
AWS Security Reference Architecture

AWS Professional Services team
[Last update](#): July 2022



Contents

Introduction	3
The value of the AWS SRA	5
How to use the AWS SRA	6
Key implementation guidelines of the AWS SRA	7
Security foundations	10
Security capabilities	11
Security design principles	11
SRA building blocks - AWS Organizations, accounts, and guardrails	13
Using AWS Organizations for security	13
The management account, trusted access, and delegated administrators	15
Dedicated accounts structure	16
AWS organization and account structure of the AWS SRA	17
Apply security services across your AWS organization	19
The AWS Security Reference Architecture	26
Org Management account	29
Security OU - Security Tooling account	35
Security OU - Log Archive account	49
Infrastructure OU – Network account	52
Infrastructure OU - Shared Services account	62
Workloads OU - Application account	65
IAM resources	75
Code repository for AWS SRA examples	79
Acknowledgments	82
Primary authors	82
Contributors	82
Appendix: AWS security, identity, and compliance services	83
Document history	85

Introduction

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 recommended 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 Documentation 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](#) were designed for the specific architecture illustrated in this reference. Each customer will have different requirements. As a result, the design of your AWS environment might 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 is 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](#).

Although we rely on a one-page diagram as our foundation, the 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 six sections and an appendix:

- [The value of the AWS SRA](#) discusses the motivation for building the AWS SRA, describes how you can use it to help improve your security, and lists key takeaways.
- [Security foundations](#) 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](#) 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](#) is a single-page architecture diagram that shows functional AWS accounts, and the security services and features that are generally available.
- [IAM resources](#) 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](#) provides an overview of the associated [GitHub repository](#) that contains example AWS CloudFormation templates and code for deploying some of the patterns discussed in the AWS SRA.

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

The value of the AWS SRA

AWS has a large (and growing) [set of security and security-related services](#). Customers have expressed appreciation for the detailed information available through our service documentation, blog posts, tutorials, summits, and conferences. They also informed us that they want to better understand the big picture and get a strategic view of AWS security services. When we worked with customers to get a deeper appreciation for what they needed, three priorities emerged:

1. Customers want more information and recommended patterns for how they can deploy, configure, and operate the AWS security services holistically. In which accounts and toward which security objectives should the services be deployed and managed? Is there one security account where all or most services should operate? How does the choice of location (organizational unit or AWS account) inform security objectives? Which trade-offs (design considerations) should customers be aware of?
2. Customers are interested in seeing different perspectives for logically organizing the many AWS security services. Beyond the primary function of each service (for example, identity services or logging services), these alternate viewpoints help customers plan, design, and implement their security architecture. An example shared later in this document groups the services based on the layers of protection aligned to the recommended structure of your AWS environment.
3. Customers are looking for guidance and examples to integrate security services in the most effective way. For example, how should they best align and connect AWS Config with other services to do the heavy lifting in automated audit and monitoring pipelines? Customers are asking for guidance on how each AWS security service relies on, or supports, other security services.

We address each of these in the AWS SRA. The first priority in the list (where things go) is the focus of the main architecture diagram and the accompanying discussions in this document. We provide a recommended AWS Organizations architecture and an account-by-account description of which services go where. To get started with the second priority in the list (how to think about the full set of security services), read the section, [Apply security services across your AWS organization](#). This section describes a way to group security services according to the structure of the elements in your AWS organization. In addition, those same ideas are reflected in the discussion of the [Application account](#), which highlights how security services can be operated to focus on certain layers of the account: Amazon Elastic Compute Cloud (Amazon EC2) instances, Amazon Virtual Private Cloud (Amazon VPC) networks, and the broader account. Finally, the third priority (service integration) is reflected throughout the guidance—particularly in the discussion of individual services in the account deep-dives sections of this documentation and the code in the AWS SRA code repository.

How to use the AWS SRA

There are different ways to use the AWS SRA depending on where you are in your cloud adoption journey. Here is a list of ways to gain the most insight from the AWS SRA assets (architecture diagram, written guidance, and code samples).

- *Define* the target state for your own security architecture.

Whether you are just starting your AWS Cloud journey—setting up your first set of accounts—or planning to enhance an established AWS environment, the AWS SRA is the place to start building your security architecture. Begin with a comprehensive foundation of account structure and security services, and then adjust based on your particular technology stack, skills, security objectives, and compliance requirements. If you know you will be building and launching more workloads, you can take your customized version of the AWS SRA and use it as the basis for your organization's security reference architecture.

- *Review* (and revise) the designs and capabilities that you have already implemented.

If you already have a security design and implementation, it is worth taking some time to compare what you have to the AWS SRA. The AWS SRA is designed to be comprehensive and provides a diagnostic baseline for reviewing your own security. Where your security designs align to the AWS SRA, you can feel more confident that you are following best practices when using AWS services. If your security designs diverge or even disagree with the guidance in the AWS SRA, this isn't necessarily a sign that you're doing something wrong. Instead, this observation provides you with the opportunity to review your decision process. There are legitimate business and technology reasons why you might deviate from the AWS SRA best practices. Perhaps your particular compliance, regulatory, or organization security requirements necessitate specific service configurations. Or, instead of using AWS services, you might have a feature preference for a product from the AWS Partner Network or a custom application that you built and manage. Sometimes, during this review, you might discover that your previous decisions were made based on older technology, AWS features, or business constraints that no longer apply. This is a good opportunity to review, prioritize any updates, and add them to the appropriate place of your engineering backlog. Whatever you discover as you assess your security architecture in light of the AWS SRA, you will find it valuable to document that analysis. Having that historical record of decisions and their justifications can help inform and prioritize future decisions.

- *Bootstrap* the implementation of your own security architecture.

The AWS SRA infrastructure as code (IaC) modules provide a fast, reliable way to start building and implementing your security architecture. These modules are described more deeply in the [code repository](#) section and in the [public GitHub repository](#). They not only enable engineers to build upon high-quality examples of the patterns in the AWS SRA guidance, but they also incorporate recommended security controls such as AWS Identity and Access Management (IAM) password policies, Amazon Simple Storage Service (Amazon S3) block account public

access, Amazon EC2 default Amazon Elastic Block Store (Amazon EBS) encryption, and integration with AWS Control Tower so that the controls are applied or removed as new AWS accounts are onboarded or decommissioned.

- *Learn* more about AWS security services and capabilities.

The guidance and discussions in the AWS SRA include important features as well as deployment and management considerations for individual AWS security and security-related services. One feature of the AWS SRA is that it provides a high-level introduction to the breadth of the AWS security services and how they work together in a multi-account environment. This complements the deep dive into the features and configuration for each service found in other sources. One example of this is the [discussion](#) of how AWS Security Hub ingests security findings from a variety of AWS services, AWS Partner products, and even your own applications.

- *Drive* a discussion of organizational governance and responsibilities for security.

An important element of designing and implementing any security architecture or strategy is understanding who in your organization has which security-related responsibilities. For example, the question of where to aggregate and monitor security findings is tied to the question of which team will be responsible for that activity. Are all findings across the organization monitored by a central team that needs access to a dedicated Security Tooling account? Or are individual application teams (or business units) responsible for certain monitoring activities and therefore need access to certain alerting and monitoring tools? As another example, if your organization has a group that manages all encryption keys centrally, that will influence who has permission to create AWS Key Management Service (AWS KMS) keys and which accounts those keys will be managed in. Understanding the characteristics of your organization—the various teams and responsibilities—will help you tailor the AWS SRA to best fit your needs. Conversely, sometimes the discussion of the security architecture becomes the impetus for discussing the existing organizational responsibilities and considering potential changes. AWS recommends a decentralized decision-making process where workload teams are responsible for defining the security controls based on their workload functions and requirements. The goal of centralized security and governance team is to build a system that allows the workload owners to make informed decisions and for all parties to get visibility of configuration, findings, and events. The AWS SRA can be a vehicle for identifying and informing these discussions.

Key implementation guidelines of the AWS SRA

Here are eight key takeaways from the AWS SRA to keep in mind as you design and implement your security.

- AWS Organizations and an appropriate multi-account strategy are necessary elements of your security architecture. Properly separating workloads, teams, and functions provides the foundations for separation of duties and defense-in-depth strategies. The guide covers this further in a [later section](#).

- Defense-in-depth is an important design consideration for selecting security controls for your organization. It helps you inject the appropriate security controls at different layers of the AWS Organizations structure, which helps minimize the impact of an issue: If there is an issue with one layer, there are controls in place that isolate other valuable IT resources. The AWS SRA demonstrates how different AWS services function at different layers of the AWS technology stack, and how using those services in combination helps you achieve defense-in-depth. This defense-in-depth concept on AWS is further discussed in a [later section](#) with design examples shown under [Application account](#).
- Use the wide variety of security building blocks across multiple AWS services and features to build a robust and resilient cloud infrastructure. When tailoring the AWS SRA to your particular needs, consider not only the primary function of AWS services and features (for example, authentication, encryption, monitoring, permission policy) but also how they fit into the structure of your architecture. A [later section](#) in the guide describes how some services operate across your entire AWS organization. Other services operate best within a single account, and some are designed to grant or deny permission to individual principals. Considering both of these perspectives helps you build a more flexible, layered security approach.
- Where possible (as detailed in later sections), make use of AWS services that can be deployed in every account (distributed instead of centralized) and build a consistent set of shared guardrails that can help protect your workloads from misuse and help reduce the impact of security events. The AWS SRA uses AWS Security Hub (centralized finding monitoring and compliance checks), Amazon GuardDuty (threat detection and anomaly detection), AWS Config (resource monitoring and change detection), IAM Access Analyzer (resource access monitoring), AWS CloudTrail (logging service API activity across your environment) and Amazon Macie (data classification) as a base set of AWS services to be deployed across every AWS account.
- Make use of the delegated administration feature of AWS Organizations, where it is supported, as explained later in the [delegated administration](#) section of the guide. This enables you to register an AWS member account as an administrator for supported services. Delegated administration provides flexibility for different teams within your enterprise to use separate accounts, as appropriate for their responsibilities, to manage AWS services across the environment. In addition, using a delegated administrator helps you limit access to, and manage the permissions overhead of, the AWS Organizations management account.
- Implement centralized monitoring, management, and governance across your AWS organizations. By using AWS services that support multi-account (and sometimes multi-Region) aggregation, along with delegated administration features, you empower your central security, network, and cloud engineering teams to have broad visibility and control over appropriate security configuration and data collection. Additionally, the data can be provided back to workload teams to empower them to make effective security decisions earlier in the software development lifecycle (SDLC).
- Use AWS Control Tower to set up and govern your multi-account AWS environment with the implementation of pre-built security controls to bootstrap your security reference architecture build. AWS Control Tower provides a blueprint to provide identity management, federated access to accounts, centralized logging, and defined workflows for provisioning additional accounts. You can then use the [Customizations for Control Tower \(CfCT\)](#) solution

to baseline the accounts managed by AWS Control Tower with additional security controls, service configurations, and governance, as demonstrated by the AWS SRA code repository. The account factory feature automatically provisions new accounts with configurable templates based on approved account configuration to standardize accounts within your AWS Organizations. You can also extend the governance to an individual existing AWS account by enrolling it into an organizational unit (OU) that is already governed by AWS Control Tower.

- The AWS SRA code examples demonstrate how you can automate the implementation of patterns within the AWS SRA guide by using infrastructure as code (IaC). Codifying the patterns provides the ability to treat IaC similar to other applications in your organization where testing can be automated before deployments are done. IaC also helps ensure consistency and repeatability with deploying guardrails across multiple (for example, SDLC or Region-specific) environments. The SRA code examples use AWS Control Tower with Customizations for AWS Control Tower (CfCT) to accelerate incorporating IaC into an AWS environment.

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 help 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. There are additional Well-Architected Framework focus areas that provide more context for specific domains such as governance, serverless, AI/ML, and gaming. These are known as [AWS Well-Architected lenses](#).

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 foundational documents, two sets of concepts are particularly relevant to the design and understanding of the AWS SRA: security capabilities and security design principles.

Security capabilities

The security perspective of AWS CAF outlines nine capabilities that help you achieve the confidentiality, integrity, and availability of your data and cloud workloads.

- *Security governance* to develop and communicate security roles, responsibilities, policies, processes, and procedures across your organization's AWS environment.
- *Security assurance* to monitor, evaluate, manage, and improve the effectiveness of your security and privacy programs.
- *Identity and access management* to manage identities and permissions at scale.
- *Threat detection* to understand and identify potential security misconfigurations, threats, or unexpected behaviors.
- *Vulnerability management* to continuously identify, classify, remediate, and mitigate security vulnerabilities.
- *Infrastructure protection* to help validate that systems and services within your workloads are protected.
- *Data protection* to maintain visibility and control over data, and how it is accessed and used in your organization.
- *Application security* to help detect and address security vulnerabilities during the software development process.
- *Incident response* to reduce potential harm by effectively responding to security incidents.

Security design principles

The [security pillar](#) of the Well-Architected Framework captures a set of six design principles that turn specific security areas into practical guidance that can help you strengthen your workload security. Where the security capabilities frame the overall security strategy, these Well-Architected principles describe what you can 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, instance and compute services, 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.

SRA building blocks - AWS Organizations, accounts, and guardrails

AWS security services, their controls, and interactions are best employed on a foundation of [AWS multi-account strategy](#) and identity and access management guardrails. These guardrails set the ability 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 and permissions 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 can organize your workloads in separate accounts and group accounts within an organizational unit (OU) 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 robust boundaries and controls between workloads. Account-level separation, in combination with AWS Organizations, 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.

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 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 principals within an organization's account, with the exception of the management account (which is one reason not to run workloads in this account). When you attach an SCP to an OU, it is inherited by the child OUs and accounts under the OU. 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 example, if an SCP denies access to all of Amazon S3, a principal affected by the SCP will not have access to Amazon S3 even if they are explicitly granted access through an IAM policy. For detailed information about how IAM policies are evaluated, the role of SCPs, and how access is ultimately granted or denied, see [policy evaluation logic](#) in the IAM documentation.

[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*. 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 might be a fit for highly regulated and sensitive workloads. This approach requires you to explicitly allow every IAM service or action in the path from the AWS account to the OU.

- Design consideration: Ideally, you would use a combination of deny list and allow list strategies. Use the allow list to define the list of allowed AWS services approved to be used within an AWS organization and attach this SCP at the root of your AWS organization. If you have a different set of services allowed per your development environment, you would attach the respective SCPs at each OU. You can then use the deny list to define enterprise guardrails by explicitly denying specific IAM actions.

The management account, trusted access, and delegated administrators

The management account (also called the AWS Organization Management account or Org Management account) is unique and differentiated from every other account in AWS Organizations. 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 deploys 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 can 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](#) 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 IAM Identity Center (successor to AWS Single Sign-On), AWS Config, AWS Firewall Manager, Amazon GuardDuty, AWS IAM Access Analyzer, Amazon Macie, AWS Security Hub, Amazon Detective, AWS Audit Manager, Amazon Inspector, 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. 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 sensitivities 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.
Security Tooling	Security	Dedicated AWS accounts for operating broadly applicable security services (such as Amazon GuardDuty, AWS Security Hub, AWS Audit Manager, Amazon Detective,

Account	OU	Security role
		Amazon Inspector, 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 Identity Center 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 <i>workload accounts</i> .) 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.

