AWS Prescriptive Guidance

AWS Security Reference Architecture

AWS Prescriptive Guidance: AWS Security Reference Architecture

Table of Contents

AWS Security Reference Architecture (AWS SRA)

AWS Professional Services team

June 2021

The Amazon Web Services (AWS) Security Reference Architecture (AWS SRA) is a holistic set of guidelines for deploying the full complement of AWS security services in a multi-account environment. It can be used to help design, implement, and manage AWS security services so that they align with AWS best practices. The recommendations are built around a single-page architecture that includes AWS security services—how they help achieve security objectives, where they can be best deployed and managed in your AWS accounts, and how they interact with other security services. This overall architectural guidance complements detailed, service-specific recommendations such as those found on the AWS security website.

The architecture and accompanying recommendations are based on our collective experiences with AWS enterprise customers. This document is a reference—a comprehensive set of guidance for using AWS services to secure a particular environment—and the solution patterns in the AWS SRA code repository (p. 45) were designed for the specific architecture illustrated in this reference. Each enterprise has some unique requirements. As a result, the design of your AWS environment may differ from the examples provided here. **You will need to modify and tailor these recommendations to suit your individual environment and security needs.** Throughout the document, where appropriate, we suggest options for frequently seen alternative scenarios.

The AWS SRA is a living set of guidance and will be updated periodically based on new service and feature releases, customer feedback, and the constantly changing threat landscape. Each update will include the revision date and the associated change log (p. 50).

Although we rely on a one-page diagram as our foundation, an architecture goes deeper than a single block diagram and must be built on a well-structured foundation of fundamentals and security principles. You can use this document in two ways: as a narrative or as a reference. The topics are organized as a story, so you can read them from the beginning (foundational security guidance) to the end (discussion of code samples you can implement). Alternatively, you can navigate the document to focus on the security principles, services, account types, guidance, and examples that are most relevant to your needs.

This document is divided into five sections and an appendix:

- Security foundations (p. 3) reviews the AWS Cloud Adoption Framework (AWS CAF), the AWS Well-Architected Framework, and the AWS Shared Responsibility Model, and highlights elements that are especially relevant to the AWS SRA.
- AWS Organizations, accounts, and IAM guardrails (p. 5) introduces the AWS Organizations service, discusses the foundational security capabilities and guardrails, and gives an overview of our recommended multi-account strategy.
- The AWS Security Reference Architecture (p. 15) is a single-page architecture diagram that shows functional AWS accounts, and the security services and features that are generally available.
- IAM resources (p. 42) presents a summary and set of pointers for AWS Identity and Access Management (IAM) guidance that are important to your security architecture.
- Code repository for AWS SRA examples (p. 45) provides an overview of the associated public Github repo that contains example AWS CloudFormation templates and code for deploying some of the patterns discussed in the AWS SRA.

The appendix (p. 48) contains a list of the individual AWS security, identity, and compliance services, and provide links to more information about each service. The Document history (p. 50) section provides a change log for tracking versions of this document. You can also subscribe to an RSS feed for change notifications.

Security foundations

The AWS Security Reference Architecture aligns to three AWS security foundations: the AWS Cloud Adoption Framework (AWS CAF), AWS Well-Architected, and the AWS Shared Responsibility Model.

AWS Professional Services created AWS CAF to help companies design and follow an accelerated path to successful cloud adoption. The guidance and best practices provided by the framework help you build a comprehensive approach to cloud computing across your enterprise and throughout your IT lifecycle. The AWS CAF organizes guidance into six areas of focus, called *perspectives*. Each perspective covers distinct responsibilities owned or managed by functionally related stakeholders. In general, the business, people, and governance perspectives focus on business capabilities; whereas the platform, security, and operations perspectives focus on technical capabilities.

- The security perspective of the AWS CAF helps you structure the selection and implementation of controls across your business. Following the current AWS recommendations in the security pillar can help you meet your business and regulatory requirements.

AWS Well-Architected helps cloud architects build a secure, high-performing, resilient, and efficient infrastructure for their applications and workloads. The framework is based on five pillars—operational excellence, security, reliability, performance efficiency, and cost optimization—and provides a consistent approach for AWS customers and Partners to evaluate architectures and implement designs that can scale over time. We believe that having well-architected workloads greatly increases the likelihood of business success.

- The Well-Architected security pillar describes how to take advantage of cloud technologies to protect data, systems, and assets in a way that can improve your security posture. This will help you meet your business and regulatory requirements by following current AWS recommendations.

Security and compliance are a shared responsibility between AWS and the customer. This shared model can help relieve your operational burden as AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the service operates. For example, you assume responsibility and management of the guest operating system (including updates and security patches), application software, server-side data encryption, network traffic route tables, and the configuration of the AWS provided security group firewall. For abstracted services such as Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB, AWS operates the infrastructure layer, the operating system, and platforms, and you access the endpoints to store and retrieve data. You are responsible for managing your data (including encryption options), classifying your assets, and using AWS Identity and Access Management (IAM) tools to apply the appropriate permissions. This shared model is often described by saying that AWS is responsible for the security *of* the cloud (that is, for protecting the infrastructure that runs all the services offered in the AWS Cloud), and you are responsible for the security *in* the cloud (as determined by the AWS Cloud services that you select).

Within the guidance provided by these security foundations, two sets of concepts are particularly relevant to the design and understanding of the AWS SRA: security epics (also called security areas) and security design principles.

Security epics

Both the security perspective of the AWS CAF and the security pillar of Well-Architected outline five core security areas (called *epics* or *areas*, respectively) on which you can build your cloud security:

- *Identity and access management* forms the backbone of your AWS deployment. In the cloud you must establish an account and be granted privileges before you can provision or orchestrate resources.
- *Detection (logging and monitoring)* – AWS services provide a wealth of logging data to help you monitor your activity and changes within each service.
- *Infrastructure security* – When you treat infrastructure as code, security infrastructure becomes a first-tier workload that must also be deployed as code.
- *Data protection* – Safeguarding important data is a critical piece of building and operating information systems, and AWS provides services and features that give you robust options to help protect your data throughout its lifecycle.
- *Threat detection and incident response* – Automating aspects of your incident management process improves reliability, increases the speed of your response, and often creates an environment that is easier to assess in after-action reviews (AARs)

Security design principles

The security pillar of the Well-Architected Framework captures a set of design principles that turn the five security areas into practical guidance that can help you strengthen your workload security. Where the security epics frame the overall security strategy, these Well-Architected principles describe what you should start doing. They are reflected very deliberately in this AWS SRA and consist of the following:

- *Implement a strong identity foundation* – Implement the principle of least privilege, and enforce separation of duties with appropriate authorization for each interaction with your AWS resources. Centralize identity management, and aim to eliminate reliance on long-term static credentials.
- *Enable traceability* – Monitor, generate alerts, and audit actions and changes to your environment in real time. Integrate log and metric collection with systems to automatically investigate and take action.
- *Apply security at all layers* – Apply a defense-in-depth approach with multiple security controls. Apply multiple types of controls (for example, preventive and detective controls) to all layers, including edge of network, virtual private cloud (VPC), load balancing, every instance and compute service, operating system, application configuration, and code.
- *Automate security best practices* – Automated, software-based security mechanisms improve your ability to securely scale more rapidly and cost-effectively. Create secure architectures, and implement controls that are defined and managed as code in version-controlled templates.
- *Protect data in transit and at rest* – Classify your data into sensitivity levels and use mechanisms such as encryption, tokenization, and access control where appropriate.
- *Keep people away from data* – Use mechanisms and tools to reduce or eliminate the need to directly access or manually process data. This reduces the risk of mishandling or modification and human error when handling sensitive data.
- *Prepare for security events* – Prepare for an incident by having an incident management and investigation policy and processes that align to your business requirements. Run incident response simulations, and use automated tools to increase your speed for detection, investigation, and recovery.

AWS Organizations, accounts, and IAM guardrails

AWS security services, their controls, and interactions are employed on a foundation of AWS account strategy and identity and access management guardrails. These set the foundation for your implementation of least privilege, separation of duties, and privacy, and provide the support for decisions about what types of controls are needed, where each security service is managed, and how they may share data in the AWS SRA.

An AWS account provides security, access, and billing boundaries for your AWS resources and enables you to achieve resource independence and isolation. Use of multiple AWS accounts plays an important role in how you meet your security requirements, as discussed in the Benefits of using multiple AWS accounts section of the *Organizing Your AWS Environment Using Multiple Accounts* whitepaper. For example, you should organize your workloads in separate accounts and group accounts based on function, compliance requirements, or a common set of controls instead of mirroring your enterprise's reporting structure. Keep security and infrastructure in mind to enable your enterprise to set common guardrails as your workloads grow. This approach provides boundaries and controls between workloads. Account-level separation is used to isolate production environments from development and test environments, or to provide a strong logical boundary between workloads that process data of different classifications such as Payment Card Industry Data Security Standard (PCI DSS) or Health Insurance Portability and Accountability Act (HIPAA). Although you might begin your AWS journey with a single account, AWS recommends that you set up multiple accounts, as your workloads grow in size and complexity.

Permissions let you specify access to AWS resources. Permissions are granted to IAM entities known as *principals* (users, groups, and roles). By default, principals start with no permissions. IAM principals can do nothing in AWS until you grant them permissions, and you can set up guardrails that apply as broadly as your entire AWS organization or as fine-grained as an individual combination of principal, action, resource, and conditions.

In this section

Using AWS Organizations for security

AWS Organizations helps you centrally manage and govern your environment as you grow and scale your AWS resources. By using AWS Organizations, you can programmatically create new AWS accounts, allocate resources, group accounts to organize your workloads, and apply policies to accounts or groups of accounts for governance. An AWS organization consolidates your AWS accounts so that you can administer them as a single unit. It has one management account along with zero or more member accounts. Most of your workloads should reside in member accounts, except for some centrally managed processes that must reside in either the management account or in accounts assigned as delegated administrators for specific AWS services. You can provide tools and access from a central location for your security team to manage security needs on behalf of an AWS organization. You can reduce resource duplication by sharing critical resources within your AWS organization. You can group accounts into AWS organizational units (OUs), which can represent different environments based on the workload's requirements and purpose.

With AWS Organizations, you can use service control policies (SCPs) to apply permission guardrails at the AWS organization, OU, or account level. These guardrails apply to all users and roles within the covered accounts. When you attach an SCP to an OU, it flows down and affects all the branches (OUs) and leaves (accounts) beneath it. SCPs do not grant any permissions. Instead, SCPs specify the maximum permissions for an AWS organization, OU, or account. You still need to attach identity-based or resource-based policies to principals or resources in your AWS accounts to actually grant permissions to them. For more information about the types of IAM policies, see the IAM resources (p. 42) section. When you attach an SCP to your OU, the SCP limits permissions for principals in all associated member accounts. For example, you can apply an SCP that restricts users from launching resources in AWS Regions that you have not explicitly allowed.

AWS Control Tower offers a simplified way to set up and govern multiple accounts. It automates the setup of accounts in your AWS organization, automates provisioning, applies guardrails (which include preventive and detective controls), and provides you with a dashboard for visibility. An additional IAM management policy, a permissions boundary, is attached to specific IAM principals (users or roles) and sets the maximum permissions that an identity-based policy can grant to an IAM principal.

AWS Organizations helps you configure AWS services that apply to all your accounts. For example, you can configure central logging of all actions performed across your AWS organization by using AWS CloudTrail, and prevent member accounts from disabling logging. You can also centrally aggregate data for rules that you've defined by using AWS Config, so you can audit your workloads for compliance and react quickly to changes. You can use AWS CloudFormation StackSets to centrally manage AWS CloudFormation stacks across accounts and OUs in your AWS organization, so you can automatically provision a new account to meet your security requirements.

The default configuration of AWS Organizations supports using SCPs as *deny lists*, and that is the approach we recommend. By using a deny list strategy, member account administrators can delegate all services and actions until you create and attach an SCP that denies a specific service or set of actions. Deny statements require less maintenance than an allow list, because you don't have to update them when AWS adds new services. Deny statements are usually shorter in character length, so it's easier to stay within the maximum size for SCPs. In a statement where the `Effect` element has a value of `Deny`, you can also restrict access to specific resources, or define conditions for when SCPs are in effect. By contrast, an `Allow` statement in an SCP applies to all resources (`"*"`) and cannot be restricted by conditions. For more information and examples, see Strategies for using SCPs in the AWS Organizations documentation.

Design consideration

Alternatively, to use SCPs as an *allow list*, you must replace the AWS managed `FullAWSAccess` SCP with an SCP that explicitly permits only those services and actions that you want to allow. For a permission to be enabled for a specified account, every SCP (from the root through each OU in the direct path to the account, and even attached to the account itself) must allow that permission. This model is more restrictive in nature and can be a good fit for highly regulated and sensitive workloads. This approach requires maintenance, because you must explicitly allow every IAM action in the path from the AWS account to the OU, and some of the actions might need to be implemented by a separate team, such as central security or identity and access management.

The management account, trusted access, and delegated administrators

The management account (also called the AWS Organization Management account or Org Management account) is unique. It is the account that creates the AWS organization. From this account, you can create AWS accounts in the AWS organization, invite other existing accounts to the AWS organization (both types are considered *member accounts*), remove accounts from the AWS organization, and apply IAM policies to the root, OUs, or accounts within the AWS organization. The management account can deploy the universal security guardrails through SCPs and service deployments (such as AWS CloudTrail)

that will affect all member accounts in the AWS organization. To further restrict permissions in the management account, those permissions should be delegated to another appropriate account, such as a security account, where possible. The management account has the responsibilities of a payer account and is responsible for paying all charges that are accrued by the member accounts. You cannot switch an AWS organization's management account. An AWS account can be a member of only one AWS organization at a time.

Because of the functionality and scope of influence the management account holds, we recommend that you limit access to this account and grant permissions only to roles that need them. Two features that help you do this are trusted access and delegated administrator. You can use trusted access to enable an AWS service that you specify, called the *trusted service*, to perform tasks in your AWS organization and its accounts on your behalf. This involves granting permissions to the trusted service but does not otherwise affect the permissions for IAM users or roles. You can use trusted access to specify settings and configuration details that you would like the trusted service to maintain in your AWS organization's accounts on your behalf. For example, the Org Management account (p. 17) section of the AWS SRA explains how to grant the AWS CloudTrail service trusted access to create a CloudTrail "organization trail" in all accounts in your AWS organization.

Some AWS services support the delegated administrator feature in AWS Organizations. With this feature, compatible services can register an AWS member account in the AWS organization as an administrator for the AWS organization's accounts in that service. This capability provides flexibility for different teams within your enterprise to use separate accounts, as appropriate for their responsibilities, to manage AWS services across the environment. The AWS security services in the AWS SRA that currently support delegated administrator include AWS Config, AWS Firewall Manager, Amazon GuardDuty, AWS IAM Access Analyzer, Amazon Macie, AWS Security Hub, and AWS Systems Manager. Use of the delegated administrator feature is emphasized in the AWS SRA as a best practice, and we delegate administration of security-related services to the Security Tooling account.

Dedicated accounts structure

An AWS account provides security, access, and billing boundaries for your AWS resources, and enables you to achieve resource independence and isolation. By default, no access is allowed between accounts.

When designing your OU and account structure, start with security and infrastructure in mind. Most enterprises have centralized teams that serve the security and infrastructure needs of the entire business. We recommend creating a set of foundational OUs for these specific functions, split into Infrastructure and Security OUs. These OU and account recommendations capture a subset of our broader, more comprehensive guidelines for AWS Organizations and multi-account structure design. For a full set of recommendations, see Organizing Your AWS Environment Using Multiple Accounts in the AWS documentation and the blog post Best Practices for Organizational Units with AWS Organizations.

