AWS Security Reference Architecture (AWS SRA) – identity management

AWS Prescriptive Guidance

AWS Prescriptive Guidance: AWS Security Reference Architecture (AWS SRA) – identity management

Table of Contents

AWS Security Reference Architecture (AWS SRA) – identity management

Global Services Security Team, Amazon Web Services (contributors)

December 2025 (document history)

Influence the future of the AWS Security Reference Architecture (AWS SRA) by taking a short survey.

This guidance provides architectural patterns for building identity management capability on AWS. This is an extension of the AWS SRA – Core Architecture guide. It dives deep into AWS identity management services and how they fit into the core security architecture defined by the AWS SRA. Identities include workforce, application, and consumer identities.

To operate securely in the cloud, your starting point is to determine who can access what in your environment. This guide provides recommendations on how you can implement a scalable, robust, and centralized identity and access management solution on AWS. You can design a centralized identity and access management system, a delegated identity and access management system, or a combination of both while ensuring strict adherence to security standards. Achieving these requirements means ensuring that the right identities can access the right resources under the right conditions. These identities could be humans within your organizations (workforce identities), applications or services within and outside AWS (machine identities), or your customers who want to sign into your applications in ways that are comfortable for them (customer identities).

Identity is now considered the primary perimeter for security. This means that getting identity management right can significantly improve your cloud security posture by eliminating unauthorized use of access, preventing accidental or intentional introduction of malicious code to systems, and ensuring secure, efficient, and compliant operations.

AWS provides fault-tolerant and highly available identity services that can help you to adequately meet your identity management requirements. These services include AWS IAM Identity Center, AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) to centrally manage workforce access to multiple AWS accounts and applications, AWS Identity and Access Management (IAM) roles and IAM Roles Anywhere for secure machine-to-machine

communications, and Amazon Cognito to implement secure and frictionless customer identity and access management into your web and mobile applications.

The sections in this guide provide detailed information about managing different identity types and recommendations for implementing AWS identity services, to help you scale as your identities scale with your environment.

In this guide:

About the AWS SRA library

Influence the future of the AWS Security Reference Architecture (AWS SRA) by taking a short survey.

This guide is part of a library that provides architectural blueprints and technical guidance for designing and building security architectures on AWS. The library consists of implementation code (AWS SRA code library), a validation tool (SRA Verify), and two complementary categories of guides that cover the core architecture and deep dive architectures.

AWS SRA – core architecture

This guide represents a foundation for the recommended AWS security architecture. It is the starting point that applies to all organizations, regardless of their industry, application type, or any other considerations. This foundation helps you build a strong and scalable architecture on AWS and helps create a strong AWS multi-account security baseline that securely scales as your business grows.

AWS SRA – deep dive architectures

The *AWS SRA – core architecture* guide is complemented by additional publications that provide architectural patterns aligned to specific security capabilities, application types, and compliance or regulatory requirements. These patterns extend the core architecture and should be used in conjunction with the *AWS SRA – core architecture* guide.

The following guides provide architectural patterns aligned to specific security capabilities:

- *AWS SRA – identity management* (this guide) provides guidance on how to implement a scalable, robust, and centralized identity and access management solution on AWS.
- AWS SRA – perimeter security discusses architecture patterns and AWS services for implementing edge security in a central account or in individual accounts.
- AWS SRA – cyber forensics describes how to configure an AWS Forensics account as a starting point to develop your organization's forensic capabilities and to help improve your security incident response (IR) preparedness.

The following guides provide architectural patterns for specific application types. You might want to focus on these after you build your baseline security architecture:

- AWS SRA – AI security provides security architectural recommendations for designing and building applications that incorporate generative AI capabilities by using AWS generative AI services.
- AWS SRA – IoT provides security architectural recommendations for designing and building IoT applications on AWS.

In addition, the following guide describes architectural patterns that are aligned with specific compliance or regulatory frameworks:

- AWS Privacy Reference Architecture (AWS PRA) provides a security architecture for applications that process personal data and must support broad privacy compliance requirements such as the General Data Protection Regulation (GDPR), the California Consumer Privacy Act (CCPA), or the Brazilian General Data Protection Law (LGPD). The AWS PRA provides a set of guidelines that are specific to the design and configuration of privacy controls in AWS services.

We recommend that you start with the *AWS SRA – core architecture* guide to understand the foundational architecture and then consult the complementary guides to take advantage of advanced functionality and implementations. For more information about this content set, see AWS Security Reference Architecture.

Tip

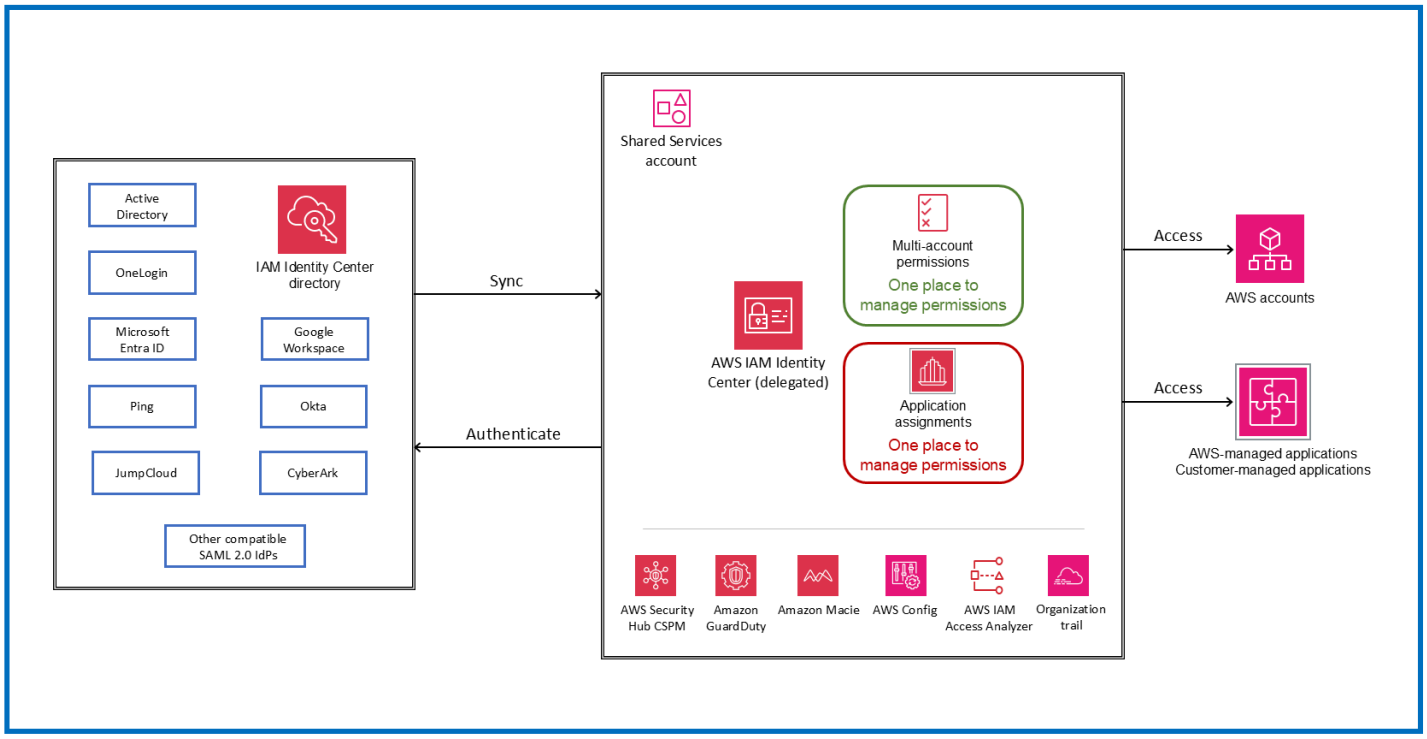
To customize the reference architecture diagrams in the AWS SRA library based on your business needs, you can download the following .zip file and extract its contents.

Download the diagram source file (Microsoft PowerPoint format)

Workforce identity management

Influence the future of the AWS Security Reference Architecture (AWS SRA) by taking a short survey.

Workforce identity management, which is illustrated in the following diagram, refers to managing human access to resources that help build and manage your businesses within your cloud infrastructure and applications. It supports secure provisioning, managing, and removing access, as employees join an organization, move between roles, and leave an organization. Identity administrators can create identities directly in AWS or connect to an external identity provider (IdP) to enable employees to use their corporate credentials to securely access AWS accounts and business applications from one place.



By using AWS IAM Identity Center to manage access to AWS managed applications, you can benefit from new capabilities such as trusted identity propagation from your query application to the AWS data service, and new services such as Amazon Q that provide a continuous user experience as users move from one Amazon Q-enabled service to another. The use of IAM Identity Center for AWS account access prevents the creation and use of IAM users, which have long-term access to resources. Instead, it enables workforce identities to access resources in AWS accounts by using

temporary credentials from IAM Identity Center, which is a security best practice. Workforce identity management services let you define fine-grained access control for AWS resources or applications in your multi-account AWS environment based on specific job functions or user attributes. These services also help audit and review user activities within your AWS environment.

AWS offers several several options for workforce identity and access management: IAM Identity Center, IAM SAML federation, and AWS Managed Microsoft AD.

- IAM Identity Center is the recommended service for managing workforce access to AWS applications and multiple AWS accounts. You can use this service with an existing identity source, such as Okta, Microsoft Entra ID, or on-premises Active Directory, or by creating users in its directory. IAM Identity Center supplies all AWS services with a shared understanding of your workforce users and groups. AWS managed applications integrate with it, so you do not need to connect your identity source individually to each service, and you can manage and view your workforce access from a central location. You can use IAM Identity Center to manage access to AWS applications while you continue to use your established configuration to access AWS accounts. For new multi-account environments, IAM Identity Center is the recommended service to manage your workforce access to the environment. You can assign permissions consistently across AWS accounts, and your users receive single sign-on access across AWS.
- An alternate way to grant your workforce access to AWS accounts is by using IAM SAML 2.0 federation. This involves creating one-to-one trust between your organization's IdP and each AWS account, and isn't recommended for multi-account environments. Inside your organization, you must have an IdP that supports SAML 2.0, such as Microsoft Entra ID, Okta, or another compatible SAML 2.0 provider.
- Another option is to use Microsoft Active Directory (AD) as a managed service to run directory-aware workloads in AWS. You can also configure a trust relationship between AWS Managed Microsoft AD in the AWS Cloud and your existing on-premises Microsoft Active Directory, to provide users and groups with access to resources in either domain by using IAM Identity Center.

Design considerations

- Although this section discusses several services and options, we recommend that you use IAM Identity Center to manage workforce access, because it has advantages over the other two approaches. Later sections discuss the advantages and use cases for individual approaches. A growing number of AWS managed applications require the use of IAM

Identity Center. If you are currently using IAM federation, you can enable and use IAM Identity Center with AWS applications without changing your existing configurations.

- To improve federation resiliency, we recommend that you configure your IdP and AWS federation to support multiple SAML sign-in endpoints. For details, see the AWS blog post How to use regional SAML endpoints for failover.

AWS IAM Identity Center

AWS IAM Identity Center provides a single place to create or connect your growing workforce identities and centrally manage secure access for those identities across your AWS environment. You can enable IAM Identity Center in conjunction with AWS Organizations. This is the recommended approach to provide centrally managed access to multiple AWS accounts within your AWS organization and AWS managed applications.

AWS managed services, including Amazon Q, Amazon Q Developer, Amazon SageMaker Studio, and Amazon Quick, integrate and use IAM Identity Center for authentication and authorization. You connect your identity source only once to IAM Identity Center and manage workforce access to all onboarded AWS-managed applications. Identities from your existing corporate directories, such as Microsoft Entra ID, Okta, Google Workspace, and Microsoft Active Directory, must be provisioned into IAM Identity Center before you can look up users or groups to grant them single sign-on access to AWS managed services. IAM Identity Center also powers application-specific, user-centric experiences. For example, users of Amazon Q experience continuity as they move from one Amazon Q-integrated service to another.

Note

You can use IAM Identity Center capabilities individually. For example, you might choose to use IAM Identity Center only to manage access to AWS managed services such as Amazon Q while using direct account federation and IAM roles to manage access to your AWS accounts.

Trusted identity propagation provides a streamlined single sign-on experience for users of query tools and business intelligence (BI) applications who require access to data in AWS services. Data access management is based on a user's identity, so administrators can grant access based on the user's existing user and group memberships. Trusted identity propagation is built on the OAuth 2.0

Authorization Framework, which allows applications to access and share user data securely without sharing passwords.

AWS managed services that integrate with trusted identity propagation, such as Amazon Redshift query editor v2, Amazon EMR, and Amazon Quick, obtain tokens from IAM Identity Center directly. IAM Identity Center also provides an option for applications to exchange identity tokens and access tokens from an external OAuth 2.0 authorization server. User access to AWS services and other events is recorded in service-specific logs and in AWS CloudTrail events, so auditors know what actions the users took and which resources they accessed.

To use trusted identity propagation, you must enable IAM Identity Center and provision users and groups. We recommend that you use an organization instance of IAM Identity Center.

> **Note**
>
> Trusted identity propagation doesn't require you to set up multi-account permissions (permission sets). You can enable IAM Identity Center and use it for trusted identity propagation only.

For more information, see the prerequisites and considerations for using trusted identity propagation and view the specific use cases supported by applications that can initiate identity propagation.

The AWS access portal provides authenticated users with single sign-on access to their AWS accounts and cloud applications. You can also use the credentials generated from the AWS access portal to configure AWS Command Line Interface (AWS CLI) or AWS SDK access to resources in your AWS accounts. This helps you eliminate the use of long-term credentials for programmatic access ,which significantly reduces the chances of credentials becoming compromised and improves your security posture.

You can also automate management of account and application access by using IAM Identity Center APIs.

IAM Identity Center is integrated with AWS CloudTrail, which provides a record of the actions taken by a user in IAM Identity Center. CloudTrail records API events such as a **CreateUser** API call, which is recorded when a user is either manually created or provisioned or synchronized to IAM Identity Center from an external IdP by using the System for Cross-domain Identity Management (SCIM) protocol. Every event or log entry recorded in CloudTrail contains information about who

generated the request. This capability helps you identify unexpected changes or activities that might require further investigation. For a complete list of supported IAM Identity Center operations in CloudTrail, see the IAM Identity Center documentation.

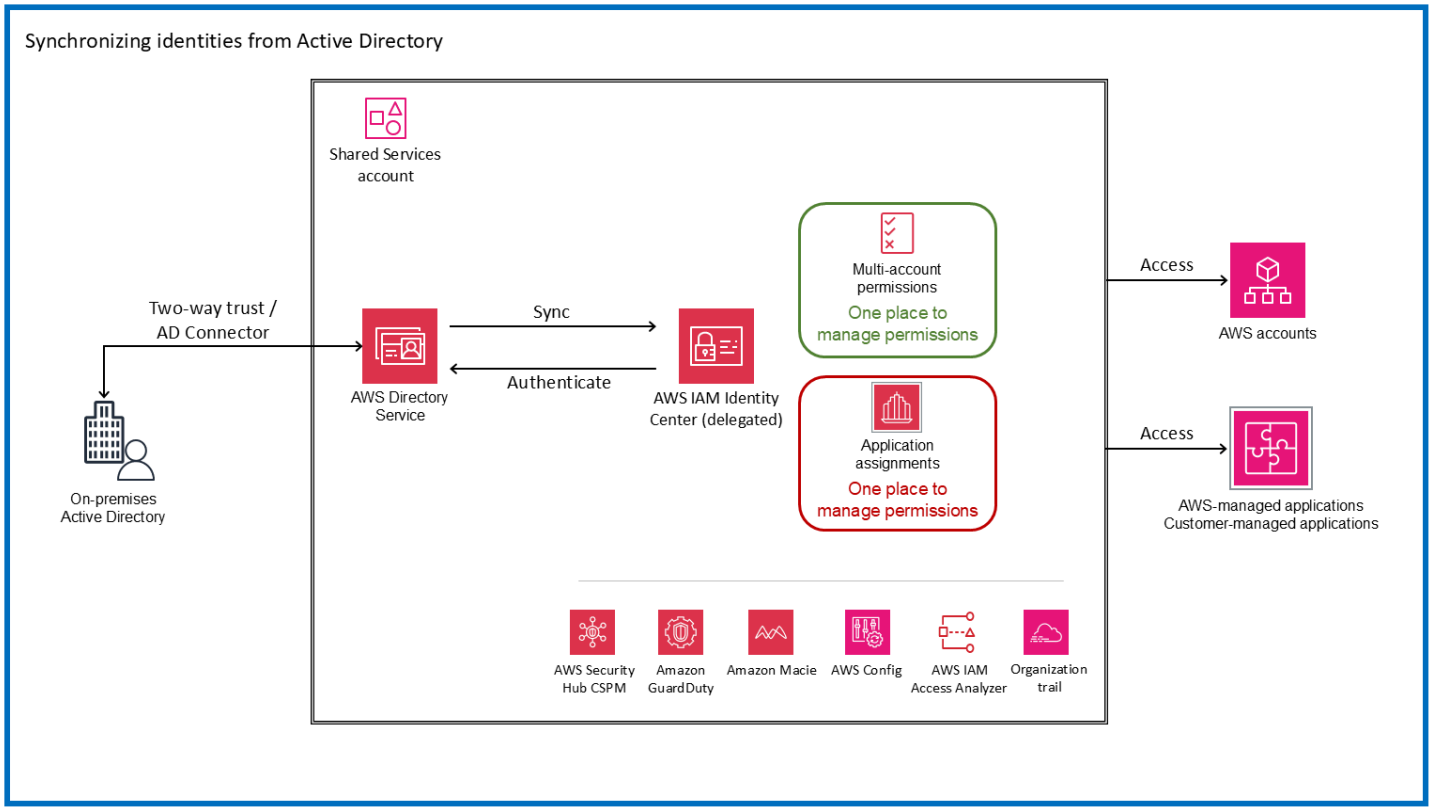
Connecting your existing identity source to IAM Identity Center

Identity federation is a common approach to building access control systems, which manage user authentication by using a central IdP and govern their access to multiple applications and services that act as service providers (SPs). IAM Identity Center gives you the flexibility to bring identities from your existing corporate identity source, including Okta, Microsoft Entra ID, Ping, Google Workspace, JumpCloud, OneLogin, on-premises Active Directory, and any SAML 2.0 compatible identity source.

Connecting your existing identity source to IAM Identity Center is the recommended approach, because it gives your workforce single sign-on access and a consistent experience across AWS services. It is also best practice to manage identities from a single location instead of maintaining multiple sources. IAM Identity Center supports identity federation with SAML 2.0, which is an open identity standard that allows IAM Identity Center to authenticate users from external IdPs. IAM Identity Center also provides support for the SCIM v2.0 standard. This standard enables automatic provisioning, updating, and deprovisioning of users and groups between any of the supported external IdPs and IAM Identity Center, except Google Workspace and PingOne, which currently support provisioning of users only through SCIM.

You can also connect other SAML 2.0-based external IdPs to IAM Identity Center, if they conform to specific standards and considerations.

You can also connect your existing Microsoft Active Directory to IAM Identity Center. This option allows you to synchronize users, groups, and group memberships from an existing Microsoft Active Directory by using AWS Directory Service. This option is suitable for large enterprises that are already managing identities, either in a self-managed Active Directory that's located on premises or in a directory in AWS Managed Microsoft AD. You can connect a directory in AWS Managed Microsoft AD to IAM Identity Center. You can also connect your self-managed directory in Active Directory to IAM Identity Center by establishing a two-way trust relationship that permits IAM Identity Center to trust your domain for authentication. Another method is to use AD Connector, which is a directory gateway that can redirect directory requests to your self-managed Active Directory without caching any information in the cloud. The following diagram illustrates this option.



Benefits:

- Connect your existing identity source to IAM Identity Center to streamline access and provide a consistent experience to your workforce across AWS services.
- Efficiently manage workforce access to AWS applications. You can manage and audit user access to AWS services more easily by making user and group information from your identity source available through IAM Identity Center.
- Improve control and visibility of user access to data in AWS services. You can enable the transfer of user identity context from your business intelligence tool to the AWS data services you use while continuing to use your chosen identity source and other AWS access management configurations.
- Manage workforce access to a multi-account AWS environment. You can use IAM Identity Center with your existing identity source or create a new directory, and manage workforce access to part or all of your AWS environment.
- Provide an additional layer of protection in the event of service disruption in the AWS Region where you enabled IAM Identity Center by setting up emergency access to the AWS Management Console.

