# AN INTRODUCTION TO AWS CLOUD SECURITY

## NOW INDISPENSABLE TO BUSINESS

Bishop Fox Guide - Updated August 2018 - G.K.

# CONTENTS

# INTRODUCTION

Amazon Web Services (AWS) isn't the novelty it was a decade ago. Resource-intensive, computer-heavy work today flows upward from giant enterprises 24/7 to the nebulous cloud where its processed by virtual servers, stored in digital containers, and eventually returned in a manner that supports the bottom line of tens of thousands of businesses.

## AWS IS INDISPENSABLE TO BUSINESS

AWS is an indispensable part of business strategy for companies worldwide that make use of its infrastructure, platform, and software services. For example, giant pharmaceuticals slice into the time it takes to complete clinical trials on billion-dollar drugs by spinning up instance after instance on the AWS platform. Financial services organizations—including NASDAQ—host solutions on AWS that keep data secure and available while maintaining compliance with stringent industry regulations. Hundreds of other organizations store and share data through AWS, allowing not only insiders to collaborate safely on projects, but also business partners, vendors, and contractors.

## AWS SECURITY IS A SHARED RESPONSIBILITY

Security is no longer the barrier to adoption it once was with the cloud; gone is the hesitancy of sending business-critical data off to the cloud to have it handled by third parties. Yet that doesn't relieve organizations from the responsibility of putting controls around their information and limiting access to it. Amazon may keep its data centers safe and the physical servers hosting millions and millions of virtual servers secure from hackers, but the onus on protecting data rests squarely on the AWS user's shoulders. This is the essence of Amazon's Shared Responsibility Model where it protects the infrastructure running its services while the customer is responsible for secure configurations of Elastic Compute Cloud EC2, for example, Amazon's infrastructure as a service offering.

This guide provides a brief introduction to securing Amazon Web Services installations, looking specifically at Amazon Simple Storage Service (S3), Elastic Compute Cloud (EC2), Identity and Access Management (IAM), and its Relational Database Service (RDS), the risks associated with each, and security best practices that should be followed to ensure the integrity of business data moving through each service.

## AMAZON WEB SERVICE INSTALLATIONS

# SIMPLE S3 ACCESS CONFIGURATION MISTAKE


Amazon Simple Storage Service, better known as S3, is one of the most popular abstracted AWS services, providing users with almost infinite storage and collaboration capabilities. For all of its convenience and efficiency, it's also one place where a whole lot of good can be undone by one simple mistake.

Massive S3 leaks have resulted in uncomfortable headlines for many companies, including some of the world's largest businesses, such as telecommunications giant Verizon and global management and consulting firm Accenture.

A Verizon partner failed to properly secure an S3 bucket and leaked personal information belonging to tens of millions of customers, while Accenture suffered what could have been a catastrophic leak of private encryption keys that could have been used to unlock private data for customers around the world.

## AVOID ACCESS CONFIGURATION MISTAKES

Simple mistakes, such as changing the default access configuration for an S3 bucket from private to public, were behind many of the leaks and forced organizations to re-evaluate the security of these installations and how to move forward. The scope of the risk posed by each leak is immeasurable. In each case, it's difficult to tell whether any of the data had been accessed or downloaded before the researchers stumbled upon it. Unless logging and other tracking had been turned on, it's nearly impossible to tell whether the exposed data had fallen into a malicious party's hands. Another commonality is that in many cases, a third party was responsible for the exposure. Data had been shared with a vendor, business partner, or contractor who needed a particular data set for a project and uploaded it to Amazon S3 without proper security controls in place, largely for the sake of convenience and sharing among colleagues.

Amazon recommends that administrators provision access based on the principle of least privilege. AWS makes four routes available to users for access provisioning:

### ROUTE 01

## IDENTITY AND ACCESS MANAGEMENT POLICIES (IAM)

Because of its ability to centrally manage access, IAM provides administrators with the strongest controls. It's advised to specify permissions in a policy, attach the policy to a group, and place users into groups to provide them with permissions. Policies should never directly apply to users because this would complicate management as the Amazon Web Services environment grows relative to its organization. In addition, use AWS managed policies rather than creating new ones. Many AWS managed policies exist that are intended to provide organizational roles with access according to the principle of least privilege.

