



Architecting on AWS

Student Guide

Version 7.10.1

200-ARCHIT-710-EN-SG

© 2025, 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, feedback, or other questions? Contact us at
<https://support.aws.amazon.com/#/contacts/aws-training>.

All trademarks are the property of their owners.

Contents

Course Introduction	4
Module 1: Architecting Fundamentals	26
Module 2: Account Security	67
Module 3: Networking 1	114
Module 4: Compute	160
Module 5: Storage	218
Module 6: Database Services	275
Module 7: Monitoring and Scaling	330
Module 8: Automation	393
Module 9: Containers	425
Module 10: Networking 2	461
Module 11: Serverless	499
Module 12: Edge Services	546
Module 13: Backup and Recovery	602
Course Summary	649





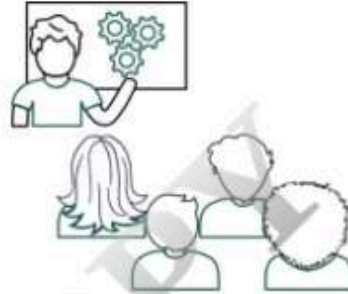
Optimize your learning

- Rest during breaks. ☕
- Have fun. 🎉
- Participate. 🗣️
- Ask questions. ❓
- Make your learning important. !
- Do the labs. 🔬

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Logistics

- Breaks and lunch
- Security
- Cell phones
- Virtual classroom features
 - Audio
 - Chat
 - Raise hand



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

7

Prerequisites

We recommend that course attendees complete one of the following prerequisites:

1. AWS Cloud Practitioner Essentials
2. AWS Technical Essentials
3. Build a working knowledge of:
 - Distributed systems
 - Networking concepts
 - IP addressing
 - Multi-tier architectures
 - Cloud computing concepts



AWS Cloud Practitioner
Essentials course



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

8

If you have this working knowledge and have not taken AWS Cloud Practitioner Essentials, familiarize yourself with basic cloud services from Amazon Web Services (AWS). For more information, see AWS Cloud Practitioner Essentials in AWS Skill Builder at

<https://explore.skillbuilder.aws/learn/course/external/view/elearning/134/aws-cloud-practitioner-essentials>.

Register for access to guides and lab environments

Make sure you register for AWS Builder Labs.

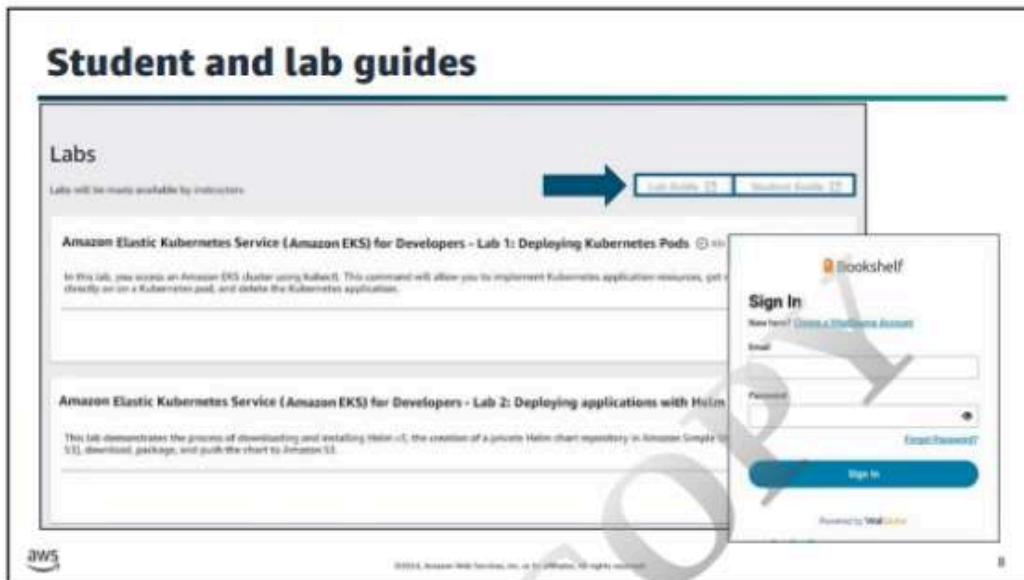
- Refer to your welcome email for registration information.



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

7

Check your inbox for a welcome email from your instructor. In this email, you will find your unique student registration URL for the class. Use this URL link to create an account or log in to your existing AWS Builder Labs account. In AWS Builder Labs, you can access your lab environments, Lab Guide, and Student Guide.



At this time, you should be logged in to AWS Builder Labs. From here, you can access your Lab Guide and Student Guide, which are located in eVantage Bookshelf (VitalSource). Buttons to the Lab Guide and Student Guide are located at the top-right corner of your AWS Builder Labs dashboard. The labs and buttons will be inactive until the start of the class.

After the class starts, choose either button to access your guides. You will be prompted to log in with your existing eVantage Bookshelf (VitalSource) account or to create a new account. After you are logged in to eVantage Bookshelf (VitalSource), you will have access to the Student Guide and Lab Guide for the class. You can access your guides online or download them. Use these guides to follow along with the course and as a reference after the training.

Lab requirements

- Computer running:
 - Windows
 - macOS
 - Linux: Ubuntu, SUSE, or Red Hat
- Recommended web browsers:
 - Google Chrome
 - Mozilla Firefox
 - Microsoft Edge
- Reliable internet connection able to browse the internet by using HTTPS
- Register for AWS Builder Labs:
 - Turn off ad blockers and script blockers



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

10

About you

Tell us the following:

- First name
- Organization and role
- What do you expect of this course?



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

11



Course objective


- Identify Amazon Web Services (AWS) services, compare features, and explore best practices to architect resilient, secure, and highly available IT solutions on AWS.




© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

13


Day 1 agenda	
Section title	Estimated time
Module 1: Architecting Fundamentals	45 minutes
Lab 1: Explore and Interact with the AWS Management Console and AWS Command Line Interface	35 minutes
Module 2: Account Security	60 minutes
Lunch	60 minutes
Module 3: Networking 1	60 minutes
Module 4: Compute	75 minutes
Lab 2: Build your Amazon VPC Infrastructure	45 minutes

 © 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. 14



Day 2 agenda	
Section title	Estimated time
Module 5: Storage	70 minutes
Module 6: Database Services	70 minutes
Lab 3: Create a Database Layer in Your Amazon VPC Infrastructure	45 minutes
Lunch	60 minutes
Module 7: Monitoring and Scaling	70 minutes
Lab 4: Configure High Availability in Your Amazon VPC	45 minutes
Module 8: Automation	30 minutes
Module 9: Containers	40 minutes

 © 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. 15

Day 3 agenda	
Section title	Estimated time
Module 10: Networking 2	45 minutes
Module 11: Serverless	45 minutes
Lab 5: Build a Serverless Architecture	45 minutes
Module 12: Edge Services	60 minutes
Lunch	60 minutes
Lab 6: Configure an Amazon CloudFront Distribution with an Amazon S3 Origin	60 minutes
Module 13: Backup and Recovery	40 minutes
Capstone Lab: Build an AWS Multi-Tier Architecture	90 minutes
Course Summary	10 minutes

 © 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. 16

Capstone lab



- Review and analyze architectural solutions based on project data, best practices, and the AWS Well-Architected Framework.
- Design the architecture in a lab, without specific guidance.

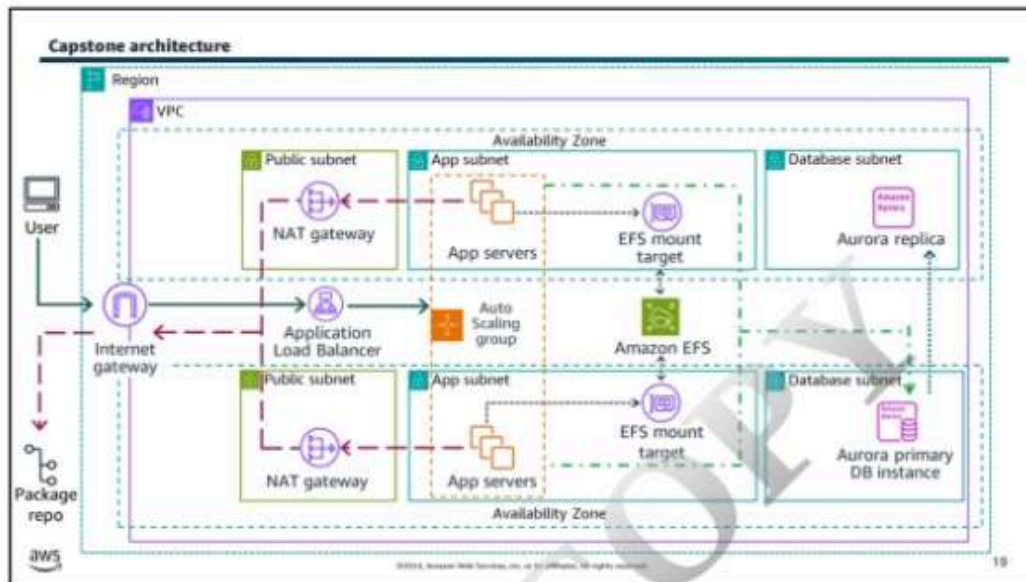
© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

18

The capstone lab is the final project for this course.

During the lab, you are provided with a scenario that discusses a business need. Review the requirements and use what you have learned in this course to complete the list of tasks.

You learn more about the capstone lab at the end of this course.



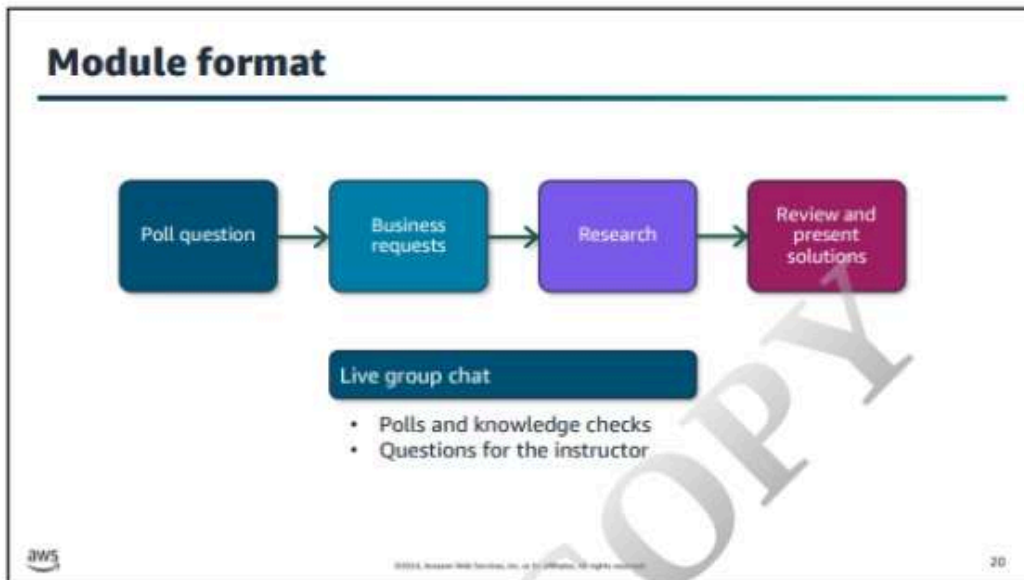
In the capstone lab, you build this multi-tier architecture.

During check-ins, you discuss specific services that are outlined here and how they interact with each other.

Note: The diagram on this slide contains several abbreviations. The following list includes some services that are not covered in this module, but are included for completeness:

- EFS = Amazon Elastic File Service (Amazon EFS)
- Aurora = Amazon Aurora
- NAT = network address translation
- VPC = Amazon Virtual Private Cloud (Amazon VPC)

[Image description: The diagram shows a single AWS Region with one virtual private cloud (VPC) and two Availability Zones. Each Availability Zone contains a public subnet, an app subnet, and a database subnet. A unidirectional arrow outside the VPC routes from the remote user through an internet gateway, to an Application Load Balancer. The Application Load Balancer is depicted as not residing in a subnet or an Availability Zone, though it does reside in both public subnets. The arrow continues to the Auto Scaling group that has app servers in the app subnets of both Availability Zones. Each app server communicates with an Amazon EFS mount target in its own subnet to reach the EFS file system, which does not reside inside the VPC and is shown this way for simplicity. All app servers communicate with an Amazon Aurora primary DB instance in one of the database subnets. The other database subnet holds the Aurora replica. A unidirectional arrow points between the Aurora primary and Aurora replica. A unidirectional arrow points from the app servers and travels through each Availability Zone's NAT gateway. The NAT gateways are in the public subnet of each Availability Zone. The arrows travel through each NAT gateway, and exit the VPC through the internet gateway and reach the remote package repository. Note: The arrows in the diagram are shown as unidirectional to indicate where the request originated. Assume that the response from the target travels the path in reverse unless otherwise indicated. **End description.]**



Each module starts with a check-in poll question. Then you are introduced to a stakeholder at the beginning of each module. The stakeholder has brought business requests to you. Their questions have informed you of what you should research to support Example Corp. in their cloud journey. Your instructor prepares you to present solutions to the stakeholder by teaching you about services and best practices for building on AWS.

At the end of each module, your instructor asks 2–5 questions to help you review the topics and services that are covered in the module. In some sections, you also check in on the capstone architecture to see what you have learned that relates to the final capstone lab.

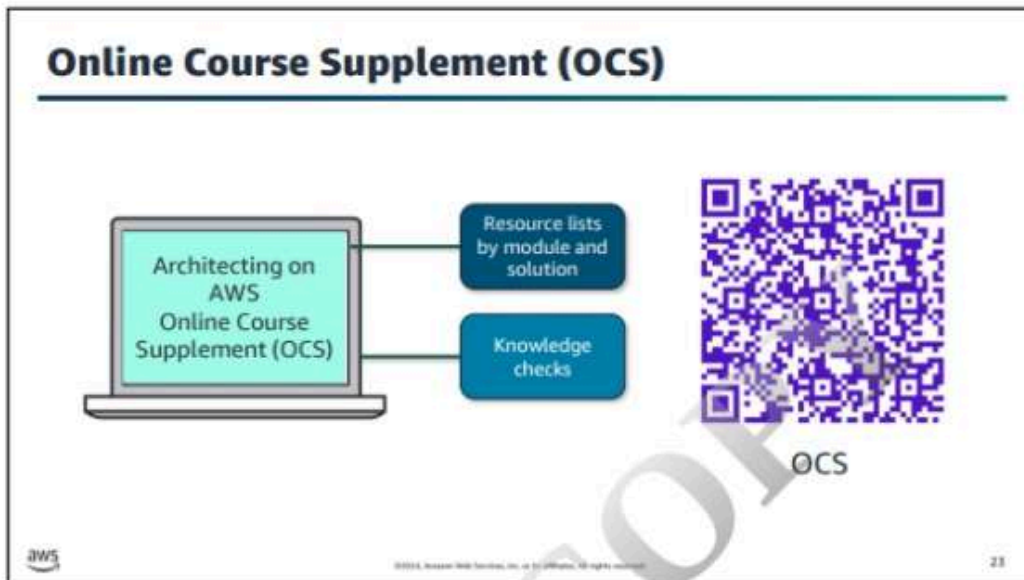
If you are attending virtually, use the live group chat to raise your hand and ask questions.



The business request page at the beginning of the module is structured in this way.

