Implementation Guide

Workload Discovery on AWS

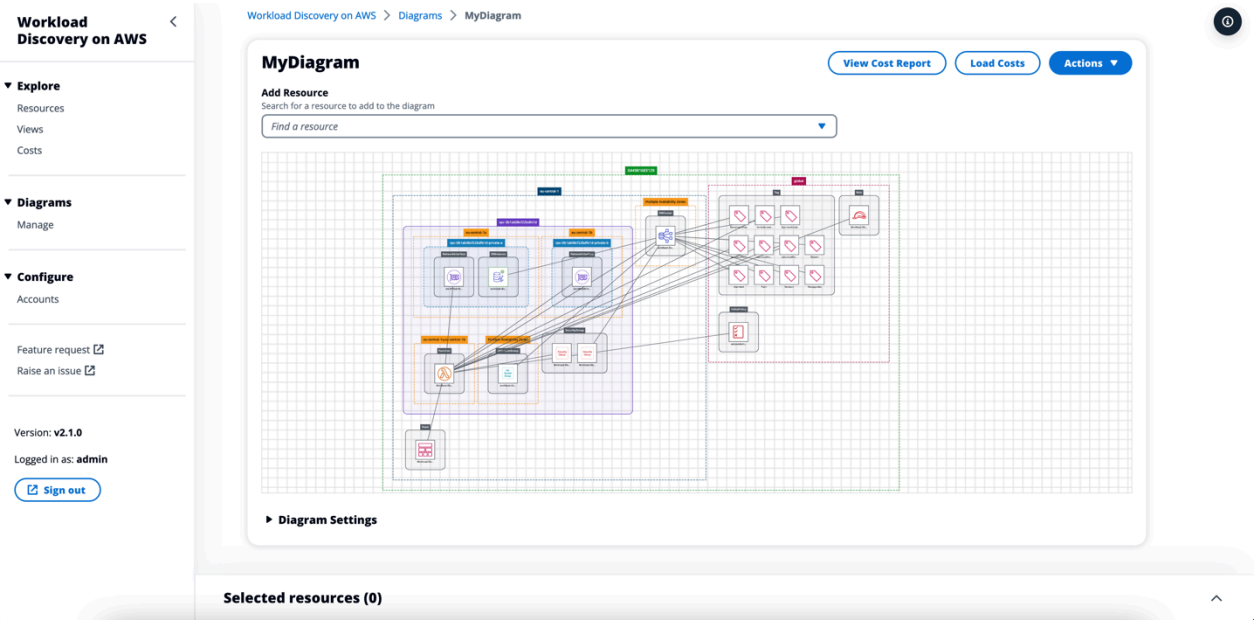
Workload Discovery on AWS: Implementation Guide

Table of Contents

Deploy a visualization tool that automatically generates architecture diagrams of AWS Cloud workloads

Monitoring your Amazon Web Services (AWS) Cloud workloads is key to maintaining operational health and efficiency. However, keeping track of the AWS resources and the relationships between them can be a challenge. Workload Discovery on AWS is a visualization tool that automatically generates architecture diagrams of your workload on AWS. You can use this solution to build, customize, and share detailed workload visualizations based on live data from AWS.

This solution works by maintaining an inventory of the AWS resources across your accounts and Regions, mapping relationships between them, and displaying them in a web user interface (web UI). When making changes to a resource, Workload Discovery on AWS saves you time by providing a link to the resource in the AWS Management Console.



Sample architecture diagram generated by Workload Discovery on AWS

This implementation guide describes architectural considerations and configuration steps for deploying Workload Discovery on AWS in the AWS Cloud. It includes links to an AWS CloudFormation template that launches and configures the AWS services required to deploy this solution using AWS best practices for security and availability.

The intended audience for implementing the Workload Discovery on AWS solution in their environment includes solution architects, business decision makers, DevOps engineers, data scientists, and cloud professionals.

Use this navigation table to quickly find answers to these questions:

If you want to . . .	Read . . .
Know the cost for running this solution. The estimated cost for running this solution in the US East (N. Virginia) Region is USD $425.19 per month.	Cost
Understand the security considerations for this solution.	Security
Know how to plan for quotas for this solution.	Quotas
Know which AWS Regions support this solution.	Supported AWS Regions
View or download the AWS CloudFormation template included in this solution to automatically deploy the infrastructure resources (the "stack") for this solution.	AWS CloudFormation template
Access the source code.	GitHub repository

Features and benefits

Workload Discovery on AWS provides the following features:

Build architecture diagrams using near real-time data

Workload Discovery on AWS scans your accounts every 15 minutes to ensure that the diagrams you create are an accurate and current representation of your workloads.

View resources from multiple accounts and Regions in one place

The solution maintains an inventory of the AWS resources across your AWS accounts and Regions in a centralized graph database, allowing you to explore multiple accounts and Regions and their relationships to each other in a single UI.

AWS Organizations integration

When deploying the solution with AWS Organizations, Workload Discovery on AWS will automatically discover all the supported resources in your organization. In this configuration, there is no need to directly manage the deployment of account specific CloudFormation templates to make these accounts available for discovery.

Collate cost data across your workloads

When enabled, the cost feature allows you to search for resources in your account by cost and add the resources you find to a diagram. You can also add cost data to already existing diagrams.

Export to diagrams.net (formerly draw.io)

Workload Discovery on AWS can export your diagrams so that you can further annotate them using this third-party drawing software.

Integration with AWS Service Catalog AppRegistry and Application Manager, a capability of AWS Systems Manager

This solution includes a Service Catalog AppRegistry resource to register the solution's CloudFormation template and its underlying resources as an application in both Service Catalog AppRegistry and Application Manager. With this integration, you can centrally manage the solution's resources and enable application search, reporting, and management actions.

Use cases

Design and security reviews

Use this solution to generate architecture diagrams to validate that the implementation of a workload matches the proposed design.

Explore and document existing workloads

Create architecture diagrams to explore workloads where little documentation exists or that were deployed manually without infrastructure as code.

Visualize costs

Generate a cost report for your architecture diagrams that contains an overview of the estimated cost.

Concepts and definitions

This section describes key concepts and defines terminology specific to this solution:

resource

An AWS resource, such as an Amazon Simple Storage Service (Amazon S3) bucket or AWS Lambda function.

relationship

A link between two resources, such as an AWS Identity and Access Management (IAM) role and an associated AWS Lambda function.

resource type

The classification category of a resource. Always follows the CloudFormation naming convention, such as `AWS::Lambda::Function`.

discovery

The process that the solution initiates to map resources and their relationships in your AWS accounts and Regions.

account discovery mode

The method of discovering accounts and adding them to the solution: either self-managed through the Workload Discovery on AWS UI or delegated to AWS Organizations.

> **Note**
>
> For a general reference of AWS terms, see the AWS Glossary.

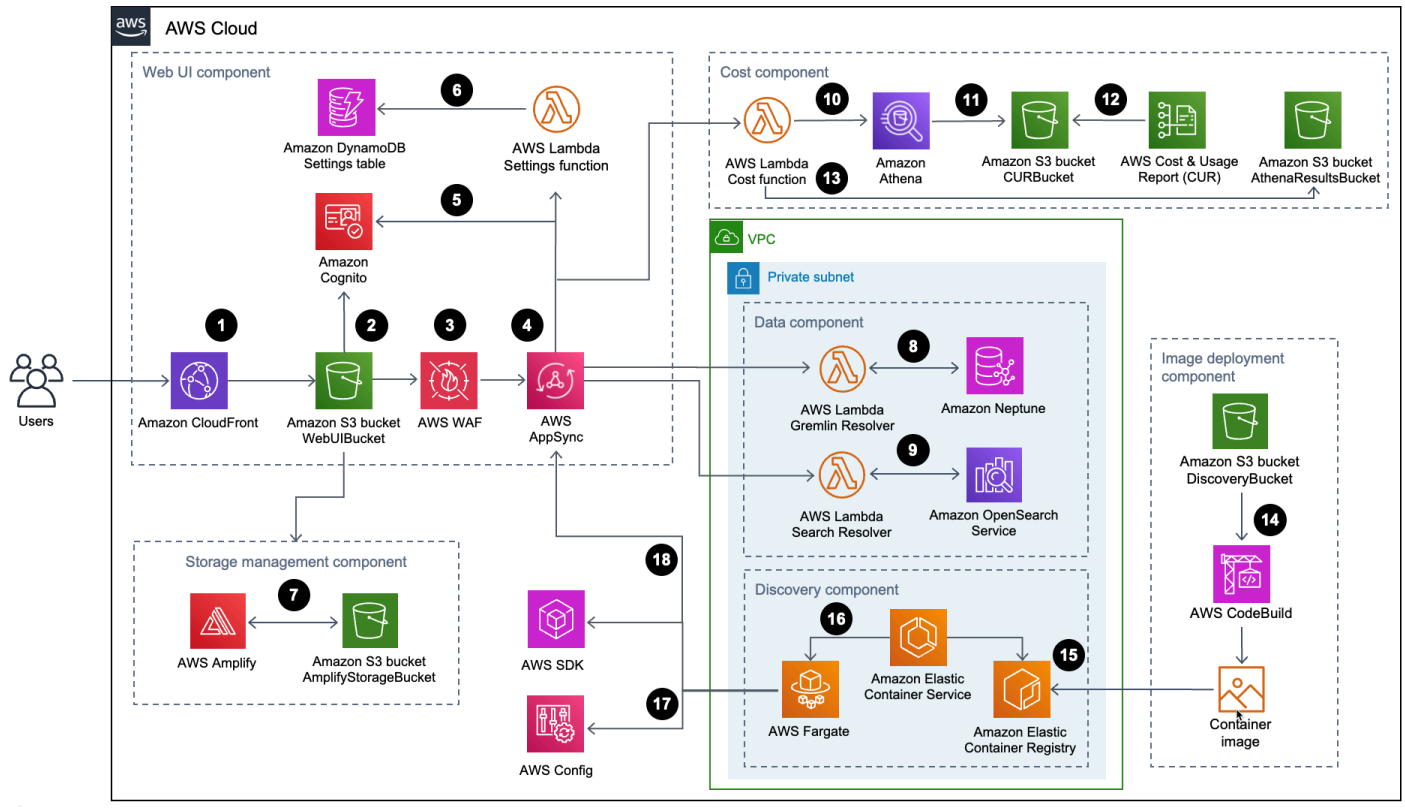
Architecture overview

This section provides a reference implementation architecture diagram for the components deployed with this solution.

Architecture diagram

Deploying this solution with the default parameters builds the following environment in the AWS Cloud.

Workload Discovery on AWS architecture



The high-level process flow for the solution components deployed with the AWS CloudFormation template is as follows:

1. HTTP Strict-Transport-Security (HSTS) adds security headers to each response from the Amazon CloudFront distribution.
2. An Amazon Simple Storage Service (Amazon S3) bucket hosts the web UI, which is distributed with Amazon CloudFront. Amazon Cognito authenticates user access to the web UI.

3. AWS WAF protects the AppSync API from common exploits and bots that can affect availability, compromise security, or consume excessive resources.
4. AWS AppSync endpoints allow the web UI component to request resource relationship data, query costs, import new AWS Regions, and update preferences. AWS AppSync also allows the discovery component to store persistent data in the solution's databases.
5. AWS AppSync uses JSON Web Tokens (JWTs) provisioned by Amazon Cognito to authenticate each request.
6. The `Settings` AWS Lambda function persists imported Regions and other configurations to Amazon DynamoDB.
7. The solution deploys AWS Amplify and an Amazon S3 bucket as the storage management component to store user preferences and saved architecture diagrams.
8. The data component uses the `Gremlin Resolver` AWS Lambda function to query and return data from an Amazon Neptune database.
9. The data component uses the `Search Resolver` Lambda function to query and persist resource data into an Amazon OpenSearch Service domain.
10. The `Cost` Lambda function uses Amazon Athena to query AWS Cost and Usage Reports (AWS CUR) to provide estimated cost data to the web UI.
11. Amazon Athena runs queries on AWS CUR.
12. AWS CUR delivers the reports to the `CostAndUsageReportBucket` Amazon S3 bucket.
13. The `Cost` Lambda function stores the Amazon Athena results in the `AthenaResultsBucket` Amazon S3 bucket.
14. AWS CodeBuild builds the discovery component container image in the image deployment component.
15. Amazon Elastic Container Registry (Amazon ECR) contains a Docker image provided by the image deployment component.
16. Amazon Elastic Container Service (Amazon ECS) manages the AWS Fargate task and provides the configuration required to run the task. AWS Fargate runs a container task every 15 minutes to refresh inventory and resource data.
17. AWS Config and AWS SDK calls help the discovery component maintain an inventory of resource data from imported Regions, then store its results in the data component.
18. The AWS Fargate task persists the results of the AWS Config and AWS SDK calls into an Amazon Neptune database and an Amazon OpenSearch Service domain with API calls to the AppSync API.

