

AWS Cloud Foundations

# Course Introduction

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



Welcome to AWS Academy Cloud Foundations!

# Module overview



## Topics

- Course objectives and overview
- AWS certification exam information
- AWS Documentation

## Activities

- AWS Documentation scavenger hunt

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

2

This course introduction module will address the following topics:

- Course objectives and overview
- AWS certification exam information
- AWS Documentation

The module concludes with a scavenger hunt activity that challenges you to locate information in the AWS Documentation.

## Module objectives



After completing this module, you should be able to:

- Recognize the purpose of the Cloud Foundations course
- Recognize the course structure
- Recognize the AWS certification process
- Navigate the AWS Documentation website

After completing this module, you should be able to:

- Recognize the purpose of the AWS Academy Cloud Foundations course
- Recognize the course structure
- Recognize the AWS certification process
- Navigate the AWS Documentation website

**Course Introduction**

## Section 1: Course objectives and overview

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



Introducing Section 1: Course objectives and overview.

# Course prerequisites



- General Required Knowledge
  - IT technical knowledge
  - IT business knowledge
- Preferred Knowledge
  - Familiarity with cloud computing concepts
  - Working knowledge of distributed systems
  - Familiarity with general networking concepts
  - Working knowledge of multi-tier architectures



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

5

To begin, it is important to have an understanding of the prerequisites for this course.

First, you should have general **IT technical knowledge**. The foundational computer literacy skills you will need to be successful in this course include a knowledge of basic computer concepts, file management, and a good understanding of the internet.

Second, you should have general **IT business knowledge**. This includes insight into how information technology is used by businesses and other organizations.

Additionally, to ensure success in this course, it is preferred that you have:

- A general familiarity with cloud computing concepts
- A working knowledge of distributed systems
- Familiarity with general networking concepts
- A working knowledge of multi-tier architectures

# Course objectives



After completing this course, you should be able to:

- Define the AWS Cloud.
- Explain the AWS pricing philosophy.
- Identify the global infrastructure components of AWS.
- Describe security and compliance measures of the AWS Cloud including AWS Identity and Access Management (IAM).
- Create an AWS Virtual Private Cloud (Amazon VPC).
- Demonstrate when to use Amazon Elastic Compute Cloud (EC2), AWS Lambda and AWS Elastic Beanstalk.
- Differentiate between Amazon S3, Amazon EBS, Amazon EFS and Amazon S3 Glacier.
- Demonstrate when to use AWS Database services including Amazon Relational Database Service (RDS), Amazon DynamoDB, Amazon Redshift, and Amazon Aurora.
- Explain AWS Cloud architectural principles.
- Explore key concepts related to Elastic Load Balancing (ELB), Amazon CloudWatch, and Auto Scaling.



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

6

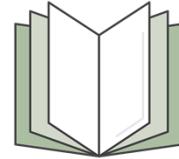
After completing this course, you should be able to:

- Define the AWS Cloud.
- Explain the AWS pricing philosophy.
- Identify the global infrastructure components of AWS.
- Describe security and compliance measures of the AWS Cloud including AWS Identity and Access Management (IAM).
- Create an AWS Virtual Private Cloud (Amazon VPC).
- Demonstrate when to use Amazon Elastic Compute Cloud (EC2), AWS Lambda and AWS Elastic Beanstalk.
- Differentiate between Amazon S3, Amazon EBS, Amazon EFS and Amazon S3 Glacier.
- Demonstrate when to use AWS Database services including Amazon Relational Database Service (RDS), Amazon DynamoDB, Amazon Redshift, and Amazon Aurora.
- Explain AWS Cloud architectural principles.
- Explore key concepts related to Elastic Load Balancing (ELB), Amazon CloudWatch, and Auto Scaling.

# Course outline



- Module 1: Cloud Concepts Overview
- Module 2: Cloud Economics and Billing
- Module 3: AWS Global Infrastructure Overview
- Module 4: AWS Cloud Security
- Module 5: Networking and Content Delivery
- Module 6: Compute
- Module 7: Storage
- Module 8: Databases
- Module 9: Cloud Architecture
- Module 10: Automatic Scaling and Monitoring



7

To achieve the course objectives, the course explores the following topics:

- Cloud concepts
- Cloud economics and billing
- AWS Global Infrastructure
- AWS Cloud security
- Networking and content delivery
- Compute
- Storage
- Databases
- Cloud architecture
- Automatic scaling and monitoring

The next ten slides provide more detail on what subtopics are covered in each module.

# Module 1: Cloud Concepts Overview



## Module sections:

- Introduction to cloud computing
- Advantages of cloud computing
- Introduction to Amazon Web Services (AWS)
- Moving to the AWS Cloud – The AWS Cloud Adoption Framework (AWS CAF)



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

8

In this module, Section 1 introduces **cloud computing**.

In Section 2, you learn about the **advantages that cloud computing provides** over a traditional, on-premises computing model.

In Section 3, you learn about what **AWS** is and the broad range of AWS products and services. You become familiar with the idea that AWS services are designed to work together to build solutions that meet business goals and technology requirements.

The module concludes with Section 4, which is about the **AWS Cloud Adoption Framework** (AWS CAF). It covers the fundamental changes that must be supported for an organization to successfully migrate its IT portfolio to the cloud.

## Module 2: Cloud Economics and Billing



### Module sections:

- Fundamentals of pricing
- Total Cost of Ownership
- AWS Organizations
- AWS Billing and Cost Management
- Technical support



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

9

The purpose of this module is to introduce you to the business advantages of moving to the cloud.

Section 1 describes the principles for **how AWS sets prices** for the various services. This includes the AWS pricing model and a description of the [AWS Free Tier](#).

Section 2 describes the **Total Cost of Ownership** and how customers can reduce their overall costs by moving IT services to the cloud. The section outlines four types of costs that are reduced by using cloud computing, and provides examples that illustrate each of these types.

Section 3 describes how customers can use AWS Organizations to manage their costs.

Section 4 describes **billing** and the components of the AWS Billing dashboard. This section includes a demonstration of how customers can use the dashboard to understand and manage their costs.

Finally, Section 5 describes the four different options for **AWS Technical Support**: Basic Support, Developer Support, Business Support, and Enterprise Support. The section also includes an activity that will help you understand the benefits of each support option.

### Module sections:

- AWS Global Infrastructure
- AWS services and service category overview



Module 3 provides an overview of the AWS global infrastructure.

In Section 1, you are introduced to the major parts of the **AWS Global Infrastructure**, including Regions, Availability Zones, the network infrastructure, and Points of Presence.

In Section 2, you are shown a listing of all the **AWS service categories**, and then you are provided with a listing of each of the services that this course will discuss. The module ends with an AWS Management Console clickthrough activity.

# Module 4: AWS Cloud Security



## Module sections:

- 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



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

11

This module provides an introduction to the AWS approach to security.

In Section 1, you are introduced to the **AWS shared responsibility model**, which specifies which responsibilities belong to the customer and which responsibilities belong to AWS.

Section 2 introduces you to the key concepts of **AWS Identity and Access Management (IAM)**, including users, groups, policies, and roles.

Section 3 provides guidance on **how to secure a new AWS account**. It discusses how you should avoid using the AWS account root user for day-to-day activities. It also discusses best practices, such as creating IAM users that have multi-factor authentication (MFA) enabled.

Section 3 highlights other ways to **secure accounts**. It discusses the security-related features of AWS Organizations, which include service control policies. This section also discusses AWS Shield, Amazon Cognito, and AWS Key Management Service (AWS KMS).

Section 5 discusses how to **secure data on AWS**. Topics include encryption of data at rest

and data in transit, and discusses options for securing data that is stored on Amazon Simple Storage Service (Amazon S3).

Finally, Section 6 discusses how AWS supports customer efforts to deploy solutions that are in **compliance** with laws and regulations. It also discusses the certifications that AWS maintains and AWS services—such as AWS Config and AWS Artifact—that support compliance.

## Module sections:

- Networking basics
- Amazon VPC
- VPC networking
- VPC security
- Amazon Route 53
- Amazon CloudFront



The purpose of this module is to introduce you to the fundamental of AWS networking and content delivery services: Amazon Virtual Private Cloud (Amazon VPC), Amazon Route 53, and Amazon CloudFront. You will have the opportunity to label a virtual private cloud (VPC) network architecture diagram, design a VPC, watch how a VPC is built, and finally build a VPC yourself.

Section 1 discusses **networking concepts** that will be referenced throughout the rest of the module: network, subnet, IPv4 and IPv6 addresses, and Classless Inter-Domain Routing (CIDR) notation.

Section 2 provides an overview of the key terminology and features of **Amazon VPC**, which you must be familiar with when you design and build your own virtual private clouds (VPCs).

In Section 3, you learn about several important **VPC networking** options: internet gateway, network address translation (NAT) gateway, VPC endpoints, VPC sharing, VPC peering, AWS Site-to-Site VPN, AWS Direct Connect, and AWS Transit Gateway.

In Section 4, you learn **how to secure VPCs** with network access control lists (network ACLs) and security groups.

Section 5 covers Domain Name System (DNS) resolution and **Amazon Route 53**. It also covers the topic of DNS failover, which introduces the topic of high availability that you will learn about in more detail in module 10.

Finally, section 6 covers the features and benefits of **Amazon CloudFront**.

# Module 6: Compute



## Module sections:

- Compute services overview
- Amazon EC2
- Amazon EC2 cost optimization
- Container services
- Introduction to AWS Lambda
- Introduction to AWS Elastic Beanstalk



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

13

This module provides an introduction to many of the compute services offered by AWS.

Section 1 provides a high-level, **compute services overview**.

Section 2 introduces you to the key concepts of **Amazon Elastic Compute Cloud (Amazon EC2)**, including Amazon Machine Images (AMIs), instance types, network settings, user data scripts, storage options, security groups, key pairs, instance lifecycle phases, Elastic IP addresses, instance metadata, and the benefits of using Amazon CloudWatch for monitoring.

Section 3 focuses on the four pillars of **cost optimization**, with an emphasis on cost optimization as it relates to Amazon EC2.

Section 4 covers **container services**. It introduces Docker and the differences between virtual machines and containers. It then discusses Amazon Elastic Container Service (Amazon ECS), AWS Fargate, Kubernetes, Amazon Elastic Kubernetes Service (Amazon EKS), and Amazon Elastic Container Registry (Amazon ECR).

Section 5 introduces serverless computing with **AWS Lambda**. Event sources and Lambda

function configuration basics are introduced, and the section ends with examples of a schedule-based Lambda function and an event-based Lambda function.

Finally, Section 6 describes the advantages of using **AWS Elastic Beanstalk** for web application deployments. It concludes with a hands-on activity where you deploy a simple web application to Elastic Beanstalk.

# Module 7: Storage



## Module sections:

- Amazon Elastic Block Store (Amazon EBS)
- Amazon Simple Storage Service (Amazon S3)
- Amazon Elastic File System (Amazon EFS)
- Amazon Simple Storage Service Glacier



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

14

Module 7 introduces you to the various options for storing data with AWS. The module provides an overview of storage services—which are based on four different storage technologies—so that you can choose a storage service for various use cases.

Section 1 provides you with an overview of the functionality of **Amazon Elastic Block Store (Amazon EBS)** and a summary of common use cases. It also introduces the concept of block versus object storage, and how to interact with Amazon EBS through the AWS Management Console.

Section 2 provides an overview of the functionality of **Amazon Simple Storage Service (Amazon S3)** and a summary of common use cases. It also describes how Amazon S3 scales as demand grows and discusses the concept of data redundancy. The section also contains a general overview of Amazon S3 pricing.

Section 3 starts with an overview of the functionality of **Amazon Elastic File Store (Amazon EFS)** and a summary of common use cases. It also provides an overview of the Amazon EFS architecture and a list of common Amazon EFS resources.

Finally, in Section 4, you are provided an overview of the functionality of **Amazon Simple Storage Service Glacier** and a summary of common use cases. This last section also

describes the lifecycle of migrating data from Amazon S3 to Amazon S3 Glacier.

# Module 8: Databases



## Module sections:

- Amazon Relational Database Service (Amazon RDS)
- Amazon DynamoDB
- Amazon Redshift
- Amazon Aurora



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

15

This module introduces you to four of the most commonly used AWS database services, with an emphasis on differentiating which database service to select for various use cases.

Section 1 provides an overview of the **Amazon Relational Database Service (Amazon RDS)**. It describes the difference between a managed and unmanaged service, and provides an overview of how to provide a highly available Amazon RDS implementation.

In Section 2, an overview of the **Amazon DynamoDB** services is provided. The section also describes how DynamoDB uses data partitioning to address scenarios that call for high data volumes and the ability to scale out on demand.

Section 3 provides an overview of **Amazon Redshift**. The section describes the parallel processing architecture of Amazon Redshift, and how this architecture supports processing very large datasets. It also reviews some of the more common use cases for Amazon Redshift.

Finally, Section 4 provides an overview of **Amazon Aurora**. The module describes the use cases where Amazon Aurora is a better solution than Amazon RDS. It also discusses how Amazon Aurora provides a more resilient database solution through the use of multiple Availability Zones.

# Module 9: Cloud Architecture



## Module sections:

- AWS Well-Architected Framework
- Reliability and availability
- AWS Trusted Advisor



The purpose of this module is to introduce you to designing and building cloud architectures according to best practices.

In Section 1, you learn about the **AWS Well-Architected Framework** and its purpose, how the framework is organized, and its design principles and best practices. You will also learn how to use it to design a cloud architecture solution that is secure, performant, resilient, and efficient. Finally, this section also introduces the AWS Well-Architected Tool, which can be used to evaluate your architectural designs against AWS Well-Architected Framework best practices.

In Section 2, you learn about **reliability and high availability**, which are two factors to consider when you design an architecture that can withstand failure.

In Section 3, you learn about **AWS Trusted Advisor**. You can use this tool to evaluate and improve your AWS environment when you implement your architectural designs.

### Module sections:

- Elastic Load Balancing
- Amazon CloudWatch
- Amazon EC2 Auto Scaling



The purpose of this module is to introduce you to three fundamental AWS services that can be used together to build dynamic, scalable architectures.

Section 1 introduces you to **Elastic Load Balancing**, which is a service that automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions.

Section 2 introduces you to **Amazon CloudWatch**, which is a service that provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health.

Finally, Section 3 introduces you to the **Amazon EC2 Auto Scaling** features that help you maintain application availability and enable you to automatically add or remove EC2 instances according to conditions that you define.

**Course Introduction**

## Section 2: AWS certification exam information

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



Introducing Section 2: AWS certification exam information.

# AWS certification exams



## Available AWS Certifications

aws certified

Updated May 2019

### Professional

Two years of comprehensive experience designing, operating, and troubleshooting solutions using the AWS Cloud

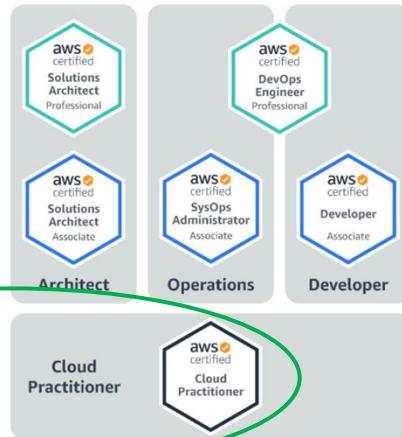
### Associate

One year of experience solving problems and implementing solutions using the AWS Cloud

*This course helps prepare you for the AWS Cloud Practitioner certification exam*

### Foundational

Six months of fundamental AWS Cloud and industry knowledge



### Specialty

Technical AWS Cloud experience in the Specialty domain as specified in the exam guide



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

19

AWS Certification helps learners build credibility and confidence by validating their cloud expertise with an industry-recognized credential, and it helps organizations identify skilled professionals who can lead cloud initiatives by using AWS.

You must earn a passing score via a proctored exam to earn an AWS Certification. After receiving a passing score, you will receive your certification credentials.

AWS Certification does not publish a list of all services or features that are covered in a certification exam. However, the exam guide for each exam lists the current topic areas and objectives covered in the exam. Exam guides can be found on the [Prepare for Your AWS Certification Exam](#) webpage.

You will be required to update your certification (or recertify) every 3 years. View the [AWS Certification Recertification](#) page for more details.

The information on this slide is current as of November 2019. However, exams are frequently updated and the details regarding which exams are available—and what is tested by each exam—are subject to change.

For the latest AWS certification exam information, go to

<https://aws.amazon.com/certification/>.

# AWS Certified Cloud Practitioner exam



- Details about the exam—including how to register for it—are at <https://aws.amazon.com/certification/certified-cloud-practitioner/>

- Download and carefully read the [AWS Certified Cloud Practitioner Exam Guide](#)

- Download the [sample exam questions](#)

- See the recommended path to attain the certification at <https://aws.amazon.com/training/path-cloudpractitioner/>

- AWS Academy Cloud Foundations covers much of the same material found in the Cloud Practitioner Essentials course, but in greater depth.

- There is additional free digital training available at [aws.training](#)



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

20

The **AWS Certified Cloud Practitioner certification** provides individuals in various cloud and technology roles with a way to validate their AWS Cloud knowledge and enhance their professional credibility. This exam covers four domains, including cloud concepts, security, technology, and billing and pricing.

The AWS Certified Cloud Practitioner exam is the only AWS certification exam that is classified as *foundational* (as shown on the previous slide). It is often the first AWS exam that IT professionals attempt to obtain.

Though this **AWS Academy Cloud Foundations** course is not listed in the AWS Certified Cloud Practitioner Exam Guide as one of the AWS training options recommended to prepare for the exam, this course does cover many of the same topics that are covered by AWS commercial courses, such as AWS Technical Essentials, AWS Business Essentials, and AWS Cloud Practitioner Essentials. Therefore, the AWS Academy Cloud Foundations course you are taking now is a good way to help prepare yourself to take this exam.

The services included in the AWS Certified Cloud Practitioner exam change as new services are added. At a minimum, you should be able to describe the overall functionality of a broad range of AWS services before taking the exam. For an overview of

the AWS services see the [Amazon Web Services Cloud Platform](#) section of the Overview of Amazon Web Services whitepaper.

**Course Introduction**

## Section 3: AWS Documentation

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



Introducing Section 3: AWS Documentation.

- Find user guides, developer guides, API references, tutorials, and more.
  - <https://docs.aws.amazon.com/>
- **Whitepapers** are also available at <https://aws.amazon.com/whitepapers/>, including these which are recommended reading for the AWS Cloud Practitioner exam:
  - [Overview of Amazon Web Services](#)
  - [Architecting for the Cloud: AWS Best Practices](#)
  - [How AWS Pricing Works](#)
  - [The Total Cost of \(Non\) Ownership of Web Applications in the Cloud](#)

AWS provides extensive and detailed documentation for each AWS service. Guides and application programming interface (API) references are organized by service category. There are also many general resources and tutorials that can be accessed from the AWS Documentation pages. General resources include case studies, an A-to-Z glossary of AWS terms, whitepapers, FAQs, information about AWS Training and Certification, and more.

Also, each SDK and toolkit has documentation—for example, the AWS Command Line Interface (AWS CLI), the boto3 libraries for AWS SDK for Python, and many others.

**AWS whitepapers** and guides can be filtered by product, category, or industry, so that you can find the information that is most relevant to your needs.

## Activity - AWS Documentation Scavenger Hunt

23



- Navigate the AWS Documentation website
- Start from the main page at <https://docs.aws.amazon.com>
- Five challenge questions for the class appear in the following slides



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

In this educator-led activity, you will be challenged to access the AWS Documentation pages and practice locating specific information.

## AWS Documentation Scavenger Hunt – Question 1

24



- Question #1: What guides and references exist for the Amazon EC2 service?

- Answer:

- User Guides for Linux and Windows
- API Reference
- AWS CLI Reference
- EC2 Instance Connect Reference
- User Guide for Auto Scaling
- VM Import/Export User Guide

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

What guides and references exist for the Amazon EC2 service?

Browse to <https://docs.aws.amazon.com> and see if you can identify at least six guides or references.

## AWS Documentation Scavenger Hunt – Question 2

25



- Question #2: Can you find the documentation that describes how to create an Amazon S3 bucket?

- Answer:

- From <https://docs.aws.amazon.com/> click **S3**
- Click the **Getting Started Guide**
- Click **Create a Bucket**

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

Can you find the documentation that describes how to create an Amazon S3 bucket?

Browse to <https://docs.aws.amazon.com> and figure out how to navigate to documentation that provides this information. Be prepared to discuss your findings with the class.

## AWS Documentation Scavenger Hunt – Question 3

26



- Question #3: Can you find a one-sentence summary of the AWS Cloud9 service?

- Answer:

- AWS Cloud9 is a cloud-based integrated development environment (IDE) that you use to write, run, and debug code.

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

Can you find a one sentence summary of the AWS Cloud9 service?

Starting at <https://docs.aws.amazon.com>, see if you can find a page that provides the summary. Be prepared to share your findings.

## AWS Documentation Scavenger Hunt – Question 4

27



- Question #4: Which programming languages does the AWS Lambda service API support?

- Answer:

- From the main AWS Documentation page, click the **AWS Lambda** link
- Click the **API Reference** link
- Click **Getting Started > Tools** to find a table that lists the following languages:  
**Node.js, Java, C#, Python, Ruby, Go, and PowerShell**

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

Which programming languages does the AWS Lambda service API support?

Browse to <https://docs.aws.amazon.com> and figure out how to navigate to documentation that provides this information. Be prepared to discuss your findings with the class.

## AWS Documentation Scavenger Hunt – Question 5

28



- Question #5: Find the tutorial that describes how to run a serverless Hello World application, then scroll through the documented steps. What two AWS services does the tutorial have you use?

- Answer:

- From the main AWS Documentation page, click **Tutorials and Projects**
- In the **Websites & Web Apps** area, click the tutorial.
- The tutorial has you use **AWS Lambda** and **Amazon CloudWatch**.

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

Find the tutorial that describes how to run a serverless Hello World application, then scroll through the documented steps. What two AWS services does the tutorial have you use?

Browse to <https://docs.aws.amazon.com> and figure out how to navigate to documentation that provides this information. Be prepared to discuss your findings with the class.

**Course Introduction**

## Module wrap-up

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



It's now time to review the module, and wrap up with a module summary and a listing of additional resources.

## Module summary



In summary, in this module, you learned how to:

- Recognize the purpose of the AWS Academy Cloud Foundations course
- Recognize the course structure
- Recognize the AWS certification process
- Navigate the AWS Documentation website

In summary, in this module, you learned how to:

- Recognize the purpose of the AWS Academy Cloud Foundations course
- Recognize the course structure
- Recognize the AWS certification process
- Navigate the AWS Documentation website

## Additional resources



- [AWS Certification](#)
- [AWS Certified Cloud Practitioner](#)
- [AWS Documentation](#)

The following resources provide more detail on the topics that are discussed in this module:

- [AWS Certification](#)
- [AWS Certified Cloud Practitioner](#)
- [AWS Documentation](#)

# Thank you

© 2019 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections or feedback on the course, please email us at: [aws-course-feedback@amazon.com](mailto:aws-course-feedback@amazon.com). For all other questions, contact us at: <https://aws.amazon.com/contact-us/aws-training/>. All trademarks are the property of their owners.



Thank you for completing this module.

AWS Academy Cloud Foundations

# Module 1: Cloud Concepts Overview

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



Welcome to Module 1: Cloud Concepts Overview.

# Module overview



## Topics

- Introduction to cloud computing
- Advantages of cloud computing
- Introduction to Amazon Web Services (AWS)
- AWS Cloud Adoption Framework (AWS CAF)



## Knowledge check

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

2

This module addresses the following topics:

- Introduction to cloud computing
- Advantages of cloud computing
- Introduction to Amazon Web Services (AWS)
- AWS Cloud Adoption Framework (AWS CAF)

Finally, you will be asked to complete a knowledge check that will be used to test your understanding of the key concepts that are covered in this module.

## Module objectives



After completing this module, you should be able to:

- Define different types of cloud computing models
- Describe six advantages of cloud computing
- Recognize the main AWS service categories and core services
- Review the AWS Cloud Adoption Framework (AWS CAF)

After completing this module, you should be able to:

- Define different types of cloud computing
- Describe six advantages of cloud computing
- Recognize the main AWS service categories and core services
- Review the AWS Cloud Adoption Framework (AWS CAF)

Module 1: Cloud Concepts Overview

## Section 1: Introduction to cloud computing

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



## Section 1: Introduction to cloud computing

# What is cloud computing?



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

5

What does cloud computing mean to you?

Take a moment to think of what cloud computing means to you and write a short sentence.

# Cloud computing defined



**Cloud computing** is the **on-demand** delivery of compute power, database, storage, applications, and other IT resources **via the internet** with **pay-as-you-go** pricing.



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

6

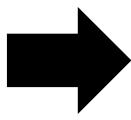
**Cloud computing** is the on-demand delivery of compute power, database, storage, applications, and other IT resources via the internet with pay-as-you-go pricing. These resources run on server computers that are located in large data centers in different locations around the world. When you use a cloud service provider like AWS, that service provider owns the computers that you are using. These resources can be used together like building blocks to build solutions that help meet business goals and satisfy technology requirements.

To learn more about cloud computing and how it works, see [this AWS webpage](#).

# Infrastructure as software



Cloud computing enables you to **stop thinking of your infrastructure as hardware**, and instead **think of (and use) it as software**.



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

1

Cloud computing enables you to stop thinking of your infrastructure as hardware, and instead think of (and use) it as software. But what does this mean?

## Traditional computing model



- Infrastructure as hardware
- Hardware solutions:
  - Require space, staff, physical security, planning, capital expenditure
  - Have a long hardware procurement cycle
  - Require you to provision capacity by guessing theoretical maximum peaks

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

8

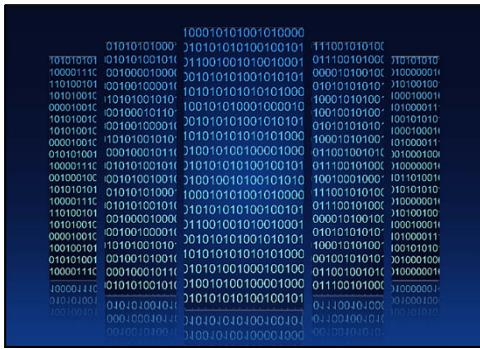
In the traditional computing model, infrastructure is thought of as hardware. Hardware solutions are physical, which means they require space, staff, physical security, planning, and capital expenditure.

In addition to significant upfront investment, another prohibitive aspect of traditional computing is the long hardware procurement cycle that involves acquiring, provisioning, and maintaining on-premises infrastructure.

With a hardware solution, you must ask if there is enough resource capacity or sufficient storage to meet your needs, and you provision capacity by guessing theoretical maximum peaks. If you don't meet your projected maximum peak, then you pay for expensive resources that stay idle. If you exceed your projected maximum peak, then you don't have sufficient capacity to meet your needs. And if your needs change, then you must spend the time, effort, and money required to implement a new solution.

For example, if you wanted to provision a new website, you would need to buy the hardware, rack and stack it, put it in a data center, and then manage it or have someone else manage it. This approach is expensive and time-consuming.

# Cloud computing model



- Infrastructure as software
- Software solutions:
  - Are flexible
  - Can change more quickly, easily, and cost-effectively than hardware solutions
  - Eliminate the undifferentiated heavy-lifting tasks

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

9

By contrast, cloud computing enables you to think of your infrastructure as software. Software solutions are flexible. You can select the cloud services that best match your needs, provision and terminate those resources on-demand, and pay for what you use. You can elastically scale resources up and down in an automated fashion. With the cloud computing model, you can treat resources as temporary and disposable. The flexibility that cloud computing offers enables businesses to implement new solutions quickly and with low upfront costs.

Compared to hardware solutions, software solutions can change much more quickly, easily, and cost-effectively.

Cloud computing helps developers and IT departments avoid undifferentiated work like procurement, maintenance, and capacity planning, thus enabling them to focus on what matters most.

As cloud computing has grown in popularity, several different service models and deployment strategies have emerged to help meet the specific needs of different users. Each type of cloud service model and deployment strategy provides you with a different level of control, flexibility, and management. Understanding the differences between these cloud service models and deployment strategies can help you decide what set of services is right for your needs.

# Cloud service models



IaaS  
(infrastructure as a service)

PaaS  
(platform as a service)

SaaS  
(software as a service)



More control  
over IT resources

Less control  
over IT resources

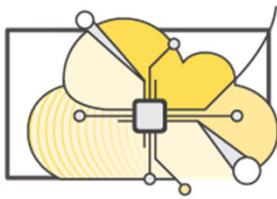
© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

10

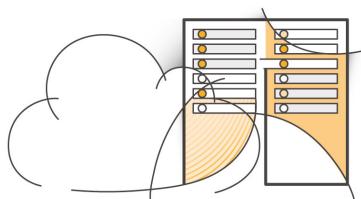
There are three main cloud service models. Each model represents a different part of the cloud computing stack and gives you a different level of control over your IT resources:

- **Infrastructure as a service (IaaS):** Services in this category are the basic building blocks for cloud IT and typically provide you with access to networking features, computers (virtual or on dedicated hardware), and data storage space. IaaS provides you with the highest level of flexibility and management control over your IT resources. It is the most similar to existing IT resources that many IT departments and developers are familiar with today.
- **Platform as a service (PaaS):** Services in this category reduce the need for you to manage the underlying infrastructure (usually hardware and operating systems) and enable you to focus on the deployment and management of your applications.
- **Software as a service (SaaS):** Services in this category provide you with a completed product that the service provider runs and manages. In most cases, *software as a service* refers to end-user applications. With a SaaS offering, you do not have to think about how the service is maintained or how the underlying infrastructure is managed. You need to think only about how you plan to use that particular piece of software. A common example of a SaaS application is web-based email, where you can send and receive email without managing feature additions to the email product or maintaining the servers and operating systems that the email program runs on.

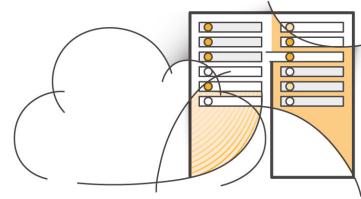
# Cloud computing deployment models



Cloud



Hybrid



On-premises  
(private cloud)

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

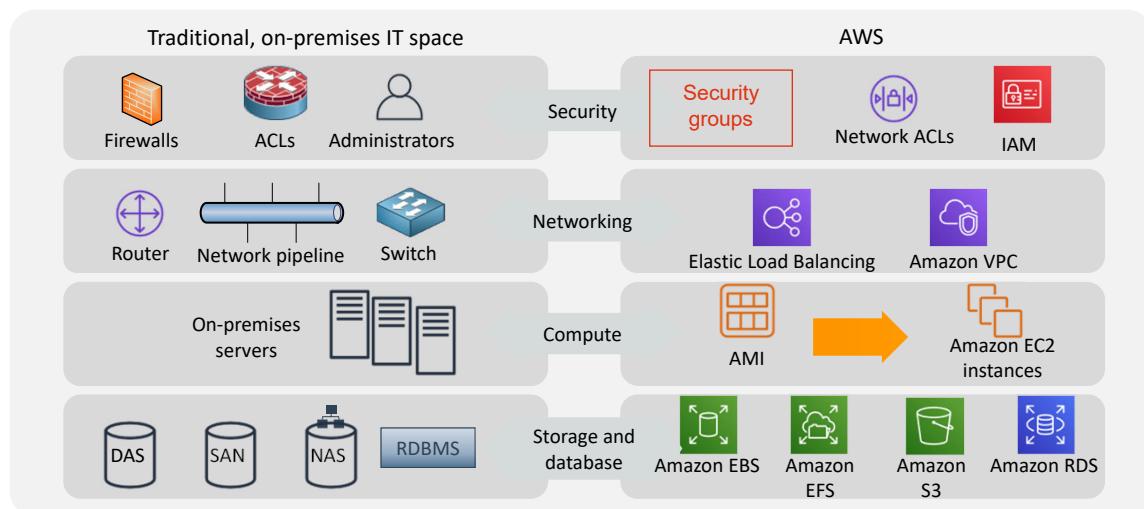
11

There are three main cloud computing deployment models, which represent the cloud environments that your applications can be deployed in:

- **Cloud:** A cloud-based application is fully deployed in the cloud, and all parts of the application run in the cloud. Applications in the cloud have either been created in the cloud or have been migrated from an existing infrastructure to take advantage of the [benefits of cloud computing](#). Cloud-based applications can be built on low-level infrastructure pieces or they can use higher-level services that provide abstraction from the management, architecting, and scaling requirements of core infrastructure.
- **Hybrid:** A hybrid deployment is a way to connect infrastructure and applications between cloud-based resources and existing resources that are not located in the cloud. The most common method of hybrid deployment is between the cloud and existing on-premises infrastructure. This model enables an organization to extend and grow their infrastructure into the cloud while connecting cloud resources to internal systems.
- **On-premises:** Deploying resources on-premises, using virtualization and resource management tools, is sometimes called *private cloud*. While on-premises deployment does not provide many of the benefits of cloud computing, it is sometimes sought for its ability to provide dedicated resources. In most cases, this deployment model is the same as legacy IT infrastructure, but it might also use application management and

virtualization technologies to increase resource utilization.

## Similarities between AWS and traditional IT



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

12

There are many similarities between AWS and the traditional, on-premises IT space:

- AWS security groups, network access control lists (network ACLs), and AWS Identity and Access Management (IAM) are similar to firewalls, access control lists (ACLs), and administrators.
- Elastic Load Balancing and Amazon Virtual Private Cloud (Amazon VPC) are similar to routers, network pipelines, and switches.
- Amazon Machine Images (AMIs) and Amazon Elastic Compute Cloud (Amazon EC2) instances are similar to on-premises servers.
- Amazon Elastic Block Store (Amazon EBS), Amazon Elastic File System (Amazon EFS), Amazon Simple Storage Service (Amazon S3), and Amazon Relational Database Service (Amazon RDS) are similar to direct attached storage (DAS), storage area networks (SAN), network attached storage (NAS), and a relational database management service (RDBMS).

With AWS services and features, you can do almost everything that you would want to do with a traditional data center.

## Section 1 key takeaways



13



- Cloud computing is the on-demand delivery of IT resources via the internet with pay-as-you-go pricing.
- Cloud computing enables you to think of (and use) your infrastructure as software.
- There are three cloud service models: IaaS, PaaS, and SaaS.
- There are three cloud deployment models: cloud, hybrid, and on-premises or private cloud.
- Almost anything you can implement with traditional IT can also be implemented as an AWS cloud computing service.

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

Some key takeaways from this section of the module include:

- Cloud computing is the on-demand delivery of IT resources via the internet with pay-as-you-go pricing.
- Cloud computing enables you to think of (and use) your infrastructure as software.
- There are three cloud service models: IaaS, PaaS, and SaaS.
- There are three cloud deployment models: cloud, hybrid, and on-premises or private cloud.
- There are many AWS service analogs for the traditional, on-premises IT space.

Module 1: Cloud Concepts Overview

## Section 2: Advantages of cloud computing

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



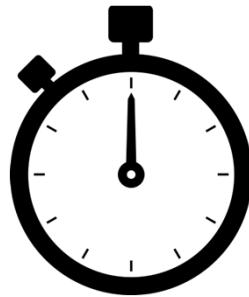
### Section 2: Advantages of cloud computing

Why are so many companies interested in moving to the cloud? This section presents six advantages of cloud computing.

## Trade capital expense for variable expense



Data center investment  
based on forecast



Pay only for the amount  
you consume

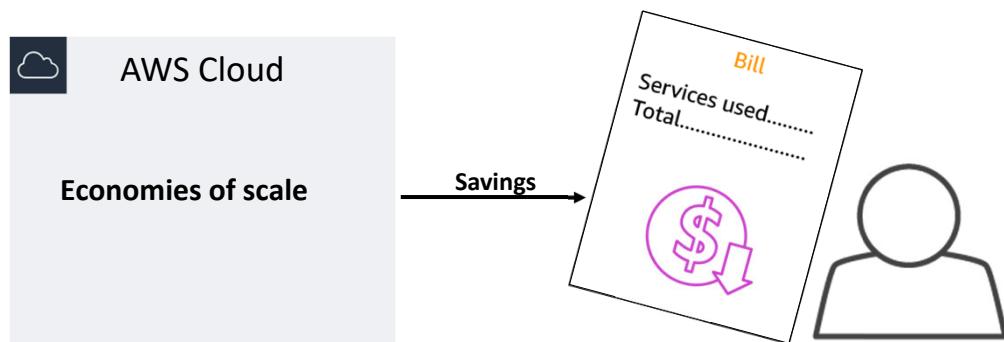
**Advantage #1—Trade capital expense for variable expense:** *Capital expenses (capex)* are funds that a company uses to acquire, upgrade, and maintain physical assets such as property, industrial buildings, or equipment. Do you remember the data center example in the traditional computing model where you needed to rack and stack the hardware, and then manage it all? You must pay for everything in the data center whether you use it or not.

By contrast, a *variable expense* is an expense that the person who bears the cost can easily alter or avoid. Instead of investing heavily in data centers and servers before you know how you will use them, you can pay only when you consume resources and pay only for the amount you consume. Thus, you save money on technology. It also enables you to adapt to new applications with as much space as you need in minutes, instead of weeks or days. Maintenance is reduced, so you can spend focus more on the core goals of your business.

# Massive economies of scale



Because of aggregate usage from all customers, AWS can achieve higher economies of scale and pass savings on to customers.

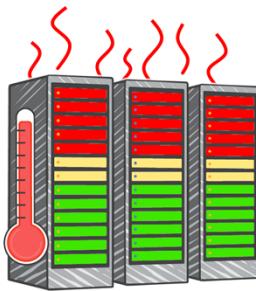


**Advantage #2—Benefit from massive economies of scale:** By using cloud computing, you can achieve a lower variable cost than you can get on your own. Because usage from hundreds of thousands of customers is aggregated in the cloud, providers such as AWS can achieve higher economies of scale, which translates into lower pay-as-you-go prices.

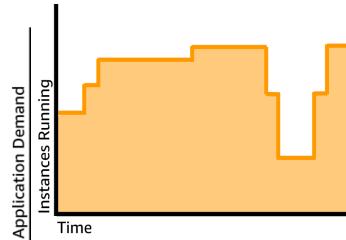
## Stop guessing capacity



Overestimated server capacity



Underestimated server capacity



Scaling on demand

**Advantage #3—Stop guessing capacity:** Eliminate guessing about your infrastructure capacity needs. When you make a capacity decision before you deploy an application, you often either have expensive idle resources or deal with limited capacity. With cloud computing, these problems go away. You can access as much or as little as you need, and scale up and down as required with only a few minutes' notice.

## Increase speed and agility



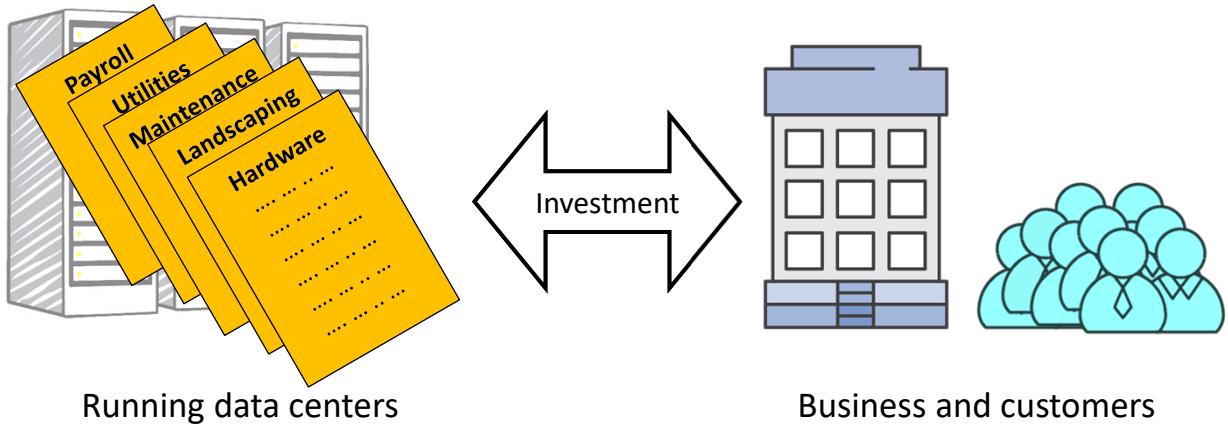
*Weeks* between wanting resources and having resources



*Minutes* between wanting resources and having resources

**Advantage #4—Increase speed and agility:** In a cloud computing environment, new IT resources are only a click away, which means that you reduce the time it takes to make those resources available to your developers from weeks to just minutes. The result is a dramatic increase in agility for the organization because the cost and time that it takes to experiment and develop are significantly lower.

## Stop spending money on running and maintaining data centers

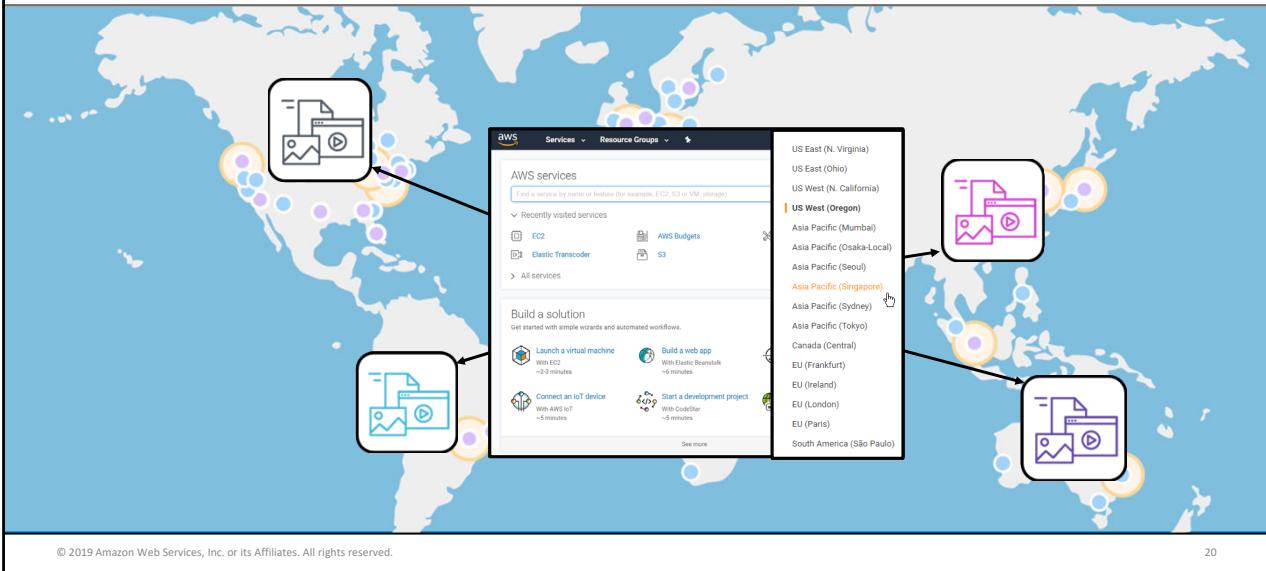


© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

19

**Advantage #5—Stop spending money on running and maintaining data centers:** Focus on projects that differentiate your business instead of focusing on the infrastructure. Cloud computing enables you to focus on your own customers instead of the heavy lifting of racking, stacking, and powering servers.

# Go global in minutes



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

20

**Advantage #6—Go global in minutes:** You can deploy your application in multiple AWS Regions around the world with just a few clicks. As a result, you can provide a lower latency and better experience for your customers simply and at minimal cost.

## Section 2 key takeaways



- Trade capital expense for variable expense
- Benefit from massive economies of scale
- Stop guessing capacity
- Increase speed and agility
- Stop spending money on running and maintaining data centers
- Go global in minutes

The key takeaways from this section of the module include the six advantages of cloud computing:

- Trade capital expense for variable expense
- Massive economies of scale
- Stop guessing capacity
- Increase speed and agility
- Stop spending money on running and maintaining data centers
- Go global in minutes

Module 1: Cloud Concepts Overview

## Section 3: Introduction to Amazon Web Services (AWS)

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

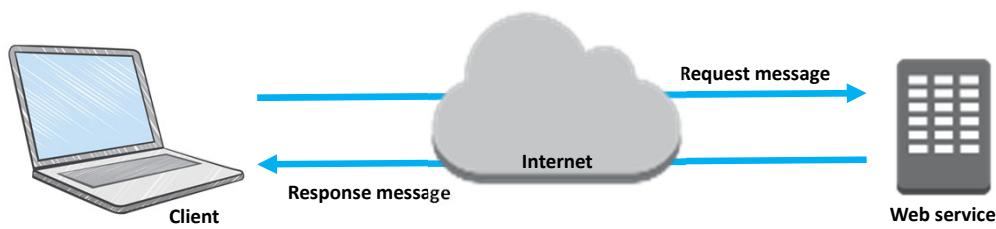


### Section 3: Introduction to Amazon Web Services (AWS)

# What are web services?



A **web service** is any piece of software that makes itself available over the internet and uses a **standardized format**—such as Extensible Markup Language (XML) or JavaScript Object Notation (JSON)—for the request and the response of an **application programming interface (API) interaction**.



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

23

In general, a web service is any piece of software that makes itself available over the internet or on private (intranet) networks. A web service uses a standardized format—such as Extensible Markup Language (XML) or JavaScript Object Notation (JSON)—for the request and the response of an application programming interface (API) interaction. It is not tied to any one operating system or programming language. It's self-describing via an interface definition file and it is discoverable.

# What is AWS?



- AWS is a **secure cloud platform** that offers a **broad set of global cloud-based products**.
- AWS provides you with **on-demand access** to compute, storage, network, database, and other IT resources and management tools.
- AWS offers **flexibility**.
- You **pay only for the individual services you need**, for **as long as you use them**.
- AWS services **work together** like building blocks.

Amazon Web Services (AWS) is a secure cloud platform that offers a broad set of global cloud-based products. Because these products are delivered over the internet, you have on-demand access to the compute, storage, network, database, and other IT resources that you might need for your projects—and the tools to manage them. You can immediately provision and launch AWS resources. The resources are ready for you to use in minutes.

AWS offers flexibility. Your AWS environment can be reconfigured and updated on demand, scaled up or down automatically to meet usage patterns and optimize spending, or shut down temporarily or permanently. The billing for AWS services becomes an operational expense instead of a capital expense.

AWS services are designed to work together to support virtually any type of application or workload. Think of these services like building blocks, which you can assemble quickly to build sophisticated, scalable solutions, and then adjust them as your needs change.

# Categories of AWS services



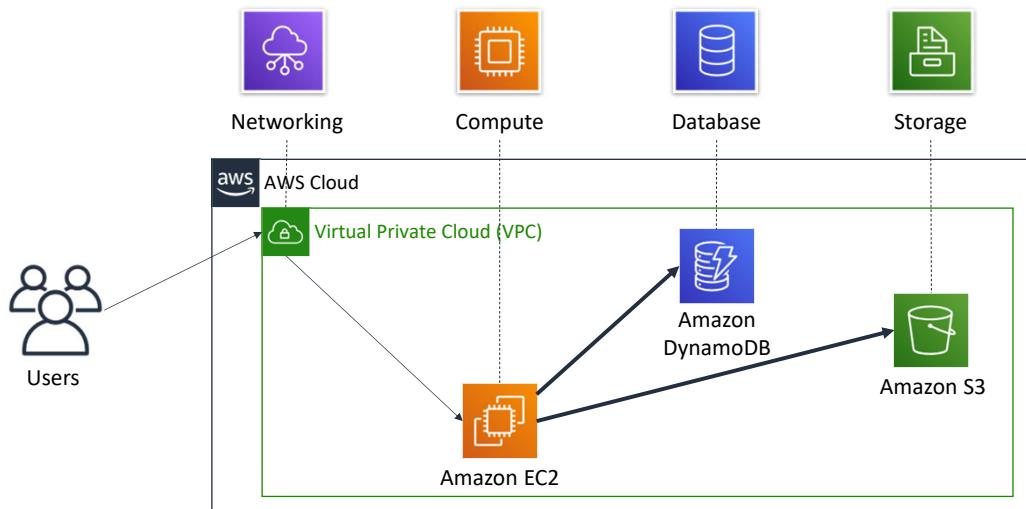
					
					
					
					

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

25

AWS services fall under different categories, and each category contains one or more services. You can select the services that you want from these different categories to build your solutions.

