Building a scalable vulnerability management program on AWS

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Building a scalable vulnerability management program on AWS

Table of Contents

Building a scalable vulnerability management program on AWS

Anna McAbee and Megan O'Neil, Amazon Web Services (AWS)

October 2023 (document history)

Depending on the underlying technology you're using, a variety of tools and scans can generate security findings in a cloud environment. Without processes in place to handle these findings, they can begin to accumulate, often leading to thousands to tens of thousands of findings in a short amount of time. However, with a structured vulnerability management program and proper operationalization of your tooling, your organization can handle and triage a large number of findings from diverse sources.

Vulnerability management focuses on discovering, prioritizing, assessing, remediating, and reporting on vulnerabilities. *Patch management*, on the other hand, focuses on patching or updating software to remove or remediate security vulnerabilities. Patch management is just one aspect of vulnerability management. Generally, we recommend establishing both a *patch-in-place process* (also known as a *mitigate-in-place* process) to address critical, patch-now scenarios, and a standard process that you run on a regular cadence in order to release patched Amazon Machine Images (AMIs), containers, or software packages. These processes help prepare your organization to respond quickly to a zero-day vulnerability. For critical systems in a production environment, using a patch-in-place process can be faster and more reliable than rolling out a new AMI across the fleet. For regularly scheduled patches, such as operating system (OS) and software patches, we recommend that you build and test using standard development processes, as you would any software-level change. This provides better stability for standard operating modes. You can use Patch Manager, a capability of AWS Systems Manager, or other third-party products as patch-in-place solutions. For more information about using Patch Manager, see Patch management in *AWS Cloud Adoption Framework: Operations Perspective*. Also, you can use EC2 Image Builder to automate the creation, management, and deployment of customized and up-to-date server images.

Building a scalable vulnerability management program on AWS involves managing traditional software and network vulnerabilities in addition to cloud configuration risks. A cloud configuration risk, such as an unencrypted Amazon Simple Storage Service (Amazon S3) bucket, should follow a similar triage and remediation process as a software vulnerability. In both of these cases, the application team must own and be accountable for the security of their application, including

the underlying infrastructure. This distribution of ownership is key for an effective and scalable vulnerability management program.

This guide discusses how to streamline the identification and remediation of vulnerabilities in order to reduce overall risk. Use the following sections to build and iterate on your vulnerability management program:

1. Prepare – Prepare your people, processes, and technology to identify, assess, and remediate vulnerabilities in your environment.
2. Triage and remediate – Route security findings to the relevant stakeholders, identify the appropriate remediation action, and then take the remediation action.
3. Report and improve – Use reporting mechanisms to identify opportunities for improvement, and then iterate on your vulnerability management program.

Building a cloud vulnerability management program often involves iteration. Prioritize recommendations in this guide and regularly revisit your backlog to stay current with technology changes and your business requirements.

Intended audience

This guide is intended for large enterprises that have three primary teams who are responsible for security related findings: a security team, a Cloud Center of Excellence (CCoE) or cloud team, and application (or *developer*) teams. This guide uses the most common enterprise operating models and builds upon those operating models to enable a more efficient response to security findings and improve security outcomes. Organizations using AWS might have different structures and different operating models; however, you can modify many of the concepts in this guide to fit different operating models and smaller organizations.

Objectives

This guide can help you and your organization:

- Develop policies to streamline vulnerability management and ensure accountability
- Establish mechanisms to distribute responsibility for security to the application teams
- Configure relevant AWS services according to best practices for scalable vulnerability management

- Distribute ownership of security findings
- Establish mechanisms to report on and iterate on your vulnerability management program
- Improve security finding visibility and improve overall security posture

Prepare your scalable vulnerability management program

Preparing to build a scalable vulnerability management program involves educating people, developing processes, and implementing the proper technology according to best practices. People, processes, and technology are equally important for an effective vulnerability management program, and you must tightly integrate them to manage vulnerabilities at scale.

This section of the guide reviews the foundational actions you can take to prepare your scalable vulnerability management program on AWS.

Topics

Define a vulnerability management plan

The first step when preparing your cloud vulnerability management program is defining your *vulnerability management plan*. This plan includes the policies and processes your organization follows. This plan should be documented and accessible by all stakeholders. A vulnerability management plan is a high-level document that typically includes the following sections:

- **Goals and scope** – Outline the goals, functions, and scope of vulnerability management.
- **Roles and responsibilities** – List the vulnerability management stakeholders and detail their responsibilities.
- **Vulnerability severity and prioritization definitions** – Determine how to classify the severity of a vulnerability and how to prioritize it.
- **Service level agreements (SLAs) for remediation** – For each severity level, define the maximum amount of time a remediation owner has to resolve a security finding. Because SLA compliance is

an integral part of having an effective and scalable vulnerability management program, consider how to track whether you're meeting these SLAs.

- **Exception process** – Detail the process of submitting, approving, and updating exceptions. This process should make sure that exceptions are legitimate, time-bound, and tracked.
- **Sources of vulnerability information** – List the sources or tools that generate security findings. For more information about AWS services that could be sources for security findings, see Configure AWS security services in this guide.

While these sections are common throughout companies of different sizes and industries, each organization's vulnerability management plan is unique. You need to build a vulnerability management plan that works best for your organization. Expect to iterate your plan over time to incorporate lessons learned and evolving technologies.

Distribute security ownership

The AWS shared responsibility model defines how AWS and its customers share responsibility for cloud security and compliance. In this model, AWS secures the infrastructure that runs all of the services offered in the AWS Cloud, and AWS customers are responsible for securing their data and applications.

You can mirror this model inside your organization and distribute the responsibilities between your cloud and application teams. This helps you scale your cloud security programs more effectively because the application teams take ownership of certain security aspects of their applications. The simplest interpretation of the shared responsibility model is that if you have access to configure the resource, then you are responsible for the security of that resource.

