AWS Cloud Adoption Framework: Security Perspective

AWS Cloud Adoption Framework: Security Perspective: AWS Whitepaper

Table of Contents

This whitepaper is for historical reference only. Some content might be outdated and some links might not be available.

AWS Cloud Adoption Framework: Security Perspective

Publication date: **December 12, 2023** (*Document history*)

Strong security is a core enabler of digital transformation, helping organizations adopt machine learning (ML), artificial intelligence (AI), big data, and the speed and scale of the cloud to meet changing business conditions and evolving customer needs. Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform. It can help you transform your organization while reducing business risk, improving environmental, social, and governance (ESG) performance, growing revenue, and improving operational efficiency.

The AWS Cloud Adoption Framework (AWS CAF) uses AWS best practices to help you accelerate your business outcomes. Use the AWS CAF to identify and prioritize transformation opportunities, evaluate and improve your cloud readiness, and iteratively evolve your transformation roadmap.

AWS CAF groups its guidance in six perspectives: *Business*, *People*, *Governance*, *Platform*, *Security*, and *Operations*. Each perspective is covered in a separate whitepaper. This whitepaper covers the *Security* perspective, and will help you achieve the confidentiality, integrity, and availability of your data and cloud workloads.

Introduction

AWS CAF is an enterprise architecture framework that helps work backwards from your strategic priorities and associated business outcomes to prioritize transformation initiatives, identify capability gaps, and iteratively evolve your digital transformation roadmap. AWS CAF identifies specific organizational capabilities that underpin successful cloud transformations. These capabilities may be used to inform the construction of a cloud-ready operating model, develop cloud skills and teams, set up centers of excellence, and adapt organizational structures.

The organizational ability to effectively use the cloud to digitally transform (organizational cloud readiness) is bolstered by a set of foundational capabilities. A *capability* is an organizational ability to use processes to deploy resources (people, technology, and any other tangible or intangible assets) to achieve a particular outcome. The AWS CAF identifies these capabilities and provides prescriptive guidance that thousands of organizations around the world have successfully used to improve their cloud readiness and accelerate their cloud transformation journeys.

AWS Well-Architected is a complementary mechanism that can help you build secure, high-performing, resilient, and efficient workloads for a variety of applications. It provides a consistent

approach for evaluating your architectures and implementing scalable designs. You should perform AWS Well-Architected reviews whenever you are ready to deploy new or optimize existing workloads on AWS.

AWS CAF groups its capabilities in six perspectives:

- Business

- People

- Governance

- Platform

- Security

- Operations

The *Security* perspective helps you achieve the confidentiality, integrity, and availability of your data and cloud workloads. It comprises nine capabilities shown in Figure 1. These are managed by stakeholders who are functionally related in their cloud transformation journey. Common stakeholders include the Board of Directors and chief executive officer (CEO). It can also include other individuals *directly* responsible for mitigating and managing risk, such as chief information security officer (CISO), chief compliance officer (CCO), internal audit leaders, and security architects and engineers.

AWS CAF Security Perspective Capabilities

Security Governance	Develop and communicate security roles, responsibilities, policies, processes, and procedures
Threat Detection	Understand and identify potential security misconfigurations, threats, or unexpected behaviors
Data Protection	Maintain visibility and control over data and how it is accessed and used in your organization
Security Assurance	Monitor, evaluate, manage, and improve the effectiveness of your security and privacy programs
Vulnerability Management	Continuously identify, classify, remediate, and mitigate security vulnerabilities
Application Security	Detect and address security vulnerabilities during the software development process
Identity and Access Management	Manage identities and permissions at scale
Infrastructure Protection	Validate that systems and services within your workload are protected
Incident Response	Reduce potential harm by effectively responding to security incidents

Figure 1. AWS CAF Security perspective capabilities

The goal of the Security perspective is to help you achieve the confidentiality, integrity, and availability of your data and workloads in the AWS Cloud, while improving your security posture. This whitepaper organizes the principles of the nine capabilities that will help you drive the transformation of your organization's security culture. For each capability, we'll discuss specific actions you can take and methods to measure progress.

Security is a top priority for AWS. As organizations embrace the scalability and flexibility of the cloud, AWS helps them evolve their security, identity, and compliance leveraging this new environment. AWS builds security into the very core of the AWS Cloud infrastructure. It offers foundational services to help you meet your unique security requirements in the AWS Cloud.

The goal of your security program remains the same, whether on-premises, in the cloud, or in a hybrid environment. AWS CAF helps you increase program maturity and efficacy, while shortening timelines and reducing costs. The difference in using the cloud is fundamental and impactful - you no longer manage physical security of your data centers, nor the related design, implementation, training, deployment, or maintenance of them. AWS provides and secures the data centers and manages all physical upgrades and maintenance. You can use software-based security tools to monitor and protect the flow of information into and out of your cloud resources. As an AWS customer, you reap the benefit of all the best practices of AWS policies, architecture, and operational processes that satisfy the requirements our most security-sensitive customers.

[AWS Compliance](#) outlines the robust controls in place at AWS for security and data protection in the AWS Cloud. AWS regularly achieves [third-party validation](#) for thousands of global compliance requirements that we continually monitor to help you meet security and compliance needs. Security and Compliance is a [shared responsibility](#) between you and AWS, with AWS being responsible for "Security of the Cloud" while you remain responsible for "Security in the Cloud".

AWS also provides you with a wide range of information about its Information Technology (IT) control environment in whitepapers, reports, certifications, accreditations, and other third-party attestations. More information is available in the [Risk and Compliance whitepaper](#) and at the [AWS Security Center](#).

AWS and the [AWS Partner Network](#) provide tools and services, such as workshops and trainings that can help you on this journey to implement and improve your security posture.

AWS collaborates extensively with the security community to increase security of the AWS Cloud.

[AWS Professional Services](#) is a global team of experts that can help you achieve specific outcomes related to your cloud transformation through a collection of AWS CAF aligned offerings.

Are you Well-Architected?

The [AWS Well-Architected Framework](#) explains the pros and cons of decisions you make when building systems in the cloud. The six pillars of the Framework describe architectural best practices for designing and operating reliable, secure, efficient, cost-effective, and sustainable systems. By answering a set of questions in the [AWS Well-Architected Tool](#), you can evaluate your workloads alongside the best practices for each pillar. This tool is available at no charge in the [AWS Management Console](#).

For more expert guidance and best practices for your cloud architecture—reference architecture deployments, diagrams, and whitepapers—refer to the [AWS Architecture Center](#).

Planning your security journey

> ⓘ Improve your security posture over time. First, implement the security recommendations that mitigate the largest risks, with the least effort. Then, advance your security posture *coherently*, by investing in diverse capabilities to reduce the overall risk as soon as possible.

Today, organizations must innovate quickly. To enable rapid innovation, security teams need to prioritize critical initiatives, business goal focused security risk reduction, and iterate often to improve their security posture over time. With so many security recommendations available, customers often ask: *"How should we prioritize what to do first?"*

In this document, you will find multiple security capabilities. In each capability, we show:

- Which security recommendations are foundational for that capability (*Start*)

- Which ones are a more advanced implementation of that capability (*Advance*)

- How can you get to an ideal state (*Excel*)

Begin with the Start phase of each capability to be comprehensive with your approach, but priorities will vary depending on many factors, including:

- Your security and compliance requirements

- Industry

- Use cases, types of workloads

- Sensitivity of the data managed in the organization

- How critical cybersecurity is for the core business of the organization

And even though we should make every effort to reduce the time that risks are unmitigated, some security controls take more time and effort to implement. Consider what recommendations will strengthen your security posture more quickly—the quick wins.

- **Ease of implementation** - An easy implementation is one that is quick to implement, has lower effort, and lower cost.

- **Increased security benefits** - A higher security posture mitigates *critical risks*, defined as high likelihood of occurrence, and greater impact.

Once you've identified the quickest wins for your organization, plan security activities that will strengthen your security posture and coherence over time. If you are currently running workloads in the cloud, perform a quick assessment to identify gaps, and then start the improvements for each phase:

- **Start** - Important recommendations that form the basis of your security posture, but may take time.
- **Advance** - Recommendations that enable efficient governance of cloud security.
- **Excel** - Recommendations for nearly continuous improvement.

Always strive to maintain the maturity and coherence of your security controls, and plan accordingly. A strong AWS Identity and Access Management (IAM) foundation is important, but if you spend months focused only on defining IAM Governance and refinding IAM policies to least privilege, and you don't have threat visibility in place during that time, you might fail to detect and incident. By enabling Amazon GuardDuty, you can get threat detection capabilities within minutes.

A sample journey with specific guidance can be found in the AWS Security Maturity Model.

Security governance

> **ⓘ Note**
>
> Develop, maintain, and effectively communicate security roles, responsibilities, accountabilities, policies, processes, and procedures.

Security governance defines the organizational structure, roles, responsibilities, accountabilities, and capabilities required to implement security effectively. It provides the ability to drive, track, and report on security controls while maintaining oversight that information security risk is being managed within your respective risk appetite. Successful security governance is a continuous endeavor that must adapt to changing business requirements and threat landscapes in a planned and measurable way.

When done well, security governance will effectively orchestrate and align your organization's security activities, providing security information and insights to enable decision making. Security governance flows from a security strategy that should reflect the priorities of the Board of Directors and have commitment from the Chief Executive. It should be driven by people who understand the business and rely on mechanisms focused on efficiency, visibility, and control; often through automation and simplification.

Adopting modern, automated workflows and a [Zero Trust](#) security model early in software and infrastructure development processes creates a scalable, effective environment. This will help you address relevant legislative, regulatory, and internal policy requirements while also supporting your business objectives.

Start

The starting point for establishing a robust security strategy in your organization involves aligning your security risks with your business objectives. These risks should be mitigated with defined and implemented controls, including detective, responsive, and preventative measures. These controls often manifest as guardrails; automated technical implementations that help your organization operate within its risk appetite and budget. This approach frees up other functions within your organization to focus on their core business, while trusting that decisions made are aligned with the organization's security framework and risk appetite.

An effective security function must be supported by a clear set of roles and responsibilities, such as a *responsible, accountable, consulted, informed* (RACI) model. This model, underpinned by a clear understanding of your security responsibilities in the cloud, forms the basis for your organization's Governance, Risk, and Compliance (GRC). Establishing a GRC model early on outlines critical roles, risk-based requirements, and compliance obligations in a concise and precise manner.

During this initial stage, begin to foster a blameless culture where mistakes are viewed as opportunities for learning and improvement. By doing so, you encourage open communication about potential security risks and incidents and promote a proactive approach to security.

The agility offered by cloud computing and DevOps/DevSecOps practices has opened new avenues for security governance that differ significantly from traditional models. The DevOps mantra to have "everything as code" offers improved visibility and control, reducing friction, costs, and many of the risks associated with legacy solutions. To realize these advantages, begin to introduce new ways of working and modern operating models that are conducive of DevOps with a deep security focus (often referred to as DevSecOps) as you improve your security governance practices.

Security should be considered at every step of the cloud journey. However, incorporating it into the initial design often results in the largest impact. This is because doing so reduces the operational effort to certify that a solution meets the security requirements for launch into production. Incorporating security controls into the development lifecycle, such as the design, build, test, or deploy stages, becomes more cost-effective, more objective, and easier to manage at scale. Development teams benefit from both flexibility and certainty, which are derived (in part) from security policies, processes, procedures, and controls.

Artifacts that can accelerate your security governance journey include:

- A security operating model that defines the "people and processes" aspects of security governance, including responsibilities under the Shared Responsibility Model.
- A security risk appetite statement that defines expectations for different workload categories; not all workloads require the same level of protection and compliance.
- A security controls framework that meets applicable laws, industry and government regulations, and external security frameworks such as NIST, CIS, ISO/IEC 27001, or others.
- A cloud security policy that extends the IT security policy framework and defines new or updated policies to address the improved security management capabilities that cloud computing offers.
- Documented principles that help decision making in the absence of specific guidance.

AWS services (examples):

- AWS Identity and Access Management (IAM) allows you to manage user access and permissions to AWS services and resources, establishing the foundation for secure access control.

- AWS Control Tower helps you set up and govern a secure and compliant multi-account AWS environment. It provides a centralized framework for managing multiple AWS accounts, allowing you to enforce security policies, configure preventive and detective guardrails, and streamline the provisioning and management of AWS resources.

- AWS CloudTrail enables you to log, monitor, and retain AWS API activity, providing visibility into actions taken within your AWS account and supporting compliance and security analysis.

- AWS Config allows you to assess, audit, and evaluate the configurations of your AWS resources, helping to achieve compliance with security policies and best practices.

Advance

Once the fundamental security governance capabilities have been established and embedded, continue to measure, iterate, and improve your governance model. Your governance model must remain agile and flexible, so that it can accommodate changing business goals, risk appetite, compliance requirements, and capabilities. Continuous refinement of your GRC model, driven by these evolving factors, should be expected and encouraged. This includes improvements not only to policies, processes, and standards but also to the broader operating model; the existing security governance capabilities should not be considered static and unchanging as new information and ways of working are introduced.

With your fundamental security framework established, broaden the applicability of the existing security measures across all departments and operations within your organization. Begin the transition to modern security practices within your governance model. These include the implementation of dynamic authentication methods and stringent access controls on all resources into your existing framework. Supplement this approach with continuous monitoring of all systems and network activities, fostering situational awareness of potential threats, and deploying granular control over resource access. This comprehensive and evolving approach is not only necessary for maintaining an effective security posture but also forms the foundation of advanced approaches to security such as the Zero Trust model.