## Simple solution example



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

26

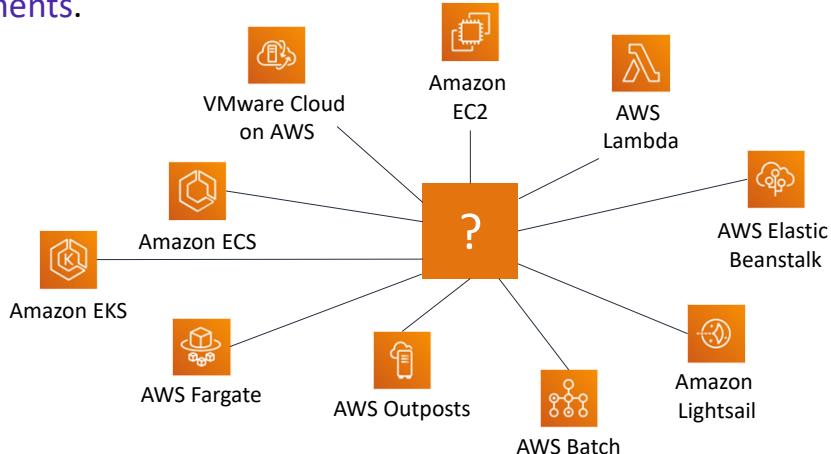
For example, say you're building a database application. Your customers might be sending data to your Amazon Elastic Compute Cloud (Amazon EC2) instances, which is a service in the compute category. These EC2 servers batch the data in one-minute increments and add an object per customer to Amazon Simple Storage Service (Amazon S3), the AWS storage service you've chosen to use. You can then use a nonrelational database like Amazon DynamoDB to power your application, for example, to build an index so that you can find all the objects for a given customer that were collected over a certain period. You might decide to run these services inside an Amazon Virtual Private Cloud (Amazon VPC), which is a service in the networking category.

The purpose of this simple example is to illustrate that you can select web services from different categories and use them together to build a solution (in this case, a database application). Of course, the solutions you build can be quite complex.

# Choosing a service



The service you select **depends on** your **business goals and technology requirements**.



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

27

Which service you choose to use will depend on your business goals and technology requirements. In the example you just looked at, the solution made use of Amazon EC2 as the compute service. However, that is only one of many compute services that AWS offers. Here are some other AWS compute offerings that you might choose to use for the following example use cases:

- [Amazon EC2](#): You want complete control over your AWS computing resources.
- [AWS Lambda](#): You want to run your code and not manage or provision servers.
- [AWS Elastic Beanstalk](#): You want a service that deploys, manages, and scales your web applications for you.
- [Amazon Lightsail](#): You need a lightweight cloud platform for a simple web application.
- [AWS Batch](#): You need to run hundreds of thousands of batch workloads.
- [AWS Outposts](#): You want to run AWS infrastructure in your on-premises data center.
- [Amazon Elastic Container Service](#) (Amazon ECS), [Amazon Elastic Kubernetes Service](#) (Amazon EKS), or [AWS Fargate](#): You want to implement a containers or microservices architecture.
- [VMware Cloud on AWS](#): You have an on-premises server virtualization platform that you want to migrate to AWS.

Similarly, there are a variety of services for you to choose from in the other categories, and the number of offerings keeps growing.

# Some Services



## Compute services –

- Amazon EC2
- AWS Lambda
- AWS Elastic Beanstalk
- Amazon EC2 Auto Scaling
- Amazon ECS
- Amazon EKS
- Amazon ECR
- AWS Fargate



## Storage services –

- Amazon S3
- Amazon S3 Glacier
- Amazon EFS
- Amazon EBS



## Management and Governance services –

- AWS Trusted Advisor
- AWS CloudWatch
- AWS CloudTrail
- AWS Well-Architected Tool
- AWS Auto Scaling
- AWS Command Line Interface
- AWS Config
- AWS Management Console
- AWS Organizations



## Security, Identity, and Compliance services –

- AWS IAM
- Amazon Cognito
- AWS Shield
- AWS Artifact
- AWS KMS



## Database services –

- Amazon RDS
- Amazon DynamoDB
- Amazon Redshift
- Amazon Aurora



## Networking and Content Delivery services –

- Amazon VPC
- Amazon Route 53
- Amazon CloudFront
- Elastic Load Balancing



## AWS Cost Management services –

- AWS Cost & Usage Report
- AWS Budgets
- AWS Cost Explorer



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

28

The array of AWS services can be intimidating as you start your journey into the cloud. This course focuses on some of the more common services in the following service categories: compute, storage, database, networking and content delivery, security, identity, and compliance, management and governance, and AWS cost management.

### Legend:

- Amazon Elastic Block Store (Amazon EBS)
- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon Elastic Container Service (Amazon ECS)
- Amazon Elastic File System (Amazon EFS)
- Amazon Elastic Kubernetes Service (Amazon EKS)
- Amazon Relational Database Service (Amazon RDS)
- Amazon Simple Storage Service (Amazon S3)
- Amazon Virtual Private Cloud (Amazon VPC)
- AWS Identity and Access Management (IAM)
- AWS Key Management Service (AWS KMS)

# Three ways to interact with AWS



## AWS Management Console

Easy-to-use graphical interface



## Command Line Interface (AWS CLI)

Access to services by discrete commands or scripts



## Software Development Kits (SDKs)

Access services directly from your code (such as Java, Python, and others)

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

29

You might wonder how to access the broad array of services that are offered by AWS. There are three ways to create and manage resources on the AWS Cloud:

- **AWS Management Console:** The console provides a rich graphical interface to a majority of the features offered by AWS. (Note: From time to time, new features might not have all of their capabilities included in the console when the feature initially launches.)
- **AWS Command Line Interface (AWS CLI):** The AWS CLI provides a suite of utilities that can be launched from a command script in Linux, macOS, or Microsoft Windows.
- **Software development kits (SDKs):** AWS provides packages that enable accessing AWS in a variety of popular programming languages. This makes it easy to use AWS in your existing applications and it also enables you to create applications that deploy and monitor complex systems entirely through code.

All three options are built on a common REST-like API that serves as the foundation of AWS.

To learn more about tools you can use to develop and manage applications on AWS, see [Tools to Build on AWS](#).

## Section 3 key takeaways



30



- AWS is a secure cloud platform that offers a broad set of global cloud-based products called services that are designed to work together.
- There are many categories of AWS services, and each category has many services to choose from.
- Choose a service based on your business goals and technology requirements.
- There are three ways to interact with AWS services.

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

The key takeaways from this section of the module include:

- AWS is a secure cloud platform that offers a broad set of global cloud-based products called services that are designed to work together.
- There are many categories of AWS services, and each category has many services to choose from.
- Choose a service based on your business goals and technology requirements.
- There are three ways to interact with AWS services.

**Module 1: Cloud Concepts Overview**

## **Section 4: Moving to the AWS Cloud – The AWS Cloud Adoption Framework (AWS CAF)**

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



### **Section 4: Moving to the AWS Cloud – The AWS Cloud Adoption Framework (AWS CAF)**

As you learned so far in this module, cloud computing offers many advantages over the traditional model. However, for most organizations, cloud adoption does not happen instantly. Technology is one thing, but an organization also consists of people and processes, and these three elements must all be in alignment for successful cloud adoption. Cloud computing introduces a significant shift in how technology is obtained, used, and managed. It also shifts how organizations budget and pay for technology services. Cloud adoption requires that fundamental changes are discussed and considered across an entire organization. It also requires that stakeholders across all organizational units—both within and outside IT—support these new changes. In this last section, you learn about the AWS CAF, which was created to help organizations design and travel an accelerated path to successful cloud adoption.

# AWS Cloud Adoption Framework (AWS CAF)



AWS CAF perspectives

- AWS CAF provides guidance and best practices to help organizations build a comprehensive approach to cloud computing across the organization and throughout the IT lifecycle to accelerate successful cloud adoption.

- AWS CAF is organized into six perspectives.
- Perspectives consist of sets of capabilities.

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

32

Each organization's cloud adoption journey is unique. However, in order for any organization to successfully migrate its IT portfolio to the cloud, three elements (that is, people, process, and technology) must be in alignment. Business and technology leaders in an organization must understand the organization's current state, target state, and the transition that is needed to achieve the target state so they can set goals and create processes for staff.

The AWS Cloud Adoption Framework (AWS CAF) provides guidance and best practices to help organizations identify gaps in skills and processes. It also helps organizations build a comprehensive approach to cloud computing—both across the organization and throughout the IT lifecycle—to accelerate successful cloud adoption.

At the highest level, the AWS CAF organizes guidance into six areas of focus, called *perspectives*. Perspectives span people, processes, and technology. Each perspective consists of a set of *capabilities*, which covers distinct responsibilities that are owned or managed by functionally related stakeholders.

Capabilities within each perspective are used to identify which areas of an organization require attention. By identifying gaps, prescriptive work streams can be created that support a successful cloud journey.

# Six core perspectives



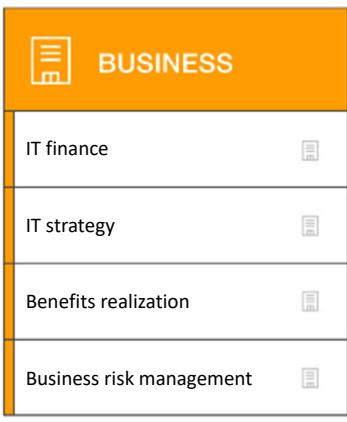
Focus on **business** capabilities



Focus on **technical** capabilities

In general, the Business, People, and Governance perspectives focus on business capabilities, while the Platform, Security, and Operations perspectives focus on technical capabilities.

# Business perspective



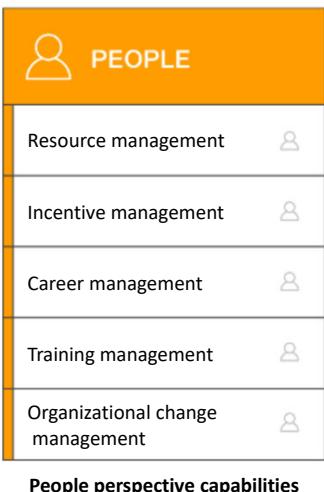
We must ensure that **IT is aligned with business needs**, and that IT investments can be traced to demonstrable business results.



Business managers, finance managers, budget owners, and strategy stakeholders

Stakeholders from the Business perspective (for example, business managers, finance managers, budget owners, and strategy stakeholders) can use the AWS CAF to create a strong business case for cloud adoption and prioritize cloud adoption initiatives. Stakeholders should ensure that an organization's business strategies and goals align with its IT strategies and goals.

# People perspective



People perspective capabilities

We must prioritize **training, staffing, and organizational changes** to build an agile organization.



Human resources, staffing, and people managers

Stakeholders from the People perspective (for example, human resources, staffing, and people managers) can use the AWS CAF to evaluate organizational structures and roles, new skill and process requirements, and identify gaps. Performing an analysis of needs and gaps can help prioritize training, staffing, and organizational changes to build an agile organization.

# Governance perspective



Governance perspective capabilities

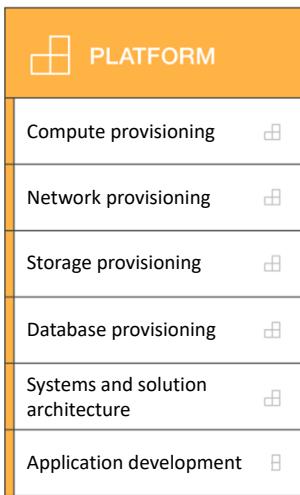
We must ensure that **skills and processes align IT strategy and goals with business strategy and goals** so the organization can maximize the business value of its IT investment and minimize business risks.



CIO, program managers, enterprise architects, business analysts, and portfolio managers

Stakeholders from the Governance perspective (for example, the Chief Information Officer or CIO, program managers, enterprise architects, business analysts, and portfolio managers) can use the AWS CAF to focus on the skills and processes that are needed to align IT strategy and goals with business strategy and goals. This focus helps the organization maximize the business value of its IT investment and minimize the business risks.

# Platform perspective



Platform perspective capabilities

We must **understand and communicate the nature of IT systems and their relationships**. We must be able to **describe the architecture of the target state environment** in detail.



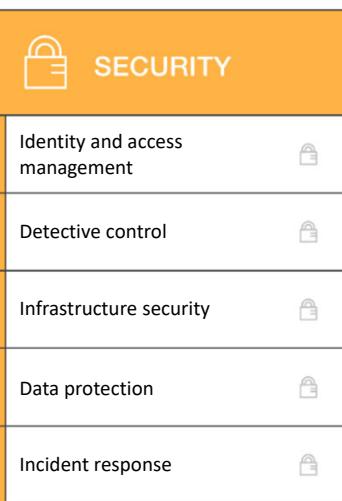
CTO, IT managers, and solutions architects

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

37

Stakeholders from the Platform perspective (for example, Chief Technology Officer or CTO, IT managers, and solutions architects) use a variety of architectural dimensions and models to understand and communicate the nature of IT systems and their relationships. They must be able to describe the architecture of the target state environment in detail. The AWS CAF includes principles and patterns for implementing new solutions on the cloud, and for migrating on-premises workloads to the cloud.

# Security perspective



We must ensure that the organization **meets its security objectives.**



CISO, IT security managers,  
and IT security analysts

## Security perspective capabilities

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

38

Stakeholders from the Security perspective (for example, Chief Information Security Officer or CISO, IT security managers, and IT security analysts) must ensure that the organization meets security objectives for visibility, auditability, control, and agility. Security perspective stakeholders can use the AWS CAF to structure the selection and implementation of security controls that meet the organization's needs.

# Operations perspective



OPERATIONS	
Service monitoring	⚙️
Application performance monitoring	⚙️
Resource inventory management	⚙️
Release management/change management	⚙️
Reporting and analytics	⚙️
Business continuity/Disaster recovery	⚙️
IT service catalog	⚙️

Operations perspective capabilities

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

We align with and support the operations of the business, and **define how day-to-day, quarter-to-quarter, and year-to-year business will be conducted.**



IT operations managers and  
IT support managers

39

Stakeholders from the Operations perspective (for example, IT operations managers and IT support managers) define how day-to-day, quarter-to-quarter, and year-to-year business is conducted. Stakeholders from the Operations perspective align with and support the operations of the business. The AWS CAF helps these stakeholders define current operating procedures. It also helps them identify the process changes and training that are needed to implement successful cloud adoption.

## Section 4 key takeaways



40

- Cloud adoption is not instantaneous for most organizations and requires a thoughtful, deliberate strategy and alignment across the whole organization.
- The AWS CAF was created to help organizations develop efficient and effective plans for their cloud adoption journey.
- The AWS CAF organizes guidance into six areas of focus, called perspectives.
- Perspectives consist of sets of business or technology capabilities that are the responsibility of key stakeholders.

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

The key takeaways from this section of the module include:

- Cloud adoption is not instantaneous for most organizations and requires a thoughtful, deliberate strategy and alignment across the whole organization.
- The AWS CAF was created to help organizations develop efficient and effective plans for their cloud adoption journey.
- The AWS CAF organizes guidance into six areas of focus, called perspectives.
- Perspectives consist of sets of business or technology capabilities that are the responsibility of key stakeholders.

Module 1: Cloud Concepts Overview

## Module wrap-up

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



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:

- Define different types of cloud computing models
- Describe six advantages of cloud computing
- Recognize the main AWS service categories and core services
- Review the AWS Cloud Adoption Framework

In summary, in this module you learned how to:

- Define different types of cloud computing
- Describe six advantages of cloud computing
- Recognize the main AWS service categories and core services
- Reviewed the AWS Cloud Adoption Framework

To finish this module, complete the knowledge check.

## Complete the knowledge check



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

43

Now, complete the knowledge check.

## Sample exam question



Why is AWS more economical than traditional data centers for applications with varying compute workloads?

- A. Amazon Elastic Compute Cloud (Amazon EC2) costs are billed on a monthly basis.
- B. Customers retain full administrative access to their Amazon EC2 instances.
- C. Amazon EC2 instances can be launched on-demand when needed.
- D. Customers can permanently run enough instances to handle peak workloads.

Look at the answer choices and rule them out based on the keywords that were previously highlighted.

## Additional resources



- [What is AWS? YouTube video](#)
- [Cloud computing with AWS website](#)
- [Overview of Amazon Web Services whitepaper](#)
- [An Overview of the AWS Cloud Adoption Framework whitepaper](#)
- [6 Strategies for Migrating Applications to the Cloud AWS Cloud Enterprise Strategy blog post](#)

If you want to learn more about the topics covered in this module, you might find the following additional resources helpful:

- [What is AWS? YouTube video](#)
- [Cloud computing with AWS website](#)
- [Overview of Amazon Web Services whitepaper](#)
- [An Overview of the AWS Cloud Adoption Framework whitepaper](#)

# Thank You

© 2019 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections or feedback on the course, please email us at: [aws-course-feedback@amazon.com](mailto:aws-course-feedback@amazon.com). For all other questions, contact us at: <https://aws.amazon.com/contact-us/aws-training/>. All trademarks are the property of their owners.



Thanks for participating!

AWS Academy Cloud Foundations

## Module 2: Cloud Economics and Billing

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



Welcome Module 2: Cloud Economics and Billing

# Module overview



## Topics

- Fundamentals of pricing
- Total Cost of Ownership
- AWS Organizations
- AWS Billing and Cost Management
- Technical Support

## Activities

- AWS Pricing Calculator
- Support plans scavenger hunt

## Demo

- Overview of the Billing Dashboard



## Knowledge check

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

2

This module will address the following topics:

- Fundamentals of Pricing
- Total Cost of Ownership
- AWS Organizations
- AWS Billing and Cost Management
- Technical Support

The module also includes an instructor-led demonstration that will show you how to interact with the billing dashboard.

The module also includes an activity that challenges you to estimate the costs for a company by using the AWS Pricing Calculator.

Finally, you will be asked to complete a knowledge check that will be used to test your understanding of the key concepts that are covered in this module.

## Module objectives



After completing this module, you should be able to:

- Explain the AWS pricing philosophy
- Recognize fundamental pricing characteristics
- Indicate the elements of total cost of ownership
- Discuss the results of the AWS Pricing Calculator
- Identify how to set up an organizational structure that simplifies billing and account visibility to review cost data.
- Identify the functionality in the AWS Billing Dashboard
- Describe how to use AWS Bills, AWS Cost Explorer, AWS Budgets, and AWS Cost and Usage Reports
- Identify the various AWS technical support plans and features

After completing this module, you should be able to:

- Explain the AWS pricing philosophy
- Recognize fundamental pricing characteristics
- Indicate the elements of total cost of ownership
- Discuss the results of the AWS Pricing Calculator
- Identify how to set up an organizational structure that simplifies billing and account visibility to review cost data.
- Identify the functionality in the AWS Billing Dashboard
- Describe how to use AWS Bills, AWS Cost Explorer, AWS Budgets, and AWS Cost and Usage Reports
- Identify the various AWS technical support plans and features

Module 2: Cloud Economics and Billing

## Section 1: Fundamentals of pricing

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



Introducing Section 1: Fundamentals of pricing.

## Three fundamental drivers of cost with AWS

### Compute

- Charged per hour/second\*
- Varies by instance type

\*Linux only

### Storage

- Charged typically per GB

### Data transfer

- Outbound is aggregated and charged
- Inbound has no charge (with some exceptions)
- Charged typically per GB

There are three fundamental drivers of cost with AWS: **compute**, **storage**, and **outbound data transfer**. These characteristics vary somewhat, depending on the AWS product and pricing model you choose.

In most cases, there is no charge for inbound data transfer or for data transfer between other AWS services within the same AWS Region. There are some exceptions, so be sure to verify data transfer rates before you begin to use the AWS service.

Outbound data transfer is aggregated across services and then charged at the outbound data transfer rate. This charge appears on the monthly statement as *AWS Data Transfer Out*.

# How do you pay for AWS?



**Pay for what you use**



**Pay less when you reserve**



**Pay less when you use more and as AWS grows**



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

6

This philosophy is what underlies AWS pricing. While the number and types of services offered by AWS have increased dramatically, our philosophy on pricing has not changed. At the end of each month, you pay for what you use. You can start or stop using a product at any time. No long-term contracts are required.

AWS offers a range of cloud computing services. For each service, you pay for exactly the amount of resources that you actually need. This utility-style pricing model includes:

- Pay for what you use
- Pay less when you reserve
- Pay less when you use more
- Pay even less as AWS grows

You will now take a closer look at these core concepts of pricing.

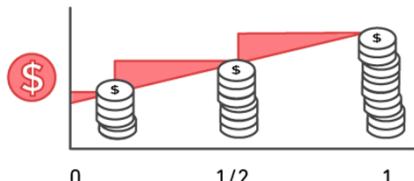
To learn more about AWS pricing, see: [AWS pricing overview](#)

## Pay for what you use

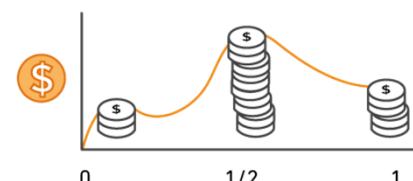


Pay only for the services that you consume, with no large upfront expenses.

On premises



AWS



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

7

Unless you build data centers for a living, you might have spent too much time and money building them. With AWS, you pay only for the services that you consume with no large upfront expenses. You can lower variable costs, so you no longer need to dedicate valuable resources to building costly infrastructure, including purchasing servers, software licenses, or leasing facilities.

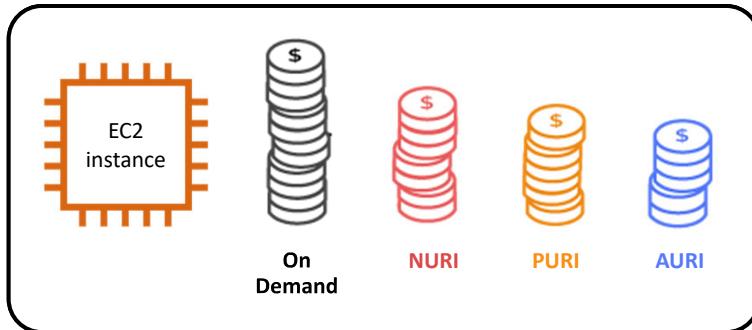
Quickly adapt to changing business needs and redirect your focus on innovation and invention by paying only for what you use and for as long as you need it. All AWS services are available on demand, require no long-term contracts, and have no complex licensing dependencies.

## Pay less when you reserve



### Invest in Reserved Instances (RIs):

- Save up to 75 percent
- Options:
  - All Upfront Reserved Instance (**AURI**) → largest discount
  - Partial Upfront Reserved Instance (**PURI**) → lower discounts
  - No Upfront Payments Reserved Instance (**NURI**) → smaller discount



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

8

For certain services like Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Relational Database Service (Amazon RDS), you can invest in reserved capacity. With Reserved Instances, you can save up to 75 percent over equivalent on-demand capacity. Reserved Instances are available in three options:

- All Upfront Reserved Instance (or AURI)
- Partial Upfront Reserved Instance (or PURI)
- No Upfront Payments Reserved Instance (or NURI)

When you buy Reserved Instances, you receive a greater discount when you make a larger upfront payment. To maximize your savings, you can pay all upfront and receive the largest discount. Partial Upfront RIs offer lower discounts, but they give you the option to spend less upfront. Lastly, you can choose to spend nothing upfront and receive a smaller discount, which enables you to free capital to spend on other projects.

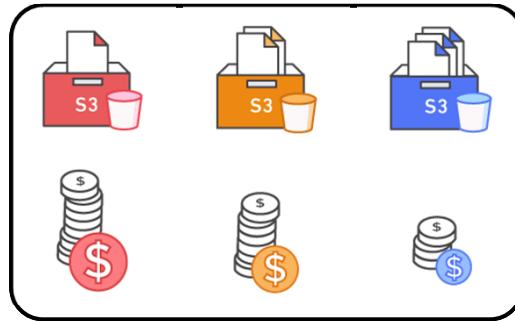
By using reserved capacity, your organization can minimize risks, more predictably manage budgets, and comply with policies that require longer-term commitments.

## Pay less by using more



Realize volume-based discounts:

- **Savings** as usage increases.
- **Tiered pricing** for services like Amazon Simple Storage Service (Amazon S3), Amazon Elastic Block Store (Amazon EBS), or Amazon Elastic File System (Amazon EFS) → the more you use, the less you pay per GB.
- Multiple storage services deliver **lower** storage costs based on needs.



With AWS, you can get volume-based discounts and realize important savings as your usage increases. For services like Amazon Simple Storage Service (Amazon S3), pricing is tiered, which means that you pay less per GB when you use more. In addition, data transfer *in* is always free. Multiple storage services deliver lower storage costs based on your needs. As a result, as your AWS usage needs increase, you benefit from the economies of scale that enable you to increase adoption and keep costs under control.

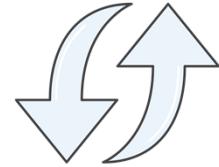
As your organization evolves, AWS also gives you options to acquire services that help you address your business needs. For example, the AWS storage services portfolio offers options to help you lower pricing based on how frequently you access data and the performance that you need to retrieve it. To optimize your savings, you can choose the right combination of storage solutions that help you reduce costs while preserving performance, security, and durability.

## Pay even less as AWS grows



As AWS grows:

- AWS focuses on lowering cost of doing business.
- This practice results in AWS passing savings from economies of scale to you.
- Since 2006, AWS has **lowered pricing 75 times** (as of September 2019).
- Future higher-performing resources replace current resources for no extra charge.



AWS constantly focuses on reducing data center hardware costs, improving operational efficiencies, lowering power consumption, and generally lowering the cost of doing business.

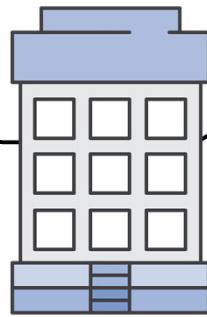
These optimizations and the substantial and growing economies of scale of AWS result in passing savings back to you as lower pricing. Since 2006, AWS has lowered pricing 75 times (as of September 2019).

Another benefit of AWS growth is that future, higher-performing resources replace current ones for no extra charge.

## Custom pricing



- Meet varying needs through custom pricing.
- Available for high-volume projects with unique requirements.



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

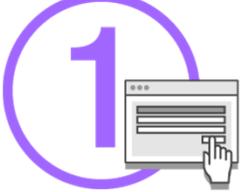
11

AWS realizes that every customer has different needs. If none of the AWS pricing models work for your project, custom pricing is available for high-volume projects with unique requirements.

# AWS Free Tier



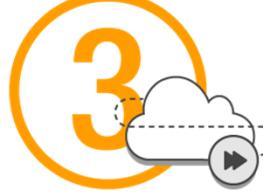
Enables you to gain free hands-on experience with the AWS platform, products, and services. Free for 1 year for new customers.



Sign up for an AWS account



Learn with 10-minute tutorials



Start building with AWS

To help new AWS customers get started in the cloud, AWS offers a free usage tier (the AWS Free Tier) for new customers for up to 1 year. The AWS Free Tier applies to certain services and options. If you are a new AWS customer, you can run a free Amazon Elastic Compute Cloud (Amazon EC2) T2 micro instance for a year, while also using a free usage tier for Amazon S3, Amazon Elastic Block Store (Amazon EBS), Elastic Load Balancing, AWS data transfer, and other AWS services.

To learn more, see [AWS Free Tier](#).

## Services with no charge



Amazon VPC



Elastic Beanstalk\*\*



Auto Scaling\*\*



AWS CloudFormation\*\*



AWS Identity and Access Management (IAM)

**\*\*Note:** There might be charges associated with other AWS services that are used with these services.

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

13

AWS also offers a variety of services for no additional charge.

- **Amazon Virtual Private Cloud (Amazon VPC)** enables you to provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define.
- **AWS Identity and Access Management (IAM)** controls your users' access to AWS services and resources.
- **Consolidated Billing** is a billing feature in AWS Organizations to consolidate payment for multiple AWS accounts or multiple Amazon Internet Services Private Limited (AISPL) accounts\*. Consolidated billing provides:
  - One bill for multiple accounts.
  - The ability to easily track each account's charges.
  - The opportunity to decrease charges as a result of volume pricing discounts from **combined usage**.
  - And you can consolidate all of your accounts using Consolidated Billing and get tiered benefits.
- **AWS Elastic Beanstalk** is an even easier way for you to quickly deploy and manage applications in the AWS Cloud.
- **AWS CloudFormation** gives developers and systems administrators an easy way to create a collection of related AWS resources and provision them in an orderly and

predictable fashion.

- **Automatic Scaling** automatically adds or removes resources according to conditions you define. The resources you are using increase seamlessly during demand spikes to maintain performance and decrease automatically during demand lulls to minimize costs.
- **AWS OpsWorks** is an application management service that makes it easy to deploy and operate applications of all shapes and sizes.

Though there is no charge for these services, there might be charges associated with other AWS services used with these services. For example, when you automatically scale additional EC2 instances, there will be charges for those instances.

\* Note: The main difference between AWS accounts and AISPL accounts is the [seller of record](#). AWS accounts are administered by Amazon Web Services, Inc., but AISPL accounts are administered by Amazon Internet Services Private Limited. If you used an Indian address when you created your account, your account's default seller of record is AISPL. By default, AISPL accounts are billed in Indian Rupees (INR).

## Key takeaways



14

- There is no charge (with some exceptions) for:
  - Inbound data transfer.
  - Data transfer between services within the same AWS Region.
- Pay for what you use.
- Start and stop anytime.
- No long-term contracts are required.
- Some services are free, but the other AWS services that they provision might not be free.

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

In summary, while the number and types of services offered by AWS have increased dramatically, our philosophy on pricing has not changed. At the end of each month, you pay only for what you use, and you can start or stop using a product at any time. No long-term contracts are required.

The best way to estimate costs is to examine the fundamental characteristics for each AWS service, estimate your usage for each characteristic, and then map that usage to the prices that are posted on the AWS website. The service pricing strategy gives you the flexibility to choose the services that you need for each project and to pay only for what you use.

There are several free AWS services, including:

- Amazon VPC
- Elastic Beanstalk
- AWS CloudFormation
- IAM
- Automatic scaling services
- AWS OpsWorks
- Consolidated Billing

While the services themselves are free, the resources that they provision might not be free. In most cases, there is no charge for inbound data transfer or for data transfer between other AWS services within the same AWS Region. There are some exceptions, so be sure to verify data transfer rates before you begin to use the AWS service.

Outbound data transfer costs are tiered.

To learn more about pricing, see:

[AWS pricing](#)

[AWS pricing overview](#)

Module 2: Cloud Economics and Billing

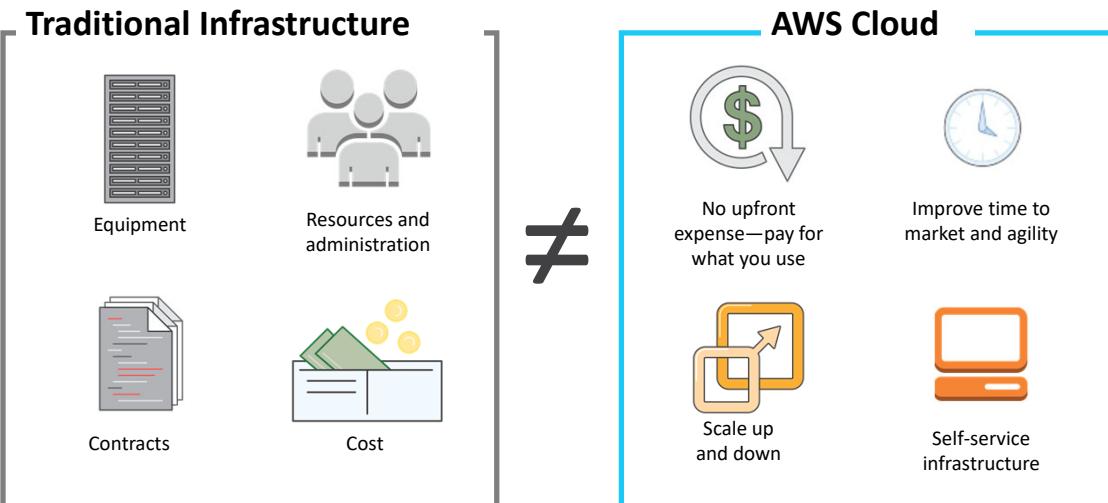
## Section 2: Total Cost of Ownership

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



Introducing Section 2: Total Cost of Ownership.

## On-premises versus cloud



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

16

On-premises versus cloud is a question that many businesses ask. The difference between these two options is how they are deployed.

An on-premises infrastructure is installed locally on a company's own computers and servers. There are several fixed costs, also known as *capital expenses*, that are associated with the traditional infrastructure. Capital expenses include facilities, hardware, licenses, and maintenance staff. Scaling up can be expensive and time-consuming. Scaling down does not reduce fixed costs.

A cloud infrastructure is purchased from a service provider who builds and maintains the facilities, hardware, and maintenance staff. A customer pays for what is used. Scaling up or down is simple. Costs are easy to estimate because they depend on service use.

It is difficult to compare an on-premises IT delivery model with the AWS Cloud. The two are different because they use different concepts and terms.

Using on-premises IT involves a discussion that is based on capital expenditure, long planning cycles, and multiple components to buy, build, manage, and refresh resources over time.

Using the AWS Cloud involves a discussion about flexibility, agility, and consumption-based costs.

So, how can you identify the best option?

# What is Total Cost of Ownership (TCO)?



**Total Cost of Ownership (TCO)** is the financial estimate to help identify direct and indirect costs of a system.

## Why use TCO?

- To compare the costs of running an **entire infrastructure environment or specific workload** on-premises versus on AWS
- To budget and **build the business case** for moving to the cloud



You can identify the best option by comparing the on-premises solution to a cloud solution. Total Cost of Ownership (or TCO) is a financial estimate that is intended to help buyers and owners determine the direct and indirect costs of a product or system. TCO includes the cost of a service, plus all the costs that are associated with owning the service.

You might want to compare the costs of running an entire infrastructure environment for a specific workload in an on-premises or collocation facility to the same workload running on a cloud-based infrastructure. This comparison is done for budgeting purposes or to build a business case for business decisions about the optimal deployment solution.

# TCO considerations



<b>1</b> Server Costs	Hardware: Server, rack chassis power distribution units (PDUs), top-of-rack (TOR) switches (and maintenance)	Software: Operating system (OS), virtualization licenses (and maintenance)	Facilities cost		
			Space	Power	Cooling
<b>2</b> Storage Costs	Hardware: Storage disks, storage area network (SAN) or Fibre Channel (FC) switches	Storage administration costs	Facilities cost		
			Space	Power	Cooling
<b>3</b> Network Costs	Network hardware: Local area network (LAN) switches, load balancer bandwidth costs	Network administration costs	Facilities cost		
			Space	Power	Cooling
<b>4</b> IT Labor Costs	Server administration costs				

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

18

Some of the costs that are associated with data center management include:

- **Server** costs for both hardware and software, and facilities costs to house the equipment.
- **Storage** costs for the hardware, administration, and facilities.
- **Network** costs for hardware, administration, and facilities.
- And **IT labor** costs that are required to administer the entire solution.

When you compare an on-premises to cloud solution, it is important to accurately assess the true costs of both options. With the cloud, most costs are upfront and readily calculated. For example, cloud providers give transparent pricing based on different usage metrics, such as RAM, storage, and bandwidth, among others. Pricing is frequently fixed per unit of time.

Customers gain certainty over pricing and are then able to readily calculate costs based on several different usage estimates.

Compare this process to on-premises technology. Though they are sometimes difficult to determine, calculations of in-house costs must take into account all:

- **Direct costs** that accompany running a server—like power, floor space, storage, and IT operations to manage those resources.
- **Indirect costs** of running a server, like network and storage infrastructure.

This diagram is conceptual, and it does not include every cost item. For example, depending on the solution you are implementing, software costs can include database, management, and middle-tier costs. Facilities costs can include upgrades, maintenance, building security, taxes, and so on. IT labor costs can include security administration and application administration costs. This diagram includes an abbreviated list to demonstrate the type of costs that are involved in data center maintenance.

## On-premises versus all-in-cloud

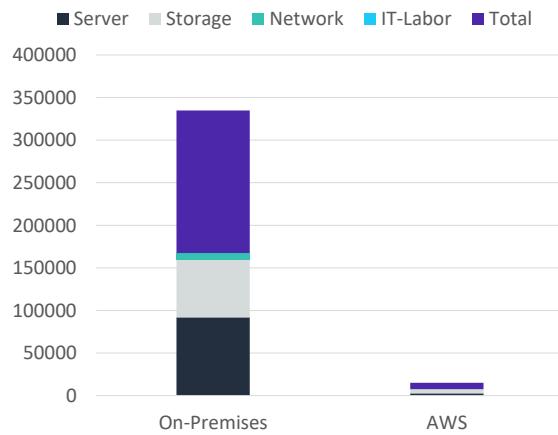


You could save up to **96 percent** a year by moving your infrastructure to AWS.

Your 3-year total savings would be **\$159,913**.

3-Year Total Cost of Ownership		
	On-Premises	AWS
Server	\$91,922	\$2,547
Storage	\$67,840	\$4,963
Network	\$7,660	\$-----
IT – Labor	\$ -----	\$-----
--		
Total	\$167,422	\$7,509

AWS cost includes business-level support and  
a 3-year PURI EC2 instance



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

19

Here is a sample cost comparison. This example shows a cost comparison for an on-premises solution and a cloud solution over 3 years. For this comparison, two similar environments were constructed to represent the on-premises and AWS environments. Additional direct and indirect costs that are associated with the on-premises solution were not included. The components of the on-premises solution include:

- 1 virtual machine with 4 CPUs, 16 GB of RAM, and a Linux operating system
- Average utilization is 100 percent
- Optimized by RAM

The components of a comparable AWS environment include:

- 1 m4.xlarge instance with 4 CPUs, 16 GB of RAM
- The instance type is a 3-year Partial Upfront Reserved Instance

The on-premises 3-year total cost is \$167,422. The AWS Cloud 3-year total cost is \$7,509, which is a 96 percent savings over the on-premises solution. Thus, the 3-year total savings on cloud infrastructure would be \$159,913. This comparison helps a business clearly understand the differences between the alternatives.

### What is the difference in the costs?

Remember, the on-premises solution is predicted. It continues to incur costs whether the capacity is used.

In contrast, the AWS solution is commissioned when needed and decommissioned when

the resources are no longer in use, which results in lower overall costs.

# AWS Pricing Calculator



Use the AWS Pricing Calculator to:

- Estimate monthly costs
- Identify opportunities to reduce monthly costs
- Model your solutions before building them
- Explore price points and calculations behind your estimate
- Find the available instance types and contract terms that meet your needs
- Name your estimate and create and name **groups** of services

The screenshot shows the AWS Pricing Calculator interface. At the top, there's a header with the AWS Academy logo and a 'Create an estimate' button. Below that is a 'Getting started' section with links for 'What is the AWS Pricing Calculator?' and 'Generating estimates'. On the right, there's a 'More resources' sidebar with links for 'User guide', 'FAQs', and 'Pricing assumptions and variations'. The main area is titled 'How it works' and shows a four-step process: 1. AWS Pricing Calculator (Estimate the cost for your architecture solution), 2. Add services (Select the AWS services that you need), 3. Configure service (Configure your usage to see service costs), and 4. View estimate totals (View your estimated costs per service, service group, and total).

Access the [AWS Pricing Calculator](#)

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

20

AWS offers the **AWS Pricing Calculator** to help you estimate a monthly AWS bill. You can use this tool to explore AWS services and create an estimate for the cost of your use cases on AWS. You can model your solutions before building them, explore the price points and calculations behind your estimate, and find the available instance types and contract terms that meet your needs. This enables you to make informed decisions about using AWS. You can plan your AWS costs and usage or price out setting up a new set of instances and services.

The **AWS Pricing Calculator** helps you:

- Estimate monthly costs of AWS services
- Identify opportunities for cost reduction
- Model your solutions before building them
- Explore price points and calculations behind your estimate
- Find the available instance types and contract terms that meet your needs

The AWS Pricing Calculator enables you to name your estimate and create and name groups of services. *Groups* are containers that you add services to in order to organize and build your estimate. You can organize your groups and services by cost-center, department, product architecture, etc.

For more information, see the [AWS Pricing Calculator website](#).

# Reading an estimate



Your estimate is broken into: first 12 months total, total upfront, and total monthly.

The screenshot shows the AWS Pricing Calculator interface with the following details:

Category	Description	Value
First 12 months total	886.92 USD	
Total upfront	0.00 USD	
Total monthly	73.91 USD	
Services (2)		
Amazon Simple Storage Service (S3)	Region: US East (Ohio)	
S3 Standard storage (100 GB per month)	Monthly:	2.37 USD
Amazon EC2	Region: US East (Ohio)	
Quick estimate	Operating system (Linux), Quantity (1), Pricing strategy (EC2 Instance Savings Plans 1 Year No Upfront), Storage for each EC2 instance (General Purpose SSD (gp2)), Storage amount (100 GB), Instance type (t4g.xlarge)	Monthly: 71.54 USD

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

21

AWS Pricing Calculator estimates are broken into:

- The total for your first 12 months – The total estimate for your current group and all of the services and groups in your current group. It combines the upfront and monthly estimates.
- Your total upfront – How much you are estimated to pay upfront as you set up your AWS stack.
- Your total monthly – How much you're estimated to spend every month while you run your AWS stack.

Within a group, you can see how much each service is estimated to cost. If you want to price out different ways to build your AWS setup, you can use different groups for each variation of your setup and compare the estimates for the different setups.

For more information, see [Reading an estimate](#).

## Activity: AWS Pricing Calculator activity



- Break up into groups of four or five and use the [AWS Pricing Calculator](#) and specifications provided to develop a cost estimate.
- Be prepared to report your findings back to the class.

**Create an estimate**

Start your estimate with no commitment, and explore AWS services and pricing for your architecture needs.

**Create estimate**

[AWS Pricing calculator website](#)

**aws pricing calculator**

AWS Pricing Calculator > My Estimate > Add service

Step 1 Select service

Step 2 Configure service

Select service Info

AWS services (63)

Q

Amazon API Gateway

Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. APIs act as the front door for applications to access data, business logic, or functionality from your backend services.

Product page Configure

Amazon Athena

Amazon Athena is an interactive query and analysis service that makes it easy to analyze data in Amazon S3 so there is no need to provision or manage infrastructure to run queries that you run.

Product page

Amazon Aurora PostgreSQL-Compatible DB

Amazon Aurora is a MySQL and PostgreSQL-compatible relational database built for the cloud, that combines the performance and

Amazon Carrier IP

A Carrier IP address is the interface, which receives

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

22

Break up into groups of four or five and use the AWS Pricing Calculator and the specifications provided to develop a cost estimate.

Be prepared to report your findings back to the class.

# Additional benefit considerations



## Hard benefits

- Reduced spending on compute, storage, networking, security
- Reductions in hardware and software purchases (capex)
- Reductions in operational costs, backup, and disaster recovery
- Reduction in operations personnel



## Soft Benefits

- Reuse of service and applications that enable you to define (and redefine solutions) by using the same cloud service
- Increased developer productivity
- Improved customer satisfaction
- Agile business processes that can quickly respond to new and emerging opportunities
- Increase in global reach

Hard benefits include reduced spending on compute, storage, networking, and security. They also include reductions in hardware and software purchases; reductions in operational costs, backup, and disaster recovery; and a reduction in operations personnel.

**Cloud Total Cost of Ownership** defines what will be spent on the technology after adoption—or what it costs to run the solution. Typically, a TCO analysis looks at the as-is on-premises infrastructure and compares it with the cost of the to-be infrastructure state in the cloud. While this difference might be easy to calculate, it might only provide a narrow view of the total financial impact of moving to the cloud.

A **return on investment (ROI)** analysis can be used to determine the value that is generated while considering spending and saving. This analysis starts by identifying the hard benefits

in terms of direct and visible cost reductions and efficiency improvements.

Next, **soft savings** are identified. Soft savings are value points that are challenging to accurately quantify, but they can be more valuable than the hard savings. It is important for you to understand both hard and soft benefits to understand the full value of the cloud. Soft benefits include:

- Reusing service and applications that enable you to define (and redefine solutions) by using the same cloud service
- Increased developer productivity
- Improved customer satisfaction
- Agile business processes that can quickly respond to new and emerging opportunities
- Increased global reach

Now, you will review a case study from Delaware North to see an actual TCO example.

# Case study: Total Cost Of Ownership



## Background:

- Growing global company with over 200 locations
- 500 million customers, \$3 billion annual revenue

## Background:

Delaware North originated in 1915 as a peanut and popcorn concessions vendor; today, it's a major food and hospitality company. Although the company deliberately keeps a low profile, it is a leader in the food-service and hospitality industry.

Delaware North serves more than **500 million customers** annually at more than **200 locations** around the world, including venues the Kennedy Space Center in Florida, London Heathrow Airport, Kings Canyon Resort in Australia, and the Green Bay Packers' Lambeau Field in Wisconsin. This global presence has turned Delaware North into a **\$3 billion enterprise**.

# Case study: Total Cost of Ownership



**Background:**

- Growing global company with over 200 locations
- 500 million customers, \$3 billion annual revenue

**Challenge:**

- Meet demand to rapidly deploy new solutions
- Constantly upgrade aging equipment

The company's on-premises data center was becoming too expensive and inefficient to support its global business operations.

Kevin Quinlivan, Delaware North's Chief Information Officer, said, "As the company continued to grow, the **demand to rapidly deploy new solutions** to meet customer requirements increased as well. This fact, combined with the **need to constantly upgrade aging equipment**, required an even greater commitment of resources on our part. We had to find a better strategy."

Delaware North turned to AWS for a solution.

# Case study: Total Cost of Ownership



**Background:**

- Growing global company with over 200 locations
- 500 million customers, \$3 billion annual revenue

**Challenge:**

- Meet demand to rapidly deploy new solutions
- Constantly upgrade aging equipment

**Criteria:**

- Broad solution to handle all workloads
- Ability to modify processes to improve efficiency and lower costs
- Eliminate busy work (such as patching software)
- Achieve a positive return on investment (ROI)

After a successful migration of about 50 websites to AWS in 2013, Delaware North evaluated the cost benefit and Total Cost of Ownership to move their IT infrastructure to AWS. Their focus was to answer executive-level business demands for measurable benefits that could convince an executive committee that the AWS Cloud was the right approach.

The evaluation process centered on three criteria:

- First, a cloud solution needed a broad set of technologies that could **handle all of Delaware North's enterprise workloads** while delivering support for critical functions.
- From an operational perspective, Delaware North wanted the features and flexibility to **modify core IT processes to improve efficiencies and lower costs**. This included **eliminating redundant or time-consuming tasks** like patching software or pushing test and development tasks through outdated systems that, in the past, added months to the deployment of new services.
- Finally, financial requirements needed to **demonstrate a return on investment** with a solid cost-benefit justification for moving away from their existing data center environment.