AWS Well-Architected design considerations

This solution uses the best practices from the AWS Well-Architected Framework which helps customers design and operate reliable, secure, efficient, and cost-effective workloads in the cloud.

This section describes how the design principles and best practices of the Well-Architected Framework benefit this solution.

Operational excellence

We architected this solution using the principles and best practices of the operational excellence pillar to benefit this solution.

- Resources defined as infrastructure as code using CloudFormation.
- The solution pushes metrics to Amazon CloudWatch to provide observability into the infrastructure, Lambda functions, Amazon ECS tasks, AWS S3 buckets, and the rest of the solution components.

Security

We architected this solution using principles and best practices of the security pillar to benefit this solution.

- Amazon Cognito authenticates and authorizes web UI app users.
- All roles used by the solution follow least-privilege access. In other words, they only contain minimum permissions required so that the service can function properly.
- Data at rest and transit is encrypted using keys stored in AWS Key Management Service (AWS KMS)--a dedicated key management store.
- Credentials have a short expiration and follow a strong password policy.
- AWS AppSync security GraphQL directives give fine-grained control over what operations can be invoked by the frontend and backend.
- Logging, tracing, and versioning is turned on where applicable.
- Automatic patching (minor version) and snapshot creation is turned on where applicable.
- Network access is private by default with Amazon Virtual Private Cloud (Amazon VPC) endpoints being turned on where available.

Reliability

We architected this solution using principles and best practices of the reliability pillar to benefit this solution.

- The solution uses AWS serverless services wherever possible to ensure high availability and recovery from service failure.
- All compute processing uses Lambda functions or Amazon ECS on AWS Fargate.
- All custom code uses the AWS SDK and requests are throttled on the client side to prevent reaching API rate quotas.

Performance efficiency

We architected this solution using principles and best practices of the performance efficiency pillar to benefit this solution.

- The solution uses AWS serverless architecture where possible. This removes the operational burden of managing physical servers.
- The solution can launch in any Region that supports AWS services used in this solution such as: AWS Lambda, Amazon Neptune, AWS AppSync, Amazon S3, and Amazon Cognito.
- In supported Regions, Amazon Neptune serverless allows you to run and instantly scale graph workloads, without the need to manage and optimize database capacity.
- The solution uses managed services throughout to reduce the operational burden of resource provisioning and management.

Cost optimization

We architected this solution using principles and best practices of the cost optimization pillar to benefit this solution.

- AWS ECS on AWS Fargate uses Lambda functions exclusively for compute and only charges based on use.
- Amazon DynamoDB scales capacity on demand, so you only pay for the capacity you use.

Sustainability

We architected this solution using principles and best practices of the sustainability pillar to benefit this solution.

- The solution uses managed and serverless services where possible to minimize the environmental impact of the backend services.

Architecture details

This section describes the components and AWS services that make up this solution and the architecture details on how these components work together.

Authentication mechanism

Workload Discovery on AWS uses an Amazon Cognito user pool for both the UI and AWS AppSync authentication. Once authenticated, Amazon Cognito provides a JSON Web Token (JWT) to the web UI that will be provided with all subsequent API requests. If a valid JWT is not provided, the API request will fail and return an HTTP 403 Forbidden response.

Supported resources

For a list of AWS resource types that Workload Discovery on AWS can discover within your accounts and Regions, refer to Supported resources.

Workload Discovery on AWS architecture diagram management

You can save Workload Discovery on AWS architecture diagrams using the web UI where create, read, update, and delete (CRUD) operations can be performed. The AWS Amplify storage API allows Workload Discovery on AWS to store architecture diagrams in an Amazon S3 bucket. There are two levels of permissions available:

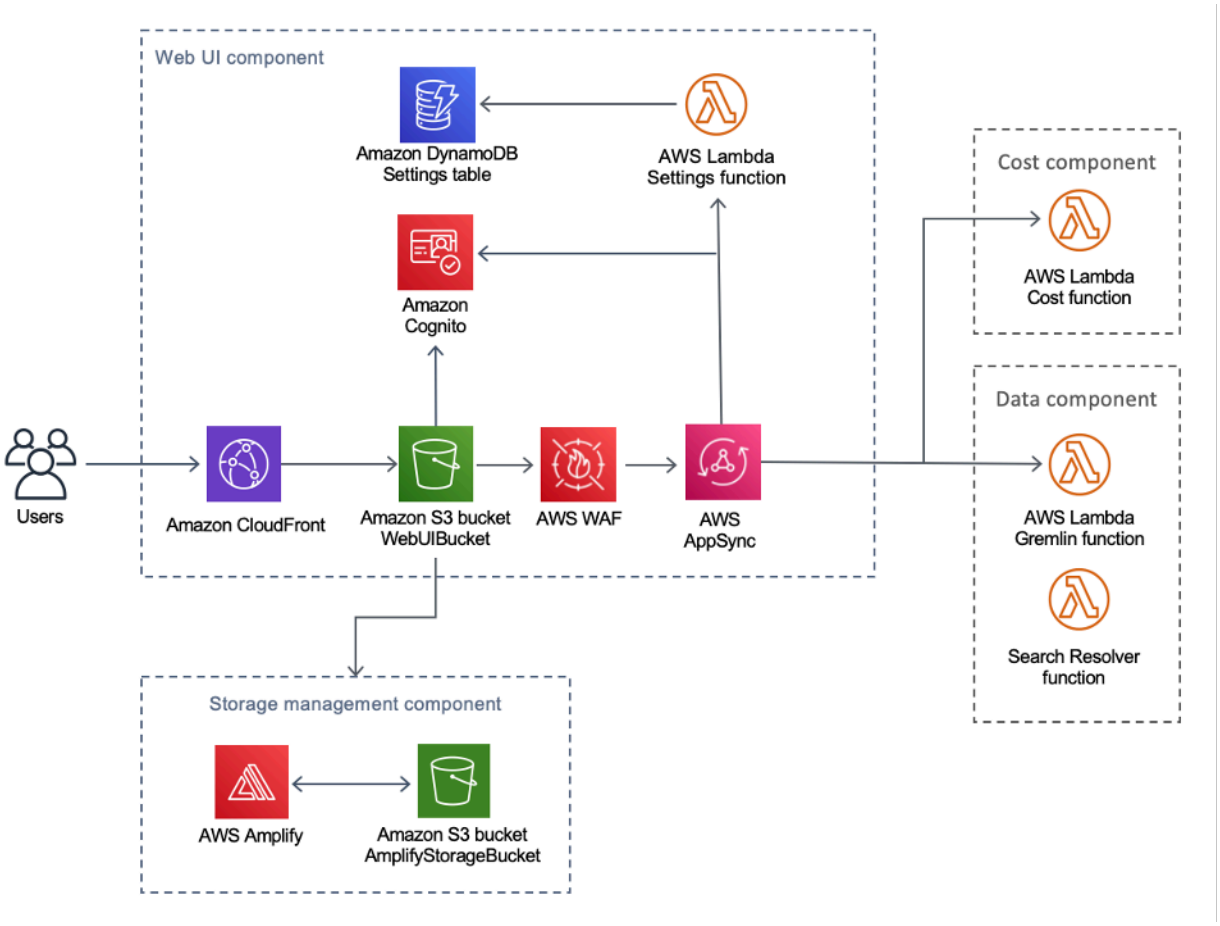
- **All users** - Allows Workload Discovery on AWS architecture diagrams to be visible to Workload Discovery on AWS users in your deployment. Users can download and edit these diagrams.
- **You** - Allows Workload Discovery on AWS architecture diagrams to be visible only to the creator. Other users will not be able to view them.

Web UI and storage management

We developed the web UI using React. The web UI provides a frontend console to allow users to interact with Workload Discovery on AWS.

Amazon CloudFront is configured to append secure headers to every HTTP request to the web UI. This provides an additional layer of security, protecting against attacks such as cross-site scripting (XSS).

Workload Discovery on AWS web UI and storage management components

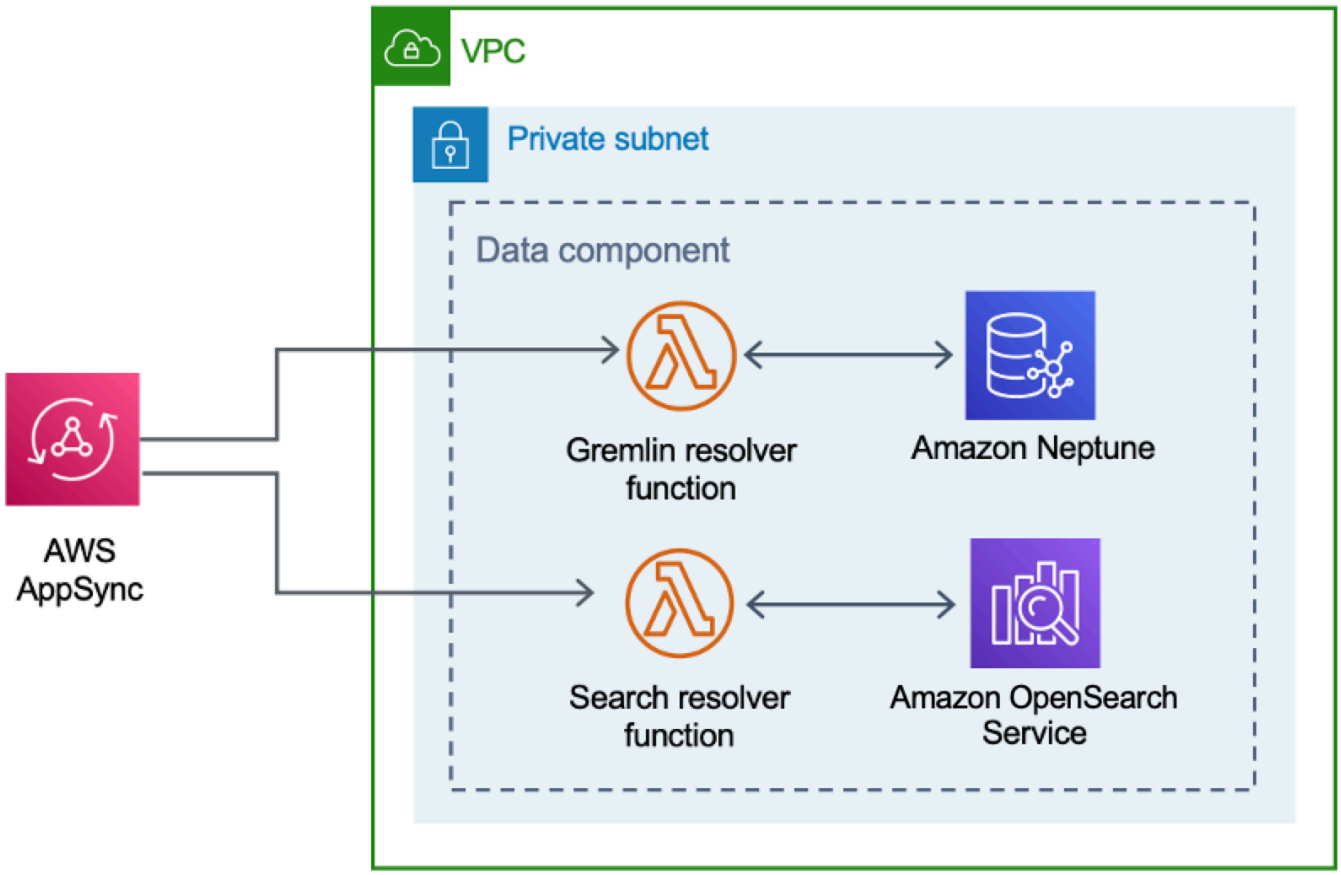


The web UI resources are hosted in the `WebUIBucket` Amazon S3 bucket and distributed by Amazon CloudFront. AWS Amplify provides an abstraction layer to simplify the integrations to AWS AppSync and Amazon S3.

This solution uses AWS AppSync to facilitate interaction with various configurations available to Workload Discovery on AWS, including managing imported Regions. AWS AppSync utilizes the `Settings` AWS Lambda function to handle requests such as importing a new account or Region.