Service consideration

IAM Identity Center doesn't currently support the use of idle timeout, where the user's session times out or is extended based on activity. It does support session duration for the AWS access portal and IAM Identity Center integrated applications. You can configure session duration between 15 minutes and 90 days. You can view and delete active AWS access portal sessions for IAM Identity Center users. However, modifying and ending AWS access portal sessions have no effect on the session duration of the AWS Management Console, which is defined in permission sets.

Design considerations

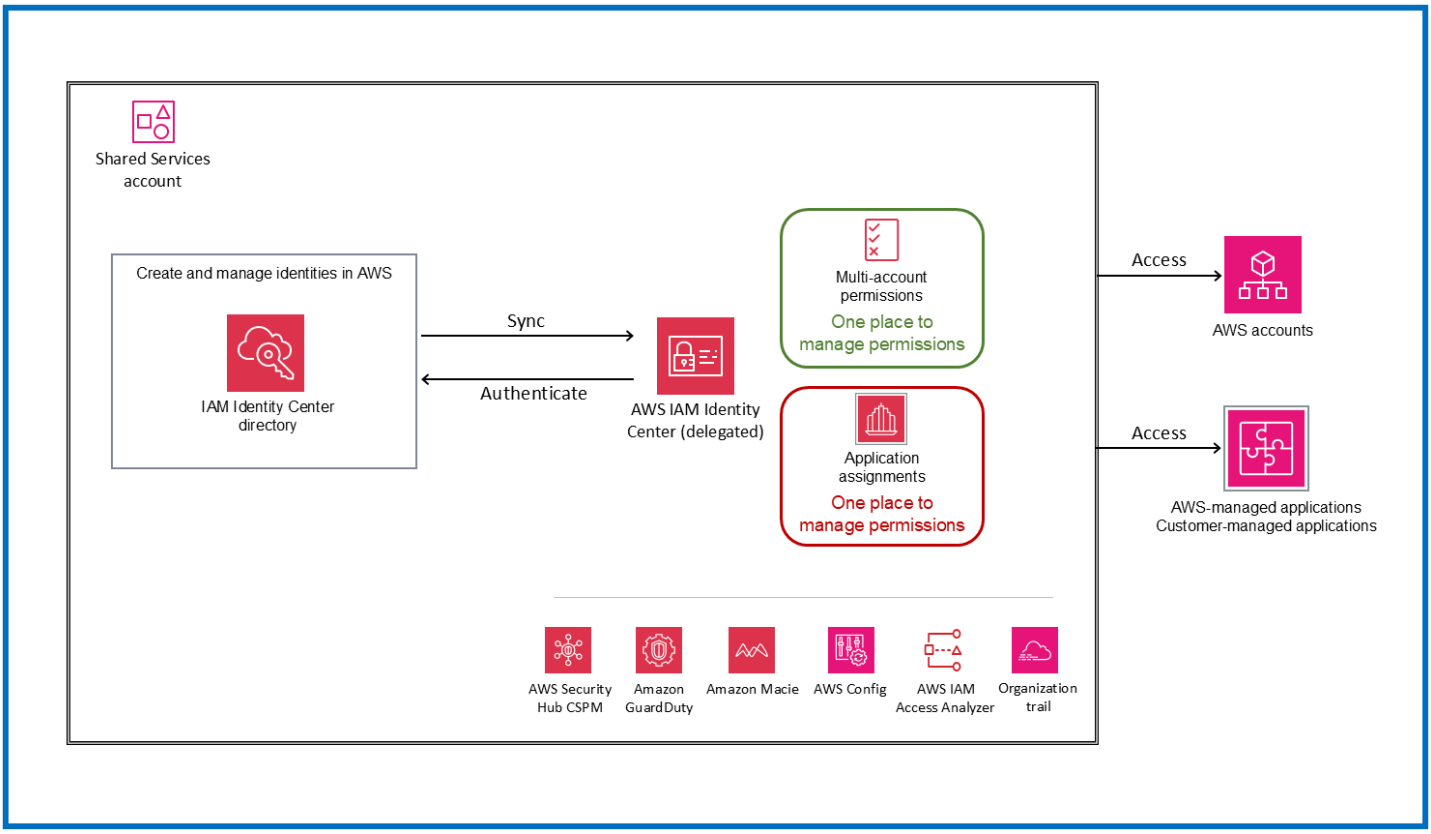
- You can enable an instance of IAM Identity Center in a single AWS Region at one time. When you enable IAM Identity Center, it controls access to its permission sets and integrated applications from the primary Region. This means that in the unlikely event of a disruption of the IAM Identity Center service in this Region, users will not be able to sign in to access accounts and applications. To provide extra protection, we recommend that you set up emergency access to the AWS Management Console by using SAML 2.0-based federation.

  Note: This emergency access recommendation is applicable if you are using a third-party external IdP as your identity source and works when the IAM service data plane and your external IdP are available.
- If you use Active Directory or create users in IAM Identity Center, follow the standard AWS break-glass guidance.
- If you plan to use AD Connector to connect your on-premises Active Directory to IAM Identity Center, consider that AD Connector has a one-on-one trust relationship with your Active Directory domain and doesn't support transitive trusts. This means that IAM Identity Center can access only the users and groups of the single domain that's attached to the AD Connector you created. If you need to support multiple domains or forests, use AWS Managed Microsoft AD.
- If you are using an external IdP, multi-factor authentication (MFA) is managed from the external IdP and not in IAM Identity Center. IAM Identity Center supports MFA

capabilities only when your identity source is configured with the IAM Identity Center identity store, AWS Managed Microsoft AD, or AD Connector.

Creating and managing identities in AWS

We recommend that you use IAM Identity Center with an external IdP. However, if you don't have an existing IdP, you can create and manage users and groups in the IAM Identity Center directory, which is the default identity source for the service. This option is illustrated in the following diagram. It is preferred over creating IAM users or roles in each AWS account for workforce users. For more information, see the IAM Identity Center documentation.



Service considerations

- When you create and manage identities in IAM Identity Center, your users must adhere to the default password policy, which can't be modified. If you want to define and use your own password policy for your identities, change your identity source to either Active Directory or to an external IdP.

- When you create and manage identities in IAM Identity Center, consider planning for disaster recovery. IAM Identity Center is a regional service that is built to operate across multiple Availability Zones to withstand the failure of an Availability Zone. However, in the unlikely event of a disruption in the Region where your IAM Identity Center is enabled, you will not be able to implement and use the emergency access setup recommended by AWS, because the IAM Identity Center directory that contains your users and groups will also be affected by any disruption in that Region. To implement disaster recovery, you need to change your identity source to either an external SAML 2.0 IdP or to Active Directory.

Design considerations

- IAM Identity Center supports the use of only one identity source at a time. However, you can change your current Identity source to one of the other two identity source options. Before you make this change, evaluate the impact by reviewing the considerations for changing your identity source.
- When you use the IAM Identity Center directory as your identity source, MFA is enabled by default for instances that were created after November 15, 2023. New users are prompted to register an MFA device when they sign in to IAM Identity Center for the first time. Administrators can update MFA settings for their users based on their security requirements.

General design considerations for IAM Identity Center

- IAM Identity Center supports attribute-based access control (ABAC), which is an authorization strategy that enables you to create fine-grained permissions by using attributes. There are two ways to pass attributes for access control to IAM Identity Center:
  - If you're using an external IdP, you can pass attributes directly in the SAML assertion by using the prefix `https://aws.amazon.com/SAML/Attributes/AccessControl`.
  - If you're using IAM Identity Center as an identity source, you can add and use attributes that are in the IAM Identity Center identity store.
  - To use ABAC in all cases, you must first select the access control attribute on the **Attributes for access control** page on the IAM Identity Center console. To pass it by using SAML

assertion, you must set the attribute name in the IdP to `https://aws.amazon.com/SAML/Attributes/AccessControl:<AttributeName>`.

- The attributes that are defined on the IAM Identity Center console **Attributes for access control** page take precedence over the attributes passed through SAML assertions from your IdP. If you want to use attributes passed from SAML assertion only, don't define any attributes manually in IAM Identity Center. After you define attributes either in the IdP or in IAM Identity Center, you can create custom permissions policies in your permission set by using the aws:PrincipalTag global condition key. This ensures that that only users with attributes that match the tags on your resources have access to those resources in your AWS accounts.

- IAM Identity Center is a workforce identity management service, so it requires human interaction to complete the authentication process for programmatic access. If you need short-term credentials for machine-to-machine authentication, explore Amazon Elastic Compute Cloud (Amazon EC2) instance profiles for workloads in AWS or IAM Roles Anywhere for workloads outside AWS.
- IAM Identity Center provides access to resources in AWS accounts within your organizations. However, if you want to provide single sign-on access to external accounts (that is, AWS accounts outside your organization) by using IAM Identity Center without inviting those accounts into your organizations, you can configure the external accounts as SAML applications in IAM Identity Center.
- IAM Identity Center supports integration with temporary elevated access management (TEAM) solutions (also known as just-in-time access). This integration provides time-bound elevated access to your multi-account AWS environment at scale. Temporary elevated access allows users to request access to perform a specific task for a specific period of time. An approver reviews each request and decides whether to approve or reject it. IAM Identity Center supports both vendor-managed TEAM solutions from supported AWS security partners or self-managed solutions, which you maintain and tailor to address your time-bound access requirements.

IAM federation

> **Note**
>
> If you already have a central user directory for managing users and groups, we recommend that you use IAM Identity Center as your primary workforce access service. If any of the

design considerations discussed later in this section prevent you from using IAM Identity Center, use IAM federation instead of creating separate IAM users in AWS.

IAM federation establishes a trust system between two parties for the purpose of authenticating users and sharing the information needed to authorize their access to resources. This system requires an identity provider (IdP) that's connected to your user directory and a service provider (SP) that is managed in IAM. The IdP is responsible for authenticating users and supplying relevant authorization context data to IAM, and IAM controls access to resources in AWS accounts and environments.

IAM federation supports commonly used standards such as SAML 2.0 and OpenID Connect (OIDC). SAML-based federation is supported by many IdPs and enables federated single sign-on access for users to sign in to the AWS Management Console or call an AWS API without having to create IAM users. You can create user identities in AWS by using IAM or connect to your existing IdP (for example, Microsoft Active Directory, Okta, Ping Identity, or Microsoft Entra ID). Alternatively, you can use an IAM OIDC identity provider when you want to establish trust between an OIDC-compatible IdP and your AWS account.

There are two design patterns for IAM federation: multi-account federation or single-account federation.

Multi-account IAM federation

In this multi-account IAM pattern, you establish a separate SAML-trust relationship between the IdP and all AWS accounts that need to be integrated. The permissions are mapped and provisioned on an individual account basis. This design pattern provides a distributed approach to managing roles and policies, and gives you the flexibility to enable a separate SAML or OIDC IdP for each account and use federated user attributes for access control.

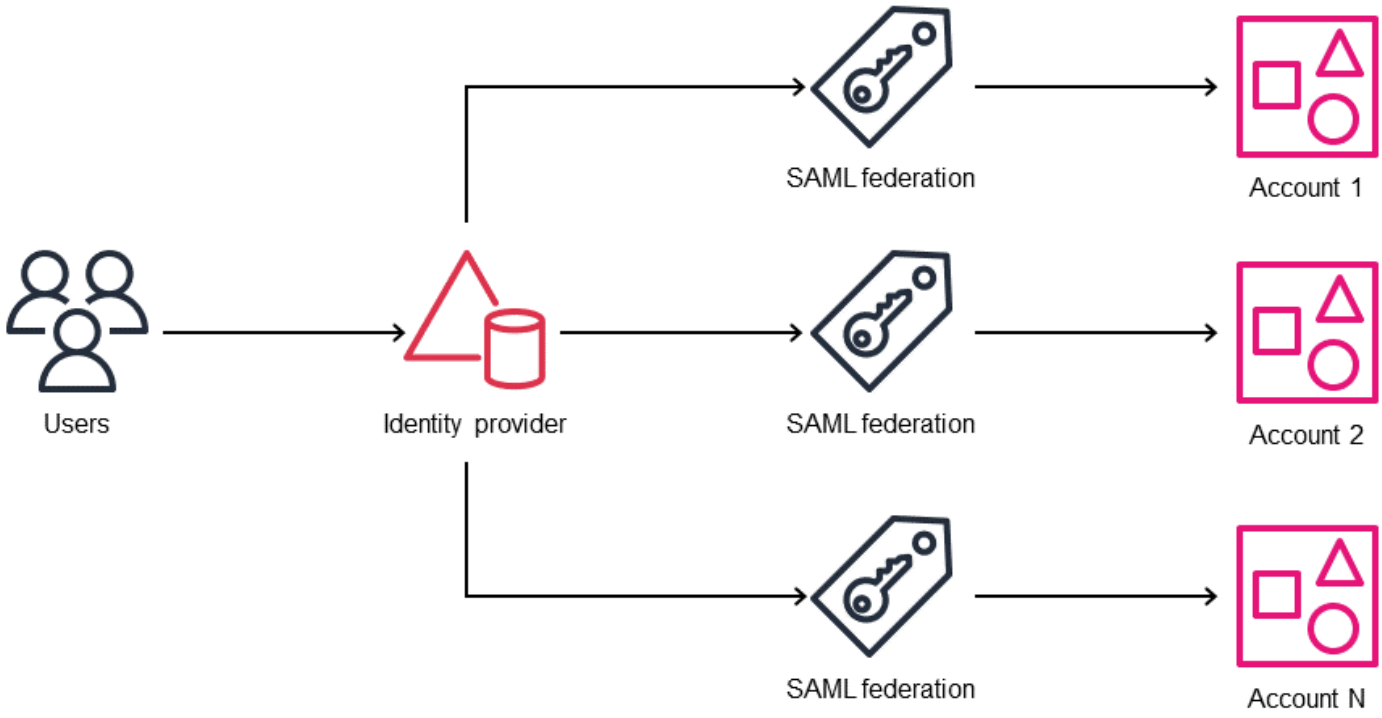
Multi-account IAM federation provides these benefits:

- Provides central access to all your AWS accounts and lets you manage permissions in a distributed way for each AWS account.
- Achieves scalability in a multi-account setup.
- Meets compliance requirements.
- Lets you manage identities from a central location.

The design is particularly helpful if you want to manage permissions in a distributed manner, separated by AWS accounts. It also helps in scenarios where you do not have repeatable IAM permissions across Active Directory users in their AWS accounts. For example, it supports network administrators who might provide resource access with slight variations across accounts.

SAML providers have to be created separately in each account, so each AWS account requires processes to manage the creation, update, and deletion of IAM roles and their permissions. This means that you can define precise and distinct IAM role permissions for AWS accounts with different levels of sensitivity for the same job function.

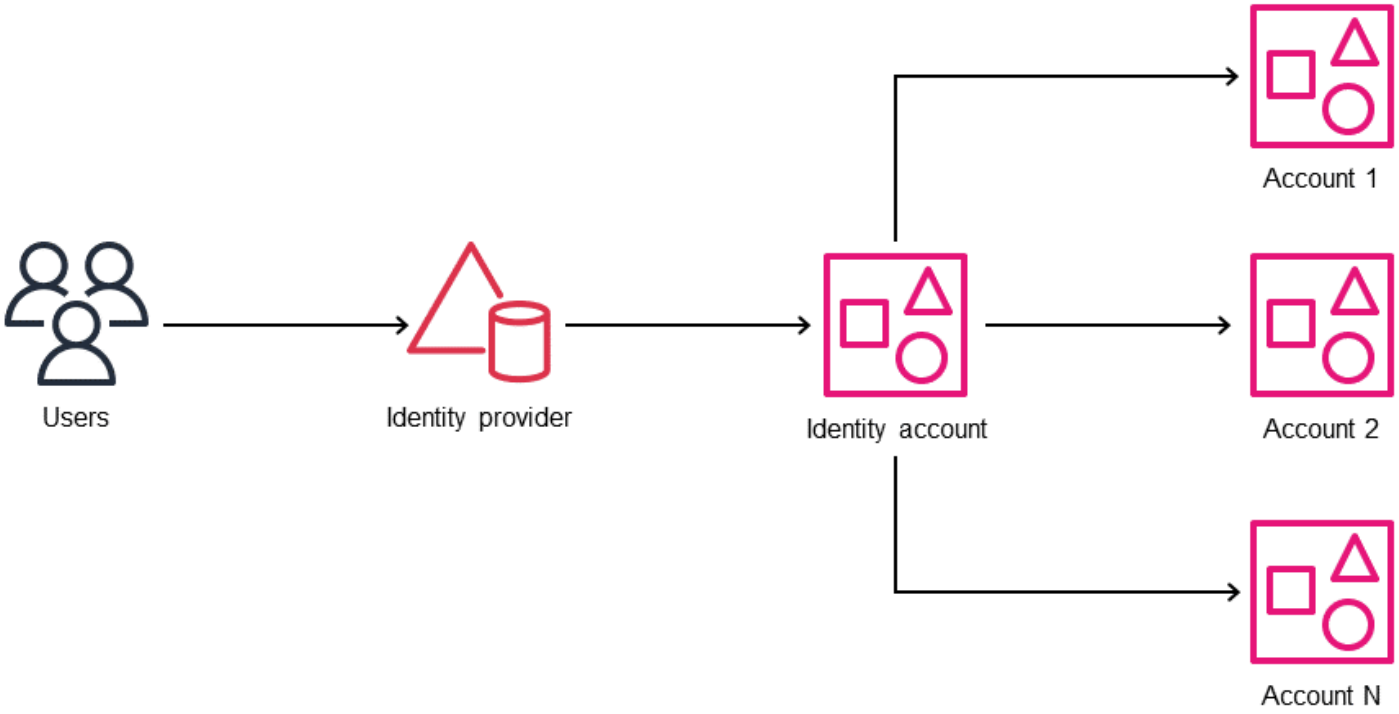
The following diagram illustrates the multi-account IAM federation pattern.



Single-account IAM federation (hub-and-spoke model)

> **Note**
>
> Use this design pattern for the specific scenarios described in this section. For most scenarios, IAM Identity Center-based federation or multi-account IAM federation is the recommended approach. For questions, contact AWS Support.

In the single-account federation pattern, the SAML trust relationship is established between the IdP and a single AWS account (the identity account). The permissions are mapped and provisioned through the centralized identity account. This design pattern provides simplicity and efficiency. The identity provider provides SAML assertions that are mapped to specific IAM roles (and permissions) in the identity account. Federated users can then assume cross-account-roles to access other AWS accounts from the identity account. The following diagram illustrates the single-account IAM federation pattern.



Use cases:

- Companies that have a single AWS account, but sometimes need to create short-lived AWS accounts for isolated sandbox or testing.
- Educational institutions that maintain their production services in a main account but provide temporary, project-based student accounts.

> **Note**
>
> These use cases require strong governance and time-bound recycling processes to ensure that production data doesn't pass into the federated accounts and to remove potential security risks. The auditing process is also difficult in these scenarios.