# Case study: Total Cost of Ownership



- Background:**
- Is a growing global company with over 200 locations
  - Have 500 million customers, \$3 billion (USD) annual revenue

- Challenge:**
- Meet demand to rapidly deploy new solutions
  - Constantly upgrade aging equipment

- Criteria:**
- Have a broad solution to handle all workloads
  - Be able to modify processes to improve efficiency and lower costs
  - Eliminate busy work (such as patching software)
  - Achieve a positive return on investment (ROI)

- Solution:**
- Moved their on-premises data center to AWS
    - Eliminated 205 servers (90 percent)
    - Moved nearly all applications to AWS
  - Used 3-year Amazon EC2 Reserved Instances

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

27

A cost comparison completed by Delaware North demonstrated that it could save \$3.5 million US dollars based on a 5-year run rate by **moving its on-premises data center to AWS** and using 3-year Amazon EC2 Reserved Instances and Reserved Instance renewals.

Quinlivan noted that the deep technology stack available on AWS was more than sufficient to meet the company's technical and operational requirements. The pricing structure of the AWS offerings—which includes paying only for what is used—provided total cost of ownership benefits that were presented to senior leaders.

Quinlivan stated, “We compared the costs of keeping our on-premises data center versus moving to the AWS Cloud, measuring basic infrastructure items such as hardware cost and maintenance.” He also says “We estimate that moving to AWS will save us at least \$3.5 million over five years by **reducing our server hardware by more than 90 percent**. But the cost savings will likely be greater due to additional benefits, like the increased compute capacity we can get using AWS. That lets us continually add more and larger workloads than we could using a traditional data center infrastructure, and achieve savings by only paying for what we use.”

Delaware North moved almost all of its applications to AWS, including enterprise software such as its Fiorano middleware, Crystal Reports and QLIK business intelligence solutions, its Citrix virtual desktop system, and Microsoft System Center Configuration Manager, which is used to manage workstations.

The most dramatic physical change was the **elimination of 205 servers**. Everything that ran on that hardware was migrated to AWS. The IT department decided to keep about 20 servers on-premises at the new headquarters building to run communications and file-and-print tasks.

“We erred on the side of caution to ensure there is no latency with these tasks, but once we reach a certain comfort level, we may move these to the cloud as well,” said Scott Mercer, head of the IT department’s service-oriented architecture team.

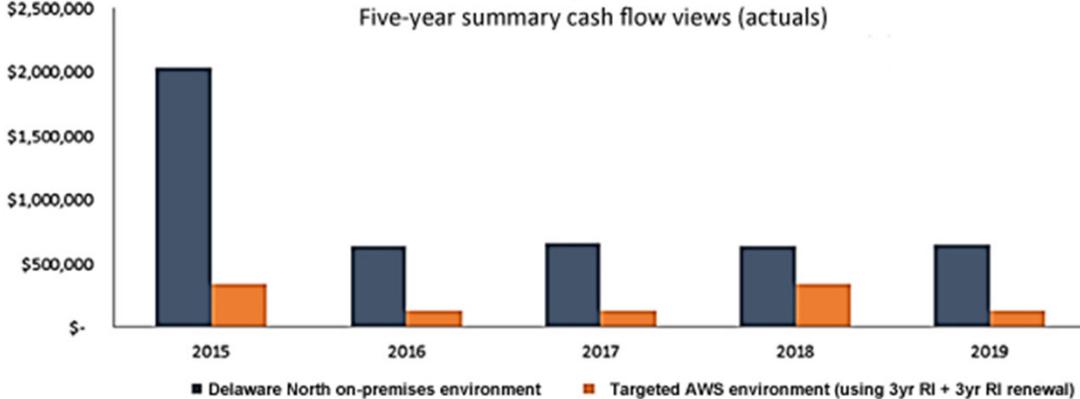
## Case study: Total Cost of Ownership



Delaware  
North

### Cost comparison: On-premises data center vs. AWS

Five-year summary cash flow views (actuals)



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

28

This chart displays the cost comparison done by Delaware North showing the costs of their on-premises environment and the proposed AWS environment. The estimates showed a \$3.5 million savings based on a five-year run rate by moving from an on-premises data center to AWS.

## Case study: Total Cost of Ownership



### Results:

#### Business Goals:

Growth  
Enhanced 24/7 business  
Operational efficiency

#### Resource optimization

- Robust security compliance
- Enhanced disaster recovery
- Increased computing capacity

#### Speed to market

- One day to provision new businesses
- Just minutes to push out a service

#### Operational efficiency

- Continuous cost optimization and reduction

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

29

About 6 months into its cloud migration, Delaware North realized benefits in addition to its data center consolidation, including cost-effective security compliance, enhanced disaster recovery, and faster deployment times for new services.

"Robust security in a retail environment is critical for us because of our many retail operations, and AWS is enormously helpful for that," said Brian Mercer, the senior software architect for the project. "By leveraging the security best practices of AWS, we've been able to eliminate a lot of compliance tasks that in the past took up valuable time and money."

Brian Mercer added that the company also increased its disaster recovery capabilities at a lower cost than what was available in its previous data center deployment. "It significantly improved our business continuity capabilities, including seamless failovers," he said.

The solution is also helping Delaware North operate with greater speed and agility. For example, it can bring in new businesses—either through contracts or acquisitions—and get them online more quickly than in the past by eliminating the need for traditional IT procurement and provisioning. It used to take between 2 and 3 weeks to provision new business units; now it takes 1 day. The Delaware North IT team is also using AWS to overhaul its operations by eliminating outdated and cumbersome processes, cleaning up documentation, and using the benefits of running test and development tasks in

combination with rapid deployment of services through the cloud.

“Our DevOps team can now spin up the resources to push out a service in just minutes, compared to the weeks it used to take,” said Brian Mercer. “With AWS, we can respond much faster to business needs. And we can start repurposing time and resources to deliver more value and services to our internal teams and to our customers.”

Module 2: Cloud Economics and Billing

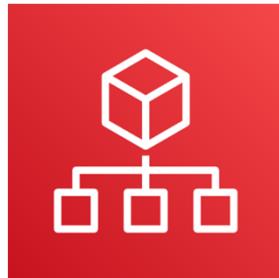
## Section 3: AWS Organizations

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



Introducing Section 3: AWS Organizations.

# Introduction to AWS Organizations



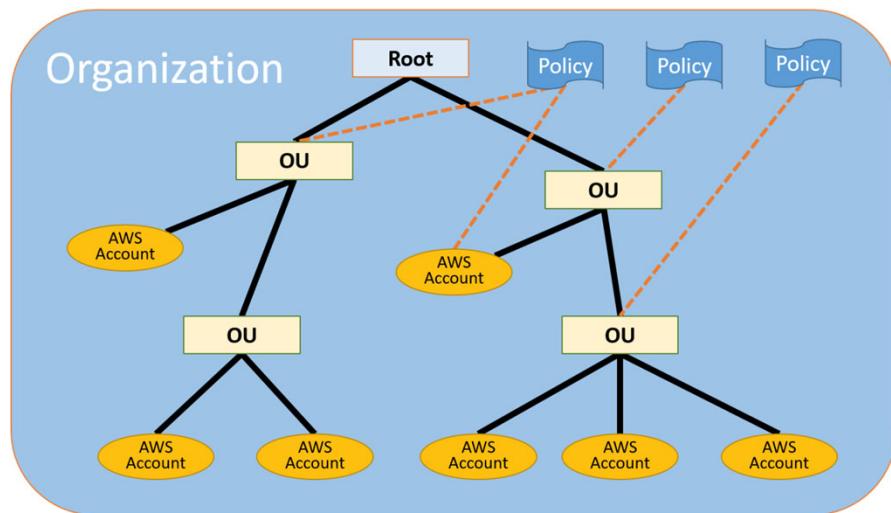
AWS Organizations

AWS Organizations is a free account management service that enables you to consolidate multiple AWS accounts into an **organization** that you create and centrally manage. AWS Organizations include consolidated billing and account management capabilities that help you to better meet the budgetary, security, and compliance needs of your business.

The main benefits of AWS Organizations are:

- Centrally managed access policies across multiple AWS accounts.
- Controlled access to AWS services.
- Automated AWS account creation and management.
- Consolidated billing across multiple AWS accounts.

# AWS Organizations terminology



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

\*Organizational Units (OUs)

32

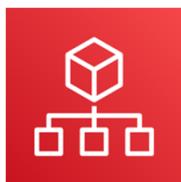
Here is some terminology to understand the structure of AWS Organizations.

The diagram shows a basic *organization*, or *root*, that consists of seven accounts that are organized into four organizational units (or OUs). An OU is a container for accounts within a root. An OU can also contain other OUs. This structure enables you to create a hierarchy that looks like an upside-down tree with the root at the top. The branches consist of child OUs and they move downward until they end in accounts, which are like the leaves of the tree.

When you attach a policy to one of the nodes in the hierarchy, it flows down and it affects all the branches and leaves. This example organization has several policies that are attached to some of the OUs or are attached directly to accounts.

An OU can have only one parent and, currently, each account can be a member of exactly one OU. An account is a standard AWS account that contains your AWS resources. You can attach a policy to an account to apply controls to only that one account.

## Key features and benefits



AWS  
Organizations



Policy-based account management



Group based account management



Application programming interfaces (APIs)  
that automate account management



Consolidated billing

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

33

AWS Organizations enables you to:

- Create **service control policies (SCPs)** that centrally control AWS services across multiple AWS accounts.
- Create **groups of accounts** and then attach policies to a group to ensure that the correct policies are applied across the accounts.
- Simplify account management by using **application programming interfaces (APIs)** to automate the creation and management of new AWS accounts.
- Simplify the billing process by setting up a single payment method for all the AWS accounts in your organization. With **consolidated billing**, you can see a combined view of charges that are incurred by all your accounts, and you can take advantage of pricing benefits from aggregated usage. Consolidated billing provides a central location to manage billing across all of your AWS accounts, and the ability to benefit from volume discounts.

# Security with AWS Organizations



Control access with AWS Identity and Access Management (IAM).

IAM policies enable you to allow or deny access to AWS services for users, groups, and roles.

Service control policies (SCPs) enable you to allow or deny access to AWS services for individuals or group accounts in an organizational unit (OU).

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

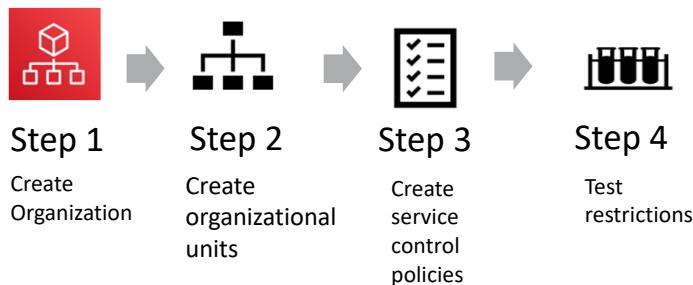
34

AWS Organizations does not replace associating **AWS Identity and Access Management (IAM)** policies with users, groups, and roles within an AWS account.

With IAM policies, you can allow or deny access to AWS services (such as Amazon S3), individual AWS resources (such as a specific S3 bucket), or individual API actions (such as s3:CreateBucket). An IAM policy can be applied only to IAM users, groups, or roles, and it can never restrict the AWS account root user.

In contrast, with Organizations, you use **service control policies (SCPs)** to allow or deny access to particular AWS services for individual AWS accounts or for groups of accounts in an OU. The specified actions from an attached SCP affect all IAM users, groups, and roles for an account, including the AWS account root user.

# Organizations setup



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

35

Keep in mind that this process assumes that you have access to two existing AWS accounts, and that you can sign in to each account as an administrator.

Review these steps for setting up AWS Organizations:

- Step 1 is to create your organization with your current AWS account as the primary account. You also invite one AWS account to join your organization and create another account as a member account.
- Step 2 is to create two organizational units in your new organization and place the member accounts in those OUs.
- Step 3 is to create service control policies, which enable you to apply restrictions to what actions can be delegated to users and roles in the member accounts. A service control policy is a type of organization control policy.
- Step 4 is to test your organization's policies. Sign in as a user for each of the roles (such as OU1 or OU2) and see how the service control policies impact account access. Alternatively, you can use the IAM policy simulator to test and troubleshoot IAM and resource-based policies that are attached to IAM users, groups, or roles in your AWS account.

To learn more about the IAM policy simulator, see:

[IAM policy simulator](#)

# Limits of AWS Organizations



Limits	
Limits on Names	Names must be composed of Unicode characters. Names must not exceed 250 characters in length.
Maximum and Minimum Values	Number of AWS accounts
	Number of roots
	Number of OUs
	Number of policies
	Maximum size of a service control policy document
	Maximum nesting of OUs in a root
	Invitations sent per day
	Number of member accounts you can create concurrently
	Number of entities to which you can attach a policy

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

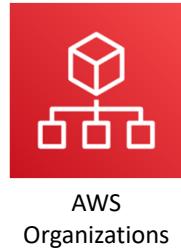
36

There are restrictions on names that you can create in AWS Organizations, which includes names of accounts, OUs, roots, and policies.

Names must be composed of Unicode characters and not exceed 250 characters in length.

AWS Organizations has several maximum and minimum values for entities.

# Accessing AWS Organizations



AWS  
Organizations



AWS Management Console



AWS Command Line  
Interface (AWS CLI) tools



Software development kits  
(SDKs)



HTTPS Query application  
programming interfaces (API)

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

37

AWS Organizations can be managed through different interfaces.

The **AWS Management Console** is a browser-based interface that you can use to manage your organization and your AWS resources. You can perform any task in your organization by using the console.

The **AWS Command Line Interface (AWS CLI) tools** enable you to issue commands at your system's command line to perform AWS Organizations tasks and AWS tasks. This method can be faster and more convenient than using the console.

You can use also **AWS software development kits (SDKs)** to handle tasks such as cryptographically signing requests, managing errors, and retrying requests automatically. AWS SDKs consist of libraries and sample code for various programming languages and platforms, such as Java, Python, Ruby, .NET, iOS, and Android.

The **AWS Organizations HTTPS Query API** gives you programmatic access to AWS Organizations and AWS. You can use the API to issue HTTPS requests directly to the service. When you use the HTTPS API, you must include code to digitally sign requests by using your credentials.

Module 2: Cloud Economics and Billing

## Section 4: AWS Billing and Cost Management

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



Introducing Section 4: AWS Billing and Cost Management.

# Introducing AWS Billing and Cost Management



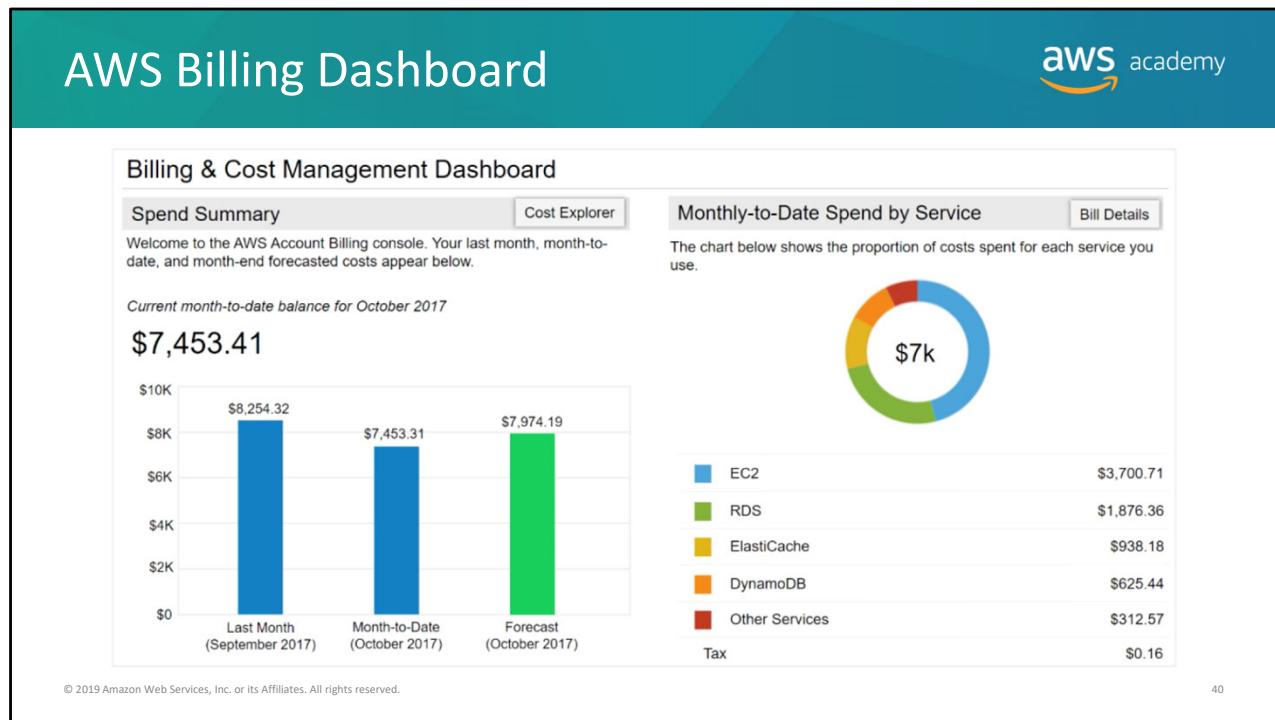
© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

39

**AWS Billing and Cost Management** is the service that you use to pay your AWS bill, monitor your usage, and budget your costs. Billing and Cost Management enables you to forecast and obtain a better idea of what your costs and usage might be in the future so that you can plan ahead.

You can set a custom time period and determine whether you would like to view your data at a monthly or daily level of granularity.

With the filtering and grouping functionality, you can further analyze your data using a variety of available dimensions. The **AWS Cost and Usage Report Tool** enables you to identify opportunities for optimization by understanding your cost and usage data trends and how you are using your AWS implementation.



The **AWS Billing Dashboard** lets you view the status of your month-to-date AWS expenditure, identify the services that account for the majority of your overall expenditure, and understand at a high level how costs are trending.

One of the graphs that is located on the dashboard is the **Spend Summary**. The Spend Summary shows you how much you spent last month, the estimated costs of your AWS usage for the month to date, and a forecast for how much you are likely to spend this month.

Another graph is **Month-to-Date Spend by Service**, which shows the top services that you use most and the proportion of costs that are attributed to that service.

## Tools



AWS Budgets



AWS Cost and Usage Report



AWS Cost Explorer

From the billing dashboard, you can access several other cost management tools that you can use to estimate and plan your AWS costs. These tools include AWS Bills, AWS Cost Explorer, AWS Budgets, and AWS Cost and Usage Reports.

## Monthly bills



BILLS | COST EXPLORER | BUDGETS | REPORTS

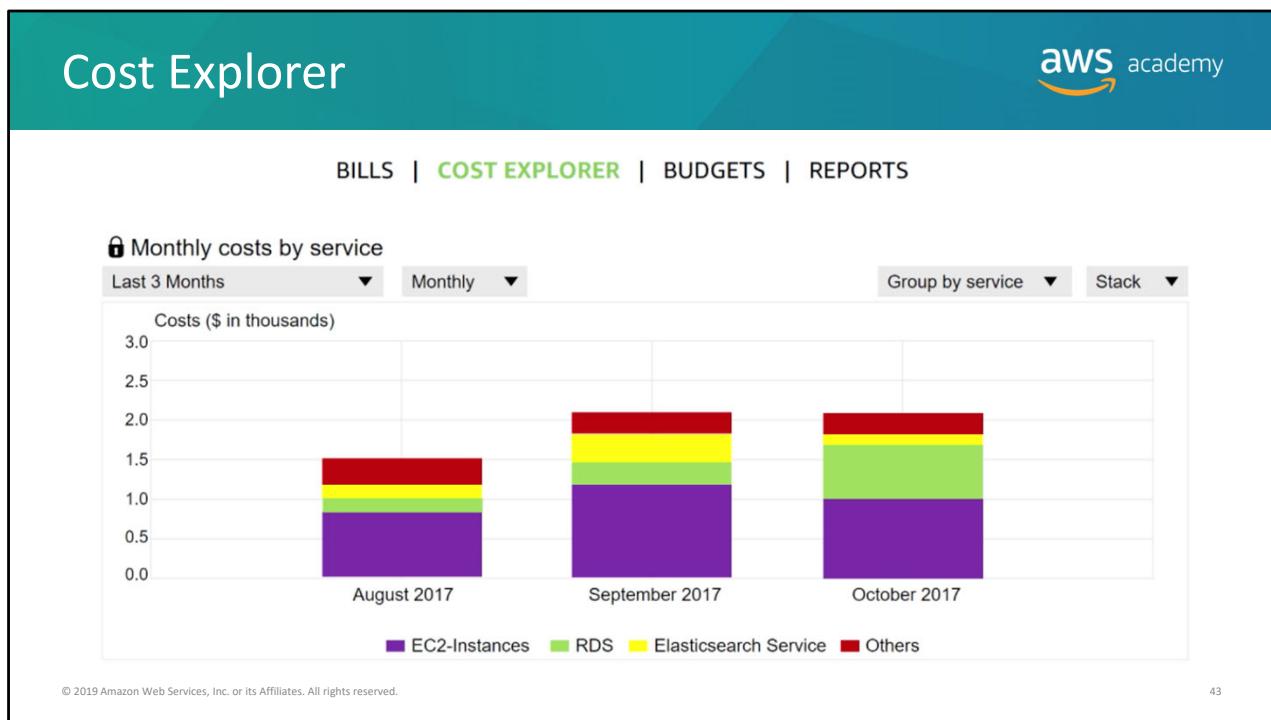
Total	\$7,453.41 USD
<b>AWS Marketplace Charges</b>	\$15.00
▼ Usage Charges and Recurring Fees	\$15.00
Invoice 32342548 – AWS Service Charges: Usage charge for this statement period	2017-10-10
Invoice 32342513 – AWS Service Charges: Usage charge for this statement period	2017-10-10
<b>AWS Service Charges</b>	\$7,438.41
▼ Usage Charges and Recurring Fees	\$7,414.41
Invoice 32342513 – AWS Service Charges: Usage charge for this statement period	2017-10-10
Invoice 32342507 – AWS Service Charges: Subscription charge	2017-10-10

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

42

The **AWS Bills page** lists the costs that you incurred over the past month for each AWS service, with a further breakdown by AWS Region and linked account.

This tool gives you access to the most up-to-date information on your costs and usage, including your monthly bill and the detailed breakdown of the AWS services that you use.



The **AWS Billing and Cost Management** console includes the **Cost Explorer** page for viewing your AWS cost data as a graph.

With Cost Explorer, you can visualize, understand, and manage your AWS costs and usage over time.

The Cost Explorer includes a default report that visualizes your costs and usage for your top cost-incurring AWS services. The monthly running costs report gives you an overview of all your costs for the past 3 months. It also provides forecasted numbers for the coming month, with a corresponding confidence interval.

The Cost Explorer is a free tool that enables you to:

- View charts of your costs.
- View cost data for the past 13 months.
- Forecast how much you are likely to spend over the next 3 months.
- Discover patterns in how much you spend on AWS resources over time and identify cost problem areas.
- Identify the services that you use the most
- View metrics, like which Availability Zones have the most traffic or which linked AWS account is used the most.

# Forecast and track costs



BILLS | COST EXPLORER | **BUDGETS** | REPORTS

Create budget		Copy	Edit	Delete	Download CSV		⋮
Filter by budget name							
	Budget name	Current	Forecasted	Budgeted	Current vs. budgeted	Forecasted vs. budgeted	⋮
<input type="checkbox"/>	▶ Total Monthly Cost	\$760.27	\$787.44	\$1,000.00			
<input type="checkbox"/>	▼ S3 Usage Bucket	2978.00 Req	3650.16 Req	3000.00 Req			

Budget details

Start date 10/01/17

End date -

Budget Period Monthly

Variance analysis

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

44

**AWS Budgets** uses the cost visualization that is provided by Cost Explorer to show you the status of your budgets and to provide forecasts of your estimated costs.

You can also use AWS Budgets to create notifications for when you go over your budget for the month, or when your estimated costs exceed your budget. Budgets can be tracked at the monthly, quarterly, or yearly level, and you can customize the start and end dates. Budget alerts can be sent via email or via **Amazon Simple Notification Service (Amazon SNS)**.

# Cost and usage reporting



BILLS | COST EXPLORER | BUDGETS | REPORTS

Product Code	Usage Type	Operation	Availability Zone	Usage Amount	Currency Code	Line Item Description
Amazon S3	Requests – Tier 1	ListAllMyBuckets		2	USD	\$0.00 per request – PUT, COPY, POST, LIST under the global free tier
Amazon EC2	USW2-Boxusage:t2.micro	Runinstnaces:0002	us-west-2a	1	USD	\$0.00 per Windows t2.micro instance-hour under monthly free tier
Amazon S3	Requests – Tier 1	ListAllMyBuckets		2	USD	\$0.00 per request – PUT, COPY, POST, LIST under the global free tier
Amazon EC2	USW2-Boxusage:t2.micro	Runinstnaces:0002	us-west-2a	1	USD	\$0.00 per Windows t2.micro instance-hour under monthly free tier
Amazon S3	Requests – Tier 1	ListAllMyBuckets		2	USD	\$0.00 per request – PUT, COPY, POST, LIST under the global free tier
Amazon S3	Requests – Tier 1	ListAllMyBuckets		2	USD	\$0.00 per request – PUT, COPY, POST, LIST under the global free tier

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

45

The **AWS Cost and Usage Report** is a single location for accessing comprehensive information about your AWS costs and usage. This tool lists the usage for each service category that is used by an account (and its users) in hourly or daily line items, and any tax that you activated for tax allocation purposes.

You can choose to have AWS to publish billing reports to an S3 bucket. These reports can be updated once a day.

Recorded demo:  
Amazon Billing  
dashboard

46



Amazon Billing  
dashboard demo

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

Show the [Billing dashboard demo](#)

# Billing dashboard demonstration



**Getting Started with AWS Billing & Cost Management**

- Manage your costs and usage using [AWS Budgets](#)
- View your cost drivers and usage trends via [Cost Explorer](#)
- Dive deeper into your costs using the [Cost and Usage Reports](#) with Athena integration
- [Learn more:](#) Check out the [AWS What's New](#) webpage
- Do you have [Reserved Instances \(RIs\)](#)?
- Access the RI Utilization & Coverage reports—and RI purchase recommendations—via [Cost Explorer](#).

**Spend Summary**

Welcome to the AWS Billing & Cost Management console. Your last month, month-to-date, and month-end forecasted costs appear below.

Current month-to-date balance for September 2019

**\$168.20**

**Month-to-Date Spend by Service**

The chart below shows the proportion of costs spent for each service you use.

**Bill Details**

**\$168.2**

Service	Cost
ES	\$74.52
DatabaseMigrationSvc	\$32.12
SageMaker	\$29.99
EC2	\$16.59
Other Services	\$14.98
Tax	\$0.00
<b>Total</b>	<b>\$168.20</b>

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

47

Show the Amazon Billing dashboard demo

Module 2: Cloud Economics and Billing

## Section 5: Technical support

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

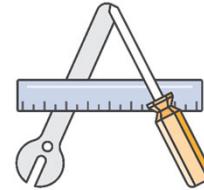


Introducing Section 5: Technical support.

# AWS support



- Provide unique combination of tools and expertise:
  - AWS Support
  - AWS Support Plans
- Support is provided for:
  - Experimenting with AWS
  - Production use of AWS
  - Business-critical use of AWS



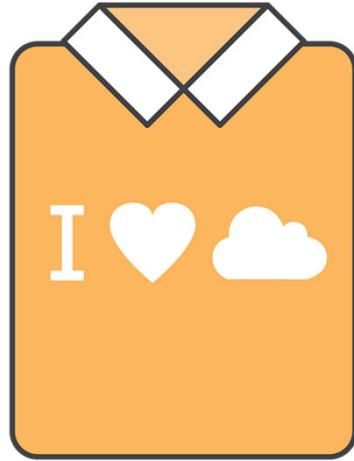
Whether you are new or continuing to adopt AWS services and applications as your business solutions, AWS want help you do amazing things with AWS. AWS Support can provide you with a unique combination of tools and expertise based on your current or future planned use cases.

AWS Support was developed to provide complete support and the right resources to aid your success. We want to support all our customers, including customers that might be experimenting with AWS, those that are looking for production uses of AWS, and also customers that use AWS as a business-critical resource. AWS Support can vary the type of support that is provided, depending on the customer's needs and goals.

## AWS support



- Proactive guidance :
  - Technical Account Manager (TAM)
- Best practices :
  - AWS Trusted Advisor
- Account assistance :
  - AWS Support Concierge



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

50

With AWS, customers can plan, deploy, and optimize with confidence.

If you would like proactive guidance, AWS Support has **Technical Account Managers (TAMs)** who are designated as that user's primary point of contact. The TAM can provide guidance, architectural review, and continuous ongoing communication to keep you informed and prepared as you plan, deploy, and optimize your solutions.

If you want to ensure that you follow best practices to increase performance and fault tolerance in the AWS environment, AWS Support has **AWS Trusted Advisor**. AWS Trusted Advisor is like a customized cloud expert. It is an online resource that checks for opportunities to reduce monthly expenditures and increase productivity.

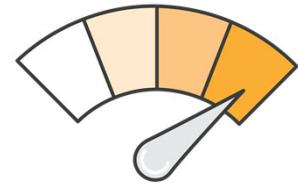
For account assistance, the **Support Concierge** is a billing and account expert who will provide quick and efficient analysis on billing and account issues. The concierge addresses all non-technical billing and account-level inquiries.

## Support plans



AWS Support offers four support plans:

- **Basic Support** – Resource Center access, Service Health Dashboard, product FAQs, discussion forums, and support for health checks
- **Developer Support**: Support for early development on AWS
- **Business Support**: Customers that run production workloads
- **Enterprise Support**: Customers that run business and mission-critical workloads



AWS wants you to be able to plan, deploy, and optimize with confidence. We have developed specific plans to support you, including Basic, Developer, Business, and Enterprise support plans.

- The **Basic Support Plan** offers:
  - 24/7 access to customer service, documentation, whitepapers and support forums.
  - Access to six core Trusted Advisor checks.
  - Access to Personal Health Dashboard.
- The **Developer Support Plan** offers resources for customers that are testing or doing early development on AWS, and any customers who:
  - Want access to guidance and technical support.
  - Are exploring how to quickly put AWS to work.
  - Use AWS for non-production workloads or applications.
- The **Business Support Plan** offers resources for customers that are running production workloads on AWS, and any customers who:
  - Run one or more applications in production environments.
  - Have multiple services activated, or use key services extensively.

- Depend on their business solutions to be available, scalable, and secure.
- The **Enterprise Support Plan** offers resources for customers that are running business and mission-critical workloads on AWS, and any customers who want to:
  - Focus on proactive management to increase efficiency and availability.
  - Build and operate workloads that follow AWS best practices.
  - Use AWS expertise to support launches and migrations.
  - Use a Technical Account Manager (TAM), who provides technical expertise for the full range of AWS services and obtains a detailed understanding of your use case and technology architecture. The Technical Account Manager is the primary point of contact for ongoing support needs.

## Case severity and response times



	Critical	Urgent	High	Normal	Low
Basic	No Case Support				
Developer Plan (Business hours)				12 hours or less	24 hours or less
Business Plan (24/7)		1 hour or less	4 hours or less	12 hours or less	24 hours or less
Enterprise Plan (24/7)	15 minutes or less	1 hour or less	4 hours or less	12 hours or less	24 hours or less

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

52

In addition to understanding the costs that are associated with different support plans, it is critical that you understand the service levels that are associated with each plan. In addition to the support plan you select, the case severity will drive the type of response that you receive. There are five different severity levels:

- **Critical** – Your business is at risk. Critical functions of your application are unavailable.
- **Urgent** – Your business is significantly impacted. Important functions of your application are unavailable.
- **High** – Important functions of your application are impaired or degraded.
- **Normal** – Non-critical functions of your application are behaving abnormally, or you have a time-sensitive development question.
- **Low** – You have a general development question, or you want to request a feature.

Note that there is no case support with the Basic Support Plan. These response times should be considered when you determine which support plan is best for your organization.

To learn more about AWS Support plans, see [Compare AWS Support Plans](#).

## Activity: Support plan scavenger hunt



- Break up into groups of four or five and develop a recommendation for the best support plan for one of the business cases that are provided.
- Be prepared to report your findings back to the class.

In this activity, your group will read the description of a business and develop a recommendation for the appropriate support plan. When you report back to the class, describe the support plan that you selected, and the decision-making criteria that you used to develop your recommendation.

Module 2: Cloud Economics and Billing

## Module wrap-up

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



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



- Explored the fundamental of AWS pricing
- Reviewed TCO concepts
- Reviewed an AWS Pricing Calculator estimate
- Reviewed the Billing dashboard
- Reviewed Technical Support options and costs

In summary you:

- Explored the fundamentals of AWS pricing
- Reviewed Total Cost of Ownership concepts
- Reviewed an AWS Pricing Calculator estimate.

Total Cost of Ownership is a concept to help you understand and compare the costs that are associated with different deployments. AWS provides the AWS Pricing Calculator to assist you with the calculations that are needed to estimate cost savings.

Use the **AWS Pricing Calculator** to:

- Estimate monthly costs
- Identify opportunities to reduce monthly costs
- Model your solutions before building them
- Explore price points and calculations behind your estimate
- Find the available instance types and contracts that meet your needs

**AWS Billing and Cost Management** provides you with tools to help you access, understand, allocate, control, and optimize your AWS costs and usage. These tools include AWS Bills, AWS Cost Explorer, AWS Budgets, and AWS Cost and Usage Reports.

These tools give you access to the most comprehensive information about your AWS costs and usage including which AWS services are the main cost drivers. Knowing and understanding your usage and costs will enable you to plan ahead and improve your AWS implementation.

## Complete the knowledge check



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

56

Now, complete the knowledge check.

## Sample exam question



Which AWS service provides infrastructure security optimization recommendations?

- A. AWS Price List Application Programming Interface (API)
- B. Reserved Instances
- C. AWS Trusted Advisor
- D. Amazon Elastic Compute Cloud (Amazon EC2) Spot Fleet

Let's look at the answer choices and rule them out based on the keywords we have previously highlighted.

## Additional resources



- AWS Economics Center: <http://aws.amazon.com/economics/>
- AWS Pricing Calculator: <https://calculator.aws/#/>
- Case studies and research: <http://aws.amazon.com/economics/>
- Additional pricing exercises: <https://dx1572sre29wk.cloudfront.net/cost/>

If you want to learn more about the topics covered in this module, you might find the following additional resources helpful:

- AWS Economics Center: <http://aws.amazon.com/economics/>
- AWS Pricing Calculator: <https://calculator.aws/#/>
- Case studies and research: <http://aws.amazon.com/economics/>
- Additional pricing exercises: <https://dx1572sre29wk.cloudfront.net/cost/>

# Thank you

© 2019 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections or feedback on the course, please email us at: [aws-course-feedback@amazon.com](mailto:aws-course-feedback@amazon.com). For all other questions, contact us at: <https://aws.amazon.com/contact-us/aws-training/>. All trademarks are the property of their owners.



Thanks for participating!

AWS Academy Cloud Foundations

## Module 3: AWS Global Infrastructure Overview

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



Welcome to Module 3: AWS Global Infrastructure Overview.

# Module overview



## Topics

- AWS Global Infrastructure
- AWS service and service category overview

## Demo

- AWS Global Infrastructure

## Activities

- AWS Management Console clickthrough



### Knowledge check

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

2

This module will address the following topics:

- AWS Global Infrastructure
- AWS service and service category overview

The module includes an educator-led demonstration that focuses on the details of the AWS Global Infrastructure. The module also includes a hands-on activity where you will explore the AWS Management Console.

Finally, you will be asked to complete a knowledge check that will test your understanding of the key concepts that are covered in this module.

## Module objectives



After completing this module, you should be able to:

- Identify the difference between AWS Regions, Availability Zones, and edge locations
- Identify AWS service and service categories

After completing this module, you should be able to:

- Identify the difference between AWS Regions, Availability Zones, and edge locations
- Identify AWS service and service categories

Module 3: AWS Global Infrastructure Overview

## Section 1: AWS Global Infrastructure

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



Introducing Section 1: AWS Global Infrastructure.

# AWS Global Infrastructure



- The **AWS Global Infrastructure** is designed and built to deliver a **flexible, reliable, scalable**, and **secure** cloud computing environment with high-quality **global network performance**.
- AWS continually updates its global infrastructure footprint. Visit one of the following web pages for current infrastructure information:

- [AWS Global Infrastructure Map](#)

Choose a circle on the map to view summary information about the Region represented by the circle.

- [Regions and Availability Zones](#)

Choose a tab to view a map of the selected geography and a list of Regions, Edge locations, Local zones, and Regional Caches.



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

5

To learn more about the AWS Regions that are currently available, use one of the following links:

- <https://aws.amazon.com/about-aws/global-infrastructure/#AWS Global Infrastructure Map>
- [https://aws.amazon.com/about-aws/global-infrastructure/regions\\_az/](https://aws.amazon.com/about-aws/global-infrastructure/regions_az/)

These resources are updated frequently to show current and planned AWS infrastructure.

## Educator-Led Demo: AWS Global Infrastructure Details



6

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

The educator might now choose to conduct a live demonstration of the AWS Global Infrastructure map introduced on the previous slide. This resource provides an interactive way to learn about the AWS Global Infrastructure. The remaining slides in this section cover many of the same topics and go into greater detail on some topics.

# AWS Regions



- An **AWS Region** is a geographical area.
  - **Data replication** across Regions is controlled by you.
  - **Communication** between Regions uses AWS backbone network infrastructure.
- Each Region provides full redundancy and connectivity to the network.
- A Region typically consists of two or more **Availability Zones**.



Example: London Region

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

7

The AWS Cloud infrastructure is built around Regions. AWS has 22 Regions worldwide. An **AWS Region** is a physical geographical location with one or more **Availability Zones**. Availability Zones in turn consist of one or more **data centers**.

To achieve fault tolerance and stability, Regions are isolated from one another. Resources in one Region are not automatically replicated to other Regions. When you store data in a specific Region, it is not replicated outside that Region.

It is your responsibility to replicate data across Regions, if your business needs require it.

AWS Regions that were introduced before March 20, 2019 are *enabled* by default. Regions that were introduced after March 20, 2019—such as Asia Pacific (Hong Kong) and Middle East (Bahrain)—are *disabled* by default. You must enable these Regions before you can use them. You can use the AWS Management Console to enable or disable a Region.

Some Regions have restricted access. An Amazon AWS (**China**) account provides access to the Beijing and Ningxia Regions only. To learn more about AWS in China, see: <https://www.amazonaws.cn/en/about-aws/china/>. The isolated **AWS GovCloud (US)** Region is designed to allow US government agencies and customers to move sensitive workloads into the cloud by addressing their specific regulatory and compliance requirements.

# Selecting a Region



Determine the right Region for your services, applications, and data based on these factors



Data governance, legal requirements



Proximity to customers (latency)



Services available within the Region



Costs (vary by Region)

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

8

There are a few factors that you should consider when you select the optimal Region or Regions where you store data and use AWS services.

One essential consideration is **data governance and legal requirements**. Local laws might require that certain information be kept within geographical boundaries. Such laws might restrict the Regions where you can offer content or services. For example, consider the European Union (EU) Data Protection Directive.

All else being equal, it is generally desirable to run your applications and store your data in a Region that is as close as possible to the user and systems that will access them. This will help you **reduce latency**. CloudPing is one website that you can use to test latency between your location and all AWS Regions. To learn more about CloudPing, see: <http://www.cloudping.info/>

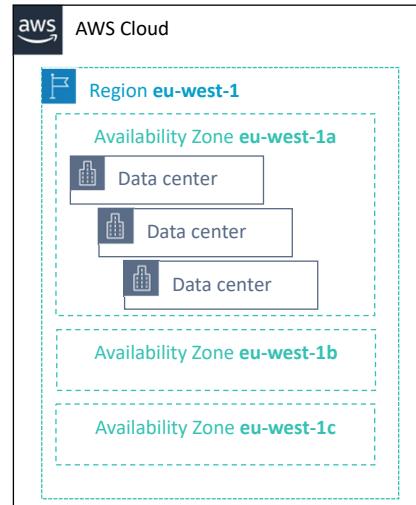
Keep in mind that not all services are available in all Regions. To learn more, see: <https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/?p=tgi&loc=4>.

Finally, there is some variation in the **cost** of running services, which can depend on which Region you choose. For example, as of this writing, running an On-Demand t3.medium size Amazon Elastic Compute Cloud (Amazon EC2) Linux instance in the US East (Ohio) Region costs \$0.0416 per hour, but running the same instance in the Asia Pacific (Tokyo) Region costs \$0.0544 per hour.

# Availability Zones



- Each **Region** has multiple Availability Zones.
- Each **Availability Zone** is a fully isolated partition of the AWS infrastructure.
  - Availability Zones consist of discrete **data centers**
  - They are designed for fault isolation
  - They are interconnected with other Availability Zones by using high-speed private networking
  - You choose your Availability Zones.
  - **AWS recommends replicating data and resources across Availability Zones** for resiliency.



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

9

Each AWS Region has multiple, isolated locations that are known as *Availability Zones*.

Each Availability Zone provides the ability to operate applications and databases that are more highly available, fault-tolerant, and scalable than would be possible with a single data center. Each Availability Zone can include multiple data centers (typically three), and at full-scale, they can include hundreds of thousands of servers. They are fully isolated partitions of the AWS Global Infrastructure. Availability Zones have their own power infrastructure, and they are physically separated by many kilometers from other Availability Zones—though all Availability Zones are within 100 km of each other.

All Availability Zones are interconnected with high-bandwidth, low-latency networking over fully redundant, dedicated fiber that provides high-throughput between Availability Zones. The network accomplishes synchronous replication between Availability Zones.

Availability Zones help build highly available applications. When an application is partitioned across Availability Zones, companies are better isolated and protected from issues such as lightning, tornadoes, earthquakes, and more.

You are responsible for selecting the Availability Zones where your systems will reside. Systems can span multiple Availability Zones. AWS recommends replicating across Availability Zones for resiliency. You should design your systems to survive the temporary or prolonged failure of an Availability Zone if a disaster occurs.

- AWS data centers are **designed for security**.
- Data centers are where the data resides and data processing occurs.
- Each data center has redundant power, networking, and connectivity, and is housed in a separate facility.
- A data center typically has 50,000 to 80,000 physical servers.



The foundation for the AWS infrastructure is the data centers. Customers do not specify a data center for the deployment of resources. Instead, an Availability Zone is the most granular level of specification that a customer can make. However, a data center is the location where the actual data resides. Amazon operates state-of-the-art, highly available data centers. Although rare, failures can occur that affect the availability of instances in the same location. If you host all your instances in a single location that is affected by such a failure, none of your instances will be available.

Data centers are securely designed with several factors in mind:

- Each location is carefully evaluated to **mitigate environmental risk**.
- Data centers have a **redundant design** that anticipates and tolerates failure while maintaining service levels.
- To ensure availability, **critical system components are backed up** across multiple Availability Zones.
- To ensure capacity, AWS continuously monitors service usage to deploy infrastructure to support availability commitments and requirements.
- Data center **locations are not disclosed** and all access to them is restricted.
- In case of failure, automated processes move data traffic away from the affected area.

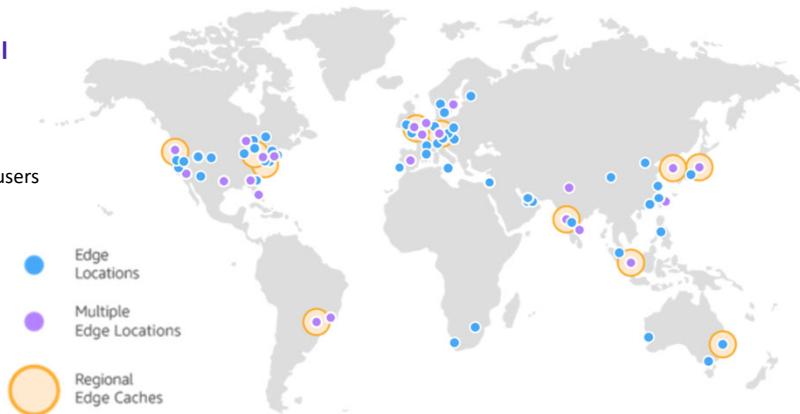
AWS uses **custom network equipment** sourced from **multiple original device manufacturers (ODMs)**. ODMs design and manufacture products based on specifications

from a second company. The second company then rebrands the products for sale.

# Points of Presence



- AWS provides a global network of **Points of Presence** locations
- Consists of **edge locations** and a much smaller number of **Regional edge caches**
- Used with Amazon CloudFront
  - A global Content Delivery Network (CDN), that delivers content to end users with **reduced latency**
- Regional edge caches used for content with infrequent access.



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

11

**Amazon CloudFront** is a **content delivery network** (CDN) used to distribute content to end users to reduce latency. **Amazon Route 53** is a Domain Name System (DNS) service. Requests going to either one of these services will be routed to the nearest **edge location** automatically in order to lower latency.

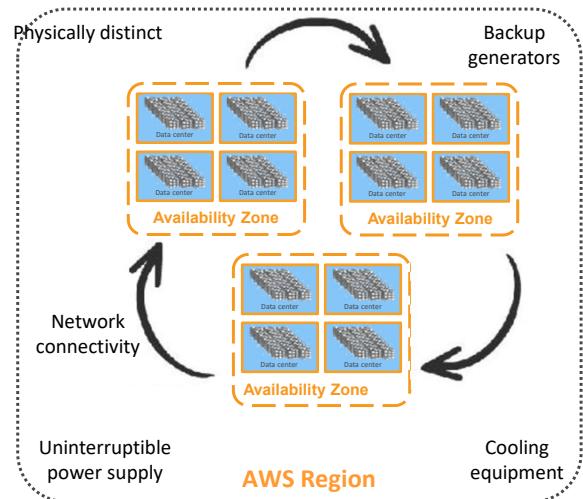
AWS **Points of Presence** are located in most of the major cities around the world. By **continuously measuring internet connectivity, performance and computing to find the best way to route requests**, the Points of Presence deliver a better near real-time user experience. They are used by many AWS services, including Amazon CloudFront, Amazon Route 53, AWS Shield, and AWS Web Application Firewall (AWS WAF) services.

**Regional edge caches** are used by default with Amazon CloudFront. Regional edge caches are used 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 that content having to be fetched from the origin server.

# AWS infrastructure features



- Elasticity and scalability
  - Elastic infrastructure; dynamic adaption of capacity
  - Scalable infrastructure; adapts to accommodate growth
- Fault-tolerance
  - Continues operating properly in the presence of a failure
  - Built-in redundancy of components
- High availability
  - High level of operational performance
  - Minimized downtime
  - No human intervention



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

12

Now that you have a good understanding of the major components that comprise the AWS Global Infrastructure, let's consider the benefits provided by this infrastructure.

The AWS Global Infrastructure has several valuable features:

- First, it is **elastic** and **scalable**. This means resources can dynamically adjust to increases or decreases in capacity requirements. It can also rapidly adjust to accommodate growth.
- Second, this infrastructure is **fault tolerant**, which means it has built-in component redundancy which enables it to continue operations despite a failed component.
- Finally, it requires minimal to no human intervention, while providing **high availability** with minimal down time.

## Key takeaways



13



- The **AWS Global Infrastructure** consists of **Regions** and **Availability Zones**.
- Your choice of a **Region** is typically based on **compliance requirements** or to **reduce latency**.
- Each **Availability Zone** is physically separate from other Availability Zones and has redundant power, networking, and connectivity.
- **Edge locations**, and **Regional edge caches** improve performance by **caching** content closer to users.