Data component

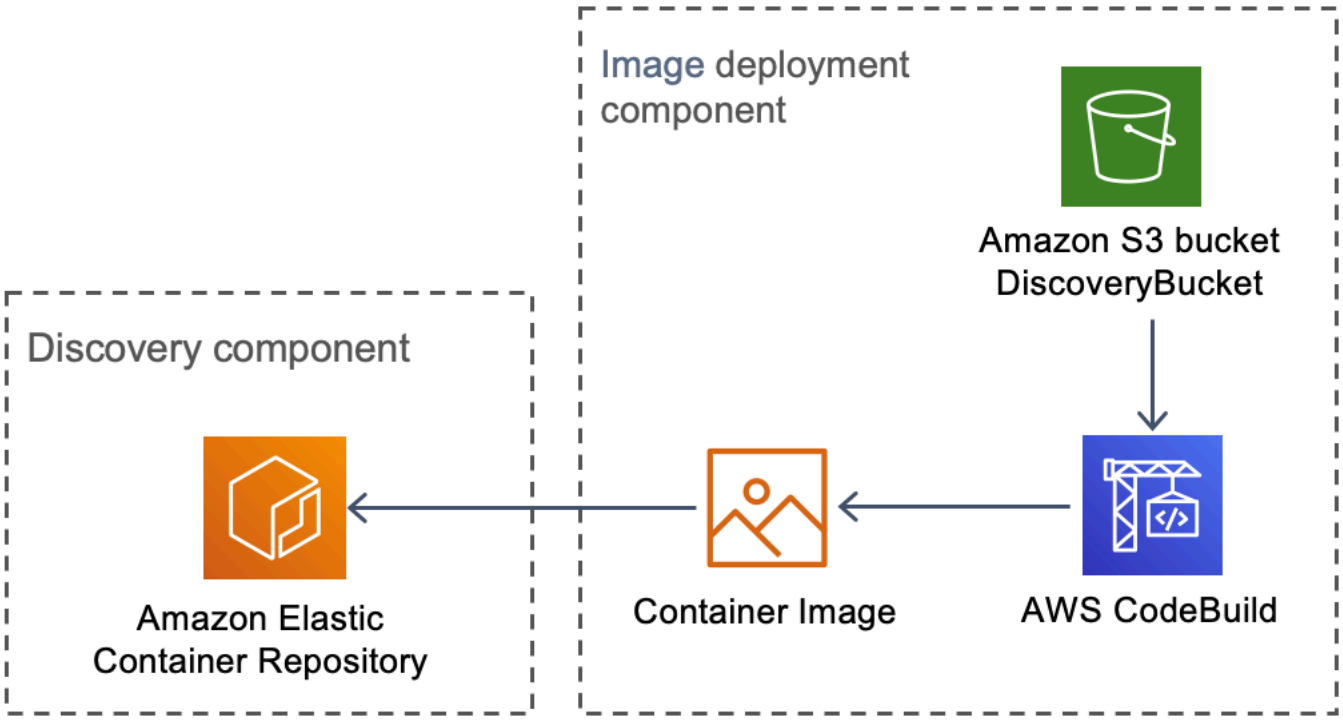
Workload Discovery on AWS data component



The web UI sends requests to the AppSync API, which invokes either the `Gremlin Resolver` or `Search Resolver` Lambda functions. These functions process the requests and query Amazon Neptune or OpenSearch Service to retrieve data about the provided resources. AWS AppSync also supports requests for the estimated cost data from the AWS CUR.

The discovery component sends requests to the AppSync API to read from and persist data in the Amazon Neptune and OpenSearch Service databases. The API receives requests from the AWS Fargate task in the discovery component. The API is then authenticated using an IAM role that provides access to the databases.

Image deployment component



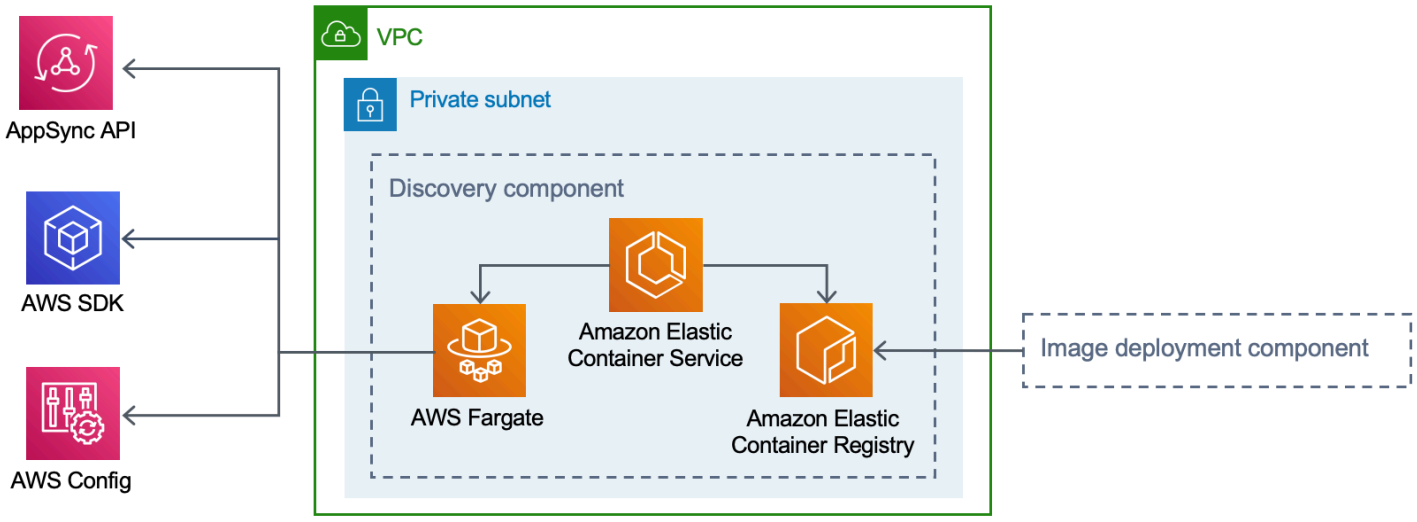
Workload Discovery on AWS image deployment component

The image deployment component builds the container image that the discovery component uses. The `DiscoveryBucket` and Amazon S3 bucket host the code which can be downloaded at time of deployment by an AWS CodeBuild job that builds the container image and uploads it to Amazon ECR.

Discovery component

The discovery component is the main data-gathering element of the Workload Discovery on AWS architecture. It is responsible for querying AWS Config and making *describe* API calls to maintain the inventory of resources and their relationships between one another.

Workload Discovery on AWS discovery component



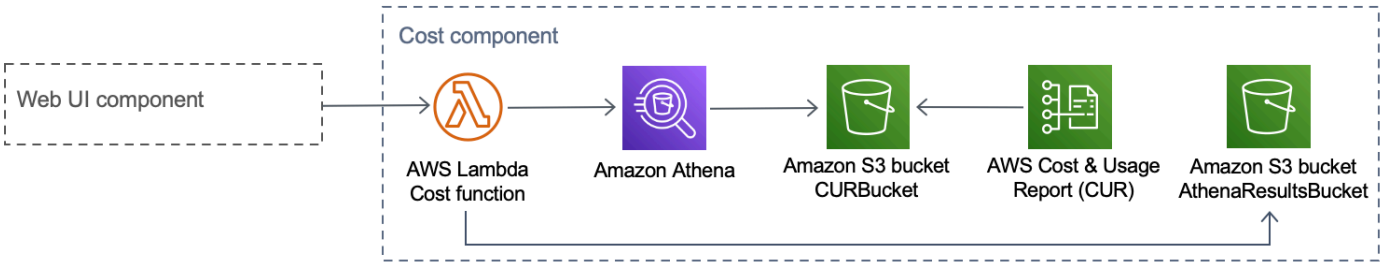
This solution configures Amazon ECS to run an AWS Fargate task using the container image downloaded from Amazon ECR. The AWS Fargate task is scheduled to run at 15-minute intervals. The resource relationship data that is collected is inserted into an Amazon Neptune graph database and Amazon OpenSearch Service.

The discovery component workflow consists of the following three steps:

1. Amazon ECS invokes an AWS Fargate task at 15-minute intervals.
2. The Fargate task gathers resource data from AWS Config, AWS API *describe* calls, and from the Amazon Neptune database.
3. The Fargate task calculates the difference between what is present in the Amazon Neptune database and what it has received from AWS Config and the *describe* calls.
4. The Fargate task sends requests to the AppSync API to persist the changes to resources and relationships discovered into Amazon Neptune and Amazon OpenSearch Service.

Cost component

Workload Discovery on AWS cost component



You can create an AWS CUR in AWS Billing and Cost Management and Cost Management. This publishes a Parquet formatted file to the `CostAndUsageReportBucket` Amazon S3 bucket. The web UI makes requests to the AWS AppSync endpoint that invokes the Cost Lambda function. The function sends predefined queries to Amazon Athena that return estimated cost information from AWS CUR.

Due to the size of the AWS CUR, the responses from Amazon Athena can be very large. The solution stores the results in the `AthenaResultsBucket` Amazon S3 bucket and paginates the results back to the web UI. The lifecycle policy configured on this bucket removes items that are more than seven days old.

AWS services in this solution

AWS service	Description
AWS AppSync	**Core.** This solution uses AppSync to provide a serverless GraphQL API that the Web UI consumes.
Amazon CloudFront	**Core**. This solution uses CloudFront with an Amazon S3 bucket as the origin. This restricts access to the Amazon S3 bucket so that it is not publicly accessible and prevents direct access from the bucket.
AWS Config	**Core**. The solution uses AWS Config as the primary data source for the resources and relationships the solution discovers.

AWS service	Description
Amazon OpenSearch Service	**Core**. The solution uses Amazon OpenSearch Service for application monitoring, log analytics, and observability.
Amazon DynamoDB	**Core**. This solution uses DynamoDB to store configuration data for the solution.
Amazon Elastic Container Service (ECS)	**Core**. This solution uses Amazon ECS to orchestrate running the task that discovers resources and relationships in your AWS accounts.
AWS Fargate	**Core**. This solution uses AWS Fargate on Amazon ECS as the compute layer for the discovery task.
AWS Lambda	**Core.** This solution uses serverless Lambda functions, with Node.js and Python runtimes, to handle API calls.
Amazon Neptune	**Core.** This solution uses Neptune as the primary datastore for the resources and relationships the solution discovers.
Amazon Simple Storage Service	**Core.** This solution uses Amazon S3 for frontend and backend storage purposes.
Amazon CloudWatch	**Supporting.** This solution uses CloudWatch to collect and visualize real-time logs, metrics, and event data in automated cases. Additionally, you can monitor the deployed solution's resource usage and performance issues.
AWS CodeBuild	**Supporting**. This solution uses CodeBuild to build the Docker container that contains the code for the discovery task and to deploy the assets for the frontend to Amazon S3.

AWS service	Description
Amazon Cognito	**Supporting.** This solution uses Cognito user pools to authenticate and authorize users to access the solution web UI.
AWS Systems Manager	**Supporting.** This solution uses AWS Systems Manager to provide application-level resource monitoring and visualization of resource operations and cost data.
Amazon Virtual Private Cloud	**Supporting**. This solutions uses a VPC to launch Neptune and OpenSearch databases in.
AWS WAF	**Supporting.** This solution uses AWS WAF to protect the AppSync API from common exploits and bots that can affect availability, compromise security, or consume excessive resources.
Amazon Athena	**Optional.** This solution uses Athena to query Cost and Usage Reports if the cost feature is enabled.

Plan your deployment

This section describes the Region, cost, security, and other considerations prior to deploying the solution.

Supported AWS Regions

This solution uses the Amazon Cognito service, which is not currently available in all AWS Regions. For the most current availability of AWS services by Region, see the AWS Regional Services List.

Workload Discovery on AWS is available in the following AWS Regions:

Region Name	
US East (N. Virginia)	Canada (Central)
US East (Ohio)	Europe (London)
US West (Oregon)	Europe (Frankfurt)
Asia Pacific (Mumbai)	Europe (Ireland)
Asia Pacific (Seoul)	Europe (Paris)
Asia Pacific (Singapore)	Europe (Stockholm)
Asia Pacific (Sydney)	South America (São Paulo)
Asia Pacific (Tokyo)	

Workload Discovery on AWS is not available in the following AWS Regions:

Region Name	Unavailable Service
AWS GovCloud (US-East)	AWS AppSync
AWS GovCloud (US-West)	AWS AppSync

Region Name	Unavailable Service
China (Beijing)	Amazon Cognito
China (Ningxia)	Amazon Cognito

Cost

You are responsible for the cost of the AWS services provisioned while running this solution. As of this revision, the cost for running this solution using the single instance deployment option in the US East (N. Virginia) Region is approximately **$0.58 per hour** or **$425.19 per month**.