Automation and continuous improvements will help move from a *reactive* to a *proactive* approach, which will help reduce risks and increase the security return on investment (ROI). Having a blameless culture in your organization is crucial to ensuring that lessons are learned from incidents or issues. As collaborative incident response and remediation of vulnerabilities become a recurring

part of your security governance strategy, emphasizing learning and improvement will help drive proactive attitudes towards cybersecurity. Without this culture being present, those in your organization may fear making changes required to improve processes and procedures at the pace required.

Enhancing the effectiveness of your security governance calls for continuous improvements to policies, processes, standards, and the operating model, made possible by regular reviews and amendments. Automation is a key enabler in transitioning from a reactive to a proactive security approach, elevating your security standard, reducing risks, and optimizing the security return on investment. At this phase, automation should move from an aspirational goal to the default mode of operation, reducing manual efforts and increasing predictability.

Automating more aspects of your security policies and procedures makes sure that decisions are made based on near real-time risk assessments. This step to favor automation paves the way for more complex security models. Previously manual tasks, like asset inventory management, require advanced methods at this stage. Use automated tools for asset discovery and tracking to continuously provide a comprehensive view of your asset landscape, aiding risk mitigation and effective security governance. For example, you can use Workload Discovery on AWS to maintain an inventory of AWS resources across your accounts and Regions, mapping relationships between them, and displaying them in a web UI.

Capabilities that are measured and reported on provide assurance that security investments are delivering value and mitigating risk. Establish a reporting framework that gives your management team confidence that security risks are being managed effectively and within the agreed risk appetite. Make use of the cloud's capabilities to automate the production of report data. This practice not only saves time but also boosts confidence in your metrics and provides a single chain of reporting based on a single data source. Use AWS Security Hub to automate security best practice checks, aggregate security alerts into a single place and format, and understand your overall security posture across all of your AWS accounts.

Excel

An organization that excels in security governance leverages the cloud to maintain control over its cloud-based assets and operations. Decisions are made with confidence, powered by reliable cloud metrics (such as incident response and vulnerability resolution metrics), which guide both investment decisions and performance evaluations of previous investments. These organizations can readily navigate changing business goals, evolving customer requirements, the shifting threat landscape, and large-scale technical modifications.

Regularly re-evaluate your security model so that roles and responsibilities align with evolving business goals. It's common to switch from a project-based approach to a product-based mentality, focusing on the needs of various stakeholders, including business units, development teams, and cloud platform teams. At AWS, we embed security within our engineering teams by developing our builders to think like security practitioners. By empowering these security-minded Amazonians called *Guardians*, we foster a culture of informed security ownership throughout the development lifecycle. The goal is to have a security presence within the engineering team that allows for quick decision-making. This helps reduce hand-off of security responsibilities, while spreading security knowledge throughout the organization.

Expand upon the effective asset management strategies already in place. At this stage, your organization should be using advanced technologies like AI and predictive analytics to track asset usage patterns and detect potential vulnerabilities before they arise. This provides a holistic view of your security landscape and equips you with actionable insights for strategic decision-making based on historic events.

Prioritization of security efforts should continue to hinge upon a risk-based approach. Use risk metrics to forecast demand for security services and resources. Such a proactive approach not only drives more informed investment decisions but also enables your team to preemptively meet customer needs. Continual monitoring of the consumption and effectiveness of your security services provides valuable insight into areas that are performing well and those needing improvement.

Excel at automated governance by codifying your security policies and processes. By doing so, you can facilitate continuous measurement of automated controls effectiveness over time, assuring that improvements to security posture can be reported accurately. Embed and automate risk management processes at each stage of every workload's lifecycle by defining risk thresholds that can be measured, and allow teams to request exceptions where required. Embracing a continuous risk management approach contributes to the overall risk management processes of your cloud ecosystem, fostering a proactive rather than a reactive approach to security governance.

By integrating automated governance practices, your organization can provide consistent compliance, reduce manual effort, such as those required by approval boards, and enhance the overall security of your systems and data.

In addition to control automation, AWS uses weekly, monthly, and quarterly meetings and reports for communication of risks across all components of the risk management process. AWS also implements an escalation process to provide management visibility into high priority risks across

the organization. All these efforts manage risk consistently within the complexity of the AWS business model.

AWS services (examples):

- Amazon Macie: Amazon Macie is a data security and privacy service that uses machine learning and pattern matching to discover, classify, and protect sensitive data stored in AWS. It helps organizations identify and secure their sensitive data, such as personally identifiable information (PII), intellectual property, and financial data, by automatically recognizing sensitive data patterns and providing alerts and recommendations for securing the data.
- AWS Systems Manager (SSM): AWS Systems Manager is a management service that helps organizations manage and automate operational tasks in their AWS environment. It provides a unified interface for managing resources, configuring operating systems, automating patch management, and collecting software inventory. SSM also includes the Parameter Store, which securely stores and manages configuration data, secrets, and other sensitive information that can be securely accessed by authorized AWS services and applications.

Security assurance

> ⓘ Continuously monitor, evaluate, manage, and improve the effectiveness of your security programs.

Your organization and the customers you serve need trust and confidence that you have implemented adequate security and data protection measures to meet your legal, regulatory, and compliance obligations. In the cloud, compliance responsibilities are shared with your Cloud Services Provider (CSP). *Security assurance* implies that commitments made to, or from, your organization have sufficient evidence that the required level of trustworthiness has been achieved. Commitments take many forms, including data confidentiality, availability of a service, and accuracy of processing. They include promises made to customers, suppliers, AWS partners, and employees. The most successful security assurance programs can provide that assurance on a continuous basis, and indeed deliver automated responses when failures are known or predicted.

All organizations require security assurance. Some may also have legal, regulatory, or contractual compliance requirements that are best met from evidence produced by the assurance activities. While compliance is a critical milestone, it isn't an end state. Your teams must integrate security and privacy assurance throughout the entire capability development lifecycle.

Start

Start by reviewing all the regulatory requirements that your organization must comply with. These can include global, local, industry, or sector-specific security and privacy regulations. Document, communicate, and track your requirements from a business risk management perspective. As you perform this exercise, determine if each of the regulatory requirements has been updated to enable the use of cloud services. Requirements that haven't been updated to reflect the differences between traditional on-premises environments and the cloud can hinder adoption. If you identify regulatory requirements that aren't cloud friendly, engage with an industry association or directly with the regulator to identify the problem. Then, you can propose changes to evolve and update the requirements.

Next, identify the controls you must implement. AWS assurance programs provide [templates and control mappings](#) to help customers establish the compliance of their environments running on AWS. Review these and document your controls into a comprehensive framework. Once

you've established security and privacy controls to meet your requirements, develop enterprise-wide policies and standards to set expectations. Then, review to determine if the controls are implemented and operating effectively. You can use self-assessments, self-attestation, internal audits, or independent assessments by third parties to verify requirements and identify gaps or risks.

Traditional security assurance approaches involve gathering samples of evidence, sample testing performed by auditors, and evaluating the narrative of how systems work. This is often manual, periodic, time intensive, and can lack technical depth. This leads to reports being often out-of-date before they are published. The challenge with traditional assurance approaches is that systems are constantly changing, and it could take months or even years to manually assess controls for every scenario to verify control statements. Once your organization reaches this threshold of complexity, you must transition to the *advance* level of security assurance.

Advance

Compare and evaluate your traditional on-premises environments and security approaches to those enabled by the cloud. Many customers quickly determine that regulated and audited data is better suited for the cloud. Review your organization's security and compliance program to determine the number of compliance programs where you use independent auditors. Auditors can validate the presence and operation of controls and provide you with associated reports, such as a SOC 2 Type II compliance. Review the audit reports, compliance certifications, or attestations that your cloud vendor has obtained. This will help you understand the controls they have in place, how they have been validated, and that they are operating effectively. Compare your on-premises processes with AWS; note that AWS regularly achieves third-party validation for thousands of global compliance requirements.

The scale of AWS allows us to embed security capabilities and invest more than most large companies could afford themselves. We provide an environment that gives customers comprehensive visibility and control over their cloud environment and data that simplifies compliance identification, tracking, and reporting. Investigate how AWS can help your organization, security teams, regulators, and auditors get the technical depth and understanding to move regulated workloads to the cloud.

If you are using a non-AWS Cloud Service Provider (CSP), determine if the CSP undergoes security assessments by independent third parties, including the physical and environmental security of its hardware and data centers. Make sure that they've obtained certifications and accreditations from recognized accreditation bodies. To confirm controls are implemented and effective, review

the CSP certification and accreditation documentation. This will give you assurance that the environmental security, security practices, and CSP cloud environment support your needs. By operating in an accredited cloud environment, you can reduce the scope and cost of audits you must perform. You can use the CSP's certifications and inherit security controls to simplify and streamline your compliance.

As your company matures and advances its security assurance capabilities, there will most likely be changes in your environment, and you'll need a higher frequency of assurance activities. These provide you with an ability to detect drift and reduce the chance that you'll miss deviations. Use tools and automation capabilities to continuously monitor and evaluate your environment to verify the operating effectiveness of your controls and demonstrate compliance with regulations and open standards.

Operating in an AWS environment, allows customers to take advantage of embedded, automated tools such as [AWS Audit Manager](#), [AWS Security Hub](#), [AWS Config](#), and [AWS CloudTrail](#) for validating compliance. Use these tools to reduce the effort needed to perform audits, and make these tasks routine, ongoing, and automated. Evolve the role of compliance in your company from one of a necessary administrative burden, to one that manages your risk and improves your security posture.

Apply cloud-specific verification techniques to your audits. Contemporary control frameworks and language are focused towards on-premises environments, and your security IT auditing techniques must be updated for the cloud. Prepare for auditing security in the cloud by identifying the differences between auditing in the cloud and on-premises. Provide your team and your auditors with education and tools to audit for security in the cloud using a risk-based approach. ISACA has launched the [Certificate in Cloud Auditing Knowledge (CCAK)](#), a vendor-neutral technical training, and credentialing for cloud auditing. The [AWS Cloud Audit Academy](#) program enables organizations to establish common audit knowledge between customers and external IT auditors. You'll be able to apply security auditing best practices and use AWS services to assess industry-recognized frameworks, standards, and statutory regulations. This will help reduce time-to-market for regulated workloads.

Excel

As you reach the *excel* stage, your organization will be the most mature. You now build security and privacy compliance into all business processes. You also monitor legislative and regulatory developments to proactively understand and address new regulatory requirements. And, you partner with regulatory entities and associations to proactively shape positive, enabling

regulations. You demand that regulators and auditors become familiar with the advanced tools and security that you're using. You contribute to and use existing security compliance frameworks and controls instead of building frameworks for a particular sector or industry. Your environment is based upon infrastructure, policy, and compliance as code, with immutable infrastructure, detailed logging, and anomaly detection.

Your organization will move away from traditional assurance methods; they will be less effective as your complexity grows. You're implementing new assurance models, and embedding application security reviews and comprehensive verification and validation (V&V) review, analysis, and testing. These reviews check that the requirements are correctly defined, and validate that the security requirements have been met. You use emerging technologies, such as automated reasoning and provable security to perform assurance activities.

Automated reasoning infers the future behavior of computer systems, considering all possible actions, requests, and configurations, and provides the highest levels of security assurance. For example, the AWS Automated Reasoning Group (ARG) is developing mathematical proofs of certain aspects of a system. A mathematical proof might be used to prove that there's *no* instance of a weak cryptographic key being used anywhere in the entire system. In this case however, the objective of contemporary audits is to achieve "reasonable assurance". In the contemporary approach, auditors can't evaluate all the code or all of the instances where keys are being used. Automated reasoning provides much greater assurance, as the mathematical proof can examine the entire system. This provides a much higher bar than today's most advanced control measures, such as automated controls, preventive controls, or detective controls.

You also use AWS services that have integrated ARG capabilities such as AWS Zelkova. This service provides insight into access control permissions and uses automated reasoning to analyze policies and the future consequences of policies. This includes AWS Identity and Access Management (IAM), Amazon Simple Storage Service (Amazon S3), and other resource policies. These policies dictate who can (or can't) do what with which resources. AWS Zelkova is used by Amazon Macie, Amazon GuardDuty, Amazon S3, and AWS IoT Device Defender to automatically derive the questions you ask about your policies, which increases confidence in your security configurations. Another example is AWS Tiros, which maps connections between network mechanisms, including open internet connectivity. It checks all network pathways and data permission levels in milliseconds. Amazon Inspector uses VPC Reachability Analyzer to determine network reachability. AWS offers tools and services that provide higher levels of assurance about the security of your infrastructure and data than traditional approaches or on-premises tools.

Identity and access management

> **ℹ Note**
>
> Securely manage human and machine identities and their permissions to cloud services and resources.

Identity and access management determines who has access to what in AWS. You need robust identity and permissions management to make sure that the right people, machines, and services have access to the right resources under the right conditions.

In this section, we're focused on how people and machines access your AWS accounts and resources; identity and access management for the applications you deploy in AWS is a topic for another paper.

The scope of this topic extends beyond the AWS Identity and Access Management IAM service itself, to include all of the identity-related controls and features in AWS. Even if you don't use every capability, a comprehensive perspective will verify that your identity and access management practices remain scalable and responsive to business needs.

