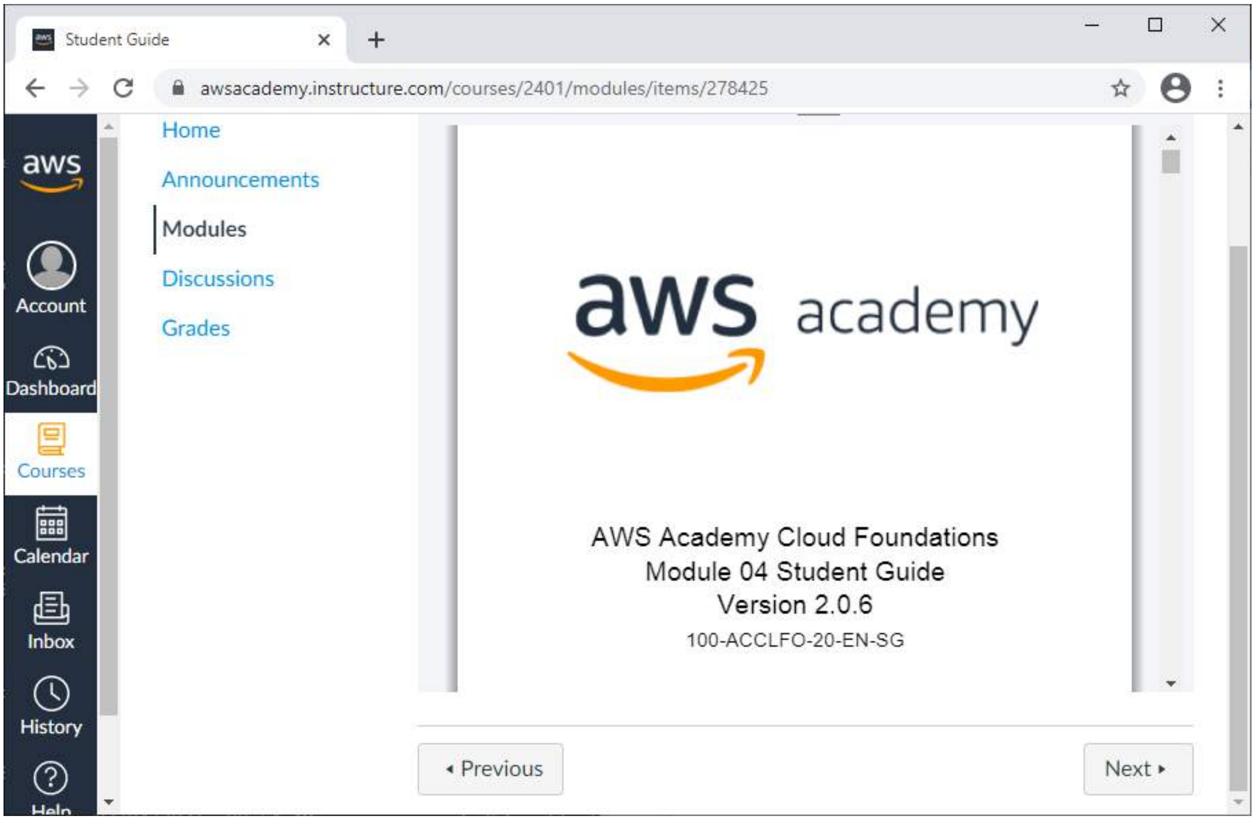


AWS Academy Cloud Foundations
Module 04 Student Guide
Version 2.0.6
100-ACCLFO-20-EN-SG

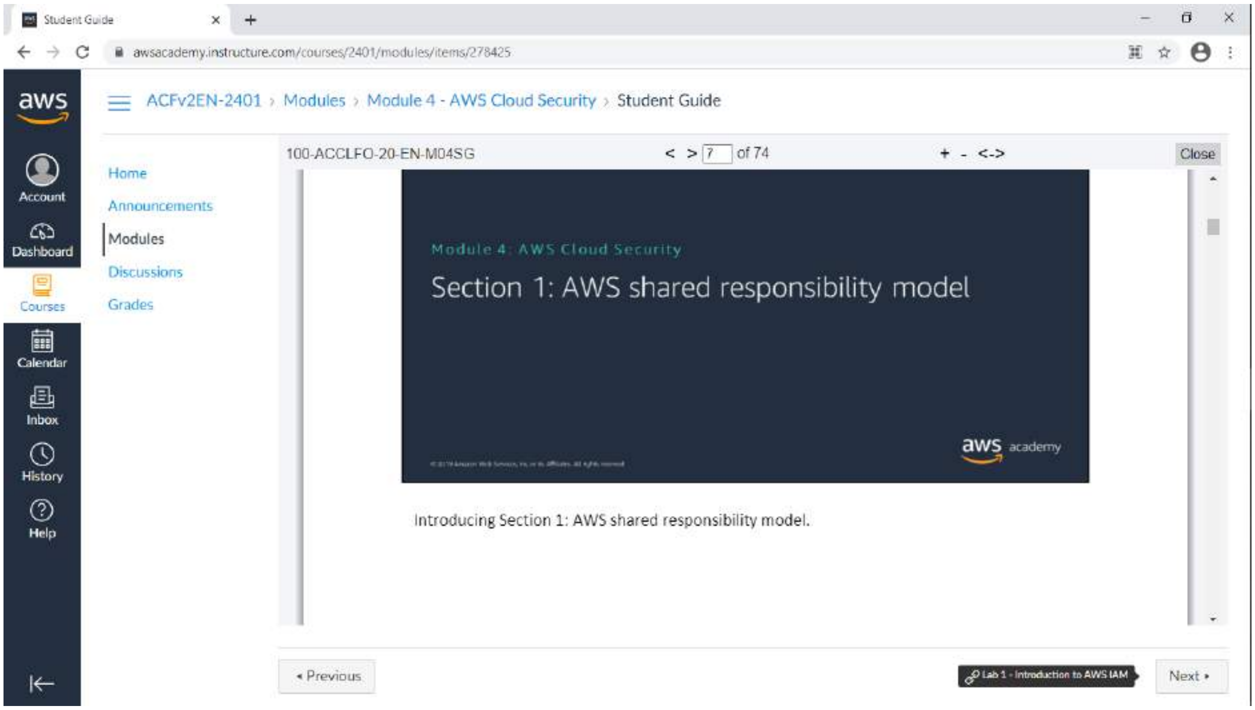
Welcome to Module 4: AWS Cloud Security.
Security is the highest priority at Amazon Web Services (AWS). AWS delivers a scalable cloud computing environment that is designed for high availability and dependability, while providing the tools that enable you to run a wide range of applications. Helping to protect the confidentiality, integrity, and availability of your systems and data is critical to AWS, and so is maintaining customer trust and confidence. This module provides an introduction to the AWS approach to security, which includes both the controls in the AWS environment and some of the AWS products and features customers can usThis module will address the following topics:

- AWS shared responsibility model
- AWS Identity and Access Management (IAM)
- Securing a new AWS account
- Securing accounts
- Securing data on AWS
- Working to ensure compliance
- Additional security services and resources

Section one includes an educator-led
Activity on the AWS shared responsibility model.
Section two includes a recorded
IAM demo, and the end of this same section there includes a hands-on lab that provides you with practice configuring IAM by using the AWS Management Console.
Finally, you will be asked to complete a
knowledge check to test your understanding of the
key concepts that are covered in this module.

After completing this module, you should be able to
:

- Recognize the shared responsibility model
- Identify the responsibility of the customer and AWS
- Recognize IAM users, groups, and roles
- Describe different types of security credentials in IAM
- Identify the steps to securing a new AWS account
- Explore IAM users and groups
- Recognize how to secure AWS data
- Recognize AWS compliance programs

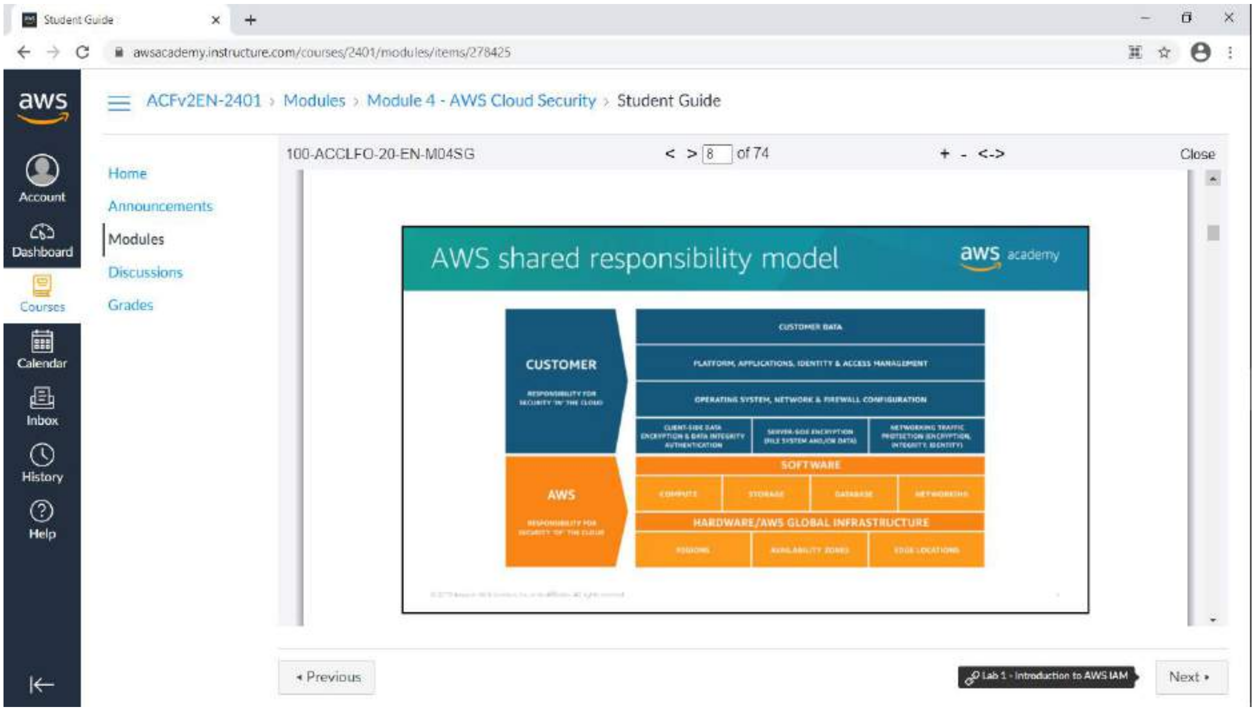


Student Guide
awsacademy.instructure.com/courses/2401/modules/items/278425
aws
Account
Dashboard
Courses
Calendar
Inbox
History
Help
ACFv2EN-2401 › Modules › Module 4 - AWS Cloud Security › Student Guide
Home
Announcements
Modules
Discussions
Grades
100-ACCLFO-20-EN-M04SG
7 of 74
Close
Module 4: AWS Cloud Security
aws academy
◂ Previous
Lab 1 - Introduction to AWS IAM
Next ▸

Section 1: AWS shared responsibility model

Introducing Section 1: AWS shared responsibility model.

Introducing Section 1: AWS shared responsibility model.



AWS shared responsibility model

CUSTOMER

RESPONSIBILITY FOR SECURITY 'IN' THE CLOUD

CUSTOMER DATA

PLATFORM, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT

OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION

CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION

SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA)

NETWORKING TRAFFIC PROTECTION (ENCRYPTION, INTEGRITY, IDENTITY)

AWS

RESPONSIBILITY FOR SECURITY 'OF' THE CLOUD

SOFTWARE

COMPUTE

STORAGE

DATABASE

NETWORKING

HARDWARE/AWS GLOBAL INFRASTRUCTURE

REGIONS

AVAILABILITY ZONES

EDGE LOCATIONS

Security and compliance are a shared responsibility between AWS and the customer. This shared responsibility model is designed to help relieve the customer's operational burden. At the same time, to provide the flexibility and customer control that enables the deployment of customer solutions on AWS, the customer remains responsible for some aspects of the overall security. The differentiation of who is responsible for what is commonly referred to as security "of" the cloud versus security "in" the cloud.

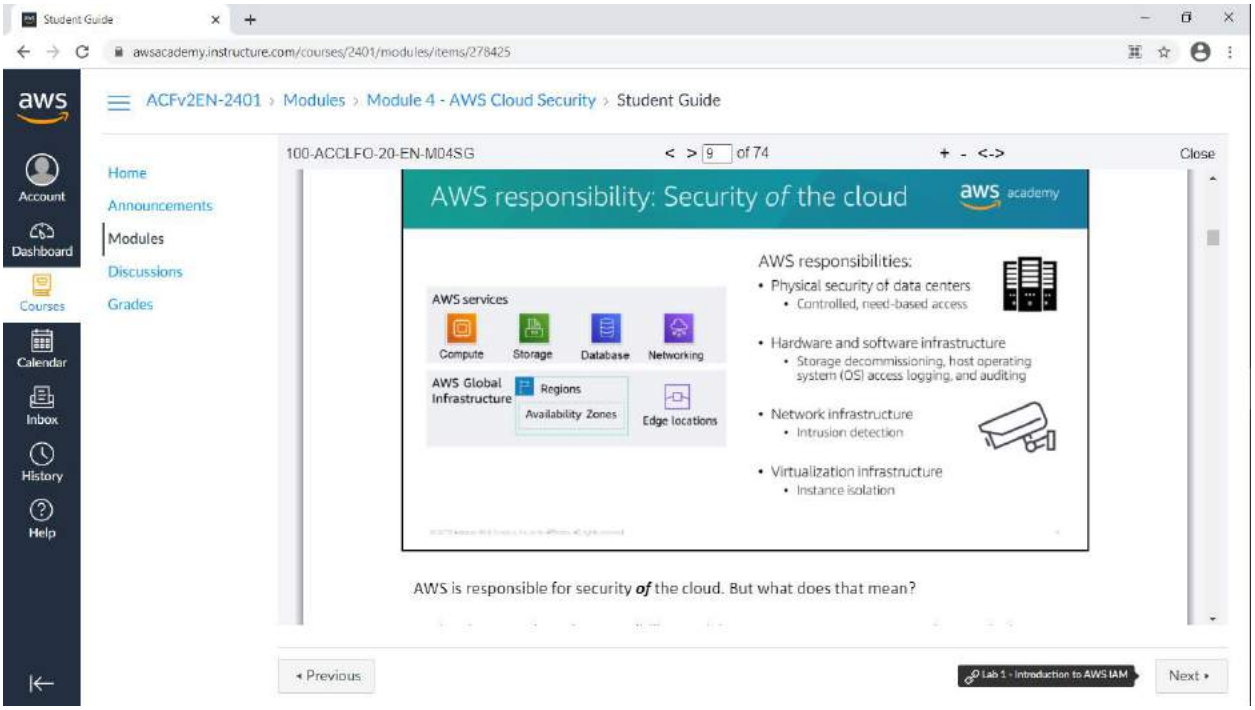
AWS operates, manages, and controls the components from the software virtualization layer down to the physical security of the facilities where AWS services operate.

AWS is responsible for protecting the infrastructure that runs all the services that are offered in the AWS Cloud. This infrastructure is composed of the hardware, software, networking, and facilities that run the AWS Cloud services.

The customer is responsible

for the encryption of data at rest and data in transit. The
customer should also ensure that the network is configured for security and that security
credentials and logins are managed safely. Additionally, the customer is responsible for the
configuration of security groups and the configuration of the operating system that run on
compute instances that they launch (including updates and security patches).

8



AWS responsibility: Security *of* the cloud

AWS responsibilities:

- Physical security of data centers
  - Controlled, need-based access
- Hardware and software infrastructure
  - Storage decommissioning, host operating system (OS) access logging, and auditing
- Network infrastructure
  - Intrusion detection
- Virtualization infrastructure
  - Instance isolation

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

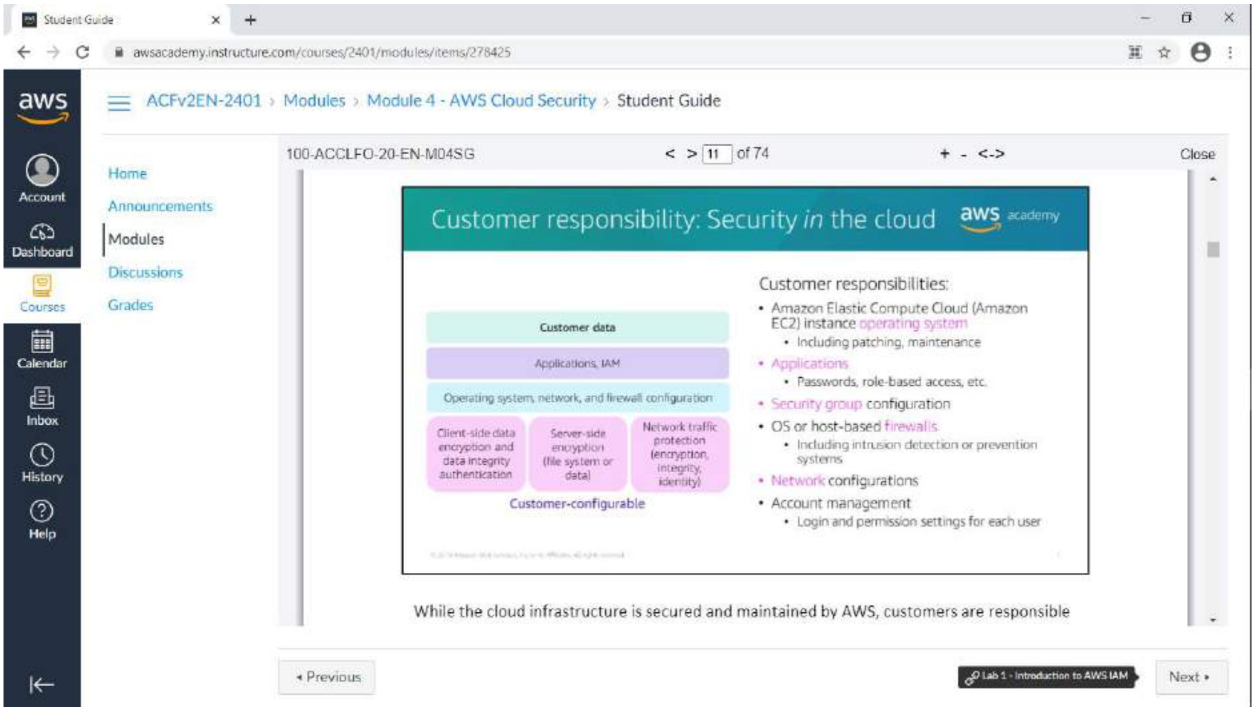
AWS is responsible for security
Of the cloud. But what does that mean?
Under the AWS shared responsibility model, AWS operates, manages, and controls the components from the bare metal host operating system and hypervisor virtualization layer
down to the physical security of the facilities where the services operate.
It means that AWS is responsible for protecting the global infrastructure that runs all the services that are offered in the AWS Cloud. The global infrastructure includes AWS Regions, Availability Zones,
and edge locations.
AWS is responsible for the physical infrastructure

that hosts your resources, including:

- Physical security of data centers with controlled, need-based access; located in nondescript facilities, with 24/7 security guards; two-factor authentication; access logging and review; video surveillance; and disk degaussing and destruction.
- Hardware infrastructure, such as servers, storage devices, and other appliances that AWS relies on.
- Software infrastructure, which hosts operating systems, service applications, and virtualization software.
- Network infrastructure, such as routers, switches, load balancers, firewalls, and cabling.

AWS also continuously monitors the network at external boundaries, secures access points, and provides redundant infrastructure with intrusion detection.

9Protecting this infrastructure is the top priority for AWS. While you cannot 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.



Customer responsibility: Security *in* the cloud

Customer data

Applications, IAM

Operating system, network, and firewall configuration

Client-side data encryption and data integrity authentication	Server-side encryption (file system or data)	Network traffic protection (encryption, integrity, identity)

Customer-configurable

Customer responsibilities:

- Amazon Elastic Compute Cloud (Amazon EC2) instance operating system
  - Including patching, maintenance
- Applications
  - Passwords, role-based access, etc.
- Security group configuration
- OS or host-based firewalls
  - Including intrusion detection or prevention systems
- Network configurations
- Account management
  - Login and permission settings for each user

While the cloud infrastructure is secured and maintained by AWS, customers are responsible

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 by using AWS services and for the applications that are connected to AWS. The security steps that you must take depend on the services that you use and the complexity of your system.

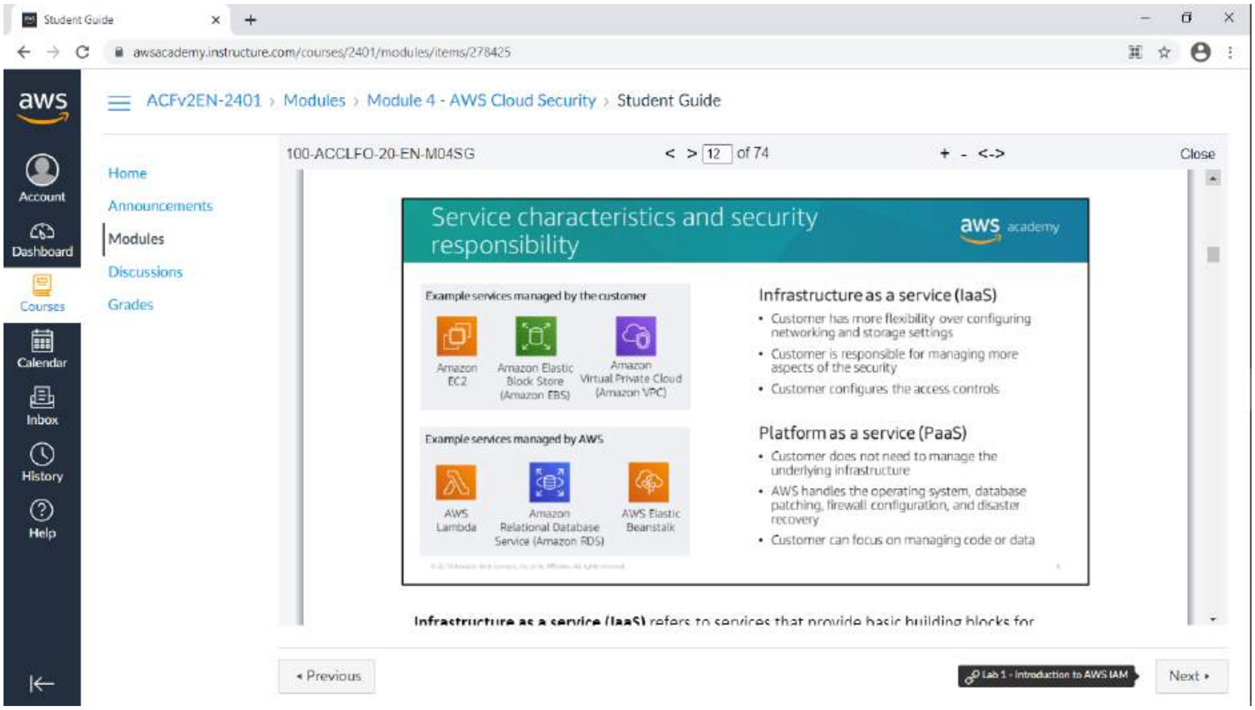
Customer responsibilities include selecting and securing any instance operating systems, securing the applications that are launched on AWS resources, security group configurations, firewall configurations, network configurations, and secure account management.

When customers use AWS services, they maintain complete control over their content.

Customers are responsible for managing critical con tent 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 configurations, and operating systems.


Student Guide
awsacademy.instructure.com/courses/2401/modules/items/278425
aws
Account
Dashboard
Courses
Calendar
Inbox
History
Help
ACFv2EN-2401 › Modules › Module 4 - AWS Cloud Security › Student Guide
Home
Announcements
Modules
Discussions
Grades
100-ACCLFO-20-EN-M04SG
< > 12 of 74
+ - <->
Close
aws academy
Amazon EC2
Amazon Elastic Block Store (Amazon EBS)
Amazon Virtual Private Cloud (Amazon VPC)
AWS Lambda
Amazon Relational Database Service (Amazon RDS)
AWS Elastic Beanstalk
◂ Previous
Lab 1 - Introduction to AWS IAM
Next ▸

Service characteristics and security responsibility

Example services managed by the customer

Example services managed by AWS

Infrastructure as a service (IaaS)

- Customer has more flexibility over configuring networking and storage settings
- Customer is responsible for managing more aspects of the security
- Customer configures the access controls

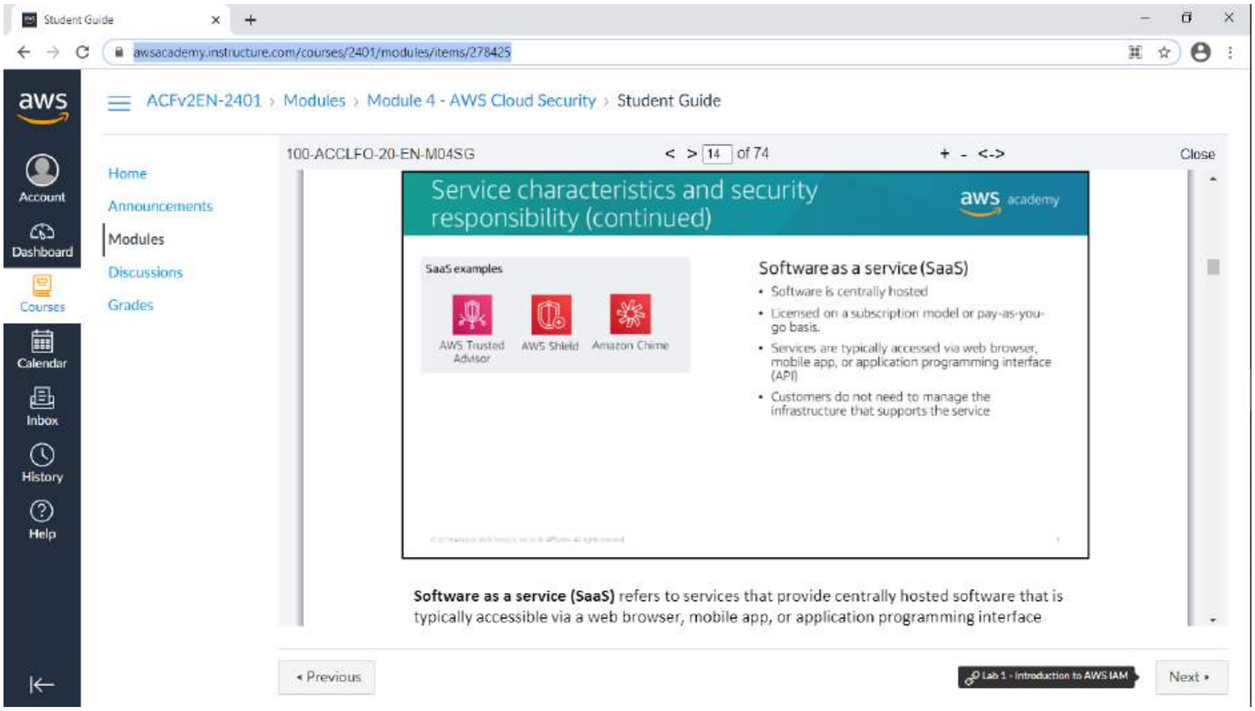
Platform as a service (PaaS)

- Customer does not need to manage the underlying infrastructure
- AWS handles the operating system, database patching, firewall configuration, and disaster recovery
- Customer can focus on managing code or data

Infrastructure as a service (IaaS) refers to services that provide basic building blocks for nfrastructure as a service (IaaS) refers to services that provide basic building blocks for cloud IT, typically including access to configure networking, computers (virtual or on dedicated hardware), and data storage space. Cloud services that can be characterized as IaaS provide the customer with the highest level of flexibility and management control over IT resources. IaaS services are most similar to existing on-premises computing resources that many IT departments are familiar with today. AWS services—such as Amazon EC2—can be categorized as IaaS and thus require the customer to perform all necessary security configuration and management tasks. Customers who deploy EC2 instances are responsible for managing the guest operating system (including updates and security patches), any application software that is installed on the instances, and the configuration of the security groups that were provided by AWS.

