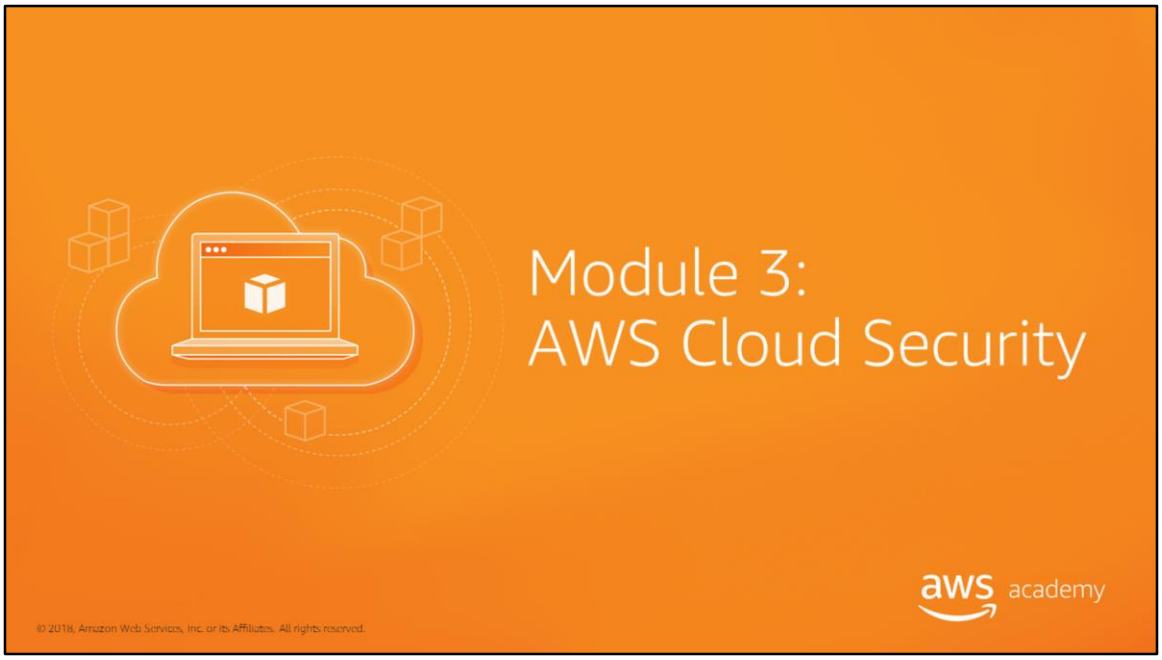
AWS Academy - Cloud Foundations
Module 03 Student Guide
Version 1.0.0

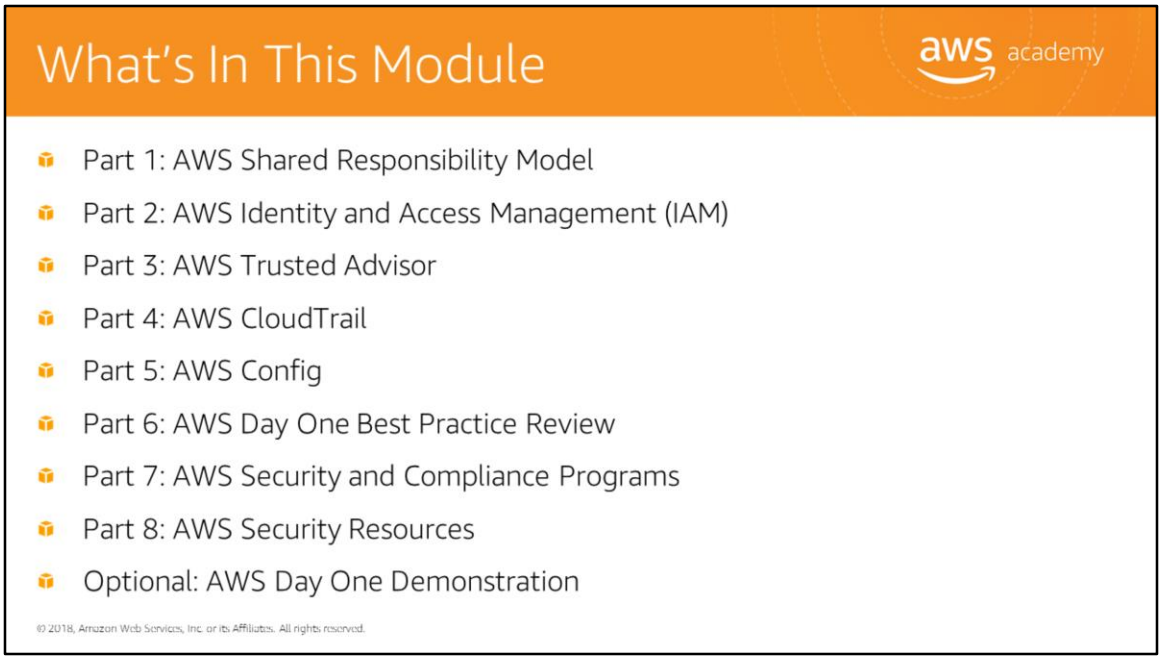
100-ACFNDS-10-EN-SG

Corrections or feedback on the course, please email us at:
aws-course-feedback@amazon.com.
For all other questions, contact us at:
https://aws.amazon.com/contact-us/aws-training/.

Contents



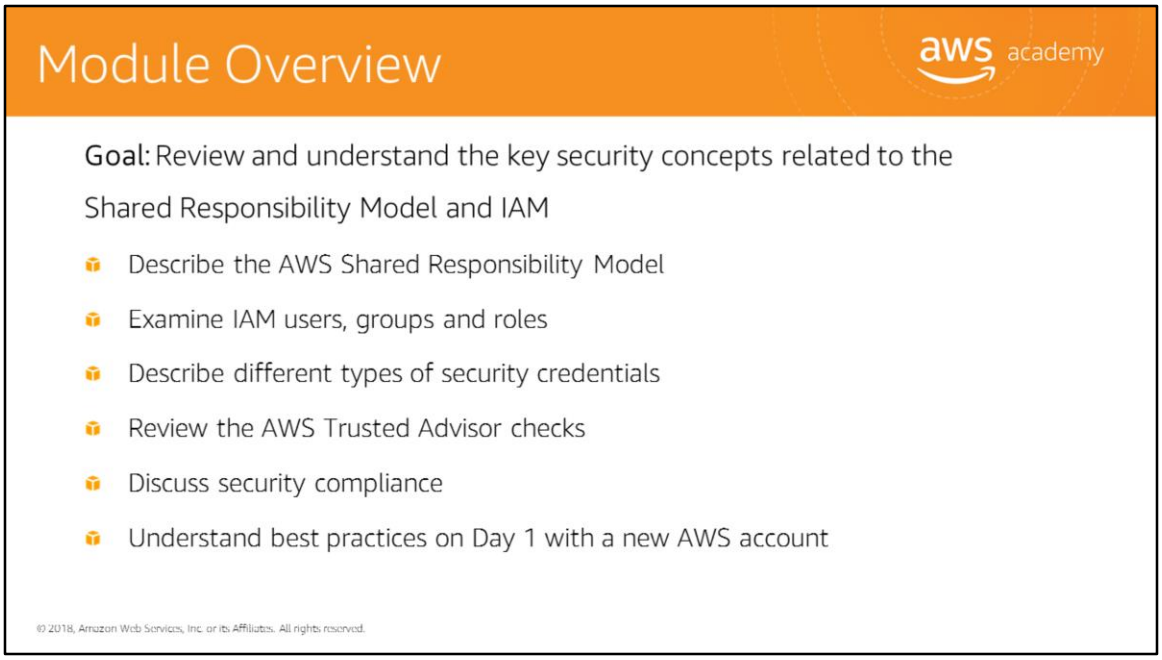
Welcome to Module 3 – AWS Cloud Security.



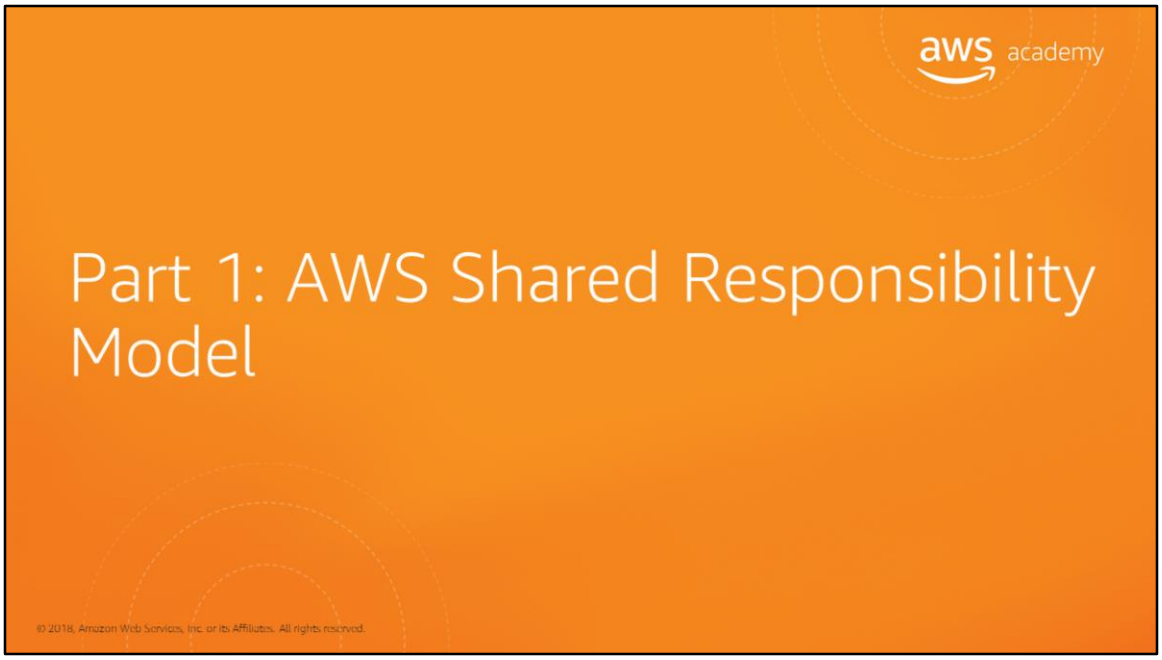
Security is of the utmost importance to AWS. AWS delivers a scalable cloud computing environment designed for high availability and dependability, providing the tools that enable you to run a wide range of applications. Helping to protect the confidentiality, integrity, and availability of your systems and data is critical to AWS, as is maintaining customer trust and confidence. This module is intended to provide an introduction to AWS's approach to security, including the controls in the AWS environment and some of the products and features that AWS makes available to customers to meet their security objectives.

In this section, we will:
Part 1: Review the Shared Responsibility Model
Part 2: Examine IAM including users, groups, and roles
Parts 3, 4 and 5: Explore AWS Trusted Advisor, AWS CloudTrail, and AWS Config
Part 6: Review Day One best practices
Part 7 & 8: Discuss security and compliance issues and resources

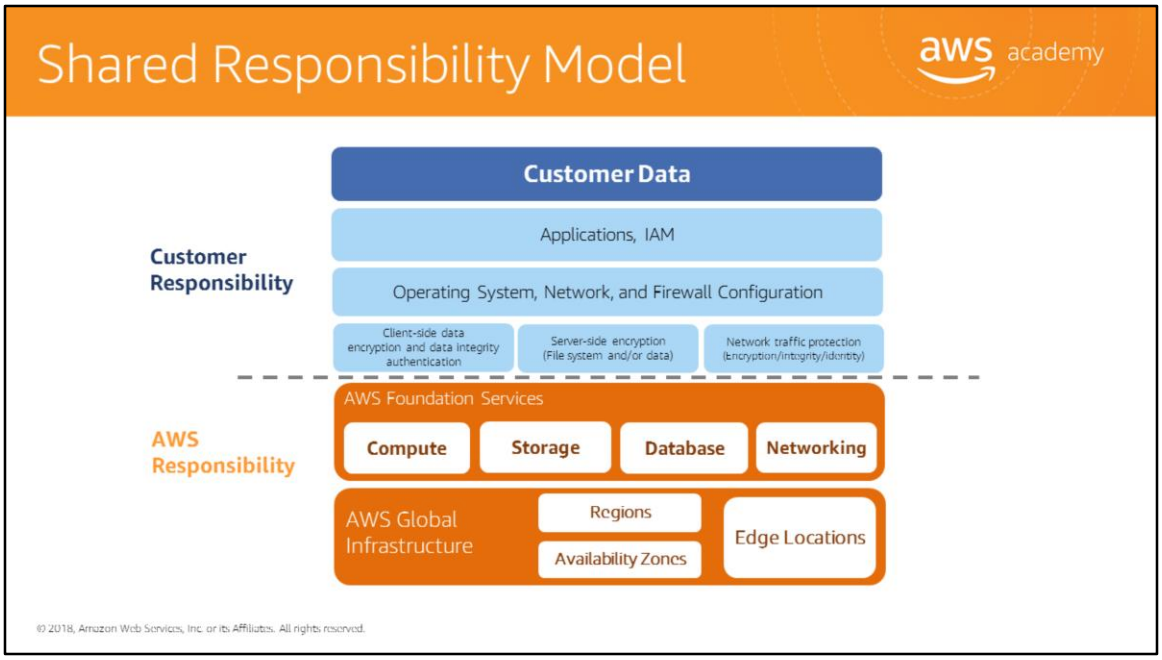
An optional step by step walk though of Day is available at the end of this module.



The goal of this module is to familiarize you with all of the security considerations for your cloud solution. We review security tools and best practices to help you understand the things that should be address as you architect a cloud solution. The review of Day 1 best practices will walk you through the best practices for setting up an account from beginning to end.



Welcome to Part 1 of the AWS Security module, the AWS Shared Responsibility Model.

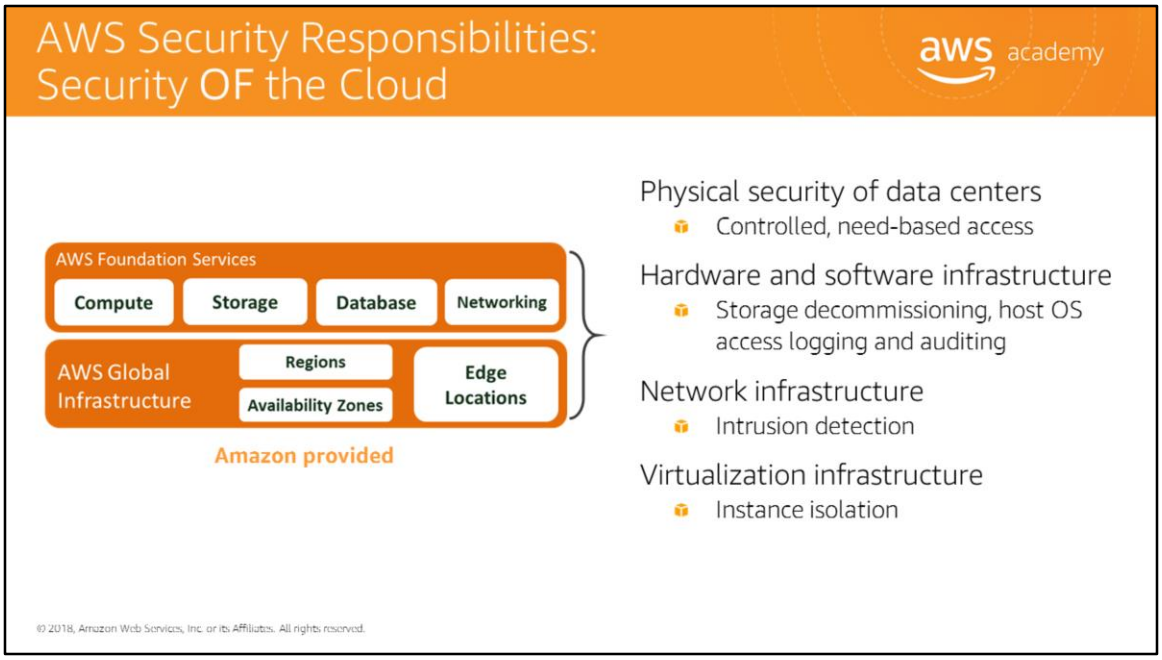


AWS provides the same approach to security that companies have been using for decades—while allowing the flexibility and low cost of cloud computing. There is nothing inherently inconsistent about providing on-demand infrastructure while also providing the security isolation that companies expect in their existing, privately owned environments.

Shared Responsibility Model

Once the customer starts using AWS, Amazon shares the responsibility of securing the customer's data in AWS cloud with its customers, making AWS security a shared responsibility. This concept is known as the Shared Responsibility Model.

Let's take a closer look at who's responsible for which aspects of security in the Shared Responsibility Model.



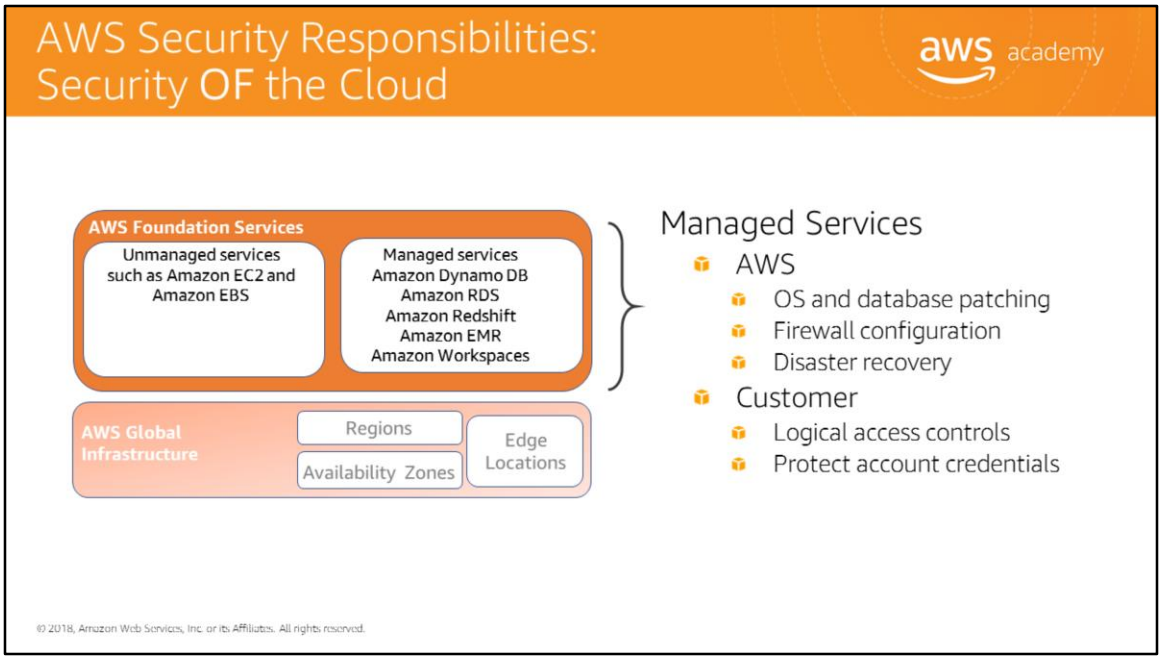
Security of the Cloud
AWS is responsible for security ***of*** the cloud. But what does that mean?

Under the shared responsibility model, AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the services operate. It means that AWS is responsible for protecting the global infrastructure that runs all of the services offered in the AWS cloud which include AWS Regions, Availability Zones, and edge locations.

For example, AWS handles the security of the cloud—specifically, the physical infrastructures that host your resources.

- **Data centers:** Nondescript facilities, 24/7 security guards, two-factor authentication, access logging and review, video surveillance, and disk degaussing and destruction
- **Hardware infrastructure:** Servers, storage devices, and other appliances that AWS services rely on
- **Software infrastructure:** Host operating systems, service applications, and virtualization software
- **Network infrastructure:** Routers, switches, load balancers, firewalls, cabling, etc. (including continuous network monitoring at external boundaries, secure access points, and redundant infrastructure)

Protecting this infrastructure is the number one priority for AWS. While you can't visit AWS data centers or offices to see this protection firsthand, Amazon provides several reports from third-party auditors who have verified our compliance with a variety of computer security standards and regulations.



Note that in addition to protecting this global infrastructure, AWS is responsible for the security configuration of its products that are considered foundational or managed services, which include Compute, Storage, Database, and Networking. These services include Amazon DynamoDB, Amazon RDS, Amazon Redshift, Amazon Elastic MapReduce, Amazon WorkSpaces, and others.

For these services, AWS will handle basic security tasks like operating system and database patching, firewall configuration, and disaster recovery. As a customer, this is valuable because you do not need to worry about patching, maintaining or installing antivirus software. Amazon takes care of it so the customer can focus on what goes into the environment. For most of the managed services, the customer needs to configure logical access controls and protect account credentials. Some managed services may require additional tasks such as setting up database user accounts, but overall the security configuration work is performed by AWS.

Here are some examples of controls that are managed by AWS, AWS customers, or both.

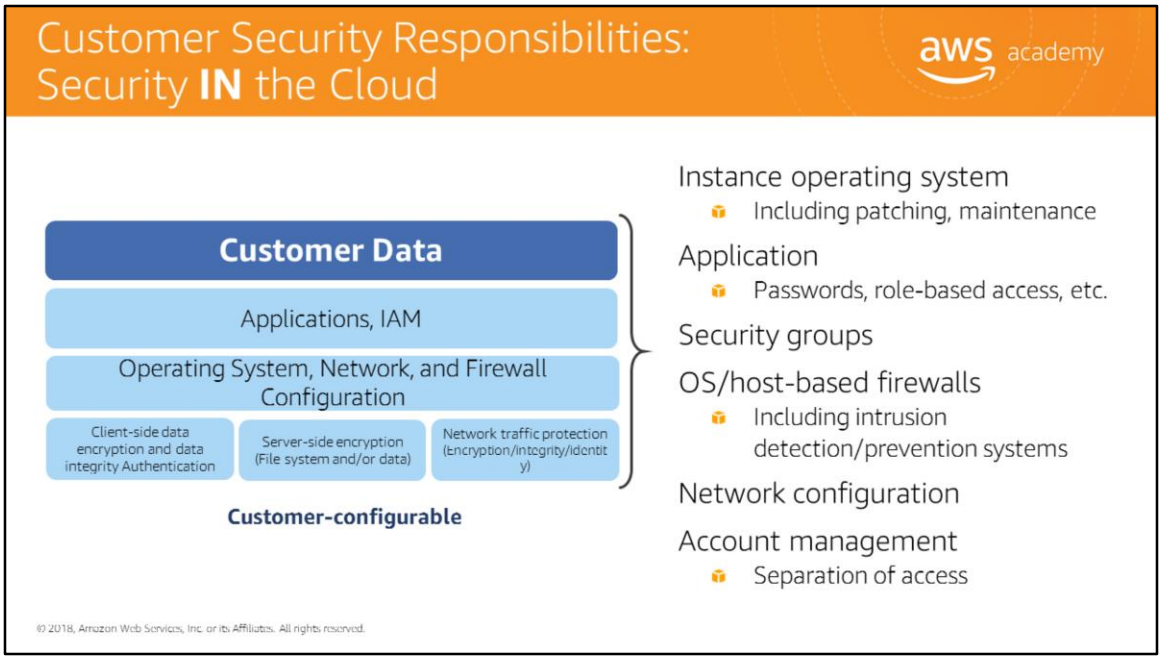
Inherited Controls – Controls that a customer fully inherits from AWS, such as physical and environmental controls

Shared Controls – Controls that apply to both the infrastructure layer and customer layers, but in completely separate contexts or perspectives. In a shared control, AWS provides the requirements for the infrastructure, and the customer must provide their own control implementation within their use of AWS services. Examples include:

- **Patch Management** – AWS is responsible for patching and fixing flaws within the infrastructure, but customers are responsible for patching their guest OS and applications.
- **Configuration Management** – AWS maintains the configuration of its infrastructure devices, but customers are responsible for configuring their own guest operating systems, databases, and applications.
- **Awareness and Training** - AWS trains AWS employees, but a customer must train their own employees.

Customer-specific – Controls that are solely the responsibility of the customer based on the application they are deploying within AWS services. Examples include:

- **Service and Communications Protection** or **Zone Security**, which may require a customer to route or zone data within specific security environments.

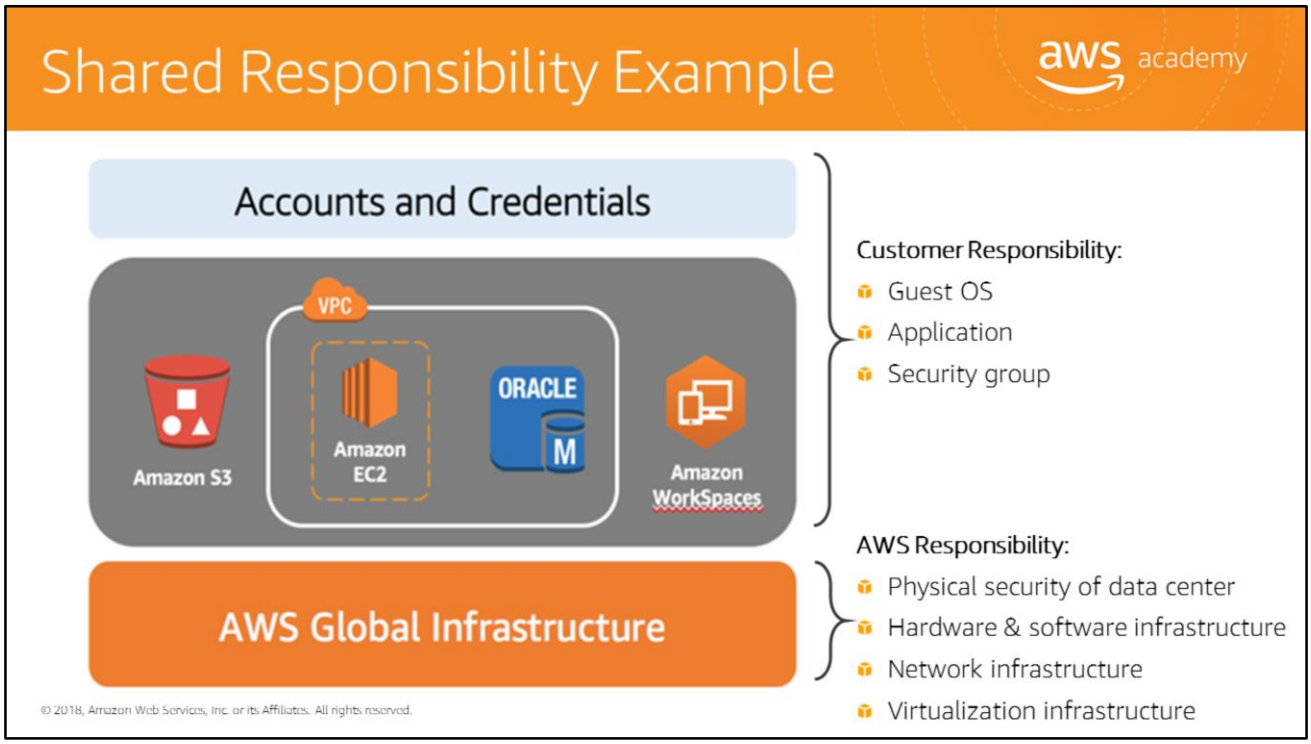


While the cloud infrastructure is secured and maintained by AWS, customers are responsible for security of everything they put ***in*** the cloud. The customer is responsible for what is implemented using AWS and for the applications connected to AWS. The security steps you need to take depend on the services you use and the complexity of your system.

When using AWS services, customers maintain complete control over their content and are responsible for managing critical content security requirements, including:

- What content they choose to store on AWS
- Which AWS services are used with the content
- In what country that content is stored
- The format and structure of that content and whether it is masked, anonymized, or encrypted
- Who has access to that content and how those access rights are granted, managed, and revoked

Customers retain control of what security they choose to implement to protect their own data, environment, applications, IAM, and operating systems. This basically means that the Shared Responsibility Model changes depending on the AWS services the customer uses.



In order to visualize the AWS Shared Responsibility Model, let's take a look at an example.

Let's say a customer is using Amazon S3 for storage and Amazon Workspaces for desktop and application streaming. They also have a VPC which consists of their Amazon EC2 instance and Oracle database instance.

AWS is responsible for protecting the global infrastructure that runs all of the services offered in the AWS cloud. The AWS global infrastructure is designed and managed according to security best practices as well as a variety of security compliance standards.

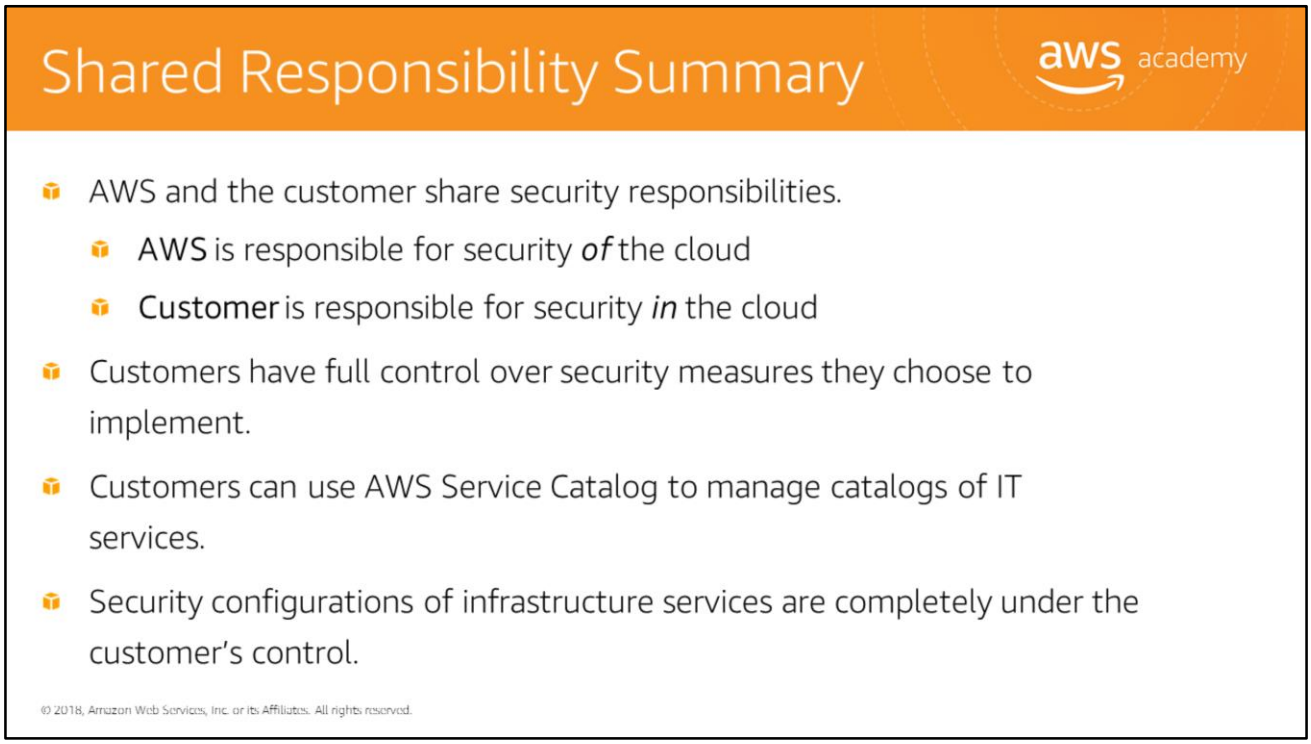
AWS products that fall into the category of Infrastructure as a Service (IaaS), such as Amazon EC2 and Amazon VPC, are completely under your control and require you to perform all the necessary security configuration and management tasks. For example, for EC2 instances, you are responsible for management of the guest OS (including updates and security patches), any application software or utilities you install on the instances, and the configuration of the AWS-provided firewall (called a security group) on each instance. These are basically the same security tasks that you're used to performing no matter where your servers are located.