> **Note**
>
> The cost for running Workload Discovery on AWS in the AWS Cloud depends on the deployment configuration you choose. The following examples provide cost breakdown for single instance and multiple instances deployment configurations in the US East (N. Virginia) Region. AWS services listed in the example tables below are billed on a monthly basis.

We recommend creating a budget through AWS Cost Explorer to help manage costs. Prices are subject to change. For full details, refer to the pricing webpage for each AWS service used in this solution.

Example cost tables

Option 1: Single instance deployment (default)

When deploying this solution using an AWS CloudFormation template, modifying the **OpensearchMultiAz** parameter to No deploys a single instance for the OpenSearch Service domain, and modifying the **CreateNeptuneReplica** parameter to No deploys a single instance for the Neptune data store. The single instance deployment option incurs a lower cost, but it reduces the availability of Workload Discovery on AWS in the event of an Availability Zone failure.

AWS service	Instance type	Hourly cost [USD]	Monthly cost [USD]
Amazon Neptune	`db.r7g.large`	$0.276	$201.48
Amazon OpenSearch Service	`m6g.large .search`	$0.128	$93.44
AppSync	`cache.small`	$0.044	$32.12
Amazon VPC (NAT Gateway)	N/A	$0.090	$65.70
AWS Config	N/A	$0.003 per resource	$0.003 per resource
Amazon ECS (AWS Fargate Task)	N/A	$0.02	$12.01
Total		**$0.558**	**$404.75**

Option 2: Multiple instances deployment

When deploying this solution using an AWS CloudFormation template, modifying the **OpensearchMultiAz** parameter to `Yes` deploys two instances in two Availability Zones for the OpenSearch Service domain, and modifying the **CreateNeptuneReplica** parameter to `Yes` deploys two instances in two Availability Zones for the Neptune data store. The multiple instances deployment option will cost more to run, but it increases the availability of Workload Discovery on AWS in the event of an Availability Zone failure.

AWS service	Instance type	Hourly cost	Monthly cost [USD]
Amazon Neptune	`db.r7.xlarge`	$0.552	$402.96
Amazon OpenSearch Service	`m6g.large .search`	$0.256	$186.88
AppSync	`cache.small`	$0.044	$32.12

AWS service	Instance type	Hourly cost	Monthly cost [USD]
Amazon VPC (NAT Gateway)	N/A	$0.09	$65.70
AWS Config	N/A	$0.003 per resource	$0.003 per resource
Amazon ECS (AWS Fargate Task)	N/A	$0.02	$12.01
Total		**$0.962**	**$699.677**

- Your final cost depends on the number of resources that AWS Config detects. $0.003 per resource item recorded will be incurred in addition to the amount provided in the table.

> **Important**
>
> The cost for Amazon Neptune and Amazon OpenSearch Service varies, depending on the instance type you select.

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared responsibility model reduces your operational burden because AWS operates, manages, and controls the components including the host operating system, the virtualization layer, and the physical security of the facilities in which the services operate. For more information about AWS security, visit the AWS Security Center.

Resource access

IAM roles

IAM roles allow customers to assign granular access policies and permissions to services and users on the AWS Cloud. Multiple roles are required to run Workload Discovery on AWS and discover resources in AWS accounts.

Amazon Cognito

Amazon Cognito is used to authenticate access with short-lived, strong credentials granting access to components needed by Workload Discovery on AWS.

Network access

Amazon VPC

Workload Discovery on AWS is deployed within an Amazon VPC and configured according to best practices to deliver security and high availability. For additional details, refer to Security best practices for your VPC. VPC endpoints allow non-internet transit between services and are configured where available.

Security groups are used to control and isolate network traffic between the components needed to run Workload Discovery on AWS.

We recommend that you review the security groups and further restrict access as needed once the deployment is up and running.

Amazon CloudFront

This solution deploys a web console UI hosted in an Amazon S3 bucket which is distributed by Amazon CloudFront. By using the origin access identity feature, the contents of this Amazon S3 bucket are accessible only through CloudFront. For more information, refer to Restricting access to an Amazon S3 origin in the *Amazon CloudFront Developer Guide*.

CloudFront activates additional security mitigations to append HTTP security headers to each viewer response. For additional details, refer to Adding or removing HTTP headers in CloudFront responses.

This solution uses the default CloudFront certificate which has a minimum supported security protocol of TLS v1.0. To enforce the use of TLS v1.2 or TLS v1.3, you must use a custom SSL certificate instead of the default CloudFront certificate. For more information, refer to How do I configure my CloudFront distribution to use an SSL/TLS certificate.

Application configuration

AWS AppSync

Workload Discovery on AWS GraphQL APIs have request validation provided by AWS AppSync according to the GraphQL specification. Furthermore, authentication and authorization are implemented using IAM and Amazon Cognito, which use the JWT provided by Amazon Cognito when a user authenticates successfully in the web UI.

AWS Lambda

By default, the Lambda functions are configured with the most recent stable version of the language runtime. No sensitive data or secrets are logged. Service interactions are carried out with the least required privilege. Roles that define these privileges are not shared between functions.

Amazon OpenSearch Service

Amazon OpenSearch Service domains are configured with an access policy that restricts access to stop any unsigned requests made to the OpenSearch Service cluster. This is restricted to a single Lambda function.

The OpenSearch Service cluster is built with node-to-node encryption activated to add an extra layer of data protection on top of the existing OpenSearch Service security features.

Log Retention

This solution captures application and service logs by creating CloudWatch logs groups in your account. By default, logs are kept for 1 year. You can adjust the LogRetentionPeriod parameter for each log group, keeping the default retention period, or choosing a period between one day and 10 years based on your requirements.

Quotas

Service quotas, also referred to as limits, are the maximum number of service resources or operations for your AWS account.

Quotas for AWS services in this solution

Make sure you have sufficient quota for each of the services implemented in this solution. For more information, see AWS service quotas.

Use the following links to go to the page for that service. To view the service quotas for all AWS services in the documentation without switching pages, view the information in the Service endpoints and quotas page in the PDF instead.

Amplify	Amazon ECR
Athena	Lambda
CloudFront	OpenSearch Service
Cognito	Neptune
Config	Amazon S3
Amazon ECS	

AWS CloudFormation quotas

Your AWS account has AWS CloudFormation quotas that you should be aware of when launching the stack in this solution. By understanding these quotas, you can avoid limitation errors that would prevent you from deploying this solution successfully. For more information, see AWS CloudFormation quotas in the in the *AWS CloudFormation User's Guide*.

AWS Lambda quotas

Your account has an AWS Lambda concurrent execution quota of 1000. If the solution is used in an account where there are other workloads running and using Lambda, then set this quota to an appropriate value. This value is adjustable; for more information, see AWS Lambda quotas in the *AWS Lambda User's Guide*.

Note

This solution requires 150 executions from the concurrent execution quota to be available in the account to which the solution is being deployed. If there are fewer than 150 executions available in that account, the CloudFormation deployment will fail.

Amazon VPC quotas

Your AWS account can contain five VPCs and two Elastic IPs (EIPs). If the solution is used in an account with other VPCs or EIPs, this could prevent you from deploying this solution successfully. If you are at risk of reaching this quota, you may provide your own VPC for deployment by providing it when following the steps in the Launch the Stack section. For more information, see Amazon VPC quotas in the *Amazon VPC User's Guide*.

Choosing the deployment account

If you are deploying Workload Discovery on AWS to an AWS Organization, the solution must be installed in a delegated admin account where StackSets and multi-Region AWS Config capabilities have been enabled.

If you are not using AWS Organizations, we recommend that you deploy Workload Discovery on AWS into a dedicated AWS account created specifically for this solution. This approach means Workload Discovery on AWS is isolated from your existing workloads and provides a single location for configuring the solution, such as adding users and importing new Regions. It is also easier to track the costs incurred while running the solution.

After Workload Discovery on AWS is deployed, you can then import Regions from any accounts you already provisioned.

Deploy the solution

This solution uses AWS CloudFormation templates and stacks to automate its deployment. The CloudFormation template specifies the AWS resources included in this solution and their properties. The CloudFormation stack provisions the resources that are described in the template.

Deployment process overview

> **Note**
>
> If you previously deployed Workload Discovery on AWS and would like to upgrade to the latest version, refer to Update the solution.

Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Time to deploy: Approximately 30 minutes

Before you launch the solution, review the cost, architecture, network security, and other considerations discussed in this guide.

> **Important**
>
> This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. AWS owns the data gathered though this survey. Data collection is subject to the AWS Privacy Notice.

Prerequisites

Gather deployment parameter details

Before deploying Workload Discovery on AWS, review your configuration details for the Amazon OpenSearch Service service-linked role and AWS Config.

Verify whether you have an AWSServiceRoleForAmazonOpenSearchService role

The deployment creates an Amazon OpenSearch Service cluster inside an Amazon Virtual Private Cloud (Amazon VPC). The template uses a service-linked role to create the OpenSearch Service cluster. However, if you already have the role created in your account, use the existing role.

To check if you already have this role:

1. Sign in to the Identity and Access Management (IAM) console for the account you plan to deploy this solution to.
2. In the **Search** box, enter `AWSServiceRoleForAmazonOpenSearchService`.
3. If your search returns a role, select `No` for the **CreateOpensearchServiceRole** parameter when you launch the stack.

Verify AWS Config is set up

Workload Discovery on AWS uses AWS Config to gather the majority of resource configurations. When deploying the solution or importing a new Region, you must confirm whether AWS Config is already set up and working as expected. The **AlreadyHaveConfigSetup** CloudFormation parameter informs Workload Discovery on AWS of whether to set up AWS Config.

The following snippet is taken from the AWS CLI Command Reference. Run the command in the Region you intend to deploy Workload Discovery on AWS or import into Workload Discovery on AWS.

Enter the following command:

```
aws configservice get-status
```

If you receive a response similar to the output, then there is a Configuration Recorder and Delivery Channel running in that Region. Select `Yes` for the **AlreadyHaveConfigSetup** CloudFormation parameter.

Output:

```
Configuration Recorders:

name: default
recorder: ON
last status: SUCCESS
```

```
Delivery Channels:

name: default
last stream delivery status: SUCCESS
last history delivery status: SUCCESS
last snapshot delivery status: SUCCESS
```

If you are configuring AWS CloudFormation StackSets, then you must include this Region in the batch of Regions that already have AWS Config configured.

Verify your AWS Config details in your account

The deployment will attempt to set up AWS Config. If you already use AWS Config in the account that you plan to either deploy to or make discoverable by Workload Discovery on AWS, select the relevant parameters when you deploy this solution. Furthermore, for successful deployment, ensure that you haven't restricted the resources that AWS Config scans.

To check your current AWS Config configuration:

1. Sign in to the AWS Config console.
2. Choose **Settings** and ensure the **Record all resources supported in this Region** and **Include global resources** boxes are selected.

Verify AWS Config aggregator type

If supplying an existing AWS Config aggregator (only supported in `AWS_ORGANIZATIONS` mode), ensure that the aggregator is an AWS Organization wide aggregator. Run the following command and verify the presence of the `OrganizationAggregationSource` field:

```
aws configservice describe-configuration-aggregators
```

Output:

```
{
    "ConfigurationAggregators": [
        {
            "ConfigurationAggregatorName": "aggregator-name",
            "ConfigurationAggregatorArn": "arn:aws:config:eu-
west-1:123456789012:config-aggregator/config-aggregator-5jfoefab",
```

```
            "OrganizationAggregationSource": {
                "RoleArn": "arn:aws:iam::123456789012:role/service-role/
ConfigAggregator-ConfigOrganizationsRole",
                "AllAwsRegions": true
            },
            "CreationTime": "2024-11-19T15:13:22.979000+00:00",
            "LastUpdatedTime": "2024-11-19T15:13:22.990000+00:00"
        }
    ]
}
```

Verify your VPC configuration

If deploying to an existing VPC, verify your private subnets can route requests to AWS services.

If you choose the option to deploy the solution in an existing VPC, you must ensure that the Workload Discovery on AWS Lambda functions and the Amazon ECS tasks running in the private subnets of your VPC can connect to other AWS services. The standard way to enable this is with NAT gateways. You can list the NAT gateways in your account as shown in the following code sample.

```
aws ec2 describe-route-tables --filters Name=association.subnet-id,Values=<private-
subnet-id1>,<private-subnet-id2> --query 'RouteTables[].Routes[].NatGatewayId'
```

Output:

```
[
    "nat-1111111111111111",
    "nat-2222222222222222"
]
```

> **Note**
>
> If less than two results return, the subnets do not have the correct number of NAT gateways.

If your VPC doesn't have NAT gateways, then you must either provision them or ensure that you have VPC endpoints for all the AWS services listed in the AWS APIs section.