Design considerations for choosing between IAM federation and IAM Identity Center

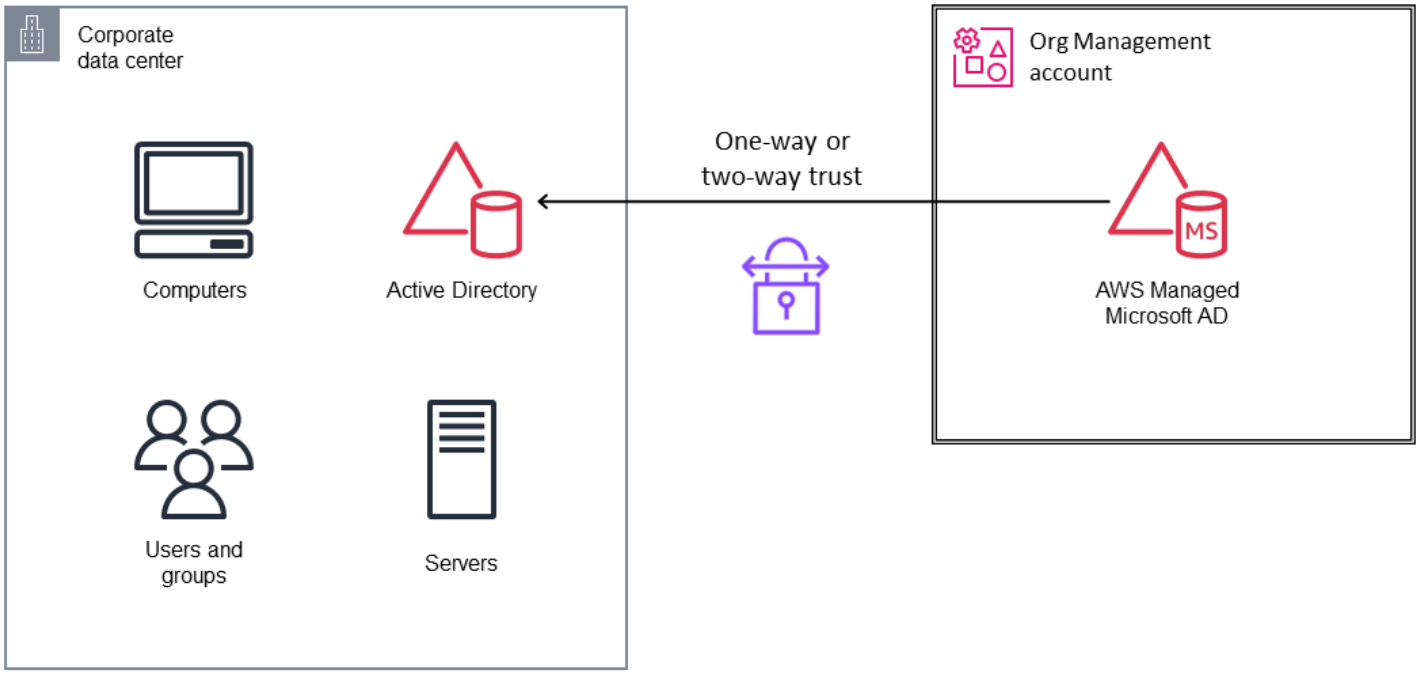
- IAM Identity Center supports connecting accounts to only one directory at a time. If you use multiple directories or want to manage permissions based on user attributes, consider using IAM federation as a design alternative. You should have an IdP that supports the SAML 2.0 protocol, such as Microsoft Active Directory Federation Service (AD FS), Okta, or Microsoft Entra ID. You can establish two-way trust by exchanging IdP and SP metadata, and configuring SAML assertions to map IAM roles to corporate directory groups and users.
- If you use an IAM OIDC identity provider to establish trust between an OIDC-compatible IdP and your AWS account, consider using IAM federation. When you use the IAM console to create an OIDC identity provider, the console attempts to fetch the thumbprint for you. We recommend that you also obtain the thumbprint for your OIDC IdP manually and verify that the console fetched the correct thumbprint. For more information, see Create an OIDC identity provider in IAM in the IAM documentation.
- Use IAM federation if your corporate directory users don't have repeatable permissions for a job function. For example, different network or database administrators might needs customized IAM role permissions in AWS accounts. To achieve this in IAM Identity Center, you can create separate customer managed policies and reference them in your permission sets. For more information, see the AWS blog post How to use customer managed policies in AWS IAM Identity Center for advanced use cases.
- If you are using a distributed permissions model, where each account manages their own permissions, or a centralized permissions model through AWS CloudFormation StackSets, consider using IAM federation. If you are using a hybrid model that involves both centralized and distributed permissions, consider using IAM Identity Center. For more information, see Identity providers and federation in the IAMdocumentation.
- Services and features such as Amazon Q Developer Professional and AWS CLI version 2 have built-in support for IAM Identity Center. However, some of those capabilities aren't supported with IAM federation.
- IAM Access Analyzer currently doesn't support the analysis of IAM Identity Center users actions.

AWS Managed Microsoft AD

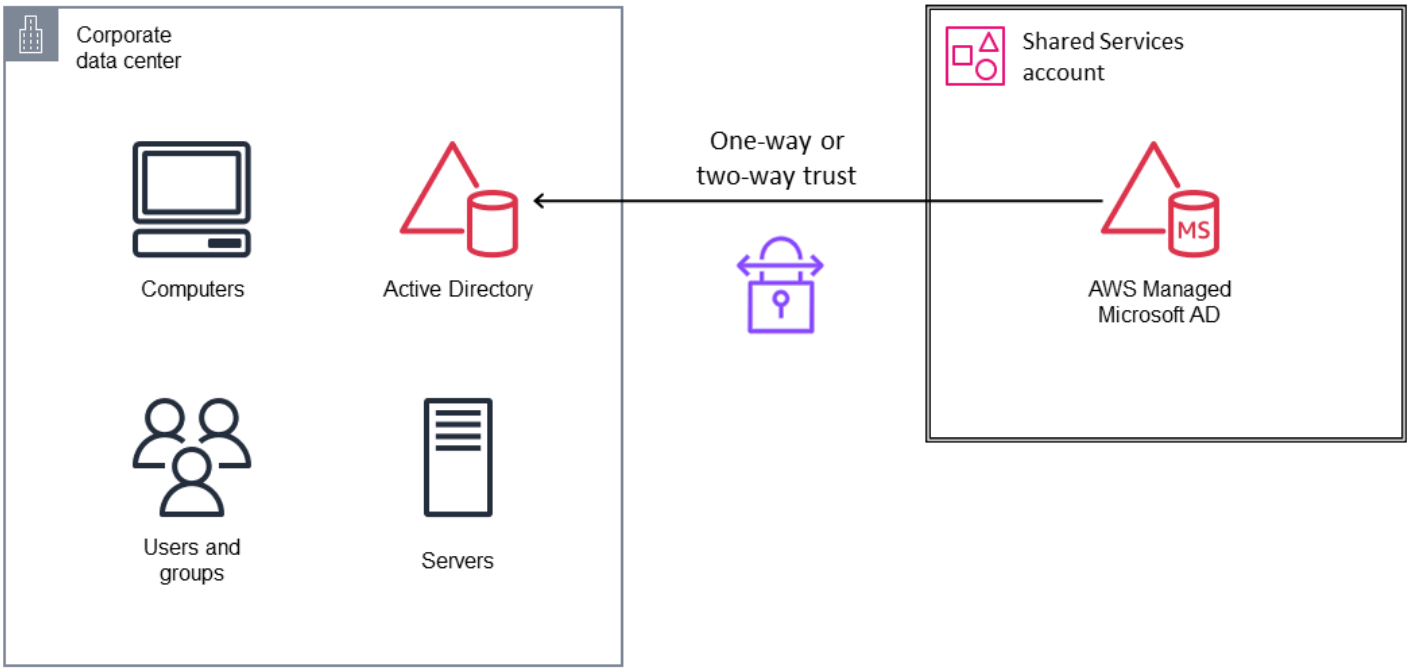
AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) is an AWS managed service that provides a managed Active Directory solution based on Microsoft Windows Server Active Directory Domain Services (AD DS). The domain controllers run in different Availability Zones in a Region of your choice. Host monitoring and recovery, data replication, snapshots, and software updates are automatically configured and managed for you. You can configure a trust relationship between AWS Managed Microsoft AD in the AWS Cloud and your existing on-premises Microsoft Active Directory. This gives users and groups access to resources in either domain by using IAM Identity Center.

For strict access restriction, you can create a separate AWS account or AWS organizational unit (OU) within your organization for identity services such as Active Directory, including AWS Managed Microsoft AD, and give only a very limited group of administrators access to this account. Generally, we recommend that you treat Active Directory on AWS in the same manner as on-premises Active Directory. Make sure to limit administrative access to the AWS account, similar to how you would limit access to a physical data center. Whoever owns the AWS account that contains Active Directory can own the Active Directory. For more information, see Design consideration for AWS Managed Microsoft AD in the *Active Directory Domain Services on AWS* whitepaper.

When you use AWS Managed Microsoft AD sharing by using AWS Organizations, you must deploy AWS Managed Microsoft AD to the Org Management account as shown in the following diagram.



If you use sharing by using the handshake method, where consumer accounts accept the directory sharing request, you can deploy AWS Managed Microsoft AD to any account within or outside your organization in AWS Organizations. In the AWS SRA, AWS Managed Microsoft AD is deployed in the Shared Services account, as shown in the following diagram. This AWS Organizations sharing method makes it easier to share the directory within your organization because you can browse and validate the Active Directory consumer accounts.



All AWS services observe a shared responsibility model. This model divides the responsibilities for AWS Managed Microsoft AD between AWS and customers.

AWS responsibility:

- Directory availability
- Directory patching and service improvements
- Security of directory infrastructure
- Domain controller security posture through group policy objects (GPOs) and other methods
- Improving security posture when needed; for example, for Server Message Block (SMB) version 1 depreciation
- Management and creation of objects outside the customer's OU

Customer responsibility:

- Setting fine-grained password policies for users
- Security of objects within the customer's OU
- Initializing a directory restore operation
- Active Directory trust creation and security
- Server-side and client-side Lightweight Directory Access Protocol (LDAP) over SSL implementation
- Implementing multi-factor authentication (MFA)
- Disabling legacy network ciphers and protocols

Based on these responsibilities, you have some influence over the security of your directory. Because AWS provides managed services, it doesn't give customers full control. In this model, the security controls you manage are smaller in scope than for a self-managed Active Directory.

Design considerations

- Use fine-grained password policies to set advanced password policies. The default password policy in AWS Managed Microsoft AD offers compatibility with this practice, but it is relatively weak because of a short password length. We recommend that you use passwords that contain 15 or more characters so that Active Directory won't store LAN Manager (LM) hashes for your account. For more information, see the Microsoft documentation.
- Disable any unused network and protocol ciphers on AWS Managed Microsoft AD. For details, see Editing AWS Managed Microsoft AD directory security settings in the Directory Service documentation.
- To further enhance the security of your AWS Managed AD, you can restrict the network ports and sources of the AWS security group that's attached to your AWS Managed Microsoft AD. For more information, see Enhancing your AWS Managed Microsoft AD network security configuration in the Directory Service documentation.
- Enable log forwarding for your AWS Managed Microsoft AD. This allows AWS Managed Microsoft AD to forward the raw Windows security event logs of your AWS Managed Microsoft AD domain controllers to an Amazon CloudWatch log group in your account.
- Create a group policy object (GPO) that denies domain and enterprise administrators network or remote access rights to domain-joined computer accounts. For more

information, see the Microsoft documentation for the security policy settings Deny log on locally and Deny log on through Remote Desktop Services.

- Implement a public key infrastructure (PKI) to issue certificates to their domain controllers to encrypt LDAP traffic. For more information, see the AWS blog post How to enable server-side LDAPS for your AWS Managed Microsoft AD directory.
- To establish Active Directory trust relationships with AWS Managed Microsoft AD, create a forest trust. This type of trust allows for maximum Kerberos compatibility. We recommend that you use a one-way trust whenever possible, although some use cases require a two-way trust. Another option for trust security is to enable selective authentication on the trust. When you enable selective authentication, you must set the **Allowed to Authenticate** permission on each computer object the trusted user will access in addition to any other permissions that are required to access the computer object. For details, see the AWS blog post Everything you wanted to know about trusts with AWS Managed Microsoft AD.
- Each AWS Managed Microsoft AD deployment has an Active Directory account that's provisioned to administer the directory. This account is named *Admin*. After you deploy the directory, we recommend that you create individual Active Directory user accounts for each elevated person who needs to access the directory. After you create these accounts, we recommend that you set the account credentials for the Admin to a random password and store it for break-glass scenarios. Do not use shared or generic accounts such as the Admin account for standard administration. Otherwise, it will be difficult to audit the directory.

Machine-to-machine identity management

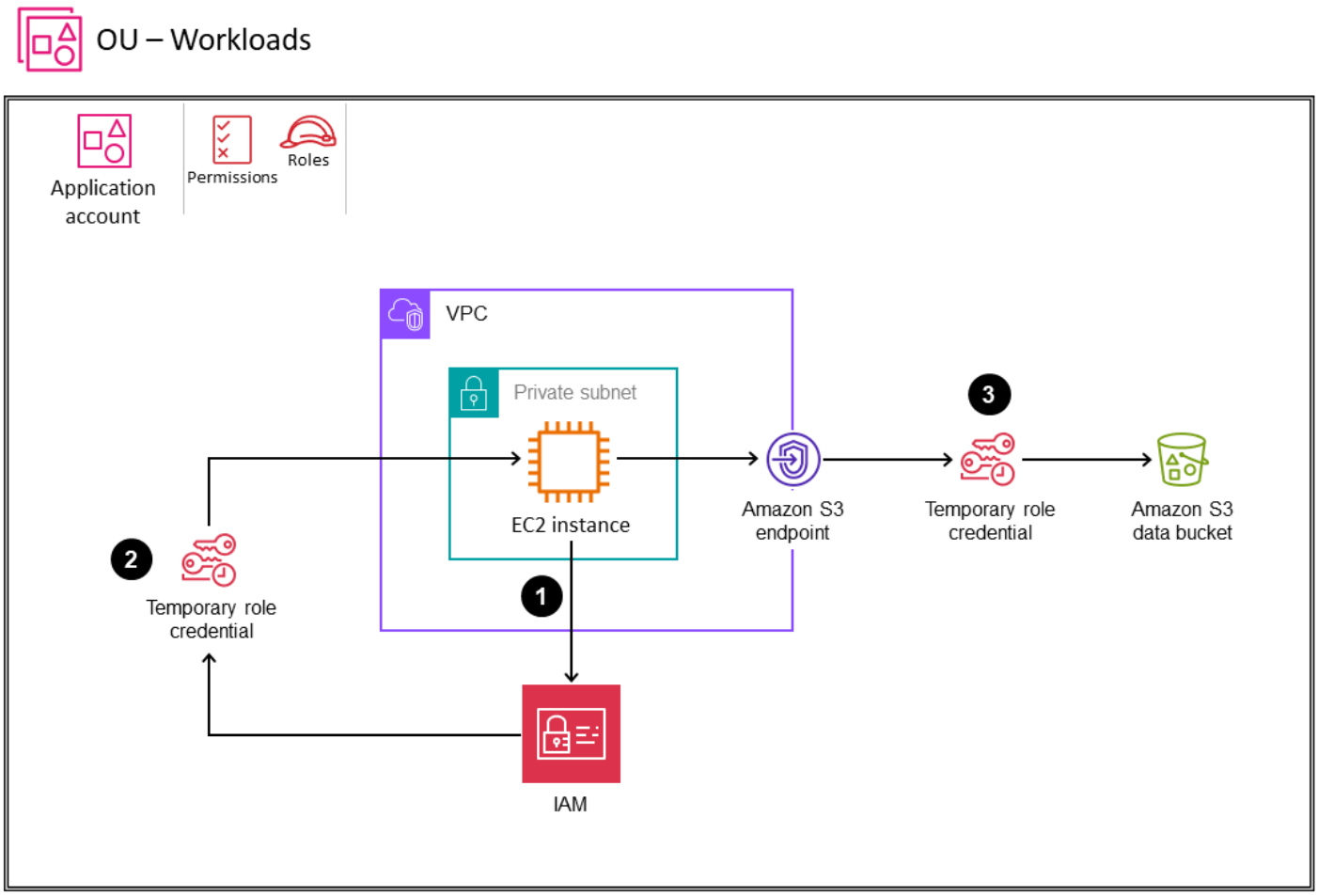
Influence the future of the AWS Security Reference Architecture (AWS SRA) by taking a short survey.

Machine-to-machine (M2M) authentication enables services and applications that run on AWS to securely communicate with one another to access resources and data. Instead of using long-term static credentials, machine authentication systems issue temporary credentials or tokens to identify trusted machines. They allow precise control over which machines can access specific parts of the environment without human intervention. Well-designed machine authentication helps improve your security posture by limiting broad credential exposure, enabling dynamic revocation of permissions, and simplifying credential rotation. Typical methods for machine authentication include EC2 instance profiles, the Amazon Cognito client credentials grant, mutually authenticated TLS (mTLS) connections, and IAM Roles Anywhere. This section provides guidance on implementing secure and scalable M2M authentication flows on AWS.

EC2 instance profiles

For scenarios where you have an application or service running on Amazon Elastic Compute Cloud (Amazon EC2) that needs to call AWS APIs, consider using EC2 instance profiles. Instance profiles allow applications that run on EC2 instances to securely access other AWS services without requiring static, long-lived IAM access keys. Instead, you should assign an IAM role to your instance to provide the required permissions through the instance profile. The EC2 instance can then automatically obtain temporary security credentials from the instance profile to access other AWS services.

The following diagram illustrates this scenario.



1. An application on the EC2 instance that needs to call an AWS API retrieves the security credentials provided by the role from the instance metadata item `iam/security-credentials/<role-name>`.
2. The application receives the `AccessKeyId`, `SecretAccessKey`, and a secret token that can be used to sign AWS API requests.
3. The application calls an AWS API. If the role permits the API action, the request is successful.

To learn more about using temporary credentials with AWS resources, see Use temporary credentials with AWS resources in the IAM documentation.

Benefits:

- **Improved security**. This method avoids the distribution of long-term credentials to EC2 instances. Credentials are provided temporarily through the instance profile.

- **Easy integration**. Applications that run on the instance can automatically obtain credentials without any additional coding or configuration. The AWS SDKs automatically use the instance profile credentials.
- **Dynamic permissions**. You can change the permissions that are available to the instance by updating the IAM role that's assigned to the instance profile. New credentials that reflect the updated permissions are automatically obtained.
- **Rotation**. AWS automatically rotates the temporary credentials to reduce the risk from compromised credentials.
- **Revocation**. You can revoke the credentials immediately by removing the role assignment from the instance profile.

Design considerations

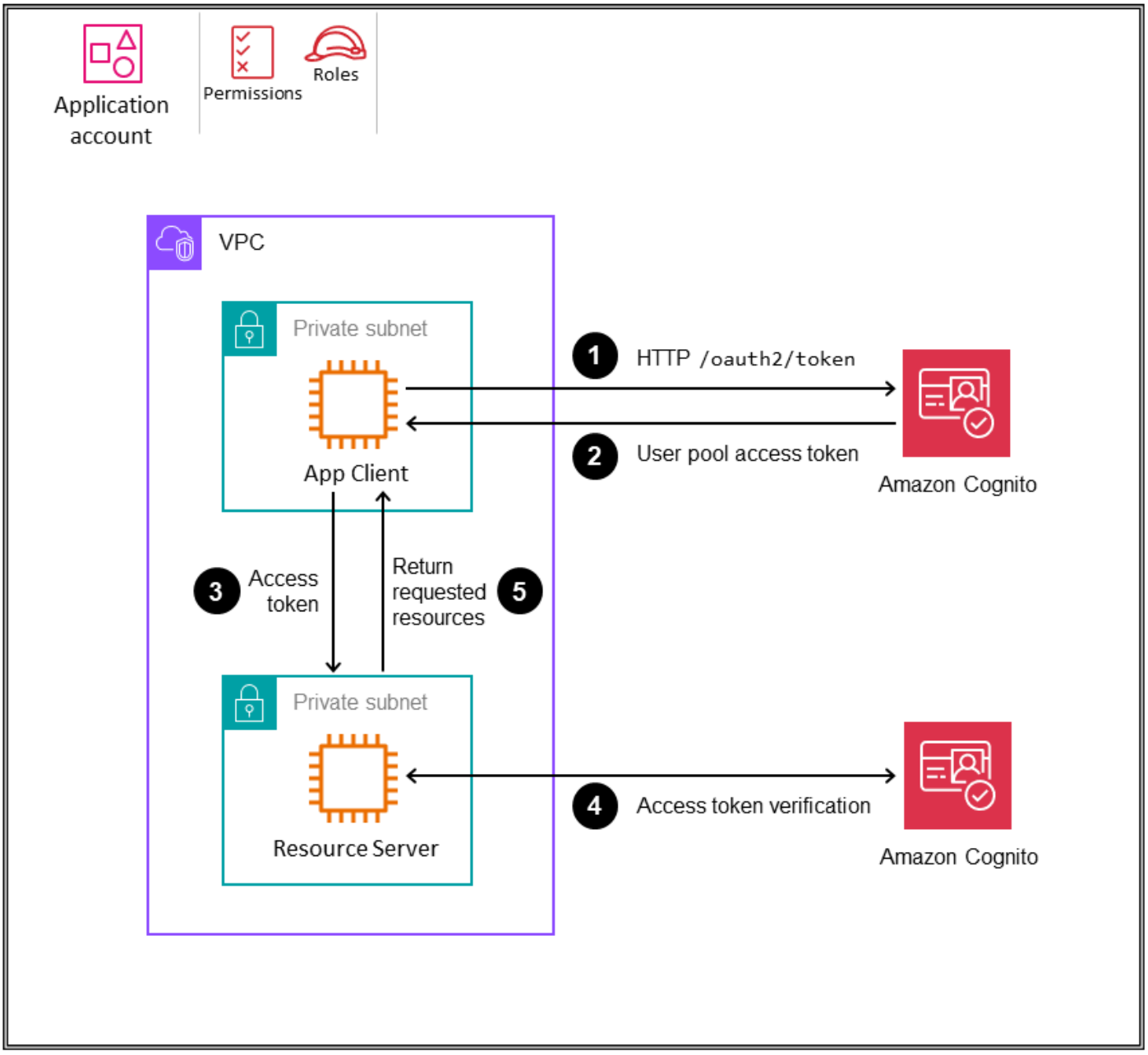
- An EC2 instance can have only one attached instance profile.
- Use least privilege IAM roles. Assign only the permissions that your application requires to the IAM role for the instance profile. Start with minimum permissions and add more permissions later if needed.
- Use IAM conditions in the role policy to restrict permissions based on tags, IP address ranges, time of day, and so on. This limits the services and resources the application can access.
- Consider how many instance profiles you require. All applications that run on an EC2 instance share the same profile and have the same AWS permissions. You can apply the same instance profile to multiple EC2 instances, so you can reduce administrative overhead by reusing instance profiles where appropriate.
- Monitor activity. Use tools such as AWS CloudTrail to monitor API calls that use the instance profile credentials. Watch for unusual activity that could indicate compromised credentials.
- Delete unneeded credentials. Remove role assignments from unused instance profiles to prevent the use of credentials. You can use IAM access advisor to identify unused roles.
- Use the PassRole permission to restrict which role a user can pass to an EC2 instance when they launch the instance. This prevents the user from running applications that have more permissions than the user has been granted.