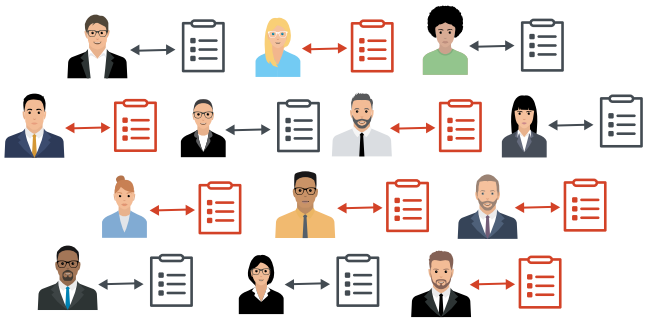


**ILLUSTRATION 01**

Unscalable way to apply permissions at the user level

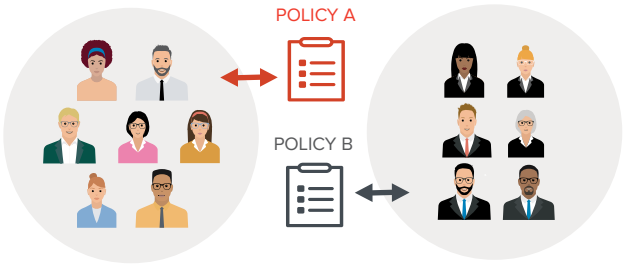



ILLUSTRATION 02

Advised way to specify permissions at the group level

## ROUTE 02

## BUCKET ACCESS CONTROL LISTS (ACL) FOR S3 BUCKETS

ACLs grant the Amazon S3 Log Delivery group write-access to the bucket. To have Amazon S3 deliver access logs to a user's bucket, an admin will need to grant write permission on the bucket to the Log Delivery group. The only way to grant necessary permissions to the Log Delivery group is via a bucket ACL.

## ROUTE 03

## BUCKET POLICIES

A bucket policy is applied to a bucket rather than a user or group, unlike an identity and access management policy. A bucket policy can be used to provide bucket access for another AWS account; IAM policies do not allow for cross-account access.

## ROUTE 04

## OBJECTS ACCESS CONTROL LIST (ACL)

Objects ACL for objects stored in a S3 bucket may also grant cross-account access to objects rather than buckets. Object ACLs should be used if access permissions need to vary between objects in a bucket in situations where cross-account access is required.

## AVOID CONTROL CONFIGURATION MISTAKES

Admins may also struggle with S3 controls known as Everyone and Authenticated User groups, which are quite similar despite their disparate names. Many newsworthy data leaks can be traced back to misconfigurations here, so it's advisable to proceed with caution. The confusion arises because the Authenticated User includes anyone authenticated to any AWS account, and not just an organizational AWS bucket. In other words, there is little difference between the "Everyone" and the "Authenticated Users" group because anyone can sign up for a free AWS account.

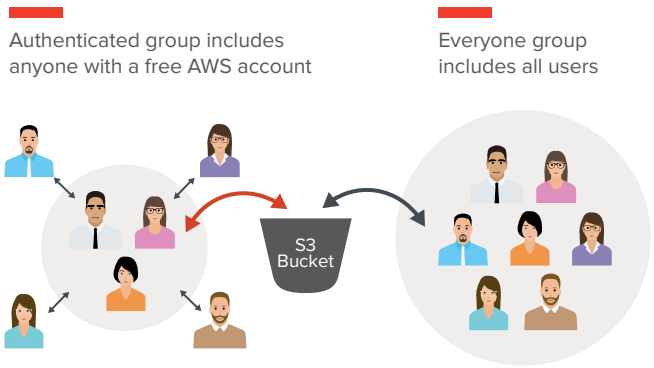



ILLUSTRATION 03

Authenticated User groups and Everyone groups are similar

If an S3 bucket is accessible to "Authenticated Users," all that someone needs to do is sign into their personal account and AWS will provide access to the S3 bucket objects. Alternatively, S3 buckets configured to grant access to the "Everyone" and the "Authenticated Users" groups can easily be accessed by an attacker via the AWS command line interface.

# THE ABCs OF EC2 SECURITY


Another popular AWS service is Amazon Elastic Compute Cloud (EC2). EC2 is the web service that allows for users and organizations to spin up virtual servers called instances, which are configured through Amazon Machine Images (AMI) to run applications at scale. It's a utility pricing-based model where a company can run thousands of virtual servers and applications without the overhead costs involved with buying and managing expensive hardware.