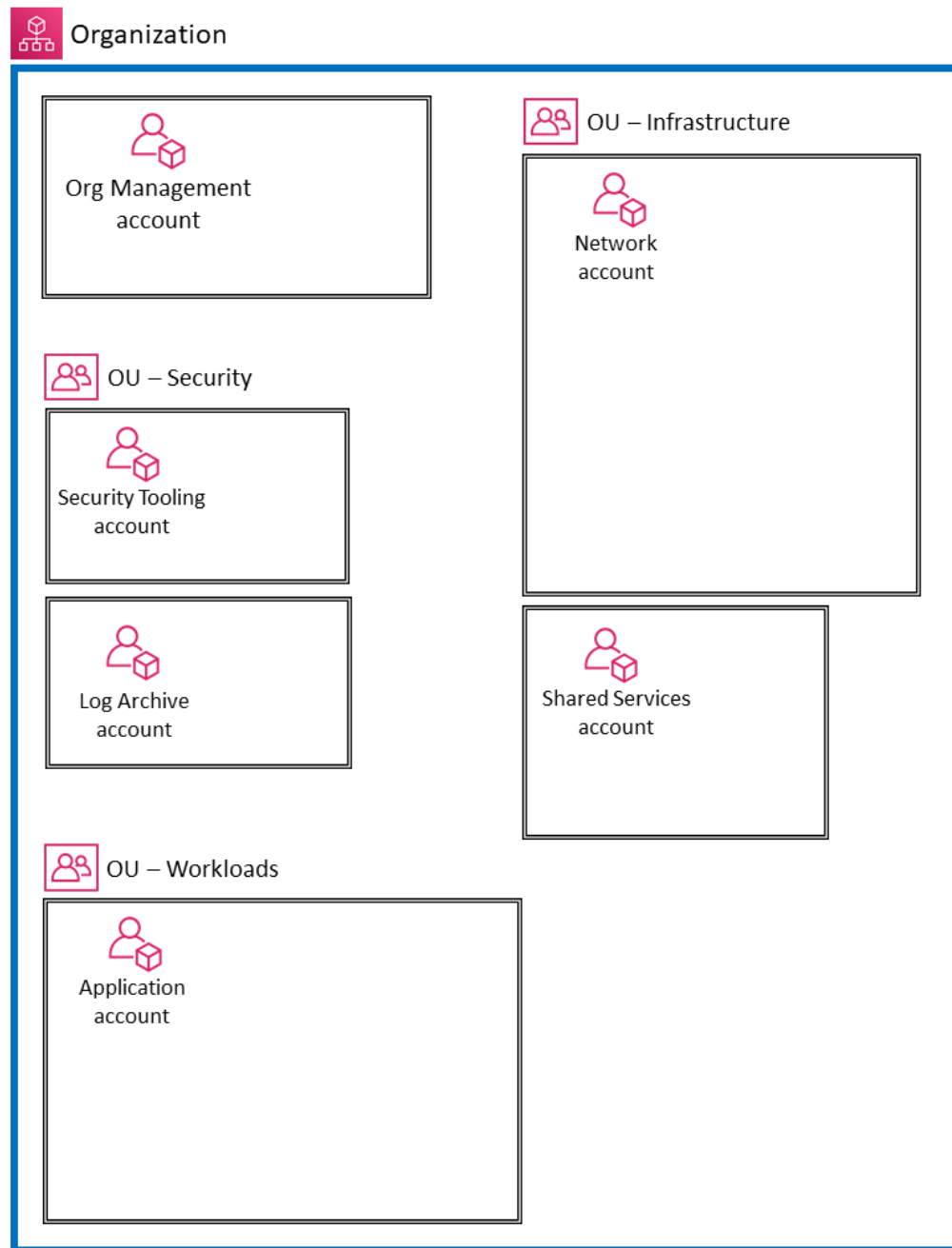
AWS organization and account structure of the AWS SRA

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 are considered production (prod) accounts that operate in a single AWS Region. When a regional service such as 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.

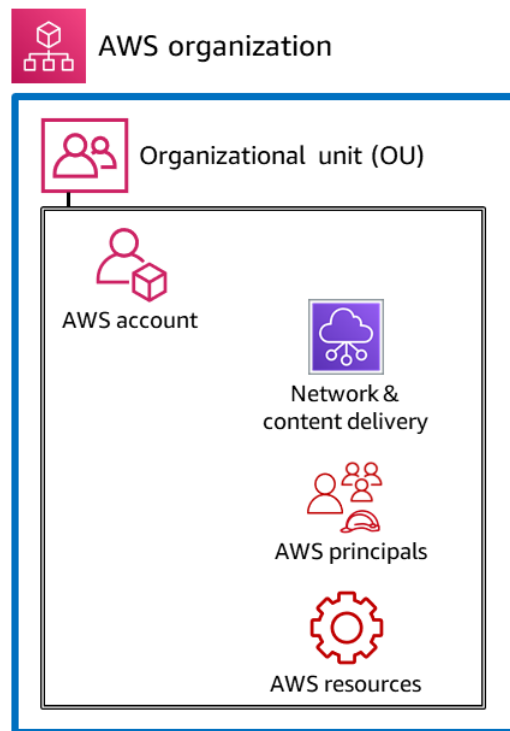
When hosting an AWS organization with a large set of accounts, it's beneficial to have an orchestration layer that facilitates account deployment and account governance. AWS Control Tower offers a straightforward way to set up and govern an AWS multi-account environment. The AWS SRA code samples in the [GitHub repository](#) demonstrate how you can use the [Customizations for AWS Control Tower](#) (CfCT) solution to deploy AWS SRA recommended structures.



Apply security services across your AWS organization

As described in a [previous section](#), customers are looking for an additional way to think about and strategically organize the full set of AWS security services. The most common organizational approach today is to group security services by primary function—according to what each service *does*. The security perspective of the AWS CAF lists nine functional capabilities, including identity and access management, infrastructure protection, data protection, and threat detection. Matching AWS services with these functional capabilities is a practical way to make implementation decisions in each area. For example, when looking at identity and access management, IAM and IAM Identity Center are services to consider. When architecting your threat detection approach, Amazon GuardDuty might be your first consideration.

As a complement to this functional view, you can also view your security with a cross-cutting, structural view. That is, in addition to asking, “Which AWS services should I use to control and protect my identities, logical access, or threat detection mechanisms?”, you can also ask, “Which AWS services should I apply across my entire AWS organization? What are the layers of defense I should put in place to protect the Amazon EC2 instances at the core of my application?” In this view, you map AWS services and features to layers in your AWS environment. Some services and features are a great fit for implementing controls across your full AWS organization. For example, blocking public access to Amazon S3 buckets is a specific control at this layer. It should preferably be done at the root organization instead of being part of the individual account setup. Other services and features are best used to help protect individual resources within an AWS account. Implementing a subordinate certificate authority (CA) within an account that requires private TLS certificates is an example of this category. Another equally important grouping consists of services that have an effect on the virtual network layer of your AWS infrastructure. The following diagram shows six layers in a typical AWS environment: AWS organization, organizational unit (OU), account, network infrastructure, principals, and resources.



Understanding the services in this structural context, including the controls and protections at each layer, helps you plan and implement a defense-in-depth strategy across your AWS environment. With this perspective, you can answer questions both from the top down (for example, “Which services am I using to implement security controls across my entire AWS organization?”) and from the bottom up (for example, “Which services manage controls on this EC2 instance?”). In this section, we walk through the elements of an AWS environment and identify associated security services and features. Of course, some AWS services have broad feature sets and support multiple security objectives. These services might support multiple elements of your AWS environment.

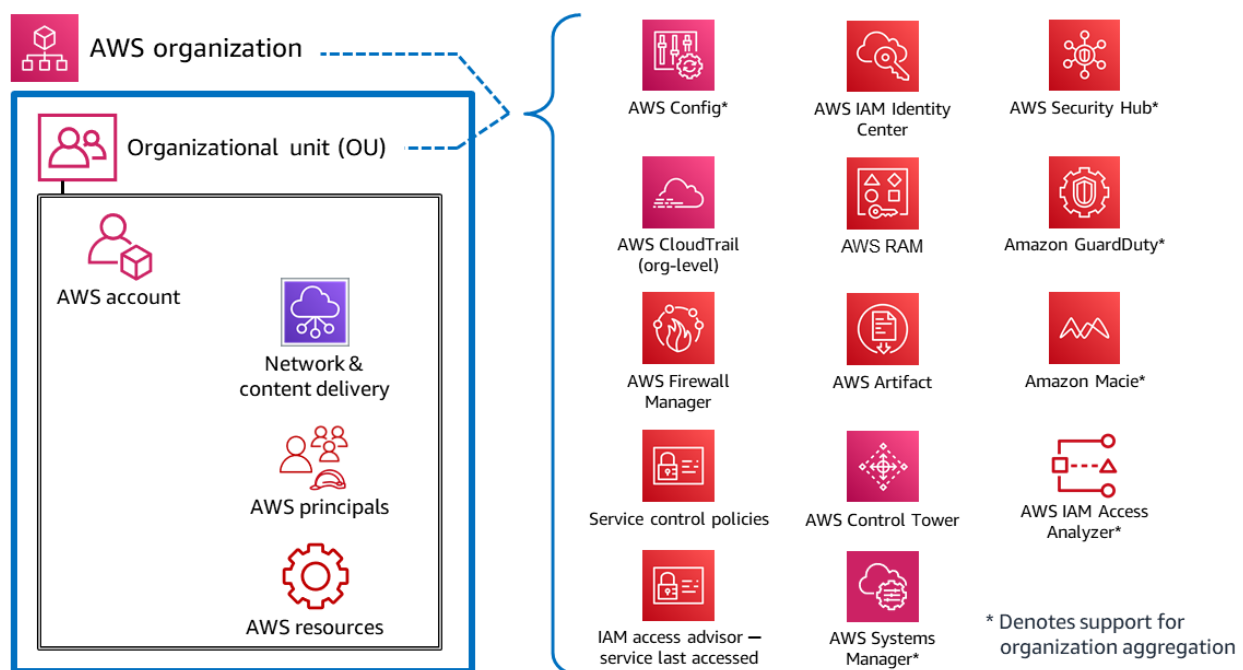
For clarity, we provide brief descriptions of how some of the services fit the stated objectives. The [next section](#) provides further discussion of the individual services within each AWS account.

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 provides a preventive, AWS organization-wide guardrail. Another example is AWS CloudTrail, which provides monitoring through an *organization trail* that logs 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, apply, and manage multiple resources across all accounts in your AWS organization: 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.

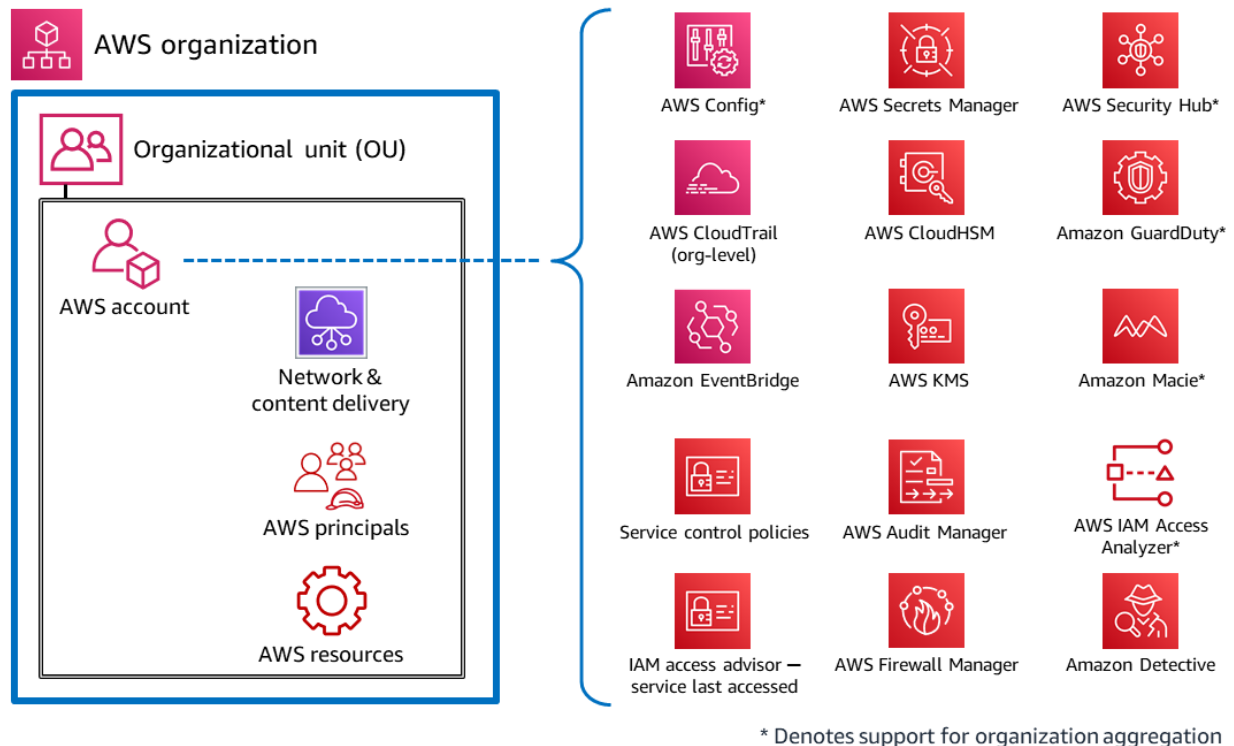
The services marked with an asterisk * in the following diagram operate with a dual scope: organization-wide and account-focused. These services fundamentally monitor or help control security within an individual account. However, they also support the ability to aggregate their results from multiple accounts into an organization-wide account for centralized visibility and management. For clarity, consider SCPs that apply across an entire OU, AWS account, or AWS organization. In contrast, you can configure and manage Amazon GuardDuty both at the account level (where individual findings are generated) and at the AWS organization level (by using the delegated administrator feature) where findings can be viewed and managed in aggregate.



AWS accounts

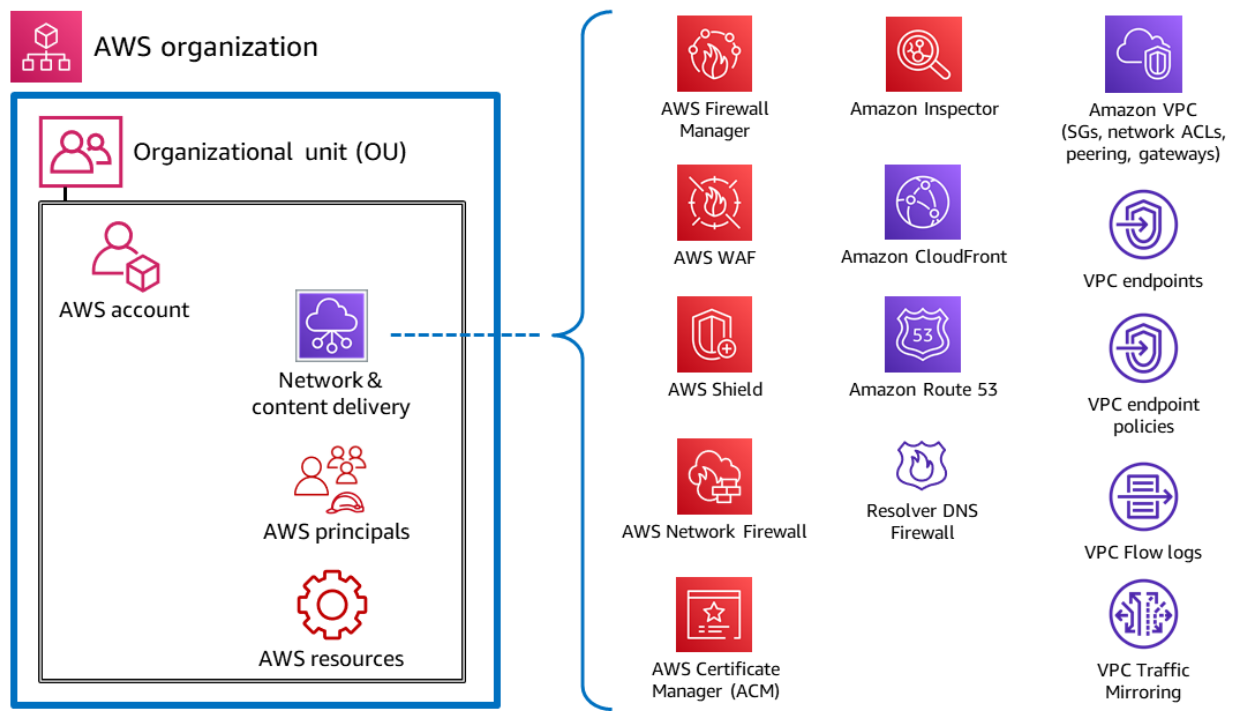
Within OUs, there are services that help protect multiple types of elements within an AWS account. For example, AWS Secrets Manager is often typically managed from a specific account and protects resources (such as database credentials or authentication information), applications, and AWS services in that account. AWS IAM Access Analyzer can be configured to generate findings when specified resources are accessible by principals outside the AWS account. As mentioned in the previous section, many of these services can also be configured and administered within AWS Organizations, so they can be managed across multiple accounts. These services are marked with an asterisk (*) in the diagram. They 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 security needs that are specific to their workload while also allowing governance and visibility to centralized security teams. Amazon GuardDuty is an example of such a service. GuardDuty monitors resources and activity associated with a single account, and 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, compute, and content delivery

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 is a vulnerability management service that continuously scans your AWS workloads for vulnerabilities. These scans include network reachability checks that indicate that there are allowed network paths to Amazon EC2 instances in your environment. [Amazon Virtual Private Cloud](#) (Amazon VPC) lets you define a virtual network into which you can launch AWS resources. This virtual network closely resembles a traditional network and includes a variety of features and benefits. VPC endpoints enable you to privately connect your VPC to supported AWS services and to the endpoint services powered by AWS PrivateLink without requiring a path to the internet. The following diagram illustrates security services that focus on network, compute, and content delivery infrastructure.