- If your architecture spans multiple AWS accounts, consider how EC2 instances in one account might need to access resources in another account. Use cross-account roles appropriately to ensure secure access without having to embed long-term AWS security credentials.
- To manage instance profiles at scale, you can use one of these options:
  - Use AWS Systems Manager Automation runbooks to automate the association of instance profiles to EC2 instances. This can be done at launch time, or after an instance is running.
  - Use AWS CloudFormation to apply instance profiles to EC2 instances programmatically at creation time, instead of configuring them through the AWS console.
- It's good practice to use VPC endpoints to privately connect to supported AWS services such as Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB from applications that run on EC2 instances.

Amazon Cognito client credentials grant

Amazon Cognito is a managed customer identity and access management service. Amazon Cognito provides OAuth-compliant authentication flows, including the ability to authenticate machines or applications instead of users through the client credentials grant type. This grant allows an application to directly retrieve temporary AWS credentials to access AWS services. Amazon Cognito client credentials are a secure way to provide AWS permissions to applications without human user interaction. Applications present their client ID and client secret to the Amazon Cognito token endpoint. In return, they receive an access token, which they can use to authenticate subsequent requests to various resources and services. The scope of this access is dictated by the permissions that are associated with the client ID. The application that receives the request must validate the token by checking its signature, expiration timestamp, and audience. After these checks, the application verifies that the requested action is allowed by validating the claims in the token.

The following diagram illustrates this method.



1. The application (App Client) that wants to request resources from a server (Resource Server) requests a token from Amazon Cognito.
2. Amazon Cognito user pools return an access token.
3. App Client sends a request to Resource Server and includes the access token.

4. Resource Server validates the token with Amazon Cognito.
5. If validation is successful and the requested action is allowed, Resource Server responds with the requested resource.

Benefits:

- **Machine authentication**. This method doesn't require user context or logins. The application authenticates directly with tokens.
- **Short-term credentials**. Applications can obtain an access token first from Amazon Cognito and then use the time-bound access token to access data from the resource server.
- **OAuth2 support**. This method reduces inconsistencies and helps with application development because it follows the established OAuth2 standard.
- **Enhanced security**. Using the client credentials grant provides enhanced security, because the client ID and client secret aren't transferred to the resource server, unlike an API key authorization mechanism. The client ID and secret are shared and used only when making calls to Amazon Cognito to get time-bound access tokens.
- **Fine-grained access control through scopes**. The application can define and request scopes and additional claims to limit access to only specific resources.
- **Audit trail**. You can use the information collected by CloudTrail to determine the request that was made to Amazon Cognito, the IP address from which the request was made, who made the request, when it was made, and additional details.

Design considerations

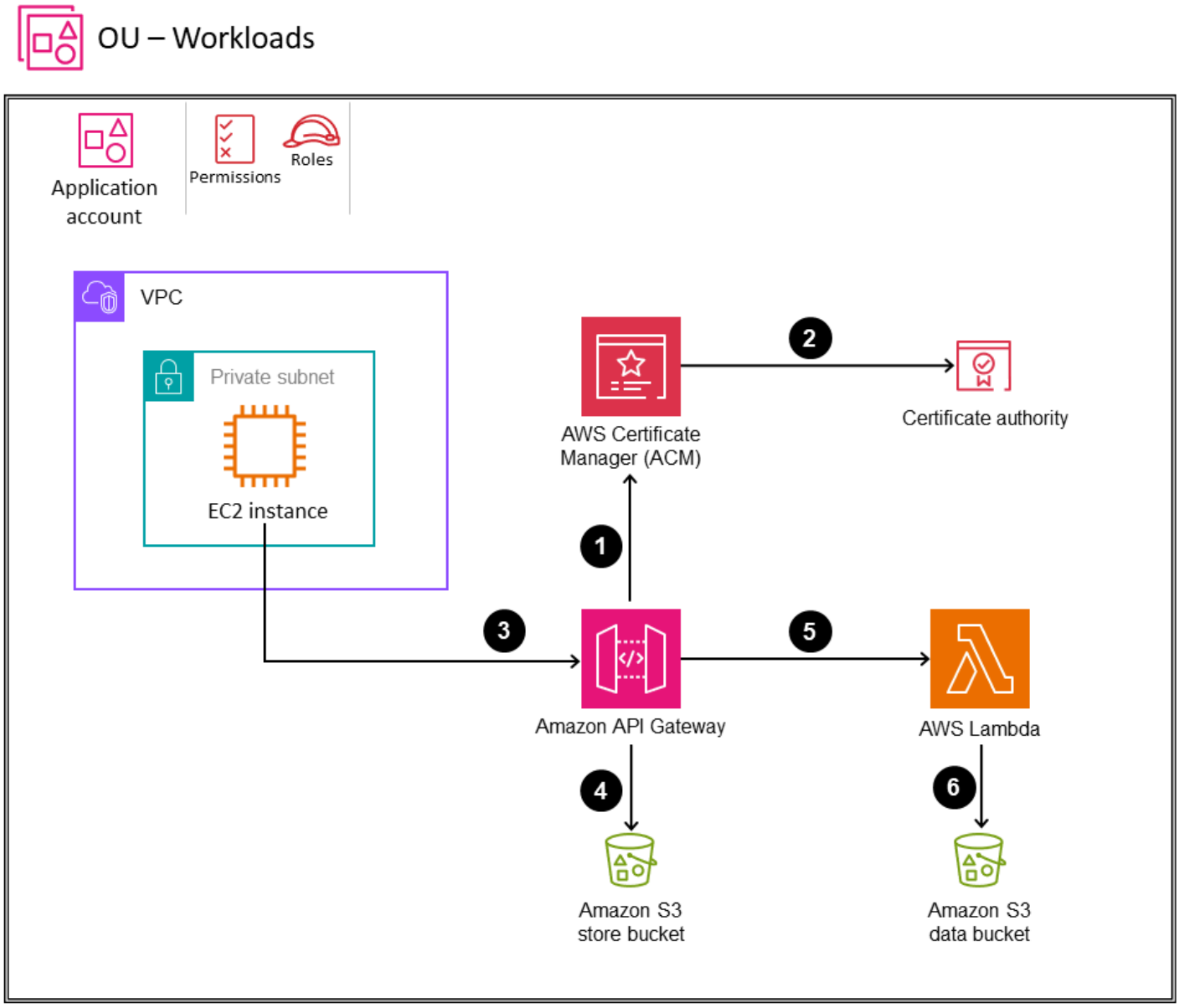
- Carefully define and constrain the scope of access for each client ID to the minimum required. Tight scopes help reduce potential vulnerabilities and ensure that services have access only to necessary resources.
- Protect client IDs and secrets by using secure storage services such as AWS Secrets Manager to store credentials. Do not check credentials into source code.
- Monitor and audit token requests and usage with tools such as CloudTrail and CloudWatch. Watch for unexpected activity patterns that could indicate issues.
- Automate the rotation of client secrets on a regular schedule. With each rotation, create a new application client, delete the old client, and update the client ID and secret. Facilitate these rotations without disrupting service communications.

- Enforce rate limits on token endpoint requests to help prevent abuse and denial of service (DoS) attacks.
- Have a strategy ready for revoking tokens in the event of a security breach. Although tokens are short-lived, compromised tokens should be invalidated immediately.
- Use AWS CloudFormation to programmatically create Amazon Cognito user pools and the application clients that represent the machines that need to authenticate to other services.
- Where appropriate, cache tokens to provide performance efficiency and cost optimization.
- Ensure that the expiration of access tokens aligns with your organization's security posture.
- If you use a custom resource server, always verify the access token to ensure that the signature is valid, the token hasn't expired, and the correct scopes are present. Verify any additional claims as needed.
- To manage client credentials at scale, you can use one of these options:
  - Centralize the management of all client credentials in a single centralized Amazon Cognito instance. This can reduce the management overhead of multiple Amazon Cognito instances, and can make configuration and auditing simpler. However, make sure to plan for scale and consider the Amazon Cognito service quotas.
  - Federate the responsibility for client credentials to workload accounts and allow multiple Amazon Cognito instances. This option promotes flexibility but can increase overhead and overall complexity compared with the centralized option.

mTLS authentication

Mutual TLS (mTLS) authentication is a mechanism that allows both the client and the server to authenticate to each other before they communicate by using certificates with TLS. Common use cases for mTLS include industries with high regulations, Internet of Things (IoT) applications, and business-to-business (B2B) applications. Amazon API Gateway currently supports mTLS in addition to its existing authorization options. You can enable mTLS on custom domains to authenticate against Regional REST and HTTP APIs. Requests can be authorized by using Bearer, JSON Web Tokens (JWTs), or sign requests with IAM-based authorization.

The following diagram shows the mTLS authentication flow for an application that's running on an EC2 instance and an API that's set up on API Gateway.



1. API Gateway requests a publicly trusted certificate directly from AWS Certificate Manager (ACM).
2. ACM generates the certificate from its certificate authority (CA).
3. The client that calls the API presents a certificate with the API request.
4. API Gateway checks the Amazon S3 trust store bucket that you have created. This bucket contains the X.509 certificates that you trust to access your API. For API Gateway to proceed with the request, the certificate's issuer and the complete chain of trust up to the root CA certificate must be in your trust store.

5. If the clients' certificate is trusted, API Gateway approves the request and calls the method.
6. The associated API action (in this case, an AWS Lambda function) processes the request and returns a response that is sent to the requestor.

Benefits

- **M2M authentication**. Services authenticate one another directly instead of using shared secrets or tokens. This removes the need to store and manage static credentials.
- **Tamper protection**. TLS encryption protects data in transit between services. Communications cannot be read or altered by third parties.
- **Easy integration**. mTLS support is built into major programming languages and frameworks. Services can enable mTLS with minimal code changes.
- **Granular permissions**. Services trust only specific certificates, which allows fine-grained control over permitted callers.
- **Revocation**. Compromised certificates can be revoked immediately so they are no longer trusted, preventing further access.

Design considerations

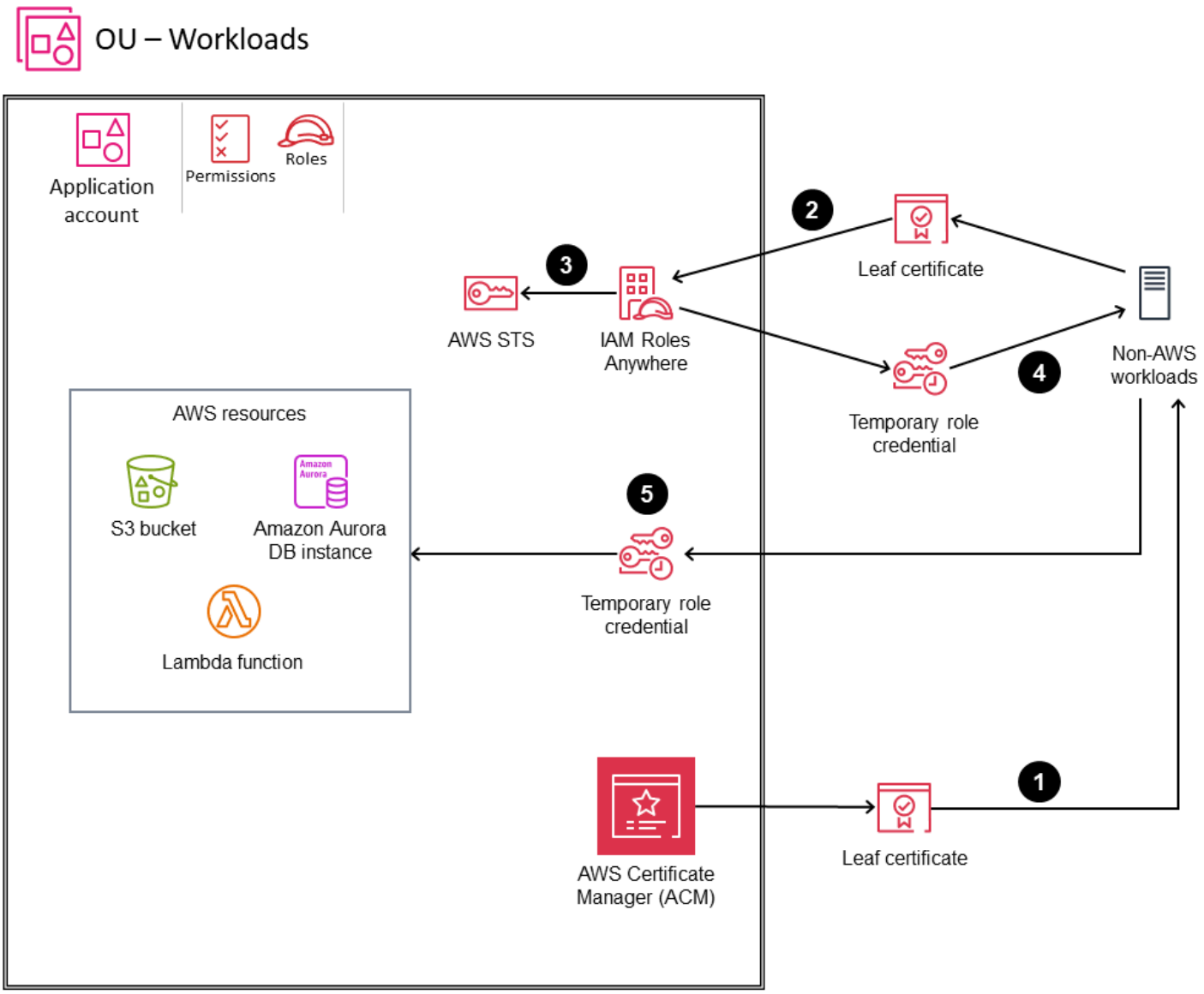
- When you use API Gateway:
  - By default, clients can call your API by using the `execute-api` endpoint that API Gateway generates for your API. To ensure that clients can access your API only by using a custom domain name with mTLS, disable this default endpoint. To learn more, see Disable the default endpoint for REST APIs in the API Gateway documentation.
  - API Gateway doesn't verify whether certificates have been revoked.
  - To configure mTLS for a REST API, you must use a Regional custom domain name for your API, with a minimum TLS version of 1.2. mTLS isn't supported for private APIs.
- You can issue certificates for API Gateway from your own CA or import them from AWS Private Certificate Authority.
- Create processes to securely issue, distribute, renew, and revoke service certificates. Automate issuance and renewal where possible. If one side of your M2M communication is an API gateway, you can integrate with AWS Private CA.

- Safeguard access to the private CA. Compromising the CA compromises trust in all certificates it issued.
- Store private keys securely and separately from certificates. Rotate keys periodically to limit impact if compromised.
- Revoke certificates immediately when they're no longer needed or if they're compromised. Distribute certificate revocation lists to services.
- Where possible, issue certificates that are intended for only specific purposes or resources to limit their utility if they're compromised.
- Have contingency plans for certificate expirations and outages of the CA or certificate revocation list (CRL) infrastructure.
- Monitor your system for certificate failures and outages. Watch for spikes in failures that could indicate issues.
- If you are using ACM with AWS Private CA, you can use AWS CloudFormation to programmatically request public and private certificates.
- If you are using ACM, use AWS Resource Access Manager (AWS RAM) to share the certificate from a security account to the workload account.

IAM Roles Anywhere

We recommend that you use IAM Roles Anywhere for M2M identity management when machines or systems need to connect to AWS services but do not support IAM roles. IAM Roles Anywhere is an extension of IAM that uses public key infrastructure (PKI) to grant access to workloads by using temporary security credentials. You can use X.509 certificates, which can be issued either through a CA or by AWS Private CA, to establish a trust anchor between the CA and IAM Roles Anywhere. As with IAM roles, the workload can access AWS services based on its permission policy, which is attached to the role.

The following diagram shows how you can use IAM Roles Anywhere to connect AWS with external resources.



1. You create a trust anchor to establish trust between your AWS account and the CA that issues certificates to your on-premises workloads. The certificates are issued by a CA that you register as a trust anchor (root of trust) in IAM Roles Anywhere. The CA can be part of your existing public key infrastructure (PKI) system, or it can be a CA that you created with AWS Private CA and manage with ACM. In this example, we are using ACM.
2. Your application makes an authentication request to IAM Roles Anywhere, and sends its public key (encoded in a certificate) and a signature signed by the corresponding private key. Your application also specifies the role to assume in the request.
3. When IAM Roles Anywhere receives the request, it first validates the signature with the public key, and then validates that the certificate was issued by a trust anchor. After both validations

succeed, your application is authenticated and IAM Roles Anywhere creates a new role session for the role specified in the request by calling AWS Security Token Service (AWS STS).

4. You use the credential helper tool that IAM Roles Anywhere provides to manage the process of creating a signature with the certificate and to call the endpoint to obtain session credentials. The tool returns the credentials to the calling process in a standard JSON format.
5. By using this bridged trust model between IAM and PKI, on-premises workloads use these temporary credentials (access key, secret key, and session token) to assume the IAM role to interact with AWS resources without needing long-term credentials. You can also configure these credentials by using the AWS CLI or AWS SDKs.

Benefits:

- **No permanent credentials**. Applications don't need long-term AWS access keys with broad permissions.
- **Fine-grained access**. Policies determine which IAM role can be assumed for a specific entity.
- **Context-aware roles**. The role can be customized based on the details of the authenticated entity.
- **Revocation**. Revoking trust permissions immediately blocks an entity from assuming a role.