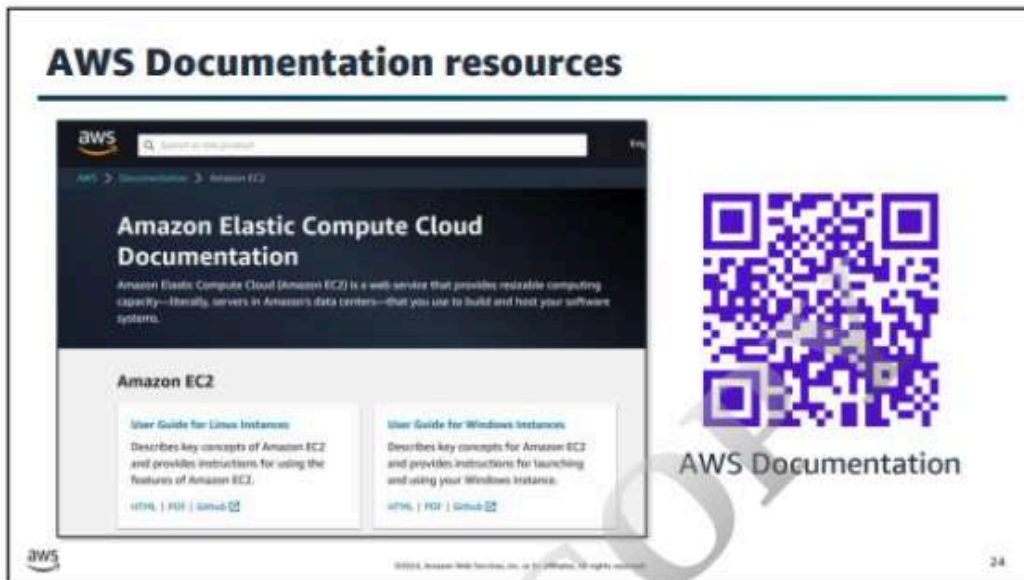
At the end of the module, you review the questions that the stakeholder asked, and you provide solutions to fit their use case.





You can use the Online Course Supplement (OCS) to continue your journey after you complete this course. You can also use it to dive deeper on topics that are not covered in detail in this course.

To find the OCS for this course, see Architecting on AWS – Online Course Supplement on AWS Skill Builder at <https://explore.skillbuilder.aws/learn/course/external/view/elearning/8319/architecting-on-aws-online-course-supplement>.



As part of their role, solutions architects research topics and find additional information about features and services to make decisions. AWS services are constantly improving and evolving. Use AWS Documentation to find user guides, developer guides, API references, tutorials, and more.



AWS Documentation is provided in HTML, PDF, and GitHub.

For more information about user guides, developer guides, API references, and tutorials, see AWS Documentation at <https://docs.aws.amazon.com/>.





Poll question



How far is your organization in its journey to the AWS Cloud?

- A. We're just getting started.
- B. We already have prototypes running.
- C. We have production workloads running.
- D. We run 100 percent of our operations in the AWS Cloud.

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

2

Module overview

- Business request
- Amazon Web Services (AWS) services
- AWS infrastructure
- AWS Well-Architected Framework
- Present solutions
- Knowledge check
- Lab 1: Explore and Interact with the AWS Management Console and AWS Command Line Interface



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

1

Business request



Chief technology officer



The chief technology officer (CTO) wants you to explore the following questions:

- What are the benefits of using AWS services?
- How is the AWS Global Infrastructure organized?
- How can you build your cloud infrastructure according to best practices?

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.


Imagine that you are meeting with your chief technology officer (CTO) as you prepare to build in AWS. As you familiarize yourself with AWS, here are some questions to consider as you navigate this module. During this module, you learn about topics that answer these questions.



The CTO asks during the project meeting, “What are the benefits of using AWS services?” The company is interested in learning about AWS services and tools that would best fit their needs.

Amazon Web Services

- Global data centers
- More than 200 services
- Secure and robust
- Pay as you go
- Built for business needs

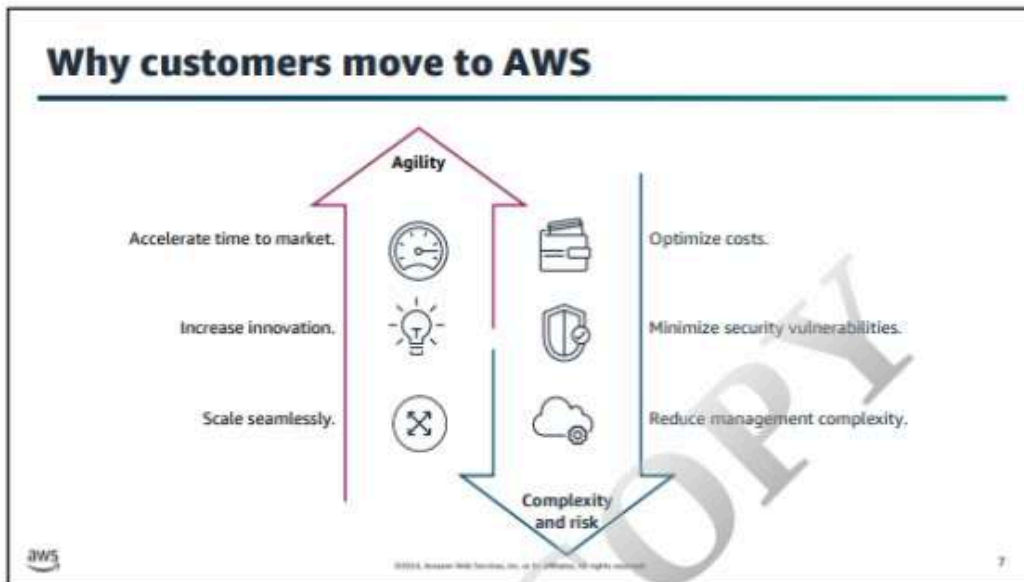


© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS is the world's most comprehensive and adopted cloud solution. AWS offers services such as compute, database, and storage. The AWS pay-as-you-go model, and its security practices, have made AWS the preferred cloud solution for businesses and public organizations.

AWS has been delivering cloud services to customers around the world by running a wide variety of use cases. AWS has the most operational experience of any cloud provider, and at a greater scale. AWS has unmatched experience, reliability, and performance, and an unmatched security record.

Millions of customers, small and large, are using AWS to lower costs, become more agile, and innovate faster. AWS is continually accelerating its pace of innovation to invent new technologies that you can use to transform your business.



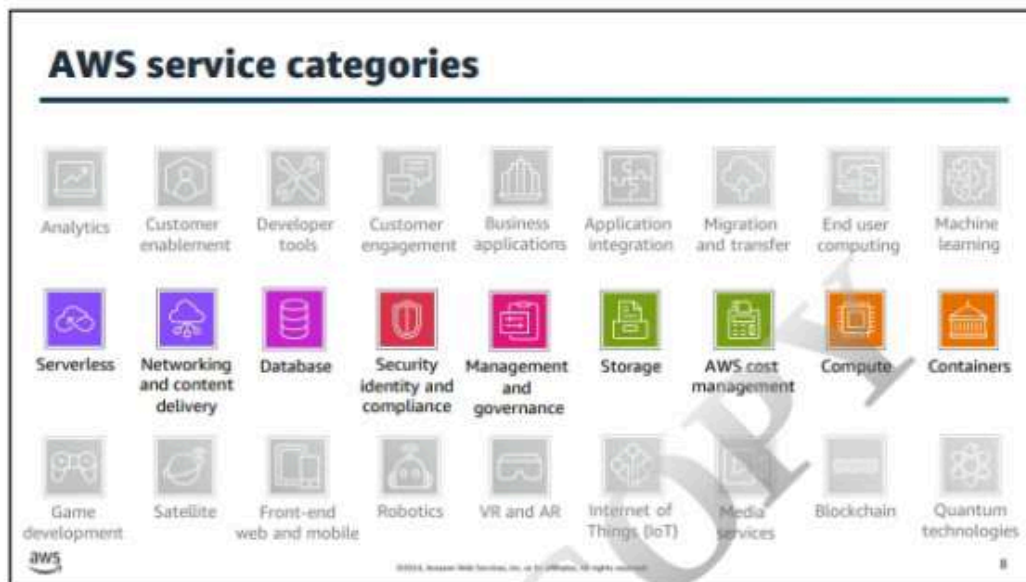
Customers move to AWS to increase agility.

- **Accelerate time to market** – By spending less time acquiring and managing infrastructure, you can focus on developing features that deliver value to your customers.
- **Increase innovation** – You can speed up your digital transformation by using AWS, which provides tools to more easily access the latest technologies and best practices. For example, you can use AWS to develop automations, adopt containerization, and use machine learning.
- **Scale seamlessly** – You can provision additional resources to support new features and scale existing resources up or down to match demand.

Customers also move to AWS to reduce complexity and risk.

- **Optimize costs** – You can reduce costs by paying for only what you use. Instead of paying for on-premises hardware, which you might not use at full capacity, you can pay for compute resources only while you're using them.
- **Minimize security vulnerabilities** – Moving to AWS puts your applications and data behind the advanced physical security of the AWS data centers. With AWS, you have many tools to manage access to your resources.
- **Reduce management complexity** – Using AWS services can reduce the need to maintain physical data centers, perform hardware maintenance, and manage physical infrastructure.

For more information about the advantages of migrating your business to the cloud, see "The future of business is here" at <https://aws.amazon.com/campaigns/migrating-to-the-cloud/>.



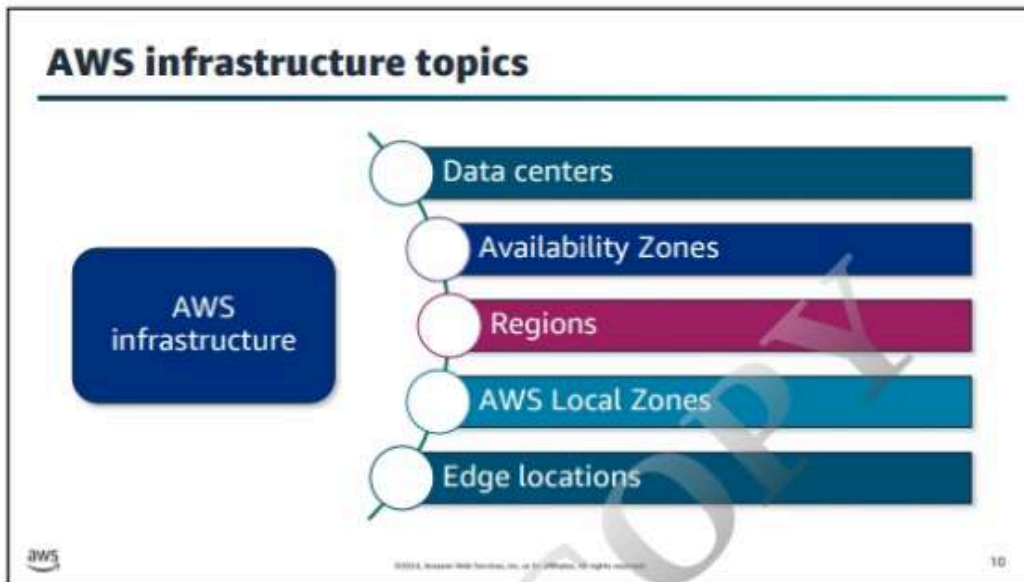
AWS offers a broad set of global cloud-based products. They include compute, storage, database, analytics, networking, mobile, developer tools, management tools, Internet of Things (IoT), security, and enterprise applications. These services help organizations move faster, scale, and lower IT costs. AWS covers infrastructure, foundation, and application services.

This course focuses on the AWS services that are highlighted on this slide. These services are serverless, networking and content delivery, database, security identity and compliance, management and governance, storage, AWS cost management, compute, and containers.

For more information, see AWS Cloud Products at <https://aws.amazon.com/products/>.




The CTO asks during the project meeting, "How is AWS global infrastructure organized?" In this section, you explore the AWS infrastructure.



AWS data centers

- AWS services operate within AWS data centers.
- Data centers host thousands of servers.
- Each location uses AWS proprietary network equipment.
- Data centers are organized into Availability Zones.



The diagram illustrates two data center locations. Each location is represented by a large server rack icon with a security camera on top. To the right of each rack are several smaller server rack icons, representing thousands of servers. The entire diagram is enclosed in a rectangular frame with the AWS logo in the bottom left corner.

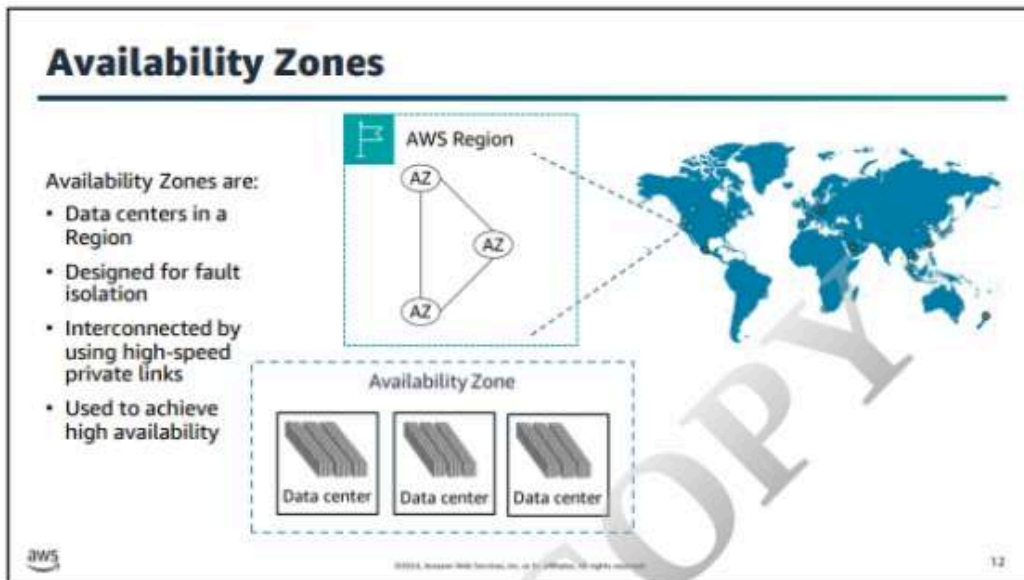
© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

11

AWS pioneered cloud computing in 2006 to provide rapid and secure infrastructure. AWS continuously innovates on the design and systems of data centers to protect them from man-made and natural risks. Today, AWS provides data centers at a large, global scale.

AWS implements controls, builds automated systems, and conducts third-party audits to confirm security and compliance. As a result, the most highly regulated organizations in the world trust AWS every day.

To learn how AWS secures the data centers, see Our Data Centers at <https://aws.amazon.com/compliance/data-center/data-centers/>.



A group of one or more data centers is called an Availability Zone.

An Availability Zone is one or more discrete data centers with redundant power, networking, and connectivity in an AWS Region. When you launch an instance, you can choose an Availability Zone or let AWS choose one for you. If you distribute your instances across multiple Availability Zones and one instance fails, you can design your application so that an instance in another Availability Zone can handle requests.

For more information about Availability Zones, see AWS Global Infrastructure at <https://aws.amazon.com/about-aws/global-infrastructure/>.



Each AWS Region consists of multiple isolated and physically separate Availability Zones within a geographic area. This arrangement achieves the greatest possible fault tolerance and stability. In your account, you determine which Regions you need.

Regions are isolated from each other, and AWS doesn't automatically replicate resources across Regions. Therefore, when you view your resources, you see only the resources that are tied to the Region that you specify in the console.

