integration is available as an open-source solution on GitHub, co-developed by AWS and Collibra. For detailed setup instructions, see Unifying metadata governance across Amazon SageMaker and Collibra.

Capabilities

Metadata synchronization

The Collibra integration synchronizes the following metadata between Amazon SageMaker Catalog and Collibra:

- Bidirectional synchronization of glossary terms and descriptions.
- Preservation of glossary structure, including parent-child relationships.
- Association of terms with data assets such as datasets, tables, and columns.
- Synchronization of classifications, data categories, and tags.
- Alignment of technical descriptions for datasets and columns.

Core metadata elements synchronize every 5 minutes. Subscription requests that originate in Amazon SageMaker Catalog synchronize to Collibra instantly.

Access request workflows

The Collibra integration extends Collibra's access governance workflows to assets cataloged in Amazon SageMaker Catalog. Users can discover and request access to datasets from within Collibra or Amazon SageMaker Unified Studio using familiar approval processes.

Key capabilities of the access request workflow include:

- Access request initiation from either Collibra or Amazon SageMaker Unified Studio.
- Centralized review and approval managed within Collibra by designated business stewards.
- Automatic access provisioning through the Amazon SageMaker Catalog grant mechanism.
- Status tracking of subscription requests across both platforms.

How it works

The integration uses the APIs of both Amazon SageMaker and Collibra Data Governance Center. You deploy an AWS CloudFormation template that provisions the required AWS resources, including IAM roles and AWS Lambda functions. On the Collibra side, you configure operating model changes, import workflows, and assign business stewards to assets.

The solution is available as an open-source project on GitHub.

Alation integration

The integration between Amazon SageMaker Catalog and Alation synchronizes catalog metadata between both systems. Alation is a data intelligence platform that helps organizations make data discoverable, governed, and actionable. This integration creates a unified metadata experience where technical teams working in Amazon SageMaker Unified Studio and business teams working in Alation collaborate on top of the same metadata. For detailed setup instructions, see Build a trusted foundation for data and AI using Alation and Amazon SageMaker Unified Studio.

Capabilities

The current phase of the Alation integration extracts metadata from Amazon SageMaker Catalog into Alation. The integration synchronizes the following metadata:

- Domains, projects, and asset names.
- Descriptions, owners, and glossary terms.
- Custom metadata fields (metadata forms).
- Provenance metadata, including the originating service, the actor who made the change, and the timestamp.

You can run metadata extractions on demand or schedule them to run automatically. The system performs an initial bulk extraction and then keeps data current through incremental updates.

How it works

The integration connects through AWS Identity and Access Management authentication. You can use either an IAM role (recommended) or an IAM user with access keys. The connector uses scoped IAM permissions following least-privilege principles. Communication uses encrypted APIs, and only metadata is synchronized. Your data files and artifacts remain in their original AWS locations.

You set up this integration by installing the SageMaker enhanced connector in Alation and configuring a data source connection.

Catalog and governance use cases

The following use cases walk through real-world scenarios that show how to combine catalog and governance features in Amazon SageMaker Unified Studio to solve common data management challenges. Each use case demonstrates an end-to-end workflow that ties together multiple features such as glossaries, metadata forms, AI-assisted curation, and access management.

Topics

- From raw tables to a governed data catalog

From raw tables to a governed data catalog

Your AWS Glue Data Catalog likely has dozens or hundreds of tables with names like `rtl_cust_churn_30d`, `daily_sales_agg_v3`, and `cust_ltv_scores_202401`. Even if some of these have descriptions at the AWS Glue level, they may lack the standardized business terminology, structured metadata, and governed access controls that help people across the organization find and trust the right data.

The business catalog in Amazon SageMaker Unified Studio adds a governance and discovery layer on top of your AWS Glue Data Catalog. You can define shared business glossaries, attach structured metadata forms, use AI to generate business context at scale, and manage access through subscriptions. The result is that tables become discoverable by business concepts (not just technical names) and accessible through a governed request flow.

This page walks through a realistic example of how to approach this for a set of retail analytics tables.

The scenario

Suppose you have these tables in your AWS Glue Data Catalog, already visible in the **Data** tab of your project:

Example retail analytics tables

Technical table name	What it actually contains
`rtl_cust_churn_30d`	Customer churn predictions (30-day window)

Technical table name	What it actually contains
`daily_sales_agg_v3`	Daily sales aggregates by store and category
`cust_ltv_scores_202401`	Customer lifetime value scores, January 2024 refresh
`inv_reorder_pts`	Inventory reorder point calculations

Right now, these tables may have short descriptions and comments, but they lack standardized business terminology, structured metadata (like freshness or quality tier), and a governed access request flow. Someone outside your immediate team would need to know the exact table name or ask around to find the right data.

The approach

Building a business catalog for these tables involves three layers of work, each building on the previous one.

Set up a governance framework first

Before curating individual tables, define the governance constructs that will keep your metadata consistent and searchable across all tables. Think of this as building the vocabulary and structure that consumers will use to find and evaluate data.

Business glossary: Give people a shared language to search with

Without a glossary, different teams use different words for the same concept. Your ML team says "churn," marketing says "attrition," and the exec dashboard says "customer loss." When someone searches the catalog, they shouldn't need to guess which synonym the data engineer used.

A glossary solves this by defining canonical terms with precise definitions. When you attach a glossary term to a table, that term becomes a searchable attribute. Someone searching "Churn Rate" will find every table tagged with that term, regardless of the table's technical name.

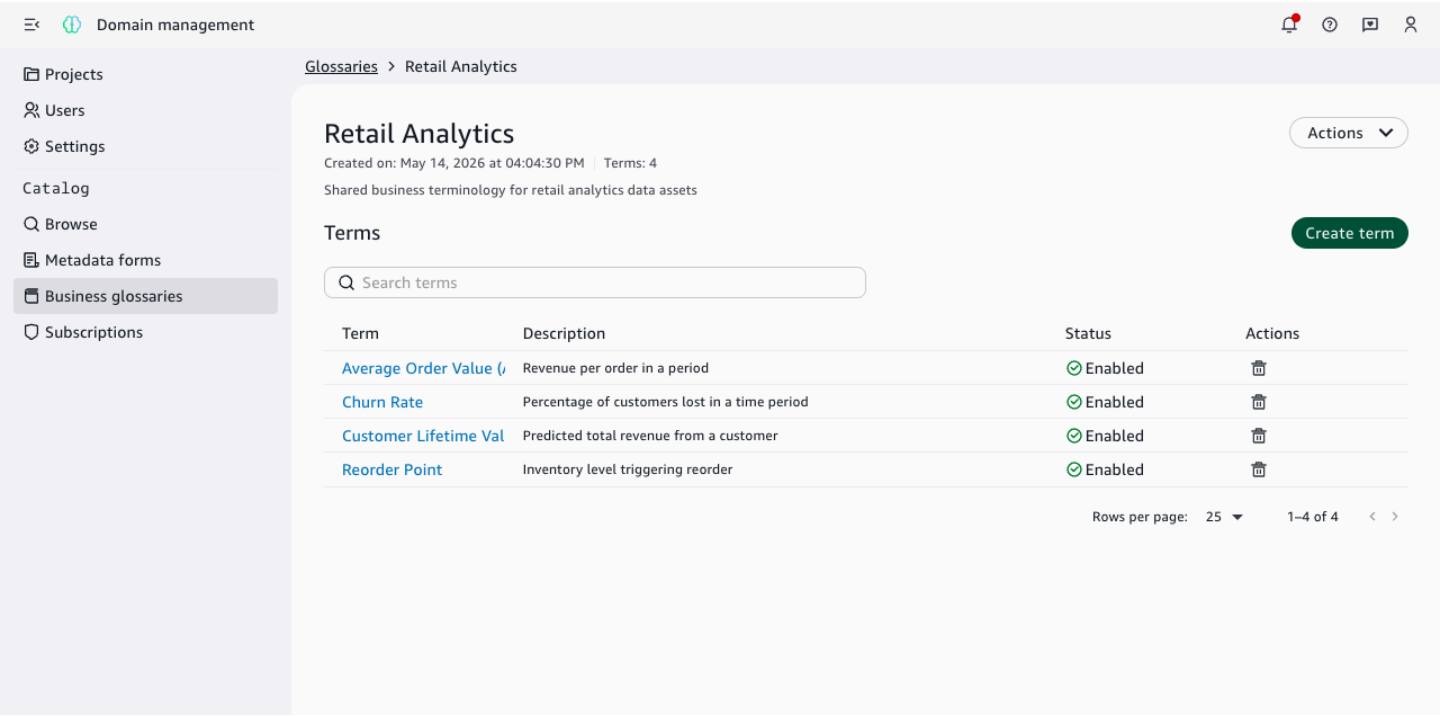
For this retail analytics scenario, create a glossary called "Retail Analytics" with terms like:

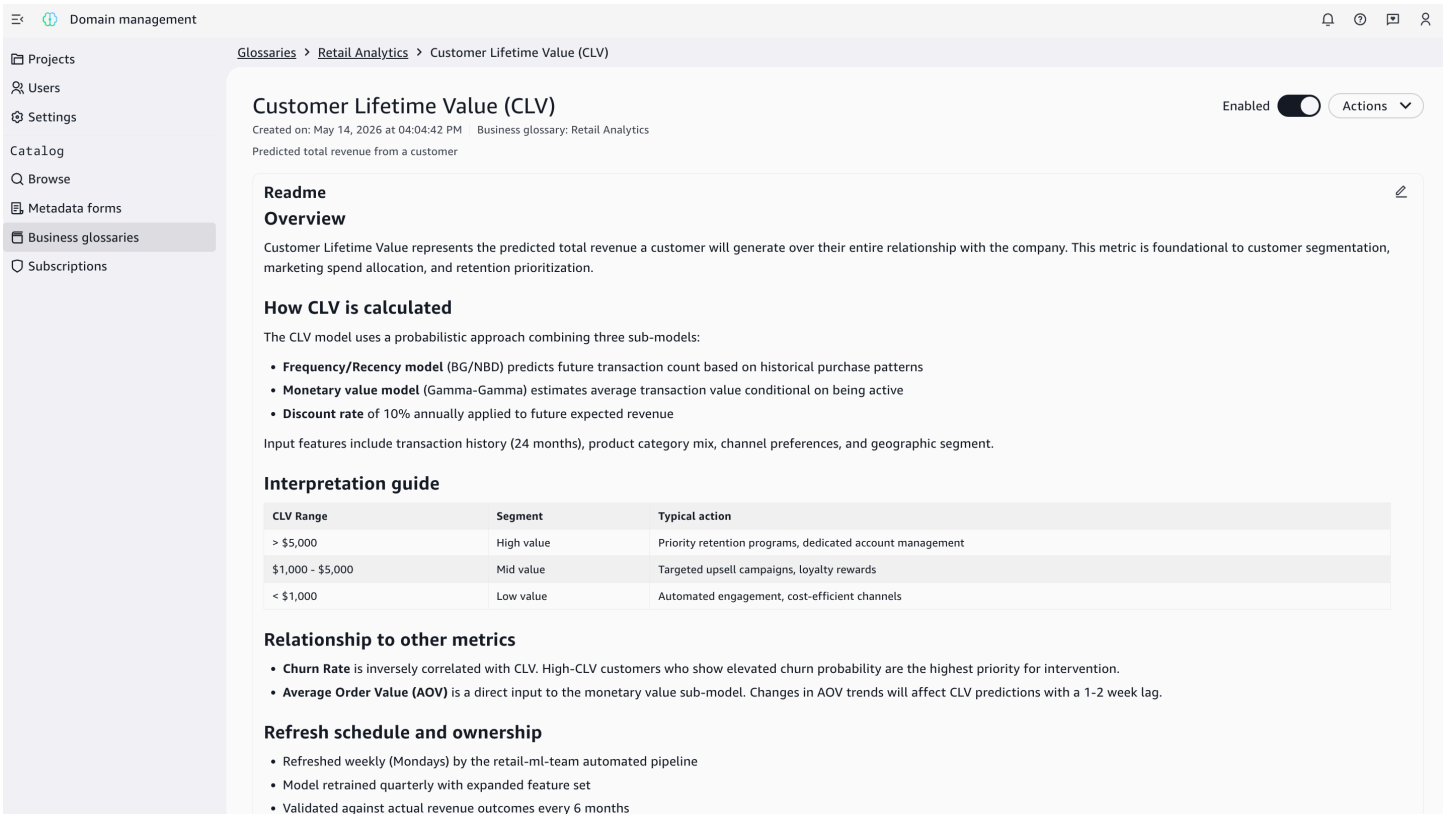
- **Churn Rate** – The percentage of customers who stop purchasing within a defined time window. Calculated as (customers lost / total customers at start of period).

- **Customer Lifetime Value (CLV)** – The predicted total revenue a customer will generate over their entire relationship with the company.
- **Average Order Value (AOV)** – Total revenue divided by number of orders in a given period.
- **Reorder Point** – The inventory level at which a new purchase order should be placed to avoid stockouts.

The consumer payoff: A finance analyst searching "Average Order Value" immediately finds `daily_sales_agg_v3` because you tagged it with the AOV term. They choose the term definition and see the exact formula, so they know whether it matches their needs before requesting access.

To create a business glossary and add terms, see Create and manage business glossaries.





Metadata forms: Structured fields that help people evaluate data before requesting access

When someone finds a table in search results, their next questions are usually practical: How fresh is this data? Who owns it? Is it production-quality or experimental? Can I trust it for a board report?

Metadata forms let you capture these answers as structured, searchable fields. Unlike free-text descriptions, form values can be filtered on. A consumer can narrow search results to "only tables with Gold quality tier" or "only tables refreshed daily."

For this scenario, create a form called "Data Quality & Freshness" with:

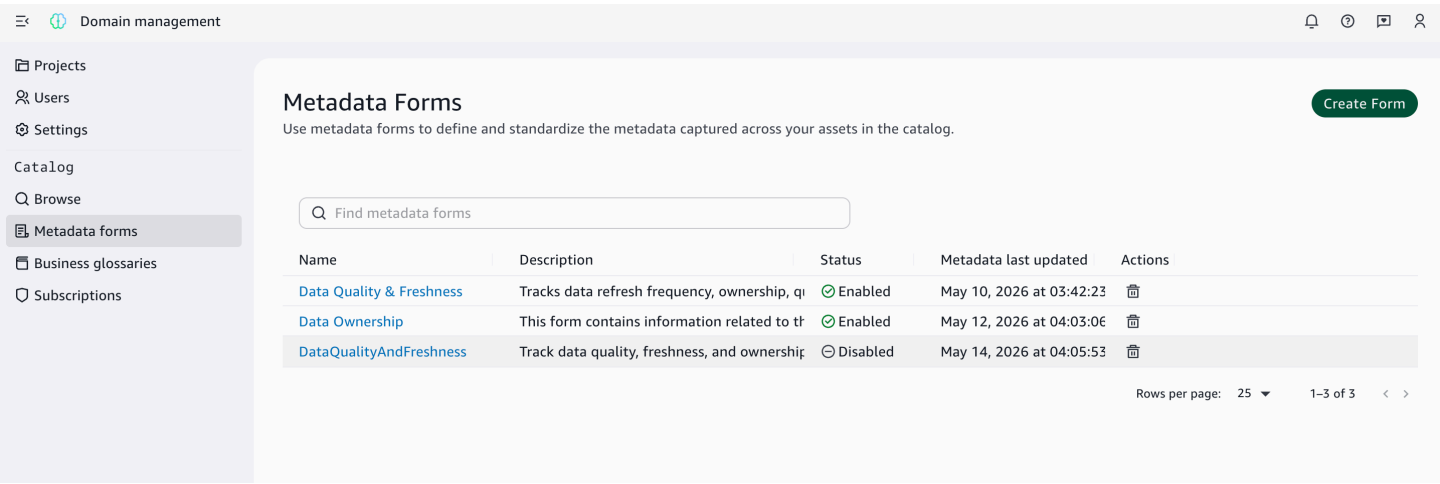
Data Quality and Freshness form fields

Field	Type	What it tells the consumer
Refresh Frequency	Text or Choice	"Can I use this for a real-time dashboard, or is it monthly?"
Last Validated Date	Date	"When did someone last confirm this data is accurate?"

Field	Type	What it tells the consumer
Data Owner	Text	"Who do I contact if something looks wrong?"
Contains PII	Boolean	"Does this data contain personally identifiable information?"
Quality Score	Number	"How reliable is this data on a 0-100 scale?"

The consumer payoff: Before even requesting access, someone can see that `rtl_cust_churn_30d` has a quality score of 92, is refreshed weekly, owned by the retail ML team, and does not contain PII. They can make an informed decision about whether to subscribe without messaging anyone.

To create a metadata form, see Standardize metadata using metadata forms.