A key part of distributing security responsibilities to application teams is building self-service security tools that help your application teams automate. Initially, this can be a joint effort. The security team can translate security requirements into code-scanning tools, and then application teams can use those tools to build and share solutions with their internal developer community. This contributes to greater efficiencies across other teams that need to meet similar security requirements.

The following table outlines the steps for distributing ownership to application teams and provides examples.

Step	Action	Example
1	**Define your security requirements** – What are you trying to achieve? This might come from a security standard or compliance requirement.	An example security requirement is least-pri vilege access for application identities.
2	**Enumerate controls for a security requirement** – What does this requirement actually mean from a control perspective? What do I need to do to achieve this?	To achieve least-privilege for application identities, the following are two sample controls: • Use AWS Identity and Access Management (IAM) roles • Do not use wildcards in IAM policies
3	**Document guidance for the controls** – With these controls, what guidance can you provide to a developer to help them comply with the control?	Initially, you might start by documenting simple example policies, including secure and unsecure IAM policies and Amazon Simple Storage Service (Amazon S3) bucket policies. Next, you can embed policy-scanning solutions within continuous integrati on and continuous delivery (CI/CD) pipelines, such as using AWS Config rules for proactive evaluation.
4	**Develop reusable artifacts** – With the guidance, can you make it even easier and	You might create infrastru cture as code (IaC) to deploy IAM policies that follow the

Step	Action	Example
	develop reusable artifacts for developers?	principle of least privilege. You can store these reusable artifacts in a code repository.

Self-service might not work for all security requirements, but it can work for standard scenarios. By following these steps, organizations can empower their application teams to handle more of their own security responsibilities in a scalable way. Overall, the distributed responsibility model leads to more collaborative security practices within many organizations.

Develop a vulnerability disclosure program

For a defense-in-depth approach to vulnerability management, create a vulnerability disclosure program so that people inside or outside your organization can report security vulnerabilities or risks.

For people inside your organization, establish a process to submit risks or vulnerabilities. This can be done through a ticketing system or email. Regardless of the process you choose, it's essential that your employees are aware of the process and can easily submit any vulnerabilities or risks that they encounter.

For people outside your organization, establish an external webpage for submitting potential security vulnerabilities. As an example, see the AWS Vulnerability Reporting webpage. This webpage should also contain disclosure guidelines to help protect your organization's data and assets. A vulnerability disclosure program should not encourage potentially harmful activity, so it's essential that you have a clear policy with guidelines. Building a mature, responsible disclosure program is a goal to strive for as you mature your program. Most don't start with an external disclosure program, and it takes time to get it right.

Prepare your AWS environment

Before implementing any vulnerability management tooling, make sure that your AWS environment is architected to support a scalable vulnerability management program. The structure of your AWS accounts and your organization's tagging policies can simplify the process of building a scalable vulnerability management program.

Develop an AWS account structure

AWS Organizations helps centrally manage and govern an AWS environment as your business grows and scales its AWS resources. An *organization* in AWS Organizations consolidates your AWS accounts into logical groups, or *organizational units*, so that you can administer them as a single unit. You manage AWS Organizations from a dedicated account, called the *management account*. For more information, see AWS Organizations terminology and concepts.

We recommend that you manage your AWS multi-account environment in AWS Organizations. This helps create a full inventory of your company's accounts and resources. This complete asset inventory is a critical aspect of vulnerability management. Application teams should not use accounts that are outside of the organization.

AWS Control Tower helps you set up and govern an AWS multi-account environment, following prescriptive best practices. If you haven't already established a multi-account environment, AWS Control Tower is a good starting point.

We recommend using the dedicated account structure and best practices described in the AWS Security Reference Architecture (AWS SRA). The Security Tooling account should serve as your delegated administrator for your security services. More information about configuring your vulnerability management tooling in this account is provided later in this guide. Host applications in dedicated accounts in the Workloads organization unit (OU). This establishes strong workload-level isolation and explicit security boundaries for each application. For information about the design principles and benefits of using a multi-account approach, see Organizing Your AWS Environment Using Multiple Accounts (AWS whitepaper).

Having an intentional account structure and centrally managing security services from a dedicated account are critical aspects of a scalable vulnerability management program.

Define, implement, and enforce tags

Tags are key-value pairs that act as metadata for organizing your AWS resources. For more information, see Tagging your AWS resources. You can use tags to provide business context, such as business unit, application owner, environment, and cost center. The following table shows a set of sample tags.

Key	Value
BusinessUnit	HumanResources

Key	Value
CostCenter	CC101
ApplicationTeam	HumanResourcesTechnology
Environment	Production

Tags can help you prioritize findings. For example, it can help you:

- Identify the owner of a resource who is responsible for patching a vulnerability
- Track which applications or business units have a large number of findings
- Escalate the severity of findings for certain data classifications, such as personally identifiable information (PII) or payment card industry (PCI) data
- Identify the type of data in the environment, such as test data in a lower-level development environment or production data

To help you achieve effective tagging at scale, follow the instructions in Building your tagging strategy in *Best Practices for Tagging AWS Resources* (AWS whitepaper).

Monitor AWS security bulletins

We highly recommend monitoring AWS security bulletins on a regular and frequent basis. Security bulletins can notify you of any new security-related vulnerabilities, affected services, and applicable updates. You can also subscribe to an RSS feed for the security bulletins and build a process to ingest and address these bulletins as part of your vulnerability management program.

Configure AWS security services

AWS offers a variety of security services that are designed to help protect your AWS environment. For your vulnerability management program, we recommend that you enable the following AWS services in each account:

- Amazon GuardDuty helps detect active threats in your environment. A GuardDuty finding could help you identify an unknown vulnerability that was exploited in your environment. It could also help you understand the effects of an unpatched vulnerability.