Principals and resources

AWS principals and AWS resources (along with IAM policies) are the fundamental elements in identity and access management on AWS. An authenticated principal in AWS can perform actions and access AWS resources. A principal can be authenticated as an AWS account root user and IAM user, or by assuming a role.

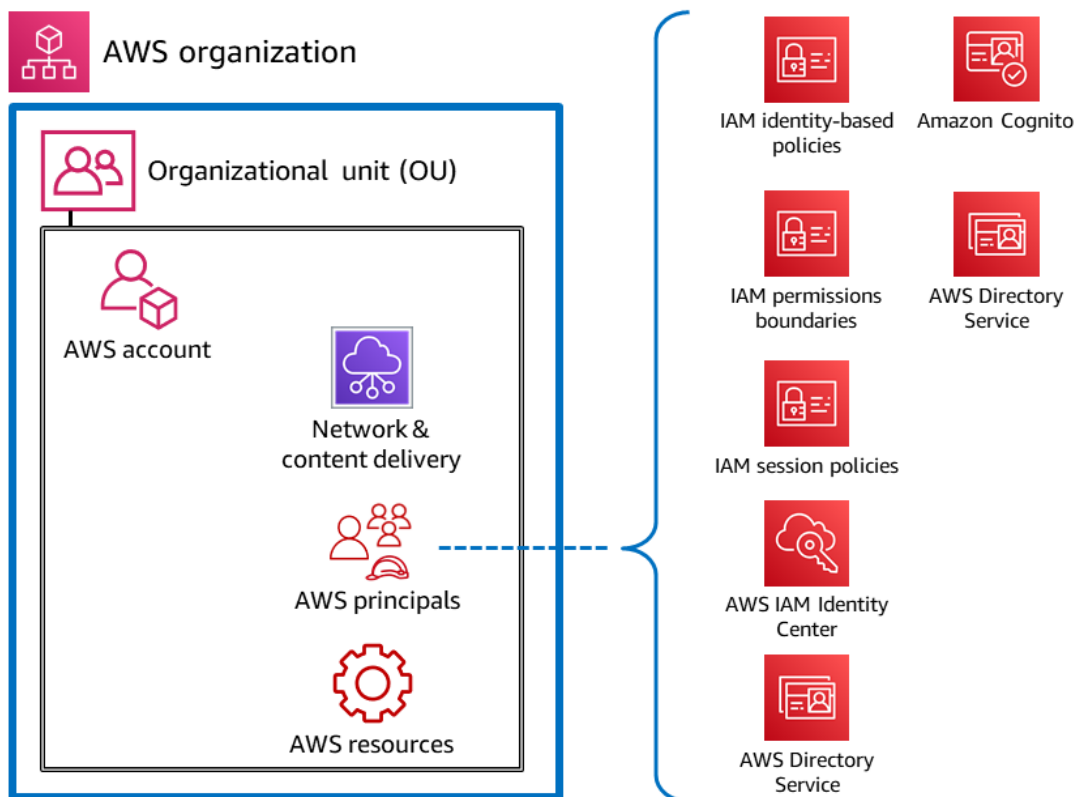
Note: Do not create persistent API keys associated with the AWS root user account. Access to the root user account should be limited only to the [tasks that require a root user](#), and then only through a rigorous exception process. For best practices to protect your account's root user, see the [AWS documentation](#).

An AWS resource is an object that exists within an AWS service that you can work with. Examples include an EC2 instance, an AWS CloudFormation stack, an Amazon Simple Notification Service (Amazon SNS) topic, and an S3 bucket. IAM policies are objects that define permissions when they are associated with an IAM principal (user, group, or role) or AWS resource. [Identity-based policies](#) are policy documents that you attach to a principal (roles, users, and groups of users) to control which actions a principal can perform, on which resources, and under which conditions. [Resource-based policies](#) are policy documents that you attach to a resource such as an S3 bucket. These policies grant the specified principal permission to perform specific actions on that resource and define the conditions for that permission. Resource-based

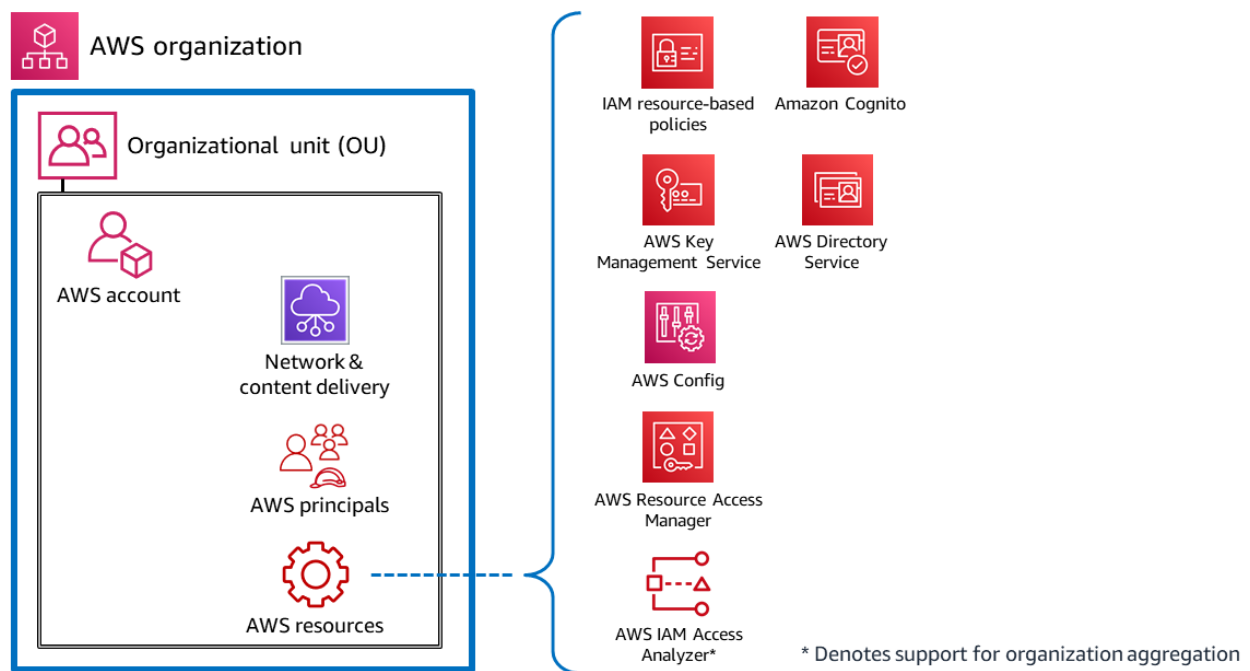
policies are in-line policies. The [IAM resources](#) section dives deeper into the types of IAM policies and how they are used.

To keep things simple in this discussion, we list AWS security services and features for IAM principals that have a primary purpose of operating on, or applying to, account principals. We keep that simplicity while acknowledging the flexibility and breadth of effects of IAM permission policies. A single statement in a policy can have effects on multiple types of AWS entities. For example, although an IAM identity-based policy is associated with an IAM principal and defines permissions (allow, deny) for that principal, the policy also implicitly defines permissions for the actions, resources, and conditions specified. In this way, an identity-based policy can be a critical element in defining permissions for a resource.

The following diagram illustrates AWS security services and features for AWS principals. Identity-based policies are attached to an IAM user, group, or role. These policies let you specify what that identity can do (its permissions). An IAM session policy is an [inline permissions policy](#) that users pass in the session when they assume the role. You can pass the policy yourself, or you can configure your identity broker to insert the policy when your [identities federate in to AWS](#). This enables your administrators to reduce the number of roles they have to create, because multiple users can assume the same role yet have unique session permissions. The IAM Identity Center service is integrated with AWS Organizations and AWS API operations, and helps you manage SSO access and user permissions across your AWS accounts in AWS Organizations.



The following diagram illustrates services and features for account resources. Resource-based policies are attached to a resource. For example, you can attach resource-based policies to S3 buckets, Amazon Simple Queue Service (Amazon SQS) queues, VPC endpoints, and AWS KMS encryption keys. You can use resource-based policies to specify who has access to the resource and what actions they can perform on it. S3 bucket policies, AWS KMS key policies, and VPC endpoint policies are types of resource-based policies. AWS IAM Access Analyzer helps you identify the resources in your organization and accounts, such as S3 buckets or IAM roles, that are shared with an external entity. This lets you identify unintended access to your resources and data, which is a security risk. AWS Config 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.

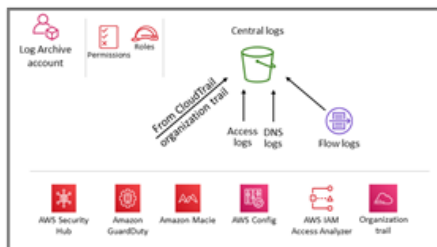


The AWS Security Reference Architecture

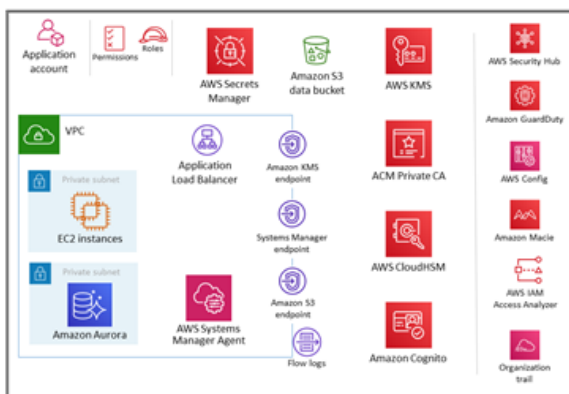
The following diagram illustrates the AWS SRA. This architectural diagram brings together all the AWS security-related services. It is built around a simple, three-tier web architecture that can fit on a single page. In such a workload, there is a *web tier* through which users connect and interact with 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 architecture is purposefully modular and provides high-level abstraction for many modern web applications.



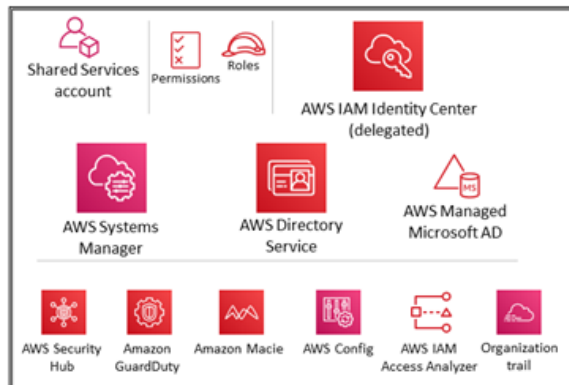
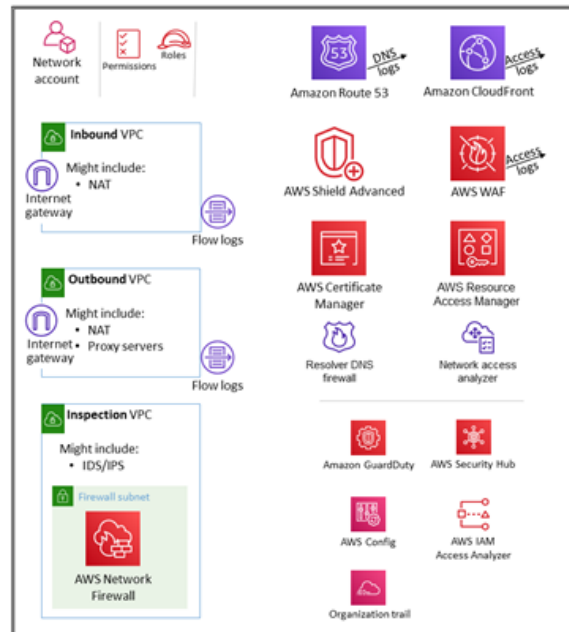
OU – Security



OU – Workloads



OU – Infrastructure



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.

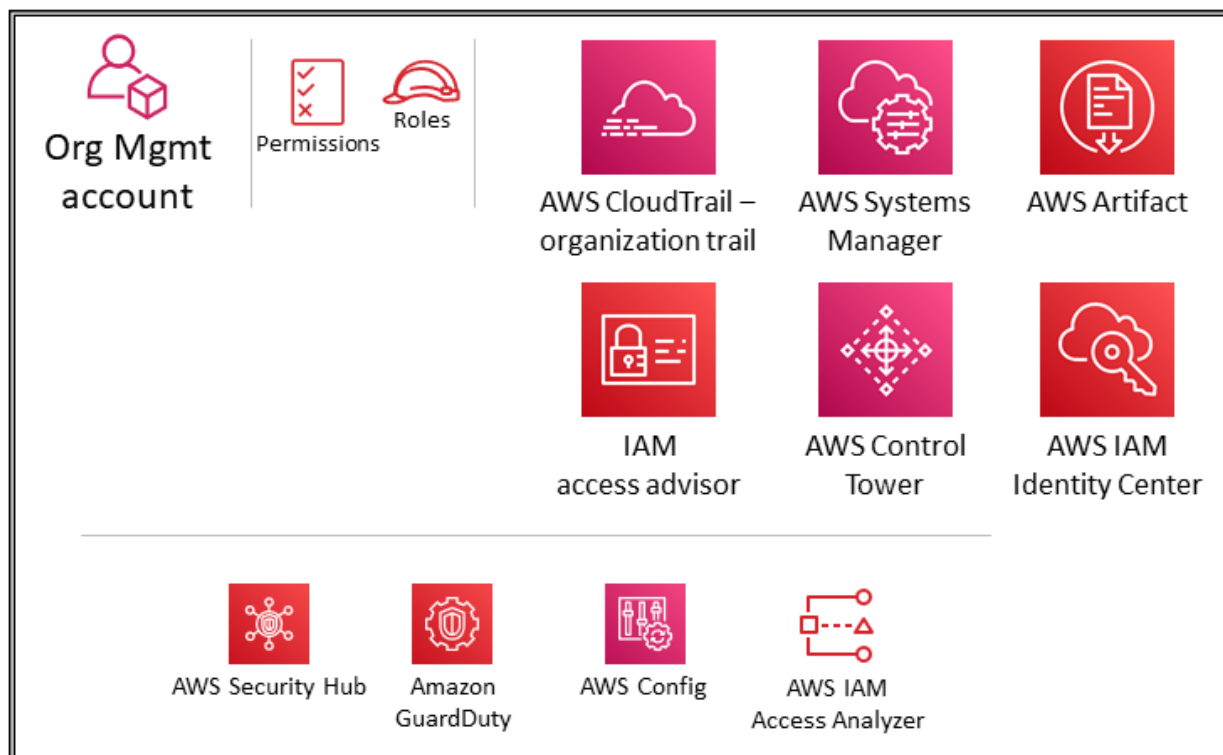
The AWS SRA contains all AWS security-related services available at the time of publication. (See [Document history](#).) However, not every workload or environment, based on its unique threat exposure, has to deploy every security service. Our goal is to provide a reference for a range of options, including descriptions of how these services fit together architecturally, so that your business can make decisions that are most appropriate for your infrastructure, workload, and security needs, based on risk.