Curate tables using AI, then refine manually

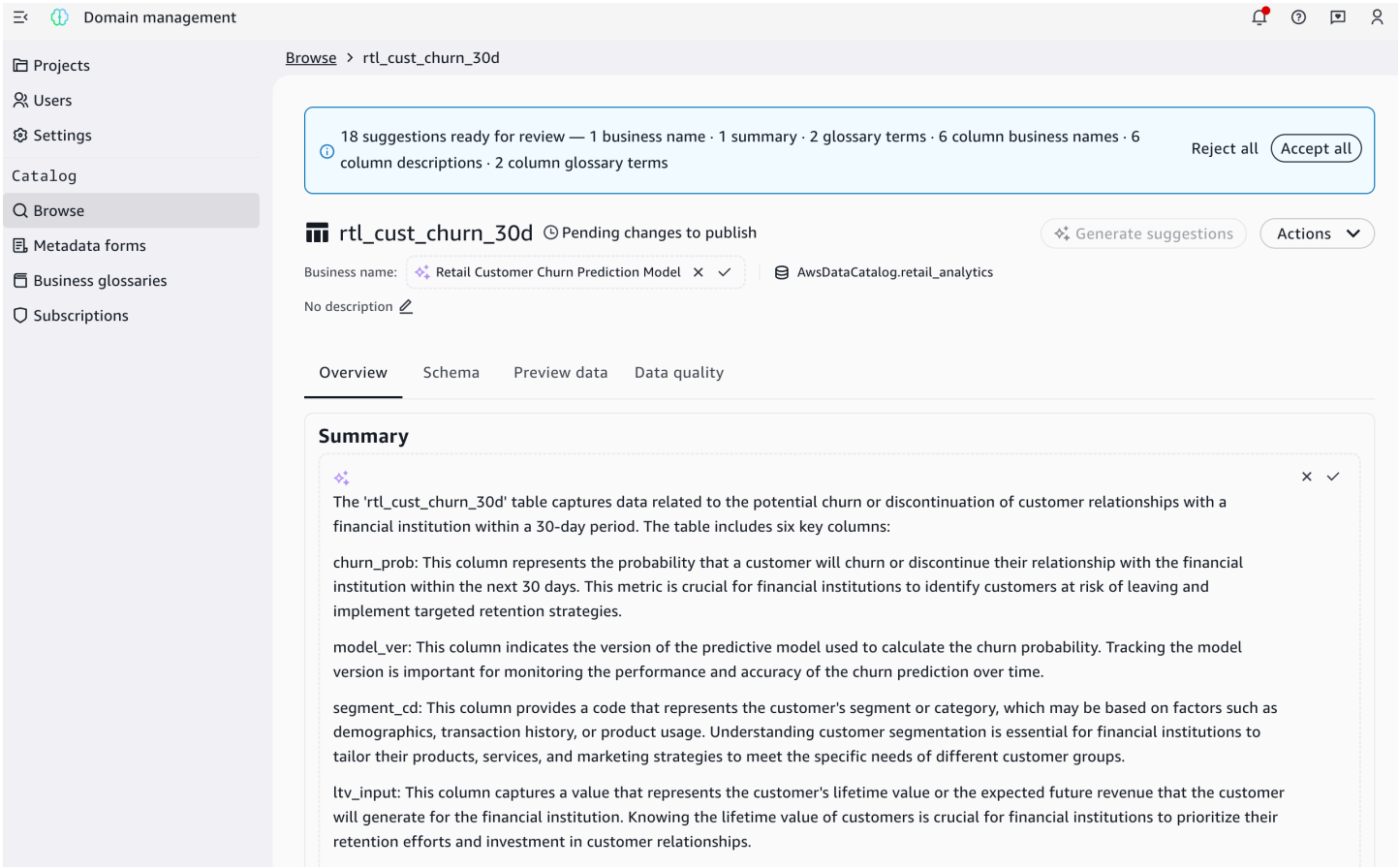
With your glossary and forms in place, curate each table. The most efficient approach is AI-first.

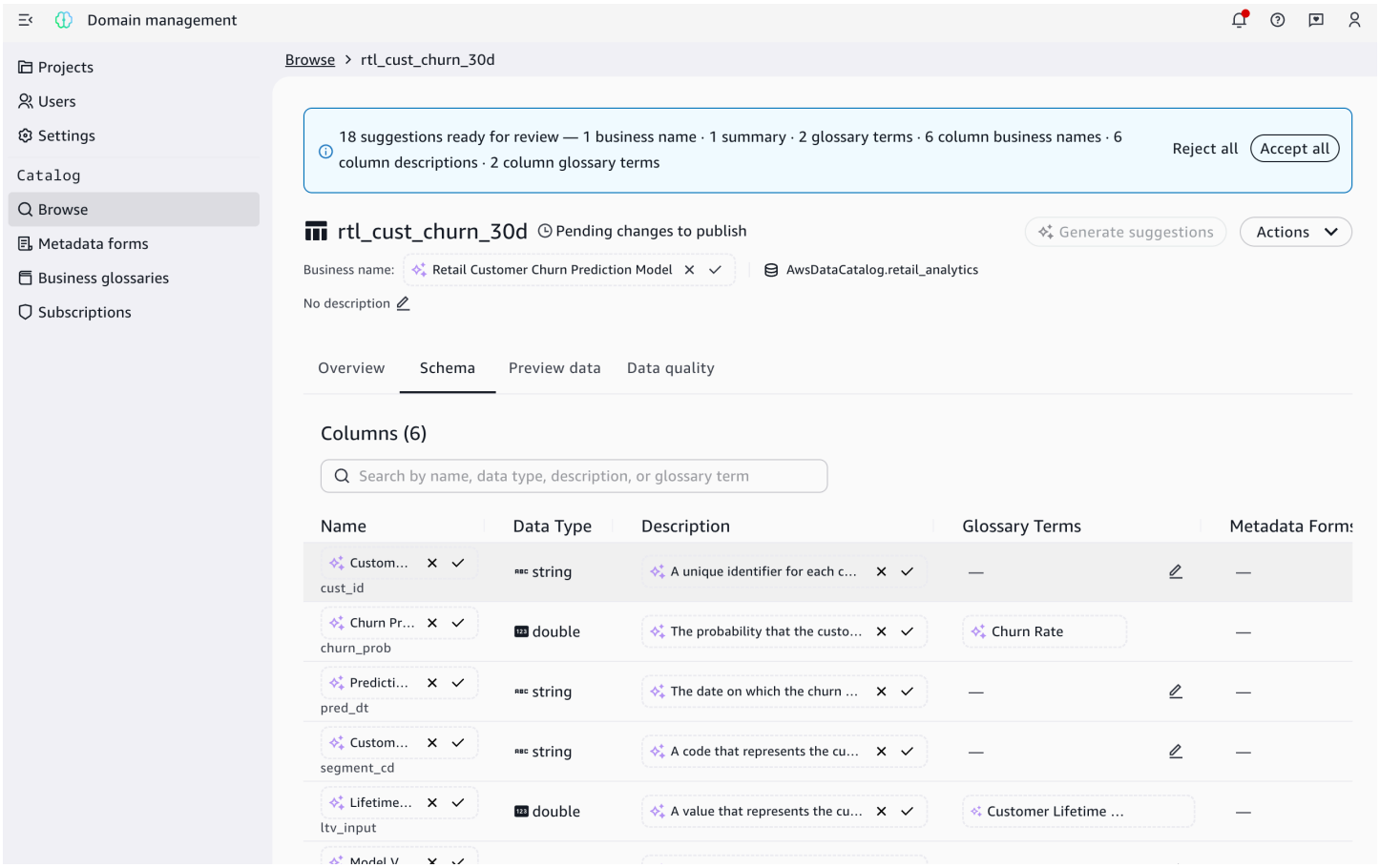
Start with AI suggestions. On any table's asset detail page in the admin portal, choose **Generate suggestions**. AI analyzes the table's column names, data types, and sample values, then proposes:

- A business name (for example, `rtl_cust_churn_30d` becomes "Retail Customer Churn Prediction Model")
- A business description
- Business names and descriptions for each column

- Glossary term associations

AI handles roughly 80% of the work well. It's particularly good at expanding abbreviated column names (`cust_id` becomes "Customer ID", `churn_prob` becomes "Churn Probability") and writing reasonable first-draft descriptions.





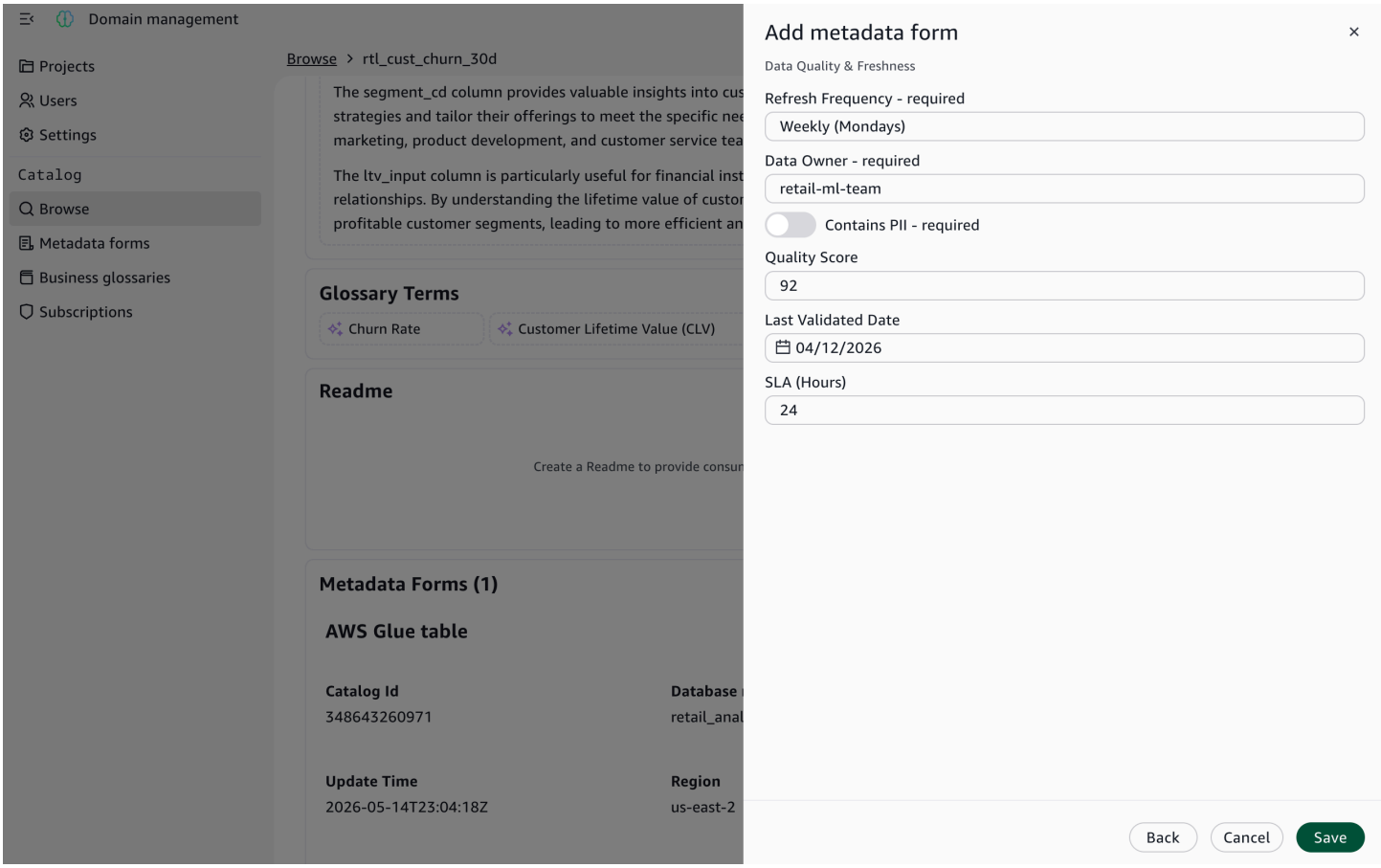
Refine with domain expertise. The 20% that needs human input:

- **Precision in business names.** AI suggested "Retail Customer Churn Prediction Model" which is good, but you might refine it further if your organization has specific naming conventions.
- **Context AI can't infer.** Add to the description that "Predictions are generated weekly by the ML team using XGBoost trained on 24 months of purchase history." AI can't know this from the schema.
- **Glossary term associations AI missed.** Manually attach "Customer Lifetime Value (CLV)" to the churn table because CLV is an input feature, even though the table name doesn't mention it.

Fill in the metadata form. Attach your "Data Quality & Freshness" form and fill in the values:

- Refresh Frequency: Weekly (Mondays)
- Last Validated Date: 2026-04-12
- Data Owner: retail-ml-team
- Contains PII: No

- Quality Score: 92



Write a README for tables that have important context beyond what fits in a description, such as known limitations, upstream dependencies, or interpretation guidance.

For all curation procedures (accepting AI suggestions, editing business names, attaching glossary terms, filling forms, writing READMEs), see Curate and enrich asset metadata.

Publish to make it all visible

Curated metadata stays in draft until you publish. Once published, the business name, description, glossary terms, form values, and README all become visible in catalog search for every project in the domain.

New tables that enter the catalog for the first time are published automatically. You only need to publish manually after making metadata changes.

To publish, see Publish data.

The end result

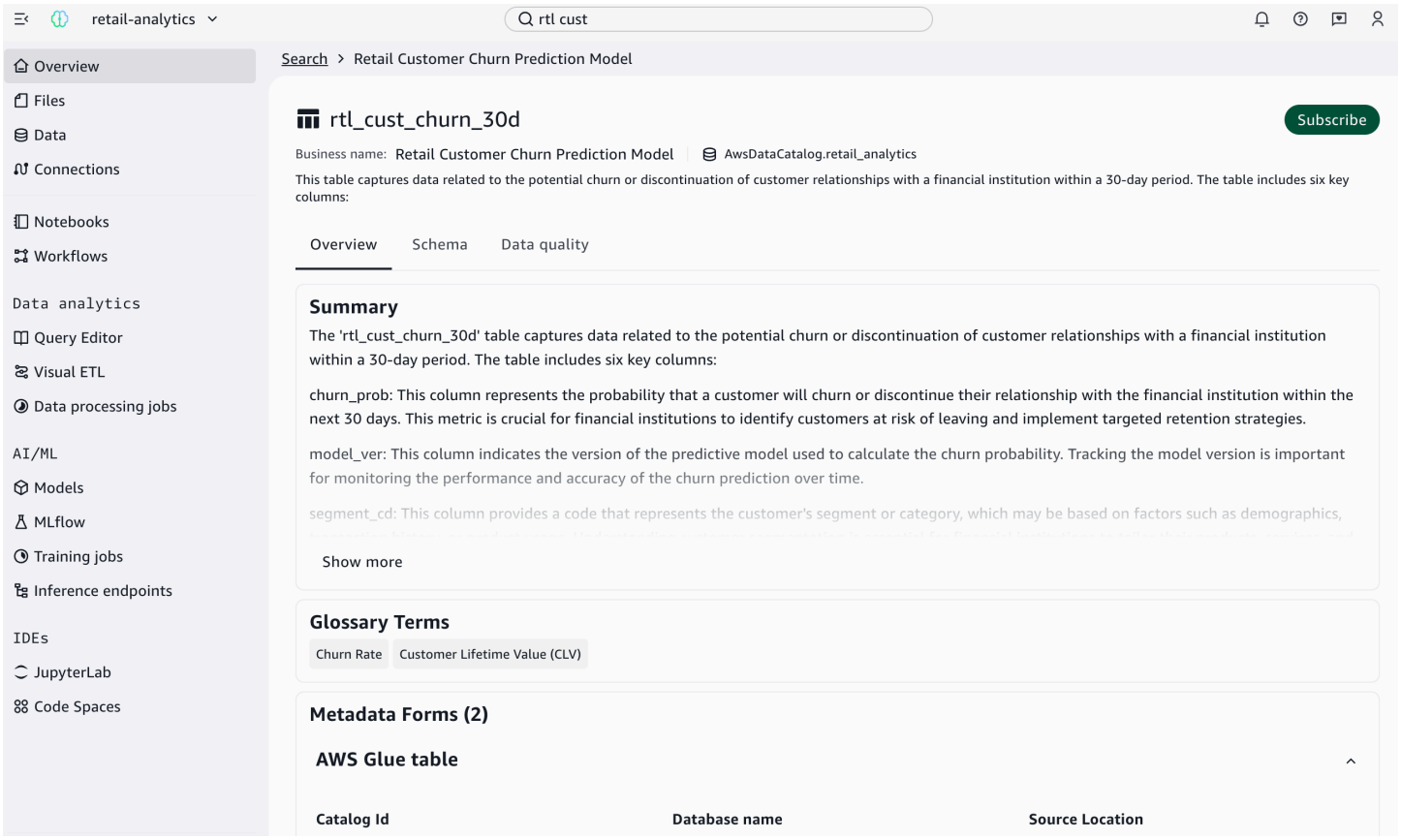
After curating and publishing, here's what changes for `rtl_cust_churn_30d`:

