Transitioning to multiple AWS accounts

AWS Prescriptive Guidance

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Prescriptive Guidance: Transitioning to multiple AWS accounts

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

Transitioning to multiple AWS accounts

Amazon Web Services (contributors)

November 2024 (document history)

Many companies begin their journey by using a single Amazon Web Services (AWS) account. Multiple roles within a company use this account to operate the business. Engineers develop code, deploy to development and test environments, and promote changes to production. Product managers query data sources to gather insights into business performance. The sales team is conducting demos from the production environment to attract new customers. The finance team is monitoring cloud spending from the AWS Billing console.

When all of these separate roles use a single AWS account, it can become difficult to enforce the security best practice of Applying the least-privilege permissions, which means you grant only the minimum permissions necessary to do the job. At a certain stage in a startup's development, someone will ask the question *Do all of our engineers need access to production?* The answer is almost always *no*, but many companies struggle with how to unwind their existing single-account environment into a multi-account environment without slowing down business.

This guide includes best practices to help you transition from a single-account environment to a multi-account environment. It discusses the decisions you need to make about account migration, user management, networking, security, and architecture. It is designed to help you succeed with minimal or no downtime for your business and daily operations. This guide focuses on the following capabilities as you transition from a single AWS account to a multi-account environment:

- Identity management and access control
- Managing permissions and access
- Network connectivity
- Security incident response
- Backups
- Account migration
- Resource migration
- Billing considerations

For more information about capabilities, see Cloud Foundation on AWS.

This guide is aligned to existing resources related to this topic, including the AWS Startup Security Baseline (AWS SSB), the Organizing Your AWS Environment Using Multiple Accounts whitepaper, the AWS Security Reference Architecture (AWS SRA) and the Establishing Your Cloud Foundation on AWS whitepaper. You should continue to use those resources for more specific guidance not covered in this guide.

Intended audience

This guide is best suited for company that wants or needs to transition to multiple AWS accounts. For startups, this need typically arises when you have found product-market fit, raised a round of funding, and are beginning to hire distinct engineering disciplines, such as infrastructure, development operations (DevOps), or security.

Even if your company isn't ready to make this transition, you can still use this guide to understand the decisions that need to be made during the transition and begin to prepare.

Objectives for transitioning to a multi-account architecture

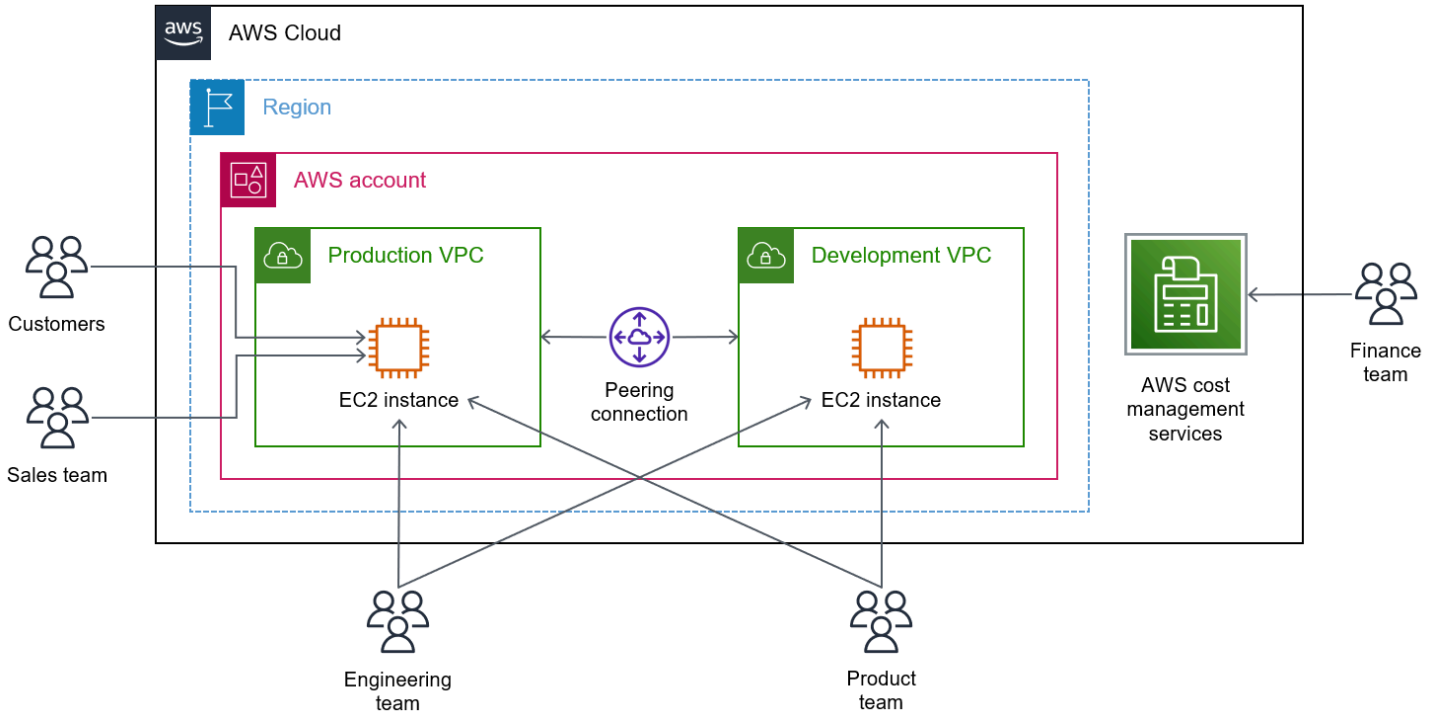
The transition to a multi-account architecture is usually driven by a business need for one or more of the following benefits:

- Grouping workloads based on business purpose or ownership
- Applying distinct security controls by environment
- Constraining access to sensitive data
- Promoting innovation and agility
- Limiting scope of impact from adverse events
- Supporting multiple IT operating models
- Managing costs
- Distributing AWS service quotas and API request rate limits

For more information about the many benefits of using a multi-account architecture, see Organizing Your AWS Environment Using Multiple Accounts (AWS whitepaper) and Guidelines to set up a well-architected environment (AWS Control Tower documentation).

Sample single-account architecture

As a starting point, it is common for startup or small companies to use a single AWS Region and have two virtual private clouds (VPCs) that are connected by VPC peering. Each VPC contains compute resources, such as Amazon Elastic Compute Cloud (Amazon EC2) instances. The engineering team develops code directly in the **Development VPC**. The product team reviews the changes, and then the engineering team manually promotes the changes to the **Production VPC**. The finance team has access to the AWS account so they can review the AWS Billing and Cost Management console.



The following are a few examples of challenges that a company might experience with this environment:

- An engineer mistakenly deleted production data when they thought they were accessing a development database.
- A sales demo was impacted when a production deployment took longer than expected.
- When development code was being load tested, the **Production VPC** became slow and generated error messages about throttling.
- The finance team can't differentiate costs for production and development environments.
- The CEO is concerned that some newly hired offshore contractors have access to customer data though the **Production VPC**.
- The finance team can't disallow access to specific AWS services that might incur high costs.

Adopting a multi-account strategy addresses all of these challenges by using compartmentalized AWS accounts to separate workloads and access.

Foundational framework and security responsibilities for transitioning to a multi-account architecture

The information and best practices in this guide are designed to complement existing AWS recommendations for infrastructure and security. As you transition from a single AWS account to multiple AWS accounts, it is important to make sure that your new multi-account architecture is consistent with the AWS Well-Architected Framework and Cloud Foundation principles. This helps you build and operate an environment that is designed for security, performance, and resiliency, while adhering to governance requirements and AWS best practices.

AWS Well-Architected Framework

AWS Well-Architected Framework helps you build a secure, high-performing, resilient, and efficient infrastructure for applications and workloads. This guide aligns to the Operational Excellence, Security, and Reliability pillars of this framework. This helps you meet your business and regulatory requirements by following current AWS recommendations.

You can assess your adherence to well-architected best practices by using the AWS Well-Architected Tool in your AWS account.

Cloud Foundation on AWS

Establishing Your Cloud Foundation on AWS (AWS Whitepaper) provides guidance that helps you tailor your AWS environment to meet the needs of your business. Using a capability-based approach, you can create an environment to deploy, operate, and manage your workloads. You can also enhance the capabilities to extend your environment as your requirements evolve and you deploy additional workloads to the cloud. For more information about the 30 capabilities defined by AWS, see Capabilities. This guide includes best practices for implementing the initial capabilities in their intended order.

You can adopt and implement capabilities according to your operational and governance needs. As your business requirements mature, the capability-based approach can be used as a mechanism to verify that your cloud environment is ready to support your workloads and scale as needed. This approach enables you to confidently establish your cloud environment for your builders and your business.

Identity management and access control for transitioning to a multi-account architecture

This first step when transitioning to a multi-account architecture is to set up your new account structure within an organization. Then you can add users and configure their access to the accounts. This section describes approaches for managing human access into multiple AWS accounts.

This section consists of the following tasks:

- Set up an organization
- Create a landing zone
- Add organizational units
- Add initial users
- Manage member accounts

Set up an organization

When you have multiple AWS accounts, you can logically manage those accounts through an organization in AWS Organizations. An *account* in AWS Organizations is a standard AWS account that contains your AWS resources and the identities that can access those resources. An *organization* is an entity that consolidates your AWS accounts so that you can administer them as a single unit.

When you use an account to create an organization, that account becomes the *management account* (also known as a *payer account* or *root account*) for the organization. An organization can only have one management account. When you add additional AWS accounts to the organization, they become *member accounts*.

> **Note**
>
> Each AWS account also has a single identity called the *root user*. You can sign in as the root user by using the email address and password you used to create the account. However, we strongly recommend that you don't use the root user for everyday tasks, even the administrative ones. For more information, see AWS account root user.

We also recommend centralizing root access for member accounts and removing the root user credentials from member accounts in your organization.

You organize accounts in a hierarchical tree-like structure that consists of the organization root, organizational units (OUs), and member accounts. The *root* is the parent container for all of the accounts in your organization. An *organizational unit* (OU) is a container for accounts within the root. An OU can contain other OUs or member accounts. An OU can have only one parent, and each account can be a member of only one OU. For more information, see Terminology and concepts (AWS Organizations documentation).

A service control policy (SCP) specifies the services and actions that users and roles can use. SCPs are similar to AWS Identity and Access Management (IAM) permissions policies except that they don't grant permissions. Instead, SCPs define the maximum permissions. When you attach a policy to one of the nodes in the hierarchy, it applies to all the OUs and accounts within that node. For example, if you apply a policy to the root, it applies to all OUs and accounts in the organization, and if you apply a policy to an OU, it applies to only the OUs and accounts in the target OU.

A resource control policy (RCP) offers central control over the maximum available permissions for resources in your organization. RCPs help you make sure that resources in your account stay within your organization's access control guidelines.