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](#).
- 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.

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](#) and [The management account, trusted access, and delegated administrators](#) earlier in this guide discussed the purpose and security objectives of the Org Management account in depth. 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 all users, 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

With [AWS Organizations](#), you can centrally manage policies across multiple AWS accounts. For example, you can apply [service control policies](#) (SCPs) across multiple AWS accounts that are members of an organization. SCPs allow you to define which AWS service APIs can and cannot

be run by [AWS Identity and Access Management](#) (IAM) principals (such as IAM users and roles) in your organization's member AWS accounts. SCPs are created and applied from the Org Management account, which is the AWS account that you used when you created your organization. Read more about SCPs in the [Using AWS Organizations for security](#) section earlier in this reference.

If you use AWS Control Tower to manage your AWS organization, it will deploy [a set of SCPs as preventive guardrails](#) (categorized as mandatory, strongly recommended, or elective). These guardrails help you govern your resources by enforcing organization-wide security controls. These SCPs automatically use an `aws-control-tower` tag that has a value of `managed-by-control-tower`.

- Design consideration: SCPs affect only *member* accounts in the AWS organization. Although they are applied from the Org Management account, they have no effect on users or roles in that account. To learn about how SCP evaluation logic works, and to see examples of recommended structures, see the AWS blog post [How to Use Service Control Policies in AWS Organizations](#).

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](#) section. For additional security best practices, see the [AWS CloudTrail documentation](#).

In the AWS SRA, CloudTrail appears within the Org Management account, because you can create an organization trail only from within the management account and with the appropriate IAM permissions. The corresponding organization trail S3 bucket to store all the logs is created in the Log Archive account.

- 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.

IAM Identity Center

AWS IAM Identity Center (successor to AWS Single Sign-On) is an identity federation service that helps you centrally manage SSO access to all your AWS accounts, principals, and cloud workloads. IAM Identity Center also helps you manage access and permissions to commonly used third-party software as a service (SaaS) applications. Identity providers integrate with IAM Identity Center by using SAML 2.0. Bulk and just-in-time provisioning can be done by using the System for Cross-Domain Identity Management (SCIM). IAM Identity Center includes a user portal where your end-users can find and access their assigned AWS accounts, roles, cloud applications, and custom applications in one place.

IAM Identity Center natively integrates with AWS Organizations and runs in the Org Management account by default. However, to exercise least privilege and tightly control access to the management account, IAM Identity Center management can be delegated to a specific member account. In the AWS SRA, the Shared Services account is the delegated administrator account for IAM Identity Center. Before you enable delegated administration for IAM Identity Center, review [these considerations](#). You will find more information about delegation in the [Shared Services account](#) section. Even after you enable delegation, IAM Identity Center still needs to run in the Org Management account to perform certain [IAM Identity Center related tasks](#), which include managing permission sets that are provisioned in the Org Management account.

Within the IAM Identity Center console, accounts are displayed by their encapsulating OU. This enables you to quickly discover your AWS accounts, apply common sets of permissions, and manage access from a central location.

IAM Identity Center includes an identity store where specific user information must be stored. However, IAM Identity Center does not have to be the authoritative source for workforce information, although it can. In cases where your enterprise already has an authoritative source, IAM Identity Center supports the following types of identity providers (IdPs).

- **IAM Identity Center Identity store** – Choose this option if the following two options are not available. Users are created, group assignments are made, and permissions are assigned in the identity store. Even if your authoritative source is external to IAM Identity Center, a copy of principal attributes will be stored with the identity store.
- **Microsoft Active Directory (AD)** – Choose this option if you want to manage users either in your own on-premises or cloud-based Active Directory, or you want to migrate or use [AWS Managed Microsoft AD](#) in AWS Directory Service.
- **External IdP** – Choose this option if you prefer to manage users in an external, third-party IdP.

You can rely on an existing IdP that is already in place within your enterprise. 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 you with an opportunity to integrate with your human resources (HR) processes.

- Design consideration: Use an external IdP if that option is available to your enterprise. Take advantage of the SCIM capability in IAM Identity Center to automate user, group, and permission provisioning (synchronization). This allows AWS access to stay in sync with your corporate workflow for new hires, employees who are moving to another team, and employees who are leaving the company.

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.

IAM access within the Org Management account lets you view service last accessed data for the Org Management account, OU, member account, or IAM policy in your AWS organization. This information is available in the IAM console within the management account and can also be obtained programmatically by using IAM access advisor APIs in AWS Command Line Interface (AWS CLI) or a programmatic client. The information indicates which principals in an organization or account last attempted to access the service and when. 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.

AWS Systems Manager

Quick Setup and Explorer, which are capabilities of AWS Systems Manager, 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. 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.

[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](#) section later in this guide discusses the use of the Systems Manager Agent (SSM Agent) on the EC2 instances in the Application account.

AWS Control Tower

AWS Control Tower provides a straightforward way to set up and govern a secure, multi-account AWS environment, which is called a *landing zone*. AWS Control Tower creates your landing zone by using AWS Organizations, and provides ongoing account management and governance as well as implementation best practices. You can use AWS Control Tower to provision new accounts in a few steps while ensuring that the accounts conform to your organizational policies. You can even add existing accounts to a new AWS Control Tower environment.

AWS Control Tower has a broad and flexible set of features. A key feature is its ability to orchestrate the capabilities of several other [AWS services](#), including AWS Organizations, AWS Service Catalog, and IAM Identity Center, to build a landing zone. For examples, by default AWS Control Tower uses AWS CloudFormation to establish a baseline, AWS Organizations service control policies (SCPs) to prevent configuration changes, and AWS Config rules to continuously detect non-conformance. AWS Control Tower employs blueprints that help you quickly align your multi-account AWS environment with [AWS Well Architected security foundation design principles](#). Among governance features, AWS Control Tower offers guardrails that prevent deployment of resources that don't conform to selected policies.

You can get started implementing AWS SRA guidance with AWS Control Tower. For example, AWS Control Tower establishes an AWS organization with the recommended multi-account architecture. It provides blueprints to provide identity management, provide federated access to accounts, centralize logging, establish cross-account security audits, define a workflow for provisioning new accounts, and implement account baselines with network configurations.

In the AWS SRA, AWS Control Tower is within the Org Management account because AWS Control Tower uses this account to set up an AWS organization automatically and designates that account as the management account. This account is used for billing across your AWS organization. It's also used for Account Factory provisioning of accounts, to manage OUs, and to manage guardrails. If you are launching AWS Control Tower in an existing AWS organization, you can use the existing management account. AWS Control Tower will use that account as the designated management account.

- Design consideration: If you want to do additional baselining of controls and configurations across your accounts, you can use [Customizations for AWS Control Tower](#) (CfCT). With CfCT, you can customize your AWS Control Tower landing zone by using an AWS CloudFormation template and service control policies (SCPs). You can deploy the custom template and policies to individual accounts and OUs within your organization. CfCT integrates with AWS Control Tower lifecycle events to ensure that resource deployments stay in sync with your landing zone.

AWS Artifact

AWS Artifact provides on-demand access to AWS security and compliance reports and select online agreements. Reports available in AWS Artifact include System and Organization Controls (SOC) reports, Payment Card Industry (PCI) reports, and certifications from accreditation bodies across geographies and compliance verticals that validate the implementation and operating effectiveness of AWS security controls. AWS Artifact helps you perform your due diligence of AWS with enhanced transparency into our security control environment. It also lets you continuously monitor the security and compliance of AWS with immediate access to new reports.

AWS Artifact Agreements enable you to review, accept, and track the status of AWS agreements such as the Business Associate Addendum (BAA) for an individual account and for the accounts that are part of your organization in AWS Organizations.

You can provide the AWS audit artifacts to your auditors or regulators as evidence of AWS security controls. You can also use the responsibility guidance provided by some of the AWS audit artifacts to design your cloud architecture. This guidance helps determine the additional security controls you can put in place to support the specific use cases of your system.

AWS Artifacts is hosted in the Org Management account to provide a central location where you can review, accept, and manage agreements with AWS. This is because agreements that are accepted at the management account flow down to the member accounts.

- Design consideration: Users within the Org Management account should be restricted to use only the Agreements feature of AWS Artifact and nothing else. To implement segregation of duties, AWS Artifact is also hosted in the Security Tooling account where you can delegate permissions to your compliance stakeholders and external auditors to access audit artifacts. You can implement this separation by defining fine-grained IAM permission policies. For examples, see [Example IAM policies](#) in the AWS documentation.

Distributed and centralized security service guardrails

In the AWS SRA, AWS Security Hub, Amazon GuardDuty, AWS Config, IAM Access Analyzer, AWS CloudTrail organization trails, and often Amazon Macie are deployed with appropriate delegated administration or aggregation to the Security Tooling account. This enables a consistent set of guardrails across accounts and also provides centralized monitoring, management, and governance across your AWS organization. You will find this group of services in every type of account represented in the AWS SRA. These should be part of the AWS services that must be provisioned as part of your account onboarding and baselining process. The [GitHub code repository](#) provides a sample implementation of AWS security-focused services across your accounts, including the AWS Org Management account.

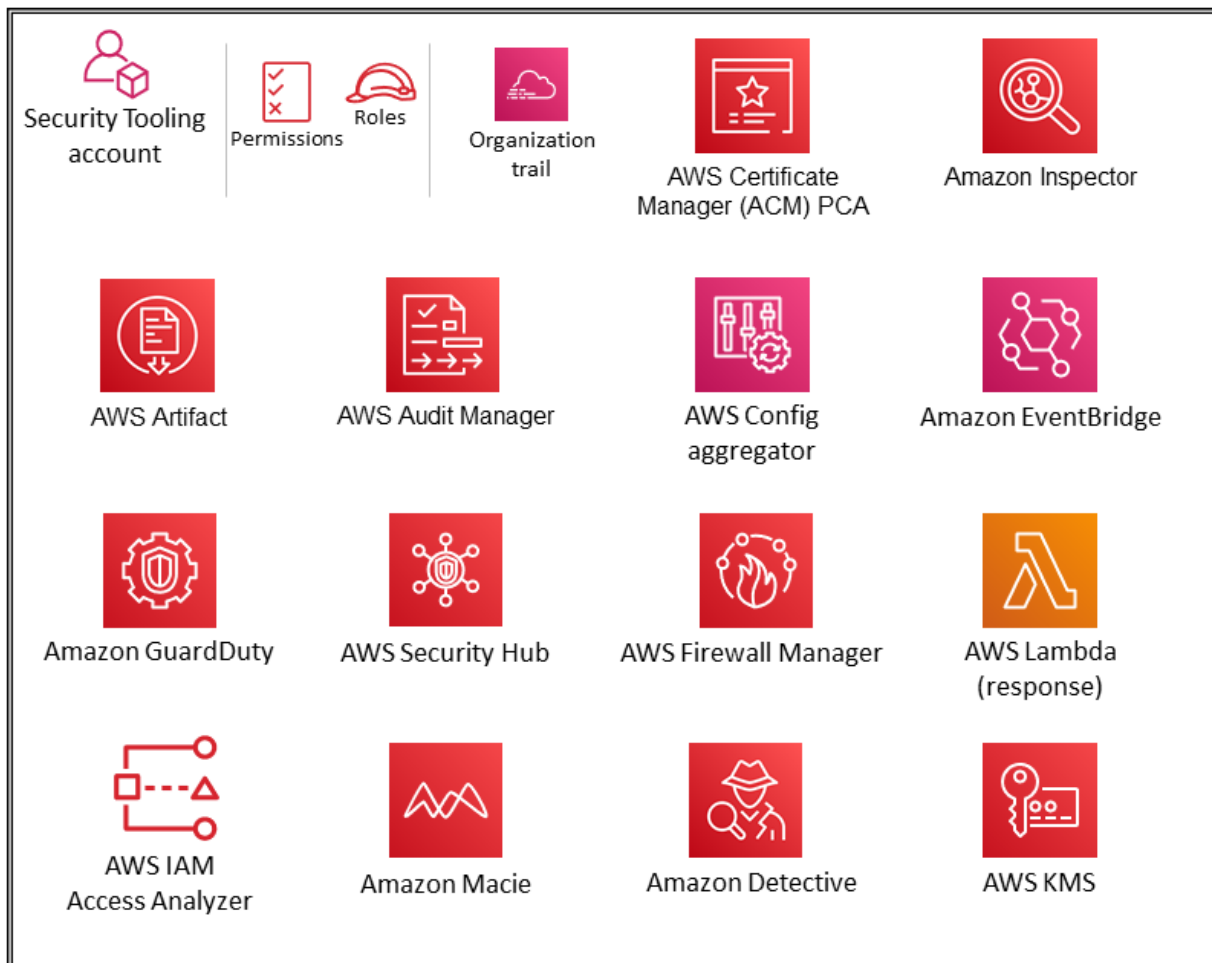
In addition to these services, AWS SRA includes two security-focused services, AWS Detective and AWS Audit Manager, which support the integration and delegated administrator

functionality in AWS Organizations. However, those are not included as part of the recommended services for account baselining. We have seen that these services are best used in the following scenarios:

- You have a dedicated team or group of resources that perform those digital forensics and IT audit functions. AWS Detective is best utilized by security analyst teams, and AWS Audit Manager is helpful to your internal audit or compliance teams.
- You want to focus on a core set of tools such as GuardDuty and Security Hub at the start of your project, and then build on these by using services that provide additional capabilities.

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 account 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 organization 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: AWS Control Tower names the account under the Security OU the *Audit Account* by default. You can rename the account during the AWS Control Tower setup.
 - Design consideration: It might be appropriate to have more than one Security Tooling account. For example, monitoring and responding to security events are 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. If you are using AWS Control Tower, it restricts the creation of additional AWS accounts under the Security OU.

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, AWS Detective, AWS Audit Manager, Amazon Inspector, and AWS Systems Manager. Your security team manages the security features of these services and monitors any security-specific events or findings.

IAM Identity Center supports delegated administration to a member account. AWS SRA uses the Shared Services account as the delegated administrator account for IAM Identity Center, as explained later in the [IAM Identity Center](#) section of the Shared Services account.

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 that you might 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. You can additionally designate an aggregation Region within the delegated administrator account to centralize your findings across your accounts and your linked Regions. Your findings are continuously and bidirectionally synced between the aggregator Region and all other Regions.

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.

You can use Security Hub with the [Network Access Analyzer](#) feature of Amazon VPC to help continuously monitor the compliance of your AWS network configuration. This will help you block unwanted network access and help prevent your critical resources from external access. For further architecture and implementation details, see the AWS blog post [Continuous verification of network compliance using Amazon VPC Network Access Analyzer and AWS Security Hub](#).