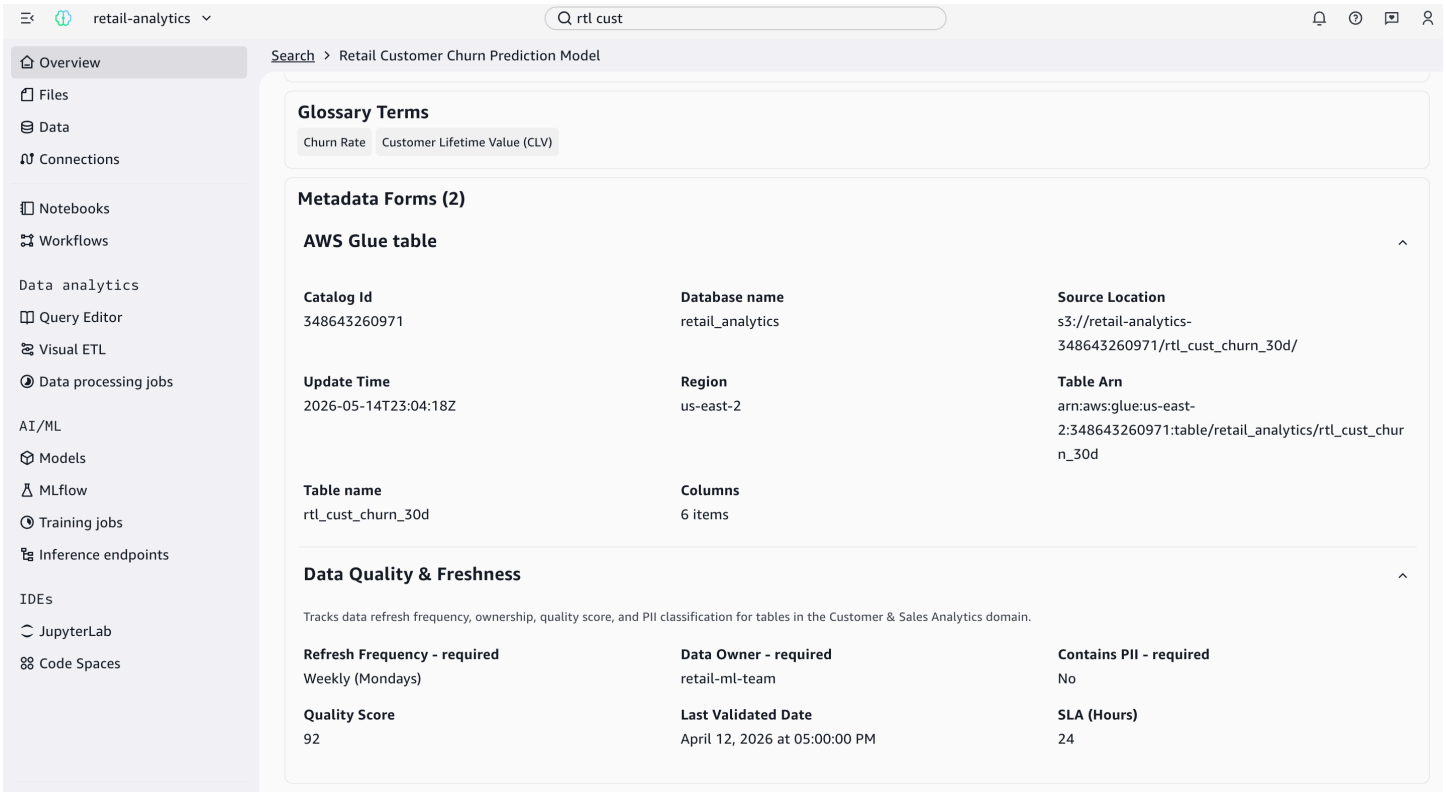
Before:

- Discoverable by table name or basic AWS Glue description (if one exists)
- No standardized business terminology attached
- Column names: `cust_id`, `churn_prob`, `pred_dt`, `segment_cd`, `ltv_input`, `model_ver`
- No structured metadata about freshness, quality, or ownership
- No governed access request flow

After:

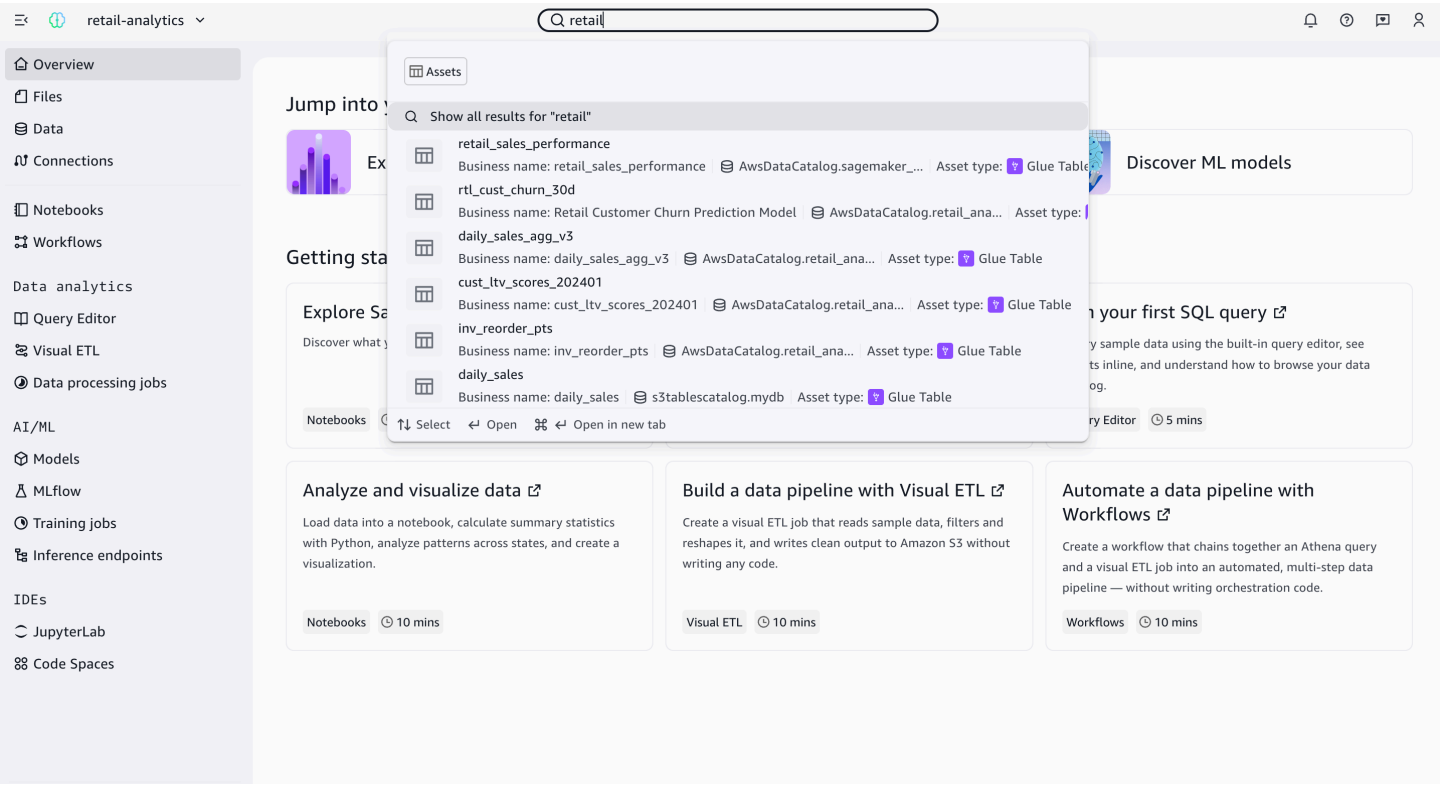
- Business name: "Retail Customer Churn Prediction Model"
- Description explains what the table contains, how predictions are generated, and how often it refreshes
- Column names: Customer ID, Churn Probability, Prediction Date, Customer Segment, LTV Input Score, Model Version
- Glossary terms: Churn Rate, Customer Lifetime Value (CLV)
- Metadata form: Refresh Frequency = Weekly (Mondays), Data Owner = retail-ml-team, Contains PII = No, Quality Score = 92, SLA = 24 hours
- Discoverable by searching "customer churn," "churn predictions," "Churn Rate," or "retail-ml-team"
- README with model details, known limitations, and upstream dependencies


retail-analytics
rtl cust
Overview
Files
Data
Connections
Notebooks
Workflows
Data analytics
Query Editor
Visual ETL
Data processing jobs
AI/ML
Models
MLflow
Training jobs
Inference endpoints
IDEs
JupyterLab
Code Spaces
Search > Retail Customer Churn Prediction Model
rtl_cust_churn_30d
Subscribe
Business name: Retail Customer Churn Prediction Model
AwsDataCatalog.retail_analytics
This table captures data related to the potential churn or discontinuation of customer relationships with a financial institution within a 30-day period. The table includes six key columns:
Overview
Schema
Data quality
Summary
The 'rtl_cust_churn_30d' table captures data related to the potential churn or discontinuation of customer relationships with a financial institution within a 30-day period. The table includes six key columns:
churn_prob: This column represents the probability that a customer will churn or discontinue their relationship with the financial institution within the next 30 days. This metric is crucial for financial institutions to identify customers at risk of leaving and implement targeted retention strategies.
model_ver: This column indicates the version of the predictive model used to calculate the churn probability. Tracking the model version is important for monitoring the performance and accuracy of the churn prediction over time.
segment_cd: This column provides a code that represents the customer's segment or category, which may be based on factors such as demographics,
Show more
Glossary Terms
Churn Rate
Customer Lifetime Value (CLV)
Metadata Forms (2)
AWS Glue table
Catalog Id
Database name
Source Location


retail-analytics
rtl cust
Overview
Files
Data
Connections
Notebooks
Workflows
Data analytics
Query Editor
Visual ETL
Data processing jobs
AI/ML
Models
MLflow
Training jobs
Inference endpoints
IDEs
JupyterLab
Code Spaces
Search > Retail Customer Churn Prediction Model
Glossary Terms
Churn Rate
Customer Lifetime Value (CLV)
Metadata Forms (2)
AWS Glue table
Catalog Id
348643260971
Database name
retail_analytics
Source Location
s3://retail-analytics-348643260971/rtl_cust_churn_30d/
Update Time
2026-05-14T23:04:18Z
Region
us-east-2
Table Arn
arn:aws:glue:us-east-2:348643260971:table/retail_analytics/rtl_cust_churn_30d
Table name
rtl_cust_churn_30d
Columns
6 items
Data Quality & Freshness
Tracks data refresh frequency, ownership, quality score, and PII classification for tables in the Customer & Sales Analytics domain.
Refresh Frequency - required
Weekly (Mondays)
Data Owner - required
retail-ml-team
Contains PII - required
No
Quality Score
92
Last Validated Date
April 12, 2026 at 05:00:00 PM
SLA (Hours)
24

Now a marketing analyst searching "customer churn" finds this table immediately, reads the description to confirm it's what they need, checks the quality tier and refresh frequency, and requests access through a subscription. The entire discovery-to-access flow works without asking the data engineering team.

For details on how search works from the consumer side, see Search and discover data. For the subscription and access request flow, see Request access to data.



Applying this pattern to your own data

The retail analytics example above is one scenario, but the pattern applies to any domain:

- **Identify your vocabulary.** What are the key business terms that people search for in your domain? Define them in a glossary so everyone uses the same language.
- **Decide what structured metadata matters.** What questions do people ask before deciding to use a table? (How fresh is it? Who owns it? Is it production-quality?) Capture those in a metadata form.
- **Let AI do the first pass.** Generate suggestions for all your tables. Accept what's good, refine what's close, reject what's wrong.

- **Add the human context.** Fill in what AI can't know: business rules, ownership, quality assessments, known limitations.
- **Publish and verify.** Check that the tables appear correctly in search from a consumer's perspective.

The goal is not to curate every table perfectly on day one. Start with your most-used or most-requested tables, get them into the catalog with good business context, and expand from there.

aws

User Guide

Amazon SageMaker Unified Studio

Reference and resources



Amazon SageMaker Unified Studio terminology and concepts

As you get started with Amazon SageMaker Unified Studio, it is important that you understand its key concepts, terminology, and components.

Amazon SageMaker Unified Studio

This is a browser-based web application where you can use all your data and tools for analytics and AI. Amazon SageMaker Unified Studio can authenticate you with your IAM user credentials or with credentials from your identity provider through the IAM Identity Center or with your SAML credentials. You can obtain the Amazon SageMaker Unified Studio URL for your domains by accessing the SageMaker AI management console at https://console.aws.amazon.com/datazone.

Amazon SageMaker management console

You can use the SageMaker AI management console at https://console.aws.amazon.com/datazone to access and configure your domains for user management, account associations, project profiles, blueprints, Amazon Bedrock models, Git connections, and Amazon Q usage.

Amazon Bedrock in SageMaker Unified Studio

Use Amazon Bedrock in SageMaker Unified Studio to build and scale generative AI applications. Amazon Bedrock in SageMaker Unified Studio provides a web interface that allow users to interact with Amazon Bedrock foundation models and use Amazon Bedrock tools, such as agents, guardrails, prompts, flows, evaluation, and functions in a seamless unified fashion. Users can interact with models in a generative AI playground or collaborate on developing generative AI applications in projects.

Amazon Q Developer

Amazon Q Developer is an AI coding assistant that can chat about code, provide inline code completions, generate net new code, scan your code for security vulnerabilities, and make code upgrades and improvements.

In the current release of Amazon SageMaker Unified Studio, by default, all users of an Amazon SageMaker Unified Studio domain have access to the Free Tier release of Amazon Q.

lakehouse architecture

lakehouse architecture unifies your data across Amazon S3 data lakes and Amazon Redshift data warehouses. lakehouse architecture helps you build powerful analytics, machine learning (ML), and generative AI applications on a single copy of data.

lakehouse architecture is accessible via Amazon SageMaker Unified Studio.

Amazon SageMaker Data Processing Visual ETL

In Amazon SageMaker Unified Studio you can author highly scalable extract, transform, load (ETL) data integration flows for distributed processing without becoming an Apache Spark expert. You can define your data integration flow in the simple visual interface and Amazon SageMaker Unified Studio automatically generates the code to move and transform your data. The code is generated in Python and written for Apache Spark. Additionally, you can choose to author your visual flows in English using generative AI prompts from Amazon Q.

Asset

In Amazon SageMaker Unified Studio, an asset is an entity that presents a single physical data object (for example, a table, a dashboard, a file) or virtual data object (for example, a view).

Asset type

Asset types define how assets are represented in the Amazon SageMaker Catalog. An asset type defines the schema for a specific type of asset. When assets are created, they are validated against the schema defined by their asset type (by default, the latest version). When an asset update occurs, Amazon SageMaker Unified Studio creates a new asset version and enables Amazon SageMaker Unified Studio users to operate on all asset versions.

Associated accounts

Use account associations in Amazon SageMaker Unified Studio to publish data from other AWS accounts into the Amazon SageMaker Catalog and create projects to work with data across multiple AWS accounts. AWS accounts where Amazon SageMaker unified root domains are created initiate the account association requests. You can request association from the Amazon SageMaker management console. Account association requests must be accepted by the administrators of the AWS accounts invited for account association. You can authorize the domain account to use data or allow infrastructure deployment with the right IAM permissions as part of approval. Once an associated account is linked to a domain, projects in Amazon SageMaker Unified Studio can use resources from those accounts and also other types of assets. You can deploy resources in specific AWS accounts through project profiles.

Authorization policy

Authorization policies are a set of controls within Amazon SageMaker Unified Studio applied to entities such as projects, blueprints, environments, glossary, and metadata forms.

Within an Amazon SageMaker Unified Studio domain unit, you can assign the following authorization policies to your users and groups to grant them specific permissions:

- Domain unit creation policy
- Project creation policy
- Project membership policy
- Domain unit ownership assumption policy
- Project ownership assumption policy

Within an Amazon SageMaker Unified Studio domain unit, you can assign the following authorization policies to your projects to grant them specific permissions:

- Glossary creation policy
- Metadata forms creation policy
- Custom asset type creation policy

Within a specific blueprint configuration, you can assign the following authorization policies to projects and domain unit owners:

- Create environment profiles using this blueprint. This policy can be assigned to Amazon SageMaker Unified Studio projects, and it authorizes them to create environment profiles using this blueprint.
- Grant permissions to create environment profiles using this blueprint . This policy can be assigned to domain unit owners and it authorizes them to grant permissions to projects to create environment profiles using this blueprint.

AWS account owner

In Amazon SageMaker Unified Studio, AWS account owners create roles, policies, and permissions in their AWS accounts that enable these AWS accounts to be associated with Amazon SageMaker Unified Studio domains.

Blueprint

A blueprint is used to create the project profile that defines which AWS tools and services project members can use as they work with data in the Amazon SageMaker Catalog.