AWS CloudFormation Template

This solution uses AWS CloudFormation to automate the deployment of Workload Discovery on AWS in the AWS Cloud. It includes the following CloudFormation template, which you can download before deployment:

workload-discovery-on-aws.template - Use this template to launch the solution and all associated components. The default configuration deploys the core and supporting solutions found in the AWS services in this solution section, but you can customize the template to meet your specific needs.

> **Note**
>
> You can customize the template to meet your specific needs; however, any changes you make could affect the upgrade process.

Launch the stack

This automated AWS CloudFormation template deploys Workload Discovery on AWS in the AWS Cloud. You must gather deployment parameter details before launching the stack. For details, refer to Prerequisites.

Time to deploy: Approximately 30 minutes

1. Sign in to the AWS Management Console and select the button to launch the `workload-discovery-on-aws.template` AWS CloudFormation template.

2. The template launches in the US East (N. Virginia) Region by default. To launch the solution in a different AWS Region, use the Region selector in the console navigation bar.

> **Note**
>
> This solution uses services that are not available in all AWS Regions. Refer to Supported AWS Regions for a list of supported AWS Regions.

3. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box, and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, refer to IAM and AWS STS quotas in the *AWS Identity and Access Management User Guide*.
5. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
AdminUserEmailAddress	*<Requires input>*	An email address to create the first user. The temporary credentials will be sent to this email address.
AlreadyHaveConfigSetup	No	Confirmation of whether or not you already have AWS Config set up in the deployment account. For details, refer to Prerequisites.
AthenaWorkgroup	`primary`	The workgroup that will be used to issue the Athena query when the Cost feature is enabled.
ApiAllowListedRanges	`0.0.0.0/1,128.0.0.0/1`	Comma separated list of CIDRs to manage access to the AppSync GraphQL API. To allow the entire internet, use 0.0.0.0/1,128.0.0.0/1. If

Parameter	Default	Description
		restricting access to specific CIDRs, you must also include the IP addresses (and a subnet mask of /32) of the NAT gateways that allow the discovery process ECS task running in its private subnet to access the internet. NOTE: This allow list does not govern access to the WebUI, only the GraphQL API.
CreateNeptuneReplica	`No`	Choose whether to create a read replica for Neptune in a separate Availability Zone. Choosing `Yes` improves resilience but increases the cost of this solution.
CreateOpenSearchServiceRole	`Yes`	Confirmation of whether or not you already have a service-linked role for Amazon OpenSearch Service. For details, refer to Prerequisites.
NeptuneInstanceClass	`db.r5.large`	The instance type used to host the Amazon Neptune database. What you select here affects the cost of running this solution.

Parameter	Default	Description
OpensearchInstanceType	`m6g.large.search`	The instance type used for your OpenSearch Service data nodes. Your selection affects the cost of running the solution.
OpensearchMultiAz	`No`	Choose whether to create an OpenSearch Service cluster that spans multiple Availability Zones. Choosing `Yes` improves resilience but increases the cost of this solution.
CrossAccountDiscovery	`SELF_MANAGED`	Choose whether Workload Discovery on AWS or AWS Organizations manages the importing of accounts. The value can be `SELF_MANAGED` or `AWS_ORGANIZATIONS`.
OrganizationUnitId	*<Optional input>*	The root organization unit ID. This parameter is only used when **CrossAccountDiscovery** is set to `AWS_ORGANIZATIONS`.

Parameter	Default	Description
AccountType	`DELEGATED_ADMIN`	The type of AWS Organizations account to install Workload Discovery on AWS in. This parameter is only used when **CrossAccountDiscovery** is set to `AWS_ORGANIZATIONS` . For details, refer to Choosing the deployment account.
ConfigAggregatorName	*<Optional input>*	The AWS Organization-wide Config aggregator to use. You must install the solution in the same account and Region as this aggregator. If you leave this parameter blank, a new aggregator will be created. This parameter is only used when **CrossAccountDiscovery** is set to `AWS;_ORGANIZATIONS` .
CpuUnits	`1 vCPU`	The number of CPUs to allocate for the Fargate task that the discovery process runs in.
Memory	`2048`	The amount of memory to allocate for the Fargate task that the discovery process runs in.
DiscoveryTaskFrequency	`15mins`	The time interval between every run of the discovery process ECS task.

Parameter	Default	Description
MinNCUs	1	Minimum Neptune Capacity Units (NCUs) to be set on the Neptune cluster (must be less than or equal to **MaxNCUs**). Required if DBInstance type is `db.serverless` .
MaxNCUs	128	Maximum NCUs to be set on the Neptune cluster (must be greater than or equal to **MinNCUs**). Required if DBInstance type is `db.serverless` .
VpcId	*<Optional input>*	The ID of an existing VPC for the solution to use. If you leave this parameter blank, a new VPC will be provisioned.
VpcCidrBlock	*<Optional input>*	The VPC CIDR block of the VPC referenced by the **VpcId** parameter. This parameter is only used if the **VpcId** parameter is set.
PrivateSubnet0	*<Optional input>*	The private subnet you wish to use. This parameter is only used if the **VpcId** parameter is set.
PrivateSubnet1	*<Optional input>*	The private subnet you wish to use. This parameter is only used if the **VpcId** parameter is set.

Parameter	Default	Description
UsesCustomIdentity	No	Confirmation of whether on not you will be using a custom identity provider, such as SAML or OIDC.
CognitoCustomDomain	*<Optional input>*	The domain prefix for the Amazon Cognito custom domain that hosts the sign-up and sign-in pages for your application. Leave empty if you are not using a custom IdP, otherwise must include only lowercase letters, numbers, and hyphens.
CognitoAttributeMapping	*<Optional input>*	The mapping of IdP attribute s to standard and custom Cognito user pool attribute s. Leave empty if you are not using a custom IdP, otherwise must be a valid JSON string.
IdentityType	*<Optional input>*	The type of Identity Provider to use (`Google`, `SAML`, or `OIDC`). Leave empty if you are not using a custom IdP.
ProviderName	*<Optional input>*	Name for the Identity Provider. Leave empty if you are not using a custom IdP.
GoogleClientId	*<Optional input>*	The Google Client ID to use. Parameter only used when **IdentityType** is set to `Google`.

Parameter	Default	Description
GoogleClientSecret	*<Optional input>*	The Google client secret to use. Parameter only used when **IdentityType** is set to `Google`.
SAMLMetadataURL	*<Optional input>*	The metadata URL for the SAML Identity Provider. Parameter only used when **IdentityType** is set to SAML.
OIDCClientId	*<Optional input>*	The OIDC client ID to use. Parameter only used when **IdentityType** is set to `OIDC`.
OIDCClientSecret	*<Optional input>*	The OIDC client secret to use. Parameter only used when **IdentityType** is set to `OIDC`.
OIDCIssuerURL	*<Optional input>*	The OIDC issuer URL to use. Parameter only used when **IdentityType** is set to `OIDC`.
OIDCAttributeRequestMethod	`GET`	The OIDC attribute request method to use. Must be either `GET` or `POST` (refer to OIDC provider or use default value). Parameter only used when **IdentityType** is set to `OIDC`.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review and create** page, review and confirm the settings. Select the boxes acknowledging that the template creates IAM resources and require certain capabilities.
9. Choose **Submit** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation Console in the Status column. You should receive a **CREATE_COMPLETE** status in approximately 30 minutes.

> **Note**
>
> If deleted, this stack removes all resources. If the stack is updated, it retains the Amazon Cognito user pool to ensure that configured users aren't lost.

Post-deployment configuration tasks

After Workload Discovery on AWS has successfully deployed, complete the following post-deployment configuration tasks.

Turn on advanced security in Amazon Cognito

To turn on the advanced security features for Amazon Cognito, follow the instructions on Adding advanced security to a user pool in the *Amazon Cognito Developer Guide*.

> **Note**
>
> There is an additional cost for activating advanced security in Amazon Cognito.

Configure Cognito

Workload Discovery on AWS uses Amazon Cognito to manage authentication. You can create users directly in the Cognito user pool or you can use a third-party IdP using SAML or OIDC.

Manage users via Cognito user pool

On deployment, the solution creates a user for you and sends an email to the address provided in the `AdminUserEmailAddress` CloudFormation parameter with temporary credentials. To add more users follow these steps.

Create additional users:

1. Sign in to the AWS Cognito console.
2. Choose **Manage User Pools**.
3. Choose **WDCognitoUserPool-** *<ID-string>*.
4. In the navigation pane, under **General Settings**, choose **Users and groups**.
5. On the **Users** tab, choose **Create user**.
6. On the **Create user** box, enter values for all required fields.

Form Field	Required?	Description
Username	Yes	The username that you will use to log in to Workload Discovery on AWS.
Send an invitation	Yes (email only)	When selected, sends a notification as a reminder of the temporary password. Select **Email** only. If you select **SMS (default),** an error message displays, but the user is still created.
Temporary Password	Yes	Enter a temporary password. The user is forced to change this when they sign in to Workload Discovery on AWS for the first time.
Phone Number	No	Enter a phone number in international format, for example, \+44. Ensure that the **Mark phone number as verified?** box is selected.
Email	Yes	Enter a valid email address. Ensure that the **Mark email as verified?** box is selected.

7. Choose **Create user**.

Repeat this process to create as many users as you need.

> **Note**
>
> Every user will have the same level of access to resources discovered. We recommend provisioning a separate deployment of Workload Discovery on AWS for accounts that contain sensitive workloads or data. This allows you to restrict access to only the users that need it.

Manage users via third-party IdP

You can set up user sign-in with an OIDC IdP or a SAML IdP.

Set up user sign-in with an OIDC IdP

1. Set up an OIDC client application in your IdP according to your provider's documentation. You will require the following values:

Field Name	Value	Description
Redirect URI	`https://<cognito-hostname>.auth.<wd-region>.amazoncognito.com/oauth2/idpresponse`	`<cognito-hostname>` can be any value and will be used later

2. The OIDC IdP will provide you with a OIDC discovery URL, client ID and a client secret. Note these values.
3. Sign in to the AWS CloudFormation console.
4. Select the main Workload Discovery on AWS stack and choose **Update**.
5. On the **Update stack** page, select **Use existing template**.
6. Update the following CloudFormation parameters:

Field Name	Value	Description
UsesCustomIdentity	**Yes**	Confirmation of whether on not you will be using a

Field Name	Value	Description
		custom identity provider, such as SAML or OIDC.
CognitoCustomDomain	*<cognito-hostname>*	The domain prefix for the Amazon Cognito custom domain that hosts the sign-up and sign-in pages for your application. This **must** match the hostname (not the full URL) in the **Redirect URI** value from step 1.
IdentityType	`OIDC`	The type of Identity Provider to use
ProviderName	*<any-value>*	Name for the Identity Provider.
OIDCIssuerURL	*<oidc-discovery-url>*	The OIDC discovery URL noted in step 2.
OIDCClientId	*<client-id>*	The client ID noted in step 2.
OIDCClientSecret	*<client-secret>*	The OIDC discovery URL noted in step 2.
CognitoAttributeMapping	*<valid-json-value>*	The mapping of IdP attributes to standard and custom Cognito user pool attributes such as `email`.
OIDCAttributeRequestMethod	`GET` or `POST`	The OIDC attribute request method to use.

7. Choose Next.
8. On the **Review** page, review and confirm the settings. Select the boxes acknowledging that the template creates IAM resources and requires certain capabilities.

9. Choose **Update stack** to deploy the stack.

Set up user sign-in with a SAML IdP

1. Sign in to the AWS CloudFormation console.
2. Choose **View nested** to display the nested stacks that make up the deployment. Depending on your preferences, nested stacks might already be displayed.
3. Select the **Workload Discovery on AWS Amazon Cognito** stack. It will be named *<wd-stack-name>* -CognitoStack- *<ID-string>*.
4. Select the **Outputs** tab and note the ID in the **Value** column associated with the **UserPoolId** key.
5. Configure your SAML IdP to accept requests and send responses to your user pool. The documentation for your SAML IdP will contain information about how to add your user pool as a relying party or application for your SAML 2.0 IdP. You will require the following values:

Field Name	Value	Description
SP entity ID	urn:amazon:cognito:sp: *<UserPoolId>*	The URN for the Cognito userpool
ACS URL	https:// *<cognito-hostname>* .auth.<wd-region>.amazoncognito.com/saml2/idpresponse	<cognito-hostname> can be any value and will be used later

6. Download SAML metadata from your IdP, or retrieve the URL to your metadata endpoint.
7. Return to the CloudFormation console.
8. Select the main Workload Discovery on AWS stack and choose **Update**.
9. On the **Update stack** page, select **Use existing template**.
10. Update the following CloudFormation parameters:

Field Name	Value	Description
UsesCustomIdentity	**Yes**	Confirmation of whether or not you will be using a

Field Name	Value	Description
		custom identity provider, such as SAML or OIDC.
CognitoCustomDomain	*<cognito-domain-value>*	The domain prefix for the Amazon Cognito custom domain that hosts the sign-up and sign-in pages for your application. This **must** match the hostname (not the full URL) in the **ACS URL** value from step 5.
IdentityType	SAML	The type of Identity Provider to use
ProviderName	*<any-value>*	Name for the Identity Provider.
SAMLMetadataURL	*<saml-metadata-url>*	The SAML metadata URL retrieved from step 6.
CognitoAttributeMapping	*<valid-json-value>*	The mapping of IdP attributes to standard and custom Cognito user pool attributes such as `email`.