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

Some key takeaways from this section of the module include:

- The AWS Global Infrastructure consists of Regions and Availability Zones.
- Your choice of a Region is typically based on compliance requirements or to reduce latency.
- Each Availability Zone is physically separate from other Availability Zones and has redundant power, networking, and connectivity.
- Edge locations, and Regional edge caches improve performance by caching content closer to users.

Module 3: AWS Global Infrastructure Overview

## Section 2: AWS services and service category overview

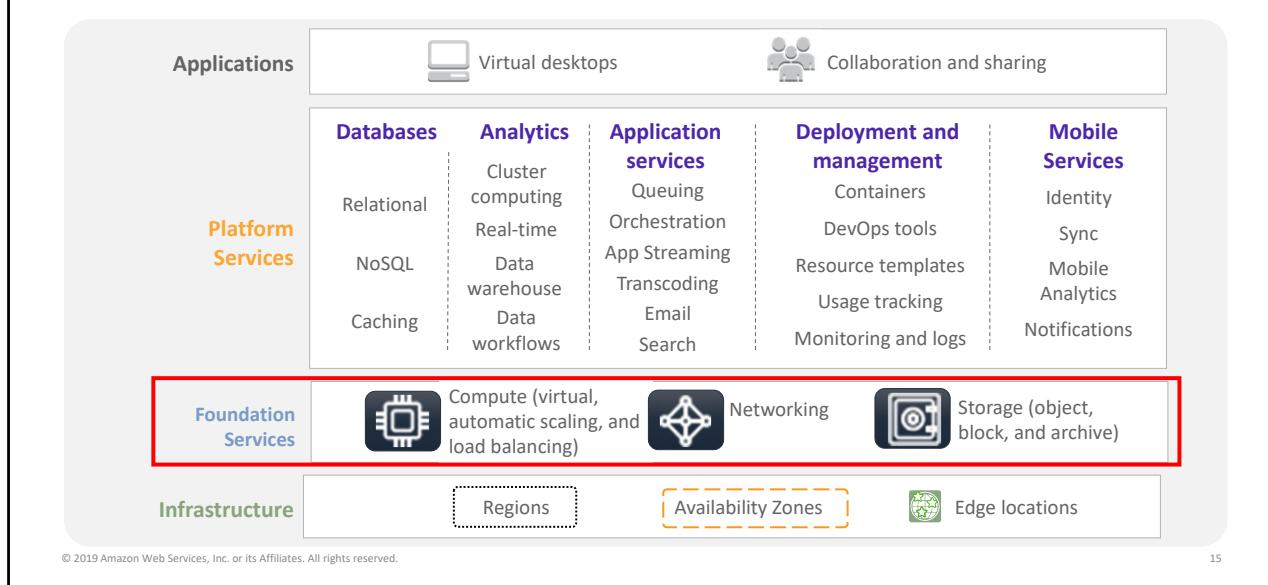
© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Introducing Part 2: AWS Service and Service Category Overview.

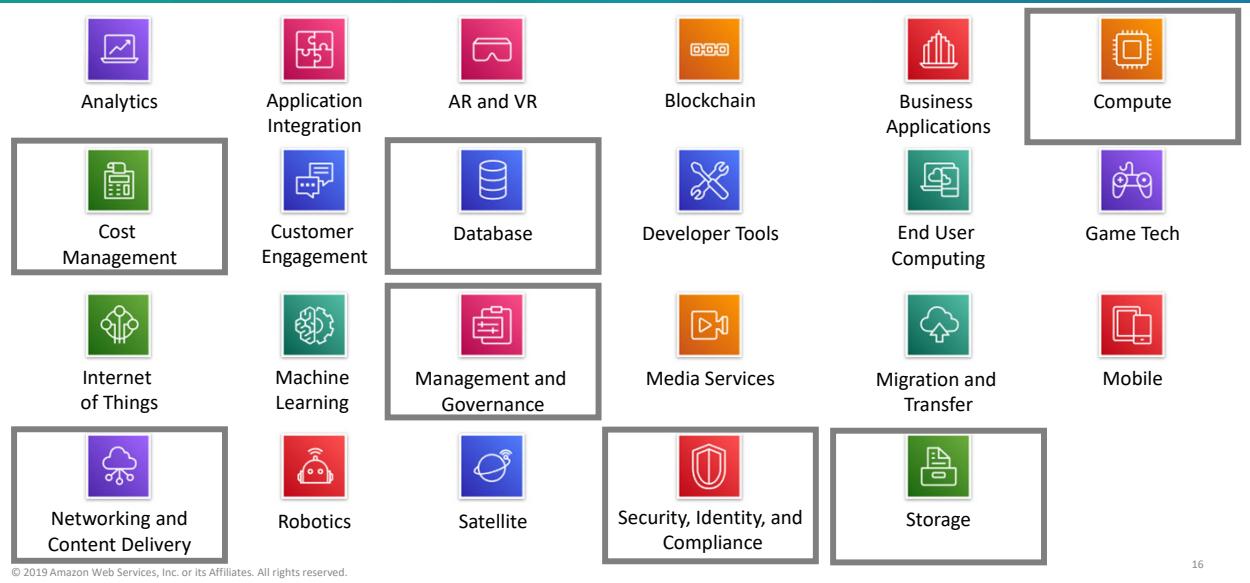
AWS offers a broad set of global cloud-based products that can be used as building blocks for common cloud architectures. Here is a look at how these cloud based products are organized.

# AWS foundational services



As discussed previously, the AWS Global Infrastructure can be broken down into three elements: Regions, Availability Zones, and Points of Presence, which include edge locations. This infrastructure provides the platform for a broad set of services, such as networking, storage, compute services, and databases—and these services are delivered as an on-demand utility that is available in seconds, with pay-as-you-go pricing.

# AWS categories of services



16

AWS offers a broad set of cloud-based services. There are 23 different product or service categories, and each category consists of one or more services. This course will not attempt to introduce you to each service. Rather, the focus of this course is on the services that are most widely used and offer the best introduction to the AWS Cloud. This course also focuses on services that are more likely to be covered in the AWS Certified Cloud Practitioner exam.

The categories that this course will discuss are highlighted on the slide: Compute, Cost Management, Database, Management and Governance, Networking and Content Delivery, Security, Identity, and Compliance, and Storage.

To learn more about AWS products, see [Cloud Products](#). All AWS products are organized into the service categories that are shown here. For example, if you click **Compute**, you will see that Amazon Elastic Compute Cloud (Amazon EC2) is first on the list. The compute category also lists many other products and services.

If you click **Amazon EC2**, it takes you to the Amazon EC2 page. Each product page provides a detailed description of the product and lists some of its benefits.

Explore the different service groups to understand the categories and services within them. Now that you know how to locate information about different services, this module will

discuss the highlighted service categories. **The next seven slides list the individual services—within each of the categories highlighted above—that this course will discuss.**

# Storage service category



Photo from <https://www.pexels.com/photo/black-and-grey-device-159282/>



## AWS storage services



Amazon Simple Storage Service (Amazon S3)



Amazon Elastic Block Store (Amazon EBS)



Amazon Elastic File System (Amazon EFS)



Amazon Simple Storage Service Glacier

17

AWS storage services include the services listed here, and many others.

**Amazon Simple Storage Service (Amazon S3)** is an object storage service that offers scalability, data availability, security, and performance. Use it to store and protect any amount of data for websites, mobile apps, backup and restore, archive, enterprise applications, Internet of Things (IoT) devices, and big data analytics.

**Amazon Elastic Block Store (Amazon EBS)** is high-performance block storage that is designed for use with Amazon EC2 for both throughput and transaction intensive workloads. It is used for a broad range of workloads, such as relational and non-relational databases, enterprise applications, containerized applications, big data analytics engines, file systems, and media workflows.

**Amazon Elastic File System (Amazon EFS)** provides a scalable, fully managed elastic Network File System (NFS) file system for use with AWS Cloud services and on-premises resources. It is built to scale on demand to petabytes, growing and shrinking automatically as you add and remove files. It reduces the need to provision and manage capacity to accommodate growth.

**Amazon Simple Storage Service Glacier** is a secure, durable, and extremely low-cost Amazon S3 cloud storage class for data archiving and long-term backup. It is designed to deliver 11 9s of durability, and to provide comprehensive security and compliance

capabilities to meet stringent regulatory requirements.

# Compute service category

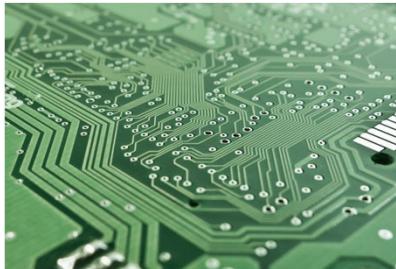


Photo from <https://www.pexels.com/photo/technology-computer-lines-board-50711/>



AWS Compute services



Amazon EC2



Amazon EC2  
Auto Scaling



Amazon Elastic  
Container Service  
(Amazon ECS)



Amazon EC2  
Container Registry



AWS Elastic  
Beanstalk



AWS Lambda



Amazon Elastic  
Kubernetes Service  
(Amazon EKS)



AWS Fargate

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

18

AWS compute services include the services listed here, and many others.

**Amazon Elastic Compute Cloud (Amazon EC2)** provides resizable compute capacity as virtual machines in the cloud.

**Amazon EC2 Auto Scaling** enables you to automatically add or remove EC2 instances according to conditions that you define.

**Amazon Elastic Container Service (Amazon ECS)** is a highly scalable, high-performance container orchestration service that supports Docker containers.

**Amazon Elastic Container Registry (Amazon ECR)** is a fully-managed Docker container registry that makes it easy for developers to store, manage, and deploy Docker container images.

**AWS Elastic Beanstalk** is a service for deploying and scaling web applications and services on familiar servers such as Apache and Microsoft Internet Information Services (IIS).

**AWS Lambda** enables you to run code without provisioning or managing servers. You pay only for the compute time that you consume. There is no charge when your code is not

running.

**Amazon Elastic Kubernetes Service (Amazon EKS)** makes it easy to deploy, manage, and scale containerized applications that use Kubernetes on AWS.

**AWS Fargate** is a compute engine for Amazon ECS that allows you to run containers without having to manage servers or clusters.

## Database service category

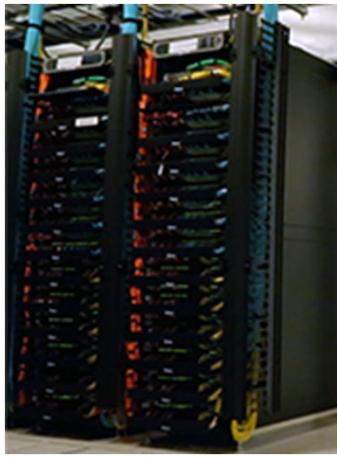


Photo from <https://aws.amazon.com/compliance/data-center/data-centers/>



AWS Database services



Amazon Relational  
Database Service



Amazon Aurora



Amazon Redshift



Amazon  
DynamoDB

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

19

AWS database services include the services listed here, and many others.

**Amazon Relational Database Service (Amazon RDS)** makes it easy to set up, operate, and scale a relational database in the cloud. It provides resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching, and backups.

**Amazon Aurora** is a MySQL and PostgreSQL-compatible relational database. It is up to five times faster than standard MySQL databases and three times faster than standard PostgreSQL databases.

**Amazon Redshift** enables you to run analytic queries against petabytes of data that is stored locally in Amazon Redshift, and directly against exabytes of data that are stored in Amazon S3. It delivers fast performance at any scale.

**Amazon DynamoDB** is a key-value and document database that delivers single-digit millisecond performance at any scale, with built-in security, backup and restore, and in-memory caching.

# Networking and content delivery service category



Photo by Umberto on Unsplash



AWS networking  
and content delivery services



Amazon VPC



Elastic Load  
Balancing



Amazon  
CloudFront



AWS Transit  
Gateway



Amazon  
Route 53



AWS Direct  
Connect



AWS VPN

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

20

AWS networking and content delivery services include the services listed here, and many others.

**Amazon Virtual Private Cloud (Amazon VPC)** enables you to provision logically isolated sections of the AWS Cloud.

**Elastic Load Balancing** automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions.

**Amazon CloudFront** is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and application programming interfaces (APIs) to customers globally, with low latency and high transfer speeds.

**AWS Transit Gateway** is a service that enables customers to connect their Amazon Virtual Private Clouds (VPCs) and their on-premises networks to a single gateway.

**Amazon Route 53** is a scalable cloud Domain Name System (DNS) web service designed to give you a reliable way to route end users to internet applications. It translates names (like `www.example.com`) into the numeric IP addresses (like `192.0.2.1`) that computers use to

connect to each other.

**AWS Direct Connect** provides a way to establish a dedicated private network connection from your data center or office to AWS, which can reduce network costs and increase bandwidth throughput.

**AWS VPN** provides a secure private tunnel from your network or device to the AWS global network.

## Security, identity, and compliance service category



Photo by Paweł Czerwiński on Unsplash



AWS security, identity,  
and compliance services



AWS Identity and Access  
Management (IAM)



AWS  
Organizations



Amazon Cognito



AWS Artifact



AWS Key  
Management  
Service



AWS Shield

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

21

AWS security, identity, and compliance services include the services listed here, and many others.

**AWS Identity and Access Management (IAM)** enables you to manage access to AWS services and resources securely. By using IAM, you can create and manage AWS users and groups. You can use IAM permissions to allow and deny user and group access to AWS resources.

**AWS Organizations** allows you to restrict what services and actions are allowed in your accounts.

**Amazon Cognito** lets you add user sign-up, sign-in, and access control to your web and mobile apps.

**AWS Artifact** provides on-demand access to AWS security and compliance reports and select online agreements.

**AWS Key Management Service (AWS KMS)** enables you to create and manage keys. You can use AWS KMS to control the use of encryption across a wide range of AWS services and in your applications.

**AWS Shield** is a managed Distributed Denial of Service (DDoS) protection service that

safeguards applications running on AWS.

## AWS cost management service category

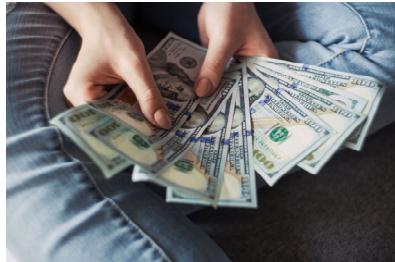


Photo by Alexander Mills on Unsplash



AWS cost management services



AWS Cost and Usage Report



AWS Budgets



AWS Cost Explorer

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

22

AWS cost management services include the services listed here, and others.

**The AWS Cost and Usage Report** contains the most comprehensive set of AWS cost and usage data available, including additional metadata about AWS services, pricing, and reservations.

**AWS Budgets** enables you to set custom budgets that alert you when your costs or usage exceed (or are forecasted to exceed) your budgeted amount.

**AWS Cost Explorer** has an easy-to-use interface that enables you to visualize, understand, and manage your AWS costs and usage over time.

## Management and governance service category



Photo by Marta Branco from Pixels



AWS management and governance services



AWS Management Console



AWS Config



Amazon CloudWatch



AWS Auto Scaling



AWS Command Line Interface



AWS Trusted Advisor



AWS Well-Architected Tool



AWS CloudTrail

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

23

AWS management and governance services include the services listed here, and others.

**The AWS Management Console** provides a web-based user interface for accessing your AWS account.

**AWS Config** provides a service that helps you track resource inventory and changes.

**Amazon CloudWatch** allows you to monitor resources and applications.

**AWS Auto Scaling** provides features that allow you to scale multiple resources to meet demand.

**AWS Command Line Interface** provides a unified tool to manage AWS services.

**AWS Trusted Advisor** helps you optimize performance and security.

**AWS Well-Architected Tool** provides help in reviewing and improving your workloads.

**AWS CloudTrail** tracks user activity and API usage.

## Activity: AWS Management Console clickthrough

24



Photo by Pixabay from Pexels.

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

In this educator-led activity, you will be asked to log in to the AWS Management Console. The activity instructions are on the next slide. You will be challenged to answer five questions. The educator will lead the class in a discussion of each question, and reveal the correct answers.

# Hands-on activity: AWS Management Console clickthrough



1. Launch the [Sandbox](#) hands-on environment and connect to the [AWS Management Console](#).
2. Explore the AWS Management Console.
  - A. Click the **Services** menu.
  - B. Notice how services are grouped into service categories. For example, the **EC2** service appears in the **Compute** service category.  
**Question #1:** Under which service category does the **IAM** service appear?  
**Question #2:** Under which service category does the **Amazon VPC** service appear?
  - C. Click the **Amazon VPC** service. Notice that the dropdown menu in the top-right corner displays an AWS Region (for example, it might display *N. Virginia*).
  - D. Click the Region menu and switch to a different Region. For example, choose **EU (London)**.
  - E. Click **Subnets** (on the left side of the screen). The Region has three subnets in it. Click the box next to one of the subnets. Notice that the bottom half of the screen now displays details about this subnet.  
**Question #3:** Does the subnet you selected exist at the level of the Region or at the level of the Availability Zone?
  - F. Click **Your VPCs**. An existing VPC is already selected.  
**Question #4:** Does the VPC exist at the level of the Region or the level of the Availability Zone?  
**Question #5:** Which services are global instead of Regional? Check Amazon EC2, IAM, Lambda, and Route 53.

The purpose of this activity is to expose you to the AWS Management Console. You will gain experience navigating between AWS service consoles (such as the Amazon VPC console). You will also practice navigating to services in different service categories. Finally, the console will help you distinguish whether a given service or service resource is global or Regional.

Follow the instructions on the slide. After most or all students have completed the steps document above, the educator will review the questions and answers with the whole class.

# Activity answer key



- **Question #1:** Under which service category does the **IAM** service appear?
  - **Answer:** **Security, Identity, & Compliance.**
- **Question #2:** Under which service category does the **Amazon VPC** service appear?
  - **Answer:** **Networking & Content Delivery**
- **Question #3:** Does the subnet that you selected exist at the level of the Region or the level of the Availability Zone?
  - **Answer:** Subnets exist at the **level of the Availability Zone**.
- **Question #4:** Does the VPC exist at the level of the Region or the level of the Availability Zone?
  - **Answer:** VPCs exist at the **Region level**.
- **Question #5:** Which of the following services are global instead of Regional? Check Amazon EC2, IAM, Lambda, and Route 53.
  - **Answer:** **IAM and Route 53 are global.** Amazon EC2 and Lambda are Regional.

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

26

This slide provides an answer key to the questions that were asked in the activity on the previous slide. The educator will use this slide to lead a discussion and debrief the hands-on activity.

Module 3: AWS Global Infrastructure Overview

## Module wrap-up

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



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:

- Identify the difference between AWS Regions, Availability Zones, and edge locations
- Identify AWS service and service categories

In summary, in this module you learned how to:

- Identify the difference between AWS Regions, Availability Zones, and edge locations
- Identify AWS service and service categories

# Complete the knowledge check



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

29

It is now time to complete the knowledge check for this module.

## Sample exam question



Which component of AWS global infrastructure does Amazon CloudFront use to ensure low-latency delivery?

- A. AWS Regions
- B. AWS edge locations
- C. AWS Availability Zones
- D. Amazon Virtual Private Cloud (Amazon VPC)

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](#). To learn more about the AWS Certified Cloud Practitioner exam, see: <https://aws.amazon.com/certification/certified-cloud-practitioner/>

## Additional resources



- [AWS Global Infrastructure](#)
- [AWS Regional Services List](#)
- [AWS Cloud Products](#)

The following resources provide more detail on the topics discussed in this module:

- [AWS Global Infrastructure](#) (<https://aws.amazon.com/about-aws/global-infrastructure/>)
- [AWS Regional Services List](#) (<https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/>)
- [AWS Cloud Products](#) (<https://aws.amazon.com/products/>)

# Thank you

© 2019 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections or feedback on the course, please email us at: [aws-course-feedback@amazon.com](mailto:aws-course-feedback@amazon.com). For all other questions, contact us at: <https://aws.amazon.com/contact-us/aws-training/>. All trademarks are the property of their owners.



Thank you for completing this module.

AWS Academy Cloud Foundations

# Module 4: AWS Cloud Security

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



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 use to meet their security objectives.

# Module overview



## 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

## Activities

- AWS shared responsibility model activity

## Demo

- Recorded demonstration of IAM

## Lab

- Introduction to AWS IAM



## Knowledge check

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

2

This 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.

## Module objectives



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

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

**Module 4: AWS Cloud Security**

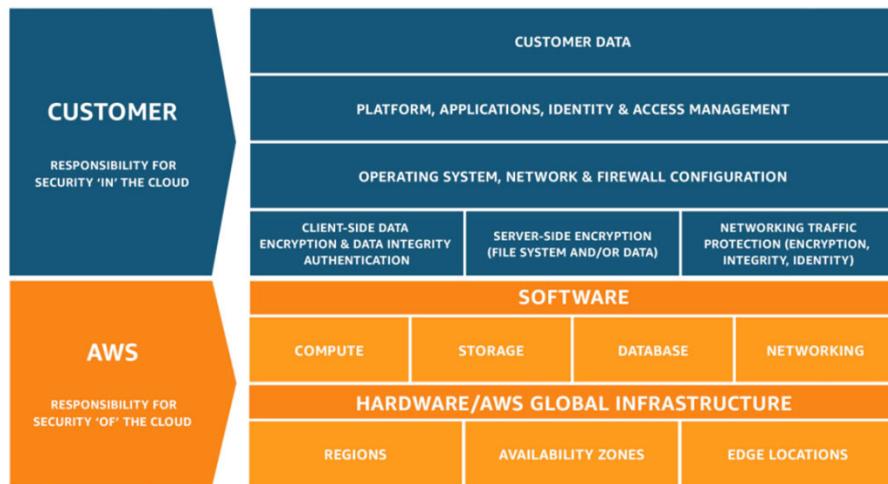
## Section 1: AWS shared responsibility model

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



Introducing Section 1: AWS shared responsibility model.

# AWS shared responsibility model



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

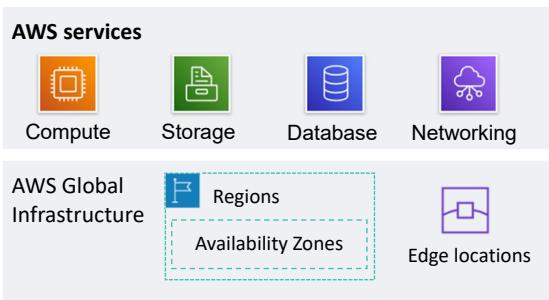
5

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).

# 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



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

6

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.

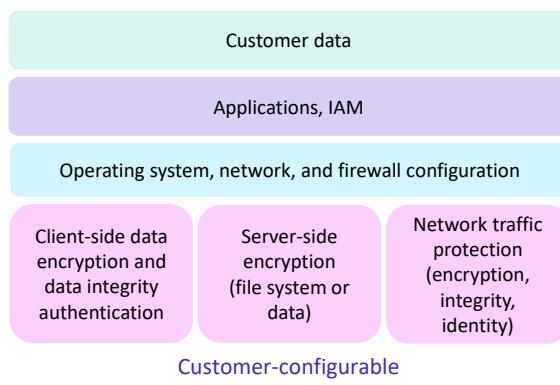
AWS is responsible for the physical infrastructure that hosts your resources, including:

- **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.

Protecting 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 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

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

7

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 content security requirements, including:

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

Customers retain control of what security they choose to implement to protect their own

data, environment, applications, IAM configurations, and operating systems.

# Service characteristics and security responsibility



## Example services managed by the customer



Amazon  
EC2



Amazon  
Elastic  
Block Store  
(Amazon EBS)



Amazon  
Virtual Private Cloud  
(Amazon VPC)

## Example services managed by AWS



AWS  
Lambda



Amazon  
Relational Database  
Service (Amazon RDS)



AWS Elastic  
Beanstalk

## 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

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

8

**Infrastructure 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 services 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)



## SaaS examples



AWS Trusted Advisor



AWS Shield



Amazon Chime

## 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 (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.

**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.

## Activity: AWS shared responsibility model



Photo by Pixabay from Pexels.

10

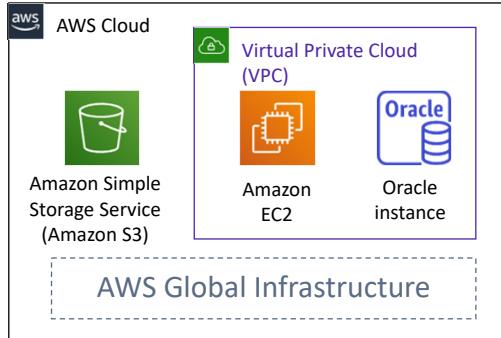
© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

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.

## 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**

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

11

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.

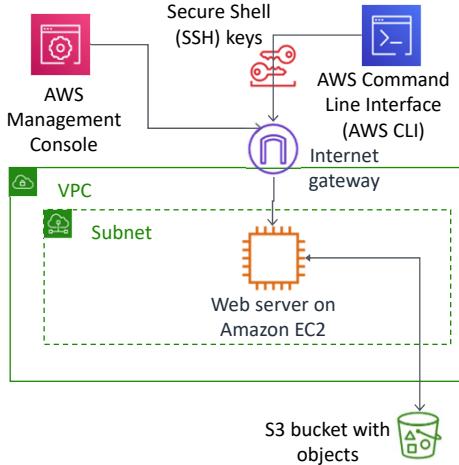
The Oracle instance in this example provides an interesting case study in terms of AWS or customer responsibility. **If the database runs on an EC2 instance**, then it is the customer's

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.

## 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

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

12

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.

## Section 1 key takeaways



13



- 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

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

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)

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



Introducing Section 2: AWS Identity and Access Management (or IAM).

# 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)

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

15

**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.

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 CLI



AWS Tools  
and SDKs

## 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.



AWS Management  
Console

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

17

**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.

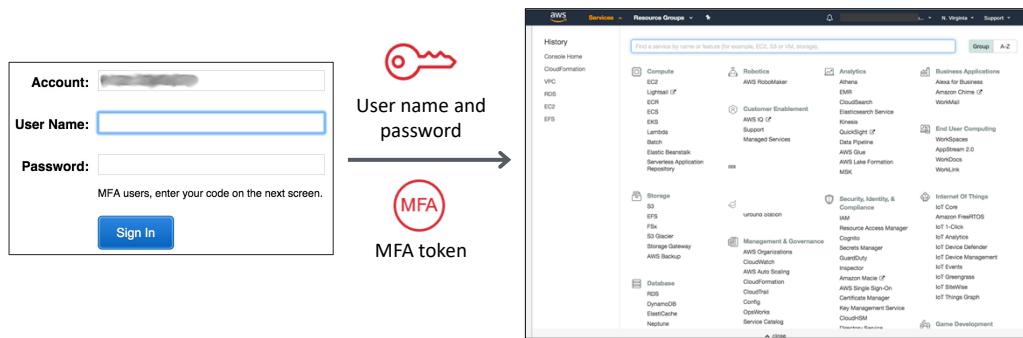
If you grant **programmatic access**, the IAM user will be required to present an **access key ID** and a **secret access key** when they make an AWS API call by using the AWS CLI, the AWS SDK, or some other development tool.

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.

# IAM MFA



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



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

18

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.

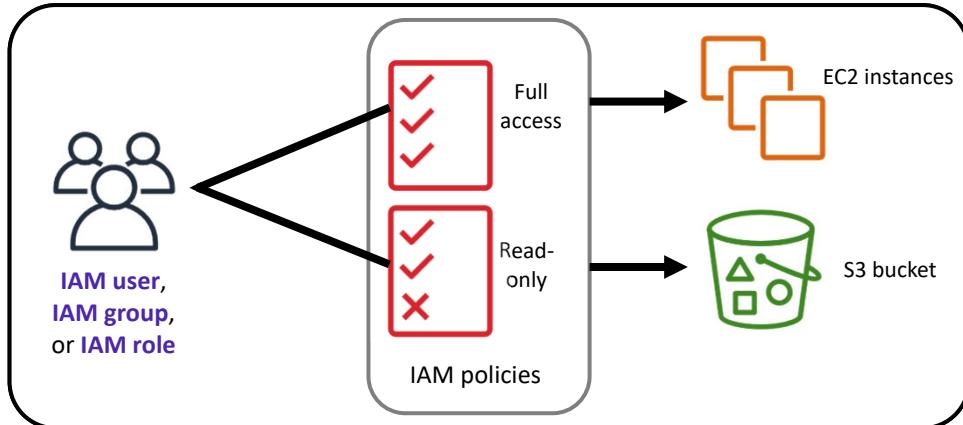
With MFA, users and systems must provide an **MFA token**—in addition to the regular sign-in credentials—before they can access AWS services and resources.

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.

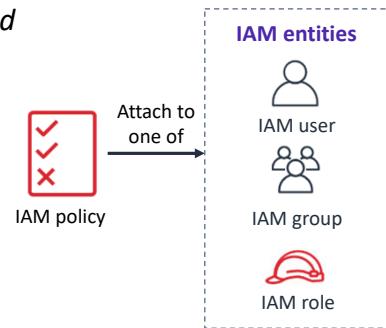
The **principle of least privilege** is an important concept in computer security. It promotes that you grant only the minimal user privileges needed to the user, based on the needs of your users. When you create IAM policies, it is a best practice to follow this security advice of granting *least privilege*. 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)



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

21

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.

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

22

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

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

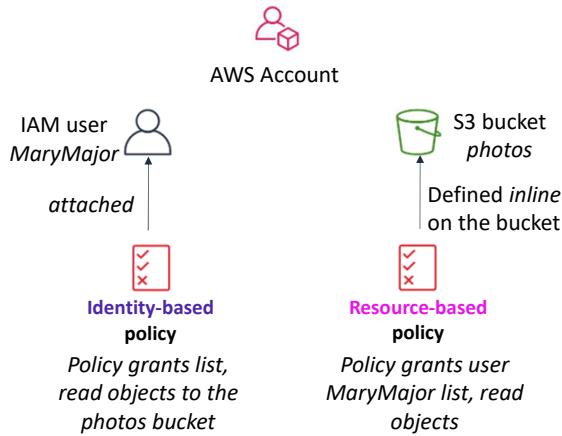
- The DynamoDB table whose name is represented by *table-name*.
- The AWS account's S3 bucket, whose name is represented by *bucket-name* and all the objects that it contains.

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



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

23

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.

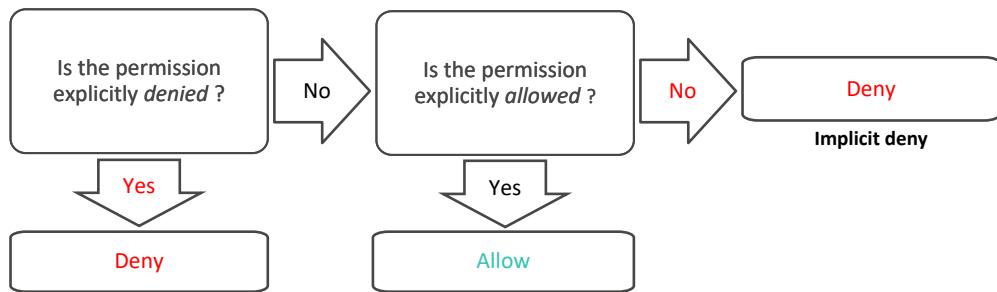
The diagram shows two different ways that the user *MaryMajor* could be granted access to objects in the S3 bucket that is named *photos*. On the left, you see an example of an identity-based policy. An IAM policy that grants access to the S3 bucket is attached to the *MaryMajor* user. On the right,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

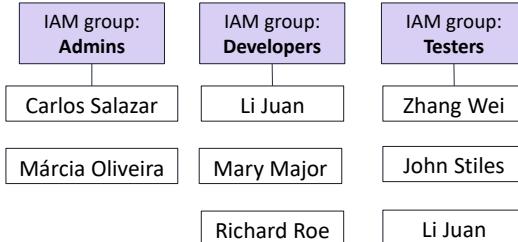
When IAM determines whether a permission is allowed, IAM first checks for the existence of any applicable **explicit denial policy**. If no explicit denial exists, it then checks for any applicable **explicit allow policy**. If neither an explicit deny nor an explicit allow policy exists, IAM reverts 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



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

25

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 user 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.

- 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.

# Example use of an IAM role

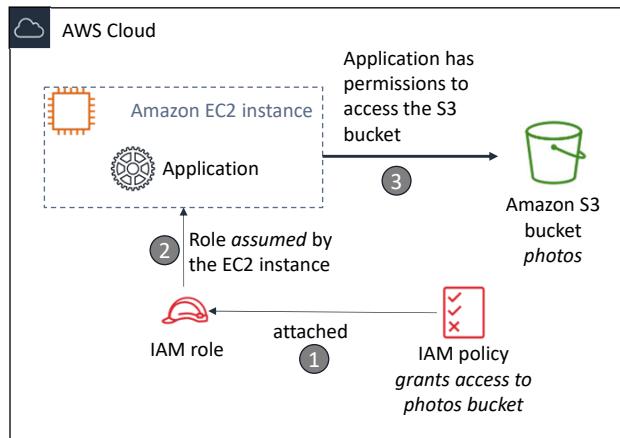


## Scenario:

- 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



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

27

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



28

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

- **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:

- **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)



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

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

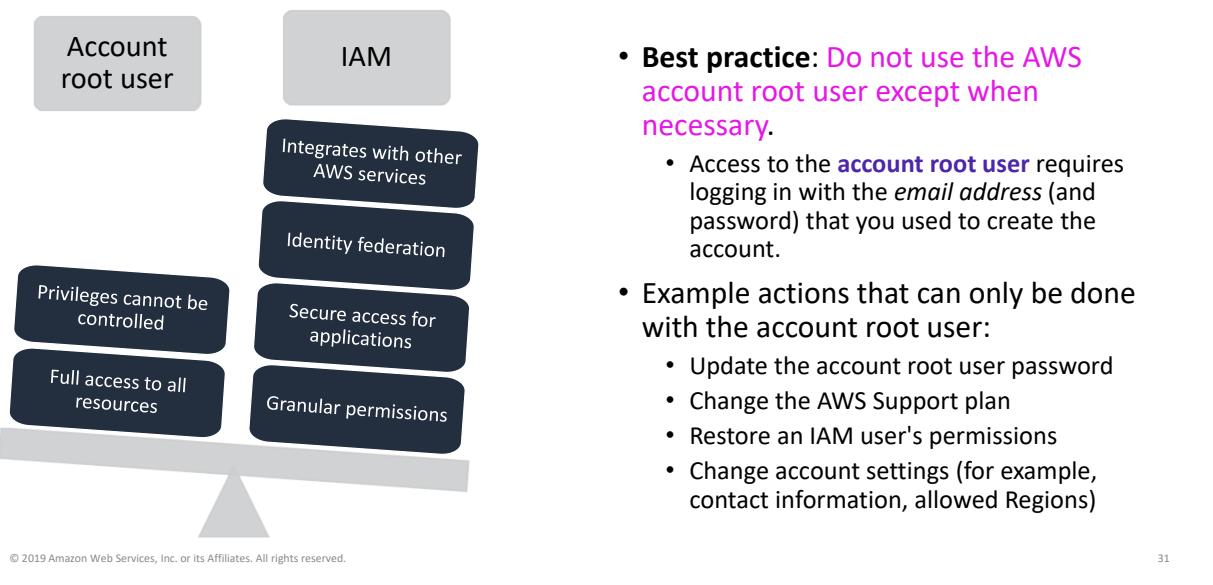
## Section 3: Securing a new AWS account

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



Introducing Section 3: Securing a new AWS account.

# AWS account root user access versus IAM access



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

31

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.

## 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:

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](#).



MFA token

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

33

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 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.

### 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, 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](#).

## Module 4: AWS Cloud Security

### Optional: Securing a new AWS account – Full walkthrough

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



The educator might optionally choose to show a full walkthrough of the first two major steps that you must complete to secure a new AWS account. (These steps were described in the previous slides.) The slides in this section provide screen captures of what it looks like to go through the process in detail.

# IAM security status review

aws academy

The screenshot shows the AWS IAM Console Dashboard. On the left, there's a sidebar with links like 'Custom Sign In Link' (which is highlighted with a red arrow), 'Dashboard', 'Groups', 'Users', 'Roles', 'Policies', 'Identity providers', 'Account settings', 'Credential report', and 'Encryption keys'. The main area has a heading 'Welcome to Identity and Access Management' and a sub-section 'IAM users sign-in link:' with a URL 'https://.signin.aws.amazon.com/console' (also highlighted with a red box). Below that is the 'IAM Resources' section with counts for 'Users: 0', 'Groups: 0', 'Roles: 0', and 'Identity Providers: 0'. To the right is the 'Customer Managed Policies: 0' section. At the bottom is the 'Security Status' panel, which displays five items: 'Delete your root access keys' (completed, checked), 'Activate MFA on your root account' (not completed, warning icon), 'Create individual IAM users' (not completed, warning icon), 'Use groups to assign permissions' (not completed, warning icon), and 'Apply an IAM password policy' (not completed, warning icon). A progress bar indicates '1 out of 5 complete.'

The screen capture shows an example of what the IAM Console Dashboard looks like when you are logged in as the AWS account root user. To access this screen in an account:

1. Log in to the **AWS Management Console** as the AWS account root user.
2. Go to the **IAM** service page and click the **Dashboard** link.
3. Review the information in the **Security Status** panel.

In the screen capture, only one of the five security status checks has been completed (*Delete your root access keys*). The goal of a person who completes the steps to secure the account is to receive green checks next to each security status item.

A review of the current **Security Status** list indicates that:

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

There is a custom IAM user sign-in link for the account. Note that the account number was hidden in this screen capture. Optionally, you can use the **Customize** link to the right of the IAM user sign-in link to change the name of the account so that it does not display the account number. This link is used to sign in to the account, and it can be sent to users after

their accounts are created.

# Activate MFA on the account root user

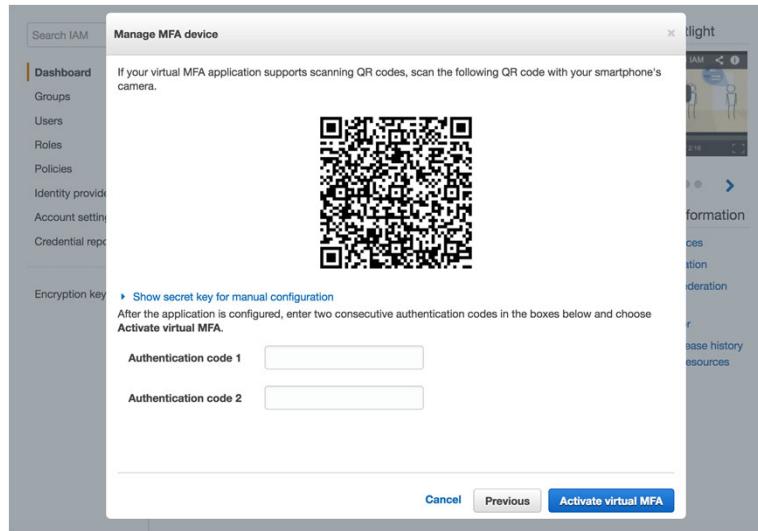
The screenshot shows the AWS Identity and Access Management (IAM) dashboard. On the left, there's a sidebar with options like 'Custom sign-in link' (which is highlighted with a red box), 'Groups', 'Users', 'Roles', 'Policies', 'Identity providers', 'Account settings', 'Credential report', and 'Encryption keys'. Below that is 'MFA activation'. The main area has a heading 'Welcome to Identity and Access Management'. It shows an 'IAM users sign-in link' (highlighted with a red box) which is <https://.signin.aws.amazon.com/console>. There are also sections for 'IAM Resources' (Users: 0, Groups: 0, Roles: 0, Identity Providers: 0, Customer Managed Policies: 0) and 'Security Status' (1 out of 5 complete). A list of tasks includes: 'Delete your root access keys' (with a checked checkbox), 'Activate MFA on your root account' (highlighted with a red box), 'Create individual IAM users', 'Use groups to assign permissions', and 'Apply an IAM password policy'. The bottom right corner of the screenshot says '38'.

Before you create IAM users in the account, activate MFA on the account root user. To log in as the account root user, use the email address that you used to create the account. The account root user has access to everything, which is why it is important to secure this account with restrictions.

To configure MFA:

1. Click the **Activate MFA on your root account** link.
2. Click **Manage MFA**.
3. Click **Assign MFA device**. You have three options: **Virtual MFA device**, **U2F security key**, and **Other hardware MFA device**. A hardware device is an actual hardware device.
4. For purposes of this demonstration, select **Virtual MFA device** and then click **Continue**.
5. A new dialog box appears and asks you to configure a virtual MFA device. An app (such as Google Authenticator) must be downloaded for this task. After the download is complete, click **Show QR code**.

## Activate MFA on account root user



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

39

6. In the authenticator application, choose the **plus sign (+)**.
7. Scan the barcode, and enter the first authentication code.
8. Wait a moment for the second code to display, and enter the second code.
9. Click the **Assign MFA** button.

# MFA on account root user is activated



The screenshot shows the AWS IAM Dashboard. On the left, there's a sidebar with options like Groups, Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. Below these, a note says "MFA activated". The main area has a title "Welcome to Identity and Access Management" and a sub-section "IAM Resources" showing 0 users, 0 roles, 0 groups, and 0 identity providers. Below that is a "Security Status" section with a progress bar at "2 out of 5 complete". It lists five items: "Delete your root access keys" (green checkmark), "Activate MFA on your root account" (green checkmark, highlighted with a red box), "Create individual IAM users" (yellow warning icon), "Use groups to assign permissions" (yellow warning icon), and "Apply an IAM password policy" (yellow warning icon). At the bottom, it says "© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved." and "40".

10. Click **Finish** and refresh your browser.

In the **Security Status** panel, it should now show a green checkmark icon, which indicates that MFA is now activated on the account root user.

# Create an individual IAM user (1)



The screenshot shows the AWS Identity and Access Management (IAM) dashboard. On the left, there's a sidebar with options like 'Dashboard', 'Groups', 'Users', 'Roles', 'Policies', 'Identity providers', 'Account settings', 'Credential report', and 'Encryption keys'. Below this, a section labeled 'IAM user creation' is highlighted with a red arrow pointing towards the main content area. The main content area has a heading 'Welcome to Identity and Access Management' and a sub-section 'IAM Resources' showing 'Users: 0', 'Groups: 0', and 'Customer Managed Policies: 0'. Below this is a 'Security Status' section with a progress bar showing '2 out of 5 complete.' and five items: 'Delete your root access keys' (checked), 'Activate MFA on your root account' (checked), 'Create individual IAM users' (highlighted with a red box), 'Use groups to assign permissions' (warning icon), and 'Apply an IAM password policy' (warning icon). At the bottom, there's a copyright notice '© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.' and a page number '41'.

Most AWS accounts are shared by multiple users in an organization. To support this practice, you can set up each user with individually assigned permissions, or you can add users to the appropriate IAM group that grants them specific permissions.

An AWS best practice is to provide each user with their own IAM user login so that they do not log in as the account root user with global privileges, or use the same credentials as someone else to log in to the account.

To configure this setup:

1. Click **Create individual IAM users** and then select **Manage Users**.

## Create an individual IAM user (2)



Add user

1 Details    2 Permissions    3 Review    4 Complete

**Set user details**

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\*  [Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type\*  **Programmatic access**  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

**AWS Management Console access**  
Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password\*  Autogenerated password  
 Custom password

Require password reset  User must create a new password at next sign-in  
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

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

42

2. Select **Add user** and specify a new user name. Note that user names cannot have spaces.
3. Select the **Access type**. There are two access types (you can grant either type or both types to the user, but for the purposes of this demonstration, grant both types):
  - **Programmatic access** enables the user to have AWS CLI access to provision resources. This option will generate an access key one time. This access key must be saved because it will be used for all future access.
  - **AWS Management Console access** enables the user to log in to the console.
4. If you chose to grant console access, either choose **Autogenerate password**, or select **Custom password** and enter one.
5. Click **Next: Permissions**.

# Create an individual IAM user (3)



Add user

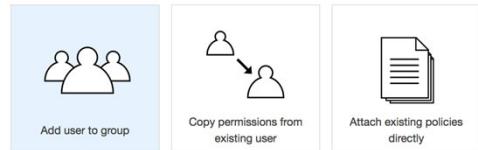
1  
Details

2  
Permissions

3  
Review

4  
Complete

Set permissions for M



**i** Get started with groups

You haven't created any groups yet. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. Get started by creating a group. [Learn more](#)

[Create group](#)

[Cancel](#) [Previous](#) **Next: Review**

43

Next, you will assign permissions. You have three options for assigning permissions:

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

6. You want to add the user to a group, so select **Add user to group** and then choose **Create group**.

Note: A group is where you put users to inherit the policies that are assigned to the group.

# Create an individual IAM user (4)

aws academy

Create group

Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. [Learn more](#)

Group name: Administrators

Create policy Refresh

Filter: Policy type ▾ Search Showing 313 results

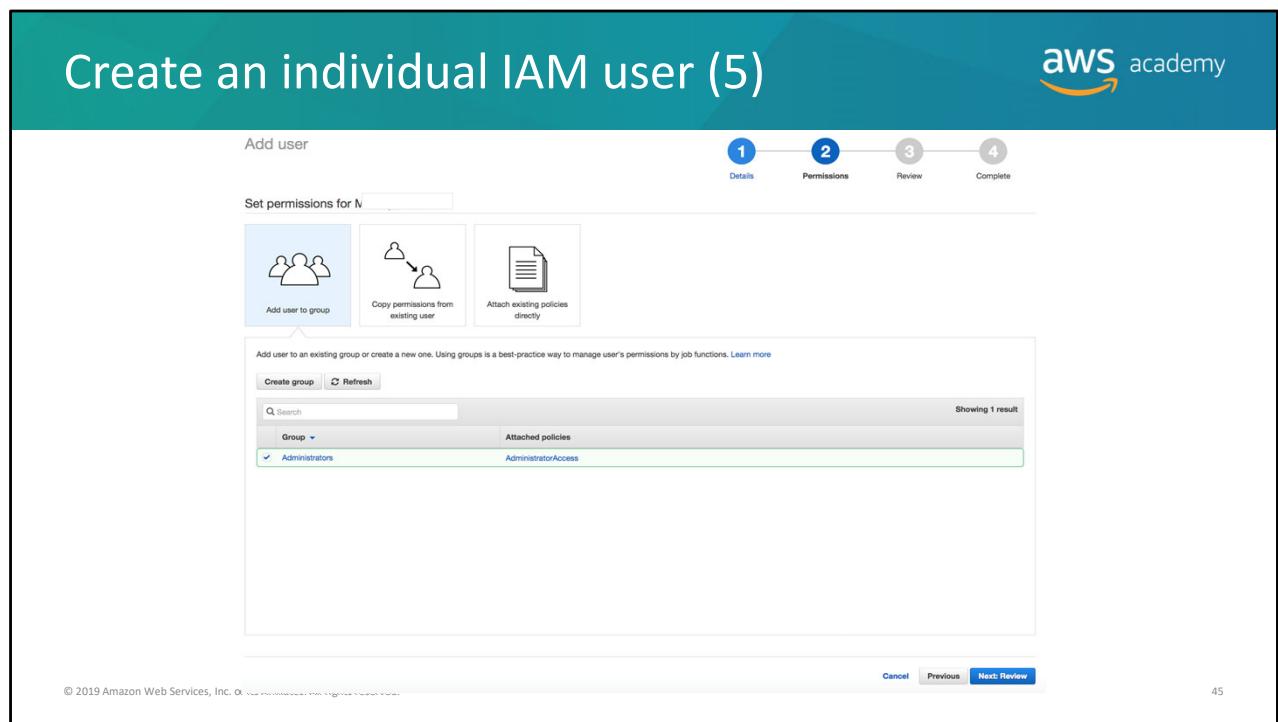
Policy name	Type	Attachments	Description
<input checked="" type="checkbox"/> AdministratorAccess	Job function	0	Provides full access to AWS services and resources.
<input type="checkbox"/> AlexaForBusinessDeviceSetup	AWS managed	0	Provide device setup access to AlexaForBusiness services
<input type="checkbox"/> AlexaForBusinessFullAccess	AWS managed	0	Grants full access to AlexaForBusiness resources and access to relat...
<input type="checkbox"/> AlexaForBusinessGatewayEx...	AWS managed	0	Provide gateway execution access to AlexaForBusiness services
<input type="checkbox"/> AlexaForBusinessReadOnlyA...	AWS managed	0	Provide read only access to AlexaForBusiness services
<input type="checkbox"/> AmazonAPIGatewayAdminist...	AWS managed	0	Provides full access to create/edit/delete APIs in Amazon API Gatew...
<input type="checkbox"/> AmazonAPIGatewayInvokeFu...	AWS managed	0	Provides full access to invoke APIs in Amazon API Gateway.
<input type="checkbox"/> AmazonAPIGatewayPushToC...	AWS managed	0	Allows API Gateway to push logs to user's account.
<input type="checkbox"/> AmazonAppStreamFullAccess	AWS managed	0	Provides full access to Amazon AppStream via the AWS Managemen...
<input type="checkbox"/> AmazonAnnStreamReadOnly...	AWS managed	0	Provides read only access to Amazon AnnStream via the AWS Mana...