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 consideration: 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](#).
- Design consideration: 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](#) 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 must always capture and store appropriate logs for monitoring and audit purposes, but Amazon GuardDuty pulls independent streams of data directly from AWS CloudTrail, Amazon 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. For example, 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.

A [conformance pack](#) is a collection of AWS Config rules and remediation actions that can be 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](#).

AWS Config integrates with AWS Security Hub to send the results of AWS Config managed and custom rule evaluations as findings into Security Hub.

AWS Config rules can be used in conjunction with AWS Systems Manager to effectively remediate noncompliant resources. You use AWS Systems Manager Explorer to gather the compliance status of AWS Config rules in your AWS accounts across AWS Regions and then use [Systems Manager Automation documents \(runbooks\)](#) to resolve your noncompliant AWS Config rules. For implementation details, see the the blog post [Remediate noncompliant AWS Config rules with AWS Systems Manager Automation runbooks](#).

If you use AWS Control Tower to manage your AWS organization, it will deploy [a set of AWS Config rules as detective guardrails](#) (categorized as mandatory, strongly recommended, or elective). These guardrails help you govern your resources and monitor compliance across accounts in your AWS organization. These AWS Config rules will automatically use an `aws-control-tower` tag that has a value of `managed-by-control-tower`.

- Design consideration: AWS Config streams 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](#).

Amazon Macie

Amazon Macie is a fully managed data security and data privacy service that uses machine learning and pattern matching to discover and help 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 consideration: If S3 objects are encrypted with an AWS Key Management Service (AWS KMS) key that you manage, you can add the Macie service-linked role as a key user to that KMS key to enable Macie to scan the data.
- Design consideration: 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 can 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 Amazon 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 straightforward 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 can 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 consideration: 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 can be integrated into your investigative playbooks and runbooks to coordinate response activities.
- Design consideration: 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 straightforward to analyze, investigate, and quickly identify the root cause of security findings or suspicious activities for your security analysts. 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 Amazon 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.

Detective integrates with AWS Organizations. The Org Management account delegates a member account as the Detective administrator account. In the AWS SRA, this is the Security Tooling account. The Detective administrator account has the ability to automatically enable all current member accounts in the organization as detective member accounts, and also add new member accounts as they get added to the AWS organization. Detective administrator accounts also have the ability to invite member accounts that currently do not reside in the AWS organization, but are within the same Region, 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 AWS 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.

AWS Audit Manager

AWS Audit Manager helps you continually audit your AWS usage to simplify how you manage audits and compliance with regulations and industry standards. It enables you to move from manually collecting, reviewing, and managing evidence to a solution that automates evidence collection, provides a simple way to track the source of audit evidence, enables teamwork collaboration, and helps to manage evidence security and integrity. When it's time for an audit, Audit Manager helps you manage stakeholder reviews of your controls.

With Audit Manager you can audit against [prebuilt frameworks](#) such as the Center for Internet Security (CIS) benchmark, the CIS AWS Foundations Benchmark, System and Organization Controls 2 (SOC 2), and the Payment Card Industry Data Security Standard (PCI DSS). It also gives you the ability to create your own frameworks with standard or custom controls based on your specific requirements for internal audits.

Audit Manager collects four types of evidence. Three types of evidence are automated: compliance check evidence from AWS Config and AWS Security Hub, management events evidence from AWS CloudTrail, and configuration evidence from AWS service-to-service API calls. For evidence that cannot be automated, Audit Manager lets you upload manual evidence.

Note: Audit Manager assists in collecting evidence that's relevant for verifying compliance with specific compliance standards and regulations. However, it doesn't assess your compliance. Therefore, the evidence that's collected through Audit Manager might not include details of your operational processes that are needed for audits. Audit Manager isn't a substitute for legal counsel or compliance experts. We recommend that you engage the services of a third-party assessor who is certified for the compliance framework(s) that you are evaluated against.

Audit Manager assessments can run over multiple accounts in your AWS organizations. Audit Manager collects and consolidates evidence into a delegated administrator account in AWS Organizations. This audit functionality is primarily used by compliance and internal audit teams, and requires only read access to your AWS accounts.

- Design consideration: Audit Manager complements other AWS security services such as Security Hub and AWS Config to help implement a risk management framework. Audit Manager provides independent risk assurance functionality, whereas Security Hub helps you oversee your risk and AWS Config conformance packs assist in managing your risks. Audit professionals who are familiar with the [Three Lines Model](#) developed by the [Institute of Internal Auditors \(IIA\)](#) should note that this combination of AWS services helps you cover the three lines of defense. For more information, see the two-part [blog series](#) on the AWS Cloud Operations & Migrations blog.
- Design consideration: In order for Audit Manager to collect Security Hub evidence, the delegated administrator account for both services has to be the same AWS account. For this reason, in the AWS SRA, the Security Tooling account is the delegated administrator for Audit Manager.

AWS Artifact

AWS Artifact is hosted within the Security Tooling account to delegate the compliance artifact management functionality from the AWS Org Management account. This delegation is important because we recommend that you avoid using the AWS Org Management account for deployments unless absolutely necessary. Instead, delegate deployments to member accounts. Because audit artifact management can be done from a member account and the function closely aligns with the security and compliance team, the Security Tooling account is designated as the delegated administrator account for AWS Artifact. You can use AWS Artifact reports to download AWS security and compliance documents, such as AWS ISO certifications, Payment Card Industry (PCI), and System and Organization Controls (SOC) reports. You can restrict this capability to only AWS Identity and Access Management (IAM) roles pertaining to your audit and compliance teams, so they can download, review, and provide those reports to external auditors as needed. You can additionally restrict specific IAM roles to have access to only specific AWS Artifact reports through IAM policies. For sample IAM policies, see the [AWS Artifact documentation](#).

- Design consideration: If you choose to have a dedicated AWS account for audit and compliance teams, you can host AWS Artifact in a security audit account, which is separate from the Security Tooling account. AWS Artifact reports provide evidence that demonstrates that an organization is following a documented process or meeting a specific requirement. Audit artifacts are gathered and archived throughout the system development lifecycle and can be used as evidence in internal or external audits and assessments.

AWS KMS

AWS Key Management Service (AWS KMS) helps you create and manage cryptographic keys and control their use across a wide range of AWS services and in your applications. AWS KMS is a secure and resilient service that uses hardware security modules to protect cryptographic keys. It follows industry standard lifecycle processes for key material, such as storage, rotation, and access control of keys. AWS KMS can help protect your data with encryption and signing keys, and can be used for both server-side encryption and client-side encryption through the [AWS Encryption SDK](#). For protection and flexibility, AWS KMS supports three types of keys: customer managed keys, AWS managed keys, and AWS owned keys. Customer managed keys are AWS KMS keys in your AWS account that you create, own, and manage. AWS managed keys are AWS KMS keys in your account that are created, managed, and used on your behalf by an AWS service that is integrated with AWS KMS. AWS owned keys are a collection of AWS KMS keys 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 [AWS KMS Cryptographic Details](#).

One deployment option is to centralize the responsibility of KMS key management to a single account while delegating the ability to use keys in the Application account by application resources by using a combination of key and IAM policies. This approach is secure and straightforward to manage, but you can encounter hurdles due to AWS KMS throttling limits, account service limits, and the security team being inundated with operational key management tasks. Another deployment option is to have a decentralized model in which you allow AWS KMS to reside in multiple accounts, and you allow those responsible for the infrastructure and workloads in a specific account to manage their own keys. This model gives your workload teams more control, flexibility, and agility over the use of encryption keys. It also helps avoid API limits, limits the scope of impact to one AWS account only, and simplifies reporting, auditing, and other compliance-related tasks. In a decentralized model it is important to deploy and enforce guardrails so that the decentralized keys are managed in the same way and usage of KMS keys is audited according to established best practices and policies. For more information, see the whitepaper [AWS Key Management Service Best Practices](#). AWS SRA recommends a distributed key management model in which the KMS keys reside locally within the account where they are used. We recommend that you avoid using a single key in one account for all cryptographic functions. Keys can be created based on function and data protection requirements, and to enforce the principle of least privilege. In some cases, encryption permissions would be kept separate from decryption permissions, and administrators would manage lifecycle functions but would not be able to encrypt or decrypt data with the keys that they manage.

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.

ACM Private CA

AWS Certificate Manager Private Certificate Authority (ACM Private CA) is a managed private CA service that helps you securely manage the lifecycle of your private end-entity TLS certificates for EC2 instances, containers, IoT devices, and on-premises resources. It allows encrypted TLS communications to running applications. With ACM Private CA, you can create your own CA hierarchy (a root CA, through subordinate CAs, to end-entity certificates) and issue certificates with it to authenticate internal users, computers, applications, services, servers, and other devices, and to sign computer code. Certificates issued by a private CA are trusted only within your AWS organization, not on the internet.

A public key infrastructure (PKI) or security team can be responsible for managing all PKI infrastructure. This includes the management and creation of the private CA. However, there must be a provision that allows workload teams to self-serve their certificate requirements. The AWS SRA depicts a centralized CA hierarchy in which the root CA is hosted within the Security Tooling account. This enables security teams to enforce stringent security control, because the root CA is the foundation of the entire PKI. However, creation of private certificates from the private CA is delegated to application development teams by sharing out the CA to an application account by using AWS Resource Access Manager (AWS RAM). AWS RAM manages the permissions required for cross-account sharing. This removes the need for a private CA in every account and provides a more cost-effective way of deployment. For more information about the workflow and implementation, see the blog post [How to use AWS RAM to share your ACM Private CA cross-account](#).

Note: ACM also helps you provision, manage, and deploy public TLS certificates for use with AWS services. To support this functionality, ACM has to reside in the AWS account that would use the public certificate. This is discussed later in this guide, in the [Application account](#) section.

- Design consideration: With ACM Private CA, you can create a hierarchy of certificate authorities with up to five levels. You can also create multiple hierarchies, each with its own root. The ACM CA hierarchy should adhere to your organization's PKI design. However, keep in mind that increasing the CA hierarchy increases the number of certificates in the certification path, which, in turn, increases the validation time of an end-entity certificate. A well-defined CA hierarchy provides benefits that include granular security control appropriate to each CA, delegation of subordinate CA to a different application, which leads to division of administrative tasks, use of CA with limited revocable trust, the ability to define different validity periods, and the ability to enforce path limits. Ideally, your root and subordinate CAs are in separate AWS accounts. For more information about planning a CA hierarchy by using ACM Private CA, see the [ACM Private CA documentation](#) and the blog post [How to secure an enterprise scale ACM Private CA hierarchy for automotive and manufacturing](#).
- Design consideration: ACM Private CA can integrate with your existing CA hierarchy, which allows you to use the automation and native AWS integration capability of ACM in conjunction with the existing root of trust that you use today. You can create a

subordinate CA in ACM Private CA backed by a parent CA on premises. For more information about implementation, see [Installing a subordinate CA certificate signed by an external parent CA](#) in the ACM Private CA documentation.

Amazon Inspector

Amazon Inspector is an automated vulnerability management service that continually scans Amazon EC2 and container workloads for software vulnerabilities and unintended network exposure.

Amazon Inspector continuously assesses your environment throughout the lifecycle of your resources by automatically scanning resources whenever you make changes to them. Events that initiate rescanning a resource include installing a new package on an EC2 instance, installing a patch, and when a new common vulnerabilities and exposures (CVE) report that affects the resource is published. When an open vulnerability is detected, Amazon Inspector calculates an Amazon Inspector risk score by correlating up-to-date CVE information with temporal and environmental factors such as network accessibility and exploitability information to provide a contextual finding.

The network reachability findings of Amazon Inspector assess the accessibility of your EC2 instances to or from VPC edges such as internet gateways, VPC peering connections, or virtual private networks (VPNs) through a virtual gateway. These rules help automate the monitoring of your AWS networks and identify where network access to your EC2 instances might be misconfigured through mismanaged security groups, access control lists (ACLs), internet gateways, and so on. For more information, see the [Amazon Inspector documentation](#).

AWS Systems Manager Agents (SSM Agents) are required for vulnerability scanning of EC2 instances. No agents are required for network reachability of EC2 instances or vulnerability scanning of container images in Amazon Elastic Container Registry (Amazon ECR).

Amazon Inspector is integrated with AWS Organizations and supports delegated administration. In the AWS SRA, the Security Tooling account is made the delegated administrator account for Amazon Inspector.

- Design consideration: Amazon Inspector integrates with AWS Security Hub automatically when both services are enabled. 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.
- Design consideration: Amazon Inspector automatically exports findings to Amazon EventBridge, and, optionally, to an Amazon Simple Storage Service (Amazon S3) bucket. To export active findings to an S3 bucket, you need a KMS key that Amazon Inspector can use to encrypt findings and an S3 bucket with permissions that allow Amazon Inspector to upload objects. EventBridge integration enables you to monitor and process findings in near real time as part of

your existing security and compliance workflows. EventBridge events are published to the Amazon Inspector delegated administrator account in addition to the member account from which they originated.

Deploying common security services within all AWS accounts

The [Apply security services across your AWS organization](#) 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.

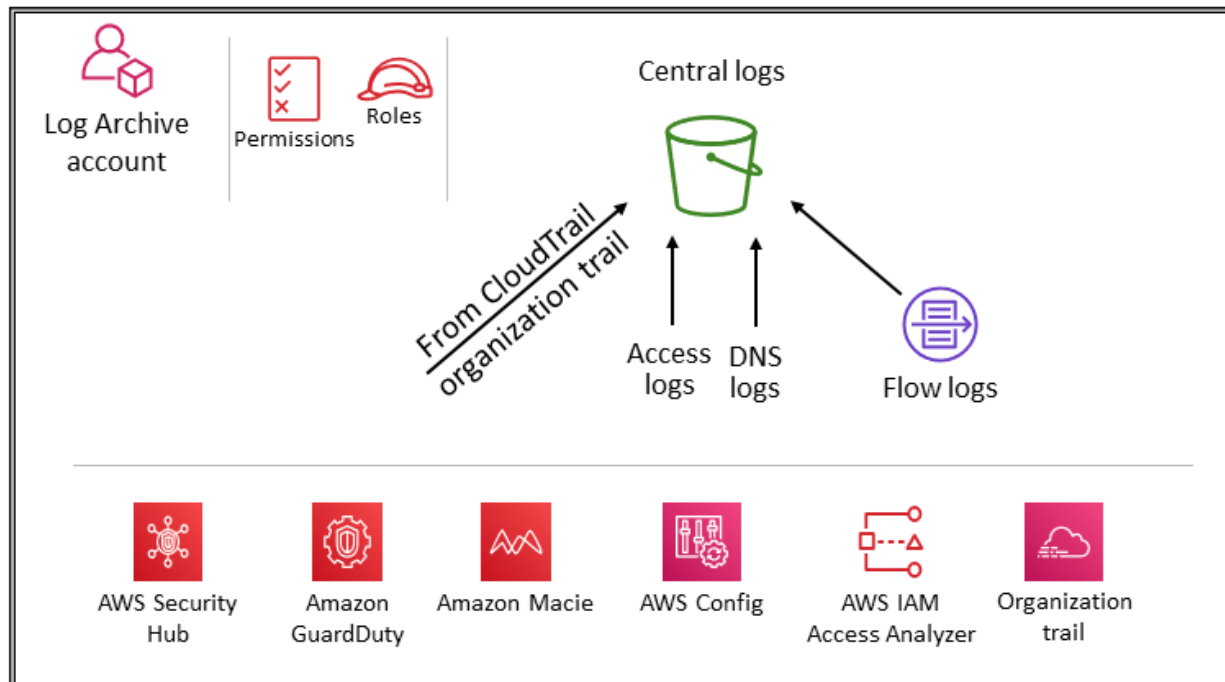
Security Hub, GuardDuty, AWS Config, Access Analyzer, and AWS CloudTrail organization trails appear in all accounts. The first three support the delegated administrator feature discussed previously in the [Management account, trusted access, and delegated administrators section](#). CloudTrail currently uses a different aggregation mechanism.