Amazon RDS database instances like Oracle are pre-configured with parameters and settings appropriate for the engine and class you have selected. AWS manages time-consuming database administration tasks including provisioning, backups, software patching, monitoring, and hardware scaling. You focus on the tasks need for application development.

Amazon WorkSpaces is a fully managed, secure Desktop-as-a-Service (DaaS) solution. You provision virtual,

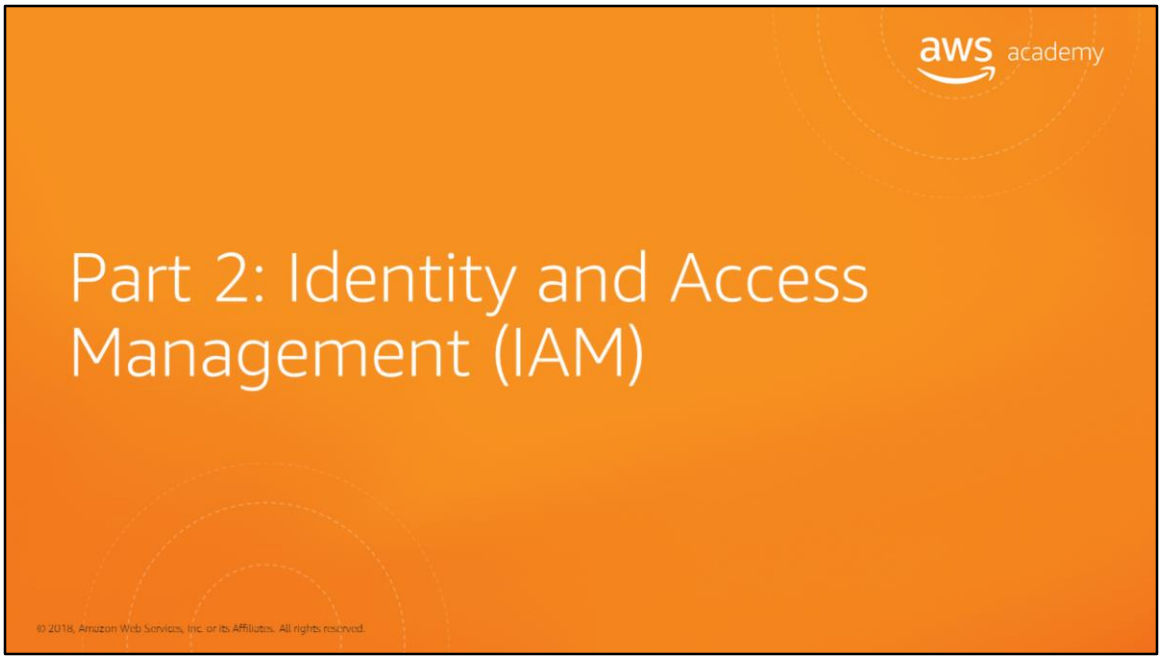
cloud-based Microsoft Windows desktops for your users, providing them access to the documents, applications, and resources they need, anywhere, anytime, from any supported device.

It is best practice for customers to protect their AWS account credentials and set up individual user accounts with IAM so that each user has their own credentials.

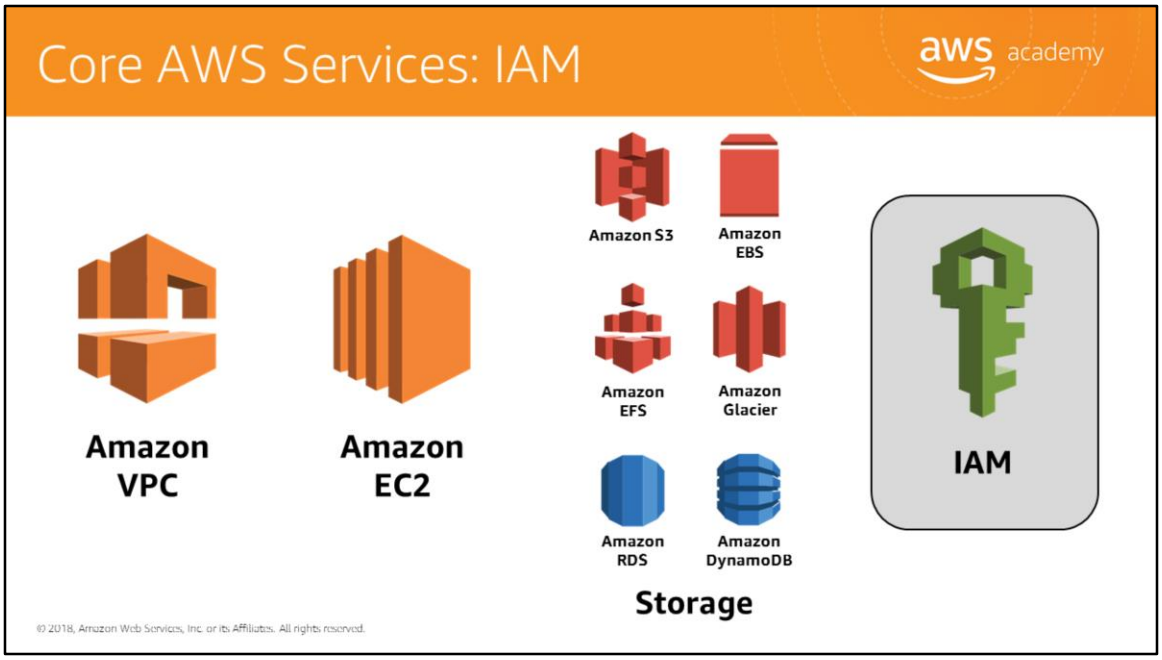


Let's quickly review the key points about the shared responsibility model:

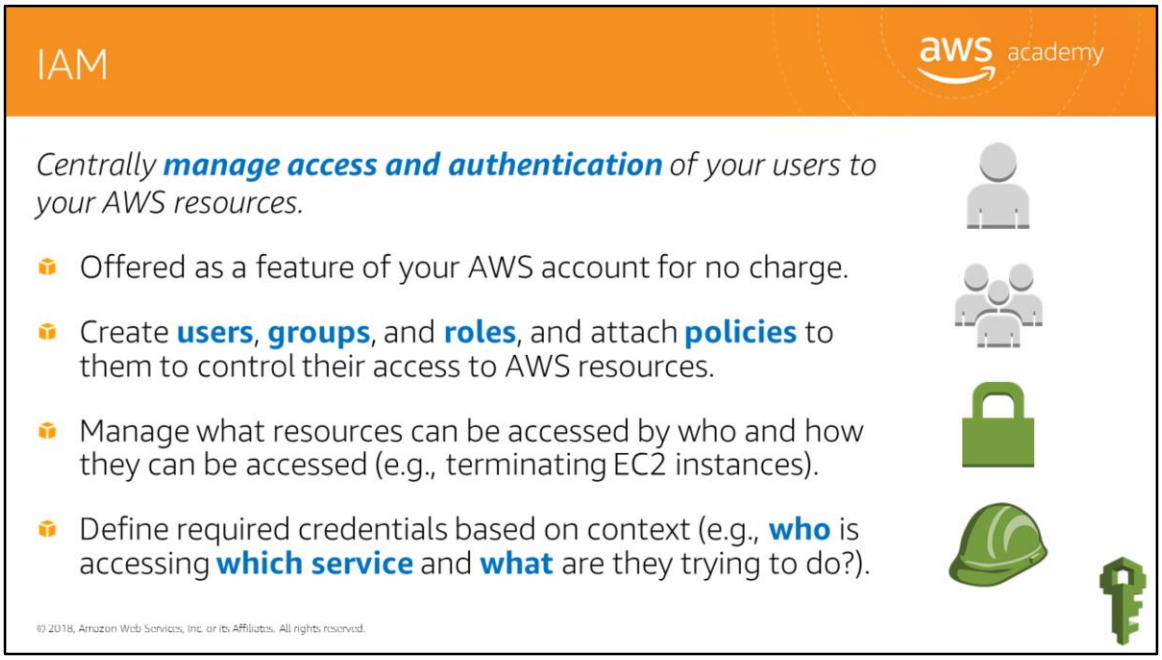
- The Shared Responsibility Model consists of AWS and the customer working together to secure data in the cloud. AWS is responsible for security *of* the cloud, while the customer is responsible for security *in* the cloud.
- Customers have full control of what security they choose to implement, given the AWS services they are using.
- Customers can use AWS Service Catalog to create and manage catalogs of IT services that have been approved for use on AWS.
- AWS products that fall into the category of IaaS, such as Amazon EC2 and Amazon VPC, are completely under the customer's control—and that requires them to perform all of the necessary security configuration and management tasks.
- By applying the Shared Responsibility Model, AWS and its customers can ensure secure and compliant data.



Welcome to Part 2 of the AWS Security module, AWS Identity and Access Management (IAM).



IAM allows you to control access to compute, storage, database, and application services in the AWS cloud (this is known as *authentication*) and how they can use resources (known as *authorization*). IAM uses access control concepts you will already be familiar with such as users, groups and permissions so you can specify which users get to access which services.



IAM is a tool to centrally manage access to launching, configuring, managing, and terminating your resources. It allows extremely granular control over access permissions, not just based on resource, but all the way down to determining exactly which API calls for each service can be made.

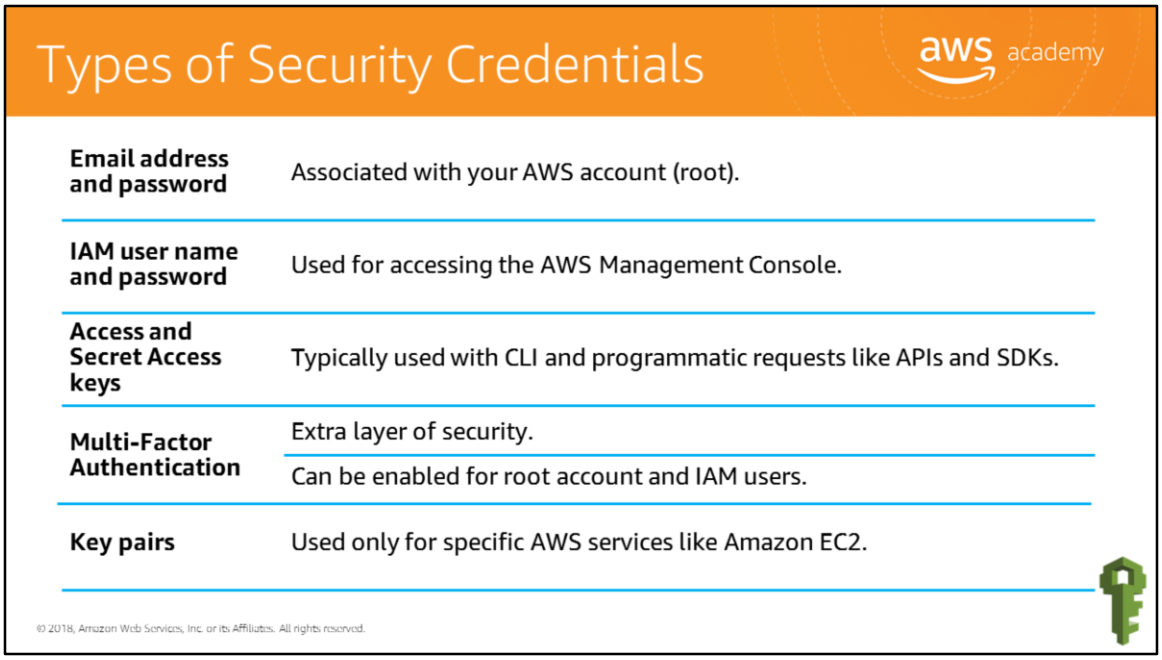
Think of the access control concepts you're already familiar with such as users (think of these as your end users), groups (think of these as a collection of users by job function), permissions (which can be applied to users or groups), and roles (think of these as trusted entities). That's exactly what IAM uses which makes it so powerful!

IAM Functionality

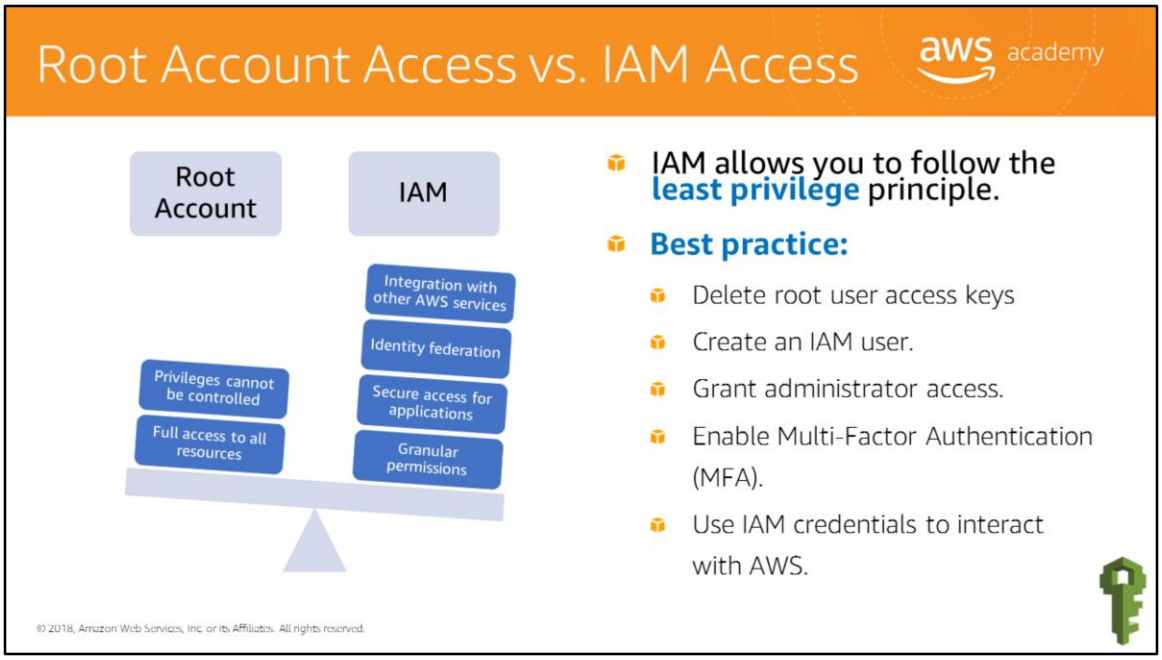
Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources. IAM allows you to:

- **Manage IAM users and their access** – You can create users in IAM, assign them individual security credentials (in other words, access keys, passwords, and multi-factor authentication devices), or request temporary security credentials to provide users access to AWS services and resources. You can manage permissions in order to control which operations a user can perform.
- **Manage IAM roles and their permissions** – You can create roles in IAM and manage permissions to control which operations can be performed by the entity, or AWS service, that assumes the role.

- **Manage federated users and their permissions** – You can enable identity federation to allow existing identities (users, groups, and roles) from your corporate directory to access the AWS Management Console, call AWS APIs, and access resources, without the need to create an IAM user for each identity.

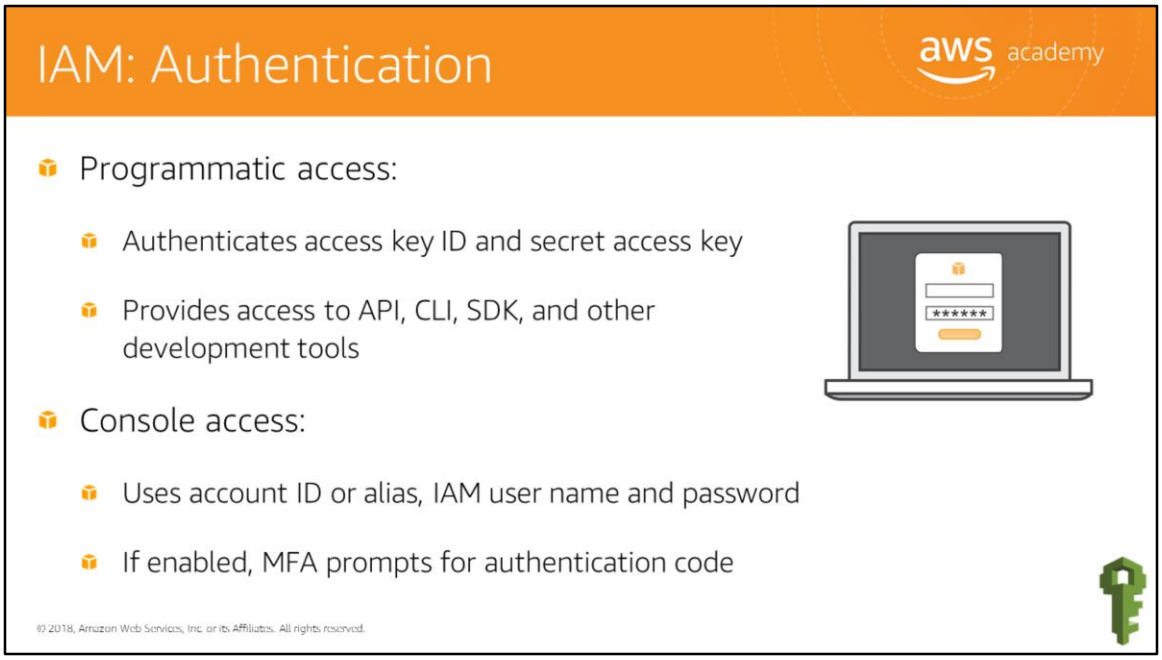


You use different types of security credentials depending on how you interact with AWS. For example, to sign in to the console, you use a user name and password. In contrast, to make programmatic calls to AWS Application Programming Interface (API) actions, you use access keys. The table on the slide summarizes the different types of AWS security credentials and when you might use each one.



When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account. AWS root accounts have ***full*** access to all resources in the account, and you cannot control the privileges of the root account credentials. Therefore, AWS strongly recommends that you not use root account credentials for day-to-day interactions with AWS.

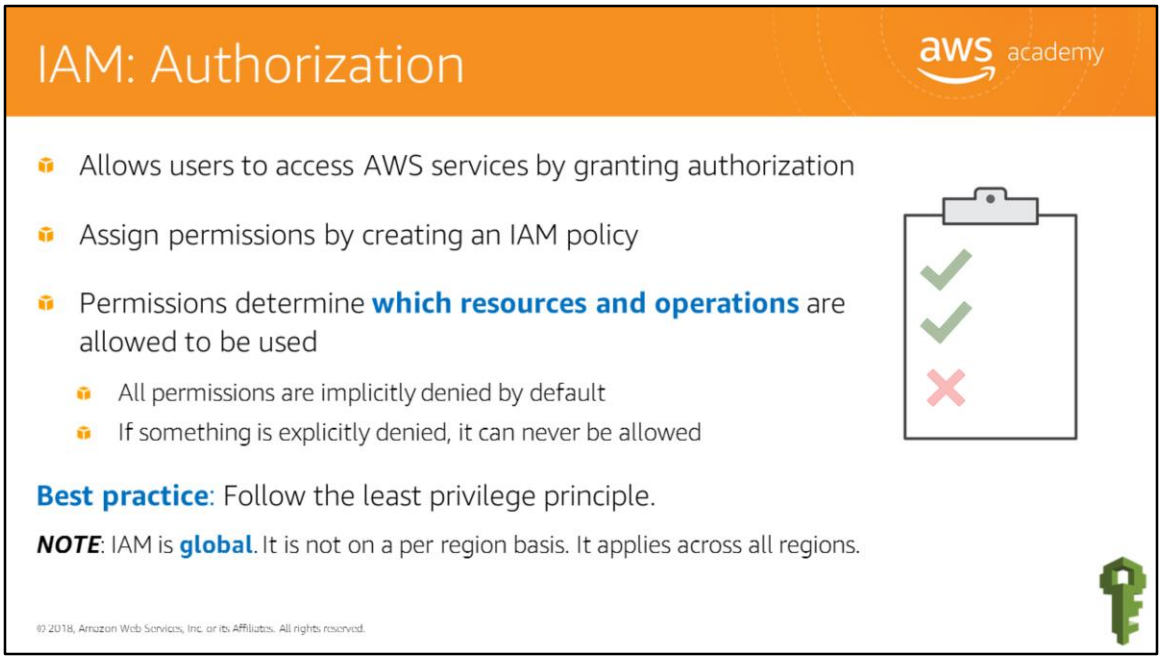
Use IAM to create additional users and assign permissions to these users, following the least privilege principle. With IAM, you can securely control access to AWS services and resources for users in your AWS account. For example, if you require administrator-level permissions, you can create an IAM user, grant that user full access, and then use those credentials to interact with AWS. Later, if you need to revoke or modify your permissions, you can delete or modify any policies that are associated with that IAM user. Additionally, if you have multiple users that require access to your AWS account, you can create unique credentials for each user and define who has access to which resources. In other words, you don't need to share credentials. For example, you can create IAM users with read-only access to resources in your AWS account and distribute those credentials to users that require read access.



AWS Authentication
When adding users, you get to select how users will access AWS. There are two different types of access you can assign users: Programmatic access and AWS Management Console access

Programmatic access enables an access key ID and secret access key for the AWS API, CLI, SDK, and other development tools.

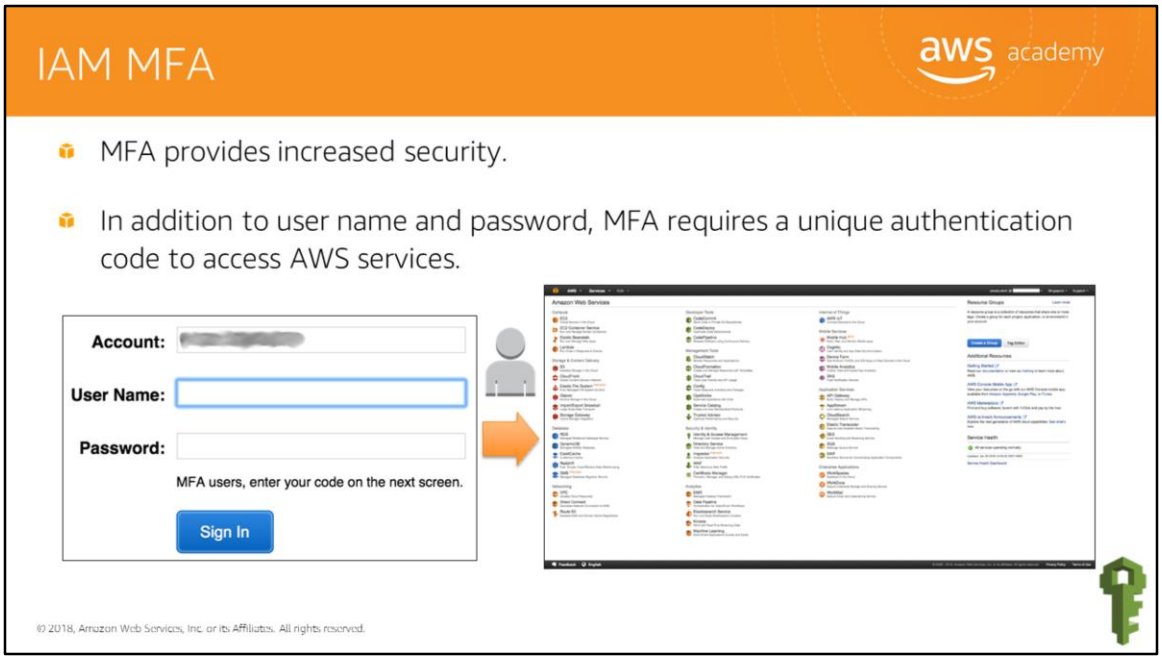
Another option is to give the user console access, which allows them to sign in to the console. The console provides a simple web interface for AWS. You can log in using your AWS account name and password. If you've enabled Multi-Factor Authentication (MFA), you will be prompted for your device's authentication code.



IAM Authorization
After a user has been authenticated, they then must be authorized to access an AWS service.

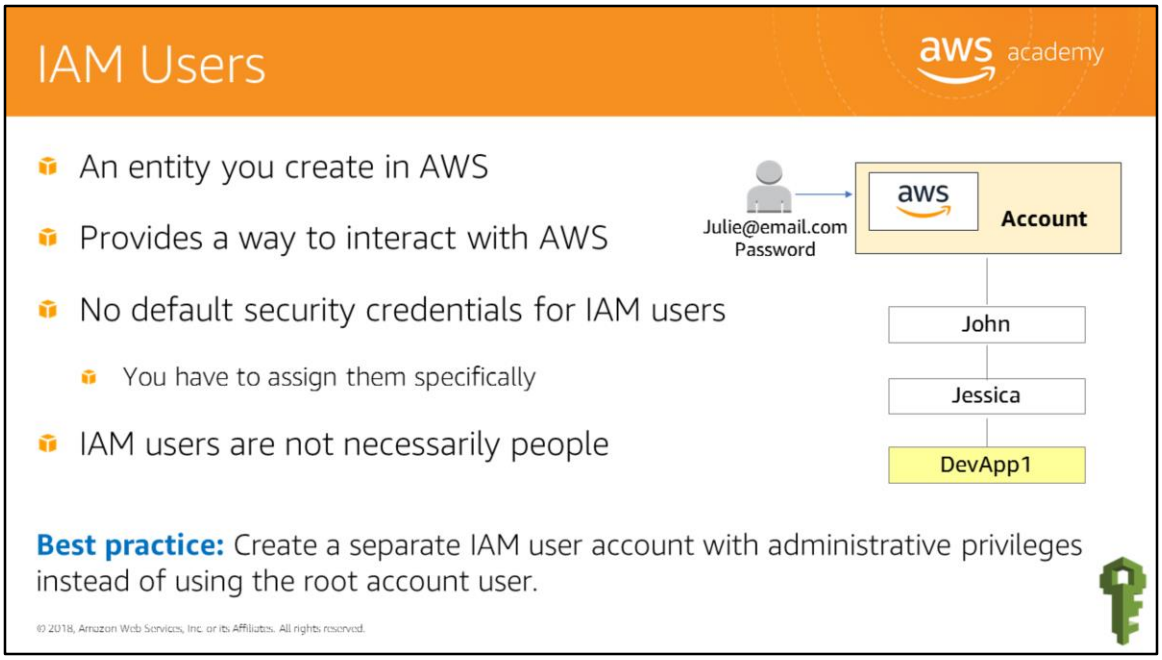
In order to assign permission to a user, group or role, you have to create an IAM policy, which is a document that explicitly lists permissions. There are no default permissions. All actions are denied by default (*implicit deny*) unless they are explicitly allowed. Any actions that you didn't explicitly allow are denied. Any actions that you explicitly deny are always denied.

The principle of least privilege is an important concept in computer security, promoting minimal user profile privileges based on users' job necessities. When you create IAM policies, follow the standard security advice of granting *least privilege*—that is, granting only the permissions required to perform a task. Determine what users need to do and then craft policies for them that let the users perform *only* those tasks. Start with a minimum set of permissions and grant additional permissions as necessary. Doing so is more secure than starting with permissions that are too lenient and then trying to tighten them later.



AWS services and resources can be accessed using the console, CLI, or through SDKs and APIs from a wide range of supported environments. For increased security, we recommend enabling MFA. With MFA, users and systems have to be authenticated before they can access AWS services and resources. There are two options for authentication devices: hardware devices and virtual MFA-compliant applications (Google Authenticator, Authy 2-Factor Authentication). SMS is another authentication alternative where you use your mobile device that can receive Short Message Service (SMS) messages to receive a code.

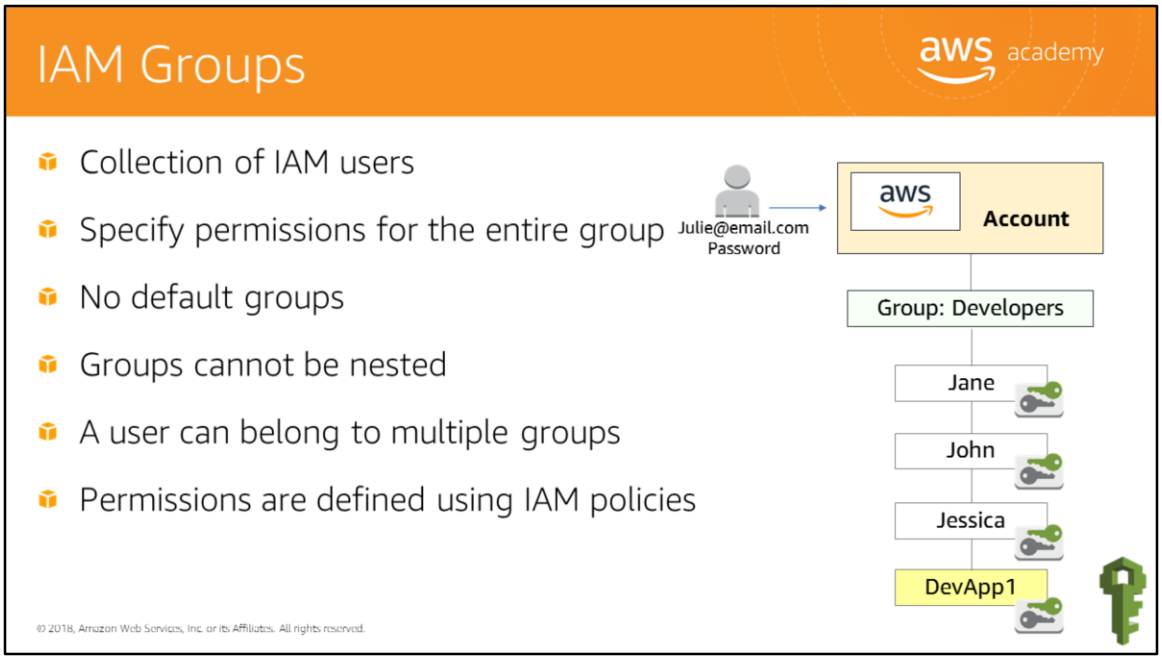
The AWS Security Token Service (STS) is a web service that also enables you to request temporary, limited-privilege credentials for IAM users or for users that you authenticate. For more information see https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa.html.



An IAM *user* is an entity that you create in AWS that provides a way to interact with AWS. An IAM user primarily gives people you work with identities that they can use to sign in to the console and make requests to AWS services.

Newly created IAM users have no default credentials to use to authenticate themselves and access AWS resources. You first need to assign security credentials to them for authentication and then attach permissions authorizing them to perform any AWS actions or to access any AWS resources. The credentials you create for users are what they use to uniquely identify themselves to AWS.