Identity and access-related decisions for AWS will be made continuously by a variety of stakeholders throughout the enterprise. Define and communicate a strategy to inform those decisions. Your strategy might include:

- **Objectives** - The desired outcomes of your IAM strategy
- **Tenets** - Guiding principles so that delegated decisions support your objectives
- **Delegation model** - To align stakeholder understanding of roles and responsibilities for access management (such as a RACI matrix)

Sample objectives

- Provide a safe place for builders to experiment, fail, and innovate at speed. This includes tools to quickly determine how and why their access is limited, and where to get help when needed.
- Minimize the possibility of over provisioning permissions and enforce separation of duties to satisfy risk appetite, best practices, compliance, and regulatory requirements.

- Focus your security experts on high-impact outcomes that only humans can deliver; automate the rest, such as guardrails checking roles and policy creation and changes.

- Nearly continuous compliance: automate attestation and reporting, to deliver on-demand the insights that auditors and leaders need. An example is security in the CI/CD pipeline, and a Professional Services offering for Automated Continuous Compliance focused on GxP in Healthcare and Life Sciences.

Sample tenets

- **Guardrails, not gates** - Design controls to minimize friction around AWS Identity and Access Management in AWS. Use centralized, blocking, manual workflows only when there is no safe alternative. Strive for self-service.

- **Security as code (SaC)** - Publish your security baseline as consumable code. Enforce governance in the toolchain. Make the best developer experience a side effect of strong security outcomes.

- **Adopt the principle of least privilege** - This is a process of continuous improvement. Start simple and iterate. Use automation, detective controls, and analytics to continually rightsize permissions for each role and environment.

- **Keep people away from data** - At each stage of your software development lifecycle (SDLC), implement IAM policies, mechanisms, and tools to decrease the need for direct access to data stores.

- **Use temporary credentials like AWS IAM roles by default** - Static users, passwords, and access keys are a last resort, issued by exception only. Where static credentials are used, secure them in a purpose-built secrets management system and rotate them regularly.

Sample delegation model

Delegation of responsibility for overlapping access controls is key to achieving defense in depth. Using AWS IAM, you can avoid delays, mistakes, and missed opportunities that may come with fully centralized access management. As you consider who should be responsible for what, and where to implement particular controls, keep in mind the following:

- Access management for *people* needs a different approach than access management for *machines and services*.

- Changes should be approved and made by people with the best knowledge of the business and data being protected.

- Agility and scalability require that developers and data custodians understand [permissions management in AWS](#).

- Permissions policy space in AWS is finite – use the best policy type for each control and make efficient use of all available policy types.

  Policy maximum size is one the few AWS service quotas that cannot be increased from defaults. For more details:

  - Watch [Choosing the Right Mix of AWS IAM Policies for Scale](#) (video)

  - Read [IAM Policy Types - How and When to Use Them](#) (blog)

Review sample access controls delegation model (R = responsible) in the following table.

AWS access control mechanism	Central cloud governance team	Central cloud IAM/ engineering team	Central security team	Workload / resource owners	AWS account owner
Organization structure	R				
Organization policies (SCPs, AWS Control Tower Guardrails)	R				
Identity policies for baseline human access		R			
Identity policies for baseline account		R			

AWS access control mechanism	Central cloud governance team	Central cloud IAM/ engineering team	Central security team	Workload / resource owners	AWS account owner
controls & services					
Permissions boundaries			R		
Resource policies for workloads				R	
Identity policies for workloads				R	
AWS account root password			R		
AWS account root MFA token					R

The example delegation model in Table 1 shows separation of duties, which makes it hard for any lone actor to circumvent all of the access controls that protect your AWS environment. Your teams may vary, but *no single team or person* should have authority over all of the access controls shown.

Use AWS services for orchestration

AWS often offers more than one solution to common problems. Sometimes this arises from customer feedback, which leads to the launch of new higher-level features to orchestrate existing lower-level ones. Two such examples are AWS Control Tower and AWS IAM Identity Center, which are offered at no extra charge. The costs of a custom solution will outweigh the benefits for most customers in the long term, especially as AWS continues to innovate.

Start

Bootstrap identity and access management for your AWS environment.

Credential management

AWS account root user management is the first capability needed to secure your AWS accounts. Your most sensitive AWS account is the one designated as your AWS Organizations management account and *access to it should be strictly controlled*. All AWS account root users, *especially* the management account root user, need a strong password in addition to multi-factor authentication (MFA). AWS account root users must be protected with due care and should be controlled by two people. For example, the individual or team that controls the root password of an AWS account should not also control its MFA token. Use a trusted password/privileged access management (PAM) solution to secure these secrets. Understand and periodically exercise the procedures for resetting AWS account root passwords and root MFA devices.

Add MFA immediately to all accounts that allow password-based login to AWS. Use your existing enterprise MFA solution, if possible. For AWS account root users and IAM-users;, configure MFA in AWS IAM. For federated users, configure MFA in your existing identity provider (IdP) or in AWS IAM Identity Center. Treat static AWS IAM credentials as toxic and prohibit them by default. People access AWS through federated authorization. Evaluate AWS IAM Roles Anywhere for on-premises or other non-AWS systems that need access to your AWS environment. Question vendors whose products require static IAM-users; and access keys; issue them only as a last resort and under a formal exception process, with automatic expiration.

Federation

To do anything in AWS, you first must grant access to the people who will build the foundation (landing zone) on which everything else is deployed. Federate your IdP with AWS. In this way, any adaptive authorization, entitlements management, compliance, and reporting capabilities associated with your IdP will apply to AWS as well. The recommended approach is to connect your IdP to AWS IAM Identity Center. AWS IAM Identity Center expands the capabilities of AWS Identity and Access Management (IAM) to centralize the administration of workforce access to AWS accounts. With Identity Center, you administer all human users and their permissions for your AWS Organizations in one place.

When enabled in your management account, AWS IAM Identity Center is deployed in the currently selected AWS Region only. This choice does not limit access to other AWS Regions. However, if your chosen Identity Center Region were unavailable, your users would be unable to access Identity

Center to authenticate to AWS or other applications. Consider the risks of such an event and create a backup access plan that satisfies your resiliency needs.

Whatever federation solution you choose, you must define the job functions, roles, and responsibilities that team members will use to do their work in AWS. Start simple and iterate. In the beginning, no more than two to three roles should be necessary. If you use AWS Control Tower to deploy a landing zone (see the following section), you can get started by using its default roles. Keep in mind the YAGNI principle and resist the urge to create several job or task-specific roles until you know how they'll be used. Role and policy sprawl can quickly become technical debt that can weaken your security posture and complicate audit and compliance matters.

Landing zone

Each AWS account is a security boundary for the identities and resources that it contains. This is why you need a multi-account strategy as described in the AWS whitepaper Organizing Your AWS Environment Using Multiple Accounts. Create a landing zone by configuring a multi-account environment according to AWS best practices and your business requirements. Workloads are then deployed atop the landing zone.

For every AWS account in your landing zone, you'll need a consistent baseline configuration including a default set of permissions for people and services. Some elements of the baseline will be constants, but overall composition will vary depending on the purpose of the target AWS account. For example, non-production accounts should allow developers freedom to experiment, while production accounts should restrict all unnecessary services and actions. In either case, AWS CloudTrail should be enabled in every account, regardless of its purpose. Enforce using various types of AWS IAM policies that work together with orchestrated management.

The preferred way to deploy and manage your landing zone is using AWS Control Tower. Control Tower automates AWS account creation and baseline configuration. It gives you a practical set of default permissions to start, and consistent governance across all of your AWS accounts as your needs evolve. Follow guidance in this AWS whitepaper Organizing Your AWS Environment Using Multiple Accounts. This embodies AWS best practices based on years of feedback and learning from our most successful customers.

Guardrails

Service Control Policies (SCPs) are a powerful tool for simplifying permissions management across your AWS Organizations. SCPs restrict permissions at the AWS account level, unlike other IAM policies that apply to individual identities or resources. To make efficient use of SCPs, you must

organize your AWS accounts in a hierarchy of organizational units (OUs) and apply SCPs to those OUs.

Use SCPs for coarse-grained controls that don't change often. SCPs can impact multiple AWS accounts, and modification requires access to the management account. Therefore, updates should be strictly controlled, tested to the extent practical, and relatively infrequent as mistakes could interrupt all users and workloads in the specific OU.

A good SCP use case is to restrict access to AWS Regions where your company doesn't operate. You can implement Regional restrictions directly in SCPs or use the Region Deny control in AWS Control Tower. Another good use of SCPs (and AWS Control Tower default) is to prevent local changes to resources that are deployed as part of your configuration baseline. Refer to AWS Control Tower recommended controls and AWS Organizations example SCP's documentation for more information.

Start using SCPs immediately. The risk of deploying major control changes via SCPs can be high, so it's best to start simple. Take an iterative approach to both the controls you implement and associated testing and deployment procedures. That way, when you need to make significant control change using SCPs, you'll have the mechanisms in place to do it safely and confidently.

Advance

Establish programs, procedures, and automation to govern identity and access management.

Metrics and monitoring

As soon as your initial landing zone is deployed, begin measuring your identity and access controls in AWS. Understanding current state (benchmarking) is the first step toward continuous improvement. It's important to measure cost-effectively, objectively, and consistently. Focus on fundamentals and measure things that are relevant to your security decision makers.

AWS Foundational Security Best Practices Standard (FSBP) is a good place to start. FSBP defines IAM best practice controls that should be monitored in all of your AWS accounts. Compliance with FSBP IAM controls alone is not sufficient. Non-compliance means that key controls are either missing or ineffective and in need of prompt attention. Monitoring FSBP compliance is a push-button operation in AWS Security Hub.

Create an AWS IAM Access Analyzer with your AWS Organization as the zone of trust, then monitor the findings. Access Analyzer findings show when your AWS resources grant access to

external principals. In other words, Access Analyzer findings help you identify security risks due to unintended access.

An Access Analyzer finding indicates that the resource in the finding allows access to an external entity. (Access Analyzer findings do not tell you if external access has occurred.)This could be intentional, such as access granted to a vendor or partner, or it could be unintentional. It's important to investigate all active findings promptly so you can correct misconfigurations and prevent unauthorized access. Use archive rules to hide findings related to expected, authorized access.

Implement an organization-wide detective control that alerts on logins by AWS account root users. Triage each alert immediately. Any alert that can't be correlated to an approved change request should be escalated immediately to your security incident response team. Establish a program to remove all unnecessary root access, including weekly reporting and reviews by your CISO to drive compliance.

Security as code

Security as code (SaC) is an extension of the infrastructure as code (IaC) concept, which says that security controls are codified in templates and managed as software. When modified templates are committed, automation verifies, tests, and deploys the updated controls to your AWS environment in a reliable, repeatable way.

As reflected in the sample delegation model shown in Table 1 preceding:

- Developers manage IAM resources for their application as part of its infrastructure code.

- Separate, central teams manage IAM resources that span your AWS environment (for example, SCPs, permissions sets, baseline roles, and policies) as standalone workloads or as components of a *security baseline* workload.

This approach lets you apply consistent governance via continuous integration/continuous deployment (CI/CD) toolchain automation, without the concentration risk of a central team that creates or approves all AWS permissions. To begin, you must get all IAM resources (for example, roles, policies, service configurations, and others) codified and stored in an enterprise-managed version control system. Ideally, your version control system should support fine-grained access control, code review, approval workflows, and simplified CI/CD integration. Now that you're relying on a decentralized IAM model, use those features to implement preventive controls that enforce separation of duties and stop policy violations before they reach AWS.

For example, consider a version control repository that contains templates defining IAM policies/permission sets used by common roles in all your AWS accounts. Block all direct commits to the repository by default and require merge requests to be peer reviewed before being merged. Configure the merge process to initiate a pipeline that uses AWS IAM Access Analyzer policy validation to check the updated IAM policies, blocking the pipeline if problems are found. When the problems are resolved by a subsequent commit, the pipeline can continue through remaining stages and deploy the updated policies.

Permissions boundaries (PB) enable you to safely delegate authority over IAM roles. A PB is an advanced policy type that sets the maximum permissions available to an identity, regardless of what's allowed in any other attached policies. Because PB can be independently controlled, you can use them to enforce conditions that can't be overridden elsewhere by delegates. Prevent over permissive access creation by developers in the identity policies. Grant them permission to use IAM, but add conditions requiring attachment of a specific PB to any role they create or update. If the PB is missing, AWS IAM will prevent the change.

Use AWS Cloud Development Kit (AWS CDK) (CDK) to generate the roles and policies needed by your applications. You can attach a permissions boundary to all roles in an application with just a few lines of code. CDKs Construct Programming Model also lets you create a library of AWS resources customized to your security policies and environment, which developers can then consume as usual. Include cdk-nag rules in your constructs to encourage best practices. CDK is a powerful tool for decentralized governance enforcement.

Least privilege automation

At scale, least privilege is a pursuit that requires automation to drive continuous improvements. Take an iterative approach, using preventive and detective controls to rightsize permissions throughout your SDLC.

To refine permissions for a role, use AWS IAM Access Analyzer policy generation to generate a policy template based on the role's access history, as captured in AWS CloudTrail. Then use AWS IAM Access Analyzer policy validation in your CI/CD pipelines to validate policies against grammar and best practices, before deployment. Together, these features take the guesswork out of creating initial permissions and periodically rightsize them to maintain least privilege.

AWS IAM features like credential reports, last-accessed information, AWS CloudTrail integration, account summary, and account authorization details provide valuable data about identities, permissions, and access history in AWS. You can also collate, enrich, and analyze outputs using services like Amazon Athena, AWS Glue, and QuickSight to identify more opportunities to rightsize

permissions and strengthen your security posture. Products from AWS Partners such as Sonrai Security and Ermetic can go further to produce sophisticated, identity, and access related insights that can help make faster, risk-informed decisions.