Platform as a service (PaaS) refers to services that remove the need for the customer to manage the underlying infrastructure (hardware, operating systems, etc.). PaaS services enable the customer to focus entirely on deploying and managing applications. Customers don’t need to worry about resource procurement, capacity planning, software maintenance, or patching.

AWS services such as AWS Lambda and Amazon RDS can be categorized as PaaS because AWS operates the infrastructure layer, the operating system, and platforms. Customers only need to access the endpoints to store and retrieve data. With PaaS services, customers are responsible for managing their data, classifying their assets, and applying the appropriate permissions. However, these service act more like managed services, with AWS handling a larger portion of the security requirements. For these services, AWS handles basic security tasks—such as operating system and database patching, firewall configuration, and disaster recovery.



Service characteristics and security responsibility (continued)

Software as a service (SaaS)

- Software is centrally hosted
- Licensed on a subscription model or pay-as-you-go basis.
- Services are typically accessed via web browser, mobile app, or application programming interface (API)
- Customers do not need to manage the infrastructure that supports the service

Software as a service (SaaS) refers to services that provide centrally hosted software that is typically accessible via a web browser, mobile app, or application programming interface

Software as a service (SaaS) refers to services that provide centrally hosted software that is typically accessible via a web browser, mobile app, or application programming interface (API). The licensing model for SaaS offerings is typically subscription or pay as you go. With SaaS offerings, customers do not need to manage the infrastructure that supports the service. Some AWS services—such as AWS Trusted Advisor, AWS Shield, and Amazon Chime—could be categorized as SaaS offerings, given their characteristics.

AWS Trusted Advisor is an online tool that analyzes your AWS environment and provides real-time guidance and recommendations to help you provision your resources by following AWS best practices. The Trusted Advisor service is offered as part of your AWS Support plan. Some of the Trusted Advisor features are free to all accounts, but Business Support and Enterprise Support customers have access to the full set of Trusted Advisor checks and recommendations.

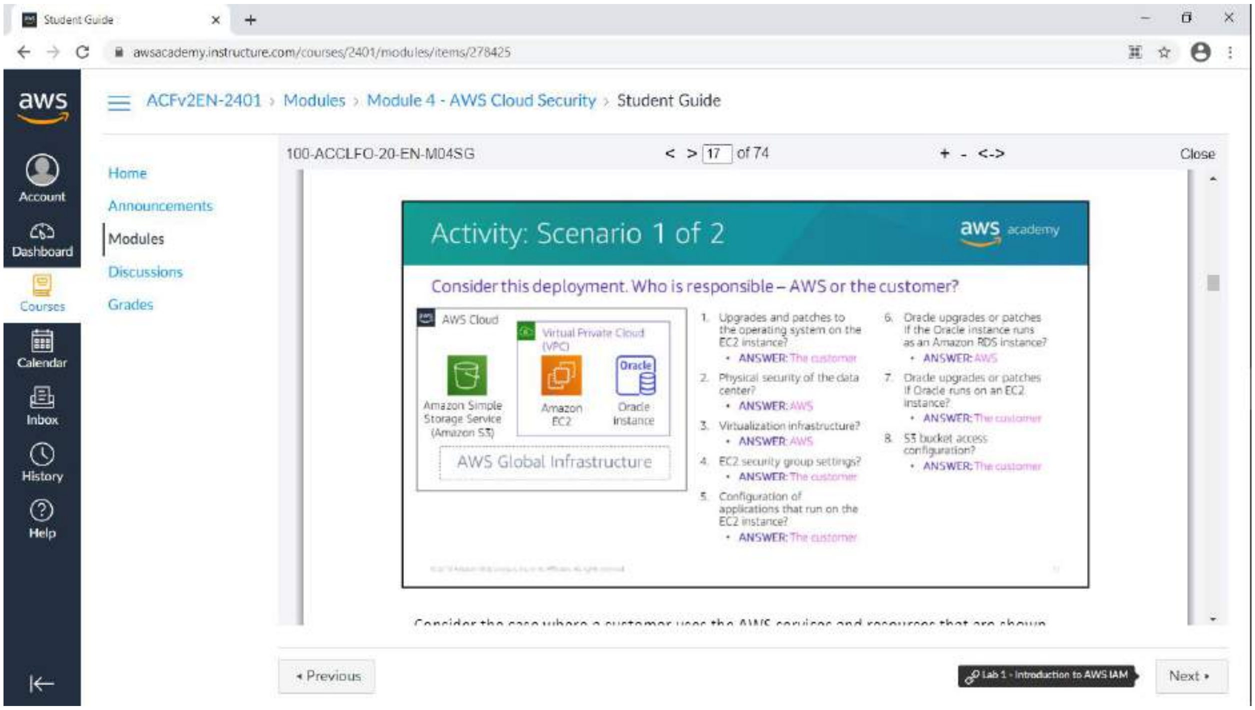
AWS Shield

is a managed distributed denial of service (DDoS) protection service that safeguards applications running on AWS. It provides always-on detection and automatic inline mitigations that minimize application downtime and latency, so there is no need to engage AWS Support to benefit from DDoS protection. AWS Shield Advanced is available to all customers. However, to contact the DDoS Response Team, customers must have either Enterprise Support or Business Support from AWS Support.

[1]Amazon Chime

is a communications service that enables you to meet, chat, and place business calls inside and outside your organization, all using a single application. It is a pay-as-you-go communications service with no upfront fees, commitments, or long-term contracts.

In this educator-led activity, you will be presented with two scenarios. For each scenario, you will be asked several questions about whose responsibility it is (AWS or the customer) to ensure security of the item in question. The educator will lead the class in a discussion of each question and reveal the correct answers one at a time.


Student Guide
awsacademy.instructure.com/courses/2401/modules/items/278425
aws
Account
Dashboard
Courses
Calendar
Inbox
History
Help
ACFv2EN-2401 › Modules › Module 4 - AWS Cloud Security › Student Guide
Home
Announcements
Modules
Discussions
Grades
100-ACCLFO-20-EN-M04SG
< > 17 of 74
+ - <->
Close
aws academy
AWS Cloud
Virtual Private Cloud (VPC)
Amazon Simple Storage Service (Amazon S3)
Amazon EC2
Oracle
Oracle instance
AWS Global Infrastructure
◂ Previous
Lab 1 - Introduction to AWS IAM
Next ▸

Activity: Scenario 1 of 2

Consider this deployment. Who is responsible – AWS or the customer?

1. Upgrades and patches to the operating system on the EC2 instance?
   - ANSWER: The customer
2. Physical security of the data center?
   - ANSWER: AWS
3. Virtualization infrastructure?
   - ANSWER: AWS
4. EC2 security group settings?
   - ANSWER: The customer
5. Configuration of applications that run on the EC2 instance?
   - ANSWER: The customer
6. Oracle upgrades or patches If the Oracle instance runs as an Amazon RDS instance?
   - ANSWER: AWS
7. Oracle upgrades or patches If Oracle runs on an EC2 instance?
   - ANSWER: The customer
8. S3 bucket access configuration?
   - ANSWER: The customer

Consider the case where a customer uses the AWS services and resources that are shown

Consider the case where a customer uses the AWS services and resources that are shown here. Who is responsible for maintaining security? AWS or the customer?

The customer uses Amazon Simple Storage Service (Amazon S3) to store data. The customer configured a virtual private cloud (VPC) with Amazon Virtual Private Cloud (Amazon VPC). The EC2 instance and the Oracle database instance that they created both run in the VPC.

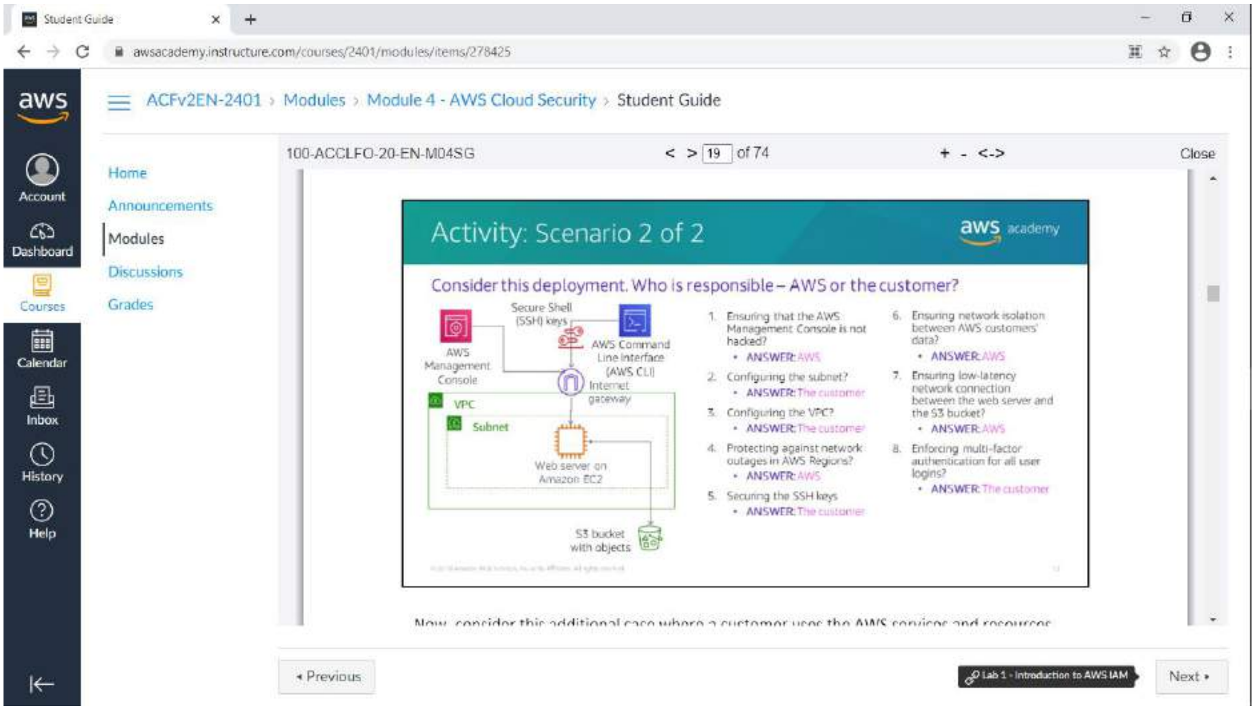
In this example, the customer must manage the guest operating system (OS) that runs on the EC2 instance. Over time, the guest OS will need to be upgraded and have security patches applied. Additionally, any application software or utilities that the customer installed on the Amazon EC2 instance must also be maintained. The customer is responsible for configuring the AWS firewall (or security group) that is applied to the Amazon EC2 instance. The customer is also responsible for the VPC configurations that specify the network conditions in which the Amazon EC2 instance runs. These tasks are the same security tasks that IT staff would perform, no matter where their servers are located.

The Oracle instance in this example provides an interesting case study in terms of AWS or customer responsibility.

If the database runs on an EC2 instance, then it is the customer's responsibility to apply Oracle software upgrades and patches. However, if the database runs as an Amazon RDS instance, then it is the responsibility of AWS to apply Oracle software upgrades and patches. Because Amazon RDS is a managed database offering, time-

consuming database administration tasks—which include provisioning, backups, software patching, monitoring, and hardware scaling—are handled by AWS. To learn more, see Best Practices for Running Oracle Database on AWS
for details.

18



Activity: Scenario 2 of 2

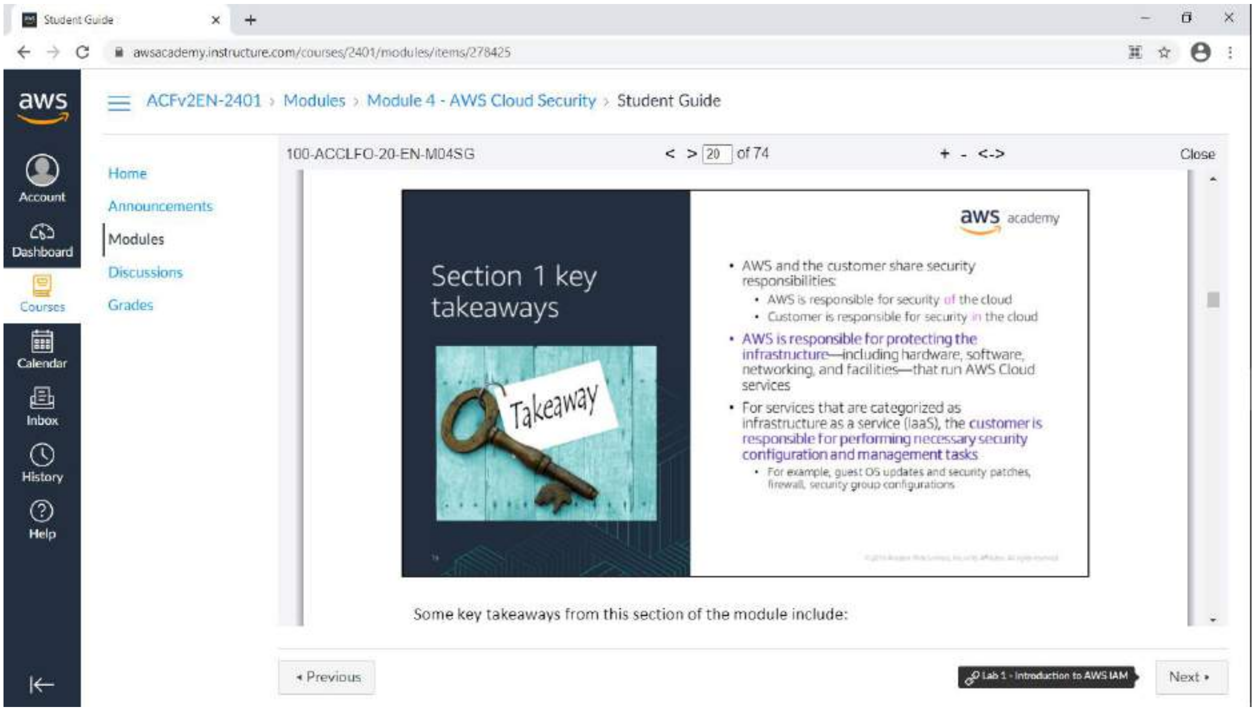
Consider this deployment. Who is responsible – AWS or the customer?

1. Ensuring that the AWS Management Console is not hacked?
   - ANSWER: AWS
2. Configuring the subnet?
   - ANSWER: The customer
3. Configuring the VPC?
   - ANSWER: The customer
4. Protecting against network outages in AWS Regions?
   - ANSWER: AWS
5. Securing the SSH keys
   - ANSWER: The customer
6. Ensuring network isolation between AWS customers' data?
   - ANSWER: AWS
7. Ensuring low-latency network connection between the web server and the S3 bucket?
   - ANSWER: AWS
8. Enforcing multi-factor authentication for all user logins?
   - ANSWER: The customer

Now, consider this additional case where a customer uses the AWS services and resources

Now, consider this additional case where a customer uses the AWS services and resources that are shown here. Who is responsible for maintaining security? AWS or the customer?

A customer uses Amazon S3 to store data. The customer configured a virtual private cloud (VPC) with Amazon VPC, and is running a web server on an EC2 instance in the VPC. The customer configured an internet gateway as part of the VPC so that the web server can be reached by using the AWS Management Console or the AWS Command Line Interface (AWS CLI). When the customer uses the AWS CLI, the connection requires the use of Secure Shell (SSH) keys.



ACFv2EN-2401 › Modules › Module 4 - AWS Cloud Security › Student Guide

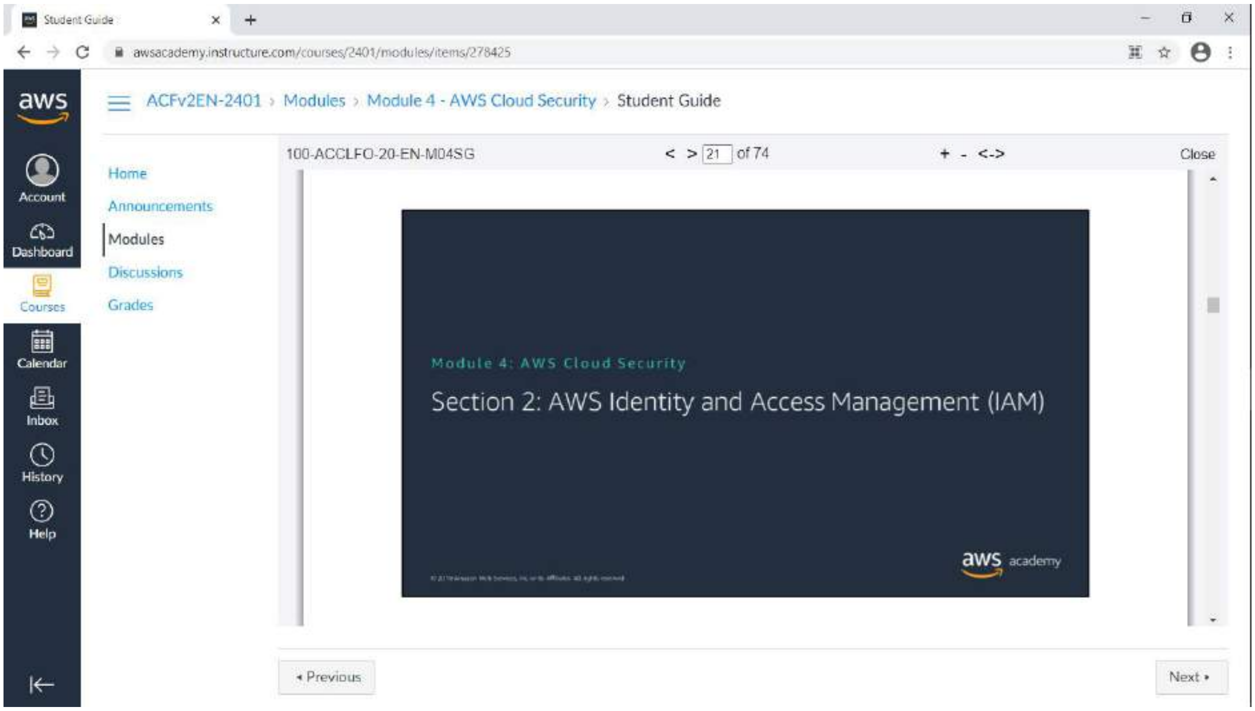
Section 1 key takeaways

- AWS and the customer share security responsibilities:
  - AWS is responsible for security of the cloud
  - Customer is responsible for security in the cloud
- AWS is responsible for protecting the infrastructure—including hardware, software, networking, and facilities—that run AWS Cloud services
- For services that are categorized as infrastructure as a service (IaaS), the customer is responsible for performing necessary security configuration and management tasks
  - For example, guest OS updates and security patches, firewall, security group configurations

Some key takeaways from this section of the module include:

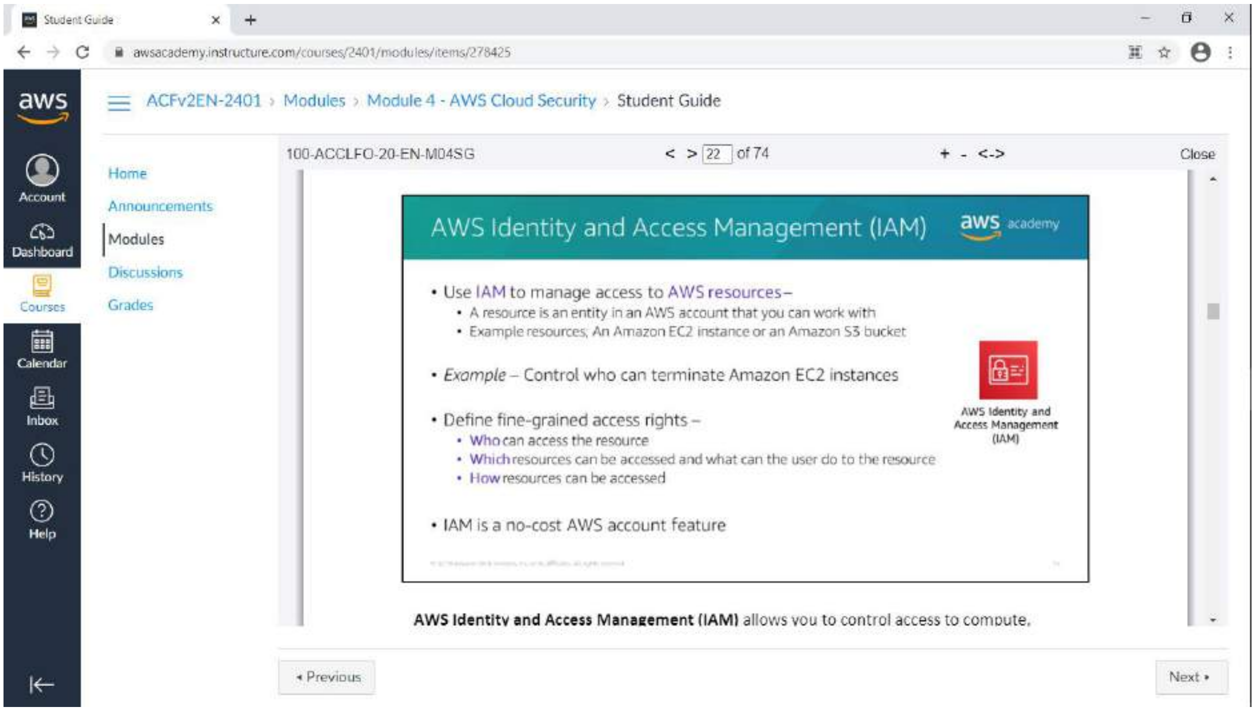
Some key takeaways from this section of the module include:

- AWS and the customer share security responsibilities –
- AWS is responsible for security *of* the cloud
- Customer is responsible for security *in* the cloud
- AWS is responsible for protecting the infrastructure —including hardware, software, networking, and facilities—that run AWS Cloud services
- For services that are categorized as infrastructure as a service (IaaS), the customer is responsible for performing necessary security configuration and management tasks
- For example, guest OS updates and security patches, firewall, security group configurations



Module 4: AWS Cloud Security

Section 2: AWS Identity and Access Management (IAM)


Student Guide
awsacademy.instructure.com/courses/2401/modules/items/278425
aws
Account
Dashboard
Courses
Calendar
Inbox
History
Help
ACFv2EN-2401 › Modules › Module 4 - AWS Cloud Security › Student Guide
Home
Announcements
Modules
Discussions
Grades
100-ACCLFO-20-EN-M04SG
22 of 74
Close
aws academy
AWS Identity and Access Management (IAM)
◂ Previous
Next ▸

AWS Identity and Access Management (IAM)

- Use IAM to manage access to AWS resources–
  - A resource is an entity in an AWS account that you can work with
  - Example resources; An Amazon EC2 instance or an Amazon S3 bucket
- *Example* – Control who can terminate Amazon EC2 instances
- Define fine-grained access rights –
  - Who can access the resource
  - Which resources can be accessed and what can the user do to the resource
  - How resources can be accessed
- IAM is a no-cost AWS account feature

AWS Identity and Access Management (IAM) allows you to control access to compute,

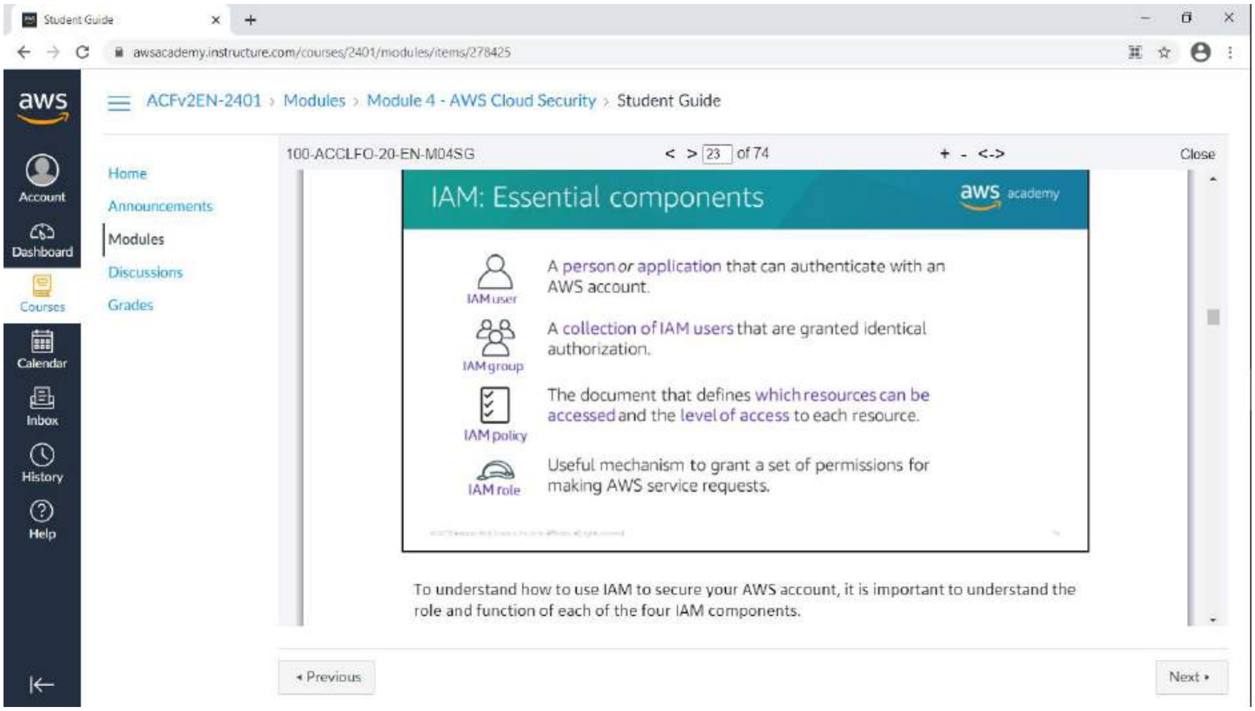
AWS Identity and Access Management (IAM) allows you to control access to compute, storage, database, and application services in the AWS Cloud. IAM can be used to handle authentication, and to specify and enforce authorization policies so that you can specify which users can access which services.

IAM is a tool that centrally manages access to launching, configuring, managing, and terminating resources in your AWS account. It provides granular control over access to resources, including the ability to specify exactly which API calls the user is authorized to make to each service. Whether you use the AWS Management Console, the AWS CLI, or the AWS software development kits (SDKs), every call to an AWS service is an API call.

With IAM, you can manage which resources can be accessed by who

, and how these resources can be accessed. You can grant different permissions to different people for different resources. For example, you might allow some users full access to Amazon EC2, Amazon S3, Amazon DynamoDB, Amazon Redshift, and other AWS services. However, for other users, you might allow read-only access to only a few S3 buckets. Similarly, you might grant permission to other users to administer only specific EC2 instances. You could also allow a few users to access only the account billing information, but nothing else.

IAM is a feature of your AWS account, and it is offered at no additional charge.



IAM: Essential components

- IAM user: A person or application that can authenticate with an AWS account.
- IAM group: A collection of IAM users that are granted identical authorization.
- IAM policy: The document that defines which resources can be accessed and the level of access to each resource.
- IAM role: Useful mechanism to grant a set of permissions for making AWS service requests.

To understand how to use IAM to secure your AWS account, it is important to understand the role and function of each of the four IAM components.