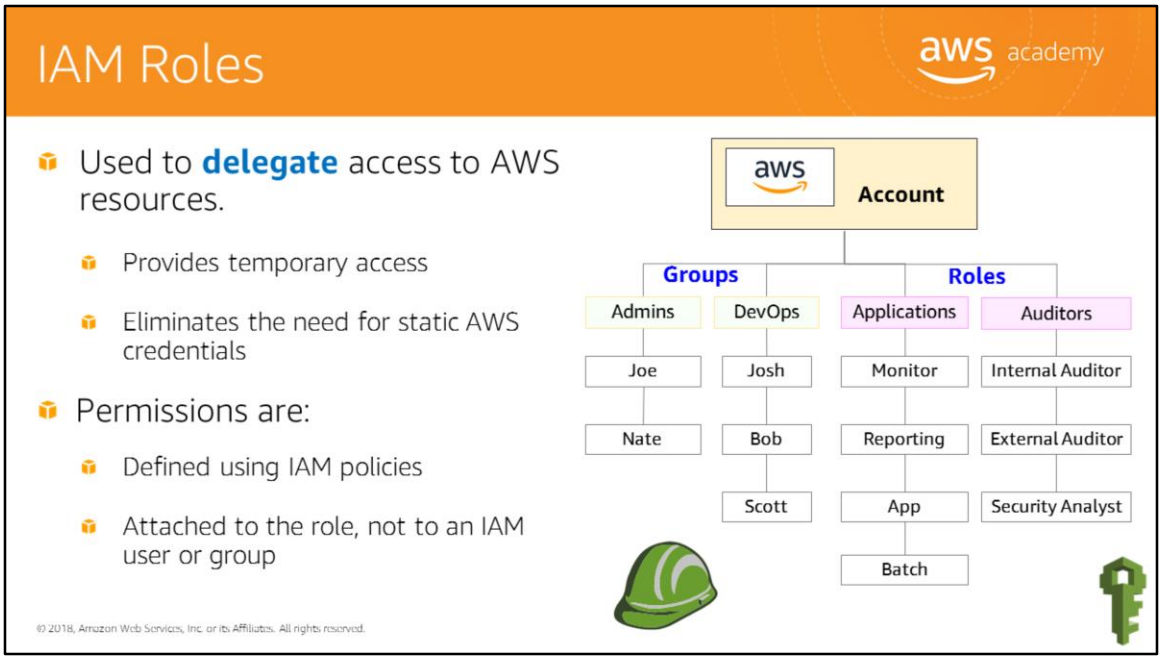
An IAM user is really just an identity with associated permissions. You might create an IAM user to represent an application that must have credentials in order to make requests to AWS. An application might have its own identity in your account and its own set of permissions, the same way that processes have their own identities and permissions in an operating system like Windows or Linux.



A *group* is a collection of IAM users. Groups let you specify permissions for a collection of users, which can make it easier to manage the permissions for those users. For example, you could have a group called *Developers* and give that group the types of permissions that developers typically need. Any user in that group automatically has the permissions that are assigned to the group. If a new user joins your organization and should have developer privileges, add that user to the Developers group. That automatically gives them the appropriate permissions. Similarly, if a person changes jobs in your organization, instead of editing that user's permissions, you can remove him or her from the old group and add him or her to the new group.

Important characteristics of groups:

- A group can contain many users, and a user can belong to multiple groups.
- Groups can't be nested; they can contain only users, not other groups.
- There's no default group that automatically includes all users in the AWS account. If you want to have a group like that, you need to create it and assign each new user to it.

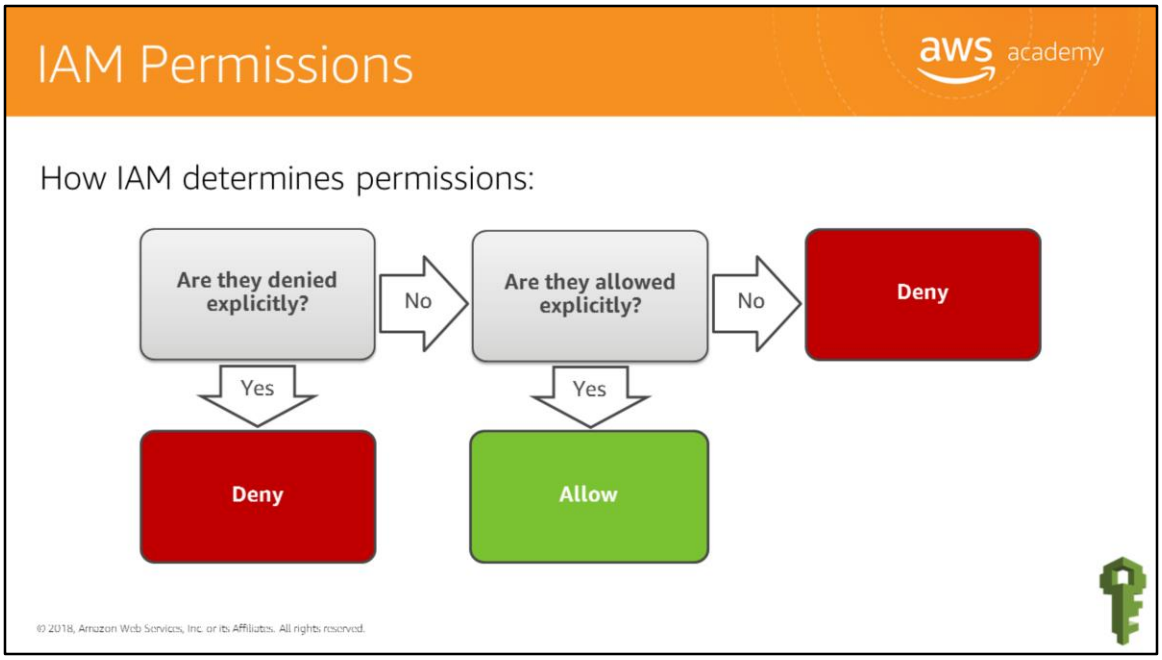


Federated users don't have permanent identities in your AWS account the way that IAM users do. To assign permissions to federated users, you can create an entity referred to as a *role.*

A role lets you define a set of permissions to access the resources that a user or service needs, but the permissions are not attached to an IAM user or group. Instead, at run time, applications or AWS services can programmatically assume a role. When a role is assumed, AWS returns temporary security credentials that the user or application can use to make programmatic requests to AWS. Consequently, you don’t have to share long-term security credentials (for example, by creating an IAM user) for each entity that requires access to a resource.

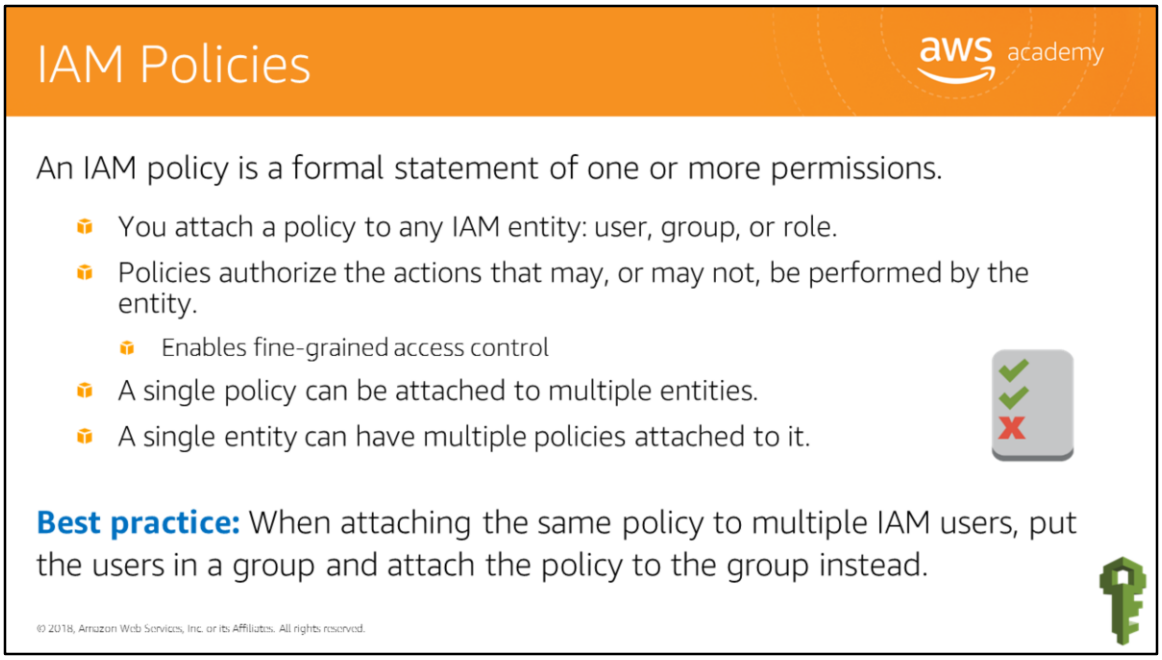
You create a role in the AWS account that contains the resources that you want to allow access to. When you create the role, you specify two policies:

- The **trust** policy specifies who is allowed to assume the role (the trusted entity, or principal).
- The **access** (or **permissions**) policy defines what actions and resources the principal is allowed access to. The principal can be an AWS account, an AWS service such as Amazon EC2, a SAML provider, or an identity provider (IdP) such as Login with Amazon, Facebook, or Google. The principal can also be an IAM user, group, or role from other AWS accounts, including the ones not owned by you.



Policies give you the opportunity to fine-tune privileges granted to IAM users, groups, and roles. Because policies are stored in JSON format, they can be used in conjunction with a version control system. It's a good idea to define least-privilege access to each user, group, or role. That way, you can customize access to specific resources using an authorization policy.

When determining whether permission is allowed, IAM first checks for an explicit denial policy. If one does not exist, it then checks for an explicit allow policy. If neither an explicit deny or explicit allow policy exists, IAM reverts to the default: implicit deny.



Policies can be attached to any IAM entity—user, group, role, or resource. For example, you can attach a policy to your AWS resources to block all requests that don't come from an approved IP address range. Policies specify what actions are allowed, which resources to allow the actions on, and what the effect will be when the user requests access to the resources.

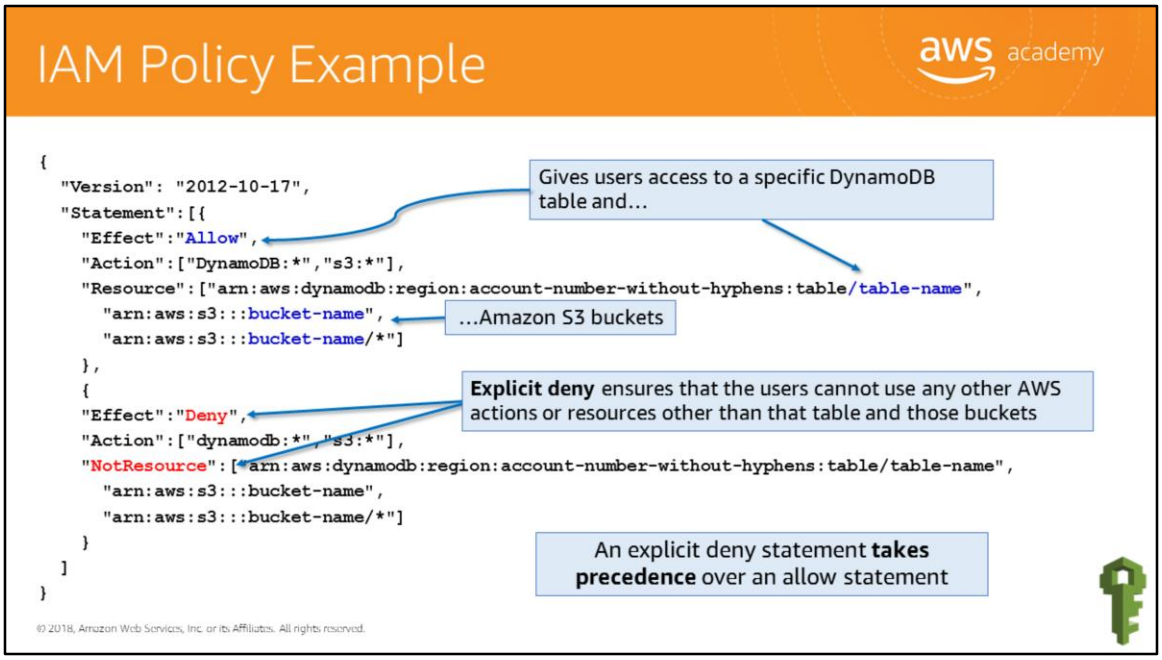
The order in which the policies are evaluated has no effect on the outcome of the evaluation. All policies are evaluated, and the result is always that the request is either allowed or denied. When there is a conflict, the most restrictive policy wins.

There are two types of IAM policies:
Identity-based policies are permission policies that you can attach to a principal (or identity), such as an IAM user, role or group. These policies control what actions that identity can perform, on which resources, and under what conditions. Identity-based policies can be further categorized:

- **Managed policies:** Standalone identity-based policies that your can attached to multiple users, groups and roles in your AWS account.
- **Inline policies:** Policies that you create and manage and that are embedded directly into a single user group or role

Resource-based policies are JSON policy documents that you attach to a resource such as an Amazon S3 bucket. These policies control what actions a specified principal can perform on that resource and what conditions. Resource-based policies are inline policies. There are no managed resource-based policies. For more information see https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html.

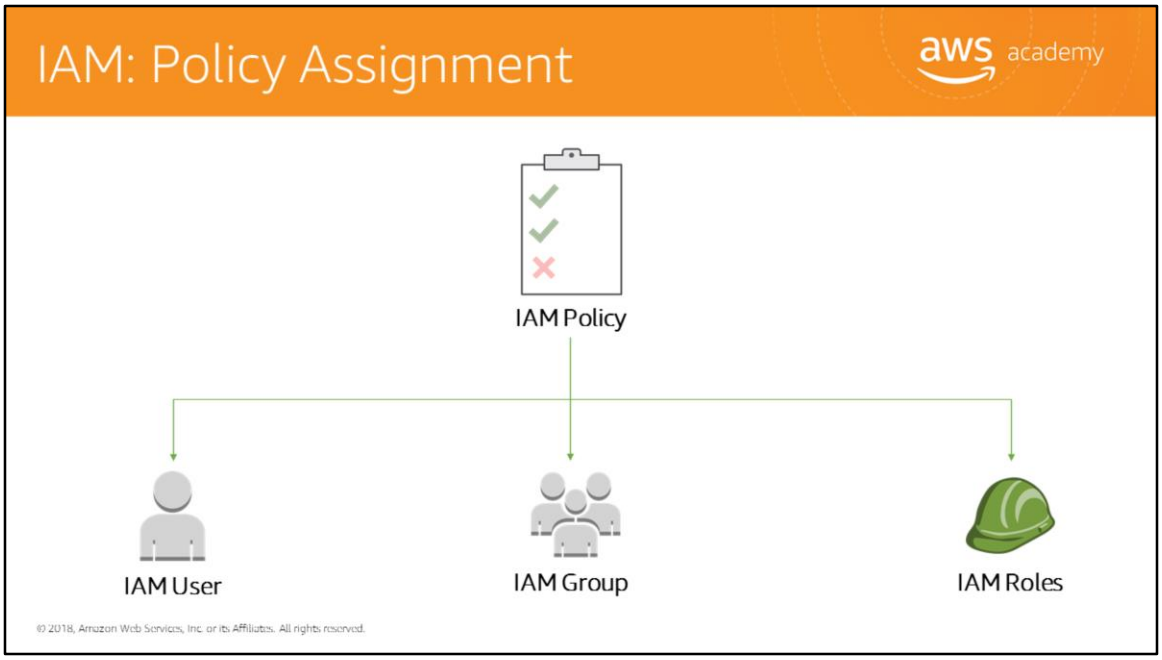
Additionally, you can use the IAM policy simulator to test and troubleshoot IAM and resource-based policies in the following ways. To learn more about the IAM policy simulator see https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_testing-policies.html



The policy gives users access to only the following:

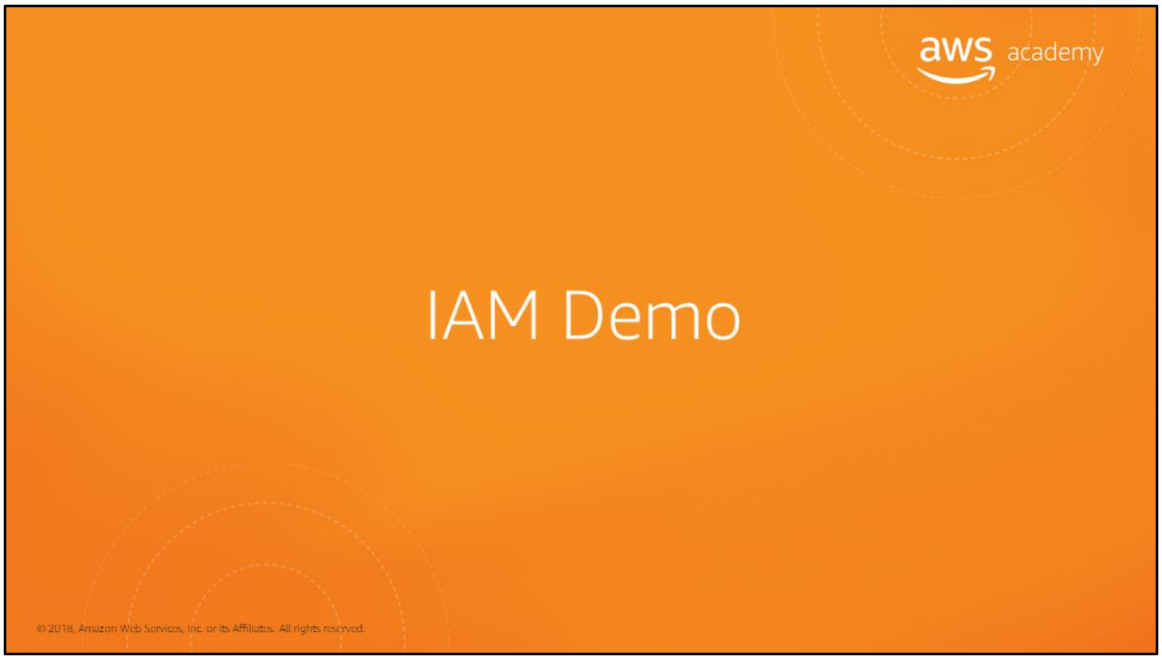
- The DynamoDB table whose name is represented by table-name.
- The AWS account's corporate Amazon S3 bucket, whose name is represented by bucket-name, and all the objects it contains.

The policy includes an explicit deny ("Effect":"Deny") element. In conjunction with the **NotResource** element, this helps to ensure that the users can't use any AWS actions or resources except those specified in the policy, even if permissions have been granted in another policy. (An explicit deny statement takes precedence over an allow statement.)



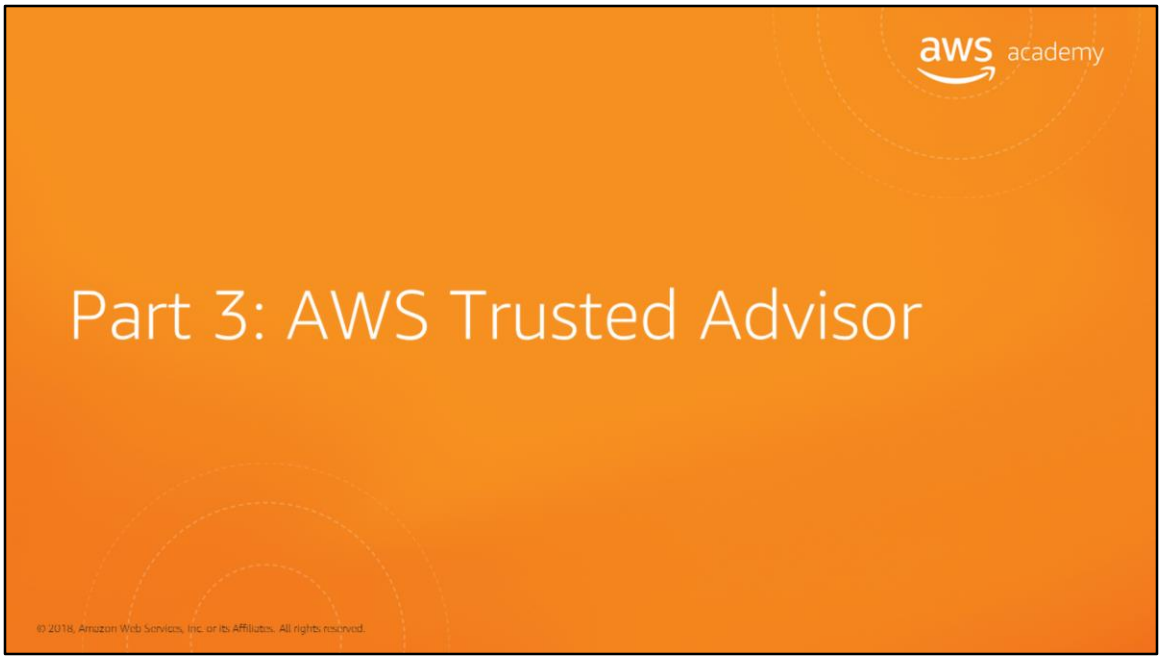
It's important to note that one policy can be assigned to an IAM user, IAM group, and IAM roles.

Now that we've covered the basic concepts of IAM, let's log in to the console, create a user, assign the user to a group, and apply permissions.



Please review the IAM console demonstration: **M3_iam v2.0.mp4**.

This video demonstration can be found in the learning management system.



Now let's look at some additional service that can be used to improve overall security and compliance.

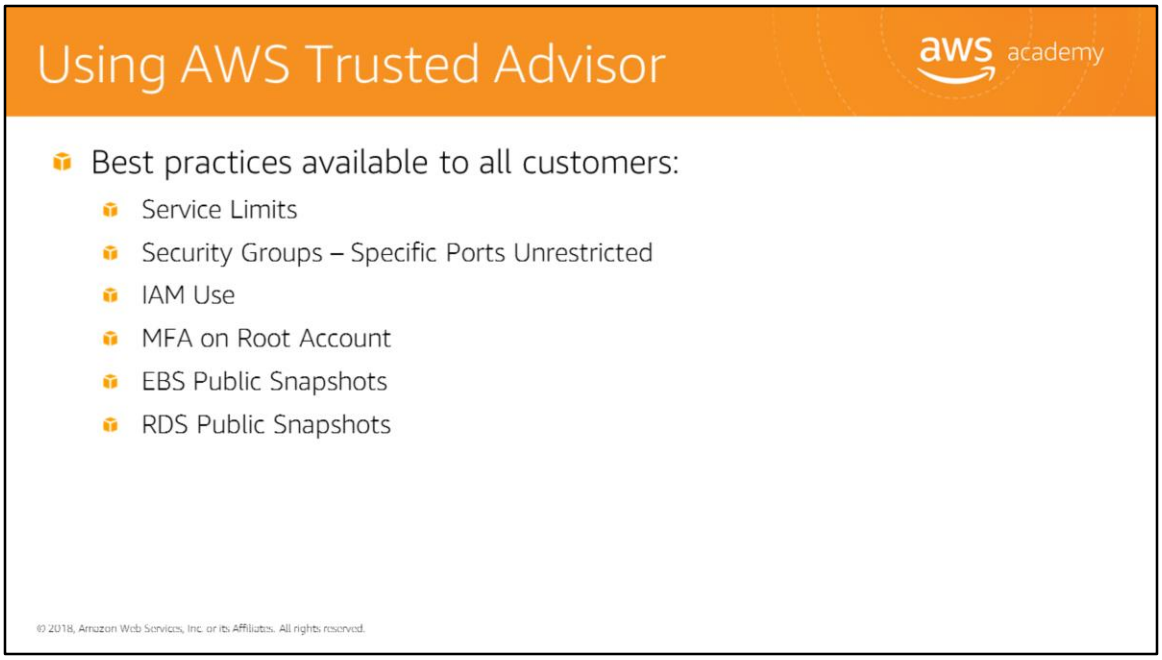
AWS Trusted Advisor is like your customized cloud expert. It provides four of the most popular performance and security recommendations to all AWS customers. Let's look at details and a case study to understand this service.

AWS Trusted Advisor is a an online resource to help you reduce cost, increase performance, and improve security by optimizing your AWS environment. It provides best practices (or checks) in five categories:

- **Cost Optimization**: See how you can save money on AWS by eliminating unused and idle resources or making commitments to reserved capacity.
- **Performance**: Improve the performance of your service by checking your service limits, ensuring you take advantage of provisioned throughput, and monitoring for over-utilized instances.
- **Security**: Improve the security of your application by closing gaps, enabling various AWS security features, and examining your permissions.
- **Fault Tolerance**: Increase the availability and redundancy of your AWS application by take advantage of automatic scaling, health checks, multiple Availability Zones, and backup capabilities.
- **Service Limits**: Checks for service usage that is more than 80% of the service limit.

The status of the check is shown by using color coding on the dashboard page:
Red: action recommended
Yellow: investigation recommended
Green: no problem detected

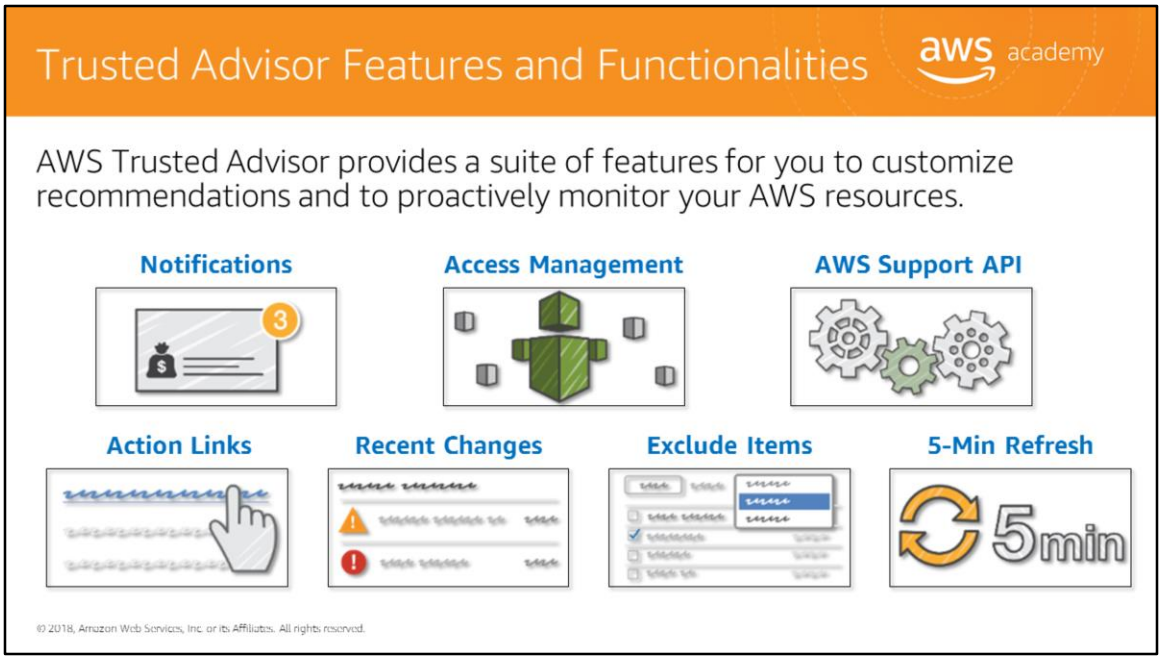
You can visit the Trusted Advisor Console here
https://console.aws.amazon.com/trustedadvisor/



AWS Trusted Advisor provides popular performance and security recommendations to all AWS customers. These six Trusted Advisor checks are available to all customers at no cost: Service Limits and Security Groups - Specific Ports Unrestricted, IAM Use, MFA on Root Account, EBS Public Snapshots, and RDS Public Snapshots.

The complete set of checks and guidance is available with Business and Enterprise Support plans. AWS Trusted Advisor helps you to provision your resources following best practices to improve system performance and reliability, increase security, and look for opportunities to save money.

For more information about Trusted Advisor best practices (checks) see https://aws.amazon.com/premiumsupport/trustedadvisor/best-practices/.



Trusted Advisor Notifications helps you stay up-to-date with your AWS resource deployment. You will be notified by weekly email when you opt in for this service, and it is free.

You can use **IAM** to control access to specific checks or check categories.

You can retrieve and refresh Trusted Advisor results programmatically using the **AWS Support API**.

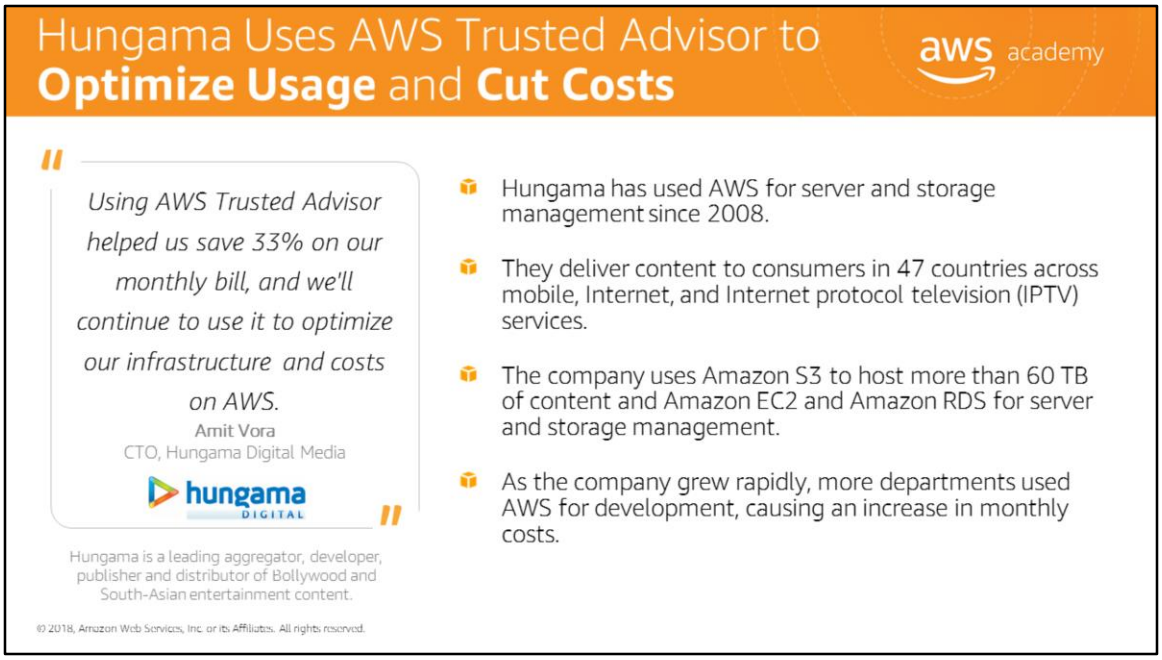
Action Links are hyperlinks on items within a Trusted Advisor report that take you directly to the console, where you can take action on the Trusted Advisor recommendations.

With the **Recent Changes** feature, you can track recent changes of check status on the console dashboard. The most recent changes appear at the top of the list to bring them to your attention.

The **Exclude Items** feature allows you to customize the Trusted Advisor report. You can exclude items from the check result if they are not relevant.

You can refresh individual checks or refresh all the checks at once by clicking the Refresh All button in the summary dashboard. A check is eligible for **refresh five minutes** after it was last refreshed.