Excel

Focus on efficiency, continuous improvements, and operational excellence.

Zero Trust

AWS has long led with a Zero Trust approach to security, using Transport Layer Security (TLS) and SigV4 to authenticate and authorize every single API call, regardless of its origin. AWS capabilities enable you to implement Zero Trust concepts for your part of the AWS Shared Responsibility Model. But Zero Trust is not a universal solution. For more details:

- Read Zero Trust architectures: An AWS perspective (blog)
- Watch Zero Trust on AWS: Steve Schmidt, VP of Security Engineering & CISO, AWS (11:12) (video)

Security controls should be chosen based on the use case and business value of the assets to be protected.

Data perimeter

Build a data perimeter to protect your AWS accounts and resources. A data perimeter overlays your existing, fine-grained controls with coarse-grained ones that allow access only if a request exclusively involves trusted identities, trusted resources, and expected networks. To establish a data perimeter, implement preventive controls that restrict access from outside of your AWS Organizations boundary. Those permissions are applied primarily using SCPs, resource-based policies, and Virtual Private Cloud (VPC) endpoint policies. For detailed guidance and examples, see the AWS whitepaper Building A Data Perimeter on AWS.

ABAC

As you scale and implement more sophisticated authorization strategies, some permissions management use cases can be simplified by using attribute-based access control (ABAC). ABAC is part of IAM, and you can use ABAC, role-based access control (RBAC), or both together based on your needs. Implement ABAC in AWS by attaching tags to IAM entities like roles and users, and to resources like Amazon Elastic Compute Cloud (Amazon EC2) instances or AWS Key Management Service (AWS KMS) keys. Then write policies that allow or deny access, based on the values of

those tags. In the case of a data perimeter, use ABAC to exempt specific users or roles from specific controls, without making high-risk policy changes.

If consistent resource tagging is already in place, ABAC will be easier to implement. Use tag policies and detective controls for enforcement. Don't wait until you need ABAC to get started with tag governance. With ABAC, tags are a *key* element of your authorization security controls. Implement trust policies over your IAM roles. Tag values can be set statically on IAM entities or you can use session tags to set values dynamically. For federated/human user access, AWS IAM Identity Center provides the most straightforward use of ABAC and session tags across all of your AWS accounts and Region

Vulnerability management

> **ⓘ Note**
>
> Automate the identification, classification, remediation, and mitigation of security vulnerabilities.

Currently, tens of thousands of security vulnerabilities exist, with more being discovered on a regular basis. With this growing threat it is critical to identify vulnerabilities, assess the vulnerability's impact to your business and security posture, and determine course of action. Some actions should be automated, yet many require a risk decision based on downstream impacts of updates, upgrades, linked systems and end-user impacts.

Organizations should strive to build a vulnerability management program that addresses your top security and business risks, and where possible automatically discovers and routes vulnerability findings in near real-time to the appropriate teams. They can then take immediate action using up-to-date common vulnerabilities and exposures (CVE) information. Review network accessibility to create context-based risk scores that help prioritize and address vulnerable resources. Scan AWS workloads for software vulnerabilities and unintended network exposures. As your environment grows and evolves, so should your vulnerability management program. Over time, procedures should be documented, tested, and shared with stakeholders. We encourage you to use automation in every step of your vulnerability management process.

Start

Your environment is a key factor in building the foundations of a vulnerability management program. Organizations should know where vulnerabilities exist, how they increase the threat surface, and avoid the risk of an unknown asset introducing vulnerabilities into the environment. Understand your assets and the asset lifecycle within the environment. Consider the following:

- How are assets provisioned onto the network?
- Are assets created via API call with traceability and explicit permissions?
- What security agents must be installed and configured on those assets?
- How are assets deployed consistently from a secure and approved known good configuration, such as Amazon Machine Images (AMIs), containers, Lambda functions, and IaC templates?

- How will those assets be monitored and tracked throughout their lifecycle for vulnerabilities?
- How will identified vulnerabilities be patched and remediated?

On AWS, there are multiple services and scaling mechanisms available, including partner solutions. These can help you gain a better understanding of your resources and the vulnerabilities affecting them. For example, you can use AWS Config to generate an asset inventory of your resources and monitor your configuration state. You also can deploy managed or custom rules to detect resources that fall out of configuration compliance. You can use Amazon Inspector for automated and scheduled vulnerability scanning at scale across EC2 instances, container images stored in Amazon Elastic Container Registry (Amazon ECR), and AWS Lambda functions. All Amazon Inspector findings are prioritized and continuously updated. EC2 Image Builder can be used to create patched, hardened, and approved AMIs from which to deploy your instances. You can also use tags to identify and classify AWS resources, and use AWS Lambda and Amazon EventBridge to automatically tag new EC2 instances that are launched manually or via Auto Scaling mechanisms. Finally, Resource Groups and AWS Systems Manager can be used to organize, manage, and automate tasks on many resources at one time, either on-demand or with a defined schedule.

In the early stages of building a vulnerability management program, start to think about roles and responsibilities, and draft runbooks to remediate identified vulnerabilities. Some considerations might include:

- Who is responsible for creating, managing, and updating the AMIs?
- Who is responsible for managing and monitoring the resource inventory and configuration compliance?
- Who is responsible for managing the vulnerability scanning solution?
- What procedure will be followed for remediating identified vulnerabilities?

Advance

Customers can identify and manage their resources and continuously scan those resources for vulnerabilities. Use a different remediation playbook for each operating system type and have clearly established maintenance windows. Automate alerts to notify resource owners of vulnerabilities, and add human testing to the automated scans that are already taking place. For example, customers can use red teaming exercises and penetration testing to identify vulnerabilities in the system architecture that may have been missed by the automated scanning tools.

On AWS, services to consider include AWS Systems Manager, which allows customers to create playbooks, maintenance windows, and automated patching strategies. These can be applied to specific operating systems or resources based on tag value. AWS Systems Manager Patch Manager and AWS Systems Manager Run Command can help perform patching and run commands on an entire fleet of instances. Amazon EventBridge, Amazon Simple Notification Service (Amazon SNS), and AWS Lambda can orchestrate alerting and remediation workflows. This reduces the mean time to respond and recover from identified vulnerabilities. Since AWS services are tightly integrated, Amazon Inspector findings can easily be forwarded to an EventBridge event bus. This takes finding details and sends them to any downstream system, such as alerting workflows or Lambda functions for auto-remediation. Finally, Systems Manager provides an audit trail of all actions, patches, and maintenance tasks, which is a requirement in many compliance frameworks, standards, and regulations.

In the *Advanced* stage of building a vulnerability management program, consider regular testing, performing Game Days and table top exercises, formalizing remediation playbooks, and automating most processes. Some examples might include:

- Performing zero-day vulnerability exercises.

- Measuring key performance indicators on a continuous basis, such as mean-time-to-detect, respond, and recover.

- Determining where you can apply force patching to workloads in non-production environments to ensure readiness for production-level patching. An example is creating a remediation playbook within Systems Manager that automates patching during defined maintenance windows.

- Regularly training non-security personnel on company policy and security best practices.

Excel

Organizations that want to *excel* in vulnerability management will have:

- Both manual and automated mechanisms in place to identify vulnerabilities continuously (scanning tools and penetration tests)

- Regularly tested remediation playbooks driven by automation pipelines

- Accurate and near real-time visibility into resources on the network and vulnerabilities affecting those resources

- Key performance indicators and metrics established to measure the program's performance and improvement over time
- Well-defined roles and responsibilities with regular employee training
- Clear vulnerability management standards and policies that are adopted across the organization

Mature vulnerability management programs include stakeholders outside the core security team, such as the governance, risk, and compliance (GRC), internal and external audit, application developers, product managers, and infrastructure engineers. A vulnerability management policy should be created that clearly outlines what the process is to identify, detect, respond, and recover from vulnerabilities. It must show what is expected from application teams that are provisioning and managing resources on the network, and have remediation service level agreements (SLAs) to follow for various operating system types.

As your environment grows, AWS recommends having a single security account within your AWS Organizations from which to run your security tooling. A delegated administrator account will perform administrative functions for the AWS Services, and may include the security account based on your security services. With this security account in place you can send all vulnerabilities, findings, and logs to this account for centralized security monitoring and alerting. For example, if you're using Amazon Inspector for vulnerability scanning and you have 100 AWS accounts to manage, delegate Amazon Inspector administration to the security account and enable from the Amazon Inspector console. This allows your security team to monitor vulnerabilities from one place rather than having to login to each account, and follows best practices for a multi-account environment. Additionally, distributed security ownership implies that each of your workload teams should have visibility to the vulnerabilities and findings in order to address them directly.

Lastly, regardless of the tooling that is implemented, you should enforce access controls that prevent account users from disabling or modifying vulnerability management tooling and workflows. For example, make sure that users are unable to modify Amazon Inspector, AWS Config, and Systems Manager settings if those services are being used. This can be achieved through IAM policies attached to the user, group, or role, or by implementing SCPs within AWS Organizations.

Review the resources, services, best practices, and partner solutions that have been widely adopted in thousands of other environments to guide your implementation.

Infrastructure protection

> **ⓘ Note**
>
> Confirm that systems and services within your workload are protected against unintended and unauthorized access and potential vulnerabilities.

Protecting your infrastructure from unintended and unauthorized access and potential vulnerabilities will help you elevate your security posture in the cloud. This includes protecting your AWS network and computing devices to detect, contain, and stop unauthorized users. In traditional data centers, you are responsible for maintaining and protecting *all* IT infrastructure. But AWS uses a shared responsibility model where both AWS and the organization share in securing the infrastructure. AWS is responsible for protecting the infrastructure that runs all of the services offered in the cloud. Your organization is responsible for the management of the guest operating system (including updates and security patches) and other associated application software and the configurations, such as firewalls. Your responsibility also depends on the services that you select. For a service like EC2, you are responsible for managing the guest operating system (including updates and security patches), any application software or utilities installed by you on the instances, and the configuration of the security groups. But for a managed service like Amazon S3, you are only responsible for managing data, choosing your encryption option, and setting up appropriate permissions.

Traditional data centers use boundary-based protection methods that define the trust boundary. They filter and restrict anything entering the trusted zone by using a stack of security solutions such as firewalls, intrusion detection systems (IDS), and intrusion prevention systems (IPS) to block malicious traffic at the boundary of the network. In contrast, infrastructure security in a cloud environment like AWS, can follow a Zero Trust model where security is applied at multiple layers. With a Zero Trust model, you don't automatically trust anyone within or outside the network. Instead, you enforce fine-grained identity-based authorization rules. This will provide a least privilege model where all the entities are given only necessary permissions to perform their duty. For example, if an unauthorized user was able to access the system using an open port, they couldn't read or modify data from a database because they would not have sufficient permissions. To reduce the risk of unauthorized users causing disruptions to your cloud deployments or gaining access to your data, use defense in depth with a Zero Trust model.

Start

We recommend using AWS Organizations in your landing zone environment that supports your multi-account strategy. Deploy security controls uniformly across your environment, and define the foundational shared services that will be used. Include network services to centrally deploy and manage your virtual private clouds (VPC) and subnets, and share them with other accounts in the environment using AWS Resource Access Manager (AWS RAM). The shared services design should also include an account or accounts designated to deploy centrally managed security services. These can be delegated admin consoles for AWS Config, Amazon GuardDuty, Amazon Macie, IAM Identity Center, and AWS Security Hub. Include an account where logging services can be centralized and logs can be aggregated. Consider shared services accounts for any other services that will be used by the other member accounts in the organization. Set up the organizational units (OUs) in AWS Organizations to create an area for the shared services account to be deployed in. You can then apply preventative and detective guardrails and other governance elements. This will give you the ability to separate accounts, such as business application workload accounts and sandbox accounts.

Next, define your infrastructure segmentation design. This includes your VPC and subnet design patterns with associated classless inter-Domain routing (CIDR) IP address ranges, routing tables, and gateways (network address translation, internet, and transit gateways). Design your infrastructure segmentation to support an n-tiered application architecture. Enforce the separation of the data tier from the business logic tier, from the presentation tier, and from other tiers as appropriate. This segmentation design should also support elements of a Zero Trust strategy with a default deny mechanism. Network communications should be restricted to only what is needed for applications to function appropriately to enable business operations. This may require an analysis of application data flows and architectural design to understand the requirements for infrastructure segmentation to be designed properly.

Now that you understand your shared services and segmentation needs, design your routing tables, network access control lists (NACLs), and security groups to support and enforce them. Use route tables to connect your subnets as needed for application data flows to other subnets and their gateways. Use network ACLs to control access to your subnets. And use security groups to control traffic to EC2 instances or other network interfaces in your subnets.

Routing tables are a set of rules used to determine where traffic flows. This could be needed when a network communication request is destined for a network interface that is *not* located in the same subnet. The design of the routing tables should enable only the necessary network

communications throughout the environment that are required for normal operations in alignment with the segmentation design requirements.

Network ACLs act as a virtual firewall for associated subnets, controlling both inbound and outbound traffic at the subnet level. They are stateless, which means that return traffic will be denied by default, and both allow and deny rules are supported. VPCs, when created, have a default network access control list that, by default, allows all inbound and outbound traffic. All custom network access control lists deny any traffic by default. Subnets must be associated with a network access control list or they will be associated with the default one. Consider setting up default and custom network ACLs associated with subnets in alignment with the infrastructure segmentation design to restrict network communications. This network access control list design may be very similar to the security group design that is implemented.