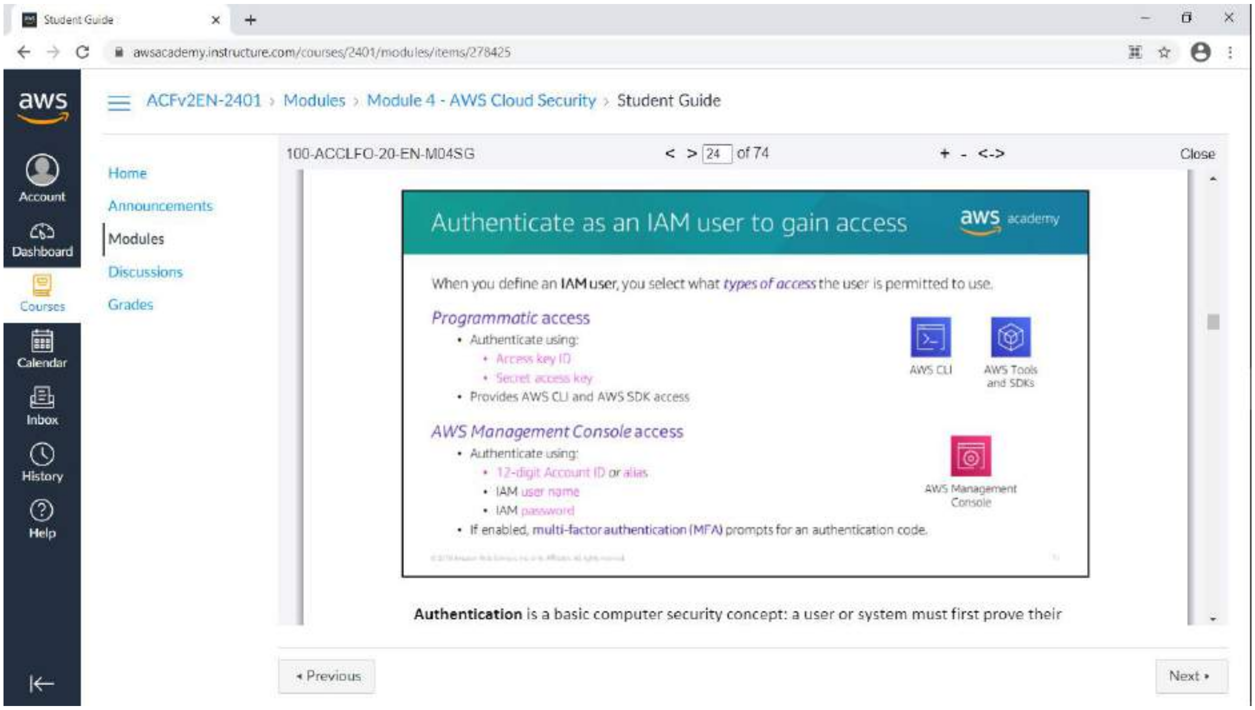
To understand how to use IAM to secure your AWS account, it is important to understand the role and function of each of the four IAM components.

An IAM user is a person or application that is defined in an AWS account, and that must make API calls to AWS products. Each user must have a unique name (with no spaces in the name) within the AWS account, and a set of security credentials that is not shared with other users. These credentials are different from the AWS account root user security credentials. Each user is defined in one and only one AWS account.

An IAM group is a collection of IAM users. You can use IAM groups to simplify specifying and managing permissions for multiple users.

An IAM policy

is a document that defines permissions to determine
what users can do in the
AWS account. A policy typically grants access to specific resources and specifies what the user
can do with those resources. Policies can also explicitly deny access.
An
IAM role
is a tool for granting temporary access to specific
AWS resources in an AWS account.



Authenticate as an IAM user to gain access

When you define an **IAM user**, you select what *types of access* the user is permitted to use.

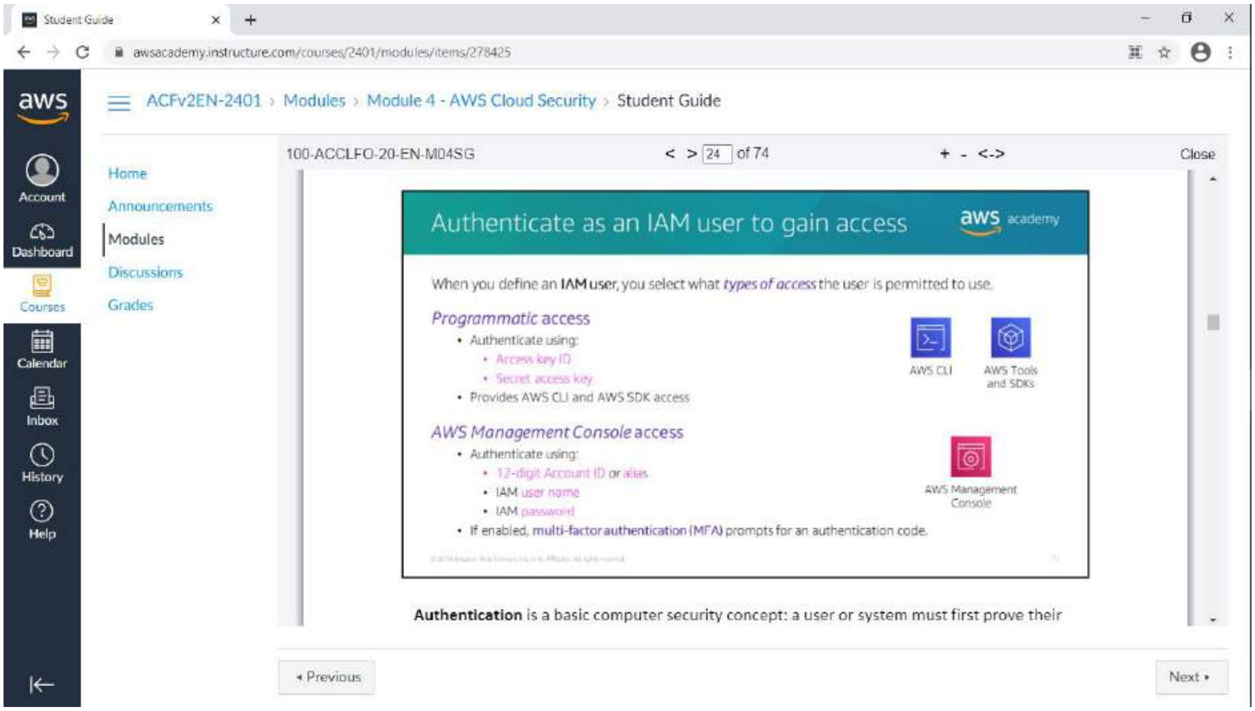
Programmatic access

- Authenticate using:
  - Access key ID
  - Secret access key
- Provides AWS CLI and AWS SDK access

AWS Management Console access

- Authenticate using:
  - 12-digit Account ID or alias
  - IAM user name
  - IAM password
- If enabled, multi-factor authentication (MFA) prompts for an authentication code.

Authentication is a basic computer security concept: a user or system must first prove their



Authenticate as an IAM user to gain access

When you define an **IAM user**, you select what *types of access* the user is permitted to use.

Programmatic access
- Authenticate using:
  - Access key ID
  - Secret access key
- Provides AWS CLI and AWS SDK access

AWS Management Console access
- Authenticate using:
  - 12-digit Account ID or alias
  - IAM user name
  - IAM password
- If enabled, multi-factor authentication (MFA) prompts for an authentication code.

Authentication is a basic computer security concept: a user or system must first prove their

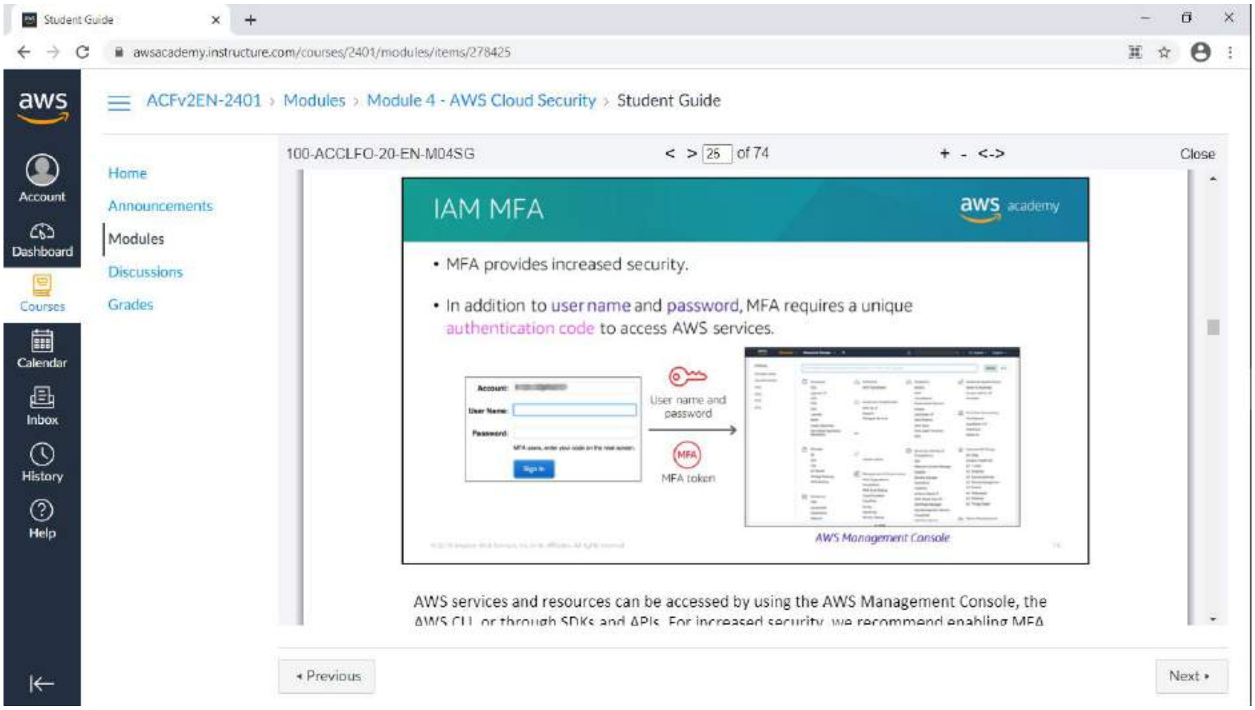
Authentication is a basic computer security concept: a user or system must first prove their identity. Consider how you authenticate yourself when you go to the airport and you want to get through airport security so that you can catch your flight. In this situation, you must present some form of identification to the security official to prove who you are before you can enter a restricted area. A similar concept applies for gaining access to AWS resources in the cloud.

When you define an IAM user, you select what type of access the user is permitted to use to access AWS resources. You can assign two different types of access to users: programmatic access and AWS Management Console access. You can assign programmatic access only, console access only, or you can assign both types of access.

If you grant programmatic access

, the IAM user will be required to present an access key ID and a secret access key when they make an AWS API call by using the AWS CLI, the AWS SDK, or some other development tool. If you grant AWS Management Console access, the IAM user will be required to fill in the fields that appear in the browser login window. The user is prompted to provide either the 12-digit account ID or the corresponding account alias. The user must also enter their IAM user name and password. If multi-factor authentication (MFA) is enabled for the user, they will also be prompted for an authentication code.

24


Student Guide

awsacademy.instructure.com/courses/2401/modules/items/278425

ACFv2EN-2401 › Modules › Module 4 - AWS Cloud Security › Student Guide

100-ACCLFO-20-EN-M04SG < > 25 of 74 + - <-> Close

IAM MFA

- MFA provides increased security.
- In addition to user name and password, MFA requires a unique authentication code to access AWS services.

Account:

User Name:

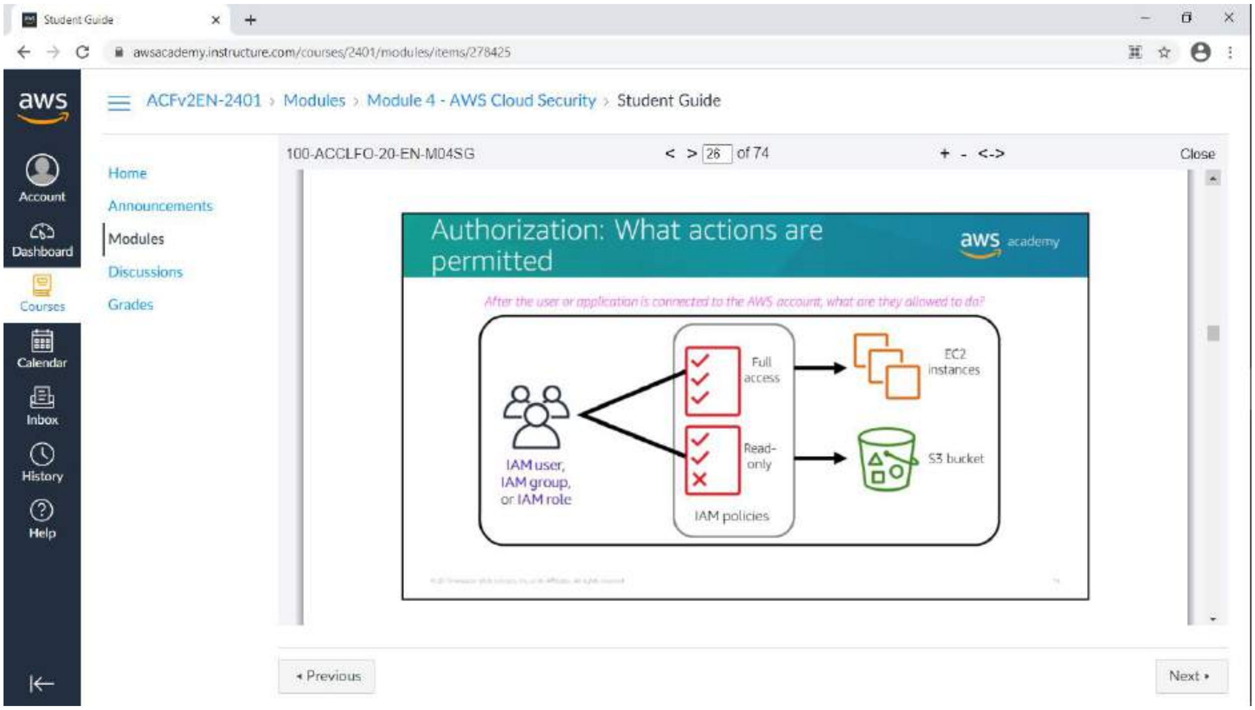
Password:

MFA users, enter your code on the next screen.

Sign In

AWS services and resources can be accessed by using the AWS Management Console, the AWS CLI, or through SDKs and APIs. For increased security, we recommend enabling MFA

AWS services and resources can be accessed by using the AWS Management Console, the AWS CLI, or through SDKs and APIs. For increased security, we recommend enabling MFA. With MFA, users and systems must provide an MFA token—in addition to the regular sign-in credentials—before they can access AWS services and resources. Options for generating the MFA authentication token include virtual MFA-compliant applications (such as Google Authenticator or Authy 2-Factor Authentication), U2F security key devices, and hardware MFA devices



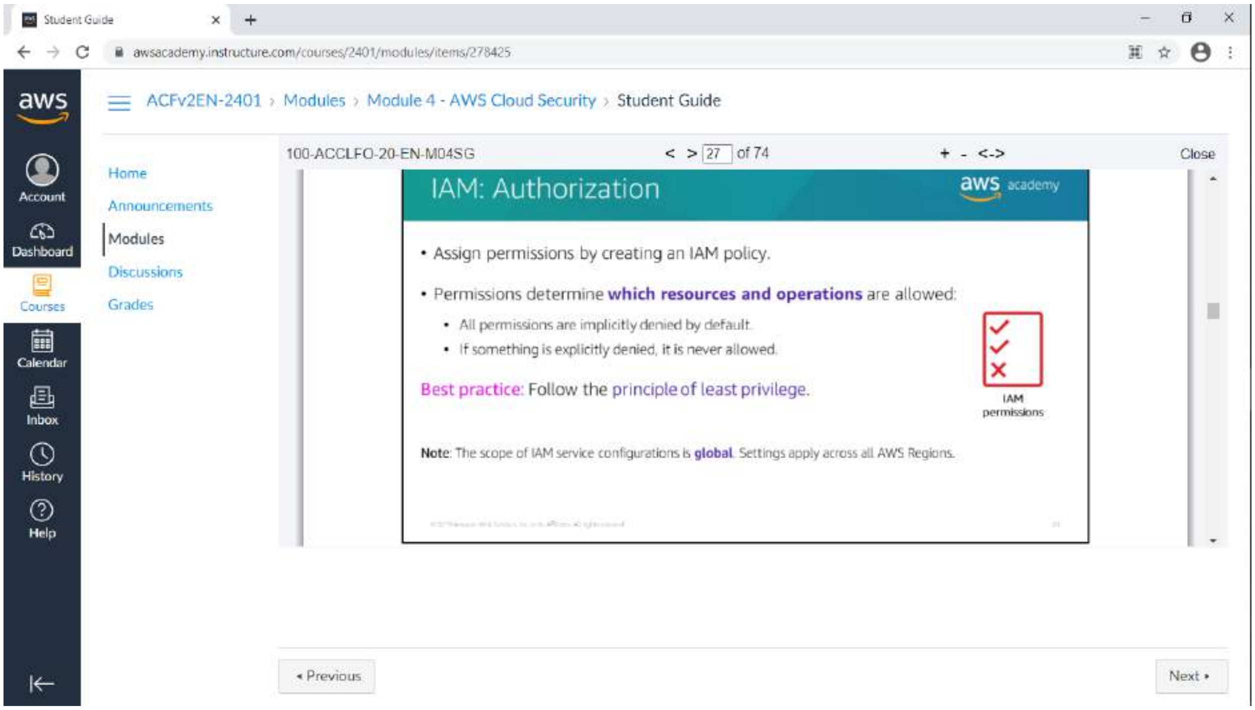
Authorization: What actions are permitted

After the user or application is connected to the AWS account, what are they allowed to do?

Authorization is the process of determining what permissions a user, service or application should be granted. After a user has been authenticated, they must be authorized to access

AWS services.

By default, IAM users do not have permissions to access any resources or data in an AWS account. Instead, you must explicitly grant permissions to a user, group, or role by creating a policy, which is a document in JavaScript Object Notation (JSON) format. A policy lists permissions that allow or deny access to resources in the AWS account



IAM: Authorization

- Assign permissions by creating an IAM policy.
- Permissions determine **which resources and operations** are allowed:
  - All permissions are implicitly denied by default.
  - If something is explicitly denied, it is never allowed.

Best practice: Follow the principle of least privilege.

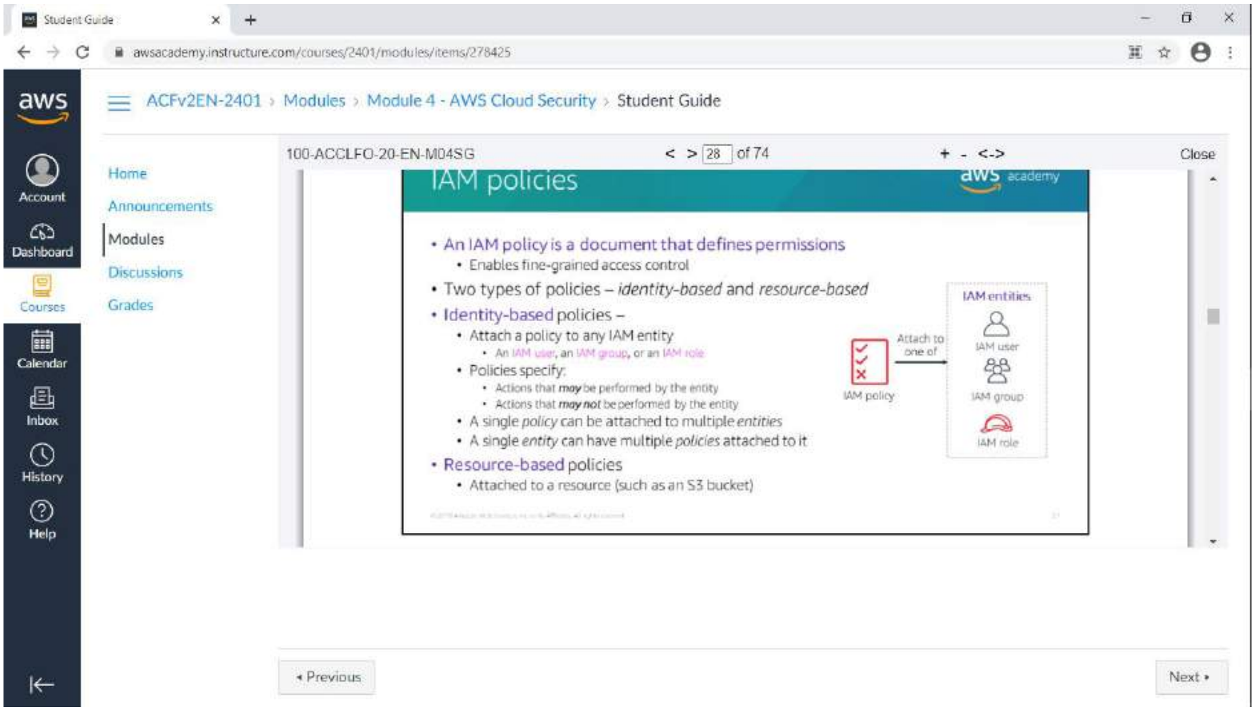
Note: The scope of IAM service configurations is **global**. Settings apply across all AWS Regions.

To assign permission to a user, group or role, you must create an IAM policy (or find an existing policy in the account). There are no default permissions. All actions in the account are denied to the user by default (implicit deny) unless those actions are explicitly allowed.

Any actions that you do not 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. It promotes that you grant only the minimal user privileges needed to the user, based on the needs of your users. When you create IAM policies, it is a best practice to follow this security advice of granting least privilege . Determine what users need to be able 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 broad and then later trying to lock down the permissions granted.

Note that the scope of the IAM service configurations is global . The settings are not defined at an AWS Region level. IAM settings apply across all AWS Regions.



IAM policies

- An IAM policy is a document that defines permissions
  - Enables fine-grained access control
- Two types of policies – *identity-based* and *resource-based*
- Identity-based policies –
  - Attach a policy to any IAM entity
    - An IAM user, an IAM group, or an IAM role
  - Policies specify:
    - Actions that ***may*** be performed by the entity
    - Actions that ***may not*** be performed by the entity
  - A single *policy* can be attached to multiple *entities*
  - A single *entity* can have multiple *policies* attached to it
- Resource-based policies
  - Attached to a resource (such as an S3 bucket)

An IAM policy is a formal statement of permissions that will be granted to an entity. Policies can be attached to any IAM entity. Entities include users, groups, roles, or resources. For example, you can attach a policy to AWS resources that will block all requests that do not come from an approved Internet Protocol (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 applies.

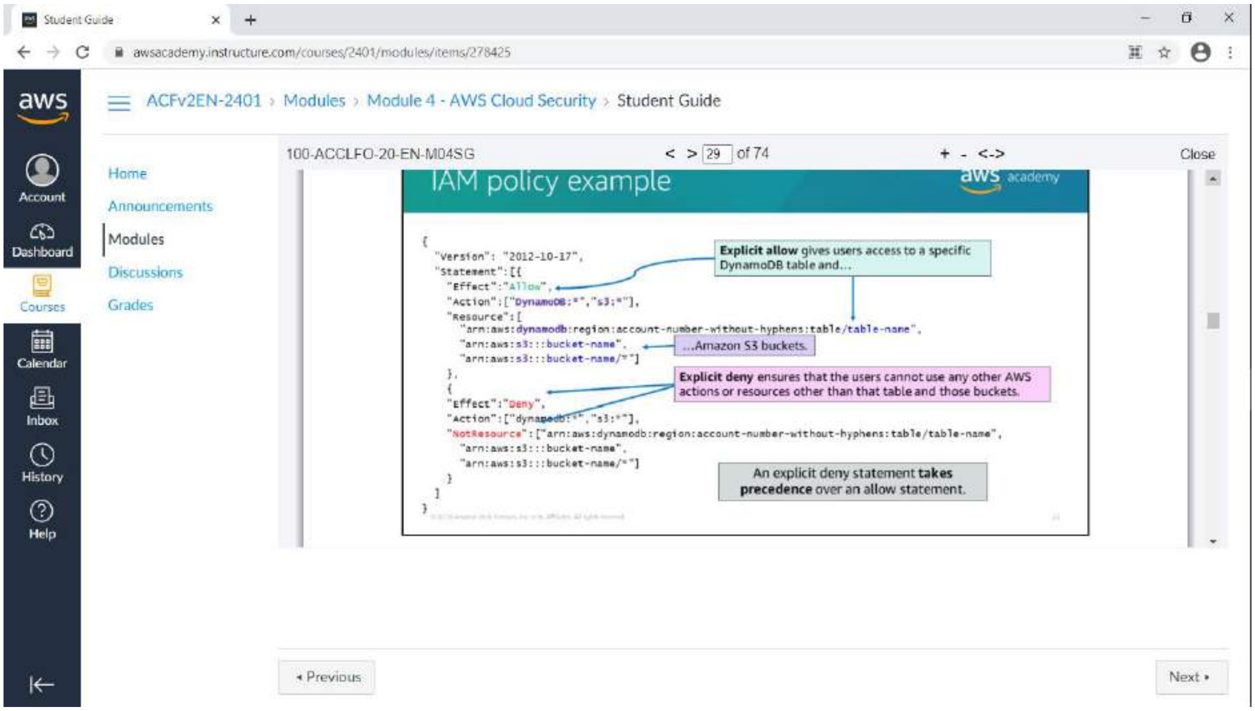
There are two types of IAM policies.

Identity-based policies are permissions 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 as:

- Managed policies –
Standalone identity-based policies that you can attach 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 S3 bucket. These policies control what actions a specified principal can perform on that resource, and under what conditions.



IAM policy example

```
{
  "Version": "2012-10-17",
  "Statement":[{
    "Effect":"Allow",
    "Action":["DynamoDB:*","s3:*"],
    "Resource":[
      "arn:aws:dynamodb:region:account-number-without-hyphens:table/table-name",
      "arn:aws:s3:::bucket-name",
      "arn:aws:s3:::bucket-name/*"]
    },
    {
    "Effect":"Deny",
    "Action":["dynamodb:*","s3:*"],
    "NotResource":["arn:aws:dynamodb:region:account-number-without-hyphens:table/table-name",
      "arn:aws:s3:::bucket-name",
      "arn:aws:s3:::bucket-name/*"]
    }
  ]
}
```

Explicit allow gives users access to a specific DynamoDB table and…

…Amazon S3 buckets.

Explicit deny ensures that the users cannot use any other AWS actions or resources other than that table and those buckets.

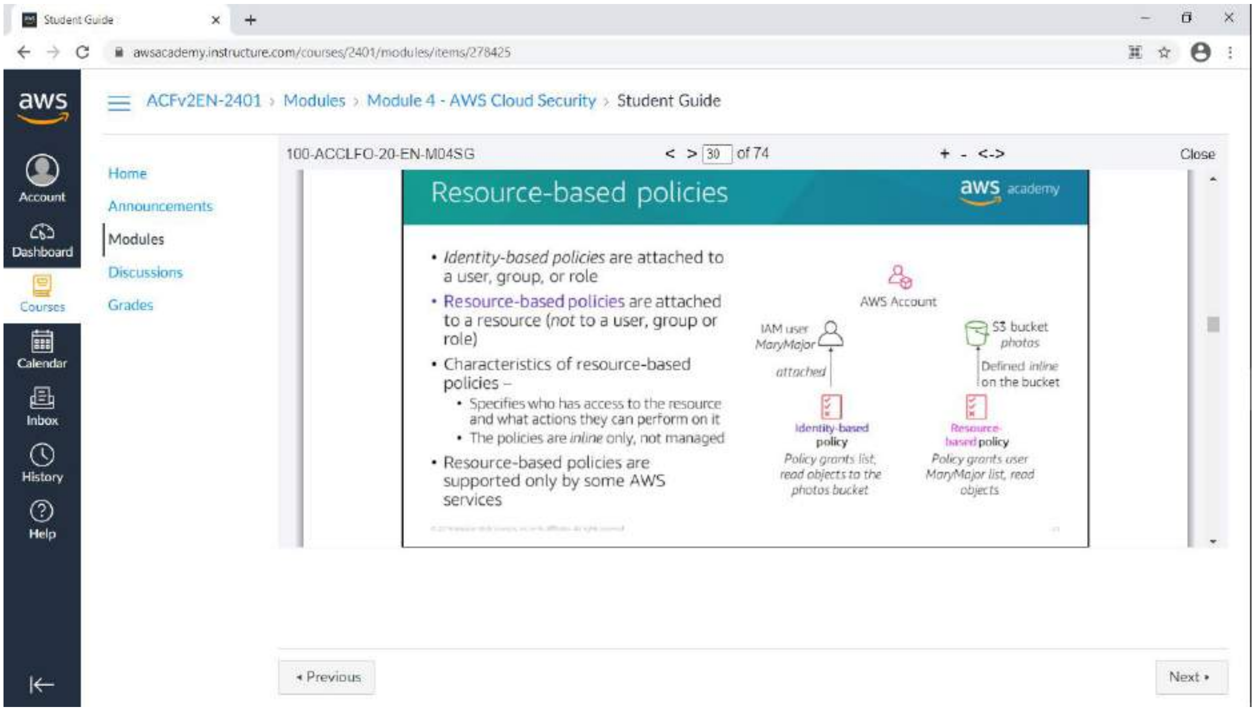
An explicit deny statement **takes precedence** over an allow statement.