Cancel Create group

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

44

7. Give the group a name. In this example, give the lead developer administrative access and then choose **Create group**.



8. Select **Next Review** to review what will be created, and then choose **Create user**.

IAM user creation successful

Add user

1 Details    2 Permissions    3 Review    4 Complete

**Success**  
You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://raysinut.sigin.aws.amazon.com/console>

[Download .csv](#)

User	Access key ID	Secret access key	Password	Email login instructions
Mi	AKI.....	***** Show	***** Show	<a href="#">Send email</a>

[Close](#)

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

46

When a user is created—and assuming you enabled both programmatic and console access when you defined the **Access type** setting and created the user—several artifacts will be generated:

1. An **access key ID** that can be used to sign AWS API calls when the user uses the AWS CLI or AWS SDKs.
2. A **secret access key** that is also used to sign AWS API calls when the user uses the AWS CLI or AWS SDKs.
3. A **password** that can be used to log in to the AWS Management Console.

Choose **Show** to display the values in each field. The credentials can also be downloaded by choosing **Download .csv**. This time is the only time when you have the option to download these credentials. You will not have an opportunity to retrieve the secret access key after this screen. Thus, you should either download the credentials, or—at the minimum—copy the secret access key, and paste it in a safe location.

**Important:** Never store these credentials in a public place (for example, never embed these credentials in code that you upload to GitHub or elsewhere). This information can be used to access your account. If you ever have a concern that your credentials have been compromised, log in as a user with IAM administrator access permissions and delete the existing access key. You can then optionally create a new access key.

# IAM Dashboard security status

aws academy

Welcome to Identity and Access Management

IAM users sign-in link:  
<https://raysiaut.signin.aws.amazon.com/console>

Customize | Copy Link

**IAM Resources**

Users: 1 Roles: 0  
Groups: 1 Identity Providers: 0  
Customer Managed Policies: 0

**Security Status** 4 out of 5 complete.

<input checked="" type="checkbox"/> Delete your root access keys	▼
<input checked="" type="checkbox"/> Activate MFA on your root account	▼
<input checked="" type="checkbox"/> Create individual IAM users	▼
<input checked="" type="checkbox"/> Use groups to assign permissions	▼
<input checked="" type="checkbox"/> Apply an IAM password policy	▼

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 47

When you return to the IAM Dashboard, the **Create individual IAM users** and **Use groups to assign permissions** security status items should show that they were addressed.

The remaining security item to address is to apply an IAM password policy.

# Set an IAM password policy



The screenshot shows the AWS IAM Password Policy configuration interface. On the left, a sidebar lists navigation options: Search IAM, Dashboard, Groups, Users, Roles, Policies, Identity providers, Account settings (which is selected), Credential report, and Encryption keys. The main area is titled 'Password Policy' and contains a message: 'You have unsaved changes to your password policy.' Below this, a note states: 'A password policy is a set of rules that define the type of password an IAM user can set. For more information about password policies, go to [Managing Passwords](#) in Using IAM.' A sub-note says: 'Currently, this AWS account does not have a password policy. Specify a password policy below.' A 'Minimum password length:' field is set to 6. A list of password requirements includes checked boxes for uppercase, lowercase, numbers, and non-alphanumeric characters, along with other options like changing passwords and preventing reuse. Buttons at the bottom are 'Apply password policy' (blue) and 'Delete password policy' (red).

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

48

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

Select the rules that the passwords should comply with and then choose **Apply password policy**.

# Security status checks completed



Search IAM

Dashboard

Groups

Users

Roles

Policies

Identity providers

Account settings

Credential report

Encryption keys

Welcome to Identity and Access Management

IAM users sign-in link:  
<https://raysiaut.signin.aws.amazon.com/console>

Customize | Copy Link

IAM Resources

Users: 1 Roles: 0

Groups: 1 Identity Providers: 0

Customer Managed Policies: 0

Security Status

5 out of 5 complete.

<input checked="" type="checkbox"/> Delete your root access keys	▼
<input checked="" type="checkbox"/> Activate MFA on your root account	▼
<input checked="" type="checkbox"/> Create individual IAM users	▼
<input checked="" type="checkbox"/> Use groups to assign permissions	▼
<input checked="" type="checkbox"/> Apply an IAM password policy	▼

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

49

All the security status checkmarks should now be green. Your account is now in compliance with the listed IAM security status checks. Congratulations!

## Section 3 key takeaways



50

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.

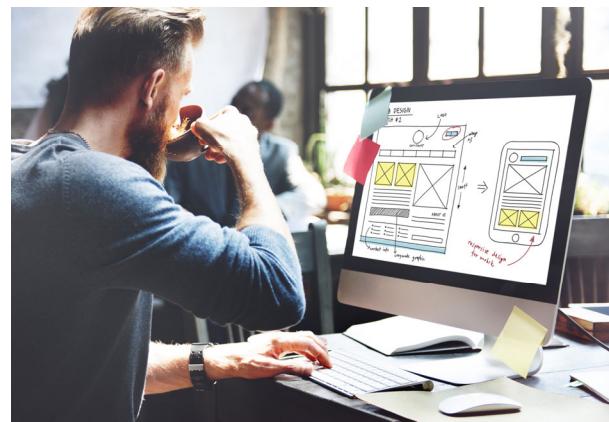
© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

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

S1



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

Introducing Lab 1: Introduction to AWS IAM.

## Lab 1: Tasks



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



AWS Identity and Access  
Management (IAM)

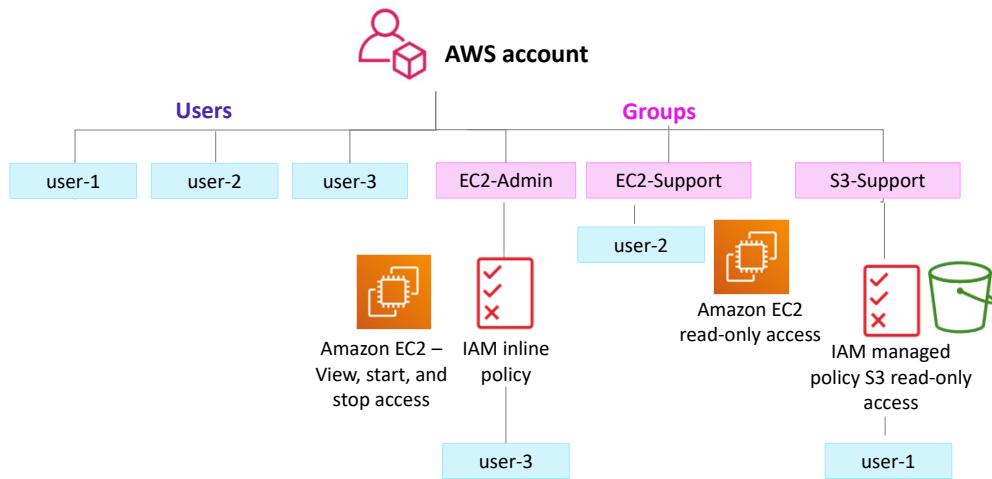
© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

52

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



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

53

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



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

54

It is now time to start the lab.



## Lab debrief: Key takeaways

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

55

The instructor will now lead a conversation about the key takeaways from the lab after you complete it.

Module 4: AWS Cloud Security

## Section 4: Securing accounts

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



Introducing Section 4: Securing accounts

# AWS Organizations



- **AWS Organizations** enables you to consolidate multiple AWS accounts so that you centrally manage them.



AWS Organizations

- **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 account 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.

- **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) 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)

**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 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

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

60

Amazon Cognito provides solutions to control access to AWS resources from your application. You can define roles and map users to different roles 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** 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

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

61

**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 protect 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.

Module 4: AWS Cloud Security

## Section 5: Securing data on AWS

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



Introducing 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



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

63

**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.

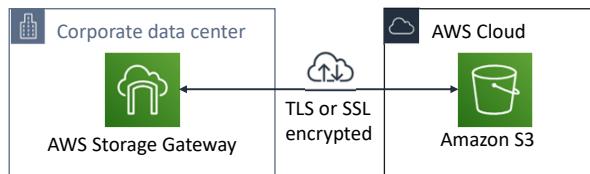
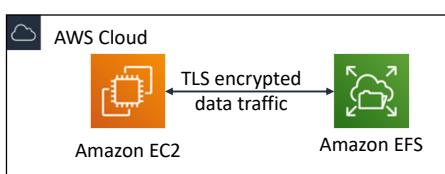
**Data at rest** refers to data that is physically stored on disk or on tape.

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:



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

64

**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 private networks. With AWS Certificate Manager, you can request a certificate and then deploy 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.

- 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.

- 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

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



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



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

67

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.

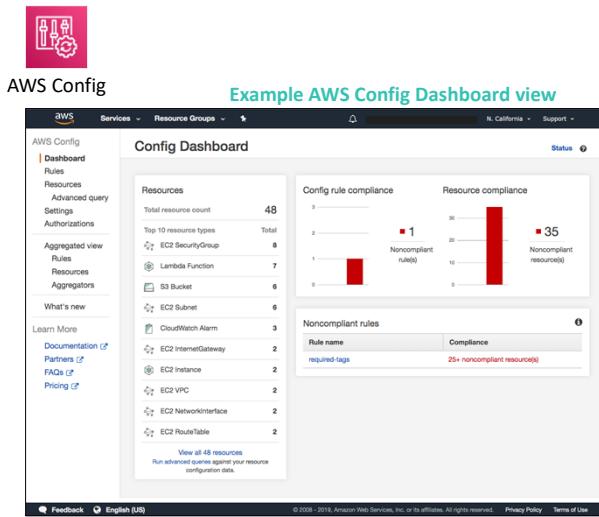
A full [Listing of AWS Compliance Programs](#) is available. Also, for details about which AWS services are in scope of AWS assurance programs, see [AWS Services in Scope by Compliance Program](#).

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



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

68

- **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



70

- **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.

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

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

## Section 7: Additional security services and resources

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



Introducing Section 7: Additional security services and resources



AWS Service Catalog

- Create and manage catalogs of IT services that are approved by your organization
  - Helps employees find and deploy *approved* IT services
  - An IT service can include one or more AWS resources
  - Example:
    - EC2 instances, storage volumes, databases, and networking components
- Control AWS service usage by specifying constraints –
  - Example constraints:
    - The AWS Region where a product can be launched
    - Allowed IP address ranges
- Centrally manage the IT service lifecycle
- Help meet compliance requirements

**AWS Service Catalog** enables organizations to create and manage catalogs of IT services that are approved for use (for example, for your employees to use) on AWS. These IT services can include everything from virtual machine images, servers, software, and databases to complete multi-tier application architectures.

From the perspective of AWS Service Catalog, an IT service can be thought of as a product. A product could be a single compute instance that runs Amazon Linux, it could be a fully configured multi-tier web application that runs in its own environment, or it could be any other useful IT service that you build on AWS. This enables your users to quickly deploy the IT services that they need, and it is designed so that users deploy only approved configurations. AWS Service Catalog can support your efforts to centrally manage deployed IT services, and it can help you achieve consistent governance and meet compliance requirements.

For more information, see [AWS Service Catalog](#) in the AWS documentation.

## Selected additional security services



Amazon  
Macie

Proactively **protect personally identifiable information (PII)** and know when it moves.



Amazon  
Inspector

Define standards and best practices for your applications and **validate adherence to these standards**.



Amazon  
GuardDuty

Intelligent **threat detection** and continuous monitoring to protect your AWS accounts and workloads.

**Amazon Macie** is a security service that uses machine learning to automatically discover, classify, and protect sensitive data in AWS. Amazon Macie recognizes sensitive data such as personally identifiable information (PII) or intellectual property. It provides you with dashboards and alerts that give visibility into how this data is being accessed or moved. Amazon Macie is a fully managed service that continuously monitors data access activity for anomalies, and it generates detailed alerts when it detects risk of unauthorized access or inadvertent data leaks. Amazon Macie is currently available to protect data that is stored in Amazon S3.

**Amazon Inspector** is an automated security assessment service that helps improve the security and compliance of applications that are deployed on AWS. Amazon Inspector automatically assesses applications for exposure, vulnerabilities, and deviations from best practices. After performing an assessment, Amazon Inspector produces a detailed list of security findings that are listed by level of severity. These findings can be reviewed directly or as part of detailed assessment reports that are available via the Amazon Inspector console or the API.

**Amazon GuardDuty** is a threat-detection service that continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts and workloads. With the

cloud, the collection and aggregation of account and network activities is simplified, but it can be time-consuming for security teams to continuously analyze event log data for potential threats. GuardDuty uses machine learning, anomaly detection, and integrated threat intelligence to identify and rank potential threats. GuardDuty analyzes tens of billions of events across multiple AWS data sources, such as AWS CloudTrail, Amazon VPC Flow Logs, and Domain Name System (DNS) logs.

Module 4: AWS Cloud Security

## Module wrap-up

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



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

## Complete the knowledge check



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

76

It is now time to complete the knowledge check for this module.

## 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](#).

# Thank you

© 2019 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections or feedback on the course, please email us at: [aws-course-feedback@amazon.com](mailto:aws-course-feedback@amazon.com). For all other questions, contact us at: <https://aws.amazon.com/contact-us/aws-training/>. All trademarks are the property of their owners.



Thank you for completing this module.

AWS Academy Cloud Foundations

# Module 5: Networking and Content Delivery

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



Welcome to Module 5: Networking and Content Delivery

This module covers three fundamental Amazon Web Services (AWS) for networking and content delivery: Amazon Virtual Private Cloud (Amazon VPC), Amazon Route 53, and Amazon CloudFront.

# Module overview



## Topics

- Networking basics
- Amazon VPC
- VPC networking
- VPC security
- Amazon Route 53
- Amazon CloudFront

## Activities

- Label a network diagram
- Design a basic VPC architecture

## Demo

- VPC demonstration

## Lab

- Build your VPC and launch a web server



## Knowledge check

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

2

This module addresses the following topics:

- Networking basics
- Amazon Virtual Private Cloud (Amazon VPC)
- VPC networking
- VPC security
- Amazon Route 53
- Amazon CloudFront

This module includes some activities that challenge you to label a network diagram and design a basic VPC architecture.

You will watch a recorded demonstration to learn how to use the VPC Wizard to create a VPC with public and private subnets.

You then get a chance to apply what you have learned in a hands-on lab where you use the VPC Wizard to build a VPC and launch a web server.

Finally, you will be asked to complete a knowledge check that test your understanding of key concepts that are covered in this module.

## Module objectives



After completing this module, you should be able to:

- Recognize the basics of networking
- Describe virtual networking in the cloud with Amazon VPC
- Label a network diagram
- Design a basic VPC architecture
- Indicate the steps to build a VPC
- Identify security groups
- Create your own VPC and add additional components to it to produce a customized network
- Identify the fundamentals of Amazon Route 53
- Recognize the benefits of Amazon CloudFront

After completing this module, you should be able to:

- Recognize the basics of networking
- Describe virtual networking in the cloud with Amazon VPC
- Label a network diagram
- Design a basic VPC architecture
- Indicate the steps to build a VPC
- Identify security groups
- Create your own VPC and add additional components to it to produce a customized network
- Identify the fundamentals of Amazon Route 53
- Recognize the benefits of Amazon CloudFront

Module 5: Networking and Content Delivery

## Section 1: Networking basics

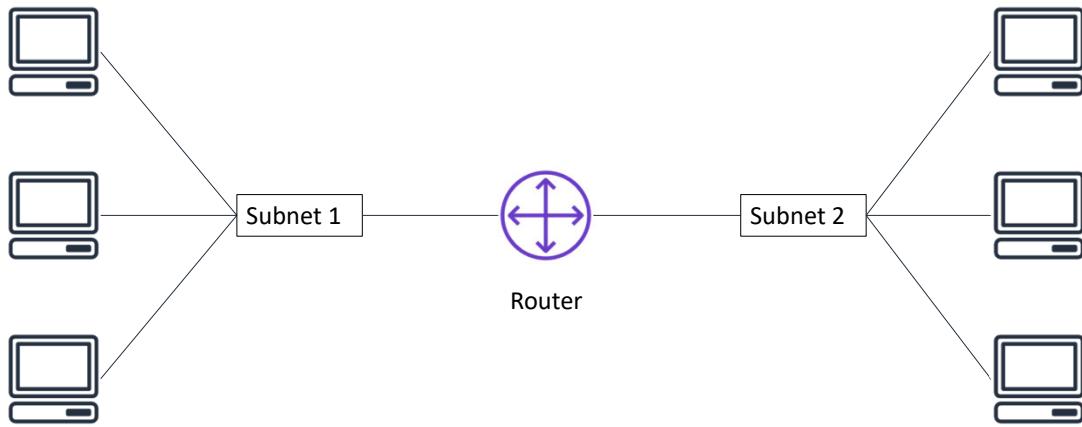
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



### Section 1: Networking basics

In this section, you will review a few basic networking concepts that provide the necessary foundation to your understanding of the AWS networking service, Amazon Virtual Private Cloud (Amazon VPC).

# Networks



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

5

A computer *network* is two or more client machines that are connected together to share resources. A network can be logically partitioned into *subnets*. Networking requires a networking device (such as a router or switch) to connect all the clients together and enable communication between them.

# IP addresses



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

6

Each client machine in a network has a unique Internet Protocol (IP) address that identifies it. An IP address is a numerical label in decimal format. Machines convert that decimal number to a binary format.

In this example, the IP address is 192.0.2.0. Each of the four dot (.)-separated numbers of the IP address represents 8 bits in octal number format. That means each of the four numbers can be anything from 0 to 255. The combined total of the four numbers for an IP address is 32 bits in binary format.

## IPv4 and IPv6 addresses



**IPv4 (32-bit) address:** 192.0.2.0

**IPv6 (128-bit) address:** 2600:1f18:22ba:8c00:ba86:a05e:a5ba:00FF

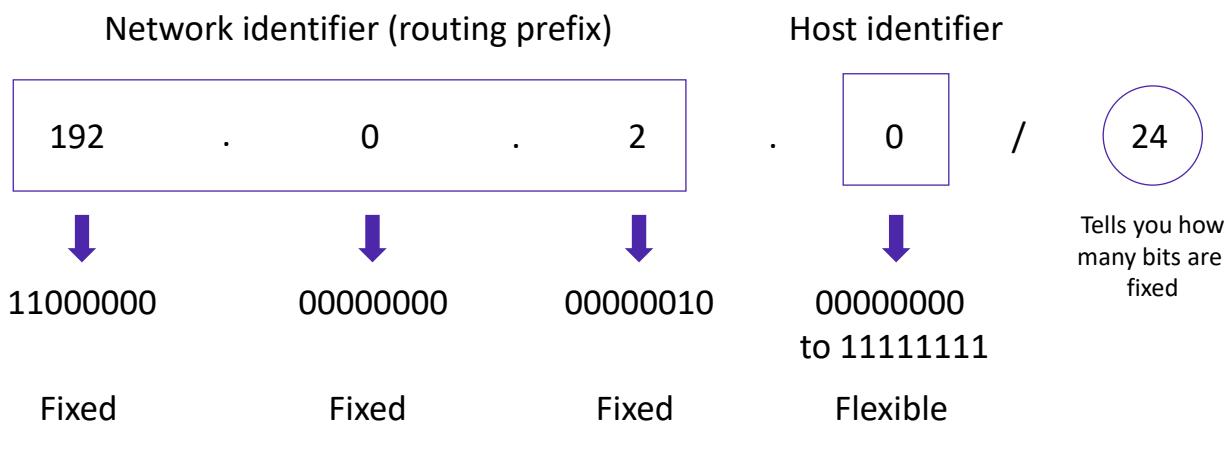
A 32-bit IP address is called an IPv4 address.

IPv6 addresses, which are 128 bits, are also available. IPv6 addresses can accommodate more user devices.

An IPv6 address is composed of eight groups of four letters and numbers that are separated by colons (:). In this example, the IPv6 address is

2600:1f18:22ba:8c00:ba86:a05e:a5ba:00FF. Each of the eight colon-separated groups of the IPv6 address represents 16 bits in hexadecimal number format. That means each of the eight groups can be anything from 0 to FFFF. The combined total of the eight groups for an IPv6 address is 128 bits in binary format.

# Classless Inter-Domain Routing (CIDR)



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

8

A common method to describe networks is Classless Inter-Domain Routing (CIDR). The CIDR address is expressed as follows:

- An IP address (which is the first address of the network)
- Next, a slash character (/)
- Finally, a number that tells you how many bits of the routing prefix must be fixed or allocated for the network identifier

The bits that are not fixed are allowed to change. CIDR is a way to express a group of IP addresses that are consecutive to each other.

In this example, the CIDR address is 192.0.2.0/24. The last number (24) tells you that the first 24 bits must be fixed. The last 8 bits are flexible, which means that  $2^8$  (or 256) IP addresses are available for the network, which range from 192.0.2.0 to 192.0.2.255. The fourth decimal digit is allowed to change from 0 to 255.

If the CIDR was 192.0.2.0/16, the last number (16) tells you that the first 16 bits must be fixed. The last 16 bits are flexible, which means that  $2^{16}$  (or 65,536) IP addresses are available for the network, ranging from 192.0.0.0 to 192.0.255.255. The third and fourth

decimal digits can each change from 0 to 255.

There are two special cases:

- Fixed IP addresses, in which every bit is fixed, represent a single IP address (for example, 192.0.2.0/32). This type of address is helpful when you want to set up a firewall rule and give access to a specific host.
- The internet, in which every bit is flexible, is represented as 0.0.0.0/0

# Open Systems Interconnection (OSI) model



Layer	Number	Function	Protocol/Address
Application	7	Means for an application to access a computer network	HTTP(S), FTP, DHCP, LDAP
Presentation	6	<ul style="list-style-type: none"><li>Ensures that the application layer can read the data</li><li>Encryption</li></ul>	ASCI, ICA
Session	5	Enables orderly exchange of data	NetBIOS, RPC
Transport	4	Provides protocols to support host-to-host communication	TCP, UDP
Network	3	Routing and packet forwarding (routers)	IP
Data link	2	Transfer data in the same LAN network (hubs and switches)	MAC
Physical	1	Transmission and reception of raw bitstreams over a physical medium	Signals (1s and 0s)

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

9

The Open Systems Interconnection (OSI) model is a conceptual model that is used to explain how data travels over a network. It consists of seven layers and shows the common protocols and addresses that are used to send data at each layer. For example, hubs and switches work at layer 2 (the data link layer). Routers work at layer 3 (the network layer). The OSI model can also be used to understand how communication takes place in a virtual private cloud (VPC), which you will learn about in the next section.

Module 5: Networking and Content Delivery

## Section 2: Amazon VPC

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



### Section 2: Amazon VPC

Many of the concepts of an on-premises network apply to a cloud-based network, but much of the complexity of setting up a network has been abstracted without sacrificing control, security, and usability. In this section, you learn about Amazon VPC and the fundamental components of a VPC.



Amazon  
VPC

- Enables you to provision a **logically isolated** section of the AWS Cloud where you can launch AWS resources in a virtual network that you define
- Gives you **control over your virtual networking resources**, including:
  - Selection of IP address range
  - Creation of subnets
  - Configuration of route tables and network gateways
- Enables you to **customize the network configuration** for your VPC
- Enables you to use **multiple layers of security**

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

11

Amazon Virtual Private Cloud (Amazon VPC) is a service that lets you provision a logically isolated section of the AWS Cloud (called a virtual private cloud, or VPC) where you can launch your AWS resources.

Amazon VPC gives you control over your virtual networking resources, including the selection of your own IP address range, the creation of subnets, and the configuration of route tables and network gateways. You can use both IPv4 and IPv6 in your VPC for secure access to resources and applications.

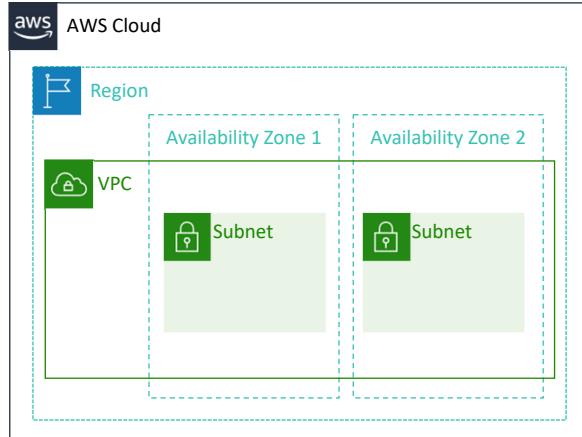
You can also customize the network configuration for your VPC. For example, you can create a public subnet for your web servers that can access the public internet. You can place your backend systems (such as databases or application servers) in a private subnet with no public internet access.

Finally, you can use multiple layers of security, including security groups and network access control lists (network ACLs), to help control access to Amazon Elastic Compute Cloud (Amazon EC2) instances in each subnet.

# VPCs and subnets



- VPCs:
  - Logically isolated from other VPCs
  - Dedicated to your AWS account
  - Belong to a single **AWS Region** and can span multiple Availability Zones
- Subnets:
  - Range of IP addresses that divide a VPC
  - Belong to a single **Availability Zone**
  - Classified as **public** or **private**



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

12

Amazon VPC enables you to provision virtual private clouds (VPCs). A *VPC* is a virtual network that is logically isolated from other virtual networks in the AWS Cloud. A VPC is dedicated to your account. VPCs belong to a single AWS Region and can span multiple Availability Zones.

After you create a VPC, you can divide it into one or more subnets. A *subnet* is a range of IP addresses in a VPC. Subnets belong to a single Availability Zone. You can create subnets in different Availability Zones for high availability. Subnets are generally classified as public or private. *Public subnets* have direct access to the internet, but *private subnets* do not.

# IP addressing



- When you create a VPC, you assign it to an IPv4 **CIDR block** (range of **private** IPv4 addresses).
- You **cannot change the address range** after you create the VPC.
- The **largest** IPv4 CIDR block size is **/16**.
- The **smallest** IPv4 CIDR block size is **/28**.
- IPv6 is also supported (with a different block size limit).
- CIDR blocks of subnets **cannot overlap**.



VPC

x.x.x.x/16 or 65,536 addresses (max)  
to  
x.x.x.x/28 or 16 addresses (min)

IP addresses enable resources in your VPC to communicate with each other and with resources over the internet. When you create a VPC, you assign an IPv4 CIDR block (a range of *private* IPv4 addresses) to it. After you create a VPC, you cannot change the address range, so it is important that you choose it carefully. The IPv4 CIDR block might be as large as /16 (which is  $2^{16}$ , or 65,536 addresses) or as small as /28 (which is  $2^4$ , or 16 addresses).

You can optionally associate an IPv6 CIDR block with your VPC and subnets, and assign IPv6 addresses from that block to the resources in your VPC. IPv6 CIDR blocks have a different block size limit.

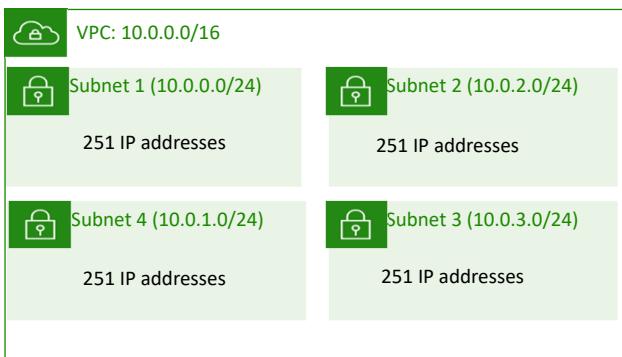
The CIDR block of a subnet can be the same as the CIDR block for a VPC. In this case, the VPC and the subnet are the same size (a single subnet in the VPC). Also, the CIDR block of a subnet can be a subset of the CIDR block for the VPC. This structure enables the definition of multiple subnets. If you create more than one subnet in a VPC, the CIDR blocks of the subnets cannot overlap. You cannot have duplicate IP addresses in the same VPC.

To learn more about IP addressing in a VPC, see the [AWS Documentation](#).

# Reserved IP addresses



**Example:** A VPC with an IPv4 CIDR block of 10.0.0.0/16 has 65,536 total IP addresses. The VPC has four equal-sized subnets. Only 251 IP addresses are available for use by each subnet.



IP Addresses for CIDR block 10.0.0.0/24	Reserved for
10.0.0.0	Network address
10.0.0.1	Internal communication
10.0.0.2	Domain Name System (DNS) resolution
10.0.0.3	Future use
10.0.0.255	Network broadcast address

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

14

When you create a subnet, it requires its own CIDR block. For each CIDR block that you specify, AWS reserves five IP addresses within that block, and these addresses are not available for use. AWS reserves these IP addresses for:

- Network address
- VPC local router (internal communications)
- Domain Name System (DNS) resolution
- Future use
- Network broadcast address

For example, suppose that you create a subnet with an IPv4 CIDR block of 10.0.0.0/24 (which has 256 total IP addresses). The subnet has 256 IP addresses, but only 251 are available because five are reserved.

# Public IP address types



## Public IPv4 address

- Manually assigned through an Elastic IP address
- Automatically assigned through the auto-assign public IP address settings at the subnet level

## Elastic IP address

- Associated with an AWS account
- Can be allocated and remapped anytime
- Additional costs might apply

When you create a VPC, every instance in that VPC gets a private IP address automatically. You can also request a public IP address to be assigned when you create the instance by modifying the subnet's auto-assign public IP address properties.

An *Elastic IP address* is a static and public IPv4 address that is designed for dynamic cloud computing. You can associate an Elastic IP address with any instance or network interface for any VPC in your account. With an Elastic IP address, you can mask the failure of an instance by rapidly remapping the address to another instance in your VPC. Associating the Elastic IP address with the network interface has an advantage over associating it directly with the instance. You can move all of the attributes of the network interface from one instance to another in a single step.

Additional costs might apply when you use Elastic IP addresses, so it is important to release them when you no longer need them.

To learn more about Elastic IP addresses, see [Elastic IP Addresses](#) in the AWS Documentation.

# Elastic network interface



- An elastic network interface is a **virtual network interface** that you can:
  - Attach to an instance.
  - Detach from the instance, and attach to another instance to redirect network traffic.
- Its **attributes follow** when it is reattached to a new instance.
- Each instance in your VPC has a **default network interface** that is assigned a private IPv4 address from the IPv4 address range of your VPC.



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

16

An *elastic network interface* is a virtual network interface that you can attach or detach from an instance in a VPC. A network interface's attributes follow it when it is reattached to another instance. When you move a network interface from one instance to another, network traffic is redirected to the new instance.

Each instance in your VPC has a default network interface (the primary network interface) that is assigned a private IPv4 address from the IPv4 address range of your VPC. You cannot detach a primary network interface from an instance. You can create and attach an additional network interface to any instance in your VPC. The number of network interfaces you can attach varies by instance type.

For more information about [Elastic Network Interfaces](#), see the AWS Documentation.

# Route tables and routes



- A **route table** contains a set of rules (or routes) that **you can configure** to direct network traffic from your subnet.
- Each **route** specifies a destination and a target.
- By default, every route table contains a **local route** for communication within the VPC.
- Each **subnet must be associated with a route table** (at most one).

Main (Default) Route Table

Destination	Target
10.0.0.0/16	local

VPC CIDR block

A *route table* contains a set of rules (called *routes*) that directs network traffic from your subnet. Each route specifies a *destination* and a *target*. The *destination* is the destination CIDR block where you want traffic from your subnet to go. The *target* is the target that the destination traffic is sent through. By default, every route table that you create contains a *local route* for communication in the VPC. You can customize route tables by adding routes. You cannot delete the local route entry that is used for internal communications.

Each subnet in your VPC must be associated with a route table. The *main route table* is the route table automatically assigned to your VPC. It controls the routing for all subnets that are not explicitly associated with any other route table. A subnet can be associated with only one route table at a time, but you can associate multiple subnets with the same route table.

To learn more about route tables, see the [AWS Documentation](#).

## Section 2 key takeaways



18



- A VPC is a logically isolated section of the AWS Cloud.
- A VPC belongs to one Region and requires a CIDR block.
- A VPC is subdivided into subnets.
- A subnet belongs to one Availability Zone and requires a CIDR block.
- Route tables control traffic for a subnet.
- Route tables have a built-in local route.
- You add additional routes to the table.
- The local route cannot be deleted.

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

Some key takeaways from this section of the module include:

- A VPC is a logically isolated section of the AWS Cloud.
- A VPC belongs to one Region and requires a CIDR block.
- A VPC is subdivided into subnets.
- A subnet belongs to one Availability Zone and requires a CIDR block.
- Route tables control traffic for a subnet.
- Route tables have a built-in local route.
- You add additional routes to the table.
- The local route cannot be deleted.

Module 5: Networking and Content Delivery

## Section 3: VPC networking

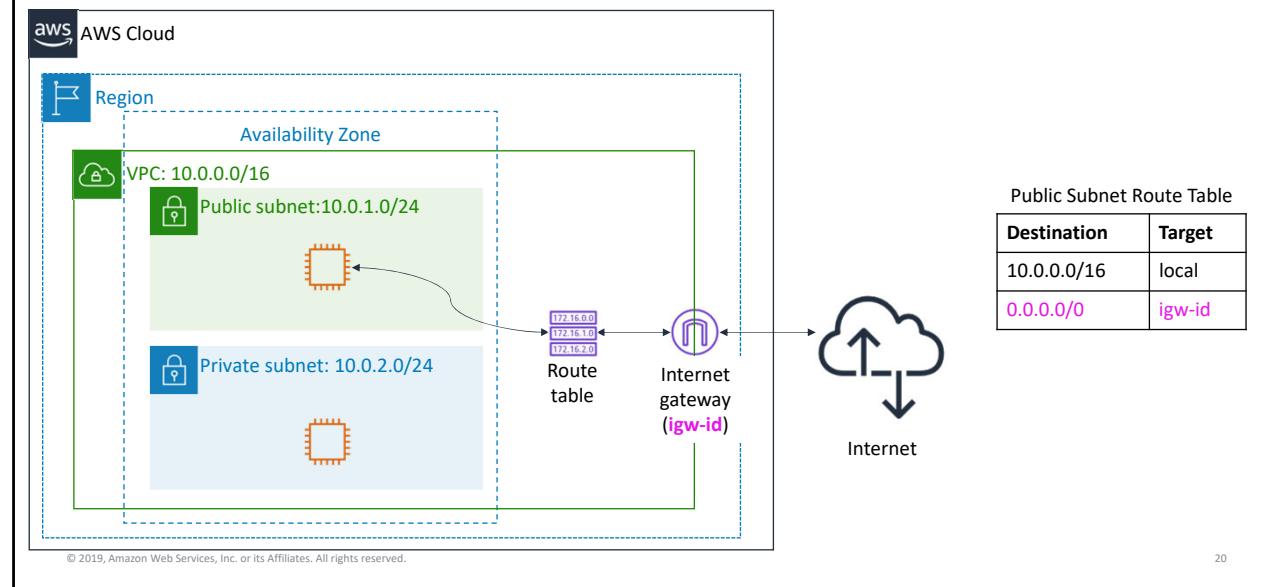
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



### Section 3: VPC networking

Now that you have learned about the basic components of a VPC, you can start routing traffic in interesting ways. In this section, you learn about different networking options.

# Internet gateway



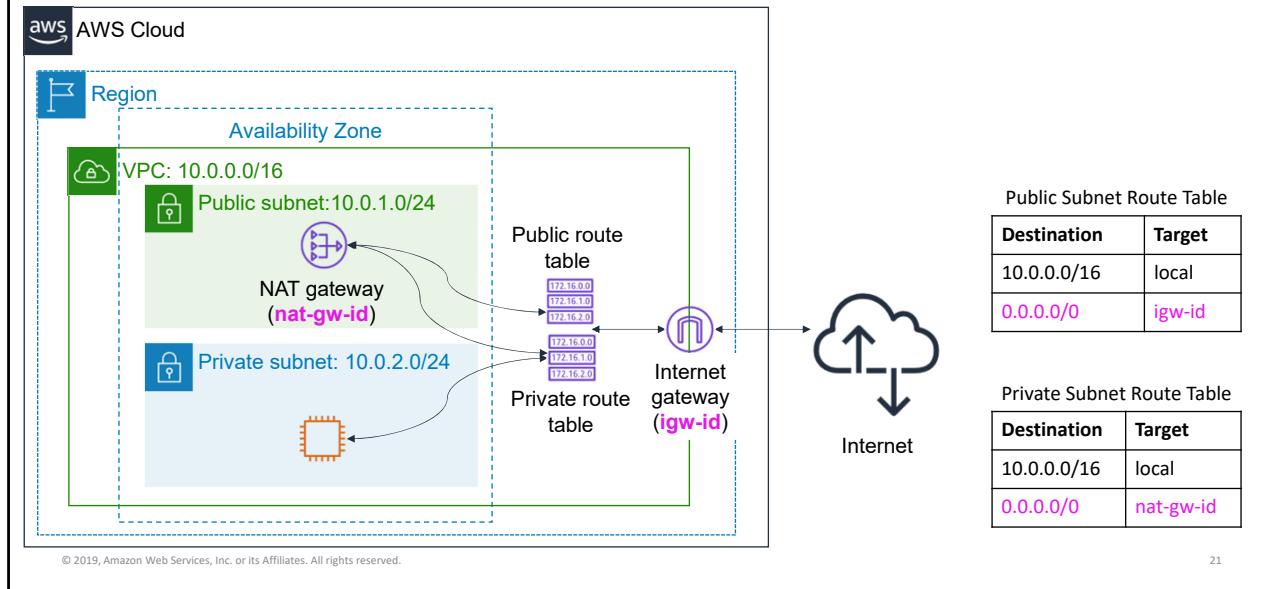
20

An *internet gateway* is a scalable, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet. An internet gateway serves two purposes: to provide a target in your VPC route tables for internet-routable traffic, and to perform network address translation for instances that were assigned public IPv4 addresses.

To make a subnet *public*, you attach an *internet gateway* to your VPC and add a route to the route table to send non-local traffic through the internet gateway to the internet (0.0.0.0/0).

For more information about internet gateways, see [Internet Gateways](#) in the AWS Documentation.

# Network address translation (NAT) gateway



A *network address translation (NAT) gateway* enables instances in a private subnet to connect to the internet or other AWS services, but prevents the internet from initiating a connection with those instances.

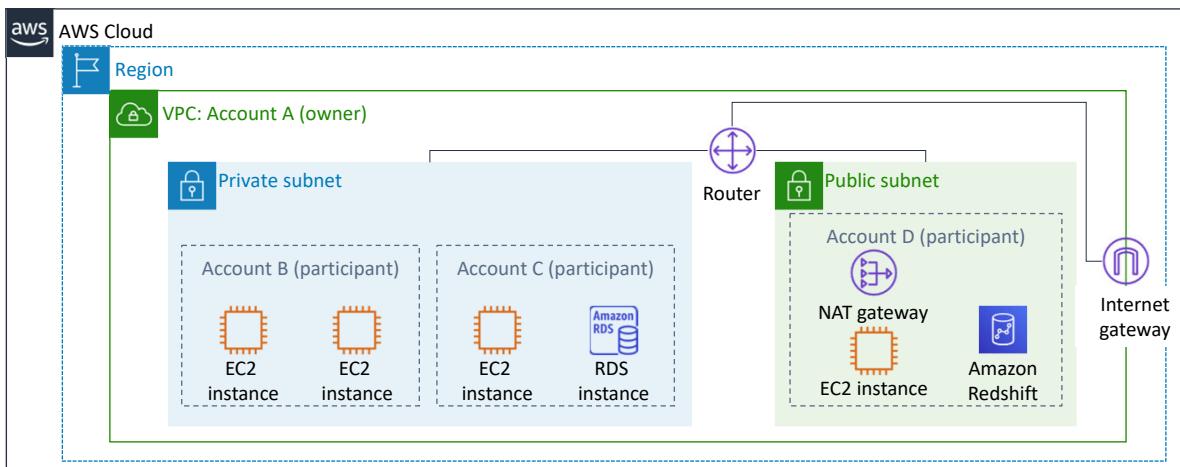
To create a NAT gateway, you must specify the public subnet in which the NAT gateway should reside. You must also specify an Elastic IP address to associate with the NAT gateway when you create it. After you create a NAT gateway, you must update the route table that is associated with one or more of your private subnets to point internet-bound traffic to the NAT gateway. Thus, instances in your private subnets can communicate with the internet.

You can also use a NAT instance in a public subnet in your VPC instead of a NAT gateway. However, a NAT gateway is a managed NAT service that provides better availability, higher bandwidth, and less administrative effort. For common use cases, AWS recommends that you use a NAT gateway instead of a NAT instance.

See the AWS Documentation for more information about

- [NAT gateways](#)
- [NAT instances](#)
- [Differences between NAT gateways and NAT instances](#)

# VPC sharing



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

22

VPC sharing enables customers to share subnets with other AWS accounts in the same organization in AWS Organizations. VPC sharing enables multiple AWS accounts to create their application resources—such as Amazon EC2 instances, Amazon Relational Database Service (Amazon RDS) databases, Amazon Redshift clusters, and AWS Lambda functions—into shared, centrally managed VPCs. In this model, the account that owns the VPC (owner) shares one or more subnets with other accounts (participants) that belong to the same organization in AWS Organizations. After a subnet is shared, the participants can view, create, modify, and delete their application resources in the subnets that are shared with them. Participants cannot view, modify, or delete resources that belong to other participants or the VPC owner.

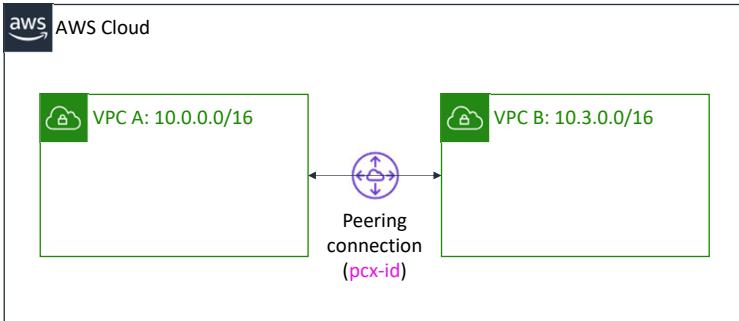
VPC sharing offers several benefits:

- Separation of duties – Centrally controlled VPC structure, routing, IP address allocation
- Ownership – Application owners continue to own resources, accounts, and security groups
- Security groups – VPC sharing participants can reference the security group IDs of each other
- Efficiencies – Higher density in subnets, efficient use of VPNs and AWS Direct Connect
- No hard limits – Hard limits can be avoided—for example, 50 virtual interfaces per AWS Direct Connect connection through simplified network architecture

- Optimized costs – Costs can be optimized through the reuse of NAT gateways, VPC interface endpoints, and intra-Availability Zone traffic

VPC sharing enables you to decouple accounts and networks. You have fewer, larger, centrally managed VPCs. Highly interconnected applications automatically benefit from this approach.

# VPC peering



Route Table for VPC A

Destination	Target
10.0.0.0/16	local
10.3.0.0/16	pcx-id

Route Table for VPC B

Destination	Target
10.3.0.0/16	local
10.0.0.0/16	pcx-id

You can connect VPCs in your own AWS account, between AWS accounts, or between AWS Regions.

## Restrictions:

- IP spaces cannot overlap.
- Transitive peering is not supported.
- You can only have one peering resource between the same two VPCs.

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

23

A *VPC peering connection* is a networking connection between two VPCs that enables you to route traffic between them privately. Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, with a VPC in another AWS account, or with a VPC in a different AWS Region.

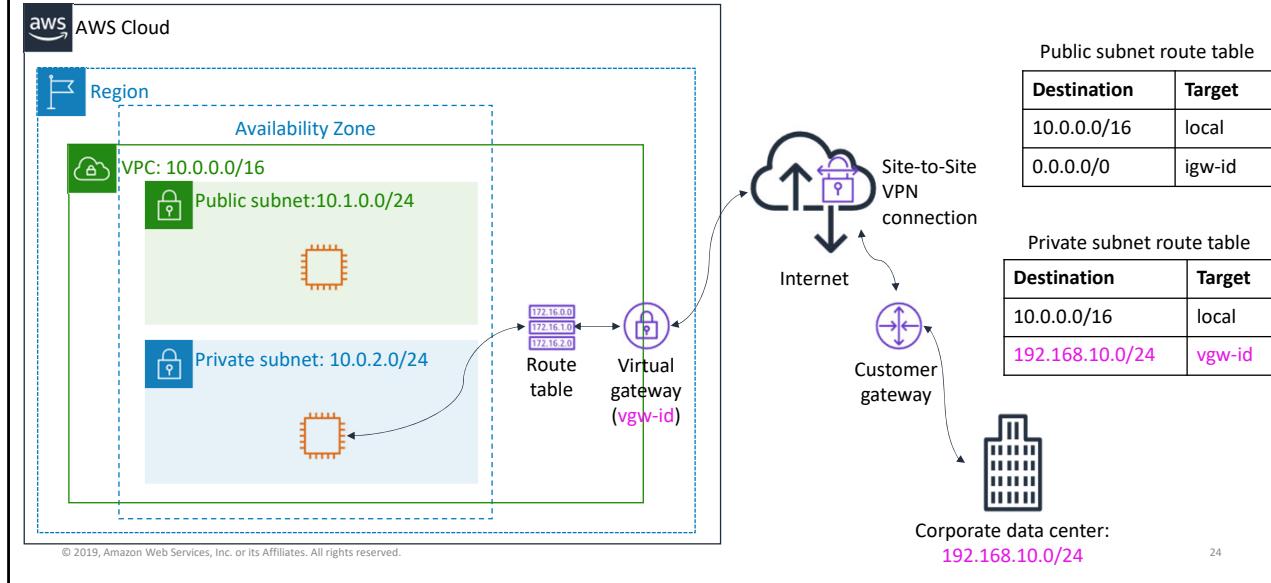
When you set up the peering connection, you create rules in your route table to allow the VPCs to communicate with each other through the peering resource. For example, suppose that you have two VPCs. In the route table for VPC A, you set the destination to be the IP address of VPC B and the target to be the peering resource ID. In the route table for VPC B, you set the destination to be the IP address of VPC A and the target to be the peering resource ID.

VPC peering has some restrictions:

- IP address ranges cannot overlap.
- Transitive peering is not supported. For example, suppose that you have three VPCs: A, B, and C. VPC A is connected to VPC B, and VPC A is connected to VPC C. However, VPC B is *not* connected to VPC C implicitly. To connect VPC B to VPC C, you must explicitly establish that connectivity.
- You can only have one peering resource between the same two VPCs.