Security groups are like virtual firewalls that control the traffic flowing in and out of resources that are associated with them, such as an EC2 instance. Security groups are stateful, so they allow return traffic, and they support allow rules only. The rules to allow traffic within the security groups should be designed so that they restrict network communications in alignment with the segmentation requirements. A security group design strategy should consist of creating them for applications that have similar functions and security requirements and use security group "chaining" where applicable. Chaining is the concept of only allowing traffic from an EC2 instance that has a particular security group applied, which allows more flexibility and control in a dynamic environment. You can have up to five security groups applied to a single network interface. It's a good practice to apply security groups in layers to create more generic security groups for reusability, and then layer additional more specific security groups for fine-grained control. The process for managing the lifecycle of security groups must be created to enable business operations (which typically includes DevOps) and maintain proper security.

When deploying EC2 virtual server instances into VPCs to host services that process or store data for business operations, several components must be in place to protect the environment. The credentials used to connect to these instances must be managed and protected. The guest operating system and software deployed within must be managed so that updates and security patches are applied regularly. This will reduce the chance for exposing vulnerabilities that can be exploited. The base "golden" image used to instantiate instances should be managed, custom built with hardened configurations, include the appropriate agent software, and updated regularly. Lastly, the instances should be ephemeral to help reduce the chance of a persistent threat in the environment.

When deploying EC2 instances, the guest operating system Amazon Machine Image (AMI) should be custom built with hardened configurations using Center for internet Security (CIS) benchmarks or other security hardening standards. This type of image is typically referred to as the "golden" image or base image. This image should be periodically revised to help confirm it is the most up-to-date to reduce vulnerability. Consider building an automated pipeline process and tooling where code is used to build and configure the golden AMI. EC2 instance images should also be built to include the appropriate agent software for logging, monitoring, patching, anti-malware, endpoint detection and response (EDR), IDS, and IPS. Consider using Amazon CloudWatch agent or other monitoring and logging solutions. Some form of anti-malware software agent and enterprise management should be in place. We strongly recommend using an EDR, IDS, and IPS solution as well.

As a part of managing the golden AMI, enforce a lifecycle rule to make sure that running EC2 instances are not using earlier images with potential vulnerabilities. To support this, use ephemeral instances that do not contain any persistent data. These can be re-instantiated (also known as rehydration) at any time without reducing operational effectiveness of the services being hosted on them. This may require storing persistent data in a database, an Amazon Elastic Block Store (Amazon EBS) volume, an Amazon Elastic File Store (Amazon EFS) volume, or perhaps an S3 bucket.

For remote host access use AWS Systems Manager Session Manager, a managed capability. This gives you the ability to log in to your hosts without the need to open inbound ports, maintain bastion hosts, or manage SSH keys. It allows authorized users to access EC2 instances in a way that can be monitored and logged without requiring SSH or RDP gateways, or other infrastructure.

You will eventually need to patch guest operating systems to remediate a vulnerability that has been discovered. Therefore, the appropriate mechanism must be in place to enable the discovery and remediation of these on a regular basis. (See the vulnerability management section of this whitepaper for more detailed information.) Software deployed within the guest operating system will also need to be updated regularly. It may be off-the-shelf software, which could have integrations with the vulnerability management solution, or a self-updating feature that must be configured and run periodically. Keep in mind that each type of operating system and software will have its own unique requirements for how to discover and remediate vulnerabilities. Use EC2 Image Builder to build secure images. This reduces the effort to create and maintain golden images without doing numerous automations.

When using container services, such as Amazon Elastic Kubernetes Service (Amazon EKS) or Elastic Container Service (Amazon ECS), several security controls should be configured. The container

images that are built and used, infrastructure (the Fargate or EC2 launch types), containers as they are running, and the networks that the container environments are using, must all be protected. Auditing, logging, and incident forensics must also be configured for observability.

- Confirm that images used by the containers are scanned for CVEs. Amazon ECR basic image scanning can be used to scan on push, or manually. Enhanced scanning in Amazon ECR can be enabled. It integrates with Amazon Inspector to provide automated scanning of image repositories for both OS and programming language package vulnerabilities.

- Remove unnecessary packages to reduce attack surface as the images are built. To use a single Dockerfile, images should be built from scratch using a multi-stage build technique. Consider using base OS images that do not have certain components, such as a package manager or shell.

- All container applications should be run as a non-root user, which can be enforced by linting the Dockerfile. Reduce risk of permission escalation by removing any files in containers that have `setuid` or `setgid` on them.

- Use VPC endpoints for accessing Amazon ECR through a private connection within the VPC. Use endpoint polies to restrict the IAM roles that can access container images.

For infrastructure security container security controls:

- Always use an operating system that is optimized for running containers, such as Amazon EKS optimized Amazon Linux 2 or Bottlerocket. Upgrade to the latest AMIs to reduce the chance of vulnerabilities. For EKS, using a managed node group will keep your application running while using a rolling-update deployment method. This will bring up new containers using the latest AMI while shutting down earlier containers. Amazon ECS also supports rolling updates using the `minimumHealthyPercent` and `maximumPercent` parameters for blue/green updates.

- Deploy container instances and worker nodes into private subnets. Any publicly accessible container instances and worker nodes should have restricted security groups associated with them.

- Minimize access to container instances or worker nodes and audit accesses to them. Do not use SSH keys. Use AWS Systems Manager Session Manager to access instances and nodes. It will also audit and log commands that were run on the node. Custom AMIs that include the SSM agent can be deployed when bootstrapping the container instances or worker nodes as they are launched, or they can be run as a daemon set.

- Periodically audit the configuration of the container cluster to achieve compliant configuration over time. For Amazon EKS, run kube-bench to continually monitor for compliant configurations of the cluster with Amazon EKS CIS benchmarks.

For runtime security of the running application container:

- Employ least privilege access to AWS resources. Keep in mind that users may not require IAM permissions to an AWS resource, they may need to be provided access to the cluster API instead. For Amazon EKS, use role-based access in the aws-auth ConfigMap to allow users to assume an IAM role for simpler management of this access. For Amazon ECS, use IAM roles for ECS tasks and IAM Roles for Service Accounts (IRSA) for EKS. Each pod should have a separate role in EKS.

- Run dynamic scans of the running containers using third-party tools such as sysdig falco, prisma cloud, or aqua.

- Use Policy-as-Code to enforce security standards using third-party solutions such as Open Policy Agent.

- For Linux-based containers, consider using tools like `seccomp` to restrict unauthorized `syscalls` from running.

To protect the network used by the container cluster environment:

- Use TLS encryption to protect all the data being transmitted to and from any pod or task, and certainly use this for sensitive data transmissions. Use mutual-TLS (mTLS) for communication between services.

- For Amazon EKS, use security groups for pods to restrict access to other AWS resources, and consider installing the Calico network policy engine to use Kubernetes network policies to restrict traffic within the cluster. For Amazon ECS, use the awsvpc network mode to attach an elastic network interface and security group to the task. The security group should be configured to restrict access to other AWS resources and other tasks within the cluster.

To enable auditing, logging, and forensics capabilities in the container environment:

- For Amazon EKS, enable the control plane logs, which include the logs for the Kubernetes API server, the controller manager, the scheduler, and the audit log. For Amazon ECS, configure the containers in your tasks to send log information to Amazon CloudWatch Logs. Consider using the AWS-provided Fluent Bit image with plugins for both Amazon CloudWatch Logs and Amazon Data Firehose.

- Centrally aggregate all logs and ingest them into a tool, such as a security information and event management (SIEM) solution. Log information can then be correlated quickly to identify indicators of compromise (IoCs) that must be investigated.

- If pods or tasks appear to be compromised, follow incident response procedures and consider isolating them to prevent further corruption in the environment. Remove or change labels and associate a network policy or security group to isolate pods or tasks. Consider cordoning worker nodes in Amazon EKS. Use instance draining in Amazon ECS to perform forensic analysis on the worker node or Amazon ECS instances (capturing memory, processes, or snapshots).

If you are planning to use a hybrid model with some hosts and services in AWS, and some in your on-premises data center, use AWS Direct Connect or AWS Site-to-Site VPN. If you use Direct Connect, make sure to use the transit encryption options for each service. This will encrypt data in transit that traverses AWS Direct Connect. Use VPN over Direct Connect to provide an IPsec-encrypted private connection that reduces network costs, increases bandwidth throughput, and provides a more consistent network experience than internet-based VPN connections.

Advance

As your cloud environment starts to grow, you may end up with hundreds of accounts and VPCs to manage. The best practices described in this section reflect the security measures that will help you configure and maintain your infrastructure at scale.

With a growing cloud environment, managing VPC peering and hybrid connectivity per VPC may become challenging. You might require additional inbound and outbound controls to support the scale and additional requirements. AWS Transit Gateway is a central hub that connects VPCs and on-premises networks. You can use Transit Gateway routing domains to route traffic between VPCs and your on-premises environment based on your routing requirements. Centralizing your networking using Transit Gateways also allows you to use AWS Network Firewall in a centralized VPC for east-west (VPC-to-VPC) and/or north-south (internet-egress and ingress-on-premises) traffic.

Create a dedicated inspection VPC comprising two subnets in each Availability Zone (AZ). One subnet is a dedicated firewall endpoint subnet, and the second is dedicated to an AWS Transit Gateway attachment.

For network layer inbound and outbound filtering, you can use AWS Network Firewall. With Network Firewall, you can write thousands of rules to filter traffic from the internet, on-premises, between VPCs, or from subnets within VPCs. If you have your VPCs connected via a Transit Gateway, you can use Network Firewall in a centralized way to inspect traffic. Network Firewall provides stateless traffic filtering using 5-tuple rule specification (source IP/port, destination IP/port, and protocol) and stateful traffic filtering with Suricata-compatible IPS rules, domain list

rules, and 5-touples. Start with managed rule groups (which are developed and maintained by AWS) and then add custom rules to fill any gaps.

If you want to use the same virtual appliances that you used in your on-premises environment, you can use [Gateway Load Balancer](#) (GWLB). With the Gateway Load Balancer, you can use the virtual appliances from independent software vendors (ISVs) and build your firewall policies without the overhead of maintaining them. For outbound domain filtering, you can use a combination of [Amazon Route 53 Resolver DNS Firewall](#) and AWS Network Firewall. DNS Firewall can be used to inspect queries that pass through the Route 53 Resolver. Network Firewall can provide FQDN-based domain filtering for both network and application layer.

For running web applications, deploy [AWS Web Application Firewall](#) (AWS WAF) to protect them from common web exploits and bots that may affect availability, compromise security, or consume excessive resources. [AWS WAF](#) provides managed rules that are crafted to provide protection against common application vulnerabilities, but you can also add custom rules. Select AWS Managed Rules based on your security and application requirements and use them in default mode so that they are automatically updated. You may need to fine-tune your rules when using AWS WAF with managed rules in default mode. Put your WAF-managed rules to a static version, analyze AWS WAF logs, and fine-tune AWS WAF rules to avoid false positives.

If you experience bot activity, use [AWS Bot Control features](#). Start with the Bot dashboard (available for free) to observe undesired patterns. Once you identify patterns, use the AWS managed Bot rule group and customize it. If you only have a few patterns and want to avoid the additional costs of a Bot control rule group, create custom rules to match the patterns instead of using the managed rule group. Use CAPTCHA to block bot traffic.

For medium and large organizations, we recommend using [AWS Shield Advanced](#) for Distributed Denial of Service (DDoS) protection. Shield Advanced provides additional detection and mitigation against large and sophisticated DDoS activities, and provides near real-time visibility into events. This is in addition to the layer 3 and 4 DDoS protection, and the application layer (layer 7) provided by [AWS Shield Standard](#) for Amazon EC2, [Elastic Load Balancing](#), [Amazon CloudFront](#), [AWS Global Accelerator](#), and [Amazon Route 53](#) resources. It also provides 24x7 access to the [Shield Response Team](#) (SRT), and protection against DDoS-related spikes in your Amazon EC2, ELB, Amazon CloudFront, AWS Global Accelerator, and Amazon Route 53 charges. We also recommend enabling proactive engagement with Shield Advanced so that the SRT team can monitor the status of the resources proactively.

In addition to the IPS (Intrusion Prevention System) capabilities provided by Network Firewall, AWS supports additional IPS and IDS capabilities. If you prefer to use an IPS/IDS that you already use in

your on-premises environment, you can use Gateway Load Balancer or VPC traffic mirroring. With Gateway Load Balancer, you can deploy virtual appliances from the AWS Partner Network and AWS Marketplace to help manage scale, availability, and service delivery. The VPC traffic mirroring feature allows you to send traffic from an elastic network interface (ENI) to an elastic network interface or Network Load Balancer (NLB). You can use this to send traffic to a virtual appliance to perform IPS/IDS activities. Additionally, you can use host-based IDS/IPS. These offer additional capabilities such as detecting changes in system files and inspecting system memory, but may be difficult to manage. We recommend selecting an IPS/IDS solution that works based on your requirements. Use a network-based IDS/IPS with a host-based IPS/IDS to provide defense in depth.

AWS Firewall Manager can centrally configure and manage firewall rules across your accounts and applications in AWS Organizations. With AWS Firewall Manager, you can configure VPC security groups and audit any existing VPC security groups for your Amazon EC2, Application Load Balancer, and ENI resources. Write AWS WAF rules and associate them to your Application Load Balancers, Amazon API Gateways, and Amazon CloudFront distributions. Activate AWS Shield Advanced protection for your Application Load Balancers, ELB Classic Load Balancers, Elastic IP addresses, and CloudFront distributions. Distribute AWS Network Firewall rules across multiple accounts, and associate your VPCs with Amazon Route 53 Resolvers DNS Firewall rules using AWS Firewall Manager.