As mentioned previously, IAM policy documents are written in JSON.

The example IAM policy grants users access only to the following resources:

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

The IAM policy also includes an explicit deny ("Effect":"Deny") element. The NotResource element helps to ensure that users cannot use any other DynamoDB or S3 actions or resources except the actions and resources that are specified in the policy—even if permissions have been granted in another policy. An explicit deny statement takes precedence over an allow statement.



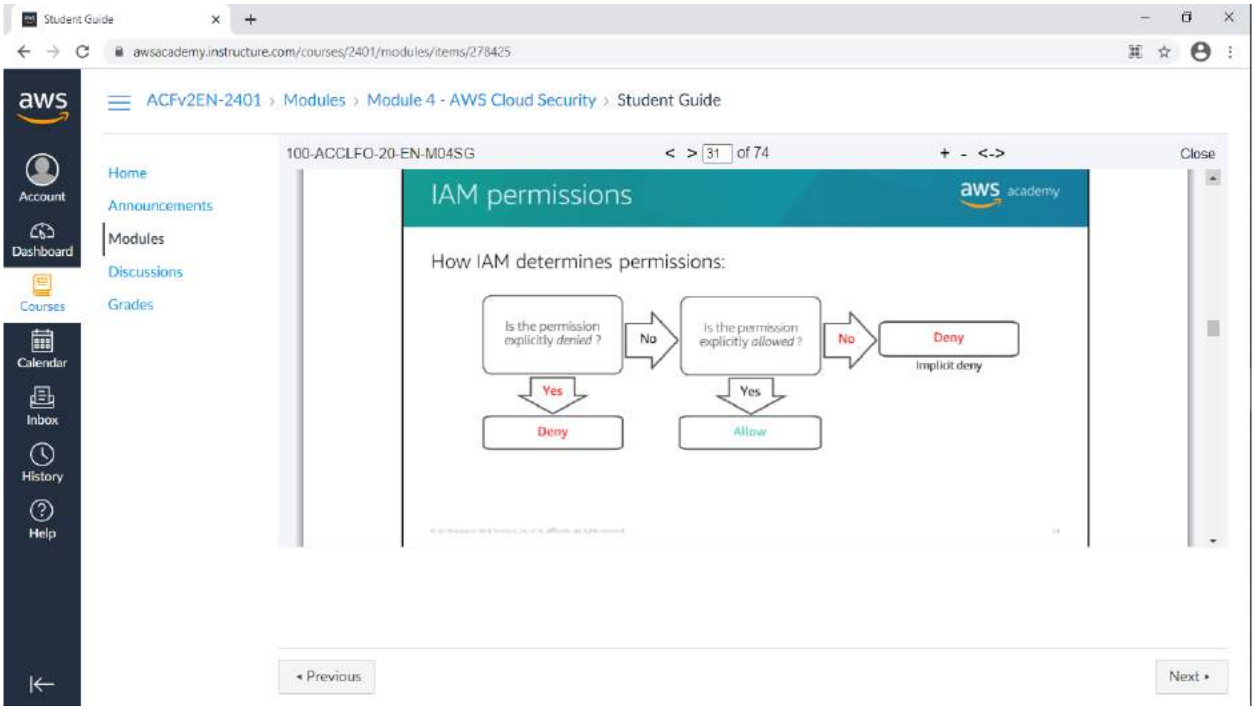
Resource-based policies

- *Identity-based policies* are attached to a user, group, or role
- Resource-based policies are attached to a resource (*not* to a user, group or role)
- Characteristics of resource-based policies –
  - Specifies who has access to the resource and what actions they can perform on it
  - The policies are *inline* only, not managed
- Resource-based policies are supported only by some AWS services

While identity-based policies are attached to a user, group, or role, resource-based policies are attached to a resource, such as an S3 bucket. These policies specify who can access the resource and what actions they can perform on it. Resource-based policies are defined inline only, which means that you define the policy on the resource itself, instead of creating a separate IAM policy document that you attach. For example, to create an S3 bucket policy (a type of resource-based policy) on an S3 bucket, navigate to the bucket, click the Permissions tab, click the Bucket Policy button, and define the JSON-formatted policy document there. An Amazon S3 access control list (ACL) is another example of a resource-based policy.

The diagram shows two different ways that the user MaryMajor could be granted access to objects in the S3 bucket that is named photos. On the left, you see an example of an identity-based policy. An IAM policy that grants access to the S3 bucket is attached to the MaryMajor user. On the right, you see an example of a resource-based policy. The S3 bucket policy for the photos bucket specifies that the user MaryMajor is allowed to list and read the objects in the bucket.

Note that you could define a deny statement in a bucket policy to restrict access to specific IAM users, even if the users are granted access in a separate identity-based policy. An explicit deny statement will always take precedence over any allow statement.



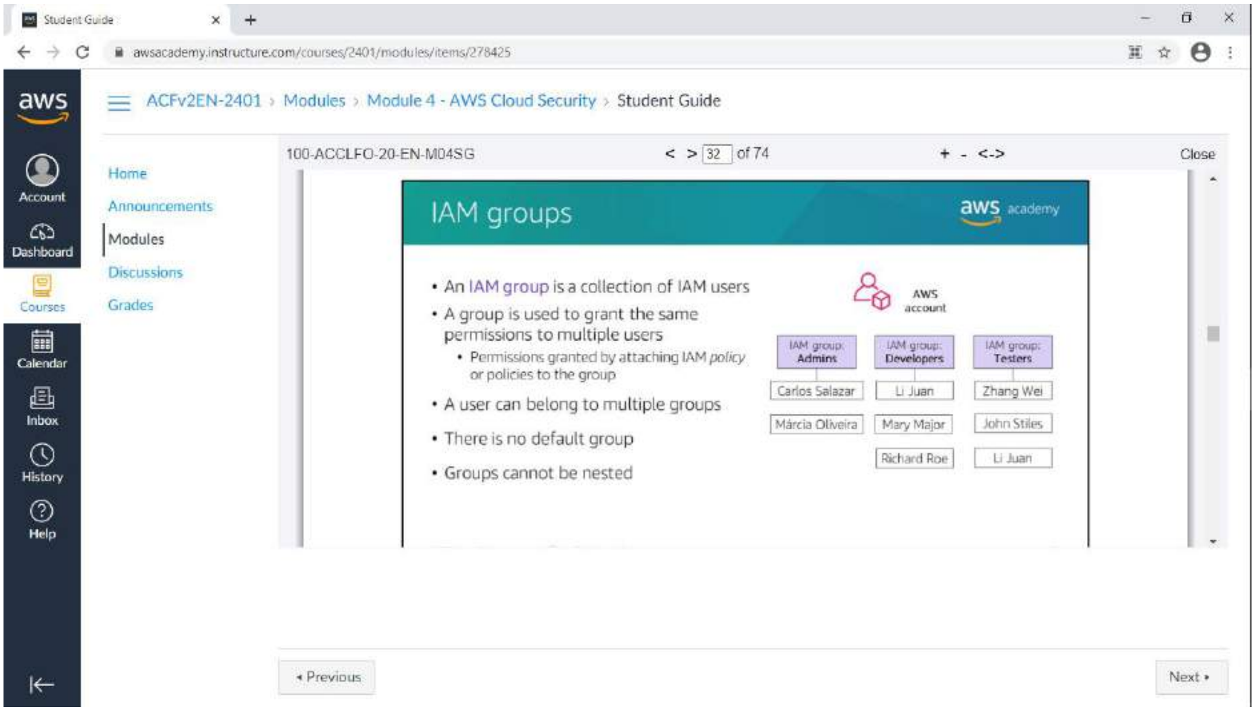
IAM permissions

How IAM determines permissions:

IAM policies enable you to fine-tune privileges that are granted to IAM users, groups, and roles.

When IAM determines whether a permission is allowed, IAM first checks for the existence of any applicable explicit denial policy. If no explicit denial exists, it then checks for any applicable explicit allow policy. If neither an explicit deny nor an explicit allow policy exists, IAM reverts to the default, which is to deny access. This process is referred to as an implicit deny. The user will be permitted to take the action only if the requested action is not explicitly denied and is explicitly allowed.

It can be difficult to figure out whether access to a resource will be granted to an IAM entity when you develop IAM policies. The IAM Policy Simulator is a useful tool for testing and troubleshooting IAM policies.



IAM groups

- An IAM group is a collection of IAM users
- A group is used to grant the same permissions to multiple users
  - Permissions granted by attaching IAM *policy* or policies to the group
- A user can belong to multiple groups
- There is no default group
- Groups cannot be nested

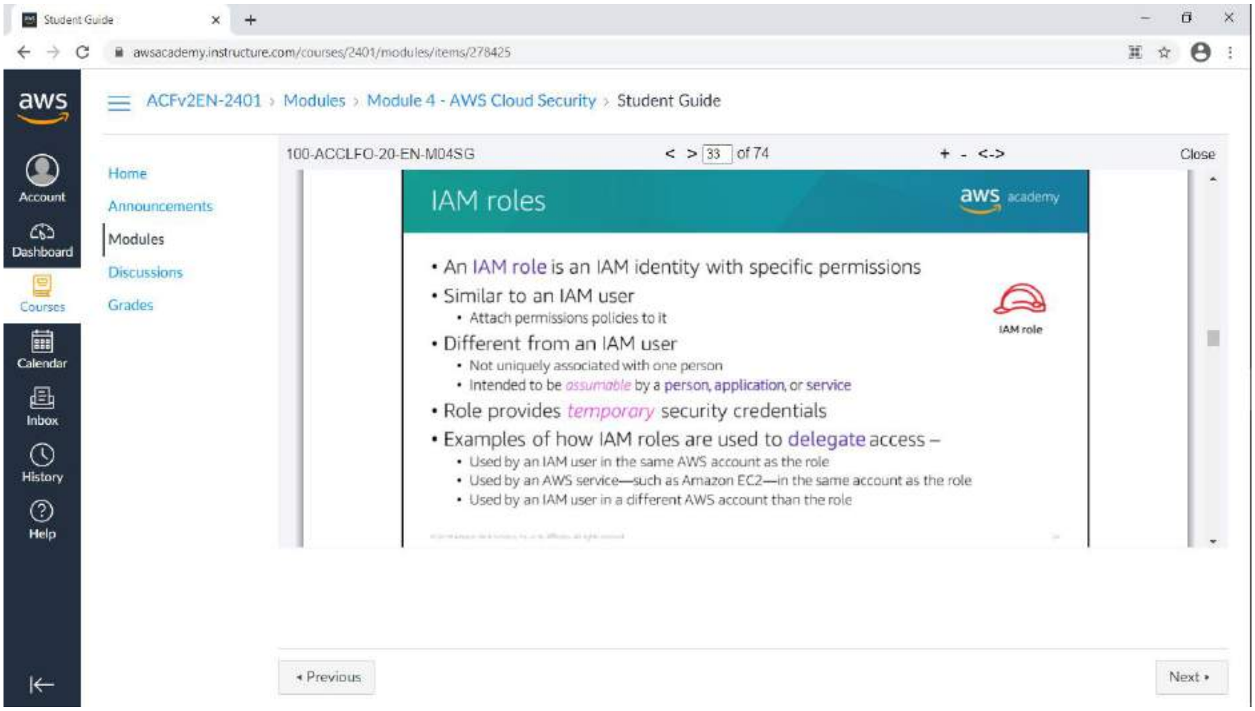
AWS account

IAM group: Admins	IAM group: Developers	IAM group: Testers
Carlos Salazar	Li Juan	Zhang Wei
Márcia Oliveira	Mary Major	John Stiles
	Richard Roe	Li Juan

An IAM group is a collection of IAM users. IAM groups offer a convenient way to specify permissions for a collection of users, which can make it easier to manage the permissions for those users. For example, you could create an IAM group that is called Developers and attach an IAM policy or multiple IAM policies to the Developers group that grant the AWS resource access permissions that developers typically need. Any use r that you then add to the Developer group will automatically have the permissions that are assigned to the group. In such a case, you do not need to attach the IAM policy or IAM policies directly to the user. If a new user joins your organization and should be granted developer privileges, you can simply add that user to the Developers group. Similarly, if a person changes jobs in your organization, instead of editing that user's permissions, simply remove the user from the group.

Important characteristics of IAM groups:

- A group can contain many users, and a user can belong to multiple groups.
- Groups cannot be nested. A group can contain only users, and a group cannot contain other groups.
- There is no default group that automatically includes all users in the AWS account. If you want to have a group with all account users in it, you need to create the group and add each new user to it.

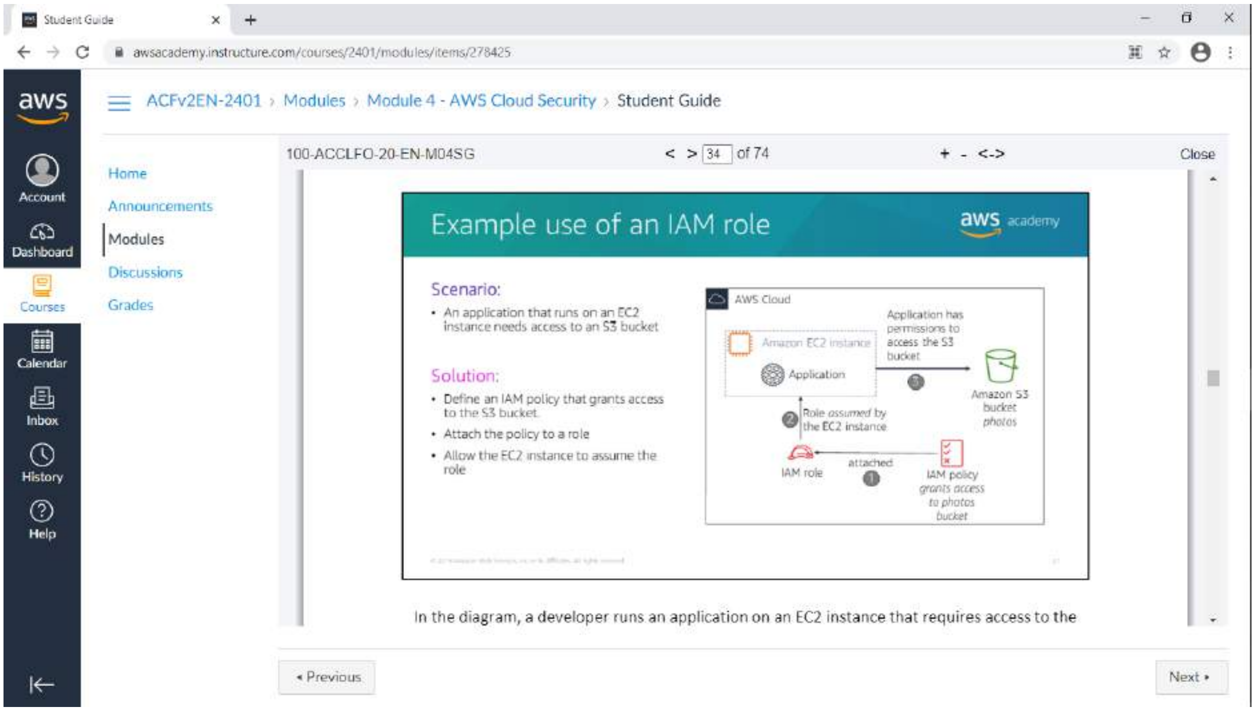


IAM roles

- An IAM role is an IAM identity with specific permissions
- Similar to an IAM user
  - Attach permissions policies to it
- Different from an IAM user
  - Not uniquely associated with one person
  - Intended to be *assumable* by a person, application, or service
- Role provides *temporary* security credentials
- Examples of how IAM roles are used to delegate access –
  - Used by an IAM user in the same AWS account as the role
  - Used by an AWS service—such as Amazon EC2—in the same account as the role
  - Used by an IAM user in a different AWS account than the role

An IAM role is an IAM identity you can create in your account that has specific permissions. An IAM role is similar to an IAM user because it is also an AWS identity that you can attach permissions policies to, and those permissions determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, the role provides you with temporary security credentials for your role session. You can use roles to delegate access to users, applications, or services that do not normally have access to your AWS resources. For example, you might want to grant users in your AWS account access to resources they don't usually have, or grant users in one AWS account access to resources in another account. Or you might want to allow a mobile app to use AWS resources, but you do not want to embed AWS keys within the app (where the keys can be difficult to rotate and where users can potentially extract them and misuse them). Also, sometimes you may want to grant AWS access to users who already have identities that are defined outside of AWS, such as in your corporate directory. Or, you might want to grant access to your account to third parties so that they can perform an audit on your resources.

For all of these example use cases, IAM roles are an essential component to implementing the cloud deployment.



Student Guide
awsacademy.instructure.com/courses/2401/modules/items/278425
aws
Account
Dashboard
Courses
Calendar
Inbox
History
Help
ACFv2EN-2401 › Modules › Module 4 - AWS Cloud Security › Student Guide
Home
Announcements
Modules
Discussions
Grades
100-ACCLFO-20-EN-M04SG
< > 34 of 74
+ - <->
Close

Example use of an IAM role

Scenario:

- An application that runs on an EC2 instance needs access to an S3 bucket

Solution:

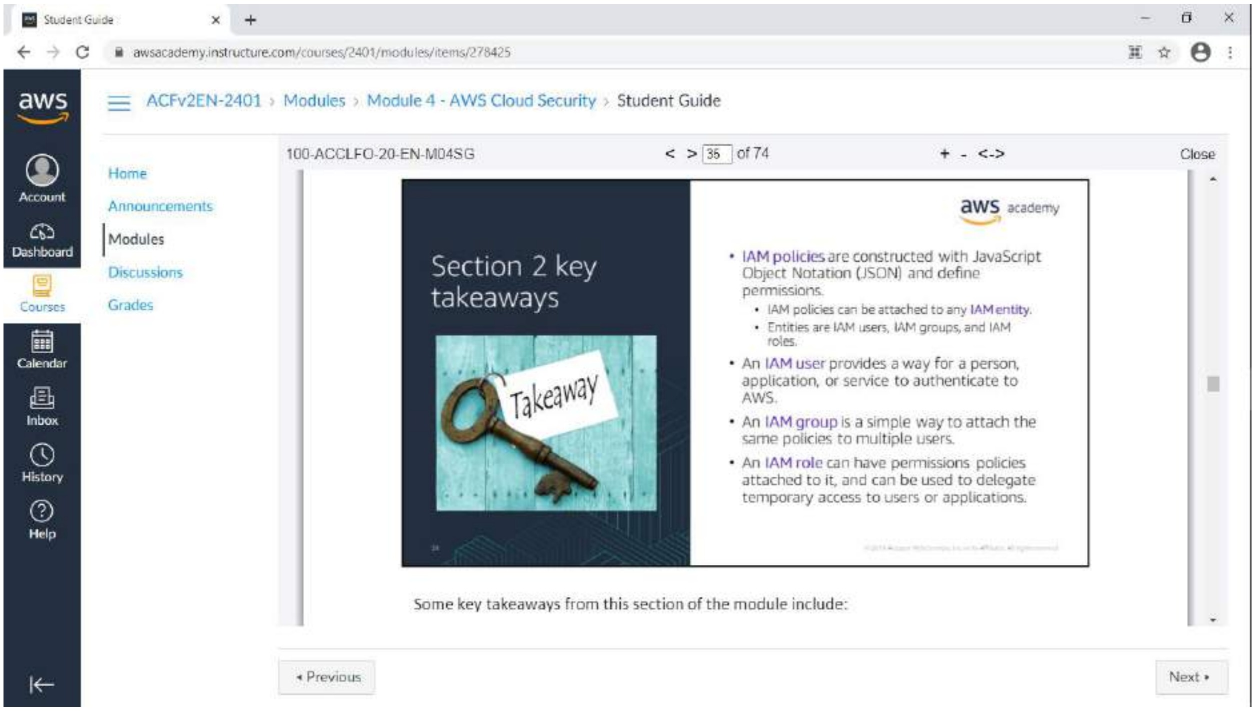
- Define an IAM policy that grants access to the S3 bucket
- Attach the policy to a role
- Allow the EC2 instance to assume the role

aws academy
AWS Cloud
Amazon EC2 instance
Application
Application has permissions to access the S3 bucket
3
Amazon S3 bucket *photos*
2
Role *assumed* by the EC2 instance
IAM role
attached
1
IAM policy *grants access to photos bucket*

In the diagram, a developer runs an application on an EC2 instance that requires access to the

In the diagram, a developer runs an application on an EC2 instance that requires access to the S3 bucket that is named photos. An administrator creates the IAM role and attaches the role to the EC2 instance. The role includes a permissions policy that grants read-only access to the specified S3 bucket. It also includes a trust policy that allows the EC2 instance to assume the role and retrieve the temporary credentials. When the application runs on the instance, it can use the role's temporary credentials to access the photos bucket. The administrator does not need to grant the application developer permission to access the photos bucket, and the developer never needs to share or manage credentials.

To learn more details about this example, see Using an IAM Role to Grant Permissions to Applications Running on Amazon EC2 Instances.



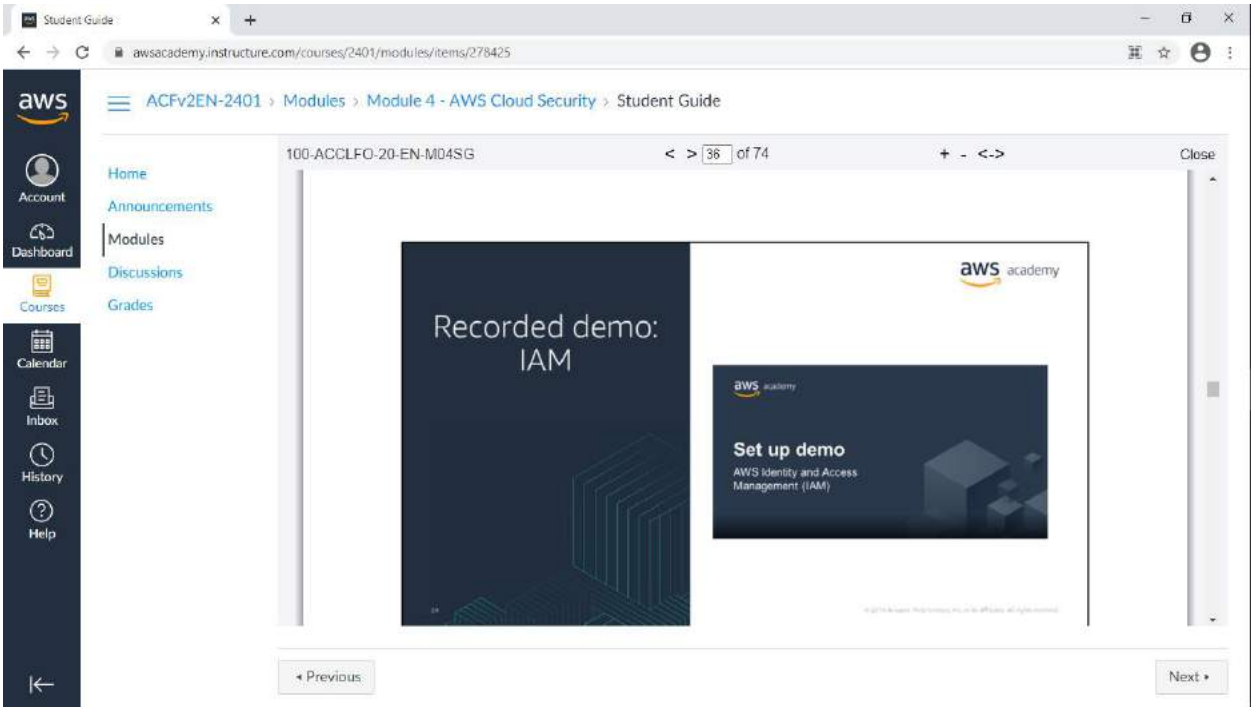
Section 2 key takeaways

- IAM policies are constructed with JavaScript Object Notation (JSON) and define permissions.
  - IAM policies can be attached to any IAM entity.
  - Entities are IAM users, IAM groups, and IAM roles.
- An IAM user provides a way for a person, application, or service to authenticate to AWS.
- An IAM group is a simple way to attach the same policies to multiple users.
- An IAM role can have permissions policies attached to it, and can be used to delegate temporary access to users or applications.

Some key takeaways from this section of the module include:

Some key takeaways from this section of the module include:

- IAM policies are constructed with JavaScript Object Notation (JSON) and define permissions.
- IAM policies can be attached to any IAM entity.
- Entities are IAM users, IAM groups, and IAM roles.
- An IAM user provides a way for a person, application, or service to authenticate to AWS.
- An IAM group is a simple way to attach the same policies to multiple users.
- An IAM role can have permissions policies attached to it, and can be used to delegate temporary access to users or applications.



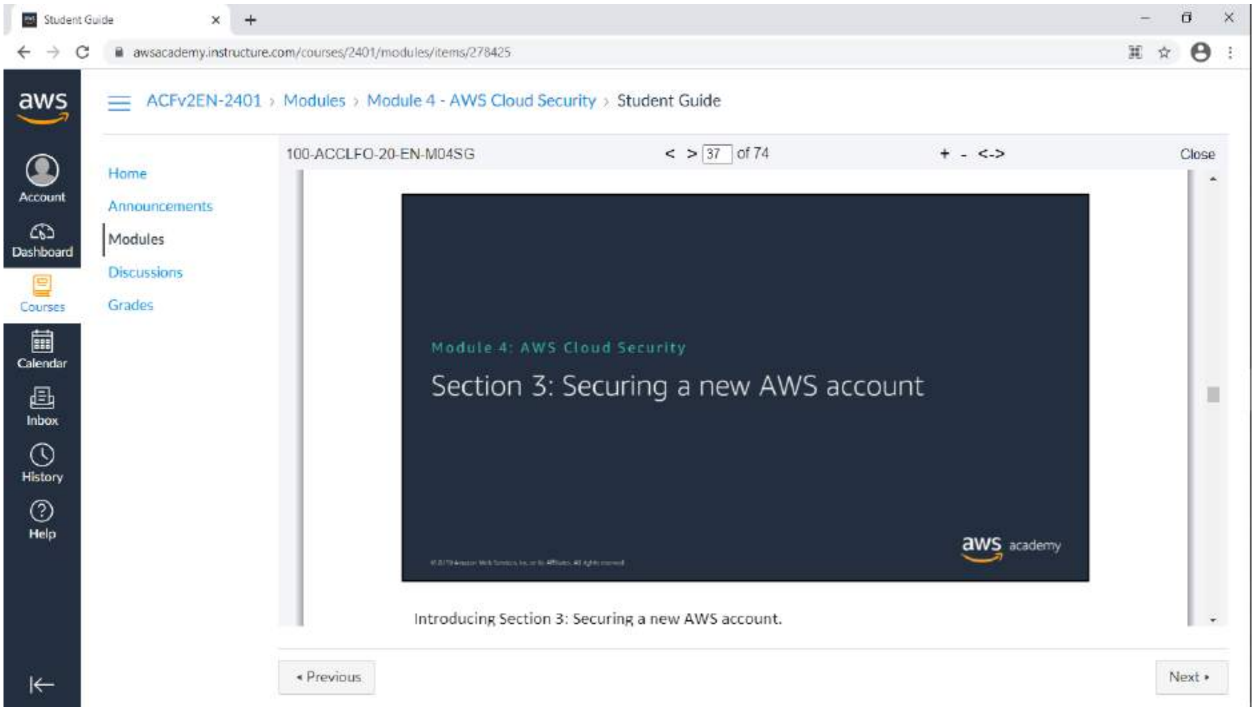
Recorded demo:
IAM

Set up demo
AWS Identity and Access Management (IAM)

Now, take a moment to watch the IAM Demo. The recording runs a little over 4 minutes, and it reinforces many of the concepts that were discussed in this section of the module.