- AWS Health provides ongoing visibility into your resource performance and the availability of your AWS services and accounts.
- AWS Identity and Access Management Access Analyzer analyzes the resource-based policies in your AWS environment to identify resources that are shared with an external entity. This can help you identify vulnerabilities associated with unintended access to your resources and data. For each instance of a resource shared outside of your account, IAM Access Analyzer generates a finding.
- Amazon Inspector is a vulnerability management service that continuously scans your AWS workloads for software vulnerabilities and unintended network exposure.
- AWS Security Hub helps you check your AWS environment against security industry standards and can identify cloud configuration risks. It also provides a comprehensive view of your AWS security state by aggregating findings from other AWS security services and third-party security tools.

This section discusses how to enable and configure Amazon Inspector and Security Hub to help you establish a scalable vulnerability management program.

Using Amazon Inspector in your vulnerability management program

Amazon Inspector is a vulnerability management service that continually scans your Amazon Elastic Compute Cloud (Amazon EC2) instances, Amazon Elastic Container Registry (Amazon ECR) container images, and AWS Lambda functions for software vulnerabilities and unintended network exposure. You can use Amazon Inspector to gain visibility and prioritize resolution of software vulnerabilities across your AWS environments.

Amazon Inspector continuously assesses your environment throughout the lifecycle of your resources. It automatically rescans resources in response to changes that could introduce a new vulnerability. For example, it rescans when you install a new package on an EC2 instance, when you install a patch, or when a new common vulnerabilities and exposures (CVE) that affects the resource is published. When Amazon Inspector identifies a vulnerability or an open network path, it produces a finding that you can investigate. The finding provides comprehensive information about the vulnerability, including the following:

- Amazon Inspector risk score
- Common Vulnerability Scoring System (CVSS) score
- Affected resource

- Vulnerability intelligence data about the CVE from Amazon, Recorded Future, and Cybersecurity and Infrastructure Security Agency (CISA)
- Remediation recommendations

For instructions on setting up Amazon Inspector, see Getting started with Amazon Inspector. The *Activate Amazon Inspector* step in this tutorial provides two configuration options: a standalone account environment and a multi-account environment. We recommend using the multi-account environment option if you want to monitor multiple AWS accounts that are members of an organization in AWS Organizations.

When you set up Amazon Inspector for a multi-account environment, you designate an account in the organization to be the Amazon Inspector delegated administrator. The delegated administrator can manage findings and some settings for organization members. For example, the delegated administrator can view the details of aggregated findings for all member accounts, enable or disable scans for member accounts, and review scanned resources. The AWS SRA recommends that you create a Security Tooling account and use it as the Amazon Inspector delegated administrator.

Using AWS Security Hub in your vulnerability managment program

Building a scalable vulnerability management program on AWS involves managing traditional software and network vulnerabilities in addition to cloud configuration risks. AWS Security Hub helps you check your AWS environment against security industry standards and can identify cloud configuration risks. Security Hub also provides a comprehensive view of your security state in AWS by aggregating security findings from other AWS security services and third-party security tools.

In the following sections, we provide best practices and recommendations for setting up Security Hub to support your vulnerability management program:

- Setting up Security Hub
- Enabling Security Hub standards
- Managing Security Hub findings
- Aggregating findings from other security services and tools

Setting up Security Hub

For setup instructions, see Setting up AWS Security Hub. To use Security Hub, you must enable AWS Config. For more information, see Enabling and configuring AWS Config in the Security Hub documentation.

If you are integrated with AWS Organizations, from the organization management account, you designate an account to be the Security Hub delegated administrator. For instructions, see Designating the Security Hub delegated administrator. The AWS SRA recommends that you create a Security Tooling account and use it as the Security Hub delegated administrator.

The delegated administrator automatically has access to configure Security Hub for all member accounts in the organization and to view findings associated with those accounts. We recommend that you enable AWS ConfigSecurity Hub in all AWS Regions and all of your AWS accounts. You can configure Security Hub to automatically treat new organization accounts as Security Hub member accounts. For instructions, see Managing member accounts that belong to an organization.

Enabling Security Hub standards

Security Hub generates findings by running automated and continuous security checks against *security controls*. The controls are associated with one or more *security standards*. The controls help you determine whether the requirements in a standard are being met.

When you enable a standard in Security Hub, Security Hub automatically enables the controls that apply to the standard. Security Hub uses AWS Config rules to perform most of its security checks for controls. You can enable or disable Security Hub standards at any time. For more information, see Security controls and standards in AWS Security Hub. For a complete list of standards, see Security Hub standards reference.

If your organization does not already have a preferred security standard, we recommend using the AWS Foundational Security Best Practices (FSBP) standard. This standard is designed to detect when AWS accounts and resource deviate from security best practices. AWS curates this standard and updates it regularly to cover new features and services. After triaging the FSBP findings, consider enabling other standards.

Managing Security Hub findings

Security Hub provides several features that help you address large volumes of findings from across your organization and understand the security state of your AWS environment. To help you manage findings, we recommend enabling the following two Security Hub features:

- Use cross-Region aggregation to aggregate findings, finding updates, insights, control compliance statuses, and security scores from multiple AWS Regions to a single aggregation Region.
- Use consolidated control findings to reduce finding noise by removing duplicate findings. When consolidated control findings is turned on in your account, Security Hub generates a single new finding or finding update for each security check of a control, even if a control applies to multiple enabled standards.

Aggregating findings from other security services and tools

In addition to generating security findings, you can use Security Hub to aggregate finding data from several AWS services and supported third-party security solutions. This section focuses on sending security findings to Security Hub. The next section, Prepare to assign security findings, discusses how you can integrate Security Hub with products that can receive findings from Security Hub.

There are many AWS services, third-party products, and open-source solutions available that you can integrate with Security Hub. If you are just getting started, we recommend doing the following:

1. **Enable integrated AWS services** – Most AWS service integrations that send findings to Security Hub are automatically activated after you enable both Security Hub and the integrated service. For your vulnerability management program, we recommend enabling Amazon Inspector, Amazon GuardDuty, AWS Health, and IAM Access Analyzer in each account. These services automatically send their findings to Security Hub. For a complete list of supported AWS service integrations, see AWS services that send findings to Security Hub.