For more information about VPC peering, see [VPC Peering](#) in the AWS Documentation.

# AWS Site-to-Site VPN



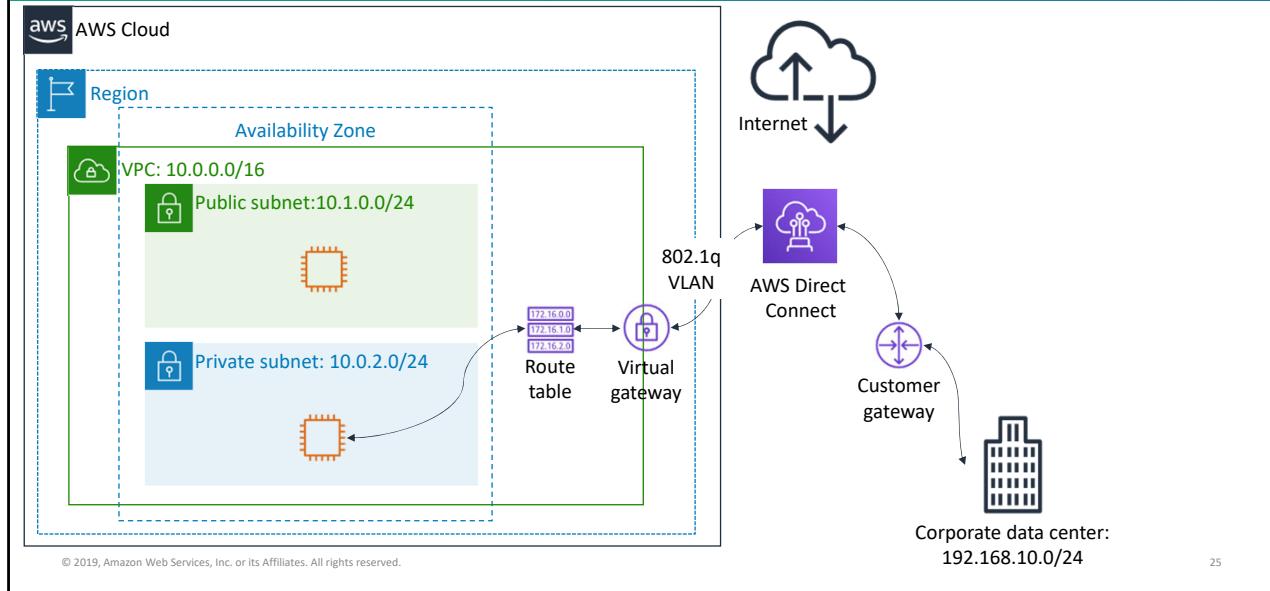
24

By default, instances that you launch into a VPC cannot communicate with a remote network. To connect your VPC to your remote network (that is, create a virtual private network or VPN connection), you:

1. Create a new virtual gateway device (called a *virtual private network (VPN) gateway*) and attach it to your VPC.
2. Define the configuration of the VPN device or the *customer gateway*. The customer gateway is not a device but an AWS resource that provides information to AWS about your VPN device.
3. Create a custom route table to point corporate data center-bound traffic to the VPN gateway. You also must update security group rules. (You will learn about security groups in the next section.)
4. Establish an *AWS Site-to-Site VPN (Site-to-Site VPN) connection* to link the two systems together.
5. Configure routing to pass traffic through the connection.

For more information about AWS Site-to-Site VPN and other VPN connectivity options, see [VPN Connections](#) in the AWS Documentation.

# AWS Direct Connect

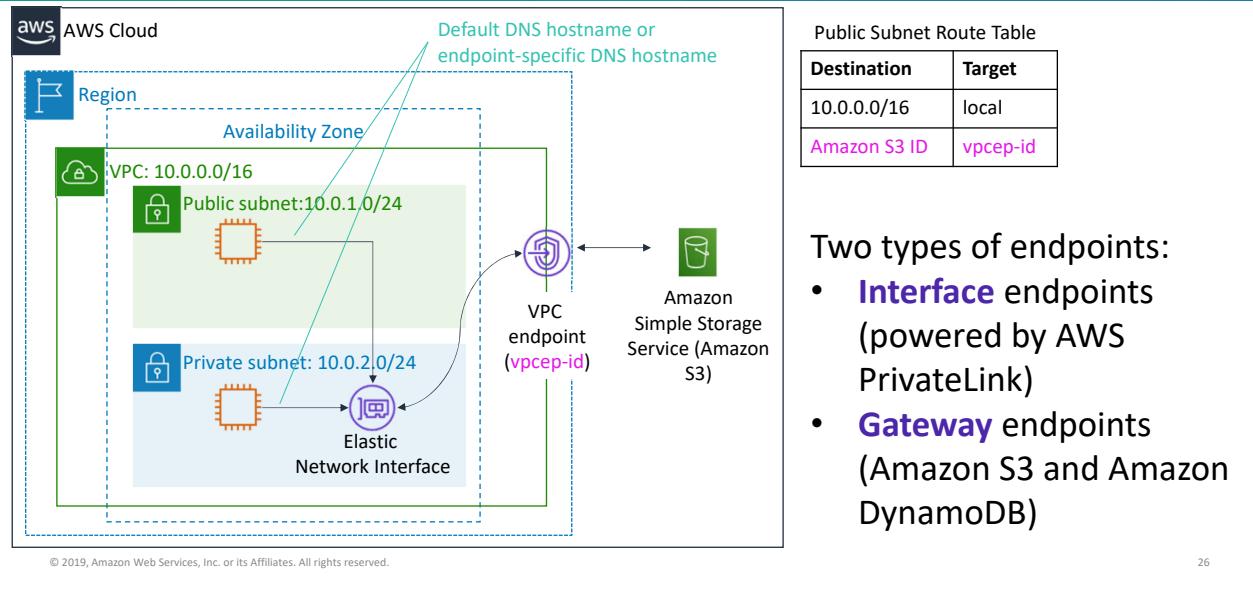


25

One of the challenges of network communication is network performance. Performance can be negatively affected if your data center is located far away from your AWS Region. For such situations, AWS offers AWS Direct Connect, or DX. *AWS Direct Connect* enables you to establish a dedicated, private network connection between your network and one of the DX locations. This private connection can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than internet-based connections. DX uses open standard 802.1q virtual local area networks (VLANs).

For more information about DX, see the [AWS Direct Connect product page](#).

# VPC endpoints



A *VPC endpoint* is a virtual device that enables you to privately connect your VPC to supported AWS services and VPC endpoint services that are powered by AWS PrivateLink. Connection to these services does not require an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC do not require public IP addresses to communicate with resources in the service. Traffic between your VPC and the other service does not leave the Amazon network.

There are two types of VPC endpoints:

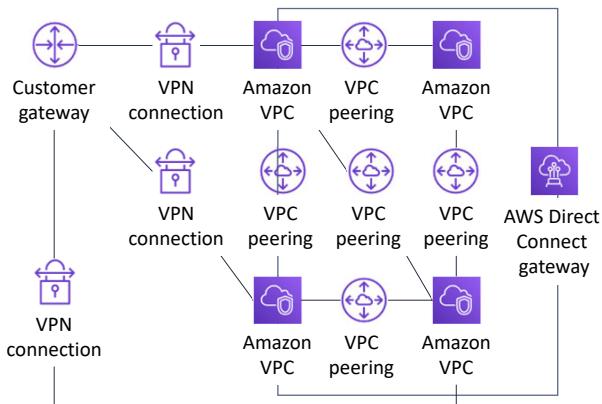
- An *interface VPC endpoint* (interface endpoint) enables you to connect to services that are powered by AWS PrivateLink. These services include some AWS services, services that are hosted by other AWS customers and AWS Partner Network (APN) Partners in their own VPCs (referred to as *endpoint services*), and supported AWS Marketplace APN Partner services. The owner of the service is the *service provider*, and you—as the principal who creates the interface endpoint—are the *service consumer*. You are charged for creating and using an interface endpoint to a service. Hourly usage rates and data processing rates apply. See the AWS Documentation for a list of supported [interface endpoints](#) and for more information about the example shown here.
- Gateway endpoints: The use of gateway endpoints incurs no additional charge. Standard charges for data transfer and resource usage apply.

For more information about VPC endpoints, see [VPC Endpoints](#) in the AWS Documentation.

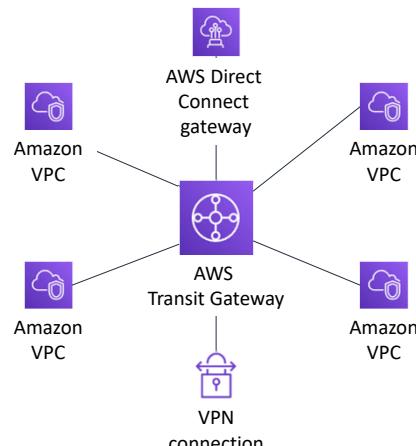
# AWS Transit Gateway



From this...



To this...



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

27

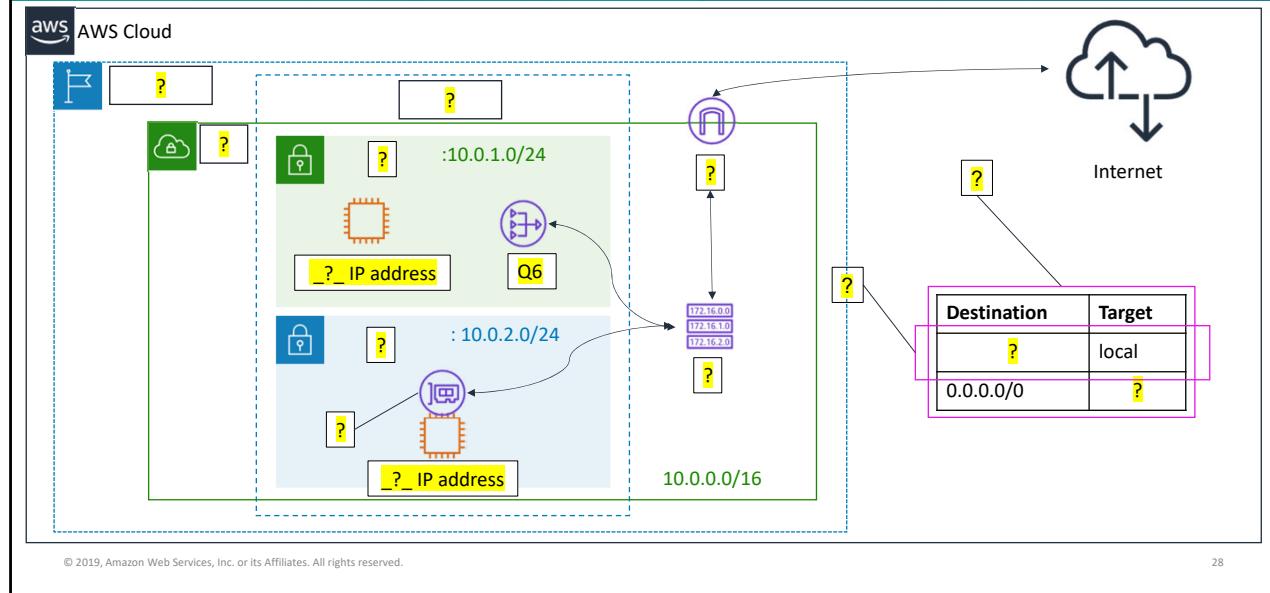
You can configure your VPCs in several ways, and take advantage of numerous connectivity options and gateways. These options and gateways include AWS Direct Connect (via DX gateways), NAT gateways, internet gateways, VPC peering, etc. It is not uncommon to find AWS customers with hundreds of VPCs distributed across AWS accounts and Regions to serve multiple lines of business, teams, projects, and so forth. Things get more complex when customers start to set up connectivity between their VPCs. All the connectivity options are strictly point-to-point, so the number of VPC-to-VPC connections can grow quickly. As you grow the number of workloads that run on AWS, you must be able to scale your networks across multiple accounts and VPCs to keep up with the growth.

Though you can use VPC peering to connect pairs of VPCs, managing point-to-point connectivity across many VPCs without the ability to centrally manage the connectivity policies can be operationally costly and difficult. For on-premises connectivity, you must attach your VPN to each individual VPC. This solution can be time-consuming to build and difficult to manage when the number of VPCs grows into the hundreds.

To solve this problem, you can use AWS Transit Gateway to simplify your networking model. With AWS Transit Gateway, you only need to create and manage a single connection from the central gateway into each VPC, on-premises data center, or remote office across your network. A transit gateway acts as a hub that controls how traffic is routed among all

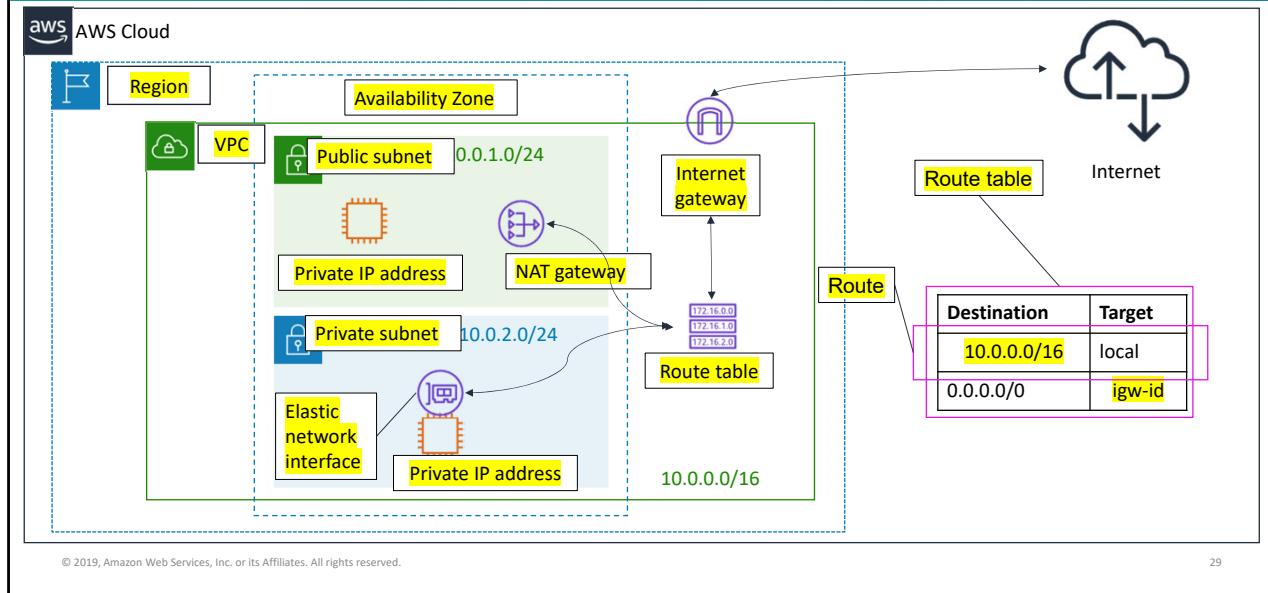
the connected networks, which act like spokes. This hub-and-spoke model significantly simplifies management and reduces operational costs because each network only needs to connect to the transit gateway and not to every other network. Any new VPC is connected to the transit gateway, and is then automatically available to every other network that is connected to the transit gateway. This ease of connectivity makes it easier to scale your network as you grow.

## Activity: Label this network diagram



See if you can recognize the different VPC networking components that you learned about by labeling this network diagram.

## Activity: Solution



Now, see how well you did.

## Recorded Amazon VPC demonstration

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



aws academy

### Set up demo

Amazon Virtual Private Cloud (VPC)

30

Now that you know how to design a VPC, watch [this demonstration](#) to learn how to use the VPC Wizard to set up a VPC with public and private subnets.

## Section 3 key takeaways



- There are several VPC networking options, which include:
  - Internet gateway
  - NAT gateway
  - VPC endpoint
  - VPC peering
  - VPC sharing
  - AWS Site-to-Site VPN
  - AWS Direct Connect
  - AWS Transit Gateway
- You can use the VPC Wizard to implement your design.

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

Some key takeaways from this section of the module include:

- There are several VPC networking options, which include:
  - Internet gateway: Connects your VPC to the internet
  - NAT gateway: Enables instances in a private subnet to connect to the internet
  - VPC endpoint: Connects your VPC to supported AWS services
  - VPC peering: Connects your VPC to other VPCs
  - VPC sharing: Allows multiple AWS accounts to create their application resources into shared, centrally-managed Amazon VPCs
  - AWS Site-to-Site VPN: Connects your VPC to remote networks
  - AWS Direct Connect: Connects your VPC to a remote network by using a dedicated network connection
  - AWS Transit Gateway: A hub-and-spoke connection alternative to VPC peering
- You can use the VPC Wizard to implement your design.

Module 5: Networking and Content Delivery

## Section 4: VPC security

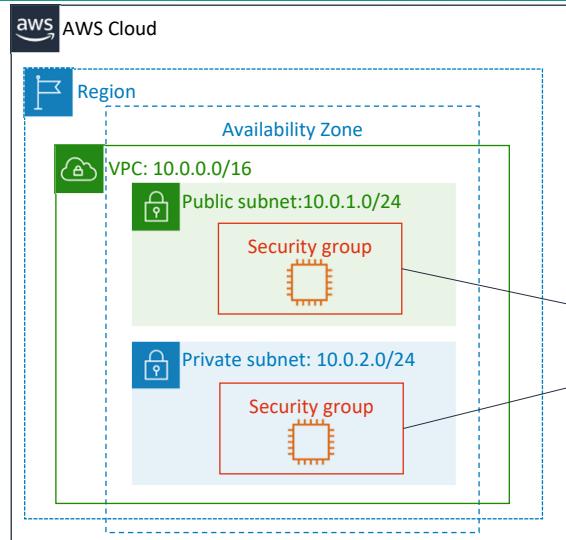
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



### Section 4: VPC security

You can build security into your VPC architecture in several ways so that you have complete control over both incoming and outgoing traffic. In this section, you learn about two Amazon VPC firewall options that you can use to secure your VPC: security groups and network access control lists (network ACLs).

# Security groups



Security groups act at the **instance level**.

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

33

A *security group* acts as a virtual firewall for your instance, and it controls inbound and outbound traffic. Security groups act at the instance level, not the subnet level. Therefore, each instance in a subnet in your VPC can be assigned to a different set of security groups.

At the most basic level, a security group is a way for you to filter traffic to your instances.

# Security groups



- Security groups have **rules** that control inbound and outbound instance traffic.
- Default security groups **deny all inbound** traffic and **allow all outbound** traffic.
- Security groups are **stateful**.

Inbound			
Source	Protocol	Port Range	Description
sg-xxxxxxxx	All	All	Allow inbound traffic from network interfaces assigned to the same security group.

Outbound			
Destination	Protocol	Port Range	Description
0.0.0.0/0	All	All	Allow all outbound IPv4 traffic.
::/0	All	All	Allow all outbound IPv6 traffic.

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

34

Security groups have *rules* that control the inbound and outbound traffic. When you create a security group, it has no inbound rules. Therefore, *no inbound traffic that originates from another host to your instance is allowed* until you add inbound rules to the security group. By default, a security group includes an outbound rule that *allows all outbound traffic*. You can remove the rule and add outbound rules that allow specific outbound traffic only. If your security group has no outbound rules, no outbound traffic that originates from your instance is allowed.

Security groups are *stateful*, which means that state information is kept even after a request is processed. Thus, if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules. Responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules.

## Custom security group examples



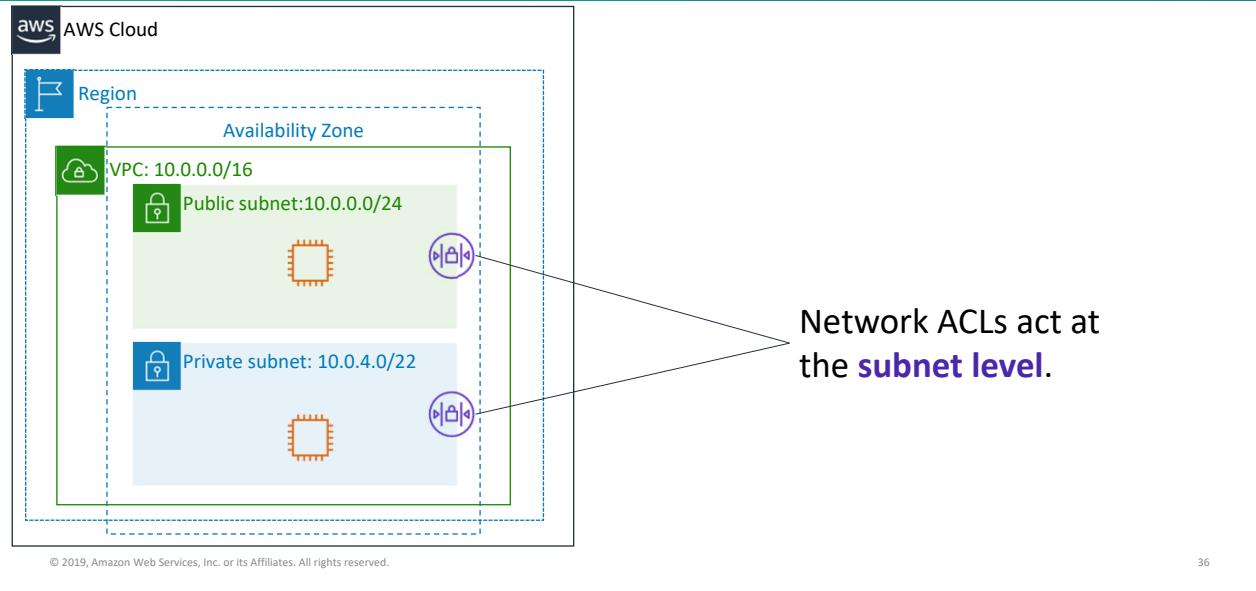
- You can **specify allow** rules, but not deny rules.
- **All rules are evaluated** before the decision to allow traffic.

Inbound			
Source	Protocol	Port Range	Description
0.0.0.0/0	TCP	80	Allow inbound HTTP access from all IPv4 addresses
0.0.0.0/0	TCP	443	Allow inbound HTTPS access from all IPv4 addresses
Your network's public IPv4 address range	TCP	22	Allow inbound SSH access to Linux instances from IPv4 IP addresses in your network (over the internet gateway)

Outbound			
Destination	Protocol	Port Range	Description
The ID of the security group for your Microsoft SQL Server database servers	TCP	1433	Allow outbound Microsoft SQL Server access to instances in the specified security group

When you create a custom security group, you can specify allow rules, but not deny rules. All rules are evaluated before the decision to allow traffic.

# Network access control lists (network ACLs)



A *network access control list (network ACL)* is an optional layer of security for your Amazon VPC. It acts as a firewall for controlling traffic in and out of one or more subnets. To add another layer of security to your VPC, you can set up network ACLs with rules that are similar to your security groups.

Each subnet in your VPC must be associated with a network ACL. If you don't explicitly associate a subnet with a network ACL, the subnet is automatically associated with the default network ACL. You can associate a network ACL with multiple subnets; however, a subnet can be associated with only one network ACL at a time. When you associate a network ACL with a subnet, the previous association is removed.

## Network access control lists (network ACLs)



- A network ACL has **separate inbound and outbound rules**, and each rule can either **allow or deny traffic**.
- **Default** network ACLs **allow** all inbound and outbound IPv4 traffic.
- Network ACLs are **stateless**.

Inbound					
Rule	Type	Protocol	Port Range	Source	Allow/Deny
100	All IPv4 traffic	All	All	0.0.0.0/0	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY

Outbound					
Rule	Type	Protocol	Port Range	Destination	Allow/Deny
100	All IPv4 traffic	All	All	0.0.0.0/0	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

37

A network ACL has separate inbound and outbound rules, and each rule can either allow or deny traffic. Your VPC automatically comes with a modifiable default network ACL. By default, it allows all inbound and outbound IPv4 traffic and, if applicable, IPv6 traffic. The table shows a default network ACL.

Network ACLs are *stateless*, which means that no information about a request is maintained after a request is processed.

## Custom network ACLs examples



- Custom network ACLs **deny** all inbound and outbound traffic until you add rules.
- You can specify **both allow and deny** rules.
- Rules are evaluated in number order, starting with the **lowest number**.

Inbound					
Rule	Type	Protocol	Port Range	Source	Allow/Deny
100	HTTPS	TCP	443	0.0.0.0/0	ALLOW
120	SSH	TCP	22	192.0.2.0/24	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY

Outbound					
Rule	Type	Protocol	Port Range	Destination	Allow/Deny
100	HTTPS	TCP	443	0.0.0.0/0	ALLOW
120	SSH	TCP	22	192.0.2.0/24	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

38

You can create a custom network ACL and associate it with a subnet. By default, each custom network ACL denies all inbound and outbound traffic until you add rules.

A network ACL contains a numbered list of rules that are evaluated in order, starting with the lowest numbered rule. The purpose is to determine whether traffic is allowed in or out of any subnet that is associated with the network ACL. The highest number that you can use for a rule is 32,766. AWS recommends that you create rules in increments (for example, increments of 10 or 100) so that you can insert new rules where you need them later.

For more information about network ACLs, see [Network ACLs](#) in the AWS Documentation.

# Security groups versus network ACLs



Attribute	Security Groups	Network ACLs
Scope	Instance level	Subnet level
Supported Rules	Allow rules only	Allow and deny rules
State	Stateful (return traffic is automatically allowed, regardless of rules)	Stateless (return traffic must be explicitly allowed by rules)
Order of Rules	All rules are evaluated before decision to allow traffic	Rules are evaluated in number order before decision to allow traffic

Here is a summary of the differences between security groups and network ACLs:

- Security groups act at the instance level, but network ACLs act at the subnet level.
- Security groups support allow rules only, but network ACLs support both allow and deny rules.
- Security groups are stateful, but network ACLs are stateless.
- For security groups, all rules are evaluated before the decision is made to allow traffic. For network ACLs, rules are evaluated in number order before the decision is made to allow traffic.

## Activity: Design a VPC



**Scenario:** You have a small business with a website that is hosted on an Amazon Elastic Compute Cloud (Amazon EC2) instance. You have customer data that is stored on a backend database that you want to keep private. You want to use Amazon VPC to set up a VPC that meets the following requirements:

- Your web server and database server must be in separate subnets.
- The first address of your network must be 10.0.0.0. Each subnet must have 256 total IPv4 addresses.
- Your customers must always be able to access your web server.
- Your database server must be able to access the internet to make patch updates.
- Your architecture must be highly available and use at least one custom firewall layer.

Now, it's your turn! In this scenario, you are a small business owner with a website that is hosted on an Amazon Elastic Compute Cloud (Amazon EC2) instance. You have customer data that is stored on a backend database that you want to keep private.

See if you can design a VPC that meets the following requirements:

- Your web server and database server must be in separate subnets.
- The first address of your network must be 10.0.0.0. Each subnet must have 256 IPv4 addresses.
- Your customers must always be able to access your web server.
- Your database server must be able to access the internet to make patch updates.
- Your architecture must be highly available and use at least one custom firewall layer.

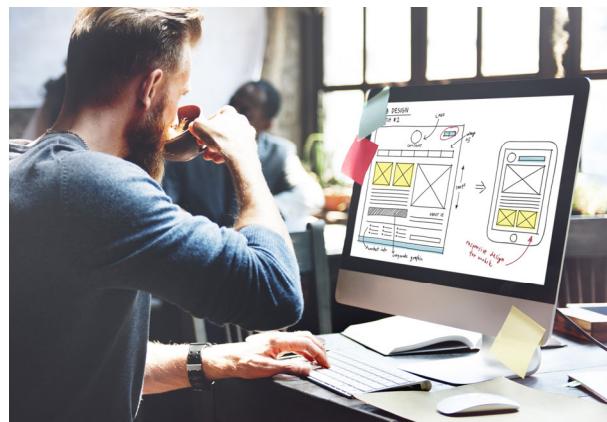
## Section 4 key takeaways



- Build security into your VPC architecture:
  - Isolate subnets if possible.
  - Choose the appropriate gateway device or VPN connection for your needs.
  - Use firewalls.
- Security groups and network ACLs are firewall options that you can use to secure your VPC.

## Lab 2: Build Your VPC and Launch a Web Server

42



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

You will now work on Lab 2: Build Your VPC and Launch a Web Server.

## Lab 2: Scenario



In this lab, you use Amazon VPC to [create your own VPC](#) and add some components to produce a customized network. You [create a security group](#) for your VPC. You also [create an EC2 instance](#) and [configure it](#) to run a web server and to use the security group. You then launch the EC2 instance into the VPC.



Amazon  
VPC



Amazon  
EC2

In this lab, you use Amazon VPC to create your own VPC and add some components to produce a customized network. You also create a security group for your VPC, and then create an EC2 instance and configure it to run a web server and to use the security group. You then launch the EC2 instance into the VPC.

## Lab 2: Tasks

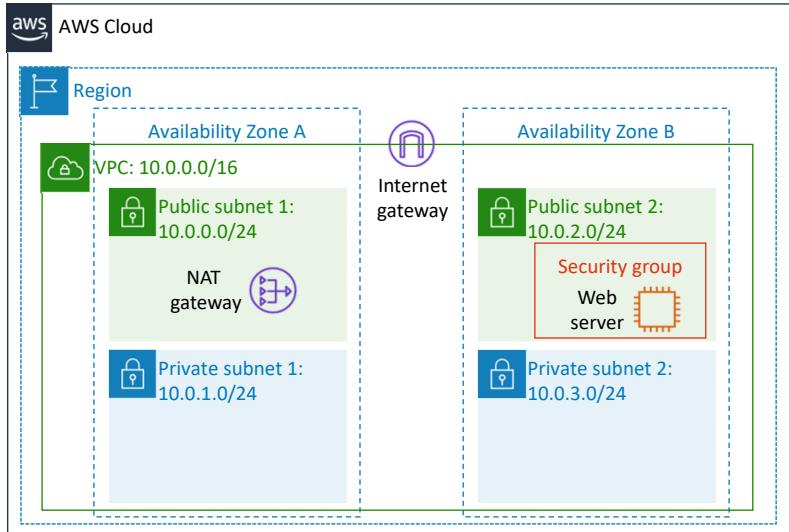


-  • Create a VPC.
-  • Create additional subnets.
-  • Create a VPC security group.
-  • Launch a web server instance.

In this lab, you complete these tasks:

- Create a VPC.
- Create additional subnets.
- Create a VPC security group.
- Launch a web server instance.

## Lab 2: Final product



Public Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	Internet gateway

Private Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	NAT gateway

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

45

This architecture diagram depicts what you create in the lab.



~ 30 minutes



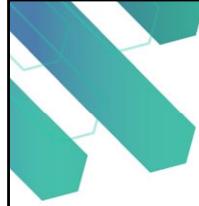
## Begin Lab 2: Build Your VPC and Launch a Web Server



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

46

It is now time to start the lab. It should take you approximately 30 minutes to complete the lab.



## Lab debrief: Key takeaways



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

47

In this lab, you:

- Created an Amazon VPC.
- Created additional subnets.
- Created an Amazon VPC security group.
- Launched a web server instance on Amazon EC2.

Module 5: Networking and Content Delivery

## Section 5: Amazon Route 53

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



## Section 5: Amazon Route 53

# Amazon Route 53



Amazon  
Route 53

- Is a highly available and scalable Domain Name System (DNS) web service
- Is used to route end users to internet applications by translating names (like [www.example.com](http://www.example.com)) into numeric IP addresses (like 192.0.2.1) that computers use to connect to each other
- Is fully compliant with IPv4 and IPv6
- Connects user requests to infrastructure running in AWS and also outside of AWS
- Is used to check the health of your resources
- Features traffic flow
- Enables you to register domain names

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

49

Amazon Route 53 is a highly available and scalable cloud [Domain Name System \(DNS\)](#) web service. It is designed to give developers and businesses a reliable and cost-effective way to route users to internet applications by translating names (like [www.example.com](http://www.example.com)) into the numeric IP addresses (like 192.0.2.1) that computers use to connect to each other. In addition, Amazon Route 53 is fully compliant with IPv6.

Amazon Route 53 effectively connects user requests to infrastructure running in AWS—such as Amazon EC2 instances, Elastic Load Balancing load balancers, or Amazon S3 buckets—and can also be used to route users to infrastructure that is outside of AWS.

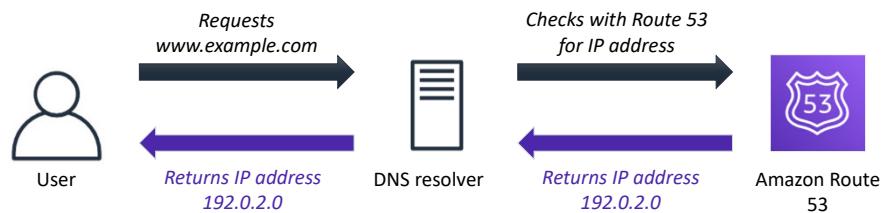
You can use Amazon Route 53 to configure DNS health checks so you that can route traffic to healthy endpoints or independently monitor the health of your application and its endpoints.

Amazon Route 53 traffic flow helps you manage traffic globally through several routing types, which can be combined with DNS failover to enable various low-latency, fault-tolerant architectures. You can use Amazon Route 53 traffic flow's simple visual editor to manage how your users are routed to your application's endpoints—whether in a single AWS Region or distributed around the globe.

Amazon Route 53 also offers Domain Name Registration—you can purchase and manage

domain names (like *example.com*), and Amazon Route 53 will automatically configure DNS settings for your domains.

# Amazon Route 53 DNS resolution



Here is the basic pattern that Amazon Route 53 follows when a user initiates a DNS request. The DNS resolver checks with your domain in Route 53, gets the IP address, and returns it to the user.

# Amazon Route 53 supported routing



- **Simple routing** – Use in single-server environments
- **Weighted round robin routing** – Assign weights to resource record sets to specify the frequency
- **Latency routing** – Help improve your global applications
- **Geolocation routing** – Route traffic based on location of your users
- **Geoproximity routing** – Route traffic based on location of your resources
- **Failover routing** – Fail over to a backup site if your primary site becomes unreachable
- **Multivalue answer routing** – Respond to DNS queries with up to eight healthy records selected at random

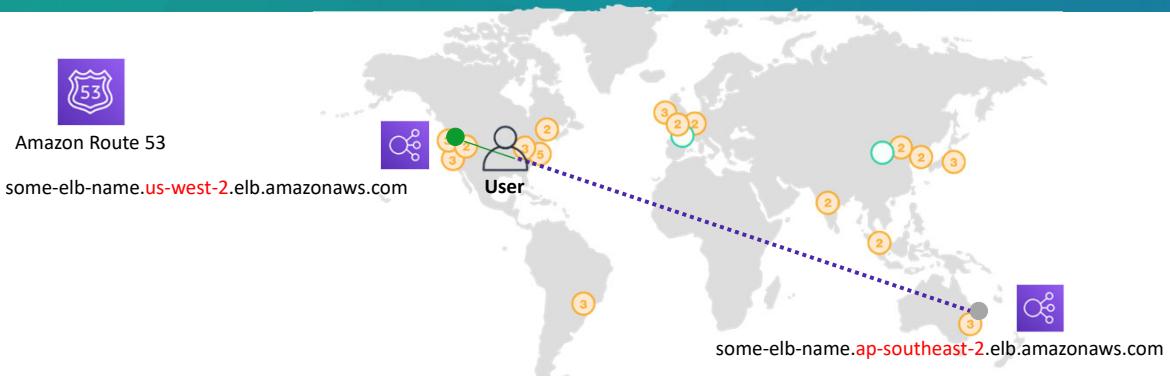
Amazon Route 53 supports several types of routing policies, which determine how Amazon Route 53 responds to queries:

- *Simple routing (round robin)* – Use for a single resource that performs a given function for your domain (such as a web server that serves content for the example.com website).
- *Weighted round robin routing* – Use to route traffic to multiple resources in proportions that you specify. Enables you to assign weights to resource record sets to specify the frequency with which different responses are served. You might want to use this capability to do A/B testing, which is when you send a small portion of traffic to a server where you made a software change. For instance, suppose you have two record sets that are associated with one DNS name: one with weight 3 and one with weight 1. In this case, 75 percent of the time, Amazon Route 53 will return the record set with weight 3, and 25 percent of the time, Amazon Route 53 will return the record set with weight 1. Weights can be any number between 0 and 255.
- *Latency routing (LBR)* – Use when you have resources in multiple AWS Regions and you want to route traffic to the Region that provides the best latency. Latency routing works by routing your customers to the AWS endpoint (for example, Amazon EC2 instances, Elastic IP addresses, or load balancers) that provides the fastest experience based on

actual performance measurements of the different AWS Regions where your application runs.

- *Geolocation routing* – Use when you want to route traffic based on the location of your users. When you use geolocation routing, you can localize your content and present some or all of your website in the language of your users. You can also use geolocation routing to restrict the distribution of content to only the locations where you have distribution rights. Another possible use is for balancing the load across endpoints in a predictable, easy-to-manage way, so that each user location is consistently routed to the same endpoint.
- *Geoproximity routing* – Use when you want to route traffic based on the location of your resources and, optionally, shift traffic from resources in one location to resources in another.
- *Failover routing (DNS failover)* – Use when you want to configure active-passive failover. Amazon Route 53 can help detect an outage of your website and redirect your users to alternate locations where your application is operating properly. When you enable this feature, Amazon Route 53 health-checking agents will monitor each location or endpoint of your application to determine its availability. You can take advantage of this feature to increase the availability of your customer-facing application.
- *Multivalue answer routing* – Use when you want Route 53 to respond to DNS queries with up to eight healthy records that are selected at random. You can configure Amazon Route 53 to return multiple values—such as IP addresses for your web servers—in response to DNS queries. You can specify multiple values for almost any record, but multivalue answer routing also enables you to check the health of each resource so that Route 53 returns only values for healthy resources. It's not a substitute for a load balancer, but the ability to return multiple health-checkable IP addresses is a way to use DNS to improve availability and load balancing.

# Use case: Multi-region deployment



Name	Type	Value
example.com	ALIAS	some-elb-name.us-west-2.elb.amazonaws.com
example.com	ALIAS	some-elb-name.ap-southeast-2.elb.amazonaws.com

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

52

Multi-Region deployment is an example use case for Amazon Route 53. With Amazon Route 53, the user is automatically directed to the Elastic Load Balancing load balancer that's closest to the user.

The benefits of multi-region deployment of Route 53 include:

- Latency-based routing to the Region
- Load balancing routing to the Availability Zone

# Amazon Route 53 DNS failover



Improve the availability of your applications that run on AWS by:

- Configuring backup and failover scenarios for your own applications
- Enabling highly available multi-region architectures on AWS
- Creating health checks

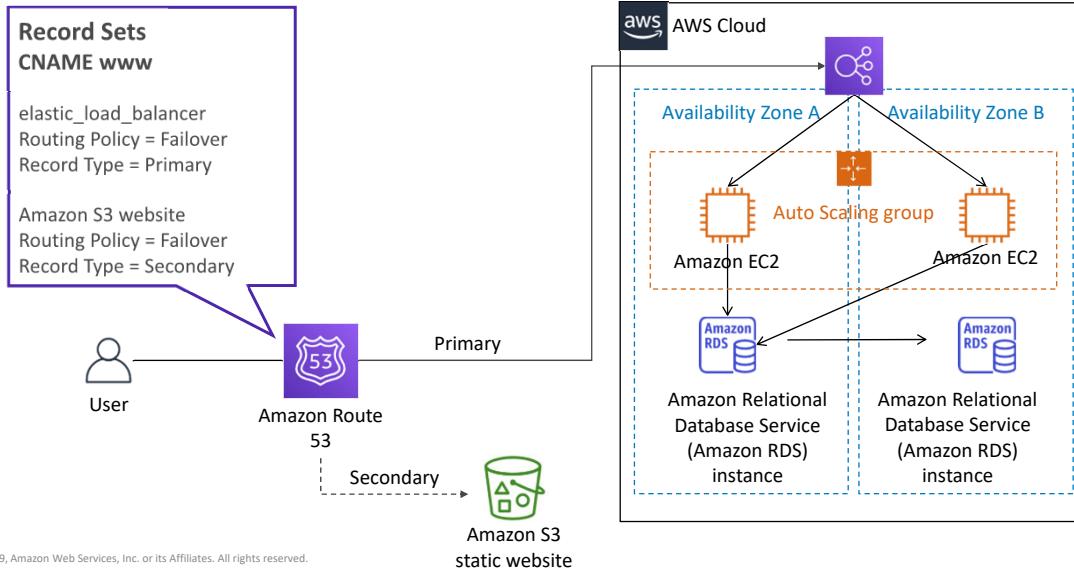
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

53

Amazon Route 53 enables you to improve the availability of your applications that run on AWS by:

- Configuring backup and failover scenarios for your own applications.
- Enabling highly available multi-Region architectures on AWS.
- Creating health checks to monitor the health and performance of your web applications, web servers, and other resources. Each health check that you create can monitor one of the following—the health of a specified resource, such as a web server; the status of other health checks; and the status of an Amazon CloudWatch alarm.

# DNS failover for a multi-tiered web application



54

Here, you see how DNS failover works in a typical architecture for a multi-tiered web application. Route 53 passes traffic to a load balancer, which then distributes traffic to a fleet of EC2 instances.

You can do the following tasks with Route 53 to ensure high availability:

1. Create two DNS records for the Canonical Name Record (CNAME) `www` with a routing policy of *Failover Routing*. The first record is the primary route policy, which points to the load balancer for your web application. The second record is the secondary route policy, which points to your static Amazon S3 website.
2. Use Route 53 health checks to make sure that the primary is running. If it is, all traffic defaults to your web application stack. Failover to the static backup site would be triggered if either the web server goes down (or stops responding), or the database instance goes down.

## Section 5 key takeaways



- Amazon Route 53 is a highly available and scalable cloud DNS web service that translates domain names into numeric IP addresses.
- Amazon Route 53 supports several types of routing policies.
- Multi-Region deployment improves your application's performance for a global audience.
- You can use Amazon Route 53 failover to improve the availability of your applications.

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

Some key takeaways from this section of the module include:

- Amazon Route 53 is a highly available and scalable cloud DNS web service that translates domain names into numeric IP addresses.
- Amazon Route 53 supports several types of routing policies.
- Multi-Region deployment improves your application's performance for a global audience.
- You can use Amazon Route 53 failover to improve the availability of your applications.

Module 5: Networking and Content Delivery

## Section 6: Amazon CloudFront

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

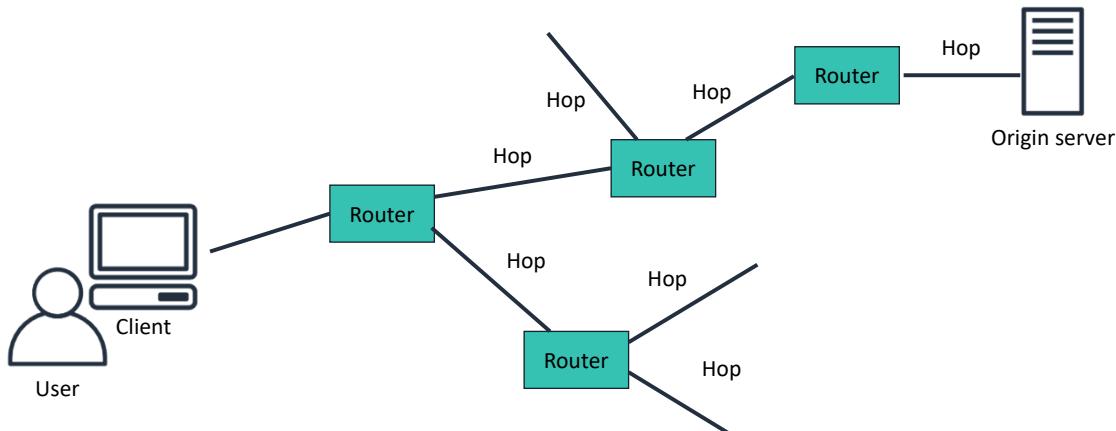


### Section 6: Amazon CloudFront

The purpose of networking is to share information between connected resources. So far in this module, you learned about VPC networking with Amazon VPC. You learned about the different options for connecting your VPC to the internet, to remote networks, to other VPCs, and to AWS services.

Content delivery occurs over networks, too—for example, when you stream a movie from your favorite streaming service. In this final section, you learn about Amazon CloudFront, which is a content delivery network (CDN) service.

# Content delivery and network latency



As explained earlier in this module when you were learning about AWS Direct Connect, one of the challenges of network communication is network performance. When you browse a website or stream a video, your request is routed through many different networks to reach an origin server. The origin server (or origin) stores the original, definitive versions of the objects (webpages, images, and media files). The number of network hops and the distance that the request must travel significantly affect the performance and responsiveness of the website. Further, network latency is different in various geographic locations. For these reasons, a content delivery network might be the solution.

# Content delivery network (CDN)



- Is a globally distributed system of caching servers
- Caches copies of commonly requested files (static content)
- Delivers a local copy of the requested content from a nearby cache edge or Point of Presence
- Accelerates delivery of dynamic content
- Improves application performance and scaling

A content delivery network (CDN) is a globally distributed system of caching servers. A CDN caches copies of commonly requested files (static content, such as Hypertext Markup Language, or HTML; Cascading Style Sheets, or CSS; JavaScript; and image files) that are hosted on the application origin server. The CDN delivers a local copy of the requested content from a cache edge or Point of Presence that provides the fastest delivery to the requester.

CDNs also deliver dynamic content that is unique to the requester and is not cacheable. Having a CDN deliver dynamic content improves application performance and scaling. The CDN establishes and maintains secure connections closer to the requester. If the CDN is on the same network as the origin, routing back to the origin to retrieve dynamic content is accelerated. In addition, content such as form data, images, and text can be ingested and sent back to the origin, thus taking advantage of the low-latency connections and proxy behavior of the PoP.



Amazon  
CloudFront

- Fast, global, and secure CDN service
- Global network of edge locations and Regional edge caches
- Self-service model
- Pay-as-you-go pricing

Amazon CloudFront is a fast CDN service that securely delivers data, videos, applications, and application programming interfaces (APIs) to customers globally with low latency and high transfer speeds. It also provides a developer-friendly environment. Amazon CloudFront delivers files to users over a global network of edge locations and Regional edge caches. Amazon CloudFront is different from traditional content delivery solutions because it enables you to quickly obtain the benefits of high-performance content delivery without negotiated contracts, high prices, or minimum fees. Like other AWS services, Amazon CloudFront is a self-service offering with pay-as-you-go pricing.

# Amazon CloudFront infrastructure

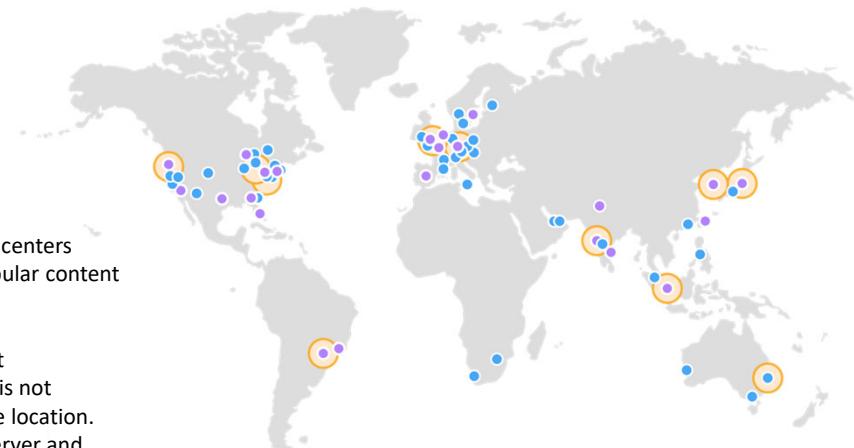


- Edge locations
- Multiple edge locations
- Regional edge caches

- **Edge locations** – Network of data centers that CloudFront uses to serve popular content quickly to customers.
- **Regional edge cache** – CloudFront location that caches content that is not popular enough to stay at an edge location. It is located between the origin server and the global edge location.