In the current release of Amazon SageMaker Unified Studio the following default blueprints are supported:

Blueprint name	Description	Resources created
AmazonBedrockGenerativeAI	This is the combined Amazon Bedrock blueprint which contains seven sub-Amazon Bedrock blueprints. Users create project profiles with this blueprint to build generative AI applications using tools such as agents, knowledge bases, guardrails, flows, functions, and model evaluation.	
AmazonBedrockChatAgent	Provides a reusable AWS CloudFormation template to create an Amazon Bedrock Agent and supportin g resources, including an execution role and a consumption role.	Bedrock Agent, Bedrock Agent Execution role, Bedrock Agent Consumption role
AmazonBedrockEvaluation	Creates one IAM role as the service role for an Amazon Bedrock evaluation job.	Bedrock Evaluation job execution role
AmazonBedrockFlow	Provides a reusable AWS CloudFormation template to create an Amazon Bedrock Prompt Flow and supportin g resources such as an execution role.	Amazon Bedrock Flow, Amazon Bedrock Flow Execution role
AmazonBedrockFunction	Provides a reusable AWS CloudFormation template	Secrets Manager secret, AWS Lambda function, AWS

Blueprint name	Description	Resources created
	to create an AWS Lamda function and supportin g resources, such as an execution role, and a secret manager.	Lambda function execution role, Log group
AmazonBedrockGuardrail	Provides an AWS CloudForm ation template to create an Amazon Bedrock Guardrail and supporting resources such as an execution role.	Amazon Bedrock Guardrail
AmazonBedrockKnowl edgeBase	Provides an AWS CloudForm ation template to create a reusable Amazon Bedrock Knowledge Base and supporting resources such as an execution role.	Amazon Bedrock Knowledge Base, OpenSearch Serverless collection, Amazon Bedrock Knowledge Base Execution role, AWS Lambdas, including OpenSearch Index Lambda and KB Ingestion Trigger Lambda, AWS Lambda Execution role, Amazon Bedrock Knowledge Base data source
AmazonBedrockPrompt	Provides a reusable AWS CloudFormation template to create an Amazon Bedrock Prompt and supportin g resources, such as an execution role, and a consumption role.	Amazon Bedrock Prompt, Amazon Bedrock Prompt Consumption role

Blueprint name	Description	Resources created
LakeHouseDatabase	Provides a reusable AWS CloudFormation template to create a data lake environment with a AWS Glue database for data management and an Amazon Athena workgroup for querying data.	AWS Glue databases, lake formation permissions, Amazon Athena workgroups
EMROnEC2	Provides a reusable AWS CloudFormation template to create an Amazon EMR on EC2 cluster to run and scale Apache Spark, Hive, and other big data workloads.	EMR on EC2 clusters
EMRServerless	Provides a reusable AWS CloudFormation template to create an Amazon EMR Serverless application that is ready to serve Apache Spark batch jobs and interactive sessions.	EMR on Serverless applications
LakehouseCatalog	Provisions a new catalog in the lakehouse architecture that is backed by Amazon Redshift Managed Storage	
MLExperiments	Provides OnDemand blueprint to enable MLflow tracking server for the experimentation inside a project.	MLflow tracking server (on demand)

Blueprint name	Description	Resources created
PartnerApps	Creates an IAM role and a Connection that enables access to Partner AI Apps. Through Partner AI Apps you can leverage integrate d and fully-managed thrid-party solutions for AI/Ml development.	Amazon SageMaker Partner AI Apps IAM role, Amazon SageMaker Partner AI Apps Connection
RedshiftServerless	Provides a reusable AWS CloudFormation template to create an Amazon Redshift Serverless environment to get insights from data without managing infrastru cture.	Amazon Redshift Serverless warehouses
Tooling	Creates resources for the project, including IAM user roles, security groups, and Amazon SageMaker unified domains.	IAM user roles, Amazon SageMaker unified domains, security groups
Workflows	Provides an AWS CloudForm ation template to create the MWAA environment for Airflow based Workflows	Enables project workflows on MWAA

Amazon SageMaker Catalog

This is a catalog of all the published assets from various projects. The scope of the Amazon SageMaker Catalog is the domain, therefore published assets are discoverable by all projects in that domain. The Amazon SageMaker Catalog enables discovery that crosses the account and Region boundary. You can publish assets to the Amazon SageMaker Catalogso that other projects can subscribe to them, or you can subscribe to assets in the catalog that were

published from other projects. Every asset that lives in the Amazon SageMaker Catalog has an owner project (also known as the producer project) which controls policies around how subscriptions can be fulfilled. A subscriber (also known as a consumer project) can make a request to the owner project to gain access to the asset. Once the request is approved, the owner project provides the necessary permissions to the subscriber project so that it may gain access to that asset.

Business glossary

In Amazon SageMaker Unified Studio, a business glossary is a collection of business terms that may be associated with assets. A business glossary helps ensure that the same terms and definitions are used across an organization throughout its various data analytics tasks. You can add terms in a business glossary to assets and columns to classify or enhance the identification of those attributes during search. You can select glossary as the value type for a field in a metadata form that is associated with an asset. When you select a particular term as the value for an asset's metadata form field, users can search for the business glossary term and find the associated assets. Glossaries are owned by the project that creates them, but they are visible to all users in the domain. Users in any project can attach glossary terms to their assets, while only members of the owning project can edit or delete the glossary and its terms.

Git connection

Use git connections to check in and check out files and manage your code repository. When you create an Amazon SageMaker unified domain, a default git connection to CodeCommit is provided for you to manage your code. You can also create and enable new 3P Git connections to GitHub, GitHub Enterprise Server, GitLab, and GitLab Self-Managed.

Data source

An entity which brings in metadata from a source and adds metadata forms (such as the ingestion job). This entity allows publishers to capture ingestion configuration including what metadata forms to attach and whether to run BNG. Since this configuration has a one-to-many mapping with the credentials provided by the publisher, we believe that it should be captured in a separate entity.

In Amazon SageMaker Unified Studio, you can use data sources to import technical metadata of assets (data) from the source databases or data warehouses into Amazon SageMaker Unified Studio. In the current release of Amazon SageMaker Unified Studio, you can create and run data sources for AWS Glue and Amazon Redshift. By creating a data source, you establish a connection between Amazon SageMaker Unified Studio and the source (AWS Glue Data Catalog

or Amazon Redshift Warehouse), which you can then use to read technical metadata, including table names, columns names, and data types. By creating a data source, you also begin the initial data source run that creates new or updates existing assets in Amazon SageMaker Unified Studio. While creating a data source or after the data source is successfully created, you also have the option to specify a schedule for your data source runs.

Data source run

In Amazon SageMaker Unified Studio, a data source run is a task that Amazon SageMaker Unified Studio performs in order to create assets in project inventories and also optionally to publish project inventory assets to the Amazon SageMaker Catalog. Data source runs can be automated (started when a data source is initially created), scheduled, or manual. Use data selection criteria to fine-tune the existing and future data sets to be ingested into project inventories or the Amazon SageMaker Catalog and the frequency of metadata updates to those inventory or catalog assets.

Domain

In Amazon SageMaker Unified Studio, a domain is the organizing entity for connecting together your assets, users, and their projects. With Amazon SageMaker unified domains, you have the flexibility to reflect the data and analytics needs of your organizational structure, whether it's creating a single Amazon SageMaker unified domain for your enterprise or multiple domains for different business units.

Domain administrator

The IAM principal ID that has the super administrative permissions to edit entities in the domain.

In Amazon SageMaker Unified Studio, an IAM principal who creates an Amazon SageMaker Unified Studio domain is the default domain administrator of that domain. Domain administrators in Amazon SageMaker Unified Studio perform key functionalities for the domain, including creating domains, assigning other domain administrators, creating and managing project profiles, configuring blueprints, user management, account associations, Amazon Bedrock models, Git connections, and Amazon Q.

Domain unit

Use domain units to organize your assets and other domain entities under specific business units and teams. To set up secure and efficient data sharing within and across business units of your organization, you can create domain units within Amazon SageMaker Unified Studio and

grant access to selected users within each business unit to log in and share their assets to the Amazon SageMaker Catalog. Domain units can also be used for resource owners, such as AWS account owners, to set up Amazon SageMaker Unified Studio authorization permissions on their resources. Domain units provide a delegated authority from account owners to domain unit owners and they can set up authorization permissions on behalf of account owners.

JupyterLab

Amazon SageMaker Unified Studio provides a JupyterLab interactive development environment (IDE) for you to use as you perform data integration, analytics, or machine learning in your projects. JupyterLab IDE is built on JupyterLab spaces and Amazon SageMaker Distribution.

Metadata form type

A metadata form type is a template that defines the metadata that is collected and saved when assets are created as inventory or published in an Amazon SageMaker unified domain. Metadata form types can be associated with a data asset. Metadata form types help domain administrators to define metadata forms needed for that domain, such as compliance information, regulation information, or classifications. Domain administrators can use this to customize additional metadata for their assets. Amazon SageMaker Unified Studio has system metadata form types such as asset-common-details-form-type, column-business-metadata-form-type, glue-table-form-type, glue-view-form-type, redshift-table-form-type, redshift-view-form-type, s3-object-collection-form-type, subscription-terms-form-type, and suggestion-form-type.

Metadata form

In Amazon SageMaker Unified Studio, metadata forms define the metadata that is collected and saved when assets are created as inventory or published in an Amazon SageMaker unified domain. Metadata form definitions are created in the domain catalog by a domain administrator. A metadata form definition is composed of one or more field definitions, with support for boolean, date, decimal, integer, string, and business glossary field value data types. A domain administrator applies a metadata form to assets in their domain by adding the metadata form to their domain. Asset publishers then provide any optional and required field values in the metadata form.

Project profile

In Amazon SageMaker Unified Studio, a project profile is a template for projects in your Amazon SageMaker unified domains. A project profile is a collection of blueprints, which are configurations used to create projects. A project profile can define if a particular blueprint is

enabled during the creation of the project, or available later for the project users to enable on demand.

You must be an administrator of a Amazon SageMaker Unified Studio domain to create and manage project profiles. In the current release of Amazon SageMaker Unified Studio, you can create the following project profiles:

- Data analytics and AI/ML model development project profile
- SQL analytics project profile
- Generative AI application development project profile
- Custom project profile

Projects in Identity Center-based domains

The project entity is the mechanism by which Amazon SageMaker Unified Studio users organize their work and provide business context over the jobs they are performing. A project is a container for all the user's code, including notebooks, queries, dashboards, workflows, and more. A project provides three capabilities:

1. Business context for the user’s work which provides a level of audit to the functionality being performed.
2. A collaboration boundary where the users can work with each other by interacting with the project’s source control repository.
3. A permissions boundary which gives users access to all the project artifacts and data/compute permissions after the users are added to the project.

A project exists within a domain. A single Amazon SageMaker unified domain can have several projects and each user can be added to multiple projects.

Each project is created using a template called a project profile, which is enabled by an administrator during the setup phase. A project profile controls the tools available within the project. Project members can request access to assets from the Amazon SageMaker Catalog and produce new artifacts using one or more of the tools available inside the project. Artifacts in a project are not accessible outside of the project unless they are published to the Amazon SageMaker Catalog which is discussed later.

Each project has one or multiple owners, who can add or remove other users (called project members) as owners or contributors and can modify or delete projects. Other restrictions on contributors can be defined with policies. When a user creates a project, they become the first owner of that project.

Project S3 path

The purpose of the project S3 path in Amazon SageMaker Unified Studio is to provide a secure, project-isolated location for storing temporary execution data and other project-related artifacts. The project S3 path follows a standardized structure of "<bucket>/<domain_id>/<project_id>/<project_scope>/" to ensure separation between projects and prevent objects from being shared across projects. The project S3 path is also used to store specific types of data, such as the location for the provisioned consumer AWS Glue database, Athena Workgroup output, and temporary storage for individual workflow runs.

Project Git repository

A project includes a dedicated git repository which serves as a central hub for users to manage version control for the code associated with their Amazon SageMaker Unified Studio projects. This enables collaboration across users within a project. All tools that generate file-based assets must use the project git repository for version control, for example, the query editor, JupyterLab IDE, and more. By default, Amazon SageMaker Unified Studio uses AWS CodeCommit as the project's repository which is created when a project is created. However, administrators can modify this to connect a third-party Git repository such as Github, Github Enterprise Server, GitLab, or BitBucket instead of the default repository.

Project member

A project member is any user who has been added to a project and given access to the project data and resources. Users can be enterprise users sourced from the IDP, or IAM Principals from one of the domain associated accounts. Project owners can add members either by adding them directly or by selecting enterprise groups. A project member is added to a project with a designation that defines the set of permissions it has within the project. Users can collaborate on various activities such as accessing data assets, performing data analysis or machine learning activities.

Subscription request

A request to use a data product.

In Amazon SageMaker Unified Studio, a subscription request is a process that an Amazon SageMaker Unified Studio project must follow in order to be granted access to a specific asset. Subscription requests can be approved, rejected, revoked, or granted.

Subscription grant

An object representing a fulfilled request for a particular project.

Querybook

Use querybooks to develop, run, and share multiple SQL queries in a single interactive notebook. They provide an environment for data scientists, analysts, and developers to query, analyze, and visualize data using Amazon Redshift or Amazon Athena as the query engine. Cells in a Querybook contain SQL statements or markdown and can be run individually, like a traditional query editor, or sequentially. Query results appear in-line with each cell, where you can toggle between multiple results and create data visualizations. To accelerate query development, Querybooks integrate with SageMaker Data Agent to generate SQL queries from natural language input, and provide auto-complete suggestions for table names, column names, and SQL keywords as you type. Amazon SageMaker Unified Studio automatically saves your work as you progress. When ready, you can publish your Querybook to your project for collaboration with teammates.

Space

A space in Amazon SageMaker Unified Studio refers to a personalized workspace that provides an isolated, sandboxed environment for users to run arbitrary code without interfering with other workers in a project. Each space consists of a compute instance, an EBS volume, and the JupyterLab application. Users can access their spaces through various entry points in Amazon SageMaker Unified Studio, the developer tools section, or by selecting Notebook files. The project Git repository is cloned into the space when you create the space. SageMaker Distribution is the image that is used to provide all the libraries, extensions, and packages in the IDE application.

Troubleshooting in Amazon SageMaker Unified Studio

Troubleshooting issues related to subscriptions in Amazon SageMaker Unified Studio

This topic contains troubleshooting instructions for issues that you might have when subscribing to assets in Amazon SageMaker Unified Studio.

Error message	Resolution
Unable to assume the IAM role *role-arn*.	This error occurs when Amazon SageMaker Unified Studio cannot assume the IAM role needed for creating permissions to grant access to assets. To fix the issue, go to the AWS Identity and Access Management console in the account where your data asset exists and ensure the IAM role in the error message has a trust relationship with Amazon DataZone service principal as shown in the following policy. `{ "Effect": "Allow", "Principal": { "Service": "datazone .amazonaws.com" }, "Action": [ "sts:AssumeRole", "sts:SetContext" ], "Condition": { "StringEquals": { "aws:SourceAccount ": accountID } } }`

Error message	Resolution
The IAM role *role-arn* does not have the necessary permissions to read the metadata of the asset you are trying to subscribe to.	This error occurs when Amazon SageMaker Unified Studio is able to assume the IAM role but the role does not have the necessary permissions. To fix this issue, go to the IAM console in the account where your data asset exists and make sure that the role you see in the error message has the SageMaker StudioProjectUserRolePolicy attached to it.
Asset is a resource link. Amazon SageMaker Unified Studio does not support subscriptions to resource links.	This error occurs because you cannot subscribe to AWS Glue table resource links in Amazon SageMaker Unified Studio. Choose a different asset type.

Error message	Resolution
Asset is not managed by Lake Formation.	This error occurs when the AWS Lake Formation permissions are not enforced on the asset you are trying to publish. This happens because the Amazon S3 location of the asset is not registered in Lake Formation. • To fix the issue, log into the Lake Formation console in the account where the table exists. • Register the Amazon S3 location in **Lake Formation mode** or **Hybrid mode**. • Ensure that the `IAMAllowedPrincipals` group is not added in the table's Lake Formation permissions. **Note** If you use encrypted Amazon S3 buckets or cross-account setups, you may need to adjust your AWS KMS and S3 settings. For more information, see Adding an Amazon S3 location to your data lake

Error message	Resolution
IAM role *role-name* does not have necessary Lake Formation permissions to grant access to this asset.	This error occurs when the IAM role in the error message does not have the necessary permissions for Amazon SageMaker Unified Studio to manage permissions on the published table. You can resolve the issue by granting the following AWS Lake Formation permissions to the the IAM role on the table you are looking to publish. 1. `Grant Describe` and `Describe Grantable` on the database where the tables exist. 2. `Describe`, `Select`, `Describe Grantable` , and `Select Grantable` on the all the tables in the above database that you want Amazon SageMaker Unified Studio to manage access on your behalf.

Amazon EBS Volume Depletion with Local Notebook Execution

Question

“Enhance the StartExecution API response when throttling occurs due to low disk space, instructing the user to delete files from the jobs folder."

Answer

1. Navigate to JupyterLab
2. In the Jobs folder, select the folder and files
3. Select delete

Domain

Question

From IAM SSO access portal URL, SAmazon SageMaker Unified Studio is not listed. When I click on Amazon DataZone, Amazon DataZone portal is shown. I clicked on SIGN IN WITH SSO, it failed due to Invalid redirectUri provided.

Answer

Visit DAmazon DataZone console, choose your domain, then click the Amazon SageMaker Unified Studio URL.

SAML Identity Provider Email Issue

Question

When using 3rd party SAML identity providers, the domain creation flow does not identify my email address.

Answer

This happens because during the user provisioning step, the email field was not populated in your local SSO instance. When sync-ing with 3rd party SAML identity providers, modify the default mapping to ensure it includes the "email" field and re-do the sync.

Project Creation Failure

Question

When I configure a project profile with resources pointing to another region/account, and try to create a new project using the project profile, it failed due to the error Project creation failed because one or more resources could not be provisioned.

Answer

Make sure that you complete following configurations:

1. Domain owner account: In the Domains menu, choose your domain. Under the Account associations tab, verify that domain is associated with the target account, and the status is Associated.
2. Target account: In the Associated domains menu, choose the associated domain. Choose your blueprint. Under the Regions tab, verify that the target region is added. Under the Authorization tab, verify that the target domain unit is shown.

3. Domain owner account: In the Domain details, under the Project profiles tab, choose your project profile. Under the Blueprint deployment settings tab, choose Name of your blueprint, under Deployment order, verify that Account ID and Region are configured correctly. Under the Authorized users and groups, verify that your SSO user is added.

Data Explorer Visibility Issue

Question

On the data explorer, I cannot see my existing databases and tables on Glue Data Catalog. How can I query them?

Answer

Amazon SageMaker Unified Studio configures AWS IAM permissions and permission boundaries. You can optionally remove the permission boundaries to allow access to the existing databases and tables.