## EC2 SECURITY FALLS ON THE USER

The security of EC2 instances falls on the user organization, and it's imperative to keep instances safe in order to maintain the integrity of data moving through the virtual servers. Security happens at obvious layers with EC2: at the operating system, the network, and at the data layer. At the guest OS layer, which lives on an Amazon-owned and managed physical server, the customer is responsible for management tasks including updates and security patches, as well as the security of applications and other software running on a virtual server, and the configuration of security groups, an Amazon tool that acts as a stateful firewall.

## KEEP YOUR SECURITY GROUP TIGHT

Security groups can be attached to an EC2 instance, and a good rule of thumb is the application of the principle of least privilege to all groups. This ensures services only communicate with other services required for a particular business function. Amazon recommends avoiding practices that expand the attack surface of a virtual infrastructure, such as exposing access to ports and IP addresses for testing for example; this a common practice with traditional on-premises firewalls.

Unlimited, or broad, external access can decrease the integrity of network segmentation and needlessly expose critical services to attackers.

In addition, our experience has shown that organizations often have public subnets, such as a guest wireless network, included in the 10.0.0.0/8 address range. If a security group with this level of access is attached to a resource containing sensitive resources, that resource has just been exposed to those on the guest wireless network, despite not containing a rule set that provides direct external access. The exposure of critical services can also result from unnecessary complexity. To minimize confusion (and complexity) with security groups, best practice is to ensure security groups are created and named based on business function while following an accepted naming convention, unused security groups are deleted, and all rules are removed from default security groups.

## AVOID CONFUSION AND COMPLEXITY

- NAME GROUPS BASED ON BUSINESS FUNCTION
- REMOVE ALL RULES FROM DEFAULT GROUPS
- FOLLOW A NAMING CONVENTION
- DELETE UNUSED SECURITY GROUPS

Security groups can also be used to restrict outbound access through the modification of the corresponding ruleset, though it's not often implemented. By default, all security groups have an outbound rule set that allows traffic using all protocols over all ports to all IP addresses (0.0.0.0/0). Updating outbound rules is every bit as simple as updating inbound rules once you understand which connections your applications need to make. Implementing restricted rule sets for both inbound and outbound traffic on your security groups is an excellent step towards a layered approach to protecting your environment.

Access to an EC2 instance, meanwhile, can be secured using Amazon EC2 key pairs, public-key cryptography

that secures credentials used to authenticate users. Users must initially create a key pair, name it, and provide a private key that the user must securely store in order to connect to an instance; Amazon EC2 keys are 2048-bit RSA keys and users are allowed 5,000 key pairs per region. Creating key pairs is done through the Amazon console and is fairly straightforward under the Network & Security tab, but the critical part is properly saving the private key file. Key pair names must be provided when instances are launched, along with the corresponding private key each time a user connects. Amazon also allows organizations to import their own key pairs for EC2 that are created using third-party tools or standard programming language libraries. EC2 accepts imported keys in three formats:

**FORMAT 01**
openSSH public key

**FORMAT 02**
Base64 encoded DER

**FORMAT 03**
SHH public key file

Please note that DSA keys are not accepted by Amazon, only RSA.

## SECURE SENSITIVE INFORMATION

Another thing to consider is secure storage, particularly of sensitive information. Amazon provides Elastic Block Storage, a network-attached block-level storage service for EC2 in the form of EBS volumes, which are persistent data stores that support frequently updated storage. To back up this data, Amazon provides the ability to take snapshots of your EBS volumes. From these snapshots, you can create additional volumes, which are exact replicas of the original volume the snapshot was taken from.

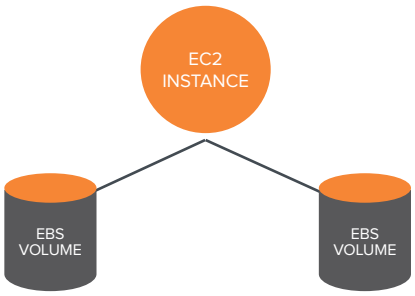



ILLUSTRATION 04
EBS volumes attached to the main EC2 instance

## ENCRYPT YOUR EBS VOLUMES

An EBS snapshot can also be created automatically when creating an Amazon Machine Image (AMI) from an EC2 instance for each EBS volume attached to the instance. Similar to AMIs, snapshots can also be shared with others. They can even be made public, but this does create a risk because of the amount of sensitive data potentially contained in a snapshot. A quick and easy way to protect your organization from one of your admins attempting to share a snapshot is to encrypt them. Not only is this a best practice, but it prevents the snapshot from being shared publicly. Instead, when attempting to share an encrypted snapshot, it is a requirement to specify the AWS accounts that the data is meant to be shared with. In addition, AWS makes this an extremely easy process through the AWS console.