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

60



Amazon CloudFront delivers content through a worldwide network of data centers that are called *edge locations*. When a user requests content that you serve with CloudFront, the user is routed to the edge location that provides the lowest latency (or time delay) so that content is delivered with the best possible performance. CloudFront edge locations are designed to serve popular content quickly to your viewers.

As objects become less popular, individual edge locations might remove those objects to make room for more popular content. For the less popular content, CloudFront has *Regional edge caches*. Regional edge caches are CloudFront locations that are deployed globally and are close to your viewers. They are located between your origin server and the global edge locations that serve content directly to viewers. A Regional edge cache has a larger cache than an individual edge location, so objects remain in the Regional edge cache longer. More of your content remains closer to your viewers, which reduces the need for CloudFront to go back to your origin server and improves overall performance for viewers.

For more information about how Amazon CloudFront works, see [How CloudFront Delivers Content](#) in the AWS Documentation.

## Amazon CloudFront benefits



- Fast and global
- Security at the edge
- Highly programmable
- Deeply integrated with AWS
- Cost-effective

Amazon CloudFront provides the following benefits:

- *Fast and global* – Amazon CloudFront is massively scaled and globally distributed. To deliver content to end users with low latency, Amazon CloudFront uses a global network that consists of edge locations and regional caches.
- *Security at the edge* – Amazon CloudFront provides both network-level and application-level protection. Your traffic and applications benefit through various built-in protections, such as AWS Shield Standard, at no additional cost. You can also use configurable features, such as AWS Certificate Manager (ACM), to create and manage custom Secure Sockets Layer (SSL) certificates at no extra cost.
- *Highly programmable* – Amazon CloudFront features can be customized for specific application requirements. It integrates with Lambda@Edge so that you can run custom code across AWS locations worldwide, which enables you to move complex application logic closer to users to improve responsiveness. The CDN also supports integrations with other tools and automation interfaces for DevOps. It offers continuous integration and continuous delivery (CI/CD) environments.
- *Deeply integrated with AWS* – Amazon CloudFront is integrated with AWS, with both physical locations that are directly connected to the AWS Global Infrastructure and other AWS services. You can use APIs or the AWS Management Console to programmatically

configure all features in the CDN.

- *Cost-effective* – Amazon CloudFront is cost-effective because it has no minimum commitments and charges you only for what you use. Compared to self-hosting, Amazon CloudFront avoids the expense and complexity of operating a network of cache servers in multiple sites across the internet. It eliminates the need to overprovision capacity to serve potential spikes in traffic. Amazon CloudFront also uses techniques like collapsing simultaneous viewer requests at an edge location for the same file into a single request to your origin server. The result is reduced load on your origin servers and reduced need to scale your origin infrastructure, which can result in further cost savings. If you use AWS origins such as Amazon Simple Storage Service (Amazon S3) or Elastic Load Balancing, you pay only for storage costs, not for any data transferred between these services and CloudFront.

## Data transfer out

- Charged for the volume of data transferred out from Amazon CloudFront edge location to the internet or to your origin.

## HTTP(S) requests

- Charged for number of HTTP(S) requests.

## Invalidation requests

- No additional charge for the first 1,000 paths that are requested for invalidation each month. Thereafter, \$0.005 per path that is requested for invalidation.

## Dedicated IP custom SSL

- \$600 per month for each custom SSL certificate that is associated with one or more CloudFront distributions that use the Dedicated IP version of custom SSL certificate support.

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

62

Amazon CloudFront charges are based on actual usage of the service in four areas:

- Data transfer out* – You are charged for the volume of data that is transferred out from Amazon CloudFront edge locations, measured in GB, to the internet or to your origin (both AWS origins and other origin servers). Data transfer usage is totaled separately for specific geographic regions, and then cost is calculated based on pricing tiers for each area. If you use other AWS services as the origins of your files, you are charged separately for your use of those services, including storage and compute hours.
- HTTP(S) requests* – You are charged for the number of HTTP(S) requests that are made to Amazon CloudFront for your content.
- Invalidation requests* – You are charged per path in your invalidation request. A path that is listed in your invalidation request represents the URL (or multiple URLs if the path contains a wildcard character) of the object that you want to invalidate from CloudFront cache. You can request up to 1,000 paths each month from Amazon CloudFront at no additional charge. Beyond the first 1,000 paths, you are charged per path that is listed in

your invalidation requests.

- *Dedicated IP custom Secure Sockets Layer (SSL)* – You pay \$600 per month for each custom SSL certificate that is associated with one or more CloudFront distributions that use the Dedicated IP version of custom SSL certificate support. This monthly fee is prorated by the hour. For example, if your custom SSL certificate was associated with at least one CloudFront distribution for just 24 hours (that is, 1 day) in the month of June, your total charge for using the custom SSL certificate feature in June is  $(1 \text{ day} / 30 \text{ days}) * \$600 = \$20$ .

For the latest pricing information, see the [Amazon CloudFront pricing page](#).

## Section 6 key takeaways



- A CDN is a globally distributed system of caching servers that accelerates delivery of content.
- Amazon CloudFront is a fast CDN service that securely delivers data, videos, applications, and APIs over a global infrastructure with low latency and high transfer speeds.
- Amazon CloudFront offers many benefits.

Some key takeaways from this section of the module include:

- A CDN is a globally distributed system of caching servers that accelerates delivery of content.
- Amazon CloudFront is a fast CDN service that securely delivers data, videos, applications, and APIs over a global infrastructure with low latency and high transfer speeds.
- Amazon CloudFront offers many benefits, including:
  - Fast and global
  - Security at the edge
  - Highly programmable
  - Deeply integrated with AWS
  - Cost-effective

Module 5: Networking and Content Delivery

## Module wrap-up

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



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

## Module summary



In summary, in this module you learned how to:

- Recognize the basics of networking
- Describe virtual networking in the cloud with Amazon VPC
- Label a network diagram
- Design a basic VPC architecture
- Indicate the steps to build a VPC
- Identify security groups
- Create your own VPC and added additional components to it to produce a customized network
- Identify the fundamentals of Amazon Route 53
- Recognize the benefits of Amazon CloudFront

In summary, in this module you learned how to:

- Recognize the basics of networking
- Describe virtual networking in the cloud with Amazon VPC
- Label a network diagram
- Design a basic VPC architecture
- Indicate the steps to build a VPC
- Identify security groups
- Create your own VPC and added additional components to it to produce a customized network
- Identify the fundamentals of Amazon Route 53
- Recognize the benefits of Amazon CloudFront

# Complete the knowledge check



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

66

Now, complete the knowledge check.

## Sample exam question



Which AWS networking service enables a company to create a virtual network within AWS?

- A. AWS Config
- B. Amazon Route 53
- C. AWS Direct Connect
- D. Amazon VPC

Look at the answer choices and rule them out based on the keywords that were previously highlighted.

## Additional resources



- [Amazon VPC overview page](#)
- [Amazon Virtual Private Cloud Connectivity Options whitepaper](#)
- [One to Many: Evolving VPC Design](#) AWS Architecture blog post
- [Amazon VPC User Guide](#)
- [Amazon CloudFront overview page](#)

If you want to learn more about the topics covered in this module, you might find the following additional resources helpful:

- [Amazon VPC overview page](#)
- [Amazon Virtual Private Cloud Connectivity Options whitepaper](#)
- [One to Many: Evolving VPC Design](#) AWS Architecture blog post
- [Amazon VPC User Guide](#)
- [Amazon CloudFront overview page](#)

# Thank you

© 2019 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections or feedback on the course, please email us at: [aws-course-feedback@amazon.com](mailto:aws-course-feedback@amazon.com). For all other questions, contact us at: <https://aws.amazon.com/contact-us/aws-training/>. All trademarks are the property of their owners.



Thanks for participating!

AWS Academy Cloud Foundations

# Module 6: Compute

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



Welcome to Module 6: Compute

# Module overview



## Topics

- Compute services overview
- Amazon EC2
- Amazon EC2 cost optimization
- Container services
- Introduction to AWS Lambda
- Introduction to AWS Elastic Beanstalk

## Activities

- Amazon EC2 versus Managed Service
- Hands-on with AWS Lambda
- Hands-on with AWS Elastic Beanstalk

## Demo

- Recorded demonstration of Amazon EC2

## Lab

- Introduction to Amazon EC2



## Knowledge check

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

2

This module will address the following topics:

- Compute services overview
- Amazon EC2
- Amazon EC2 cost optimization
- Container services
- Introduction to AWS Lambda
- Introduction to AWS Elastic Beanstalk

Section 2 includes a recorded **Amazon EC2 demonstration**. The end of this same section includes a **hands-on lab**, where you will practice launching an EC2 instance by using the AWS Management Console. There is also an activity in this section that has you compare the advantages and disadvantages of running a database deployment on Amazon EC2, versus running it on Amazon Relational Database Service (RDS).

Section 5 includes a hands-on AWS Lambda activity and section 6 includes a hands-on Elastic Beanstalk activity.

Finally, you will be asked to complete a **knowledge check** that will test your understanding of the key concepts that are covered in this module.

## Module objectives



After completing this module, you should be able to:

- Provide an overview of different AWS compute services in the cloud
- Demonstrate why to use Amazon Elastic Compute Cloud (Amazon EC2)
- Identify the functionality in the EC2 console
- Perform basic functions in Amazon EC2 to build a virtual computing environment
- Identify Amazon EC2 cost optimization elements
- Demonstrate when to use AWS Elastic Beanstalk
- Demonstrate when to use AWS Lambda
- Identify how to run containerized applications in a cluster of managed servers

After completing this module, you should be able to:

- Provide an overview of different AWS compute services in the cloud
- Demonstrate why to use Amazon Elastic Compute Cloud (Amazon EC2)
- Identify the functionality in the EC2 console
- Perform basic functions in EC2 to build a virtual computing environment
- Identify EC2 cost optimization elements
- Demonstrate when to use AWS Elastic Beanstalk
- Demonstrate when to use AWS Lambda
- Identify how to run containerized applications in a cluster of managed servers

Module 6: Compute

## Section 1: Compute services overview

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

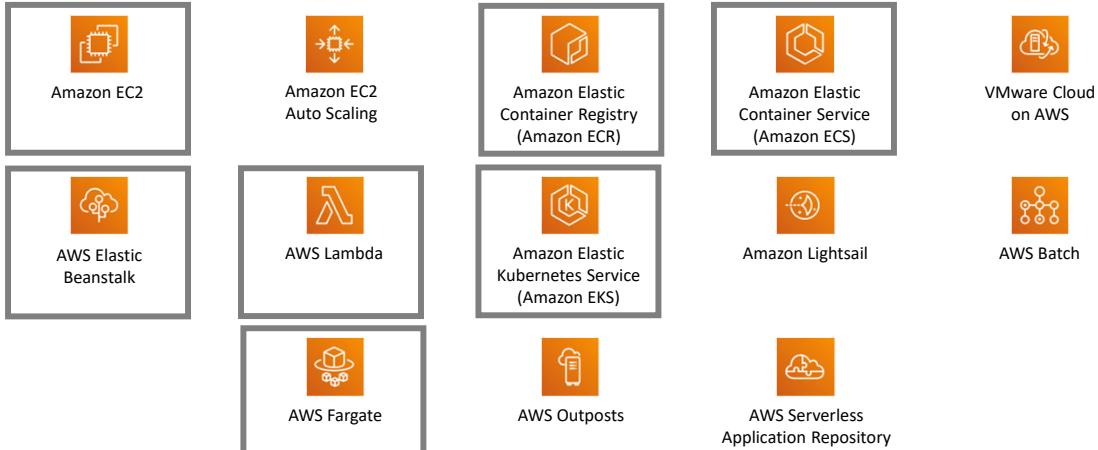


Introducing Section 1: Compute services overview.

# AWS compute services



Amazon Web Services (AWS) offers many compute services. This module will discuss the highlighted services.



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

5

Amazon Web Services (AWS) offers many compute services. Here is a brief summary of what each compute service offers:

- **Amazon Elastic Compute Cloud (Amazon EC2)** provides resizable virtual machines.
- **Amazon EC2 Auto Scaling** supports application availability by allowing you to define conditions that will automatically launch or terminate EC2 instances.
- **Amazon Elastic Container Registry (Amazon ECR)** is used to store and retrieve Docker images.
- **Amazon Elastic Container Service (Amazon ECS)** is a container orchestration service that supports Docker.
- **VMware Cloud on AWS** enables you to provision a hybrid cloud without custom hardware.
- **AWS Elastic Beanstalk** provides a simple way to run and manage web applications.
- **AWS Lambda** is a serverless compute solution. You pay only for the compute time that you use.
- **Amazon Elastic Kubernetes Service (Amazon EKS)** enables you to run managed Kubernetes on AWS.
- **Amazon Lightsail** provides a simple-to-use service for building an application or website.
- **AWS Batch** provides a tool for running batch jobs at any scale.
- **AWS Fargate** provides a way to run containers that reduce the need for you to manage

servers or clusters.

- **AWS Outposts** provides a way to run select AWS services in your on-premises data center.
- **AWS Serverless Application Repository** provides a way to discover, deploy, and publish serverless applications.

This module will discuss details of the services that are highlighted on the slide.

# Categorizing compute services



Services	Key Concepts	Characteristics	Ease of Use
• Amazon EC2	• Infrastructure as a service (IaaS) • Instance-based • <b>Virtual machines</b>	• Provision virtual machines that you can manage as you choose	A familiar concept to many IT professionals.
• AWS Lambda	• <b>Serverless</b> computing • Function-based • Low-cost	• Write and deploy code that runs on a schedule or that can be triggered by events • Use when possible (architect for the cloud)	A relatively new concept for many IT staff members, but easy to use after you learn how.
• Amazon ECS • Amazon EKS • AWS Fargate • Amazon ECR	• <b>Container-based</b> computing • Instance-based	• Spin up and run jobs more quickly	AWS Fargate reduces administrative overhead, but you can use options that give you more control.
• AWS Elastic Beanstalk	• Platform as a service (PaaS) • For <b>web applications</b>	• Focus on your code (building your application) • Can easily tie into other services—databases, Domain Name System (DNS), etc.	Fast and easy to get started.

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

6

You can think of each AWS compute service as belonging to one of four broad categories: virtual machines (VMs) that provide infrastructure as a service (IaaS), serverless, container-based, and platform as a service (PaaS).

**Amazon EC2** provides virtual machines, and you can think of it as infrastructure as a service (IaaS). IaaS services provide flexibility and leave many of the server management responsibilities to you. You choose the operating system, and you also choose the size and resource capabilities of the servers that you launch. For IT professionals who have experience using on-premises computing, virtual machines are a familiar concept. Amazon EC2 was one of the first AWS services, and it remains one of the most popular services.

**AWS Lambda** is a zero-administration compute platform. AWS Lambda enables you to run code without provisioning or managing servers. You pay only for the compute time that is consumed. This serverless technology concept is relatively new to many IT professionals. However, it is becoming more popular because it supports cloud-native architectures, which enable massive scalability at a lower cost than running servers 24/7 to support the same workloads.

Container-based services—including **Amazon Elastic Container Service**, **Amazon Elastic Kubernetes Service**, **AWS Fargate**, and **Amazon Elastic Container Registry**—enable you to run multiple workloads on a single operating system (OS). Containers spin up more quickly than virtual machines, thus offering responsiveness. Container-based solutions continue to

grow in popularity.

Finally, **AWS Elastic Beanstalk** provides a platform as a service (PaaS). It facilitates the quick deployment of applications that you create by providing all the application services that you need. AWS manages the OS, the application server, and the other infrastructure components so that you can focus on developing your application code.

# Choosing the optimal compute service



- The optimal compute service or services that you use will depend on your use case
- Some aspects to consider –
  - What is your application design?
  - What are your usage patterns?
  - Which configuration settings will you want to manage?
- Selecting the wrong compute solution for an architecture can lead to lower performance efficiency
  - A good starting place—Understand the available compute options

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

7

AWS offers many compute services because different use cases benefit from different compute environments. The optimal compute service or services that you use will depend on your use case.

Often, the compute architecture that you use is determined by legacy code. However, that does not mean that you cannot evolve the architecture to take advantage of proven cloud-native designs.

Best practices include:

- Evaluate the available compute options
- Understand the available compute configuration options
- Collect computer-related metrics
- Use the available elasticity of resources
- Re-evaluate compute needs based on metrics

Sometimes, a customer will start with one compute solution and decide to change the design based on their analysis of metrics. If you are interested in seeing an example of how a customer modified their choice of compute services for a particular use case, view this [Inventory Tracking](#) solution video.

Module 6: Compute

## Section 2: Amazon EC2

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



Introducing Section 2: Amazon EC2.

# Amazon Elastic Compute Cloud (Amazon EC2)

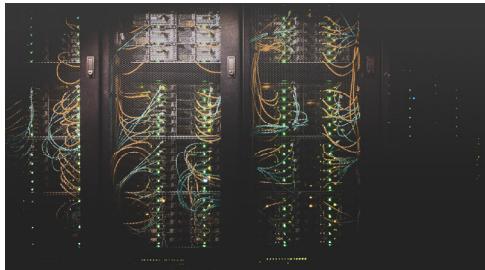
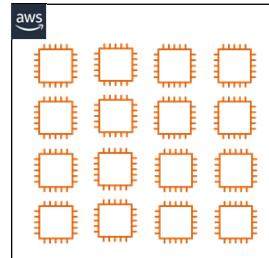


Photo by Taylor Vick on Unsplash

## On-premises servers

### Example uses of Amazon EC2 instances

- ✓ Application server
- ✓ Web server
- ✓ Database server
- ✓ Game server
- ✓ Mail server
- ✓ Media server
- ✓ Catalog server
- ✓ File server
- ✓ Computing server
- ✓ Proxy server



## Amazon EC2 instances



Photo by panumas nikhomkhai from Pexels

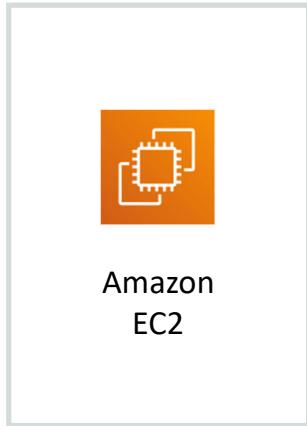
9

**Running servers on-premises** is an expensive undertaking. Hardware must be procured, and this procurement can be based on project plans instead of the reality of how the servers are used. Data centers are expensive to build, staff, and maintain. Organizations also need to permanently provision a sufficient amount of hardware to handle traffic spikes and peak workloads. After traditional on-premises deployments are built, server capacity might be unused and idle for a significant portion of the time that the servers are running, which is wasteful.

Amazon Elastic Compute Cloud (Amazon EC2) provides virtual machines where you can host the same kinds of applications that you might run on a traditional on-premises server. It provides secure, resizable compute capacity in the cloud. EC2 instances can support a variety of workloads. Common uses for EC2 instances include, but are not limited to:

- Application servers
- Web servers
- Database servers
- Game servers
- Mail servers
- Media servers

- Catalog servers
- File servers
- Computing servers
- Proxy servers



- **Amazon Elastic Compute Cloud (Amazon EC2)**
  - Provides **virtual machines**—referred to as **EC2 instances**—in the cloud.
  - Gives you *full control* over the guest operating system (Windows or Linux) on each instance.
- You can launch instances of any size into an Availability Zone anywhere in the world.
  - Launch instances from **Amazon Machine Images (AMIs)**.
  - Launch instances with a few clicks or a line of code, and they are ready in minutes.
- You can control traffic to and from instances.

The **EC2** in Amazon EC2 stands for **Elastic Compute Cloud**:

- **Elastic** refers to the fact that you can easily increase or decrease the number of servers you run to support an application automatically, and you can also increase or decrease the size of existing servers.
- **Compute** refers to reason why most users run servers in the first place, which is to host running applications or process data—actions that require compute resources, including processing power (CPU) and memory (RAM).
- **Cloud** refers to the fact that the EC2 instances that you run are hosted in the cloud.

Amazon EC2 provides virtual machines in the cloud and gives you full administrative control over the Windows or Linux operating system that runs on the instance. Most server operating systems are supported, including: Windows 2008, 2012, 2016, and 2019, Red Hat, SuSE, Ubuntu, and Amazon Linux.

An operating system that runs on a virtual machine is often called *a guest operating system* to distinguish it from the *host operating system*. The host operating system is directly installed on any server hardware that hosts one or more virtual machines.

With Amazon EC2, you can launch any number of instances of any size into any Availability Zone anywhere in the world in a matter of minutes. Instances launch from **Amazon Machine Images (AMIs)**, which are effectively virtual machine *templates*. AMIs are discussed in more detail later in this module.

You can control traffic to and from instances by using security groups. Also, because the

servers run in the AWS Cloud, you can build solutions that take use multiple AWS services.

# Launching an Amazon EC2 instance



This section of the module walks through **nine key decisions** to make when you create an EC2 instance by using the AWS Management Console **Launch Instance Wizard**.

- Along the way, essential Amazon EC2 concepts will be explored.

The screenshot shows the AWS Management Console EC2 Dashboard. The left sidebar includes links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances (selected), Launch Templates, Spot Requests, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images (AMIs), and Bundle Tasks. The main content area displays resource counts: 0 Running Instances, 0 Dedicated Hosts, 0 Volumes, 1 Key Pairs, 0 Elastic IPs, 0 Snapshots, 0 Load Balancers, and 1 Security Groups. A callout box highlights the 'Launch Instance' button. The right sidebar contains sections for Account Attributes (Supported Platforms: VPC, Default VPC: vpc-01c49451cf595b68), Console experiments (Settings), Additional Information (Getting Started Guide, Documentation, All EC2 Resources, Forums, Pricing, Contact Us), and AWS Marketplace (Find free software trial products in the AWS Marketplace from the EC2 Launch Wizard). The bottom of the page includes a feedback link, language selection (English (US)), and copyright information (© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.).

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

11

The first time you launch an Amazon EC2 instance, you will likely use the AWS Management Console Launch Instance Wizard. You will have the opportunity to experience using the Launch Wizard in the **lab** that is in this module.

The **Launch Instance Wizard** makes it easy to launch an instance. For example, if you choose to accept all the default settings, you can skip most of the steps that are provided by the wizard and launch an EC2 instance in as few as six clicks. An example of this process is shown in the **demonstration** at the end of this section.

However, for most deployments you will want to modify the default settings so that the servers you launch are deployed in a way that matches your specific needs.

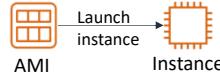
The next series of slides introduce you to the essential choices that you must make when you launch an instance. The slides cover essential concepts that are good to know when you make these choices. These concepts are described to help you understand the options that are available, and the effects of the decisions that you will make.

# 1. Select an AMI



## Choices made using the Launch Instance Wizard:

1. **AMI**
2. **Instance Type**
3. **Network settings**
4. **IAM role**
5. **User data**
6. **Storage options**
7. **Tags**
8. **Security group**
9. **Key pair**



- Amazon Machine Image (AMI)
  - Is a template that is used to create an EC2 instance (which is a **virtual machine, or VM**, that runs in the AWS Cloud)
  - Contains a **Windows** or **Linux** operating system
  - Often also has some **software** pre-installed
- AMI choices:
  - Quick Start – *Linux and Windows AMIs that are provided by AWS*
  - My AMIs – *Any AMIs that you created*
  - AWS Marketplace – *Pre-configured templates from third parties*
  - Community AMIs – *AMIs shared by others; use at your own risk*



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

12

An **Amazon Machine Image (AMI)** provides information that is required to launch an **EC2 instance**. You must specify a source AMI when you launch an instance. You can use different AMIs to launch different types of instances. For example, you can choose one AMI to launch an instance that will become a web server and another AMI to deploy an instance that will host an application server. You can also launch multiple instances from a single AMI.

An AMI includes the following components:

- A **template for the root volume** of the instance. A root volume typically contains an operating system (OS) and everything that was installed in that OS (applications, libraries, etc.). Amazon EC2 copies the template to the root volume of a new EC2 instance, and then starts it.
- **Launch permissions** that control which AWS accounts can use the AMI.
- A **block device mapping** that specifies the volumes to attach to the instance (if any) when it is launched.

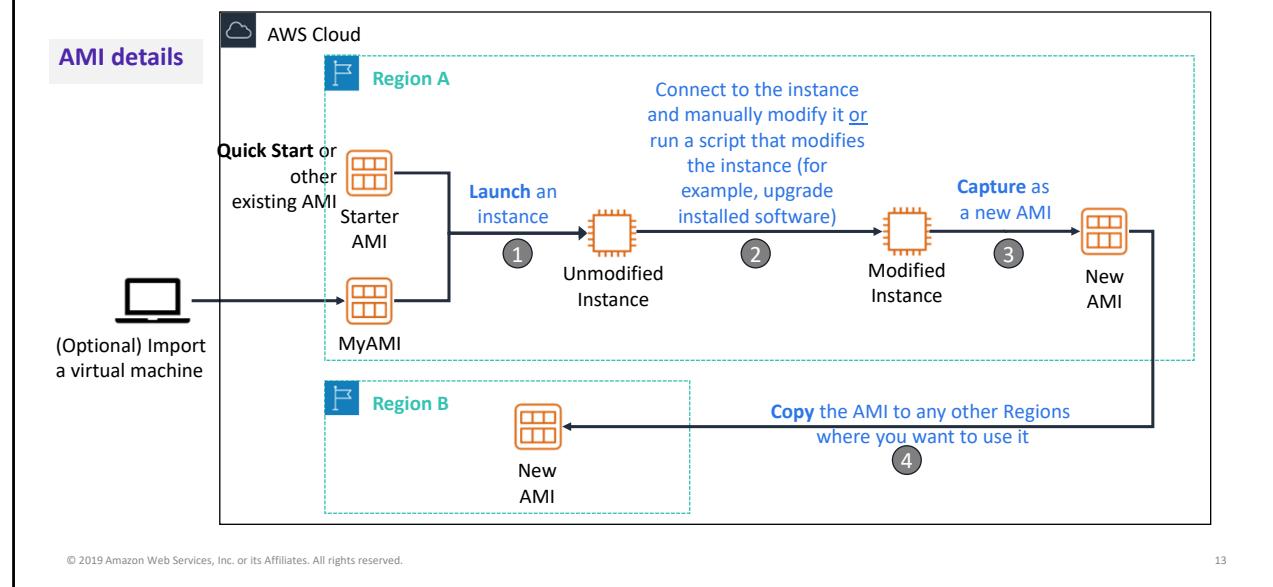
You can choose many AMIs:

- **Quick Start** – AWS offers a number of pre-built AMIs for launching your instances. These

AMIs include many Linux and Windows options.

- **My AMIs** – These AMIs are AMIs that you created.
- **AWS Marketplace** – The AWS Marketplace offers a digital catalog that lists thousands of software solutions. These AMIs can offer specific use cases to help you get started quickly.
- **Community AMIs** – These AMIs are created by people all around the world. These AMIs are not checked by AWS, so use them at your own risk. Community AMIs can offer many different solutions to various problems, but use them with care. Avoid using them in any production or corporate environment.

# Creating a new AMI: Example



An AMI is created from an EC2 instance. You can **import** a virtual machine so that it becomes an EC2 instance, and then save the EC2 instance as an AMI. You can then launch an EC2 instance from that AMI. Alternatively, you can start with an **existing AMI**—such as of the Quick Start AMIs provided by AWS—and create an EC2 instance from it.

Regardless of which options you chose (step 1), you will have what the diagram refers to as *an unmodified instance*. From that instance, you might then create a *golden instance*—that is, a virtual machine that you configured with the specific OS and application settings that you want (step 2)—and then capture that as a new AMI (step 3). When you create an AMI, Amazon EC2 stops the instance, creates a snapshot of its root volume, and finally registers the snapshot as an AMI.

After an AMI is registered, the AMI can be used to launch new instances in the same AWS Region. The new AMI can now be thought of as a new starter AMI. You might want to also copy the AMI to other Regions (step 4), so that EC2 instances can also be launched in those locations.

## 2. Select an instance type



### Choices made using the Launch Instance Wizard:

1. AMI
2. Instance Type
3. Network settings
4. IAM role
5. User data
6. Storage options
7. Tags
8. Security group
9. Key pair

- Consider your use case
  - How will the EC2 instance you create be used?
- The **instance type** that you choose determines –
  - Memory (RAM)
  - Processing power (CPU)
  - Disk space and disk type (Storage)
  - Network performance
- Instance type categories –
  - General purpose
  - Compute optimized
  - Memory optimized
  - Storage optimized
  - Accelerated computing
- Instance types offer *family*, *generation*, and *size*



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

14

After you choose the AMI for launching the instance, you must choose an instance type.

Amazon EC2 provides a selection of **instance types** that are optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity. The different instance types give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more instance sizes, which enable you to scale your resources to the requirements of your target workload.

**Instance type categories** include general purpose, compute optimized, memory optimized, storage optimized, and accelerated computing instances. Each instance type category offers many instance types to choose from.

# EC2 instance type naming and sizes



## Instance type details

### Instance type naming

- Example: **t3.large**
  - T is the family name
  - 3 is the generation number
  - Large is the size

## Example instance sizes

Instance Name	vCPU	Memory (GB)	Storage
t3.nano	2	0.5	EBS-Only
t3.micro	2	1	EBS-Only
t3.small	2	2	EBS-Only
t3.medium	2	4	EBS-Only
t3.large	2	8	EBS-Only
t3.xlarge	4	16	EBS-Only
t3.2xlarge	8	32	EBS-Only

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

15

When you look at an EC2 instance type, you will see that its name has several parts. For example, consider the T type.

T is the **family name**, which is then followed by a number. Here, that number is 3.

The number is the **generation number** of that type. So, a t3 instance is the third generation of the T family. In general, instance types that are of a higher generation are more powerful and provide a better value for the price.

The next part of the name is the **size** portion of the instance. When you compare sizes, it is important to look at the coefficient portion of the size category.

For example, a **t3.2xlarge** has twice the vCPU and memory of a **t3.xlarge**. The t3.xlarge has, in turn, twice the vCPU and memory of a t3.large.

It is also important to note that **network bandwidth** is also tied to the size of the Amazon EC2 instance. If you will run jobs that will be very network-intensive, you might be required to increase the instance specifications to meet your needs.

## Select instance type: Based on use case



### Instance type details

	General Purpose	Compute Optimized	Memory Optimized	Accelerated Computing	Storage Optimized
Instance Types	a1, m4, m5, t2, t3	c4, c5	r4, r5, x1, z1	f1, g3, g4, p2, p3	d2, h1, i3
Use Case	Broad	High performance	In-memory databases	Machine learning	Distributed file systems

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

16

**Instance types** vary in several ways, including: CPU type, CPU or core count, storage type, storage amount, memory amount, and network performance. The chart provides a high-level view of the different instance categories, and which instance type families and generation numbers fit into each category type. Consider a few of the instance types in more detail:

- **T3** instances provide burstable performance **general purpose** instances that provide a baseline level of CPU performance with the ability to burst above the baseline. Use cases for this type of instance include websites and web applications, development environments, build servers, code repositories, microservices, test and staging environments, and line-of-business applications.
- **C5** instances are optimized for **compute-intensive** workloads, and deliver cost-effective high performance at a low price per compute ratio. Use cases include scientific modeling, batch processing, ad serving, highly scalable multiplayer gaming, and video encoding.
- **R5** instances are optimized for memory-intensive applications. Use cases include high-performance databases, data mining and analysis, in-memory databases, distributed web-scale in-memory caches, applications that perform real-time processing of unstructured big data, Apache Hadoop or Apache Spark clusters, and other enterprise

applications.

To learn more about each instance type, see the [Amazon EC2 Instance Types](#) documentation.

# Instance types: Networking features



- The network bandwidth (Gbps) varies by instance type.
  - See [Amazon EC2 Instance Types](#) to compare.
- To maximize networking and bandwidth performance of your instance type:
  - If you have interdependent instances, launch them into a **cluster placement group**.
  - Enable enhanced networking.
- Enhanced networking types are supported on most instance types.
  - See the [Networking and Storage Features](#) documentation for details.
- Enhanced networking types –
  - **Elastic Network Adapter (ENA)**: Supports network speeds of up to 100 Gbps.
  - **Intel 82599 Virtual Function interface**: Supports network speeds of up to 10 Gbps.

In addition to considering the CPU, RAM, and storage needs of your workloads, it is also important to consider your network bandwidth requirements.

Each instance type provides a documented network performance level. For example, an a1.medium instance will provide up to 10 Gbps, but a p3dn.24xlarge instance provides up to 100 Gbps. Choose an instance type that meets your requirements.

When you launch multiple new EC2 instances, Amazon EC2 attempts to place the instances so that they are spread out across the underlying hardware by default. It does this to minimize correlated failures. However, if you want to specify specific placement criteria, you can use **placement groups** to influence the placement of a group of **interdependent** instances to meet the needs of your workload. For example, you might specify that three instances should all be deployed in the same Availability Zone to ensure lower network latency and higher network throughput between instances. See the [Placement Group](#) documentation for details.

Many instance types also enable you to configure enhanced networking to get significantly higher packet per second (PPS) performance, lower delay variation in the arrival of packets over the network (network jitter), and lower latencies. See the [Elastic Network Adapter \(ENA\)](#) documentation for details.

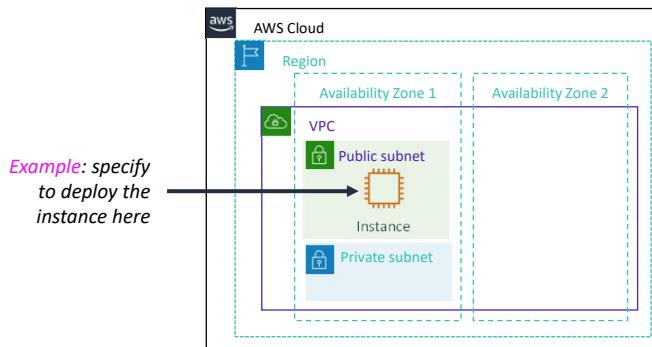
### 3. Specify network settings



#### Choices made by using the Launch Instance Wizard:

1. AMI
2. Instance Type
3. Network settings
4. IAM role
5. User data
6. Storage options
7. Tags
8. Security group
9. Key pair

- Where should the instance be deployed?
  - Identify the VPC and optionally the subnet
- Should a public IP address be automatically assigned?
  - To make it internet-accessible



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

18

After you have chosen an AMI and an instance type, you must specify the network location where the EC2 instance will be deployed. The choice of **Region** must be made before you start the Launch Instance Wizard. Verify that you are in the correct Region page of the Amazon EC2 console before you choose **Launch Instance**.

When you launch an instance in a **default VPC**, AWS will assign it a **public IP address** by default. When you launch an instance into a **nondefault VPC**, the subnet has an attribute that determines whether instances launched into that subnet receive a public IP address from the public IPv4 address pool. By default, AWS will not assign a public IP address to instances that are launched in a nondefault subnet. You can control whether your instance receives a public IP address by either modifying the public IP addressing attribute of your subnet, or by enabling or disabling the public IP addressing feature during launch (which overrides the subnet's public IP addressing attribute).

## 4. Attach IAM role (optional)

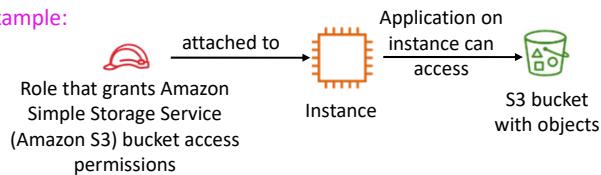


### Choices made by using the Launch Instance Wizard:

1. AMI
2. Instance Type
3. Network settings
4. **IAM role**
5. User data
6. Storage options
7. Tags
8. Security group
9. Key pair

- Will software on the EC2 instance need to interact with other AWS services?
  - If yes, attach an appropriate **IAM Role**.
- An AWS Identity and Access Management (IAM) role that is attached to an EC2 instance is kept in an **instance profile**.
- You are *not* restricted to attaching a role only at instance launch.
  - You can also attach a role to an instance that already exists.

#### Example:



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

19

It is common to use EC2 instances to run an application that must make secure API calls to other AWS services. To support these use cases, AWS enables you to **attach an AWS Identity and Access Management (IAM) role to an EC2 instance**. Without this feature, you might be tempted to place AWS credentials on an EC2 instance so an application that runs on that instance to use. However, you should never store AWS credentials on an EC2 instance. It is highly insecure. Instead, attach an IAM role to the EC2 instance. The IAM role then grants permission to make application programming interface (API) requests to the applications that run on the EC2 instance.

An **instance profile** is a container for an IAM role. If you use the AWS Management Console to create a role for Amazon EC2, the console automatically creates an instance profile and gives it the same name as the role. When you then use the Amazon EC2 console to launch an instance with an IAM role, you can select a role to associate with the instance. In the console, the list that displays is actually a list of instance profile names.

In the example, you see that an IAM role is used to grant permissions to an application that runs on an EC2 instance. The application must access a bucket in Amazon S3.

You can attach an IAM role when you launch the instance, but you can also attach a role to

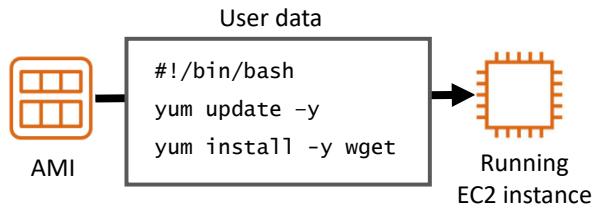
an already running EC2 instance. When you define a role that can be used by an EC2 instance, you define which accounts or AWS services can assume the role. You also define which API actions and resources the application can use after it assumes the role. If you change a role, the change is propagated to all instances that have the role attached to them.

## 5. User data script (optional)



### Choices made by using the Launch Instance Wizard:

1. AMI
2. Instance Type
3. Network settings
4. IAM role
5. **User data**
6. Storage options
7. Tags
8. Security group
9. Key pair



- Optionally specify a user data script at instance launch
- Use **user data** scripts to customize the runtime environment of your instance
  - Script runs the first time the instance starts
- Can be used strategically
  - For example, reduce the number of custom AMIs that you build and maintain

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

20

When you create your EC2 instances, you have the option of passing **user data** to the instance. User data can automate the completion of installations and configurations at instance launch. For example, a user data script might patch and update the instance's operating system, fetch and install software license keys, or install additional software.

In the example user data script, you see a simple three-line **Linux** Bash shell script. The first line indicates that the script should be run by the Bash shell. The second line invokes the Yellowdog Updater, Modified (YUM) utility, which is commonly used in many Linux distributions—such as Amazon Linux, CentOS, and Red Hat Linux—to retrieve software from an online repository and install it. In line two of the example, that command tells YUM to update all installed packages to the latest versions that are known to the software repository that it is configured to access. Line three of the script indicates that the **Wget** utility should be installed. Wget is a common utility for downloading files from the web.

For a **Windows** instance, the user data script should be written in a format that is compatible with a Command Prompt window (batch commands) or with Windows PowerShell. See the [Windows User Data Scripts](#) documentation for details.

When the EC2 instance is created, **the user data script will run with root privileges** during

the final phases of the boot process. On Linux instances, it is run by the cloud-init service. On Windows instances, it is run by the EC2Config or EC2Launch utility. **By default, user data only runs the first time that the instance starts up.** However, if you would like your user data script to run every time the instance is booted, you can create a [Multipurpose Internet Mail Extensions \(MIME\) multipart file](#) user data script (this process is not commonly done).

## 6. Specify storage



### Choices made by using the Launch Instance Wizard:

1. AMI
2. Instance Type
3. Network settings
4. IAM role
5. User data
6. Storage options
7. Tags
8. Security group
9. Key pair

- Configure the **root volume**
  - Where the guest operating system is installed
- Attach **additional storage volumes** (optional)
  - AMI might already include more than one volume
- For each volume, specify:
  - The **size** of the disk (in GB)
  - The **volume type**
    - Different types of solid state drives (SSDs) and hard disk drives (HDDs) are available
  - If the volume will be deleted when the instance is terminated
  - If **encryption** should be used



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

21

When you launch an EC2 instance, you can configure storage options. For example, you can configure the size of the root volume where the guest operating system is installed. You can also attach additional storage volumes when you launch the instance. Some AMIs are also configured to launch more than one storage volume by default to provide storage that is separate from the root volume.

For each volume that your instance will have, you can specify the size of the disks, the volume types, and whether the storage will be retained if the instance is terminated. You can also specify if encryption should be used.

# Amazon EC2 storage options



- **Amazon Elastic Block Store (Amazon EBS)** –
  - **Durable**, block-level storage volumes.
  - You can stop the instance and start it again, and the data will still be there.
- **Amazon EC2 Instance Store** –
  - **Ephemeral** storage is provided on disks that are attached to the host computer where the EC2 instance is running.
  - If the instance stops, data stored here is deleted.
- Other options for storage (not for the root volume) –
  - Mount an **Amazon Elastic File System (Amazon EFS)** file system.
  - Connect to **Amazon Simple Storage Service (Amazon S3)**.

**Amazon Elastic Block Store (Amazon EBS)** is an easy-to-use, high-performance **durable block storage** service that is designed to be used with Amazon EC2 for both throughput- and transaction-intensive workloads. With Amazon EBS, you can choose from four different volume types to balance the optimal price and performance. You can change volume types or increase volume size without disrupting your critical applications, so you can have cost-effective storage when you need it.

**Amazon EC2 Instance Store** provides ephemeral, or temporary, block-level storage for your instance. This storage is located on disks that are physically attached to the host computer. Instance Store works well when you must temporarily store information that changes frequently, such as buffers, caches, scratch data, and other temporary content. You can also use Instance Store for data that is replicated across a fleet of instances, such as a load balanced pool of web servers. If the instances are stopped—either because of user error or a malfunction—the data on the instance store will be deleted.

**Amazon Elastic File System (Amazon EFS)** provides a simple, scalable, fully managed elastic Network File System (NFS) file system for use with AWS Cloud services and on-premises resources. It is built to scale on-demand to petabytes without disrupting applications. It grows and shrinks automatically as you add and remove files, which reduces the need to

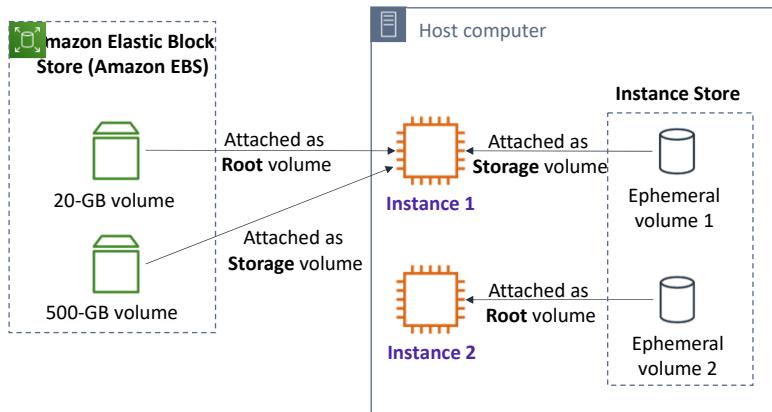
provision and manage capacity to accommodate growth.

**Amazon Simple Storage Service (Amazon S3)** is an object storage service that offers scalability, data availability, security, and performance. You can store and protect any amount of data for a variety of use cases, such as websites, mobile apps, backup and restore, archive, enterprise applications, Internet of Things (IoT) devices, and big data analytics.

# Example storage options



- **Instance 1** characteristics –
  - It has an **Amazon EBS root volume** type for the operating system.
  - **What will happen if the instance is stopped and then started again?**
- **Instance 2** characteristics –
  - It has an **Instance Store root volume** type for the operating system.
  - **What will happen if the instance stops (because of user error or a system malfunction)?**



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

23

Here, you see two examples of how storage options could be configured for EC2 instances.

The **Instance 1** example shows that the root volume—which contains the OS and possibly other data—is stored on Amazon EBS. This instance also has two attached volumes. One volume is a 500-GB Amazon EBS storage volume, and the other volume is an Instance Store volume. **If this instance was stopped and then started again**, the OS would survive and any data that was stored on either the 20-GB Amazon EBS volume or the 500-GB Amazon EBS volume would remain intact. However, any data that was stored on Ephemeral volume 1 would be permanently lost. Instance Store works well for temporarily storing information that changes frequently, such as buffers, caches, scratch data, and other temporary content.

The **Instance 2** example shows that the root volume is on an instance store (Ephemeral volume 2). **An instance with an Instance Store root volume cannot be stopped by an Amazon EC2 API call. It can only be terminated.** However, it could be stopped from within the instance's OS (for example, by issuing a shutdown command)—or it could stop because of OS or disk failure—which would cause the instance to be terminated. If the instance was terminated, all the data that was stored on Ephemeral volume 2 would be lost, including the OS. You would not be able to start the instance again. Therefore, do not rely on Instance Store for valuable, long-term data. Instead, use more durable data storage, such as Amazon EBS, Amazon EFS, or Amazon S3.

If an instance *reboots* (intentionally or unintentionally), data on the instance store root volume does persist.

## 7. Add tags



### Choices made by using the Launch Instance Wizard:

1. AMI
2. Instance Type
3. Network settings
4. IAM role
5. User data
6. Storage options
7. Tags
8. Security group
9. Key pair

- A **tag** is a label that you can assign to an AWS resource.
  - Consists of a *key* and an optional *value*.
- Tagging is how you can attach **metadata** to an EC2 instance.
- Potential benefits of tagging—Filtering, automation, cost allocation, and access control.

### Example:

Key	(128 characters maximum)	Value	(256 characters maximum)
Name	WebServer1		
<b>Add another tag</b>		(Up to 50 tags maximum)	

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

24

A tag is a label that you assign to an AWS resource. Each tag consists of a *key* and an optional *value*, both of which you define. Tags enable you to categorize AWS resources, such as EC2 instances, in different ways. For example, you might tag instances by purpose, owner, or environment.

Tagging is how you can attach metadata to an EC2 instance.

Tag keys and tag values are case-sensitive. For example, a commonly used tag for EC2 instances is a tag key that is called *Name* and a tag value that describes the instance, such as *My Web Server*. The *Name* tag is exposed by default in the Amazon EC2 console **Instances** page. However, if you create a key that is called *name* (with lower-case *n*), it will not appear in the **Name** column for the list of instances (though it will still appear in the instance details panel in the **Tags** tab).

It is a best practice to develop [tagging strategies](#). Using a consistent set of tag keys makes it easier for you to manage your resources. You can also search and filter the resources based on the tags that you add.

## 8. Security group settings



### Choices made by using the Launch Instance Wizard:

1. AMI
2. Instance Type
3. Network settings
4. IAM role
5. User data
6. Storage options
7. Tags
8. **Security group**
9. Key pair

- A **security group** is a **set of firewall rules** that control traffic to the instance.
  - It exists *outside* of the instance's guest OS.
- Create **rules** that specify the **source** and which **ports** that network communications can use.
  - Specify the **port** number and the **protocol**, such as Transmission Control Protocol (TCP), User Datagram Protocol (UDP), or Internet Control Message Protocol (ICMP).
  - Specify the **source** (for example, an IP address or another security group) that is allowed to use the rule.

Type	Protocol	Port Range	Source
SSH	TCP	22	My IP 72.21.198.67/32

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

25

A **security group** acts as a virtual firewall that controls network traffic for one or more instances. When you launch an instance, you can specify one or more security groups; otherwise, the default security group is used.