Data Catalog Visibility Issue

Question

On the data catalog, I cannot see my existing databases and tables on Glue Data Catalog. How can I view them?

Answer

1. On your project page, choose Data sources.
2. Choose CREATE DATA SOURCE to add the existing databases and tables as a data source.
3. For Data source type, choose AWS Glue, and choose NEXT.
4. Configure how to select your databases and tables here.
5. Once everything is filled, choose NEXT and go ahead to register the data sources.

Connection to Amazon RDS MySQL in Existing VPC

Question

I want to connect to my Amazon RDS MySQL database instance that exists in my existing VPC. When I add a connection, I do not see any settings about VPC. How can I configure the reachability?

Answer

Amazon SageMaker Unified Studio uses the VPC and subnets that are specified in the domain creation. If you have the data source in a separate VPC, you can configure network reachability between the domain VPC and the data source VPC using VPC peering or Transit Gateway, or alternatively you can create a new domain using the data source VPC.

Visual ETL Flow Column Selection

Question

I created a data source, and now I am adding a new transform on top of it. But I cannot choose the columns for the transform.

Answer

When you start authoring a visual ETL flow, the data preview is also started. Once the preview is completed, then schema is automatically collected and available for further transforms.

Invalid or expired auth token when accessing an IDE

Question

When I tried to access an IDE in Amazon SageMaker Unified Studio I got an invalid or expired auth token error message. How do I resolve this?

Answer

This issue often occurs when third-party cookies are blocked. Amazon SageMaker Unified Studio uses the third-party cookies for authenticating into a Amazon SageMaker Unified Studio IDE since the IDE is hosted in a separate domain from the primary site. To help resolve this issue, ensure that third-party cookies are allowed in your browser.

> **Note**
>
> To do this in Safari, go to **Preferences** or **Settings** and ensure that on the **Privacy** page, the box next to **Prevent cross-site tracking** is unchecked.

For more information, see the following sites:

- For Firefox, see Third-party cookies and Firefox tracking protection.
- For Safari, see Prevent cross-site tracking in Safari on Mac.
- For Chrome, see Delete, allow and manage cookies in Chrome.
- For Edge, see Manage cookies in Microsoft Edge: View, allow, block, delete and use.

JupyterLab Configure Magic Error

Question

When I ran %%configure magic, it returned the error Connection name cannot be empty.

Answer

The magic syntax is different from Glue Interactive Session's existing kernel. Instead, run the magic with following syntax:

```
%%configure --name (compute) (-f)
{
"key": "value"
}
```

For example, if you want to change the default Spark SQL catalog name for project default Spark connection, run following magic:

```
%%configure --name project.spark --f
{
"--conf": "spark.sql.defaultCatalog=glue_catalog"
}
```

Local IDE troubleshooting

The following table lists common issues with the local IDE extension, along with their categories and resolutions.

Issue	Category	Resolution
Invalid domain URL format	Authentication	Verify that the URL follows the format `https://dzd-xxxxxxxxx.sagemaker.region.on.aws` . Check for copy-paste errors.
SSO authentication fails or hangs in browser	Authentication	Clear your browser cache and cookies, try an incognito or private window, or restart your IDE.
Connection expired or token issues	Authentication	If using SSO, sign out and sign back in through the AWS Toolkit extension. If using IAM credentials, update your credentials file.
"Failed to get SSO instance info" or connection timeout	Authentication	Verify that the domain URL is accessible from your network.
Timeouts connecting to the Space	Non-public internet access	Add the required URLs to your allowlist for your extension . For example, for VS Code, add the Microsoft extension marketplace URLs to your allowlist.
VPC-only domain without internet access	Non-public internet access	Configure a NAT gateway or proxy for internet access. Completely isolated domains require additional setup for remote connections.
"Failed to start SageMaker session"	Session and connection	Check the browser tab that opens for the specific error. Verify your permissions, or

Issue	Category	Resolution
		reduce the number of active sessions.
Remote SSH fails with "Invalid hostname format. Expected format: `sm_<creds-type>_<AWSResourceARN> "`	Session and connection	Remove previous SSH config entries from other IDEs by choosing Open SSH Configuration File in the command palette, then reinstall the extension.
Multiple browser windows open with error messages after a Space shuts down	Session and connection	In your user settings, set `remote.SSH.serverShutdownTimeout` to a lower value, such as 5.
AccessDeniedException on project node	Data Explorer	Contact your administrator to verify or grant project access.
"Failed to obtain project credentials"	Data Explorer	Re-authenticate through the extension, or update your credentials file.
"No AWS account region found in tooling environment"	Data Explorer	Verify that your project has a valid tooling environment configured.

Quotas for Amazon SageMaker Unified Studio

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. Unless otherwise noted, each quota is account-specific and Region-specific. If the default quotas don't fit your use case, you can request a quota increase. You can use the Service Quotas console from the AWS Management Console or contact AWS Support to submit requests on your behalf. For more information about how to request a quota increase using the Service Quota console, see Requesting a quota increase in the *Service Quotas User Guide*.

The resource quota limits are applied at the account level, meaning the depletion of resource quotas in one project can affect all other projects within the account. Administrators can monitor resource usage by project and take necessary actions through the Amazon SageMaker Unified Studio console. When approaching quota limits, administrators can delete unused resources to free up quota allocation or request a quota increase.

Amazon SageMaker Unified Studio has the following default quotas and limits. These default quotas provide a balance between resource availability and cost management.

Resource	Default
Maximum number of JupyterLab instances	4000
Maximum number of user type project members per project	100
Maximum number of group type project members per project	20
Maximum number of spaces	6000
Maximum number of projects	1000
Maximum number of environments	1000
Maximum number of Micro environments	200

For more information about other AWS service quotas, see AWS service quotas.

For more quotas information, see the following:

- Amazon SageMaker Supported Regions and Quotas
- Amazon Managed Workflows for Apache Airflow endpoints and quotas
- Amazon Redshift endpoints and quotas
- Amazon EMR endpoints and quotas
- Amazon DataZone endpoints and quotas
- Amazon Q Business endpoints and quotas
- Amazon Athena endpoints and quotas
- Amazon Bedrock endpoints and quotas
- AWS Glue endpoints and quotas

Release notes for Amazon SageMaker Unified Studio

The following sections describe the feature releases for Amazon SageMaker Unified Studio.

July 2026

July 21, 2026

Amazon SageMaker Unified Studio now supports Amazon OpenSearch as a data source

Amazon SageMaker Unified Studio now supports Amazon OpenSearch as a data source, enabling you to query and analyze your search and log analytics data directly alongside your other data assets. With this new connection, you can combine operational search data in OpenSearch with data from sources like Amazon Redshift, Amazon S3, and relational databases, all within a single, governed environment.

This integration is valuable when you need to correlate data across analytical and operational workloads. For example, you can join application logs and metrics stored in OpenSearch with transactional data to gain insights into system performance and user behavior. Data engineers and analysts can build data pipelines that bring together real-time search and analytics data with structured datasets, streamlining cross-source workflows without switching between tools.

To get started, add an Amazon OpenSearch connection under your project's data section. Your OpenSearch data appears in the data explorer alongside your other project data. From there, you can query it directly using the query editor, explore it in notebooks, or incorporate it into a visual ETL job.

This feature is available in all AWS Regions where Amazon SageMaker Unified Studio is available. For more information, see the section called “Connecting to a new data source” in the *Amazon SageMaker Unified Studio User Guide*.

July 17, 2026

Amazon SageMaker Unified Studio now supports custom visual transforms in Visual ETL

With Amazon SageMaker Unified Studio, you can now create, share, and reuse custom visual transforms in visual ETL flows. This capability enables data engineers to build reusable, business-specific transformation logic and make it available to their teams without requiring coding expertise.

With custom visual transforms, you can build your own transforms and publish them to a shared library that other team members can discover and use directly from the visual ETL interface. For example, you can create a transform that standardizes customer phone numbers, masks personally identifiable information, or applies your organization's standard data-quality checks. Team members can then reuse the same logic across multiple ETL jobs, reducing duplicated effort and helping ensure consistent results.

This feature is available in all AWS Regions where Amazon SageMaker Unified Studio is available. For more information, see the section called “Custom visual transform” in the *Amazon SageMaker Unified Studio User Guide*.

July 9, 2026

Workflows operators for Amazon Bedrock, Amazon S3 Tables, Amazon S3 Vectors, and AWS Glue Catalog

With Amazon SageMaker Unified Studio Workflows, you can now use 19 new operators. These operators support Amazon Bedrock, Amazon S3 Tables (a capability of Amazon S3), Amazon S3 Vectors (a capability of Amazon S3), AWS Glue Data Catalog, and Amazon Managed Workflows for Apache Airflow (Amazon MWAA) Serverless. Use the visual workflow creator to add tasks and orchestrate these services without writing custom integration code.

With this launch, you can create workflows that do the following:

- Manage Amazon Bedrock guardrails
- Provision and delete S3 Tables and S3 Vectors resources
- Manage AWS Glue Data Catalog tables and databases
- Trigger MWAA Serverless workflow runs

This launch expands the services that you can orchestrate from Amazon SageMaker Unified Studio Workflows. You no longer need to switch between consoles or write custom DAG code.

This feature is available in all AWS Regions where Amazon SageMaker Unified Studio is available. For more information, see Supported operators for Amazon MWAA Serverless workflows. To get started, see the section called "Visual workflows" in the *Amazon SageMaker Unified Studio User Guide*.

July 9, 2026

Create custom asset types in the catalog for IAM-based domains

With Amazon SageMaker Unified Studio, you can now create custom asset types for IAM-based domains. Domain administrators can catalog any format of asset, such as medical imaging files in Amazon S3, revenue dashboards built in PowerBI, or PDF research reports generated by a third-party platform. Custom asset types bring all assets into the SageMaker catalog so teams can search, discover, and subscribe to them without needing separate tools or processes.

To get started, create a custom asset type with a name, description, and optional metadata forms that define the fields each asset should carry. After an asset is published, anyone in the domain can find it by name, type, or glossary term and request a subscription through the same governed workflow used for all other catalog assets.

This feature is available in all AWS Regions where Amazon SageMaker Unified Studio is available. For more information, see the section called "Custom asset types" in the *Amazon SageMaker Unified Studio User Guide*.

July 9, 2026

Custom asset types for IAM-based domains

Amazon SageMaker Unified Studio now supports custom asset types for IAM-based domains. With custom asset types, domain administrators can catalog any format of asset within Amazon SageMaker Unified Studio, such as medical imaging files in Amazon S3, revenue dashboards built in PowerBI, or PDF research reports generated by a third-party platform. Custom asset types bring all assets, regardless of their underlying format, into the Amazon SageMaker Unified Studio catalog so teams can search, discover, and subscribe to them without needing separate tools or processes.

To get started, an administrator creates a custom asset type with a name, description, and optional metadata forms that define the fields each asset carries. Individual assets can then be created from

that type, enriched with glossary terms and README documentation to add business context, and published for discovery. Once published, anyone in the domain can find the asset by name, type, or glossary term and request a subscription through the same governed workflow used for all other catalog assets.

Custom asset types for IAM-based domains are available in all AWS Regions where Amazon SageMaker Unified Studio is available. For more information, see the section called "Custom asset types" in the *Amazon SageMaker Unified Studio User Guide*.

July 7, 2026

Importing existing MWAA environments

With Amazon SageMaker Unified Studio, you can now connect existing Amazon Managed Workflows for Apache Airflow (MWAA) environments to projects. If you already operate MWAA environments, you can now manage your Airflow workflows from the same interface you use for analytics and machine learning, without recreating configurations or migrating DAGs.

To connect an existing environment, open the Workflows tool in your project and choose **Add connection** in the connection selector. Provide the Airflow configuration options that reference your domain and project. After you connect, project members can sync, trigger, and monitor workflows directly from Amazon SageMaker Unified Studio. Environments running Apache Airflow 3 or later also get access to the visual authoring experience for creating new workflows using the drag-and-drop editor.

This feature is available in all AWS Regions where Amazon SageMaker Unified Studio is available. For more information, see the section called "Workflow environments" in the *Amazon SageMaker Unified Studio User Guide*.

July 7, 2026

Use data lineage in IAM-based domains

With Amazon SageMaker Unified Studio, you can now capture and visualize OpenLineage-compatible data lineage in IAM-based domains. Lineage events are captured from Apache Spark jobs run on Amazon EMR, AWS Glue, SageMaker Visual ETL, and notebooks. This capability was previously available only in IAM Identity Center-based domains.

The interactive lineage graph provides an aggregate visual representation of how data moves from source to consumption, with configurable graph depth, event timestamp mode for

detailed column-level lineage, and a dataset-only view for simplified visualization. You can also programmatically publish, query, and manage data lineage from OpenLineage-compatible applications, and remove published events using the DeleteLineageEvent API.

This feature is available in all AWS Regions where Amazon SageMaker Unified Studio is available. To learn more, see the *Amazon SageMaker Unified Studio User Guide*.

July 2, 2026

Amazon SageMaker Unified Studio now supports Terraform for provisioning

Amazon SageMaker Unified Studio now supports Terraform for provisioning. Customers can use the open-source terraform-aws-sagemaker-unified-studio module to deploy a Amazon SageMaker Unified Studio domain through version-controlled templates. Platform teams can bring Amazon SageMaker Unified Studio into their existing infrastructure-as-code pipelines, maintaining consistency across development, staging, and production accounts. The Terraform module handles the infrastructure of Amazon SageMaker Unified Studio domain with provisioned IAM roles. Sub-modules let teams enable blueprints, compose blueprints into project profiles, and create projects independently.

To get started, see the terraform-aws-sagemaker-unified-studio module on GitHub.

June 2026

June 17, 2026

AWS Glue support for Spark Connect in notebooks

Amazon SageMaker Unified Studio notebooks now support AWS Glue as a Spark Connect runtime. You can select a AWS Glue connection from the notebook compute panel to run PySpark and Spark SQL code using serverless AWS Glue managed infrastructure. AWS Glue joins Amazon Athena for Apache Spark and Amazon EMR Serverless as available Spark runtimes in notebooks.

For more information, see the section called “Use AWS Glue with Spark Connect”.

June 9, 2026

Amazon SageMaker Unified Studio Notebooks now support EMR Serverless

Amazon SageMaker Unified Studio notebooks now support Amazon EMR Serverless with Apache Spark Connect, giving data engineers and analysts more flexibility in choosing their Spark runtime

for interactive analytics and data engineering workloads. In addition to Amazon Athena for Apache Spark, users can now leverage Amazon EMR Serverless as their Spark runtime, selecting the optimal engine based on their requirements. You can run PySpark and Spark SQL on an EMR Serverless Spark Application in Notebook cells. Users can select their Spark runtime from the Notebook side panel. Additionally, users can leverage SageMaker Data Agent to generate code and execution plans from natural language prompts.

For more information, see the section called “Use EMR Serverless with Spark Connect”.

June 4, 2026

SageMaker Data Agent integrates business context into conversations

Amazon SageMaker Data Agent now integrates with SageMaker Catalog business context and metadata, enabling data practitioners to discover datasets and generate more accurate SQL and Python code using business terminology instead of cryptic technical table names. This integration allows the Data Agent to leverage the business context that companies have invested months curating in their SageMaker Catalog, including those synced from Collibra, Atlan, and Alation, to deliver more accurate data discovery and code generation.

With this capability, you can ask questions like "Calculate customer retention rate" or "What data do I have on customer churn?" and the Data Agent searches glossary terms, custom metadata forms, asset summaries, and README content to identify the correct tables and columns. This integration is available in SageMaker Unified Studio notebooks and Query Editor.

For more information, see the section called “Business context integration”.

June 4, 2026

Import SageMaker Pipelines into Serverless Workflows (IAM domains)

You can now import existing SageMaker pipelines directly into Serverless Apache Airflow workflows in Amazon SageMaker Unified Studio. The import converts your pipeline definition into an interactive visual workflow DAG that you can edit using the drag-and-drop workflows editor. This feature is available for IAM domains.

For more information, see the section called “Import SageMaker Pipelines”.

June 3, 2026

Amazon SageMaker Unified Studio now supports a localized experience in twelve languages

Amazon SageMaker Unified Studio enhanced its global accessibility by introducing support for twelve languages across the user interface. Supported languages include English (American), Chinese (Simplified and Traditional), French, German, Indonesian, Italian, Japanese, Korean, Portuguese (Brazilian), Spanish, and Turkish. With this launch, data engineers, analysts, and data scientists across global teams can navigate, build, and collaborate in the language they are most comfortable with, reducing friction and improving productivity.

Your preferred language is automatically detected based on your browser's default language settings. You can also set your preferred language by choosing 'Language selector' in your profile settings and selecting the language. The selected language applies across the entire SageMaker Unified Studio user interface.

For more information, see *Navigate Amazon SageMaker Unified Studio*.

June 3, 2026

Amazon SageMaker Unified Studio now supports notebook scheduling

Amazon SageMaker Unified Studio now enables you to schedule, parameterize, and orchestrate notebook runs directly from the notebook interface without managing external orchestration infrastructure. You can trigger on-demand background runs on dedicated compute without interrupting interactive sessions and create scheduled or recurring runs. With notebook parameterization, you can reuse a single notebook across different inputs by defining parameters and overriding their values per schedule or on-demand run. You can also orchestrate multi-notebook workflows using the Notebook Operator in the Workflows tool, chaining notebooks so that outputs from one run feed as inputs to the next. When a scheduled or background run fails, AI-assisted troubleshooting using SageMaker Data Agent helps you identify the root cause and suggests fixes directly in the notebook.