# IAM IS A NECESSITY AND A PRIVILEGE


Given the off-premises nature of cloud computing, secure access to virtual resources and data is paramount. AWS provides its Identity and Access Management (IAM) service to ensure only authenticated parties have permission to access services and data moving through your virtual machines. Organizations need to think about access in two ways: managing access to AWS and managing access to services.

## MANAGING ACCESS TO AWS

Management starts at AWS account creation and the generation of credentials for a user's root account, which operates outside the reach of IAM controls; this should be used sparingly and only by a predetermined set of privileged users. Day-to-day access should be managed through the IAM service, and access granted to users should be done according to the principle of least privilege, which states that users should have only

the permissions required to satisfy the needs of their particular business role. The same principle should be applied for API access keys. When creating a new user, do not provide programmatic access by default. Programmatic access means that an access key ID and a secret access key are assigned to the user which can be used to issue AWS commands through the API. Not everyone needs this functionality so only assign API keys when a user's job function explicitly requires it.

### USE 12 CHARACTER PASSWORDS & MFA

The Amazon default password policy should also be changed immediately; it specifies a length of only six characters and no complexity requirements. NIST Special Publication 800-63-3 recommends a minimum of eight characters for privileged accounts, but since Amazon does not allow for the separate password policy designations of admin and user accounts, 12 is preferred. Complex passwords are not necessarily strong passwords; therefore, encourage users to choose passphrases that are considered secure and easier to remember. Also, passwords should not expire; this only encourages the poor practice of password reuse and less complex passwords being chosen because they are easier to remember. Instead, enforce multifactor authentication on accounts that can log into AWS.

### MANAGING ACCESS TO SERVICES

Access to services can be managed through IAM policies and groups, which define permissions granted to users and roles. Policies can be assigned to groups, for example, and users are then added to groups and absorb the policy. This centralizes the management of permissions around groups as opposed to individual users. If you create each group to fulfill the need of a specific job function, then updating permissions becomes very simple since everyone assigned to a given group should have the same job function. If a user changes job roles, they can now easily be granted new permissions by getting assigned to the relevant group or groups. In cases where a user requires special permissions unique to their specific role, AWS does still provide the ability to add policies directly to a user either by attaching an existing policy or creating an inline policy. Again though, best practice is to manage user access through groups, so this is not recommended unless absolutely necessary.

### TEST YOUR POLICIES

Amazon provides a tool called the IAM policy simulator which can be used to test policies attached to roles or users, as well as resources such as S3 buckets, SQS queues, or Amazon Glacier Vaults, for example. Policies can be tested before they're implemented, and even under real-world scenarios by providing some context to a test such as dates or IP addresses.