The AWS SRA utilizes the following accounts to achieve effective security operations on AWS. These dedicated accounts help ensure separation of duties, support different governance and access policies for different sensitives of applications and data, and help mitigate the impact of a security event. In the discussions that follow, we are focused on production (*prod*) accounts and their associated workloads. Software development lifecycle (SDLC) accounts (often called *dev* and *test* accounts) are intended for staging deliverables and can operate under a different security policy set from that of production accounts.

Account	OU	Security role
Management	—	Central governance and management of all AWS Regions and accounts. The AWS account that hosts the root of the AWS organization.

Account	OU	Security role
Security Tooling	Security	Dedicated AWS accounts for operating broadly applicable security services (such as Amazon GuardDuty, AWS Security Hub, and AWS Config), monitoring AWS accounts, and automating security alerting and response.
Log Archive	Security	Dedicated AWS accounts for ingesting and archiving all logging and backups for all AWS Regions and AWS accounts. This should be designed as immutable storage.
Network	Infrastructure	The gateway between your application and the broader internet. The Network account isolates the broader networking services, configuration, and operation from the individual application workloads, security, and other infrastructure.
Shared Services	Infrastructure	This account supports the services that multiple applications and teams use to deliver their outcomes. Examples include directory services (Active Directory), messaging services, and metadata services.
Application	Workloads	AWS accounts that host the AWS organization's applications and perform the workloads. (These are sometimes called *workload accounts*.) Application accounts should be created to isolate software services instead of being mapped to your teams. This makes the deployed application more resilient to organizational change.

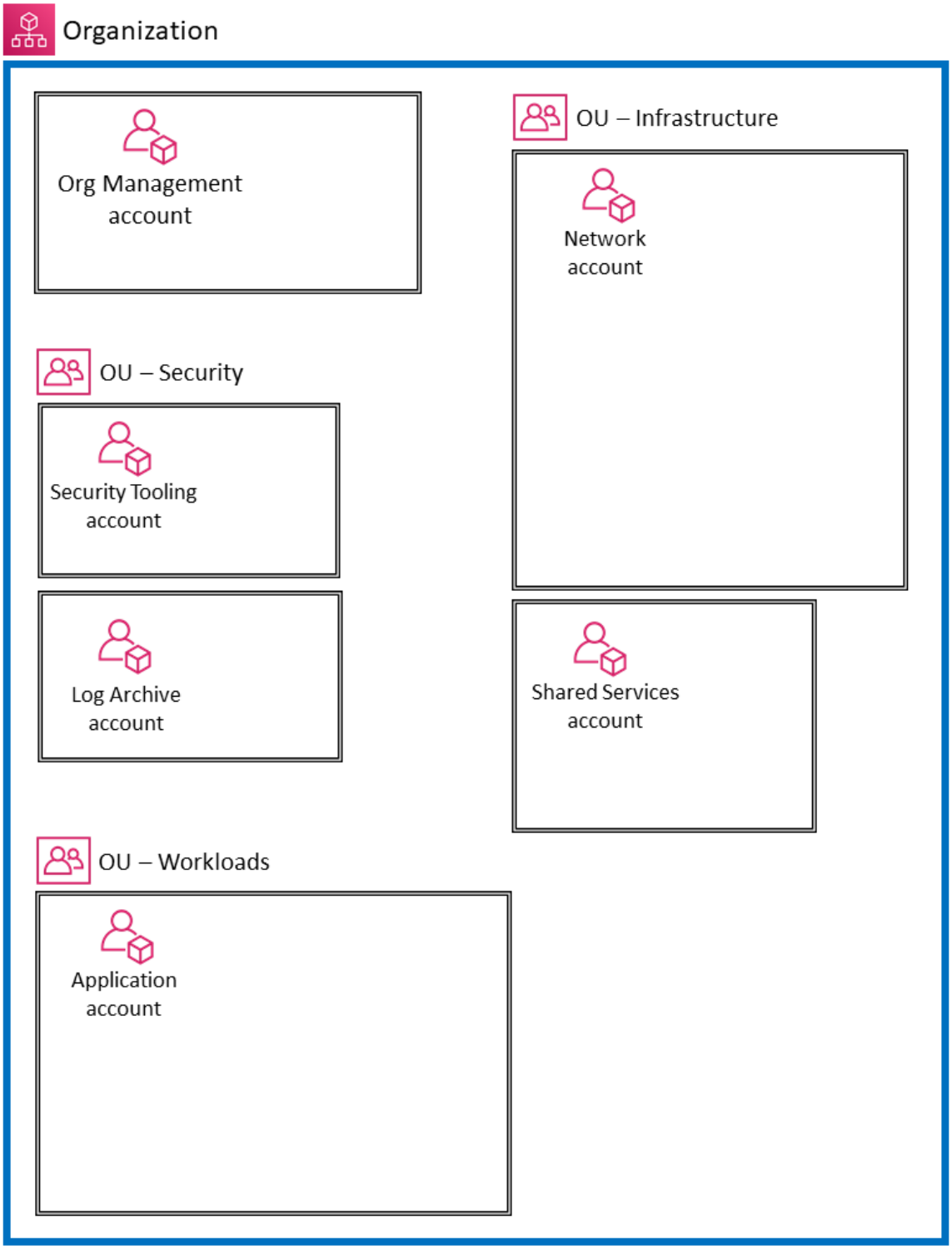
AWS organization and account structure of the AWS SRA

The following diagram captures the high-level structure of the AWS SRA without displaying specific services. It reflects the dedicated accounts structure discussed in the previous section, and we include the diagram here to orient the discussion around the primary components of the architecture:

- All accounts that are shown in the diagram are part of a single AWS organization.

- At the upper left of the diagram is the Org Management account, which is used to create the AWS organization.
- Below the Org Management account is the Security OU with two specific accounts: one for Security Tooling and the other for Log Archive.
- Along the right side is the Infrastructure OU with the Network account and Shared Services account.
- At the bottom of the diagram is the Workloads OU, which is associated with an Application account that houses the enterprise application.

For this discussion, all accounts should be considered production (prod) accounts that operate in a single AWS Region. When a regional service such as Amazon Simple Storage Service (Amazon S3), Amazon GuardDuty, or AWS Key Management Service (AWS KMS) is shown inside an account, that service is configured and managed from within that account.



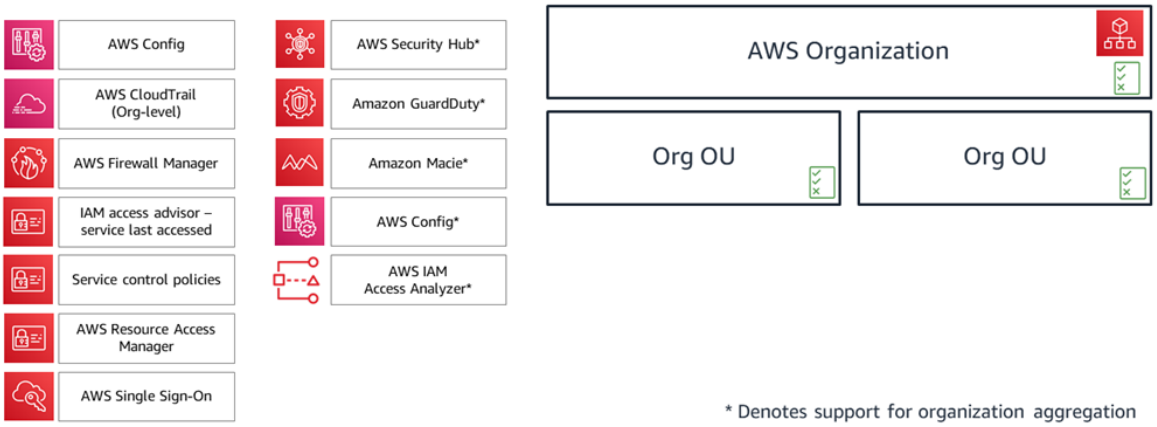
Apply security services across your AWS organization

The security categories we discussed earlier in the Security foundations (p. 3) section are organized functionally, to represent areas of focus for your cloud security strategy. Another way to group services is by their intended scope of control. This perspective focuses on where to configure and manage AWS security services to deploy appropriate layers of defense in the AWS Organizations hierarchy. For example, some services and features are best used to implement controls for your AWS organization. Others are best used to protect individual resources within an AWS account. Understanding the scope of influence of each service helps you adopt a defense-in-depth strategy. Thinking about services in this way helps ensure that your layers of security appropriately complement one another. With this perspective, you can answer questions both from the top down (for example, "Which services apply security controls across my entire AWS organization?") and from the bottom up (for example, "Which services apply controls to this particular resource?"). In this section, we walk through the elements of an AWS environment—organization, OU, account, network, principal, resource—and identify the associated security services and features. Further discussion of the individual services within each AWS account follows in the next section.

Organization-wide or multiple accounts

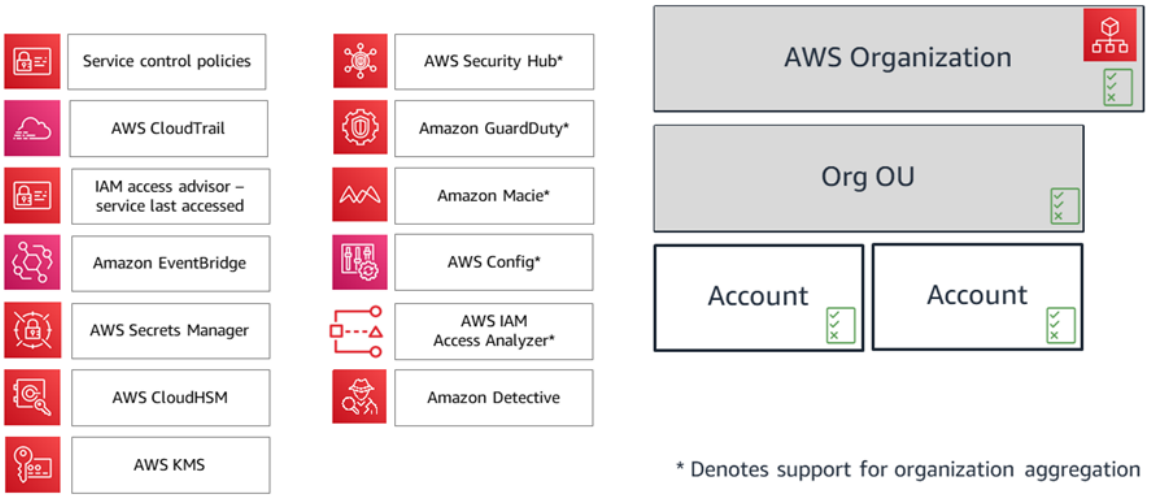
At the top level, there are AWS services and features that are designed to apply governance and control capabilities or guardrails across multiple accounts in an AWS organization (including the entire organization or specific OUs). Service control policies (SCPs) are a good example of an IAM feature that is designed as an AWS organization-wide guardrail. Another example is AWS CloudTrail, which supports an *organization trail* that will log all events for all AWS accounts in that AWS organization. This comprehensive trail is distinct from individual trails that might be created in each account. A third example is AWS Firewall Manager, which you can use to configure and apply AWS WAF rules, AWS WAF Classic rules, AWS Shield Advanced protections, Amazon Virtual Private Cloud (Amazon VPC) security groups, AWS Network Firewall policies, and Amazon Route 53 resolver DNS firewall policies across your AWS organization.

Some security services (marked with an asterisk * in the following diagram) operate with a dual scope: organization-wide and account-focused. These services fundamentally monitor or control security within an individual account. However, some configuration, and often the results from multiple accounts, can be aggregated to an organization-wide account for centralized visibility and management. For example, an SCP applies across an entire OU or AWS organization by default. In contrast, Amazon GuardDuty can be configured and managed both at the account level (where individual findings are generated) and at the AWS organization level (via the delegated administrator feature) where findings can be managed in aggregate.



AWS accounts

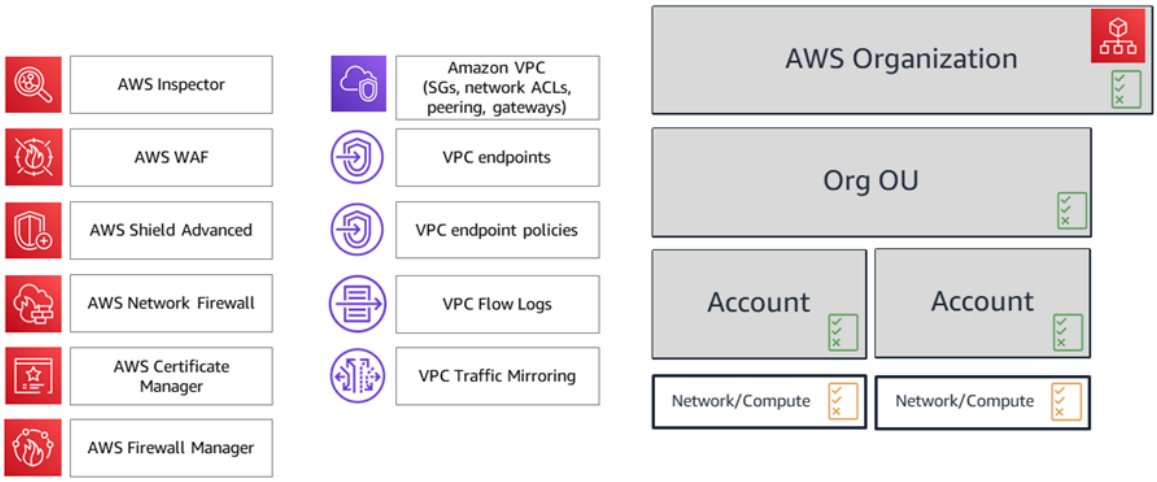
Within OUs, there are services that protect multiple types of elements within an AWS account. The following diagram illustrates these services. For example, AWS Secrets Manager is often managed from, and protects resources for, a single account. Amazon GuardDuty monitors resources and activity associated with a single account. As mentioned in the previous section, some of these services can also be configured and administered within AWS Organizations, so they can be managed across multiple accounts (which do not all have to be in the same AWS organization). These services (denoted with an asterisk *) also make it easier to aggregate results from multiple accounts and deliver those to a single account. This gives individual application teams the flexibility and visibility to manage specific security needs while also allowing governance and visibility to centralized security teams. Amazon GuardDuty is a good example of such a service. GuardDuty findings from multiple member accounts (such as all accounts in an AWS organization) can be collected, viewed, and managed from a delegated administrator account.