For more information, see the section called “Schedule and automate notebook runs”.

June 3, 2026

SageMaker Data Agent now supports conversation history

The SageMaker Data Agent in Amazon SageMaker Unified Studio now retains your past conversations for up to 90 days, allowing you to resume previous analysis sessions. You can access conversation history from the clock icon in the chat panel header, browse past conversations sorted by most recent activity, and resume any conversation with its full message history.

For more information, see the section called “Conversation history”.

June 1, 2026

Amazon SageMaker adds permissions boundaries for SCP compliance

Amazon SageMaker Unified Studio now supports custom IAM permissions boundaries, so organizations that enforce Service Control Policies (SCPs) requiring permissions boundaries on all IAM roles can adopt SageMaker Unified Studio without modifying their security posture.

When a user creates a project, SageMaker Unified Studio provisions three IAM roles: a project user role, an Amazon Bedrock service role, and a Bedrock Lambda execution role. With this launch, administrators can specify a permissions boundary in the Tooling blueprint configuration, and all three roles are created with that permissions boundary attached. This satisfies SCP requirements at creation time, and project provisioning succeeds without administrator intervention. The permissions boundary also limits what the provisioned roles can do, so administrators retain control over project-level permissions even as new projects are created. Because the permissions boundary is set at the blueprint level, it applies to every new project automatically. For more information, see What's New Post.

May 2026

May 22, 2026

Amazon SageMaker expands domain management across domain types

Amazon SageMaker Unified Studio now provides domain management experience for Identity Center and IAM-based domains outside of AWS console, allows administrators and data management teams to create and manage projects, configure workforce identity, manage users and permissions, and set networking properties for projects. Previously, this was only available for IAM based domains.

With this launch, administrators of Identity Center-based domains can access domain management capabilities in SageMaker Unified Studio portal to create projects with configurable execution roles that define which AWS analytics, AI, and ML services the project can access. VPC configuration is consistent across both domain types, inherited by all projects, and can be edited to change the VPC, subnets, or security group. Administrators can also manage associated accounts, enabling users to publish and consume data from other AWS accounts within SageMaker Unified Studio. For more information, see What's New Post.

May 21, 2026

Automated Glue connector provisioning for cross-subnet job retries

Amazon SageMaker Unified Studio now supports automatic creation of connections for AWS Glue job retries across subnets to improve data pipeline resilience. Administrators can define their domain VPC with multiple private subnets across Availability Zones, and the system provisions the connectors needed for all new projects so that failed jobs can be retried on an alternate subnet automatically. No user action is needed beyond the initial VPC configuration on the domain.

For more information, see What's New Post.

May 21, 2026

Manage feature groups in Feature Store from Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio now provides a Feature Store interface for creating, browsing, editing, and deleting feature groups directly from the project UI. You can configure online stores (including the in-memory tier with collection types), offline stores with Glue Data Catalog or Iceberg table formats, and use sample notebooks to ingest and retrieve feature data. The Feature Store interface is available in Amazon SageMaker Unified Studio domains configured with AWS IAM.

For more information, see the section called “Feature Store”.

May 22, 2026

Amazon SageMaker adds business metadata and governance in IAM-based domains

Amazon SageMaker Unified Studio now supports business context, metadata and data governance capabilities in IAM-based domains. With this launch, customers using Amazon SageMaker IAM-based domains can add business context to their AWS Glue Data Catalog tables, including business names, descriptions, and README documentation. They can use AI-generated metadata to produce business names and descriptions automatically, reducing the effort of cataloging large numbers of tables. Customers can also create business glossaries so that teams across the organization use consistent definitions for terms like "ARR" or "churn rate," and define metadata form templates to capture structured attributes such as data classification, retention policies, or ownership details.

With this business context in place, data engineers, analysts, and data scientists can search for and discover tables across the entire domain, filter results by glossary terms and metadata form fields,

and request access through subscriptions. After an administrator approves the request, Amazon SageMaker Unified Studio automatically grants the necessary AWS Lake Formation permissions to the project. Administrators can also grant access to tables directly from within Amazon SageMaker Unified Studio without waiting for a request.

For more information, see What's New Post.

May 20, 2026

Amazon SageMaker Unified Studio now supports data quality rule authoring and evaluation

Amazon SageMaker Unified Studio now supports data quality rule authoring and evaluation, powered by AWS Glue Data Quality. Data engineers, analysts, and data scientists can define data quality rules, run ruleset evaluations, and view results directly within Amazon SageMaker Unified Studio for both data at rest in catalog tables and data in transit within Visual ETL jobs. This helps you catch data quality issues before bad data enters your data lakes or affects downstream analytics and machine learning workloads.

With this launch, you can author rules using the same Data Quality Definition Language (DQDL) used in AWS Glue Data Quality and run evaluations directly in Amazon SageMaker Unified Studio across two workflows. For data at rest, a dedicated Data Quality tab on catalog assets provides rule authoring, on-demand or scheduled evaluations, and detailed per-rule pass/fail results. For data in transit, you can add an Evaluate Data Quality transform to any Visual ETL job, and review data quality results as part of the run details. You can create rulesets that check for completeness, uniqueness, freshness, accuracy, and other data quality dimensions.

For more information, see What's New Post.

May 13, 2026

Amazon SageMaker Data Agent now available for IAM Identity Center domains

SageMaker Data Agent is now available in Amazon SageMaker Unified Studio domains configured with AWS IAM Identity Center. Data Agent helps data analysts and engineers streamline their analytics workflows across both notebooks and Query Editor environments. You can describe your analysis goals in plain English and receive working Python or SQL code tailored to your connected data sources. The agent maintains conversational context across notebook cells, selected tables, and query history, proposing step-by-step plans before generating code.

For more information, see What's New Post.

May 11, 2026

Amazon SageMaker Unified Studio adds getting started tutorials and in-product release notes

Amazon SageMaker Unified Studio now helps you get productive faster with getting started tutorials and a development environment appearance that automatically adapts to your system preference, and adds in-product release notes to help you discover new capabilities.

On the homepage, a new getting started section helps you get productive in minutes by walking through core workflows such as running your first SQL query, analyzing data from a notebook, building a data pipeline with Visual ETL, and training an ML model. Each tutorial uses pre-loaded sample data and can be completed in under 10 minutes. The development environment now also defaults to match your operating system's light or dark mode setting, so the interface matches your preference from your first sign-in. A new "What's New" section surfaces recent feature announcements and release notes directly in the product, so you can stay informed about new capabilities as they launch. In 2026 alone, Amazon SageMaker Unified Studio has added over 20 new features, which you can also find in the release notes.

For more information, see What's New Post.

May 7, 2026

Amazon SageMaker Unified Studio adds identity and user management features

Amazon SageMaker Unified Studio announces new administration features that give administrators more control over identity configuration and user management for both IAM and Identity Center domain types.

In SageMaker IAM domains, administrators can now onboard users through single sign-on by configuring AWS IAM Identity Center. After configuration, administrators can add IAM roles, IAM users, IAM Identity Center users, and IAM Identity Center groups as project members. Teams can collaborate on project data and resources regardless of how individual members authenticate. Administrators can set up IAM Identity Center integration in the SageMaker Unified Studio admin portal. A new domain user management page for SageMaker IAM domains gives administrators a consolidated view of all users active in the domain, where they can manage access and update permissions from a single screen.

In SageMaker Identity Center domains, users can now access the SageMaker Unified Studio portal by federating through an IAM role. SageMaker Unified Studio creates a unique user session for each

federated user, so users sharing the same role don't overwrite each other's work. Administrators can audit individual actions even when multiple users share a single IAM role.

With these features, customers can use IAM identity or IAM Identity Center corporate identity across both domain types, giving teams flexibility to collaborate in SageMaker Unified Studio regardless of their authentication method. For more information, see What's New Post.

April 2026

April 23, 2026

Amazon SageMaker supports notebooks and data agent for IdC domains

Amazon SageMaker Unified Studio now supports serverless notebooks with a built-in data agent for AWS IAM Identity Center (IdC) domains. Previously, the notebook experience and data agent were available only in IAM domains. With this launch, customers who use IdC for authentication and access management can access the high-performance, serverless notebook environment for analytics and machine learning (ML) workloads.

The serverless notebook gives data engineers, analysts, and data scientists one place to perform SQL queries, execute Python code, process large-scale data jobs, run ML workloads, and create visualizations. A built-in AI data agent accelerates development by generating code and SQL statements from natural language prompts and guides users through their tasks. Customers can flexibly combine SQL, Python, and natural language within a single interactive workspace, removing the need to switch between different tools based on the workload. For example, you can start with SQL queries to explore your data, use Python for advanced analytics or to build ML models, or use natural language prompts to generate code automatically. The notebook is backed by Amazon Athena for Apache Spark, scaling from interactive SQL queries to petabyte-scale data processing.

For more information, see What's New Post.

Amazon SageMaker Unified Studio now supports VPC for notebook kernels

Amazon SageMaker Unified Studio now supports Amazon Virtual Private Cloud (Amazon VPC) for notebook kernels. With this launch, notebook kernels execute within the VPC configured at the domain level, giving enterprises network isolation for interactive data and machine learning (ML) workloads. This helps customers meet security and compliance requirements by keeping applicable notebook compute traffic within their VPC boundaries.

With VPC support for notebook kernels, data engineers, analysts, and data scientists can connect to private resources from their notebooks. The notebook kernel inherits the VPC settings, subnets, and security groups defined at the SageMaker Unified Studio domain level, so administrators can manage network policies centrally. This means you can query private databases, access internal APIs, and work with data sources that are not publicly accessible, all from the same notebook environment that supports SQL, Python, and natural language through the built-in data agent. This VPC configuration only applies to the notebook's interactive compute, where your Python code and dataframes execute. For VPC configurations with other compute engines, refer to the documentation for each individual engine.

For more information, see What's New Post.

Amazon SageMaker Unified Studio now offers CI/CD CLI for data and AI applications

Amazon SageMaker Unified Studio now offers the CI/CD CLI (aws-smus-cicd-cli), an open-source command line tool that automates deployment of multi-service data and AI applications across development, test, and production. Organizations building applications in SageMaker Unified Studio combine multiple AWS services, including AWS Glue, Amazon Athena, Amazon MWAA, Amazon SageMaker AI, Amazon Bedrock, and Amazon QuickSight, into single applications. The CLI allows data teams to define applications once in a YAML manifest while DevOps teams deploy with a single command, reducing deployment bottlenecks and configuration drift.

The CLI reads a declarative manifest.yaml that maps each pipeline stage to an isolated SageMaker Unified Studio project. At deploy time, it substitutes stage-specific configurations (S3 paths, IAM roles, account IDs, and connection strings) and provisions resources in dependency order. Four commands cover the lifecycle: describe validates permissions and connections, bundle packages an immutable artifact from the source target, deploy writes that artifact to the destination target, and test runs post-deployment validation. It works with existing CI/CD solutions such as GitHub Actions, Jenkins, and GitLab CI. For more information, see What's New Post.

April 22, 2026

Amazon SageMaker Unified Studio now supports multiple code spaces within projects for IAM domains

Amazon SageMaker Unified Studio now lets data workers create and manage multiple code spaces (individually configured development environments) within a single project for IAM domains. Previously, projects were limited to one JupyterLab space and one Code Editor space embedded in the project. With this launch, you can now parallelly work on different workstreams or experiments

with different compute and storage configuration needs, giving developers the flexibility they need as their workloads scale. For instance, data scientists can now work in parallel on any long running data transformation and model training workloads within the same project using separate spaces.

With multiple spaces, each one maintains its own persistent Amazon EBS volume, ensuring that your files, data, and session state are preserved independently. You can scale compute and storage up or down per space, pause and resume them at any time, and customize the runtime environment for each specific task. Spaces can either be opened in dedicated browser tabs or connected to a local IDE if you prefer your own development environment, with full functionality including Amazon Q paid tier support. This is particularly beneficial for builders who need isolated environments for parallel workstreams while still working within a single collaborative project. For more information, see What's New Post.

April 21, 2026

Amazon SageMaker now supports multi-region replication from IAM Identity Center

Amazon SageMaker now supports multi-region replication from IAM Identity Center (IdC), enabling you to deploy SageMaker Unified Studio domains in different regions from your IdC instance. This new capability empowers enterprise customers, particularly those in regulated industries like financial services and healthcare, to maintain compliance while leveraging centralized workforce identity management.

As an Amazon SageMaker Unified Studio administrator, you can deploy SageMaker domains closer to your workforce based on data residency needs while maintaining seamless single sign-on (SSO) access. Organizations can address use cases such as maintaining IdC in one region while processing sensitive data in compliance-required regions, supporting global operations with centralized identity management, and meeting data sovereignty requirements without compromising SSO capabilities. For more information, see What's New Post.

April 7, 2026

Amazon SageMaker adds serverless workflows to Identity Center domains

Amazon SageMaker Unified Studio now supports Serverless Workflows in Identity Center domains. With this launch, customers using Identity Center domains can orchestrate data processing tasks with Apache Airflow (powered by Managed Workflows for Apache Airflow) without provisioning or managing Airflow infrastructure. Serverless Workflows were previously available only in IAM-based domains.

Serverless Workflows automatically provision compute resources when a workflow runs and release them when it completes, so you only pay for actual workflow run time. Each workflow runs with its own execution role and isolated worker, providing workflow-level security and preventing cross-workflow interference. With Serverless Workflows, Identity Center domain customers also get access to the Visual Workflow experience with support for around 200 operators, including built-in integration with AWS services such as Amazon S3, Amazon Redshift, Amazon EMR, AWS Glue, and Amazon SageMaker AI.

For more information, see What's New Post.

April 6, 2026

Amazon SageMaker Unified Studio adds notebook import/export and developer acceleration features

Amazon SageMaker Unified Studio notebooks now support import/export capabilities, enabling migration from JupyterLab and other notebook platforms. This release also introduces developer acceleration features including cell reordering, keyboard shortcuts, cell renaming, and multi-line SQL support, designed to enhance productivity for data engineers and data scientists professionals working with notebook-based workflows.

The new import/export functionality supports .ipynb, .json, and .py formats while preserving cell types, metadata, and outputs, making platform migration straightforward. You can export notebooks in four formats including Jupyter notebook with requirements (.zip), standard .ipynb, Python scripts (.py), and SageMaker Unified Studio native format (.json). Developer acceleration features enable you to reorder cells without copy-paste duplication, assign custom names to cells for improved navigation in large notebooks, use familiar keyboard shortcuts for faster development, and execute multiple SQL statements in a single cell with results displayed in separate tabs for easy comparison and analysis. For more information, see What's New Post.

April 3, 2026

Amazon SageMaker Data Agent introduces charting capabilities and support for materialized views

Amazon SageMaker Data Agent now supports interactive charting, SQL analytics on Snowflake data sources, and materialized view management in Amazon SageMaker Unified Studio notebooks. Data Agent now provides a complete analytics workflow that goes beyond code generation,

enabling you to explore AWS and external data sources, visualize results, and optimize query performance, all with natural language prompts.

You can ask "plot monthly revenue trends by region for 2025" and Data Agent generates an interactive chart directly in your notebook, where you can hover over data points, and modify without writing code. When your analysis spans AWS and Snowflake, you can query Snowflake tables through external connections and join them with your AWS Glue Data Catalog data in a single prompt. Additionally, you can ask "analyze my notebook and suggest which queries would benefit from materialized views" and the agent recommends optimizations based on your query patterns, creates the views, and sets refresh schedules. For more information, see What's New Post.

April 1, 2026

Amazon SageMaker Data Agent now supports geo-specific inference for Japan and Australia

Amazon SageMaker Data Agent now supports cross-region inference profiles for Japan and Australia through Amazon Bedrock. With this update, inference requests from Data Agent in the Asia Pacific (Tokyo) and Asia Pacific (Sydney) regions are processed within their respective geographies, supporting data sovereignty requirements for customers in Japan and Australia.

Data Agent provides an AI-powered conversational experience for data exploration, Python and SQL code generation, troubleshooting, and analytics directly within Amazon SageMaker Unified Studio Notebook and Query Editor. With geo-specific inference through JP-CRIS (Japan Cross-Region Inference) and AU-CRIS (Australia Cross-Region Inference), you can use Data Agent with confidence that your inference requests are routed exclusively within your geography over the AWS Global Network. Customers in regulated industries such as financial services, healthcare, and the public sector can meet data residency requirements while using the full set of Data Agent capabilities. For more information, see What's New Post.

March 2026

March 31, 2026

Amazon SageMaker Unified Studio adds Observability for AWS Glue jobs via CloudWatch metrics

Amazon SageMaker Unified Studio adds Observability for jobs, it now displays Amazon CloudWatch metrics for AWS Glue jobs directly alongside job logs in a single, unified interface. This

enhancement adds observability to SageMaker Unified Studio, enabling data engineers and ETL developers to streamline their troubleshooting processes.

With this feature, teams can diagnose performance issues faster by correlating resource utilization patterns—including DPU utilization, memory consumption, CPU load, and data movement size—directly with job log output. Specific use cases include identifying compute bottlenecks, detecting memory pressure or out-of-memory conditions, optimizing resource allocation, and monitoring data pipeline performance at scale. By consolidating metrics and logs into one workspace, organizations can significantly reduce mean time to resolution (MTTR) for ETL pipeline issues and improve overall operational efficiency. For more information, see What's New Post.