11. Choose Next.
12. On the **Review** page, review and confirm the settings. Select the boxes acknowledging that the template creates IAM resources and requires certain capabilities.
13. Choose **Update stack** to deploy the stack.

Log in to Workload Discovery on AWS

After the solution successfully deploys, determine the URL for the Amazon CloudFront distribution that serves the solution's web UI.

1. Sign in to the AWS CloudFormation console.
2. Choose **View nested** to display the nested stacks that make up the deployment. Depending on your preferences, nested stacks might already be displayed.
3. Select the main Workload Discovery on AWS stack.
4. Select the **Outputs** tab and choose the URL in the **Value** column associated with the **WebUiUrl** key.
5. On the **Sign in to** screen, enter the sign-in credentials that you received via email. Then take the following actions:
   a. Follow the prompts to change your password.
   b. Use the verification code sent to your email to complete account recovery.

Import a Region

> **Note**
>
> The following section only applies when the solution's account discovery mode is self-managed. For information on how account discovery works in AWS Organizations mode, see the AWS Organizations Account Discovery Mode section.

Importing a Region requires certain infrastructure to be deployed. This infrastructure consists of *Global* and *Regional* resources:

Global – Resources that are deployed once in an account and reused for each Region imported.

- An IAM role (WorkloadDiscoveryRole)

Regional – Resources that are deployed in each Region imported.

- An AWS Config Delivery Channel
- An Amazon S3 bucket for AWS Config
- An IAM role (ConfigRole)

There are two options to deploy this infrastructure:

- AWS CloudFormation StackSets (recommended)
- AWS CloudFormation

Import a Region

These steps guide you through importing a Region and deploying the AWS CloudFormation templates.

1. Sign in to Workload Discovery on AWS. Refer to Log in to Workload Discovery on AWS for the URL.
2. In the navigation menu, select **Accounts**.
3. Choose **Import**.
4. Select the import method:
   a. **Add Accounts & Regions** using a CSV file.
   b. **Add Accounts & Regions** using a form.

CSV file

Provide a Comma Separated Value (CSV) file that contains the Regions to be imported in the following format.

```
"accountId","accountName","region"
123456789012,"test-account-1",eu-west-2
123456789013,"test-account-2",eu-west-1
123456789013,"test-account-2",eu-west-2
123456789014,"test-account-3",eu-west-3
```

1. Select **Upload a CSV**.
2. Locate and open your CSV file.
3. Review the **Regions** table, then select **Import**.
4. In the modal dialog, download the Global resources template and Regional Resources template.
5. Deploy the CloudFormation templates in the relevant accounts (refer to Deploy the AWS CloudFormation templates section).
6. Once the global and regional resource templates have been deployed, select both boxes to confirm that the installation is complete and choose **Import**.

Form

Provide the Regions to import using the form:

1. For **Account ID**, enter a 12-digit account ID or select an existing account ID.
2. For **Account name**, enter an account name or use a pre-populated value when selecting an existing account ID.
3. Select the Regions to import.
4. Select **Add** to populate the Regions in the **Regions** table below.
5. Review the **Regions** table, then select **Import**.
6. In the modal dialog, download the Global resources template and Regional Resources template.
7. Deploy the CloudFormation templates in the relevant accounts (refer to Deploy the AWS CloudFormation templates section).
8. Once the global and regional resource templates have been deployed, select both boxes to confirm the installation is complete and choose **Import**.

Deploy the AWS CloudFormation templates

Global resources must be deployed once per account. Do not deploy this template when importing a Region from an account that contains a Region that is already imported into Workload Discovery on AWS. If the Region has already been imported, follow the instructions in Deploy the stack to provision the Regional resources.

Use CloudFormation StackSets to provision Global resources across accounts

> **Important**
>
> First, complete the Prerequisites for stack set operations to activate StackSets in your target accounts.

1. In the administrator account, sign in to the AWS CloudFormation console.
2. From the navigation menu, select **StackSets**.
3. Choose **Create StackSet**.

4. On the **Choose a template** page, under **Permissions**:
   a. If you're using AWS Organizations, choose either **Service managed permissions** or **Self service permissions**. For details, refer to Using StackSets in an AWS Organization.
   b. If you're not using AWS Organizations, enter the IAM run role name used when following the StackSets prerequisite steps. For details, refer to Grant self-managed permissions.
5. Under **Specify template**, select **Upload a template file**. Choose the `global-resources.template` file (downloaded earlier when you imported a Region either by CSV file or form), and choose **Next**.
6. On the **Specify StackSet details** page, assign a name to your StackSet. For information about naming character limitations, refer to IAM and AWS STS quotas in the *AWS Identity and Access Management User Guide*.
7. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

Field Name	Default	Description
AccountId	*The deployment account ID*	The account ID of the original deployment account. You must leave this value as the default.

8. Choose **Next**.
9. On the **Configure StackSet options** page, choose **Next**.
10. On the **Set deployment options** page, under **Accounts**, enter the account IDs for deploying the account role in the **Account numbers** box.
11. Under **Specify regions**, select a **Region** to install the stack.
12. Under **Deployment options**, select **Parallel**, and then choose **Next**.
13. On the **Review** page, check the box acknowledging that AWS CloudFormation might create IAM resources with custom names.
14. Choose **Submit**.

Use CloudFormation StackSets to provision Regional resources

Important

First, complete the Prerequisites for stack set operations to activate StackSets in your target accounts.
If you have some Regions with AWS Config installed and some without, you must perform two StackSet operations, one for the Regions with AWS Config installed and one for those without.

1. In the administrator account, sign in to the AWS CloudFormation console.
2. From the navigation menu, select **StackSets**.
3. Choose **Create StackSet**.
4. On the Choose a template page, under **Permissions**:
   a. If you're using AWS Organizations, choose either **Service managed permissions** or **Self service permissions**. For details, refer to Using StackSets in an AWS Organization.
   b. If you're not using AWS Organizations, enter the IAM run role name used when following the StackSets prerequisite steps. For details, refer to Grant self-managed permissions.
5. Under **Specify template**, select **Upload a template file**. Choose the `regional-resources.template` file (downloaded earlier when you imported a Region either by CSV file or form), and choose **Next**.
6. On the **Specify StackSet details** page, assign a name to your StackSet. For information about naming character limitations, refer to IAM and AWS STS quotas in the *AWS Identity and Access Management User Guide*.
7. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

Field Name	Default	Description
AccountId	*The deployment account ID*	The account ID of the original deployment account. You must leave this value as the default.

Field Name	Default	Description
AggregationRegion	*The deployment Region*	The Region that was originall y deployed into. You must leave this value as the default.
AlreadyHaveConfigSetup	No	Confirmation of whether the Region already has AWS Config installed. Set to Yes if AWS Config is already installed in this Region.

8. Choose **Next**.
9. On the **Configure StackSet options** page, choose **Next**.
10. On the **Set deployment options** page, under **Accounts**, enter the account IDs to deploy the account role to in the **Account numbers** box.
11. Under **Specify regions**, select a **Region** to install the stack. This installs the stack in these Regions in all the accounts entered in step 6.
12. Under **Deployment options**, select **Parallel**, and then choose **Next**.
13. On the **Review** page, check the box acknowledging that AWS CloudFormation might create IAM resources with custom names.
14. Choose **Submit**.

Deploy the stack to provision the Global resources using CloudFormation

Global resources must be deployed once per account. Do not deploy this template when importing a Region from an account that contains a Region that is already imported into Workload Discovery on AWS.

1. Sign in to the AWS CloudFormation console.
2. Choose **Create stack**, and then select **With new resources (standard)**.
3. On the **Create stack** page, in the **Specify template** section, select **Upload a template file**.

4. Choose **Choose file** and select the `global-resources.template` file that (downloaded earlier when you imported a Region either by CSV file or form), and choose **Next**.
5. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, refer to IAM and AWS STS quotas in the _AWS Identity and Access Management_*User Guide*.
6. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

Field Name	Default	Description
Stack name	`workload-discovery`	The name of this AWS CloudFormation stack.
AccountId	*Deployment account ID*	The account ID of the original deployment account. You must leave this value as the default.

7. Choose **Next**.
8. Select the box acknowledging that AWS CloudFormation might create IAM resources with custom names.
9. Choose **Create stack**.

The new Regions will be scanned during the next discovery process, which runs at 15-minute intervals, for example: 15:00, 15:15, 15:30, 15:45.

Deploy the stack to provision the Regional resources using CloudFormation

1. Sign in to the AWS CloudFormation console.
2. Choose **Create stack**, and then select **With new resources (standard)**.
3. On the **Create stack** page, in the **Specify template** section, select **Upload a template file**.
4. Choose **Choose file** and select the `regional-resources.template` file (downloaded earlier when you imported a Region either by CSV file or form), and choose **Next**.

5. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, refer to IAM and AWS STS quotas in the *AWS Identity and Access Management User Guide*.
6. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

Field Name	Default	Description
AccountId	*Solution deployment account ID*	The account ID of the original deployment account. Must be left as default.
AggregationRegion	*Solution deployment Region*	The Region that was originall y deployed into. Must be left as default.
AlreadyHaveConfigSetup	No	Confirmation of whether the Region already has AWS Config installed. Set to `Yes` if AWS Config is already installed in this Region.

7. Choose **Next**.
8. Select the box acknowledging that AWS CloudFormation might create IAM resources with custom names.
9. Choose **Create stack**.

The new Regions will be scanned during the next discovery process, which runs at 15-minute intervals, for example, 15:00, 15:15, 15:30, 15:45.

Verify the Region was imported correctly

1. Sign in to the solution's web UI (or refresh the page if it's already loaded). Refer to Log in to Workload Discovery on AWS for the URL.
2. From the left navigation panel, under **Settings**, select **Imported Regions**.

The Region, account name, and account ID appear in the table. The **Last Scanned** column shows the last discovered resources in that Region.

> **Note**
>
> If the **Last Scanned** column stays blank for more than 30 minutes, refer to Debugging the discovery component.

Set up the cost feature

The cost feature requires manual set up of AWS Cost and Usage reports (CUR). Following the instructions below you will:

1. Set up a scheduled CUR.
2. Set up Amazon S3 replication (when CURs are outside the deployment account)

Create the AWS Cost and Usage Report in the deployment account

1. Sign in to the Billing and Cost Management console of the account from which you would like to gather cost data.
2. In the navigation menu, under **Legacy Pages**, select **Cost and Usage Reports**.
3. Choose **Create Report.**
4. Use `workload-discovery-cost-and-usage-` *`<your-workload-discovery-deployment-account-ID>`* as the **Report name.**

   > **Note**
   >
   > You must follow this naming convention because a small amount of infrastructure will be deployed to facilitate the querying of the CURs.

5. Select the **Include resource IDs** box.

> **Note**
>
> You must select the **Include resource IDs** box to view cost data. This ID must match with the resources discovered by Workload Discovery on AWS.

6. Choose **Next**.
7. On the Delivery options page, choose **Configure** *0*
8. Select the *<stack-name>* `-s3buc-costandusagereportbucket-` *<ID-string>* Amazon S3 bucket to store the CUR. Choose **Next**.
9. Review the policy, select the confirmation box, and choose **Save**.
10. Set the **Report prefix path** to `aws-perspective`.
11. Select **Daily** for the time granularity.
12. Under **Enable report data integration for,** select **Amazon Athena**.
13. Choose **Next**.
14. Choose **Review and Complete**.

To verify that the report is correctly set up, check the Amazon S3 bucket for the test file.

> **Note**
>
> It can take up to 24 hours for the reports to be uploaded to your bucket.