Virtual network and compute infrastructure

Because network access is so critical in security, and compute infrastructure is a fundamental component of many AWS workloads, there are many AWS security services and features that are dedicated to these resources. For example, Amazon Inspector helps check for unintended network accessibility of your

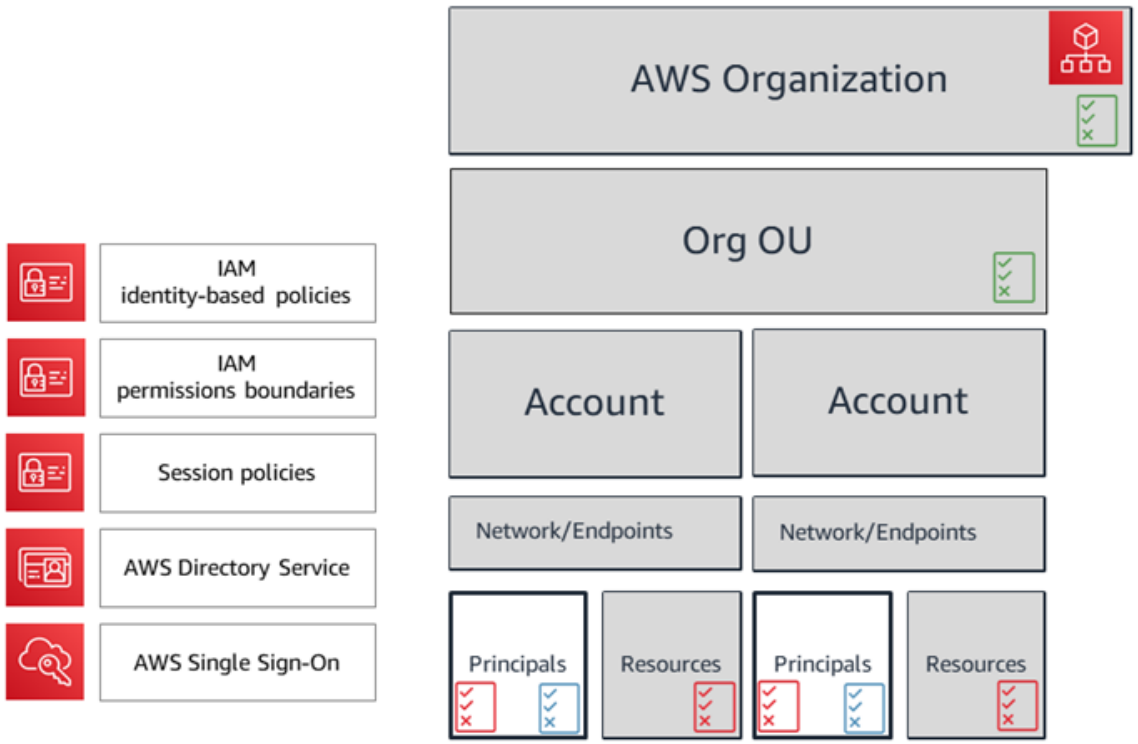
Amazon Elastic Compute Cloud (Amazon EC2) instances and for vulnerabilities on those EC2 instances. VPC endpoints enable you to privately connect your VPC to supported AWS services and VPC endpoint services powered by AWS PrivateLink without requiring additional network access services such as internet gateways. The following diagram illustrates security services that focus on network or compute infrastructure.



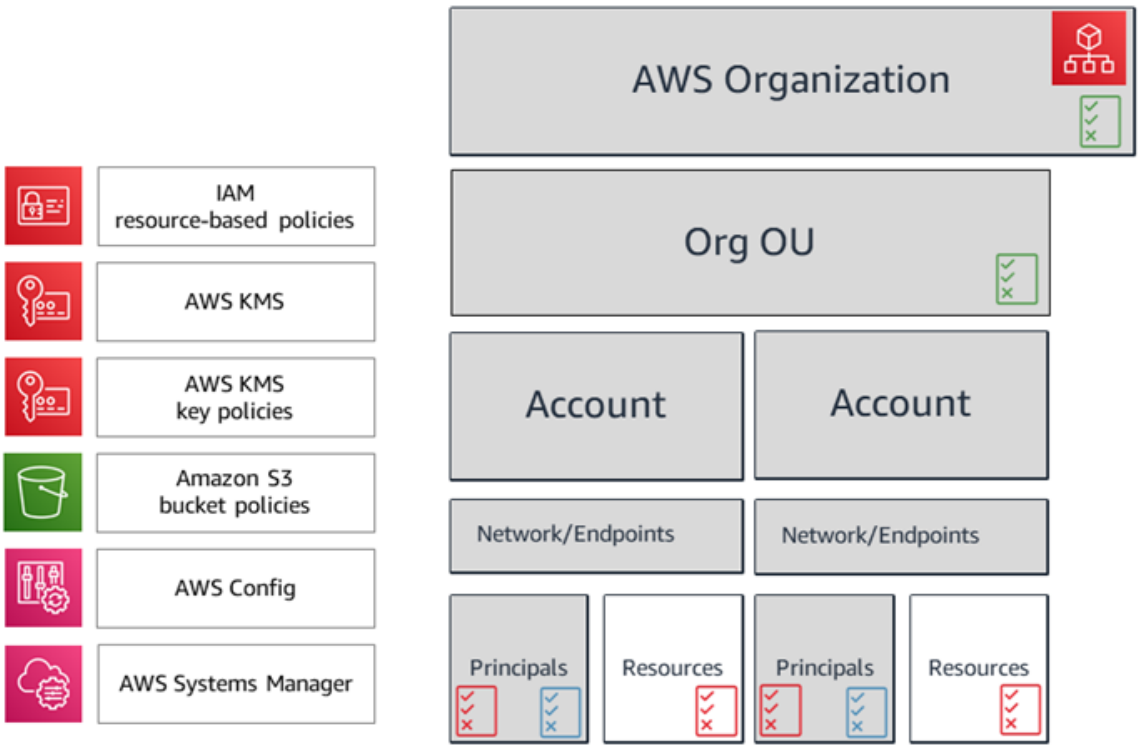
Principals and resources

IAM principals and resources are the fundamental building blocks (along with policies) of identity and access management in AWS. An IAM principal is an entity in AWS that can perform actions and access resources. A principal can be an AWS account root user, an IAM user, or a role. A resource is an object that exists within an AWS service. Examples include an EC2 instance, an Amazon Simple Notification Service (Amazon SNS) topic, and an S3 bucket. You can associate permissions with a principal to grant or restrict the principal's actions and their access to resources. You can also associate permissions with a resource to grant or restrict which principals can access or act on that resource. IAM identity-based (or resource-based) policies are typically used for these respective permission controls. The IAM resources (p. 42) section dives deeper into the types of IAM policies and how they are used.

The following diagram illustrates AWS security services and features for account principals.

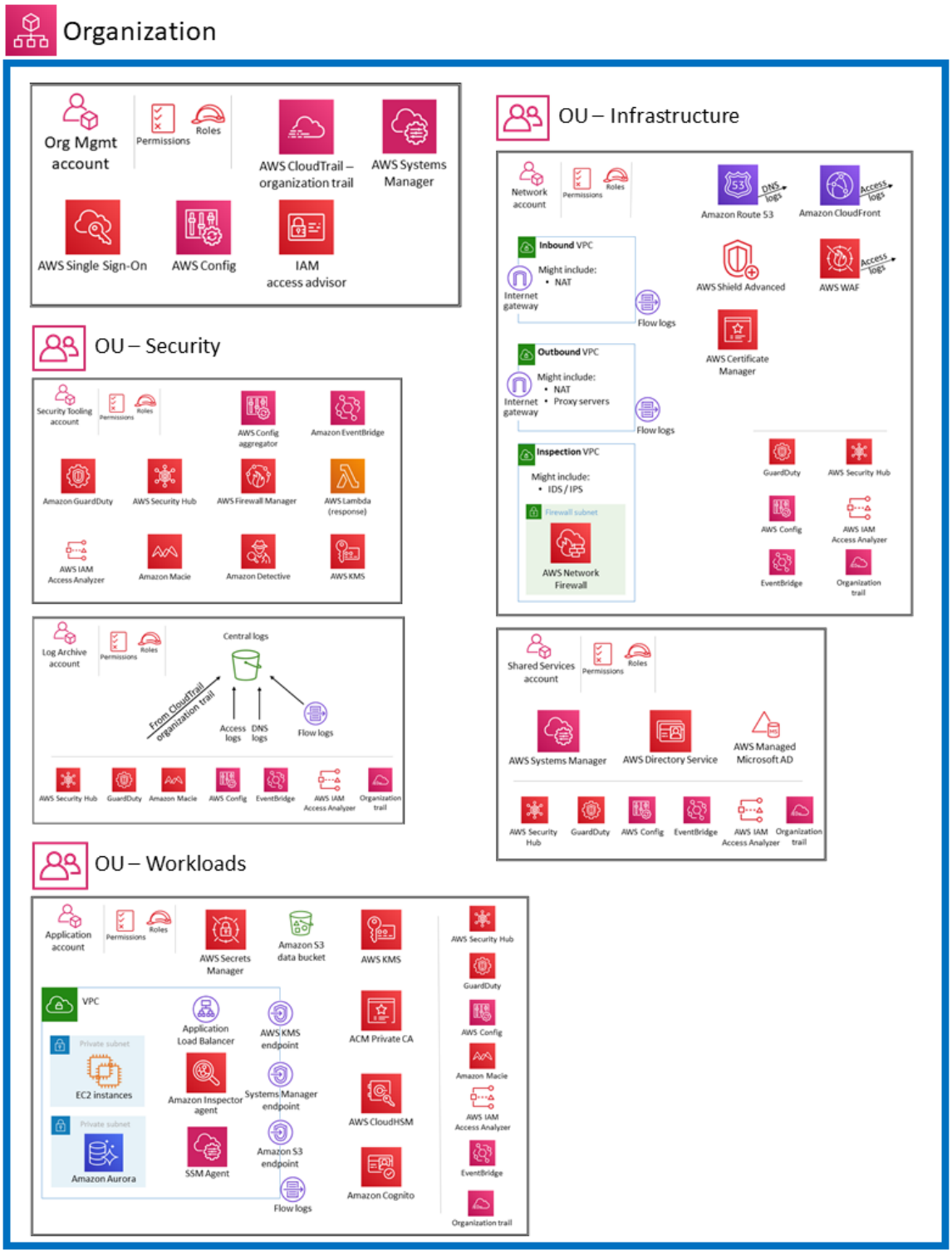


The following diagram illustrates services and features for account resources.



The AWS Security Reference Architecture

The following diagram illustrates the AWS SRA. This architectural diagram brings together all the AWS security-related services. It is built around a simple, three-tier web architecture that can fit on a single page. In such a workload, there is a *web tier* through which users connect and interact with the *application tier*, which handles the actual business logic of the application: taking inputs from the user, doing some computation, and generating outputs. The application tier stores and retrieves information from the *data tier*. The design is purposefully modular and provides the high-level abstraction for many modern web applications.



For this reference architecture, the actual web application and data tier are deliberately represented as simply as possible, through Amazon Elastic Compute Cloud (Amazon EC2) instances and an Amazon Aurora database, respectively. Most architecture diagrams focus and dive deep on the web, application, and data tiers. For readability, they often omit the security controls. This diagram flips that emphasis to

show security wherever possible, and keeps the application and data tiers as simple as necessary to show security features meaningfully.

The AWS SRA contains most of the AWS security-related services that were generally available at publication. (See Document history (p. 50).) Not every workload or environment needs to deploy every security service, but our goal is to provide a reference for all possible options, including descriptions of how these services fit together architecturally.

The following sections walk through each OU and account to understand its objectives and the individual AWS security services associated with it. For each element (typically an AWS service), this document provides the following information:

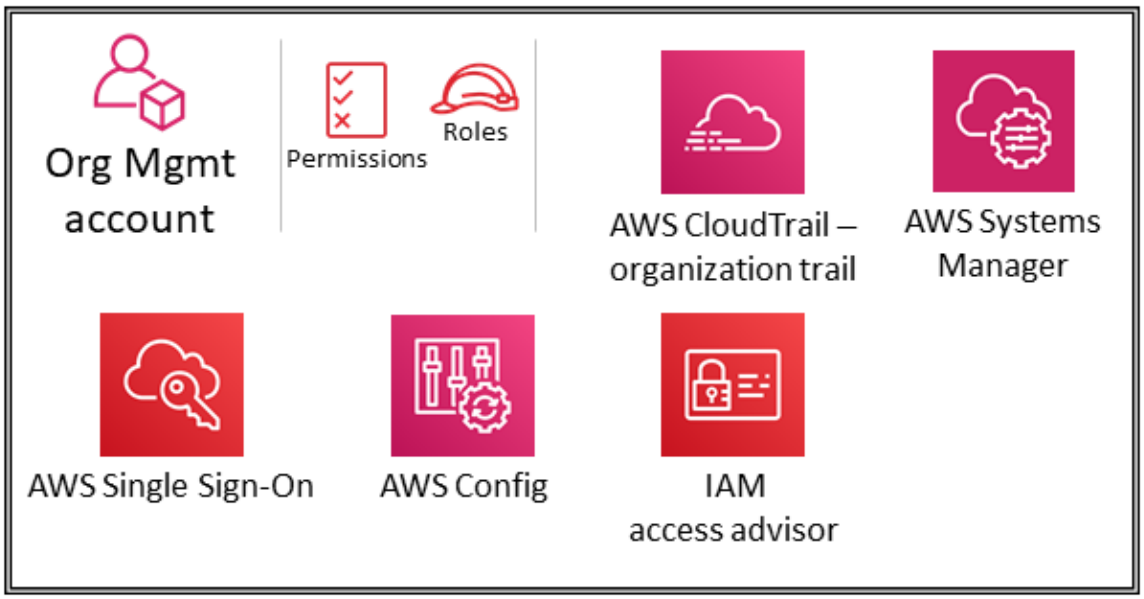
- Brief overview of the element and its security purpose in the AWS SRA. For more detailed descriptions and technical information about individual services, see the Appendix (p. 48).
- Recommended placement to most effectively enable and manage the service. This is captured in the individual architecture diagrams for each account and OU.
- Configuration, management, and data sharing links to other security services. How does this service rely on, or support, other security services?
- Design considerations. First, the document highlights optional features or configurations that have important security implications. Second, where our teams' experience includes common variations in the recommendations we make—typically as a result of alternate requirements or constraints—the document describes those options.

OUs and accounts

- Org Management account (p. 17)
- Security OU – Security Tooling account (p. 20)
- Security OU – Log Archive account (p. 26)
- Infrastructure OU – Network account (p. 28)
- Infrastructure OU – Shared Services account (p. 33)
- Workloads OU – Application account (p. 35)

Org Management account

The following diagram illustrates the AWS security services that are configured in the Org Management account.



The sections Using AWS Organizations for security (p. 5) and The management account, trusted access, and delegated administrators (p. 6) earlier in this guide discussed the purpose and security objectives of the Org Management account in depth. You should follow the security best practices for your Org Management account. These include using an email address that is managed by your business, maintaining the correct administrative and security contact information (such as attaching a phone number to the account (in the event AWS needs to contact the owner of the account), enabling multi-factor authentication (MFA) for the root user, and regularly reviewing who has access to the Org Management account. Services deployed in the Org Management account should be configured with appropriate roles, trust policies, and other permissions so that the administrators of those services (who must access them in the Org Management account) cannot also inappropriately access other services.

Service control policies

Apply service control policies (SCPs) in the Org Management account to ensure that member AWS accounts stay within your account governance strategy and access control guidelines. SCPs do not grant any permissions. Instead, SCPs deployed in the Org Management account specify the maximum permissions for this AWS organization. Additional SCPs are deployed for each OU to establish more specific guardrails for each type of account. Read more about SCPs in the Using AWS Organizations for security (p. 5) section earlier in this reference.

> **Design consideration**
> SCPs affect only member accounts in the AWS organization. They have no effect on users or roles in the Org Management account.

AWS CloudTrail

AWS CloudTrail is a service that supports governance, compliance, operational auditing, and risk auditing of your AWS account. With CloudTrail, you can log, continuously monitor, and retain account activity related to actions across your AWS infrastructure. CloudTrail is integrated with AWS Organizations, and that integration can be used to create a single trail that logs all events for all accounts in the AWS organization. This is referred to as an *organization trail*. When you create an organization trail, a trail with the name that you specify is created in every AWS account that belongs to your AWS organization. The trail logs activity for all accounts in the AWS organization and stores the logs in a single S3 bucket. All accounts in the AWS organization can see the organization trail in their list of trails, but member AWS

accounts have limited access to this trail. Additionally, by default, only the Org Management account has access to the S3 bucket. For more information about these protections, see the Amazon S3 as central log store (p. 27) section. For additional security best practices, see the AWS CloudTrail documentation.

Design consideration
If member accounts need to use CloudTrail information in a way that isn't permitted by the organization trail, the managers of each AWS account can create a local trail with the appropriate controls.

AWS SSO

AWS Single Sign-On (AWS SSO) serves as your identity source and enables federation to multiple accounts in your AWS organization. You should rely on an identity provider that lets you manage identities in a centralized place. This makes it easier to manage access across multiple applications and services, because you are creating, managing, and revoking access from a single location. For example, if someone leaves your team, you can revoke their access to all applications and services (including AWS accounts) from one location. This reduces the need for multiple credentials and provides an opportunity to integrate with your human resources (HR) processes.

