

Cloud Security Guide

# AWS Checklist for 2022

Identity, Data, and Platform Security  
for the Well-Architected Framework  
Security Pillar



## Introduction

Amazon Web Services (AWS), the top public cloud service provider, offers a broad set of global compute, storage, database, analytics, application, and deployment services that help enterprises move faster, lower IT costs, and scale applications. While this is great for development, AWS remains one of the biggest cloud security threats in 2021, as companies face more sophisticated threats.

Environments are frequently at risk from cybercrime. For example, a large-scale operation called **TeamTNT** has been installing cryptocurrency-mining malware on misconfigured container platforms. The team was breaching container platforms by targeting Docker systems containing exposed, passwordless management API ports — enabling the group to mine for cryptocurrency at their victims' expense.

At the same time, companies are facing rising internal security issues due to misconfigurations and mismanagement. In **one recent high-profile example**, Prestige Software failed to appropriately configure an AWS S3 bucket, exposing a trove of data on the public internet.

Research from McKinsey shows that insiders are present in 50% of **cyber breaches** — and 44% of root causes can be attributed to negligence. Often, breaches occur when inexperienced or understaffed IT teams are asked to handle large scale cloud migrations. AWS can be incredibly complicated, and if you're new to the platform, it's effortless to make small mistakes that can lead to catastrophic consequences.

This document guides AWS customers by recommending best practices for the highest protection level for their AWS infrastructure and the sensitive data stored in AWS.

## AWS Shared Responsibility Model

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

AWS has made platform security a priority to protect customers' critical information and applications taking responsibility for its infrastructure's security. AWS detects fraud and abuse and responds to incidents by notifying customers. However, the customer is responsible for ensuring their AWS environment is configured securely and data is not shared with someone it shouldn't be shared with inside or outside the company, identifying when an identity human or non-human misuses AWS, and enforcing compliance and governance policies.

## AWS Responsibility

AWS is focused on the security of AWS infrastructure, including protecting its computing, storage, networking, and database services against intrusions because it can't fully control how its customers use AWS. AWS is responsible for the security of the software, hardware, and the physical facilities that host AWS services. Also, AWS takes responsibility for the security configuration of its managed services such as AWS DynamoDB, RDS, Redshift, Elastic MapReduce, WorkSpaces, and others.

## Customer Responsibility

AWS customers are responsible for the secure usage of AWS services that are considered unmanaged. For example, while AWS has built several layers of security features to prevent unauthorized access to AWS, including multi-factor authentication, it is the customer's responsibility to make sure multifactor authentication is turned on for users, particularly for those with the most extensive IAM permissions in AWS.

Furthermore, the default security settings of AWS services are often the least secure. Correcting misconfigured AWS security settings, therefore, is a low hanging fruit that organizations should prioritize to fulfill their end of AWS security responsibility.

As enterprises continue to migrate to or build their custom applications in AWS, the threats they face are no longer isolated like the old world of on-premises applications as **identities are the new perimeter**. Preventing many of these threats falls on the shoulders of the AWS customer. So how are you securing your data?

Below are checklists to help you govern and secure **your AWS**, including but not limited to the following:

	Customer Responsibility	AWS Responsibility
Preventing or detecting when an AWS account has been compromised	✓	
Preventing or detecting a privileged or regular AWS user behaving in an insecure manner	✓	
Business continuity management (availability, incident response)		✓
Protecting against AWS zero-day exploits and other vulnerabilities		✓
Providing environmental security assurance against things like mass power outages, earthquakes, floods, and other natural disasters		✓
Providing physical access control to hardware/software		✓
Configuring AWS Managed Services in a secure manner		✓
Database patching	✓	✓
Ensuring network security (DoS, man-in-the-middle (MITM), port scanning)	✓	✓
Ensuring AWS and custom applications are being used in a manner compliant with internal and external policies	✓	✓
Updating guest operating systems and applying security patches	✓	
Restricting access to AWS services or custom applications to only those users who require it	✓	
Configuring AWS services (except AWS Managed Services) in a secure manner	✓	
Preventing sensitive data from being uploaded to or shared from applications in an inappropriate manner	✓	

# Top Security AWS Cloud Security Challenges

Threats to applications running on AWS and the data stored within them can take many forms:

## 1. Compromise of AWS

AWS has made significant investments in security to protect its platform from intrusion. However, the slight possibility remains that an **attacker could compromise** an element in the AWS platform and either gain access to data, take an application running on the platform offline, or permanently destroy data.