Create the AWS Cost and Usage Report in an external account

1. Sign in to the Billing and Cost Management console of the account from which you would like to gather cost data.
2. In the navigation menu, under **Legacy Pages**, select **Cost and Usage Reports**.
3. Choose **Create Report.**
4. Use `workload-discovery-cost-and-usage-` *<your-external-account-ID>* as the **Report name.**

> **Note**
>
> You must follow this naming convention because a small amount of infrastructure will be deployed to facilitate the querying of the CURs.

5. Check the **Include resource IDs** box.

> **Note**
>
> You must select the **Include resource IDs** box to view cost data. This ID is needed to match with the resources discovered by Workload Discovery on AWS.

6. Choose **Next**.
7. On the **Delivery options** page, choose **Configure** *0*
8. Create a new Amazon S3 bucket to store the CURs.
9. Review the policy, select the confirmation box, and choose **Save**.
10. Set the **Report prefix path** to `aws-perspective`.
11. Select **Daily** for the time granularity.
12. Under **Enable report data integration for**, select **Amazon Athena**.
13. Choose **Next**.
14. Choose **Review and Complete**. To verify that the report is correctly set up, check the Amazon S3 bucket for the test file.

> **Note**
>
> It can take up to 24 hours for the reports to be uploaded to your bucket.

Next, set up replication to the deployment account.

Set up replication

Set up replication into the Amazon S3 bucket created during deployment. The Amazon S3 bucket follows the following format: *<stack-name>* `-s3buc-costandusagereportbucket-` *<ID-string>*. This allows the solution to query the bucket with Amazon Athena.

1. Sign in to the AWS account in Amazon S3 console that contains the created CUR that needs to be replicated.
2. Select the Amazon S3 bucket created when configuring your CUR. For more information, look at Step 8 of *Create and schedule the AWS Cost and Usage Report*.
3. Choose the **Management** tab.
4. Under **Replication rules**, choose **Create replication rule**.
5. Under **Replication rule configuration**, in the **Replication rule name** box, enter a descriptive rule ID.
6. Under **Source bucket**, select **Apply to all objects in the bucket** to configure the rule scope.
7. Under **Destination**, configure the following:
   a. Select **Specify a bucket in another account**.
   b. Enter the account ID.
   c. Enter a value for the **Bucket name** that was created during deployment of Workload Discovery on AWS. You can find this by following the instructions in Locating deployment resources, using the logical ID `CostAndUsageReportBucket` and the stack name you specified when first deploying Workload Discovery on AWS.
   d. Select the box for **Change object ownership to destination bucket owner**.
8. Under **IAM role**, choose **Create new role**.

> **Note**
>
> A replication role might already exist. You can select it and ensure that it has the required S3 replication role actions.

9. Choose **Save**.
10. Sign in to the AWS Management Console where the CUR is installed, navigate to the **S3** service page and select the `CostAndUsageReportBucket` S3 bucket. For details, refer to Locating deployment resources.
11. Select the **Management** tab.
12. Under **Replication rules**, from the **Actions** drop-down menu, select **Receive replicated objects**.
13. Under **Source bucket account settings**:
    a. Enter the source bucket account ID.
    b. Choose **Generate policies**.

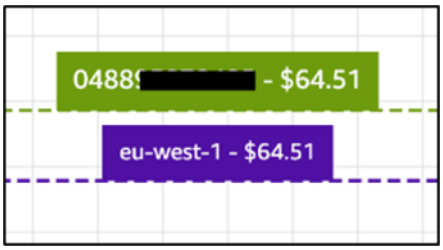
c. Under **Policies,** select **view bucket policy**.

d. Select **Include permission to change object ownership to destination bucket owner**.

e. Choose **Apply settings**. This gives it access to copy objects to it. Refer to Cost Bucket replication policy for an example S3 bucket policy.

> **Note**
>
> When replicating CURs from multiple AWS accounts. You need to ensure the bucket policy on the destination bucket (within the Workload Discovery on AWS account) has the ARN of each IAM Role you are using from each account. Refer to Cost Bucket replication policy for more details.

When the reports are in the account, cost data appears on the bounding boxes and individual resources.



Edit S3 bucket lifecycle policies

During deployment, the solution configures lifecycle policies on two buckets:

- `CostAndUsageReportBucket`
- `AccessLogsBucket`

> **Important**
>
> These lifecycle policies delete data from these buckets after 90 days. You can edit the lifecycle to fit any internal policies you have.

Monitoring the solution

This solution uses myApplications and CloudWatch AppInsights to allow you to monitor your Workload Discovery on AWS deployment.

myApplications

myApplications is an extension of Console Home that helps you manage and monitor the cost, health, security posture, and performance of your applications on AWS. You can access all applications in your account, key metrics across all applications, and an overview of cost, security, and operations metrics and insights from multiple service consoles from one view in the AWS Management Console.

To view the myApplications dashboard for Workload Discovery on AWS:

1. Sign in to the AWS Management Console.
2. In the left sidebar, choose **myApplications**.
3. Type `workload-discovery` into searchbar to find the application.
4. Select the application.

CloudWatch AppInsights

CloudWatch Application Insights helps you monitor your applications by identifying and setting up key metrics, logs, and alarms across your application resources and technology stack.It continuously monitors metrics and logs to detect and correlate anomalies and errors. To assist with troubleshooting, it creates automated dashboards for detected problems, which include correlated metric anomalies and log errors, along with additional insights to point you to a potential root cause.

To view the CloudWatch AppInsights dashboard for Workload Discovery on AWS:

1. Sign in to the CloudWatch console.
2. In the left sidebar, choose **Insights, Application Insights**.
3. Select the **Applications** tab.
4. Type `workload-discovery` into searchbar to find the dashboard.

5. Select the dashboard.
6. Select the application.

Update the solution

Important

Updating from v1.x.x to v2.x.x of Workload Discovery on AWS is not supported. We recommend that you to uninstall v1.x.x of this solution before installing v2.x.x.

To update from a 2.x.x deployment, follow these steps.

1. Download the solution's AWS CloudFormation template.
2. Sign in to the AWS CloudFormation console.
3. Select the stack with the name provided during deployment and choose **Update**.
4. On the **Update stack** page, select **Replace current template**, then select **Upload a template file**, and upload the file downloaded in step 1.
5. Choose **Next**.
6. On the **Specify stack detail** page, under **Parameters**, review the parameters and modify them as necessary.
7. Choose **Next**.
8. On the **Configure stack options** page, under **Stack failure options**, ensure the **Behavior on provisioning failure** radio button is set to **Rollback all stack resources**.
9. choose **Next**.
10. On the **Review** page, review and confirm the settings. Select the boxes acknowledging that the template creates IAM resources and requires certain capabilities.
11. Choose **Update stack** to deploy the stack.

Note

If you deployed the solution in self-managed account discovery mode, you must update the global resources you deployed when following the steps in the Import a Region section.

Troubleshooting

Known issue resolution provides instructions to mitigate known errors. If these instructions don't address your issue, see the Contact AWS Support section for instructions on opening an AWS Support case for this solution.

Known issue resolution

During the deployment of Workload Discovery on AWS and in the post-deployment phase, several common configuration errors can occur:

> **Note**
>
> To help make it easier to troubleshoot, we recommend disabling the rollback on failure feature in the AWS CloudFormation template. You can also find additional troubleshooting help in the Workload Discovery on AWS post-deployment configuration documentation.

Config Delivery Channel Error

Issue: The following error occurs when deploying the main AWS CloudFormation template:

```
Failed to put delivery channel '<stack-name>-DiscoveryImport-<ID-string>-
DeliveryChannel-<ID-string>' because the maximum number of delivery channels:
 1 is reached. (Service: AmazonConfig; Status Code: 400; Error Code:
 MaxNumberOfDeliveryChannelsExceededException; Request ID: 4edc54bc-8c85-4925-
b99d-7ef9c73215b3; Proxy: null)
```

Reason: Solution is being deployed to a region that already has AWS Config enabled.

Resolution: Follow the instructions in the pre-requisites section and deploy the solution with the CloudFormation parameter **AlreadyHaveConfigSetup** set to `Yes`.

Search Resolver Stack Deployment Times Out When Deploying To Existing VPC

Issue: Nested stack that provisions a custom resource to create an index in the OpenSearch cluster times out with the following error:

```
Embedded stack arn:aws:cloudformation:<region>::stack/<stack-name>-
SearchResolversStack-<ID-string>/<guid> was not successfullycreated: Stack creation
 time exceeded the specified timeout
```

Reason: The private subnets provided as CloudFormation parameters do not have the ability to route to S3 (custom resources must write the result of their execution to an S3 bucket using a presigned URL). There are generally two reasons for this:

1. The private subnets do not have NAT gateways associated with them so there is no access to the internet.
2. The private subnet is using VPC endpoints instead of a NAT gateway and the S3 gateway endpoint is not configured correctly.

Resolution:

1. Provision NAT gateways in the VPC to allow tasks running in private subnets to access the internet, either using CloudFormation or the AWS CLI as per the documentation.
2. Ensure that the route tables for the subnets have been updated for the S3 VPC endpoint as per the documentation.

Resources Not Discovered After Account Has Been Imported

Issue: Accounts have been imported through the Web UI but no resources appear to be discovered after the discovery process has run.

Global resources template not deployed

Reason: When the **CrossAccountDiscovery** CloudFormation parameter is set to `SELF_MANAGED`, the global resources CloudFormation template has not been deployed.

Resolution: Deploy the global resources template in the required accounts, as per the documentation.

StackSet deployment error

Reason: When the **CrossAccountDiscovery** CloudFormation parameter is set to `AWS_ORGANIZATIONS`: one or more accounts is not discovered and the **Role Status** column has

Not Deployed entries. This means there has been a problem with the automated deployment of the global resources template using StackSets.

Resolution: Go to the **WdGlobalResources** StackSet in the region that Workload Discovery has been deployed to and check the errors in the stack instances that have failed to deploy:

1. Sign in to the AWS CloudFormation console.
2. From the navigation menu, select **StackSets**.
3. Select the **Service-managed** tab.
4. In the search bar, search for `WdGlobalResources`.
5. Choose `WdGlobalResources` from the search results.
6. Select the **Stack Instances** tab.
7. Inspect the **Detailed status** column for any errors.

Discovery ECS task out of memory

Reason: The discovery process ECS task is running out of memory. This can happen when importing a large number of accounts or resources. The **Last Discovered** column in the UI will display **Not Discovered** or have a value larger than the one specified in the **DiscoveryTaskFrequency** CloudFormation parameter (the default value is 15 minutes). There will be an out of memory error in the ECS console. To verify, follow these steps:

1. Sign in to the Amazon Elastic Container Service console.
2. Select the cluster named workload-discovery-cluster.
3. Choose the Tasks tab.
4. Select the Stopped button in the Desired task status panel.
5. In the **Last Status** column check for the error message `OutOfMemoryError: Container killed due to memory usage`.

Resolution: Update the **Memory** CloudFormation parameter to a larger value: start with double and keep increasing until the error stops.

> **Note**
>
> Only certain combination of CPU units and memory values are valid so you may have to update the **CpuUnits** CloudFormation parameter as well. The full list of combinations is listed in the ECS documentation.

Gremlin Lambda times out when connecting to AWS Neptune

Issue: GraphQL queries backed by the `<stack-name>-GremlinResol-GremlinAppSyncFunction-<ID-string>` lambda function timeout when attempting to connect to the AWS Neptune database.

Reason: The VPC that the database is running has a custom DNS configuration.

Resolution: Update the security group associated with the `<stack-name>-GremlinResol-GremlinAppSyncFunction-<ID-string>` lambda function to open port 53 for the UDP protocol.

Unable to access Elastic Container Registry

Issue: When the scheduled Amazon ECS task on Fargate is launched, the task fails with the following error:

```
ResourceInitializationError: unable to pull secrets or registry auth: execution
 resource retrieval failed: unable to retrieve ecr registry auth
```

Reason: The ECS task is is running in a VPC that does not have a route to the to the ECR API endpoint.

Resolution: Add a VPC endpoint for the `com.amazonaws.<region>.ecr.api` route as per the ECR documentation.

Unable to pull container from Elastic Container Registry

Issue: When the scheduled Amazon ECS task on Fargate is launched, the task fails with the following error:

```
CannotPullContainerFromRegistry: There is a connection issue between the task and
 Amazon ECR. Check your task network configuration
```

Reason: The ECS task is is running in a VPC that does not have a route to the ECR Docker endpoint.

Resolution: Add a VPC endpoint for the `com.amazonaws.<region>.ecr.dkr route as per the ECR documentation.

Only Non-AWS Config Resources Are Being Discovered In Specific Accounts

Issue: The only resource types that the solution discovers are the ones listed in the table on the Supported resources section.