You can use the AWS Organizations console to centrally view and manage all of your accounts within an organization. One of the benefits of using an organization is that you can receive a consolidated bill that shows all charges associated with the management and member accounts. For more information, see Consolidated billing (AWS Organizations documentation).

Best practices

- Don't use an existing AWS account to create an organization. Start with a new account, which becomes your management account for the organization. Privileged operations can be performed within an organization's management account, and SCPs and RCPs do not apply to the management account. That's why you should limit the cloud resources and data contained in the management account to only those that must be managed in the management account.
- Limit access to the management account to only those individuals who need to provision new AWS accounts and to administer the organization.
- Use SCPs to define the maximum permissions for the root, organizational units, and member accounts. SCPs can't be directly applied to the management account.

- Use RCPs to define the maximum permissions for resources in member accounts. RCPs can't be directly applied to the management account.
- Adhere to the Best practices for AWS Organizations (AWS Organizations documentation).

Create a landing zone

A *landing zone* is a well-architected, multi-account AWS environment that is a starting point from which you can deploy workloads and applications. It provides a baseline to get started with multi-account architecture, identity and access management, governance, data security, network design, and logging. AWS Control Tower is a service that simplifies the maintenance and governance of a multi-account environment by providing automated guardrails. Typically, you provision a single AWS Control Tower landing zone that manages your environment across all AWS Regions. AWS Control Tower works by orchestrating other AWS services within your account. For more information, see What happens when you set up a landing zone (AWS Control Tower documentation).

When you set up a landing zone with AWS Control Tower, you identify three shared accounts: the management account, the log archive account, and the audit account. For more information, see What are the shared accounts (AWS Control Tower documentation). For the management account, you must use an existing account that isn't hosting any workloads to set up the landing zone. For the log archive and audit accounts, you can choose to reuse existing AWS accounts, or AWS Control Tower can create them for you.

For instructions about how to set up your AWS Control Tower landing zone, see Getting started (AWS Control Tower documentation).

Best practices

- Adhere to the best practices in Design principles for your multi-account strategy (AWS Whitepaper).
- Adhere to the Best practices for AWS Control Tower administrators (AWS Control Tower documentation).
- Create your landing zone in the AWS Region that hosts the majority of your workloads.

> **Important**
>
> If you decide to change this Region after deploying your landing zone, you need the assistance of AWS Support, and you must decommission the landing zone. This practice isn't recommended.

- When determining which Regions AWS Control Tower will govern, select only the Regions in which you expect to immediately deploy workloads. You can change these Regions or add more later. If AWS Control Tower governs a Region, it will deploy its detective guardrails into that Region as AWS Config Rules.
- After determining which Regions AWS Control Tower will govern, deny access to all ungoverned Regions. This helps ensure that your workloads and developers can only use approved AWS Regions. This is implemented as a service control policy (SCP) in the organization. For more information, see Configure the AWS Region deny control (AWS Control Tower documentation).
- When setting up your landing zone in AWS Control Tower, we recommend you rename the following OUs and accounts:
  - We recommend that you rename the **Security** OU to **Security_Prod** to signify that this OU will be used for production security-related AWS accounts.
  - We recommend that you allow AWS Control Tower to create an additional OU and then rename it from **Sandbox** to **Workloads**. In the next section, you create additional OUs within the **Workloads** OU, which you use to organize your AWS accounts.
  - We recommend that you rename the centralized logging AWS account from **Log Archive** to **log-archive-prod**.
  - We recommend that you rename the audit account from **Audit** to **security-tooling-prod**.
- To help prevent fraud, AWS requires that AWS accounts have a history of use before they can be added to an AWS Control Tower landing zone. If you are using a new AWS account without any usage history, in the new account, you can launch an Amazon Elastic Compute Cloud (Amazon EC2) instance that is not in the AWS Free Tier. Let the instance run for a few minutes and then terminate it.

Add organizational units

Establishing the proper organization structure is critical to setting up a multi-account environment. Because you use service control policies (SCPs) to define the maximum permissions for an OU

and the accounts within it, your organization structure must be logical from a management, permissions, and financial reporting perspective. For more information about the structure of an organization, including organizational units (OUs), see Terminology and concepts (AWS Organizations documentation).

In this section, you customize the landing zone by creating nested OUs that help you segment and structure your environments, such as production and non-production. These recommended best practices are designed to segment your landing zone to separate production and non-production resources and separate infrastructure from workloads.

For more information about how to create OUs, see Managing organizational units (AWS Organizations documentation).

Best practices

- Within the **Workloads** OU that you created in Create a landing zone, create the following nested OUs:
  - **Prod** – Use this OU for AWS accounts that store and access production data, including customer data.
  - **NonProd** – Use this OU for AWS accounts that store non-production data, such as development, staging, or testing environments

Under the organization root, create an **Infrastructure_Prod** OU. Use this OU to host a centralized networking account.

Add initial users

There are two ways to grant people access to AWS accounts:

- IAM identities, such as users, groups, and roles
- Identity federation, such as by using AWS IAM Identity Center

In smaller companies and single-account environments, it is common for administrators to create an IAM user when a new person joins the company. The access key and secret key credentials associated to an IAM user are known as *long-term credentials* because they don't expire. However, this isn't a recommended security best practice because if an attacker compromised those

credentials, you would have to generate a new set of credentials for the user. Another approach for accessing AWS accounts is through IAM roles. You can also use AWS Security Token Service (AWS STS) to temporarily request *short-term credentials*, which expire after a configurable amount of time.

You can manage people access into your AWS accounts through IAM Identity Center. You can create individual user accounts for each of your employees or contractors, they can manage their own passwords and multi-factor authentication (MFA) solutions, and you can group them to manage access. When configuring MFA, you can use software tokens, such as authenticator applications, or you can use hardware tokens, such as YubiKey devices.

IAM Identity Center also supports federation from external identity providers (IdPs) such as Okta, JumpCloud, and Ping Identity. For more information, see Supported identity providers (IAM Identity Center documentation). By federating with an external IdP, you can manage user authentication across applications and then use IAM Identity Center to authorize access to specific AWS accounts.

Best practices

- Adhere to the Security best practices (IAM documentation) for configuring user access.
- Manage account access by groups instead of by individual users. In IAM Identity Center, create new groups that represent each of your business functions. For example, you might create groups for engineering, finance, sales, and product management.
- Often, groups are defined by separating those who need access to all AWS accounts (often read-only access) and those who need access to a single AWS account. We recommend that you use the following naming convention for groups so that it is easy to identify the AWS account and permissions associated with the group.

  ```
  <prefix>-<account name>-<permission set>
  ```

- For example, for the group `AWS-A-dev-nonprod-DeveloperAccess`, `AWS-A` is a prefix that indicates access to a single account, `dev-nonprod` is the name of the account, and `DeveloperAccess` is the permission set assigned to the group. For the group `AWS-O-BillingAccess`, the `AWS-O` prefix indicates access to the entire organization, and `BillingAccess` indicates the permission set for the group. In this example, because the group has access to the entire organization, an account name isn't represented in the group name.
- If you are using IAM Identity Center with an external SAML-based IdP and want to require MFA, you can use attribute-based access control (ABAC) to pass the authentication method

from the IdP to IAM Identity Center. The attributes are sent through the SAML assertions. For more information, see Enable and configure attributes for access control (IAM Identity Center documentation).

Many IdPs, such as Microsoft Azure Active Directory and Okta, can use the Authentication Method Reference (`amr`) claim inside a SAML assertion to pass the user's MFA status to IAM Identity Center. The claim used to assert MFA status and its format varies by IdP. For more information, see the documentation for your IdP.

In IAM Identity Center, you can then create permission set policies that determine who can access your AWS resources. When you enable ABAC and specify attributes, IAM Identity Center passes the attribute value of the authenticated user to IAM for use in policy evaluation. For more information, see Create permission policies for ABAC (IAM Identity Center documentation). As shown in the following example, you use the `aws:PrincipalTag` condition key to create an access control rule for MFA.

```
"Condition": {
  "StringLike": { "aws:PrincipalTag/amr": "mfa" }
}
```

Manage member accounts

In this section, you invite your preexisting account into the organization and you begin to create new accounts within your organization. An important part of this process is defining the criteria you use to determine whether you need to provision a new account.

This section consists of the following tasks:

- Invite your preexisting account
- Customize VPC settings in AWS Control Tower
- Define scoping criteria

Invite your preexisting account

Within AWS Organizations, you can invite your company's preexisting account into your new organization. Only the management account in the organization can invite other accounts to join. When the administrator of the invited account accepts, the account immediately joins the

organization, and the organization's management account becomes responsible for all charges accrued by the new member account. For more information, see Inviting an AWS account to join your organization and Accepting or declining an invitation from an organization (AWS Organizations documentation).

> **Note**
>
> You can invite an account to join an organization only if that account isn't a currently in another organization. If the account is a member of an existing organization, you must remove it from the organization. If the account is the management account for a different organization that was created in error, you must delete the organization.

> **Important**
>
> If you need access to any historical cost or usage information from your preexisting account, you can use AWS Cost and Usage Report to export that information to an Amazon Simple Storage Service (Amazon S3) bucket. Do this prior to accepting the invitation to join the organization. When an account joins an organization, you lose access to this historical data for the account. For more information, see Setting up an Amazon S3 bucket for Cost and Usage Reports (AWS Cost and Usage Report documentation).

Best practices

- We recommend that you add your preexisting account, which likely contains production workloads, to the **Workloads** > **Prod** organizational unit that you created in Add organizational units .
- By default, the organization's management account doesn't have administrative access over member accounts that are invited to the organization. If you want the management account to have administrative control, you must create the **OrganizationAccountAccessRole** IAM role in the member account and grant permission to the management account to assume the role. For more information, see Creating the OrganizationAccountAccessRole in an invited member account (AWS Organizations documentation).
- For the preexisting account that you invited to the organization, review Best practices for member accounts (AWS Organizations documentation) and confirm that the account adheres to these recommendations.

Customize VPC settings in AWS Control Tower

We recommend that you provision new AWS accounts through the Account Factory in AWS Control Tower. By using Account Factory, you can use the AWS Control Tower integration with Amazon EventBridge to provision resources in new AWS accounts as soon as the account is created.

When you set up a new AWS account, a default virtual private cloud (VPC) is automatically provisioned. However, when you set up a new account through Account Factory, AWS Control Tower automatically provisions an additional VPC. For more information, see Overview of AWS Control Tower and VPCs (AWS Control Tower documentation). This means that, by default, AWS Control Tower provisions two default VPCs in every new account.

It is common for companies to want more control over the VPCs within their accounts. Many prefer to use other services, such as AWS CloudFormation, Hashicorp Terraform, or Pulumi, to set up and manage their VPCs. You should customize the Account Factory settings to prevent creation of the additional VPC provisioned by AWS Control Tower. For instructions, see Configure Amazon VPC settings (AWS Control Tower documentation), and apply the following settings:

1. Disable the **Internet-accessible subnet** option.
2. In **Maximum number of private subnets**, choose **0**.
3. In **Regions for VPC creation**, clear all Regions.
4. In **Availability Zones**, choose **3**.