## 2. Third-Party Account Compromise

According to the Verizon Data Breach Investigations Report, 63% of data breaches were due to a compromised account where the hacker exploited a weak, default, or stolen password. Misconfigured security settings or accounts with excessive identity and access management (IAM) permissions can increase the potential damage.

## 3. Sensitive Data Uploaded Against Policy

Many organizations have industry-specific, regional regulations, or internal policies that prohibit certain data types from being uploaded to the cloud. In some cases, data can be safely stored in the cloud, but only in specific geographic locations (for example, a data center in Ireland but not in the United States).

## 4. Software Development Lacks Security Input

Unfortunately, IT security isn't always involved in the development or security of custom applications. When it comes to their development, IT security is often circumvented, making the task of securing these applications more difficult.

## 5. Shadow IT

Shadow IT uses information technology systems, devices, software, applications, and services without explicit IT department approval. It has grown exponentially in recent years with the adoption of cloud-based applications and services. Compute departments other than the central IT department, to work around the shortcomings of the central information systems create a hidden risk.

## 6. Ephemeral Compute Pours Over Your Data

With container orchestration, the typical lifetime of a container is 12 hours. Serverless functions - already adopted by 22% of corporations - come and go in seconds. Data is the digital era's oil, but the oil rigs are ephemeral and countless in this era. EC2 instances, spot instances, containers, serverless functions, admins, and agile development teams are the countless fleeting rigs that drill into your data.

## 7. Unsecured Storage Containers

The news is filled regularly with attacks on **misconfigured cloud** servers and the leaked data that criminals obtain from them. This happens because of human error. Setting a cloud server with loose or no credentials and then forgetting to tighten them when the server is placed into production is a common mistake.

## 8. Cloud Data Sprawl

Gone are the days of a limited selection of manageable data stores. Innovations in agile cloud development have led to an explosion of new data store options, with teams utilizing Amazon MongoDB, Elasticsearch, Dynamo DB, HashiCorp Vault, and many more. Adding these to object stores makes it self-evident that new corporate infrastructures do not have a physical or logical concept of a 'data center.' This innovation can create cloud sprawl, where an organization has an uncontrolled proliferation of its cloud instances, services, or identities.

## Insider Threats and Privileged Identity Threats

The average enterprise experiences 10.9 insider threats each month and 3.3 privileged user threats each month. These incidents can include malicious and negligent behavior— ranging from taking actions that **unintentionally expose data** to the internet to **employees stealing data**.

## Manually Managing Access Rights

Keeping track of which users can access an application manually creates risk. You can't detect common privilege escalation attacks across your infrastructure manually. Also, you can create risk by giving too many admin rights to virtual machines and containers.

## Increase in Supply Chain

A lack of understanding in the chain of custody in your most-often-used projects can create a risk. Developers have lengthened their software supply chains by using more open-source tools. This means you must understand the trust relationship and protect the complete path that software takes through your entire development process and lifecycle. Lack of visibility in these trust relationships leads to unnecessary risk.

## Enable Root Account Access

Frequently, organizations still use an active root account and allow access via access keys. The root is the account that has access to all files and commands across the operating environment. Using a root access account is extremely dangerous.

## Assign Access Keys to a Root Account

Again, the root account shouldn't be used for any purpose. Assigning an access key to the root account is a recipe for disaster. If your company currently assigns access keys to any root account, put a stop to this immediately.

## Losing Track of Access Keys

Organizations often lose track of where AWS access keys are being used and by who, making it impossible to protect their environments.

## Improper Storage of Access Keys

Using the AWS Command Line Interface (AWS CLI) will automatically produce a text file on your local system — like your laptop or desktop — containing the Access Key and Access Secret Key in plain text for anyone to read or use to their advantage. You'll want to be sure you remove this file from your local system.

## Using Accounts Without Active MFA

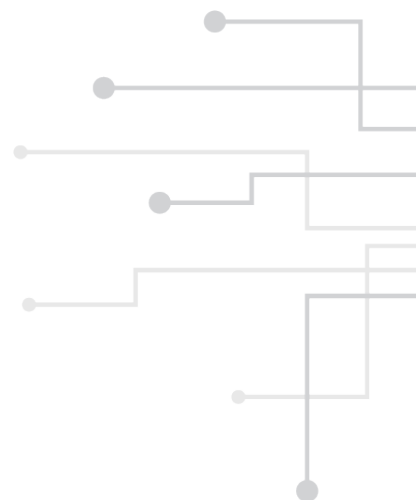
Research shows 81% of security incidents are caused by credential theft. All AWS accounts should be fortified with multi-factor authentication (MFA) to prevent getting hacked. Also, temporary credentials should be used in place of long-lived access keys — a feature that can be deployed using IAM roles.