Design considerations

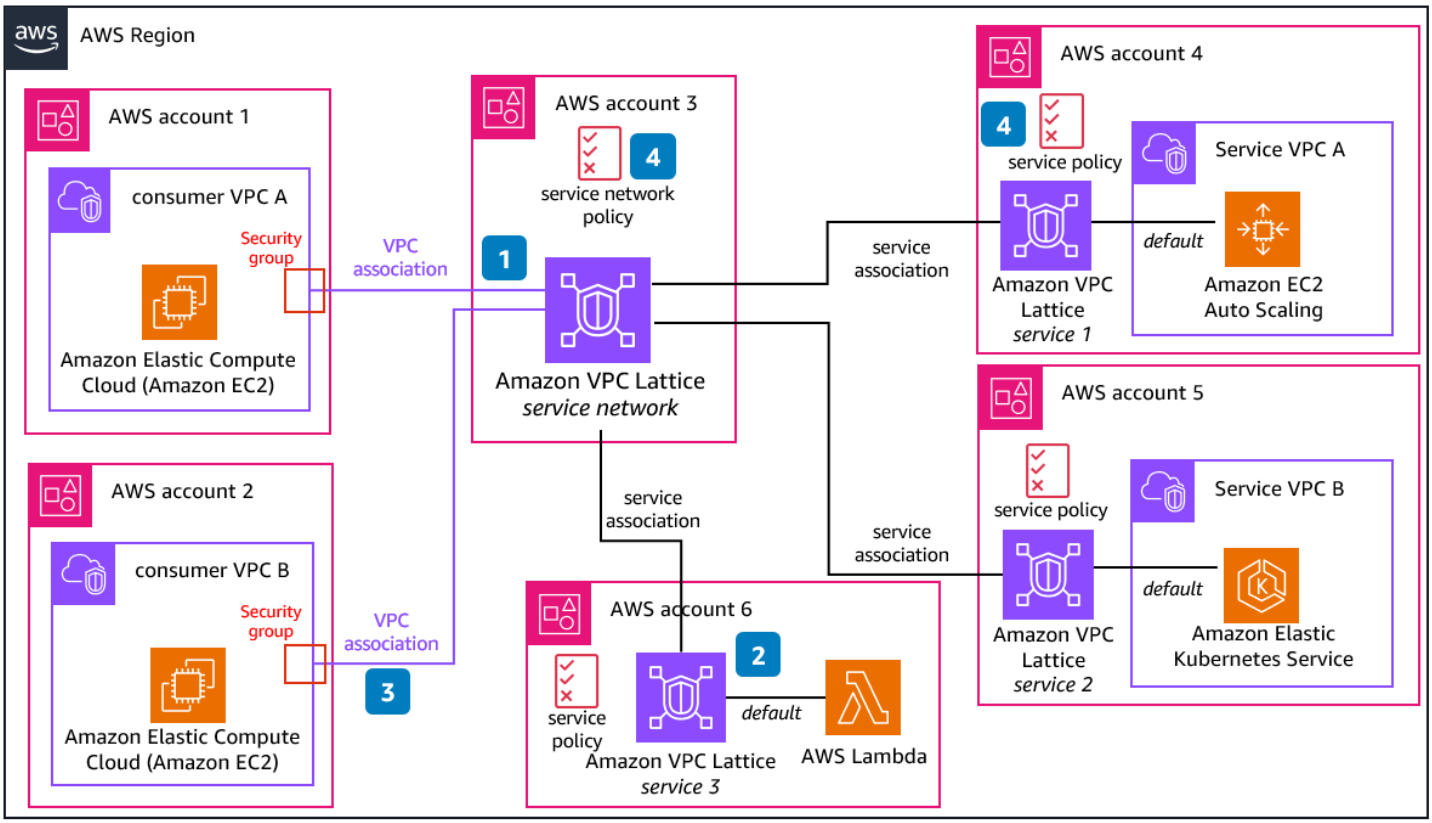
- Servers must be able to support certificate-based authentication.
- It's good practice to lock down the trust policy to use `aws:SourceArn` or `aws:SourceAccount` for the account where the trust anchor has been configured.
- Principal tags are carried forward from the certificate details. These include the common name (CN), the subject alternative name (SAN), the subject, and the issuer.
- If you are using ACM, use AWS RAM to share the certificate from a security account to the workload account.
- Use operating system (OS) file system permissions to restrict read access to the owning user.
- Never check keys into source control. Store them separately from source code to reduce the risk of accidentally including them in a change set. If possible, consider using a secure storage mechanism.

- Make sure that you have a process to rotate and revoke certificates.

Amazon VPC Lattice

For scenarios where you would like to connect multiple applications or services that run across the same or different compute platforms—such as EC2 instances, Lambda functions, or even Kubernetes pods—without increasing networking complexity, consider Amazon VPC Lattice. This application networking service connects, monitors, and secures service-to-service communications. A service, often called a *microservice*, is an independently deployable unit of software that delivers a specific task. VPC Lattice automatically manages network connectivity and application-layer routing between services across VPCs and AWS accounts without requiring you to manage the underlying network connectivity, frontend load balancers, or sidecar proxies.

The following diagram shows an example of a VPC Lattice service network, which comprises one or more VPC Lattice services. The services are part of a service directory, which is a list of all VPC Lattice services you create locally within an AWS account together with any VPC Lattice services that are shared with your account by using AWS RAM.



1. A service network is a logical boundary for a collection of services. Services that are associated with the network can be authorized for discovery, connectivity, accessibility, and observability. To make requests to services in the network, the client must be in a VPC that is associated with the service network.
2. A service represents an independently deployable unit of software that delivers a specific task or function. Each service has a listener that uses rules to target to one or several target groups. Targets can be EC2 instances, IP addresses, Lambda functions, Application Load Balancers, or Kubernetes pods.
3. Associating a service with a service network enables clients to make requests to the service, but only if the VPC where the client is located is also associated with the service network, and the policies allow it.
4. Associating a VPC with the service network enables all the targets within that VPC to be clients and communicate with other services in the service network. A security group can be attached to this association to control the network access from the VPC, and service network or service policies can be used to apply fine-grained access controls.

Authentication and authorization are enforced by using auth policies, which are IAM policy documents that are attached to service networks (for coarse-grained controls) or individual services (for fine-grained controls) to control principal access to services.

After services are associated with the service network, they can begin interacting without any networking changes required to enable the communications. This helps reduce the overhead of complex networking.

Benefits:

- **Improved security**. Creates an improved and more consistent security posture with reliable authentication and context-specific authorization by using IAM.
- **Simplified connectivity**. Using VPC Lattice to discover and securely connect services and resources across VPCs and accounts helps simplify and automate service and resource connectivity.
- **Connecting compute platforms**. You can connect platforms such as EC2 instances, Lambda functions, and Amazon Elastic Kubernetes Service (Amazon EKS services to a single service network.
- **Scalability**. You can scale compute and network resources automatically to support high-bandwidth HTTP, HTTPS, gRPC, and TCP workloads.

- **Connecting TCP resources**. You can connect to TCP resources such as Amazon Relational Database Service (Amazon RDS) databases, domain names, and IP addresses across multiple VPCs and accounts.

Design considerations

- Plan your service topology carefully, evaluate which VPCs need to be connected to the network, and identify areas where dedicated service networks are required for isolation. Design traffic routing rules and weights, plan health check configurations, and consider circuit breakers.
- Consider external connectivity patterns such as hybrid and cross-Region access.
- Design authentication and authorization policies by using IAM constructs at the network and endpoint level based on your security requirements.
- For operational aspects such as deployment automation and procedures for introducing changes to networks and services, consider how services will be discovered by clients.
- To optimize costs, evaluate pricing based on the number of services and networks. Consider costs for Availability Zone traffic, and optimize the number of service endpoints.
- Consider service quotas.

Customer identity management

Influence the future of the AWS Security Reference Architecture (AWS SRA) by taking a short survey.

Customer identity and access management (CIAM) is a technology that allows organizations to manage customer identities. It provides security and an enhanced user experience for signing up, signing in, and accessing consumer applications, web portals, or digital services offered by an organization. CIAM helps you identify your customers, create personalized experiences, and determine the correct access they need for customer-facing applications and services. A CIAM solution can also help an organization meet compliance mandates across industry regulatory standards and frameworks. For more information, see What is CIAM? on the AWS website.

Amazon Cognito is an identity service for web and mobile applications that provides CIAM capabilities to businesses of any scale. Amazon Cognito includes a user directory, an authentication server, and an authorization service for OAuth 2.0 access tokens, and can also provide temporary AWS credentials. You can use Amazon Cognito to authenticate and authorize users from the built-in user directory, from a federated identity provider such as your enterprise directory, or from social identity providers such as Google and Facebook.

The two main components of Amazon Cognito are user pools and identity pools. User pools are user directories that provide sign-up and sign-in options for your web and mobile application users. Identity pools provide temporary AWS credentials to grant your users access to other AWS services.

When to use Amazon Cognito

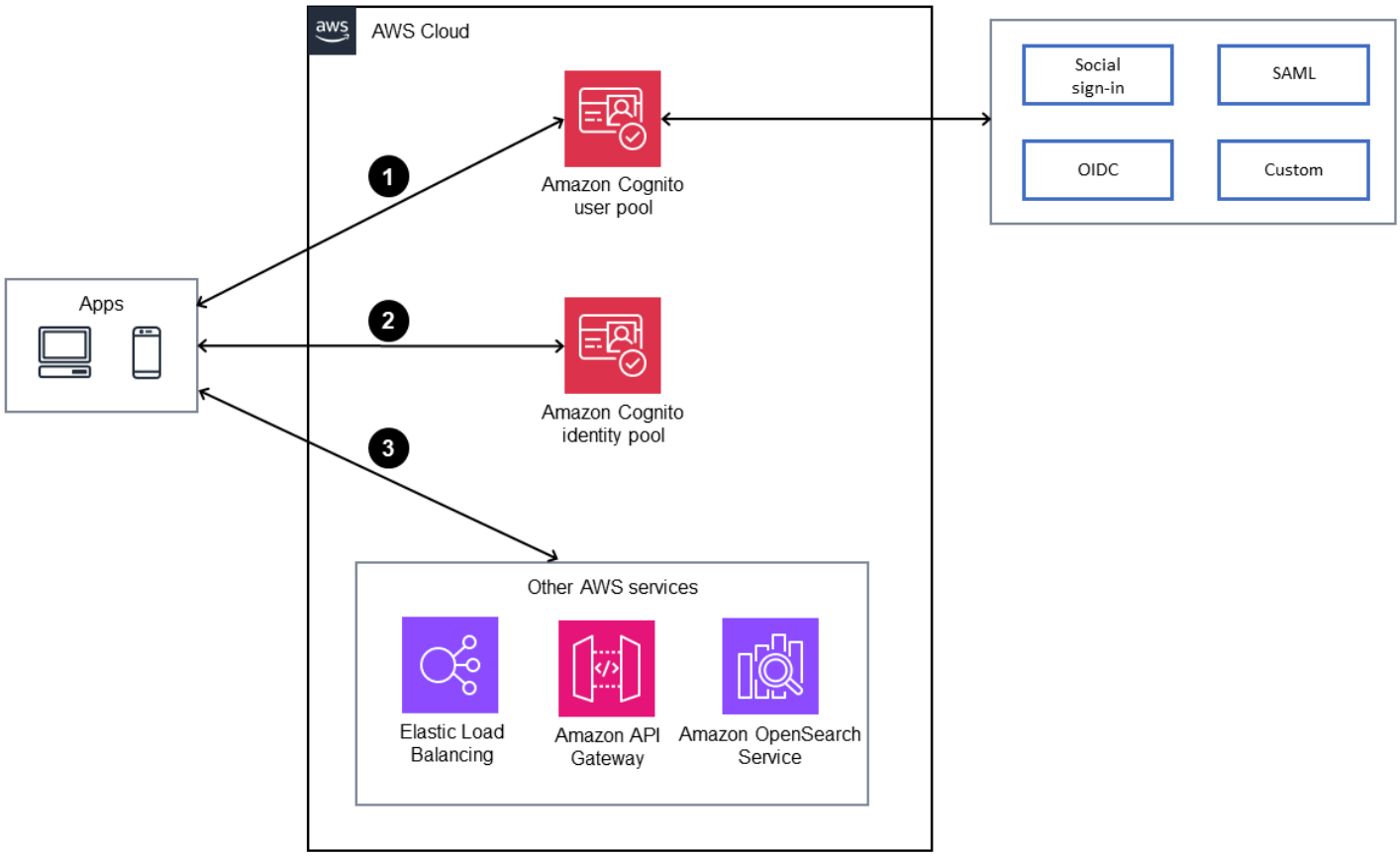
Amazon Cognito is a good choice when you require a secure and cost-effective user management solution for your web and mobile applications. Here are some scenarios where you might decide to use Amazon Cognito:

- **Authentication**. If you're prototyping an application or want to implement user login functionality quickly, you can use user pools and the hosted UI in Amazon Cognito to speed up development. You can focus on your core application features while Amazon Cognito handles user sign-up, sign-in, and security.

Amazon Cognito supports various authentication methods, including usernames and passwords, social identity providers. and enterprise identity providers through SAML and OpenID Connect (OIDC).

- **User management**. Amazon Cognito supports user management, including user registration, verification, and account recovery. Users can sign up and sign in with their preferred identity provider, and you can customize the registration process according to your application's requirements.
- **Secure access to AWS resources**. Amazon Cognito integrates with IAM to provide fine-grained access control to AWS resources. You can define IAM roles and policies to control access to AWS services based on user identity and group membership.
- **Federated identity**. Amazon Cognito supports federated identity, which allows a user to sign in by using their existing social or enterprise identities. This eliminates the need for users to create new credentials for your application, so it enhances the user experience and reduces friction during the sign-up process.
- **Mobile and web applications**. Amazon Cognito is well-suited for both mobile and web applications. It provides SDKs for various platforms, and makes it easy to integrate authentication and access control into your application code. It supports offline access and synchronization for mobile applications, so users can access their data even when they're offline.
- **Scalability**. Amazon Cognito is a highly available and fully managed service that can scale to millions of users. It processes more than 100 billion authentications per month.
- **Security**. Amazon Cognito has several built-in security features, such as encryption of sensitive data, multi-factor authentication (MFA), and protection against common web attacks such as cross-site scripting (XSS) and cross-site request forgery (CSRF). Amazon Cognito also provides advanced security features such as adaptive authentication, checking for usage of compromised credentials, and access token customization.
- **Integration with existing AWS services**. Amazon Cognito integrates seamlessly with AWS services. This can simplify development and streamline user management for functionality that relies on AWS resources.

The following diagram illustrates some of these scenarios.



1. The application authenticates with Amazon Cognito user pools and gets tokens.
2. The application uses Amazon Cognito identity pools to exchange tokens for AWS credentials.
3. The application accesses AWS services with credentials.

We recommend that you use Amazon Cognito whenever you need to add user authentication, authorization, and user management capabilities to your web or mobile applications, especially when you have multiple identity providers, require secure access to AWS resources, and have scalability requirements.

Design considerations

- Create an Amazon Cognito user pool or identity pool based on your requirements.
- Don't update the user profile too frequently (for example, with every sign-in request). If an update is required, store the updated attributes in an external database such as Amazon DynamoDB.

- Do not use Amazon Cognito workforce identity management.
- Your application should always validate JSON web tokens (JWTs) before trusting them by verifying their signature and validity. This validation should be done on the client side without sending API calls to the user pool. After the token is verified, you can trust the claims in the token and use them instead of making additional `getUser` API calls. For more information, see Verifying JSON web tokens in the Amazon Cognito documentation. You can also use additional JWT libraries for token verification.
- Enable the advanced security features in Amazon Cognito only if you aren't using a `CUSTOM_AUTH` flow, AWS Lambda triggers for custom authentication challenges, or federated sign-in. For considerations and limitations around advanced security features, see the Amazon Cognito documentation.
- Enable AWS WAF to protect Amazon Cognito user pools by using rate-based rules and combining multiple request parameters. For more information, see the AWS blog post Protect your Amazon Cognito user pool with AWS WAF.
- If you want an extra layer of protection, use an Amazon CloudFront proxy for additional processing and validation of incoming requests, as explained in the AWS blog post Protect public clients for Amazon Cognito by using an Amazon CloudFront proxy.
- All API calls after user sign-in should be made from backend services. For example, use AWS WAF to deny calls to `UpdateUserAttribute`, but then call `AdminUpdateUserAttribute` from the application backend instead, to update the user attribute.
- When you create a user pool, you choose how users will sign in—for example, with a username, email address, or phone number. This configuration cannot be changed after the user pool is created. Similarly, custom attributes cannot be changed or removed after they are added to the user pool.
- We recommend that you enable multi-factor authentication (MFA) in your user pool.
- Amazon Cognito doesn't currently provide built-in backup or export functions. To back up or export your users' data, you can use the Amazon Cognito profiles export reference architecture.
- Use IAM roles for general access to AWS resources. For fine-grained authorization requirements, use Amazon Verified Permissions. This permission management service natively integrates with Amazon Cognito. You can also use access token customization to enrich application-specific claims in order to determine the level of access and content available to the user. If your application uses Amazon API Gateway as an entry point, use

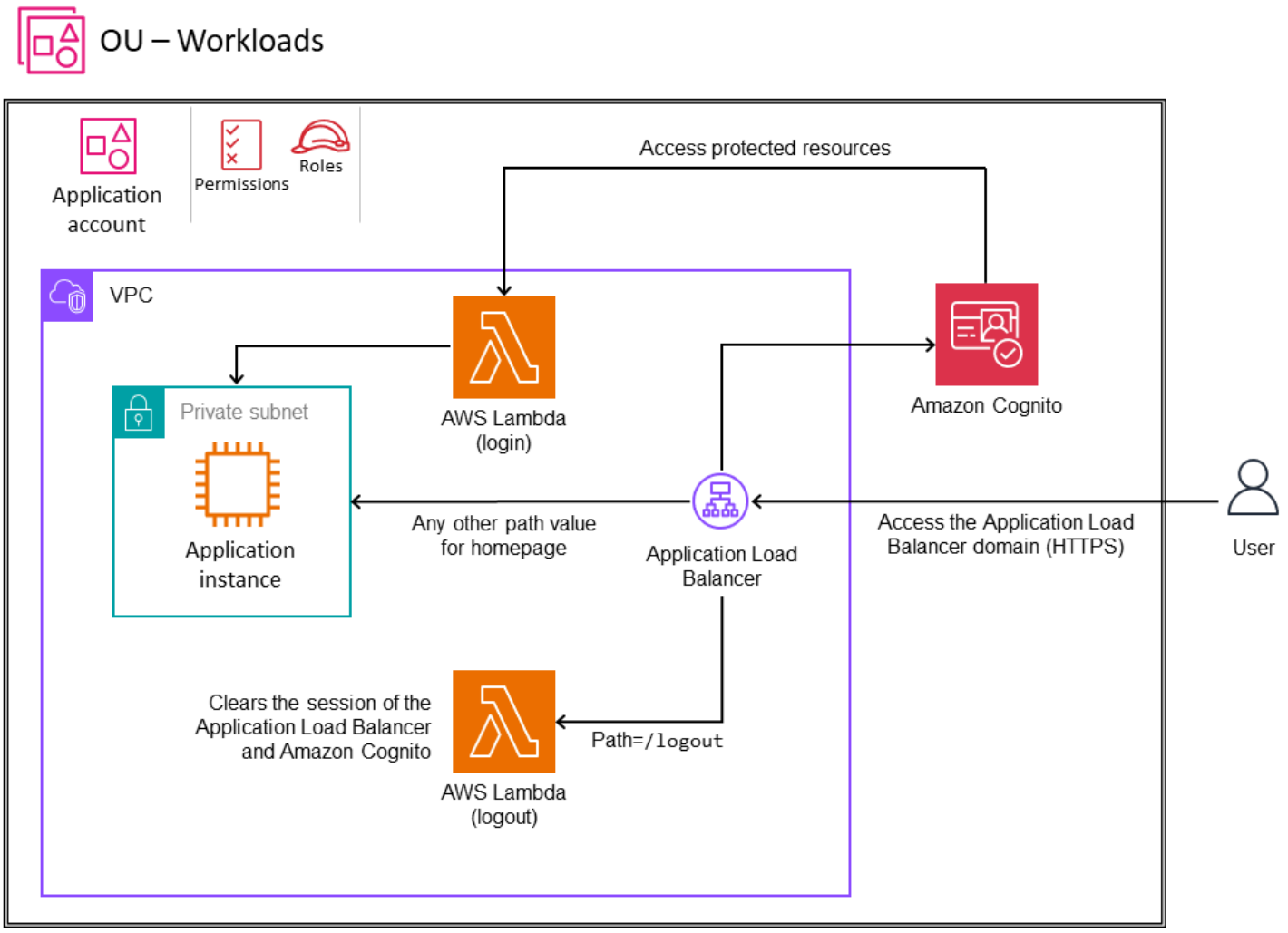
the Amazon Cognito feature to secure API Gateway by using Verified Permissions. This service manages and evaluates granular security policies that reference user attributes and groups.You can ensure that only users in authorized Amazon Cognito groups have access to the application's APIs. For more information, see the article Protect API Gateway with Amazon Verified Permissions in AWS Builder Center.

- Use AWS SDKs to access user data from the backend by calling and retrieving user attributes, statuses, and group information. You can store custom app data in Amazon Cognito user attributes and keep it synchronized across devices.

The following sections discuss three patterns for integrating Amazon Cognito with other AWS services: Application Load Balancers, API Gateway, and Amazon OpenSearch Service.

Integration with an Application Load Balancer

You can configure an Application Load Balancer with Amazon Cognito to authenticate application users, as illustrated in the following diagram.