You can use AWS SSO to quickly and easily assign your employees' access to AWS accounts that are managed with AWS Organizations, business cloud applications, and custom applications that support Security Assertion Markup Language (SAML) 2.0. AWS SSO natively integrates with AWS Organizations and is enabled in the Org Management account. Accounts are displayed by OU within the AWS SSO console. This enables you to quickly discover your AWS accounts, deploy common sets of permissions, and manage access from a central location.

Design considerations

- Administrators can use the default AWS SSO directory to manage their users. Or, they can connect their self-managed Active Directory (AD) or their AWS Managed Microsoft AD directory by using AWS Directory Service (in the Shared Services account). This Microsoft AD directory defines the pool of identities that administrators can pull from when they use the AWS SSO console to assign SSO access. AWS Directory Service helps you set up and run a standalone AWS Managed Microsoft AD directory hosted in the AWS Cloud. You can also use AWS Directory Service to connect your AWS resources with an existing self-managed AD.
- AWS SSO is one option for implementing an SSO authentication strategy. Many enterprise customers integrate SSO with their existing identity provider (IdP). If you're using another IdP with SSO, we recommend using the System for Cross-domain Identity Management (SCIM) for better security, consistency, and convenience.
- Enforce multi-factor authentication (MFA) with software or hardware mechanisms to provide an additional layer of verification. For example, when using AWS SSO as the identity source, configure the **context-aware** or **always-on** setting for MFA, and allow users to enroll their own MFA devices to accelerate adoption.

IAM access advisor

IAM access advisor provides traceability data in the form of service last accessed information for your AWS accounts and OUs. Use this detective control to contribute to a least privilege strategy. For IAM principals, you can view two types of last accessed information: allowed AWS service information and allowed action information. The information includes the date and time when the attempt was made.

From the Org Management account, you can also view service last accessed data for the Org Management account, OU, member account, or IAM policy in your AWS organization. A programmatic report for an AWS organizational entity includes a list of services that are allowed by any SCPs that apply to the entity. Last accessed information provides insight for actual service usage (see example scenarios), so you can reduce IAM permissions to only those services that are actually used.

Design consideration
Service last accessed information for an AWS Organizations entity or policy can be accessed only from the Org Management account. For this reason, we recommend that you follow both a least privilege and separation of duties approach when you set the permissions for the identities who will access this information.

AWS Systems Manager

AWS Systems Manager Quick Setup and Systems Manager Explorer both support AWS Organizations and operate from the Org Management account.

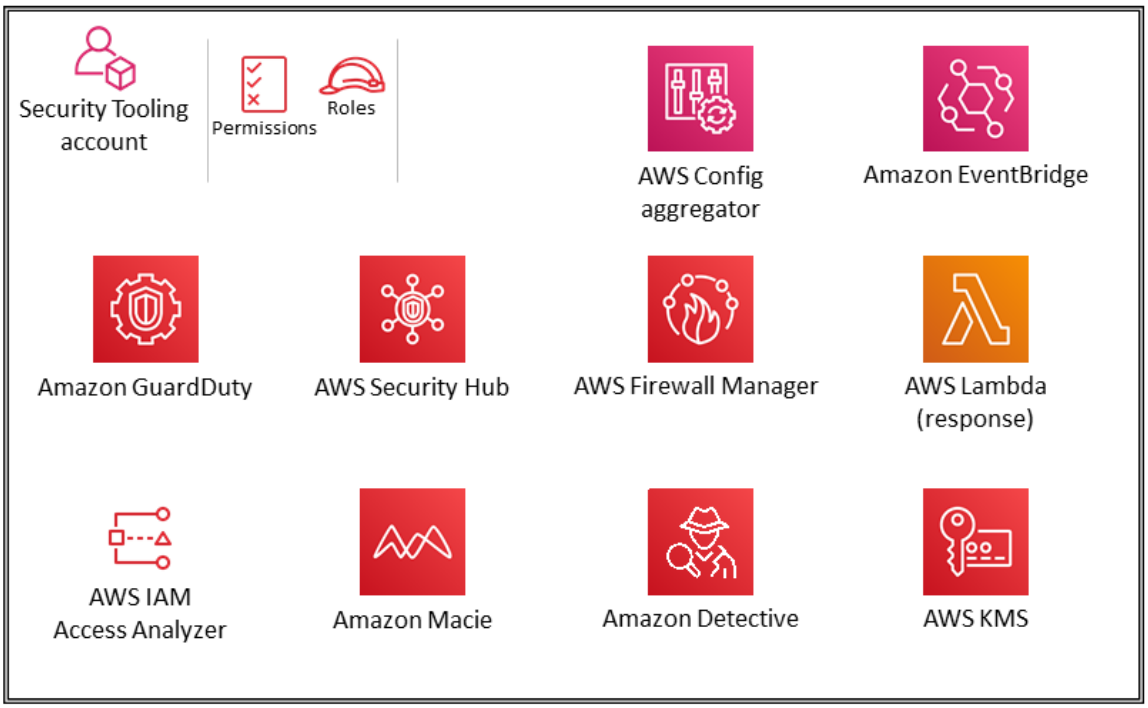
Quick Setup is an automation feature of Systems Manager. It enables the Org Management account to easily define configurations for Systems Manager to engage on your behalf across accounts in your AWS organization. You can enable Quick Setup across your entire AWS organization or choose specific OUs. Among other things, Quick Setup can schedule AWS Systems Manager Agent (SSM Agent) to run biweekly updates on your EC2 instances and can set up a daily scan of those instances to identify missing patches.

Systems Manager Explorer is a customizable operations dashboard that reports information about your AWS resources. Explorer displays an aggregated view of operations data for your AWS accounts and across AWS Regions. This includes data about your EC2 instances and patch compliance details. After you complete Integrated Setup (which also includes Systems Manager OpsCenter) within AWS Organizations, you can aggregate data in Explorer by OU or for an entire AWS organization. Systems Manager aggregates the data into the AWS Org Management account before displaying it in Explorer.

The Workloads OU (p. 35) section later in this guide discusses the use of the Systems Manager Agent (SSM Agent) on the EC2 instances in the Application account.

Security OU – Security Tooling account

The following diagram illustrates the AWS security services that are configured in the Security Tooling account.



The Security Tooling account is dedicated to operating security services, monitoring AWS accounts, and automating security alerting and response. The security objectives include the following:

- Provide a dedicated enclave with controlled access to manage access to the security guardrails, monitoring, and response.
- Maintain the appropriate centralized security infrastructure to monitor security operations data and maintain traceability. Detection, investigation, and response are essential parts of the security lifecycle and can be used to support a quality process, a legal or compliance obligation, and for threat identification and response efforts.
- Further support a defense-in-depth strategy by maintaining another layer of control over appropriate security configuration and operations such as encryption keys and security group settings. This is an account where security operators work. Read-only/audit roles to view AWS organization-wide information are typical, whereas write/modify roles are limited in number, tightly controlled, monitored, and logged.

Design consideration
It might be appropriate to have more than one Security Tooling account. For example, monitoring and responding to security events is often assigned to a dedicated team. Network security might warrant its own account and roles in collaboration with the cloud infrastructure or network team. Such splits retain the objective of separating centralized security enclaves and further emphasize the separation of duties, least privilege, and potential simplicity of team assignments.

Delegated administrator for security services

The Security Tooling account serves as the administrator account for security services that are managed in an administrator/member structure throughout the AWS accounts. As mentioned earlier, this is handled through the AWS Organizations delegated administrator functionality. Services in the AWS SRA that currently support delegated administrator include AWS Config, AWS Firewall Manager, Amazon GuardDuty, AWS IAM Access Analyzer, Amazon Macie, AWS Security Hub, and AWS Systems Manager. The

security team manages the security features of these services and monitors any security-specific events or findings.

AWS Security Hub

AWS Security Hub provides you with a comprehensive view of your security state in AWS and helps you check your environment against security industry standards and best practices. Security Hub collects security data from across AWS integrated services, supported third-party products, and other custom security products you may use. It helps you continuously monitor and analyze your security trends and identify the highest priority security issues.

Security Hub integrates with AWS Organizations to simplify security posture management across all your existing and future accounts in your AWS organization. The Security Hub delegated administrator account (in this case, Security Tooling) has Security Hub enabled automatically and can choose the AWS accounts to enable as member accounts. The Security Hub delegated administrator account can also view findings, view insights, and control details from all member accounts.

Security Hub supports integrations with several AWS services. Amazon GuardDuty, AWS Config, Amazon Macie, AWS IAM Access Analyzer, AWS Firewall Manager, Amazon Inspector, and AWS Systems Manager Patch Manager can feed findings to Security Hub. In addition, you can pivot from Security Hub to Amazon Detective to investigate an Amazon GuardDuty finding. Security Hub recommends aligning the delegated administrator accounts for these services (where they exist) for smoother integration. For example, if you do not align administrator accounts between Detective and Security Hub, pivoting from findings into Detective will not work.

In addition to monitoring, Security Hub supports integration with Amazon EventBridge to automate remediation of specific findings. You can define custom actions to take when a finding is received. For example, you can configure custom actions to send findings to a ticketing system or to an automated remediation system. Further discussion and examples are available in these two AWS blog posts: Automated Response and Remediation with AWS Security Hub and How to deploy the AWS Solution for Security Hub Automated Response and Remediation.

Security Hub uses service-linked AWS Config rules to perform most of its security checks for controls. To support these controls, AWS Config must be enabled on all accounts—including the administrator (or delegated administrator) account and member accounts—in each AWS Region where Security Hub is enabled.

Design considerations

- In addition to the specific, managed AWS Config rules that Security Hub uses, you can use automation to import other AWS Config rules to Security Hub so that your AWS Config rules show up along with your other security findings. This allows you to more easily use AWS Config rules to help ensure continuous compliance across all your AWS accounts. For more information, see the blog post How to import AWS Config rules evaluations as findings in Security Hub.
- If a compliance standard, such as PCI-DSS, is already present in Security Hub, then the fully managed Security Hub service is the easiest way to operationalize it. However, if you want to assemble your own compliance or security standard, which might include security, operational, or cost optimization checks, AWS Config conformance packs offer a simplified way to do this customization. (For more information about AWS Config and conformance packs, see the AWS Config (p. 23) section.)

Amazon GuardDuty

Amazon GuardDuty is a threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts and workloads. You should always capture and

store appropriate logs for monitoring and audit purposes, but GuardDuty pulls independent streams of data directly from AWS CloudTrail, VPC flow logs, and AWS DNS logs. You don't have to manage Amazon S3 bucket policies or modify the way you collect and store your logs. GuardDuty permissions are managed as service-linked roles that you can revoke at any time by disabling GuardDuty. This makes it easy to enable the service without complex configuration, and it eliminates the risk that an IAM permission modification or S3 bucket policy change will affect the operation of the service.

GuardDuty is enabled in all accounts through AWS Organizations, and all findings are viewable and actionable by appropriate security teams in the GuardDuty delegated administrator account (in this case, the Security Tooling account).

When AWS Security Hub is enabled, GuardDuty findings automatically flow to Security Hub. When Amazon Detective is enabled, GuardDuty findings are included in the Detective log ingest process. GuardDuty and Detective support cross-service user workflows, where GuardDuty provides links from the console that redirect you from a selected finding to a Detective page that contains a curated set of visualizations for investigating that finding. You can also integrate GuardDuty with Amazon EventBridge to automate best practices for GuardDuty, such as automating responses to new GuardDuty findings.

AWS Config

AWS Config is a service that enables you to assess, audit, and evaluate the configurations of supported AWS resources in your AWS accounts. AWS Config continuously monitors and records AWS resource configurations, and automatically evaluates recorded configurations against desired configurations. You can also integrate AWS Config with other services to do the heavy lifting in automated audit and monitoring pipelines. For example, AWS Config can monitor for changes in individual secrets in AWS Secrets Manager.

AWS Config must be enabled for each member account in the AWS organization and for each AWS Region that contains the resources that you want to protect. You can centrally manage (for example, create, update, and delete) AWS Config rules across all accounts within your AWS organization. From the AWS Config delegated administrator account, you can deploy a common set of AWS Config rules across all accounts and specify accounts where AWS Config rules should not be created. The AWS Config delegated administrator account can also aggregate resource configuration and compliance data from all member accounts to provide a single view. Use the APIs from the delegated administrator account to enforce governance by ensuring that the underlying AWS Config rules are not modifiable by your AWS organization's member accounts.

Design considerations

- AWS Config streams all configuration and compliance change notifications to Amazon EventBridge. This means that you can use the native filtering capabilities in EventBridge to filter AWS Config events so that you can route specific types of notifications to specific targets. For example, you can send compliance notifications for specific rules or resource types to specific email addresses, or route configuration change notifications to an external IT service management (ITSM) or configuration management database (CMDB) tool. For more information, see the blog post AWS Config best practices.
- A conformance pack is a collection of AWS Config rules and remediation actions that can be easily deployed as a single entity in an account and Region, or across an organization in AWS Organizations. Conformance packs are created by authoring a YAML template that contains the list of AWS Config managed or custom rules and remediation actions. To get started evaluating your AWS environment, use one of the sample conformance pack templates.

Amazon Macie

Amazon Macie is a fully managed data security and data privacy service that uses machine learning and pattern matching to discover and protect your sensitive data in AWS. You need to understand

the type and classification of data your workload is processing to ensure that appropriate controls are enforced. Macie automates the discovery of sensitive data at scale. With Macie, you can perform various sensitive content discovery and data classification tasks on objects in Amazon S3. Macie is enabled in all accounts through AWS Organizations. Principals who have the appropriate permissions in the delegated administrator account (in this case, the Security Tooling account) can enable or suspend Macie in any account, create sensitive data discovery jobs for buckets that are owned by member accounts, and view all policy findings for all member accounts. Sensitive data findings can be viewed only by the account that created the sensitive findings job. For more information, see Managing multiple accounts in Amazon Macie in the Macie documentation.

Macie findings flow to AWS Security Hub for review and analysis. Macie also integrates with Amazon EventBridge to facilitate automated responses to findings such as alerts, feeds to security information and event management (SIEM) systems, and automated remediation.

Design considerations

- If S3 objects are encrypted with an AWS Key Management Service (AWS KMS) customer master key (CMK) that you manage, you can add the Macie service-linked role as a key user to that CMK to enable Macie to scan the data.
- Macie is optimized for scanning objects in Amazon S3. As a result, any Macie-supported object type that can be placed in Amazon S3 (permanently or temporarily) can be scanned for sensitive data. This means that data from other sources—for example, periodic snapshot exports of Amazon Relational Database Service (Amazon RDS) or Amazon Aurora databases, exported Amazon DynamoDB tables, or extracted text files from native or third-party applications—can be moved to Amazon S3 and evaluated by Macie.

AWS IAM Access Analyzer

AWS IAM Access Analyzer helps you identify the resources in your AWS organization and accounts, such as Amazon S3 buckets or IAM roles, that are shared with an external entity. This detective control helps you identify unintended access to your data and resources, which is a security risk.

Access Analyzer is deployed in the Security Tooling account through the delegated administrator functionality in AWS Organizations. The delegated administrator has permissions to create and manage analyzers with the AWS organization as the zone of trust. Findings from Access Analyzer automatically flow to Security Hub. Access Analyzer also sends an event to EventBridge for each generated finding, when the status of an existing finding changes, and when a finding is deleted. EventBridge can further direct these events to notification or remediation streams.

Design consideration
To get account-scoped findings (where the account serves as the trusted boundary), you create an account-scoped analyzer in each member account. This should be done as part of the account pipeline. Account-scoped findings flow into Security Hub at the member account level. From there, they flow to the Security Hub delegated administrator account (Security Tooling).