Best practices

- Delete the default VPC that is automatically provisioned in every new account. This prevents users from launching public EC2 instances in the account without explicitly creating a dedicated VPC. For more information, see Delete your default subnets and default VPC (Amazon Virtual Private Cloud documentation). You can also configure AWS Control Tower Account Factory for Terraform (AFT) to automatically delete the default VPC in newly created accounts.
- Provision a new AWS account called **dev-nonprod** into the **Workloads** > **NonProd** organizational unit. Use this account for your development environment. For instructions, see Provision Account Factory accounts with AWS Service Catalog (AWS Control Tower documentation).

Define scoping criteria

You need to select the criteria your company will use when deciding whether to provision a new AWS account. You might decide to provision accounts for each business unit, or you might decide to provision accounts based on environment, such as production, testing, or QA. Every company has their own requirements for how large or small their AWS accounts should be. Generally, you evaluate the following three factors when deciding how to size your accounts:

- **Balancing service quotas** – *Service quotas* are the maximum values for the number of resources, actions, and items for each AWS service within an AWS account. If many workloads share the same account and one workload is consuming most or all of a service quota, that might negatively impact another workload in the same account. If so, you might need to separate those workloads into different accounts. For more information, see AWS service quotas (AWS General Reference).
- **Cost reporting** - Isolating workloads into separate accounts allows you to see costs at an account level in the cost and usage reports. When you use the same account for multiple workloads, you can use tags to help you manage and identify resources. For more information about tagging, see Tagging AWS resources (AWS General Reference).
- **Controlling access** - When workloads share an account, you need to consider how you will configure IAM policies to limit access to the account resources so that users don't have access to the workloads they don't need. As an alternative, you can use multiple accounts and permission sets in IAM Identity Center to manage access into individual accounts.

Best practices

- Adhere to the best practices in AWS multi-account strategy for your AWS Control Tower landing zone (AWS Control Tower documentation).
- Establish an effective tagging strategy that helps you identify and manage AWS resources. You can use tags to categorize resources by purpose, business unit, environment, or other criteria. For more information, see Best practices for tagging (AWS General Reference documentation).
- Don't overload an account with too many workloads. If the demand of the workload exceeds a service quota, this can cause performance issues. You can separate the competing workloads into different AWS accounts or you can request a service quota increase. For more information, see Requesting a quota increase (Service Quotas documentation).

Managing permissions and access for a multi-account architecture

This section consists of the following topics:

Engineering cultural considerations

One of the pillars of the AWS Well-Architected Framework is Operational Excellence. Teams must understand the operating model and their part in achieving your business outcomes. Teams can focus on achieving shared goals when they understand their responsibilities, can take ownership, and know how decisions are made.

With early stage companies that are building quickly, everyone on the team performs multiple roles. It isn't uncommon for these users to have highly privileged access to the entire AWS account. As companies grow, they often want to follow the principle of *least privilege* and only grant permissions that are required for the user to do their job. To help you limit scope, you can use AWS Identity and Access Management Access Analyzer to see what permissions a user or IAM role is actually using, allowing you to remove any excess permissions.

It can be challenging to decide who in your company has permissions to create IAM roles. This is commonly a vector for escalating privileges. Escalating privileges is when a user can expand their own permissions or scope of access. For example, if a user has limited permissions but can create new IAM roles, that user could escalate their privileges by creating and assuming a new IAM role that has the `AdministratorAccess` managed policy applied.

Some companies limit IAM role provisioning to a centralized team of trusted individuals. The downside of this approach is that this team can quickly become a bottleneck because almost all AWS services require an IAM role to operate. As an alternative, you can use permissions boundaries to delegate IAM access to only users who are developing, testing, launching, and managing your cloud infrastructure. For example policies, see Example Permission Boundaries (GitHub).

Development operations (DevOps) teams, also known as *platform* teams, often need to balance self-service capabilities for multiple internal development teams against application operational stability. Fostering an engineering culture that embraces autonomy, mastery, and purpose in the workplace can help motivate teams. Engineers want to do their work in a self-directed manner, without relying on others to do things for them. If DevOps teams can implement self-service solutions, this also reduces the amount of time others depend on them to get things done.

Creating permission sets

You can manage AWS account access by using permission sets in AWS IAM Identity Center. A *permission set* is a template that helps you deploy one or more IAM policies to multiple AWS accounts. When you assign a permission set to an AWS account, IAM Identity Center creates an IAM role and attaches your IAM policies to that role. For more information, see Create and manage permission sets (IAM Identity Center documentation).

AWS recommends creating permission sets that map to the different personas in your business.

For example, you might create the following permission sets:

- Billing permission set
- Developer permission set
- Production permission set

The following permissions sets are snippets from an AWS CloudFormation template. You should use this code as a starting point and customize it for your business. For more information about CloudFormation templates, see Learn template basics (CloudFormation documentation).

Billing permission set

The finance team uses **BillingAccessPermissionSet** to view the AWS Billing console dashboard and AWS Cost Explorer in each account.

```
BillingAccessPermissionSet:
  Type: "AWS::SSO::PermissionSet"
  Properties:
    Description: Access to Billing and Cost Explorer
    InstanceArn: !Sub "arn:${AWS::Partition}:sso:::instance/ssoins-instanceId"
    ManagedPolicies:
      - !Sub "arn:${AWS::Partition}:iam::aws:policy/job-function/Billing"
    Name: BillingAccess
```

```
    SessionDuration: PT8H
    RelayStateType: https://console.aws.amazon.com/billing/home
```

Developer permission set

The engineering team uses **DeveloperAccessPermissionSet** to access non-production accounts.

```
DeveloperAccessPermissionSet:
  Type: "AWS::SSO::PermissionSet"
  Properties:
    Description: Access to provision resources through CloudFormation
    InlinePolicy: !Sub |-
      {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:${AWS::Partition}:iam::*:role/CloudFormationRole",
            "Condition": {
              "StringEquals": {
                "aws:ResourceAccount": "${!aws:PrincipalAccount}",
                "iam:PassedToService": "cloudformation.${AWS::URLSuffix}"
              }
            }
          },
          {
            "Effect": "Allow",
            "Action": [
              "cloudformation:ContinueUpdateRollback",
              "cloudformation:CreateChangeSet",
              "cloudformation:CreateStack",
              "cloudformation:DeleteStack",
              "cloudformation:RollbackStack",
              "cloudformation:UpdateStack"
            ],
            "Resource": "arn:${AWS::Partition}:cloudformation:*:*:stack/app-*",
            "Condition": {
              "ArnLike": {
                "cloudformation:RoleArn": "arn:${AWS::Partition}:iam::${!
aws:PrincipalAccount}:role/CloudFormationRole"
              },
              "Null": {
```

```
              "cloudformation:ImportResourceTypes": true
            }
          }
        },
        {
          "Effect": "Allow",
          "Action": [
            "cloudformation:CancelUpdateStack",
            "cloudformation:DeleteChangeSet",
            "cloudformation:DetectStackDrift",
            "cloudformation:DetectStackResourceDrift",
            "cloudformation:ExecuteChangeSet",
            "cloudformation:TagResource",
            "cloudformation:UntagResource",
            "cloudformation:UpdateTerminationProtection"
          ],
          "Resource": "arn:${AWS::Partition}:cloudformation:*:*:stack/app-*"
        },
        {
         "Effect": "Allow",
          "Action": [
            "cloudformation:CreateUploadBucket",
            "cloudformation:ValidateTemplate",
            "cloudformation:EstimateTemplateCost"
          ],
          "Resource": "*"
        }
      ]
    }
  InstanceArn: !Sub "arn:${AWS::Partition}:sso:::instance/ssoins-instanceId"
  ManagedPolicies:
    - !Sub "arn:${AWS::Partition}:iam::aws:policy/AWSServiceCatalogEndUserFullAccess"
    - !Sub "arn:${AWS::Partition}:iam::aws:policy/AWSProtonDeveloperAccess"
    - !Sub "arn:${AWS::Partition}:iam::aws:policy/AWSBillingReadOnlyAccess"
    - !Sub "arn:${AWS::Partition}:iam::aws:policy/AWSSupportAccess"
    - !Sub "arn:${AWS::Partition}:iam::aws:policy/ReadOnlyAccess"
  Name: DeveloperAccess
  SessionDuration: PT8H
```

Production permission set

The engineering team uses **ProductionPermissionSet** to access production accounts. This permission set has limited, view-only access.

```
ProductionPermissionSet:
  Type: "AWS::SSO::PermissionSet"
  Properties:
    Description: Access to production accounts
    InlinePolicy: !Sub |-
      {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:${AWS::Partition}:iam::*:role/CloudFormationRole",
            "Condition": {
              "StringEquals": {
                "aws:ResourceAccount": "${!aws:PrincipalAccount}",
                "iam:PassedToService": "cloudformation.${AWS::URLSuffix}"
              }
            }
          },
          {
            "Effect": "Allow",
            "Action": "cloudformation:ContinueUpdateRollback",
            "Resource": "arn:${AWS::Partition}:cloudformation:*:*:stack/app-*",
            "Condition": {
              "ArnLike": {
                "cloudformation:RoleArn": "arn:${AWS::Partition}:iam::${!
aws:PrincipalAccount}:role/CloudFormationRole"
              }
            }
          },
          {
            "Effect": "Allow",
            "Action": "cloudformation:CancelUpdateStack",
            "Resource": "arn:${AWS::Partition}:cloudformation:*:*:stack/app-*"
          }
        ]
      }
    InstanceArn: !Sub "arn:${AWS::Partition}:sso:::instance/ssoins-instanceId"
    ManagedPolicies:
      - !Sub "arn:${AWS::Partition}:iam::aws:policy/AWSBillingReadOnlyAccess"
      - !Sub "arn:${AWS::Partition}:iam::aws:policy/AWSSupportAccess"
      - !Sub "arn:${AWS::Partition}:iam::aws:policy/job-function/ViewOnlyAccess"
    Name: ProductionAccess
```

```
    SessionDuration: PT2H
```

Creating a permissions boundary

After you deploy the permission sets, you establish a permissions boundary. This *permissions boundary* is a mechanism to delegate IAM access to only users who are developing, testing, launching, and managing your cloud infrastructure. Those users can perform only the actions that are permitted by the policy and the permissions boundary.

You can define the permissions boundary in an AWS CloudFormation template and then use CloudFormation StackSets to deploy the template into multiple accounts. This helps you establish and maintain standardized policies across your organization with a single operation. For more information and instructions, see Working with AWS CloudFormation StackSets (CloudFormation documentation).