The AWS SRA [GitHub code repository](#) provides a sample implementation of enabling Security Hub, GuardDuty, AWS Config, Firewall Manager, and CloudTrail organization trails across all your accounts, including the AWS Org Management account.

- Design consideration: Specific account configurations might necessitate additional security services. For example, accounts that manage S3 buckets (the Application and Log Archive accounts) 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.) Another example is Amazon Inspector, which is applicable only for accounts that host either EC2 instances or Amazon ECR images.
- Design consideration: In addition to the services described previously in this section, the AWS SRA includes two security-focused services, AWS Detective and AWS Audit Manager, which support AWS Organizations integration and the delegated administrator functionality. However, those are not included as part of the recommended services for account baselining, because we have seen that these services are best used in the following scenarios:
 - You have a dedicated team or group of resources that perform these functions. Detective is best utilized by security analyst teams and Audit Manager is helpful to your internal audit or compliance teams.
 - You want to focus on a core set of tools such as GuardDuty and Security Hub at the start of your project, and then build on these by using services that provide additional capabilities.

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 can 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 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

Many AWS services log information in Amazon S3—either by default or exclusively. AWS CloudTrail, Amazon VPC Flow Logs, AWS Config, and Elastic Load Balancing are some examples of services that log information in Amazon S3. This means that log integrity is achieved through S3 object integrity; log confidentiality is achieved through S3 object access controls; and log availability is achieved through S3 Object Lock, S3 object versions, and S3 Lifecycle rules. By logging information in a dedicated and centralized S3 bucket that resides in a dedicated account, you can manage these logs in just a few buckets and enforce strict security controls, access, and separation of duties.

In the AWS SRA, the primary logs stored in Amazon S3 come from CloudTrail, so this section describes how to protect those objects. This guidance also applies to any other S3 objects created either by your own applications or by other AWS services. Apply these patterns whenever you have data in Amazon S3 that needs high integrity, strong access control, and automated retention or destruction.

By default, CloudTrail logs in S3 buckets are encrypted by Amazon server-side encryption with Amazon S3-managed encryption keys (SSE-S3). This helps protect the data at rest, but access control is controlled exclusively by IAM policies. To provide an additional managed security layer, you can use server-side encryption with AWS KMS keys that you manage (SSE-KMS) on all security S3 buckets. This adds a second level of access control. To read log files, a user must have both Amazon S3 read permissions for the S3 object and an IAM policy or role applied that allows them permissions to decrypt by the associated key policy.

Two options help you protect or verify the integrity of CloudTrail log objects in Amazon S3. CloudTrail provides [log file integrity validation](#) to determine whether a log file was modified or deleted after CloudTrail delivered it. The other option is S3 Object Lock.

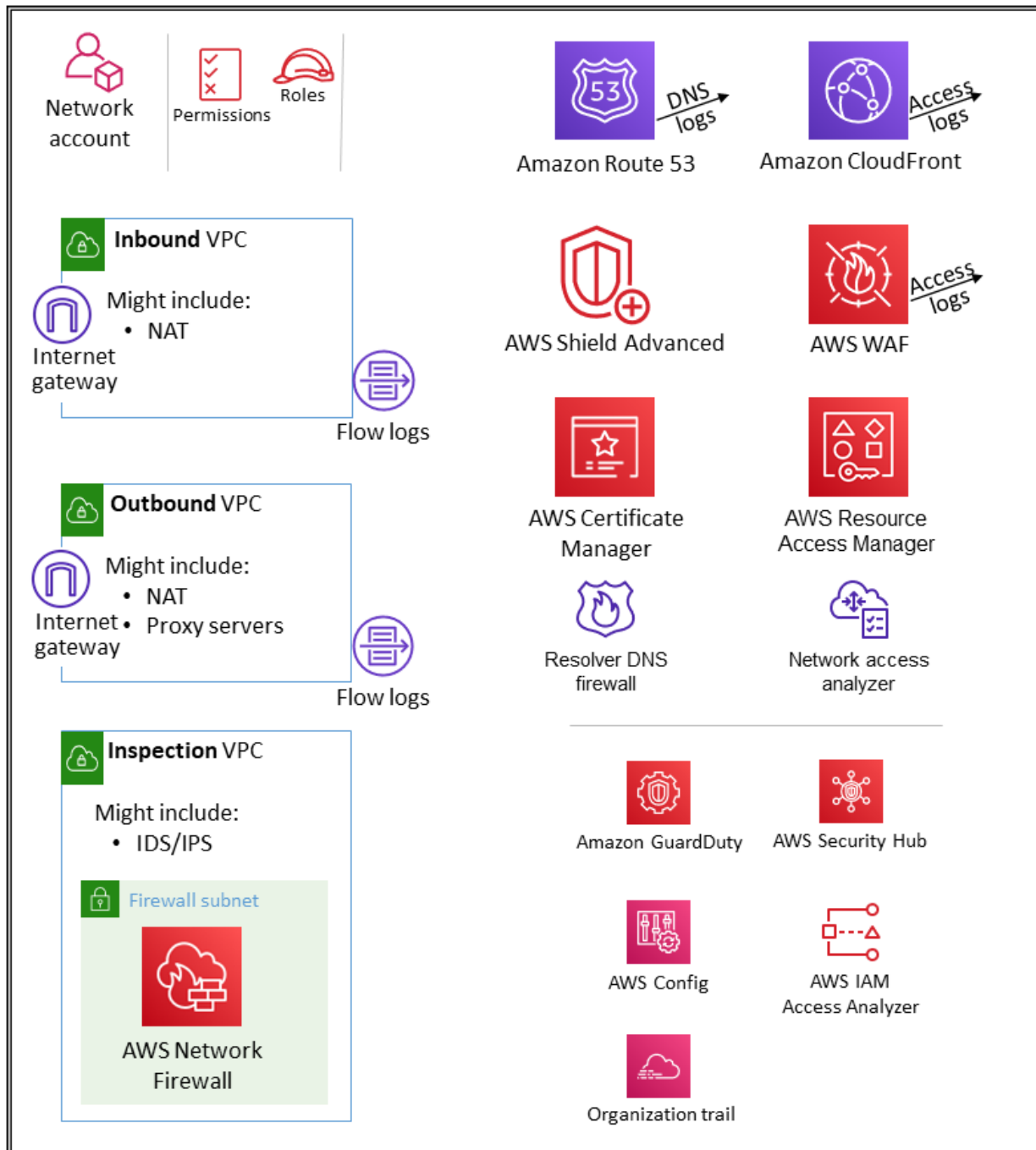
In addition to protecting the S3 bucket itself, you can adhere to the principle of least privilege for the logging services (for example, CloudTrail) and the Log Archive account. For example, users with permissions granted by the AWS managed IAM policy `AWSCloudTrail_FullAccess` can disable or reconfigure 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 the blog post [Top 10 security best practices for securing data in Amazon S3](#).

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 manage the majority of the infrastructure in this account.

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 that are in separate 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](#) – PrivateLink provides private connectivity between VPCs, services, and applications. You can create your own application in your VPC and configure it as a 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 PrivateLink, service traffic doesn't pass across a publicly routable network. Use 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 PrivateLink uses elastic network interfaces within the client VPC so that there are no IP conflicts with the service provider.
- [AWS Transit Gateway](#) – 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 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 the 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 specifics of the application, you can 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 specifics of the application, you can 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 effortlessly deploy and manage stateful inspection, intrusion prevention and detection, and web filtering to help 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 might have been written internally or sourced externally from third-party vendors or open-source platforms.

In the AWS SRA, Network Firewall is used within the network account because the network control-focused functionality of the service aligns with the intent of the account.

- Design consideration: AWS Firewall Manager supports Network Firewall, so you can 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 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.
- Design consideration: There are [multiple deployment models](#) available with Network Firewall. The right model depends on your 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) 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.
- Design consideration: 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.

Network Access Analyzer

[Network Access Analyzer](#) is a feature of Amazon VPC that identifies unintended network access to your resources. You can use Network Access Analyzer to validate network segmentation, identify resources that are accessible from the internet or accessible only from trusted IP address ranges, and validate that you have appropriate network controls on all network paths.

Network Access Analyzer uses automated reasoning algorithms to analyze the network paths that a packet can take between resources in an AWS network, and produces findings for paths that match your defined [Network Access Scope](#). Network Access Analyzer performs a static analysis

of a network configuration, meaning that no packets are transmitted in the network as part of this analysis.

The Amazon Inspector Network Reachability rules provide a related feature. The findings generated by these rules are used in the Application account. Both Network Access Analyzer and Network Reachability use the latest technology from the [AWS Provable Security initiative](#), and they apply this technology with different areas of focus. The Network Reachability package focuses specifically on EC2 instances and their internet accessibility.

The network account defines the critical network infrastructure that controls the traffic in and out of your AWS environment. This traffic needs to be tightly monitored. In the AWS SRA, Network Access Analyzer is used within the Network account to help identify unintended network access, identify internet-accessible resources through internet gateways, and verify that appropriate network controls such as network firewalls and NAT gateways are present on all network paths between resources and internet gateways.

- Design consideration: Network Access Analyzer is a feature of Amazon VPC, and it can be used in any AWS account that has a VPC. Network administrators can get tightly scoped, cross-account IAM roles to validate that approved network paths are enforced within each AWS account.

AWS Certificate Manager

AWS Certificate Manager (ACM) lets you provision, manage, and deploy public and private TLS certificates for use with AWS services and your internal connected resources. With ACM, you can quickly request a certificate, deploy it on ACM-integrated AWS resources, such as Elastic Load Balancing load balancers, Amazon CloudFront distributions, and APIs on Amazon API Gateway, and let ACM handle certificate renewals. There is no need to generate a key pair or certificate signing request (CSR), submit a CSR to a certificate authority (CA), or upload and install the certificate when it is received. ACM also provides the option to import TLS certificates issued by third-party CAs and deploy them with ACM integrated services. When you use ACM to manage certificates, certificate private keys are securely protected and stored by using strong encryption and key management best practices. With ACM there is no additional charge for provisioning public certificates, and ACM manages the renewal process.

ACM is used in the Network account to generate a public TLS certificate, which, in turn, is used by CloudFront distributions to establish the HTTPS connection between viewers and CloudFront. For more information, see the [CloudFront documentation](#).

- 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 403 (*Forbidden*) status code.

In the AWS SRA, AWS WAF is used in the Network account, because it protects CloudFront.

- Design consideration: [CloudFront provides features](#) that enhance the AWS WAF functionality and make the two services work better together.
- Design consideration: You can use AWS WAF, AWS Firewall Manager, and AWS Shield together to create a comprehensive security solution. It starts with AWS WAF. You can automate and then simplify AWS WAF management by 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 malicious events, you can consider purchasing the additional features that AWS Shield Advanced provides.

Amazon Route 53

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. You can use Route 53 to perform three main functions: domain registration, DNS routing, and health checking.

You can use Route 53 as the DNS service to map domain names to your EC2 instances, S3 buckets, CloudFront distributions, and other AWS resources. The distributed nature of our DNS servers helps ensure that your end users are routed to your application consistently. Features such as Route 53 traffic flow and routing control help you improve reliability with simply configured failover to reroute your users to an alternate location if your primary application endpoint becomes unavailable. Route 53 Resolver provides recursive DNS for your VPC and on-premises networks over AWS Direct Connect or AWS managed VPN.

By using the AWS Identity and Access Management (IAM) service with Route 53, you get fine-grained control over who can update your DNS data. You can enable DNS Security Extensions (DNSSEC) signing to let DNS resolvers validate that a DNS response came from Route 53 and has not been tampered with.

[Route 53 Resolver DNS Firewall](#) provides protection for outbound DNS requests from your VPCs. These requests route through Route 53 Resolver for domain name resolution. A primary use of DNS Firewall protections is to help prevent DNS exfiltration of your data. With DNS Firewall, you can monitor and control the domains that your applications can query. You can deny access to the domains that you know to be bad, and allow all other queries to pass through. Alternately, you can deny access to all domains except for the ones that you explicitly trust. You can also use DNS Firewall to block resolution requests to resources in private hosted zones (shared or local), including VPC endpoint names. It can also block requests for public or private EC2 instance names.

Route 53 resolvers are created by default as part of every VPC. In the AWS SRA, Route 53 is used in the Network account primarily for the DNS firewall capability.

- Design consideration: DNS Firewall and AWS Network Firewall both offer domain name filtering, but for different types of traffic. You can use DNS Firewall and Network Firewall together to configure domain-based filtering for application-layer traffic over two different network paths.
 - DNS Firewall provides filtering for outbound DNS queries that pass through the Route 53 Resolver from applications within your VPCs. You can also configure DNS Firewall to send custom responses for queries to blocked domain names.
 - Network Firewall provides filtering for both network-layer and application-layer traffic, but does not have visibility into queries made by Route 53 Resolver.

Amazon CloudFront

Amazon CloudFront is a 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 create a custom TLS certificate and enforce HTTPS communications between viewers and CloudFront, as described earlier in the [ACM section](#). You can additionally require that the communications between CloudFront and your custom origin implement end-to-end encryption in transit. For this scenario, you will need to install a TLS certificate on your origin server. If your origin is an ELB load balancer, you can use a certificate generated by ACM or a certificate that is validated by a third-party CA and imported into ACM. If S3 bucket website endpoints serve as the origin, you can't configure CloudFront to use HTTPS with your origin because Amazon S3 doesn't support HTTPS for website endpoints. (However, you can still require HTTPS between viewers and CloudFront.) For all other origins that support installing HTTPS certificates, you must use a certificate that is signed by a trusted third-party CA.

When you use CloudFront as your CDN, you can restrict access to your content by using these capabilities:

- By using signed URLs and signed cookies, you can support token authentication to restrict access to only authenticated viewers.

- By using the geo restriction capability, you can work to 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.

In the AWS SRA, Amazon CloudFront is used within the Network account, because this account provides the networking infrastructure required for workloads to communicate outside AWS.

- Design consideration: CloudFront, AWS Shield, AWS WAF, and Amazon Route 53 work seamlessly together to create a flexible, layered security perimeter against multiple types of unauthorized events, 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.
- Design consideration: 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](#).
- Design consideration: You might also want to evaluate a distributed architecture where all network infrastructure that is required for your workloads to communicate outside AWS resides locally within the Application account. This simplifies the network architecture and routing, and would help reduce costs by not requiring cross-account ingress/egress charges. In a distributed architecture, you have to enforce strong security oversight to exercise security controls for all network paths.

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, AWS 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) events. 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 unauthorized events that target your applications running on protected Amazon Elastic Compute Cloud (Amazon EC2), Elastic Load Balancing (ELB), Amazon CloudFront, AWS Global Accelerator, and Route 53 resources. Shield Advanced records metrics that you can monitor in Amazon 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 can consider the additional features that Shield Advanced provides.

AWS RAM

AWS Resource Access Manager (AWS RAM) helps you securely share the AWS resources that you create in one AWS account with other AWS accounts. AWS RAM provides a central place to manage the sharing of resources and to standardize this experience across accounts. This makes it simpler to manage resources while taking advantage of the administrative and billing isolation, and reduce the scope of impact containment benefits provided by a multi-account strategy. If your account is managed by AWS Organizations, AWS RAM lets you share resources with all accounts in the organization, or only with the accounts within one or more specified organizational units (OUs). You can also share with specific AWS accounts by account ID, regardless of whether the account is part of an organization. You can also share [some supported resource types](#) with specified IAM roles and users.