   > **Note**
   >
   > AWS Health sends findings to Security Hub if one of the following conditions are met:
   >
   > - The finding is associated with an AWS security service
   > - The finding **typecode** contains the words `security`, `abuse`, or `certificate`
   > - The finding AWS Health service is `risk` or `abuse`

2. **Set up third-party integrations** – For a list of the currently supported integrations, see Available third-party partner product integrations. Select any additional tools that can send findings to or receive findings from Security Hub. You might already have some of these third-party tools. Follow the product instructions to configure integration with Security Hub.

Prepare to assign security findings

In this section, you set up the tools that your teams use to manage and assign security findings. This section includes the following options:

- Manage findings in existing tools and workflows – This option integrates AWS Security Hub with existing systems that your teams use to manage their daily tasks, such as a product backlog. This option is recommended for teams that have established tools to manage their workflows.
- Manage findings in Security Hub – This option configures notifications for Security Hub events so that the appropriate team receives an alert and can address the finding in Security Hub.

Decide which workflow would work best for your teams, and make sure that security findings can make it promptly to their respective owners.

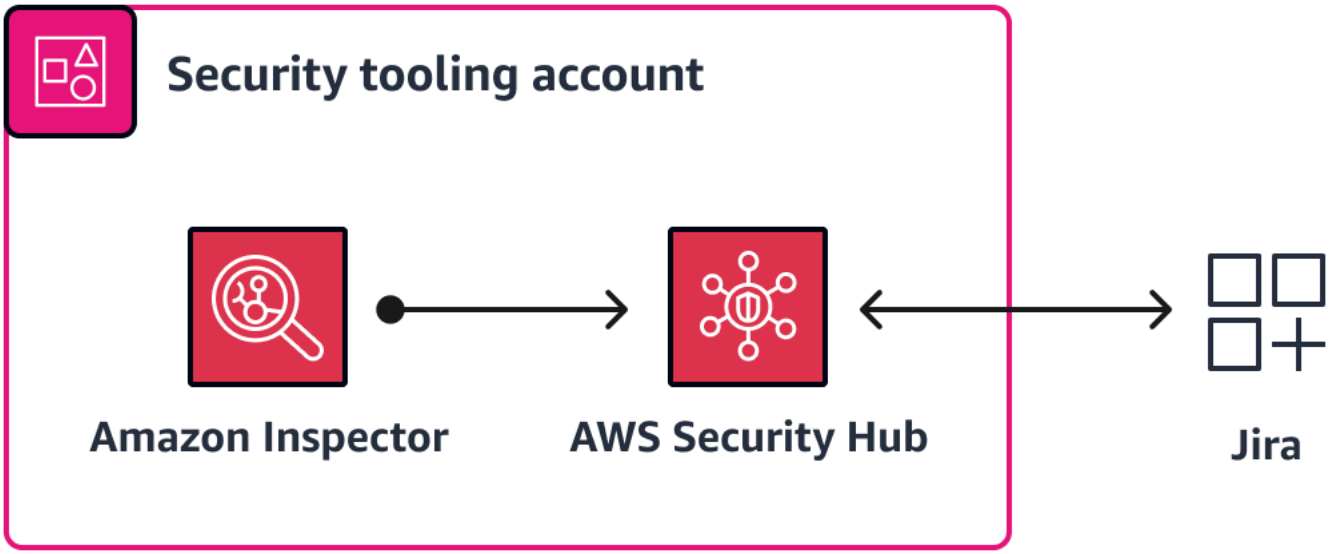
Manage findings in existing tools and workflows

We recommend additional Security Hub integrations for enterprise organizations that have established tools that teams use to manage or perform their daily tasks. You can import Security Hub finding data into several technology platforms. Examples include:

- Security information and event management (SIEM) systems help security teams triage operational security events. SIEM systems provide real-time analysis of security alerts that are generated by applications and network hardware.
- Governance, risk, and compliance (GRC) systems help compliance and governance teams monitor and report on risk management data. GRC tools are software applications that businesses can use to manage policies, assess risk, control user access, and streamline compliance. You might use GRC tools to integrate business processes, reduce costs, and improve efficiency.
- Product backlog and ticketing systems help application and cloud teams manage features and prioritize development tasks. Atlassian Jira and Microsoft Azure DevOps are examples of these systems.

Integrating Security Hub findings directly with these existing enterprise systems can improve mean time to recovery (MTTR) and security outcomes because the daily operational workflow doesn't have to change. Teams can respond and learn from security findings much faster because they don't have to use separate workflows and tools. Integration makes addressing security findings part of the normal, standard workflow.

Security Hub integrates with multiple third-party partner products. For a complete list and instructions, see Available third-party partner product integrations in the Security Hub documentation. Common integrations include Atlassian - Jira Service Management, Bidirectionally integrate AWS Security Hub with Jira software, and ServiceNow – ITSM. The following diagram shows how you can configure Amazon Inspector to send findings to Security Hub and then configure Security Hub to send all findings to Jira.



Manage findings in Security Hub

You can build a cloud-based notification system for Security Hub findings by using Amazon EventBridge rules and Amazon Simple Notification Service (Amazon SNS) topics. This system notifies the appropriate team about a finding when it is created. For this approach, the multi-account strategy described in Develop an AWS account structure is critical because applications are separated into dedicated accounts. This helps you notify the correct teams for each finding.

Security or cloud teams might choose to receive events from all AWS accounts. In this case, build an EventBridge rule within the Security Hub delegated administrator account and subscribe an Amazon SNS topic that notifies these teams. For application teams, configure an EventBridge rule and SNS topic within their respective application accounts. When a Security Hub finding occurs within an application account, the responsible team is notified about the finding.