For more information about Trusted Advisor see https://aws.amazon.com/premiumsupport/trustedadvisor/.



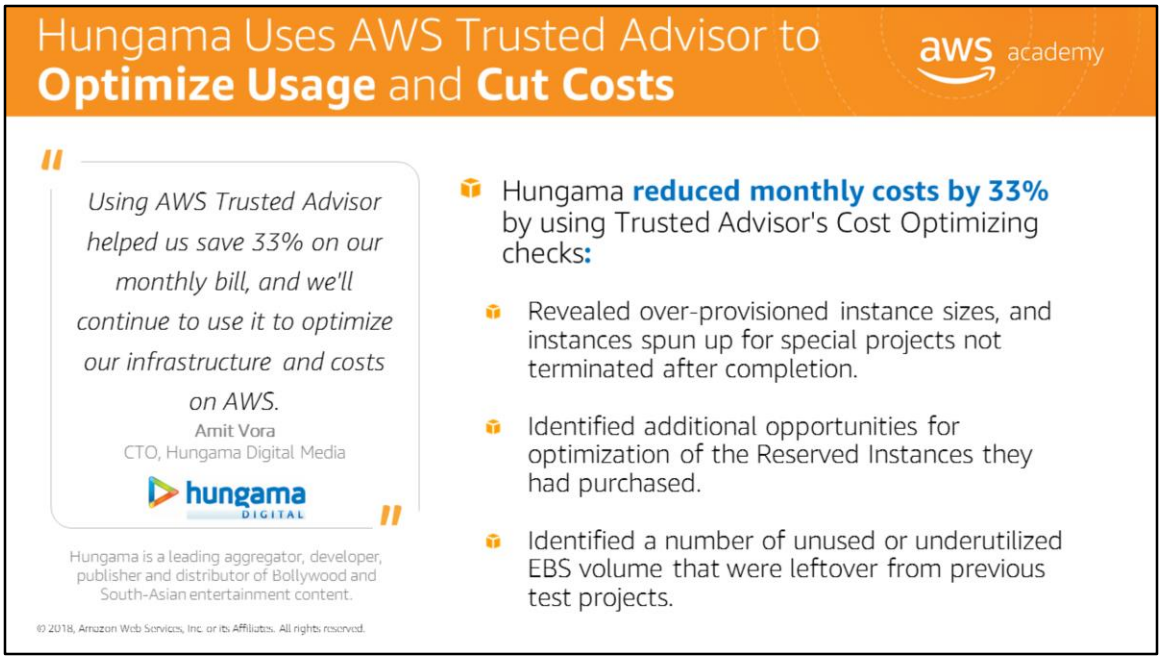
Three AWS Trusted Advisor checks were particularly helpful in optimizing usage and cutting costs:

1. The *Low Utilization Amazon EC2 Instances* check on AWS Trusted Advisor checks the Amazon Elastic Compute Cloud (Amazon EC2) instances that are running at any time during the last 14 days and alerts you if the daily CPU utilization was 10% or less and network I/O was 5 MB or less on 4 or more days.

2. The *Reserved Instance Optimization* check with AWS Trusted Advisor checks your Amazon EC2 computing consumption history and calculates an optimal number of Partial Upfront Reserved Instances. Recommendations are based on the previous calendar month's hour-by-hour usage aggregated across all consolidated billing accounts.

3. The *Underutilized Amazon EBS Volumes* check on AWS Trusted Advisor checks Amazon EBS volume configurations and warns when volumes appear to be underused. If a volume remains unattached or has very low write activity (excluding boot volumes) for a period of time, the volume is probably not being used.

For more on how Hungama uses AWS see https://aws.amazon.com/solutions/case-studies/hungama/.



Low Utilization Amazon EC2 Instances

The **Low Utilization Amazon EC2 Instances** check revealed over-provisioned instance sizes, and instances spun up for special projects were not terminated after completion. In response, the audit team used this information to **right-size** their instances. They also **categorized** production and development servers and **automated** the process of shutting down development servers during non-business hours.

Reserved Instance Optimization

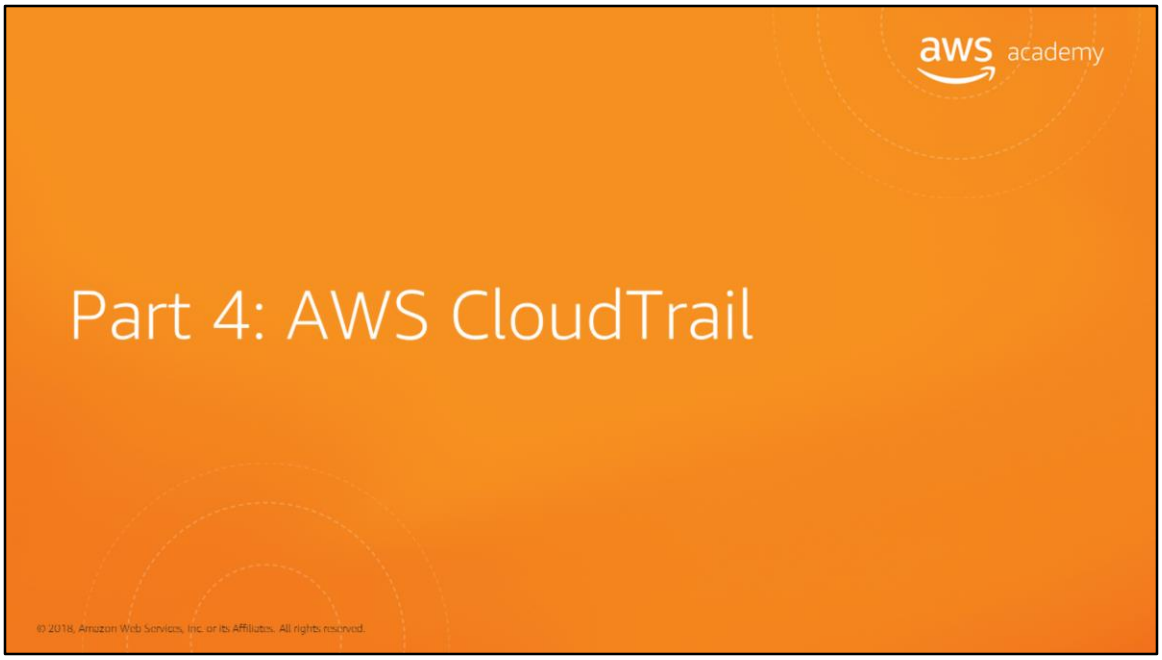
The **Reserved Instance Optimization** check identified additional opportunities for optimization of the RI instances they had purchased. In response, Hungama changed how they reserved their instances and based reservations on the **specific usage patterns** of their different instance categories (dev/prod/test/etc.).

Underutilized Amazon EBS

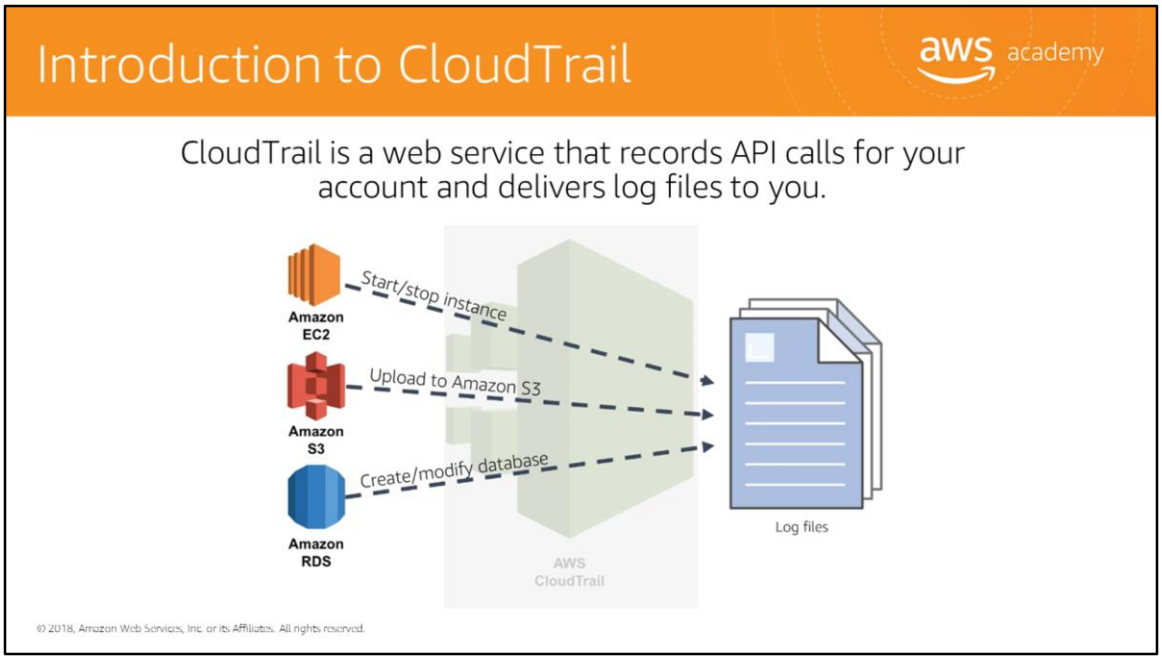
The **Underutilized Amazon EBS** volumes check identified a number of unused or underutilized EBS volumes that were often leftover from previous test projects. In response, the audit team created **snapshots** of many of the underutilized EBS volumes, which they stored on Amazon S3, and then **deleted the volumes**. This resulted in a reduction of over 90% on the number of snapshots generated weekly.

AWS Trusted Advisor

AWS Trusted Advisor is an online tool that acts like a customized cloud expert, helping you to configure your resources to follow best practices. Trusted Advisor inspects your AWS environment to help close security gaps, and finds opportunities to save money, improve system performance, and increase reliability.

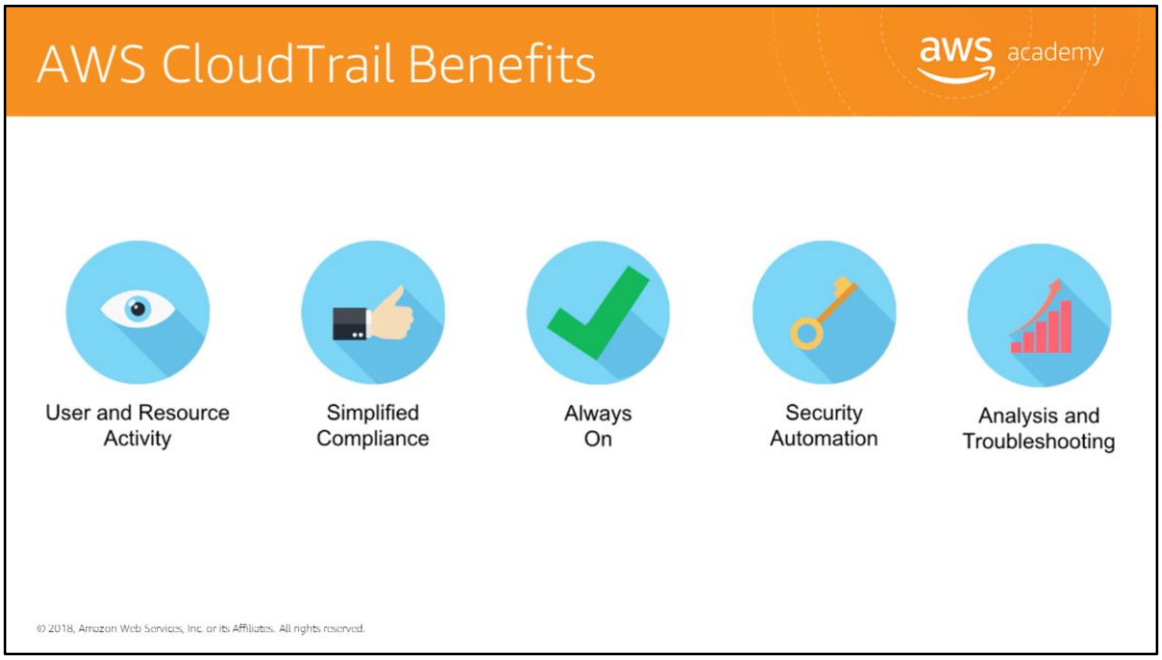


AWS CloudTrail enables you to simplify governance, compliance, and risk auditing. CloudTrail accelerates analysis of operational and security issues by providing visibility into both API and non-API actions in your AWS account.

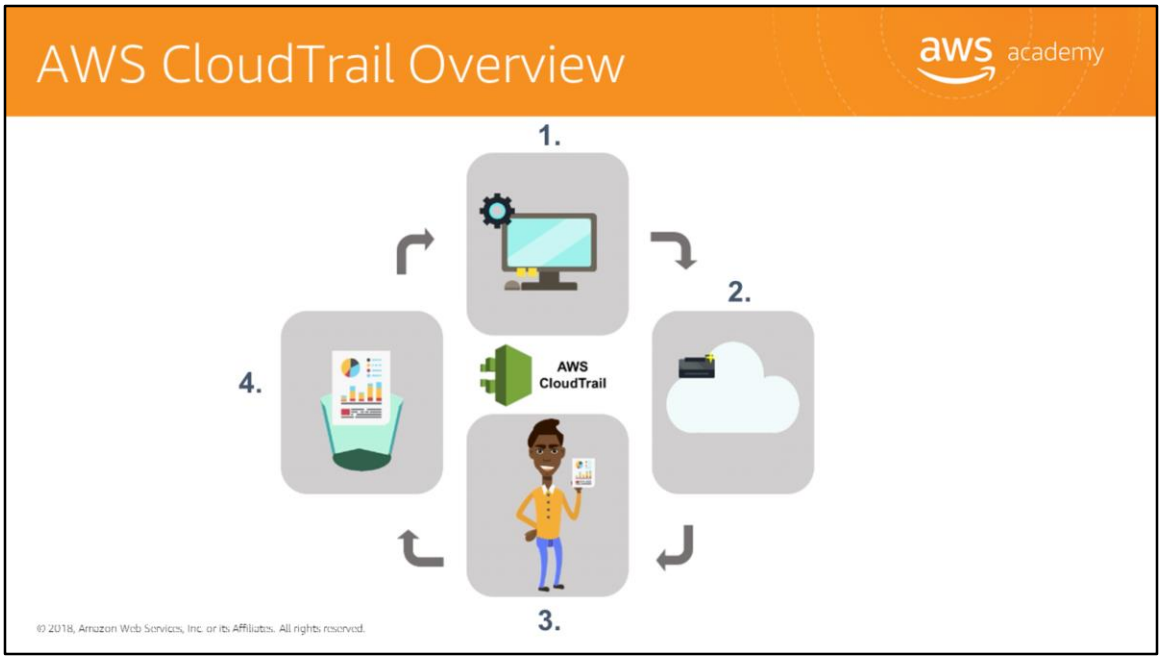


CloudTrail helps you log the API calls made in your AWS account across regions. Because everything in AWS is an API call, activity to AWS resources like starting and stopping instances, creating or modifying Amazon RDS databases, or uploading a file to Amazon S3 are logged, whether that action was performed via the CLI, an SDK, the console, or an API directly. This is a crucial tool for simplifying your governance, compliance, and risk auditing.

CloudTrail enables you to simplify governance, compliance, and risk auditing. The service accelerates analysis of operational and security issues by providing visibility into both API and non-API actions in your AWS account. With CloudWatch Logs integration, support for multi-region configurations, and log file integrity validation, CloudTrail provides comprehensive, secure, and searchable event history of activity made with the console, AWS SDKs, command line tools, and other AWS services.

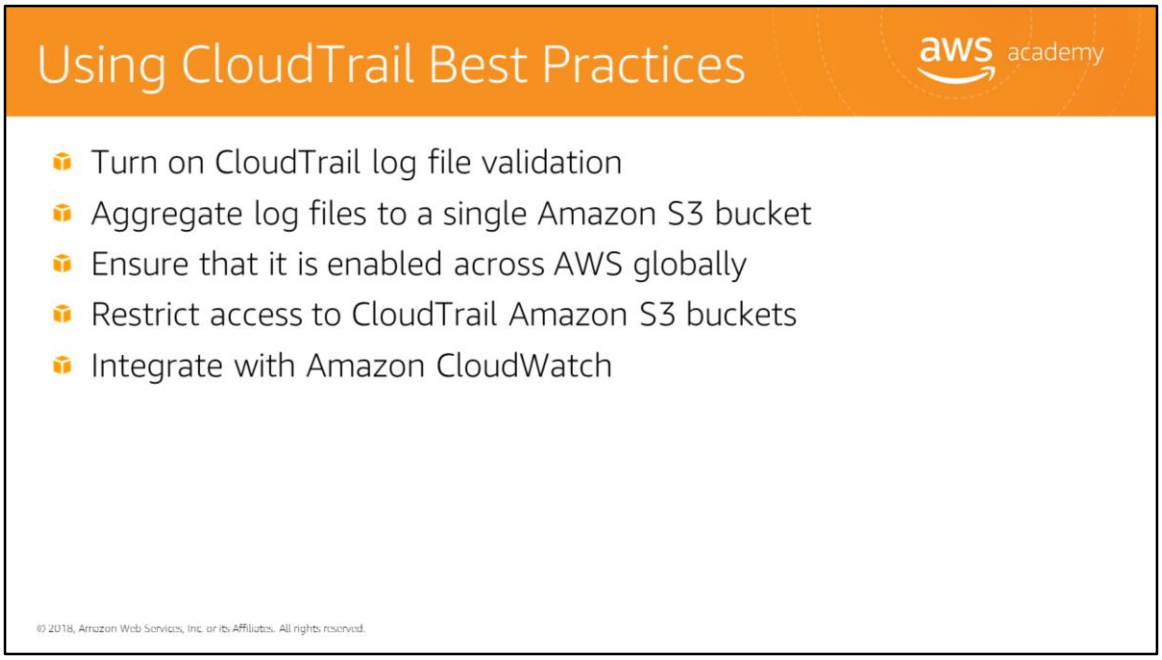


CloudTrail has several key benefits. It increases your visibility into user and resource activity, which allows you to identify who did what and when in your account. Compliance audits are simplified because they are automatically recording and storing event logs. This allows you to search through log data, identify actions that are out of compliance, accelerate investigations into incidents and then expedite response. Because you are able to capture a comprehensive history of changes made within your account, you can discover and troubleshoot any operational issues in your account.

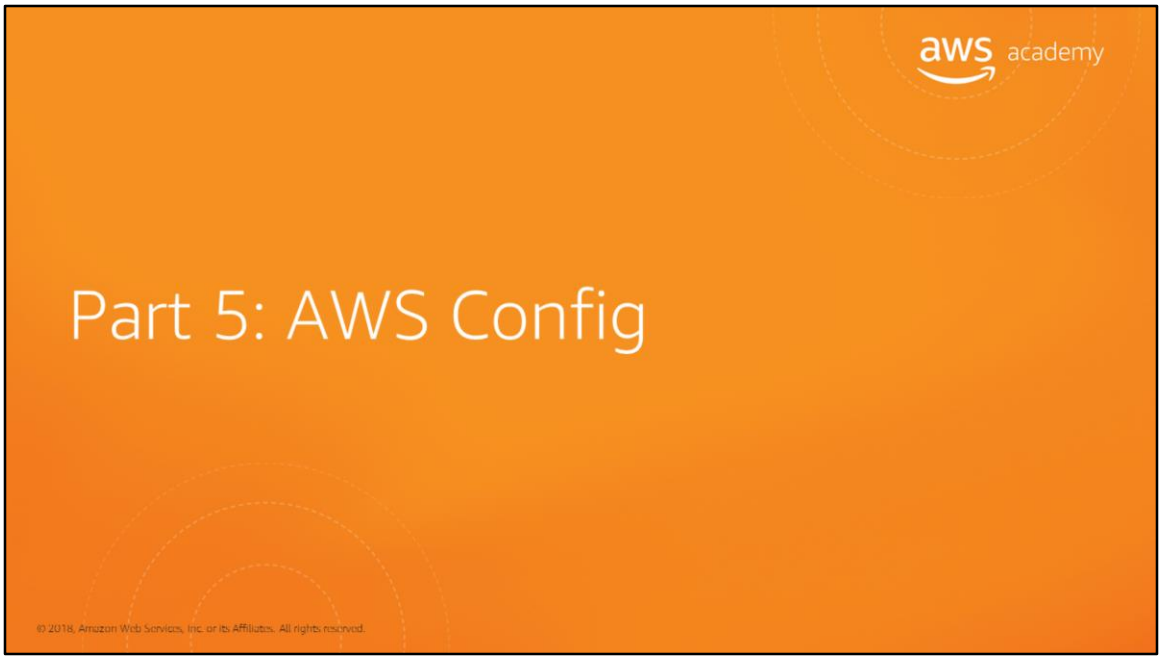


How does this work?

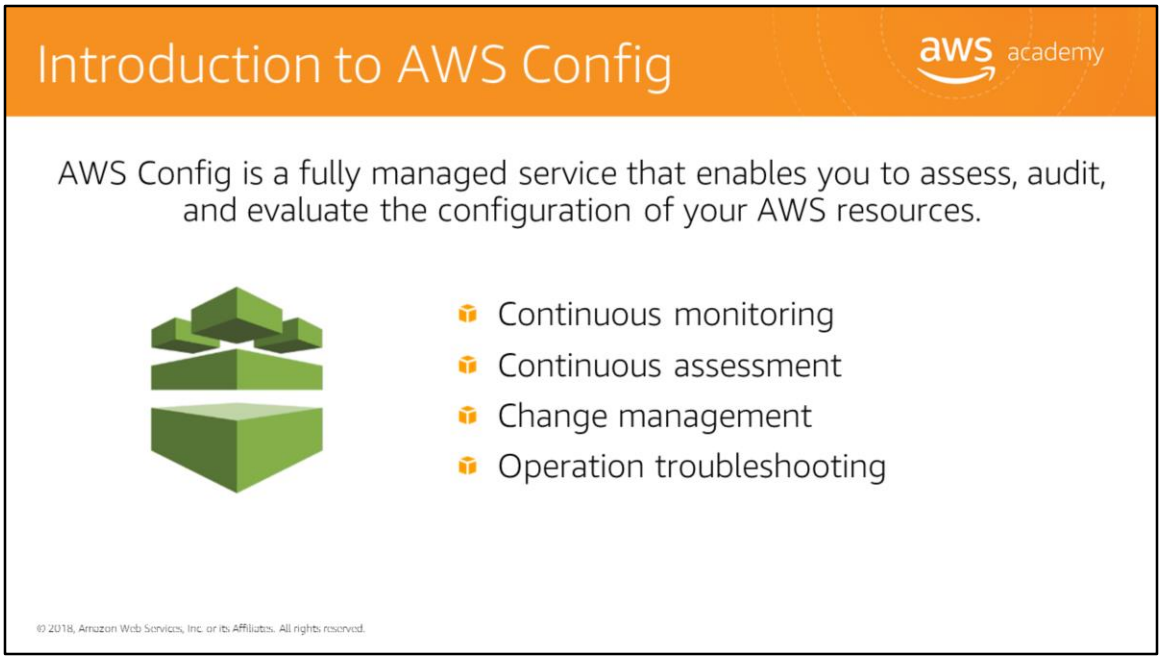
First, an activity happens in your account. Next, CloudTrail captures and records that activity and calls it a *CloudTrail event*. The event will contain details about who performed the request, the date and time of the request, the source IP and how the request was made, the action being performed, the region in which the action was taken, and the response. By default, the logs are stored for 7 days. The activity log can be sent to other AWS services, so the activity history can be retained for as long as you like.



To get the most out of CloudTrail, turn on CloudTrail log file validations. When you are configuring CloudTrail, you can aggregate all log files to a single Amazon S3 bucket. Additionally, a configuration that applies to all regions ensures that your settings are applied consistently across all existing and newly launched regions. You can also validate the integrity of log files by detecting whether or not they were changed or deleted after they were sent to the S3 bucket. It is also a good idea to run MFA to delete a CloudTrail bucket. This can be accomplished by restricting access to where they are stored. Lastly, integrating this service with Amazon CloudWatch enables you to define actions to execute when specific events are logged by CloudTrail.

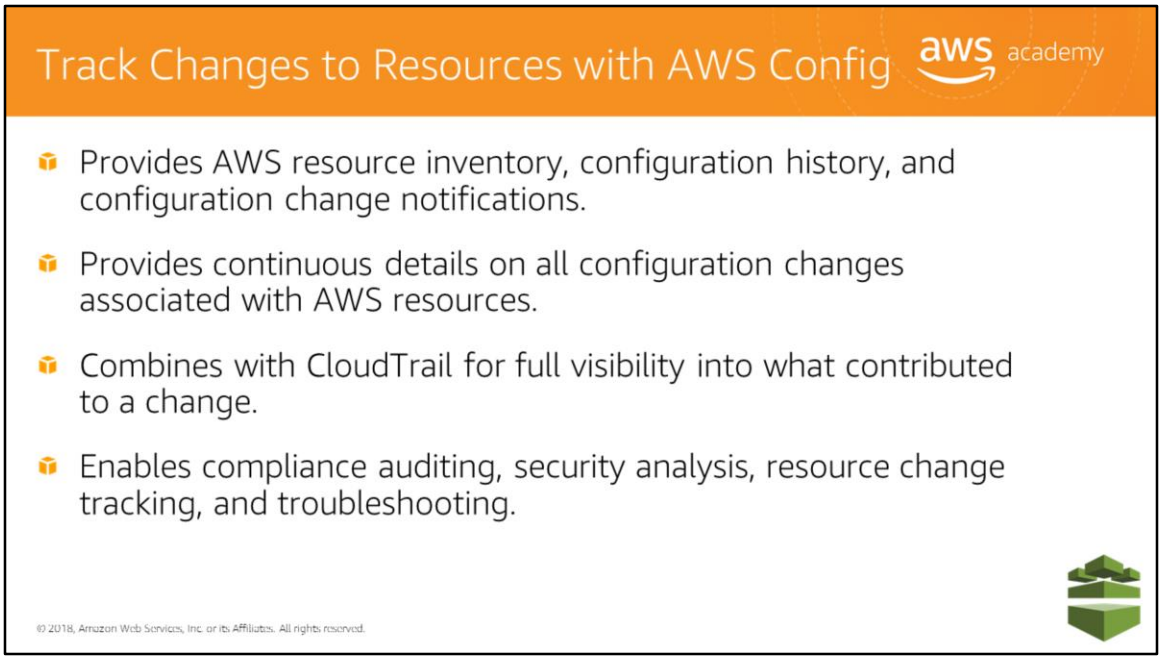


Let’s look at AWS Config to see how it can help with security.



AWS Config is a fully managed service that provides you with an AWS resource inventory, configuration history, and configuration change notifications to enable security and governance. With AWS Config, you can discover existing AWS resources, export a complete inventory of your AWS resources with all configuration details, and determine how a resource was configured at any point in time. These capabilities enable compliance auditing, security analysis, resource change tracking, and troubleshooting.

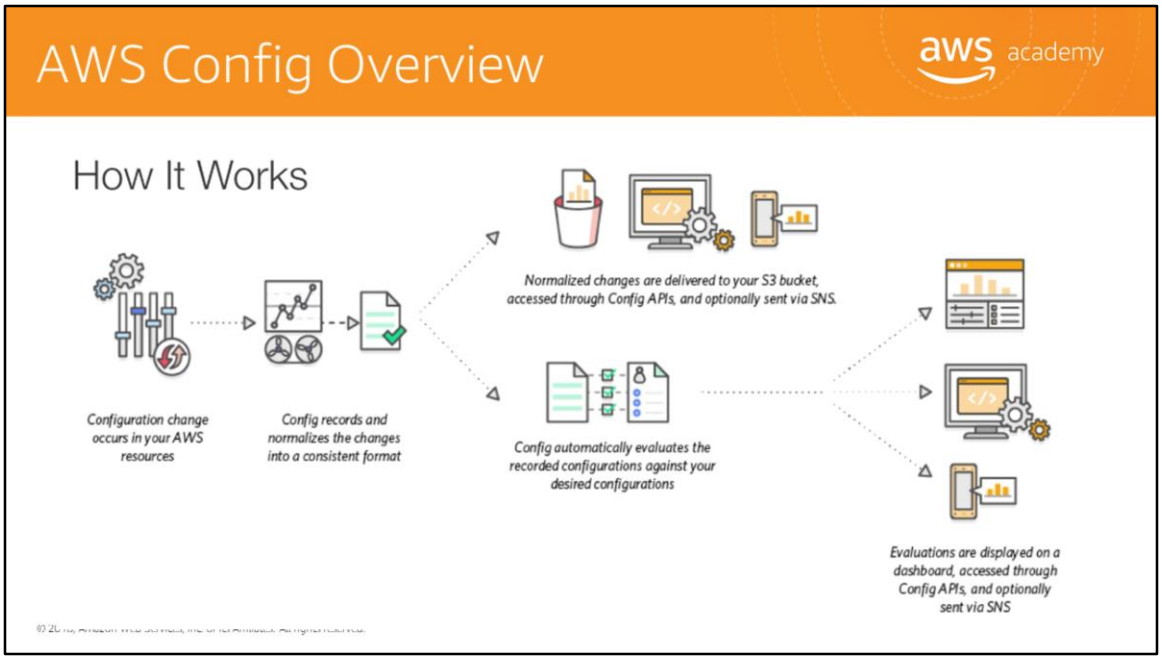
With AWS Config, you can review changes in configurations and relationships between AWS resources. You can also review detailed resource configuration histories and determine overall compliance against those configurations specified in your internal compliance. This enables you to simplify compliance auditing, security analysis, change management, as well as operational troubleshooting.



AWS Config makes it easy to track your resource's configuration without the need for upfront investments, and you can avoid the complexity of installing and updating agents for data collection or maintaining large databases. After you enable AWS Config, you can view continuously updated details of all configuration attributes associated with AWS resources. You are notified via Amazon Simple Notification Service (Amazon SNS) of every configuration change.

AWS Config gives you access to resource configuration history. You can relate configuration changes with CloudTrail events that may have contributed to the change in configuration. This information provides you full visibility from details—such as "Who made the change?" and "From what IP address?"—to the effect of this change on AWS resources and related resources. You can use this information to generate reports to aid auditing and assessing compliance over a period of time.