AWS Firewall Manager

AWS Firewall Manager helps protect your network by simplifying your administration and maintenance tasks for AWS WAF, AWS Shield Advanced, Amazon VPC security groups, AWS Network Firewall, and Route 53 Resolver DNS Firewall across multiple accounts and resources. With Firewall Manager, you set up your AWS WAF firewall rules, Shield Advanced protections, Amazon VPC security groups, AWS Network Firewall firewalls, and DNS Firewall rule group associations only once. The service automatically applies the rules and protections across your accounts and resources, even as you add new resources.

Firewall Manager is particularly useful when you want to protect your entire AWS organization instead of a small number of specific accounts and resources, or if you frequently add new resources that

you want to protect. Firewall Manager uses security policies to let you define a set of configurations, including relevant rules, protections, and actions that must be deployed and the accounts and resources (indicated by tags) to include or exclude. You can create granular and flexible configurations while still being able to scale control out to large numbers of accounts and VPCs. These policies automatically and consistently enforce the rules you configure even when new accounts and resources are created. Firewall Manager is enabled in all accounts through AWS Organizations, and configuration and management are performed by the appropriate security teams in the Firewall Manager delegated administrator account (in this case, the Security Tooling account).

You must enable AWS Config for each AWS Region that contains the resources that you want to protect. If you don't want to enable AWS Config for all resources, you must enable it for resources that are associated with the type of Firewall Manager policies that you use. When you use both AWS Security Hub and Firewall Manager, Firewall Manager automatically sends your findings to Security Hub. Firewall Manager creates findings for resources that are out of compliance and for attacks that it detects, and sends the findings to Security Hub. When you set up a Firewall Manager policy for AWS WAF, you can centrally enable logging on web access control lists (web ACLs) for all in-scope accounts and centralize the logs under a single account.

Design consideration
Account managers of individual member accounts in the AWS organization can configure additional controls (such as AWS WAF rules and VPC security groups) in the Firewall Manager managed services according to their particular needs.

Amazon EventBridge

Amazon EventBridge is a serverless event bus service that makes it easy to connect your applications with data from a variety of sources. It is frequently used in security automation. You can set up routing rules to determine where to send your data to build application architectures that react in real time to all your data sources. You can create a custom event bus to receive events from your custom applications, in addition to using the default event bus in each account. You should create an event bus in the Security Tooling account that can receive security-specific events from other accounts in the AWS organization. For example, by linking AWS Config rules, GuardDuty, and Security Hub with EventBridge, you create a flexible, automated pipeline for routing security data, raising alerts, and managing actions to resolve issues.

Design considerations

- EventBridge is capable of routing events to a number of different targets. One valuable pattern for automating security actions is to connect particular events to individual AWS Lambda responders, which take appropriate actions. For example, in certain circumstances you might want to use EventBridge to route a public S3 bucket finding to a Lambda responder that corrects the bucket policy and removes the public permissions. These responders should be integrated into your investigative playbooks and runbooks to coordinate response activities.
- A best practice for a successful security operations team is to integrate the flow of security events and findings into a notification and workflow system such as a ticketing system, a bug/issue system, or another security information and event management (SIEM) system. This takes the workflow out of email and static reports, and helps you route, escalate, and manage events or findings. The flexible routing abilities in EventBridge are a powerful enabler for this integration.

Amazon Detective

Amazon Detective supports your responsive security control strategy by making it easy to analyze, investigate, and quickly identify the root cause of security findings or suspicious activities. Detective automatically extracts time-based events such as login attempts, API calls, and network traffic from

AWS CloudTrail logs and Amazon VPC flow logs. Detective consumes these events by using independent streams of CloudTrail logs and VPC flow logs. Detective uses machine learning and visualization to create a unified, interactive view of the behavior of your resources and the interactions among them over time—this is called a *behavior graph*. You can explore the behavior graph to examine disparate actions such as failed logon attempts or suspicious API calls.

Detective also ingests findings that are detected by Amazon GuardDuty. When an account enables Detective, it becomes the administrator account for the behavior graph. Before you try to enable Detective, make sure that your account has been enrolled in GuardDuty for at least 48 hours. If you do not meet this requirement, you cannot enable Detective.

Administrator accounts invite member accounts to contribute their data to the primary account's behavior graph. When a member account accepts the invitation and is enabled, Detective begins to ingest and extract the member account's data into that behavior graph.

Design consideration
You can navigate to Detective finding profiles from the GuardDuty and Security Hub consoles. These links can help streamline the investigation process. Your account must be the administrative account for both Detective and the service you are pivoting from (GuardDuty or Security Hub). If the primary accounts are the same for the services, the integration links work seamlessly.

Deploying common security services within all AWS accounts

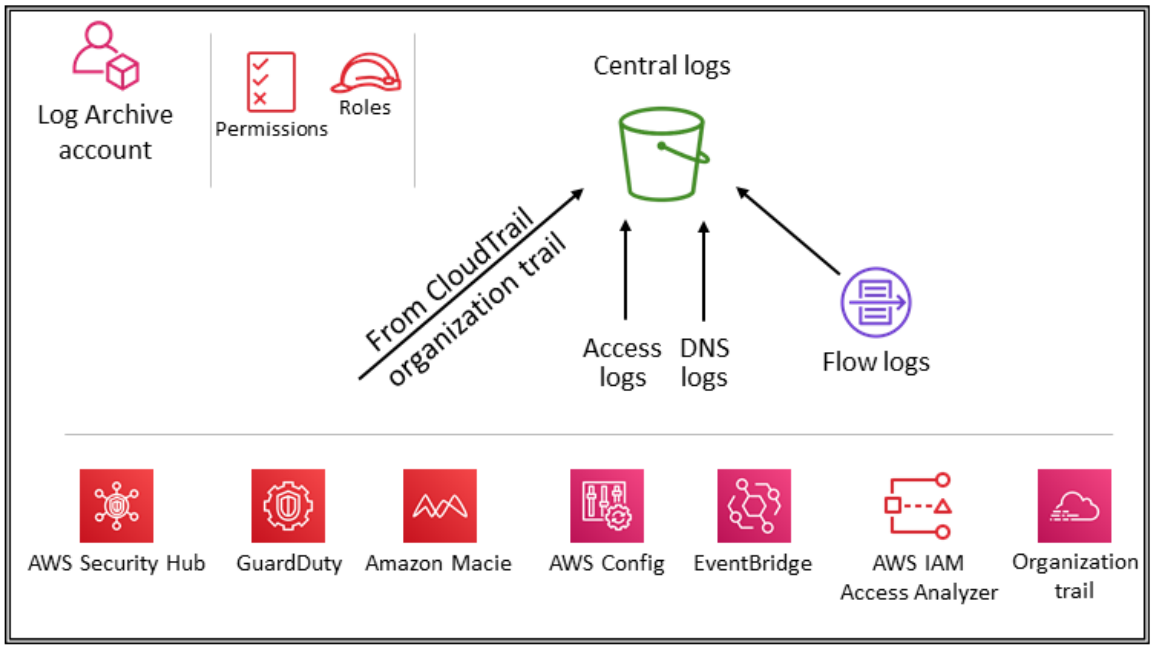
The Apply security services across your AWS organization (p. 11) section earlier in this reference highlighted security services that protect an AWS account, and noted that many of these services can also be configured and managed within AWS Organizations. Some of these services should be deployed in all accounts, and you will see them in the AWS SRA. This enables a consistent set of guardrails and provides centralized monitoring, management, and governance across your AWS organization.

Security Hub, GuardDuty, AWS Config, Access Analyzer, CloudTrail organization trails, and EventBridge appear in all accounts. The first four support the delegated administrator feature discussed previously in the management account, trusted access, and delegated administrators (p. 6) section. CloudTrail currently uses a different aggregation mechanism. EventBridge doesn't use the centralized control and monitoring mechanism, but this service is included because of its integral role in automating alerts and responses.

Design consideration
Specific account configurations might necessitate additional security services. For example, accounts that manage Amazon Simple Storage Service (Amazon S3) buckets (our Application account and Log Archive account) should also include Amazon Macie and consider turning on CloudTrail S3 data event logging in these common security services. (Macie supports delegated administration with centralized configuration and monitoring.)

Security OU – Log Archive account

The following diagram illustrates the AWS security services that are configured in the Log Archive account.



The Log Archive account is dedicated to ingesting and archiving all security-related logs and backups. With centralized logs in place, you can monitor, audit, and alert on Amazon S3 object access, unauthorized activity by identities, IAM policy changes, and other critical activities performed on sensitive resources. The security objectives are straightforward: This should be immutable storage, accessed only by controlled, automated, and monitored mechanisms, and built for durability (for example, by using the appropriate replication and archival processes). Controls should be implemented at depth to protect the integrity and availability of the logs and log management process. In addition to preventive controls, such as assigning least privilege roles to be used for access and encrypting logs with a controlled AWS KMS key, use detective controls such as AWS Config to monitor (and alert and remediate) this collection of permissions for unexpected changes.

Design consideration
Operational log data used by your infrastructure, operations, and workload teams often overlaps with the log data used by security, audit, and compliance teams. We recommend that you consolidate your operational log data into the Log Archive account. Based on your specific security and governance requirements, you might need to filter operational log data saved to this account. You might also need to specify who has access to the operational log data in the Log Archive account.

Types of logs

The primary logs shown in the AWS SRA include AWS CloudTrail (organization trail), Amazon VPC flow logs, access logs from Amazon CloudFront and AWS WAF, and DNS logs from Amazon Route 53. These logs provide an audit of actions taken (or attempted) by a user, role, AWS service, or network entity (identified, for example, by an IP address). Other log types (for example, application logs or database logs) can be captured and archived as well. For more information about log sources and logging best practices, see the security documentation for each service.

Amazon S3 as central log store

By logging to a dedicated and centralized S3 bucket that resides in a dedicated account, you can enforce strict security controls, access, and separation of duties.

By default, the log files delivered by CloudTrail to the bucket are encrypted by Amazon server-side encryption with Amazon S3-managed encryption keys (SSE-S3). To provide a security layer that is directly manageable, you should use server-side encryption with customer master keys (CMKs) that you manage (SSE-KMS) on all security log files. With this feature, in order to read log files, a user must have both Amazon S3 read permissions for the bucket that contains the log files and an IAM policy or role applied that allows decrypt permissions by the associated key policy. Additionally, CloudTrail provides *log file integrity validation* to determine whether a log file was modified, deleted, or unchanged after CloudTrail delivered it.

Configure the multi-factor authentication (MFA) **Delete** feature for the log archive bucket to ensure that any attempt to change the versioning state of the bucket or to permanently delete an object version requires additional authentication. This helps prevent any operation that could compromise the integrity of your log files.

You can use the Amazon S3 object lifecycle management rules to define your own retention policy to better meet your business and auditing needs. For example, you might want to archive log files that are more than a year old in Amazon S3 Glacier, or delete log files after a certain amount of time has passed.

In addition to protecting the S3 bucket itself, you should adhere to the principle of least privilege for the logging services (for example, CloudTrail) and the Log Archive account. For example, users with permissions granted by the AWS managed IAM policy **AWSCloudTrail_FullAccess** have the ability to disable or reconfigure the most sensitive and important auditing functions in their AWS accounts. Limit the application of this IAM policy to as few individuals as possible.

Use detective controls, such as those delivered by AWS Config and AWS IAM Access Analyzer, to monitor (and alert and remediate) this broader collective of preventive controls for unexpected changes.

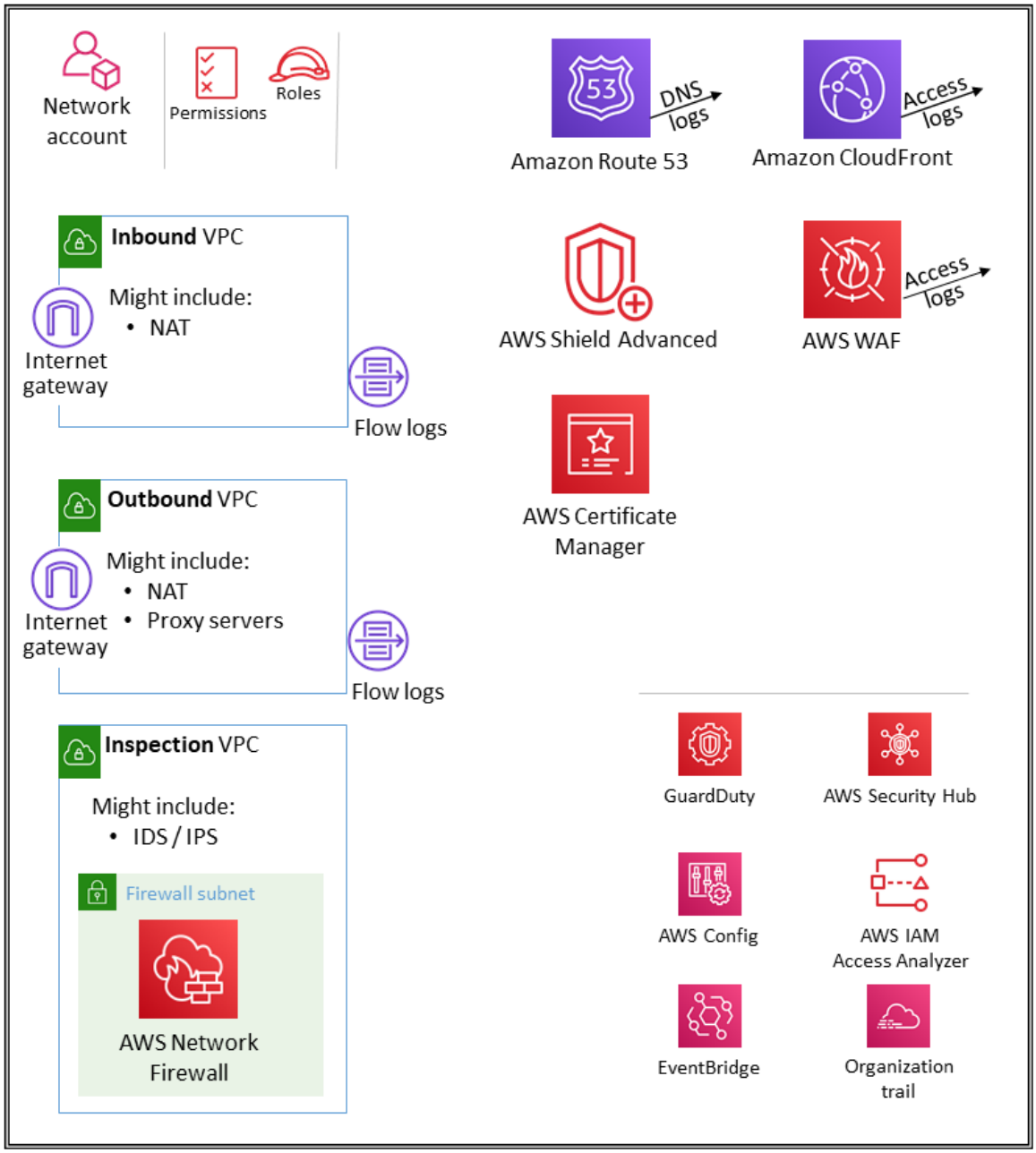
For a deeper discussion of security best practices for S3 buckets, see the Amazon S3 documentation, online tech talks, and blog posts.

Security service guardrails

In the AWS SRA, AWS Security Hub, Amazon GuardDuty, Amazon Macie, AWS Config, AWS IAM Access Analyzer, AWS CloudTrail organization trails, and Amazon EventBridge are deployed with appropriate delegated administration to the Security Tooling account. This enables a consistent set of guardrails and provides centralized monitoring, management, and governance across your AWS organization.

Infrastructure OU – Network account

The following diagram illustrates the AWS security services that are configured in the Network account.