The demonstration shows how to configure the following resources by using the AWS Management Console:

- An IAM role that will be used by an EC2 instance
- An IAM group
- An IAM user

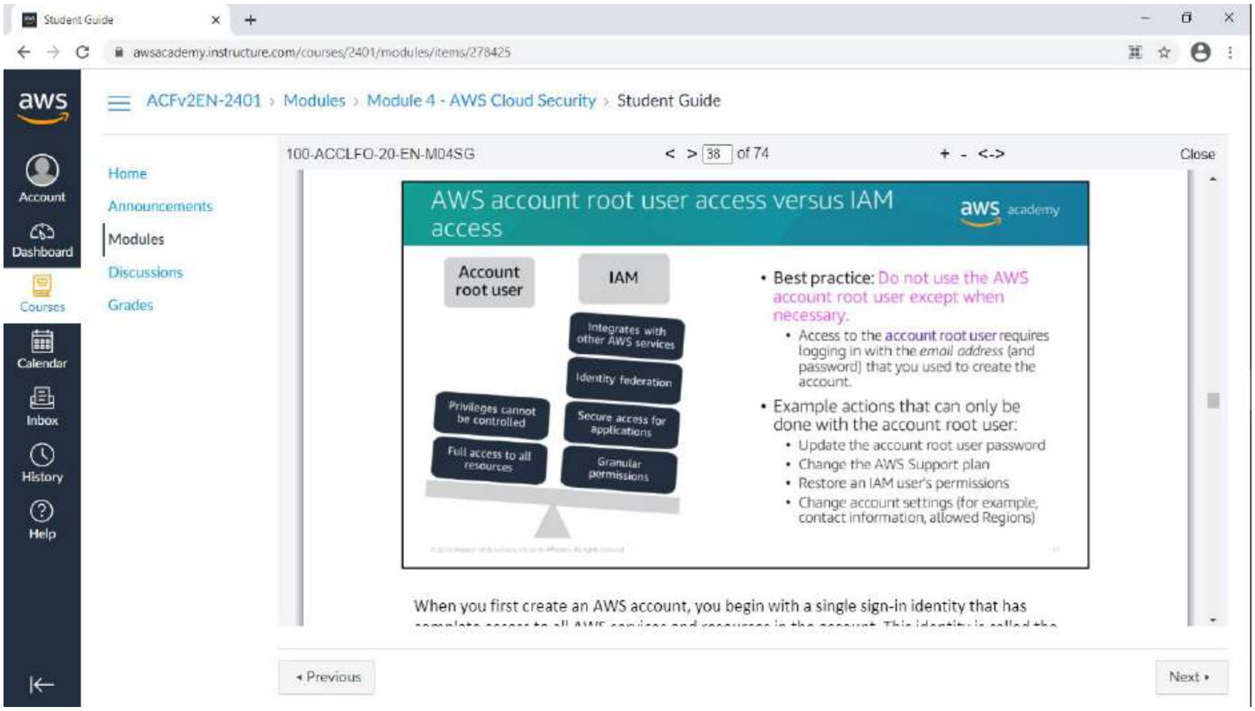


Module 4: AWS Cloud Security

Section 3: Securing a new AWS account

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Introducing Section 3: Securing a new AWS account.


Student Guide
awsacademy.instructure.com/courses/2401/modules/items/278425
aws
Account
Dashboard
Courses
Calendar
Inbox
History
Help
ACFv2EN-2401 › Modules › Module 4 - AWS Cloud Security › Student Guide
Home
Announcements
Modules
Discussions
Grades
100-ACCLFO-20-EN-M04SG
< > 38 of 74
+ - <->
Close
aws academy
Account root user
IAM
Integrates with other AWS services
Identity federation
Secure access for applications
Granular permissions
Privileges cannot be controlled
Full access to all resources
‹ Previous
Next ›

AWS account root user access versus IAM access

- Best practice: Do not use the AWS account root user except when necessary.
  - Access to the account root user requires logging in with the *email address* (and password) that you used to create the account.
- Example actions that can only be done with the account root user:
  - Update the account root user password
  - Change the AWS Support plan
  - Restore an IAM user's permissions
  - Change account settings (for example, contact information, allowed Regions)

When you first create an AWS account, you begin with a single sign-in identity that has

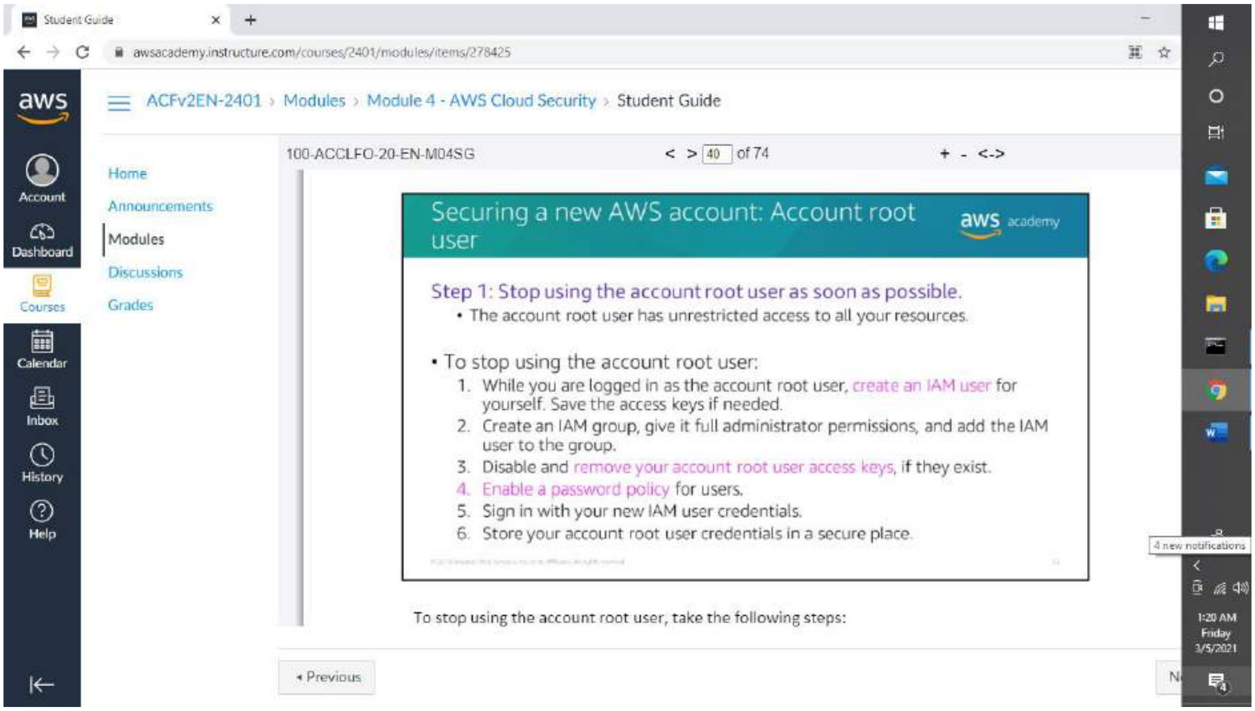
When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user and it is accessed by signing into the AWS Management Console with the email address and password that you used to create the account. AWS account root users have (and retain) full access to all resources in the account. Therefore, AWS strongly recommends that you do not use account root user credentials for day-to-day interactions with the account.

Instead, AWS recommends that you use IAM to create additional users and assign permissions to these users, following the principle of least privilege. 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 the account. 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 the account, you can create unique credentials for each user and define which user will have access to which resources.

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. You should avoid sharing the same credentials with multiple users.

While the account root user should not be used for routine tasks, there are a few tasks that can only be accomplished by logging in as the account root user. A full list of these tasks is detailed on the Tasks that require root user credentials AWS documentation page.



Securing a new AWS account: Account root user

aws academy

Step 1: Stop using the account root user as soon as possible.

- The account root user has unrestricted access to all your resources.

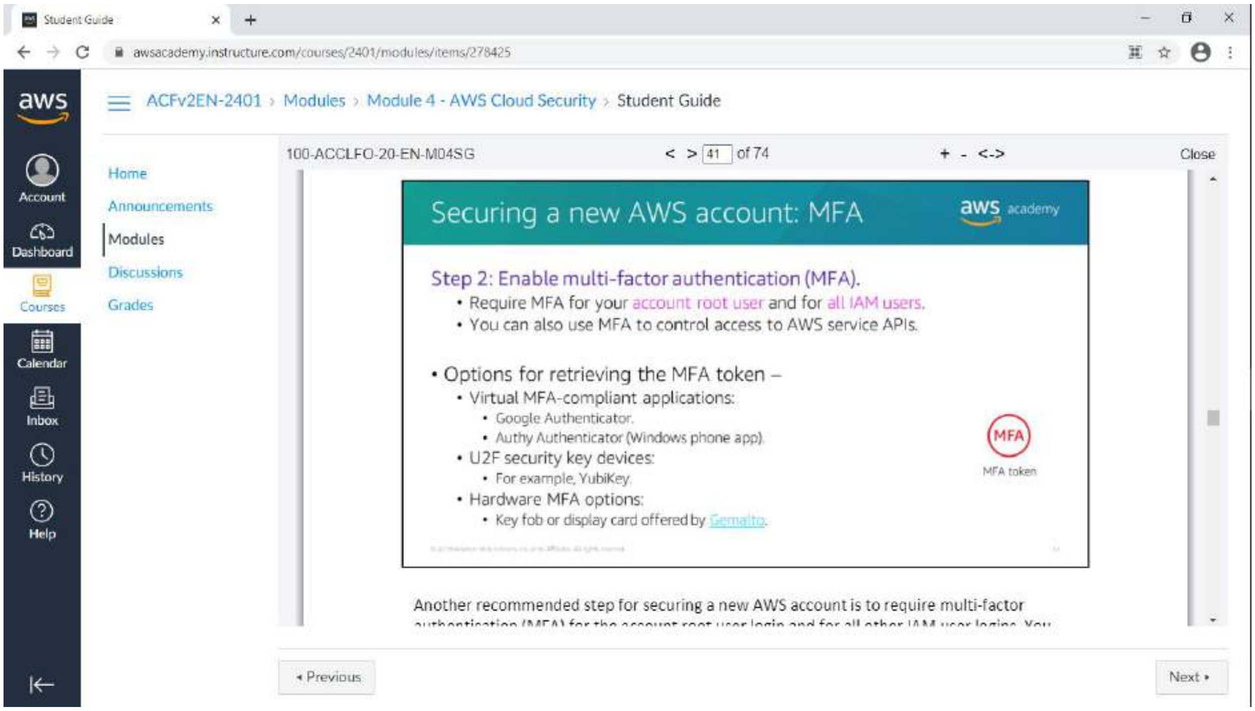
- To stop using the account root user:
  1. While you are logged in as the account root user, create an IAM user for yourself. Save the access keys if needed.
  2. Create an IAM group, give it full administrator permissions, and add the IAM user to the group.
  3. Disable and remove your account root user access keys, if they exist.
  4. Enable a password policy for users.
  5. Sign in with your new IAM user credentials.
  6. Store your account root user credentials in a secure place.

To stop using the account root user, take the following steps:

To stop using the account root user, take the following steps:

1. While you are logged into the account root user, create an IAM user for yourself with AWS Management Console access enabled (but do not attach any permissions to the user yet). Save the IAM user access keys if needed.
2. Next, create an IAM group, give it a name (such as FullAccess), and attach IAM policies to the group that grant full access to at least a few of the services you will use. Next, add the IAM user to the group.
3. Disable and remove your account root user access keys, if they exist.
4. Enable a password policy for all users. Copy the IAM users sign-in link from the IAM Dashboard page. Then, sign out as the account root user.
5. Browse to the IAM users sign-in link that you copied, and sign in to the account by using your new IAM user credentials.
6. Store your account root user credentials in a secure place.

To view detailed instructions for how to set up your first IAM user and IAM group, see Creating Your First IAM Admin User and Group



Securing a new AWS account: MFA

Step 2: Enable multi-factor authentication (MFA).

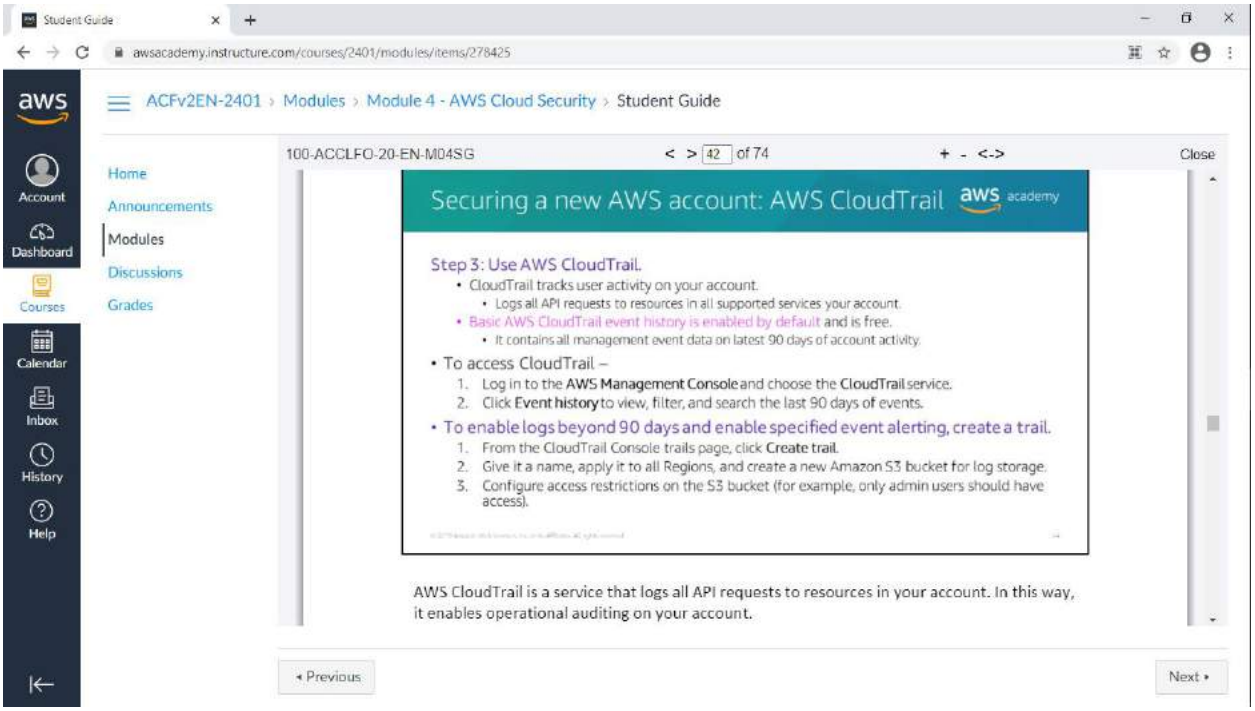
- Require MFA for your account root user and for all IAM users.
- You can also use MFA to control access to AWS service APIs.

- Options for retrieving the MFA token –
  - Virtual MFA-compliant applications:
    - Google Authenticator.
    - Authy Authenticator (Windows phone app).
  - U2F security key devices:
    - For example, YubiKey.
  - Hardware MFA options:
    - Key fob or display card offered by Gemalto.

Another recommended step for securing a new AWS account is to require multi-factor

Another recommended step for securing a new AWS account is to require multi-factor authentication (MFA) for the account root user login and for all other IAM user logins. You can also use MFA to control programmatic access. For details, see Configuring MFA-Protected API Access.

You have a few options for retrieving the MFA token that is needed to log in when MFA is enabled. Options include virtual MFA-compliant applications (such as Google Authenticator and Authy Authenticator), U2F security key devices, and hardware MFA options that provide a key fob or display card.



Securing a new AWS account: AWS CloudTrail

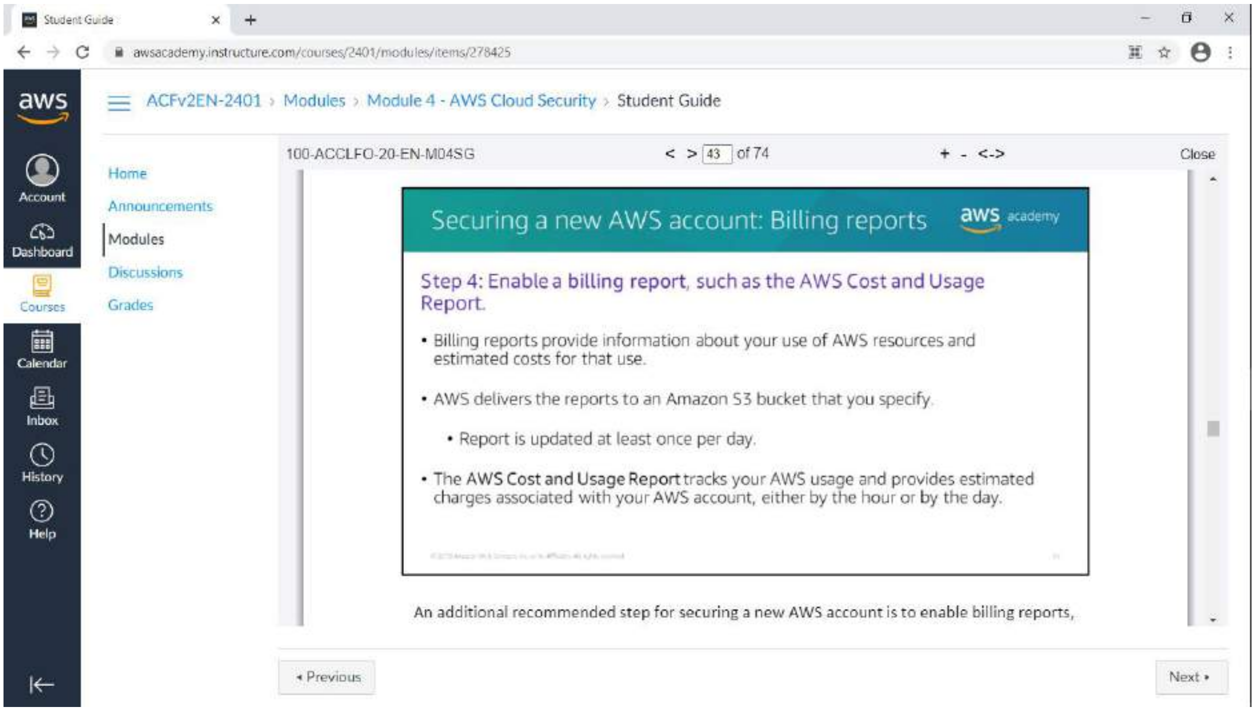
Step 3: Use AWS CloudTrail.

- CloudTrail tracks user activity on your account.
  - Logs all API requests to resources in all supported services your account.
- Basic AWS CloudTrail event history is enabled by default and is free.
  - It contains all management event data on latest 90 days of account activity.

- To access CloudTrail –
  1. Log in to the **AWS Management Console** and choose the **CloudTrail** service.
  2. Click **Event history** to view, filter, and search the last 90 days of events.
- To enable logs beyond 90 days and enable specified event alerting, create a trail.
  1. From the CloudTrail Console trails page, click **Create trail**.
  2. Give it a name, apply it to all Regions, and create a new Amazon S3 bucket for log storage.
  3. Configure access restrictions on the S3 bucket (for example, only admin users should have access).

AWS CloudTrail is a service that logs all API requests to resources in your account. In this way, it enables operational auditing on your account.

AWS CloudTrail is a service that logs all API requests to resources in your account. In this way, it enables operational auditing on your account. AWS CloudTrail is enabled on account creation by default on all AWS accounts, and it keeps a record of the last 90 days of account management event activity. You can view and download the last 90 days of your account activity for create, modify, and delete operations of services that are supported by CloudTrail without needing to manually create another trail. To enable CloudTrail log retention beyond the last 90 days and to enable alerting whenever specified events occur, create a new trail (which is described at a high level on the slide). For detailed step-by-step instructions about how to create a trail in AWS CloudTrail, see creating a trail in the AWS documentation.



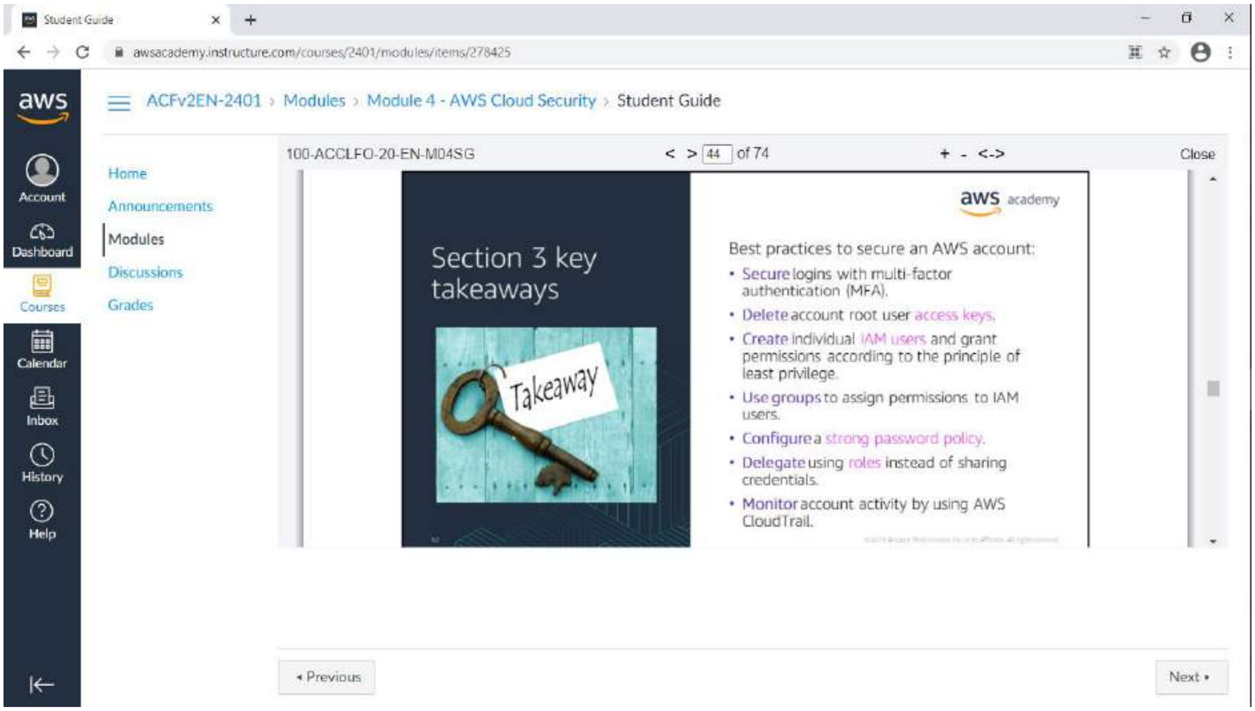
Securing a new AWS account: Billing reports

Step 4: Enable a billing report, such as the AWS Cost and Usage Report.

- Billing reports provide information about your use of AWS resources and estimated costs for that use.
- AWS delivers the reports to an Amazon S3 bucket that you specify.
  - Report is updated at least once per day.
- The **AWS Cost and Usage Report** tracks your AWS usage and provides estimated charges associated with your AWS account, either by the hour or by the day.

An additional recommended step for securing a new AWS account is to enable billing reports,

An additional recommended step for securing a new AWS account is to
enable billing reports,
such as the
AWS Cost and Usage Report
. Billing reports provide information about your use
of AWS resources and estimated costs for that use.
AWS delivers the reports to an Amazon
S3 bucket that you specify and AWS updates the reports at least once
per day.
The AWS Cost and Usage Report tracks usage in the AWS account and
provides estimated
charges, either by the hour or by the day.
For details about how to create an AWS Cost and Usage Report, see the
AWS
Documentation



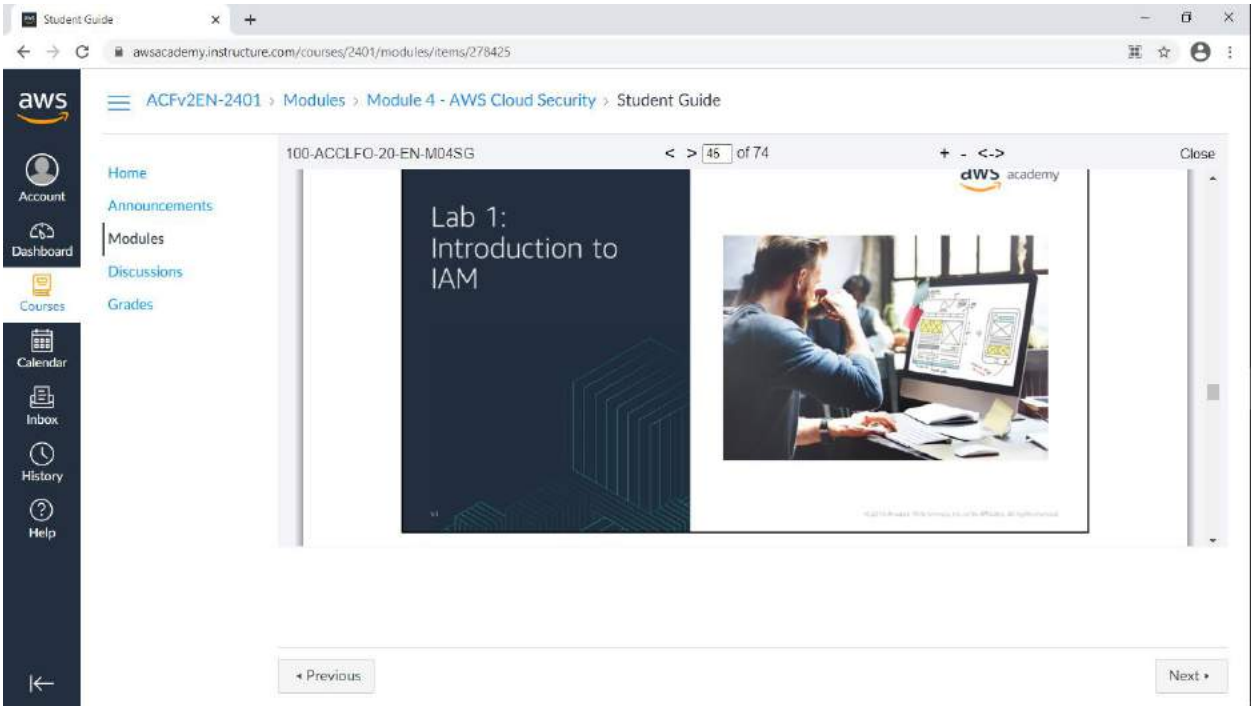
Section 3 key takeaways

Best practices to secure an AWS account:

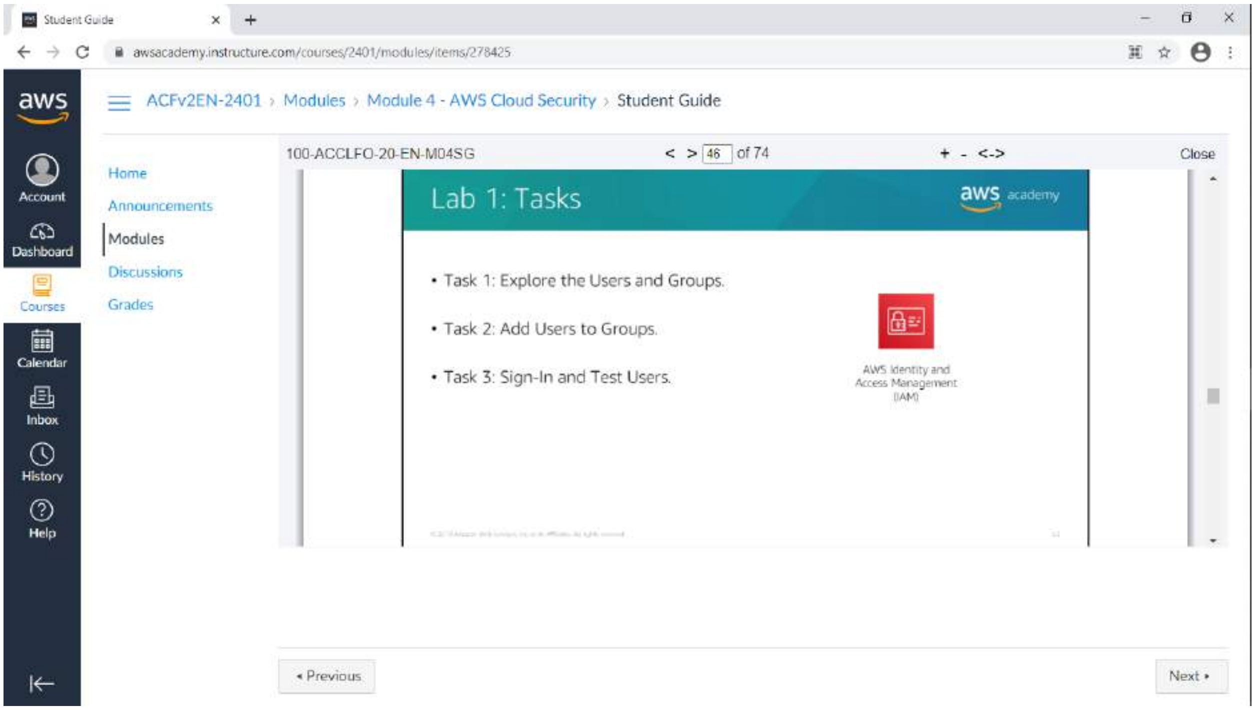
- Secure logins with multi-factor authentication (MFA).
- Delete account root user access keys.
- Create individual IAM users and grant permissions according to the principle of least privilege.
- Use groups to assign permissions to IAM users.
- Configure a strong password policy.
- Delegate using roles instead of sharing credentials.
- Monitor account activity by using AWS CloudTrail.