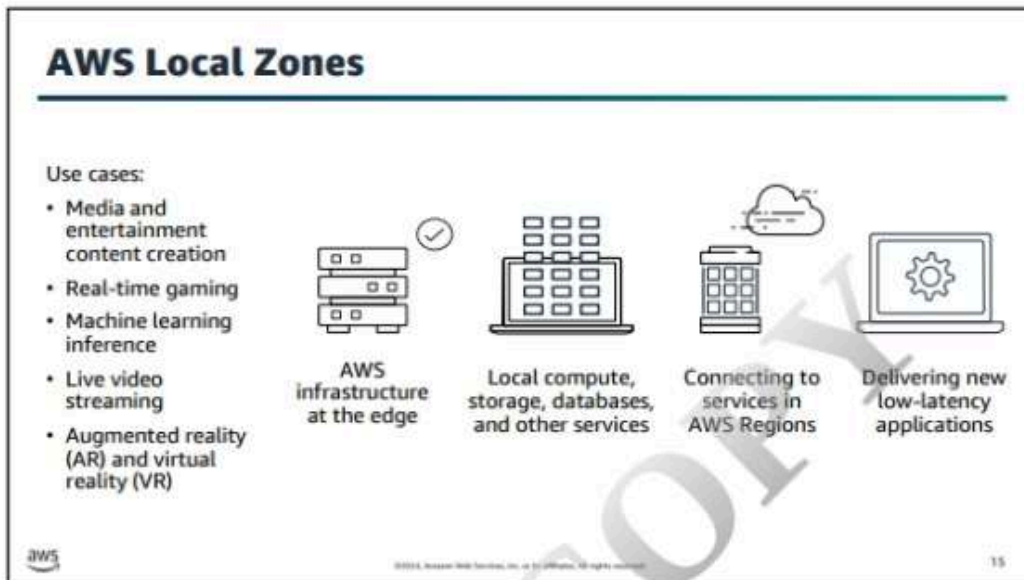
You can run applications and workloads from a Region to reduce latency to end users. In this way, you avoid the upfront expenses, long-term commitments, and scaling challenges that are associated with maintaining and operating a global infrastructure.

For more information about AWS Regions, see Regions and Availability Zones at https://aws.amazon.com/about-aws/global-infrastructure/regions_az/.



Choosing the right Region is important. You must determine the right Region for your services, applications, and data, based on the following factors:

- **Governance and legal requirements** – Consider any legal requirements based on data governance, sovereignty, or privacy laws.
- **Latency** – Close proximity to customers means better performance.
- **Service availability** – Not all AWS services are available in all Regions.
- **Cost** – Different Regions have different costs. Research the pricing for the services that you plan to use and compare costs to make the best decision for your workloads.

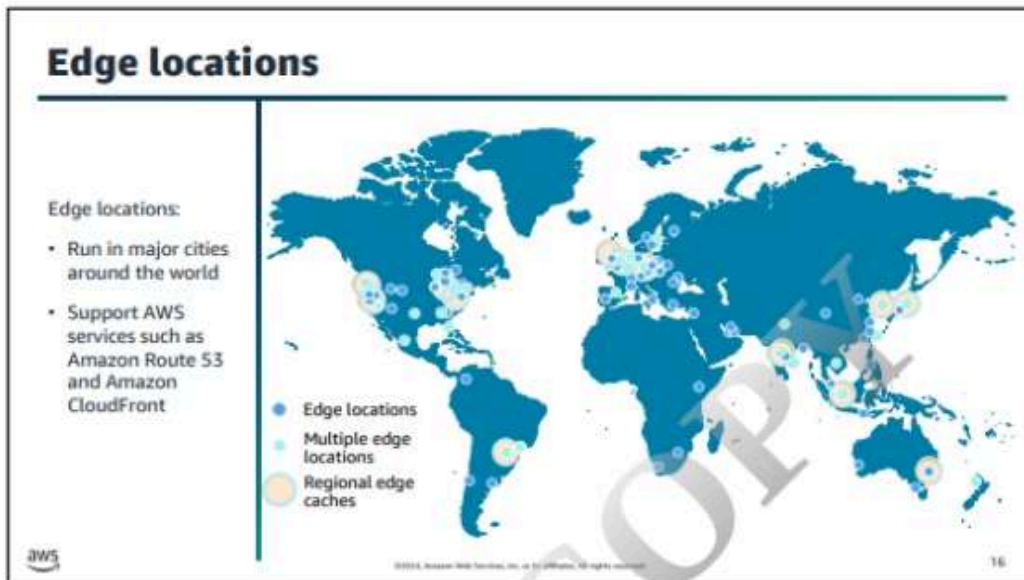


You can use AWS Local Zones for highly demanding applications that require single-digit millisecond latency to end users. Examples include the following:

- Media and entertainment content creation – Includes live production, video editing, and graphics-intensive virtual workstations for artists in geographic proximity
- Real-time multiplayer gaming – Includes real-time multiplayer game sessions, to maintain a reliable gameplay experience
- Machine learning hosting and training – For high-performance, low latency inferencing
- Augmented reality (AR) and virtual reality (VR) – Includes immersive entertainment, data driven insights, and engaging virtual training experiences

Customers can innovate faster because chip designers and verification engineers solve complex, compute-intensive, and latency-sensitive problems by using application and desktop streaming services in AWS Local Zones.

For more information, see AWS Local Zones at <https://aws.amazon.com/about-aws/global-infrastructure/localzones/>.

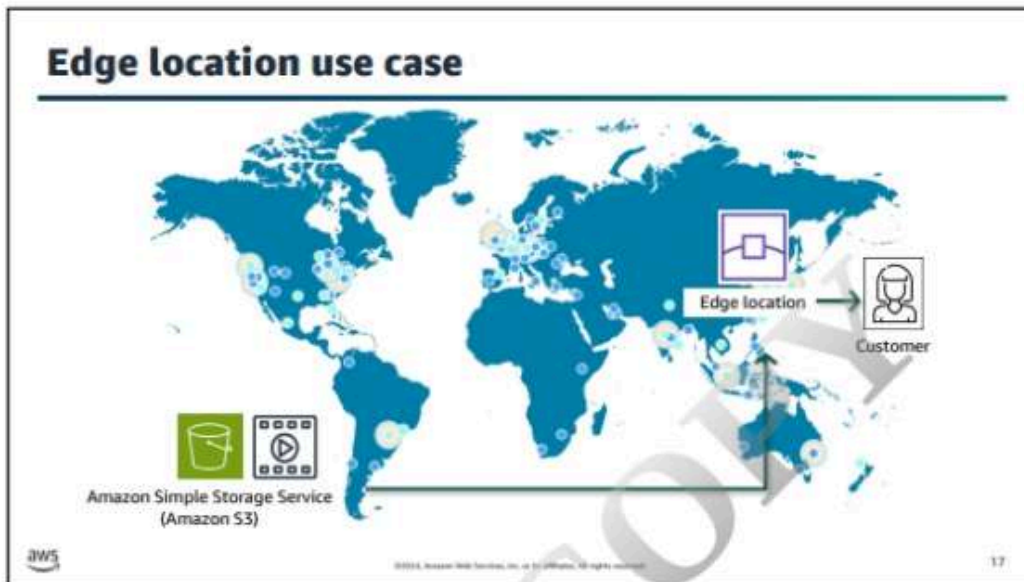


An edge location is the nearest point to a requester of an AWS service. Edge locations are in major cities around the world. They receive requests and cache copies of your content for faster delivery.

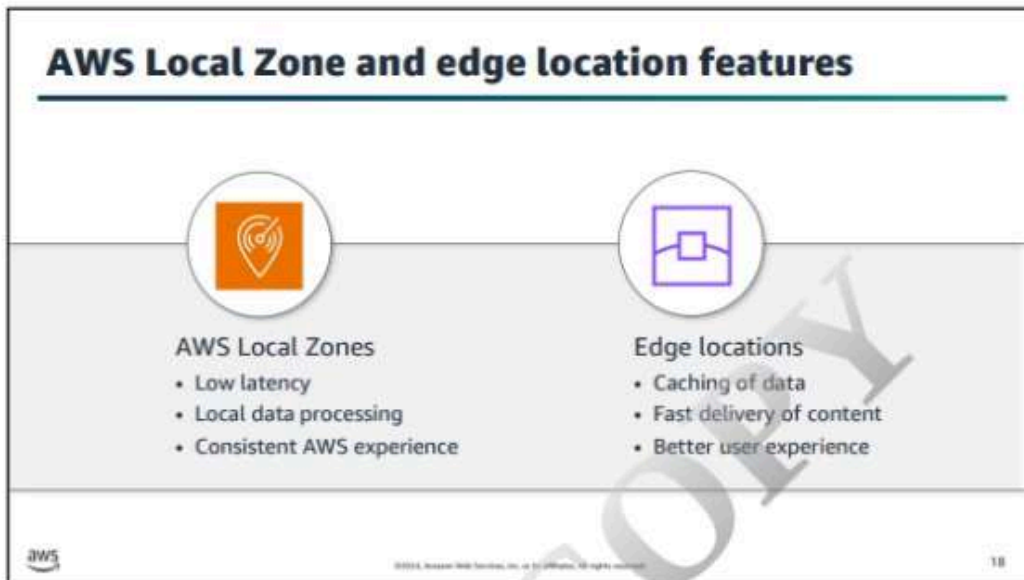
To deliver content to end users with lower latency, you use a global network of edge locations that support AWS services. Amazon CloudFront delivers customer content through a worldwide network of point of presence (PoP) locations, which consist of edge locations and Regional edge cache servers.

You use Regional edge caches, used by default with CloudFront, when you have content that is not accessed frequently enough to remain in an edge location. Regional edge caches absorb this content and provide an alternative to the need to retrieve that content from the origin server.

For more information, see Amazon CloudFront Key Features at <https://aws.amazon.com/cloudfront/features/>.



One common use for edge locations is to serve content closer to your customers. This diagram shows an example of a video file that is stored in Amazon Simple Storage Service (Amazon S3) in South America. The file is cached to an edge location near the customer to serve the video file faster to a customer in Asia.



When should you use AWS Local Zones?

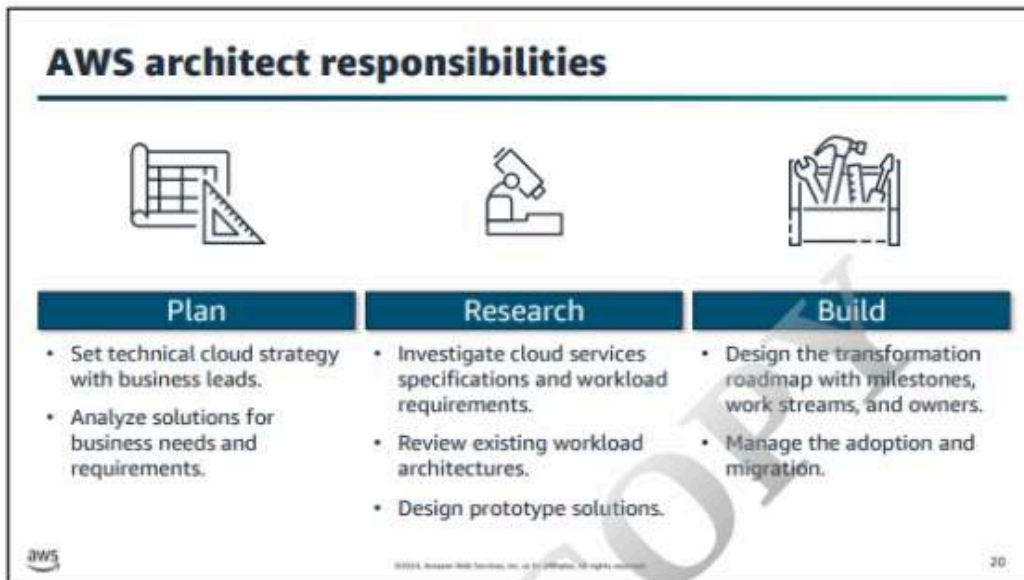
You should use AWS Local Zones to deploy AWS compute, storage, database, and other services closer to your end users for low-latency requirements. With AWS Local Zones, you can use the same AWS infrastructure, services, APIs, and tool sets that you are familiar with in the cloud.

When should you use edge locations?

You should use edge locations for caching the data (content) to provide fast delivery of content for users. Using edge locations provides a better user experience, with faster delivery to users at any location.



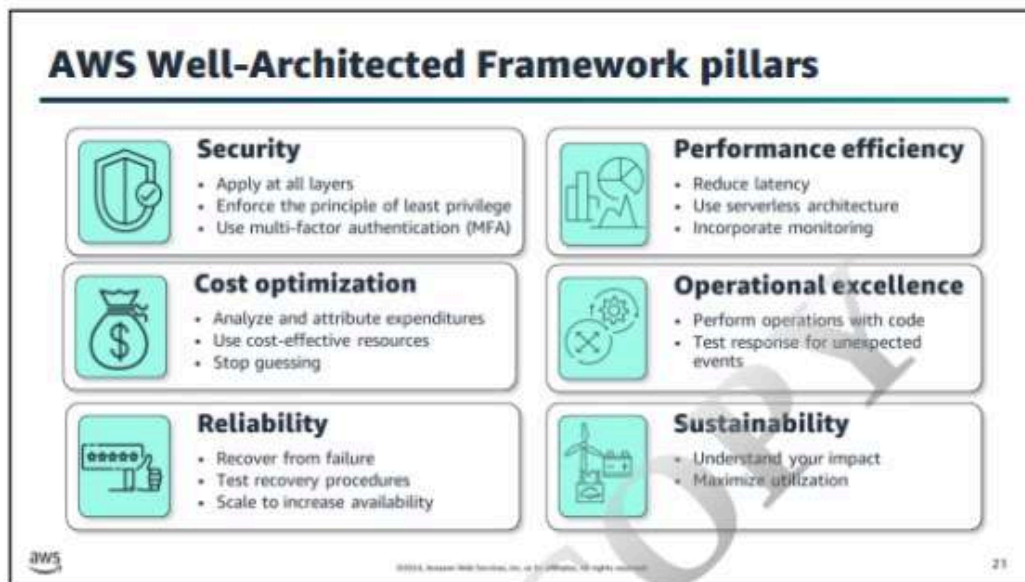
The CTO asks during the project meeting, "How can we build our cloud infrastructure according to best practices?" The AWS Well-Architected Framework provides consistent guidance for AWS architecting best practices.



Solutions architects (SAs) are responsible for managing an organization's cloud computing architecture. They have in-depth knowledge of the architectural principles and services that are used to do the following:

- Develop the technical cloud strategy based on business needs.
- Assist with cloud migration efforts.
- Review workload requirements.
- Provide guidance on how to address high-risk issues.

For more information about AWS architect responsibilities, see "Successful solutions architects do these five things" at <https://aws.amazon.com/blogs/training-and-certification/successful-solutions-architects-do-these-five-things/>.



Creating technology solutions is much like constructing a physical building. If the foundation is not solid, it can cause structural problems that undermine the integrity and function of the building. The AWS Well-Architected Framework helps cloud architects build secure, high-performing, resilient, and efficient application infrastructures. It is a consistent approach for customers and partners to evaluate architectures and implement designs that can scale over time.

The AWS Well-Architected Framework started as a whitepaper. It has expanded to include domain-specific lenses, hands-on labs, and the AWS Well-Architected Tool (AWS WA Tool).