Security Hub already automatically sends all new findings and all updates to existing findings to EventBridge as **Security Hub Findings - Imported** events. Each **Security Hub Findings - Imported** event contains a single finding. You can apply filters on EventBridge rules so that a finding initiates

the rule only if the finding matches the filters. For instructions, see Configuring an EventBridge rule for automatically sent findings. For more information about creating and subscribing Amazon SNS topics, see Configuring Amazon SNS.

Consider the following when using this approach:

- For application teams, create EventBridge rules within each AWS account and AWS Region where the application is hosted.
- For security and cloud teams, create EventBridge rules in the Security Hub delegated administrator account. This notifies teams about all findings in the member accounts.
- Amazon SNS sends a notification each day if the status of the security finding is `NEW`. If you want to turn off the daily notifications, you can create a custom AWS Lambda function that changes the status of the finding from `NEW` to `NOTIFIED` after the Amazon SNS subscriber receives the notification.

Triage and remediate security findings in your AWS environment

Triaging a security finding involves routing the finding to the appropriate stakeholder, assessing and prioritizing the finding, then remediating it. This section reviews each of these steps in detail and provides recommendations for scalability and efficiency. It also includes examples to help illustrate the triage and remediation process.

Topics

Define ownership of security findings

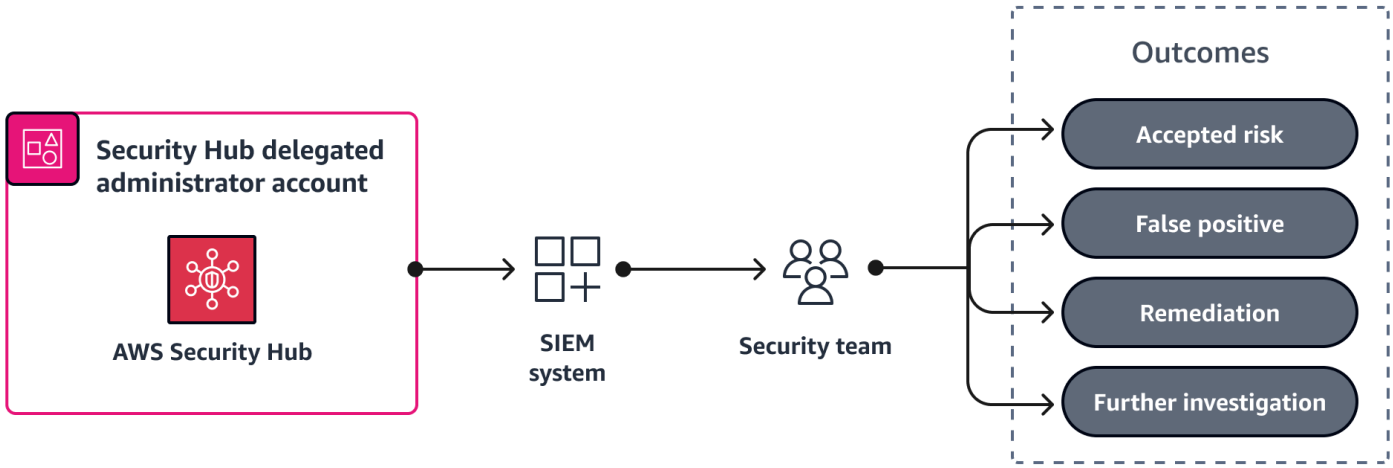
Defining an ownership model to triage security findings can be challenging, but it doesn't have to be. The security landscape changes constantly, and practitioners must be flexible to adapt to these changes. Adopt a flexible approach to developing your ownership model for security findings. Your initial model should enable your teams to act right away. We recommend starting with basic ownership logic and refining that logic over time. If you delay to define the perfect ownership criteria, the number of security findings will continue to grow.

To facilitate assignment of findings to the appropriate teams and resources, we recommend integrating AWS Security Hub with any existing systems that your teams use to manage their daily tasks. For example, you can integrate Security Hub with security information and event management (SIEM) systems or product backlog and ticketing systems. For more information, see Prepare to assign security findings in this guide.

The following is an example of an ownership model that you can use as a starting point:

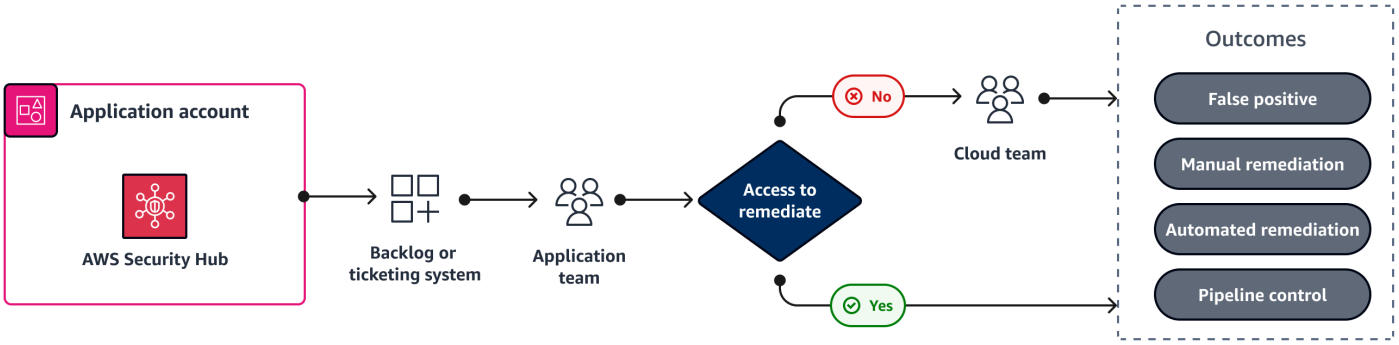
- **The security team reviews potentially active threats and helps assess and prioritize security findings**. The security team has the expertise and the tools to properly evaluate context. They understand the additional security-related data that helps them assess and prioritize vulnerabilities and investigate threat-detection events. If finding severity or additional tuning are

needed, see the Assess and prioritize security findings section in this guide. For an example, see Security team example in this guide.