The Network account manages the gateway between your application and the broader internet. It is important to protect that two-way interface. The Network account isolates the networking services, configuration, and operation from the individual application workloads, security, and other infrastructure. This arrangement not only limits connectivity, permissions, and data flow, but also supports separation of duties and least privilege for the teams that need to operate in these accounts. By splitting network flow into separate inbound and outbound virtual private clouds (VPCs), you can protect sensitive infrastructure and traffic from undesired access. The inbound network is generally considered higher risk and deserves appropriate routing, monitoring, and potential issue mitigations. These infrastructure accounts will inherit permission guardrails from the Org Management account and the Infrastructure OU. Networking (and security) teams will manage the majority of the infrastructure here.

Network architecture

Although network design and specifics are beyond the scope of this document, we recommend these three options for network connectivity between the various accounts: VPC peering, AWS PrivateLink, and AWS Transit Gateway. Important considerations in choosing among these are operational norms, budgets, and specific bandwidth needs.

- VPC peering – The simplest way to connect two VPCs is to use VPC peering. A connection enables full bidirectional connectivity between the VPCs. VPCs across accounts and AWS Regions can also be peered together. At scale, when you have tens to hundreds of VPCs, interconnecting them with peering results in a mesh of hundreds to thousands of peering connections, which can be challenging to manage and scale. VPC peering is best used when resources in one VPC must communicate with resources in another VPC, the environment of both VPCs is controlled and secured, and the number of VPCs to be connected is fewer than 10 (to allow for the individual management of each connection).
- AWS PrivateLink – AWS PrivateLink provides private connectivity between VPCs, services, and application. You can create your own application in your VPC and configure it as an AWS PrivateLink-powered service (referred to as an *endpoint service*). Other AWS principals can create a connection from their VPC to your endpoint service by using an interface VPC endpoint or a Gateway Load Balancer endpoint, depending on the type of service. When you use AWS PrivateLink, service traffic doesn't pass across a publicly routable network. Use AWS PrivateLink when you have a client-server setup where you want to give one or more consumer VPCs unidirectional access to a specific service or set of instances in the service provider VPC. This is also a good option when clients and servers in the two VPCs have overlapping IP addresses, because AWS PrivateLink uses elastic network interfaces within the client VPC so that there are no IP conflicts with the service provider.
- AWS Transit Gateway – AWS Transit Gateway provides a hub-and-spoke design for connecting VPCs and on-premises networks as a fully managed service without requiring you to provision virtual appliances. AWS manages high availability and scalability. A transit gateway is a regional resource and can connect thousands of VPCs within the same AWS Region. You can attach all your hybrid connectivity (VPN and AWS Direct Connect connections) to a single transit gateway, thereby consolidating and controlling your AWS organization's entire routing configuration in one place. A transit gateway solves the complexity involved with creating and managing multiple VPC peering connections at scale. It is a good default for most network architectures, but specific needs around cost, bandwidth, and latency might make VPC peering a better fit for your needs.

Inbound (ingress) VPC

The inbound VPC is intended to accept, inspect, and route network connections initiated outside the application. Depending on the particulars of the application, we expect to see some network address translation (NAT) in this VPC. Flow logs from this VPC are captured and stored in the Log Archive account.

Outbound (egress) VPC

The outbound VPC is intended to handle network connections initiated from within the application. Depending on the particulars of the application, we expect to see traffic NAT, AWS service-specific VPC endpoints, and hosting of external API endpoints in this VPC. Flow logs from this VPC are captured and stored in the Log Archive account.

Inspection VPC

A dedicated inspection VPC provides a simplified and central approach for managing inspections between VPCs (in the same or in different AWS Regions), the internet, and on-premises networks. For the AWS SRA, ensure that all traffic between VPCs passes through the inspection VPC, and avoid using the inspection VPC for any other workload.

AWS Network Firewall

AWS Network Firewall is a highly available, managed network firewall service for your VPC. It enables you to easily deploy and manage stateful inspection, intrusion prevention and detection, and web filtering to protect your virtual networks on AWS. For more information about configuring Network Firewall, see the AWS Network Firewall – New Managed Firewall Service in VPC blog post.

You use a firewall on a per-Availability Zone basis in your VPC. For each Availability Zone, you choose a subnet to host the firewall endpoint that filters your traffic. The firewall endpoint in an Availability Zone can protect all the subnets inside the zone except for the subnet where it's located. Depending on the use case and deployment model, the firewall subnet could be either public or private. The firewall is completely transparent to the traffic flow and does not perform network address translation (NAT). It preserves the source and destination address. In this reference architecture, the firewall endpoints are hosted in an inspection VPC. All traffic from the inbound VPC and to the outbound VPC is routed through this firewall subnet for inspection.

Network Firewall makes firewall activity visible in real time through Amazon CloudWatch metrics, and offers increased visibility of network traffic by sending logs to Amazon Simple Storage Service (Amazon S3), CloudWatch, and Amazon Kinesis Data Firehose. Network Firewall is interoperable with your existing security approach, including technologies from AWS Partners. You can also import existing Suricata rulesets, which may have been written internally or sourced externally from third-party vendors or open-source platforms.

Design considerations

- AWS Firewall Manager supports AWS Network Firewall, and makes it easy to centrally configure and deploy Network Firewall rules across your organization. (For details, see AWS Network Firewall policies in the AWS documentation.) When you configure Firewall Manager, it automatically creates a Network Firewall firewall with sets of rules in the accounts and VPCs that you specify. It also deploys an endpoint in a dedicated subnet for every Availability Zone that contains public subnets. At the same time, any changes to the centrally configured set of rules are automatically updated downstream on the deployed Network Firewall firewalls.
- There are multiple deployment models available with Network Firewall. The right model depends on the use case and requirements. Examples include the following:
  - A distributed deployment model where Network Firewall is deployed into individual VPCs.
  - A centralized deployment model, where Network Firewall is deployed into a centralized VPC for east-west (VPC-to-VPC) and/or north-south (internet egress and ingress, on-premises) traffic.
  - A combined deployment model where Network Firewall is deployed into a centralized VPC for east-west and a subset of north-south traffic.
- As a best practice, do not use the Network Firewall subnet to deploy any other services. This is because Network Firewall cannot inspect traffic from sources or destinations within the firewall subnet.

AWS Certificate Manager (ACM)

ACM handles the complexity of creating, storing, and renewing SSL/TLS X.509 certificates and keys that protect your applications. In the inbound VPC, ACM-provisioned public certificates are deployed through Amazon Route 53 (by enabling DNS ownership validation by using CNAME). For additional uses of ACM, see the section on the Workloads OU (p. 35) later in this document.

Design consideration
For externally facing certificates, ACM must reside in the same account as the resources for which it provisions certificates. Certificates cannot be shared across accounts.

AWS WAF

AWS WAF is a web application firewall that lets you monitor the HTTP and HTTPS requests that are forwarded to an Amazon CloudFront distribution, an Amazon API Gateway REST API, an Application Load Balancer, or an AWS AppSync GraphQL API. AWS WAF also lets you control access to your content. Based on conditions that you specify, such as the IP addresses that requests originate from, or the values of query strings, the fronted service responds to requests either with the requested content or with an HTTP status code 403 (Forbidden) status code. In this architecture, AWS WAF protects CloudFront.

Design considerations

- CloudFront provides features that enhance the AWS WAF functionality and make the two services work better together.
- You can use AWS WAF, AWS Firewall Manager, and AWS Shield together to create a comprehensive security solution. It all starts with AWS WAF. You can automate and then simplify AWS WAF management using Firewall Manager. Shield Advanced provides additional features on top of AWS WAF, such as dedicated support from the distributed denial of service (DDoS) Response Team (DRT) and advanced reporting. If you want granular control over the protection that is added to your resources, AWS WAF alone is the right choice. If you want to use AWS WAF across accounts, accelerate your AWS WAF configuration, or automate protection of new resources, use Firewall Manager with AWS WAF. Finally, if you own high visibility websites or are otherwise prone to frequent DDoS attacks, you should consider purchasing the additional features that Shield Advanced provides.

Amazon CloudFront

Amazon CloudFront is a highly secure content delivery network (CDN) that provides both network-level and application-level protection. You can deliver your content, APIs, or applications by using SSL/TLS certificates, and advanced SSL features are enabled automatically. You can use AWS Certificate Manager (ACM) to easily create a *custom SSL certificate* and deploy content to your CloudFront distribution for free. Additionally, you can restrict access to your content by using a number of capabilities:

- By using signed URLs and signed cookies, you can support token authentication to restrict access to only authenticated viewers.
- Through the geo restriction capability, you can prevent users in specific geographic locations from accessing content that you're distributing through CloudFront.
- You can use the origin access identity (OAI) feature to restrict access to an S3 bucket to be accessible only from CloudFront.

Design considerations

- CloudFront, AWS Shield, AWS WAF, and Amazon Route 53 work seamlessly together to create a flexible, layered security perimeter against multiple types of attacks, including network and application-layer DDoS attacks. CloudFront provides features that enhance the AWS WAF functionality and make the two work better together. For more information, see How AWS WAF works with Amazon CloudFront features in the AWS documentation.
- When you deliver web content through a CDN such as CloudFront, a best practice is to prevent viewer requests from bypassing the CDN and accessing your origin content directly. For more information, see the blog post How to enhance Amazon CloudFront origin security with AWS WAF and AWS Secrets Manager.

AWS Shield

AWS Shield is a managed DDoS protection service that safeguards applications that run on AWS. Shield provides always-on detection and automatic inline mitigations that minimize application downtime and latency, so there is no need to engage AWS Support to benefit from DDoS protection. In the AWS SRA, Shield Advanced is configured to protect Route 53 and CloudFront.

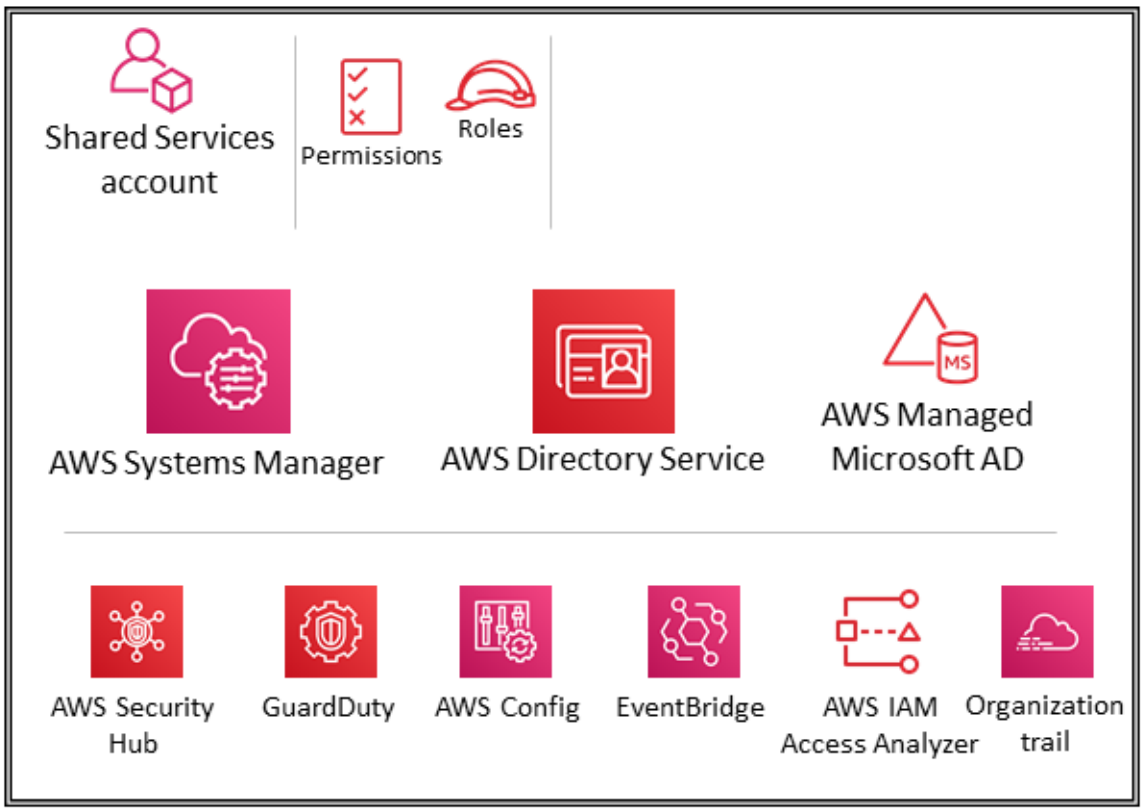
Design consideration
There are two tiers of Shield: Shield Standard and Shield Advanced. All AWS customers benefit from the automatic protections of Shield Standard at no additional charge. Shield Standard provides protection against the most common and frequently occurring infrastructure (layer 3 and 4) attacks. Shield Standard uses deterministic packet filtering and priority-based traffic shaping to automatically mitigate basic network layer attacks. Shield Advanced provides more sophisticated automatic mitigations for attacks that target your applications running on protected Amazon Elastic Compute Cloud (Amazon EC2), Elastic Load Balancing (ELB), CloudFront, AWS Global Accelerator, and Route 53 resources. Shield Advanced records metrics that you can monitor in CloudWatch. (For more information, see AWS Shield Advanced metrics and alarms in the AWS documentation.) If you own high-visibility websites or are otherwise prone to frequent DDoS attacks, you should consider the additional features that Shield Advanced provides.

Security service guardrails

In the AWS SRA, AWS Security Hub, Amazon GuardDuty, AWS Config, AWS IAM Access Analyzer, AWS CloudTrail organization trails, and Amazon EventBridge are deployed with appropriate delegated administration to the Security Tooling account. This enables a consistent set of guardrails and provides centralized monitoring, management, and governance across your AWS organization.

Infrastructure OU – Shared Services account

The following diagram illustrates the AWS security services that are configured in the Shared Services account.



The Shared Services account is part of the Infrastructure OU, and its purpose is to support the services that multiple applications and teams use to deliver their outcomes. For example, directory services (Active Directory), messaging services, and metadata services are in this category. The AWS SRA highlights the shared services that support security controls. Although the Network accounts are also part of the Infrastructure OU, they are removed from the Shared Services account to support the separation of duties. The teams that will manage these services don't need permissions or access to the Network accounts.

AWS Systems Manager

AWS Systems Manager (which is also included in the Org Management account and in the Application account) provides a collection of capabilities that enable visibility and control of your AWS resources. One of these capabilities, Systems Manager Explorer, is a customizable operations dashboard that reports information about your AWS resources. You can synchronize operations data across all accounts in your AWS organization by using AWS Organizations and Systems Manager Explorer. Systems Manager is deployed in the Shared Services account through the delegated administrator functionality in AWS Organizations.