The architectural reviews focus on the following pillars:

- **Security** – Use AWS security best practices to build policies and processes to protect data and assets. Allow auditing and traceability. Monitor, alert, and audit actions and changes to your environment in real time.
- **Cost optimization** – Achieve cost efficiency while considering fluctuating resource needs.
- **Reliability** – Meet well-defined operational thresholds for applications. This includes support to recover from failures, handling increased demand, and mitigating disruption.
- **Performance efficiency** – Deliver efficient performance for a set of resources such as instances, storage, databases, space, and time.
- **Operational excellence** – Run and monitor systems that deliver business value. Continually improve supporting processes and procedures.
- **Sustainability** – Minimize and understand your environmental impact when running cloud workloads.

With the tool, you can gather data and get recommendations to do the following:

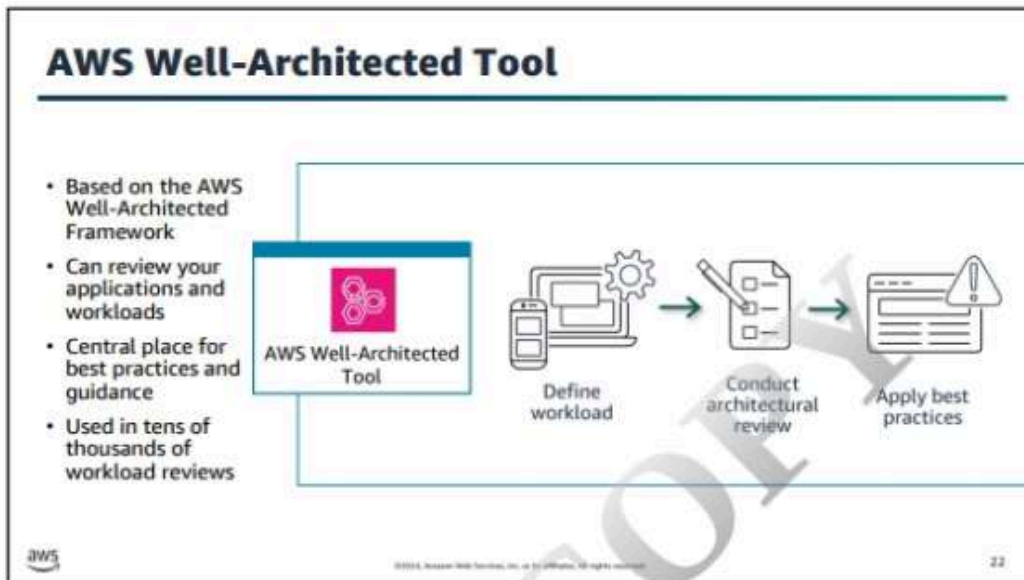
- Minimize system failures and operational costs.
- Dive deep into business and infrastructure processes.
- Provide best practice guidance.
- Deliver on the cloud computing value proposition.

For more information about related labs, see AWS Well-Architected Labs at <https://www.wellarchitectedlabs.com/>.

For more information about the AWS WA Tool, see AWS Well-Architected Tool at <https://aws.amazon.com/well-architected-tool/>.

For more information about the console, see AWS Well-Architected Tool in the AWS Management Console at <https://console.aws.amazon.com/wellarchitected/>.

DO NOT COPY
javierartiga.espublico@gmail.com



The AWS WA Tool is a self-service tool. You can use it to review the state of your existing workloads and compare them to the latest AWS architectural best practices. It is designed to help architects and their managers review AWS workloads without the need for an AWS SA. This service is based on the AWS Well-Architected Framework.


To complete a Well-Architected review, use the tool in the console. All details are stored securely in your account. You can share results with your SA or partner resource for collaboration on the review or remediation steps.

For more information about AWS WA Tool best practices, see “New – AWS Well-Architected Tool – Review Workloads Against Best Practices” in the *AWS News Blog* at <https://aws.amazon.com/blogs/aws/new-aws-well-architected-tool-review-workloads-against-best-practices/>.


For more information about the AWS Well-Architected Framework pillars, see AWS Well-Architected at <https://aws.amazon.com/architecture/well-architected/>.



Present solutions



Chief technology officer



Consider how you would answer the following questions:

- What are the benefits of using AWS services?
- How is the AWS global infrastructure organized?
- How can you build your cloud infrastructure according to best practices?

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

24

Imagine that you are now ready to talk to the chief technology officer, discuss what you have learned, and present solutions. Think about how you would answer the questions from the beginning of the lesson.

Your answers should include the following solutions:



- Use AWS services to increase agility while decreasing complexity and risk.
- AWS global infrastructure is organized into AWS Regions. These Regions contain Availability Zones. You can also use AWS Local Zones and edge locations.
- Use the Well-Architected Framework, which helps cloud architects build secure, high-performing, resilient, and efficient application infrastructures.

Module review

In this module, you learned about the following:

- ✓ AWS services
- ✓ AWS infrastructure
- ✓ AWS Well-Architected Framework

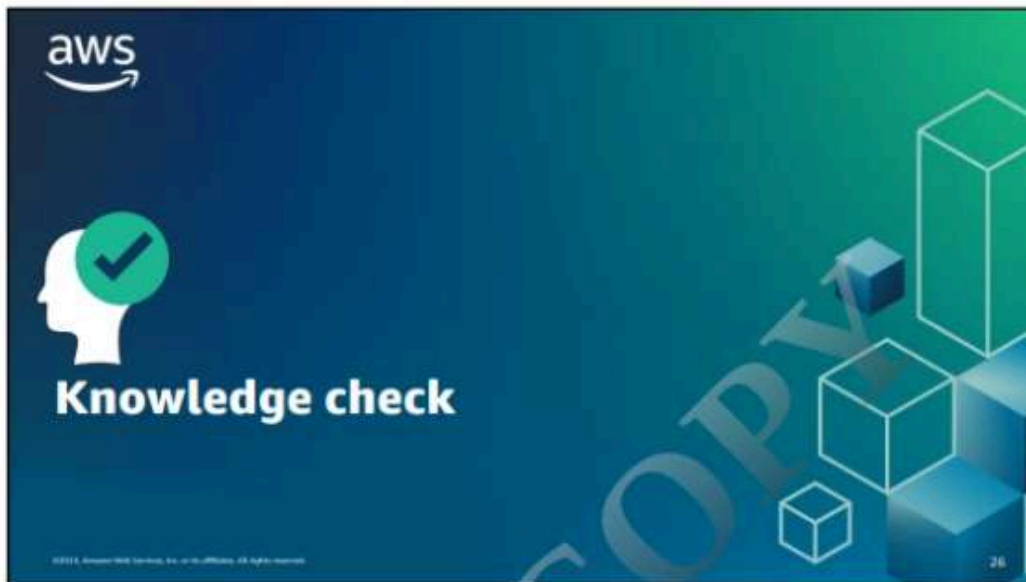
Next, you review the following:

-  Knowledge check
-  Lab introduction



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

25



Knowledge check question 1

Which item is the best example of one responsibility of an AWS architect?

Choice	Response
A	Monitor alarms for disaster response.
B	Maintain application-level code in the AWS Cloud.
C	Manage access to a group of AWS accounts.
D	Analyze solutions for business needs and requirements.



©2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

27

Knowledge check question 1 and answer

Which item is the best example of one responsibility of an AWS architect?

Choice	Response
A	Monitor alarms for disaster response.
B	Maintain application-level code in the AWS Cloud.
C	Manage access to a group of AWS accounts.
D correct	Analyze solutions for business needs and requirements.

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

28

The correct answer is D. AWS architects analyze solutions for business needs and requirements.

For more information about being a successful solutions architect on AWS, see “Successful solutions architects do these five things” on the *AWS Training and Certification Blog* at <https://aws.amazon.com/blogs/training-and-certification/successful-solutions-architects-do-these-five-things/>.

Knowledge check question 2

Which item is a cluster of data centers within a geographic location with low latency network connectivity?

Choice	Response
A	Availability Zone
B	Region
C	Edge location
D	Outposts



©2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

28

Knowledge check question 2 and answer

Which item is a cluster of data centers within a geographic location with low latency network connectivity?

Choice	Response
A correct	Availability Zone
B	Region
C	Edge location
D	Outposts

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

30

The correct answer is A, Availability Zone.

An Availability Zone is one or more discrete data centers with redundant power, networking, and connectivity in an AWS Region. For more information, see Regions and Availability Zones at https://aws.amazon.com/about-aws/global-infrastructure/regions_az/.

Knowledge check question 3

Which factors must be considered when picking an AWS Region? (Select TWO.)

Choice	Response
A	Local data regulations
B	Operating system requirements
C	Latency to end users
D	Support for hybrid networking
E	Programming language of the application

 ©2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. 11

Knowledge check question 3 and answer

Which factors must be considered when picking an AWS Region? (Select TWO.)

Choice	Response
A correct	Local data regulations
B	Operating system requirements
C correct	Latency to end users
D	Support for hybrid networking
E	Programming language of the application

©2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The correct answers are A, local data regulations, and C, latency to end users.

Choosing the right AWS Region is important. You must determine the right Region for your services, applications, and data, based on the following factors:

- Governance and legal requirements – Consider any legal requirements based on data governance, sovereignty, or privacy laws.
- Latency – Close proximity to customers means better performance.
- Service availability – Not all AWS services are available in all Regions.
- Cost – Different Regions have different costs. Research the pricing for the services that you plan to use, and compare costs to make the best decision for your workloads.

Knowledge check question 4

What is the primary benefit of deploying applications into multiple Availability Zones?

Choice	Response
A	Stronger security policies for resources
B	Decreased latency to resources
C	High availability for resources
D	There is no benefit to this design



©2025 Amazon Web Services, Inc. or its affiliates. All rights reserved.

11

Knowledge check question 4 and answer

What is the primary benefit of deploying applications into multiple Availability Zones?

Choice	Response
A	Stronger security policies for resources
B	Decreased latency to resources
C correct	High availability for resources
D	There is no benefit to this design

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The correct answer is C, high availability for resources.

Availability Zones are multiple isolated areas within a particular geographic location. When you launch an instance, you can choose an Availability Zone or let AWS choose one for you. You can distribute your instances across multiple Availability Zones. You can design your application so that if one instance fails, an instance in another Availability Zone can handle requests.

Knowledge check question 5

Which AWS Well-Architected Framework pillar contains the principle of least privilege?

Choice	Response
A	Operational excellence
B	Security
C	Reliability
D	Performance efficiency




©2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

35

Knowledge check question 5 and answer

Which AWS Well-Architected Framework pillar contains the principle of least privilege?

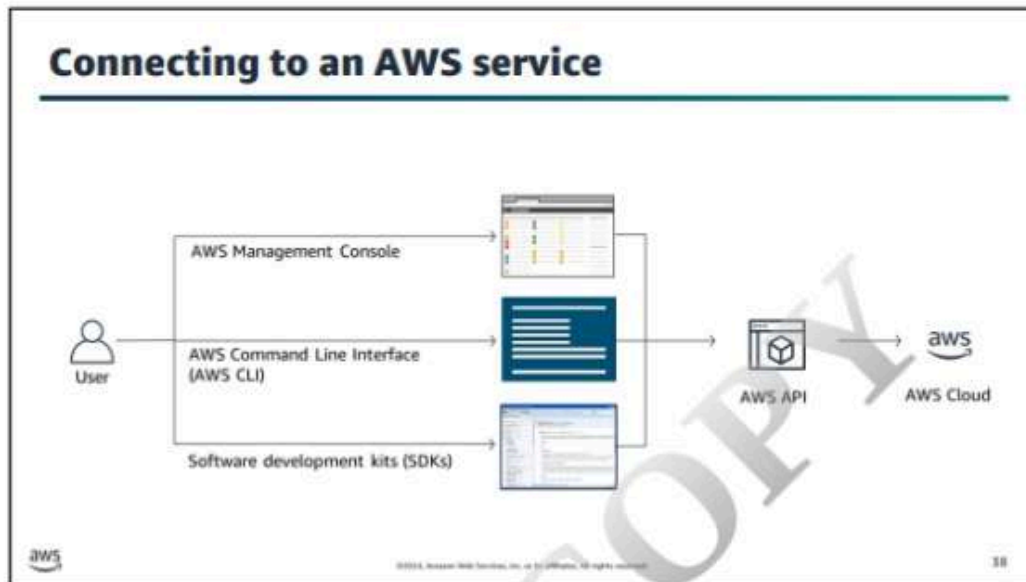
Choice	Response
A	Operational excellence
B correct	Security
C	Reliability
D	Performance efficiency

 ©2025 Amazon Web Services, Inc. or its affiliates. All rights reserved. 36

The correct answer is B, security.

The principle of least privilege (POLP) is a concept in computer security that limits users' access rights to only what is strictly required to do their jobs.





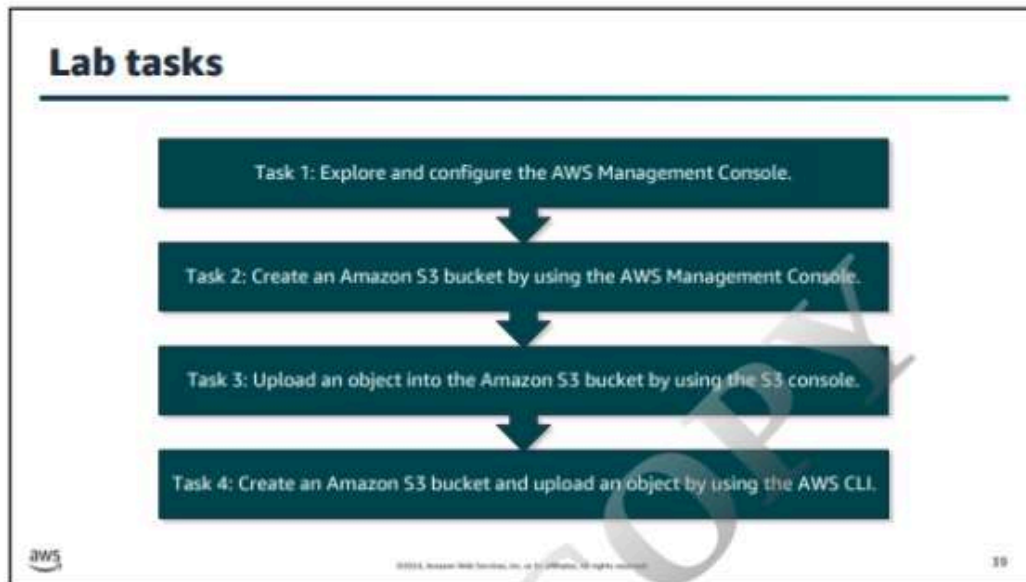
You can use the following tools to interact with AWS:

- **AWS Management Console** – This tool is the easiest place to start interacting with AWS services. It is a graphical user interface (GUI) to manage your AWS account and take actions.
- **AWS Command Line Interface (AWS CLI)** – This tool is for managing AWS services by using the command line. AWS CLI version 1 is preinstalled on Amazon Linux and Amazon Linux 2 distributions.
- **Software development kits (SDKs)** – AWS provides AWS SDKs and the AWS Cloud Development Kit (AWS CDK) in many common programming languages. You use these software development frameworks for defining and provisioning your cloud infrastructure by using code.

All of these tools connect to the same underlying AWS API to create resources and manage your AWS services.

Learn about installing, updating, and uninstalling the AWS CLI. For more information, see “Installing or updating the latest version of the AWS CLI” in the *AWS Command Line Interface User Guide for Version 2* at <https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-install.html>.

Learn how to get started with using the AWS CDK. For more information, see “Getting started with the AWS CDK” in the *AWS Cloud Development Kit (AWS CDK) v2 Developer Guide* at https://docs.aws.amazon.com/cdk/latest/guide/getting_started.html.