- **Distribute security findings between the cloud and application teams** – As discussed in the Distribute security ownership section, the team that has access to configure the resource is responsible for its secure configuration. Application teams are responsible for the security findings related to the resources that they build and configure, and the cloud team is responsible for security findings related to the wide-reaching configurations. In most cases, application teams do not have access to change wide-reaching configurations and AWS services, such as AWS Control Tower, service control policies (SCPs) in AWS Organizations, networking-related VPC configurations, and AWS IAM Identity Center.

  For multi-account environments that separate applications into dedicated accounts, you can usually integrate security-related findings for the account into the application's backlog or ticketing system. From that system, the cloud team or application team can address the finding. For examples, see Cloud team example or Application team example in this guide.



- **Assign remaining, unresolved findings to the cloud team** – Residual findings might be related to default settings or wide-reaching configurations that the cloud team can address. This

team likely has the most historical knowledge and access to resolve the finding. Overall, this is typically a significantly smaller subset of the total findings.

Assess and prioritize security findings

A critical component of an effective vulnerability management program is the ability to assess and prioritize security findings. This is where pulling in context, organizational history, and tuning detection systems comes into place. Prioritization of security findings helps establish the appropriate speed for response level.

For Amazon Inspector, AWS Security Hub, and Amazon GuardDuty, findings contain a severity label or score. We recommend prioritizing the investigation of all critical and high severity findings in Security Hub, including findings related to the Foundational Security Best Practices (FSBP) standard, Amazon Inspector, and GuardDuty. Finding severity labels are scores are determined as follows:

- The Amazon Inspector score is a highly contextualized score for each finding. It's calculated by correlating Common Vulnerability Scoring System (CVSS) base score information with network reachability results and exploitability data. Using this score, you can prioritize findings to focus on the most critical findings and vulnerable resources. In addition to the score, Amazon Inspector also provides enhanced vulnerability intelligence about Common Vulnerabilities and Exposures (CVE). This is a summary of the available intelligence about the CVE from Amazon as well as industry-standard security intelligence sources, such as Recorded Future and Cybersecurity and Infrastructure Security Agency (CISA). For example, Amazon Inspector can provide the names of known malware kits used to exploit a vulnerability. For more information, see Vulnerability Intelligence.
- Each GuardDuty finding has an assigned severity level and value that reflects the finding's potential risk to your environment. This level and value are determined by AWS security engineers. For example, a `High` severity level indicates that a resource is compromised and is actively being used for unauthorized purposes. We recommend that you treat a `High` severity GuardDuty finding as a priority and immediately remediate to prevent further unauthorized use.
- The severity of an Security Hub control finding is determined by the difficulty to exploit and the likelihood of compromise. The difficulty is determined by the amount of sophistication or complexity that is required to use the weakness to carry out a threat scenario. The likelihood of compromise indicates how likely it is that the threat scenario will result in a disruption or breach of your AWS services or resources.

In order to tune findings, you can suppress or archive specific findings directly in the respective service console or by using the service's API. In addition, you can make changes to findings in Security Hub by using automation rules. GuardDuty and Amazon Inspector findings are automatically sent to Security Hub. You can use automation rules to automatically update (such as changing the severity) or suppress findings in near real-time, based on criteria that you define. As you create automation rules, we recommend adding context to the rule description, such as the date of creation or modification, who created it, and why the rule is needed. This information is often helpful for future reference.

Remediate security findings

After assessing and prioritizing a finding, the next action is remediating the finding. There are many different actions you could take to remediate a finding. For software vulnerabilities, you might update the operating system or apply a patch. For cloud configuration findings, you might update the resource configuration. In general, the actions you take to remediate can be grouped into one of the following outcomes:

- **Manual remediation** – You manually provide a fix to the vulnerability, such as modifying the properties of an AWS resource to enable encryption. If the finding is from one a managed check in Security Hub, then the finding includes a link to instructions for manually remediating the finding.
- **Reusable artifact** – You update the infrastructure as code (IaC) to fix the vulnerability and know that others could benefit from a similar solution. Consider uploading the updated IaC and a brief summary of the resolution to an internal shared code repository.
- **Automated remediation** – The vulnerability is automatically remediated through mechanisms you created.
- **Pipeline control** – You apply a control within your continuous integration and continuous delivery (CI/CD) pipeline that prevents deployment if the vulnerability is present.
- **Accepted risk** – You take no action or implement a compensating control, and you accept the risk that the vulnerability presents. Track the accepted risk in a dedicated location, such as a risk registry.
- **False positive** – You take no action because you have determined the finding didn't correctly identify a vulnerability.

A complete list of the various actions you can take and tools you can use to remediate a vulnerability is out of scope for this guide. However, there are some services and tools that you can help you remediate vulnerabilities at scale that are worth noting, including:

- Patch Manager, a capability of AWS Systems Manager, automates the process of patching managed nodes with both security-related updates and other types of updates. You can use Patch Manager to apply patches for both operating systems and applications.
- AWS Firewall Manager helps you centrally configure and manage firewall rules across your accounts and applications in AWS Organizations. As new applications are created, Firewall Manager makes it easier to bring new applications and resources into compliance by enforcing a common set of security rules.
- Automated Security Response on AWS is an AWS Solution that works with Security Hub and provides predefined response and remediation actions based on industry compliance standards and best practices for security threats.

Examples of triaging and remediating security findings

This section provides examples of the triage process for the security, cloud, and application teams. It discusses the types of findings each team commonly addresses and provides an example of how to respond. High-level remediation guidance is also included.

The following examples are included in this section:

- Security team example: Creating a Security Hub automation rule
- Cloud team example: Changing VPC configurations
- Application team example: Creating an AWS Config rule

Security team example: Creating a Security Hub automation rule

The security team receives findings related to threat detection, including Amazon GuardDuty findings. For a complete list of GuardDuty finding types that are categorized by AWS resource type, see Finding types in the GuardDuty documentation. Security teams must be familiar with all of these finding types.