Regional resources template not deployed

Reason: When the **CrossAccountDiscovery** CloudFormation parameter is set to `SELF_MANAGED`, the regional resources CloudFormation template has not been deployed in the regions of each account to be discovered.

Resolution: Deploy the regional resources templates in the required accounts, as per the documentation.

Regional resources template deployed incorrectly

Reason: When the **CrossAccountDiscovery** CloudFormation parameter is set to `SELF_MANAGED`, the regional resources CloudFormation template has been deployed in the regions of a number of accounts that did not have Config enabled but the CloudFormation parameter **AlreadyHaveConfigSetup** was erroneously set to `Yes`.

Resolution: Delete the previous deployed regional resources stack (AWS Config will be in an inconsistent state otherwise) and re-deploy with the CloudFormation parameter **AlreadyHaveConfigSetup** set to `No`.

Config not enabled in required regions

Reason: When the **CrossAccountDiscovery** CloudFormation parameter is set to `AWS_ORGANIZATIONS`, AWS Config is not enabled in the regions of each account to be discovered. In `AWS_ORGANIZATIONS` mode, you are responsible for enabling Config as per your organization's policies.

Resolution: Enable AWS Config in the regions of each account to be discovered.

Contact AWS Support

If you have AWS Developer Support, AWS Business Support, or AWS Enterprise Support, you can use the Support Center to get expert assistance with this solution. The following sections provide instructions.

Create case

1. Sign in to Support Center.
2. Choose **Create case**.

How can we help?

1. Choose **Technical**.
2. For **Service**, select **Solutions**.
3. For **Category**, select **Other Solutions**.
4. For **Severity**, select the option that best matches your use case.
5. When you enter the **Service**, **Category**, and **Severity**, the interface populates links to common troubleshooting questions. If you can't resolve your question with these links, choose **Next step: Additional information**.

Additional information

1. For **Subject**, enter text summarizing your question or issue.
2. For **Description**, describe the issue in detail.
3. Choose **Attach files**.
4. Attach the information that AWS Support needs to process the request.

Help us resolve your case faster

1. Enter the requested information.
2. Choose **Next step: Solve now or contact us**.

Solve now or contact us

1. Review the **Solve now** solutions.
2. If you can’t resolve your issue with these solutions, choose **Contact us**, enter the requested information, and choose **Submit**.

Uninstall the solution

To uninstall the solution, use the AWS Management Console or the AWS Command Line Interface (AWS CLI). First, stop all running tasks from the Amazon ECS cluster. Otherwise, stack deletion can fail.

Using the AWS Management Console

1. Sign in to the AWS CloudFormation console.
2. Select the stack with the name provided during deployment.
3. Choose **Delete stack**.

Using AWS Command Line Interface

Determine whether the AWS CLI is available in your environment. For installation instructions, refer to What Is the AWS Command Line Interface in the *AWS CLI User Guide*.

After confirming that the AWS CLI is available, run the following command:

```
$ aws cloudformation delete-stack --stack-name <customer-defined-stack-name>
```

Developer guide

This section provides the source code for the solution and additional customizations.

Source code

Visit the Workload Discovery on AWS GitHub repository to download the templates and scripts for this solution, and to share your customizations with others.

Locating deployment resources

Follow these steps to locate resources that deployed into your account.

1. Sign in to the AWS CloudFormation console.
2. Select the Region you deployed the solution in.

   Depending on the usage of this account, it may contain multiple stacks for different workloads. There will be a main stack with the name provided during deployment and multiple nested stacks beneath it.
3. Select each stack to access the resources deployed using that template.
4. Select the **Resources** tab and choose the **Physical ID** link for the relevant resource to view the resource in its respective service console.

   If you know the **Logical ID** of a resource, you can also search using the search bar above the table.

Supported resources

The solution supports all the resource types that AWS Config supports, as listed here. The following table contains the supported resources that Workload Discovery on AWS discovers that aren't supported by AWS Config. Details are provided in the corresponding AWS documentation listing.

Resource type	Source	Description
AWS::ApiGateway::Method	SDK	getMethod

Resource type	Source	Description
AWS::ApiGateway::Resource	SDK	getResource
AWS::APIGateway::Authorizer	SDK	getAuthorizers
AWS::Bedrock::Agent	SDK	GetAgent
AWS::Bedrock::AgentVersion	SDK	ListAgentVersions
AWS::Bedrock::CustomModel	SDK	GetCustomModel
AWS::Bedrock::DataSource	SDK	GetDataSource
AWS::Bedrock::FoundationModel	SDK	ListFoundationModels
AWS::Bedrock::ImportedModel	SDK	GetImportedModel
AWS::Bedrock::InferenceProfile	SDK	GetInferenceProfile
AWS::Bedrock::KnowledgeBase	SDK	GetKnowledgeBase
AWS::DynamoDB::Stream	SDK	describeStream
AWS::EC2::Spot	SDK	describeSpotInstanceRequests
AWS::EC2::SpotFleet	SDK	describeSpotFleetRequests
AWS::ECS::Task	SDK	describe-tasks
AWS::EKS::Nodegroup	SDK	describeNodegroup
AWS::ElasticLoadBalancingV2::TargetGroup	SDK	describeTargetGroups
AWS::Glue::Connection	SDK	GetConnections

Resource type	Source	Description
AWS::Glue::Crawler	SDK	BatchGetCrawlers
AWS::Glue::Database	SDK	GetDatabases
AWS::Glue::Tables	SDK	GetTables
AWS::IAM::AWSManagedPolicy	SDK	getAccountAuthorizationDetails
AWS::OpenSearchServerless::Collection	SDK	BatchGetCollection

AWS Organizations account discovery mode

When Workload Discovery on AWS is deployed in an AWS Organization, the discovery of accounts is no longer managed through the solution's web UI. In this case, you don't need to manage the deployment of CloudFormation templates to discover accounts.

Instead, the solution uses an AWS Organization-wide AWS Config aggregator to discover resources in all accounts in the organization that have AWS Config enabled.

For resource types that aren't supported by AWS Config, the solution automatically deploys an IAM role in each account in the organization using AWS CloudFormation StackSets. This role allows the discovery process to make SDK calls in all the organization's accounts to discover these supplementary resources.

This StackSet is configured to automatically deploy the role in any new accounts that are added to the organization and delete the role from any accounts removed from the organization.

> **Note**
>
> It is not possible for a StackSet to deploy stack instance to the Management account. If you want Workload Discovery to discover this account then you must deploy the global resources template using the standard AWS CloudFormation deployment method described in the Deploy the stack to provision the Global resources using CloudFormation section.

Amazon S3 replication role actions

The IAM role used to perform the replication needs to have the following actions:

s3:ReplicateObject
s3:ReplicateDelete
s3:ReplicateTags
s3:ObjectOwnerOverrideToBucketOwner
s3:ListBucket
s3:GetReplicationConfiguration
s3:GetObjectVersionForReplication
s3:GetObjectVersionAcl
s3:GetObjectVersionTagging
s3:GetObjectRetention
s3:GetObjectLegalHold

To verify the role has the replication role actions:

1. Copy the name of the role name in the S3 Replication wizard.
2. Sign in to the IAM Console within the account you are setting up the replication in.
3. Paste the name of the role into the **Search IAM** box.
4. Select the top item from the list. This is the IAM role that will be used.
5. Under **Permissions policies**, expand the **Managed policy**.
6. Ensure that the policy has the actions detailed in the preceding table.

S3 bucket policy

Below is an example of an S3 bucket policy that will allow CURs to be uploaded to the bucket along with permissions to allow external accounts to replicate objects into it. You need to add the IAM Role from each external AWS account to this policy to grant permissions for the replication to take place.

```
   {
      "Version":"2012-10-17",
      "Id":"",
      "Statement":[
          {
            "Sid":"Set permissions for objects"
            "Effect":"Allow",
            "Principal":{
                "AWS":"arn-of-role-selected-in-replication-setup-in-source-account"
          },
      "Action":["s3:ReplicateObject",
      "s3:ReplicateDelete"],
"s3:ObjectOwnerOverrideToBucketOwner",
        "Resource":"arn:aws:s3:::destination-bucket-name/*"
      },
      {
          "Sid":"Set permissions on bucket",
          "Effect":"Allow",
          "Principal":{
                "AWS":"arn-of-role-selected-in-replication-setup-in-source-account"
      },

      "Action":["s3:GetBucketVersioning",
"s3:PutBucketVersioning"],
        "Resource":"arn:aws:s3:::destination-bucket-name "
      },
      {
          "Sid": "Stmt1335892150622",
          "Effect": "Allow",
          "Principal": {
              "Service": "billingreports.amazonaws.com"
          },
          "Action": [
              "s3:GetBucketAcl",
              "s3:GetBucketPolicy"
           ],
```

```
            "Resource": "arn:aws:s3:::destination-bucket-name"
        },
        {

            "Sid": "Stmt1335892526596",
            "Effect": "Allow",
            "Principal": {
                "Service": "billingreports.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::destination-bucket-name/*"

        }
    ]
}
```

AWS APIs

As detailed in the prerequisites, if you are deploying the solution to an existing VPC, the following services must be accessible from your private subnets.

API Gateway

- GetAuthorizers
- GetIntegration
- GetMethod
- GetResources
- GetRestApis

Amazon Bedrock

- GetAgent
- GetCustomModel
- GetDataSource
- GetInferenceProfile
- GetImportedModel

- GetKnowledgeBase
- ListAgentVersions
- ListFoundationModels

Amazon Cognito

- DescribeUserPool

AWS Config

- BatchGetAggregateResourceConfig
- DescribeConfigurationAggregators
- ListAggregateDiscoveredResources
- SelectAggregateResourceConfig

DynamoDB Streams

- DescribeStream

Amazon EC2

- DescribeInstances
- DescribeSpotFleetRequests
- DescribeSpotInstanceRequests
- DescribeTransitGatewayAttachments

Amazon Elastic Load Balancer

- DescribeLoadBalancers
- DescribeListeners
- DescribeTargetGroups
- DescribeTargetHealth

Amazon Elastic Kubernetes Service

- DescribeNodegroup
- ListNodegroups

AWS Glue

- BatchGetCrawlers
- GetConnections
- GetDatabases
- GetTables

IAM

- GetAccountAuthorizationDetails
- ListPolicies

AWS Lambda

- GetFunction
- GetFunctionConfiguration
- ListEventSourceMappings

Amazon OpenSearch Service

- DescribeDomains
- ListDomainNames

Amazon OpenSearch Serverless

- BatchGetCollection

AWS Organizations

- ListAccounts
- ListAccountsForParent
- ListOrganizationalUnitsForParent
- ListRoots

Amazon Simple Notification Service

- ListSubscriptions

Amazon Security Token Service

- AssumeRole

Reference

This section includes information about an optional feature for collecting unique metrics for this solution and a list of builders who contributed to this solution.

Anonymized data collection

This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. When activated, the following information is collected and sent to AWS:

- Solution ID - The AWS solution identifier
- Unique ID (UUID) - Randomly generated, unique identifier for each deployment
- Timestamp - Data collection timestamp
- Cost Feature Enabled - Information on whether the user is using the cost feature
- Number of Accounts - Number of accounts user has onboarded in their deployment
- Number of Diagrams - Number of diagrams created in each deployment
- Number of Resources - Number of resources discovered in all onboarded accounts

AWS owns the data gathered through this survey. Data collection is subject to the Privacy Notice. To opt out of this feature, complete the following steps before launching the AWS CloudFormation template.

1. Download the AWS CloudFormation template to your local hard drive.
2. Open the AWS CloudFormation template with a text editor.
3. Modify the AWS CloudFormation template mapping section from:

```
Mappings:
  Solution:
    Metrics:
      CollectAnonymizedUsageMetrics: 'true'
```

to:

```
Mappings:
```

```
  Solution:
    Metrics:
      CollectAnonymizedUsageMetrics: 'false'
```

1. Sign in to the AWS CloudFormation console.
2. Select **Create stack**.
3. On the **Create stack page**, **Specify template section**, select **Upload a template file.**
4. Under **Upload a template file**, choose **Choose file** and select the edited template from your local drive.
5. Choose **Next** and follow the steps in Launch the stack.

Contributors

- Mohsan Jaffery
- Matthew Ball
- Stefano Vozza
- Connor Kirkpatrick
- Chris Deigan
- Nick Lee
- Tim Mekari

Revisions

Publication date: *September 2020. For updates, refer to CHANGELOG.md file in the GitHub repository.*

Refer to the CHANGELOG.md file in the GitHub repository.

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers, or licensors. AWS products or services are provided "as is" without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

The solution is licensed under the terms of the Apache License, Version 2.0.