The key takeaways from this section of the module are all related to best practices for securing an AWS account. Those best practice recommendations include:

- Secure logins with multi-factor authentication (MFA).
- Delete account root user access keys.
- Create individual IAM users and grant permissions according to the principle of least privilege.
- Use groups to assign permissions to IAM users.
- Configure a strong password policy.
- Delegate using roles instead of sharing credentials.
- Monitor account activity using AWS CloudTrail.



Lab 1: Introduction to IAM

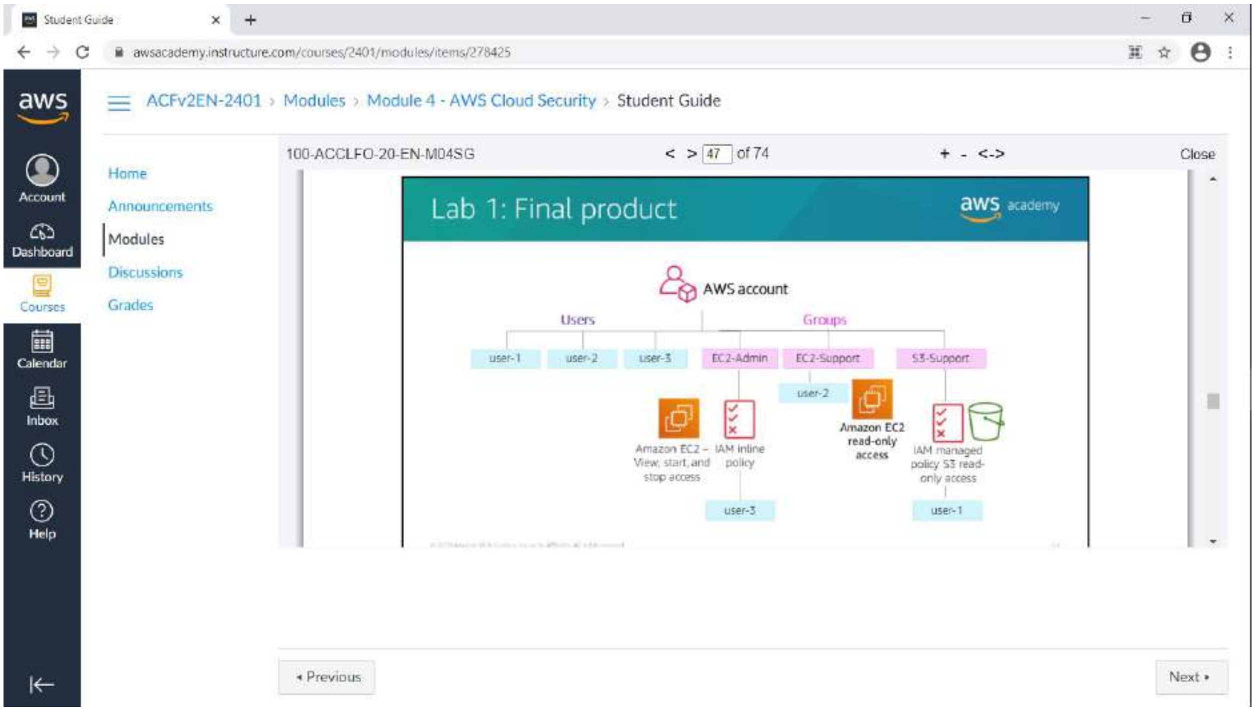


Lab 1: Tasks

- Task 1: Explore the Users and Groups.
- Task 2: Add Users to Groups.
- Task 3: Sign-In and Test Users.

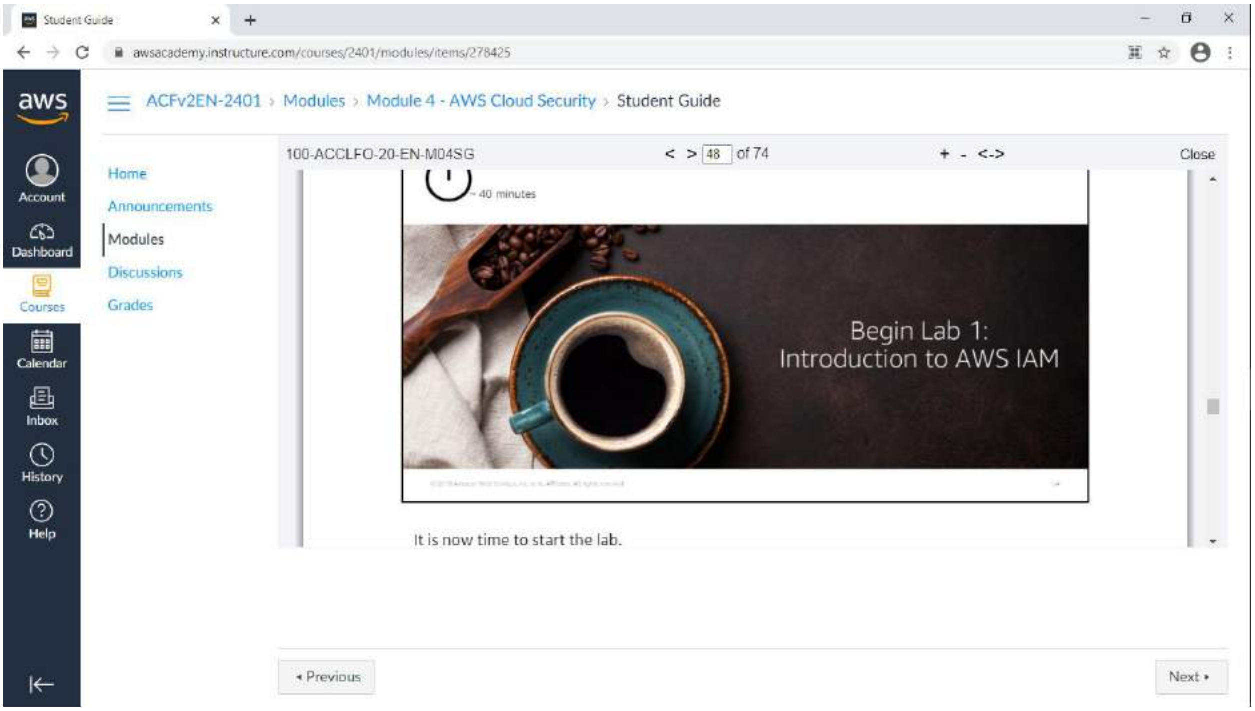
In this hands-on lab, you will:

- Explore pre-created IAM users and groups.
- Inspect IAM policies as they are applied to the pre-created groups.
- Follow a real-world scenario and add users to groups that have specific capabilities enabled.
- Locate and use the IAM sign-in URL.
- Experiment with the effects of IAM policies on access to AWS resources.



Lab 1: Final product

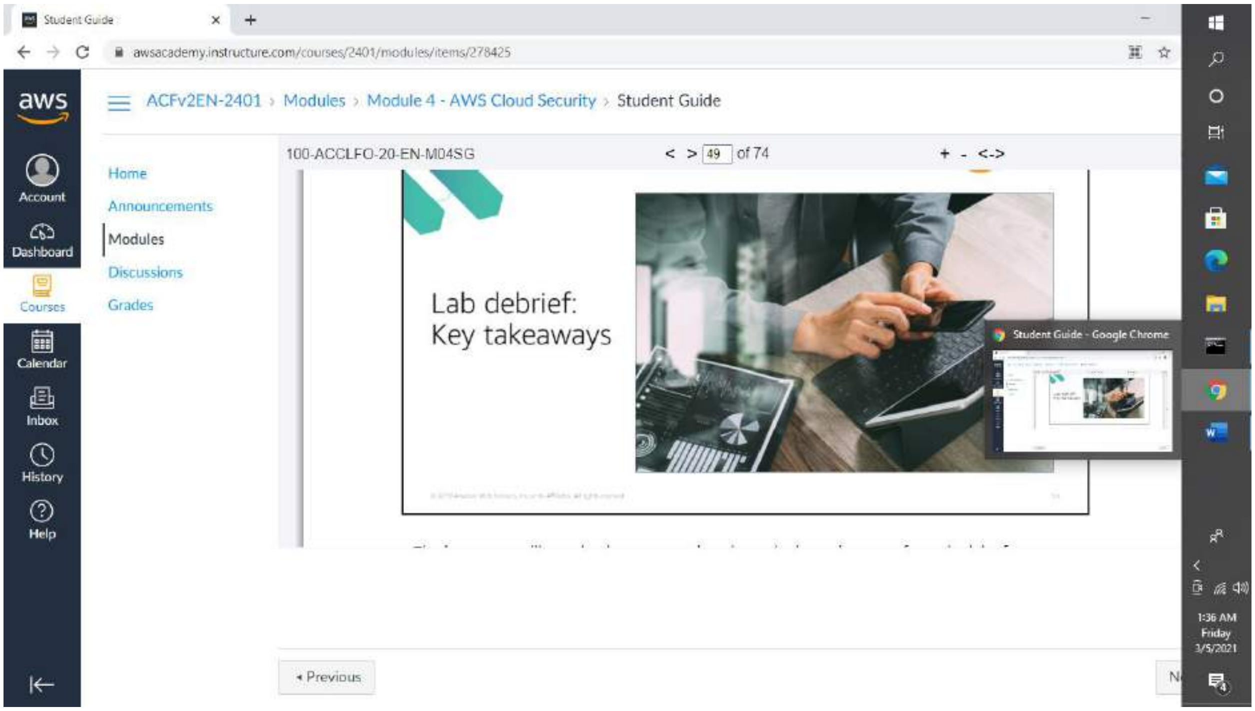
The diagram shows the resources that your AWS account will have after you complete the lab steps. It also describes how the resources will be configured.



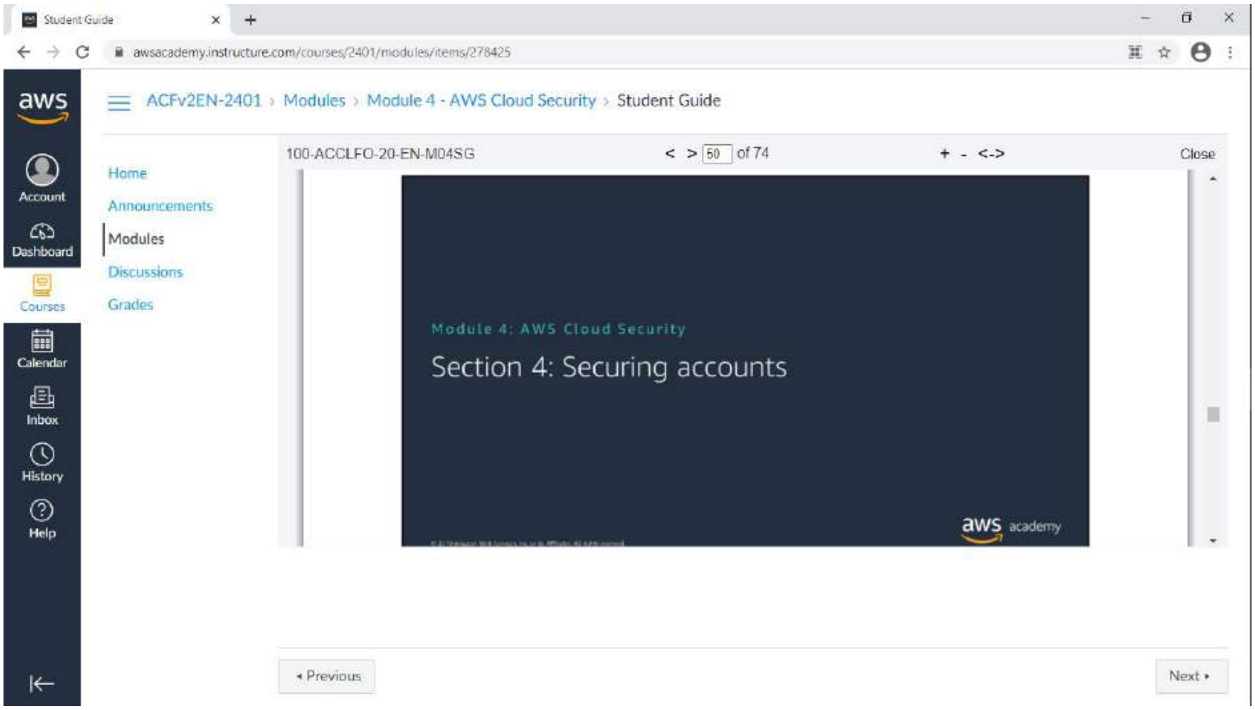
~ 40 minutes

Begin Lab 1:
Introduction to AWS IAM

It is now time to start the lab.

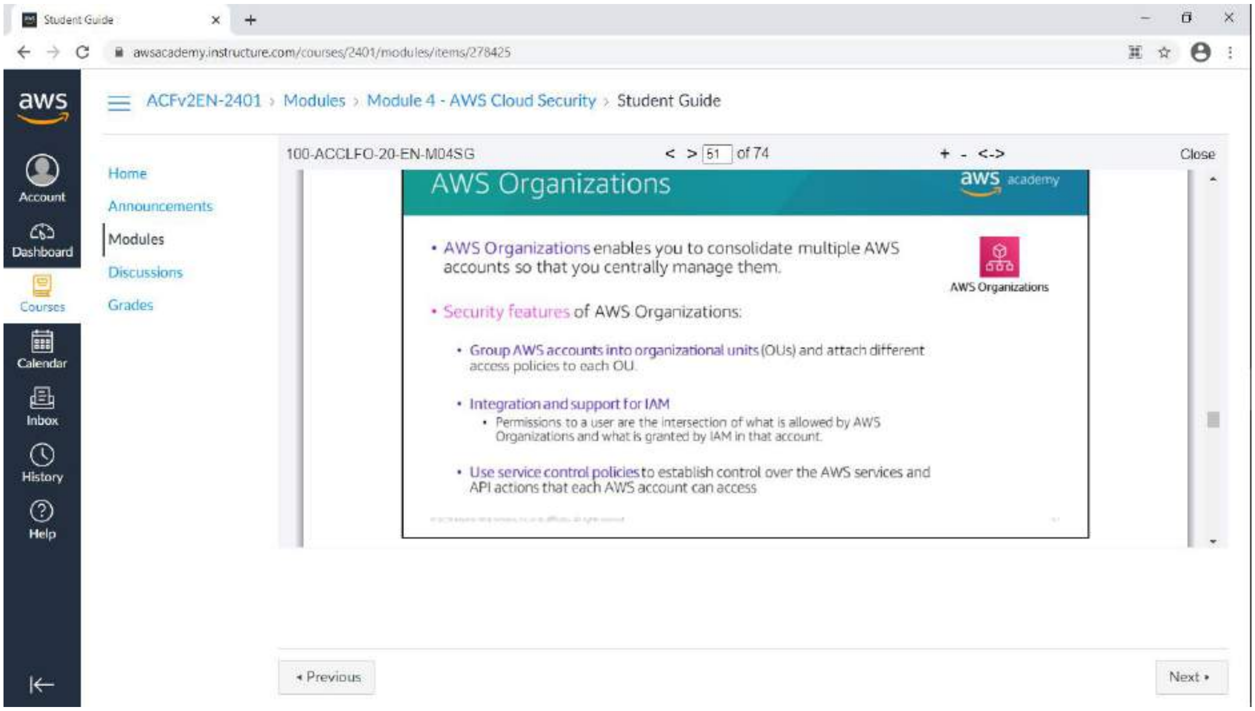


Lab debrief:
Key takeaways



Module 4: AWS Cloud Security

Section 4: Securing accounts

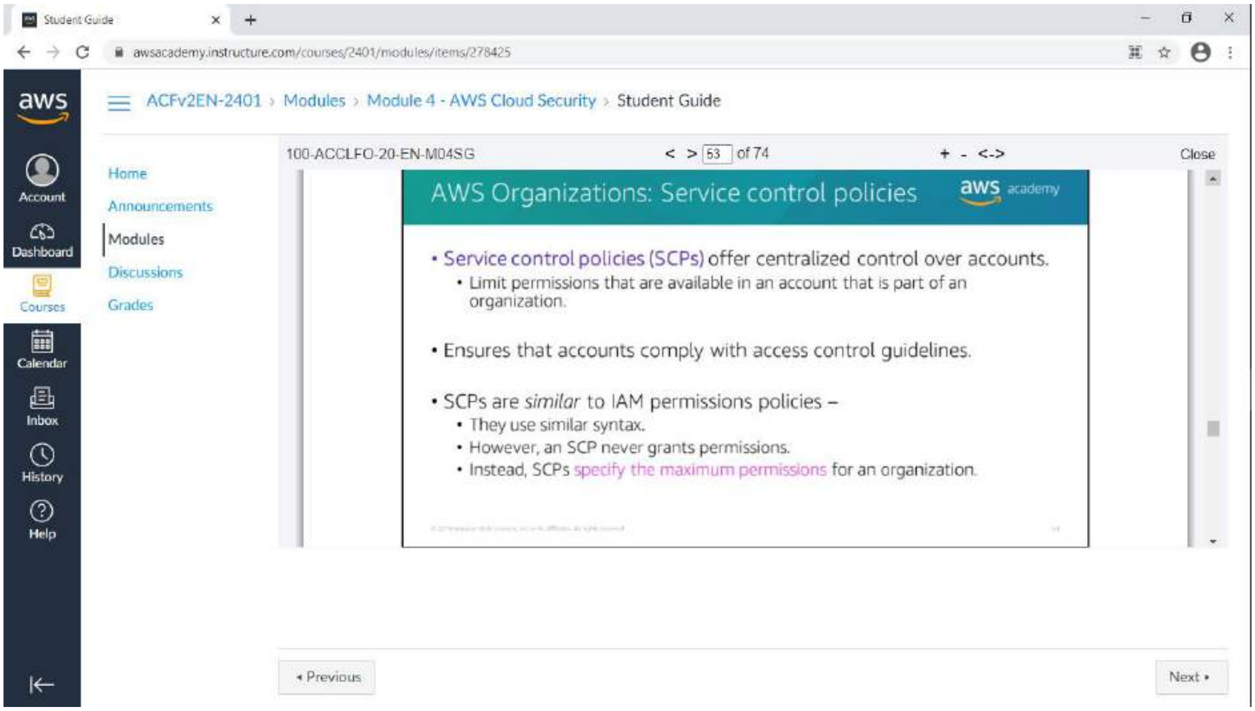


AWS Organizations

- AWS Organizations enables you to consolidate multiple AWS accounts so that you centrally manage them.
- Security features of AWS Organizations:
  - Group AWS accounts into organizational units (OUs) and attach different access policies to each OU.
  - Integration and support for IAM
    - Permissions to a user are the intersection of what is allowed by AWS Organizations and what is granted by IAM in that account.
  - Use service control policies to establish control over the AWS services and API actions that each AWS account can access

AWS Organizations is an account management service that enables you to consolidate multiple AWS accounts into an
organization
that you create and centrally manage. Here, the focus is on the security features that AWS Organizations provides.
One helpful security feature is that you can
group accounts into organizational units
(OUs)
and attach different access policies to each OU. For example, if you have accounts that
should only be allowed to access AWS services that
meet certain regulatory requirements,
you can put those accounts into one OU. You then can define a policy that blocks OU access
to services that do not meet those regulatory requirements, and then attach the policy to the
OU.
Another security feature is that
AWS Organizations integrates with and supports IAM.
AWS
Organizations expands that control to the account level by giving you control over what users
and roles in an account or a group of accounts can
do. The resulting permissions are the
logical intersection of what is allowed by the AWS
Organizations policy settings and what
permissions are explicitly granted by IAM in the ac
count for that user or role. The user can
access only what is allowed by
both
the AWS Organizations policies and IAM policies.
Finally, AWS Organizations
provides service control policies (SCPs)
that enable you to specify
the maximum permissions that member accounts in the
organization can have. In SCPs, you can restrict which AWS services, resources, and individual actions the users and roles in each
member account can access.

These restrictions even override the administrators of member accounts. When AWS Organizations blocks access to a service, resource, or API action, a user or role in that account can't access it, even if an administrator of a member account explicitly grants such permissions.

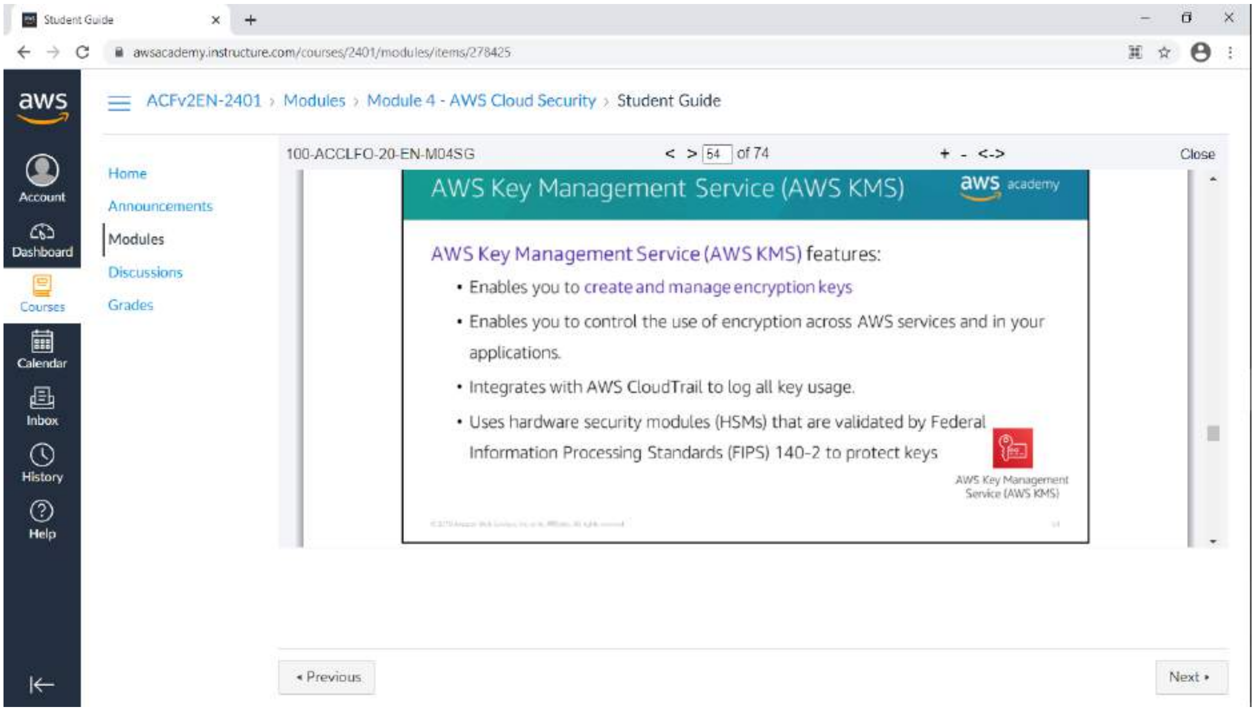


AWS Organizations: Service control policies

- Service control policies (SCPs) offer centralized control over accounts.
  - Limit permissions that are available in an account that is part of an organization.
- Ensures that accounts comply with access control guidelines.
- SCPs are *similar* to IAM permissions policies –
  - They use similar syntax.
  - However, an SCP never grants permissions.
  - Instead, SCPs specify the maximum permissions for an organization.

Here is a closer look at the Service control policies (SCPs) feature of AWS Organizations. SCPs offer central control over the maximum available permissions for all accounts in your organization, enabling you to ensure that your accounts stay in your organization’s access control guidelines. SCPs are available only in an organization that has all features enabled, including consolidated billing. SCPs aren't available if your organization has enabled only the consolidated billing features. For instructions about enabling SCPs, see Enabling and Disabling a Policy Type on a Root.

SCPs are similar to IAM permissions policies and they use almost the same syntax. However, an SCP never grants permissions. Instead, SCPs are JSON policies that specify the maximum permissions for an organization or OU. Attaching an SCP to the organization root or an organizational unit (OU) defines a safeguard for the actions that accounts in the organization root or OU can do. However, it is not a substitute for well-managed IAM configurations within each account. You must still attach IAM policies to users and roles in your organization's accounts to actually grant permissions to them.