When an organization grows, the number of hosts it must manage also grows significantly. Manually managing these instances can add additional overhead and introduce error. We recommend leveraging AWS Systems Manager to automate maintenance and deployment tasks such as running commands, collecting software inventory, applying OS patches, and so on. For example, use the *Run Command* feature of AWS Systems Manager to run `chmod og-rwx /etc/ssh/sshd_config`. This will affirm permissions on `/etc/ssh/sshd_config` are configured as a part of security hardening of the Amazon Linux 2 operating system in an AMI. Use services like Amazon Inspector or other vulnerability scanning software to discover vulnerabilities. AWS Systems Manager Patch Manager or other patch management solution can assist in deploying remediations.

To further reduce security maintenance, you can use managed services, such as Amazon Relational Database Service (Amazon RDS) and AWS Lambda, whenever feasible. Amazon RDS will help you deploy and manage relational databases without the maintenance overhead of hardware provisioning, patch management, or backup. Similarly, you can use Lambda to run code without maintaining servers, so you can focus on the functionality and security of your code without the overhead of managing infrastructure.

Excel

At this stage of maturity, best practices include enhancing architectures that have already been set to achieve Zero Trust, while fully managing infrastructure as code. Use automation to avoid human errors and embed security into DevOps. Whether you have dedicated teams managing infrastructure security, or have chosen a distributed DevOps model, it is important to have a RACI matrix that helps identify task or activity ownership while clearly defining roles and responsibilities.

Use AWS CloudFormation, Terraform, or the AWS Cloud Development Kit (AWS CDK) (AWS CDK), and make sure that all your infrastructure can be provisioned via code. For example, to achieve compliance, the security team should create a list of allowed security group rules and automate monitoring in your AWS environment for creation of inbound and outbound rules. Take automated action to revoke rules that are not on your *allowlist* in higher environments. Automation should be a main part of your infrastructure processes.

To simplify the configuration of AWS WAF rules, create and deploy a solution that will automatically deploy a web access control list (web ACL). The list should include AWS WAF rules with protective features that control which traffic to allow to web applications and APIs deployed on Amazon CloudFront, Application Load Balancers, or Amazon API Gateways. And it will filter out web-based attacks. Design and deploy automation for AWS Firewall Manager to simplify the configuration, management, and auditing of firewall, AWS WAF rules, VPC security groups, and Route 53 DNS firewall rules. Use published solutions such as the *Security Automations for AWS WAF* and *Automations for AWS Firewall Manager*.

In order for security enabled DevOps, the DevOps teams and security teams must find a balance between control and speed; between what are centralized controls and what a workload team can configure. A fast-feedback loop can influence your automation and focus on managing the risks. The DevOps team must be able to iterate quickly; so, the security team should build a faster process for changes in lower environments (such as development or test environments). The security team should maintain the ability to regulate the changes in both the lower and higher environments.

To further enable DevOps, a tagging strategy should be implemented in lower environments only. This permits the rules to be evaluated against the *allowlist*, but not shut down. An alert should be sent to the development team notifying them when the rule they created is not in compliance. Then they can follow a process to get these rules added to the *allowlist* in production.

Data protection

> **ℹ Note**
>
> Maintain visibility and control over data, and how it is accessed and used in your organization.

[Protecting](#) your data from unintended unauthorized access and potential unauthorized data changes, will help develop and drive the data protection component of your cloud security strategy. As you migrate data to the AWS Cloud, maintain visibility and control over how your data is being accessed and used throughout your organization. Before architecting any system, foundational practices that influence security should be in place. For example, [data classification](#) provides a way to categorize organizational data based on levels of sensitivity. [Encryption](#) protects that data by way of rendering it unintelligible to unauthorized access. These tools and techniques support such objectives as preventing financial loss, and complying with regulatory obligations.

Organizations will often look to compliance frameworks such as the [National Institute of Standards and Technology](#) (NIST), [Center for internet Security](#) (CIS), or their general industry for regulatory requirements and guidance on security controls that should be implemented. In this section, we will draw from the AWS Well-Architected Framework in the context of [Data Protection](#).

Start

To start building out a data protection strategy in AWS, focus on the following four elements.

Identify the data within your workloads.

Understand the type and classification of data your workloads are processing. What are the associated business processes, data owners, applicable legal and compliance requirements? Where are your data stored? These answers will help you identify the security controls that must be enforced to secure your environment. This may include classifications to indicate if the data are intended to be:

- Publicly available,
- Internal use only, such as customer personally identifiable information (PII), or

- Intended for more restricted access, such as intellectual property, legally privileged, or marked sensitive.

By carefully managing an appropriate data classification system and each workload's level of protection requirements, you can accurately map the controls and level of access and protection appropriate for your data. For example, public content is available for anyone to access, but internal or sensitive content may be encrypted and stored in a protected manner that requires authorized access to a key for decrypting the content.

Define data protection controls.

Use resource tags to separate AWS accounts per sensitivity and potentially community of interest. Use IAM policies, SCPs, AWS KMS, and AWS CloudHSM to define and implement your requirements for data classification and protection with encryption. For example, if you have a project with S3 buckets that contain highly critical data, or EC2 instances that process confidential data, they can be tagged with a "Project=ABC" tag. Only your immediate team knows what the project code means, and it provides a way to use attribute-based access control. If you are making authorization decisions based on tags, you should make sure that the permissions on the tags are defined appropriately using tag policies in AWS Organizations.

Define data lifecycle management.

Base your defined lifecycle strategy on sensitivity level as well as legal and organization requirements. Include the duration for which you retain data, data destruction processes, data access management, data transformation, and data sharing. When choosing a data classification methodology, balance usability versus access. Always use a defense in depth approach and reduce human access to data and mechanisms for transforming, deleting, or copying data. For example, require users to strongly authenticate access to an application. Give the application, rather than the users, the requisite access permissions to perform "action at a distance." In addition, confirm that users come from a trusted network path and require access to the decryption keys. Use tools, such as dashboards and automated reporting, to give users information from the data, rather than giving them direct access to the data.

Automate identification and classification.

Automating the identification and classification of data can help you implement the correct controls. Rather than manual, use automation to reduce the risk of human error and exposure. Amazon Macie uses machine learning to automatically discover, classify, and protect sensitive data in AWS and recognizes sensitive data, such as PII or intellectual property. Macie provides you with

dashboards and alerts that give visibility into how this data is being accessed or moved. To help you determine appropriate protection and retention controls, classify your data based on criticality and sensitivity.

Summary

We draw from the four areas that should initially be focused on to develop the foundation in which other aspects of data protection will rely (such as encryption at rest or in transit).

AWS has prescriptive guidance within the Well-Architected Framework and security whitepapers to help organizations explore developing a strategy to support their broader data protection initiatives in AWS.

First, develop a data classification strategy, and discover and identify where and what data resides in your workloads on AWS. Then, tag your data, resources, and assets to allow for proper scoping of security controls. Developing a tagging strategy is equally important as the development of your data classification strategy and subsequent discovery of classified data. Tagging can be an effective scaling mechanism for implementing cloud management and governance strategies. It helps identify, protect, detect, and respond to events in AWS. Tags can simplify ABAC, as well as streamline automation and operations, grouping of resources for enhanced visibility, and provide effective cost management.

Advance

Now that the foundation has been set in AWS, protect resources that have been classified, discovered, and tagged. Think of data protection in AWS as layers, and build out a defense in depth approach to protect your resources. Don't limit your protection to a single control, whether it is for an EC2 instance hosting an application, an EBS volume, or a file containing sensitive information in Amazon S3. The AWS Well-Architected Framework provides similar guidance.

> ⓘ **Apply security at all layers**
>
> Apply a defense in depth approach with multiple security controls. Apply to all layers (for example, edge of network, VPC, load balancing, every instance and compute service, operating system, application, and code).

Focus your data protection controls on these core areas:

Identity and access management

Embracing least privilege is critical to the protection of data, both from accidental exposure, as well as from potential malicious activities. When crafting IAM policies, use specific Amazon Resource Names (ARNs) and condition statements to limit over exposure to resources. Avoid using wildcard values whenever possible to reduce the overall permissions being granted to AWS entities. Enable AWS IAM Access Analyzer to monitor these access policies and alert you when it detects associate IAM policies not following protocol, or with access to external accounts.

Different controls including access (using least privilege), backups (see Reliability Pillar - AWS Well-Architected Framework), isolation, and versioning can all help protect your data at rest. Access to your data should be audited using detective mechanisms covered earlier in this whitepaper including AWS CloudTrail and service level logs, such as S3 Access Logs. You should inventory what data is publicly accessible, and plan for how you can reduce the amount of data available over time. Amazon Glacier Vault Lock and S3 Object Lock are capabilities providing mandatory access control. This means that once a vault policy is locked with the compliance option, not even the root user can change it until the lock expires. Access should also directly link to the organizations data classification strategy. This embraces the principal of least privilege, as well as proper scoping of business functions.

Resource policies

Many resources in AWS, such as VPC Endpoints, S3 buckets, and AWS KMS keys, use resource-based policies. These policies grant the specified principal permission to perform specific actions on that resource, and define under what conditions this applies. They complement Identity policies and offer additional layers of protection.

Encryption

A critical component of any data protection strategy is encryption of data at rest, in transit, and in use.

Data at rest represents any data that you persist in non-volatile storage for any duration in your workload. This includes block storage, object storage, databases, archives, IoT devices, and any other storage medium on which data is persisted. Protecting your data at rest reduces the risk of unauthorized access, when encryption and appropriate access controls are implemented.

Data in transit is any data that is sent from one system to another. This includes communication between resources within your workload as well as communication between other services and your end users. By providing the appropriate level of protection for your data in transit, you protect the confidentiality and integrity of your workload's data.

Data in use refers to data that is not simply being passively stored in a stable destination, such as a central data warehouse. It is working its way through other parts of an IT architecture. Data in use may be in the process of being generated, amended or updated, erased, or viewed through various interface endpoints.

Using encryption should be the only way to store sensitive data. AWS KMS integrates seamlessly with many AWS services to make it easier for you to encrypt all your data at rest. For example, in Amazon S3 you can set default encryption on a bucket so that all new objects are automatically encrypted. Additionally, Amazon EBS and Amazon S3 support the enforcement of encryption by setting default encryption. Use AWS Config Managed Rules to automatically check that you are using encryption, for EBS volumes, Amazon RDS instances, and S3 buckets.

Key management

To support the encryption components listed preceding, usage of key management services such as AWS KMS is recommended. The KMS keys that you create are customer managed keys. AWS services that use KMS keys to encrypt your service resources often create keys for you. KMS keys that AWS services create in your AWS account are AWS managed keys. KMS keys that AWS services create in a service account are AWS owned keys. Generate keys for specific applications, workloads, services, and environments of use AWS KMS Customer Managed Key (CMK). There should be different keys for Human Resources applications and Finance applications, and for Amazon S3 and Amazon RDS. And you should have different keys for development environments and production environments. Fine-grained access controls can be applied to keys to limit scope and any potential impact. More information about AWS KMS and best practices can be located in the Security Best Practices for AWS Key Management Service,

Logging, monitoring, and alerting

To confirm that the data protection controls are working as expected, it is important to set up detective controls, and then responsive controls as your maturity in cloud increases. Use automated tools to validate and enforce data at rest controls continuously. For example, verify that there are only encrypted storage resources. You can automate validation that all EBS volumes are encrypted using AWS Config Rules. AWS Security Hub can also verify a number of different controls through automated checks against security standards. Additionally, your AWS Config Rules can automatically remediate noncompliant resources.

Audit the use of encryption keys to validate that the access control mechanisms on the keys are appropriately implemented. For example, any AWS service using an AWS KMS key logs each use in

AWS CloudTrail. You can then query AWS CloudTrail and make sure that all uses of your keys are valid, by using a tool such as [Amazon CloudWatch Insights](#).

Amazon GuardDuty automatically detects suspicious activity or attempts to move data outside of defined boundaries. For example, GuardDuty can detect S3 read activity that is unusual with the [Exfiltration:S3/ObjectRead.Unusual](#) finding. In addition to Amazon GuardDuty, [Amazon VPC Flow Logs](#), which capture network traffic information, can be used with Amazon EventBridge to trigger detection of abnormal connections, both successful and denied. [S3 Access Analyzer](#) can assess *what* data is accessible to *who* in your [S3 buckets](#).

Summary

Putting all of this guidance together, a real-world example would be a principal (human or machine) who must upload data to S3. This data will then need to be pulled down and processed by an application. As part of their [IAM role policy](#), the principal would have access to Amazon S3. Through constraints in the policy, they would have access only to *specific* S3 buckets, with the ability to upload. This principal also has permissions to use a [Customer Managed Key (CMK) in KMS](#) to encrypt the data in Amazon S3. The CMK itself has a resource policy attached, permitting the principal to only invoke the encrypt API. This same resource policy also defines the application [IAM role](#) and permits the decrypt API to be invoked as part of the download and processing by the application. The S3 bucket has a resource policy attached to only allow uploads from defined principals, and download from defined applications IAM roles. It also has restrictions to only select [VPCs](#) inside of the AWS account that is part of a defined [AWS Organization](#). The bucket policy also only allows uploads if a specific [KMS key](#) is defined as part of the PUT API.