AWS RAM enables you to share resources that do not support IAM resource-based policies, such as VPC subnets and Route 53 rules. Furthermore, with AWS RAM, the owners of a resource can see which principals have access to individual resources that they have shared. IAM principals can retrieve the list of resources shared with them directly, which they can't do with resources shared by IAM resource policies. If AWS RAM is used to share resources outside your AWS organization, an invitation process is initiated. The recipient must accept the invitation before access to the resources is granted. This provides additional checks and balances.

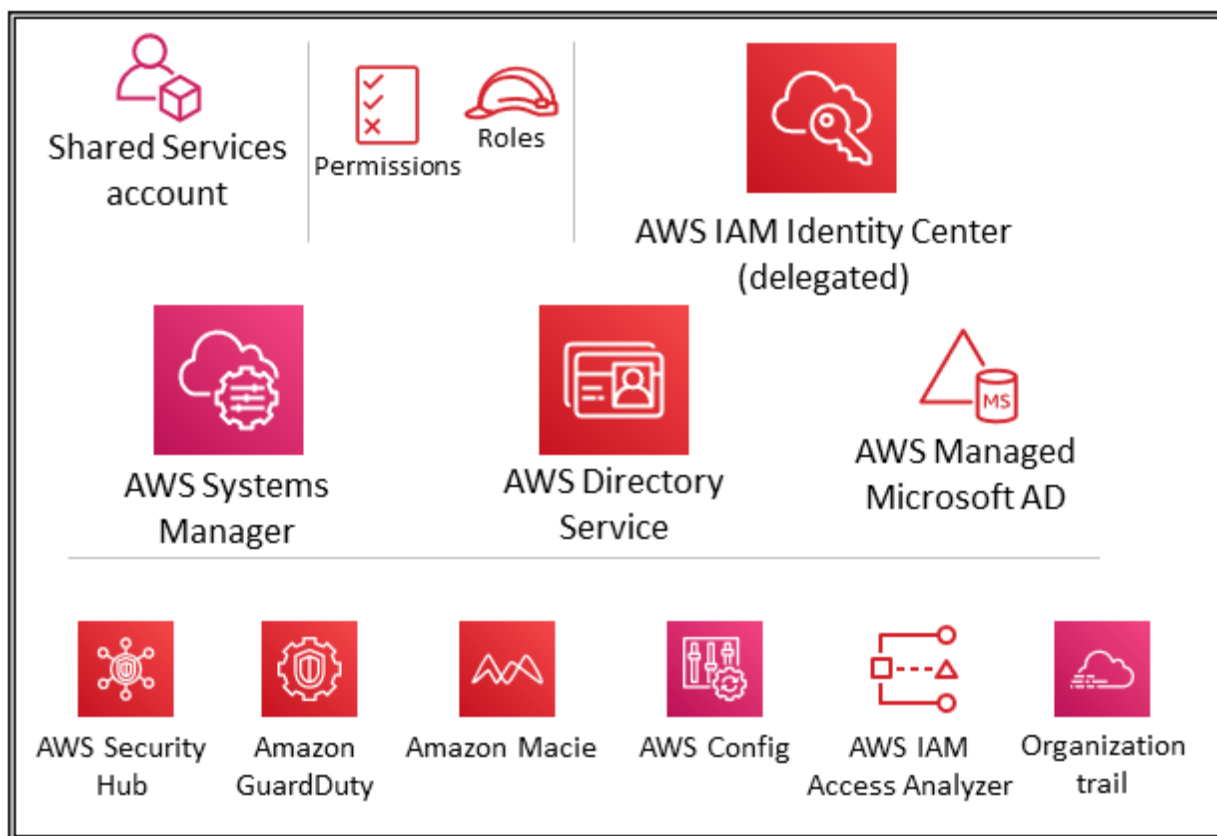
AWS RAM is invoked and managed by the resource owner, in the account where the shared resource is deployed. One common use case for AWS RAM illustrated in the AWS SRA is for network administrators to share VPC subnets and transit gateways with the entire AWS organization. This provides the ability to decouple AWS account and network management functions and helps achieve separation of duties. For more information about VPC sharing, see the AWS blog post [VPC sharing: A new approach to multiple accounts and VPC management](#) and the [AWS network infrastructure whitepaper](#).

- Design consideration: Although AWS RAM as a service is deployed only within the Network account in the AWS SRA, it would typically be deployed in more than one account. For example, you can centralize your data lake management to a single data lake account, and then share the AWS Lake Formation data catalog resources (databases and tables) with other accounts in your AWS organization. For more information, see the [AWS Lake Formation documentation](#) and the AWS blog post [Securely share your data across AWS accounts using AWS Lake Formation](#). Additionally, security administrators

can use AWS RAM to follow best practices when they build an ACM Private CA hierarchy. CAs can be shared with external third parties, who can issue certificates without having access to the CA hierarchy. This allows origination organizations to limit and revoke third-party access.

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.

Systems Manager helps you work to maintain security and compliance by scanning your managed instances and reporting (or taking corrective action) on any policy violations it detects. By pairing Systems Manager with appropriate deployments in individual member AWS accounts (for example, the Application account), you can coordinate instance inventory data collection and centralize automation such as patching and security updates.

AWS Managed Microsoft AD

AWS Directory Service for Microsoft Active Directory, also known as AWS Managed Microsoft AD, enables your directory-aware workloads and AWS resources to use managed Active Directory on AWS. You can use AWS Managed Microsoft AD to join [Amazon EC2 for Windows Server](#), [Amazon EC2 for Linux](#), and [Amazon RDS for SQL Server](#) instances to your domain, and use [AWS end user computing](#) (EUC) services, such as [Amazon WorkSpaces](#), with Active Directory users and groups.

AWS Managed Microsoft AD helps you extend your existing Active Directory to AWS and use your existing on-premises user credentials to access cloud resources. You can also administer your on-premises users, groups, applications, and systems without the complexity of running and maintaining an on-premises, highly available Active Directory. You can join your existing computers, laptops, and printers to an AWS Managed Microsoft AD domain.

AWS Managed Microsoft AD is built on Microsoft Active Directory and doesn't require you to synchronize or replicate data from your existing Active Directory to the cloud. You can use familiar Active Directory administration tools and features, such as Group Policy Objects (GPOs), domain trusts, fine-grained password policies, group Managed Service Accounts (gMSAs), schema extensions, and Kerberos-based single sign-on. You can also delegate administrative tasks and authorize access using Active Directory security groups.

Multi-Region replication enables you to deploy and use a single AWS Managed Microsoft AD directory across multiple AWS Regions. This makes it easier and more cost-effective for you to deploy and manage your Microsoft Windows and Linux workloads globally. When you use the automated multi-Region replication capability, you get higher resiliency while your applications use a local directory for optimal performance.

AWS Managed Microsoft AD supports Lightweight Directory Access Protocol (LDAP) over SSL/TLS, also known as LDAPS, in both client and server roles. When acting as a server, AWS Managed Microsoft AD supports LDAPS over ports 636 (SSL) and 389 (TLS). You enable server-side LDAPS communications by installing a certificate on your AWS Managed Microsoft AD domain controllers from an AWS-based Active Directory Certificate Services (AD CS) certificate authority (CA). When acting as a client, AWS Managed Microsoft AD supports LDAPS over ports 636 (SSL). You can enable client-side LDAPS communications by

registering CA certificates from your server certificate issuers into AWS, and then enable LDAPS on your directory.

In the AWS SRA, AWS Directory Service is used within the Shared Services account to provide domain services for Microsoft-aware workloads across multiple AWS member accounts.

- Design consideration: You can grant your on-premises Active Directory users access to sign in to the AWS Management Console and AWS Command Line Interface (AWS CLI) with their existing Active Directory credentials by using IAM Identity Center and selecting AWS Managed Microsoft AD as the identity source. This enables your users to assume one of their assigned roles at sign-in, and to access and take action on the resources according to the permissions defined for the role. An alternative option is to use AWS Managed Microsoft AD to enable your users to assume an [AWS Identity and Access Management](#) (IAM) role.

IAM Identity Center

The AWS SRA uses the delegated administrator feature supported by IAM Identity Center to delegate most of the administration of IAM Identity Center to the Shared Services account. This helps restrict the number of users who require access to the Org Management account. IAM Identity Center still needs to be enabled in the Org Management account to perform certain tasks, including the management of permission sets that are provisioned within the Org Management account.

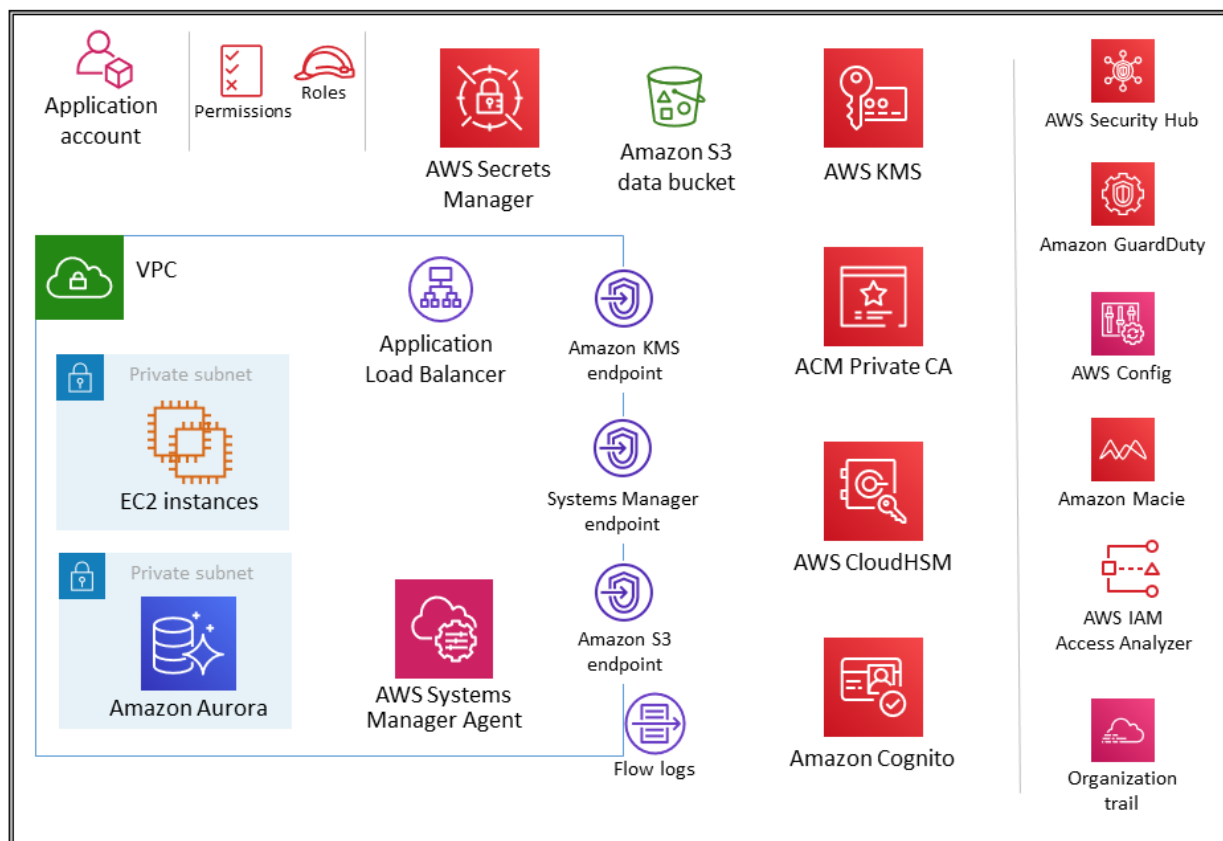
The primary reason for using the Shared Services account as the delegated administrator for IAM Identity Center is the Active Directory location. If you plan to use Active Directory as your IAM Identity Center identity source, you will need to locate the directory in the member account where you have enabled the IAM Identity Center delegated administrator feature. In the AWS SRA, the Shared Services account hosts AWS Managed Microsoft AD, so that account is made the delegated administrator for IAM Identity Center.

IAM Identity Center supports the registration of a single member account as a delegated administrator at one time. You can register a member account only when you sign in with credentials from the management account. To enable delegation, you have to consider the prerequisites listed in the [IAM Identity Center documentation](#). The delegated administrator account can perform most IAM Identity Center management tasks, but with some restrictions, which are listed in the [IAM Identity Center documentation](#). Access to the delegated administrator account for IAM Identity Center should be tightly controlled.

- Design consideration: If you decide to change the IAM Identity Center identity source from any other source to Active Directory, or change it from Active Directory to any other source, the directory must reside in (be owned by) the IAM Identity Center delegated administrator member account, if one exists; otherwise, it must be in the management account.

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, resources inside a VPC are routable to one another. There are two private subnets: one to host the 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. Amazon 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](#).

Use separate VPCs (as subset of account boundaries) to isolate infrastructure by workload segments. Use subnets to isolate the tiers of your application (for example, web, application, and database) within a single VPC. Use private subnets for your instances if they should not be accessed directly from the internet. To call the Amazon EC2 API from your VPC without sending traffic over the public internet, use AWS PrivateLink. Restrict access to your instances by using [security groups](#). Use [VPC Flow Logs](#) to monitor the traffic that reaches your instances. Use [Session Manager](#), a capability of AWS Systems Manager, to access your instances remotely instead of opening inbound SSH ports and managing SSH keys. Use separate Amazon Elastic Block Store (Amazon EBS) volumes for the operating system and your data. You can [configure your AWS account](#) to enforce the encryption of the new EBS volumes and snapshot copies that you create.

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 (TLS server) public 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 consideration: For common scenarios such as strictly internal applications that require a private TLS certificate on the Application Load Balancer, you can use ACM within this account to generate a private certificate from ACM Private CA. In the AWS SRA, the ACM root Private CA is hosted in the Security Tooling account and can be shared with the whole AWS organization or with specific

AWS accounts to issue end-entity certificates, as described earlier in the [Security Tooling account](#) section.

- Design consideration: For public certificates, you can use ACM to generate those certificates and manage them, including automated rotation. Alternatively, you can generate your own certificates by using 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.
- Design consideration: 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](#).

ACM Private CA

AWS Certificate Manager Private Certificate Authority (ACM Private CA) is used in the Application account to generate private certificates to be used with an Application Load Balancer. It is a common scenario for Application Load Balancers to serve secure content over TLS. This requires TLS certificates to be installed on the Application Load Balancer. For applications that are strictly internal, private TLS certificates can provide the secure channel.

In the AWS SRA, ACM Private CA is hosted in the Security Tooling account and is shared out to the Application account by using AWS RAM. This allows developers in an Application account to request a certificate from a shared private CA. Sharing CAs across your organization or across AWS accounts helps reduce the cost and complexity of creating and managing duplicate CAs in all your AWS accounts. When you use ACM to issue private certificates from a shared CA, the certificate is generated locally in the requesting account, and ACM provides full lifecycle management and renewal.

Amazon Inspector

The AWS SRA uses Amazon Inspector to automatically discover and scan EC2 instances and container images that reside in the Amazon Elastic Container Registry (Amazon ECR) for software vulnerabilities and unintended network exposure.

Amazon Inspector is placed in the Application account, because it provides vulnerability management services to EC2 instances in this account. Additionally, Amazon Inspector reports on [unwanted network paths](#) to and from EC2 instances.

Amazon Inspector in member accounts is centrally managed by the delegated administrator account. In the AWS SRA, the Security Tooling account is the delegated administrator account. The delegated administrator account can manage findings data and certain settings for members of the organization. This includes viewing aggregated findings details for all member accounts, enabling or disabling scans for member accounts, and reviewing scanned resources within the AWS organization.