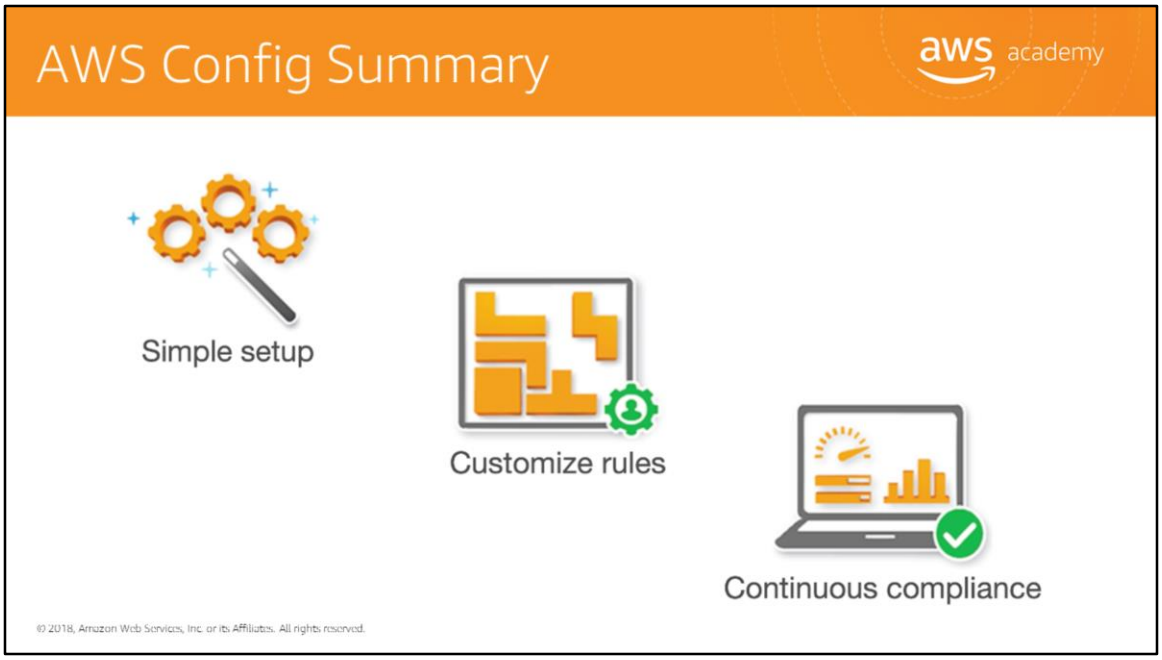
If you want to track changes to resources configuration, answer questions about resource configurations, demonstrate compliance, troubleshoot, or perform security analysis, use AWS Config.



So, how does this work?

Looking at this diagram from left to right, the first thing that happens is a change occurs in one of your AWS resources. Next, the AWS Config engine records and normalizes that change in a consistent format. Then those changes are delivered to an S3 bucket, they are assessed through the AWS Config APIs and, optionally, they can be sent out via a notification service like Amazon SMS.

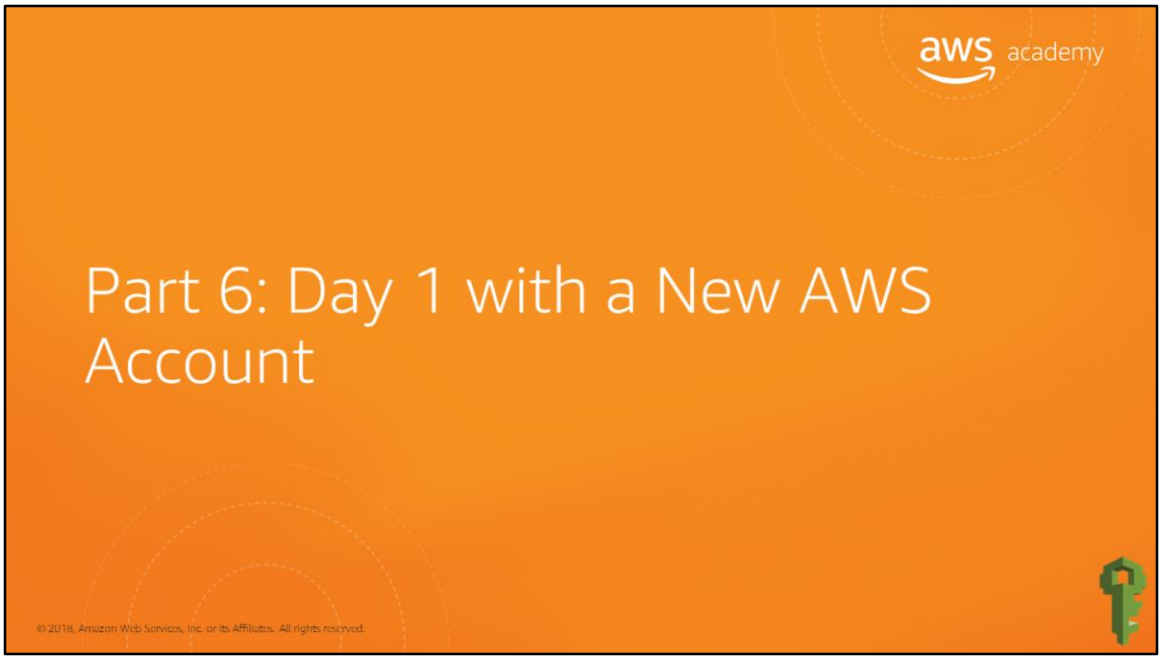
AWS Config will automatically evaluate the recorded configuration against your desired configuration. Those evaluations will be displayed on the dashboard or they are accessible via the AWS Config APIs. They can also be sent out via Amazon SMS.



In summary, AWS Config is a simple service to set up, but it is a very powerful tool that allows you to take advantage of custom rules to automatically discover your AWS resources.

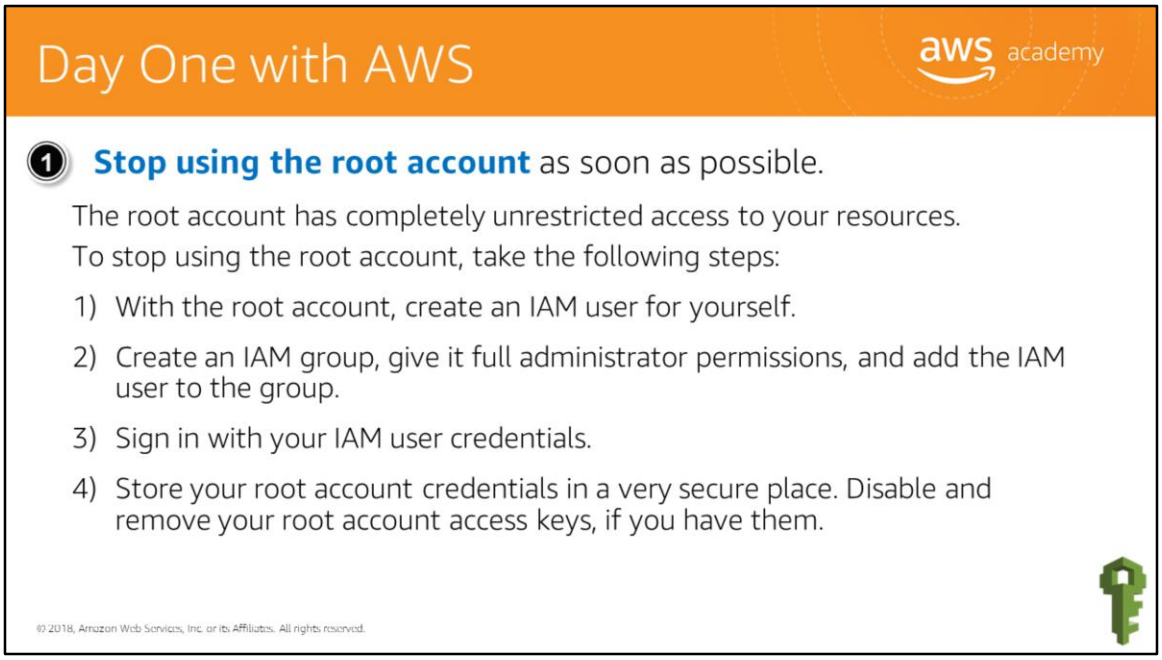
With AWS Config, you can do the following:

- Evaluate your AWS resource configurations for desired settings.
- Get a snapshot of the current configurations of the supported resources that are associated with your AWS account.
- Retrieve configurations of one or more resources that exist in your account.
- Retrieve historical configurations of one or more resources.
- Receive a notification whenever a resource is created, modified, or deleted.
- View relationships between resources. For example, you might want to find all resources that use a particular security group.



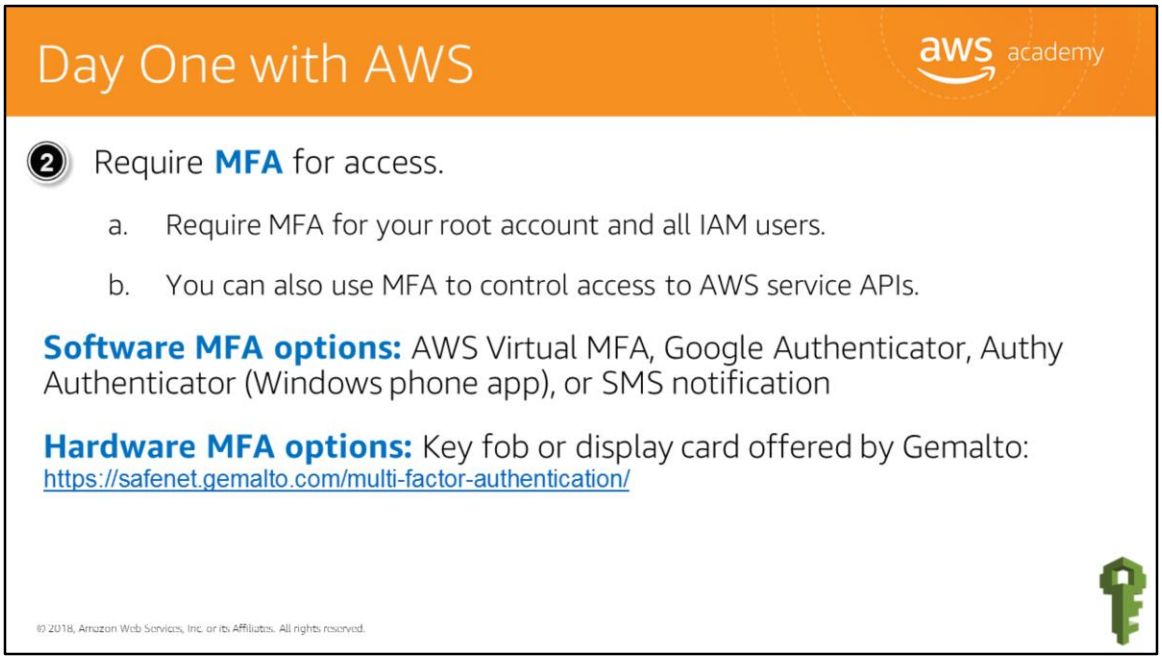
Let's take a look at how we put these services into action by starting from the beginning.

This is Day 1 with a new AWS account. What are the best practices for setting up the account?



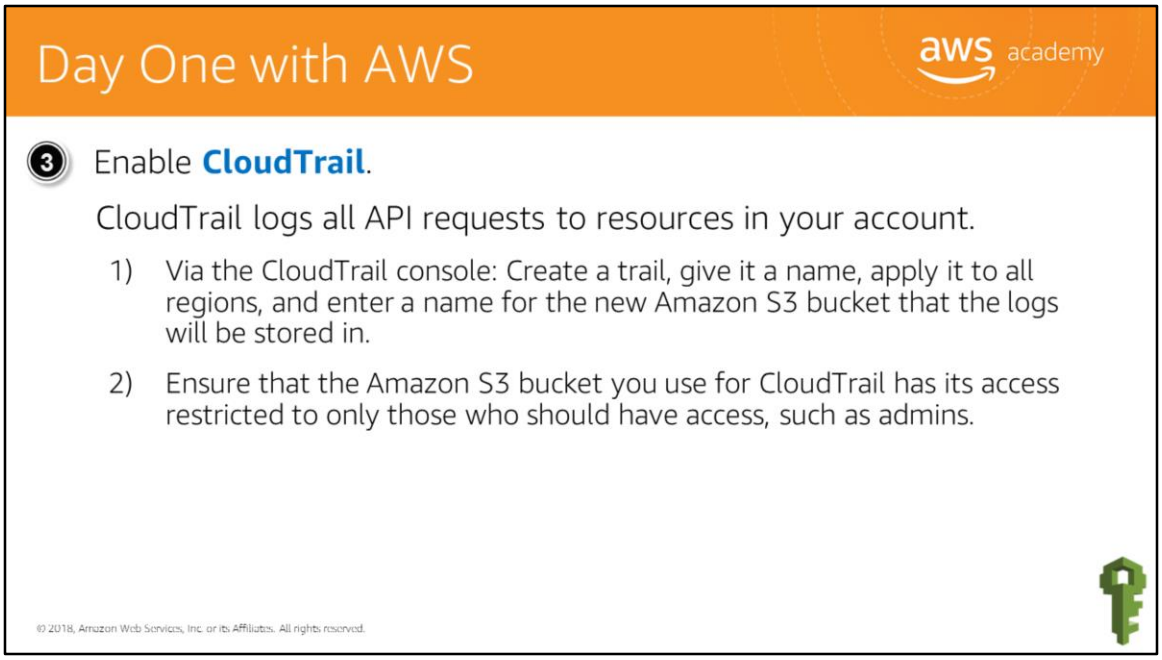
AWS also recommends if you have access keys for your root account, you remove them once you've established that they are not being used anywhere in your applications.

For instructions for setting up your first IAM user and administrators group see http://docs.aws.amazon.com/IAM/latest/UserGuide/getting-started_create-admin-group.html.



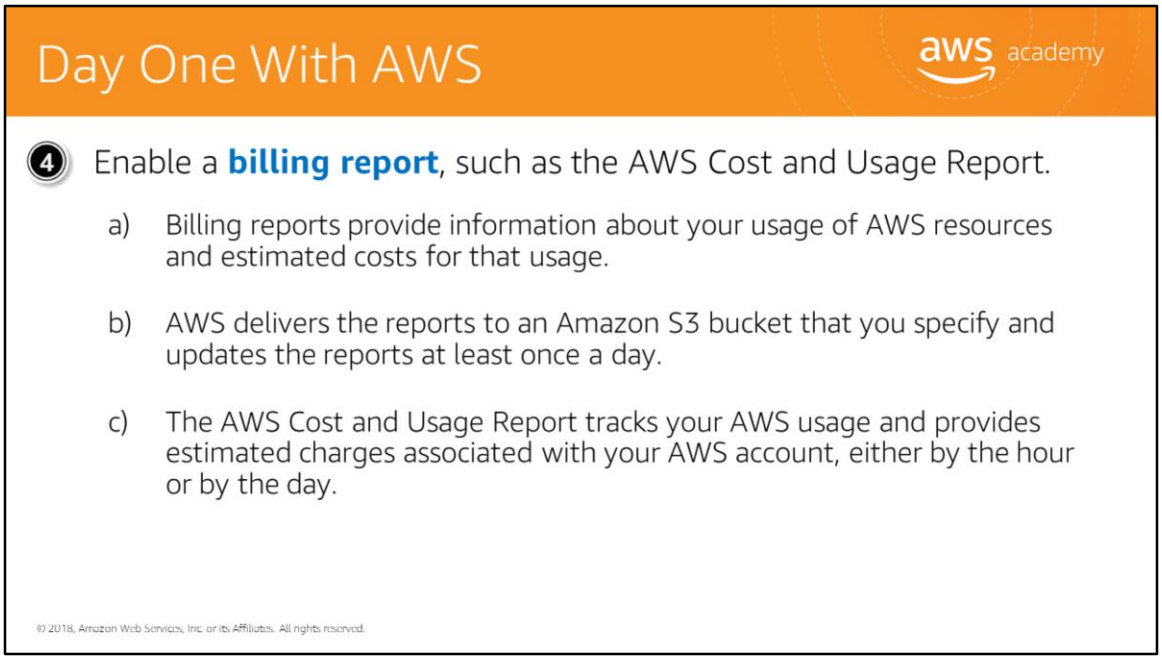
SMS-based MFA authentication is currently in Preview release only. To request access to it, see https://aws.amazon.com/iam/details/mfa/smsmfa/.

For more information, including links to the approved MFA devices and applications see https://aws.amazon.com/iam/details/mfa/.

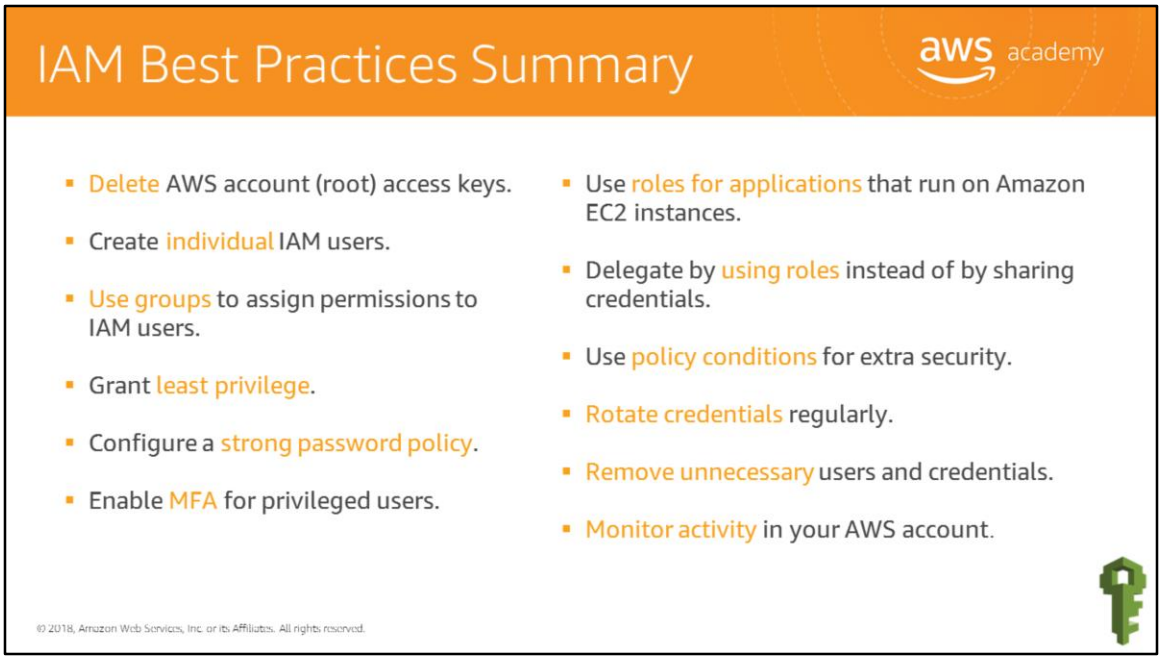


For step-by-step instructions for creating a trail in CloudTrail see http://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-create-a-trail-using-the-console-first-time.html.

CloudTrail is now enabled by default for ALL CUSTOMERS. It will provide visibility into the past seven days of account activity without the need for you to configure a trail in the service to get started. This new always-on capability provides the ability to view, search, and download the aforementioned account activity through the CloudTrail Event History.

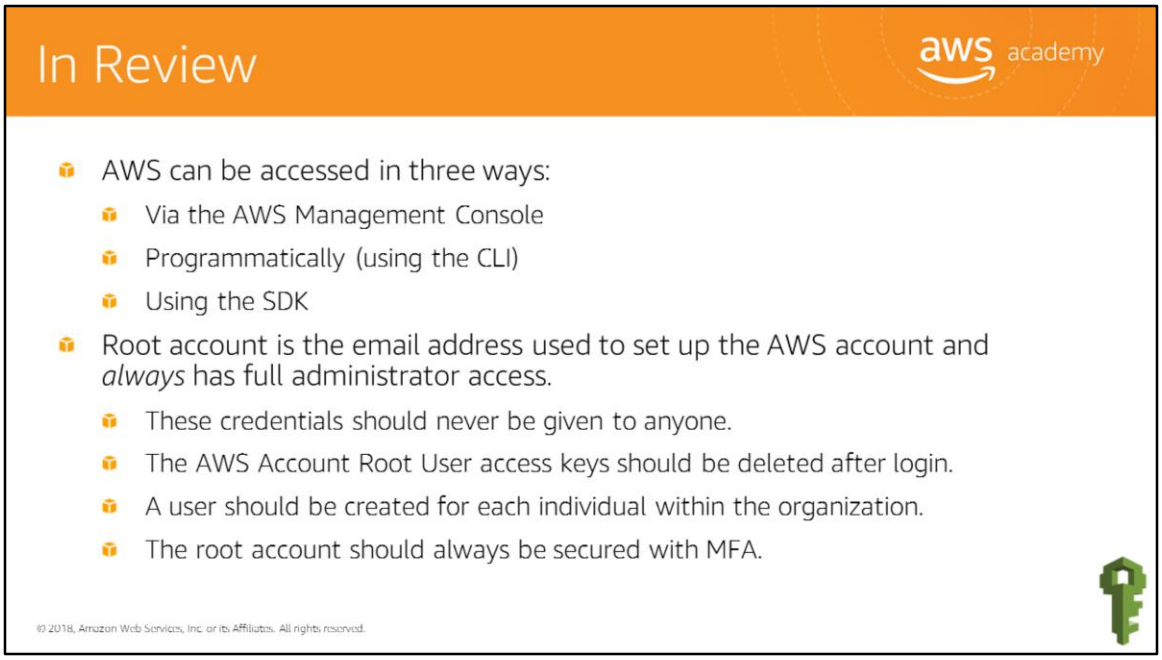


To create an AWS Cost and Usage report see http://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/detailed-billing-reports.html#turnonreports.

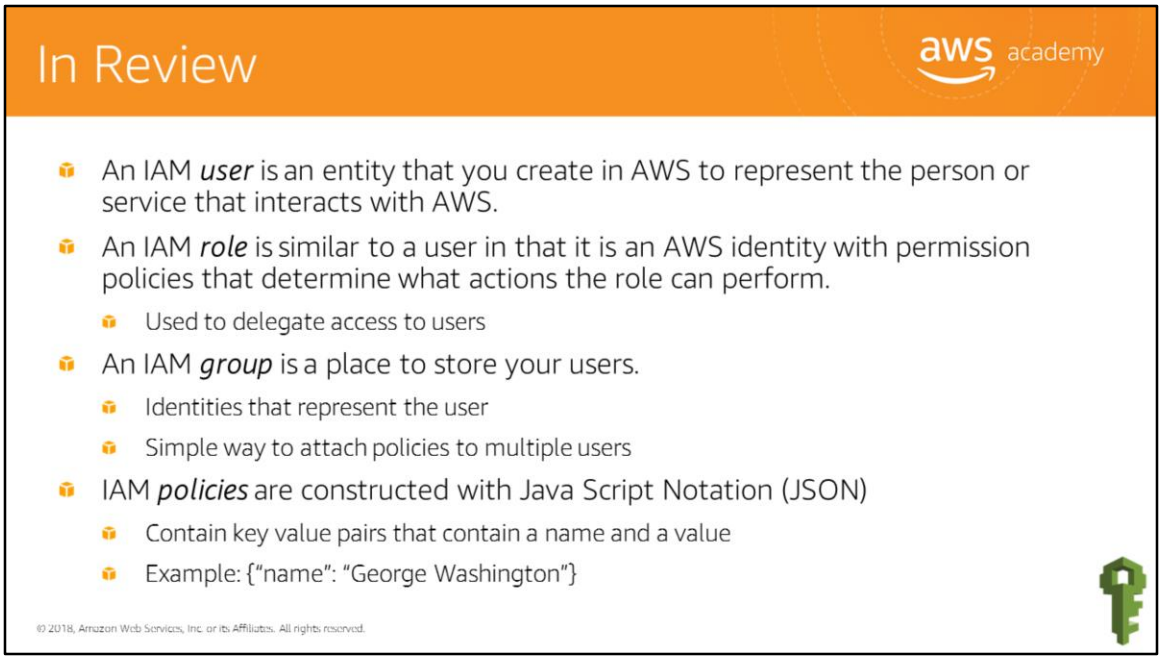


The slide shows some best practices to follow with IAM.

For more information see http://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html



For more information see http://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html



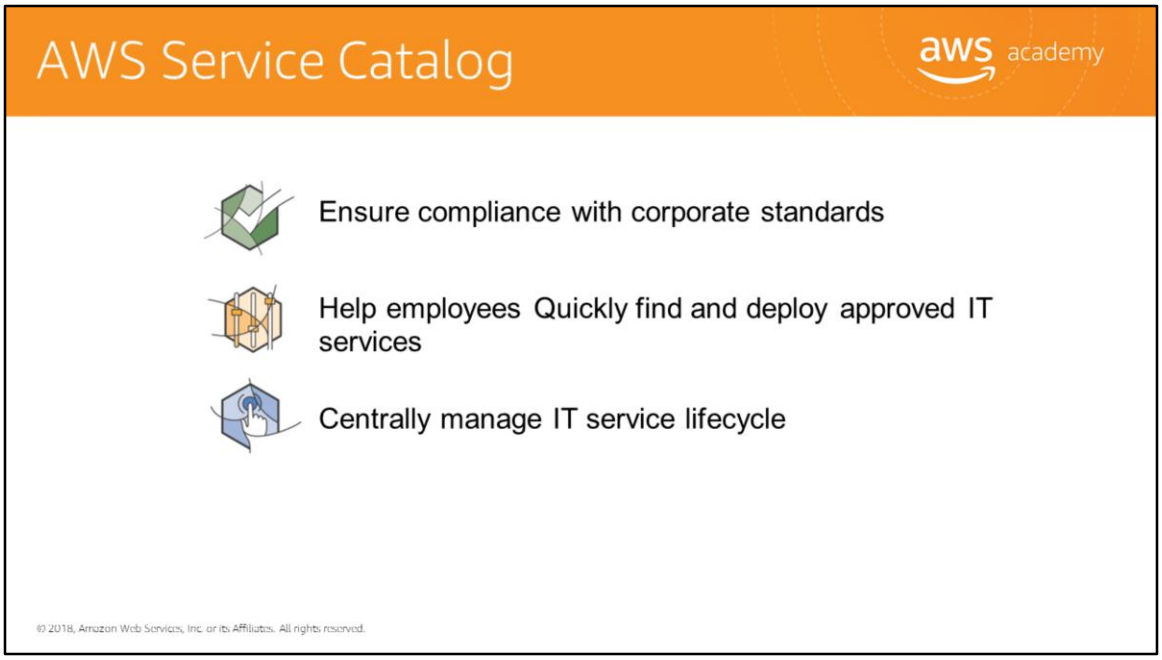
For more information see http://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html

Welcome to Part 3 of the AWS Security Module, AWS Security Compliance Program.

The success of our security and compliance program is primarily measured by our customers' success. Our customers drive our portfolio of compliance reports, attestations, and certifications that support their efforts in running a secure and compliant cloud environment.

You can take advantage of this effort to achieve the savings and security at scale that AWS offers while still maintaining robust security and regulatory compliance.
In this module, we'll be discussing:

- AWS' Compliance Approach, which includes Assurance Programs
- AWS Risk and Compliance Programs such as Risk Management, Control Environment, and Information Security
- AWS Customer Compliance responsibilities



Customers can also use AWS Service Catalog to create and manage catalogs of IT services that they have approved for use on AWS, including virtual machine images, servers, software, and databases to complete multi-tier application architectures.

AWS Service Catalog allows you to centrally manage commonly deployed IT services and helps you achieve consistent governance and meet compliance requirements while enabling users to quickly deploy only the approved IT services they need.

AWS Service Catalog an be integrated with AWS CloudFormation for stack developments to ensure compliance with corporate standards.

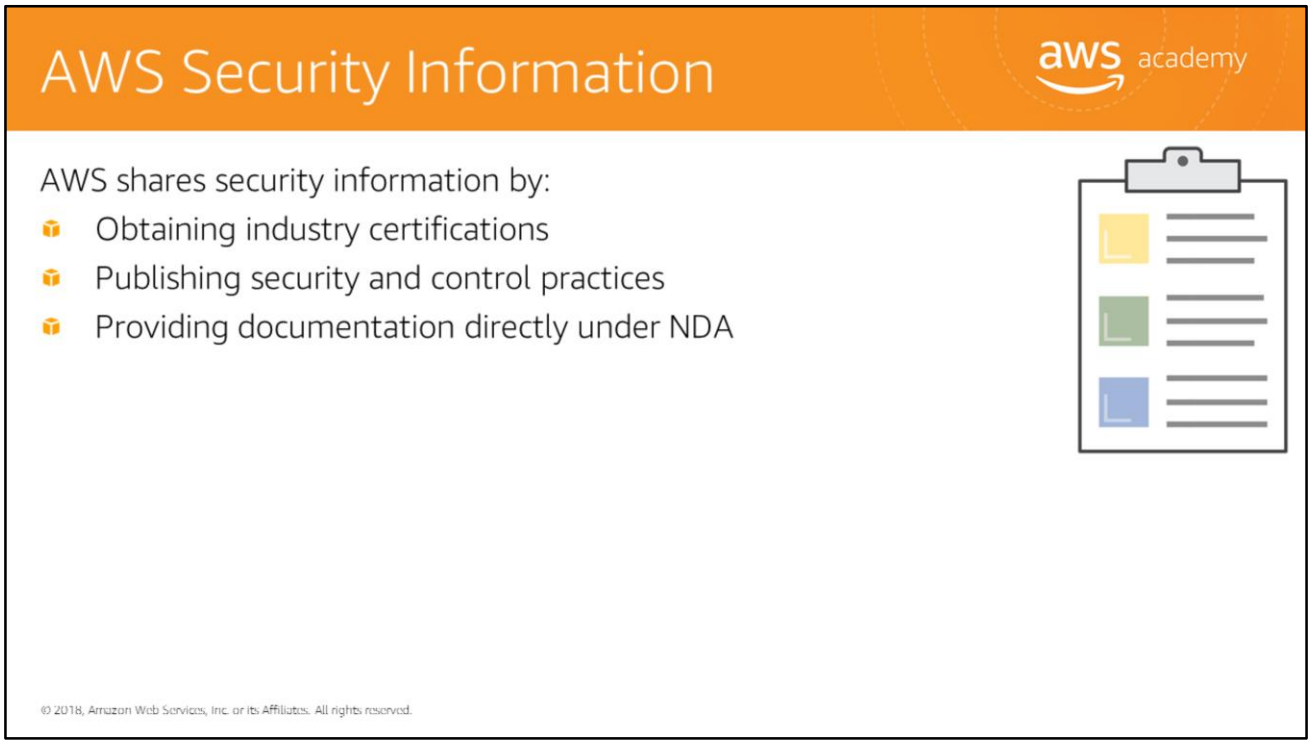
For more more information on the AWS Service Catalog see https://aws.amazon.com/servicecatalog/.

AWS Compliance Approach

- AWS and customers share control
- Responsibility of AWS:
  - Provide highly secure and controlled environment
  - Provide wide array of security features
- Responsibility of the customer:
  - Configure IT

As we discussed in the Shared Security Responsibility Model, AWS and its customers share control over the IT environment—which means both parties have responsibility for managing the IT environment. The responsibility of AWS in this model includes providing its services on a highly secure and controlled environment and providing a wide array of security features for customers to use.

The customers' responsibility includes configuring their IT environments in a secure and controlled manner for their purposes.



While customers don't communicate their use and configurations to AWS, AWS does communicate its security and control environment relevant to customers. AWS does this by doing the following:

- Obtaining industry certifications and independent third-party attestations
- Publishing information about the AWS security and control practices in whitepapers and web site content
- Providing certificates, reports, and other documentation directly to AWS customers under NDA (as required)

AWS Assurance Programs

AWS, certifying bodies, and independent auditors provide:

- Certifications/attestations
- Laws, regulations, and privacy
- Alignments/frameworks

AWS engages with external certifying bodies and independent auditors to provide customers with considerable information regarding the policies, processes, and controls established and operated by AWS.

Certifications/Attestations: Compliance certifications and attestations are assessed by a third-party, independent auditor and result in a certification, audit report, or attestation of compliance.