In the preceding example, if one control fails, the data is not exposed, and in many cases, if two controls fail, the data is *still* protected. This might happen if the S3 bucket was accidentally made public and the principal's account credentials were compromised. This is an example of a defense in depth approach.

Excel

Excel

At this stage of cloud maturity, data protection best practices include enhancing the foundation that was initially set, iterating on the capabilities built upon that foundation, moving towards *shift left* culture in [DevOps](#), and focusing on automated remediations.

Advanced permissions management

Protecting data depends on the permission structure that you have built. While basic IAM provisions are an essential part of the foundation for all data protection strategies, advanced topics of IAM should be implemented when possible. ABAC is an advanced permissions management model that scopes all permissions by the tags of resources and entities. For example, an IAM role with the tag `Project X` should only have access to resources also tagged with `Project X`. More advanced AWS services can be used to further analyze and refine AWS permissions within your environment to adhere to the principle of least privilege. If you are using AWS Organizations, SCPs prevent sensitive actions from being taken on your most critical resources unless stringent conditions are met. IAM Access Analyzer is a tool that can identify all external access into your AWS environment. It has policy generation and validation capabilities to help you further scope your AWS IAM policies. It uses machine learning (ML) to understand the minimum permissions needed by your everyday workloads. These tools should be leveraged to automatically redefine IAM policies on principals based on access needs observed over time.

Advanced encryption

Beyond the concepts of encryption of data at rest and in transit, advanced encryption should focus on using specific ciphers and elliptic curves for varying data protection needs. You should consider quantum cryptography protections, tokenization, format preserving encryption (FPE), and masking. These types of encryption concepts are meant to solve specific industry and regulatory needs in addition to enhancement of encryption protections. For example, use Federal Information Processing Standard (FIPS) 140-2 L3 technologies for US government systems or regulated financial organizations, or use tokenization to meet Payment Card Industry (PCI) requirements on secure storage of customer data. The hardware security modules (HSMs) used in AWS KMS have been awarded FIPS 140-2 Security Level 3 certification from the U.S. National Institute of Standards and Technology (NIST).

Key management

Another domain of data protection to further develop along with your organizations' cloud maturity is *key management*. Using AWS KMS, you can enforce automated key rotation schedules, confirm that resources are encrypted with distinct keys, define key policies to restrict which principals have admin permissions (delete, create keys) and use (encrypt, decrypt), and more. To meet other use cases, or for customers who want to control the generation of key material, AWS CloudHSM offers a managed hardware security module solution that allows you to generate and use your own keys within a dedicated hardware space on AWS. AWS CloudHSM can be connected to AWS KMS to create a Custom Key Store where you can generate and manage keys inside of a FIPS 140-2 L3 boundary.

Replication, backups, and recovery

A robust data protection strategy should have measures in place to prevent unintended data deletion. Confirm that replication, backup, and recovery mechanisms are implemented so your data is resilient against unforeseen incidents of data loss. Carefully consider replication capabilities across AWS services that may handle your data and determine the best replication needs for your workloads. AWS Backup is a service that allows you to centrally manage and automate backups across AWS services and hybrid workloads. Implement automated recovery plans for your resources in the event of unexpected failure so your workloads experience minimal downtime.

Dynamic data classification

As you expand on your data classification strategy, build out automated mechanisms to identify new types of data. This data is generated by using different datasets through a process commonly known as *data fusion*. This generally occurs when querying data across datasets or in data lakes. For example, if you take one dataset of city names, and one dataset of customer last names, the resulting combined dataset now contains PII data. This new data must be classified and protected. Use tools like Amazon Macie to identify these new datasets and then apply data classification, tagging, and protective controls.

Data usage monitoring

As you start to acquire data in varying services such as Amazon EBS, Amazon EFS, Amazon S3, data lakes, data warehouses, it is increasingly important to monitor this data and its usage. Build mechanisms that allow for reporting, analytics, alerts, and automated remediation across data types and locations. Understand how, where, and by who your data is being used, to identify norms.

Summary

As part of advanced data protection operations in AWS, you will want to use all of the guidance outlined in this whitepaper. Add guidance and best practices that are applicable to your organizations regulatory and compliance needs. It is important that you create a defense in depth approach to data protection in AWS, and implement preventative, detective, and responsive mechanisms. Use mechanisms such as automated rescoping of IAM permissions based off of observed usage patterns, and auto enforcement of data classification and tagging. You can automate enforcement of encrypted resources, alerting on abnormal data usage, and backup and recovery as well. Mine logs and analytics to identify new patterns and iterate on your data protection policy. Automate wherever possible, and push into the development pipeline to enforce

data protection and security early on. Finally, ascertain that your mechanisms remain agile, so when new services and features are released, you are able to explore in development and securely promote to production.

Application security

> **ⓘ Note**
>
> Detect and address security vulnerabilities during the software development process and while operating the application.

Organizations are adopting transformative approaches, such as DevSecOps (Development, Security, and Operations), to consistently use, operate, and improve the quality and security within the SDLC using CI/CD pipelines. More components of the end-to-end application architecture and software supply chain are expressed as code. This includes core delivery infrastructure (code repository, pipeline, artifact store, application code, and underlying infrastructure). It's imperative to utilize automation to achieve a high level of security assurance at scale. Integrate dedicated application security capabilities into the various stages of the SDLC using tools and services (AWS services, AWS Partners, AWS Marketplace, and third parties). These can detect and prevent instability, misconfiguration, and vulnerabilities.

Start

Your customers and product owners are requesting development teams to release more, faster, and with more confidence. Begin by selecting a single application team to transform the way they deliver to meet these goals. In order to integrate security analysis and review processes into the design stage of the SDLC, make use of risk analysis techniques like attack and threat modeling. Threat modeling should be performed during the design phase of a given workload feature or feature change, as these changes may introduce new threats that require you to update your threat model. Take advantage of best practice assessments, like the AWS Well-Architected Framework.

To assure security and quality within the development stage, some developers (*champions*) within the team assume dedicated security and quality responsibilities to assist with reviews and communication with the centralized teams. Integrating security and quality processes into the development and testing phases of the SDLC helps expedite subsequent reviews by the centralized security team.

Centralized security and operations teams can now focus on developing security guardrails, detective capabilities, enterprise best practices, and automation to support other teams at scale. They can implement automated testing capabilities and tools that support functional, non-

functional, and security use cases, in order to rely less on manual auditing and human intervention. Developers interface with these tools during the testing stage of the SDLC in a self-service manner. They use an integrated development environment (IDE), a command line interface (CLI), and tools within the CI/CD pipeline to reduce the number of pre-existing workload vulnerabilities. Carry out these processes.

- Perform functional application tests (such as unit testing, integration testing, regression testing, and user acceptance testing).

- Pre-commit hooks using static testing tools like awslabs/git-secrets, which can be used to detect static secrets and sensitive information before code is committed to a source code repository.

- Conduct static application security testing (SAST) using Amazon CodeGuru (or another tool from the AWS Marketplace) to automatically review code and detect security vulnerabilities.

- Perform IaC static code analysis using aws-cloudformation/cloudformation-guard to enforce best practices and prevent misconfiguration of AWS resources.

- Conduct a software composition analysis (SCA) to audit known vulnerabilities within publicly available software packages, read How to automate your software-composition analysis on AWS.

- Generate a software bill of materials (SBOM) and store it with every release.

- Perform penetration testing in accordance with the AWS Customer Support Policy.

Applications can use AWS KMS and AWS Secrets Manager to protect passwords and other sensitive data while abstracting them from source code and configuration files. Adding mitigating controls at the edge using tools like AWS WAF and AWS Shield provides protection against common web exploits (also refer to the OWASP top 10). Bots may affect availability, compromise security, or consume excessive resources, and AWS WAF improves the ability to respond to incidents as they occur. Observability of applications is provided by Amazon CloudWatch, while monitoring AWS resource misconfigurations against a baseline of security frameworks using AWS Security Hub.

Advance

As the testing tools, detective and preventive guardrails, and automation are tuned and matured, the confidence in these capabilities and noticeable improvement in speed to deliver software encourages more teams to begin their transformation. AWS Organizations SCPs enable the evaluation, security customization, and use of a growing list of new AWS services. Common architecture patterns start to emerge between teams, a shared library of Well-Architected CI/CD pipeline, infrastructure, and application patterns that can be centralized and made available to

development teams in a self-service manner. General purpose, machine learning-powered code generators, such as Amazon CodeWhisperer, are used to automatically generate personalized recommendations in real time.

The detection and prevention of security and operational flaws are further *shifted left* to maximize the benefits of discovering vulnerabilities earlier in the SDLC where they are easier and cheaper to fix. Enforcement of testing capabilities using a *break-the-build* approach is adopted in accordance with the enterprise risk management strategy. More advanced testing capabilities are introduced into the testing stage of the CI/CD pipeline, such as:

- Functional acceptance testing using Amazon CloudWatch Synthetics canaries to continuously verify customer experience
- Dynamic Application Security Testing (DAST) from the AWS Marketplace used to analyze real-world vulnerabilities against the runtime operations of the application
- Resilience testing using AWS Resilience Hub to track and optimize recovery time objective (RTO) and recovery point objective (RPO) of workloads
- Performance testing using Amazon DevOps Guru to detect, analyze, and make recommendations for operational issues

Production applications begin to adopt Zero Trust architectures that do not solely depend on network perimeters. Instead, network controls are augmented with identity, device, behavior, and other rich context and signals to make more granular, intelligent, adaptive, and continuous access decisions. This may include the use of a service mesh like AWS App Mesh, application authentication and authorization using Amazon Cognito, and reducing the use of long-term credentials using AWS IAM Identity Center (successor to AWS IAM Identity Center). Amazon EC2 instances are provisioned and managed at scale using configuration management tooling like AWS Systems Manager. EC2 instances can leverage automated vulnerability management tooling like Amazon Inspector, and automated patching solutions such as AWS SSM Patch Manager. Container images should be managed through ECR and scanned with ECR image scanning for vulnerabilities. AWS Incident Manager can be implemented to build automated response plans and enable active collaboration when a security incident happens. AWS CodeArtifact can be used to store your application artifacts securely and achieve traceability and accountability through AWS CloudTrail.

Excel

As you collect output from various testing tools across your environment, you can make better use of the vulnerability trending data. You can adjust key performance indicators (KPIs) where

needed, and make more data-driven decisions. Centralize output data from your various testing tools using Amazon RDS, and create dashboards with QuickSight. Create KPIs to monitor and drive secure coding behavior at the team, leadership, and individual level. The focus should be on *time-of-detection* versus time-of-remediation, recurrence of vulnerabilities, security incident remediation timings, and preventable vulnerabilities that made it into production.

The central team should build re-usable libraries for security functionality that can be shared across application teams and *golden* EC2 AMIs, or container images using EC2 Image Builder. Containers and EC2 instances should be configured to only run signed code, to prevent malicious code from being initiated. Use AWS Signer to sign code. Include security conformance testing. High or medium risk vulnerabilities should break the build and taint artifacts to prevent them from being deployed to production. Create a bug-bounty and responsible disclosure program to crowdsource security testing and receive vulnerability reports.

Consider using interactive application security testing (IAST), bug bounty programs, or runtime application self-protection (RASP) to further reduce false positives, detect vulnerabilities, and provide defense in depth.

Infrastructure should be ephemeral. Applications should rely on blue/green deployments to roll out fixes, patches, and updates. Initiate canary deployments for multiple releases per day across multiple teams. Make your application more security self-aware to perceive attackers attempting to bypass business logic or fuzzing input. Use CloudWatch Alarms to generate alerts.

Resilience of your infrastructure should be tested using Chaos Engineering with AWS Fault Injection Service.

Threat detection

> **ⓘ Note**
>
> Understand security threats and detect malicious activity, data breaches, or other security events.

Understanding the threats that are applicable to your organization and identifying the services and controls to implement is key to an effective threat detection program. This helps you identify possible threats early, reduces the dwell time of unauthorized users, and enables you to respond more quickly to mitigate the event.

Start

When creating *directive*, *preventative*, *detective*, and *responsive* controls, you should adhere to the governing controls and requirements relevant to your organization. Threat modeling can help you map and understand the severity and business criticality of components within a workload. You can then effectively address the risks and impacts, and identify the specific controls and mitigations that can be implemented. Establish a team of individuals responsible for monitoring and responding to events, tuning detections, and proactively looking for anomalous behavior. This team should be actively involved and consulted to review the architecture for enabling *threat detection* and *incident response.* It should provide the logging requirements needed for building detections and alerting. Log ingestion should be verified periodically to validate that all logs necessary for detection are enabled and available, log ingestion pipelines are functioning, and alerts are initiating as intended. A security review will provide insight into which types of logs should be collected and centralized to a restricted AWS account for effective threat monitoring and detection, including what applicable detective controls will generate findings (such as GuardDuty findings). Examples of such logs may include, but are not limited to:

- Service-level logs, with alerting and associated automated response, including:
  - AWS CloudTrail logs, Command Line Interface and APIs
  - Amazon GuardDuty findings
  - Elastic Load Balancing logs
  - Amazon Route 53 resolver query logs

- Amazon S3 data event logs or server access logs
- Virtual Private Cloud (VPC) Flow Logs
- Host-level logs, including:
  - Application specific logs, such as web server logs
  - Audit logs
  - Endpoint detection logs
  - System logs

Enabling detective controls on an AWS account or across AWS Organizations should be carefully planned with scaling in mind. Use AWS Organizations functionalities built-in to AWS Security Services such as Amazon GuardDuty, AWS Config, and AWS Security Hub. This will provide the security teams the visibility into the services' functionality into each account, permitting them to audit their detection capabilities. Organizations should plan for how custom controls can be deployed and audited at scale, and provide teams the ability to audit those controls.