The following CloudFormation template provisions an IAM role and creates an IAM policy that acts as a permission boundary. Using a stack set, you can deploy this template to all of the member accounts in your organization.

```
CloudFormationRole:
  Type: "AWS::IAM::Role"
  Properties:
    AssumeRolePolicyDocument:
      Version: "2012-10-17"
      Statement:
        Effect: Allow
        Principal:
          Service: !Sub "cloudformation.${AWS::URLSuffix}"
        Action: "sts:AssumeRole"
        Condition:
          StringEquals:
            "aws:SourceAccount": !Ref "AWS::AccountId"
    Description: !Sub "DO NOT DELETE - Used by CloudFormation. Created by
 CloudFormation ${AWS::StackId}"
    ManagedPolicyArns:
      - !Sub "arn:${AWS::Partition}:iam::aws:policy/AdministratorAccess"
    PermissionsBoundary: !Ref DeveloperBoundary
    RoleName: CloudFormationRole

DeveloperBoundary:
```

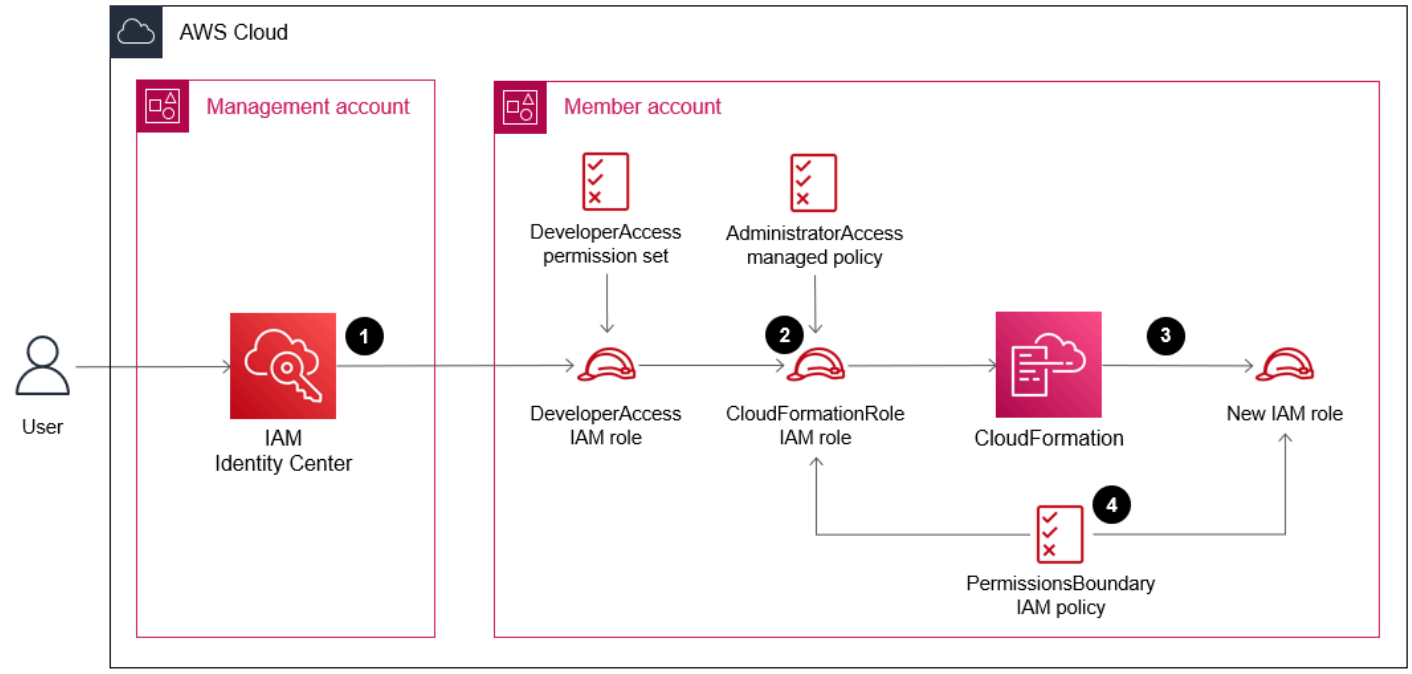
```
  Type: "AWS::IAM::ManagedPolicy"
  Properties:
    Description: Permission boundary for developers
    ManagedPolicyName: PermissionsBoundary
    PolicyDocument:
      Version: "2012-10-17"
      Statement:
        - Sid: AllowModifyIamRolesWithBoundary
          Effect: Allow
          Action:
            - "iam:AttachRolePolicy"
            - "iam:CreateRole"
            - "iam:DeleteRolePolicy"
            - "iam:DetachRolePolicy"
            - "iam:PutRolePermissionsBoundary"
            - "iam:PutRolePolicy"
          Resource: !Sub "arn:${AWS::Partition}:iam::${AWS::AccountId}:role/app/*"
          Condition:
            ArnEquals:
              "iam:PermissionsBoundary": !Sub "arn:${AWS::Partition}:iam::
${AWS::AccountId}:policy/PermissionsBoundary"
        - Sid: AllowModifyIamRoles
          Effect: Allow
          Action:
            - "iam:DeleteRole"
            - "iam:TagRole"
            - "iam:UntagRole"
            - "iam:UpdateAssumeRolePolicy"
            - "iam:UpdateRole"
            - "iam:UpdateRoleDescription"
          Resource: !Sub "arn:${AWS::Partition}:iam::${AWS::AccountId}:role/app/*"
        - Sid: OverlyPermissiveAllowedServices
          Effect: Allow
          Action:
            - "lambda:*"
            - "apigateway:*"
            - "events:*"
            - "s3:*"
            - "logs:*"
          Resource: "*"
```

The **CloudFormationRole** role, **PermissionsBoundary** policy, and the **DeveloperAccess** permission set work together to grant the following permissions:

- Users have read-only access to most AWS services, through the **ReadOnlyAccess** AWS managed policy.
- Users have access to open support cases, through the **AWSSupportAccess** AWS managed policy.
- Users have read-only access to the AWS Billing console dashboard, through the **AWSBillingReadOnlyAccess** AWS managed policy.
- Users are able to provision new environments from AWS Proton, through the **AWSProtonDeveloperAccess** AWS managed policy.
- Users are able to provision products from Service Catalog, through the **AWSServiceCatalogEndUserFullAccess** AWS managed policy.
- Users are able to validate and estimate the cost of any CloudFormation template, through the inline policy.
- By using the **CloudFormationRole** IAM role, users are able to create, update, or delete any CloudFormation stack that starts with **app/**.
- Users are able to use CloudFormation to create, update, or delete IAM roles that start with **app/**. The **PermissionsBoundary** IAM policy prevents users from escalating their privileges.
- Users can provision AWS Lambda, Amazon EventBridge, Amazon CloudWatch, Amazon Simple Storage Service (Amazon S3), and Amazon API Gateway resources only by using CloudFormation.

The following image shows how an authorized user, such as a developer, can create a new IAM role in a member account by using the permissions sets, IAM roles, and permissions boundaries described in this guide:

1. The user authenticates in IAM Identity Center and assumes the **DeveloperAccess** IAM role.
2. The user initiates the `cloudformation:CreateStack` action and assumes the **CloudFormationRole** IAM role.
3. The user initiates the `iam:CreateRole` action and uses CloudFormation to create a new IAM role.
4. The **PermissionsBoundary** IAM policy is applied to the new IAM role.



The **CloudFormationRole** role has the AdministratorAccess managed policy attached, but due to the **PermissionsBoundary** IAM policy, the **CloudFormationRole** role's effective permissions become equal to the **PermissionsBoundary** policy. The **PermissionsBoundary** policy references itself when allowing the `iam:CreateRole` action, which ensures that roles can be created only if the permissions boundary is applied.

Managing permissions for individuals

By using permissions sets, the permissions boundary, and the **CloudFormationRole** IAM role, you can limit the amount of permissions that you need to assign directly to individual principals. This helps you manage access as your company grows and helps you apply the security best practice of granting least privilege.

You can also use *service-linked roles*, which grant permissions to an AWS service to provision resources on your behalf. Instead of granting permissions to the IAM principal (user, user group, or role), you can grant the permissions to the service. For example, the service-linked roles for AWS Proton and AWS Service Catalog allow you to provision your own templates, resources, and environments, without assigning permissions to the IAM principal. For more information, see AWS services that work with IAM and Using service-linked roles (IAM documentation).

Another best practice is to limit the amount of access individuals have to the AWS Management Console. By limiting access to the console, you can require individuals to provision resources by

using infrastructure as code (IaC) technologies, such as AWS CloudFormation, HashiCorp Terraform, or Pulumi. Managing infrastructure through IaC you to track changes to resources over time and introduce mechanisms for approving changes, such as GitHub pull requests.

Network connectivity for a multi-account architecture

Connecting VPCs

Many companies use VPC peering in Amazon Virtual Private Cloud (Amazon VPC) to connect development and production VPCs. Using a VPC peering connection, you can route traffic between two VPCs by using private IP addressing. The connected VPCs can be in different AWS accounts and in different AWS Regions. For more information, see What is VPC peering (Amazon VPC documentation). As companies grow and the number of VPCs increases, maintaining peering connections between all of the VPCs can become a maintenance burden. You might also be limited by the maximum number of VPC peering connections per VPC. For more information, see the VPC peering connection quota (Amazon VPC documentation).

If you have multiple development, test, and staging environments that host non-production data across multiple AWS accounts, you might want to provide network connectivity between all of those VPCs but disallow any access to production environments. You can use AWS Transit Gateway to connect multiple VPCs across multiple accounts. You can separate the route tables to prevent development VPCs from communicating to production VPCs through the transit gateway, which acts as centralized router. For more information, see Centralized router (Transit Gateway documentation).

Transit Gateway also supports peering with other transit gateways, including those in different AWS accounts or AWS Regions. Because Transit Gateway is a fully managed, highly available service, you need to provision only one transit gateway for each Region.

For more information and detailed network architectures, see Building a Scalable and Secure Multi-VPC AWS Network Infrastructure (AWS Whitepaper).

Connecting applications

If you need to establish communication between applications in different AWS accounts in the same environment (such as production), you can use one of the following options:

- VPC peering or AWS Transit Gateway can provide connectivity at the network level if you want to open broad access to multiple IP addresses and ports.
- AWS PrivateLink creates endpoints in a private subnet of the VPC, and these endpoints are registered as DNS entries in Amazon Route 53 Resolver. By using DNS, applications can resolve

the endpoints and connect to the registered services, without requiring NAT gateways or internet gateways in the VPC.

- Amazon VPC Lattice associates services, such as applications, across multiple accounts and VPCs and collects them into a service network. Clients in VPCs associated with the service network can send requests to all other services that are associated with the service network, regardless of whether they're in the same account. VPC Lattice integrates with AWS Resource Access Manager (AWS RAM) so that you can share resources with other accounts or through AWS Organizations. You can associate a VPC with only one service network. This solution doesn't require use of VPC peering or AWS Transit Gateway to communicate across accounts.