For this example, the security team is accepting the level of associated risk for security findings in an AWS account that is used strictly for learning purposes and does not include important or sensitive data. The name of this account is `sandbox`, and the account ID is `123456789012`. The

security team can create an AWS Security Hub automation rule that suppresses all GuardDuty findings from this account. They can either create a rule from a template, which covers many common use cases, or they can create a custom rule. In Security Hub, we recommend previewing the results of the criteria to confirm that the rule returns the intended findings.

> **Note**
>
> This example highlights the functionality of automation rules. We don't recommend suppressing all GuardDuty findings for an account. Context matters, and each organization must choose which findings to suppress based on data type, classification, and mitigation controls.

The following are the parameters used to create this automation rule:

- **Rule:**
  - **Rule name** is `Suppress findings from Sandbox account`
  - **Rule description** is `Date: 06/25/23 Authored by: John Doe Reason: Suppress GuardDuty findings from the sandbox account`
- **Criteria:**
  - `AwsAccountId` = `123456789012`
  - `ProductName` = `GuardDuty`
  - `WorkflowStatus` = `NEW`
  - `RecordState` = `ACTIVE`
- **Automated action:**
  - `Workflow.status` is `SUPPRESSED`

For more information, see Automation rules in the Security Hub documentation. Security teams have many options for investigating and remediating findings for detected threats. For extensive guidance, see the AWS Security Incident Response Guide. We recommend reviewing this guide to confirm that you have established strong incident response processes.

Cloud team example: Changing VPC configurations

The cloud team is responsible for triaging and remediating security findings that have common trends, such as changes to AWS default settings that might not suit your use case. These findings

tend to affect many AWS accounts or resources, such as VPC configurations, or they include a restriction that should be placed across the entire environment. For the most part, the cloud team makes manual, one-time changes, such as adding or updating a policy.

After your organization has used an AWS environment for some time, you might find a set of anti-patterns developing. An *anti-pattern* is a frequently used solution for a recurring issue where the solution is counter-productive, ineffective, or less effective than an alternative. As an alternative to these anti-patterns, your organization can use environment-wide restrictions that are more effective, such as AWS Organizations service control policies (SCPs) or IAM Identity Center permissions sets. SCPs and permissions sets can provide additional restrictions for resource types, such as preventing users from configuring a public Amazon Simple Storage Service (Amazon S3) bucket. Although it can be tempting to restrict every possible security configuration, there are policy size limits for SCPs and permissions sets. We recommend a balanced approach to preventative and detective controls.

The following are some controls from the AWS Security Hub Foundational Security Best Practices (FSBP) standard that the cloud team might be responsible for:

- [EC2.2] The VPC default security group should not allow inbound and outbound traffic
- [EC2.6] VPC flow logging should be enabled in all VPCs
- [EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests
- [CloudTrail.1] CloudTrail should be enabled and configured with at least one multi-Region trail that includes read and write management events
- [Config.1] AWS Config should be enabled

For this example, the cloud team is addressing a finding for FSBP control EC2.2. The documentation for this control recommends not using the default security group because it allows broad access through the default inbound and outbound rules. Because the default security group cannot be deleted, the recommendation is to change the rule settings to restrict inbound and outbound traffic. To efficiently address this issue, the cloud team should use established mechanisms to modify the security group rules for all VPCs because each VPC has this default security group. In most cases, cloud teams manage VPC configurations by using AWS Control Tower customizations or an infrastructure as code (IaC) tool, such as HashiCorp Terraform or AWS CloudFormation.

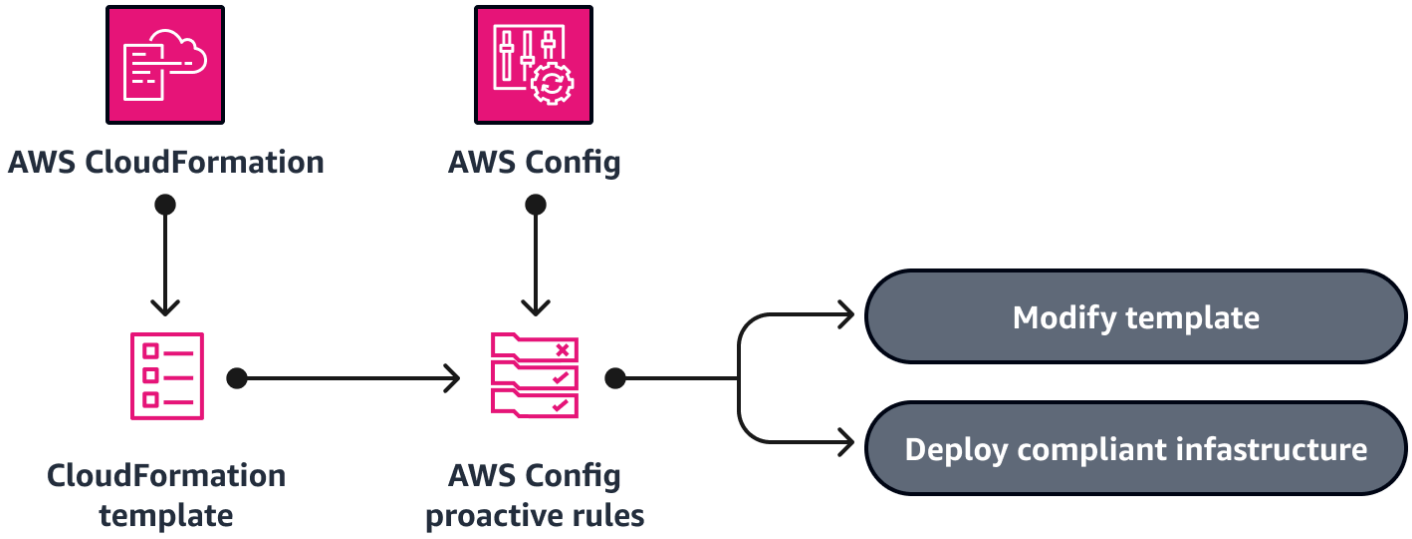
Application team example: Creating an AWS Config rule