****Accessibility note for screen reader users:** To make code examples in this presentation more accessible, Start of code and End of code labels have been added before and after blocks of code on slides in this presentation.

Poll question



How many Amazon Web Services (AWS) accounts does your organization use?

- A. 1
- B. 2–10
- C. More than 10
- D. I don't know

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

2

Module overview

- Business requests
- Principals and identities
- Security policies
- Managing multiple accounts
- Module review
- Knowledge check



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

1

Business request



Security specialist



The security specialist needs to know the following:

- What are the best practices to manage access to AWS accounts and resources?
- How can we give users access to only the resources that they need?
- What is the best way to manage multiple accounts?

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

4

Imagine that your security specialist meets with you to discuss how to start building accounts with least privilege in AWS. Here are some questions that they are asking about account security.

At the end of this module, you meet with the security specialist and present some solutions.



The security specialist asks, “What are the best practices to manage access to AWS accounts and resources?”
The security team must start setting up accounts. The company wants your advice about how to provide access.

AWS account root user

A root user:

- Has full access to all AWS services
- Cannot be restricted in a single account model
- Should not be used for day-to-day interactions with AWS

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

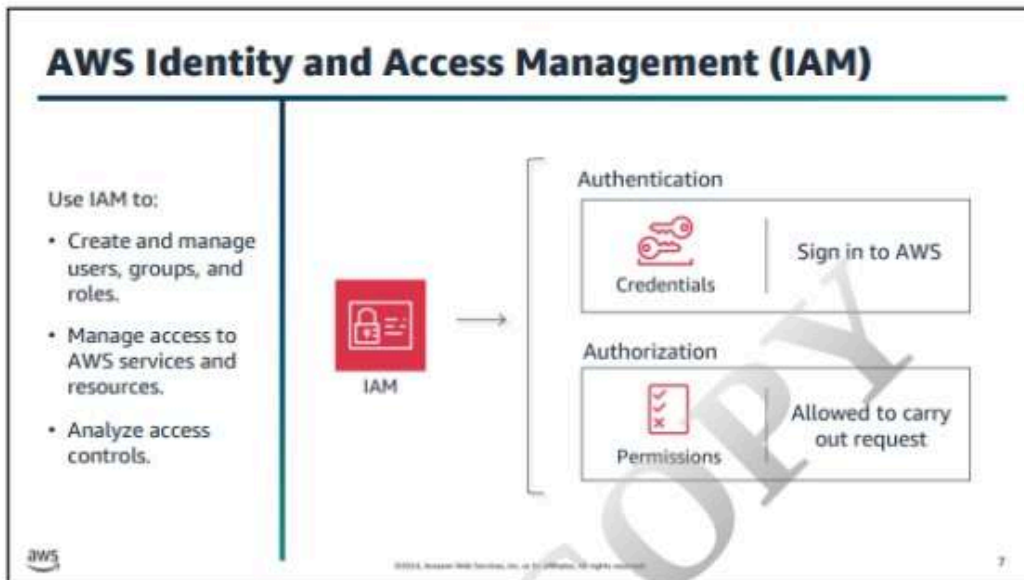
When you first create an AWS account, you begin with a *root user*. A root user has complete access to all AWS services and resources in the account. You access the root user identity by signing in with the email address and password that you were provided when you created the account. AWS strongly recommends that you not use root account credentials for day-to-day interactions with AWS. Create users for everyday tasks. You can manage and audit users with relative ease.

Create your additional users and assign permissions to these users by following the *principle of least privilege*. This principle grants users only the level of access that they require and nothing more. You can start by creating an administrator user. Manage the account with the administrator user instead of the root user.

As a best practice, require multi-factor authentication (MFA) for your root user. It provides you with an extra layer of security for your AWS accounts. Use your root user only for tasks that require it.

For more information about the root user, see “AWS account root user” in the *AWS Identity and Access Management User Guide* at https://docs.aws.amazon.com/IAM/latest/UserGuide/id_root-user.html.

For more information about least privilege and IAM best practices, see “Grant least privilege” in the *AWS Identity and Access Management User Guide* at <https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#grant-least-privilege>.



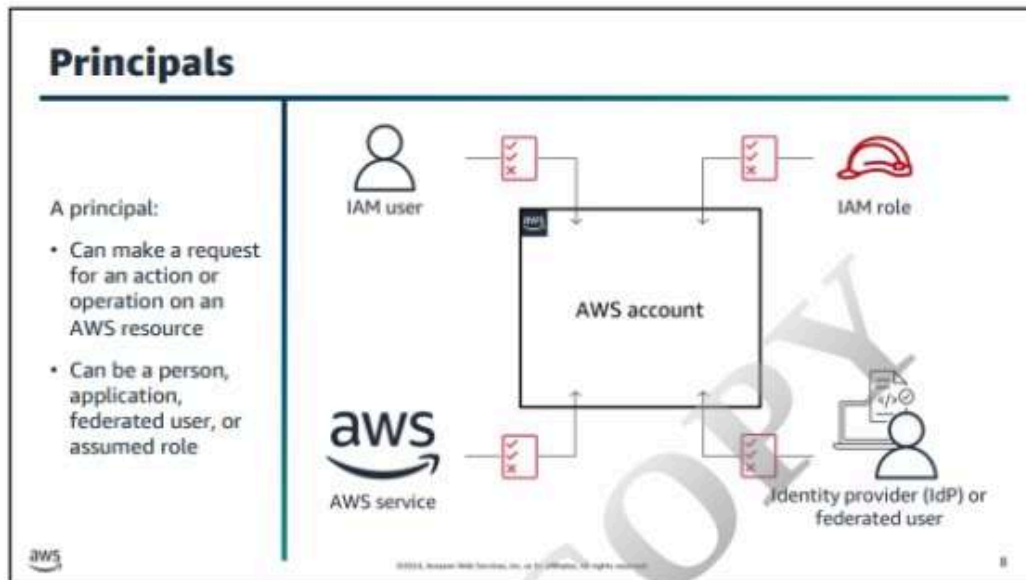
AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. Use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

Think of IAM as the tool to centrally manage access to launching, configuring, managing, and terminating your resources. You have granular control over access permissions. This control is based on resources and helps you define who has permissions to which API calls.

You manage access in AWS by creating and using security policies. You learn about IAM users, IAM user groups, and roles in this section.

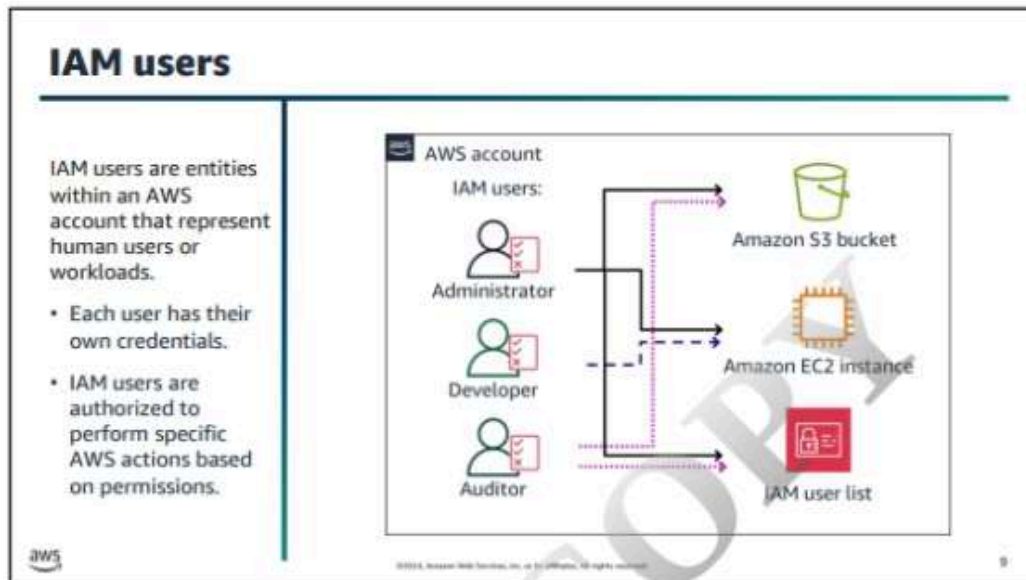
For more information about IAM, see “What is IAM?” in the *AWS Identity and Access Management User Guide* at <https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html>.

For more information about policy types and their uses, see “Policies and permissions in IAM” at https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html.



A *principal* is an entity that can request an action or operation on an AWS resource. IAM users and IAM roles are the most common principals that you work with, and you learn about them in this lesson. The principal can also be an AWS service, such as Amazon Elastic Compute Cloud (Amazon EC2), a Security Assertion Markup Language 2.0 (SAML 2.0) provider, or an identity provider (IdP). With an IdP, you manage identities outside IAM—for example, login with Amazon, Facebook, or Google. You can give these external identities permissions to use AWS resources in your account. Federated users are external identities that IAM does not manage directly.

For more information about federated users, see Identity federation in AWS at <https://aws.amazon.com/identity/federation/>.

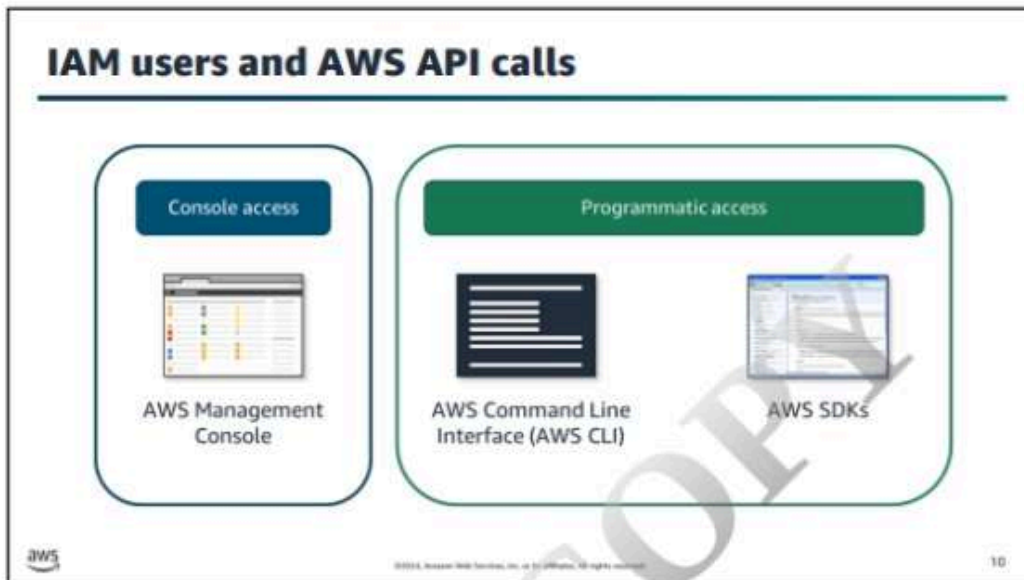


By default, a new IAM user has no permissions to do anything. The user is not authorized to perform any AWS operations or access any AWS resources. An advantage of having individual IAM users is that you can assign permissions individually to each user.

For example, this diagram shows three IAM users—an administrator, developer, and auditor—and their permissions within an AWS account. The administrator has permissions to access an Amazon Simple Storage Service (Amazon S3) bucket, an EC2 instance, and a list of IAM users in your account. The auditor has read-only permissions to Amazon S3 and IAM, but not to Amazon EC2. The developer only has permissions to the EC2 instance.

As a best practice, require multi-factor authentication (MFA) for your IAM users and set up an IAM user password policy.

For more information about IAM users, see “IAM users” in the *AWS Identity and Access Management User Guide* at https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users.html.



Provide the type of credentials that are required for the type of access that a user will need.

Ways to access AWS services include the following:

- AWS Management Console access – Create a password for a user.
- Programmatic access – The IAM user might need to make API calls, use the AWS CLI, or use the AWS SDKs. In that case, you create an access key (access key ID and a secret access key) for that user.

As a best practice, apply the principle of least privilege, which means that you create only the credentials that the user needs. For example, do not create access keys for a user who requires access only through the console.

AWS requires different types of security credentials, depending on how you access AWS.

For more information, see "Understanding and getting your AWS credentials" in the *AWS General Reference* at <https://docs.aws.amazon.com/general/latest/gr/aws-sec-cred-types.html>.

For more information about password creation, see "Managing passwords for IAM users" in the *AWS Identity Access and Management User Guide* at https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_admin-change-user.html.

Programmatic access



Access Key ID: AKIAIOSFODNN7EXAMPLE
 Secret Access Key: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

AWS CLI

```
$ aws configure
AWS Access Key ID [*****MPLE]:
AWS Secret Access Key [*****EKEY]:
Default region name [ap-southeast-1]:
Default output format [json]:
```

AWS SDK





Java Python .NET

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

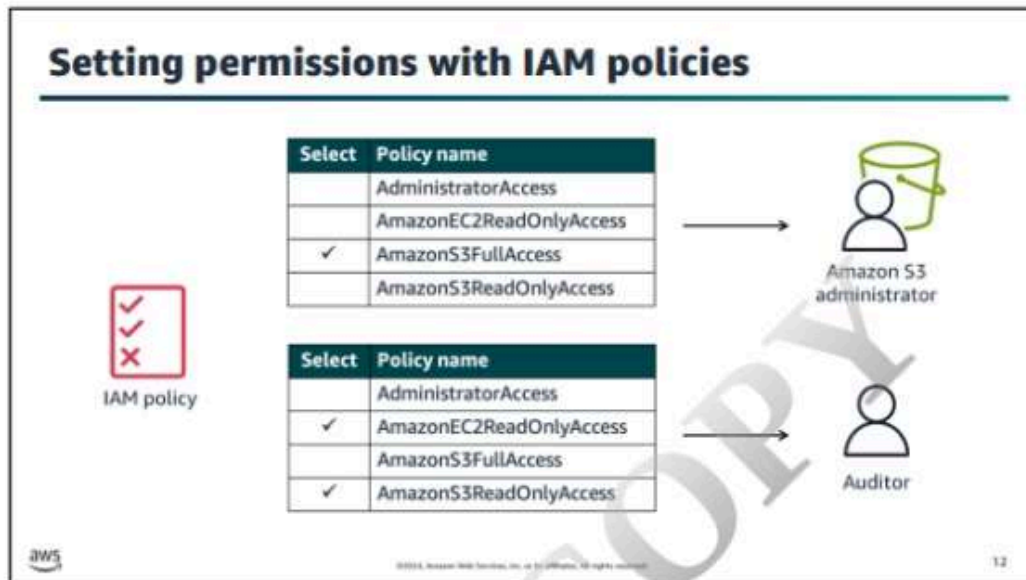
Programmatic access gives your IAM user the credentials to make API calls in the AWS CLI or AWS SDKs. AWS provides an SDK for programming languages such as Java, Python, and .NET.

When programmatic access is granted to your IAM user, it creates a unique key pair that comprises an access key ID and a secret access key. Use your key pair to configure the AWS CLI, or make API calls through an AWS SDK.

To set up AWS CLI in your client, enter the `aws configure` command. The example code shows the four elements that are required to configure your IAM user in AWS CLI:

- AWS access key ID
- AWS secret access key
- Default Region name
- Default output format (JSON, YAML, YAML stream, text, table)

For more information about configuring your key pair in AWS CLI, see “Configuration basics” in the *AWS Command Line Interface User Guide* at <https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-quickstart.html>.