Laws, Regulation, and Privacy: AWS customers remain responsible for complying with applicable compliance laws and regulations. In some cases, AWS offers functionality (such as security features), enablers, and legal agreements (such as the AWS Data Processing Agreement and Business Associate Addendum) to support customer compliance.

Alignments/Frameworks: Compliance alignments and frameworks include published security or compliance requirements for a specific purpose, such as a specific industry or function. AWS provides functionality (such as security features) and enablers (including compliance playbooks, mapping documents, and whitepapers) for these types of programs.

AWS Risk and Compliance Programs

AWS Risk and Compliance Programs:

- Provide information about AWS controls
- Assist customers in documenting their framework

Components of AWS Risk and Compliance Programs

- Risk management
- Control environment
- Information security (IS)

AWS provides information about its Risk and Compliance Program to enable customers to incorporate AWS controls into their governance framework. This information can assist customers in documenting a complete control and governance framework with AWS included as an important part of that framework.

The AWS Risk and Compliance Program is made up of three components:

- Risk Management
- Control Environment
- Information Security

Let's take a look at each of the AWS Risk and Compliance Programs in more detail.

AWS Risk Management

- Business plan
  - Includes risk management
  - Plan re-evaluated at least biannually
- Responsibilities
  - Identifies risks
  - Implements appropriate measures to address risks
  - Assesses various internal/external risks
- Information security framework and policies based on:
  - Control Objectives for Information and related Technology (COBIT)
  - American Institute of Certified Public Accountants (AICPA)
  - National Institute of Standards and Technology (NIST)

AWS management has developed a strategic business plan that includes risk identification and the implementation of controls to mitigate or manage risks. AWS management re-evaluates the strategic business plan at least biannually. This process requires management to identify risks within its areas of responsibility and to implement appropriate measures designed to address those risks.

In addition, the AWS control environment is subject to various internal and external risk assessments.

The AWS Compliance and Security teams have established an information security framework and policies that are based on the following governing bodies:

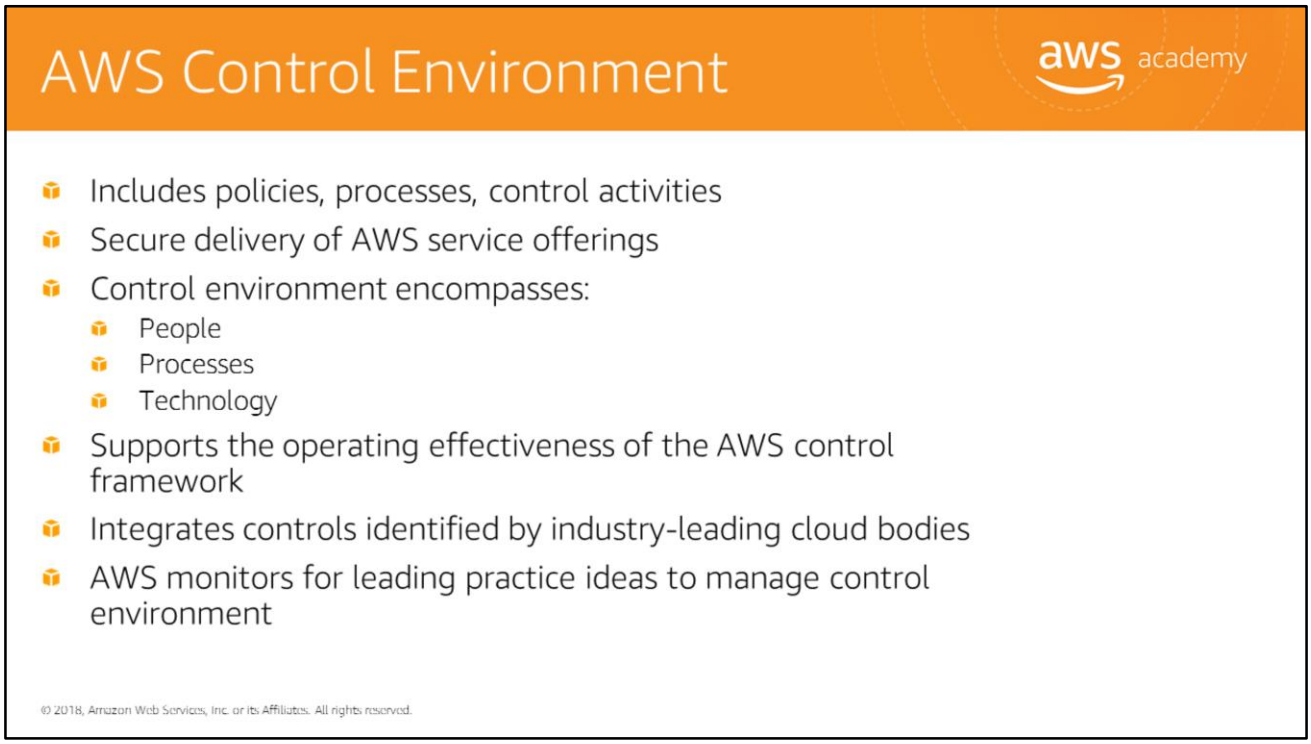
- Control Objectives for Information and related Technology (COBIT)
- American Institute of Certified Public Accountants (AICPA)
- National Institute of Standards and Technology (NIST)

AWS Risk Management

- AWS takes care of:
  - Maintaining the security policy
  - Providing security training to employees
  - Performing application security reviews to assess:
    - Data confidentiality, integrity, availability
    - Conformance to IS policy
- AWS security
  - Scans service endpoints for vulnerabilities
  - Notifies for remediation of vulnerabilities
- Independent security firms
  - Scans are not a replacement for customer scans
  - Customers can ask to scan cloud infrastructure

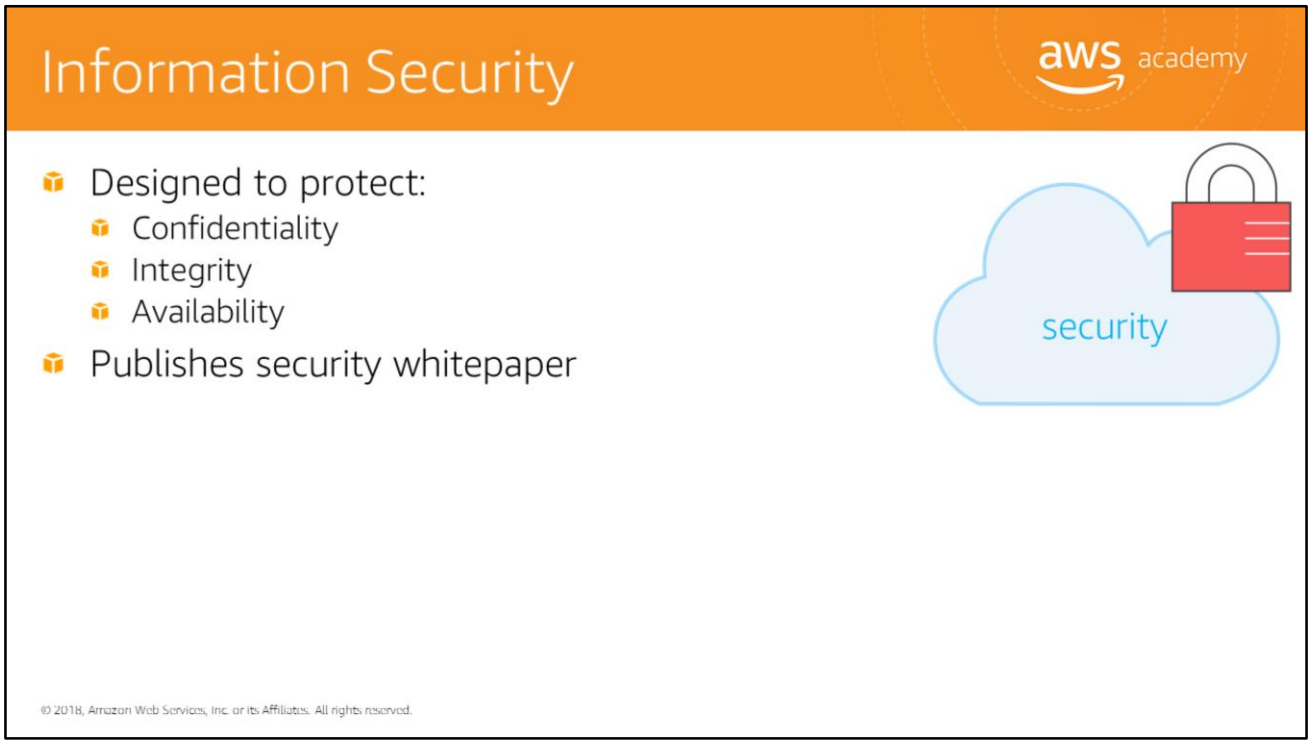
AWS maintains the security policy, provides security training to employees, and performs application security reviews. These reviews assess the confidentiality, integrity, and availability of data, as well as conformance to the information security policy.

AWS Security regularly scans all Internet facing service endpoint IP addresses for vulnerabilities (scans are not performed on customer EC2 instance interfaces). AWS Security notifies the appropriate parties to remediate any identified vulnerabilities. In addition, external vulnerability threat assessments are performed regularly by independent security firms. Findings and recommendations resulting from these assessments are categorized and delivered to AWS leadership. These scans are done in a manner for the health and viability of the underlying AWS infrastructure and are not meant to replace the customer's own vulnerability scans required to meet their specific compliance requirements. Customers can request permission to conduct scans of their cloud infrastructure as long as they are limited to the customer's instances and do not violate the AWS Acceptable Use Policy.



Control Environment

AWS manages a comprehensive control environment that includes policies, processes, and control activities that leverage various aspects of Amazon's overall control environment. This control environment is in place for the secure delivery of AWS service offerings. The collective control environment encompasses the people, processes, and technology necessary to establish and maintain an environment that supports the operating effectiveness of the AWS control framework. AWS has integrated applicable cloud-specific controls identified by leading cloud computing industry bodies into the AWS control framework. AWS continues to monitor these industry groups for ideas on which leading practices can be implemented to better assist customers with managing their control environment.



Information Security
AWS has implemented a formal information security program designed to protect the confidentiality, integrity, and availability of customers' systems and data. AWS publishes a security whitepaper that is available on the public website that addresses how AWS can help customers secure their data.

To learn more about compliance and find additional resources for this topic, see https://aws.amazon.com/compliance/.

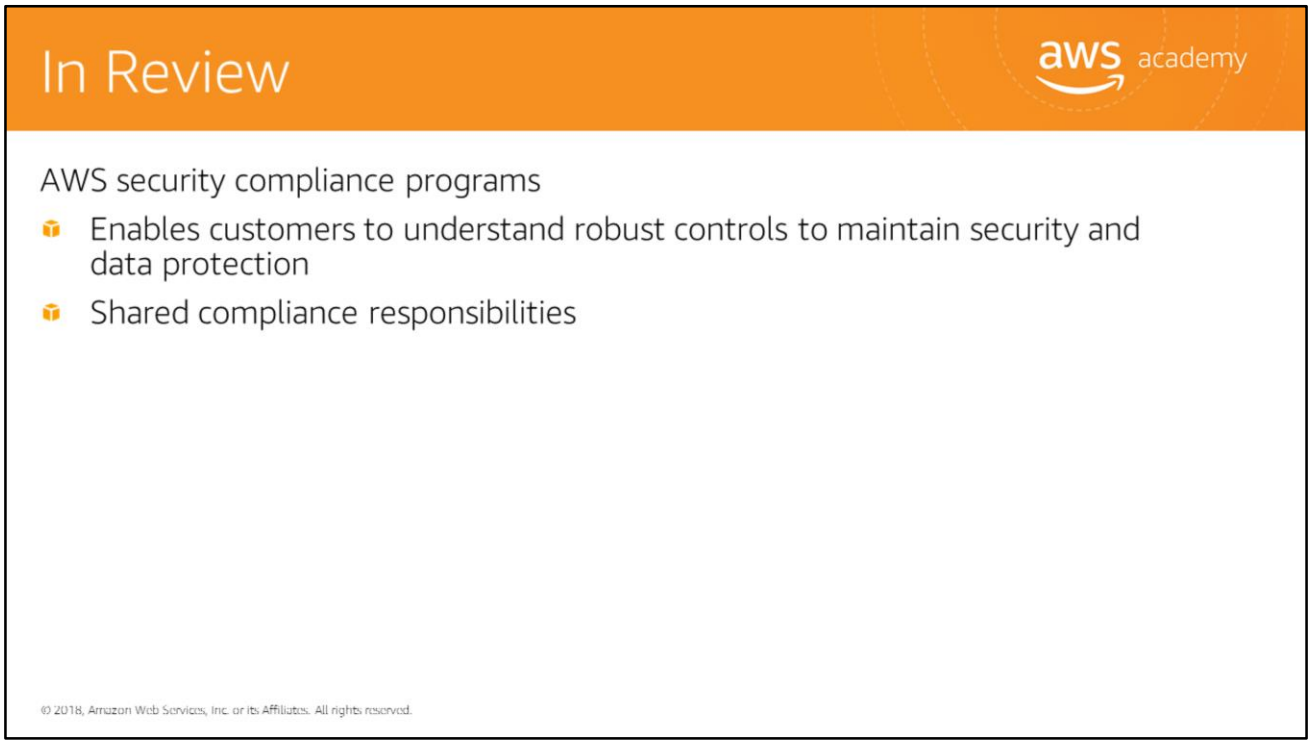
Customer Compliance Requirements

- Maintain governance over the entire IT control environment
- Customers should understand:
  - Required compliance objectives
  - Validation-based risk tolerance
- Establish control environment
- Verify effectiveness of control environment
- Customer compliance basic approach:
  - Review
  - Design
  - Identify
  - Verify

AWS customers are required to continue to maintain adequate governance over the entire IT control environment regardless of how IT is deployed. Leading practices include an understanding of required compliance objectives and requirements (from relevant sources), establishment of a control environment that meets those objectives and requirements, an understanding of the validation required based on the organization's risk tolerance, and verification of the operating effectiveness of their control environment. Deployment in the AWS cloud gives enterprises different options to apply various types of controls and various verification methods.

Strong customer compliance and governance might include the following basic approach:
Review information available from AWS together with other information to understand as much of the entire IT environment as possible, and then document all compliance requirements.
Design and implement control objectives to meet the enterprise compliance requirements.
Identify and document controls owned by outside parties.
Verify that all control objectives are met and all key controls are designed and operating effectively.

By staying engaged in the compliance and governance process with AWS, customers can ensure compliance requirements are being met.



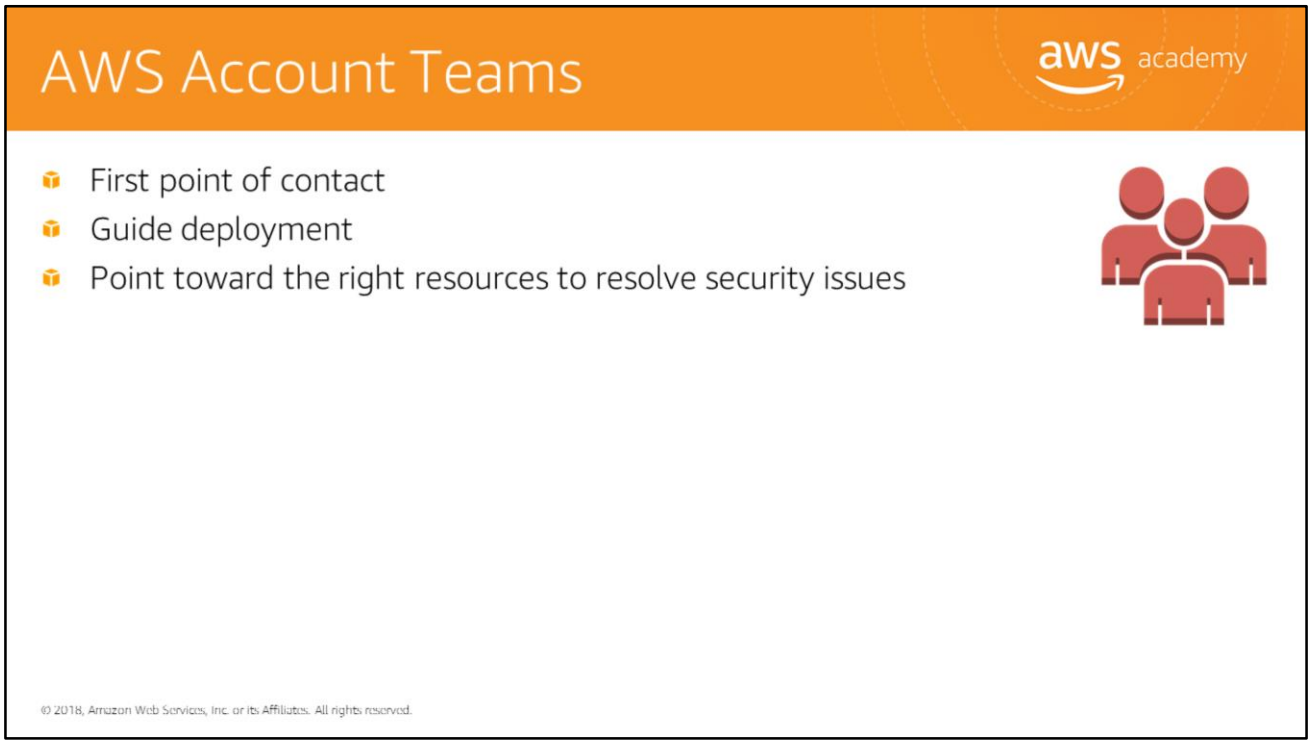
AWS Cloud Compliance enables customers to understand the robust controls in place at AWS to maintain security and data protection in the cloud. As systems are built on top of AWS cloud infrastructure, compliance responsibilities will be shared.

By tying together governance-focused, audit-friendly service features with applicable security compliance regulations or audit standards, AWS Compliance enablers build on traditional programs; helping customers to establish and operate in an AWS security controlled environment.

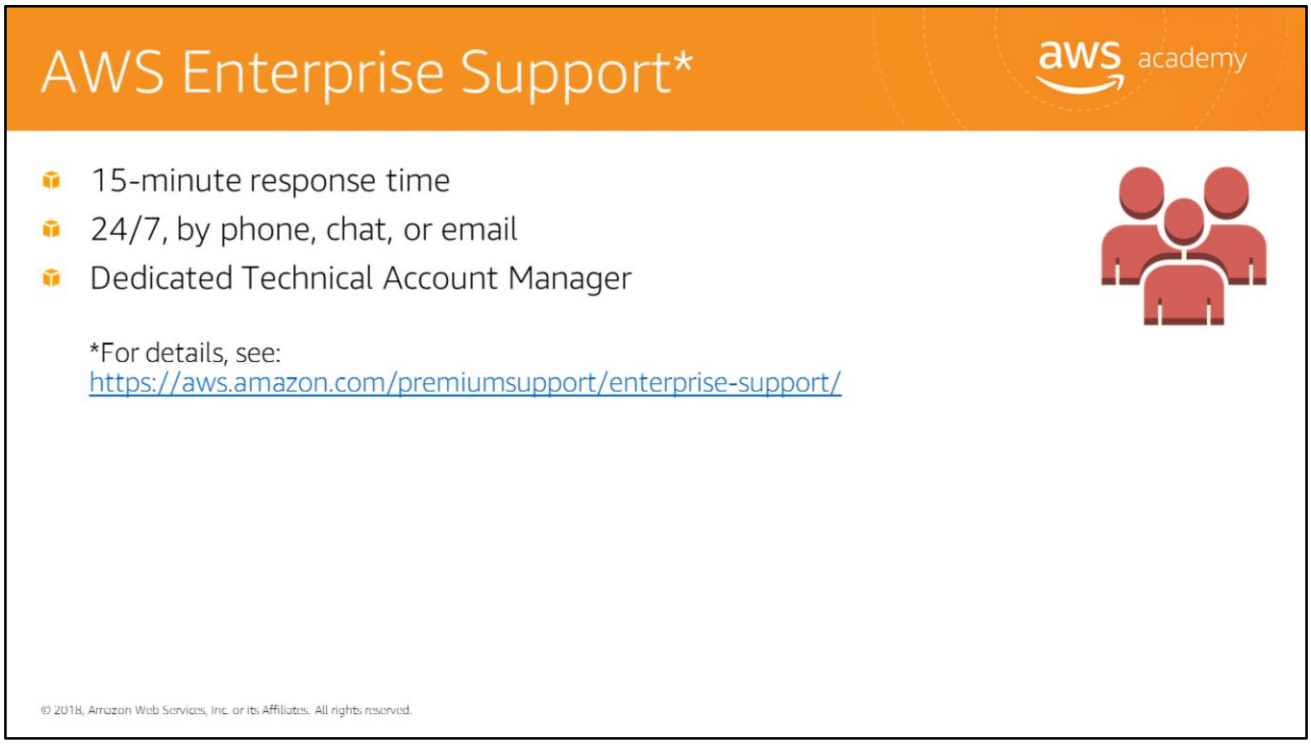
As we mentioned before, AWS communicates its security and control environment relevant to customers by doing the following:

- Industry certifications and independent third-party attestations
- Information about AWS security and control practices in whitepapers and web content
- Certificates, reports, and other documentation provided directly to AWS customers under NDA

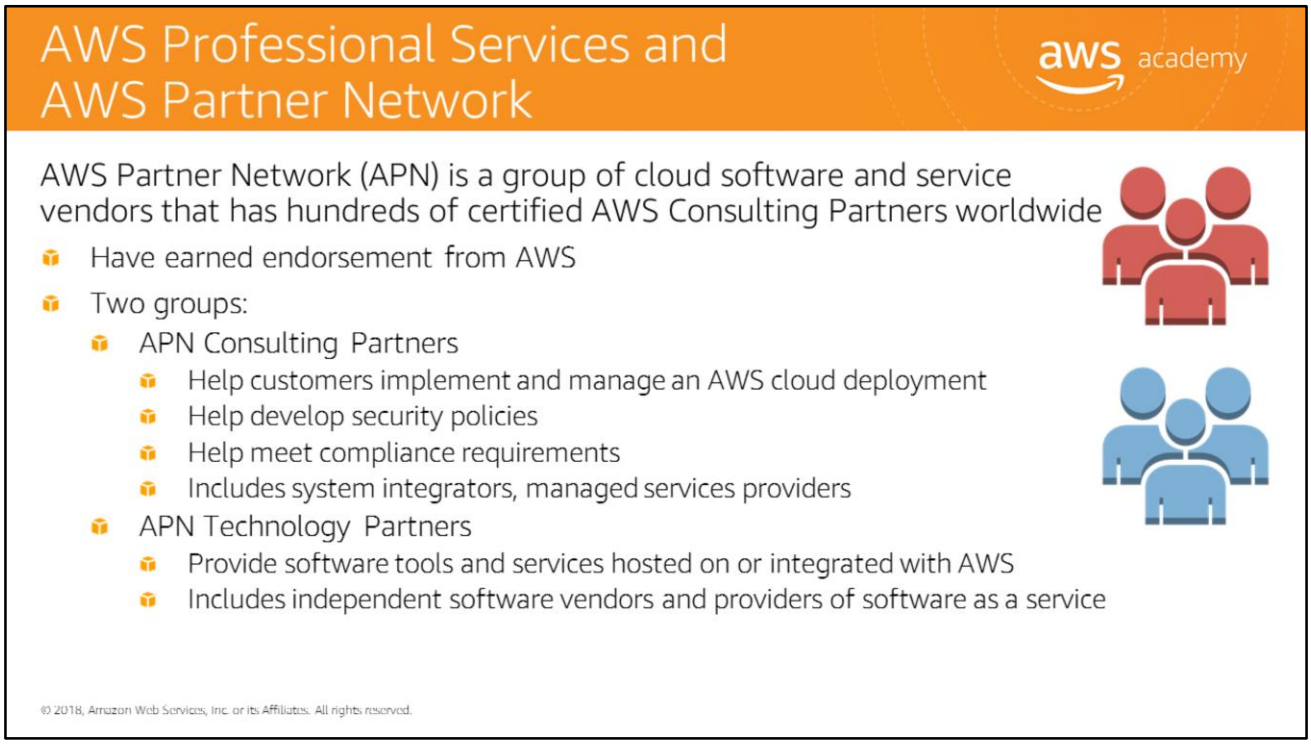
Let's take a closer look at how AWS provides customers with guidance and expertise through online tools, resources, support, and professional services to secure their data in the cloud.



AWS Account Teams
AWS Account Teams provide a first point of contact, guiding you through your deployment and implementation, and pointing you toward the right resources to resolve security issues you may encounter.

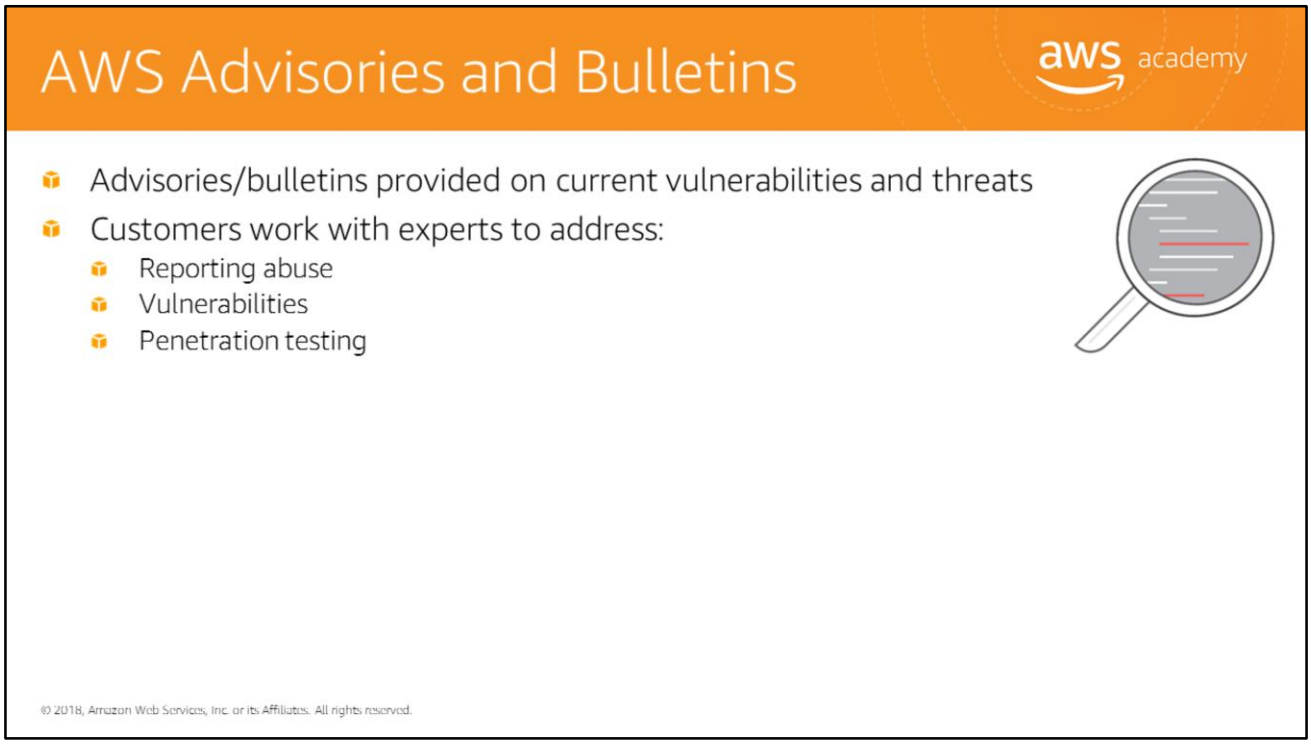


AWS Enterprise Support
AWS Enterprise Support provides 15-minute response time and is available 24×7 by phone, chat, or email; along with a dedicated Technical Account Manager. This concierge service ensures that customers' issues are addressed as swiftly as possible.



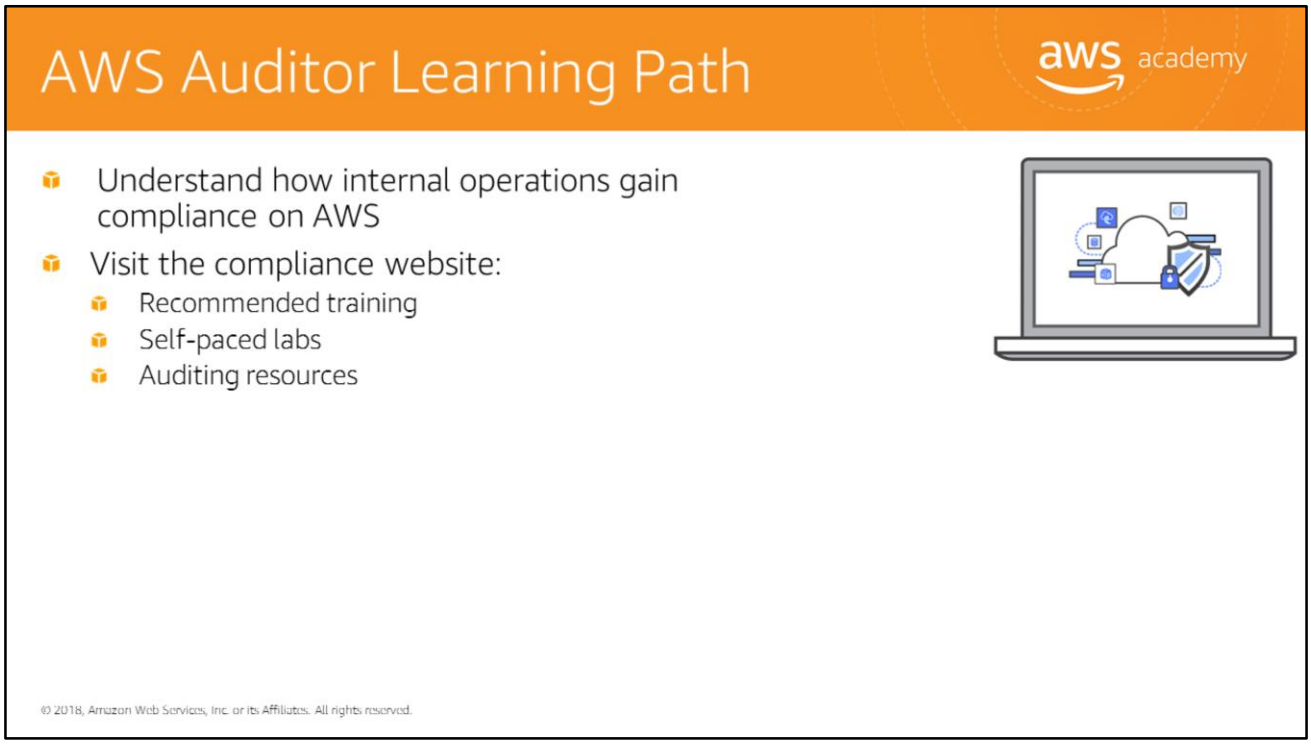
AWS Professional Services and AWS Partner Network
AWS Professional Services and AWS Partner Network both help customers develop security policies and procedures based on well-proven designs, and help to ensure that customers' security design meets internal and external compliance requirements.

The AWS Partner Network has hundreds of certified AWS Consulting Partners worldwide to help customers with their security and compliance needs.