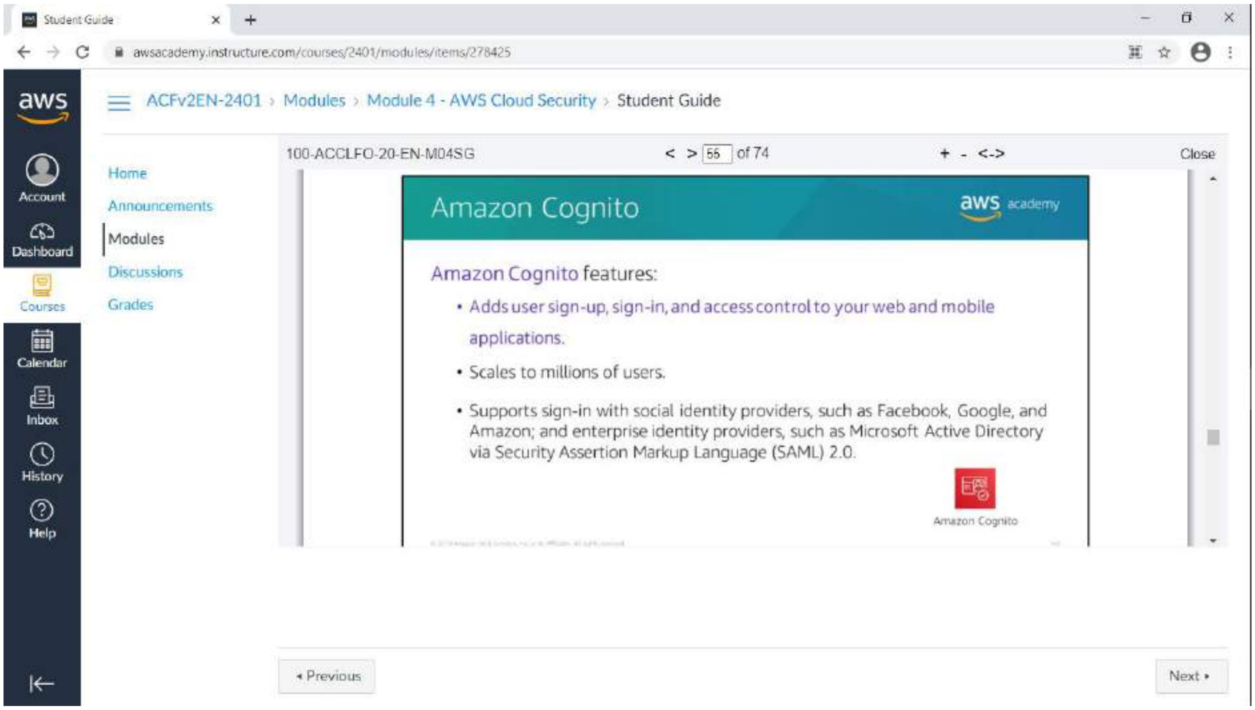


AWS Key Management Service (AWS KMS)

AWS Key Management Service (AWS KMS) features:

- Enables you to create and manage encryption keys
- Enables you to control the use of encryption across AWS services and in your applications.
- Integrates with AWS CloudTrail to log all key usage.
- Uses hardware security modules (HSMs) that are validated by Federal Information Processing Standards (FIPS) 140-2 to protect keys

AWS Key Management Service (AWS KMS)
is a service that enables you to create and
manage encryption keys, and to control the use of encryption across a wide range of AWS
services and your applications. AWS KMS is a secure
and resilient service that uses hardware
security modules (HSMs) that were validated under
Federal Information Processing
Standards (FIPS) 140-2
(or are in the process of being validated) to protect your keys. AWS KMS also integrates with AWS CloudTrail to provide
you with logs of all key usage to help
meet your regulatory and compliance needs.
Customer master keys (CMKs)
are used to control access to data encryption keys
that
encrypt and decrypt your data. You can create new keys when you want, and you can
manage who has access to these keys and who can use
them. You can also import keys from
your own key management infrastructure into AWS KMS
.
AWS KMS integrates with most AWS services, which means that you can use AWS KMS CMKs
to control the encryption of the data that you store in these services. To learn more, see
AWS
Key Management Service features



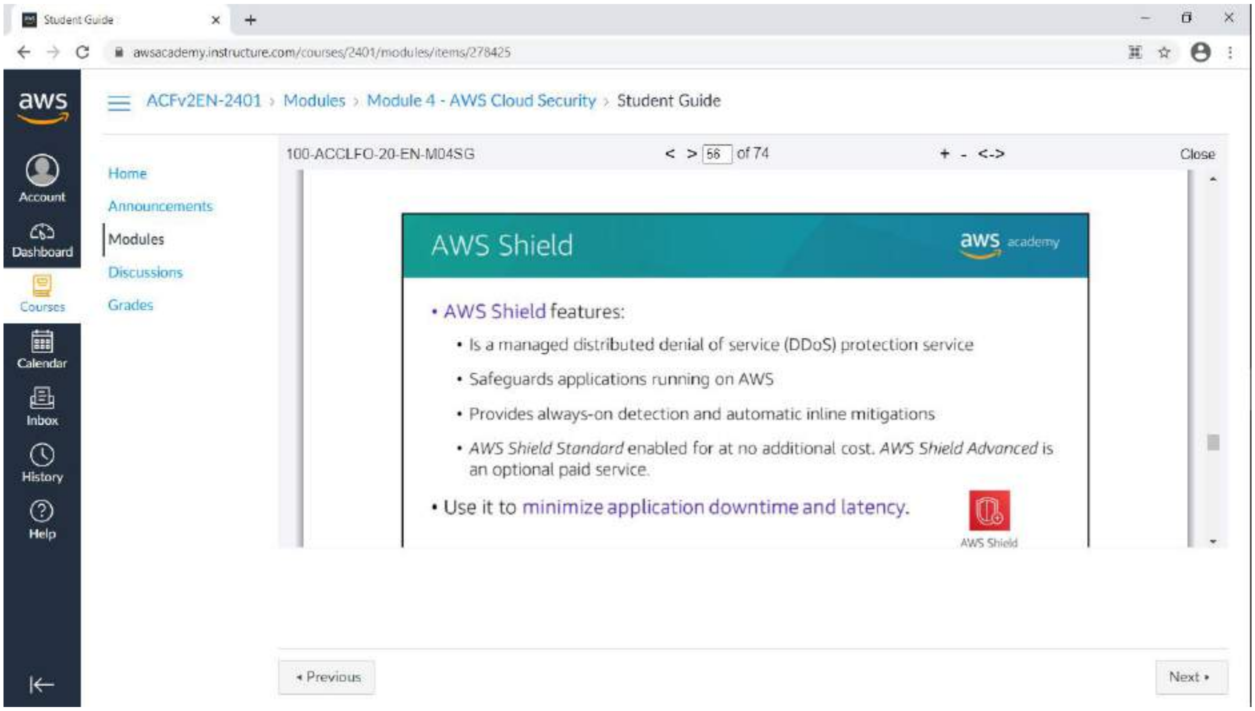
Amazon Cognito

Amazon Cognito features:

- Adds user sign-up, sign-in, and access control to your web and mobile applications.
- Scales to millions of users.
- Supports sign-in with social identity providers, such as Facebook, Google, and Amazon; and enterprise identity providers, such as Microsoft Active Directory via Security Assertion Markup Language (SAML) 2.0.

Amazon Cognito provides solutions to control access
to AWS resources from your application.
You can define roles and map users to different rol
es so your application can access only the
resources that are authorized for each user.
Amazon Cognito uses common identity management standards, such as
Security Assertion
Markup Language (SAML) 2.0
. SAML is an open standard for exchanging identity and security
information with applications and service providers
. Applications and service providers that
support SAML enable you to sign in by using your corporate directory
credentials, such as
your user name and password from Microsoft Active Directory. With
SAML, you can use
single sign-on (SSO) to sign in to all of your SAML
-enabled applications by using a single set of
credentials.

Amazon Cognito helps you meet multiple security and compliance requirements, including requirements for highly regulated organizations such as healthcare companies and merchants. Amazon Cognito is eligible for use with the US Health Insurance Portability and Accountability Act (HIPAA). It can also be used for workloads that are compliant with the Payment Card Industry Data Security Standard (PCI DSS); the American Institute of CPAs (AICPA) Service Organization Control (SOC); the International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) standards ISO/IEC 27001, ISO/IEC 27017, and ISO/IEC 27018; and ISO 9001.



AWS Shield

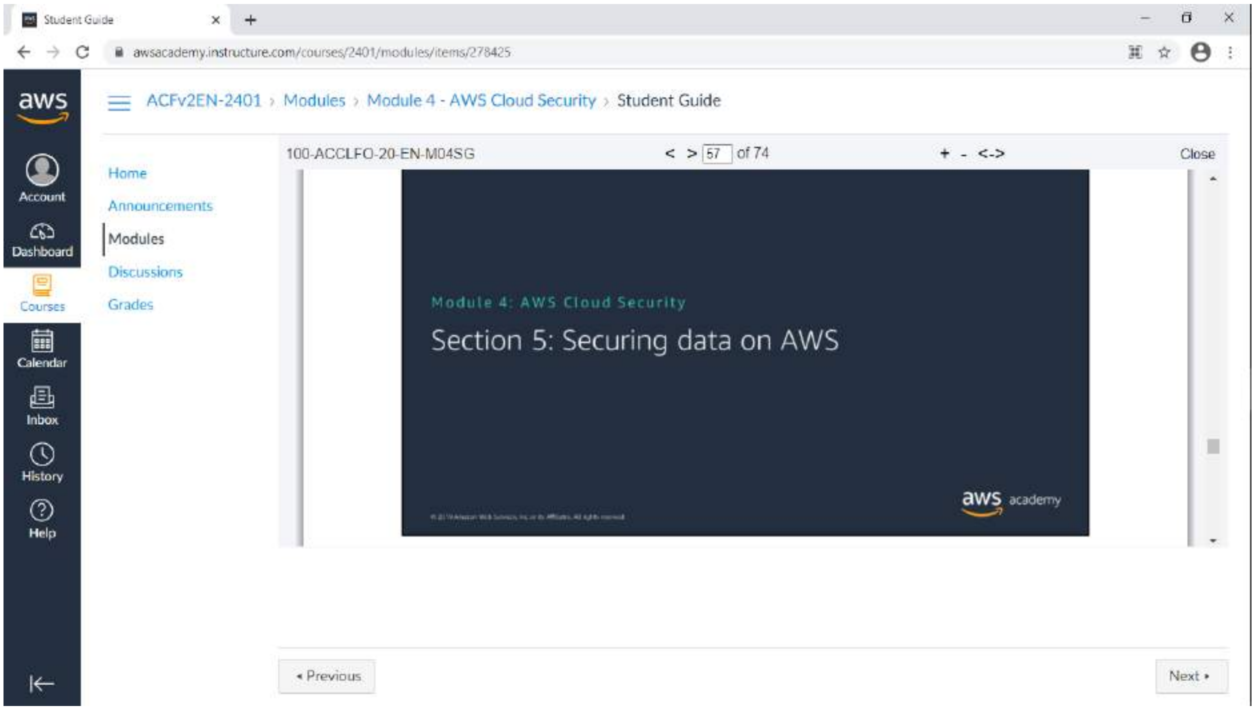
- AWS Shield features:
  - Is a managed distributed denial of service (DDoS) protection service
  - Safeguards applications running on AWS
  - Provides always-on detection and automatic inline mitigations
  - *AWS Shield Standard* enabled for at no additional cost. *AWS Shield Advanced* is an optional paid service.
- Use it to minimize application downtime and latency.

AWS Shield is a managed distributed denial of service (DDoS) protection service that safeguards applications that run on AWS. It provides always-on detection and automatic inline mitigations that minimize application downtime and latency, so there is no need to engage AWS Support to benefit from DDoS protection. AWS Shield helps protects your website from all types of DDoS attacks, including Infrastructure layer attacks (like User Datagram Protocol—or UDP—floods), state exhaustion attacks (like TCP SYN floods), and application-layer attacks (like HTTP GET or POST floods). For examples, see the AWS WAF Developer Guide.

AWS Shield Standard is automatically enabled to all AWS customers at no additional cost.

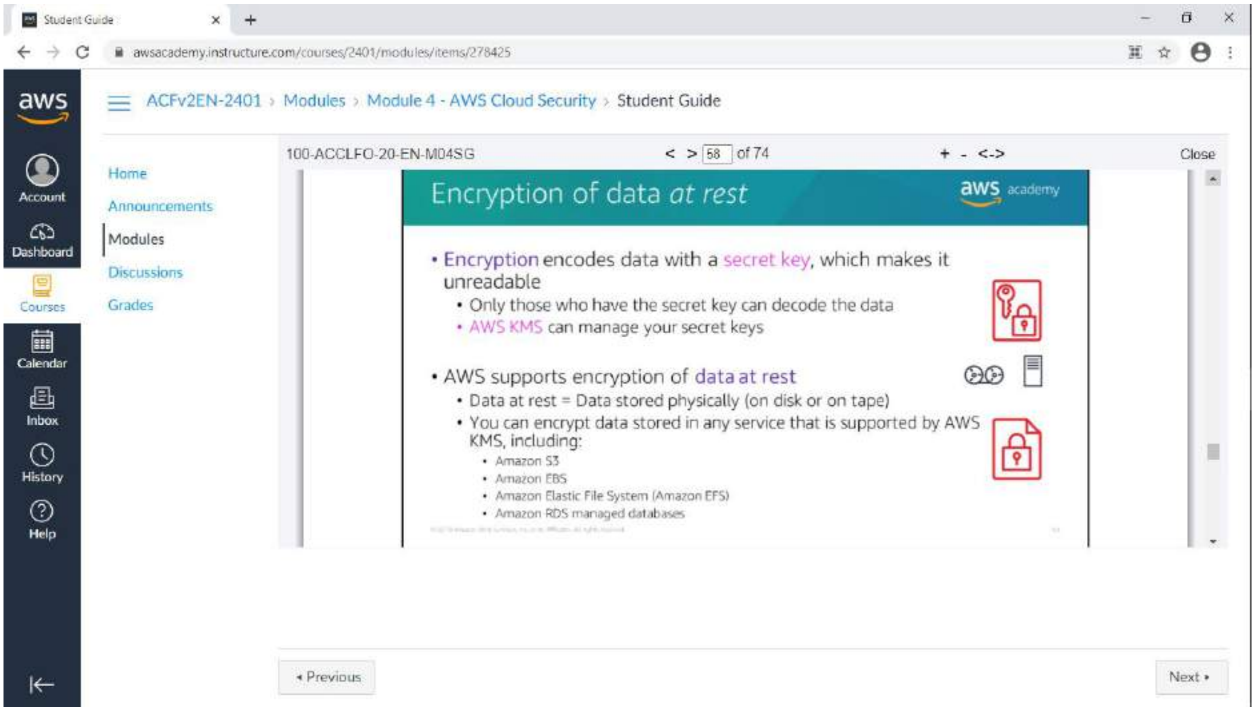
AWS Shield Advanced

is an optional paid service. AWS Shield Advanced provides additional protections against more sophisticated and larger attacks for your applications that run on
Amazon EC2, Elastic Load Balancing, Amazon CloudFront, AWS Global Accelerator, and
Amazon Route 53. AWS Shield Advanced is available to all customers. However, to contact
the DDoS Response Team, customers need to have either Enterprise Support or Business
Support from AWS Support.
56



Module 4: AWS Cloud Security

Section 5: Securing data on AWS



Encryption of data *at rest*

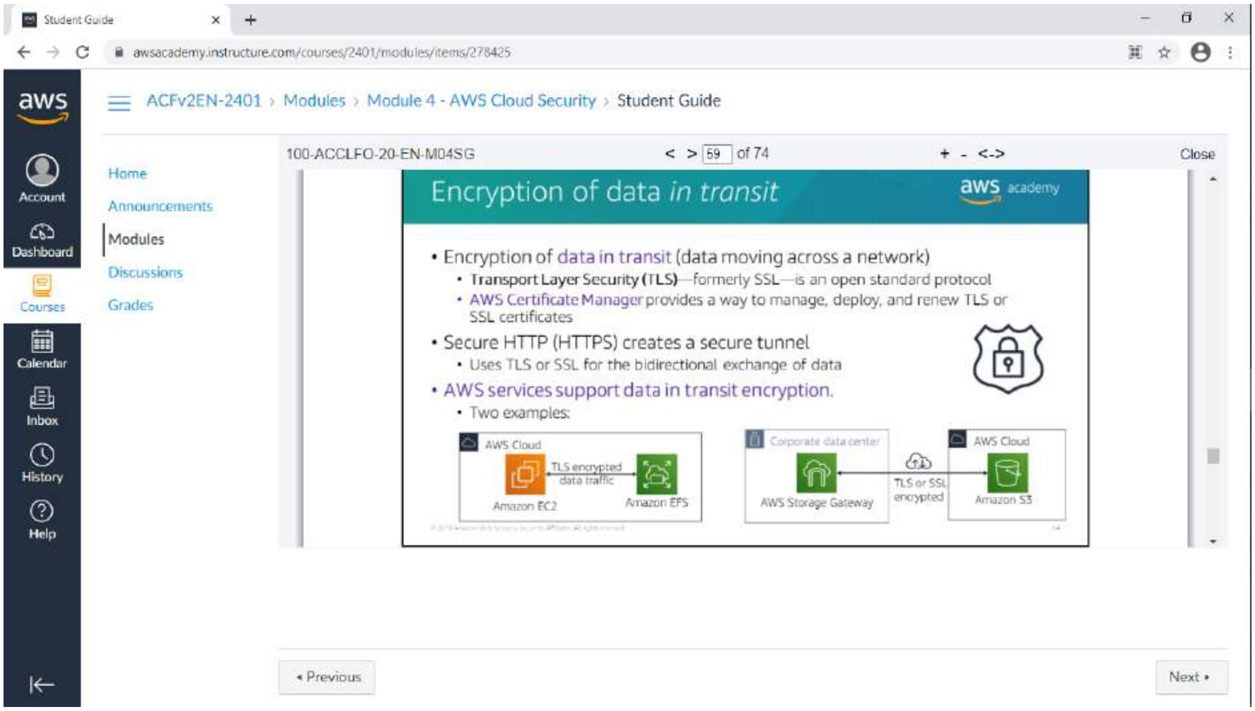
- Encryption encodes data with a secret key, which makes it unreadable
  - Only those who have the secret key can decode the data
  - AWS KMS can manage your secret keys
- AWS supports encryption of data at rest
  - Data at rest = Data stored physically (on disk or on tape)
  - You can encrypt data stored in any service that is supported by AWS KMS, including:
    - Amazon S3
    - Amazon EBS
    - Amazon Elastic File System (Amazon EFS)
    - Amazon RDS managed databases

Data encryption is an essential tool to use when your objective is to protect digital data. Data encryption takes data that is legible and encodes it so that it is unreadable to anyone who does not have access to the secret key that can be used to decode it. Thus, even if an attacker gains access to your data, they cannot make sense of it.

Data at rest refers to data that is physically stored on disk or on tape.

You can create encrypted file systems on AWS so that all your data and metadata is encrypted at rest by using the open standard Advanced Encryption Standard (AES)-256 encryption algorithm. When you use AWS KMS, encryption and decryption are handled automatically and transparently, so that you do not need to modify your applications. If your organization is subject to corporate or regulatory policies that require encryption of data and metadata at rest, AWS recommends enabling encryption on all services that store your data.

You can encrypt data stored in any service that is supported by AWS KMS. See How AWS Services use AWS KMS for a list of supported services.



Encryption of data *in transit*

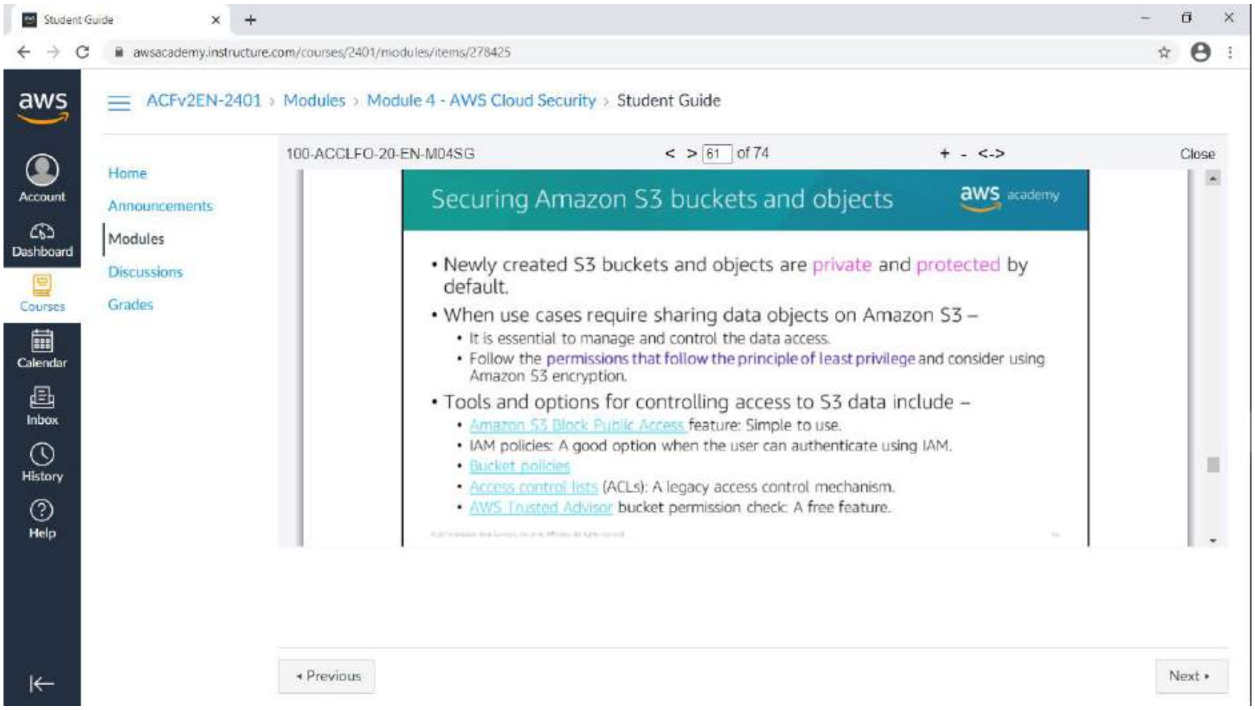
- Encryption of data in transit (data moving across a network)
  - **Transport Layer Security (TLS)**—formerly SSL—is an open standard protocol
  - AWS Certificate Manager provides a way to manage, deploy, and renew TLS or SSL certificates
- Secure HTTP (HTTPS) creates a secure tunnel
  - Uses TLS or SSL for the bidirectional exchange of data
- AWS services support data in transit encryption.
  - Two examples:

Data in transit
refers to data that is moving across the network. Encryption of data in
transit
is accomplished by using Transport Layer Security (
TLS) 1.2 with an open standard AES-256
cipher. TLS was formerly called Secure Sockets Layer
(SSL).
AWS Certificate Manager
is a service that enables you to provision, manage,
and deploy SSL
or TLS certificates for use with AWS services and your internal
connected resources. SSL or
TLS certificates are used to secure network communications and
establish the identity of
websites over the internet, and also resources on p
rivate networks. With AWS Certificate
Manager, you can request a certificate and then dep
loy it on AWS resources (such as load
balancers or CloudFront distributions). AWS Certificate Manager also
handles certificate
renewals.

Web traffic that runs over HTTP is not secure. However, traffic that runs over Secure HTTP (HTTPS) is encrypted by using TLS or SSL. HTTPS traffic is protected against eavesdropping and man-in-the-middle attacks because of the bidirectional encryption of the communication.

AWS services support encryption for data in transit . Two examples of encryption for data in transit are shown. The first example shows an EC2 instance that has mounted an Amazon EFS shared file system. All data traffic between the instance and Amazon EFS is encrypted by using TLS or SSL. For further details about this configuration, see Encryption of EFS Data in Transit

The second example shows the use of AWS Storage Gateway, a hybrid cloud storage service that provides on-premises access to AWS Cloud storage. In this example, the storage gateway is connected across the internet to Amazon S3, and the connection encrypts the data in transit.



Securing Amazon S3 buckets and objects

- Newly created S3 buckets and objects are private and protected by default.
- When use cases require sharing data objects on Amazon S3 –
  - It is essential to manage and control the data access.
  - Follow the permissions that follow the principle of least privilege and consider using Amazon S3 encryption.
- Tools and options for controlling access to S3 data include –
  - Amazon S3 Block Public Access feature: Simple to use.
  - IAM policies: A good option when the user can authenticate using IAM.
  - Bucket policies
  - Access control lists (ACLs): A legacy access control mechanism.
  - AWS Trusted Advisor bucket permission check: A free feature.

By default, all Amazon S3 buckets are private and can be accessed only by users who are explicitly granted access. It is essential to manage and control access to Amazon S3 data.

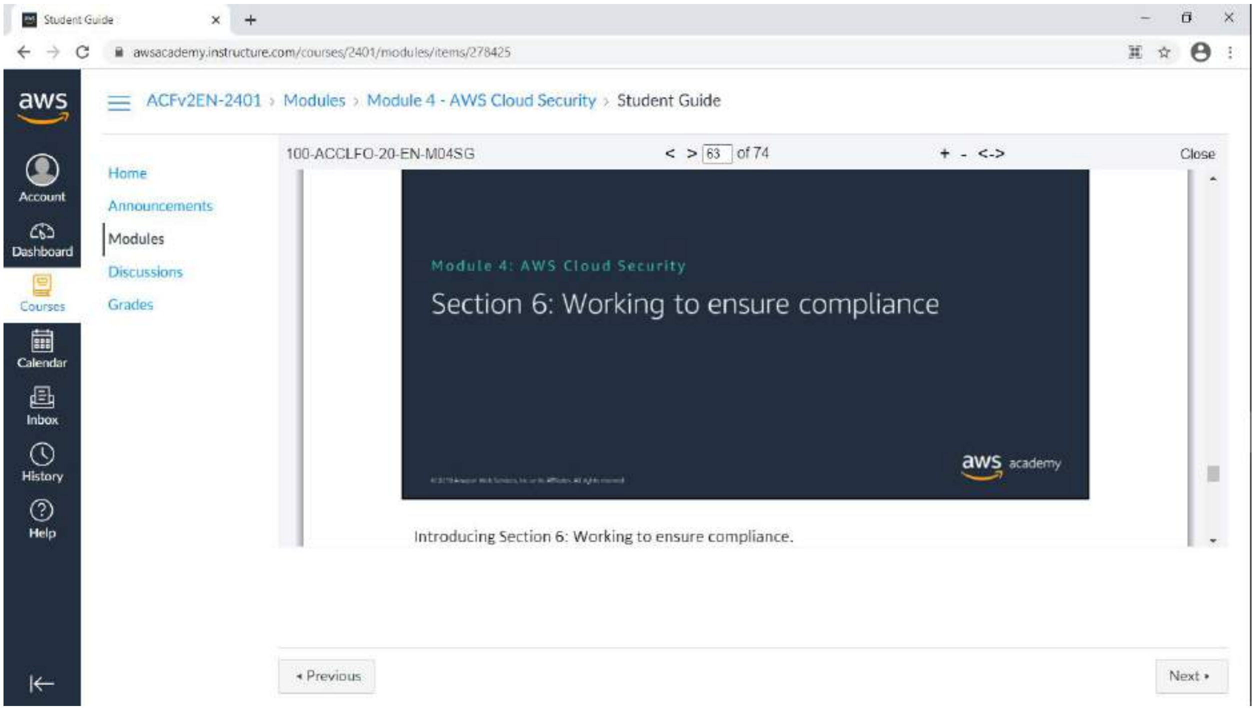
AWS provides many tools and options for controlling access to your S3 buckets or objects, including:

- Using Amazon S3 Block Public Access. These settings override any other policies or object permissions. Enable Block Public Access for all buckets that you don't want to be publicly accessible. This feature provides a straightforward method for avoiding unintended exposure of Amazon S3 data.
- Writing IAM policies that specify the users or roles that can access specific buckets and objects. This method was discussed in detail earlier in this module.
- Writing bucket policies that define access to specific buckets or objects. This option is typically used when the user or system cannot authenticate by using IAM. Bucket policies can be configured to grant access across AWS accounts or to grant public or anonymous access to Amazon S3 data. If bucket policies are used, they should be written carefully and tested fully. You can specify a deny statement in a bucket policy to restrict access. Access

will be restricted even if the users have permissions that are granted in an identity-based
policy that is attached to the users.
61

Setting
access control lists (ACLs)
on your buckets and objects. ACLs are less commonly
used (ACLs predate IAM). If you do use ACLs, do not
 set access that is too open or
permissive.

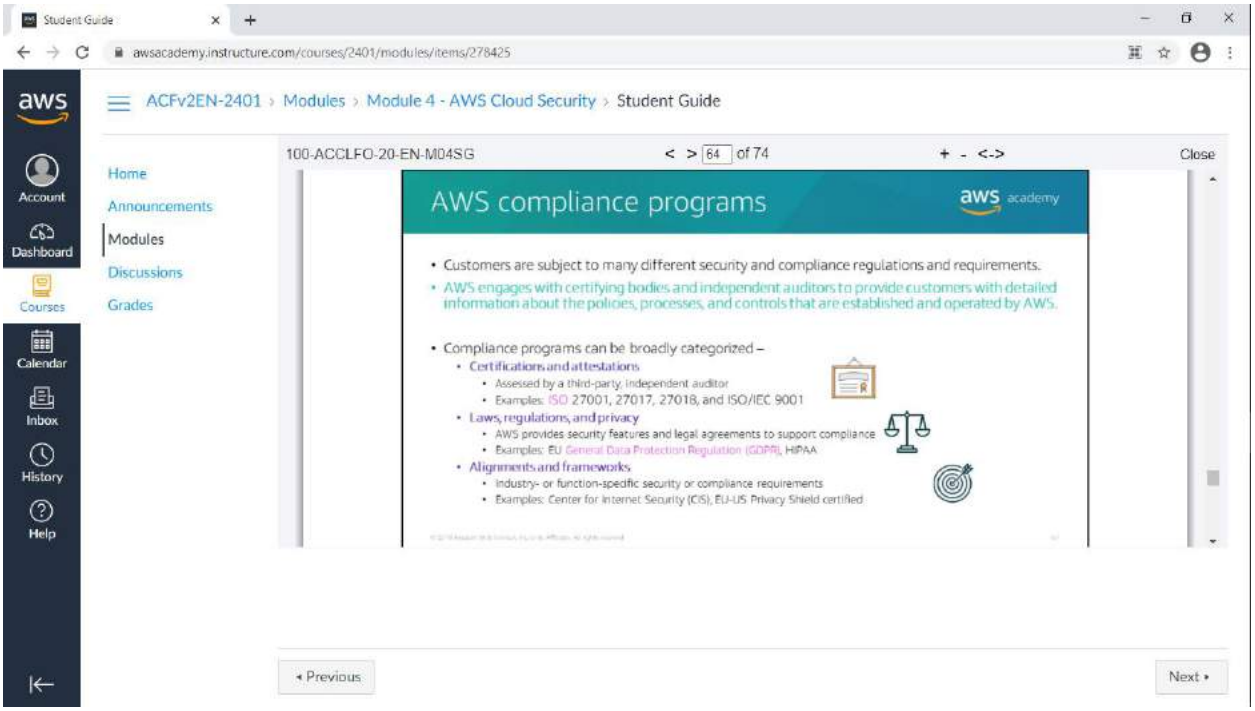
- AWS Trusted Advisor
provides a bucket permission check feature that is
a useful tool for
discovering if any of the buckets in your account have permissions that grant global access.