# AWS Checklist

Amazon has built a set of security controls for its customers to use across AWS services, and it is up to the customer to make the most of these built-in capabilities. Here are best practices security experts recommend you follow:

- Enable CloudTrail logging across all of AWS.
- Turn on CloudTrail log file validation.
- Enable CloudTrail multi-region logging.
- Integrate CloudTrail with CloudWatch.
- Enable access logging for CloudTrail S3 buckets.
- Enable access logging for Elastic Load Balancer (ELB).
- Enable Redshift audit logging.
- Activate Virtual Private Cloud (VPC) flow logging.
- Require multi-factor authentication (MFA) to delete CloudTrail buckets.
- Turn on multi-factor authentication for the “root” account.
- Turn on multi-factor authentication for IAM users.
- Enable IAM users for multi-mode access.
- Attach IAM policies to groups or roles.
- Rotate IAM access keys regularly, and standardize on the selected number of days.
- Set up a strict password policy.
- Set the password expiration period to 90 days and prevent reuseCustomer Visualforce pages with standard headers.
- Don't use expired SSL/TLS certificates.
- User HTTPS for CloudFront distributions.
- Restrict access to CloudTrail buckets.
- Encrypt CloudTrail log files at rest.
- Encrypt Elastic Block Store (EBS) database.
- Provision access to resources using IAM roles.
- Control inbound and outbound traffic to your EC2 with structured security groups that don't have large ranges of ports open.
- Configure EC2 security groups to restrict inbound access to EC2.
- Protect EC2 Key Pairs.
- Avoid using root user accounts.

- Lock root user accounts and prevent anyone in the organization from accessing them.
- Use secure SSL ciphers when connecting between the client and ELB.
- Use secure SSL versions when connecting between client and ELB.
- Use a standard naming (tagging) convention for EC2.
- Encrypt AWS' Relational Database Service (RDS).
- Ensure access keys are not being used with root accounts.
- Use secure CloudFront SSL versions.
- Enable the require\_ssl parameter in all Redshift clusters.
- Rotate SSH keys periodically.
- Minimize the number of discrete security groups.
- Reduce the number of IAM groups.
- Terminate unused access keys.
- Rotate your keys regularly. This will reduce the risk of a compromised key. Do this even if someone has a "read-only" API key.
- Remove access keys that haven't been used in the last 90 days. (You can always create a new one.)
- Disable access for inactive or unused IAM users.
- Remove unused IAM access keys.
- Delete unused SSH Public Keys.
- Restrict access to Amazon Machine Images (AMIs).
- Restrict access to EC2 security groups.
- Restrict access to RDS instances.
- Restrict access to Redshift clusters.
- Restrict access to outbound access.
- Disallow unrestricted ingress access on uncommon ports.
- Restrict access to well-known ports such as CIFS, FTP, ICMP, SMTP, SSH, Remote desktop.
- Inventory and categorize all existing custom applications by the types of data stored, compliance requirements, and possible threats they face.
- Involve IT security throughout the development process.
- Grant the fewest privileges as possible for application users.
- Enforce a single set of data loss prevention policies across custom applications and all other cloud services.
- Encrypt highly sensitive data such as protected health information (PHI) or personally identifiable information (PII).
- Control access to S3 buckets.
- Don't create any public access S3 buckets.
- Activate S3 access logging.



## Are You Ready to Secure Your AWS Environment?

There's a lot to unpack here, and the truth is these are just a few issues you need to watch out for when using AWS. If you have questions, don't hesitate to reach out — Sonrai's technical team of security experts are standing by to help.

REQUEST A DEMO


## Learn More

Sonrai Security offers a demo for AWS organizations and brings you that much closer to data security and compliance. See how we identify holes, such as excessive privilege, escalations, separation of duty, and potential risks that may arise from the identities, permissions, and data exchange within and across clouds.

Request a demo today!

 [sonraisecurity.com](https://sonraisecurity.com)

 [info@sonraisecurity.com](mailto:info@sonraisecurity.com)

 646.389.2262





### **Legal Notice**

This document is provided for informational purposes only. It represents Sonrai Security practices as of the date of issue of this document, subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS' products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions, or assurances from Sonrai Security, its affiliates, suppliers, or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, Sonrai Security agreements, and this document is not part of, nor does it modify, any agreement between AWS, Sonrai Security, and its customers.