AWS Directory Service

AWS Directory Service enables administrators to connect their self-managed Microsoft Active Directory (AD) or their AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) directory to AWS Single Sign-On (AWS SSO). (See also AWS organization and account structure of the AWS SRA (p. 8) earlier in this document.) This Microsoft AD directory defines the pool of identities that administrators can pull from when using the AWS SSO console to assign SSO access. After administrators connect their corporate directory to AWS SSO, they can grant their AD users or groups access to AWS

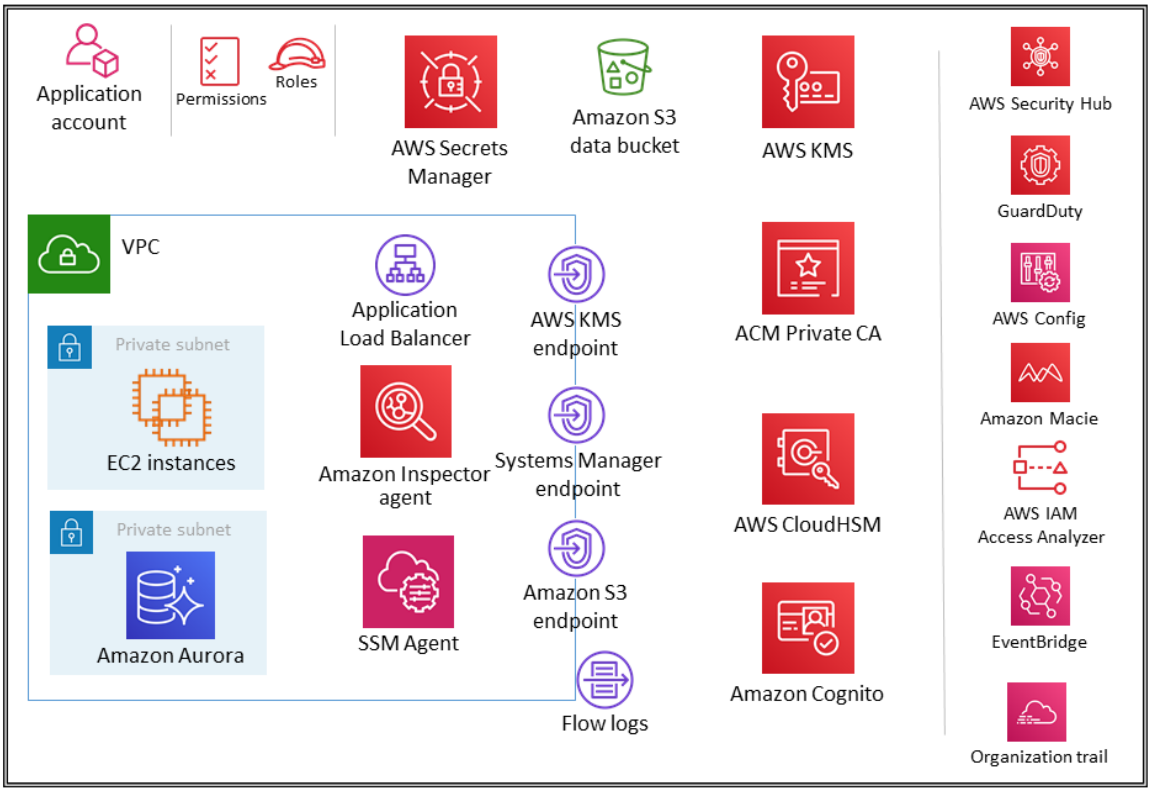
accounts, cloud applications, or both. AWS Directory Service helps you set up and run a standalone AWS Managed Microsoft AD directory that is hosted in the AWS Cloud. You can also use AWS Directory Service to connect your AWS resources with an existing, self-managed AD.

Security service guardrails

In the AWS SRA, AWS Security Hub, Amazon GuardDuty, AWS Config, AWS IAM Access Analyzer, AWS CloudTrail organization trails, and Amazon EventBridge are deployed with appropriate delegated administration to the Security Tooling account. This enables a consistent set of guardrails and provides centralized monitoring, management, and governance across your AWS organization.

Workloads OU – Application account

The following diagram illustrates the AWS security services that are configured in the Application account (along with the application itself).



The Application account hosts the primary infrastructure and services to run and maintain an enterprise application. The Application account and Workloads OU serve a few primary security objectives. First, you create a separate account for each application to provide boundaries and controls between workloads so that you can avoid issues of comingling roles, permissions, data, and encryption keys. You want to provide a separate account container where the application team can be given broad rights to manage their own infrastructure without affecting others. Next, you add a layer of protection by providing a mechanism for the security operations team to monitor and collect security data. Employ an organization trail and local deployments of account security services (Amazon GuardDuty, AWS Config, AWS Security Hub, Amazon EventBridge, AWS IAM Access Analyzer), which are configured and monitored by the security team. Finally, you enable your enterprise to set controls centrally. You align

the application account to the broader security structure by making it a member of the Workloads OU through which it inherits appropriate service permissions, constraints, and guardrails.

Application VPC

The virtual private cloud (VPC) in the Application account needs both inbound access (for the simple web services that you are modeling) and outbound access (for application needs or AWS service needs). By default, all resources inside a VPC are routable to each other. There are two private subnets: one to host the Amazon Elastic Compute Cloud (Amazon EC2) instances (application layer) and the other for Amazon Aurora (database layer). Network segmentation between different tiers, such as the application tier and database tier, is accomplished through VPC security groups, which restrict traffic at the instance level. For resiliency, the workload spans two or more Availability Zones and utilizes two subnets per zone.

Design consideration
You can use Traffic Mirroring to copy network traffic from an elastic network interface of EC2 instances. You can then send the traffic to out-of-band security and monitoring appliances for content inspection, threat monitoring, or troubleshooting. For example, you might want to monitor the traffic that is leaving your VPC or the traffic whose source is outside your VPC. In this case, you will mirror all traffic except for the traffic passing within your VPC and send it to a single monitoring appliance. VPC flow logs do not capture mirrored traffic; they generally capture information from packet headers only. Traffic Mirroring provides deeper insight into the network traffic by allowing you to analyze actual traffic content, including payload. Enable Traffic Mirroring only for the elastic network interface of EC2 instances that might be operating as part of sensitive workloads or for which you expect to need detailed diagnostics in the event of an issue.

VPC endpoints

VPC endpoints provide another layer of security control as well as scalability and reliability. Use these to connect your application VPC to other AWS services. (In the Application account, the AWS SRA employs VPC endpoints for AWS KMS, AWS Systems Manager, and Amazon S3.) Endpoints are virtual devices. They are horizontally scaled, redundant, and highly available VPC components. They allow communication between instances in your VPC and services without imposing availability risks or bandwidth constraints on your network traffic. You can use a VPC endpoint to privately connect your VPC to supported AWS services and VPC endpoint services powered by AWS PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC do not require public IP addresses to communicate with other AWS services. Traffic between your VPC and the other AWS service does not leave the Amazon network.

Another benefit of using VPC endpoints is to enable the configuration of endpoint policies. A VPC endpoint policy is an IAM resource policy that you attach to an endpoint when you create or modify the endpoint. If you do not attach an IAM policy when you create an endpoint, AWS attaches a default IAM policy for you that allows full access to the service. An endpoint policy does not override or replace IAM user policies or service-specific policies (such as S3 bucket policies). It is a separate IAM policy for controlling access from the endpoint to the specified service. In this way, it adds another layer of control over which AWS principals can communicate with resources or services.

Amazon EC2

The EC2 instances that compose our application make use of version 2 of the Instance Metadata Service (IMDSv2). IMDSv2 adds protections for four types of vulnerabilities that could be used to try to access the IMDS: website application firewalls, open reverse proxies, server-side request forgery (SSRF) vulnerabilities, open layer 3 firewalls, and NATs. For more information, see the blog post Add defense in depth against open firewalls, reverse proxies, and SSRF vulnerabilities with enhancements to the EC2 Instance Metadata Service.

Note
Although in practice, AWS Systems Manager Session Manager and Amazon Inspector agents are deployed on the EC2 instances, the Workloads OU diagram shows them separately for ease of readability.

Application Load Balancers

Application Load Balancers distribute incoming application traffic across multiple targets, such as EC2 instances, in multiple Availability Zones. In the AWS SRA, the target group for the load balancer are the application EC2 instances. The AWS SRA uses HTTPS listeners to ensure that the communication channel is encrypted. The Application Load Balancer uses a server certificate to terminate the front-end connection, and then to decrypt requests from clients before sending them to the targets.

AWS Certificate Manager (ACM) natively integrates with Application Load Balancers, and the AWS SRA uses ACM to generate and manage the necessary X.509 (SSL/TLS server) certificates. You can enforce TLS 1.2 and strong ciphers for front-end connections through the Application Load Balancer security policy. For more information, see the Elastic Load Balancing documentation.

Design considerations

- You can alternatively use SSL/TLS tools to create a certificate signing request (CSR), get the CSR signed by a certificate authority (CA) to produce a certificate, and then import the certificate into ACM or upload the certificate to IAM for use with the Application Load Balancer. If you import a certificate into ACM, you must monitor the expiration date of the certificate and renew it before it expires. If you import a certificate into IAM, you must create a new certificate, import the new certificate to ACM or IAM, add the new certificate to your load balancer, and remove the expired certificate from your load balancer. In addition to enforcing the HTTPS connection, you can configure the load balancer to securely authenticate users as they access your backend application. For identity management you can use an identity provider (IdP) that is OpenID Connect (OIDC) compliant, use social IdPs, or authenticate users through corporate identities, using SAML 2.0 (Security Assertion Markup Language 2.0), Lightweight Directory Access Protocol (LDAP), or Microsoft AD. For more information, see the Elastic Load Balancing documentation.
- For additional layers of defense, you can deploy AWS WAF policies to protect the Application Load Balancer. Having edge policies, application policies, and even private or internal policy enforcement layers adds to the visibility of communication requests and provides unified policy enforcement. For more information, see the blog post Deploying defense in depth using AWS Managed Rules for AWS WAF.

Amazon Inspector

Amazon Inspector implements two types of detective controls, which test the network accessibility of your EC2 instances and the security state of your applications that run on those instances.

The Network Reachability rules package of Amazon Inspector assesses the accessibility of your EC2 instances to or from the internet. These rules help automate the monitoring of your AWS networks and identify where network access to your EC2 instances might be misconfigured. The findings show whether your EC2 instance ports are reachable from the internet. These findings also highlight network configurations that allow for potentially malicious access, such as mismanaged security groups, access control lists (ACLs), internet gateways, and so on. For more information, see the Amazon Inspector documentation.

By installing the Amazon Inspector agent, you can further assess the EC2 host itself for exposure to common vulnerabilities and exposures (CVEs), alignment to Center for Internet Security (CIS) benchmarks, and alignment with AWS security best practices. For more information, see the Amazon Inspector documentation.

Design considerations

- Amazon Inspector integrates with AWS Security Hub if both services are enabled in the same AWS account. You can use this integration to send all findings from Amazon Inspector to Security Hub, which will then include those findings in its analysis of your security posture.
- The Amazon Inspector agent initiates all communication with the Amazon Inspector service. This means that the agent must have an outbound network path to public AWS endpoints so that it can send telemetry data. The agent periodically communicates with Amazon Inspector over a TLS-protected channel, which is authenticated in one of two ways: by using the AWS identity associated with the role of the EC2 instance, or, if no role is assigned, by using the AWS identity associated with the instance's metadata document.

AWS Systems Manager

AWS Systems Manager is an AWS service that you can use to view operational data from multiple AWS services and automate operational tasks across your AWS resources. With automated approval workflows and runbooks, you can reduce human error and simplify maintenance and deployment tasks on AWS resources.

In addition to these general automation capabilities, Systems Manager supports a number of preventive, detective, and responsive security features. Systems Manager Agent (SSM Agent) is Amazon software that can be installed and configured on an EC2 instance, an on-premises server, or a virtual machine (VM). SSM Agent makes it possible for Systems Manager to update, manage, and configure these resources. SSM helps you maintain security and compliance by scanning these managed instances and reporting (or taking corrective action) on any violations it detects in your patch, configuration, and custom policies.

The AWS SRA uses AWS Systems Manager Session Manager to provide an interactive, browser-based shell and CLI experience. This provides secure and auditable instance management without the need to open inbound ports, maintain bastion hosts, or manage SSH keys. The AWS SRA uses AWS Systems Manager Patch Manager to apply patches to EC2 instances for both operating systems and applications.

Design considerations

- Systems Manager relies on EC2 instance metadata to function correctly. Systems Manager can access instance metadata by using either version 1 or version 2 of the Instance Metadata Service (IMDSv1 and IMDSv2).
- SSM Agent has to communicate with different AWS services and resources such as EC2 messages, Systems Manager, and Amazon S3. For this communication to happen, the subnet requires either outbound internet connectivity or provisioning of appropriate VPC endpoints. The AWS SRA uses the endpoints.

Amazon Aurora

In the AWS SRA, Amazon Aurora and Amazon S3 make up the logical data tier. Aurora is a fully managed relational database engine that's compatible with MySQL and PostgreSQL. An application that is running on the EC2 instances communicates with Aurora and Amazon S3 as needed. Aurora is configured with a database cluster inside a DB subnet group.

Design consideration
As in many database services, security for Aurora is managed at three levels. To control who can perform Amazon Relational Database Service (Amazon RDS) management actions on Aurora DB clusters and DB instances, you use IAM. To control which devices and EC2 instances can open connections to the cluster endpoint and port of the DB instance for Aurora DB clusters in a VPC, you use a VPC security group. To authenticate logins and permissions for an Aurora DB cluster, you can take the same approach as with a stand-alone DB instance of MySQL or PostgreSQL, or

you can use IAM database authentication for Aurora MySQL-Compatible Edition. With this latter approach, you authenticate to your Aurora MySQL-Compatible DB cluster by using an IAM role and an authentication token.

Amazon S3

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance. It is the data backbone of many applications built on AWS, and appropriate permissions and security controls are critical for protecting sensitive data. For recommended security best practices for Amazon S3, see the documentation, online tech talks, and deeper dives in blog posts. The most important best practice is to block overly permissive access (especially public access) to S3 buckets.

AWS KMS

AWS Key Management Service (AWS KMS) makes it easy for you to create and manage cryptographic keys and control their use across a wide range of AWS services and in your applications. By defining an encryption approach that includes the storage, rotation, and access control of keys, you can help provide protection for your content against unauthorized users and against unnecessary exposure to authorized users. AWS KMS is a secure and resilient service that uses hardware security modules. Customer master keys (CMKs) are the primary resources in AWS KMS. A CMK is a logical representation of a master key. For protection and flexibility, AWS KMS supports three types of CMKs: customer managed CMKs, AWS managed CMKs, and AWS owned CMKs. Customer managed CMKs are CMKs in your AWS account that you create, own, and manage. AWS managed CMKs are CMKs in your account that are created, managed, and used on your behalf by an AWS service that is integrated with AWS KMS. AWS owned CMKs are a collection of CMKs that an AWS service owns and manages for use in multiple AWS accounts. For more information about using KMS keys, see the AWS KMS documentation and the AWS Key Management Service Best Practices technical paper.

Keep CMKs in the same accounts and AWS Regions as the data encryption keys that they encrypt. In the Application account, AWS KMS is used to manage keys that are specific to the application, and permissions can be granted to local application roles as well as to appropriate security teams or administrators for some separation of duties. In the Security Tooling account, AWS KMS is used to manage the encryption of centralized security services such as the AWS CloudTrail organization trail that is managed by the AWS organization.

Design considerations

- You have a choice for how to deploy AWS KMS. You can allow the workload teams that handle customer data to create, manage, and use KMS keys in the AWS accounts that they manage. This model affords the workload teams more control, flexibility, and agility over the use of encryption keys. Alternatively, you can separate the responsibility for the creation and management of KMS keys into a centralized security account while delegating only the ability to use the keys to the workload teams. This second option facilitates better separation of responsibilities and prevents the workload teams from accidentally deleting or escalating privilege on KMS keys without also involving the security account.
- You should use appropriate monitoring and detective controls for additional security layers. AWS KMS is integrated with AWS CloudTrail and AWS Config. CloudTrail provides you with logs of all key usage to help meet your regulatory and compliance needs. AWS Config monitors and records all changes in your KMS keys and the associated KMS key policies (IAM resource policies).

AWS CloudHSM

AWS CloudHSM provides managed hardware security modules (HSMs) in the AWS Cloud. It enables you to generate and use your own encryption keys on AWS by using FIPS 140-2 level 3 validated HSMs that

you control access to. You can use AWS CloudHSM to offload SSL/TLS processing for your web servers. This reduces the burden on the web server and provides extra security by storing the web server's private key in AWS CloudHSM. You could similarly deploy an HSM from AWS CloudHSM in the inbound VPC in the Network account to store your private keys and sign certificate requests if you need to act as an issuing certificate authority.