Best practices for network connectivity

- Create an AWS account that you use for the centralized networking. Name this account **network-prod**, and use it for AWS Transit Gateway and Amazon VPC IP Address Manager (IPAM). Add this account to the **Infrastructure_Prod** organizational unit.
- Use AWS Resource Access Manager (AWS RAM) to share the transit gateway, VPC Lattice service networks, and IPAM pools with the rest of the organization. This allows any AWS account within your organization to interact with these services.
- By using IPAM pools to centrally manage IPv4 and IPv6 address allocations, you can allow your end-users to self-provision VPCs by using AWS Service Catalog. This helps you appropriately size VPCs and prevent overlapping IP address spaces.
- Use a centralized egress approach for traffic bound to the internet, and use a decentralized ingress approach for traffic coming into your environment from the internet. For more information, see Centralized egress and Decentralized ingress.

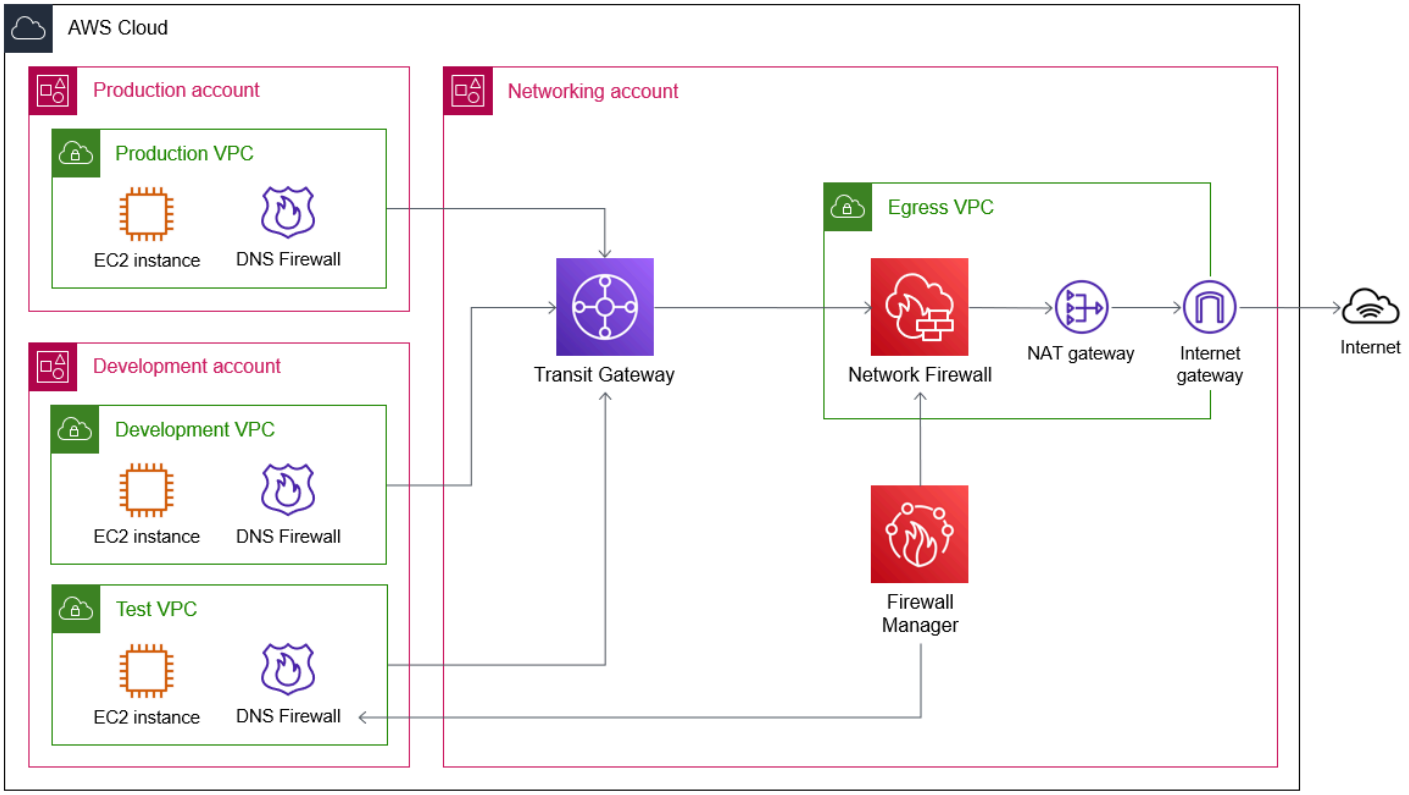
Centralized egress

Centralized egress is the principle of using a single, common inspection point for all network traffic destined to the internet. At this inspection point, you can allow traffic only to specified domains or only through specified ports or protocols. Centralizing egress also can help you reduce costs by eliminating the need to deploy NAT gateways in each of your VPCs in order to reach the internet. This is beneficial from a security perspective because it limits exposure to externally accessible malicious resources, such as malware command and control (C&C) infrastructure. For more information and architecture options for centralized egress, see Centralized egress to internet (AWS Whitepaper).

You can use AWS Network Firewall, which is a stateful, managed, network firewall and intrusion detection and prevention service, as a central inspection point for egress traffic. You set up this firewall in a dedicated VPC for egress traffic. Network Firewall supports stateful rules that you can use to limit internet access to specific domains. For more information, see Domain filtering (Network Firewall documentation).

You can also use the Amazon Route 53 Resolver DNS Firewall to limit egress traffic to specific domain names, primarily to prevent unauthorized exfiltration of your data. In DNS Firewall rules, you can apply domain lists (Route 53 documentation), which allow or deny access to specified domains. You can use AWS managed domain lists, which contain domain names that are associated with malicious activity or other potential threats, or you can create custom domain lists. You create DNS Firewall rule groups and then apply them to your VPCs. Outbound DNS requests route through a Resolver in the VPC for domain name resolution, and DNS Firewall filters the requests based on the rule groups applied to the VPC. Recursive DNS requests going to the Resolver don't flow through the transit gateway and Network Firewall path. Route 53 Resolver and DNS Firewall should be considered to be a separate egress path out of the VPC.

The following image shows a sample architecture for centralized egress. Before network communication begins, DNS requests are sent to the Route 53 Resolver, where the DNS firewall allows or denies resolution of the IP address used for communication. Traffic destined to the internet is routed to a transit gateway in a centralized networking account. The transit gateway forwards the traffic to Network Firewall for inspection. If the firewall policy permits the egress traffic, the traffic routes through an NAT gateway, through an internet gateway, and to the internet. You can use AWS Firewall Manager to centrally manage DNS Firewall rule groups and Network Firewall policies across your multi-account infrastructure.



Best practices for securing egress traffic

- Start in logging-only mode (Route 53 documentation). Change to block mode after you have validated that legitimate traffic isn't affected.
- Block DNS traffic going to the internet by using AWS Firewall Manager policies for network access control lists or by using AWS Network Firewall. All DNS queries should route through a Route 53 Resolver, where you can monitor them with Amazon GuardDuty (if enabled) and filter them with Route 53 Resolver DNS Firewall (if enabled). For more information, see Resolving DNS queries between VPCs and your network (Route 53 documentation).
- Use the AWS Managed Domain Lists (Route 53 documentation) in DNS Firewall and Network Firewall.
- Consider blocking high-risk, unused top-level domains, such as .info, .top, .xyz, or some country code domains.
- Consider blocking high-risk, unused ports, such as ports 1389, 4444, 3333, 445, 135, 139, or 53.
- As a starting point, you can use a deny list that includes the AWS managed rules. You can then work over time toward implementing an allow-list model. For example, instead of including only a strict list of fully qualified domain names in the allow list, begin by using some wildcards, such

as **.example.com*. You can even allow only the top-level domains you expect and block all others. Then, over time, narrow those down too.

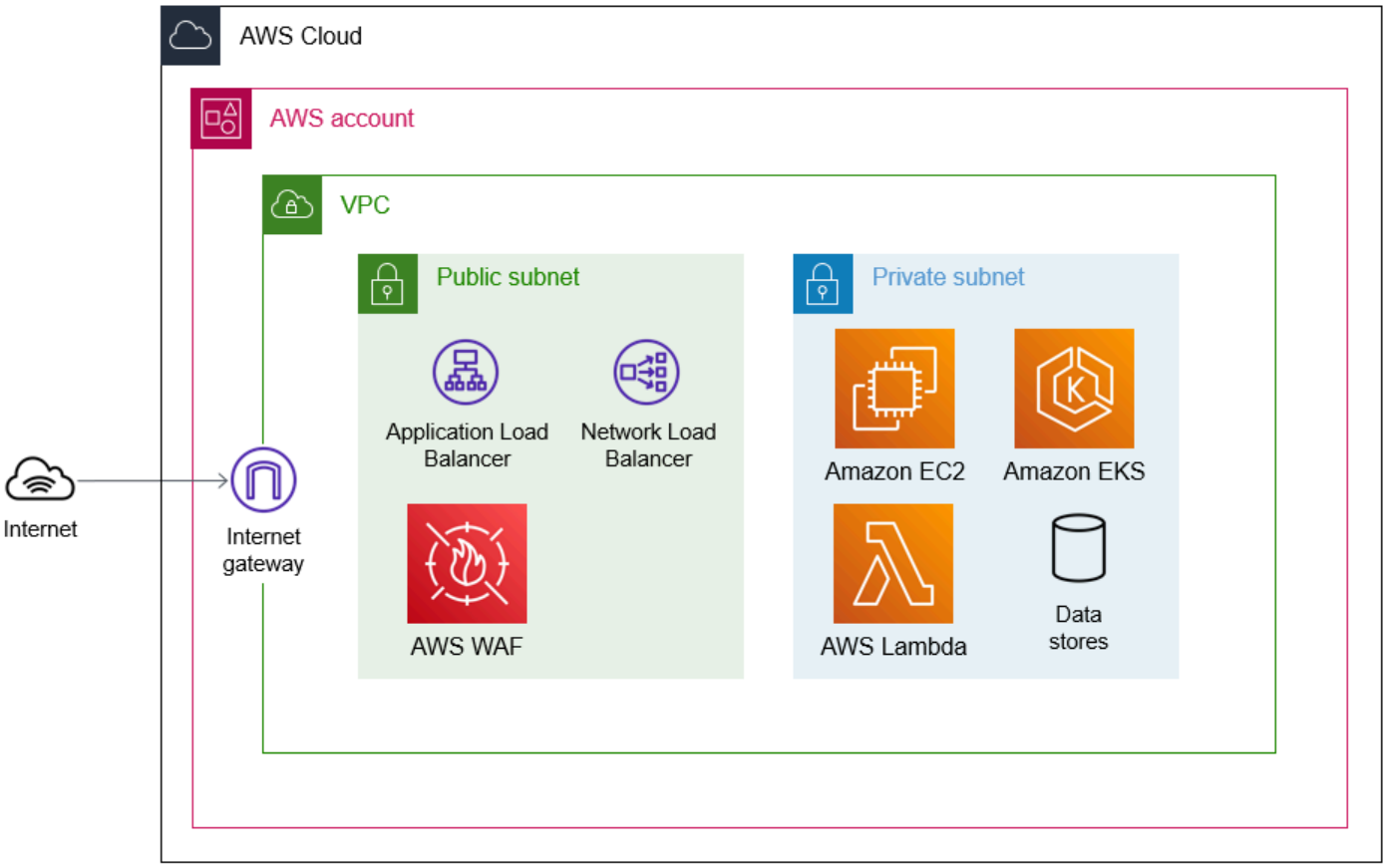
- Use Route 53 Profiles (Route 53 documentation) to apply DNS-related Route 53 configurations across many VPCs and in different AWS accounts.
- Define a process for handling exceptions to these best practices.