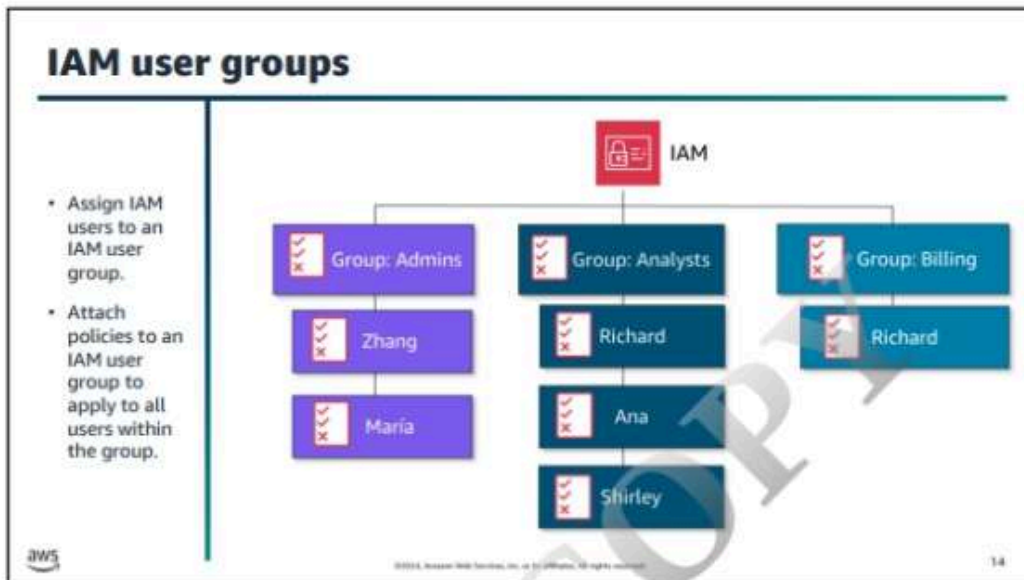
To allow IAM users to create or modify resources and perform tasks, do the following tasks:

1. Create or choose IAM policies that grant IAM users permission to access the specific resources and API actions that they will need.
2. Attach the policies to the IAM users or groups that require those permissions.

Users only have the permissions that you specify in the policy. Most users have multiple policies. Together, they represent the permissions for that user.

In the diagram, you choose to give the Amazon S3 administrator full access to Amazon S3, but you do not grant full access to all services in your AWS account. You attach the AmazonEC2ReadOnlyAccess and AmazonS3ReadOnlyAccess policies to an auditor who needs to know what resources exist in your account. The auditor should not be able to modify or delete anything.

As a best practice, attach only the policies that the user requires to complete the work.



An IAM user group is a collection of IAM users. With user groups, you can specify permissions for multiple users, which helps you to manage the permissions. A user can be a member of more than one user group. In the diagram, Richard is a member of the Analysts group and the Billing group. Richard gets permissions from both IAM user groups.

The IAM example is an on-screen diagram that shows how IAM users are broken into three groups:

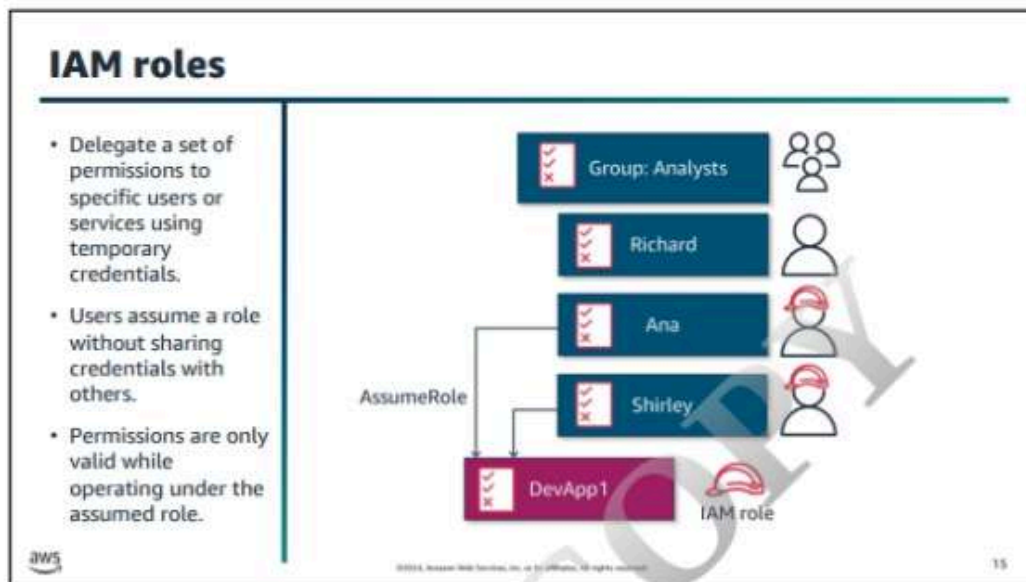
- Admins
- Analysts
- Billing

Zhang and Maria are in the Admins group.

Richard, Ana, and Shirley are in the Analysts group.

Richard is the only one in the Billing group. He is also the only one in two groups.

For more information about user groups, see "IAM user groups" at https://docs.aws.amazon.com/IAM/latest/UserGuide/id_groups.html.

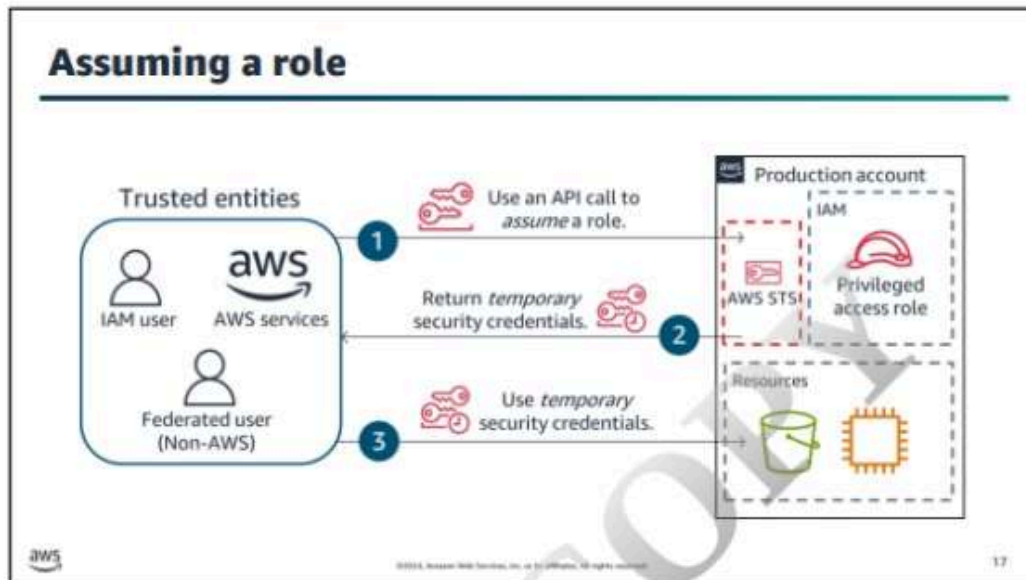


IAM roles deliver temporary AWS credentials. Multiple employees and applications can use the same role, which helps you to manage the roles. There are no charges for using roles.

For example, in this diagram the IAM users Richard, Ana, and Shirley are members of the Analysts user group. As members of the Analysts group, these users inherit permissions that are assigned to the group. Another IAM role, which is called DevApp1, is being used for testing purposes. DevApp1 has its own set of permissions. Ana and Shirley can assume the role and temporarily use the permissions specific to the DevApp1 role. While they assume this role, Ana and Shirley only have the permissions that are granted to the role and do not follow their group's inherited permissions.

The following examples show how you might use IAM roles:

- Cross-account access – Developer Diego requires access to an S3 bucket in the Prod account.
- Temporary account access – Contractor Carlos requires temporary access to an S3 bucket in the Prod account.
- Least privilege – Require Diego to use IAM roles to delete an Amazon DynamoDB table.
- Audit – Administrator Ana wants to track who used an IAM role.
- Access for AWS services – Amazon Lex must use Amazon Polly to synthesize speech responses for your AI assistant.
- IAM roles for Amazon EC2 – An application that is running on Amazon EC2 requires access to an S3 bucket and a DynamoDB table.
- SAML federation – Administrator Ana wants to use IAM with identities that are stored in an external IdP.



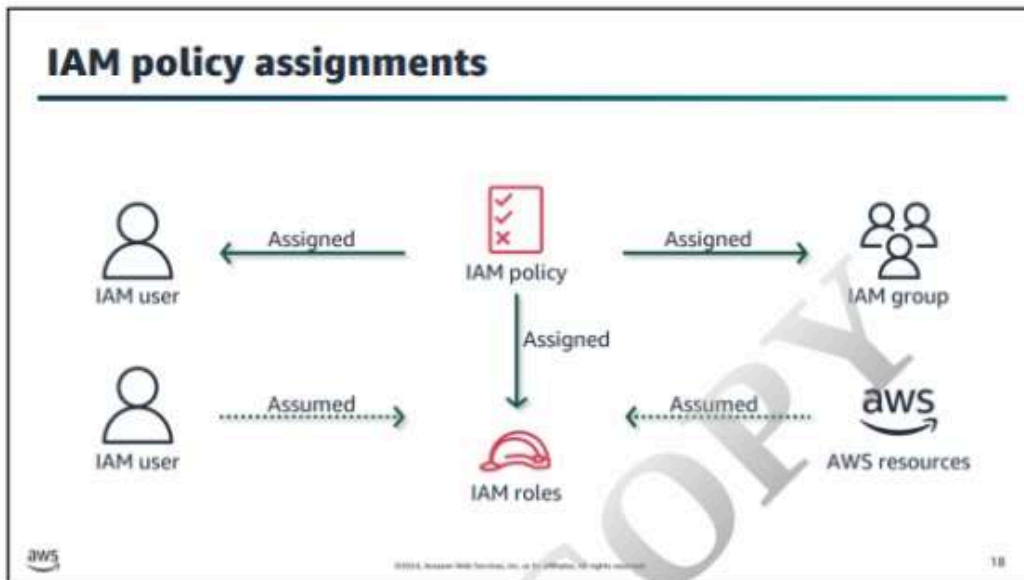
You assume a role by using a trusted entity, such as an IAM user, an AWS service, or a federated user.

IAM users assume roles in the AWS Management Console or AWS CLI. This action uses the AssumeRole API. AWS services can use the same API call to assume roles in your AWS accounts. Your federated users use either the AssumeRoleWithSAML or the AssumeRoleWithWebIdentity API calls.

The API call is made to AWS Security Token Service (AWS STS). AWS STS is a web service that provides temporary, limited-privilege credentials for IAM users or federated users. It returns a set of temporary security credentials that consist of an access key ID, a secret access key, and a security token. These credentials are then used to access AWS resources. The AssumeRole API is typically used for cross-account access or federation.

For more information about AWS STS, see the *AWS Security Token Service API Reference* at <https://docs.aws.amazon.com/STS/latest/APIReference/Welcome.html>.

For more information about using IAM roles, see "Using IAM roles" in the AWS Identity and Access Management User Guide at https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use.html.

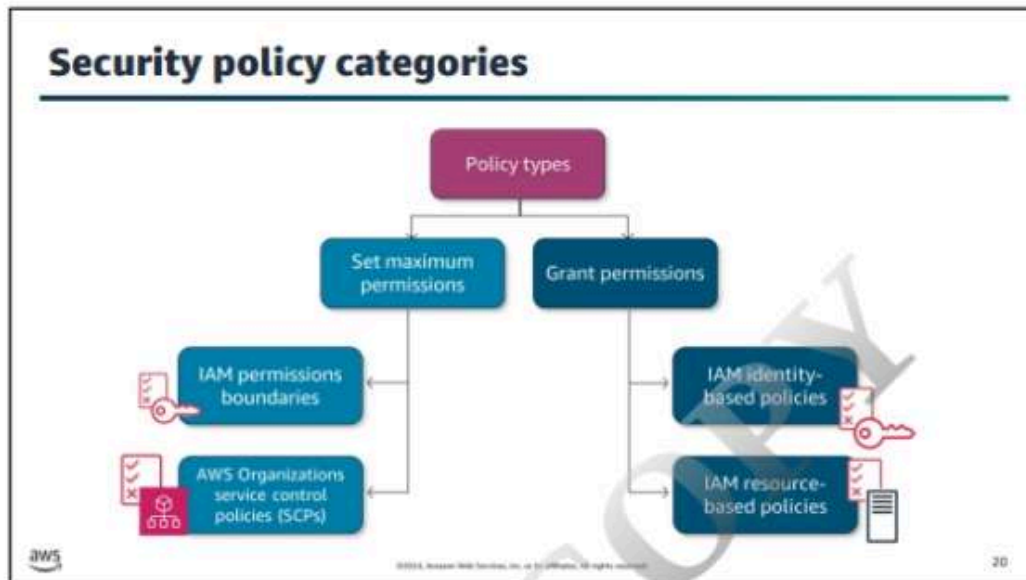


IAM provides you with the tools to create and manage all types of IAM policies (managed policies and inline policies). To add permissions to an IAM identity (IAM user, group, or role), you create a policy, validate the policy, and then attach the policy to the identity. You can attach multiple policies to an identity, and each policy can contain multiple permissions. You learn more about IAM policies in the next section.

Use roles to delegate access to users, applications, or services that don't normally have access to your AWS resources.



The security specialist asks, "How can we give users access to only the resources they need?" The security team has users and roles set up. The company wants your advice about setting up permissions in security policies.



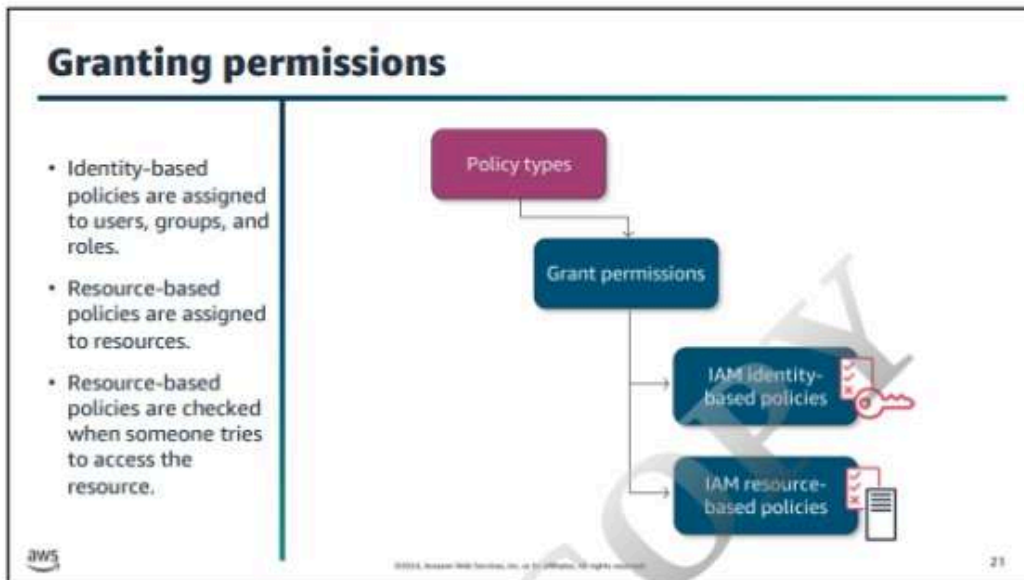
A *policy* is attached to an identity or resource to define its permissions. AWS evaluates these policies when a principal, such as a user, makes a request.