March 30, 2026

Amazon SageMaker Data Agent is now available in the Amazon SageMaker Unified Studio Query Editor

Amazon SageMaker Data Agent is now available in the Query Editor in Amazon SageMaker Unified Studio, extending beyond notebook experience. With Data Agent in Query Editor, you can generate SQL queries from natural language, debug failed queries, and explore your data through a conversational, interactive experience.

Data Agent brings the same conversational experience available in notebooks to your SQL analytics workflow. You can ask "calculate quarterly revenue growth rate by product category for 2025," and the agent proposes a step-by-step plan for you to review before generating contextually accurate SQL for Amazon Redshift and Amazon Athena. This helps you build analytics queries faster, going from question to executable SQL without writing complex joins and aggregations manually. When a query fails, you can use Fix with AI to analyze the error and get suggested corrections. Data Agent maintains awareness of your connected data sources and schema information, so follow-up questions and modifications build on your previous context. For more information, see What's New Post.

March 25, 2026

Amazon SageMaker Unified Studio launches support for remote connection from Cursor IDE

Today, AWS announces remote connection from Cursor IDE to Amazon SageMaker Unified Studio via the AWS Toolkit extension. This new capability allows data scientists, ML engineers, and developers to leverage their Cursor setup - including its AI-powered code completion, natural language editing, and multi-file editing capabilities - while accessing the scalable compute

resources of Amazon SageMaker. By connecting Cursor to SageMaker Unified Studio using the AWS Toolkit extension, you can eliminate context switching between your local IDE and cloud infrastructure, maintaining your existing AI-assisted development workflows within a single environment for all your AWS analytics and AI/ML services. For more information, see What's New Post.

March 18, 2026

Amazon SageMaker Unified Studio adds custom metadata filters

Amazon SageMaker Unified Studio adds custom metadata search filters, enabling customers to narrow catalog search results using organization-specific attributes. This helps customers find the right assets faster by filtering on fields like business region, data classification, or study name, in addition to existing keyword and semantic search.

With custom metadata search filters, customers can add filters based on any custom metadata fields available in their catalog, such as sample type or study ID. Filters support string fields with a "contains" operator and numeric fields (Integer, Long) with equals, greater than, and less than operators. Customers can also filter by asset name, description, and date range. Multiple filters can be combined, and filter selections persist across browser sessions.

For more information, see What's New Post.

March 17, 2026

Amazon SageMaker Unified Studio supports aggregated view of data lineage

Amazon SageMaker Unified Studio now provides an aggregated view of data lineage, displaying all jobs contributing to your dataset. The aggregated view gives you a complete picture of data transformations and dependencies across your entire lineage graph, helping you quickly identify all upstream sources and downstream consumers of your datasets.

Previously, SageMaker Unified Studio showed the lineage graph as it existed at a specific point in time, which is useful for troubleshooting and investigating specific data processing events. The aggregated view now provides a complete picture of data transformations and dependencies across multiple levels of the lineage graph. You can use this view to understand the full scope of jobs impacting your datasets and to identify all upstream sources and downstream consumers.

The aggregated view is available as the default lineage view in Amazon SageMaker Unified Studio for IdC-based domains. You can switch to the previous view by toggling the "display in event

timestamp order" option. You can also query the lineage graph using the new QueryGraph API, which provides lineage node graphs with metadata and augmented business context.

For more information, see What's New Post.

March 9, 2026

Amazon SageMaker Unified Studio now supports faster data preview in Visual ETL

Amazon SageMaker Unified Studio introduces data preview v2.0 for Visual ETL, a new data preview mode that delivers near-instant results when building and iterating on visual ETL jobs. With data preview v2.0, data engineers and analysts can see the output of each transform in about one second, with no session startup required and at no additional compute cost.

Data preview v2.0 uses an in-browser query engine to load and process data locally, removing the dependency on server-side Spark sessions for preview operations. Source data is fetched once and cached in the browser, so subsequent transforms apply instantly without re-querying the underlying data source. For Amazon Redshift users, this means you can iterate on transforms without additional queries against your Redshift cluster, keeping your preview workflow fast and your cluster resources focused on production workloads. Data preview v2.0 supports CSV, Parquet, and JSON files from Amazon S3, in addition to data from Amazon Redshift, Amazon S3 Tables, AWS Glue Data Catalog, and third-party sources including Snowflake, MySQL, PostgreSQL, SQL Server, Oracle, Google BigQuery, Amazon DynamoDB, and Amazon DocumentDB. A toggle in the Visual ETL editor gives you the option to switch between data preview v2.0 and the original Spark-based preview at any time. For more information, see What's New Post.

March 5, 2026

Amazon SageMaker Unified Studio adds light mode support for IAM-based domains

Today, AWS announces light mode support in Amazon SageMaker Unified Studio for IAM-based domains. Customers can now configure the visual interface mode to match their preference, choosing between dark and light themes.

Light mode helps improve readability in bright environments and provides a familiar visual experience for customers who prefer lighter interfaces. Combined with the existing dark mode, this update gives you full control over your development environment's appearance, improving accessibility and reducing eye strain across varying lighting conditions. In SageMaker Unified Studio settings, you can click on 'customize appearance' under your Profile settings to choose between

visual modes including dark and light. The setting persists across browsers and devices. For more information, see What's New Post.

March 3, 2026

Amazon SageMaker Unified Studio adds metadata sync with third-party catalogs

Amazon SageMaker Unified Studio now supports metadata and context sync across Atlan, Collibra, and Alation. These integrations synchronize catalog metadata between Amazon SageMaker Catalog and each partner platform, giving teams a consistent view of their data and AI assets regardless of which tool they use day to day. Organizations can maintain aligned glossary terms, asset descriptions, and ownership information across platforms without manual reconciliation.

All three integrations synchronize key metadata elements including projects, assets, descriptions, glossary terms, and their hierarchies. With the Collibra integration, you can synchronize metadata in both directions between SageMaker Catalog and the partner platform, so updates you make in one are reflected in the other. Also, you can manage SageMaker Unified Studio data access requests from Collibra. With the Atlan and Alation integration, you can ingest metadata from SageMaker Catalog into Alation with additional enhancements coming soon. You set up these integrations by setting up a connection to SageMaker Unified Studio from within Atlan and Alation, while the Collibra integration is available as an open-source solution on GitHub. For more information, see What's New Post.

Amazon SageMaker Unified Studio now supports AWS Glue 5.1 for data processing jobs

Amazon SageMaker Unified Studio now supports AWS Glue 5.1 for Visual ETL, notebook, and code-based data processing jobs. With AWS Glue 5.1 in Amazon SageMaker Unified Studio, data engineers and data scientists can run jobs on Apache Spark 3.5.6 with Python 3.11 and Scala 2.12.18, and use updated open table format libraries including Apache Iceberg 1.10.0, Apache Hudi 1.0.2, and Delta Lake 3.3.2.

You can use AWS Glue 5.1 in Amazon SageMaker Unified Studio when creating data processing jobs by selecting Glue 5.1 from the version dropdown in job settings. This applies to Visual ETL jobs, notebook jobs, and code-based jobs, so you can take advantage of the latest Spark runtime and open table format libraries across all your data processing workflows.

For more information, see What's New Post.

Amazon SageMaker Unified Studio launches support for remote connection from Kiro IDE

Today, AWS announces the ability to remotely connect from Kiro IDE to Amazon SageMaker Unified Studio. This new capability allows data scientists, ML engineers, and developers to leverage their Kiro setup - including its spec-driven development, conversational coding, and automated feature generation capabilities - while accessing the scalable compute resources of Amazon SageMaker. By connecting Kiro to SageMaker Unified Studio using the AWS toolkit extension, you can eliminate context switching between your local IDE and cloud infrastructure, maintaining your existing agentic development workflows within a single environment for all your AWS analytics and AI/ML services. For more information, see What's New Post.

February 2026

February 4, 2026

Apache Spark lineage now available in Amazon SageMaker Unified Studio for IDC based domains

Amazon SageMaker announces general availability of Data Lineage for Apache Spark jobs executed on Amazon EMR and AWS Glue in SageMaker Unified Studio for IDC based domains. Data Lineage provides you with the information you need to identify the root cause of complex issues and understand the impact of changes.

This feature supports lineage capture of schema and transformations of data assets and columns from Spark executions in EMR-EC2, EMR-Serverless, EMR-EKS, and AWS Glue. You can then explore this lineage visually as a graph in SageMaker Unified Studio or query it using APIs. You can also use lineage to compare transformations across Spark job's history.

For more information, see What's New Post.

January 2026

January 30, 2026

Amazon SageMaker Unified Studio now supports AWS PrivateLink

Today, Amazon SageMaker announced a new capability allowing you to establish connectivity between your Amazon Virtual Private Cloud (VPC) and Amazon SageMaker Unified Studio without customer data traffic going through the public internet. Customers needing to go beyond the

standard data transfer protocol (HTTPS/TLS2) can choose to configure their VPC so data transfer stays within the AWS network.

Through AWS PrivateLink , Network Administrators can now onboard AWS service endpoints to their VPC used by Amazon SageMaker Unified Studio. With the endpoints are onboarded, IAM policies used by Amazon SageMaker will enforce that customer data stay within the AWS network. For more information, see What's New Post.

January 20, 2026

SageMaker Unified Studio adds cross-Region subscriptions support for IDC based domains

Amazon SageMaker Unified Studio now supports cross-Region subscriptions for comprehensive and flexible data access and governance. With cross-Region support, you can subscribe to AWS Glue tables and views, as well as Amazon Redshift tables and views published in a different AWS Region than your project. This capability helps break down data silos and enable better collaboration across your organization by allowing teams to access curated data assets from any AWS Region without manual replication. For more information, see What's New Post.

December 2025

December 2, 2025

Amazon SageMaker Catalog now exports asset metadata as queryable dataset

Amazon SageMaker Catalog now exports asset metadata as an Apache Iceberg table through Amazon S3 Tables. This allows data teams to query catalog inventory and answer questions such as, "How many assets were registered last month?", "Which assets are classified as confidential?", or "Which assets lack business descriptions?" using standard SQL without building custom ETL infrastructure for reporting.

This capability automatically converts catalog asset metadata into a queryable table accessible from Amazon Athena, SageMaker Unified Studio notebooks, AI agents, and other analytics and BI tools. The exported table includes technical metadata (such as resource_id, resource_type), business metadata (such as asset_name, business_description), ownership details, and timestamps. Data is partitioned by snapshot_date for time travel queries and automatically appears in SageMaker Unified Studio under the aws-sagemaker-catalog bucket. For more information, see What's New Post.

November 2025

November 21, 2025

Introducing one-click onboarding of existing datasets to Amazon SageMaker

Amazon SageMaker Unified Studio now offers one-click onboarding that helps customers start working with their existing AWS data in minutes. Customers can start directly from Amazon SageMaker, Amazon Athena, Amazon Redshift, or Amazon S3 Tables, giving them a fast path from their existing tools and data to the simple experience in SageMaker Unified Studio. After clicking "Get Started" and specifying an AWS IAM role, SageMaker automatically creates a project with all existing data permissions intact from AWS Glue Data Catalog, AWS Lake Formation, and Amazon S3. A notebook and serverless compute are pre-configured to accelerate first use. For more information, see What's New Post.

Announcing notebooks with a built-in AI agent in Amazon SageMaker

The new SageMaker notebooks provide data and AI teams a high-performance, serverless programming environment for analytics and machine learning jobs. Customers can quickly get started working with data without pre-provisioning data processing infrastructure. The notebook gives data engineers, analysts, and data scientists one place to perform SQL queries, execute Python code, process large-scale data jobs, run machine learning workloads and create visualizations, without having to switch between tools. It is powered by Amazon Athena for Apache Spark, automatically scaling from interactive queries to petabyte-scale processing. A built-in AI agent accelerates development by generating code and SQL statements from natural language prompts while guiding users through their tasks. For more information, see What's New Post.

Introducing Amazon SageMaker Data Agent for analytics and AI/ML development

SageMaker Data Agent works within new SageMaker notebooks to break down complex analytics and ML tasks into manageable steps. Customers can describe objectives in natural language and the agent creates a detailed execution plan and generates the required SQL and Python code. The agent maintains awareness of the notebook context, including available data sources, schemas, and catalog information, managing common tasks including data transformation, statistical analysis, and model development. This helps data engineers, analysts, and data scientists who spend significant time on manual setup tasks and boilerplate code build analytics and ML applications faster. For more information, see What's New Post.

Amazon Athena for Apache Spark is now available in Amazon SageMaker notebooks

Amazon SageMaker now supports Amazon Athena for Apache Spark, bringing a new notebook experience and fast serverless Spark experience together within a unified workspace. Now, data engineers, analysts, and data scientists can easily query data, run Python code, develop jobs, train models, visualize data, and work with AI from one place, with no infrastructure to manage and second-level billing. Athena for Apache Spark scales in seconds to support any workload, from interactive queries to petabyte-scale jobs. Athena for Apache Spark now runs on Spark 3.5.6, the same high-performance Spark engine available across AWS, optimized for open table formats including Apache Iceberg and Delta Lake. It brings you new debugging features, real-time monitoring in the Spark UI, and secure interactive cluster communication through Spark Connect. As you use these capabilities to work with your data, Athena for Spark now enforces table-level access controls defined in AWS Lake Formation. For more information, see What's New Post.

Visual Workflows now supports Amazon MWAA Serverless for IAM-based domains in Amazon SageMaker Unified Studio

Visual Workflows now supports Amazon MWAA Serverless for IAM-based domains in SageMaker unified studio. Visual Workflows is a drag-and-drop interface for creating and managing workflows. This low-code feature allows users to visually represent a series of tasks, such as data processing and analysis, with the option to customize further by switching from visual to code. Leveraging Amazon Managed Workflows for Apache Airflow (MWAA) Serverless, it enables users to create, modify, schedule, and monitor workflows without writing code or managing infrastructure. This addition simplifies workflow management and enhances usability for data engineers and scientists. For more information, see User Guide, Blog Post, and IAM-based domains and projects.

November 14, 2025

Amazon SageMaker Catalog now supports read and write access to Amazon S3

Amazon SageMaker Catalog now supports read and write access to Amazon S3 general purpose buckets. This capability helps data scientists and analysts search for unstructured data, process it alongside structured datasets, and share transformed datasets with other teams. Data publishers gain additional controls to support analytics and generative AI workflows within SageMaker Unified Studio while maintaining security and governance controls over shared data. For more information, see What's New Post.

November 10, 2025

Amazon SageMaker Unified Studio adds support for catalog notifications

Amazon SageMaker Unified Studio now provides real-time notifications for data catalog activities, enabling data teams to stay informed of subscription requests, dataset updates, and access approvals. With this launch, customers receive real-time notifications for catalog events including new dataset publications, metadata changes, and access approvals directly within the SageMaker Unified Studio notification center. This launch streamlines collaboration by keeping teams updated as datasets are published or modified. For more information, see What's New Post.

November 6, 2025

Custom tags for Project resources in Amazon SageMaker Unified Studio

Amazon SageMaker Unified Studio announced new capability of adding custom tags to resources created through the Amazon SageMaker Unified Studio projects. This helps customers enforce tagging standards that conform to Service Control Policies (SCP) and helps enable cost tracking reporting practices on resources created across the organization. For more information, see What's New Post and or the API Reference documentation.

October 2025

October 27, 2025

Amazon SageMaker adds additional search context for search results

Amazon SageMaker enhances search results in Amazon SageMaker Unified Studio with additional context that improves transparency and interpretability. Users can see which metadata fields matched their query and understand why each result appears, increasing clarity and trust in data discovery. The capability introduces inline highlighting for matched terms and an explanation panel that details where and how each match occurred across metadata fields such as name, description, glossary, schema, and other metadata. The enhancement reduces time spent evaluating irrelevant assets by presenting match evidence directly in search results. Users can quickly validate relevance without opening individual assets. For more information, see What's New Post and AWS Big Data Blog.

October 12, 2025

Continuous real-time ingestion of metadata from AWS Glue Data Catalog to SageMaker Catalog

We have launched a new capability within SageMaker Catalog that enables continuous real-time ingestion of metadata from AWS Glue Data Catalog to SageMaker Catalog. With this launch, once

you onboard your tables and views in AWS Glue Data Catalog to SageMaker Catalog, SageMaker Catalog continuously keeps metadata current via real-time ingestion. Any changes, such as new tables or schema updates made in AWS Glue Data Catalog, are automatically reflected in the SageMaker Catalog. This eliminates the need for periodic ingestion jobs, reduces stale metadata risk, and lowers operational costs for you while ensuring access to the freshest metadata. For more information, see Onboarding data in Amazon SageMaker Unified Studio.

October 11, 2025

Amazon SageMaker Unified Studio now supports remote connections from Visual Studio Code

Amazon SageMaker Unified Studio now supports remote connections from Visual Studio Code. Use your familiar VS Code environment with SageMaker's scalable compute resources. This integration lets you keep your development workflows while accessing AWS analytics and AI/ML services in a unified environment. For more information, see What's New Post, User Guide, Admin Guide, and Blog post.

Command Palette for Amazon SageMaker Unified Studio

TUse the Command Palette to navigate Unified Studio and perform actions with your keyboard. Press Cmd + K (macOS) or Ctrl + K (Windows/Linux) to open the Command Palette. The popup dialog lets you execute commands, navigate through Unified Studio, and discover keyboard shortcuts.