Module 4: AWS Cloud Security

Section 6: Working to ensure compliance

Introducing Section 6: Working to ensure compliance.



AWS compliance programs

- Customers are subject to many different security and compliance regulations and requirements.
- AWS engages with certifying bodies and independent auditors to provide customers with detailed information about the policies, processes, and controls that are established and operated by AWS.

- Compliance programs can be broadly categorized –
  - Certifications and attestations
    - Assessed by a third-party, independent auditor
    - Examples: ISO 27001, 27017, 27018, and ISO/IEC 9001
  - Laws, regulations, and privacy
    - AWS provides security features and legal agreements to support compliance
    - Examples: EU General Data Protection Regulation (GDPR), HIPAA
  - Alignments and frameworks
    - Industry- or function-specific security or compliance requirements
    - Examples: Center for Internet Security (CIS), EU-US Privacy Shield certified

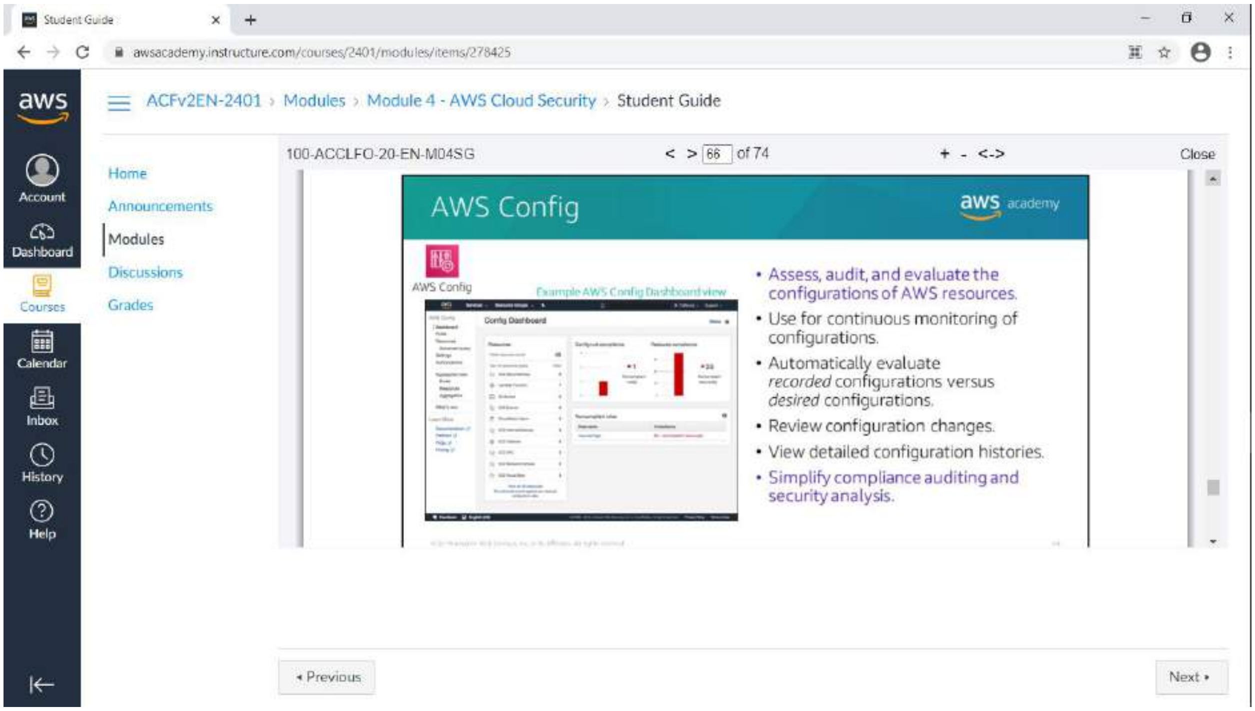
AWS engages with external certifying bodies and independent auditors to provide customers with information about the policies, processes, and controls that are established and operated by AWS.

A full Listing of AWS Compliance Programs is available. Also, for details about which AWS services are in scope of AWS assurance programs, see AWS Services in Scope by Compliance Program

.

As an example of a certification for which you can use AWS services to meet your compliance goals, consider the ISO/IEC 27001:2013 certification. It specifies the requirements for establishing, implementing, maintaining, and continually improving an Information Security Management System. The basis of this certification is the development and implementation of a rigorous security program, which includes the development and implementation of an Information Security Management System. The Information Security Management System defines how AWS perpetually manages security in a holistic, comprehensive manner.

AWS also provides security features and legal agreements that are designed to help support customers with common regulations and laws. One example is the Health Insurance Portability and Accountability Act (HIPAA) regulation. Another example, the European Union (EU) General Data Protection Regulation (GDPR) protects European Union data subjects' fundamental right to privacy and the protection of personal data. It introduces robust requirements that will raise and harmonize standards for data protection, security, and compliance. The GDPR Center contains many resources to help customers meet their compliance requirements with this regulation.

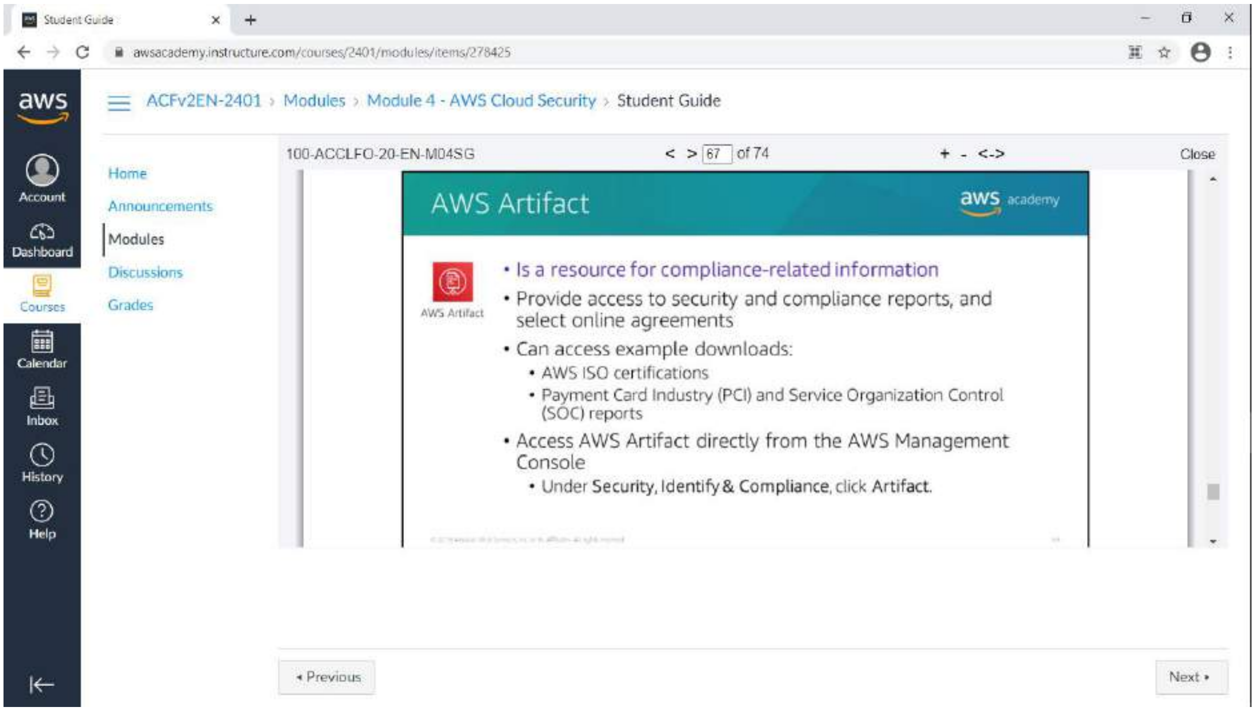


AWS Config

- Assess, audit, and evaluate the configurations of AWS resources.
- Use for continuous monitoring of configurations.
- Automatically evaluate *recorded* configurations versus *desired* configurations.
- Review configuration changes.
- View detailed configuration histories.
- Simplify compliance auditing and security analysis.

AWS Config
is a service that enables you to assess, audit, and
 evaluate the configurations of
your AWS resources. AWS Config continuously monitors and records your AWS resource
configurations, and it enables you to automate the
evaluation of recorded configurations
against desired configurations. With AWS Config, you can review changes in configurations
and relationships between AWS resources, review detailed resource configuration histories,
and determine your overall compliance against the configurations that are specified in your
internal guidelines. This enables you to simplify compliance auditing, security analysis,
change management, and operational troubleshooting.
As you can see in the AWS Config Dashboard screen capture shown here, AWS Config keeps
an inventory listing of all resources that exist in
 the account, and it then checks for

configuration rule compliance and resource compliance. Resources that are found to be
noncompliant are flagged, which alerts you to the configuration issues that should be
addressed within the account.
AWS Config is a Regional service. To track resources
across Regions, enable it in every Region
that you use. AWS Config offers an aggregator feature that can show an aggregated view of resources across multiple Regions and even multiple accounts.

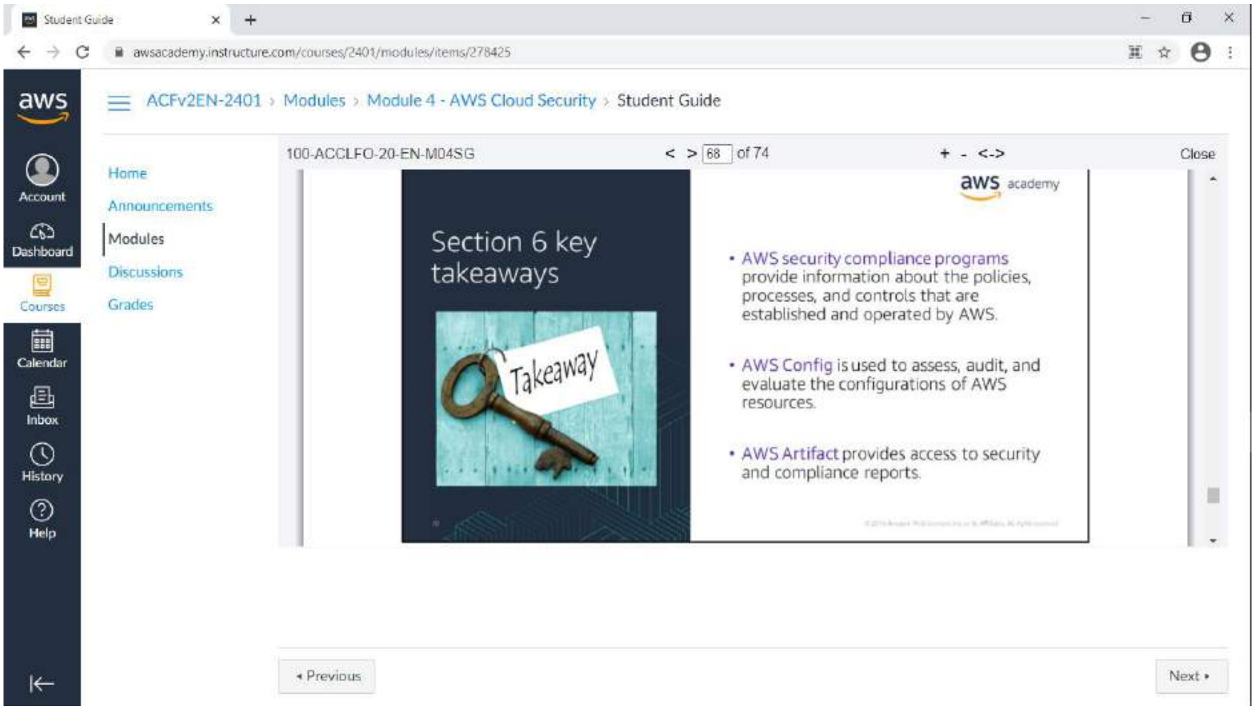


AWS Artifact

- Is a resource for compliance-related information
- Provide access to security and compliance reports, and select online agreements
- Can access example downloads:
  - AWS ISO certifications
  - Payment Card Industry (PCI) and Service Organization Control (SOC) reports
- Access AWS Artifact directly from the AWS Management Console
  - Under **Security, Identify & Compliance**, click **Artifact**.

AWS Artifact provides on-demand downloads of AWS security and compliance documents, such as AWS ISO certifications, Payment Card Industry (PCI), and Service Organization Control (SOC) reports. You can submit the security and compliance documents (also known as audit artifacts) to your auditors or regulators to demonstrate the security and compliance of the AWS infrastructure and services that you use. You can also use these documents as guidelines to evaluate your own cloud architecture and assess the effectiveness of your company's internal controls. AWS Artifact provides documents about AWS only. AWS customers are responsible for developing or obtaining documents that demonstrate the security and compliance of their companies.

You can also use AWS Artifact to review, accept, and track the status of AWS agreements such as the Business Associate Agreement (BAA). A BAA typically is required for companies that are subject to HIPAA to ensure that protected health information (PHI) is appropriately safeguarded. With AWS Artifact, you can accept agreements with AWS and designate AWS accounts that can legally process restricted information. You can accept an agreement on behalf of multiple accounts. To accept agreements for multiple accounts, use AWS Organizations to create an organization. To learn more, see Managing agreements in AWS Artifact



Section 6 key takeaways

- AWS security compliance programs provide information about the policies, processes, and controls that are established and operated by AWS.
- AWS Config is used to assess, audit, and evaluate the configurations of AWS resources.
- AWS Artifact provides access to security and compliance reports.

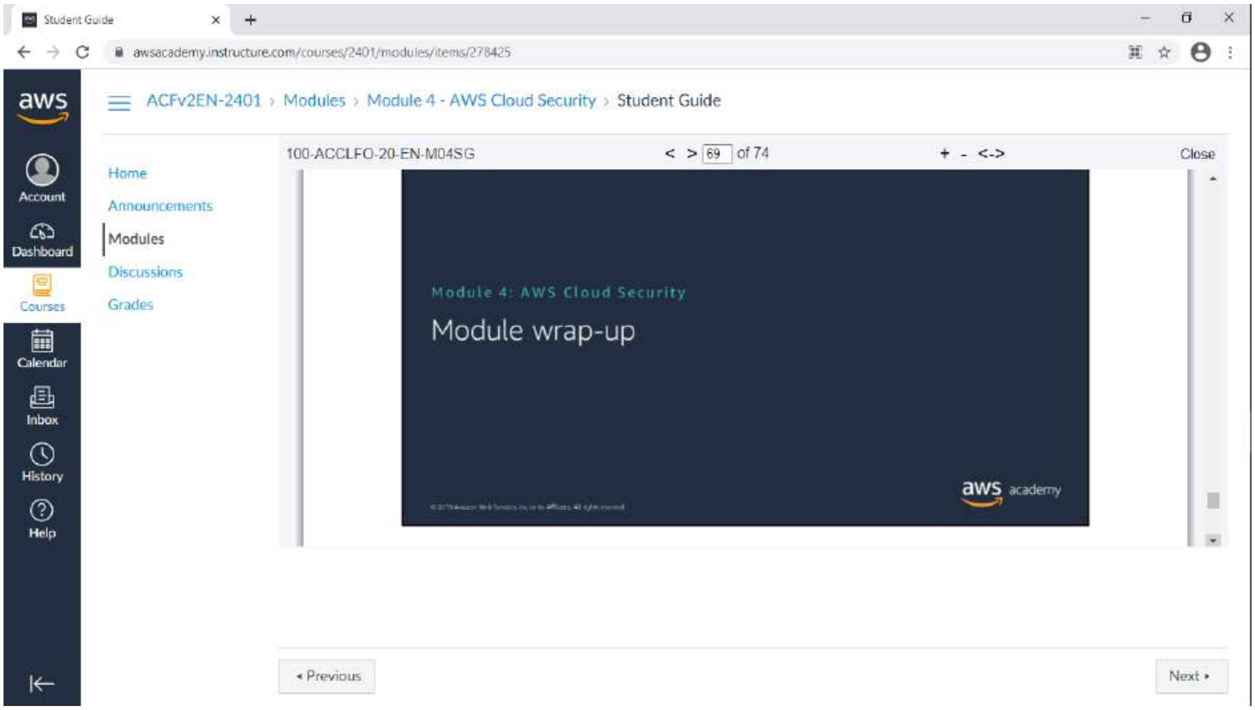
Some key takeaways from this section of the module include:

- AWS security compliance programs provide information about the policies, processes, and controls that are established and operated by AWS.
-

AWS Config is used to assess, audit, and evaluate the configurations of AWS resources.

•

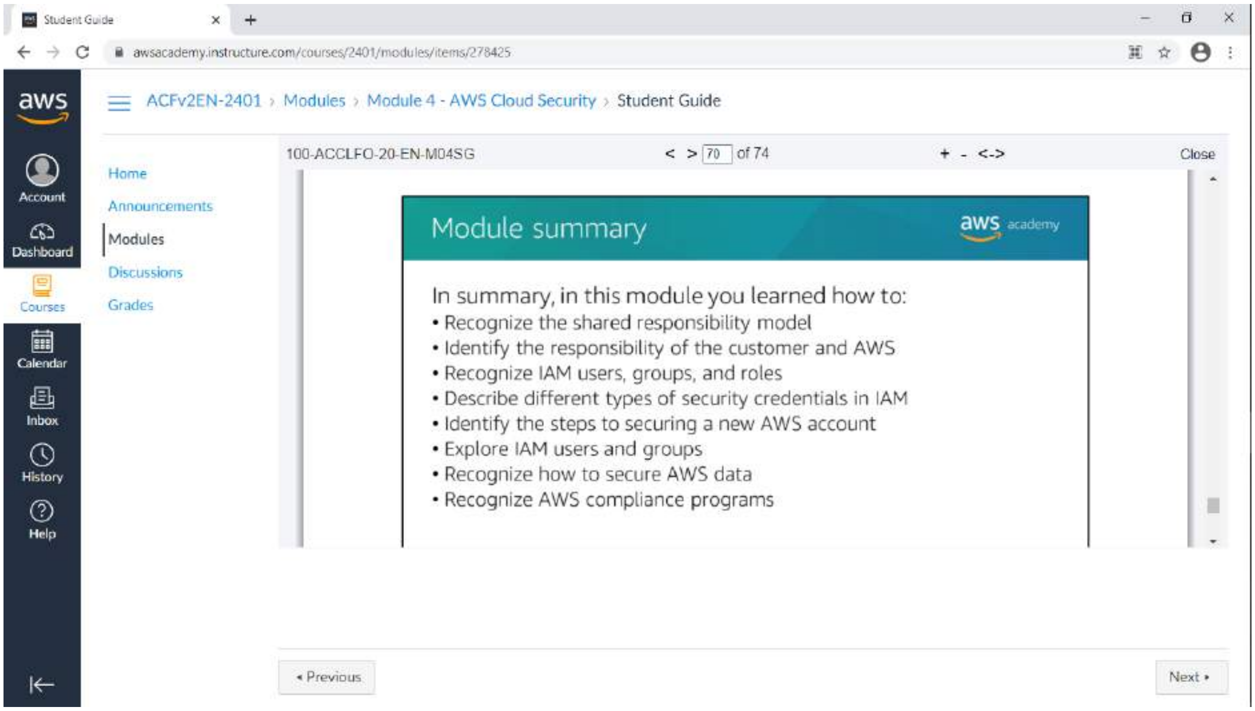
AWS Artifact provides access to security and compliance reports.



Module 4: AWS Cloud Security

Module wrap-up

It’s now time to review the module and wrap up with a knowledge check and discussion of a practice certification exam question.



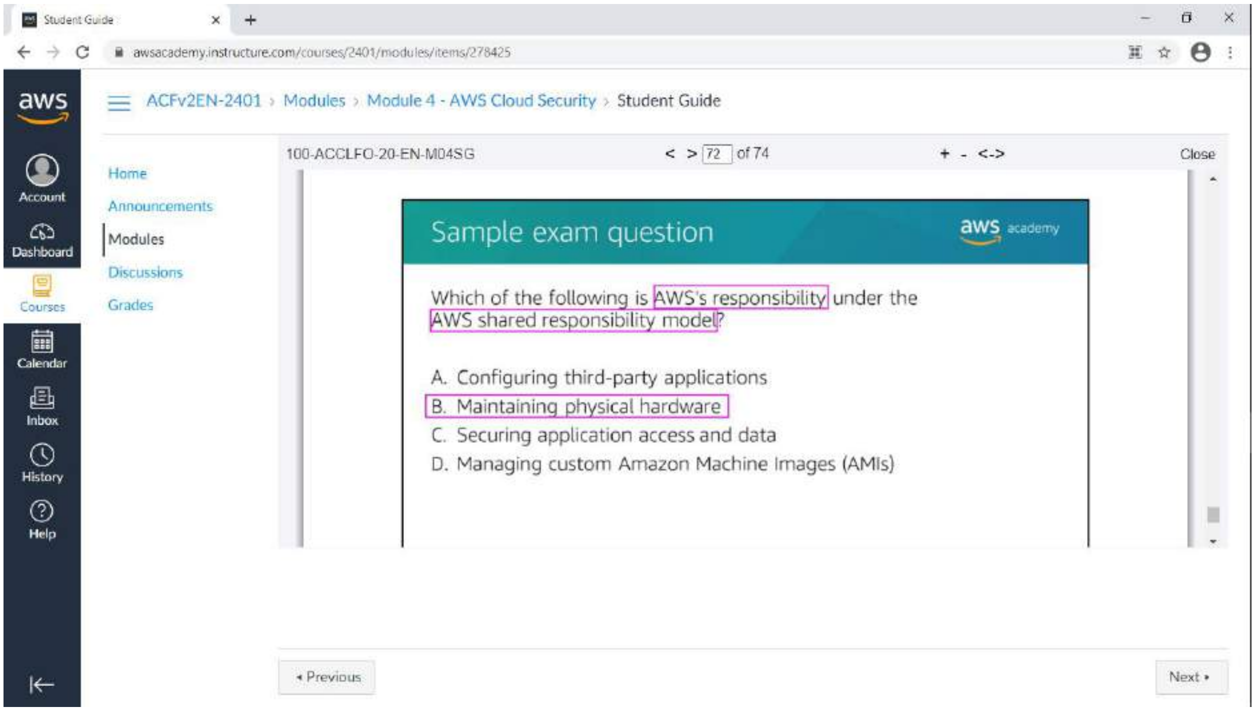
Module summary

In summary, in this module you learned how to:

- Recognize the shared responsibility model
- Identify the responsibility of the customer and AWS
- Recognize IAM users, groups, and roles
- Describe different types of security credentials in IAM
- Identify the steps to securing a new AWS account
- Explore IAM users and groups
- Recognize how to secure AWS data
- Recognize AWS compliance programs

In summary, in this module you learned how to:

- Recognize the shared responsibility model
- Identify the responsibility of the customer and AWS
- Recognize IAM users, groups, and roles
- Describe different types of security credentials in IAM
- Identify the steps to securing a new AWS account
- Explore IAM users and groups
- Recognize how to secure AWS data
- Recognize AWS compliance programs

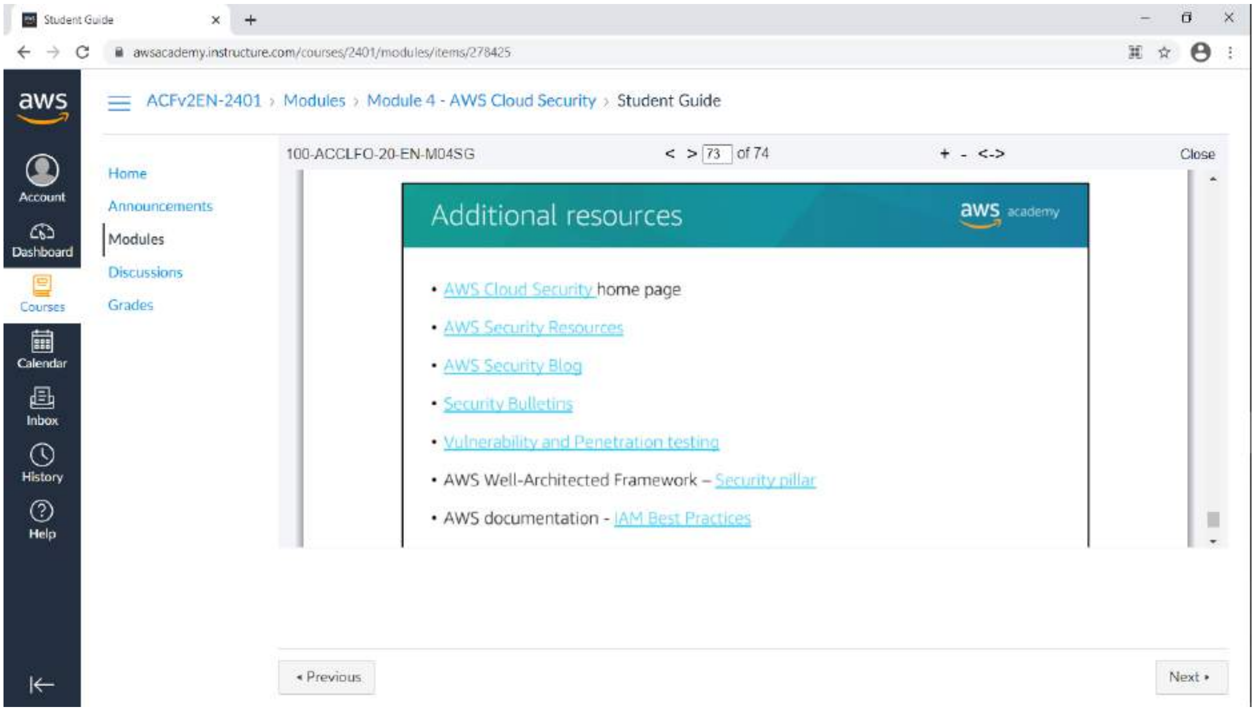


Sample exam question

Which of the following is AWS's responsibility under the AWS shared responsibility model?

A. Configuring third-party applications
B. Maintaining physical hardware
C. Securing application access and data
D. Managing custom Amazon Machine Images (AMIs)

Look at the answer choices and rule them out based on the keywords that were previously
highlighted.
This sample exam question comes from the AWS Certified Cloud Practitioner sample exam
questions document that is linked to from the main
AWS Certified Cloud Practitioner exam
information page



Additional resources

- AWS Cloud Security home page
- AWS Security Resources
- AWS Security Blog
- Security Bulletins
- Vulnerability and Penetration testing
- AWS Well-Architected Framework – Security pillar
- AWS documentation - IAM Best Practices

Security is a large topic and this module has only
provided an introduction to the subject. The
following resources provide more detail:

•

The
AWS Cloud Security
home page – Provides links to many security resources

.

•

AWS Security Resources

.

•

AWS Security Blog

.

•

Security Bulletins
notify the customer about the latest security and privacy events with
AWS services.

•

The Vulnerability and Penetration testing
page – Describes which types of testing are

permitted without prior approval, which types of testing require approval, and which types of testing are prohibited.

- AWS Well-Architected Framework – Security pillar
- .

AWS documentation – IAM Best Practices