In the diagram, the policy types are responsible for either setting maximum permissions or granting permissions. IAM permissions boundaries and AWS Organizations service control policies (SCPs) help set maximum permissions on actions in your account. IAM identity-based policies and resource-based policies grant permissions to allow or deny actions in your account.

The following policy types, listed in order of frequency, are available for use in AWS. You learn about each of these policy types in more detail later in this module.

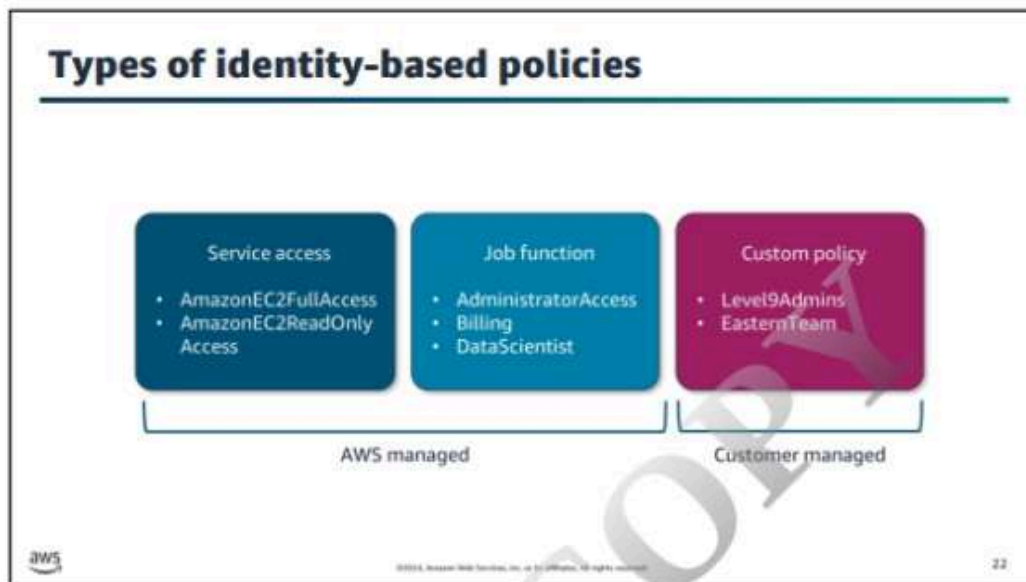
Policy types

- **Identity-based policies** – Attach managed and inline policies to IAM identities. These identities include users, groups to which users belong, and roles.
- **Resource-based policies** – Attach inline policies to resources. The most common examples of resource-based policies are Amazon S3 bucket policies and IAM role trust policies.
- **AWS Organizations service control policies (SCPs)** – Use Organizations SCPs to define the maximum permissions for account members of an organization or organizational unit (OU).
- **IAM permissions boundaries** – AWS supports *permissions boundaries* for IAM entities (users or roles). Use IAM permissions boundaries to set the maximum permissions that an IAM entity can receive.



Identity-based policies are JSON permissions policy documents that control which actions an IAM identity (users, groups of users, and roles) can perform. These policies also control which resources an IAM identity can perform these actions on, and under what conditions they can perform these actions.

Resource-based policies are JSON policy documents that you attach to a resource, such as an Amazon S3 bucket. These policies grant the principal permission to perform specific actions on that resource and define under what conditions this applies. Resource-based policies are inline policies. There are no managed resource-based policies.



You can choose to use existing AWS policies. AWS manages some of these policies. You also have the option to create your own policies.

Identity-based policies can be categorized by the following types:

- **Managed policies** – Standalone identity-based policies that you can attach to multiple users, groups, and roles in your AWS account. There are two types of managed policies:
 - **AWS managed policies** – Managed policies that AWS creates and manages. They are built to provide specific service access or permissions for job functions.
 - **Customer managed policies** – Managed policies that you create and manage in your AWS account. Customer managed policies provide more precise control over your policies than AWS managed policies do.
- **Inline policies** – Policies that you add directly to a single user, group, or role. Inline policies maintain a strict one-to-one relationship between a policy and an identity. They are deleted when you delete the identity.

An *inline policy* is a policy that you create and embed directly to an IAM group, user, or role. Inline policies can't be reused on other identities or managed outside of the identity where they exist.

As a best practice, use customer managed policies instead of inline policies.

For more information, see “Security best practices in IAM” at

<https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#best-practice-managed-vs-inline>.

Identity-based policy example

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Resource": "arn:aws:ec2:*:*:instance/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Owner": "${aws:username}"
        }
      }
    }
  ]
}

```

A

Use this version date to use all of the available policy features.

B

Indicate whether the policy allows or denies an action.

C

Include a list of actions that the policy allows or denies.

D

Choose a list of resources that the effect applies to.

E

Optional: Specify the conditions under which the policy applies.

© 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved.
24

A JSON identity-based policy document includes these elements:

- **Version** – The Version policy element specifies the language syntax rules to use to process a policy. To use all of the available policy features, include the value 2012-10-17 for the version in your policies.
- **Effect** – Use Allow or Deny to indicate whether the policy allows or denies access.
- **Action** (or NotAction) – Include a list of actions that the policy allows or denies.
- **Resource** (or NotResource) – You must specify a list of resources that the actions apply to.
- **Condition** (or NotCondition) – Specify the circumstances under which the policy grants permission.

Note: This course does not cover the NotAction, NotResource, and NotCondition policy elements.

For example, you can attach the example policy statement to your IAM user. Then, that user is allowed to stop and start EC2 instances in your account if the condition is met. Here, the EC2 instances that your IAM user can control must have a tag with key Owner and value equal to the IAM user name.

In the Resource element, the policy lists an Amazon Resource Name (ARN) with a wildcard (asterisk) character. You use wildcards to apply a policy element to more than one resource or action. This policy applies for resources in any account number and Region with any resource ID. You can reuse it in multiple accounts without having to rewrite the policy with your AWS account ID.

For more information, see “Policies and permissions in AWS Identity and Access Management” in the *AWS Identity and Access Management User Guide* at https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html.

Explicit allow and explicit deny

This section from a policy allows access.
It is called an *explicit allow*.

```
{
  "Effect": "Allow",
  "Action": [
    "s3:ListBucket",
    "s3:GetObject"
  ],
  "Resource": [
    "arn:aws:s3:::BUCKET-NAME",
    "arn:aws:s3:::BUCKET-NAME/*"
  ]
}
```

This section from a policy denies access.
It is called an *explicit deny*.