Design consideration
If you have a hard requirement for FIPS 140-2 level 3, you can also choose to configure AWS KMS to use the AWS CloudHSM cluster as a custom key store rather than using the native CMK store. By doing this, you benefit from the integration between AWS KMS and AWS services that encrypt your data, while being responsible for the HSMs that protect your CMKs. This combines single-tenant HSMs under your control with the ease of use and integration of AWS KMS. To manage your AWS CloudHSM infrastructure, you should employ a public key infrastructure (PKI) or security team that has experience managing HSMs.

ACM Private CA

AWS Certificate Manager Private Certificate Authority (ACM Private CA) provisions and deploys a private TLS certificate that will be exported and used on the Application Load Balancer at the front end of the application EC2 instances. This allows encrypted TLS communication to the running applications. With ACM Private CA, you can create your own CA hierarchy and issue certificates with it for authenticating internal users, computers, applications, services, servers, and other devices, and for signing computer code. Certificates issued by a private CA are trusted only within your AWS organization, not on the internet. A PKI or security team should be responsible for managing all PKI infrastructure. This includes the management and creation of the private CA. However, creation of private certificates from the private CA can be delegated to application development teams if appropriate. For additional uses of ACM, see the Network account (p. 28) section earlier in this document.

AWS Secrets Manager

AWS Secrets Manager helps you protect the credentials (*secrets*) that you need to access your applications, services, and IT resources. The service enables you to easily rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle. You can replace hardcoded credentials in your code with an API call to Secrets Manager to retrieve the secret programmatically. This helps ensure that the secret can't be compromised by someone who is examining your code, because the secret no longer exists in the code. With Secrets Manager, you can manage access to secrets by using fine-grained IAM policies and resource-based policies. You can help secure secrets by encrypting them with encryption keys that you manage by using AWS KMS. Secrets Manager also integrates with AWS logging and monitoring services for centralized auditing.

In the AWS SRA, Secrets Manager is located in the Application account to support local application use cases and to manage secrets close to their usage. In this example, an instance profile is attached to the EC2 instances in the Application account. Separate secrets can then be configured in Secrets Manager to allow that instance profile to retrieve secrets—for example, to join the appropriate Active Directory or LDAP domain and to access the Aurora database.

Design considerations

- In general, configure and manage Secrets Manager in the account that is closest to where the secrets will be used. This approach takes advantage of the local knowledge of the use case and provides speed and flexibility to application development teams. For tightly controlled information, where an additional layer of control may be appropriate, secrets can be centrally managed by Secrets Manager in the Security Tooling account.
- AWS Config can provide detective controls on these secrets. For example, it can track and monitor changes to secrets in Secrets Manager, such as the secret's description, rotation configuration, tags, the secret's relationship to other AWS sources such as the KMS encryption key or the AWS Lambda functions used for secret rotation. You can also configure Amazon

EventBridge, which receives all configuration and compliance change notifications from AWS Config, to route particular secrets events for notification or remediation actions.

IAM resources

Although AWS Identity and Access Management (IAM) is not a service that is drawn in a traditional architecture diagram, it touches every aspect of the AWS organization, AWS accounts, and AWS services. You cannot deploy any AWS services without creating IAM principals and granting permissions first. A full treatment of IAM is beyond the scope of this document, but this section provides important summaries of best practice recommendations and pointers to additional resources.

- For IAM best practices, see Security best practices in IAM in the AWS documentation, IAM articles in the AWS Security blog, and AWS re:Invent presentations.
- The AWS Well-Architected security pillar outlines key steps in the permissions management process: define permissions guardrails, grant least privilege access, analyze public and cross-account access, share resources securely, reduce permissions continuously, and establish emergency access process.
- The following table and its accompanying notes provide a high-level overview of recommended guidance on types of available IAM permission policies and how to use them in your security architecture. To learn more, see the AWS re:Invent 2020 video on choosing the right mix of IAM policies.

Policy type	Effect	Managed by	Purpose	Pertains to	Affects	Deployed in
Service control policies (SCPs)	Restrict	Central team, such as platform or security team[1]	Guardrails, governance	Organization (group of accounts)	All principals in organization, OU, and accounts	Org Management account[2]
Baseline account automation policies (the IAM roles used by the platform to operate an account)	Grant and restrict	Central team, such as platform, security, or IAM team[1]	Permissions for (baseline) non-workload performance roles[3]	Single account[4]	A principal in a member account	Member accounts
Baseline human policies (the roles that grant users permissions to perform their work)	Grant and restrict	Central team, such as platform, security, or IAM team[1]	Permissions for human roles[5]	Single account[4]	Federated principals[5] and IAM users[6]	Member accounts
Permissions boundaries (maximum permissions that an empowered	Restrict	Central team, such as platform, security, or IAM team[1]	Guardrails for workload performance roles	Single account[4]	The workload performance role principals in this account	Member accounts

Policy type	Effect	Managed by	Purpose	Pertains to	Affects	Deployed in
principal can assign to another principal)						
Machine role policies for applications (role attached to infrastructure deployed by delegated administrators)	Grant and restrict	Delegated to workload owner[8]	Permission for workload compute[9]	Single account	A principal in this account	Member accounts
Resource policies	Grant and restrict	Delegated to workload owner[8,10]	Permissions to resources	Single account	A principal in an account[11]	Member accounts

Notes from table:

1. Enterprises have many centralized teams (such as cloud platform, security operations, or identity and access management teams) that divide the responsibilities of these independent controls, and peer review one another's policies. The examples in the table are placeholders. You will need to determine the most effective separation of duties for your enterprise.
2. To use SCPs, you must enable all features within AWS Organizations.
3. Common baseline roles and policies are generally needed to enable automation, such as permissions for the pipeline, deployment tools, monitoring tools (for example, AWS Lambda and AWS Config Rules), and other permissions. This configuration is typically delivered when the account is provisioned.
4. Although these pertain to a resource (such as a role or a policy) in a single account, they can easily be replicated or deployed to multiple accounts by using AWS CloudFormation StackSets.
5. Define a core set of baseline human roles and policies that are deployed to all member accounts by a central team (often during account provisioning). Examples include the workload owners (delegated administrators), the platform team, the IAM team, and security audit teams.
6. Use identity federation (instead of local IAM users) whenever possible.
7. Permissions boundaries are used by delegated administrators. This IAM policy defines the maximum permissions and overrides other policies (including `"*:*"` policies that allow all actions on resources). Permissions boundaries should be required in baseline human policies as a condition to create roles (such as workload performance roles) and attach policies. Additional configurations such as SCPs enforce the attachment of the permissions boundary.
8. This assumes that sufficient guardrails (for example, SCPs and permissions boundaries) have been deployed.
9. These optional policies could be delivered during account provisioning or as part of the workload development process. The permission to create and attach these policies will be governed by the application developer's own permissions.
10. In addition to local account permissions, a centralized team (such as the cloud platform team or the security operations team) often manages resource-based policies to enable cross-account access to operate the accounts (for example, to provide access to S3 buckets for logging).
11. A resource-based IAM policy can enumerate any principal in any account. It can even enumerate anonymous principals (public access).

Ensuring that IAM identities have only those permissions that are necessary for a well-delineated set of tasks is critical for reducing the risk of malicious or unintentional abuse of permissions. Establishing and maintaining a least privilege model requires a deliberate plan to continually update, evaluate, and mitigate excess privilege. Here are some additional recommendations for that plan:

- Use your organization's governance model and established risk appetite to establish specific guardrails and permissions boundaries.
- Implement least privilege through a continually iterative process. This is not a one-time exercise.
- Use SCPs to reduce actionable risk. These are intended to be broad guardrails, not narrowly targeted controls.
- Use permissions boundaries to delegate IAM administration in a safer way.
  - Make sure that the delegated administrators attach the appropriate IAM boundary policy to the roles and users they create.
- As a defense-in-depth approach (in conjunction with identity-based policies), use resource-based IAM policies to deny broad access to resources.
- Use IAM access advisor, AWS CloudTrail, AWS IAM Access Analyzer, and related tooling to regularly analyze historical usage and permissions granted. Immediately remediate obvious over-permissions.
- Scope broad actions to specific resources where applicable instead of using an asterisk as a wildcard to indicate all resources.
- Implement a mechanism to quickly identify, review, and approve IAM policy exceptions based upon requests.

Code repository for AWS SRA examples

This document is accompanied by a GitHub repository at https://github.com/aws-samples/aws-security-reference-architecture-examples. This repository contains templates and deployable examples that illustrate some of the patterns presented previously in this document. The AWS services and infrastructure deployed in these templates are deliberately least privilege, and are intended for you to tailor and extend to fit the needs of your environment.

The initial set of solutions are built by using AWS CloudFormation and Python scripts. Deployment configurations are based on the customizations for AWS Control Tower solution, the AWS Landing Zone solution, and AWS CloudFormation StackSets. The AWS Control Tower and AWS Landing Zone solutions help customers quickly set up a secure, multi-account AWS environment based on AWS best practices. These solutions help save time by automating the setup of an environment for running secure and scalable workloads while implementing an initial security baseline through the creation of accounts and resources. They also provide a baseline environment to get started with a multi-account architecture, identity and access management, governance, data security, network design, and logging. The solutions in the AWS SRA repository provide configurations to implement the patterns described in this document.

Here is a summary of the initial solutions in the AWS SRA repository. Each solution folder includes a README.md file with details.

- The **Organization CloudTrail** solution creates an organization trail within the Org Management account. This trail is encrypted with a customer master key (CMK) that is managed in the Security Tooling account, and delivers logs to an S3 bucket in the Log Archive account. Optionally, data events can be enabled for Amazon S3 and Lambda functions. An organization trail logs events for all AWS accounts in the AWS organization while preventing member accounts from modifying the configurations.
- The **AWS Config Aggregator Account** solution enables an AWS Config aggregator in a specified account and creates authorizations within each member account. This solution assumes that AWS Config has already been enabled in each member account of the AWS organization. The solution includes a scheduled Lambda function that checks each day for any new member accounts that have been added to the AWS organization. If so, the solution adds them to the aggregator.
- The **Organization AWS Config Conformance Pack** solution deploys AWS Config Rules by delegating administration to a member account within the AWS organization. It then creates an organization conformance pack within the delegated administrator account for all existing and future accounts in the AWS organization. This solution deploys the following sample templates: AWS Control Tower Detective Guardrails Conformance Pack and Operational Best Practices for Encryption and Key Management.
- The **Organization GuardDuty** solution enables Amazon GuardDuty by delegating administration to a member account within the AWS organization. It configures GuardDuty within the delegated administrator account for all existing and future AWS organization accounts. The GuardDuty findings are also encrypted with a KMS key and sent to an S3 bucket in the Log Archive account.
- The **Organization Macie** solution enables Amazon Macie by delegating administration to a member account within the AWS organization. It configures Macie within the delegated administrator account for all existing and future AWS organization accounts. Macie is further configured to send its discovery results to a central S3 bucket that is encrypted with a KMS key.
- The **SecurityHub Enabler** solution enables AWS Security Hub within each AWS organization account and AWS Region that is configured with the Security Tooling account as the administrator account. The solution also provides optional configurations to enable security standards and third-party partner integrations. Centralizing Security Hub within the Security Tooling account provides a cross-account

view of security standards compliance and findings from both AWS services and third-party partner integrations.

- The **Organization AWS Firewall Manager** solution configures AWS Firewall Manager security policies by delegating administration to the Security Tooling account and configuring Firewall Manager with a security group policy and multiple AWS WAF policies. The security group policy requires a maximum allowed security group within a VPC (existing or created by the solution), which is deployed by the solution.

Contributors

Contributors to this document include:

- Brian Andrzejewski, AWS Senior Consultant
- Scott Conklin, AWS Senior Consultant
- Josh Du Lac, AWS Principal Solutions Architect
- Michael Haken, AWS Principal Technologist
- Jorg Huser, AWS Principal Consultant
- Mehial Mendrin, AWS Senior Consultant
- Avik Mukherjee, AWS Senior Consultant
- Eric Rose, AWS Principal Consultant
- Neal Rothleder, AWS Principal Consultant
- Handan Selamoglu, AWS Senior Technical Writer
- Andy Wickersham, AWS Consultant

Appendix: AWS security, identity, and compliance services

As an introduction or a refresher, see the Security, Identity, and Compliance on AWS webpage for a list of the AWS services that are aligned with the security perspective of the AWS Cloud Adoption Framework (AWS CAF). Within this perspective, the AWS CAF outlines five core programmatic epics for properly managing AWS Cloud security: identity and access management, detective controls, infrastructure security, data protection, and incident response. In addition to the epics, AWS offers services that help you determine your compliance status.

Identity and access management – AWS identity services enable you to securely manage identities, resources, and permissions at scale.

- IAM – Securely control access to AWS services and resources.
- AWS SSO – Centrally manage SSO access to multiple AWS accounts and business applications.
- Amazon Cognito – Add user sign-up, sign-in, and access control to your web and mobile applications.
- AWS Directory Service – Use managed Microsoft Active Directory in the AWS Cloud.
- AWS Resource Access Manager – Share AWS resources simply and securely.
- AWS Organizations – Implement policy-based management for multiple AWS accounts.

Detective controls – AWS monitoring and detection services provide guidance to help identify potential security incidents within your AWS environment.

- AWS Security Hub – View and manage security alerts and automate compliance checks from a central location.
- Amazon GuardDuty – Protect your AWS accounts and workloads with intelligent threat detection and continuous monitoring.
- Amazon Inspector – Automate security assessments to help improve the security and compliance of your applications that are deployed on AWS.
- AWS Config – Record and evaluate the configurations of your AWS resources to enable compliance auditing, resource change tracking, and security analysis.
- AWS CloudTrail – Track user activity and API usage to enable governance, compliance, and operational and risk auditing of your AWS account.

Infrastructure security – AWS network and application protection services enable you to enforce fine-grained security policy at network control points across your AWS organization.

- AWS Shield – Safeguard your web applications running on AWS with managed DDoS protection.
- AWS WAF – Protect your web applications from common web exploits and ensure availability and security.
- AWS Firewall Manager – Configure and manage AWS WAF rules across AWS accounts and applications from a central location.
- AWS Systems Manager – Easily configure and manage Amazon EC2 and on-premises systems to apply OS patches, create secure system images, and configure secure operating systems.
- Amazon VPC – Provision a logically isolated section of AWS where you can launch AWS resources in a virtual network that you define.
- AWS Network Firewall – Easily deploy essential network protections for all your VPCs.

Data protection – AWS provides services that help you protect your data, accounts, and workloads from unauthorized access.

- Amazon Macie – Discover, classify, and protect sensitive data with machine learning-powered security features.
- AWS KMS – Easily create and control the keys used to encrypt your data.
- AWS CloudHSM – Manage your hardware security modules (HSMs) in the AWS Cloud.
- AWS Certificate Manager – Easily provision, manage, and deploy SSL/TLS certificates for use with AWS services.
- AWS Secrets Manager – Easily rotate, manage, and retrieve database credentials, API keys, and other secrets through their lifecycle.

Incident response – AWS identifies threats by continuously monitoring the network activity and account behavior within your cloud environment.

- Detective – Analyze and visualize security data to rapidly get to the root cause of potential security issues.
- AWS Config Rules – Create rules that automatically take action in response to changes in your environment, such as isolating resources, enriching events with additional data, or restoring configuration to a known good state.
- AWS Lambda – Run code without provisioning or managing servers so you can scale your programmed, automated response to incidents.

Compliance – AWS gives you a comprehensive view of your compliance status and continuously monitors your environment by using automated compliance checks based on the AWS best practices and industry standards your business follows.

- AWS Artifact – No-cost, self-service portal that provides on-demand access to AWS security and compliance reports and select online agreements.
- AWS Audit Manager – Continuously audits your AWS usage to simplify how you assess risk and compliance with regulations and industry standards.

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.

update-history-change	update-history-description	update-history-date
— (p. 50)	Initial publication. This version doesn't include several AWS services (such as AWS Directory Service, Amazon Cognito, AWS Resource Access Manager, and AWS Audit Manager), which we plan to add in future versions.	June 23, 2021