- Design consideration: You can use [Patch Manager](#), a capability of AWS Systems Manager, to trigger on-demand patching to remediate Amazon Inspector zero-day or other critical security vulnerabilities. Patch Manager helps you patch those vulnerabilities without having to wait for your normal patching schedule. The remediation is carried out by using the Systems Manager Automation runbook. For more information, see the two part blog series [Automate vulnerability management and remediation in AWS using Amazon Inspector and AWS Systems Manager](#).

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 work to 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. [AWS 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. Systems Manager 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 [Session Manager](#), a capability of Systems 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 Patch Manager, a capability of Systems Manager, to apply patches to EC2 instances for both operating systems and applications.

The AWS SRA also uses [Automation](#), a capability of Systems Manager, to simplify common maintenance and deployment tasks of Amazon EC2 instances and other AWS resources. Automation can simplify common IT tasks such as changing the state of one or more nodes (using an approval automation) and managing node states according to a schedule. Systems Manager includes features that help you target large groups of instances by using tags, and velocity controls that help you roll out changes according to the limits you define. Automation offers one-click automations for simplifying complex tasks such as creating golden Amazon Machine Images (AMIs) and recovering unreachable EC2 instances. Additionally, you can enhance operational security by giving IAM roles access to specific runbooks to perform certain functions, without directly giving permissions to those roles. For example, if you want an IAM

role to have permissions to restart specific EC2 instances after patch updates, but you don't want to grant the permission directly to that role, you can instead create an Automation runbook and give the role permissions to only run the runbook.

- Design consideration: 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).
- Design consideration: SSM Agent has to communicate with different AWS services and resources such as Amazon 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 VPC endpoints for the SSM Agent to establish private network paths to various AWS services.
- Design consideration: Using Automation, you can share best practices with the rest of your organization. You can create best practices for resource management in runbooks and share the runbooks across AWS Regions and groups. You can also constrain the allowed values for runbook parameters. For these use cases, you might have to create Automation runbooks in a central account such as Security Tooling or Shared Services and share them with the rest of the AWS organization. Common use cases include the capability to centrally implement patching and security updates, remediate drift on VPC configurations or S3 bucket policies, and manage EC2 instances at scale. For implementation details, see the [Systems Manager documentation](#).

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

The AWS SRA illustrates the recommended distribution model for key management, where the KMS key resides within the same AWS account as the resource to be encrypted. For this reason, AWS KMS is used in the Application account in addition to being included in the Security Tooling account. In the Application account, AWS KMS is used to manage keys that are specific to the application resources. You can implement a separation of duties by using [key policies](#) to grant key usage permissions to local application roles and to restrict management and monitoring permissions to your key custodians.

- Design consideration: In a distributed model, the AWS KMS key management responsibility resides with the application team. However, your central security team can be responsible for the governance and [monitoring](#) of important cryptographic events such as the following:
 - The imported key material in a KMS key is nearing its expiration date.
 - A KMS key that is pending deletion is still being used.
 - The key material in a KMS key was automatically rotated.
 - A KMS key was deleted.
 - There is a high rate of decryption failure.

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 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 CloudHSM. You could similarly deploy an HSM from 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 CloudHSM cluster as a custom key store rather than using the native KMS key 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 KMS keys. This combines single-tenant HSMs under your control with the ease of use and integration of AWS KMS. To manage your CloudHSM infrastructure, you have to employ a public key infrastructure (PKI) and have a team that has experience managing HSMs.

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 efficiently 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. Additionally, Secrets Manager helps you move your applications between environments (development, pre-production, production). Instead of changing the code, you can ensure that an appropriately named and referenced secret is available in the environment. This promotes the consistency and reusability of application code across different environments, while requiring fewer changes and human interactions after the code has been tested.

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.

Secrets Manager uses [envelope encryption](#) with AWS KMS keys and data keys to protect each secret value. When you create a secret, you can choose any symmetric customer managed key in the AWS account and Region, or you can use the AWS managed key for Secrets Manager.

As a best practice, you can monitor your secrets to log any changes to them. This helps you ensure that any unexpected usage or change can be investigated. Unwanted changes can be rolled back. Secrets Manager currently supports two AWS services that enable you to monitor your organization and activity: AWS CloudTrail and AWS Config. CloudTrail captures all API calls for Secrets Manager as events, including calls from the Secrets Manager console and from code calls to the Secrets Manager APIs. In addition, CloudTrail captures other related (non-API) events that might have a security or compliance impact on your AWS account or might help you troubleshoot operational problems. These include certain secrets rotation events and deletion of secret versions. AWS Config can provide detective controls by tracking and monitoring changes to secrets in Secrets Manager. These changes include a secret's description, rotation configuration, tags, and 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 configuration and compliance change notifications from AWS Config, to route particular secrets events for notification or remediation actions.

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. Here, 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 consideration: 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 might be appropriate, secrets can be centrally managed by Secrets Manager in the Security Tooling account.

Amazon Cognito

Amazon Cognito lets you add user sign-up, sign-in, and access control to your web and mobile apps quickly and efficiently. Amazon Cognito scales to millions of users and supports sign-in with social identity providers, such as Apple, Facebook, Google, and Amazon, and enterprise identity providers through SAML 2.0 and OpenID Connect. The two main components of Amazon Cognito are [user pools](#) and [identity pools](#). User pools are user directories that provide sign-up and sign-in options for your application users. Identity pools enable you to grant your users access to other AWS services. You can use identity pools and user pools separately or together. For common usage scenarios, see the [Amazon Cognito documentation](#).

Amazon Cognito provides a built-in and customizable UI for user sign-up and sign-in. You can use Android, iOS, and JavaScript SDKs for Amazon Cognito to add user sign-up and sign-in pages to your apps. [Amazon Cognito Sync](#) is an AWS service and client library that enables cross-device syncing of application-related user data.

Amazon Cognito supports multi-factor authentication and encryption of data at rest and data in transit. Amazon Cognito user pools provide [advanced security features](#) to help protect access to user accounts in your application. These advanced security features provide risk-based adaptive authentication and protection from the use of compromised credentials.

- Design consideration: You can create an AWS Lambda function and then trigger that function during user pool operations such as user sign-up, confirmation, and sign-in (authentication) with an AWS Lambda trigger. You can add authentication challenges, migrate users, and customize verification messages. For common operations and user flow, see the [Amazon Cognito documentation](#). Amazon Cognito calls Lambda functions synchronously.
- Design consideration: You can use Amazon Cognito user pools to secure small, multi-tenant applications. A common use case of multi-tenant design is to run workloads to support testing multiple versions of an application. Multi-tenant design is also useful for testing a single application with different datasets, which allows full use of your cluster

resources. However, make sure that the number of tenants and expected volume align with the related Amazon Cognito [service quotas](#). These quotas are shared across all tenants in your application.

Layered defense

The Application account provides an opportunity to illustrate layered defense principals that AWS enables. Consider the security of the EC2 instances that make up the core of a simple example application represented in the AWS SRA and you can see the way AWS services work together in a layered defense. This approach aligns to the structural view of AWS security services, as described in the section [Apply security services across your AWS organization](#) earlier in this guide.

- The innermost layer is the EC2 instances. As mentioned earlier, EC2 instances include many native security features either by default or as options. Examples include [IMDSv2](#), the [Nitro system](#), and [Amazon EBS storage encryption](#).
- The second layer of protection focuses on the operating system and software running on the EC2 instances. Services such as [Amazon Inspector](#) and [AWS Systems Manager](#) enable you to monitor, report, and take corrective action on these configurations. Inspector [monitors your software for vulnerabilities](#) and Systems Manager helps you work to maintain security and compliance by scanning managed instances for their [patch](#) and [configuration status](#), and then reporting and taking any [corrective actions](#) you specify.
- The instances, and the software running on these instances, sit with your AWS networking infrastructure. In addition to using the [security features of Amazon VPC](#), the AWS SRA also makes use of VPC endpoints to provide private connectivity between the VPC and supported AWS services, and to provide a mechanism to place access policies at the network boundary.
- The activity and configuration of the EC2 instances, software, network, and IAM roles and resources are further monitored by AWS account-focused services such as AWS Security Hub, Amazon GuardDuty, AWS CloudTrail, AWS Config, AWS IAM Access Analyzer, and Amazon Macie.
- Finally, beyond the Application account, AWS RAM helps control which resources are shared with other accounts, and IAM service control policies help you enforce consistent permissions across the AWS organization.

IAM resources

Although AWS Identity and Access Management (IAM) is not a service that is included 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 explanation 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 an 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](#).

Use case or policy	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, OU, account	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 automation roles [3]	Single account [4]	Principals used by automation within a member account	Member accounts
Baseline human policies (the IAM 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 developer can assign to another principal)	Restrict	Central team, such as platform, security, or IAM team [1]	Guardrails for application roles (must be applied)	Single account [4]	Individual roles for an application or workload in this account [7]	Member accounts
Machine role policies for applications (role attached to infrastructure deployed by developers)	Grant and restrict	Delegated to developers [8]	Permission for the application or workload [9]	Single account	A principal in this account	Member accounts
Resource policies	Grant and restrict	Delegated to developers [8,10]	Permissions to resources	Single account	A principal in an account [11]	Member accounts

Notes from the 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 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 developers in 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 to 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 application 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 some 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 refer to any principal in any account to allow or deny access to its resources. It can even refer to anonymous principals to enable 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

To help you get started building and implementing the guidance in the AWS SRA, an infrastructure as code (IaC) repository at <https://github.com/aws-samples/aws-security-reference-architecture-examples> accompanies this guide. This repository contains code to help developers and engineers deploy some of the guidance and architecture patterns presented in this document. This code is drawn from AWS Professional Services consultants' first-hand experience with customers. The templates are general in nature—their goal is to illustrate an implementation pattern rather than provide a complete solution. The AWS service configurations and resource deployments are deliberately very restrictive. You might need to modify and tailor these solutions to suit your environment and security needs.

The examples within this repository have been deployed and tested within an AWS Control Tower environment by using AWS CloudFormation and the [Customizations for AWS Control Tower \(CfCT\)](#) solution. The CfCT solution helps customers quickly set up a secure, multi-account AWS environment based on AWS best practices. It helps 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. AWS Control Tower also provides 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 additional security configurations to implement the patterns described in this document.

Here is a summary of the solutions in the [AWS SRA repository](#). Each solution includes a README.md file with details.

- The [CloudTrail Organization](#) solution creates an organization trail within the Org Management account. This trail is encrypted with a customer managed key created 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 AWS Lambda functions. An organization trail logs events for all AWS accounts in the AWS organization while preventing member accounts from modifying the configurations.
- The [GuardDuty Organization](#) solution enables Amazon GuardDuty by delegating administration to the Security Tooling account. It configures GuardDuty within the Security Tooling 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 [Security Hub Organization](#) solution configures AWS Security Hub by delegating administration to the Security Tooling account. It configures Security Hub within the Security Tooling account for all existing and future AWS organization accounts. The solution also provides parameters for synchronizing the enabled security standards across all accounts and Regions as well as configuring a Region aggregator within the Security Tooling account. 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 AWS Partner integrations.

- The [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.
- The [Macie Organization](#) solution enables Amazon Macie by delegating administration to the Security Tooling account. It configures Macie within the Security Tooling 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.
- AWS Config
 - The [Config Aggregator](#) solution configures an AWS Config aggregator by delegating administration to the Security Tooling account. The solution then configures an AWS Config aggregator within the Security Tooling account for all existing and future accounts in the AWS organization.
 - The [Conformance Pack Organization Rules](#) solution deploys AWS Config rules by delegating administration to the Security Tooling account. It then creates an organization conformance pack within the delegated administrator account for all existing and future accounts in the AWS organization. The solution is configured to deploy the [Operational Best Practices for Encryption and Key Management](#) conformance pack sample template.
 - The [AWS Config Control Tower Management Account](#) solution enables AWS Config in the AWS Control Tower management account and updates the AWS Config aggregator within the Security Tooling account accordingly. The solution uses the AWS Control Tower CloudFormation template for enabling AWS Config as a reference to ensure consistency with the other accounts in the AWS organization.
- IAM
 - The [Access Analyzer](#) solution enables AWS IAM Access Analyzer by delegating administration to the Security Tooling account. It then configures an organization-level Access Analyzer within the Security Tooling account for all existing and future accounts in the AWS organization. The solution also deploys Access Analyzer to all member accounts and Regions to support analyzing account-level permissions.
 - The [IAM Password Policy](#) solution updates the AWS account password policy within all accounts in an AWS organization. The solution provides parameters for configuring the password policy settings to help you align with industry compliance standards.
- The [EC2 Default EBS Encryption](#) solution enables account-level, default Amazon EBS encryption within each AWS account and AWS Region in the AWS organization. It enforces the encryption of new EBS volumes and snapshots that you create. For example, Amazon EBS encrypts the EBS volumes that are created when you launch an instance and the snapshots that you copy from an unencrypted snapshot.
- The [S3 Block Account Public Access](#) solution enables Amazon S3 account-level settings within each AWS account in the AWS organization. The Amazon S3 Block Public Access feature provides settings for access points, buckets, and accounts to help you

manage public access to Amazon S3 resources. By default, new buckets, access points, and objects don't allow public access. However, users can modify bucket policies, access point policies, or object permissions to allow public access. Amazon S3 Block Public Access settings override these policies and permissions so that you can limit public access to these resources.

Acknowledgments

Primary authors

- Avik Mukherjee, AWS Senior Consultant
- Andy Wickersham, AWS Senior Consultant
- Neal Rothleder, AWS Principal Consultant
- Paul Grassi, AWS Principal Consultant

Contributors

- 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
- Eric Rose, AWS Principal Consultant
- Ilya Epshteyn, Senior Manager, Identity Solutions
- Handan Selamoglu, AWS Senior Technical Writer

Appendix: AWS security, identity, and compliance services

For an introduction or a refresher, see [Security, Identity, and Compliance on AWS](#) on the AWS website for a list of the AWS services that help you secure your workloads and applications in the cloud. These services are grouped into five categories: data protection, identity & access management, network & application protection, threat detection & continuous monitoring, and compliance & data privacy.

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](#) – 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](#) – Provision, manage, and deploy SSL/TLS certificates for use with AWS services.
- [AWS Secrets Manager](#) – Rotate, manage, and retrieve database credentials, API keys, and other secrets through their lifecycle.

Identity & 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.
- [IAM Identity Center](#) – 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.

Network & application protection – These categories of services enable you to enforce fine-grained security policy at network control points across your organization. AWS services help you inspect and filter traffic to help prevent unauthorized resource access at the host-level, network-level, and application-level boundaries.

- [AWS Shield](#) – Safeguard your web applications that run 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](#) – 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](#) – Deploy essential network protections for your VPCs.
- [Amazon Route 53 DNS Firewall](#) – Protect your outbound DNS requests from your VPCs.

Threat detection & continuous monitoring – 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 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 CloudTrail](#) – Track user activity and API usage to enable governance and operational and risk auditing of your AWS account.
- [Amazon Detective](#) – Analyze and visualize security data to rapidly get to the root cause of potential security issues.
- [AWS Lambda](#) – Run code without provisioning or managing servers so you can scale your programmed, automated response to incidents.

Compliance & data privacy – 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](#) – Use a no-cost, self-service portal to get on-demand access to AWS security and compliance reports and select online agreements.
- [AWS Audit Manager](#) – Continuously audit 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](#).

Change	Description	Date
Major additions and updates	<ul style="list-style-type: none">Added information about how to use the AWS SRA and key implementation guidelines.Added architectural guidance for additional AWS services such as AWS Artifact, Amazon Inspector, AWS RAM, Amazon Route 53, AWS Control Tower, AWS Audit Manager, AWS Directory Service, Amazon Cognito, and Network Access Analyzer.Updated existing guidance to reflect new AWS service features and best practices.	July 25, 2022
—	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