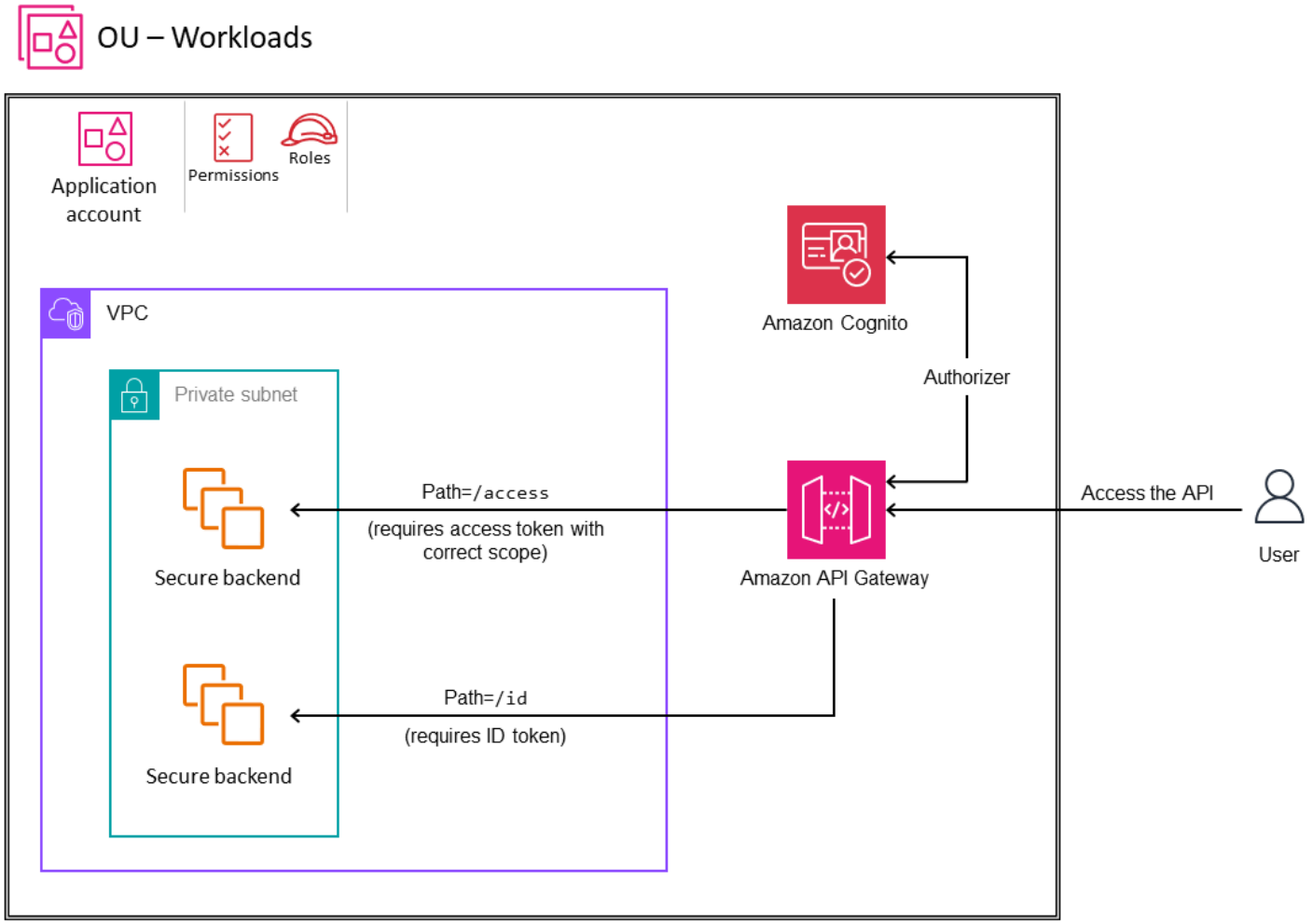


By configuring the HTTPS listener default rule, you can offload user identification to the Application Load Balancer and create an automatic authentication process. For details, see How do I set up an Application Load Balancer to authenticate users through an Amazon Cognito user pool in the AWS Knowledge Center. If your application is hosted on Kubernetes, see the AWS blog post How to use Application Load Balancer and Amazon Cognito to authenticate users for your Kubernetes web apps.

Integration with Amazon API Gateway

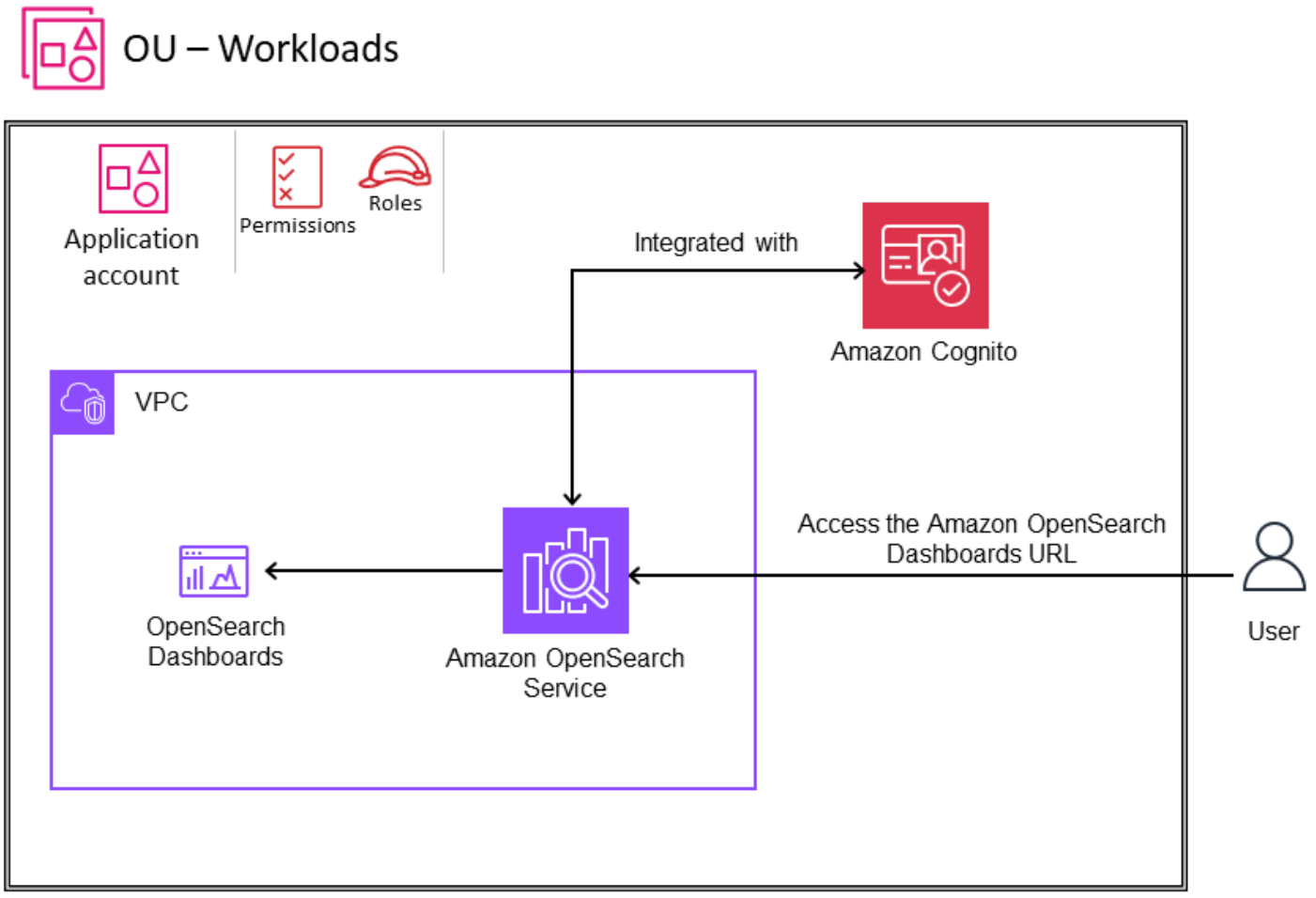
Amazon API Gateway is a fully managed, cloud-based API gateway service that makes it easy to create, publish and manage APIs at scale. It is an entry point for user traffic into the backend services. You can integrate Amazon Cognito with API Gateway to implement authentication and access control, either to protect the APIs from misuse or for any other security or business use

case. There are two methods for securing access to API Gateway: by using an Amazon Cognito authorizer (as illustrated in the following diagram) or by using an AWS Lambda authorizer. For more information about these implementations, see How do I set up an Amazon Cognito user pool as an authorizer on an API Gateway REST API? in the AWS Knowledge Base.



Integration with Amazon OpenSearch Service

You can use Amazon Cognito to secure Amazon OpenSearch Service domains. For example, if a user might need access to OpenSearch Dashboards from the internet, as illustrated in the following diagram. In this scenario, Amazon Cognito can provide access permissions, including fine-grained permissions, by mapping Amazon Cognito groups and users to internal OpenSearch Service permissions. For more information, see Configuring Amazon Cognito authentication for OpenSearch Dashboards in the OpenSearch Service documentation.


OU – Workloads
Application account
Permissions
Roles
Integrated with
Amazon Cognito
VPC
Access the Amazon OpenSearch Dashboards URL
OpenSearch Dashboards
Amazon OpenSearch Service
User

Contributors

The following individuals contributed to this guide.

Authoring:

- Avik Mukherjee, AWS Senior Security SA
- Ashwin Phadke, AWS Senior Solutions Architect
- Sowjanya Rajavaram, AWS Senior Security SA
- Meg Peddada, AWS Senior Security Consultant
- Victor Okonyia, AWS Technical Account Manager
- Jeremy Girven, AWS Specialist SA
- Rodney Underkoffler, AWS Specialist Senior SA
- James Thompson, AWS Senior Solutions Architect
- Farhan Farooq, AWS Senior Solutions Architect
- Jonathan VanKim, AWS Principal Security SA

Reviewing:

- Eric Rose, AWS Principal Security SA
- Manoj Kumar, AWS Delivery Consultant

Technical writing:

- Handan Selamoglu, AWS Senior Technical Writer

Document history

The following table describes significant changes to this guide. If you want to be notified about future updates, you can subscribe to an RSS feed.

Change	Description	Date
Initial publication as standalone guide	Converted from a chapter in the AWS SRA – core architecture guide to an individual guide.	December 22, 2025

AWS Prescriptive Guidance glossary

The following are commonly used terms in strategies, guides, and patterns provided by AWS Prescriptive Guidance. To suggest entries, please use the **Provide feedback** link at the end of the glossary.

Numbers

7 Rs

Seven common migration strategies for moving applications to the cloud. These strategies build upon the 5 Rs that Gartner identified in 2011 and consist of the following:

- Refactor/re-architect – Move an application and modify its architecture by taking full advantage of cloud-native features to improve agility, performance, and scalability. This typically involves porting the operating system and database. Example: Migrate your on-premises Oracle database to the Amazon Aurora PostgreSQL-Compatible Edition.
- Replatform (lift and reshape) – Move an application to the cloud, and introduce some level of optimization to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Amazon Relational Database Service (Amazon RDS) for Oracle in the AWS Cloud.
- Repurchase (drop and shop) – Switch to a different product, typically by moving from a traditional license to a SaaS model. Example: Migrate your customer relationship management (CRM) system to Salesforce.com.
- Rehost (lift and shift) – Move an application to the cloud without making any changes to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Oracle on an EC2 instance in the AWS Cloud.
- Relocate (hypervisor-level lift and shift) – Move infrastructure to the cloud without purchasing new hardware, rewriting applications, or modifying your existing operations. You migrate servers from an on-premises platform to a cloud service for the same platform. Example: Migrate a Microsoft Hyper-V application to AWS.
- Retain (revisit) – Keep applications in your source environment. These might include applications that require major refactoring, and you want to postpone that work until a later time, and legacy applications that you want to retain, because there's no business justification for migrating them.

- Retire – Decommission or remove applications that are no longer needed in your source environment.

A

A2A (Agent-to-Agent)

A stateful protocol for agent-to-agent collaboration supporting task delegation and state transfer.

ABAC

See attribute-based access control.

abstracted services

See managed services.

ACID

See atomicity, consistency, isolation, durability.

active-active migration

A database migration method in which the source and target databases are kept in sync (by using a bidirectional replication tool or dual write operations), and both databases handle transactions from connecting applications during migration. This method supports migration in small, controlled batches instead of requiring a one-time cutover. It's more flexible but requires more work than active-passive migration.

active-passive migration

A database migration method in which the source and target databases are kept in sync, but only the source database handles transactions from connecting applications while data is replicated to the target database. The target database doesn't accept any transactions during migration.

Agent

An AI system that can autonomously reason, plan, and take actions using tools to achieve goals.

Agent Ops

Operational practices for building, testing, deploying, and running AI agents in production at scale.

aggregate function

A SQL function that operates on a group of rows and calculates a single return value for the group. Examples of aggregate functions include SUM and MAX.

AI

See artificial intelligence.

AIOps

See artificial intelligence operations.

anonymization

The process of permanently deleting personal information in a dataset. Anonymization can help protect personal privacy. Anonymized data is no longer considered to be personal data.

anti-pattern

A frequently used solution for a recurring issue where the solution is counter-productive, ineffective, or less effective than an alternative.

application control

A security approach that allows the use of only approved applications in order to help protect a system from malware.

application portfolio

A collection of detailed information about each application used by an organization, including the cost to build and maintain the application, and its business value. This information is key to the portfolio discovery and analysis process and helps identify and prioritize the applications to be migrated, modernized, and optimized.

artificial intelligence (AI)

The field of computer science that is dedicated to using computing technologies to perform cognitive functions that are typically associated with humans, such as learning, solving problems, and recognizing patterns. For more information, see What is Artificial Intelligence?

artificial intelligence operations (AIOps)

The process of using machine learning techniques to solve operational problems, reduce operational incidents and human intervention, and increase service quality. For more information about how AIOps is used in the AWS migration strategy, see the operations integration guide.

asymmetric encryption

An encryption algorithm that uses a pair of keys, a public key for encryption and a private key for decryption. You can share the public key because it isn't used for decryption, but access to the private key should be highly restricted.

atomicity, consistency, isolation, durability (ACID)

A set of software properties that guarantee the data validity and operational reliability of a database, even in the case of errors, power failures, or other problems.

attribute-based access control (ABAC)

The practice of creating fine-grained permissions based on user attributes, such as department, job role, and team name. For more information, see ABAC for AWS in the AWS Identity and Access Management (IAM) documentation.

authoritative data source

A location where you store the primary version of data, which is considered to be the most reliable source of information. You can copy data from the authoritative data source to other locations for the purposes of processing or modifying the data, such as anonymizing, redacting, or pseudonymizing it.

Availability Zone

A distinct location within an AWS Region that is insulated from failures in other Availability Zones and provides inexpensive, low-latency network connectivity to other Availability Zones in the same Region.

AWS Cloud Adoption Framework (AWS CAF)

A framework of guidelines and best practices from AWS to help organizations develop an efficient and effective plan to move successfully to the cloud. AWS CAF organizes guidance into six focus areas called perspectives: business, people, governance, platform, security, and operations. The business, people, and governance perspectives focus on business skills and processes; the platform, security, and operations perspectives focus on technical skills and processes. For example, the people perspective targets stakeholders who handle human resources (HR), staffing functions, and people management. For this perspective, AWS CAF provides guidance for people development, training, and communications to help ready the organization for successful cloud adoption. For more information, see the AWS CAF website and the AWS CAF whitepaper.

AWS Workload Qualification Framework (AWS WQF)

A tool that evaluates database migration workloads, recommends migration strategies, and provides work estimates. AWS WQF is included with AWS Schema Conversion Tool (AWS SCT). It analyzes database schemas and code objects, application code, dependencies, and performance characteristics, and provides assessment reports.

B

bad bot

A bot that is intended to disrupt or cause harm to individuals or organizations.

BCP

See business continuity planning.

behavior graph

A unified, interactive view of resource behavior and interactions over time. You can use a behavior graph with Amazon Detective to examine failed logon attempts, suspicious API calls, and similar actions. For more information, see Data in a behavior graph in the Detective documentation.

big-endian system

A system that stores the most significant byte first. See also endianness.

binary classification

A process that predicts a binary outcome (one of two possible classes). For example, your ML model might need to predict problems such as “Is this email spam or not spam?" or "Is this product a book or a car?"

bloom filter

A probabilistic, memory-efficient data structure that is used to test whether an element is a member of a set.

blue/green deployment

A deployment strategy where you create two separate but identical environments. You run the current application version in one environment (blue) and the new application version in the other environment (green). This strategy helps you quickly roll back with minimal impact.

bot

A software application that runs automated tasks over the internet and simulates human activity or interaction. Some bots are useful or beneficial, such as web crawlers that index information on the internet. Some other bots, known as *bad bots*, are intended to disrupt or cause harm to individuals or organizations.

botnet

Networks of bots that are infected by malware and are under the control of a single party, known as a *bot herder* or *bot operator*. Botnets are the best-known mechanism to scale bots and their impact.

branch

A contained area of a code repository. The first branch created in a repository is the *main branch*. You can create a new branch from an existing branch, and you can then develop features or fix bugs in the new branch. A branch you create to build a feature is commonly referred to as a *feature branch*. When the feature is ready for release, you merge the feature branch back into the main branch. For more information, see About branches (GitHub documentation).

break-glass access

In exceptional circumstances and through an approved process, a quick means for a user to gain access to an AWS account that they don't typically have permissions to access. For more information, see the Implement break-glass procedures indicator in the AWS Well-Architected guidance.

brownfield strategy

The existing infrastructure in your environment. When adopting a brownfield strategy for a system architecture, you design the architecture around the constraints of the current systems and infrastructure. If you are expanding the existing infrastructure, you might blend brownfield and greenfield strategies.

buffer cache

The memory area where the most frequently accessed data is stored.

business capability

What a business does to generate value (for example, sales, customer service, or marketing). Microservices architectures and development decisions can be driven by business capabilities.

For more information, see the Organized around business capabilities section of the Running containerized microservices on AWS whitepaper.

business continuity planning (BCP)

A plan that addresses the potential impact of a disruptive event, such as a large-scale migration, on operations and enables a business to resume operations quickly.

C

CAF

See AWS Cloud Adoption Framework.

canary deployment

The slow and incremental release of a version to end users. When you are confident, you deploy the new version and replace the current version in its entirety.

CCoE

See Cloud Center of Excellence.

CDC

See change data capture.

change data capture (CDC)

The process of tracking changes to a data source, such as a database table, and recording metadata about the change. You can use CDC for various purposes, such as auditing or replicating changes in a target system to maintain synchronization.

chaos engineering

Intentionally introducing failures or disruptive events to test a system's resilience. You can use AWS Fault Injection Service (AWS FIS) to perform experiments that stress your AWS workloads and evaluate their response.

CI/CD

See continuous integration and continuous delivery.

classification

A categorization process that helps generate predictions. ML models for classification problems predict a discrete value. Discrete values are always distinct from one another. For example, a model might need to evaluate whether or not there is a car in an image.

Citizen Developer

A business user who creates AI applications using no-code/low-code platforms without specialized technical skills.

client-side encryption

Encryption of data locally, before the target AWS service receives it.

Cloud Center of Excellence (CCoE)

A multi-disciplinary team that drives cloud adoption efforts across an organization, including developing cloud best practices, mobilizing resources, establishing migration timelines, and leading the organization through large-scale transformations. For more information, see the CCoE posts on the AWS Cloud Enterprise Strategy Blog.

cloud computing

The cloud technology that is typically used for remote data storage and IoT device management. Cloud computing is commonly connected to edge computing technology.

cloud operating model

In an IT organization, the operating model that is used to build, mature, and optimize one or more cloud environments. For more information, see Building your Cloud Operating Model.

cloud stages of adoption

The four phases that organizations typically go through when they migrate to the AWS Cloud:

- Project – Running a few cloud-related projects for proof of concept and learning purposes
- Foundation – Making foundational investments to scale your cloud adoption (e.g., creating a landing zone, defining a CCoE, establishing an operations model)
- Migration – Migrating individual applications
- Re-invention – Optimizing products and services, and innovating in the cloud

These stages were defined by Stephen Orban in the blog post The Journey Toward Cloud-First & the Stages of Adoption on the AWS Cloud Enterprise Strategy blog. For information about how they relate to the AWS migration strategy, see the migration readiness guide.