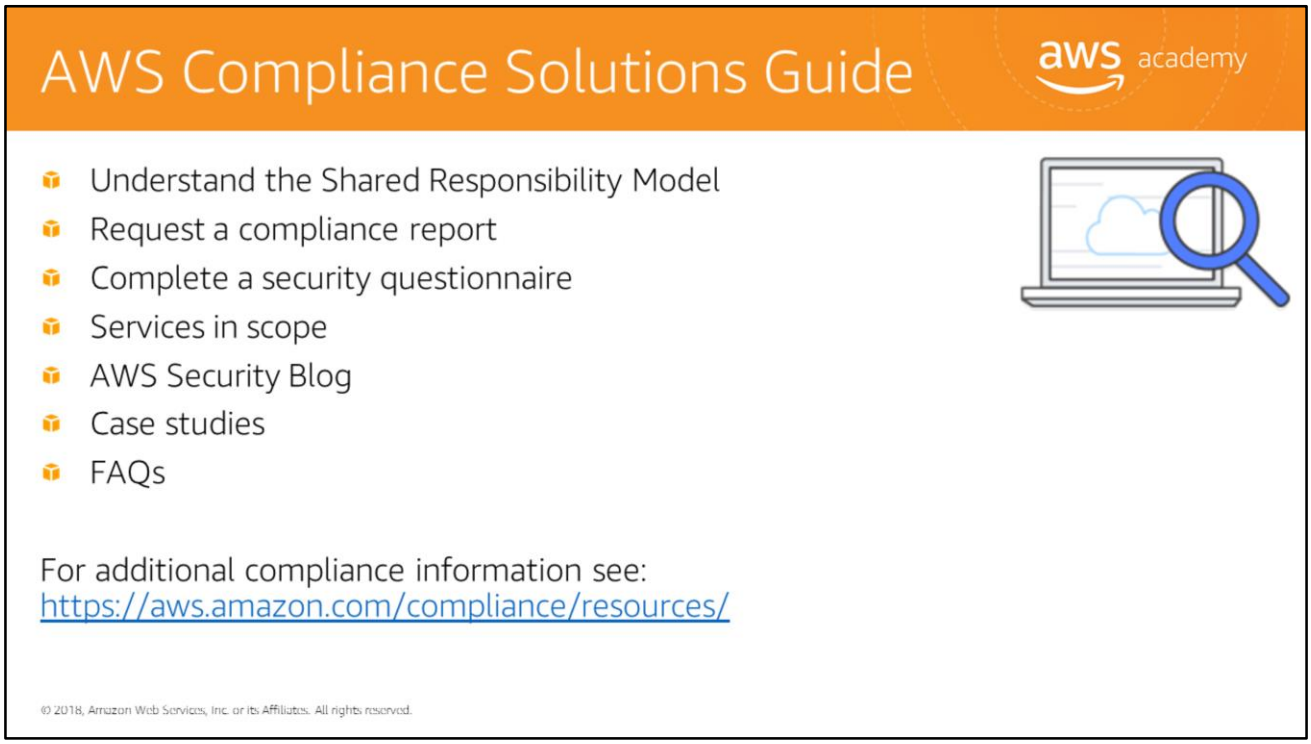
AWS Advisories and Bulletins

With AWS Advisories and Bulletins, AWS provides advisories around current vulnerabilities and threats, and enables customers to work with AWS security experts to address concerns like reporting abuse, vulnerabilities, and penetration testing.



AWS Auditor Learning Path

If you're in an auditing, compliance, or legal role, check out AWS Auditor Learning Path to get a better understanding of how your internal operations can demonstrate compliance using AWS' services. You can access Recommended Training, self-paced labs, and auditing resources from the Compliance website.

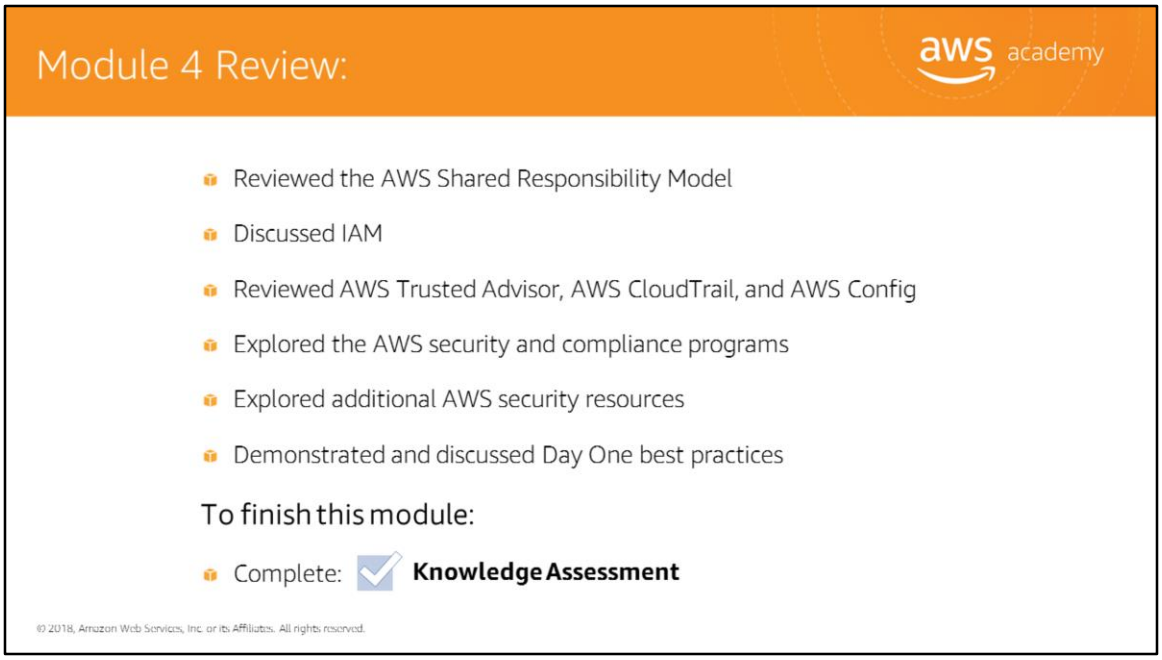


AWS Compliance Solutions Guide
If you don't know where to start with compliance or need to access frequently used resources and processes, check out the AWS Compliance Solutions Guide. Learn about the available compliance solutions available such as:
Understanding the Shared Responsibility Model
Requesting a Compliance Report
Completing a Security Questionnaire

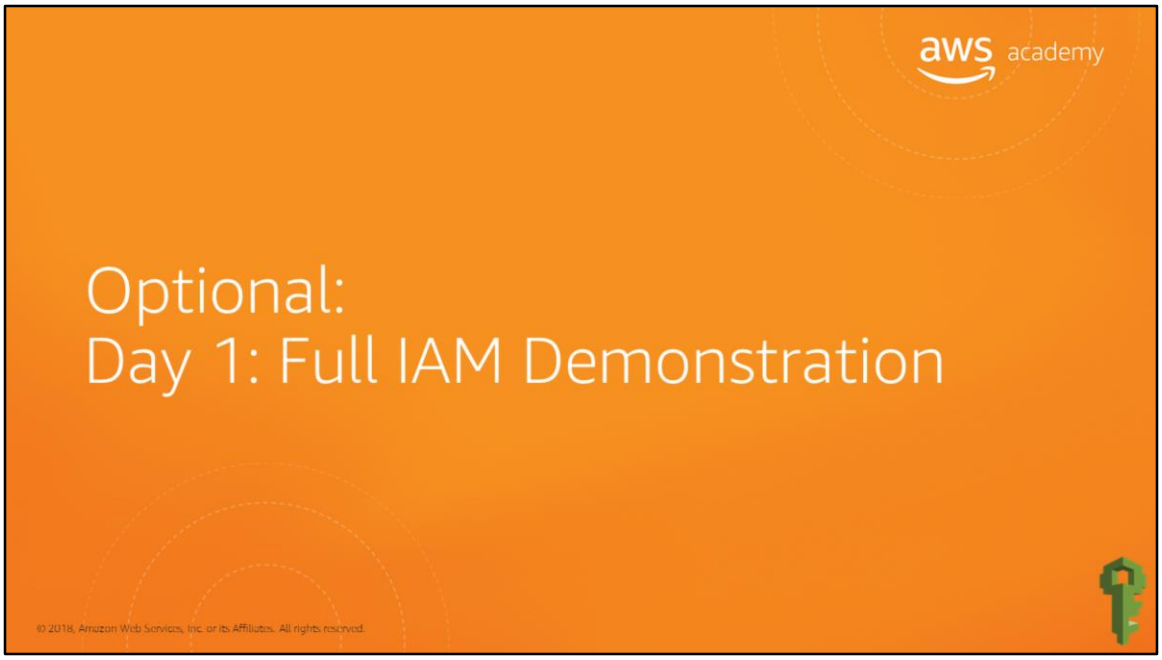
More AWS Compliance Resources
Other helpful compliance resources include:
- **Services in Scope** –Details which services are currently in scope and which are in progress.
- **AWS Security Blog** – The blog is a great way to track all the newest updates to AWS security programs.
- **Case Studies** –Provide insightful information on some of the AWS current customer experiences with security.

You can also get answers to frequently asked questions for specific compliance types, such as:
- Certifications and attestations
    - Payment Card Industry (PCI)

    - System & Organization Control (SOC)
    - Federal Risk and Authorization Management Program (FedRAMP)
- Laws and regulations
    - U.S. Health Insurance Portability and Accountability Act (HIPAA)



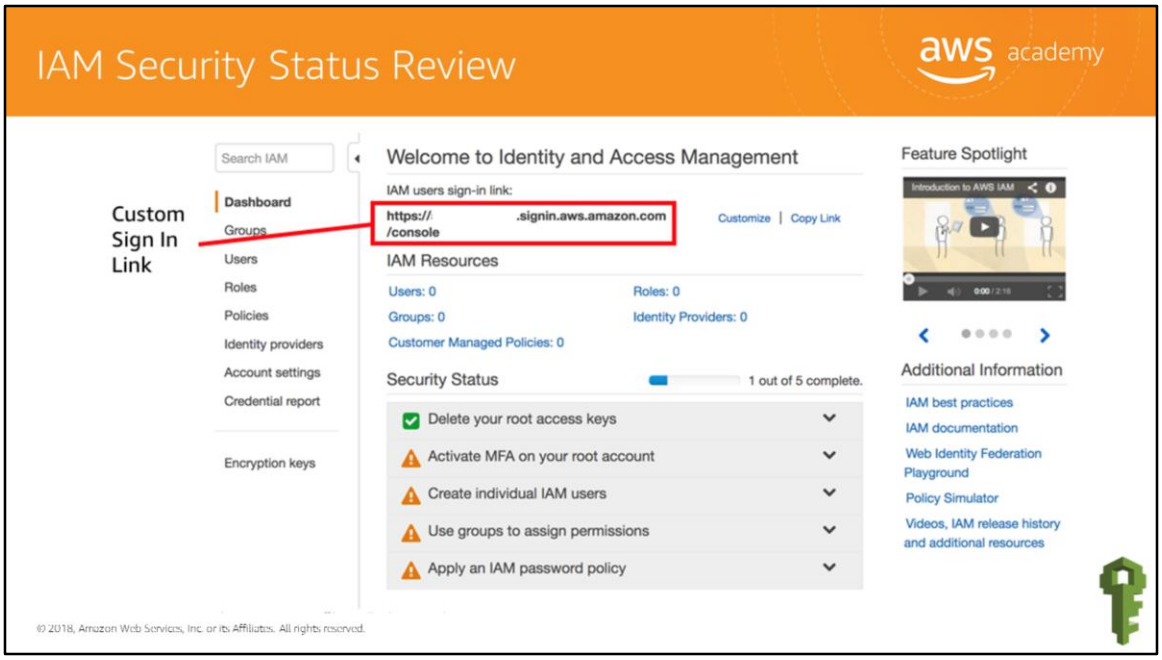
Now that we have a better understanding some of the security considerations and related AWS services, in Module 4 we look at the principles to consider whether you are migrating existing apps to AWS or designing new applications for the cloud.



Day 1 Demo:

1. Go to IAM.
2. Review Security Status. Do these four things:
    - Activate MFA on your root account
    - Create individual IAM users
    - Use groups to assign permissions
    - Apply an IAM password policy

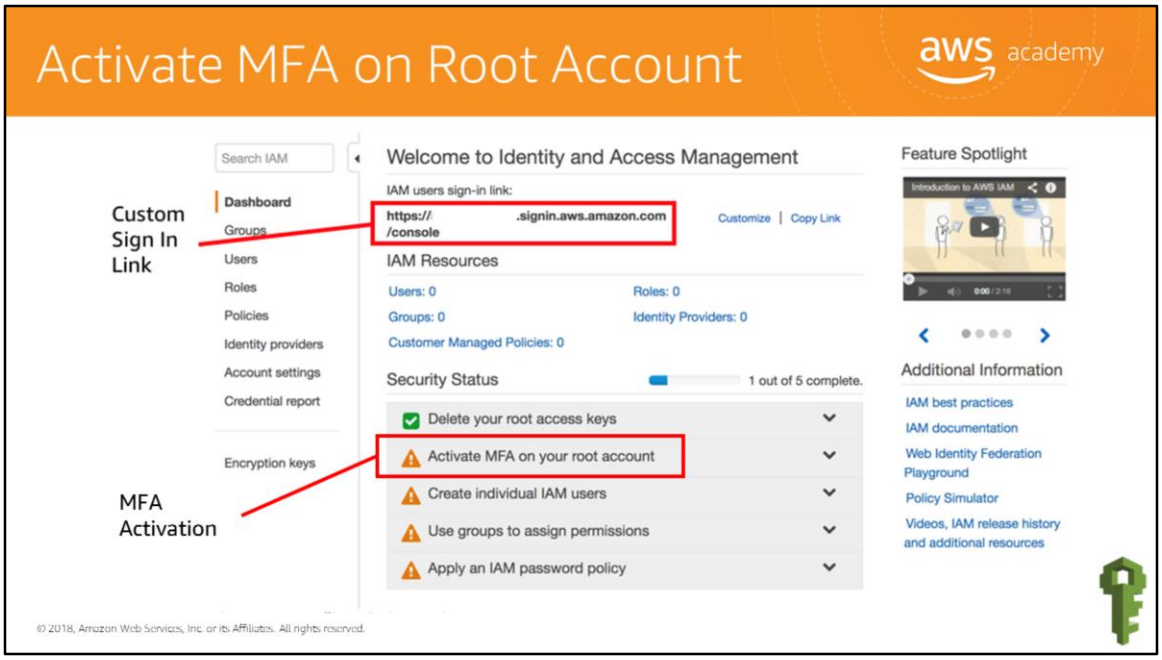
- GOAL: Green checks by each security status item.



A review of the current Security Status indicates that:

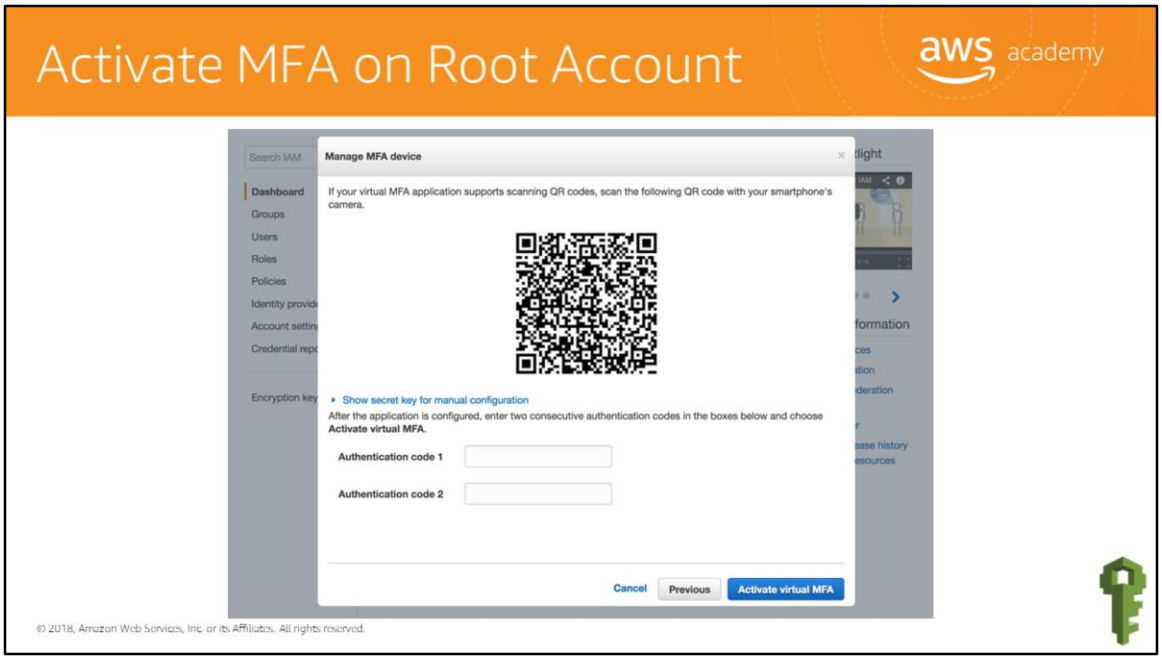
- MFA has not been activated on the root account
- No individual IAM users have been created
- No permissions have been assigned to groups
- No IAM password policy has been applied

There is a custom sign-in link for the account (account number hidden for security purposes). Use the **Customize** button to change the name of the account so that the account number is not displayed. This link is used to sign in to the account and can be sent to users as they are set up.

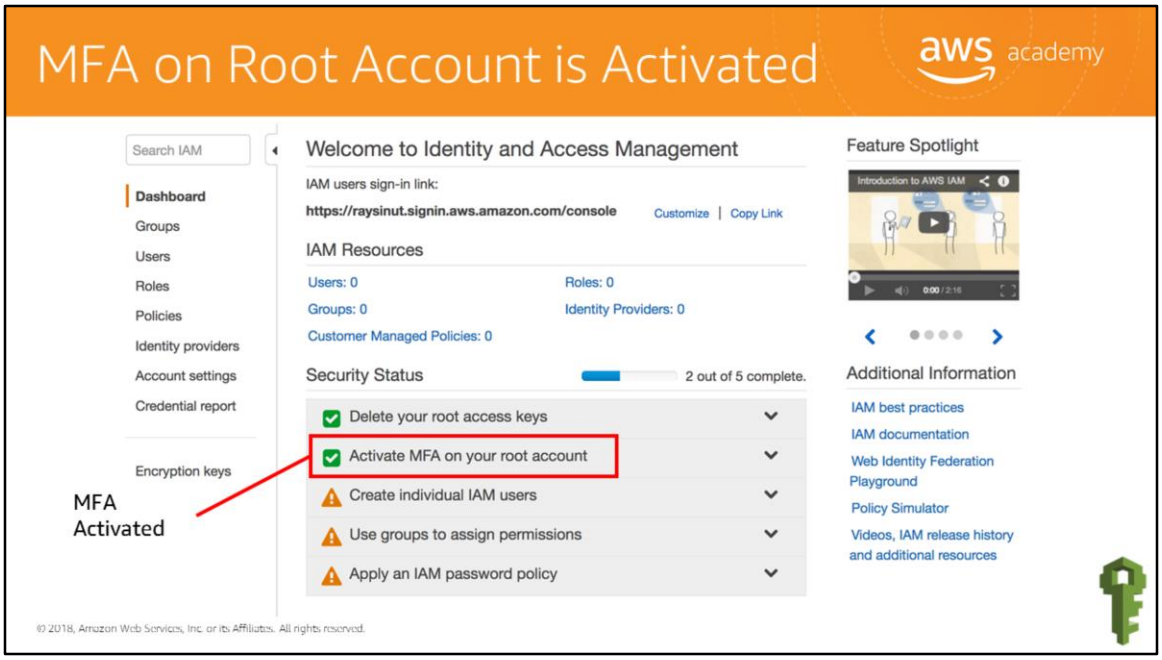


Before creating the users, activate MFA on the root account. The root account is the email address that you used to sign up for the AWS account. The root account has access to everything—that's why it's important to secure this account with restrictions.

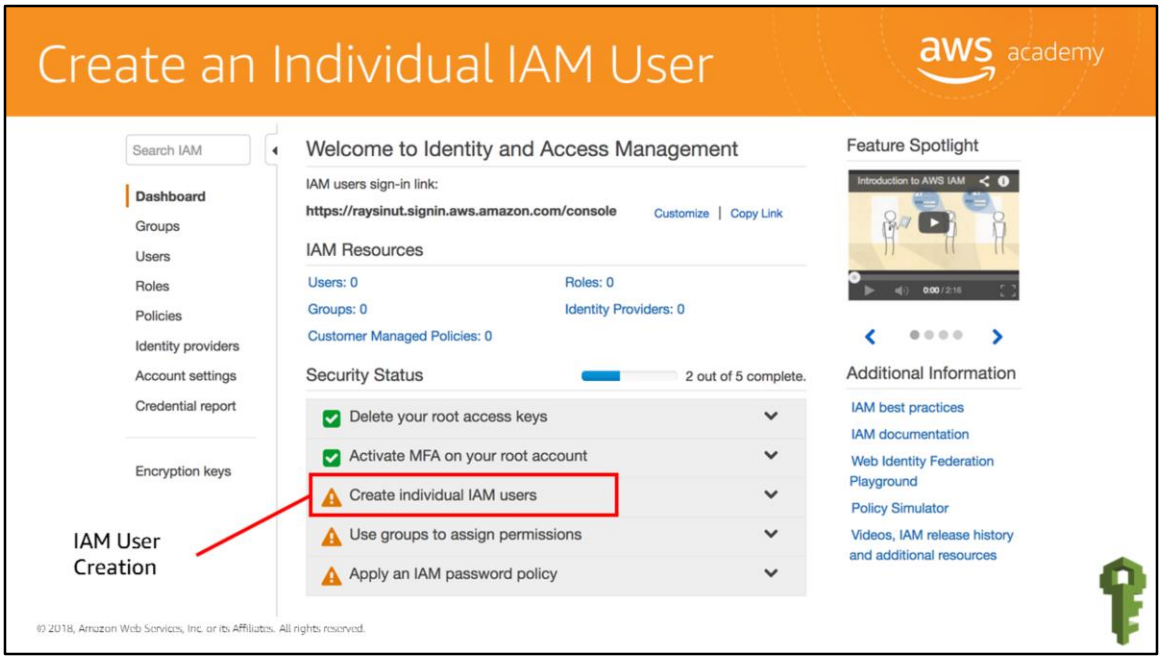
To set up MFA, click **Activate MFA on your root account** and then click **Manage MFA**. Two options will be presented: Virtual and Hardware. A hardware device is an actual hardware device. For purposes of this demonstration, select **Virtual** and then click **Next Step**. A new dialog box appears and asks us to configure a virtual MFA device. An app must be downloaded for this task. After that is complete, click **Next**.



In the authenticator application, press the **plus sign**. Scan the barcode, enter the two authentication codes, and click the **Activate Virtual MFA** button.

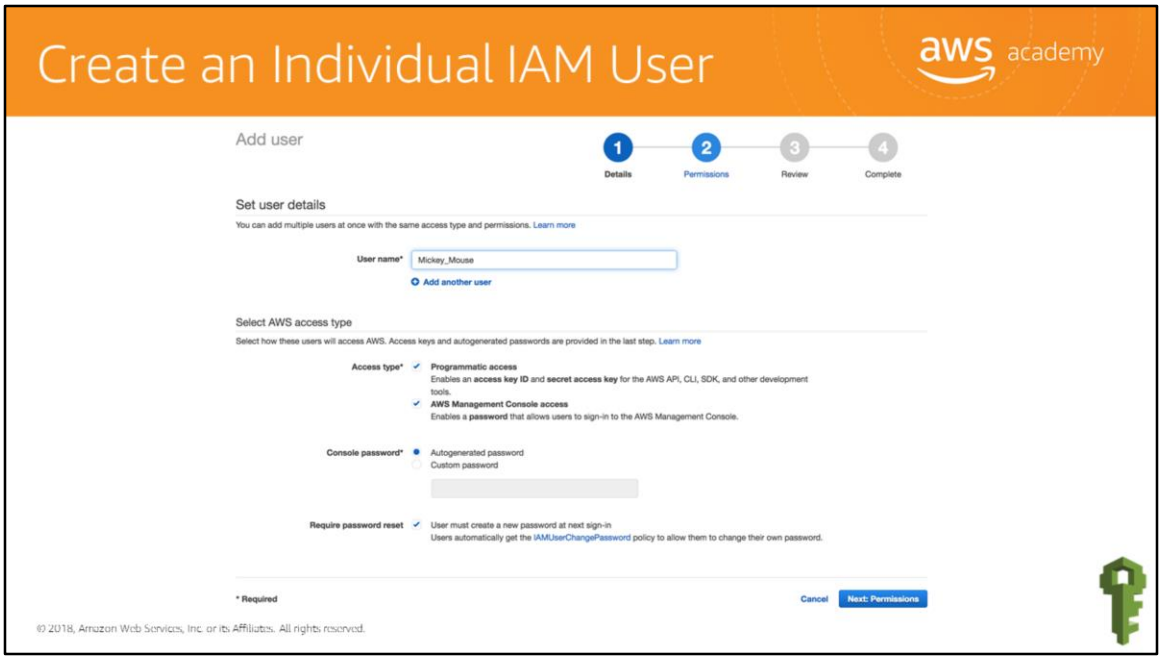


Click **Finish** and refresh your browser. The MFA should now show that it is set up.



Most AWS accounts are set up as company accounts with multiple user. Each user is set up with individual permissions or included as part of a group with specific permissions. A best practice is to have each user have their own account so they are not logging in a root with global privileges.

Click **Create individual IAM users** and then click **Manage Users**.

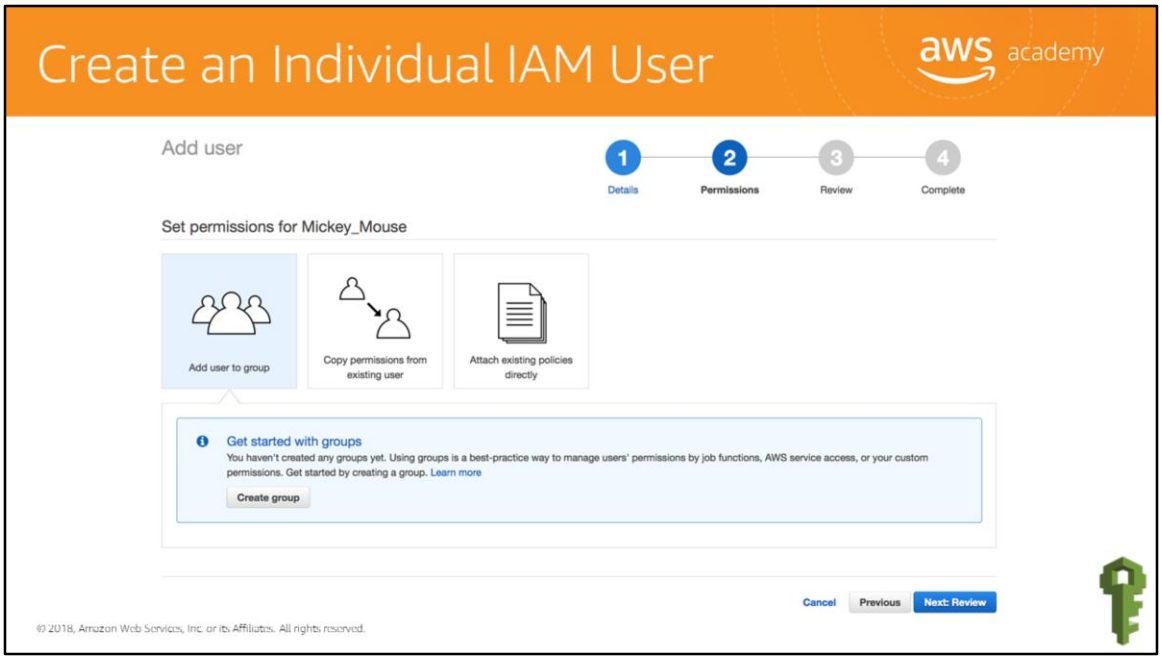


Click **Add another user**. Add a username. Note that usernames cannot have spaces.

Select the **Access types**. There are two access types:

- **Programmatic access** enables the user to have command line access to provision resources. This option will generate an access key one time. This access key must be saved as it will be used for all future access.
- **AWS Management Console access** enables user to log in to the AWS console.

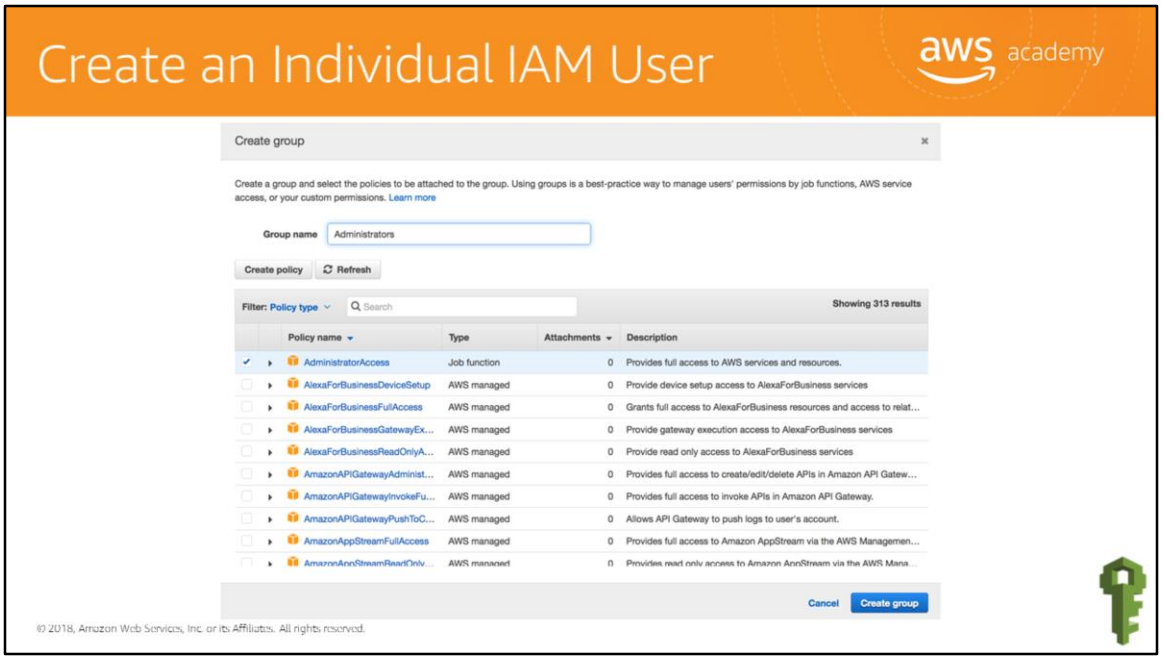
Select a password type.



