

# Checklist: Securing your AWS Workloads

Amazon Web Services (AWS) is the leader in the public cloud market. They offer a broad set of services that help organizations move faster, lower IT costs, and scale applications.

Like most cloud providers, AWS operates under a shared responsibility model — managing security of the cloud while AWS customers are responsible for security in the cloud.

Threats to your workloads running in AWS can take many forms:

- Compromise of the AWS account
- Data leakage or system compromise through insecure configurations
- Breaches through publicly presented applications that are not thoroughly assessed or monitored
- And more...

Use this checklist as a guide to the activities and references you need to start building a secure foundation for your workloads or assess existing setups.

## Start with a Solid Foundation

### **ENSURE INTERNAL ALIGNMENT**

Identifying your internal stakeholders, their expectations and requirements, as well as meeting with individuals who will be impacted by the project is critical.

*Engage security stakeholders during requirements gathering.*

*Include IT security staff throughout the project delivery processes.*

*Consider forming a cloud COE (Center of Excellence) that includes a stakeholder from each appropriate BU (Business Unit.)*

## FAMILIARIZE YOURSELF WITH AWS GUIDANCE

### *The AWS Shared Responsibility Model*

AWS provides clear guidance on where responsibilities lie between their customers and them, as it relates to security. Ensure you fully understand your responsibilities. [Learn More](#)

### *The AWS Well-architected Framework*

The [AWS Well-Architected Framework](#) helps you understand the pros and cons of decisions you make while building systems on AWS. By using the framework, you will learn architectural best practices for designing and operating reliable, secure, efficient, and cost-effective systems in the cloud.

## Architect For Security

### *Map security boundaries using AWS controls.*

Inventory and categorize workloads — segmenting environments based on your organization and the security of your data. Consider the following:

- Environment type
- Regulatory scope
- Change control requirements
- Application and infrastructure tiers

### *Use a strategic security framework to understand risks and identify areas for gap analysis.*

For example, the NIST framework is a useful tool to assess and improve your ability to prevent, detect, and respond to cyber-attacks.

### *Plan to automate security best practices:*

*Standardize on the least access and privilege security controls.*

*Define and enforce base standards and controls for reusable system component.*

Standardize a tagging strategy, AMI, database instance, and service configurations used to build applications.

*Define organizations/roles that secure and control those components.*

*Implement these as orchestration code for environment builds.*

### *Prepare for security incidents.*

Leverage infrastructure-as-code methodologies to enable rapid response in the event of a security incident.

### TIP

Leverage the AWS Organization's service and use separate AWS accounts based on development, testing and production. Where compliance requirements differ, these environments often have very different access requirements and data sensitivity.

[Learn More](#)

### TIP

Many of the benefits infrastructure-as-code brings to application availability, stability, and scalability can be leveraged for security response.

[Learn More](#)

## Adopt Assessment Standards

**Use a standard set of assessment criteria** to identify drift in your environment away from best practices.

You should use third party assessment criteria combined with a set of your own internal checks which align with your unique standards. A good example of third-party assessments would be the AWS CIS Benchmarks.

**Adopt a third-party assessment tool** to understand risks in your environments.

To reduce workload and drive a consistent approach, identify solutions that allow you to understand your risk and prioritize changes to improve security posture.

### TIP

Alert Logic provides automated checks against the CIS Benchmarks as well as our Threat Risk Index which assesses your system and application vulnerabilities based on our proprietary algorithm

## Define Access Standards

**Protect the Root Account**

This account has ultimate control over your AWS environment — its security is paramount. Leaked root account access keys are the source of many AWS account breaches.

**Use IAM policies, groups, and roles that have:**

*Unique accounts for all individuals*

*Multi-factor authentication turned on as default*

*Strict password policies*

*User permissions configured at group and role level*

*Different configurations for AWS Console, AWS API, and service or application permissions*

**Identify where IAM roles can be leveraged** in place of IAM users.

**Consider federation and single-sign-on** options for access management.

**Terminate unused** access keys.

**Disable access** for inactive or unused IAM users.

**Remove unused IAM policy** privileges.

**Remove unused IAM access** keys.

### TIP

The easiest way to protect your root account is not to use it. Set a very strong password, enable multi-factor authentication and lock it away

[Learn More](#)

**Protect systems from network threats:**

*Disallow unrestricted ingress access on uncommon ports.*

*Restrict access to well-known ports such as CIFS, FTP, ICMP, SMTP, SSH, and remote desktop.*

*Restrict outbound access.*

## Protect Data

**Encrypt data wherever possible to mitigate lateral spread in the event of compromise:**

*Enable EBS encryption by default.*

*Use the AWS: SecureTransport condition for Amazon S3 bucket policies.*

**Enable S3 Block Public Access** for all accounts and buckets that you do not want publicly accessible.

**Use AWS IAM user policies** to specify who and what can access specific S3 buckets and objects.

**Enable MFA delete** for S3.

**Set up MFA-protected API access** for S3.

## Visibility and Threat Detection

**Enable logging and auditing through Cloudtrail:**

*Turn on CloudTrail log file validation.*

*Enable CloudTrail multi-region logging.*

*Enable access logging for CloudTrail S3 buckets.*

*Disallow deletion of CloudTrail buckets.*

*Ensure CloudTrail logs are encrypted at rest.*

**Turn on AWS Security services:**

*Amazon GuardDuty*

*AWS IAM Access Analyzer*

*AWS Security Hub*

*AWS Inspector*

*AWS Config*

**Employ security tooling and services that automatically assess changes and discover new assets:**

*Asset Discovery*

*Configuration Monitoring*

**Implement security monitoring for your workloads that enables rapid response to security incidents and provides coverage for your architectures:**

*Covers all supporting services, from EC2 to AWS container services.*

*Integrates with AWS services for complete visibility, e.g. AWS CloudTrail.*

*Provides 24/7 response capabilities.*

*Incorporates the latest threat intelligence continuously to protect from new and emerging threats.*

While it is impossible to list every security measure and configuration that may be required for the myriad of ways customers use AWS services, we believe this list provides the fundamentals and methodology that can lead to a secure foundation.

With services that integrate tightly with AWS, providing security posture assessment, 24/7 security detection and response built on a platform providing comprehensive coverage for your workloads, Alert Logic MDR™ is the industry standard for securing AWS.

**Contact us at [www.AlertLogic.com](http://www.AlertLogic.com) to speak with one of our AWS security experts.**