Using automation for the deployment of detective controls and logging services will allow the threat detection team to match the growth as an organization expands and creates new accounts. Tools like Assisted Log Enabler for AWS give threat detection teams the ability to enable and audit AWS CloudTrail logs, Amazon Elastic Kubernetes Service (Amazon EKS) logs, Amazon Route 53 resolver query logs, Elastic Load Balancing logs, Amazon S3 server access logs, and VPC Flow Logs across accounts, within a single account, or across AWS Organizations.

Individuals responsible for threat detection should be trained and knowledgeable with hands-on experience in both the operation and security of AWS Cloud Products. This knowledge and experience will aid in understanding the shared responsibility across AWS Cloud services and how to build the most effective detective controls with those responsibilities in mind.

Advance

At this stage, consider analyzing logs to identify configurations that do not follow best practices.

When building new detections to proactively identify threats, use a dedicated testing environment. Teams can use DevOps methods for building detections with version control, backlog tracking, testing, and deployment. Apply a DevOps framework for the detection development process. This will promote the ability to fail fast and roll back to a known functioning version should an update or new detection have unintended effects.

Detections are implemented at each layer of an application, including:

- Operating system detection:

  - Unauthorized software is installed

  - Outdated software is installed

  - Abnormal audit/system activity

- Application detection:

  - An SQL injection is performed against a web server

  - A vulnerability in an application is exploited

  - An application is leveraged for malicious purposes

- Control plane detection:

  - CloudTrail logs are disabled

  - Amazon EC2 instance is deployed using an earlier or unauthorized Amazon Machine Image (AMI)

  - A security group that allows all inbound and outbound traffic is created and applied to an EC2 instance

  - An S3 bucket is made public

    Finally, close the loop by ensuring that any identified issues feed into your *lessons learned* mechanism so that corresponding preventative controls can be implemented or adjusted as required.

Excel

Threat detection teams should exercise and test detections regularly through real-world scenarios and production environments to confirm detections are operating as intended. Incorporate lessons learned into the threat detection development process, as well as current and future architectures for detecting threats. To proactively identify threats, threat detection teams should implement and perform regular threat hunting (searching) throughout their AWS environments. This will allow the team to proactively identify and deploy new detections and measures otherwise unknown, further reducing dwell time of unauthorized users within an environment.

The threat detection team should employ mechanisms to share artifacts and intelligence to inform and educate internal teams on the applicable threats. This will help internal development and operational teams build more informed and effective threat models to better protect and defend their environments. The threat detection team should also consider joining and participating in external threat intelligence sharing communities. They can contribute intelligence to the community and gain additional insights and data into the threats targeting their specific business, applications, and industry vertical.

Incident response

> ℹ️ **Note**
>
> Reduce potential harm by effectively responding to security incidents.

Incident Response (IR), as a structured process to detect, identify, and manage security events, remains largely the same in the cloud. Well-known IR frameworks, such as NIST SP 800-61 Computer Security Incident Handling Guide and the AWS Security Incident Response Guide continue to be effective guides to building and operating IR programs in the cloud. Cloud IR does not exist in isolation. Building and operating an IR capability requires a mesh of security processes and disciplines that span the entirety of the AWS CAF security capability model. In this prescriptive guidance, important dependencies with other security capabilities will be highlighted.

Start

Start with a plan.

A written Incident Response Plan should define the purpose and response objectives of IR for the organization. Align the response goals to legal and regulatory requirements and the risks to the goals of the organization. Identify both internal stakeholders and important external parties, and establish roles and responsibilities. If you are adapting an existing IR plan to cloud, update the plan to incorporate an understanding of the AWS Shared Responsibility Model. Communicate with stakeholders to educate them on what's different about responding in the cloud.

Security governance – inventory *(dependency).*

Know what is in your cloud environment through continuously updating and managing your asset inventory. Work with security governance teams to help confirm that assets are properly tagged, inventories are up-to-date, and both are available to incident responders.

Prioritize continuous education and training.

Incident response is a complex topic, requiring a wide range of technical knowledge about the systems and services that are involved with security incidents. While the incident response process is largely the same in the cloud, you must train your operations and incident response staff on cloud technologies and how your organization uses them.

Establish an incident classification framework.

Define what a "security incident" is for your organization and develop a method for classifying incidents. A classification framework will assist you in triaging incidents so that both response and preparation activities can be prioritized. You'll be able to collect useful metrics that will help in determining the performance of your incident response program and areas needing improvement. A simple framework may assign incidents to a *category* such as, denial of service, malware, or unauthorized access, along with an impact-based *severity* such as, critical, high, medium, or low. More complex systems may be created that include threat types as well as tactics and techniques. Not having a classification framework in place may result in either over or under reporting of security incidents and lead to inefficiencies as you are unable to easily prioritize your response activities.

Build a framework of playbooks.

A [playbook](#) is a written plan for investigating and responding to specific types of incidents. It defines tasks and the detailed procedures for completing them. Start by creating playbooks that focus on specific types of alerts that are common in your environment. Begin with playbooks that capture manual tasks. To discover and capture exception conditions, iterate on the playbook processes to improve the logical flow. The primary aim of the playbook is consistency and repeatability. The [Customer Response Framework](#) provides an example framework for customers to create, develop, and integrate security playbooks in preparation for potential attack scenarios when using AWS services. These playbooks can be used as a guide for what detections to build as well as how to effectively respond to each. Continuously manage and expand your playbook library to cover more complex playbooks. The [MITRE ATT&CK](#) framework can be a useful tool in helping to build playbooks, supporting completeness of investigations, and understanding where you might have gaps in the coverage of your playbooks.

Automate responses to incidents.

Automation is a force multiplier, which enables your responders to scale efforts to match the organization. It frees up your time to spend improving the security of your cloud environment. For example:

- [Amazon Detective](#) automatically collects log data from your AWS resources, and then uses machine learning, statistical analysis, and graph theory to generate visualizations that help you to conduct faster and more efficient security investigations.

- [AWS Security Hub](#) collects security data across AWS accounts, AWS services, and supported third-party products and helps you analyze your security trends and identify the highest priority security issues.

- The [Automated Security Response on AWS](#) solution is an add-on that works with AWS Security Hub and provides predefined response and remediation actions based on industry compliance standards and best practices for security threats.

Look for automation opportunities within the incidents you receive frequently. Then, take well-defined and tested manual tasks from your playbooks and [automate them](#). Start with the most time-consuming tasks first, and build your automation in the cloud in order to match the resilience, scale, and elasticity of your workloads. Your response to incidents should become more frequently expressed as code, either through serverless functions or automatically building response environments through infrastructure code. Start using the security practices of your organization's SDLC and CI/CD pipelines.

Implement threat detection *(dependency).*

The IR team frequently does not directly own detection mechanisms; however, they are critical consumers of them. Many other capabilities in AWS CAF besides threat detection (such as AWS Identity and Access Management, vulnerability management, and data and infrastructure protection) will have detective functions. One of the most significant vulnerabilities in any environment is lack of visibility. Stay connected with these detection capabilities, understand what detections you have, know where the gaps are, and work with your partners in these other areas to improve detection coverage.

Advance

Define measures and KPIs.

Begin by defining incident service level agreements (SLAs). The SLA will form the basis for mean time to respond, an important measurement of performance for the IR program. Introduce additional metrics to further measure performance, such as mean time to detection and mean time to recovery. Regular reporting on these metrics will provide valuable performance data that can help guide IR program improvements and demonstrate effectiveness.

Security assurance*(dependency).*

When developing metrics, measures, and other KPIs or risk indicators, work with security assurance staff. Verify that the data you have about the effectiveness of your cloud IR capability is reported regularly and to the right audience.

Incorporate lessons learned into your incident processes.

After each security event, establish the practice of learning from what went well, and what could have been better. This step comes *after* returning to normal operations, and should result in a list of improvement actions for IR processes, plans, and playbooks. Some ideas of the kinds of questions you might answer in a *lessons learned* exercise are:

- **Detection** – Could you improve time to detection? Are there updates to metrics and alarms that would detect the incident sooner?

- **Diagnosis** – Can you improve the time to diagnosis? Are there updates to your response plans or escalation plans that would engage the correct responders sooner?

- **Mitigation** – Can you improve the time to mitigation? Are there runbook steps that you could add or improve?

- **Prevention** – Can you prevent future incidents from occurring? To discover the root causes of an incident, Amazon uses the 5-Whys approach in problem investigation.

To follow through with the implementation of these lessons and assign ownership to the identified tasks, items from lessons learned should be added to a tracked backlog of work.

Test your response processes.

Periodically test your plans and playbooks with Security Response Simulations. Simulations may consist of assembling key IR staff and stakeholders from outside of the IR team, and talking through specific scenarios. Simulations help prepare responders under less stressful conditions so that they may perform better under real conditions. A simulation also details specific response plans and playbooks to validate readiness or expose possible issues. Simulation exercises should likewise conclude with lessons learned, and produce a list of improvement actions to processes, plans, and playbooks.

Threat Detection *(dependency).*

As you consider threat intelligence collection, partner with those responsible for threat detection solutions. These tools may have useful intelligence feeds that come as part of the tool or that can be added for an additional cost. There are both commercial and open source options available for

threat intelligence platforms. Choosing the best option for your organization will take a broad understanding of your security technology stack, as well as your IR goals.

Excel

Integrate your IR processes with a repository of identify indicators of compromise (IoCs), tactics, techniques, and procedures (TTPs) and threat research.

Use threat intelligence platforms (TIPs) to aggregate threat data from intelligence feeds, and store in a repository for your own threat data. Create, store, and share IoCs that are relevant to your environment in a centralized repository. This will speed the IR process by improving the accuracy and completeness of investigations. It will enable automated threat analysis using the latest information from threat feeds or collected from a threat analysis. Amazon GuardDuty threat intelligence is provided by AWS and third-party providers, such as Proofpoint and CrowdStrike. These threat intelligence feeds are pre-integrated and continuously updated in GuardDuty at no additional cost.

Implement threat detection*(dependency)*.

A Well-Architected logging infrastructure and threat detection model will enable you to build automated response use cases faster and easier. As you build and improve automated response functions, use response feedback to improve the log data you collect and store, and your protective controls. You will be able to optimize all of the components in your entire security program as you iterate.

Conclusion

As you embark on your cloud transformation journey, update your security posture to include the AWS portion of your environment. This Security perspective whitepaper prescriptively guides you on an approach to take advantage of the benefits that AWS has for your security posture. Much more security information is available on the AWS website, where security features are described in detail. More detailed prescriptive guidance is provided for common implementations. There is a comprehensive list of security-focused content that your security team can review, as you prepare for AWS adoption initiatives.

Contributors

Contributors to this document include:

- Alex Torres, Senior Solutions Architect – Cloud Foundations
- Anne Coulombe, Principal Security Advisor, Healthcare & Life Sciences
- Carl Mathis, Senior Privacy Consultant
- Chris Lamont-Smith, Senior Security Consultant
- Cliff Donathan, Principal Security Advisor
- Damindra Bandara, Senior Security Consultant
- Dario Goldfarb, Principal Security Solutions Architect
- Don Edwards, Worldwide Tech Leader – Identity
- Dustin Ellis, Senior Solutions Architect
- Dutch Schwartz, Principal Security Specialist
- Eva Mineva, Senior Product Manager – Identity Customer Experience
- Fred Charlot, Senior Assurance Consultant
- Fritz Kunstler, Principal Security Consultant
- Greg Eppel, Tech Leader – Management and Governance (Cloud Ops)
- Istvan Berko, Practice Manager, Security GTM Specialists
- Jack Huang, Security Consultant
- Jamie Rubbi-Clarke, Senior Security Consultant
- Jonathon Poling, Principal Security Consultant
- Kyle Dickinson, Senior Security Solutions Architect – Threat Detection / Incident Response
- Liam Schneider, Security Practice Manager
- Lucas Kauffman, Security Consultant
- Mark Lieberg, Senior Security Consultant
- Max Farnga, Security Consultant
- Mayank Jain, Principal Security Advisor
- Michael Rhyndress, DevSecOps Consultant
- Paul Duvall, Director – DevSecOps
- Payal Vadhani, Director – Security

- Richard Billington, Senior Security Consultant
- Rob Samuel, Principal – Security Assurance
- Saša Baškarada, Worldwide Lead AWS CAF
- Sausan Yazji, Senior Practice Manager
- Scott Conklin, Senior Security Consultant
- Tyson Martin, Principal Security Advisor
- Volker Rath, Principal Security Consultant

Further reading

For additional information, see:

- AWS Architecture Center
- AWS Case Studies
- AWS General Reference
- AWS Glossary
- AWS Knowledge Center
- AWS Prescriptive Guidance
- AWS Quick Starts
- AWS Security Documentation
- AWS Solutions Library
- AWS Training and Certification
- AWS Well-Architected
- AWS Whitepapers & Guides
- Getting Started with AWS
- Overview of Amazon Web Services

Document history

To be notified about updates to this whitepaper, subscribe to the RSS feed.

Change	Description	Date
Whitepaper updated	Major updates throughout.	December 12, 2023
Initial publication	Whitepaper first published.	August 1, 2016

> ⓘ **Note**
>
> To subscribe to RSS updates, you must have an RSS plug-in enabled for the browser that you are using.

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided "as is" without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

AWS Glossary

For the latest AWS terminology, see the [AWS glossary](#) in the *AWS Glossary Reference*.