September 2025

Septemeber 18, 2025

Format change for Amazon SageMaker Unified Studio domain

New Amazon SageMaker Unified Studio domain identifiers now contain a hyphen character instead of an underscore. This change aligns with established host naming standards. Existing domains with the underscore character in the domain identifier will continue to work without any changes needed. Domain identifiers for newly created domains will have a hyphen. New format example, dzd-123456789abcde.

September 12, 2025

Continuous real-time ingestion of metadata from AWS Glue Data Catalog to SageMaker Catalog

We have launched a new capability within SageMaker Catalog that enables continuous real-time ingestion of metadata from AWS Glue Data Catalog to SageMaker Catalog. With this launch, once you onboard your tables and views in AWS Glue Data Catalog to SageMaker Catalog, SageMaker Catalog continuously keeps metadata current via real-time ingestion. Any changes, such as new tables or schema updates made in AWS Glue Data Catalog, are automatically reflected in the SageMaker Catalog. This eliminates the need for periodic ingestion jobs, reduces stale metadata risk, and lowers operational costs for you while ensuring access to the freshest metadata. For more information, see Onboarding data in Amazon SageMaker Unified Studio.

Septemeber 08, 2025

Enhanced AI assistance through improved Amazon Q Developer chat and integrations with Amazon Q CLI

Amazon SageMaker Unified Studio now offers enhanced AI assistance through improved Amazon Q Developer chat and integrations with Amazon Q CLI. By integrating with Model Context Protocol (MCP) servers, Amazon Q Developer is aware of your SageMaker Unified Studio project resources, including data, compute, and code, and provides personalized, relevant responses for data engineering and machine learning development. Users can now receive AI assistance for tasks such as code refactoring, file modification, and more with full transparency into the AI's reasoning and actions. These new capabilities are included at no additional cost with the Amazon Q Developer Free Tier and are available in all AWS Regions where SageMaker Unified Studio is supported.
To make even more use of these features, users can enable Amazon Q Developer Pro. For more information, see What's New Post, Admin Guide, and Blog Post.

Custom blueprints in Amazon SageMaker Unified Studio

Custom blueprint capability in Amazon SageMaker Unified Studio, allows customers to use their own managed policies or roles as per their corporate security policies for creating the project in SageMaker Unified Studio. Organizations can incorporate their specific dependencies, security controls using their own managed AWS Identity and Access Management (https://aws.amazon.com/iam/) (IAM) policies, and best practices, making it straightforward for them to align with internal standards. Because Custom Blueprints are defined through infrastructure as code (IaC), they are straightforward to version control, share across teams, and evolve over time. This speeds up onboarding and keeps projects consistent and governed, no matter how big or distributed customers' data organization becomes. For enterprises, this means more time focusing on insights, models, and innovation. The custom blueprints feature is designed to help teams move faster and stay consistent while maintaining their organization's security controls and best

practices. Sample templates can be found here(https://docs.aws.amazon.com/sagemaker-unified-studio/latest/adminguide/custom-blueprints.html). For more information, see What's New Post, Admin Guide, and Blog Post.

Introducing restricted classification terms for governed classification in Amazon SageMaker Catalog

AWS announced restricted classification terms in SageMaker Catalog. This new capability allows domain administrators to define governance-controlled terms and enforce which teams and users are authorized to apply them. Restricted classification terms are designed to allow organizations to set standards for consistent classification of sensitive data, help prevent misuse of regulatory tags, and enable downstream workflows such as automatic access grants across the enterprise. For more information, see Blog Post.

August 2025

August 22, 2025

Amazon SageMaker Unified Studio adds S3 file sharing options to projects experience

Amazon SageMaker Unified Studio now supports Trusted Identity Propagation (TIP) for SQL analytics use cases with Amazon Redshift and Amazon Athena. This feature enables administrators to grant permissions based on user attributes from their corporate identity center. Users can single-sign into databases with permissions based on their identity or group membership, while auditors can track access across Unified Studio query editor and services like Redshift, Athena, and Lake Formation. To implement, create or update your domain with TIP enabled in the project profile. TIP automatically enables for Amazon Athena workgroup, while Amazon Redshift requires creating a connection with IAM Identity Center connection type. For more information, see User Guide.For more information, see What's New User Guide and Admin Guide.

August 7, 2025

Amazon SageMaker Unified Studio Redshift Managed Workgroup experience

Amazon SageMaker Unified Studio launched Redshift Managed Workgroup, a collection of Amazon Redshift compute resources that AWS Glue manages. It becomes visible as a managed workgroup in Amazon Redshift when users register a serverless namespace to AWS Glue Data Catalog and create a Lake Formation catalog. All management of these workgroups must be done through

the AWS Glue Data Catalog interface. This eliminates the need for dedicated Redshift compute resources when querying Lakehouse catalogs, reducing costs, preserving data continuity, and accelerating project setup time.

August 5, 2025

Project deletion progress experience

Users can now track project deletion progress through a visual interface, providing clear visibility and status updates until the process is complete, enhancing the project management experience.

July 2025

July 16, 2025

Amazon SageMaker streamlines the S3 Tables workflow experience

This update simplifies table creation and management by eliminating the need to navigate multiple AWS consoles. Users can now create tables, load data, and run queries directly within Amazon SageMaker Unified Studio using either Query Editor or Jupyter Notebook. The update enables administrators to configure analytics integration with Amazon S3 and create custom profiles, allowing project owners to set up pre-configured catalogs and S3 Tables support more efficiently. For more information, see What's New and User Guide.

July 15, 2025

Automatic approval of subscription requests based on project membership

A subscription request is now automatically approved if the requester is already authorized to approve it manually. Specifically, if they are a member of both the project that published the asset and the project requesting access. To qualify for auto-approval, the requester must be listed as an owner or contributor in the project where the asset was originally published. And the requester must also be listed as an owner or contributor in the project making the subscription request. For more information, see User Guide.

July 16, 2025

Data lineage enhancements

Amazon SageMaker Unified Studio has now contributed a 'custom transport' to the OpenLineage community that allows builders to download the transport along with OpenLineage plugins to augment and automate lineage events captured from OpenLineage-enabled systems. With this, customers can automate lineage capture and send these lineage events to the Amazon SageMaker unified domain, enhancing data governance and traceability within their data workflows. Additionally lineage enhancements include automated lineage events capture for additional data sources using AWS Glue crawlers, improved its SQL lineage support for stored procedures and materialized views in Amazon Redshift, and from tools such as vETL processes and notebooks (interactive executions and remote workflows). For more information, see What's New and User Guide.

July 15, 2025

Amazon QuickSight Integration

Amazon QuickSight Integration is now available, enabling users to seamlessly build and visualize dashboards using project data directly from Amazon SageMaker Unified Studio. The integration automatically creates secured QuickSight datasets, organizes dashboards within project folders, and allows publishing to the SageMaker Catalog for broader discovery and reuse. This keeps your dashboards organized, discoverable, shareable, and governed, making cross-team collaboration and asset reuse much easier. For more information, see What's New, User Guide, Admin Guide, AWS News Blog, and the AWS Big Data Blog.

July 15, 2025

Data management with automated Lakehouse onboarding and metadata ingestion

Data management is simplified with the introduction of two key capabilities in Amazon SageMaker. First, automated metadata ingestion for datasets into SageMaker Catalog during domain creation or updates, and direct sharing of assets between projects. The automated onboarding eliminates manual IAM configuration and metadata ingestion tasks, making datasets immediately available for governance and analysis. Second, direct sharing enables seamless cross-team collaboration while maintaining governance controls, reducing administrative overhead and accelerating project timelines. For more information, see What's New, Admin Guide, and AWS News Blog.

July 15, 2025

Amazon SageMaker Catalog adds support for Amazon S3 general purpose buckets

Amazon SageMaker Catalog now supports Amazon S3 general purpose buckets to enable data producers to share unstructured data as "S3 Object Collections" within the SageMaker Catalog. Users can curate these assets with business metadata, making them discoverable and accessible to data scientists, engineers, and analysts while maintaining granular security controls. This feature streamlines cross-team data sharing, enhances data governance, and allows consumers to subscribe to assets for ongoing access and updates. For more information, see What's New and User Guide.

July 15, 2025

Unified Metadata Governance with SageMaker-Collibra solution

Unified Metadata Governance with SageMaker-Collibra solution addresses organizational challenges by connecting Amazon SageMaker Catalog with Collibra, enabling centralized metadata management and streamlined access workflows. This integration reduces metadata duplication, enhances governance controls, and builds data trust across analytics and AI platforms. For more information, see AWS Big Data Blog.

July 15, 2025

Visual Workflows

Visual Workflows introduces a drag-and-drop interface for creating and managing workflows within Amazon SageMaker Unified Studio. This low-code feature allows users to visually represent a series of tasks, such as data processing and analysis, with the option to customize further by switching from visual to code. Leveraging Amazon Managed Workflows for Apache Airflow (MWAA), it enables users to create, modify, schedule, and monitor workflows easily. This addition simplifies workflow management and enhances usability for data engineers and scientists. For more information, see What's New, User Guide, and Blog Post.

July 15, 2025

Data processing jobs experience

A new data processing jobs experience is available within Amazon SageMaker Unified Studio. Jobs enables users to author, manage, monitor and troubleshoot data processing workloads across their organization and collaborate in projects to securely build and share data processing jobs and workflows. Users can create jobs through multiple methods including ETL scripts,

interactive notebooks, or the Visual ETL editor, with options for on-demand execution, scheduling, or workflow orchestration. The feature includes monitoring capabilities and AI-powered troubleshooting for failed jobs. For more information, see What's New, User Guide, and Blog Post.

July 3, 2025

Enhanced Data Explorer object actions

Data Explorer is being enhanced with user-friendly actions that streamline database operations, including a UI-based table definition creator, auto-generation of common SQL queries, a quick actions menu, and direct object manipulation capabilities. These features reduce the need for manual query writing and boost overall productivity.

July 3, 2025

Bring S3 buckets to your project

To bring in Amazon S3 data either in the same account or a different account to your project, you must first gain access to the data and then add the data to your project. You can gain access to the data by using the project role or an access role. Then you have to create a S3 connection using the Add Data option from data explorer. For more information, see User Guide.

July 3, 2025

Uploading data from your desktop to create a Redshift table

You can now upload data from your desktop in CSV, JSON, Parquet, or Delimiter formats to create either a data lake or Amazon Redshift table using Amazon SageMaker Unified Studio. You can select Add data option from the data explorer and then select create table. For more information, see User Guide.

July 25, 2025

Text search capability of Query history experience

Amazon SageMaker Unified Studio introduces text search capability for query history in both Athena and Redshift engines. This enhancement enables users to efficiently search through their historical queries, improving workflow productivity and query reusability. For more information, see User Guide.

July 23, 2025

Trusted identity propagation experience

Amazon SageMaker Unified Studio now supports Trusted Identity Propagation (TIP) for SQL analytics use cases with Amazon Redshift and Amazon Athena. This feature enables administrators to grant permissions based on user attributes from their corporate identity center. Users can single-sign into databases with permissions based on their identity or group membership, while auditors can track access across Unified Studio query editor and services like Redshift, Athena, and Lake Formation. To implement, create or update your domain with TIP enabled in the project profile. TIP automatically enables for Amazon Athena workgroup, while Amazon Redshift requires creating a connection with IAM Identity Center connection type. For more information, see User Guide.For more information, see User Guide and Admin Guide.

July 2, 2025

Support of SQL generation actions in Data Explorer experience

Amazon SageMaker Unified Studio now supports SQL generation actions in Data Explorer, introducing automated generation of table definitions and common SQL queries (INSERT/UPDATE/SELECT/DROP). The feature includes a quick actions menu for frequently used operations and direct object manipulation capabilities. These enhancements significantly reduce manual query writing effort and improve overall productivity for data analysts and scientists.

June 2025 (Additional)

June 30, 2025

Next Generation Amazon SageMaker user guide

Next Generation Amazon SageMaker user guide was launched on 6/30 and it introduces the main components of Next Generation Amazon SageMaker and includes 7 new use case-based tutorials to help customers get started. For more information, see Next Generation Amazon SageMaker user guide.

June 25, 2025

Automatic file synchronization between project Git repositories and Amazon S3 buckets

Amazon SageMaker Unified Studio now offers automatic file synchronization between project Git repositories and Amazon S3 buckets. This new capability streamlines data and AI development workflows by ensuring your Git repository files are automatically mirrored to S3. For more information, see What's New, Admin Guide, and AWS Big Data Blog.

June 2025

June 12, 2025

Workflow Improvements

Enhanced default workflow template: The default start_date now uses a static date instead of a relative date. This prevents potential workflow trigger failures, improves workflow reliability and predictability.

Notebook output capabilities: Download notebook output directly, export results for offline analysis and share findings more easily with your team.

June 19, 2025

Auto complete support for large number of tables and columns experience

Amazon SageMaker Unified Studio now delivers enhanced autocomplete capabilities for enterprise-scale Amazon Redshift databases. The feature supports seamless autocomplete functionality for massive databases containing up to 200,000 tables, while eliminating previous restrictions on the number of schemas or columns that can be indexed and suggested. Using real-time database metadata, it provides comprehensive coverage across your entire Redshift environment, including both user-defined native tables and critical system tables. This enables data analysts, engineers, and scientists working with complex, large-scale data warehouses to navigate their database structures effortlessly, dramatically reducing query development time and minimizing syntax errors.

June 5, 2025

CloudFormation template

Amazon SageMaker Unified Studio now provides an CloudFormation template to setup and create a new Amazon SageMaker Unified domain and project.

June 2, 2025

Amazon DataZone and Amazon SageMaker integration

Amazon DataZone and Amazon SageMaker introduced a new UI feature that enables DataZone domains to be upgraded and utilized within the next generation of Amazon SageMaker. This update preserves customers' Amazon DataZone investments when transferring to Amazon SageMaker. The integration allows all content and assets created in DataZone, including metadata forms, glossaries, and subscriptions, to remain accessible through Amazon SageMaker Unified Studio post-upgrade. This upgrade pathway demonstrates Amazon's commitment to providing continuity and value preservation for customers using their machine learning and data management services. For more information, see What's New and Upgrade Amazon DataZone domains to Amazon SageMaker unified domains.

May 2025

May 30, 2025

Increased concurrent spaces support

Amazon SageMaker Unified Studio now supports up to 6,000 concurrent spaces per customer account, representing a 2.4x increase from the previous limit. This significant scaling improvement enables organizations to support larger teams and more concurrent workloads within their Amazon SageMaker Unified Studio environment.

May 30, 2025

Visual ETL improvements

Amazon SageMaker Unified Studio visual ETL now supports cloning a visual ETL flow as a Jupyter notebook, allowing users to continue editing the flow with code. This streamlines the workflow for users who prefer to start authoring ETL pipelines visually and then transition to a code-based approach. Visual ETL also added support for automatically inferring schemas from CSV files within S3 nodes, improving the authoring process and reducing manual work needed.

May 29, 2025

Native Redshift table creation enhancements

You can now create native Redshift tables directly through local file uploads, supporting multiple file formats (CSV, JSON, PARQUET) with various compression options and increased file size limit to 100MB. The enhancement includes automatic schema inference with preview capabilities and adds PARQUET support in Glue upload flow. This eliminates the previous multi-step process that required manual S3 file copying, table creation, and data insertion, making the workflow more efficient and less error-prone.

May 23, 2025

Customer Managed Custom Master (CM-CMK) support for Persistent App UI

Customer Managed Custom Master (CM-CMK) support for Persistent App UI, an essential tool for debugging big data applications. This feature extends SSE encryption applied to logs managed in the EMR service bucket to honor Customer Managed Custom Master Key (CM-CMK). This feature enables customers to manage accessibility of logs after their clusters terminate. For more information, see About Amazon EMR Releases and Configure Amazon EMR cluster logging and debugging.

May 22, 2025

Global Search Bar

Amazon SageMaker Unified Studio introduced a new Global Search Bar, providing a persistent search interface in the top navigation and landing page, simplifying discover and access data without navigating through multiple menus. The intuitive search functionality currently supports business catalog data and projects, offering clear visual indicators for search result scope. This user-centric enhancement represents AWS's commitment to improving the overall SageMaker Studio experience, making data and resources more accessible to users.

May 12, 2025

Code Editor and Multiple Spaces support

AWS launched two complementary capabilities in the next generation of Amazon SageMaker that enhance the development experience for analytics, machine learning (ML), and GenAI teams: Code Editor and Multiple Spaces support. The Code Editor, based on Code-OSS (Open Source Software) like VS Code, offers a powerful IDE experience with familiar shortcuts, terminal access, and advanced development tools, while supporting thousands of VS Code-compatible extensions from Open VSX. It enables seamless version control through major Git platforms and

comes preconfigured with Amazon SageMaker distribution for ML frameworks. To maximize the benefits of Code Editor alongside other coding interfaces in Unified Studio, including JupyterLab, SageMaker now supports multiple spaces per user per project, allowing users to manage parallel workstreams with different computational needs. For more information, see What's New and Using the Code Editor IDE in Amazon SageMaker Unified Studio.

May 12, 2025

Bring Your Own Image (BYOI)

Amazon SageMaker Unified Studio now allows you to bring your own image (BYOI). This feature benefits customers who have regulatory and compliance requirements or who prefer not to use the framework containers that come with the default SageMaker Distribution image. For more information, see What's New and Bring your own image (BYOI).