CMDB

See configuration management database.

code repository

A location where source code and other assets, such as documentation, samples, and scripts, are stored and updated through version control processes. Common cloud repositories include GitHub or Bitbucket Cloud. Each version of the code is called a *branch*. In a microservice structure, each repository is devoted to a single piece of functionality. A single CI/CD pipeline can use multiple repositories.

cold cache

A buffer cache that is empty, not well populated, or contains stale or irrelevant data. This affects performance because the database instance must read from the main memory or disk, which is slower than reading from the buffer cache.

cold data

Data that is rarely accessed and is typically historical. When querying this kind of data, slow queries are typically acceptable. Moving this data to lower-performing and less expensive storage tiers or classes can reduce costs.

computer vision (CV)

A field of AI that uses machine learning to analyze and extract information from visual formats such as digital images and videos. For example, Amazon SageMaker AI provides image processing algorithms for CV.

configuration drift

For a workload, a configuration change from the expected state. It might cause the workload to become noncompliant, and it's typically gradual and unintentional.

configuration management database (CMDB)

A repository that stores and manages information about a database and its IT environment, including both hardware and software components and their configurations. You typically use data from a CMDB in the portfolio discovery and analysis stage of migration.

conformance pack

A collection of AWS Config rules and remediation actions that you can assemble to customize your compliance and security checks. You can deploy a conformance pack as a single entity in

an AWS account and Region, or across an organization, by using a YAML template. For more information, see Conformance packs in the AWS Config documentation.

continuous integration and continuous delivery (CI/CD)

The process of automating the source, build, test, staging, and production stages of the software release process. CI/CD is commonly described as a pipeline. CI/CD can help you automate processes, improve productivity, improve code quality, and deliver faster. For more information, see Benefits of continuous delivery. CD can also stand for *continuous deployment*. For more information, see Continuous Delivery vs. Continuous Deployment.

CV

See computer vision.

D

data at rest

Data that is stationary in your network, such as data that is in storage.

data classification

A process for identifying and categorizing the data in your network based on its criticality and sensitivity. It is a critical component of any cybersecurity risk management strategy because it helps you determine the appropriate protection and retention controls for the data. Data classification is a component of the security pillar in the AWS Well-Architected Framework. For more information, see Data classification.

data drift

A meaningful variation between the production data and the data that was used to train an ML model, or a meaningful change in the input data over time. Data drift can reduce the overall quality, accuracy, and fairness in ML model predictions.

data in transit

Data that is actively moving through your network, such as between network resources.

data mesh

An architectural framework that provides distributed, decentralized data ownership with centralized management and governance.

data minimization

The principle of collecting and processing only the data that is strictly necessary. Practicing data minimization in the AWS Cloud can reduce privacy risks, costs, and your analytics carbon footprint.

data perimeter

A set of preventive guardrails in your AWS environment that help make sure that only trusted identities are accessing trusted resources from expected networks. For more information, see Building a data perimeter on AWS.

data preprocessing

To transform raw data into a format that is easily parsed by your ML model. Preprocessing data can mean removing certain columns or rows and addressing missing, inconsistent, or duplicate values.

data provenance

The process of tracking the origin and history of data throughout its lifecycle, such as how the data was generated, transmitted, and stored.

data subject

An individual whose data is being collected and processed.

data warehouse

A data management system that supports business intelligence, such as analytics. Data warehouses commonly contain large amounts of historical data, and they are typically used for queries and analysis.

database definition language (DDL)

Statements or commands for creating or modifying the structure of tables and objects in a database.

database manipulation language (DML)

Statements or commands for modifying (inserting, updating, and deleting) information in a database.

DDL

See database definition language.

deep ensemble

To combine multiple deep learning models for prediction. You can use deep ensembles to obtain a more accurate prediction or for estimating uncertainty in predictions.

deep learning

An ML subfield that uses multiple layers of artificial neural networks to identify mapping between input data and target variables of interest.

defense-in-depth

An information security approach in which a series of security mechanisms and controls are thoughtfully layered throughout a computer network to protect the confidentiality, integrity, and availability of the network and the data within. When you adopt this strategy on AWS, you add multiple controls at different layers of the AWS Organizations structure to help secure resources. For example, a defense-in-depth approach might combine multi-factor authentication, network segmentation, and encryption.

delegated administrator

In AWS Organizations, a compatible service can register an AWS member account to administer the organization's accounts and manage permissions for that service. This account is called the *delegated administrator* for that service. For more information and a list of compatible services, see Services that work with AWS Organizations in the AWS Organizations documentation.

deployment

The process of making an application, new features, or code fixes available in the target environment. Deployment involves implementing changes in a code base and then building and running that code base in the application's environments.

development environment

See environment.

detective control

A security control that is designed to detect, log, and alert after an event has occurred. These controls are a second line of defense, alerting you to security events that bypassed the preventative controls in place. For more information, see Detective controls in *Implementing security controls on AWS*.

development value stream mapping (DVSM)

A process used to identify and prioritize constraints that adversely affect speed and quality in a software development lifecycle. DVSM extends the value stream mapping process originally designed for lean manufacturing practices. It focuses on the steps and teams required to create and move value through the software development process.

digital twin

A virtual representation of a real-world system, such as a building, factory, industrial equipment, or production line. Digital twins support predictive maintenance, remote monitoring, and production optimization.

dimension table

In a star schema, a smaller table that contains data attributes about quantitative data in a fact table. Dimension table attributes are typically text fields or discrete numbers that behave like text. These attributes are commonly used for query constraining, filtering, and result set labeling.

disaster

An event that prevents a workload or system from fulfilling its business objectives in its primary deployed location. These events can be natural disasters, technical failures, or the result of human actions, such as unintentional misconfiguration or a malware attack.

disaster recovery (DR)

The strategy and process you use to minimize downtime and data loss caused by a disaster. For more information, see Disaster Recovery of Workloads on AWS: Recovery in the Cloud in the AWS Well-Architected Framework.

DML

See database manipulation language.

domain-driven design

An approach to developing a complex software system by connecting its components to evolving domains, or core business goals, that each component serves. This concept was introduced by Eric Evans in his book, *Domain-Driven Design: Tackling Complexity in the Heart of Software* (Boston: Addison-Wesley Professional, 2003). For information about how you can use domain-driven design with the strangler fig pattern, see Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

DR

See disaster recovery.

drift detection

Tracking deviations from a baselined configuration. For example, you can use AWS CloudFormation to detect drift in system resources, or you can use AWS Control Tower to detect changes in your landing zone that might affect compliance with governance requirements.

DVSM

See development value stream mapping.

E

EDA

See exploratory data analysis.

EDI

See electronic data interchange.

edge computing

The technology that increases the computing power for smart devices at the edges of an IoT network. When compared with cloud computing, edge computing can reduce communication latency and improve response time.

electronic data interchange (EDI)

The automated exchange of business documents between organizations. For more information, see What is Electronic Data Interchange.

encryption

A computing process that transforms plaintext data, which is human-readable, into ciphertext.

encryption key

A cryptographic string of randomized bits that is generated by an encryption algorithm. Keys can vary in length, and each key is designed to be unpredictable and unique.

endianness

The order in which bytes are stored in computer memory. Big-endian systems store the most significant byte first. Little-endian systems store the least significant byte first.

endpoint

See service endpoint.

endpoint service

A service that you can host in a virtual private cloud (VPC) to share with other users. You can create an endpoint service with AWS PrivateLink and grant permissions to other AWS accounts or to AWS Identity and Access Management (IAM) principals. These accounts or principals can connect to your endpoint service privately by creating interface VPC endpoints. For more information, see Create an endpoint service in the Amazon Virtual Private Cloud (Amazon VPC) documentation.

enterprise resource planning (ERP)

A system that automates and manages key business processes (such as accounting, MES, and project management) for an enterprise.

envelope encryption

The process of encrypting an encryption key with another encryption key. For more information, see Envelope encryption in the AWS Key Management Service (AWS KMS) documentation.

environment

An instance of a running application. The following are common types of environments in cloud computing:

- development environment – An instance of a running application that is available only to the core team responsible for maintaining the application. Development environments are used to test changes before promoting them to upper environments. This type of environment is sometimes referred to as a *test environment*.
- lower environments – All development environments for an application, such as those used for initial builds and tests.
- production environment – An instance of a running application that end users can access. In a CI/CD pipeline, the production environment is the last deployment environment.

- upper environments – All environments that can be accessed by users other than the core development team. This can include a production environment, preproduction environments, and environments for user acceptance testing.

epic

In agile methodologies, functional categories that help organize and prioritize your work. Epics provide a high-level description of requirements and implementation tasks. For example, AWS CAF security epics include identity and access management, detective controls, infrastructure security, data protection, and incident response. For more information about epics in the AWS migration strategy, see the program implementation guide.

ERP

See enterprise resource planning.

exploratory data analysis (EDA)

The process of analyzing a dataset to understand its main characteristics. You collect or aggregate data and then perform initial investigations to find patterns, detect anomalies, and check assumptions. EDA is performed by calculating summary statistics and creating data visualizations.

F

fact table

The central table in a star schema. It stores quantitative data about business operations. Typically, a fact table contains two types of columns: those that contain measures and those that contain a foreign key to a dimension table.

fail fast

A philosophy that uses frequent and incremental testing to reduce the development lifecycle. It is a critical part of an agile approach.

fault isolation boundary

In the AWS Cloud, a boundary such as an Availability Zone, AWS Region, control plane, or data plane that limits the effect of a failure and helps improve the resilience of workloads. For more information, see AWS Fault Isolation Boundaries.

feature branch

See branch.

features

The input data that you use to make a prediction. For example, in a manufacturing context, features could be images that are periodically captured from the manufacturing line.

feature importance

How significant a feature is for a model's predictions. This is usually expressed as a numerical score that can be calculated through various techniques, such as Shapley Additive Explanations (SHAP) and integrated gradients. For more information, see Machine learning model interpretability with AWS.

feature transformation

To optimize data for the ML process, including enriching data with additional sources, scaling values, or extracting multiple sets of information from a single data field. This enables the ML model to benefit from the data. For example, if you break down the "2021-05-27 00:15:37" date into "2021", "May", "Thu", and "15", you can help the learning algorithm learn nuanced patterns associated with different data components.

few-shot prompting

Providing an LLM with a small number of examples that demonstrate the task and desired output before asking it to perform a similar task. This technique is an application of in-context learning, where models learn from examples (*shots*) that are embedded in prompts. Few-shot prompting can be effective for tasks that require specific formatting, reasoning, or domain knowledge. See also zero-shot prompting.

FGAC

See fine-grained access control.

fine-grained access control (FGAC)

The use of multiple conditions to allow or deny an access request.

flash-cut migration

A database migration method that uses continuous data replication through change data capture to migrate data in the shortest time possible, instead of using a phased approach. The objective is to keep downtime to a minimum.

FM

See foundation model.

foundation model (FM)

A large deep-learning neural network that has been training on massive datasets of generalized and unlabeled data. FMs are capable of performing a wide variety of general tasks, such as understanding language, generating text and images, and conversing in natural language. For more information, see What are Foundation Models.

FM gateway

A centralized intermediary that controls and normalizes access to foundation models. Also known as an *LLM gateway*.

G

generative AI

A subset of AI models that have been trained on large amounts of data and that can use a simple text prompt to create new content and artifacts, such as images, videos, text, and audio. For more information, see What is Generative AI.

geo blocking

See geographic restrictions.

geographic restrictions (geo blocking)

In Amazon CloudFront, an option to prevent users in specific countries from accessing content distributions. You can use an allow list or block list to specify approved and banned countries. For more information, see Restricting the geographic distribution of your content in the CloudFront documentation.

Gitflow workflow

An approach in which lower and upper environments use different branches in a source code repository. The Gitflow workflow is considered legacy, and the trunk-based workflow is the modern, preferred approach.

golden image

A snapshot of a system or software that is used as a template to deploy new instances of that system or software. For example, in manufacturing, a golden image can be used to provision software on multiple devices and helps improve speed, scalability, and productivity in device manufacturing operations.

greenfield strategy

The absence of existing infrastructure in a new environment. When adopting a greenfield strategy for a system architecture, you can select all new technologies without the restriction of compatibility with existing infrastructure, also known as brownfield. If you are expanding the existing infrastructure, you might blend brownfield and greenfield strategies.

guardrail

A high-level rule that helps govern resources, policies, and compliance across organizational units (OUs). *Preventive guardrails* enforce policies to ensure alignment to compliance standards. They are implemented by using service control policies and IAM permissions boundaries. *Detective guardrails* detect policy violations and compliance issues, and generate alerts for remediation. They are implemented by using AWS Config, AWS Security Hub CSPM, Amazon GuardDuty, AWS Trusted Advisor, Amazon Inspector, and custom AWS Lambda checks.

guardrails (AI)

Safety mechanisms that filter, validate, and constrain agent inputs and outputs to help ensure responsible and safe AI behavior.

H

HA

See high availability.

heterogeneous database migration

Migrating your source database to a target database that uses a different database engine (for example, Oracle to Amazon Aurora). Heterogeneous migration is typically part of a re-architecting effort, and converting the schema can be a complex task. AWS provides AWS SCT that helps with schema conversions.

high availability (HA)

The ability of a workload to operate continuously, without intervention, in the event of challenges or disasters. HA systems are designed to automatically fail over, consistently deliver high-quality performance, and handle different loads and failures with minimal performance impact.

historian modernization

An approach used to modernize and upgrade operational technology (OT) systems to better serve the needs of the manufacturing industry. A *historian* is a type of database that is used to collect and store data from various sources in a factory.

holdout data

A portion of historical, labeled data that is withheld from a dataset that is used to train a machine learning model. You can use holdout data to evaluate the model performance by comparing the model predictions against the holdout data.

human-in-the-loop (HitL)

A workflow pattern where agent execution pauses for human review and approval at critical decision points.

homogeneous database migration

Migrating your source database to a target database that shares the same database engine (for example, Microsoft SQL Server to Amazon RDS for SQL Server). Homogeneous migration is typically part of a rehosting or replatforming effort. You can use native database utilities to migrate the schema.

hot data

Data that is frequently accessed, such as real-time data or recent translational data. This data typically requires a high-performance storage tier or class to provide fast query responses.

hotfix

An urgent fix for a critical issue in a production environment. Due to its urgency, a hotfix is usually made outside of the typical DevOps release workflow.

hypercare period

Immediately following cutover, the period of time when a migration team manages and monitors the migrated applications in the cloud in order to address any issues. Typically, this

period is 1–4 days in length. At the end of the hypercare period, the migration team typically transfers responsibility for the applications to the cloud operations team.

I

IaC

See infrastructure as code.

identity-based policy

A policy attached to one or more IAM principals that defines their permissions within the AWS Cloud environment.

idle application

An application that has an average CPU and memory usage between 5 and 20 percent over a period of 90 days. In a migration project, it is common to retire these applications or retain them on premises.

IIoT

See industrial Internet of Things.

immutable infrastructure

A model that deploys new infrastructure for production workloads instead of updating, patching, or modifying the existing infrastructure. Immutable infrastructures are inherently more consistent, reliable, and predictable than mutable infrastructure. For more information, see the Deploy using immutable infrastructure best practice in the AWS Well-Architected Framework.

inbound (ingress) VPC

In an AWS multi-account architecture, a VPC that accepts, inspects, and routes network connections from outside an application. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

incremental migration

A cutover strategy in which you migrate your application in small parts instead of performing a single, full cutover. For example, you might move only a few microservices or users to the new system initially. After you verify that everything is working properly, you can incrementally

move additional microservices or users until you can decommission your legacy system. This strategy reduces the risks associated with large migrations.

Industry 4.0

A term that was introduced by Klaus Schwab in 2016 to refer to the modernization of manufacturing processes through advances in connectivity, real-time data, automation, analytics, and AI/ML.

infrastructure

All of the resources and assets contained within an application's environment.

infrastructure as code (IaC)

The process of provisioning and managing an application's infrastructure through a set of configuration files. IaC is designed to help you centralize infrastructure management, standardize resources, and scale quickly so that new environments are repeatable, reliable, and consistent.

industrial Internet of Things (IIoT)

The use of internet-connected sensors and devices in the industrial sectors, such as manufacturing, energy, automotive, healthcare, life sciences, and agriculture. For more information, see Building an industrial Internet of Things (IIoT) digital transformation strategy.

inspection VPC

In an AWS multi-account architecture, a centralized VPC that manages inspections of network traffic between VPCs (in the same or different AWS Regions), the internet, and on-premises networks. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

Internet of Things (IoT)

The network of connected physical objects with embedded sensors or processors that communicate with other devices and systems through the internet or over a local communication network. For more information, see What is IoT?

interpretability

A characteristic of a machine learning model that describes the degree to which a human can understand how the model's predictions depend on its inputs. For more information, see Machine learning model interpretability with AWS.

IoT

See Internet of Things.

IT information library (ITIL)

A set of best practices for delivering IT services and aligning these services with business requirements. ITIL provides the foundation for ITSM.

IT service management (ITSM)

Activities associated with designing, implementing, managing, and supporting IT services for an organization. For information about integrating cloud operations with ITSM tools, see the operations integration guide.

ITIL

See IT information library.

ITSM

See IT service management.

L

label-based access control (LBAC)

An implementation of mandatory access control (MAC) where the users and the data itself are each explicitly assigned a security label value. The intersection between the user security label and data security label determines which rows and columns can be seen by the user.

landing zone

A landing zone is a well-architected, multi-account AWS environment that is scalable and secure. This is a starting point from which your organizations can quickly launch and deploy workloads and applications with confidence in their security and infrastructure environment. For more information about landing zones, see Setting up a secure and scalable multi-account AWS environment.

large language model (LLM)

A deep learning AI model that is pretrained on a vast amount of data. An LLM can perform multiple tasks, such as answering questions, summarizing documents, translating text into other languages, and completing sentences. For more information, see What are LLMs.

large migration

A migration of 300 or more servers.

LBAC

See label-based access control.

least privilege

The security best practice of granting the minimum permissions required to perform a task. For more information, see Apply least-privilege permissions in the IAM documentation.

lift and shift

See 7 Rs.

little-endian system

A system that stores the least significant byte first. See also endianness.

LLM

See large language model.

lower environments

See environment.

M

machine learning (ML)

A type of artificial intelligence that uses algorithms and techniques for pattern recognition and learning. ML analyzes and learns from recorded data, such as Internet of Things (IoT) data, to generate a statistical model based on patterns. For more information, see Machine Learning.

main branch

See branch.

malware

Software that is designed to compromise computer security or privacy. Malware might disrupt computer systems, leak sensitive information, or gain unauthorized access. Examples of malware include viruses, worms, ransomware, Trojan horses, spyware, and keyloggers.

managed services

AWS services for which AWS operates the infrastructure layer, the operating system, and platforms, and you access the endpoints to store and retrieve data. Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB are examples of managed services. These are also known as *abstracted services*.

manufacturing execution system (MES)

A software system for tracking, monitoring, documenting, and controlling production processes that convert raw materials to finished products on the shop floor.

MAP

See Migration Acceleration Program.

MCP

See Model Context Protocol.

Model Context Protocol (MCP)

A stateless protocol for agent-to-tool communication.

MCP server

A service that exposes one or more tools through the Model Context Protocol.

mechanism

A complete process in which you create a tool, drive adoption of the tool, and then inspect the results in order to make adjustments. A mechanism is a cycle that reinforces and improves itself as it operates. For more information, see Building mechanisms in the AWS Well-Architected Framework.

member account

All AWS accounts other than the management account that are part of an organization in AWS Organizations. An account can be a member of only one organization at a time.

MES

See manufacturing execution system.

Message Queuing Telemetry Transport (MQTT)

A lightweight, machine-to-machine (M2M) communication protocol, based on the publish/subscribe pattern, for resource-constrained IoT devices.

microservice

A small, independent service that communicates over well-defined APIs and is typically owned by small, self-contained teams. For example, an insurance system might include microservices that map to business capabilities, such as sales or marketing, or subdomains, such as purchasing, claims, or analytics. The benefits of microservices include agility, flexible scaling, easy deployment, reusable code, and resilience. For more information, see Integrating microservices by using AWS serverless services.

microservices architecture

An approach to building an application with independent components that run each application process as a microservice. These microservices communicate through a well-defined interface by using lightweight APIs. Each microservice in this architecture can be updated, deployed, and scaled to meet demand for specific functions of an application. For more information, see Implementing microservices on AWS.

Migration Acceleration Program (MAP)

An AWS program that provides consulting support, training, and services to help organizations build a strong operational foundation for moving to the cloud, and to help offset the initial cost of migrations. MAP includes a migration methodology for executing legacy migrations in a methodical way and a set of tools to automate and accelerate common migration scenarios.

migration at scale

The process of moving the majority of the application portfolio to the cloud in waves, with more applications moved at a faster rate in each wave. This phase uses the best practices and lessons learned from the earlier phases to implement a *migration factory* of teams, tools, and processes to streamline the migration of workloads through automation and agile delivery. This is the third phase of the AWS migration strategy.

migration factory

Cross-functional teams that streamline the migration of workloads through automated, agile approaches. Migration factory teams typically include operations, business analysts and owners, migration engineers, developers, and DevOps professionals working in sprints. Between 20 and 50 percent of an enterprise application portfolio consists of repeated patterns that can be optimized by a factory approach. For more information, see the discussion of migration factories and the Cloud Migration Factory guide in this content set.

migration metadata

The information about the application and server that is needed to complete the migration. Each migration pattern requires a different set of migration metadata. Examples of migration metadata include the target subnet, security group, and AWS account.

migration pattern

A repeatable migration task that details the migration strategy, the migration destination, and the migration application or service used. Example: Rehost migration to Amazon EC2 with AWS Application Migration Service.