The following are some controls from the Security Hub Foundational Security Best Practices (FSBP) security standard that the application or development team might be responsible for:

- [CloudFront.1] CloudFront distributions should have a default root object configured
- [EC2.19] Security groups should not allow unrestricted access to ports with high risk
- [CodeBuild.1] CodeBuild GitHub or Bitbucket source repository URLs should use OAuth
- [ECS.4] ECS containers should run as non-privileged
- [ELB.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS

For this example, the application team is addressing a finding for FSBP control EC2.19. This control checks whether unrestricted incoming traffic for the security groups is accessible to the specified ports that have the highest risk. This control fails if any of the rules in a security group allow ingress traffic from `0.0.0.0/0` or `::/0` for those ports. The documentation for this control recommends deleting the rules that allow this traffic.

In addition to addressing the individual security group rule, this is a great example of a finding that should result in a new AWS Config rule. By using the proactive evaluation mode, you can help prevent the deployment of risky security group rules in the future. Proactive mode evaluates resources before they have been deployed so that you can prevent misconfigured resources and their associated security findings. When implementing a new service or a new functionality, application teams can run rules in proactive mode as part of their continuous integration and continuous delivery (CI/CD) pipeline to identify noncompliant resources. The following image shows how you can use a proactive AWS Config rule to confirm that the infrastructure defined in an AWS CloudFormation template is compliant.



Another important efficiency can be gained in this example. When an application team creates a proactive AWS Config rule, they can share it in a common code repository so that other application teams can use it.

Each finding associated with a Security Hub control contains details about the finding and a link to the instructions for remediating the issue. Although cloud teams might encounter findings that require manual, one-time remediation, when appropriate, we recommend building proactive checks that identify issues as early as possible in the development process.

Report and improve your vulnerability management program

Effective reporting for vulnerability management involves reviewing data, monitoring trends, and sharing knowledge. This provides visibility and helps teams improve their organizations security posture in the AWS Cloud.

Conduct monthly security operations meetings

Monthly security operations meetings are an effective mechanism to promote continued ownership, accountability, and alignment across teams. In the meeting, the stakeholders from the security, cloud, and application teams review data for outstanding security findings, findings outside of service level agreements (SLAs), and the teams that have the most findings.

These meetings help your teams identify anti-patterns, such as opportunities to add more restrictions. Preventative controls and automation opportunities can also be discovered and shared. The meetings also help identify what is working and not working well within the vulnerability management program so that you can make improvements.

By reviewing data, identifying anti-patterns and issues, and sharing information about controls and automations, teams can gain valuable insights and make ongoing refinements that can strengthen their security posture and reduce their security-related SLAs.

Use Security Hub insights to identify anti-patterns

AWS Security Hub insights can also help you identify anti-patterns and track your progress in remediating findings. A Security Hub *insight* is a collection of related findings. It identifies a security area that requires attention and intervention. Security Hub insights can help you identify specific requirements and develop reports. Security Hub offers several built-in, managed insights. To track security issues that are unique to your AWS environment and usage, you can create custom insights.

Conclusion and next steps

In summary, an effective vulnerability management program requires thorough preparation and requires that you enable the right tools and integrations, fine-tune those tools, efficiently triage issues, and continuously report and improve. By following the best practices in this guide, organizations can build a scalable vulnerability management program on AWS to help secure their cloud environments.

You can expand on this program to include additional security-related vulnerabilities and findings, such as application security vulnerabilities. AWS Security Hub supports custom product integrations. Consider using Security Hub as the integration point for additional security tools and products. This integration allows you to take advantage of the processes and workflows you've already established in your vulnerability management program, such as the direct integration with product backlogs and the monthly security review meetings.

The following table summarizes the phases and action items described in this guide.

Phase	Action items
Prepare	• Define a vulnerability management plan. • Distribute ownership of findings. • Develop vulnerability disclosure program. • Develop an AWS account structure. • Define, implement, and enforce tags. • Monitor AWS security bulletins. • Enable Amazon Inspector with a delegated administrator. • Enable Security Hub with a delegated administrator. • Enable Security Hub standards. • Set up Security Hub cross-Region aggregati on. • Enable consolidated control findings in Security Hub.

Phase	Action items
	• Set up and manage Security Hub integrations, including applicable downstream integrations with SIEM, GRC, or product backlog or ticketing systems
Triage and remediate	• Route findings based on multi-account strategy. • Route findings to security, cloud, and application or developer teams. • Tune security findings to make sure that they are actionable for your specific environment. • Develop automated remediation mechanisms, when possible. • Implement CI/CD pipeline controls or other guardrails that help prevent security findings, when possible. • Use Security Hub automation rules to escalate or suppress findings.
Report and improve	• Hold monthly security operations meetings. • Use Security Hub insights to identify anti-patterns.

Resources

AWS service documentation

- Product integrations (AWS Security Hub)
- Integrating AWS Security Hub in Jira Service Management Cloud (AWS Security Hub)
- Automation rules (AWS Security Hub)
- Proactive evaluation rules (AWS Config)
- Patch Manager (AWS Systems Manager)

Other AWS resources

- Best practices for tagging AWS resources (AWS whitepaper)
- Automated Security Response on AWS (AWS Solutions Library)
- AWS Security Incident Response Guide (AWS Technical Guide)
- AWS security bulletins

Document history

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

Change	Description	Date
Initial publication	—	October 12, 2023

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 AWS CodeCommit. 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, AWS Panorama offers devices that add CV to on-premises camera networks, and Amazon SageMaker 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.

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.

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.

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.

G

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.

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.

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 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.

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

An approach in system engineering that takes privacy into account throughout the whole engineering 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.

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).

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.

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 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.

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.

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.