Decentralized ingress

Decentralized ingress is the principle of defining, at an individual account-level, how traffic from the internet reaches the workloads in that account. In multi-account architectures, one of the benefits of decentralized ingress is that each account can use the most appropriate ingress service or resource for its workloads, such as an Application Load Balancer, Amazon API Gateway, or Network Load Balancer.

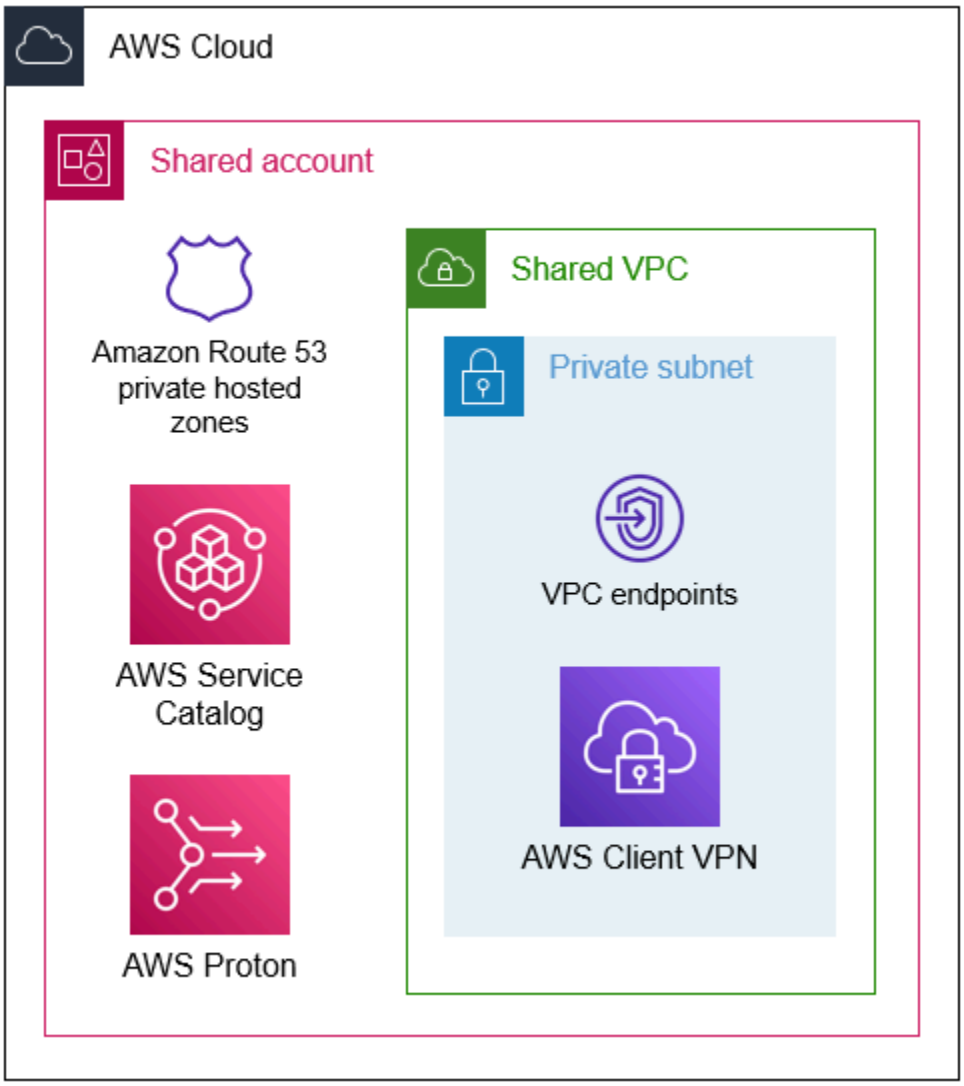
Although decentralized ingress means that you have to manage each account individually, you can centrally administer and maintain your configurations through AWS Firewall Manager. Firewall Manager supports protections such as AWS WAF and Amazon VPC security groups. You can associate AWS WAF to an Application Load Balancer, Amazon CloudFront, API Gateway, or AWS AppSync. If you are using an egress VPC and transit gateway, as described in Centralized egress, each spoke VPC contains public and private subnets. However, there is no need to deploy NAT gateways because traffic routes through the egress VPC in the networking account.

The following image shows an example of an individual AWS account that has a single VPC that contains an internet-accessible workload. Traffic from the internet accesses the VPC through an internet gateway and reaches load balancing and security services hosted in a public subnet. (A *public subnet* contains a default route to an internet gateway). Deploy load balancers into public subnets, and attach AWS WAF access control lists (ACLs) to help protect against malicious traffic, such as cross-site scripting. Deploy workloads that host applications into *private subnets*, which don't have direct access to and from the internet.



If you have a lot of VPCs in your organization, you might want to share common AWS services by creating interface VPC endpoints or private hosted zones in a dedicated and shared AWS account. For more information, see Access an AWS service using an interface VPC endpoint (AWS PrivateLink documentation) and Working with private hosted zones (Route 53 documentation).

The following image shows an example of an AWS account that hosts resources that can be shared across the organization. VPC endpoints can be shared across multiple accounts by creating them in a dedicated VPC. When you create a VPC endpoint, you can optionally have AWS manage the DNS entries for the endpoint. To share an endpoint, clear this option, and create the DNS entries in a separate Route 53 private hosted zone (PHZ). You can then associate the PHZ to all of the VPCs in your organization for centralized DNS resolution of the VPC endpoints. You also need to ensure that the transit gateway route tables include routes for the shared VPC to the other VPCs. For more information, see Centralized access to interface VPC endpoints (AWS Whitepaper).



A shared AWS account is also a good place to host AWS Service Catalog portfolios. A *portfolio* is a collection of IT services that you want to make available for deployment on AWS, and the portfolio contains configuration information for those services. You can create the portfolios in the shared account, share them to the organization, and then each member account imports the portfolio into its own regional Service Catalog instance. For more information, see Sharing with AWS Organizations (Service Catalog documentation).

Similarly, with AWS Proton, you can use the shared account to centrally manage your environment and service templates and then set up account connections with the organization member accounts. For more information, see Environment account connections (AWS Proton documentation).

Security incident response for a multi-account architecture

As you transition to multiple AWS accounts, it is important that you maintain visibility into security events that might occur within your organization. In Identity management and access control, you used AWS Control Tower to set up your landing zone. During that setup process, AWS Control Tower designated an AWS account for security. You should delegate administration of security services into the **security-tooling-prod** account and use this account to centrally managed these services.

This guide reviews the use of the following AWS services to help protect your AWS accounts and organization:

- Amazon GuardDuty
- Amazon Macie
- AWS Security Hub

Amazon GuardDuty

Amazon GuardDuty is a continuous security monitoring service that analyzes data sources, such as AWS CloudTrail event logs. For a complete list of supported data sources, see How Amazon GuardDuty uses its data sources (GuardDuty documentation). It uses threat intelligence feeds, such as lists of malicious IP addresses and domains, and machine learning to identify unexpected and potentially unauthorized and malicious activity within your AWS environment.

When you use GuardDuty with AWS Organizations, the management account in the organization can designate any account in the organization to be the GuardDuty *delegated administrator*. The delegated administrator becomes the GuardDuty administrator account for the Region. GuardDuty is automatically enabled in that :AWS Region, and the delegated administrator account has permissions to enable and manage GuardDuty for all accounts in the organization within that Region. For more information, see Managing GuardDuty accounts with AWS Organizations (GuardDuty documentation).

GuardDuty is a regional service. This means that you must enable GuardDuty in each Region that you want to monitor.

Best practices

- Enable GuardDuty in all supported AWS Regions. GuardDuty can generate findings about unauthorized or unusual activity, even in Regions that you aren't actively using. Pricing for GuardDuty is based on the number of analyzed events. Even in Regions where you aren't operating workloads, enabling GuardDuty is an effective and cost-efficient detection tool to alert you about potentially malicious activity. For more information about the Regions where GuardDuty is available, see Amazon GuardDuty service endpoints (AWS General Reference).
- Within every Region, delegate the **security-tooling-prod** account to administer GuardDuty for your organization. For more information, see Designating a GuardDuty delegated administrator (GuardDuty documentation).
- Configure GuardDuty to automatically enroll new AWS accounts as they are added to the organization. For more information, see *Step 3 - automate the addition of new organization accounts as members* in Managing accounts with AWS Organizations (GuardDuty documentation).

Amazon Macie

Amazon Macie is a fully managed data security and data privacy service that uses machine learning and pattern matching to help you discover, monitor, and protect sensitive data in Amazon Simple Storage Service (Amazon S3). You can export data from Amazon Relational Database Service (Amazon RDS) and Amazon DynamoDB an S3 bucket and then use Macie to scan the data.

When you use Macie with AWS Organizations, the management account in the organization can designate any account in the organization to be the Macie *administrator account*. The administrator account can enable and manage Macie for the member accounts in the organization, can access Amazon S3 inventory data, and can run sensitive data discovery jobs for the accounts. For more information, see Managing accounts with AWS Organizations (Macie documentation).

Macie is a regional service. This means that you must enable Macie in each Region that you want to monitor and that the Macie administrator account can manage member accounts only within the same Region.

Best practices

- Adhere to the Considerations and recommendations for using Macie with AWS Organizations (Macie documentation).

- Within every Region, delegate the **security-tooling-prod** account to administer Macie for your organization. To centrally manage Macie accounts in multiple AWS Regions, the management account must log in to each Region where the organization currently uses or will use Macie, and then designate the Macie administrator account in each of those Regions. The Macie administrator account can then configure the organization in each of those Regions. For more information, see Integrating and configuring an organization (Macie documentation).
- Macie provides a monthly free tier for sensitive data discovery jobs. If you might have sensitive data stored in Amazon S3, use Macie to analyze your S3 buckets as part of the monthly free tier. If you exceed the free tier, sensitive data discovery charges begin to accrue for your account.