Migration Portfolio Assessment (MPA)

An online tool that provides information for validating the business case for migrating to the AWS Cloud. MPA provides detailed portfolio assessment (server right-sizing, pricing, TCO comparisons, migration cost analysis) as well as migration planning (application data analysis and data collection, application grouping, migration prioritization, and wave planning). The MPA tool (requires login) is available free of charge to all AWS consultants and APN Partner consultants.

Migration Readiness Assessment (MRA)

The process of gaining insights about an organization's cloud readiness status, identifying strengths and weaknesses, and building an action plan to close identified gaps, using the AWS CAF. For more information, see the migration readiness guide. MRA is the first phase of the AWS migration strategy.

migration strategy

The approach used to migrate a workload to the AWS Cloud. For more information, see the 7 Rs entry in this glossary and see Mobilize your organization to accelerate large-scale migrations.

ML

See machine learning.

modernization

Transforming an outdated (legacy or monolithic) application and its infrastructure into an agile, elastic, and highly available system in the cloud to reduce costs, gain efficiencies, and take advantage of innovations. For more information, see Strategy for modernizing applications in the AWS Cloud.

modernization readiness assessment

An evaluation that helps determine the modernization readiness of an organization's applications; identifies benefits, risks, and dependencies; and determines how well the organization can support the future state of those applications. The outcome of the assessment is a blueprint of the target architecture, a roadmap that details development phases and milestones for the modernization process, and an action plan for addressing identified gaps. For more information, see Evaluating modernization readiness for applications in the AWS Cloud.

monolithic applications (monoliths)

Applications that run as a single service with tightly coupled processes. Monolithic applications have several drawbacks. If one application feature experiences a spike in demand, the entire architecture must be scaled. Adding or improving a monolithic application's features also becomes more complex when the code base grows. To address these issues, you can use a microservices architecture. For more information, see Decomposing monoliths into microservices.

MPA

See Migration Portfolio Assessment.

MQTT

See Message Queuing Telemetry Transport.

multiclass classification

A process that helps generate predictions for multiple classes (predicting one of more than two outcomes). For example, an ML model might ask "Is this product a book, car, or phone?" or "Which product category is most interesting to this customer?"

mutable infrastructure

A model that updates and modifies the existing infrastructure for production workloads. For improved consistency, reliability, and predictability, the AWS Well-Architected Framework recommends the use of immutable infrastructure as a best practice.

O

OAC

See origin access control.

OAI

See origin access identity.

OCM

See organizational change management.

offline migration

A migration method in which the source workload is taken down during the migration process. This method involves extended downtime and is typically used for small, non-critical workloads.

OI

See operations integration.

OLA

See operational-level agreement.

online migration

A migration method in which the source workload is copied to the target system without being taken offline. Applications that are connected to the workload can continue to function during the migration. This method involves zero to minimal downtime and is typically used for critical production workloads.

OPC-UA

See Open Process Communications - Unified Architecture.

Open Process Communications - Unified Architecture (OPC-UA)

A machine-to-machine (M2M) communication protocol for industrial automation. OPC-UA provides an interoperability standard with data encryption, authentication, and authorization schemes.

operational-level agreement (OLA)

An agreement that clarifies what functional IT groups promise to deliver to each other, to support a service-level agreement (SLA).

operational readiness review (ORR)

A checklist of questions and associated best practices that help you understand, evaluate, prevent, or reduce the scope of incidents and possible failures. For more information, see Operational Readiness Reviews (ORR) in the AWS Well-Architected Framework.

operational technology (OT)

Hardware and software systems that work with the physical environment to control industrial operations, equipment, and infrastructure. In manufacturing, the integration of OT and information technology (IT) systems is a key focus for Industry 4.0 transformations.

operations integration (OI)

The process of modernizing operations in the cloud, which involves readiness planning, automation, and integration. For more information, see the operations integration guide.

organization trail

A trail that's created by AWS CloudTrail that logs all events for all AWS accounts in an organization in AWS Organizations. This trail is created in each AWS account that's part of the organization and tracks the activity in each account. For more information, see Creating a trail for an organization in the CloudTrail documentation.

organizational change management (OCM)

A framework for managing major, disruptive business transformations from a people, culture, and leadership perspective. OCM helps organizations prepare for, and transition to, new systems and strategies by accelerating change adoption, addressing transitional issues, and driving cultural and organizational changes. In the AWS migration strategy, this framework is called *people acceleration*, because of the speed of change required in cloud adoption projects. For more information, see the OCM guide.

origin access control (OAC)

In CloudFront, an enhanced option for restricting access to secure your Amazon Simple Storage Service (Amazon S3) content. OAC supports all S3 buckets in all AWS Regions, server-side encryption with AWS KMS (SSE-KMS), and dynamic PUT and DELETE requests to the S3 bucket.

origin access identity (OAI)

In CloudFront, an option for restricting access to secure your Amazon S3 content. When you use OAI, CloudFront creates a principal that Amazon S3 can authenticate with. Authenticated principals can access content in an S3 bucket only through a specific CloudFront distribution. See also OAC, which provides more granular and enhanced access control.

ORR

See operational readiness review.

OT

See operational technology.

outbound (egress) VPC

In an AWS multi-account architecture, a VPC that handles network connections that are initiated from within an application. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

P

permissions boundary

An IAM management policy that is attached to IAM principals to set the maximum permissions that the user or role can have. For more information, see Permissions boundaries in the IAM documentation.

personally identifiable information (PII)

Information that, when viewed directly or paired with other related data, can be used to reasonably infer the identity of an individual. Examples of PII include names, addresses, and contact information.

PII

See personally identifiable information.

playbook

A set of predefined steps that capture the work associated with migrations, such as delivering core operations functions in the cloud. A playbook can take the form of scripts, automated runbooks, or a summary of processes or steps required to operate your modernized environment.

PLC

See programmable logic controller.

PLM

See product lifecycle management.

policy

An object that can define permissions (see identity-based policy), specify access conditions (see resource-based policy), or define the maximum permissions for all accounts in an organization in AWS Organizations (see service control policy).

polyglot persistence

Independently choosing a microservice's data storage technology based on data access patterns and other requirements. If your microservices have the same data storage technology, they can encounter implementation challenges or experience poor performance. Microservices are more easily implemented and achieve better performance and scalability if they use the data store best adapted to their requirements.

portfolio assessment

A process of discovering, analyzing, and prioritizing the application portfolio in order to plan the migration. For more information, see Evaluating migration readiness.

predicate

A query condition that returns `true` or `false`, commonly located in a `WHERE` clause.

predicate pushdown

A database query optimization technique that filters the data in the query before transfer. This reduces the amount of data that must be retrieved and processed from the relational database, and it improves query performance.

preventative control

A security control that is designed to prevent an event from occurring. These controls are a first line of defense to help prevent unauthorized access or unwanted changes to your network. For more information, see Preventative controls in *Implementing security controls on AWS*.

principal

An entity in AWS that can perform actions and access resources. This entity is typically a root user for an AWS account, an IAM role, or a user. For more information, see *Principal* in Roles terms and concepts in the IAM documentation.

privacy by design

A system engineering approach that takes privacy into account through the whole development process.

private hosted zones

A container that holds information about how you want Amazon Route 53 to respond to DNS queries for a domain and its subdomains within one or more VPCs. For more information, see Working with private hosted zones in the Route 53 documentation.

proactive control

A security control designed to prevent the deployment of noncompliant resources. These controls scan resources before they are provisioned. If the resource is not compliant with the control, then it isn't provisioned. For more information, see the Controls reference guide in the AWS Control Tower documentation and see Proactive controls in *Implementing security controls on AWS*.

product lifecycle management (PLM)

The management of data and processes for a product throughout its entire lifecycle, from design, development, and launch, through growth and maturity, to decline and removal.

production environment

See environment.

programmable logic controller (PLC)

In manufacturing, a highly reliable, adaptable computer that monitors machines and automates manufacturing processes.

prompt chaining

Using the output of one LLM prompt as the input for the next prompt to generate better responses. This technique is used to break down a complex task into subtasks, or to iteratively refine or expand a preliminary response. It helps improve the accuracy and relevance of a model's responses and allows for more granular, personalized results.

pseudonymization

The process of replacing personal identifiers in a dataset with placeholder values. Pseudonymization can help protect personal privacy. Pseudonymized data is still considered to be personal data.

publish/subscribe (pub/sub)

A pattern that enables asynchronous communications among microservices to improve scalability and responsiveness. For example, in a microservices-based MES, a microservice can

publish event messages to a channel that other microservices can subscribe to. The system can add new microservices without changing the publishing service.

Q

query plan

A series of steps, like instructions, that are used to access the data in a SQL relational database system.

query plan regression

When a database service optimizer chooses a less optimal plan than it did before a given change to the database environment. This can be caused by changes to statistics, constraints, environment settings, query parameter bindings, and updates to the database engine.

R

RACI matrix

See responsible, accountable, consulted, informed (RACI).

RAG

See Retrieval Augmented Generation.

ransomware

A malicious software that is designed to block access to a computer system or data until a payment is made.

RASCI matrix

See responsible, accountable, consulted, informed (RACI).

RCAC

See row and column access control.

read replica

A copy of a database that's used for read-only purposes. You can route queries to the read replica to reduce the load on your primary database.

re-architect

See 7 Rs.

recovery point objective (RPO)

The maximum acceptable amount of time since the last data recovery point. This determines what is considered an acceptable loss of data between the last recovery point and the interruption of service.

recovery time objective (RTO)

The maximum acceptable delay between the interruption of service and restoration of service.

refactor

See 7 Rs.

Region

A collection of AWS resources in a geographic area. Each AWS Region is isolated and independent of the others to provide fault tolerance, stability, and resilience. For more information, see Specify which AWS Regions your account can use.

regression

An ML technique that predicts a numeric value. For example, to solve the problem of "What price will this house sell for?" an ML model could use a linear regression model to predict a house's sale price based on known facts about the house (for example, the square footage).

rehost

See 7 Rs.

release

In a deployment process, the act of promoting changes to a production environment.

relocate

See 7 Rs.

replatform

See 7 Rs.

repurchase

See 7 Rs.

resiliency

An application's ability to resist or recover from disruptions. High availability and disaster recovery are common considerations when planning for resiliency in the AWS Cloud. For more information, see AWS Cloud Resilience.

resource-based policy

A policy attached to a resource, such as an Amazon S3 bucket, an endpoint, or an encryption key. This type of policy specifies which principals are allowed access, supported actions, and any other conditions that must be met.

responsible, accountable, consulted, informed (RACI) matrix

A matrix that defines the roles and responsibilities for all parties involved in migration activities and cloud operations. The matrix name is derived from the responsibility types defined in the matrix: responsible (R), accountable (A), consulted (C), and informed (I). The support (S) type is optional. If you include support, the matrix is called a *RASCI matrix*, and if you exclude it, it's called a *RACI matrix*.

responsive control

A security control that is designed to drive remediation of adverse events or deviations from your security baseline. For more information, see Responsive controls in *Implementing security controls on AWS*.

retain

See 7 Rs.

retire

See 7 Rs.

Retrieval Augmented Generation (RAG)

A generative AI technology in which an LLM references an authoritative data source that is outside of its training data sources before generating a response. For example, a RAG model might perform a semantic search of an organization's knowledge base or custom data. For more information, see What is RAG.

rotation

The process of periodically updating a secret to make it more difficult for an attacker to access the credentials.

row and column access control (RCAC)

The use of basic, flexible SQL expressions that have defined access rules. RCAC consists of row permissions and column masks.

RPO

See recovery point objective.

RTO

See recovery time objective.

runbook

A set of manual or automated procedures required to perform a specific task. These are typically built to streamline repetitive operations or procedures with high error rates.

S

SAML 2.0

An open standard that many identity providers (IdPs) use. This feature enables federated single sign-on (SSO), so users can log into the AWS Management Console or call the AWS API operations without you having to create user in IAM for everyone in your organization. For more information about SAML 2.0-based federation, see About SAML 2.0-based federation in the IAM documentation.

SCADA

See supervisory control and data acquisition.

SCP

See service control policy.

secret

In AWS Secrets Manager, confidential or restricted information, such as a password or user credentials, that you store in encrypted form. It consists of the secret value and its metadata.

The secret value can be binary, a single string, or multiple strings. For more information, see What's in a Secrets Manager secret? in the Secrets Manager documentation.

security by design

A system engineering approach that takes security into account through the whole development process.

security control

A technical or administrative guardrail that prevents, detects, or reduces the ability of a threat actor to exploit a security vulnerability. There are four primary types of security controls: preventative, detective, responsive, and proactive.

security hardening

The process of reducing the attack surface to make it more resistant to attacks. This can include actions such as removing resources that are no longer needed, implementing the security best practice of granting least privilege, or deactivating unnecessary features in configuration files.

security information and event management (SIEM) system

Tools and services that combine security information management (SIM) and security event management (SEM) systems. A SIEM system collects, monitors, and analyzes data from servers, networks, devices, and other sources to detect threats and security breaches, and to generate alerts.

security response automation

A predefined and programmed action that is designed to automatically respond to or remediate a security event. These automations serve as detective or responsive security controls that help you implement AWS security best practices. Examples of automated response actions include modifying a VPC security group, patching an Amazon EC2 instance, or rotating credentials.

server-side encryption

Encryption of data at its destination, by the AWS service that receives it.

service control policy (SCP)

A policy that provides centralized control over permissions for all accounts in an organization in AWS Organizations. SCPs define guardrails or set limits on actions that an administrator can delegate to users or roles. You can use SCPs as allow lists or deny lists, to specify which services

or actions are permitted or prohibited. For more information, see Service control policies in the AWS Organizations documentation.

service endpoint

The URL of the entry point for an AWS service. You can use the endpoint to connect programmatically to the target service. For more information, see AWS service endpoints in *AWS General Reference*.

service-level agreement (SLA)

An agreement that clarifies what an IT team promises to deliver to their customers, such as service uptime and performance.

service-level indicator (SLI)

A measurement of a performance aspect of a service, such as its error rate, availability, or throughput.

service-level objective (SLO)

A target metric that represents the health of a service, as measured by a service-level indicator.

shared responsibility model

A model describing the responsibility you share with AWS for cloud security and compliance. AWS is responsible for security *of* the cloud, whereas you are responsible for security *in* the cloud. For more information, see Shared responsibility model.

Shadow AI

Unauthorized AI applications built or used outside of governed channels within an organization.

SIEM

See security information and event management system.

single point of failure (SPOF)

A failure in a single, critical component of an application that can disrupt the system.

SLA

See service-level agreement.

SLI

See service-level indicator.

SLO

See service-level objective.

split-and-seed model

A pattern for scaling and accelerating modernization projects. As new features and product releases are defined, the core team splits up to create new product teams. This helps scale your organization's capabilities and services, improves developer productivity, and supports rapid innovation. For more information, see Phased approach to modernizing applications in the AWS Cloud.

SPOF

See single point of failure.

star schema

A database organizational structure that uses one large fact table to store transactional or measured data and uses one or more smaller dimensional tables to store data attributes. This structure is designed for use in a data warehouse or for business intelligence purposes.

strangler fig pattern

An approach to modernizing monolithic systems by incrementally rewriting and replacing system functionality until the legacy system can be decommissioned. This pattern uses the analogy of a fig vine that grows into an established tree and eventually overcomes and replaces its host. The pattern was introduced by Martin Fowler as a way to manage risk when rewriting monolithic systems. For an example of how to apply this pattern, see Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

subnet

A range of IP addresses in your VPC. A subnet must reside in a single Availability Zone.

supervisory control and data acquisition (SCADA)

In manufacturing, a system that uses hardware and software to monitor physical assets and production operations.

symmetric encryption

An encryption algorithm that uses the same key to encrypt and decrypt the data.

synthetic testing

Testing a system in a way that simulates user interactions to detect potential issues or to monitor performance. You can use Amazon CloudWatch Synthetics to create these tests.

system prompt

A technique for providing context, instructions, or guidelines to an LLM to direct its behavior. System prompts help set context and establish rules for interactions with users.

T

tags

Key-value pairs that act as metadata for organizing your AWS resources. Tags can help you manage, identify, organize, search for, and filter resources. For more information, see Tagging your AWS resources.

target variable

The value that you are trying to predict in supervised ML. This is also referred to as an *outcome variable*. For example, in a manufacturing setting the target variable could be a product defect.

task list

A tool that is used to track progress through a runbook. A task list contains an overview of the runbook and a list of general tasks to be completed. For each general task, it includes the estimated amount of time required, the owner, and the progress.

test environment

See environment.

training

To provide data for your ML model to learn from. The training data must contain the correct answer. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer that you want to predict). It outputs an ML model that captures these patterns. You can then use the ML model to make predictions on new data for which you don't know the target.

tool

A function or API that an agent can invoke to perform operations in external systems.

transit gateway

A network transit hub that you can use to interconnect your VPCs and on-premises networks. For more information, see What is a transit gateway in the AWS Transit Gateway documentation.

trunk-based workflow

An approach in which developers build and test features locally in a feature branch and then merge those changes into the main branch. The main branch is then built to the development, preproduction, and production environments, sequentially.

trusted access

Granting permissions to a service that you specify to perform tasks in your organization in AWS Organizations and in its accounts on your behalf. The trusted service creates a service-linked role in each account, when that role is needed, to perform management tasks for you. For more information, see Using AWS Organizations with other AWS services in the AWS Organizations documentation.

tuning

To change aspects of your training process to improve the ML model's accuracy. For example, you can train the ML model by generating a labeling set, adding labels, and then repeating these steps several times under different settings to optimize the model.

two-pizza team

A small DevOps team that you can feed with two pizzas. A two-pizza team size ensures the best possible opportunity for collaboration in software development.

U

uncertainty

A concept that refers to imprecise, incomplete, or unknown information that can undermine the reliability of predictive ML models. There are two types of uncertainty: *Epistemic uncertainty* is caused by limited, incomplete data, whereas *aleatoric uncertainty* is caused by the noise and randomness inherent in the data.

undifferentiated tasks

Also known as *heavy lifting*, work that is necessary to create and operate an application but that doesn't provide direct value to the end user or provide competitive advantage. Examples of undifferentiated tasks include procurement, maintenance, and capacity planning.

upper environments

See environment.

V

vacuuming

A database maintenance operation that involves cleaning up after incremental updates to reclaim storage and improve performance.

version control

Processes and tools that track changes, such as changes to source code in a repository.

VPC peering

A connection between two VPCs that allows you to route traffic by using private IP addresses. For more information, see What is VPC peering in the Amazon VPC documentation.

vulnerability

A software or hardware flaw that compromises the security of the system.

W

warm cache

A buffer cache that contains current, relevant data that is frequently accessed. The database instance can read from the buffer cache, which is faster than reading from the main memory or disk.

warm data

Data that is infrequently accessed. When querying this kind of data, moderately slow queries are typically acceptable.

window function

A SQL function that performs a calculation on a group of rows that relate in some way to the current record. Window functions are useful for processing tasks, such as calculating a moving average or accessing the value of rows based on the relative position of the current row.

workload

A collection of resources and code that delivers business value, such as a customer-facing application or backend process.

workstream

Functional groups in a migration project that are responsible for a specific set of tasks. Each workstream is independent but supports the other workstreams in the project. For example, the portfolio workstream is responsible for prioritizing applications, wave planning, and collecting migration metadata. The portfolio workstream delivers these assets to the migration workstream, which then migrates the servers and applications.

WORM

See write once, read many.

WQF

See AWS Workload Qualification Framework.

write once, read many (WORM)

A storage model that writes data a single time and prevents the data from being deleted or modified. Authorized users can read the data as many times as needed, but they cannot change it. This data storage infrastructure is considered immutable.

Z

zero-day exploit

An attack, typically malware, that takes advantage of a zero-day vulnerability.

zero-day vulnerability

An unmitigated flaw or vulnerability in a production system. Threat actors can use this type of vulnerability to attack the system. Developers frequently become aware of the vulnerability as a result of the attack.

zero-shot prompting

Providing an LLM with instructions for performing a task but no examples (*shots*) that can help guide it. The LLM must use its pre-trained knowledge to handle the task. The effectiveness of zero-shot prompting depends on the complexity of the task and the quality of the prompt. See also few-shot prompting.

zombie application

An application that has an average CPU and memory usage below 5 percent. In a migration project, it is common to retire these applications.