```
{
  "Effect": "Deny",
  "Action": [
    "ec2:*",
    "s3:*"
  ],
  "Resource": "*"
}
```



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

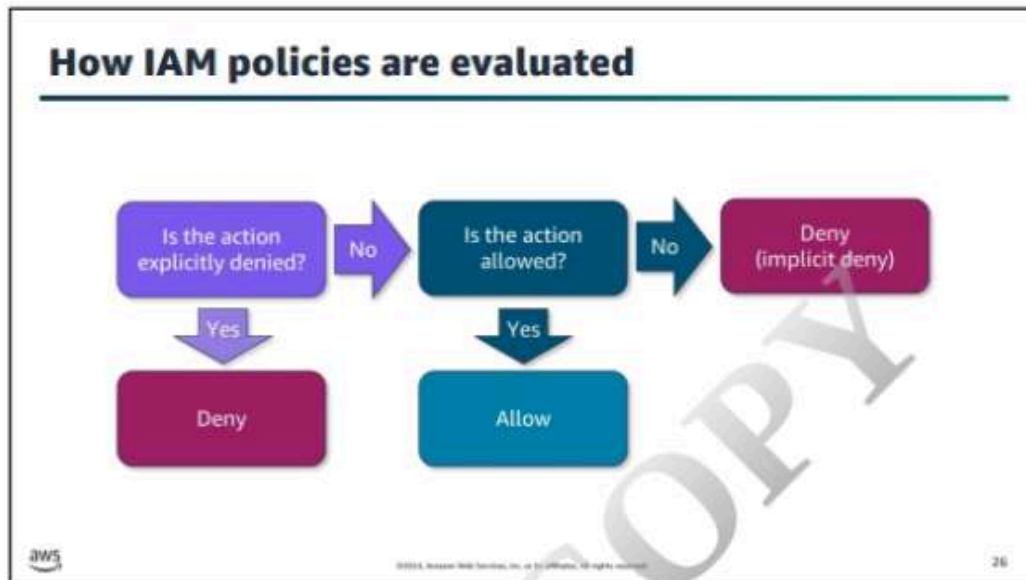
25

An IAM policy is made up of explicit allow statements, explicit deny statements, or both.

The first policy shows an *explicit allow*. It authorizes your IAM user, group, or role to take the listed actions against a set of your resources. The policy allows list and get actions on all objects in an S3 bucket called BUCKET-NAME. When you use a wildcard character after the bucket name and slash, it covers all objects in that bucket.

The second policy shows an *explicit deny*. It stops your IAM user, group, or role when trying to take an action that is listed for a set of your resources. In the second policy example, all actions in Amazon EC2 or Amazon S3 on any resource are denied.

Use allow and deny in your statement to guide which actions your principals can take in your account.



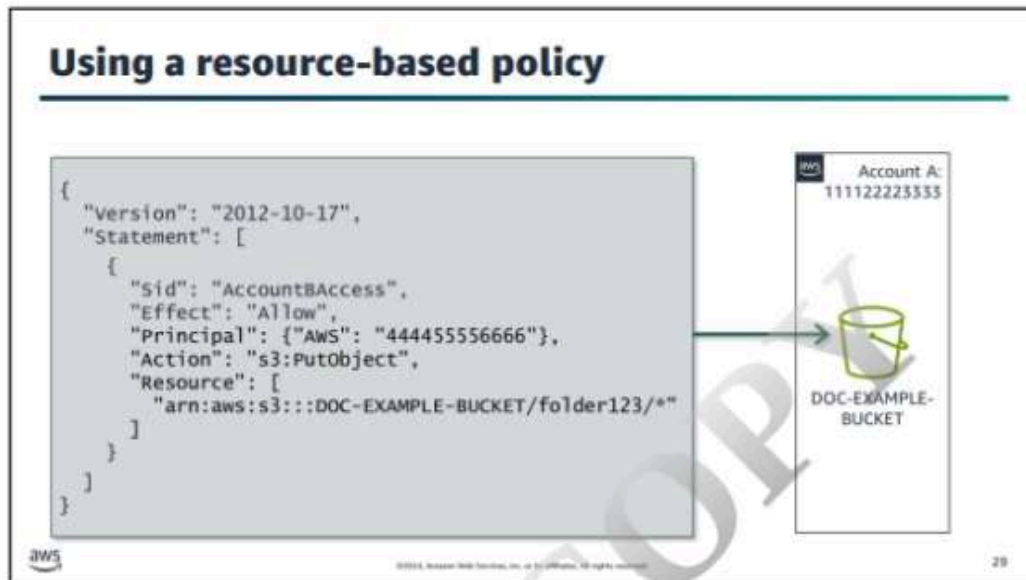
It is important to know the AWS evaluation logic when building IAM policies for your account. This way, you can give your users and applications only the access they need.

AWS evaluates all policies that are applicable to the request context. The following list summarizes the AWS evaluation logic for policies within a single account:

- By default, all requests are implicitly denied with the exception of the AWS account root user, which has full access. This policy is called an *implicit deny*.
- An explicit allow in an identity-based policy or resource-based policy overrides this default. There are additional security controls that can override an explicit allow with an implicit deny, such as permissions boundaries and SCPs. Both of these security controls are covered later in this module.
- An explicit deny in any policy overrides any allows.

Explicit deny is useful as a safety measure because it overrides explicit allow.

For more information, see "Determining whether a request is allowed or denied within an account" under "Policy evaluation logic" in the *AWS Identity and Access Management User Guide* at https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_evaluation-logic.html#policy-eval-denyallow.



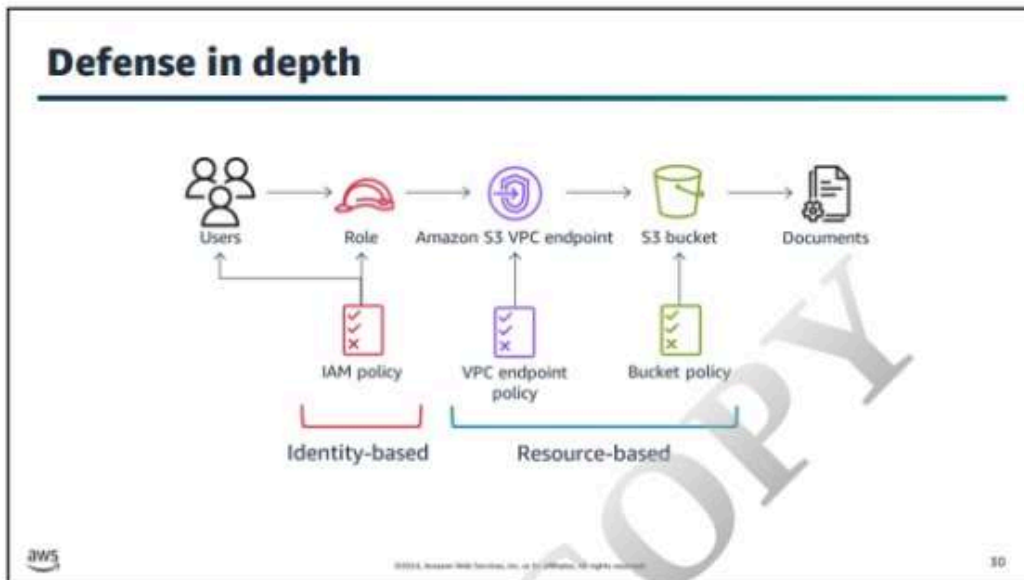
Resource-based policies are attached to a single resource, such as an S3 bucket or AWS Lambda function. You learn more about S3 buckets and Lambda functions later in this course. With resource-based policies, you choose who has access to the resource and what actions they can perform on it.

In the example, the principal is an AWS account ID. The set of resources are all objects in the bucket DOC-EXAMPLE-BUCKET that are within the folder called folder123. The bucket policy allows an administrator in the specific AWS account to assign permission to upload objects to your bucket's folder.

For more information about cross-account policy evaluation, see "Determining whether a cross-account request is allowed" at https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_evaluation-logic-cross-account.html#policy-eval-cross-account.

AWS identity-based policies and resource-based policies are evaluated together. Recall how IAM policies are evaluated. If any explicit deny statement is found in any IAM policy, the action is denied. If at least one allow statement exists with no explicit deny, the action is allowed.

For more information about identity-based policies, see "Identity-based policies and resource-based policies" in the *AWS Identity and Access Management User Guide* at https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_identity-vs-resource.html.

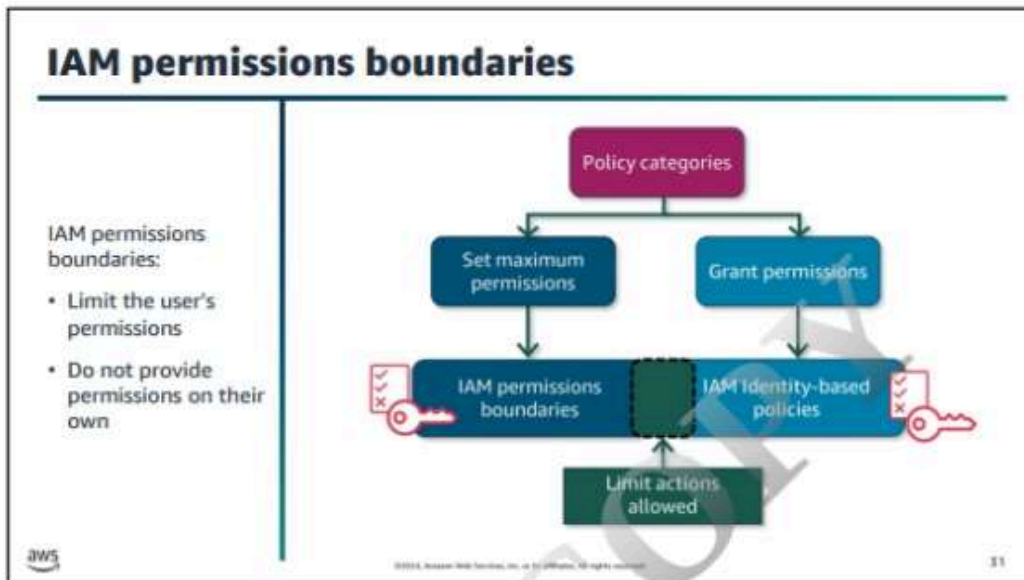


Defense in depth is a strategy that is focused on creating multiple layers of security.

Apply a defense in depth approach with multiple security controls to all layers. For example, apply it to the edge of the network, virtual private cloud (VPC), load balancing, and every instance, compute service, operating system, application, and code. Application security is as critical as instance security.

In the diagram, different users try to access a document in your S3 bucket. Each user needs an identity-based policy that is assigned to either their user or a role that they assume to access AWS. They then navigate through layers of resource-based policies—first a VPC endpoint policy, then a bucket policy for the S3 bucket. Your users are able to access the documents that they need for their task. You will learn more about VPC endpoints and S3 buckets later in this course.

For more information, see “Policy evaluation logic” in the *AWS Identity and Access Management User Guide* at https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_evaluation-logic.html.



AWS supports *permissions boundaries* for IAM entities—users or roles. A permissions boundary is an advanced feature for using a managed policy to set the *maximum permissions that an identity-based policy can grant to an IAM entity*. Permissions boundaries act as a filter.

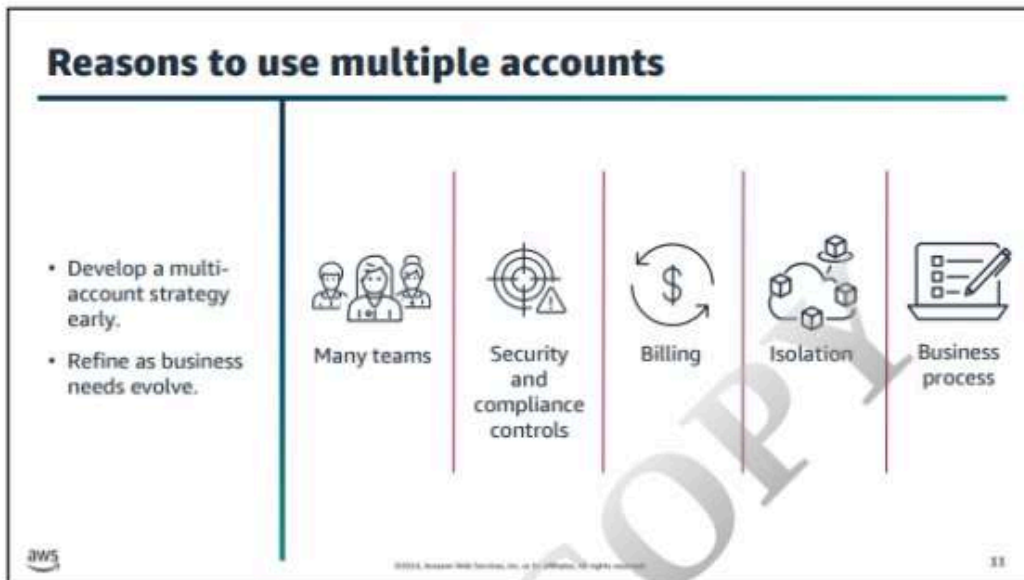
An entity's permissions boundary allows it to perform only the actions that are allowed by both its identity-based policies and its permissions boundaries.

For more information about permissions boundaries, see “Permissions boundaries for IAM entities” in the *AWS Identity and Access Management User Guide* at https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_boundaries.html.

[Image Description: Diagram that shows the two policy categories, set maximum permissions and grant permissions. Connected to set maximum permissions are IAM permission boundaries. Partially overlapping IAM permission boundaries and connected to grant permissions are IAM identity-based policies. The area of overlap is labeled “limits actions allowed.” **End Description.]**

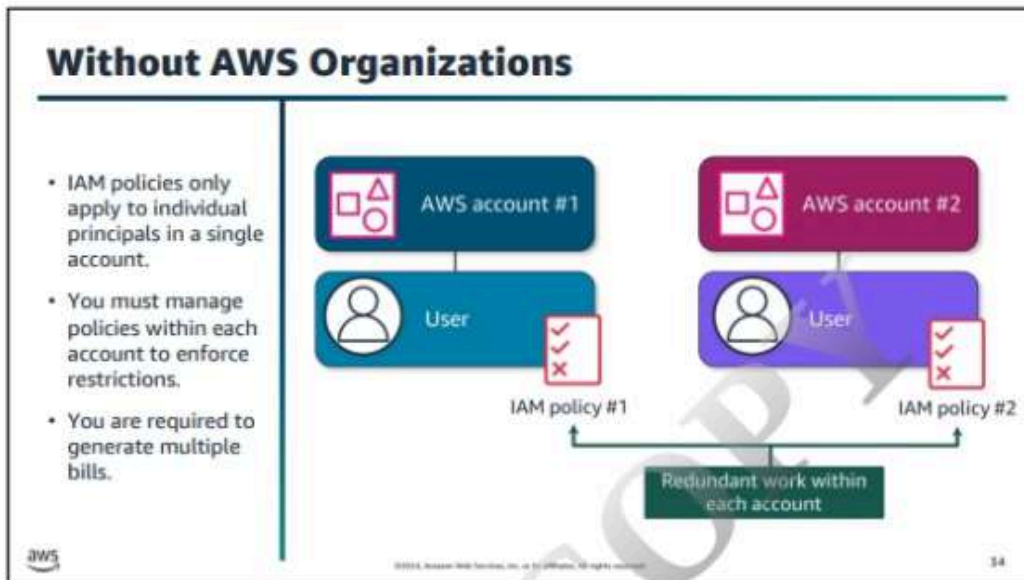


The security specialist asks, "What is the best way to manage multiple accounts?" The company wants your advice about ways to manage more than one account.

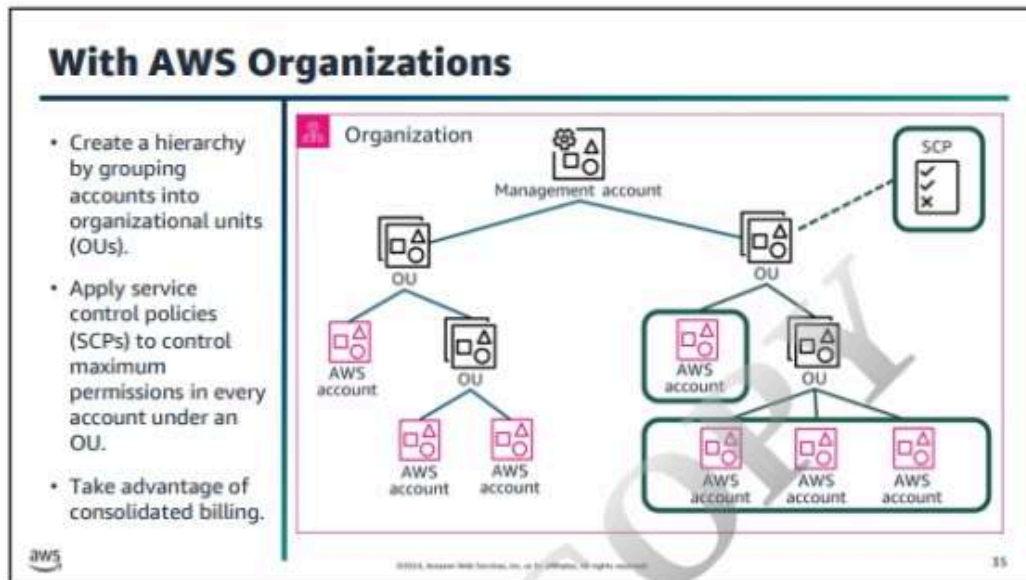


As you expand your use of AWS, you have several reasons that you might want to create a multi-account structure in your organization:

- To group resources for categorization and discovery
- To improve your security posture with a logical boundary
- To limit potential impact in case of unauthorized access
- To simplify management of user access to different environments



Managing multiple accounts is more challenging without AWS Organizations. For example, IAM policies only apply to a specific AWS account. Therefore, you must duplicate and manage IAM policies in each account to deploy standardized permissions across all accounts.

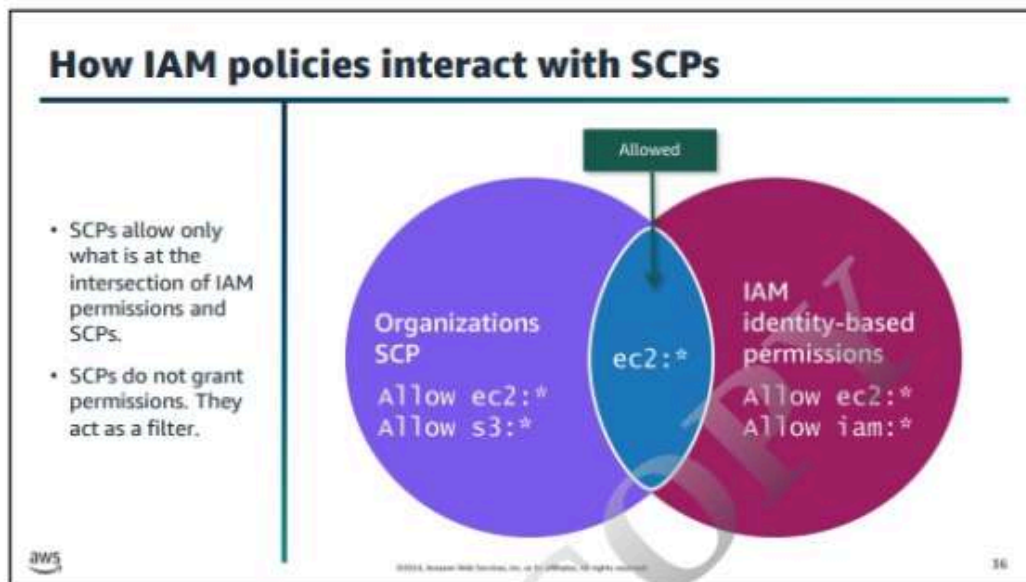


AWS Organizations provides these key features:

- Centralized management of all your AWS accounts
- Consolidated billing for all member accounts
- Hierarchical grouping of your accounts to meet your budgetary, security, or compliance needs
- Policies to centralize control over the AWS services and API actions that each account can access
- Policies to standardize tags across the resources in your organization's accounts
- Policies to control how AWS AI and machine learning (ML) services can collect and store data
- Policies that configure automatic backups for the resources in your organization's accounts
- Integration and support for IAM
- Integration with other AWS services
- Global access
- Data replication that is eventually consistent
- No cost for use

For more information about inheritance for SCPs, see "SCP evaluation" in the *AWS Organizations User Guide* at https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_inheritance_auth.html.

[Image Description: An AWS organization contains a management account, which has two OUs. Each of these OUs has one child AWS account and child OU. Each of these child OUs has multiple child AWS accounts. A policy is applied to a top OU and is active on all child AWS accounts and child OUs. **End description.]**



An SCP is a type of organization policy that you can use to manage permissions in your organization. SCPs have the following characteristics:

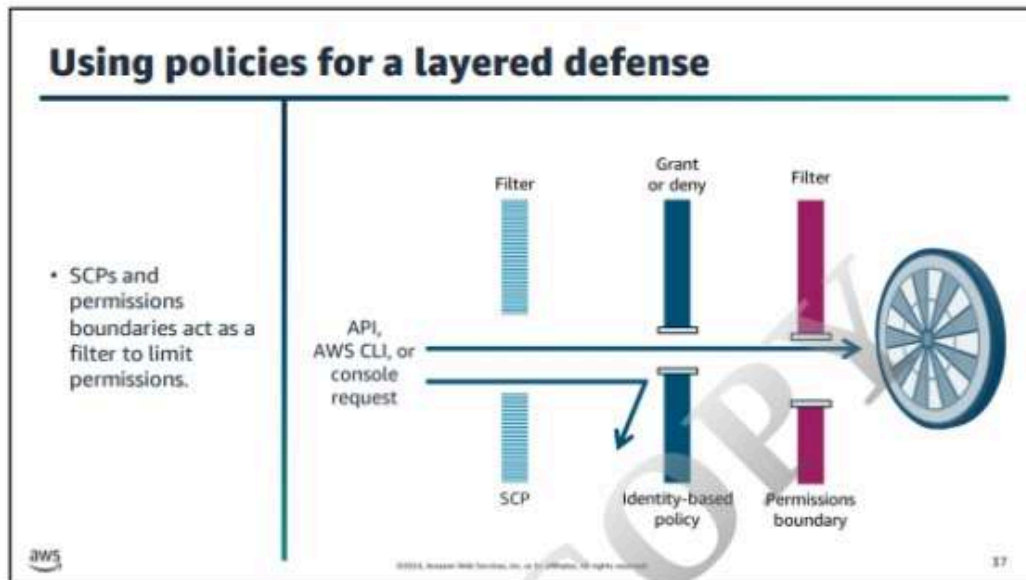
- Offer central control over the maximum available permissions for all accounts in your organization
- Help your accounts stay within your organization's access control guidelines
- Are available only in an organization that has all features turned on

SCPs aren't available if your organization turns on only the consolidated billing features.

Attaching an SCP to an Organizations entity (root, OU, or account) defines a guardrail. SCPs set limits on the actions that the IAM users and roles in the affected accounts can perform. To grant permissions, you must attach identity-based policies or resource-based policies to IAM users, or to the resources in your organization's accounts. When an IAM user or role belongs to an account that is a member of an organization, the SCPs limit the user's or role's effective permissions.

In the example, an SCP allows all EC2 and S3 actions. A collection of IAM identity-based permissions allows all EC2 and IAM actions. The effective allowed permissions for the IAM identity are all EC2 actions. It excludes both S3 and IAM actions because they are not explicitly allowed in both policy types.

For more information about SCPs, see "How to Use Service Control Policies in AWS Organizations" in the *AWS Security Blog* at <https://aws.amazon.com/blogs/security/how-to-use-service-control-policies-in-aws-organizations/>.



When a principal tries to use the console, the AWS API, or the AWS CLI, that principal sends a request to AWS. With AWS, you can configure several resources to determine whether to grant or deny the request.

In this example, you observe the following layers of defense:

- First, the action must be allowed by any SCPs that are configured for the organization.
- Next, the identity-based policy must allow and not explicitly deny the action.
- Finally, the action must be included within any applied permissions boundaries.

In an IAM entity (user or role), a permissions boundary allows only the actions that both its identity-based policies and its permissions boundaries allow. This practice adds an additional layer to protect against the creation of an IAM identity-based policy that allows overly permissive actions for that entity.

For more information, see “Policy evaluation logic” at

https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_evaluation-logic.html.