AWS Security Hub

AWS Security Hub provides you with a comprehensive view of your security state in AWS. You can use it to check your environment against security industry standards and best practices. Security Hub collects security data from across all of your AWS accounts, services (including GuardDuty and Macie), and supported third-party partner products. Security Hub helps you analyze security trends and identify the highest priority security issues. Security Hub provides various security standards that you can enable to perform compliance checks in each AWS account.

When you use Security Hub with AWS Organizations, the management account in the organization can designate any account in the organization to be the Security Hub *administrator account*. The Security Hub administrator account can then enable and manage other member accounts in the organization. For more information, see Using AWS Organizations to manage accounts (Security Hub documentation).

Security Hub is a regional service. This means that you must enable Security Hub in each Region that you want to analyze, and in AWS Organizations, you must define the delegated administrator for each Region.

Best practices

- Adhere to the Prerequisites and recommendations (Security Hub documentation).
- Within every Region, delegate the **security-tooling-prod** account to administer Security Hub for your organization. For more information, see Designating a Security Hub administrator account (Security Hub documentation).
- Configure Security Hub to automatically enroll new AWS accounts when they are added into the organization.

- Enable the AWS Foundational Security Best Practices standard (Security Hub documentation) to detect when resources deviate from security best practices.
- Enable Cross-Region aggregation (Security Hub documentation) so that you can view and manage all of your Security Hub findings from a single Region.

Configuring backups for a multi-account architecture

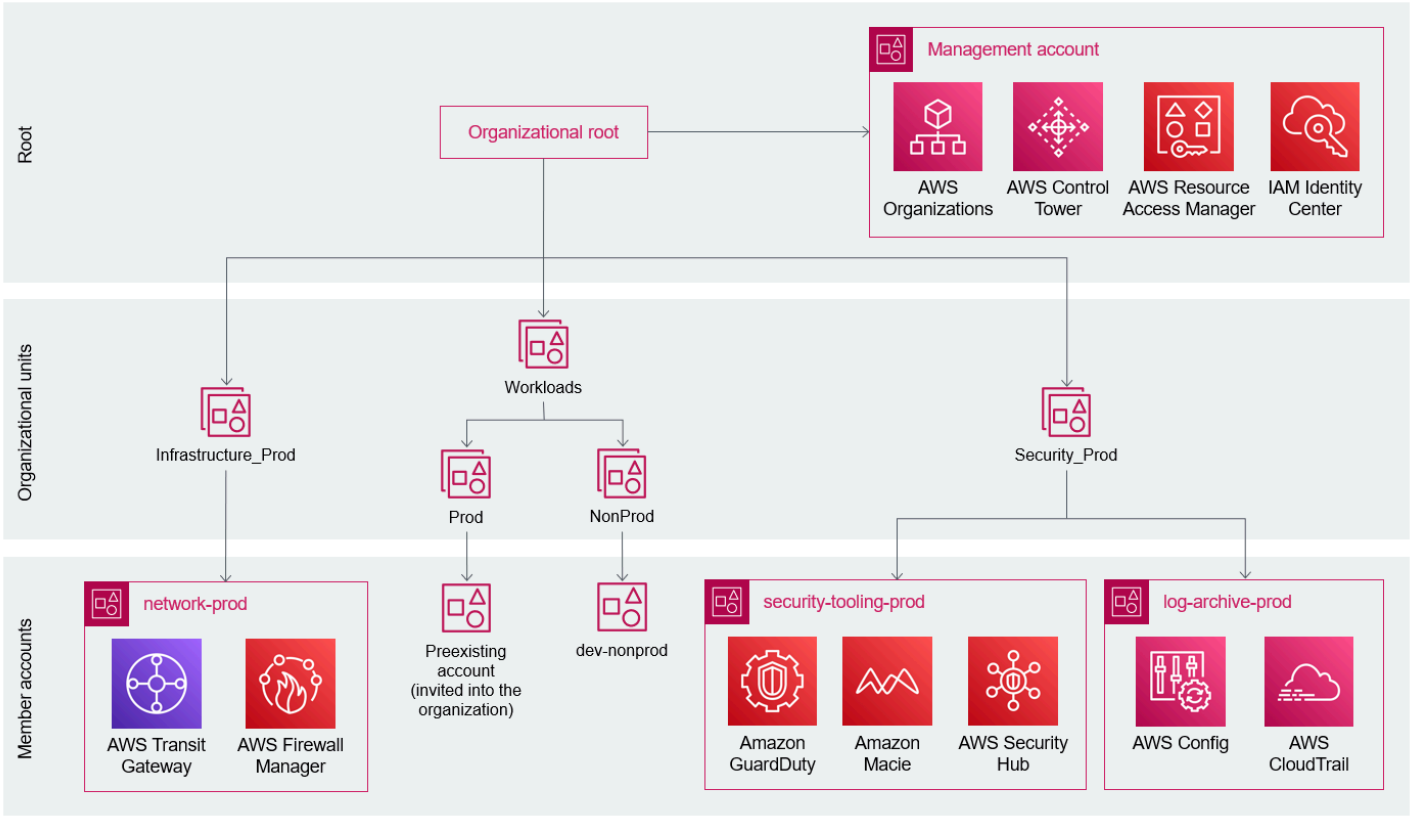
A comprehensive backup strategy is an essential part of a company's data protection plan to withstand, recover, and reduce any impact that might be sustained due to a security event. A backup policy helps you standardize and implement a backup strategy for the resources across all of the accounts in your organization. In a backup policy, you can configure and deploy backup plans for your resources. For more information, see Backup policies (AWS Organizations documentation). For more information, see Top 10 security best practices for securing backups in AWS (AWS Prescriptive Guidance).

Account migration when transitioning to a multi-account architecture

In Invite your preexisting account, you invited your preexisting account to join the **Workloads > Prod** organizational unit. This account is now managed as part of your organization.

You also provisioned a new **dev-nonprod** account in the **Workloads > NonProd** organizational unit. Team members should now be able to access the appropriate accounts through AWS IAM Identity Center. Remove any individual user accounts in AWS Identity and Access Management (IAM).

If you have followed the recommendations in this guide, your organization now has the following structure.



If there are workloads running within the preexisting account, you now migrate these workloads into independent accounts, according to the criteria you established in Define scoping criteria. Migrate any non-production workloads to the new **dev-nonprod** organizational unit, and migrate production workloads to the **network-prod** account. For more information about migrating common AWS resources, see the following section of this guide, Resource migration.

Resource replication or migration between AWS accounts

After migrating from a single AWS account to multi-account architecture, it's common to have production and non-production workloads running in the preexisting account. Migrating these resources to dedicated production and non-production accounts or organizational units helps you manage access and networking for these workloads. The following are some options for migrating common AWS resources into another AWS account.

This section focuses on strategies for replicating data between AWS accounts. You should strive to have your workloads be as stateless as possible to avoid needing to replicate compute resources between accounts. It is also beneficial to manage your resources through infrastructure as code (IaC) so that you can reprovision an environment in a separate AWS account.

This section reviews options for migrating the following data resources:

- AWS AppConfig configurations and environments
- AWS Certificate Manager certificates
- Amazon CloudFront distributions
- AWS CodeArtifact domains and repositories
- Amazon DynamoDB tables
- Amazon EBS volumes
- Amazon EC2 instances or AMIs
- Amazon ECR registries
- Amazon EFS file systems
- Amazon ElastiCache (Redis OSS) clusters
- AWS Elastic Beanstalk environments
- Elastic IP addresses
- AWS Lambda layers
- Amazon Lightsail instances
- Amazon Neptune clusters
- Amazon OpenSearch Service domains
- Amazon RDS snapshots
- Amazon Redshift clusters

- Amazon Route 53 domains and hosted zones
- Amazon S3 buckets
- Amazon SageMaker AI models
- AWS WAF web ACLs

AWS AppConfig configurations and environments

AWS AppConfig doesn't support directly copying its configuration to another AWS account. However, it is a best practice to manage the AWS AppConfig configurations and environments separately from the AWS accounts that are hosting the environments. For more information, see Cross-account configuration with AWS AppConfig (AWS blog post).

AWS Certificate Manager certificates

You can't directly export an AWS Certificate Manager (ACM) certificate from one account to another because the AWS Key Management Service (AWS KMS) key used to encrypt the certificate's private key is unique to each AWS Region and account. However, you can simultaneously provision multiple certificates with the same domain name across multiple accounts and Regions. ACM supports validating domain ownership by using DNS (recommended) or email. When you use DNS validation and create a new certificate, ACM generates a unique CNAME record for every domain on the certificate. The CNAME record is unique for each account, and it must be added to the Amazon Route 53 hosted zone or DNS provider within 72 hours for the certificate to be properly validated.

Amazon CloudFront distributions

Amazon CloudFront doesn't support migration of distributions from one AWS account to another AWS account. However, CloudFront does support the migration of an alternate domain name, also known as a *CNAME*, from one distribution to another. For more information, see How do I resolve the CNAMEAlreadyExists error when I set up a CNAME alias for my CloudFront distribution (AWS Knowledge Center).

AWS CodeArtifact domains and repositories

Although an organization can have multiple domains, the recommendation is to have a single production domain that contains all published artifacts. This helps development teams find and

share packages across an organization. The AWS account that owns the domain can be different from the account that owns any repositories associated to the domain. You can copy packages between repositories, but they must belong to the same domain. For more information, see Copy packages between repositories (CodeArtifact documentation).

Amazon DynamoDB tables

You can use one of the following services to migrate an Amazon DynamoDB table to a different AWS account:

- AWS Backup
- DynamoDB import and export to Amazon S3
- Amazon S3 and AWS Glue
- AWS Data Pipeline
- Amazon EMR

For more information, see How can I migrate my Amazon DynamoDB tables from one AWS account to another (AWS Knowledge Center).

Amazon EBS volumes

You can take a snapshot of an existing Amazon Elastic Block Store (Amazon EBS) volume, share the snapshot with the target account, and then create a copy of the volume in the target account. This effectively migrates the volume from one account to another. For more information, see How can I share an encrypted Amazon EBS snapshot or volume with another AWS account (AWS Knowledge Center).

Amazon EC2 instances or AMIs

It is not possible to directly transfer existing Amazon Elastic Compute Cloud (Amazon EC2) instances or Amazon Machine Images (AMIs) to a different AWS account. Instead, you can create a custom AMI in the source account, share the AMI with the target account, launch a new EC2 instance from the shared AMI in the target account, then deregister the shared AMI. For more information, see How do I transfer an Amazon EC2 instance or AMI to a different AWS account (AWS Knowledge Center).

Amazon ECR registries

Amazon Elastic Container Registry (Amazon ECR) supports both cross-account and cross-Region replication. You configure replication on the source registry and a registry permissions policy on the target registry. For more information, see Configuring cross-account replication (Amazon ECR documentation) and Allow the root user of a source account to replicate all repositories (Amazon ECR documentation).

Amazon EFS file systems

Amazon Elastic File System (Amazon EFS) supports cross-account and cross-Region replication. You can configure replication on the source file system. For more information, see Replicating file systems (Amazon EFS documentation).