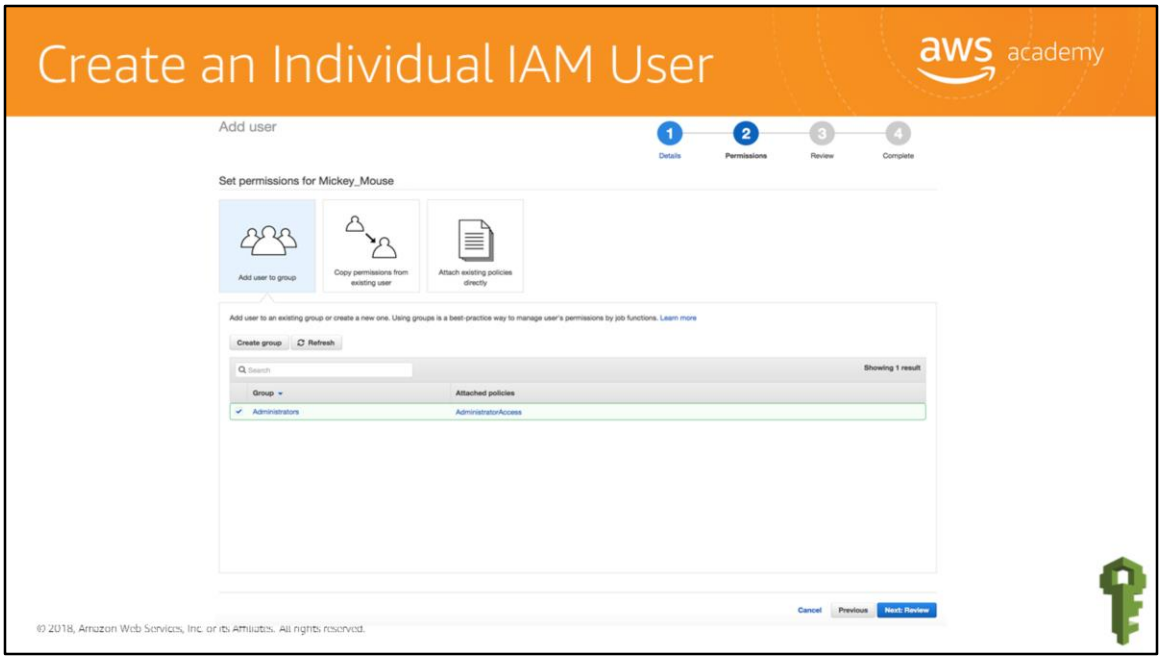
Next, you will assign permissions. There are three options:

- Add user to group
- Copy permissions from an existing user
- Attach existing policies directly

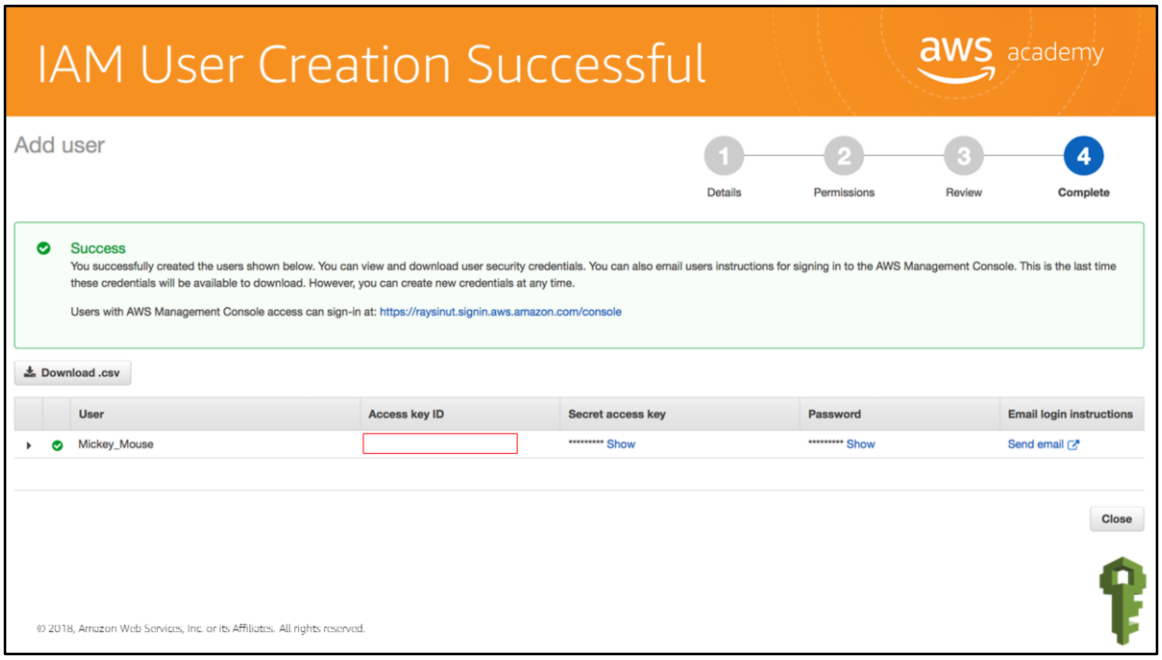
We want to add the user to a group, so select **Add user to group** and then click the **Create group** button. Group is where you put users to inherit the policies assigned to the group.



Give the group a name. For this example, give the lead developer administrative access. Click the **Create group** button.



Click **Next Review** to review what is being created, and then click **Create user**.

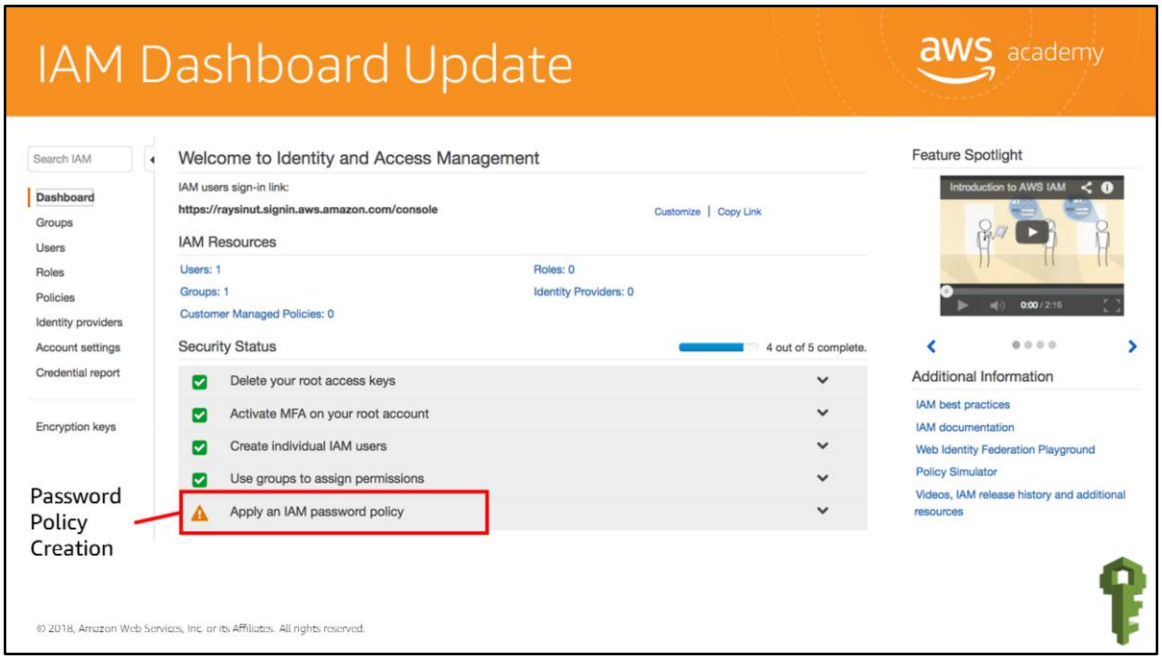


When a user is created, several things are generated:

- **Access key ID**: used to access AWS at the command line to programmatically access AWS (blocked out)
- **Secret access key**: used to access AWS at the command line to programmatically access AWS
- **Password**: used to log in to the console

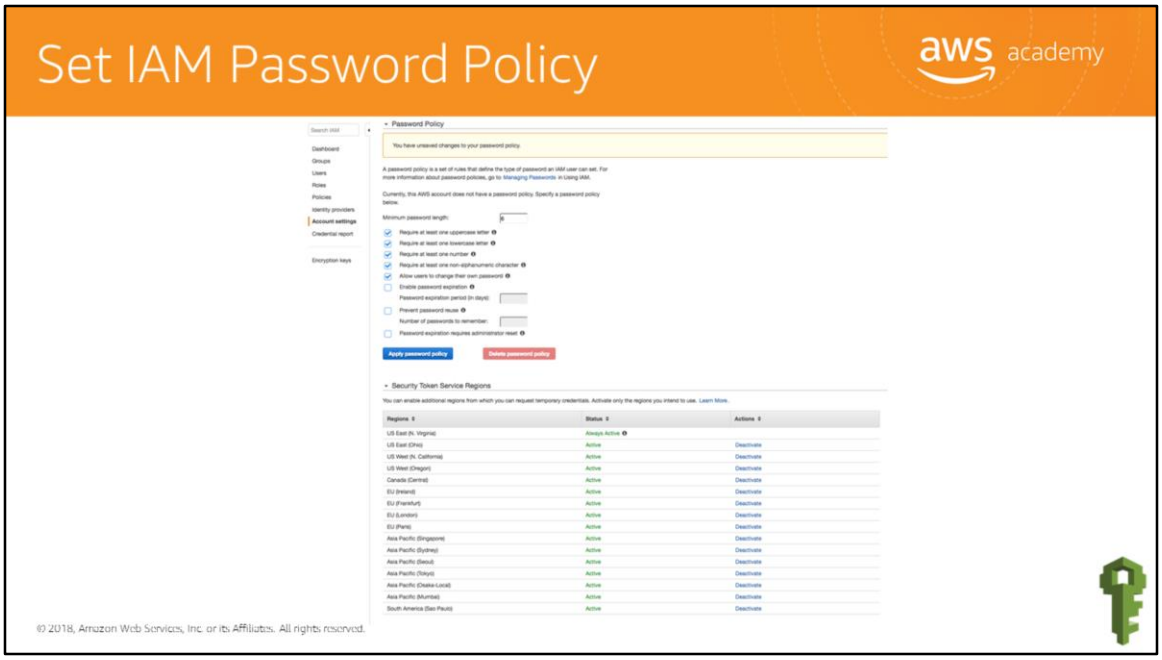
Pressing show will display the values in each of the fields. All of the information can be downloaded by selecting the **Download csv button**.

Note: Never put this information in a public place. This information can be used to access your account.



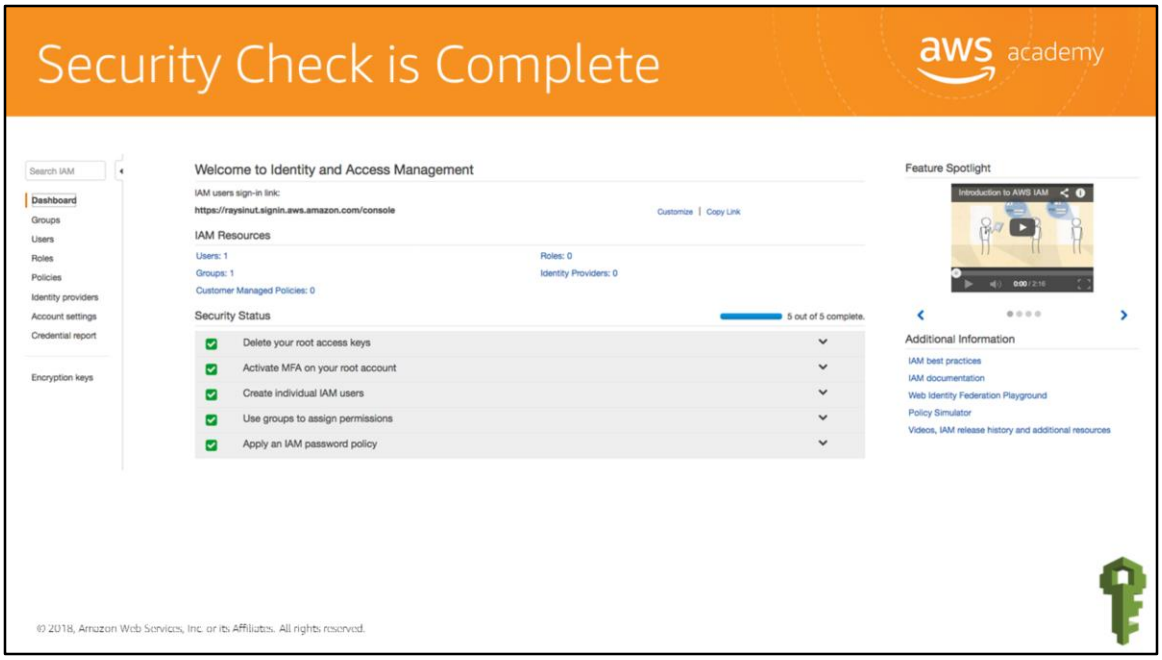
When you return to the dashboard, the individual IAM user and group security status items have been addressed.

The last thing to do is apply an IAM password policy.

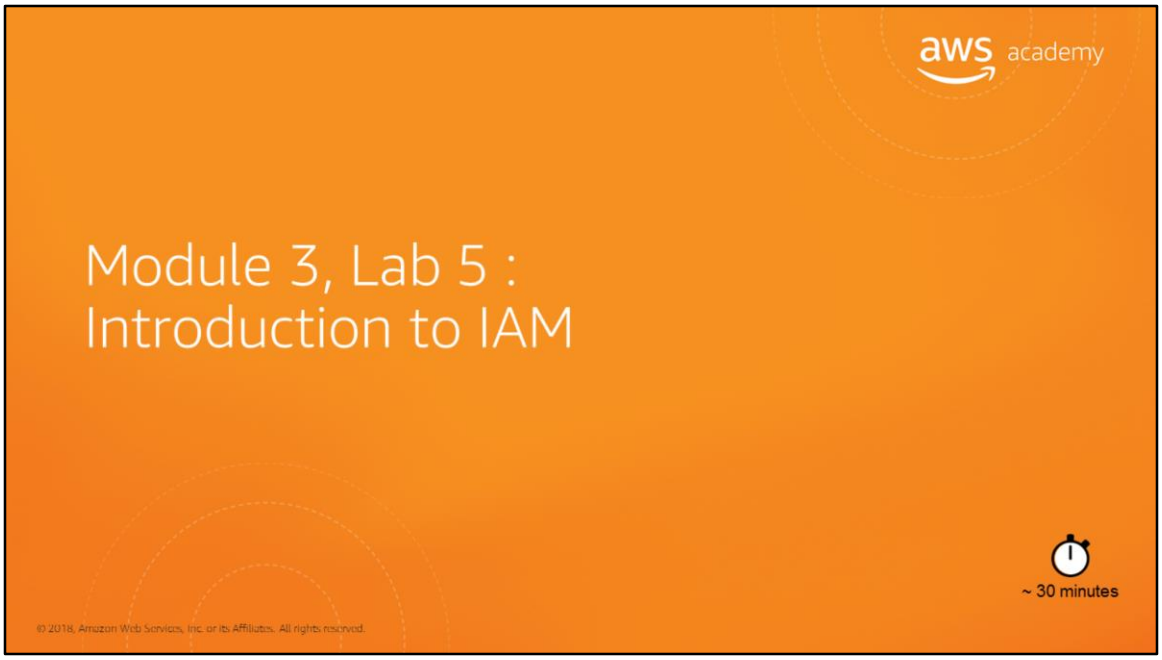


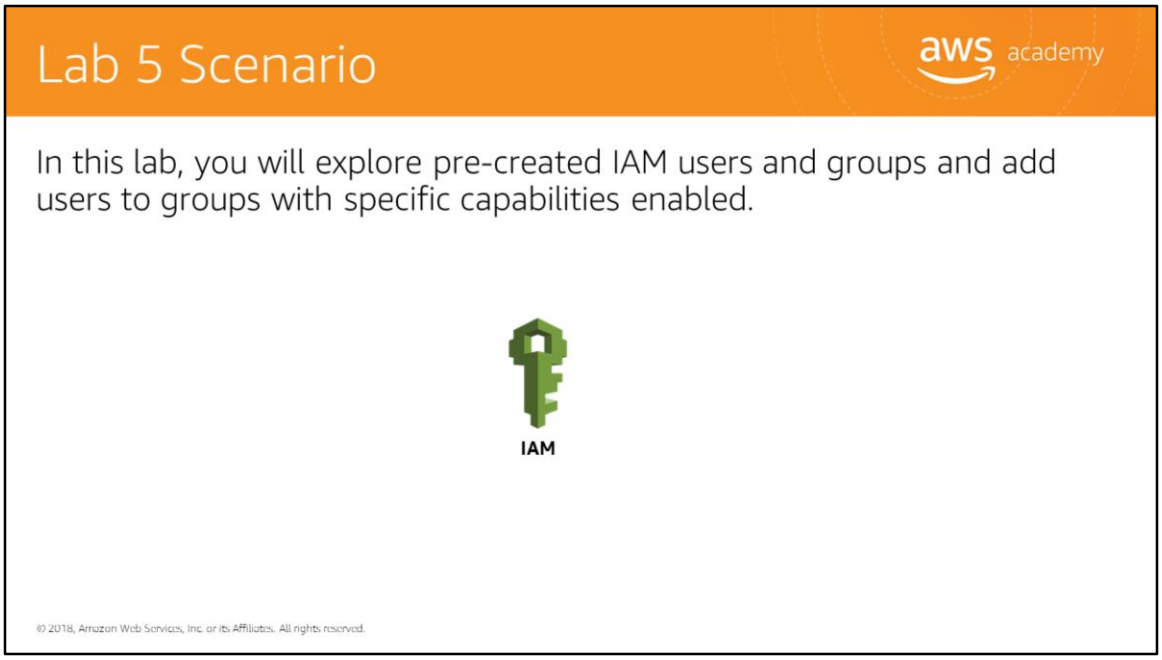
The IAM password policy is a set of rules that defines the type of password that an IAM user can set.

Select the rules that passwords should follow and click the **Apply password policy** button.



All the security status checkmarks are green, so the IAM setup requirements are complete.

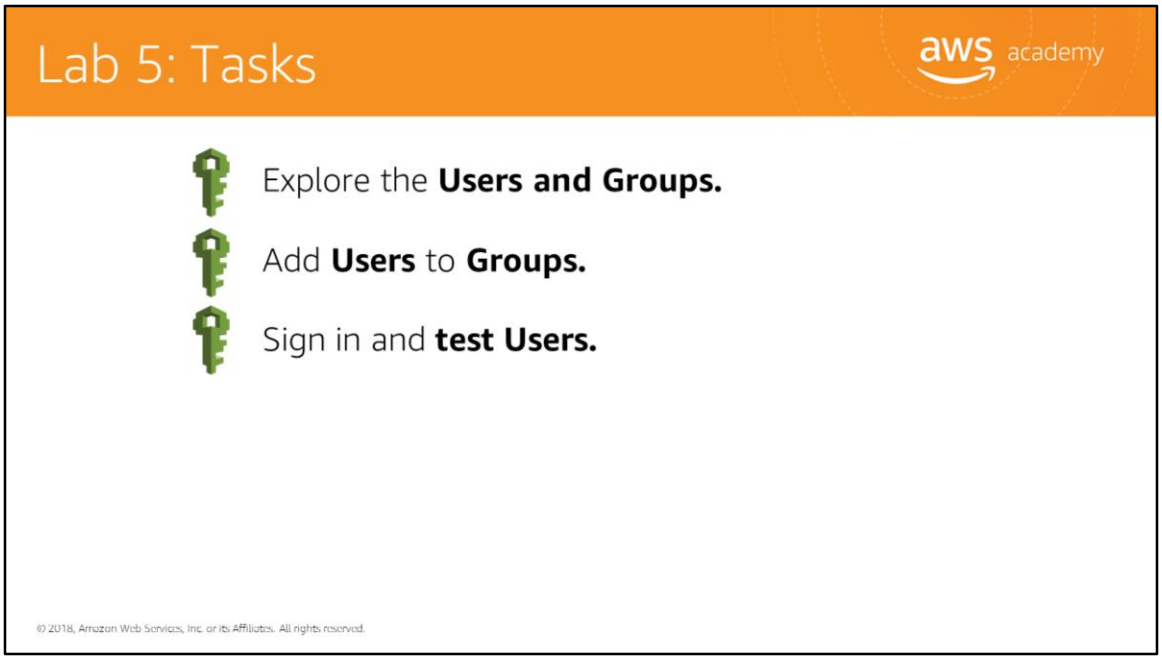


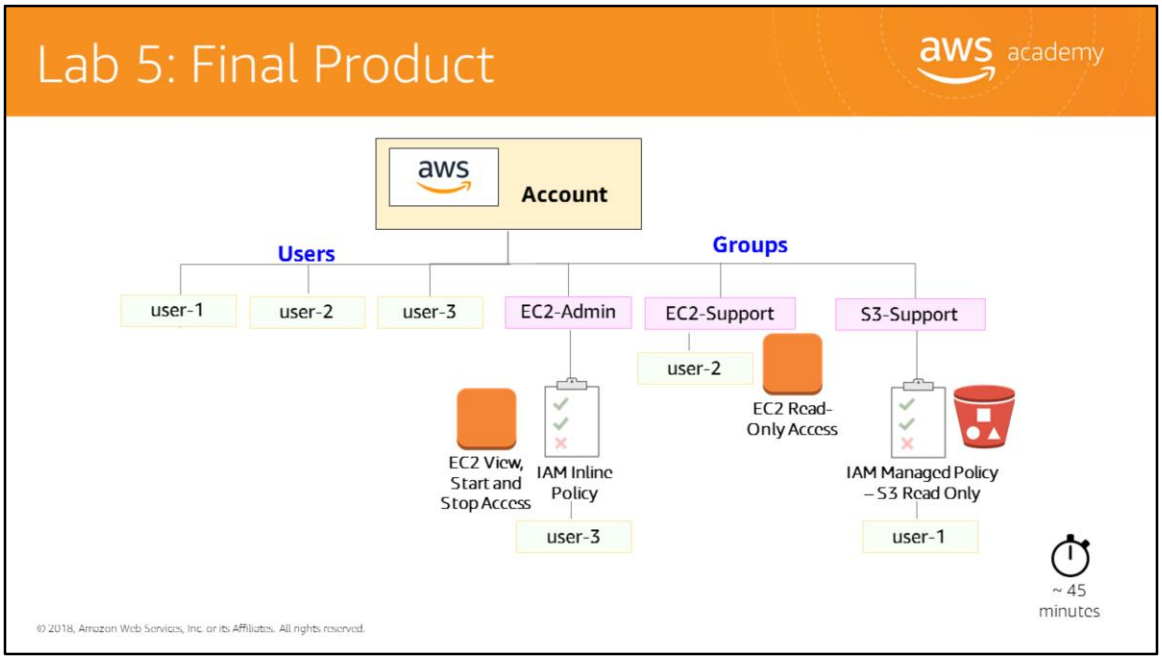


IAM is a web service that enables AWS customers to manage users and user permissions in AWS. With IAM, you can centrally manage users, security credentials such as access keys, and permissions that control which AWS resources users can access.

In this lab, you will:

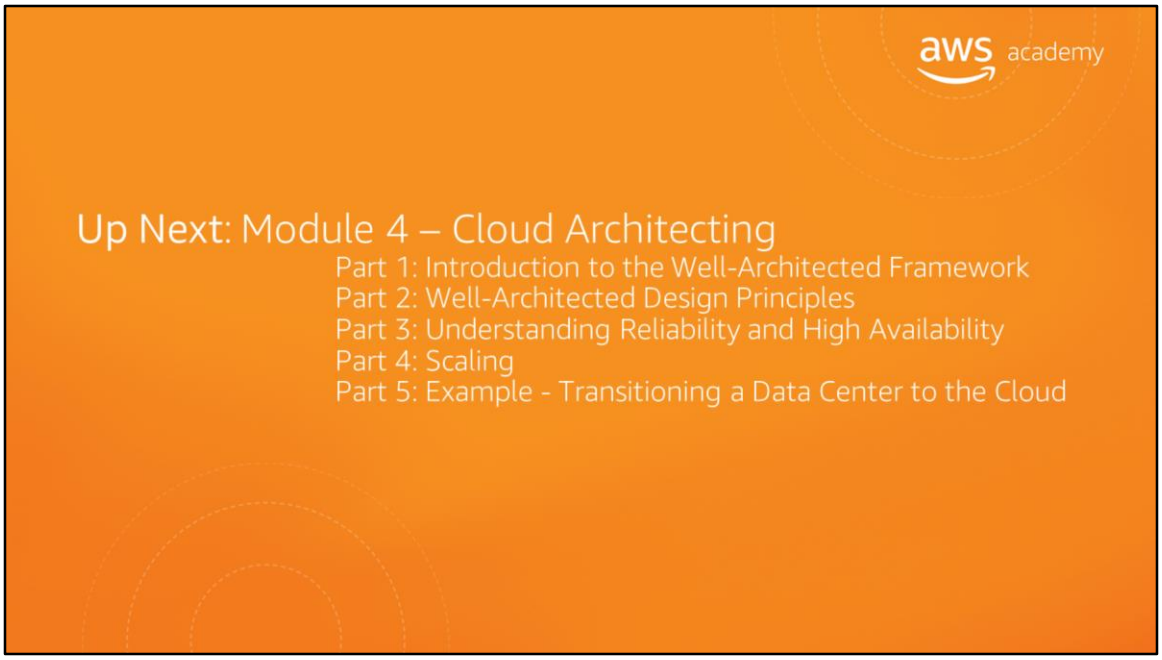
- Explore pre-created IAM users and groups
- Inspect IAM policies as applied to the pre-created groups
- Follow a real-world scenario, adding users to groups with specific capabilities enabled
- Locate and use the IAM sign-in URL
- Experiment with the effects of policies on service access



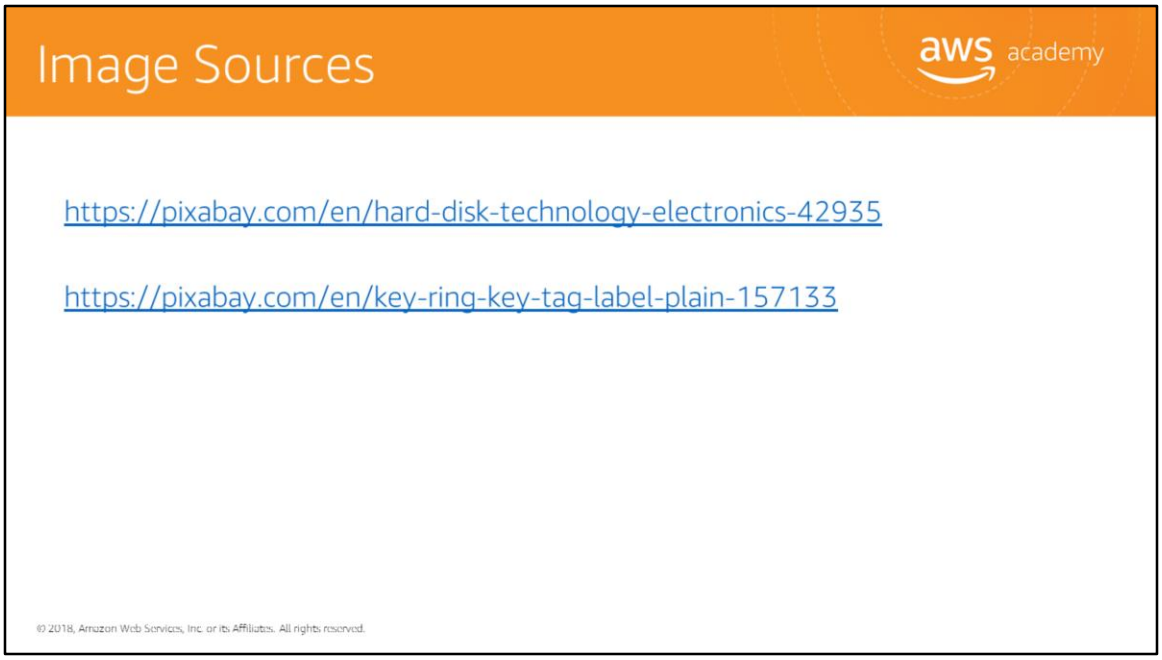


In this lab, you:

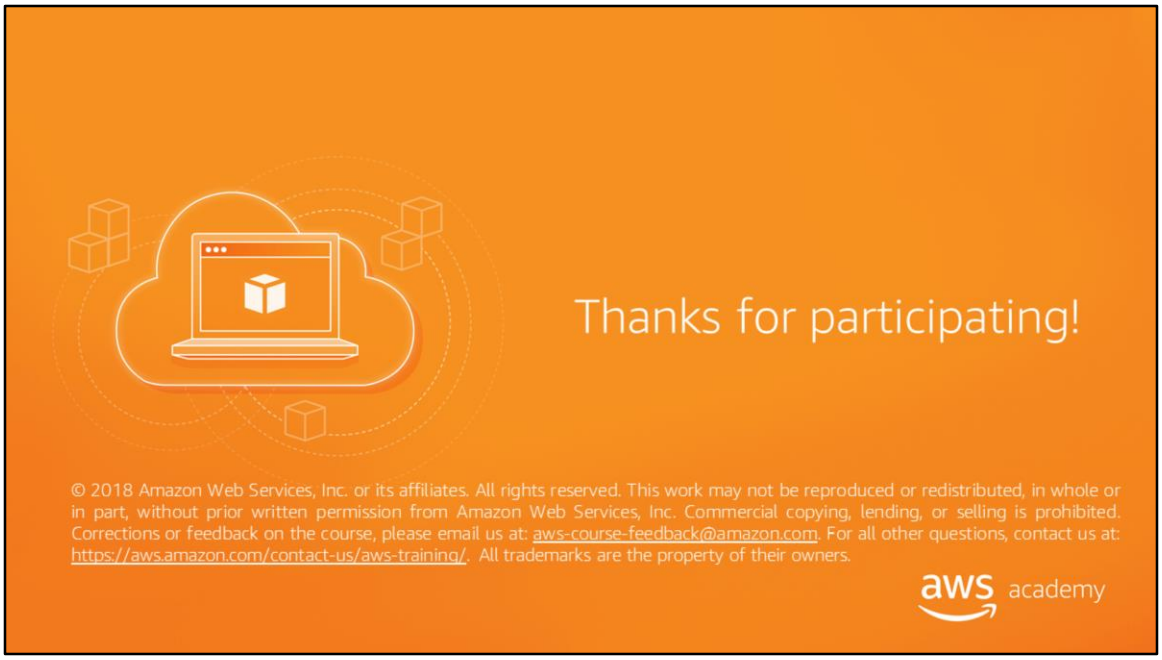
- Explored pre-created IAM Users and Groups
- Inspected IAM policies as applied to the pre-created Groups
- Followed a real-world scenario, adding Users to Groups with specific capabilities enabled
- Located and use the IAM sing-in URL
- Experimented with the effects of policies on service access



In the next module we will look at principles for architecting a cloud solution.



This slide contains attributions for any Creative Commons-licensed images used within this module.


Thanks for participating!
© 2018 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections or feedback on the course, please email us at: aws-course-feedback@amazon.com. For all other questions, contact us at: https://aws.amazon.com/contact-us/aws-training/. All trademarks are the property of their owners.
aws academy