## SECURING THE KEYS TO THE KINGDOM IN RDS

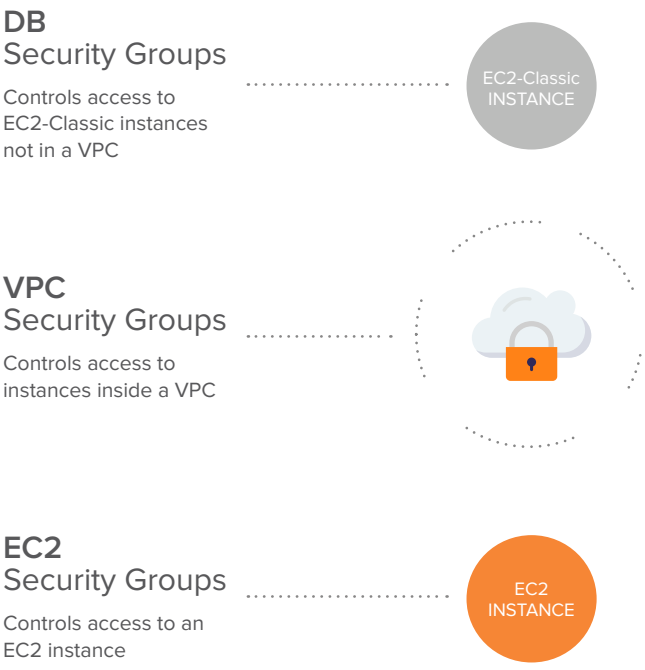

### USE A MANAGED RELATIONAL DATABASE

Amazon's Relational Database Service (RDS) is a web service that organizations may use to provision a cloud-based relational database. Organizations may use database software they're familiar with such as the open source MySQL, MariaDB, or PostgreSQL database, as well as proprietary software. The service manages backups, patching, failure and corruption detection, and recovery. Security should be the top priority given the criticality of a database to the business and the integrity of the data within.

## SECURE THE CONNECTIONS TO YOUR DATABASES

Organizations need to start with secure, private connections to databases, especially from on-premises databases, via a virtual private network, for example. This allows an administrator to control, for example, the IP range and ports allowed to access the AWS RDS implementation. Databases can also be kept private and accessible only from within your organization's cloud infrastructure. If the decision is made to make a database public, restrictive security groups should be used to enhance controls.

Three types of security groups are used within Amazon RDS: DB security groups, which control access to EC2-Classic instances not in a virtual private cloud (VPC); a VPC security group controls access to instances inside a VPC; and an EC2 security groups that controls access to an EC2 instance.




A database administrator must also be assigned and granted necessary privileges, and AWS IAM policies should be used to assign access and administrative permissions. Default master users are given certain privileges for a particular instance with pre-defined roles such as superuser, administrator, or user, and privileges that allow that individual to create, update, drop, and manage connections and much more for all six database engines supported by RDS, including Oracle, Microsoft SQL Server, and Amazon Aurora, in addition to the previously mentioned open source databases.

## SECURE DATA AT REST

Securing data at rest is a crucial component of RDS security, which means encrypting data stored in RDS instances and snapshots at rest. Data is encrypted on the server that hosts your organization's RDS instance using AES-256. Once data is encrypted, Amazon RDS transparently manages authentication of access and decryption of data. All of this can be managed through the Amazon RDS console by choosing yes for Enable Encryption. Encryption is applied to the instance storage, and also to backups, snapshots, and read-only replicas. Encryption in transit is handled using SSL to secure the connection to all database engines supported by AWS RDS.

# CONCLUSION

The advantages afforded by cloud computing in many instances outweigh the enterprise's former dependence on managing and storing homegrown applications and services in a personal data center. Most organizations have migrated some percentage of applications to cloud-based service providers such as Amazon Web Services and are benefiting from the flexibility, utility billing, and security afforded by these platforms. Security in AWS is a shared responsibility with Amazon assuming care over the security of its infrastructure, while the user is responsible for the integrity and security of its data. Organizations that choose to migrate applications and data to the cloud must do so knowing how to manage access to cloud-based resources, as well as the security of network-based connections to data and servers processing that data, the protection of data at rest and in transit. While Amazon provides a number of tools and services to help with security, it's important to understand the set of security best practices specific to cloud-based security and how to best apply them to an AWS environment.

# ADDITIONAL RESOURCES

AWS Security Best Practices Guide, August 2016

Amazon EC2 Key Pairs

Access Control List (ACL) Overview

Amazon EC2 Security Groups for Linux Instances

Shared Responsibility Model

AWS Identity and Access Management (IAM)

AWS IAM Frequently Asked Questions

Testing IAM Policies with the IAM Policy Simulator

Must-Know Features of Amazon RDS: Security & Encryption

Overview of Managing Access Permissions to Your Amazon RDS Resources

Using Identity-Based Policies (IAM Policies) for Amazon RDS

MEET GERBEN KLEIJN

# ABOUT
# THE AUTHOR

Gerben Kleijn is a Senior Security Analyst at Bishop Fox. In this role, Gerben focuses on compliance gap assessments, cloud deployment reviews, as well as firewall and VPN reviews. He also has significant experience with security monitoring and alerting. Gerben has worked on both the offensive and defensive side of security.

BISHOP FOX.

**Bishop Fox provides security consulting services to the Fortune 1000 and high-tech startups.**

We find problems before the bad guys do.

Find out more at bishopfox.com.

Keep in touch with the foxes on Twitter @bishopfox and on LinkedIn Bishop Fox