Amazon ElastiCache (Redis OSS) clusters

You can use a backup of an Amazon ElastiCache (Redis OSS) database cluster to migrate it to a different account. For more information, see What are best practices for migrating my ElastiCache (Redis OSS) cluster (AWS Knowledge Center).

AWS Elastic Beanstalk environments

For AWS Elastic Beanstalk, you can use saved configurations (Elastic Beanstalk documentation) to migrate an environment to a different AWS account. For more information, see How do I migrate my Elastic Beanstalk environment from one AWS account to another AWS account (AWS Knowledge Center).

Elastic IP addresses

You can transfer Elastic IP addresses between AWS accounts that are in the same AWS Region. For more information, see Transfer Elastic IP addresses (Amazon VPC documentation).

AWS Lambda layers

By default, an AWS Lambda layer that you create is private to your AWS account. However, you can optionally share the layer with other AWS accounts or make it public. To copy a layer, you

reprovision it in another AWS account. For more information, see Configuring layer permissions (Lambda documentation).

Amazon Lightsail instances

You can create a snapshot of an Amazon Lightsail instance and export the snapshot to an Amazon Machine Image (AMI) and an encrypted snapshot of an Amazon EBS volume. For more information, see Exporting Amazon Lightsail snapshots to Amazon EC2 (Lightsail documentation). By default, the snapshot is encrypted with an AWS managed key created in AWS Key Management Service (AWS KMS). However, this type of KMS key cannot be shared between AWS accounts. Instead, you manually encrypt a copy of the AMI with a customer managed key that can be used from the target account. For more information, see Allowing users in other accounts to use a KMS key (AWS KMS documentation). You can then share the copied AMI with the target AWS account and launch a new EC2 instance for Lightsail from the copied AMI. For more information, see Launch an instance using the new launch instance wizard (Amazon EC2 documentation).

Amazon Neptune clusters

You can copy an automated snapshot of the Amazon Neptune database cluster to another AWS account. For more information, see Copying a database (DB) cluster snapshot (Neptune documentation).

You can also share a manual snapshot with up to 20 AWS accounts that can directly restore a DB cluster from the snapshot. For more information, see Sharing a DB Cluster Snapshot (Neptune documentation).

Amazon OpenSearch Service domains

To copy data between Amazon OpenSearch Service domains, you can use Amazon S3 to create a snapshot of the source domain and then restore the snapshot into a target domain in a different AWS account. For more information, see How do I restore data from an Amazon OpenSearch Service domain in another AWS account (AWS Knowledge Center).

If you have network connectivity between the AWS accounts, you can also use the cross-cluster replication (OpenSearch Service documentation) feature in OpenSearch Service.

Amazon RDS snapshots

For Amazon Relational Database Service (Amazon RDS), you can share manual snapshots of DB instances or clusters with up to 20 AWS accounts. You can then restore the DB instance or DB cluster from the shared snapshot. For more information, see How do I share manual Amazon RDS DB snapshots or Aurora DB cluster snapshots with another AWS account (AWS Knowledge Center).

You can also use AWS Database Migration Service (AWS DMS) to configure continuous replication between database instances in different accounts. However, this requires network connectivity between the accounts, such as VPC peering or a transit gateway.

Amazon Redshift clusters

To migrate an Amazon Redshift cluster to a different AWS account, you create a manual snapshot of the cluster in the source account, share the snapshot with the target AWS account, and then restore the cluster from the snapshot. For more information, see How do I copy an Amazon Redshift provisioned cluster to a different AWS account (AWS Knowledge Center).

Amazon Route 53 domains and hosted zones

You can transfer Amazon Route 53 domains between AWS accounts. For more information, see Transfer a domain to a different AWS account (Route 53 documentation).

You can also migrate a Route 53 hosted zone to a different AWS account. For more information about when this is recommended or required, see Migrate a hosted zone to a different AWS account (Route 53 documentation). When you migrate a hosted zone, you recreate it in the target AWS account. For instructions, see Migrating a hosted zone to a different AWS account (Route 53 documentation).

Amazon S3 buckets

You can use Amazon Simple Storage Service (Amazon S3) Same-Region Replication to copy objects between S3 buckets in the same AWS Region. For more information, see Replicating objects (Amazon S3 documentation). Note the following:

- Change the replica ownership to the AWS account that owns the destination bucket. For instructions, see Changing the replica owner (Amazon S3 documentation).

- Update the bucket owner conditions to reflect the AWS account ID of the target bucket. For more information, see Verifying bucket ownership with bucket owner condition (Amazon S3 documentation).
- As of April 2023, the **Bucket owner enforced setting** is enabled for newly created buckets, making bucket access control lists (ACLs) and object ACLs ineffective. For more information, see Amazon S3 Security Changes Are Coming (AWS blog post).
- You can use S3 Batch Replication (Amazon S3 documentation) to replicate objects that existed before replication was configured.

Amazon SageMaker AI models

SageMaker AI models are stored in an Amazon S3 bucket during training. By granting access to the S3 bucket from the target account, you can deploy a model stored in the source account to the target account. For more information, see How can I deploy an Amazon SageMaker AI model to a different AWS account (AWS Knowledge Center).

AWS WAF web ACLs

AWS WAF web access control lists (web ACLs) must reside in the same account as the resources they are associated to, such as Amazon CloudFront distributions, Application Load Balancers, Amazon API Gateway REST APIs, and AWS AppSync GraphQL APIs. You can use AWS Firewall Manager to centrally manage AWS WAF web ACLs across your entire organization in AWS Organizations and across Regions. For more information, see Getting started with AWS Firewall Manager AWS WAF policies (Firewall Manager documentation).

Billing considerations when transitioning to a multi-account architecture

If you use AWS Organizations for the transition to multiple AWS accounts, you can use the consolidated billing feature (AWS Organizations documentation). This feature provides a single, combined bill that shows the charges across multiple accounts.

The following are billing best practices and recommendations for transitioning to multiple accounts:

- If you need access to your historical billing data, before you accept the invitation to join an organization, create a Cost and Usage Report (AWS Cost and Usage Report documentation) to export the account's historical billing data to an Amazon Simple Storage Service (Amazon S3) bucket. After you accept the invitation to join the organization, the account's historical billing data is no longer accessible.
- If you need to combine two organizations, such as for a merger or acquisition, you can use the Account Assessment for AWS Organizations (AWS Solutions Library) to evaluate the resource-based policies in each organization and identify any potential issues before combining them.

Conclusion

Transitioning from a single AWS account to multiple accounts can feel overwhelming at first without an adoption strategy. By implementing a multi-account strategy, you can address many challenges that companies face when using a single AWS account:

- **Mistaking production data for development data** – You can grant different permissions and access by using AWS IAM Identity Center with separate permission sets production and non-production organizational units. Only highly privileged users should have access to the production database, and that access should be for limited periods of time and audited.
- **Production deployment affecting other business operations** – You can separate stakeholders by using multiple accounts and multiple environments. For example, you could create a dedicated sales demo environment, within a non-production account, so that you can plan deployments and releases when demos aren't occurring.
- **Slow production workload performance when testing development workloads** – Each AWS account has independent service quotas that govern each service. By using multiple accounts, you can limit the scope of one environment impacting another environment.
- **Distinguishing production costs from development costs** – Consolidated billing for the organization rolls up all of the costs at the AWS account level so that the finance team can see how much production costs compared to non-production environments, such as development, testing, and demo environments. You can also use tags and tagging policies to separate costs within an account.
- **Limiting access to sensitive data** – IAM Identity Center allows you to have separate access policies for a group of people associated to a specific account.
- **Controlling costs** – By using service-control policies (SCPs) in a multi-account architecture, you can disallow access to specific AWS services that might incur high costs for your organization. SCPs can deny all access to specific services or can limit the usage of a service to a specific type, such as restricting the types of Amazon Elastic Compute Cloud (Amazon EC2) instances that can be created.

Contributors

Contributors to this document include:

- Justin Plock, Principal Solutions Architect, AWS (principal author)
- Emily Arnautovic, Principal Architect, AWS
- Jason DiDomenico, Senior Solutions Architect, AWS
- Michael Leighty, Sr. Security Specialist Solutions Architect, AWS
- Jesse Lepich, Sr. Security Specialist Solutions Architect, AWS
- Rodney Lester, Principal Solutions Architect, AWS
- Israel Lopez Moriano, Solutions Architect, AWS
- George Rolston, Senior Solutions Architect, AWS
- Alex Torres, Senior Solutions Architect, AWS
- Dave Walker, Principal Solutions Architect, AWS

Resources

AWS Prescriptive Guidance

- AWS Startup Security Baseline (AWS SSB)
- AWS Security Reference Architecture (AWS SRA)
- Top 10 security best practices for securing backups in AWS

AWS blog posts

- How Setting Up IAM Users and IAM Roles Can Help Keep Your Startup Secure
- How to let builders create IAM resources while improving security and agility for your organization

AWS Whitepapers

- Organizing Your AWS Environment Using Multiple Accounts
- Establishing Your Cloud Foundation on AWS
- Building a Scalable and Secure Multi-VPC AWS Network Infrastructure

AWS code samples

- Automate the setup of security services with AWS Control Tower (GitHub)

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
Resource control policies	We added information about resource control policies to the Set up an organization section.	November 20, 2024
Centralized egress best practices	We updated the best practices for securing egress traffic.	May 6, 2024
Organization best practices	We updated the best practices for creating an organization in AWS Organizations.	December 4, 2023
Billing considerations	We added the Billing considerations section.	September 20, 2023
Resource migration, application connectivity, and Amazon VPC Lattice	We added the Resource migration and Connecting applications sections. We also added information about a new AWS service, Amazon Virtual Private Cloud (Amazon VPC) Lattice.	April 27, 2023
Account history and ABAC	We revised the Create a landing zone section to add information about how to make sure your new AWS accounts have usage history so that you can add them to to your AWS Control Tower landing zone. We also revised	January 6, 2023

	the Add initial users section to add information about how you can use attribute-based access control (ABAC) to pass the authentication method from an external SAML-base d IdP to AWS IAM Identity Center.	
Egress traffic networking	We revised the Centraliz ed egress section to add information about using Amazon Route 53 Resolver DNS Firewall to to limit egress traffic to specific domain names.	October 13, 2022
Security of egress traffic	We added Best practices for securing egress traffic.	October 6, 2022
Permissions boundaries	We improved the definition of a permissions boundary, and in the *Resources* section, we added a new link for more information about this topic.	September 22, 2022
Initial publication	—	September 6, 2022

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

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 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.

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.

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.

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, Amazon GuardDuty, AWS Trusted Advisor, Amazon Inspector, and custom AWS Lambda checks.

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.

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.

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. For more information, see Enabling data persistence in microservices.

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.

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.

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. For more information, see the Quantifying uncertainty in deep learning systems guide.

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.