You can add **rules** to each security group. Rules allow traffic to or from its associated instances. You can modify the rules for a security group at any time, and the new rules will be automatically applied to all instances that are associated with the security group. When AWS decides whether to allow traffic to reach an instance, all the rules from all the security groups that are associated with the instance are evaluated. When you launch an instance in a virtual private cloud (VPC), you must either create a new security group or use one that already exists in that VPC. After you launch an instance, you can change its security groups.

When you **define a rule**, you can specify the allowable source of the network communication (inbound rules) or destination (outbound rules). The **source** can be an IP address, an IP address range, another security group, a gateway VPC endpoint, or anywhere (which means that all sources will be allowed). By default, a **security group** includes an **outbound rule** that allows all **outbound** traffic. You can remove the **rule** and add **outbound rules** that only allow specific **outbound** traffic. If your **security group** has no **outbound rules**, no **outbound** traffic that originates from your instance is allowed.

In the **example rule**, the rule allows Secure Shell (SSH) traffic over Transmission Control

Protocol (TCP) port 22 if the source of the request is *My IP*. The *My IP* IP address is calculated by determining what IP address you are currently connected to the AWS Cloud from when you define the rule.

Network access control lists (network ACLs) can also be used as firewalls to protect subnets in a VPC.

## 9. Identify or create the key pair



### Choices made by using the Launch Instance Wizard:

1. AMI
2. Instance Type
3. Network settings
4. IAM role
5. User data
6. Storage options
7. Tags
8. Security group
9. Key pair

- At instance launch, you specify an existing key pair *or* create a new key pair.
- A **key pair** consists of –
  - A **public key** that AWS stores.
  - A **private key** file that you store.
- It enables secure connections to the instance.
- For **Windows AMIs** –
  - Use the private key to obtain the administrator password that you need to log in to your instance.
- For **Linux AMIs** –
  - Use the private key to use SSH to securely connect to your instance.



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

26

After you specify all the required configurations to launch an EC2 instance, and after you customize any optional EC2 launch wizard configuration settings, you are presented with a **Review Instance Launch** window. If you then choose **Launch**, a dialog asks you to choose an existing key pair, proceed without a key pair, or create a new key pair before you can choose **Launch Instances** and create the EC2 instance.

Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. The technology uses a **public key** to encrypt a piece of data, and then the recipient uses the private key to decrypt the data. The public and private keys are known as a **key pair**. Public–key cryptography enables you to securely access your instances by using a private key instead of a password.

When you launch an instance, you specify a key pair. You can specify an existing key pair or a new key pair that you create at launch. If you create a new key pair, download it and save it in a safe location. This opportunity is the only chance you get to save the private key file.

To connect to a **Windows** instance, use the private key to obtain the administrator password, and then log in to the EC2 instance's Windows Desktop by using Remote Desktop Protocol (RDP). To establish an SSH connection *from* a Windows machine to an

Amazon EC2 instance, you can use a tool such as PuTTY, which will require the same private key.

With **Linux** instances, at boot time, the **public key** content is placed on the instance. An entry is created in within `~/.ssh/authorized_keys`. To log in to your Linux instance (for example, by using SSH), you must provide the **private key** when you establish the connection.

Amazon EC2 console view of a running EC2 instance

The screenshot shows the AWS EC2 Instances Management console. On the left, there's a sidebar with options like EC2 Dashboard, Events, Tags, Reports, Limits, Instances (selected), Launch Templates, Spot Requests, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images (AMIs, Bundle Tasks), and Elastic Block Store (Volumes, Snapshots). The main area has tabs for Launch Instance, Connect, and Actions. A search bar at the top right contains the instance ID: i-092b6f3efba959a53. Below it is a table with columns: Name, Instance ID, Instance Type, Instance State, Status Checks, Public DNS (IPv4), and IPv4 Public IP. One row is shown: Name is i-092b6f3efba959a53, Instance ID is i-092b6f3efba959a53, Instance Type is t2.micro, Instance State is running, Status Checks is Initializing, Public DNS (IPv4) is ec2-54-159-171-63.compute-1.amazonaws.com, and IPv4 Public IP is 54.159.171.63. Below the table, there are tabs for Description, Status Checks, Monitoring, and Tags. The Description tab is selected, showing detailed information about the instance, including its ID, state, type, and various network and VPC details. At the bottom, there are links for Feedback, English (US), and footer text: © 2008–2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Privacy Policy Terms of Use.

After you choose **Launch Instances** and then choose **View Instances**, you will be presented with a screen that looks similar to the example.

Many of the settings that you specified during launch are visible in the **Description** panel.

Information about the available instance includes IP address and DNS address information, the instance type, the unique instance ID that was assigned to the instance, the AMI ID of the AMI that you used to launch the instance, the VPC ID, the subnet ID, and more.

Many of these details provide hyperlinks that you can choose to learn more information about the resources that are relevant to the EC2 instance you launched.

## Another option: Launch an EC2 instance with the AWS Command Line Interface



- EC2 instances can also be created programmatically.
  - This example shows how simple the command can be.
    - This command assumes that the key pair and security group already exist.
    - More options could be specified. See the [AWS CLI Command Reference](#) for details.



AWS Command Line Interface (AWS CLI)

### Example command:

```
aws ec2 run-instances \
--image-id ami-1a2b3c4d \
--count 1 \
--instance-type c3.large \
--key-name MyKeyPair \
--security-groups MySecurityGroup \
--region us-east-1
```

You can also launch EC2 instances programmatically, either by using the AWS Command Line Interface (AWS CLI) or one of the AWS software development kits (SDKs).

In the example AWS CLI command, you see a single command that specifies the minimal information that is needed to launch an instance. The command includes the following information:

- **aws** – Specifies an invocation of the *aws* command line utility.
- **ec2** – Specifies an invocation of the *ec2* service command.
- **run-instances** – Is the subcommand that is being invoked.

The rest of the command specifies several parameters, including:

- **image-id** – This parameter is followed by an AMI ID. All AMIs have a unique AMI ID.
- **count** – You can specify more than one.
- **instance-type** – You can specify the instance type to create (for example) a c3.large instance
- **key-name** – In the example, assume that *MyKeyPair* already exists.
- **security-groups** - In this example, assume that *MySecurityGroup* already exists.
- **region** - AMIs exist in an AWS Region, so you must specify the Region where the AWS CLI will find the AMI and launch the EC2 instance.

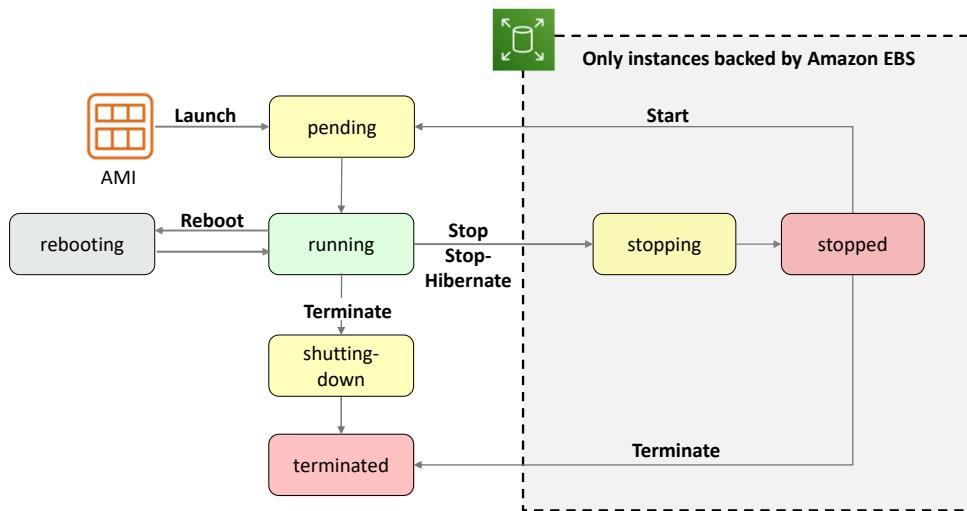
The command should successfully create an EC2 instance if:

- The command is properly formed

- The resources that the command needs already exist
- You have sufficient permissions to run the command
- You have sufficient capacity in the AWS account

If the command is successful, the API responds to the command with the instance ID and other relevant data for your application to use in subsequent API requests.

# Amazon EC2 instance lifecycle



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

29

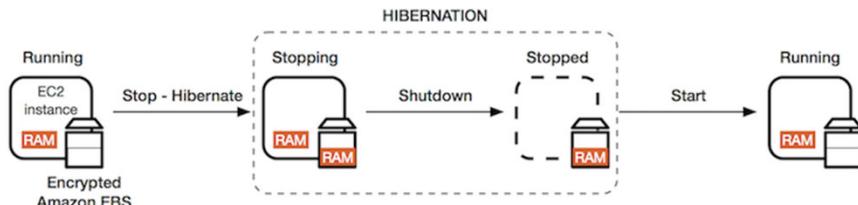
Here, you see the lifecycle of an instance. The arrows show **actions** that you can take and the boxes show the **state** the instance will enter after that action. An instance can be in one of the following states:

- **Pending** – When an instance is first launched from an AMI, or when you start a stopped instance, it enters the *pending* state when the instance is booted and deployed to a host computer. The instance type that you specified at launch determines the hardware of the host computer for your instance.
- **Running** – When the instance is fully booted and ready, it exits the *pending* state and enters the *running* state. You can connect over the internet to your running instance.
- **Rebooting** – AWS recommends you reboot an instance by using the Amazon EC2 console, AWS CLI, or AWS SDKs instead of invoking a reboot from within the guest operating system (OS). A rebooted instance stays on the same physical host, maintains the same public DNS name and public IP address, and if it has **instance store** volumes, it retains the data on those volumes.
- **Shutting down** – This state is an intermediary state between *running* and *terminated*.
- **Terminated** – A terminated instance remains visible in the Amazon EC2 console for a while before the virtual machine is deleted. However, you can't connect to or recover a

terminated instance.

- **Stopping** – Instances that are backed by Amazon EBS can be stopped. They enter the *stopping* state before they attain the fully *stopped* state.
- **Stopped** – A *stopped* instance will not incur the same cost as a *running* instance. Starting a *stopped* instance puts it back into the *pending* state, which moves the instance to a new host machine.

# Instance hibernation option



- Benefits

- It saves the contents from the instance memory (RAM).
- On instance restart, RAM contents are reloaded, previously running processes are resumed.
- You can save on cost in a hibernated state versus a running state (costs are similar to a stopped instance).

- Prerequisites

- Only certain Linux AMIs (such as Amazon Linux 2) and only certain instance families support it.
- Instance must have an encrypted Amazon EBS root volume and a maximum of 150 GB RAM.
- Hibernation must be enabled at instance launch.

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

30

Some instances that are backed by Amazon EBS support **hibernation**. When you hibernate an instance, the guest OS saves the contents from the instance memory (RAM) to your Amazon EBS root volume. When you restart the instance, the root volume is restored to its previous state, the RAM contents are reloaded, and the processes that were previously running on the instance are resumed.

Only certain Linux AMIs that are backed by Amazon EBS and other certain instance types support hibernation. Hibernation also requires that you encrypt the root EBS volume. In addition, you must enable hibernation when the instance is first launched. You cannot enable hibernation on an existing instance that did not originally have hibernation enabled.

For further details about prerequisites and cost, see the [Hibernate Your Linux Instance](#) AWS documentation page.

## Consider using an Elastic IP address



- **Rebooting** an instance will *not* change any IP addresses or DNS hostnames.
- When an instance is **stopped** and then **started** again –
  - The *public IPv4 address and external DNS hostname* will change.
  - The *private IPv4 address and internal DNS hostname* do *not* change.
- If you require a persistent public IP address –
  - Associate an **Elastic IP address** with the instance.
- Elastic IP address characteristics –
  - Can be associated with instances in the Region as needed.
  - Remains allocated to your account until you choose to release it.



Elastic IP Address

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

31

A **public IP address** is an IPv4 address that is reachable from the internet. Each instance that receives a public IP address is also given an external DNS hostname. For example, if the public IP address assigned to the instance is *203.0.113.25*, then the external DNS hostname might be *ec2-203-0-113-25.compute-1.amazonaws.com*.

If you specify that a public IP address should be assigned to your instance, it is assigned from the AWS pool of public IPv4 addresses. The public IP address is not associated with your AWS account. When a public IP address is disassociated from your instance, it is released back into the public IPv4 address pool, and you will not be able to specify that you want to reuse it. AWS releases your instance's public IP address when the instance is stopped or terminated. Your stopped instance receives a new public IP address when it is restarted.

If you require a persistent public IP address, you might want to associate an **Elastic IP address** with the instance. To associate an Elastic IP address, you must first allocate a new Elastic IP address in the Region where the instance exists. After the Elastic IP address is allocated, you can associate the Elastic IP address with an EC2 instance.

By default, all AWS accounts are limited to five (5) Elastic IP addresses per Region because public (IPv4) internet addresses are a scarce public resource. However, this is a soft limit,

and you can request a limit increase (which might be approved).

- **Instance metadata** is data about your instance.
- While you are connected to the instance, you can view it –
  - In a browser: `http://169.254.169.254/latest/meta-data/`
  - In a terminal window: `curl http://169.254.169.254/latest/meta-data/`
- Example retrievable values –
  - Public IP address, private IP address, public hostname, instance ID, security groups, Region, Availability Zone.
  - Any user data specified at instance launch can also be accessed at:  
`http://169.254.169.254/latest/user-data/`
- It can be used to configure or manage a running instance.
  - For example, author a configuration script that reads the metadata and uses it to configure applications or OS settings.

Instance metadata is data about your instance. You can view it while you are connected to the instance. To access it in a browser, go to the following URL:

`http://169.254.169.254/latest/meta-data/`. The data can also be read programmatically, such as from a terminal window that has the cURL utility. In the terminal window, run `curl http://169.254.169.254/latest/meta-data/` to retrieve it. The IP address `169.254.169.254` is a link-local address and it is valid only from the instance.

Instance metadata provides much of the same information about the running instance that you can find in the AWS Management Console. For example, you can discover the public IP address, private IP address, public hostname, instance ID, security groups, Region, Availability Zone, and more.

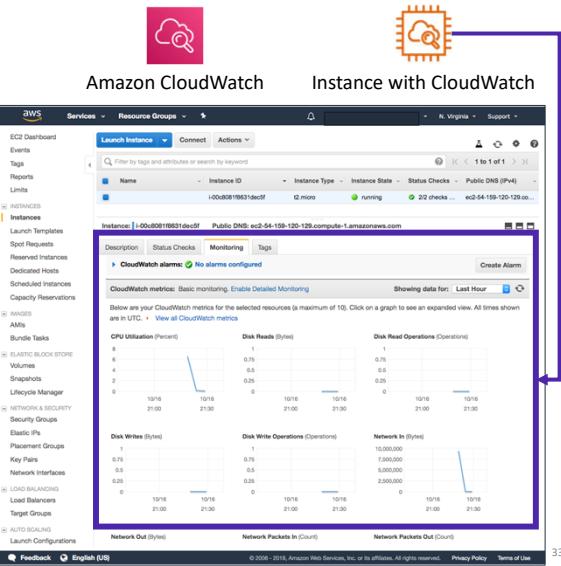
Any user data that is specified at instance launch can also be accessed at the following URL: `http://169.254.169.254/latest/user-data`.

EC2 instance metadata can be used to configure or manage a running instance. For example, you can author a configuration script that accesses the metadata information and uses it to configure applications or OS settings.

# Amazon CloudWatch for monitoring



- Use **Amazon CloudWatch** to monitor EC2 instances
  - Provides near-real-time metrics
  - Provides charts in the Amazon EC2 console **Monitoring** tab that you can view
  - Maintains 15 months of historical data
- **Basic monitoring**
  - Default, no additional cost
  - Metric data sent to CloudWatch every 5 minutes
- **Detailed monitoring**
  - Fixed monthly rate for seven pre-selected metrics
  - Metric data delivered every 1 minute



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

33

You can monitor your instances by using Amazon CloudWatch, which collects and processes raw data from Amazon EC2 into readable, near-real-time metrics. These statistics are recorded for a period of 15 months, so you can access historical information and gain a better perspective on how your web application or service is performing.

By default, Amazon EC2 provides **basic monitoring**, which sends metric data to CloudWatch in 5-minute periods. To send metric data for your instance to CloudWatch in 1-minute periods, you can enable **detailed monitoring** on the instance. For more information, see [Enable or Disable Detailed Monitoring for Your Instances](#).

The Amazon EC2 console displays a series of graphs based on the raw data from Amazon CloudWatch. Depending on your needs, you might prefer to get data for your instances from Amazon CloudWatch instead of through the graphs in the console. By default, Amazon CloudWatch does not provide RAM metrics for EC2 instances, though that is an option that you can configure if you want to CloudWatch to collect that data.

## Section 2 key takeaways



34

- Amazon EC2 enables you to run Windows and Linux virtual machines in the cloud.
- You launch EC2 instances from an AMI template into a VPC in your account.
- You can choose from many instance types. Each instance type offers different combinations of CPU, RAM, storage, and networking capabilities.
- You can configure security groups to control access to instances (specify allowed ports and source).
- User data enables you to specify a script to run the first time that an instance launches.
- Only instances that are backed by Amazon EBS can be stopped.
- You can use Amazon CloudWatch to capture and review metrics on EC2 instances.

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

Some key takeaways from this section of the module include:

- Amazon EC2 enables you to run Windows and Linux virtual machines in the cloud.
- You launch EC2 instances from an AMI template into a VPC in your account.
- You can choose from many instance types. Each instance type offers different combinations of CPU, RAM, storage, and networking capabilities.
- You can configure security groups to control access to instances (specify allowed ports and source).
- User data enables you to specify a script to run the first time that an instance launches.
- Only instances that are backed by Amazon EBS can be stopped.
- You can use Amazon CloudWatch to capture and review metrics on EC2 instances.

# Recorded Amazon EC2 demonstration

35



## Set up demo

Amazon Elastic Compute Cloud  
(Amazon EC2)

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

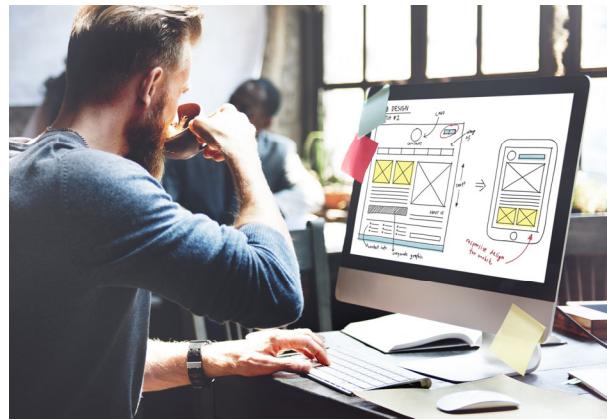
Now, take a moment to watch the [EC2 Demo](#). The recording runs just over 3 minutes and reinforces some of the concepts that were discussed in this section of the module.

The demonstration shows:

- How to use the AWS Management Console to launch an Amazon EC2 instance (with all the default instance settings accepted).
- How to connect to the Windows instance by using a Remote Desktop client and the key pair that was identified during instance launch to decrypt the Windows password for login.
- How to terminate the instance after it is no longer needed.

## Lab 3: Introduction to Amazon EC2

36



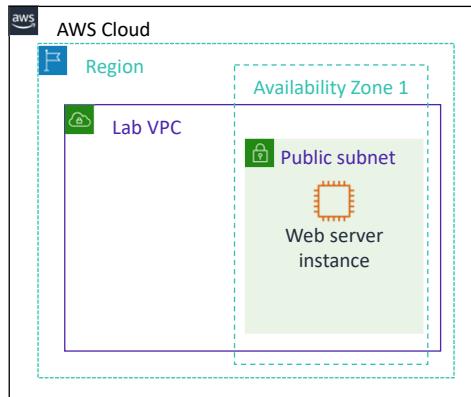
© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Introducing Lab 3: Introduction to Amazon EC2. This lab provides hands-on practice with launching, resizing, managing, and monitoring an Amazon EC2 instance.

## Lab 3 scenario



In this lab, you will launch and configure your first virtual machine that runs on Amazon EC2.



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

37

### Introducing Lab 3: Introduction to Amazon EC2.

In this lab, you will launch and configure a virtual machine that runs on Amazon EC2.

## Lab 3: Tasks



- Task 1 – Launch Your Amazon EC2 Instance
- Task 2 – Monitor Your Instance
- Task 3 – Update Your Security Group and Access the Web Server
- Task 4 – Resize Your Instance: Instance Type and EBS Volume
- Task 5 – Explore EC2 Limits
- Task 6 – Test Termination Protection

In this hands-on lab, you will:

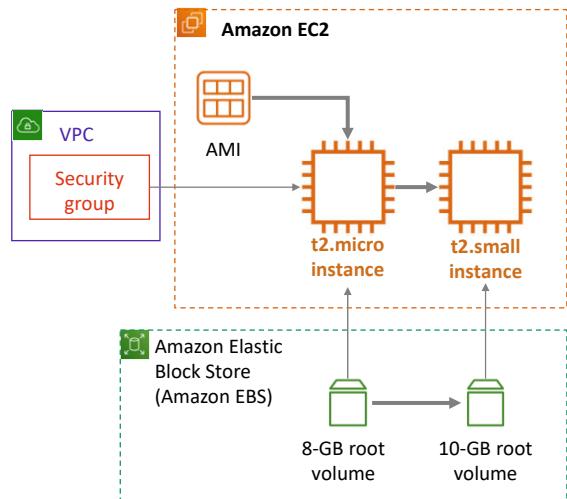
- Launch Your Amazon EC2 Instance
- Monitor Your Instance
- Update Your Security Group and Access the Web Server
- Resize Your Instance: Instance Type and EBS Volume
- Explore EC2 Limits
- Test Termination Protection

## Lab 3: Final product



By the end of the lab, you will have:

1. Launched an instance that is configured as a web server
2. Viewed the instance system log
3. Reconfigured a security group
4. Modified the instance type and root volume size



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

39

By the end of the lab, you will have:

1. Launched an instance that is configured as a web server
2. Viewed the instance system log
3. Reconfigured a security group
4. Modified the instance type and root volume size



~ 35 minutes

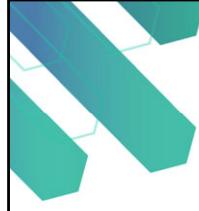


## Begin Lab 3: Introduction to Amazon EC2

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

40

It is now time to start the lab.



## Lab debrief: Key takeaways

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

41

The instructor will lead a conversation about the key takeaways from the lab after you have completed it.

## Activity: Amazon EC2

42



Photo by Pixabay from Pexels.

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

In this educator-led activity, you will discuss the advantages and disadvantages of using Amazon EC2 versus using a managed service like Amazon Relational Database Service (Amazon RDS).

## Activity: Gather information



Amazon EC2

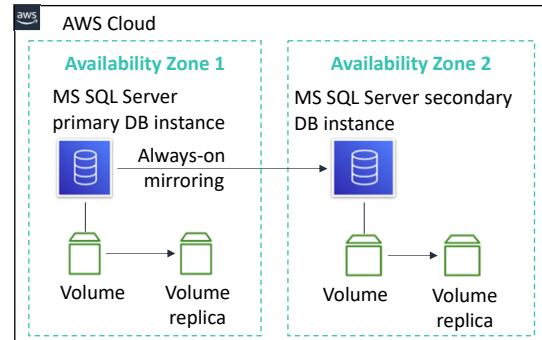
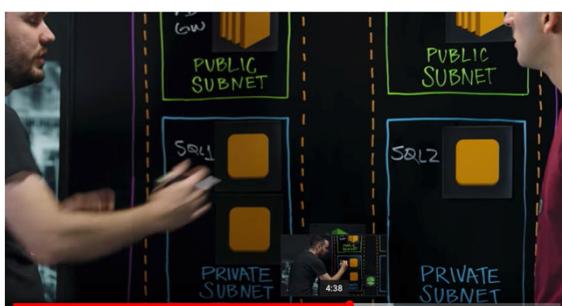


Amazon RDS



AWS Quick Starts

Automated, gold-standard deployments in the AWS Cloud



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

43

The objective of this activity is to demonstrate that you understand the differences between building a deployment that uses Amazon EC2 and using a fully managed service, such as Amazon RDS, to deploy your solution. At the end of this activity, you should be prepared to discuss the advantages and disadvantages of deploying Microsoft SQL Server on Amazon EC2 versus deploying it on Amazon RDS.

The educator will ask you to:

1. Watch an 8-minute [video](#) that explains the benefits of deploying Microsoft SQL Server on Amazon EC2 by using the [AWS Quick Start – SQL Server Reference Architecture](#) deployment. You are encouraged to take notes.
2. Read a [blog post](#) about the benefits of running Microsoft SQL Server on Amazon RDS. You are again encouraged to take notes.
3. Participate in the class conversation about the questions posed on the next slide.

## Activity: Check your understanding



1. Between Amazon EC2 or Amazon RDS, which provides a managed service? What does *managed service* mean?
  - **ANSWER:** Amazon RDS provides a managed service. Amazon RDS handles provisioning, installation and patching, automated backups, restoring snapshots from points in time, high availability, and monitoring.
2. Name at least one advantage of deploying Microsoft SQL Server on Amazon EC2 instead of Amazon RDS.
  - **ANSWER:** Amazon EC2 offers complete control over every configuration, the OS, and the software stack.
3. What advantage does the Quick Start provide over a manual installation on Amazon EC2?
  - **ANSWER:** The Quick Start is a reference architecture with proven best practices built into the design.
4. Which deployment option offers the best approach for all use cases?
  - **ANSWER:** Neither. The correct deployment option depends on your specific needs.
5. Which approach costs more: using Amazon EC2 or using Amazon RDS?
  - **ANSWER:** It depends. Managing the database deployment on Amazon EC2 requires more customer oversight and time. If time is your priority, then Amazon RDS might be less expensive. If you have in-house expertise, Amazon EC2 might be more cost-effective.

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

44

The educator will lead the class in a conversation as each question is revealed. Then, the educator will display the written suggested responses and you can discuss these points further.

Regarding **question 5**, the answer was based on the information that is listed on the AWS Pricing pages as of October, 2019.

- For **Amazon RDS**, you pay \$0.977 per hour if you run Microsoft SQL Server based on these parameters:
  - Instance – Standard (Single-AZ) instance
  - Instance size – db.m5.large
  - Region – US East (Ohio)
  - Pricing – On-Demand Instance
- For **Amazon EC2**, you pay \$0.668 per hour if you run Microsoft SQL Server based on these parameters:
  - Instance – Windows instance
  - Instance size – m5.large
  - Region – US East (Ohio)
  - Pricing – On-Demand Instance

As you consider cost, do not forget to include the cost of labor. For example, keep in mind that with a standard Single-AZ Amazon RDS deployment—which is the basis of the example price reference—automated backups are provided. With Amazon RDS, if a DB instance

component failed and a user-initiated restore operation is required, you would have a restorable backup that you could use. If you run the database on Amazon EC2, you could configure an equally robust backup procedure for Microsoft SQL Server. However, it would take time, knowledge, and technical skill to build the solution. You would also need to pre-configure the solution *before* you encounter the situation where you need it. For these reasons, when you consider the needs of your deployments holistically, you might find that using Amazon RDS is less expensive than using Amazon EC2. However, if you have skilled database administrators on staff—and you also have very specific deployment requirements that make it preferable for you to have total control over all aspects of the deployment—you could use Amazon EC2. In this case, you might find Amazon EC2 to be the more cost-effective solution.

Module 6: Compute

## Section 3: Amazon EC2 cost optimization

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



Introducing Section 3: Amazon EC2 cost optimization.

# Amazon EC2 pricing models



## On-Demand Instances

- Pay by the hour
- No long-term commitments.
- Eligible for the [AWS Free Tier](#).

## Dedicated Hosts

- A physical server with EC2 instance capacity fully dedicated to your use.

## Dedicated Instances

- Instances that run in a VPC on hardware that is dedicated to a single customer.

## Reserved Instances

- Full, partial, or no upfront payment for instance you reserve.
- Discount on hourly charge for that instance.
- 1-year or 3-year term.

## Scheduled Reserved Instances

- Purchase a capacity reservation that is always available on a recurring schedule you specify.
- 1-year term.

## Spot Instances

- Instances run as long as they are available and your bid is above the Spot Instance price.
- They can be interrupted by AWS with a 2-minute notification.
- Interruption options include terminated, stopped or hibernated.
- Prices can be significantly less expensive compared to On-Demand Instances
- Good choice when you have flexibility in when your applications can run.

*Per second billing* available for On-Demand Instances, Reserved Instances, and Spot Instances that run Amazon Linux or Ubuntu.

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

46

Amazon offers different pricing models to choose from when you want to run EC2 instances.

**Per second billing** is only available for On-Demand Instances, Reserved Instances, and Spot Instances that run Amazon Linux or Ubuntu.

**On-Demand** Instances are eligible for the [AWS Free Tier](#). They have the lowest upfront cost and the most flexibility. There are no upfront commitments or long-term contracts. It is a good choice for applications with short-term, spiky, or unpredictable workloads.

**Dedicated Hosts** are physical servers with instance capacity that is dedicated to your use. They enable you to use your existing per-socket, per-core, or per-VM software licenses, such as for Microsoft Windows or Microsoft SQL Server.

**Dedicated Instances** are instances that run in a virtual private cloud (VPC) on hardware that's dedicated to a single customer. They are physically isolated at the host hardware level from instances that belong to other AWS accounts.

**Reserved Instance** enable you to reserve computing capacity for 1-year or 3-year term with lower hourly running costs. The discounted usage price is fixed for as long as you own the

Reserved Instance. If you expect consistent, heavy use, they can provide substantial savings compared to On-Demand Instances.

**Scheduled Reserved Instances** enable you to purchase capacity reservations that recur on a daily, weekly, or monthly basis, with a specified duration, for a 1-year term. You pay for the time that the instances are scheduled, even if you do not use them.

**Spot Instances** enable you to bid on unused EC2 instances, which can lower your costs. The hourly price for a Spot Instance fluctuates depending on supply and demand. Your Spot Instance runs whenever your bid exceeds the current market price.

## Amazon EC2 pricing models: Benefits



On-Demand Instances	Spot Instances	Reserved Instances	Dedicated Hosts
<ul style="list-style-type: none"><li>Low cost and flexibility</li></ul>	<ul style="list-style-type: none"><li>Large scale, dynamic workload</li></ul>	<ul style="list-style-type: none"><li>Predictability ensures compute capacity is available when needed</li></ul>	<ul style="list-style-type: none"><li>Save money on licensing costs</li><li>Help meet compliance and regulatory requirements</li></ul>

Each Amazon EC2 pricing model provides a different set of benefits.

**On-Demand Instances** offer the most flexibility, with no long-term contract and low rates.

**Spot Instances** provide large scale at a significantly discounted price.

**Reserved Instances** are a good choice if you have predictable or steady-state compute needs (for example, an instance that you know you want to keep running most or all of the time for months or years).

**Dedicated Hosts** are a good choice when you have licensing restrictions for the software you want to run on Amazon EC2, or when you have specific compliance or regulatory requirements that preclude you from using the other deployment options.

# Amazon EC2 pricing models: Use cases



Spiky Workloads



Time-Insensitive Workloads



Steady-State Workloads



Highly Sensitive Workloads

On-Demand Instances	Spot Instances	Reserved Instances	Dedicated Hosts
<ul style="list-style-type: none"><li>Short-term, spiky, or unpredictable workloads</li><li>Application development or testing</li></ul>	<ul style="list-style-type: none"><li>Applications with flexible start and end times</li><li>Applications only feasible at very low compute prices</li><li>Users with urgent computing needs for large amounts of additional capacity</li></ul>	<ul style="list-style-type: none"><li>Steady state or predictable usage workloads</li><li>Applications that require reserved capacity, including disaster recovery</li><li>Users able to make upfront payments to reduce total computing costs even further</li></ul>	<ul style="list-style-type: none"><li>Bring your own license (BYOL)</li><li>Compliance and regulatory restrictions</li><li>Usage and licensing tracking</li><li>Control instance placement</li></ul>

Here is a review of some use cases for the various pricing options.

**On-Demand Instance** pricing works well for spiky workloads or if you only need to test or run an application for a short time (for example, during application development or testing). Sometimes, your workloads are unpredictable, and On-Demand Instances are a good choice for these cases.

**Spot Instances** are a good choice if your applications can tolerate interruption with a 2-minute warning notification. By default, instances are terminated, but you can configure them to stop or hibernate instead. Common use cases include fault-tolerant applications such as web servers, API backends, and big data processing. Workloads that constantly save data to persistent storage (such as Amazon S3) are also good candidates.

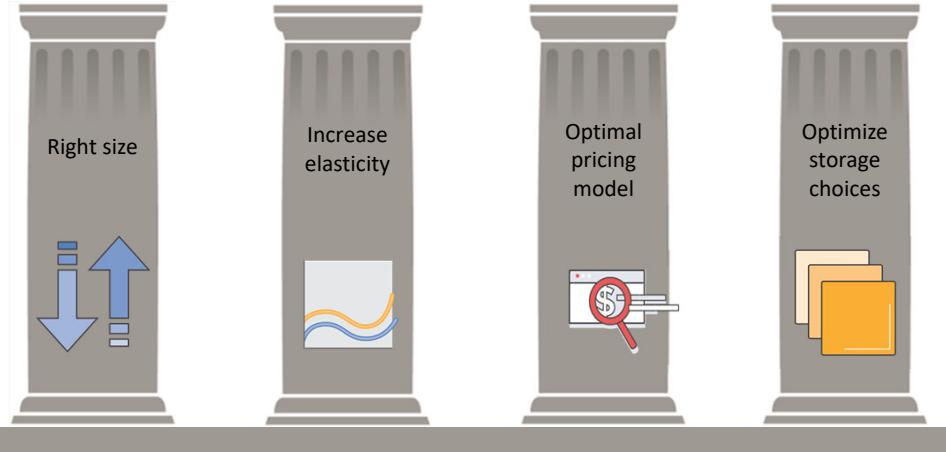
**Reserved Instances** are a good choice when you have long-term workloads with predictable usage patterns, such as servers that you know you will want to run in a consistent way over many months.

**Dedicated Hosts** are a good choice when you have existing per-socket, per-core, or per-VM software licenses, or when you must address specific corporate compliance and regulatory requirements.

# The four pillars of cost optimization



## Cost Optimization



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

49

To optimize costs, you must consider four consistent, powerful drivers:

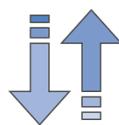
- **Right-size** – Choose the right balance of instance types. Notice when servers can be either sized down or turned off, and still meet your performance requirements.
- **Increase elasticity** – Design your deployments to reduce the amount of server capacity that is idle by implementing deployments that are elastic, such as deployments that use automatic scaling to handle peak loads.
- **Optimal pricing model** – Recognize the available pricing options. Analyze your usage patterns so that you can run EC2 instances with the right mix of pricing options.
- **Optimize storage choices** – Analyze the storage requirements of your deployments. Reduce unused storage overhead when possible, and choose less expensive storage options if they can still meet your requirements for storage performance.

# Pillar 1: Right size



## Pillars:

1. Right size
2. Increase elasticity
3. Optimal pricing model
4. Optimize storage choices



### ✓ Provision instances to match the need

- CPU, memory, storage, and network throughput
- Select appropriate [instance types](#) for your use

### ✓ Use Amazon CloudWatch metrics

- How idle are instances? When?
- Downsize instances

### ✓ Best practice: Right size, then reserve

First, consider right-sizing. AWS offers approximately 60 instance types and sizes. The wide choice of options enables customers to select the instance that best fits their workload. It can be difficult to know where to start and what instance choice will prove to be the best, from both a technical perspective and a cost perspective. **Right-sizing** is the process of reviewing deployed resources and looking for opportunities to downsize when possible.

#### To right-size:

- **Select** the cheapest instance available that still meets your performance requirements.
- **Review** CPU, RAM, storage, and network utilization to identify instances that could be downsized. You might want to provision a variety of instance types and sizes in a test environment, and then test your application on those different test deployments to identify which instances offer the best performance-to-cost ratio. For right-sizing, use techniques such as load testing to your advantage.
- **Use** Amazon CloudWatch metrics and set up custom metrics. A metric represents a time-ordered set of values that are published to CloudWatch (for example, the CPU usage of a particular EC2 instance). Data points can come from any application or business activity for which you collect data.

## Pillar 2: Increase elasticity



### Pillars:

1. Right-Size
2. Increase Elasticity
3. Optimal pricing model
4. Optimize storage choices



- ✓ Stop or hibernate Amazon EBS-backed instances that are not actively in use
  - Example: non-production development or test instances
- ✓ Use automatic scaling to match needs based on usage
  - Automated and time-based elasticity

One form of **elasticity** is to create, start, or use EC2 instances when they are needed, but then to turn them off when they are not in use. Elasticity is one of the central tenets of the cloud, but customers often go through a learning process to operationalize elasticity to drive cost savings.

The easiest way for large customers to embrace elasticity is to look for resources that look like good candidates for stopping or hibernating, such as non-production environments, development workloads, or test workloads. For example, if you run development or test workloads in a single time zone, you can easily turn off those instances outside of business hours and thus reduce runtime costs by perhaps 65 percent. The concept is similar to why there is a light switch next to the door, and why most offices encourage employees *to turn off the lights on their way out of the office each night*.

For production workloads, configuring more precise and granular automatic scaling policies can help you take advantage of horizontal scaling to meet peak capacity needs and to not pay for peak capacity all the time.

As a rule of thumb, you should target 20–30 percent of your Amazon EC2 instances to run as On-Demand Instances or Spot Instances, and you should also actively look for ways to maximize elasticity.

## Pillar 3: Optimal pricing model



### Pillars:

1. Right-Size
2. Increase Elasticity
- 3. Optimal pricing model**
4. Optimize storage choices



- ✓ Leverage the right pricing model for your use case
  - Consider your usage patterns
- ✓ Optimize and *combine* purchase types
- ✓ Examples:
  - Use **On-Demand Instances** and **Spot Instances** for variable workloads
  - Use **Reserved Instances** for predictable workloads
- ✓ Consider serverless solutions (AWS Lambda)

AWS provides a number of pricing models for Amazon EC2 to help customers save money. The models available were discussed in detail earlier in this module. Customers can combine multiple purchase types to optimize pricing based on their current and forecast capacity needs.

Customers are also encouraged to consider their application architecture. For example, does the functionality provided by your application need to run on an EC2 virtual machine? Perhaps by making use of the AWS Lambda service instead, you could significantly decrease your costs.

AWS Lambda is discussed later in this module.

## Pillar 4: Optimize storage choices



### Pillars:

1. Right-Size
2. Increase Elasticity
3. Optimal pricing model
4. **Optimize storage choices**



- ✓ Reduce costs while maintaining storage performance and availability
- ✓ Resize EBS volumes
- ✓ Change EBS volume types
  - ✓ Can you meet performance requirements with less expensive storage?
  - ✓ Example: [Amazon EBS Throughput Optimized HDD \(st1\)](#) storage typically costs half as much as the default [General Purpose SSD \(gp2\)](#) storage option.
- ✓ Delete EBS snapshots that are no longer needed
- ✓ Identify the most appropriate destination for specific types of data
  - ✓ Does the application need the instance to reside on Amazon EBS?
  - ✓ Amazon S3 storage options with lifecycle policies can reduce costs

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

53

Customers can also reduce storage costs. When you launch EC2 instances, different instance types offer different storage options. It is a best practice to try to reduce costs while also maintaining storage performance and availability.

One way you can accomplish this is by **resizing EBS volumes**. For example, if you originally provisioned a 500-GB volume for an EC2 instance that will only need a maximum of 20 GB of storage space, you can reduce the size of the volume and save on costs.

There are also a variety of **EBS volume types**. Choose the least expensive type that still meets your performance requirements. For example, Amazon EBS Throughput Optimized HDD (st1) storage typically costs half as much as the default General Purpose SSD (gp2) storage option. If an st1 drive will meet the needs of your workload, take advantage of the cost savings.

Customers often use **EBS snapshots** to create data backups. However, some customers forget to delete snapshots that are no longer needed. Delete these unneeded snapshots to save on costs.

Finally, try to identify the most **appropriate destination for specific types of data**. Does your application need the data it uses to reside on Amazon EBS? Would the application run equally as well if it used Amazon S3 for storage instead? Configuring data lifecycle policies can also reduce costs. For example, you might automate the migration of older infrequently

accessed data to cheaper storage locations, such as Amazon Simple Storage Service Glacier.

## Measure, monitor, and improve



- Cost optimization is an ongoing process.



- Recommendations –

- Define and enforce **cost allocation tagging**.
- Define metrics, set targets, and review regularly.
- Encourage teams to **architect for cost**.
- Assign the responsibility of optimization to an individual or to a team.



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

54

If it is done correctly, cost optimization is not a one-time process that a customer completes. Instead, by routinely measuring and analyzing your systems, you can continually improve and adjust your costs.

**Tagging** helps provide information about *what resources* are being used *by whom* and *for what purpose*. You can activate cost allocation tags in the Billing and Cost Management console, and AWS can generate a cost allocation report with usage and costs grouped by your active tags. Apply tags that represent business categories (such as cost centers, application names, or owners) to organize your costs across multiple services.

**Encourage teams to architect for cost.** AWS Cost Explorer is a free tool that you can use to view graphs of your costs. You can use Cost Explorer to see patterns in how much you spend on AWS resources over time, identify areas that need further inquiry, and see trends that you can use to understand your costs.

Use AWS services such as **AWS Trusted Advisor**, which provides real-time guidance to help you provision resources that follow AWS best practices.

Cost-optimization efforts are typically more successful when the responsibility for cost optimization is assigned to an individual or to a team.

## Section 3 key takeaways



55



- **Amazon EC2 pricing models** include On-Demand Instances, Reserved Instances, Spot Instances, Dedicated Instances, and Dedicated Hosts.
- **Spot Instances** can be interrupted with a 2-minute notification. However, they can offer significant cost savings over On-Demand Instances.
- The **four pillars of cost optimization** are:
  - Right size
  - Increase elasticity
  - Optimal pricing model
  - Optimize storage choices

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

Some key takeaways from this section of the module are:

- **Amazon EC2 pricing models** include On-Demand Instances, Reserved Instances, Spot Instances, Dedicated Instances, and Dedicated Hosts. Per second billing is available for On-Demand Instances, Reserved Instances, and Spot Instances that use only Amazon Linux and Ubuntu.
- **Spot Instances** can be interrupted with a 2-minute notification. However, they can offer significant cost savings over On-Demand Instances.
- The **four pillars of cost optimization** are –
  - Right size
  - Increase elasticity
  - Optimal pricing model
  - Optimize storage choices

Module 6: Compute

## Section 4: Container services

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



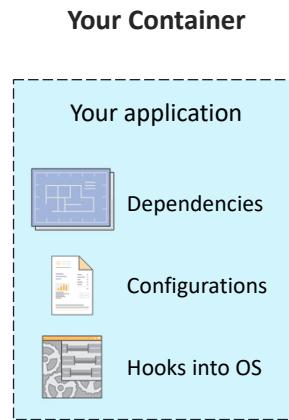
Introducing Section 4: Container services.

# Container basics



- **Containers** are a method of **operating system virtualization**.

- Benefits –
  - Repeatable.
  - Self-contained environments.
  - Software runs the same in different environments.
    - Developer's laptop, test, production.
  - Faster to launch and stop or terminate than virtual machines



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

57

Containers are a method of **operating system virtualization** that enables you to run an application and its dependencies in resource-isolated processes. By using containers, you can easily package an application's code, configurations, and dependencies into easy-to-use building blocks that deliver environmental consistency, operational efficiency, developer productivity, and version control.

Containers are smaller than virtual machines, and do not contain an entire operating system. Instead, containers *share a virtualized operating system* and run as resource-isolated processes, which ensure quick, reliable, and consistent deployments. Containers hold everything that the software needs to run, such as libraries, system tools, code, and the runtime.

Containers deliver **environmental consistency** because the application's code, configurations, and dependencies are packaged into a single object.

In terms of space, container images are usually an order of magnitude smaller than virtual machines. Spinning up a container happens in hundreds of milliseconds. Thus, by using containers, you can use a fast, portable, and infrastructure-agnostic environments.

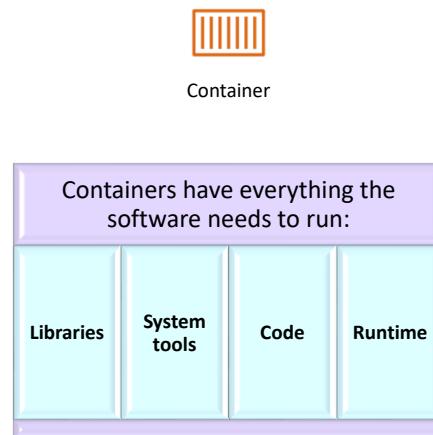
Containers can help ensure that applications deploy quickly, reliably, and consistently, regardless of deployment environment. Containers also give you more granular control

over resources, which gives your infrastructure improved efficiency.

# What is Docker?



- **Docker** is a software platform that enables you to build, test, and deploy applications quickly.
- You run containers on Docker.
  - Containers are created from a template called an *image*.
- A **container** has everything a software application needs to run.



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

58

**Docker** is a software platform that packages software (such as applications) into containers.

Docker is installed on each server that will host containers, and it provides simple commands that you can use to build, start, or stop containers.

By using Docker, you can quickly deploy and scale applications into any environment.

Docker is best used as a solution when you want to:

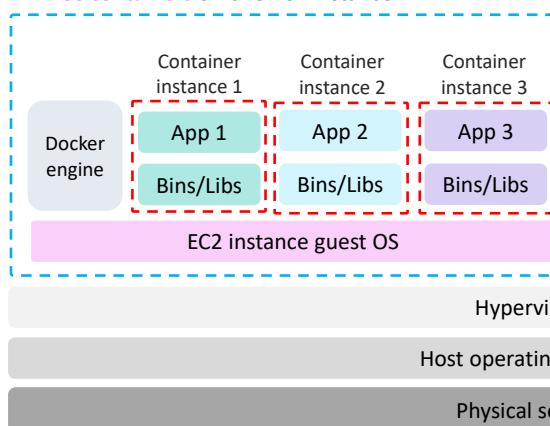
- Standardize environments
- Reduce conflicts between language stacks and versions
- Use containers as a service
- Run microservices using standardized code deployments
- Require portability for data processing

# Containers versus virtual machines

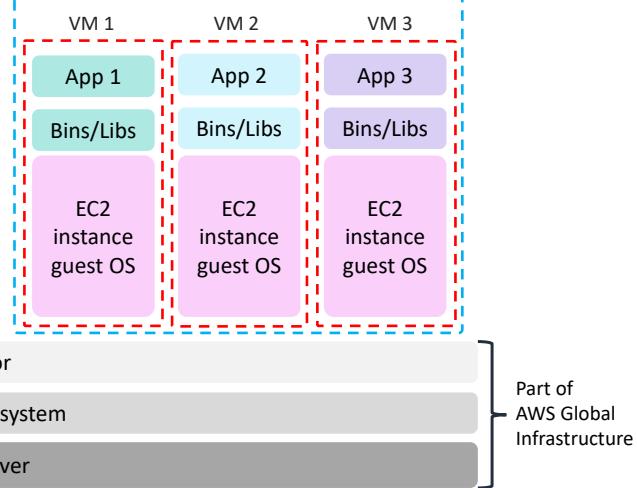


## Example

Three containers on one EC2 instance



Three virtual machines on three EC2 instances



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

59

Many people who are first introduced to the concept of a container think that containers are exactly like virtual machines. However, the differences are in the details. One significant difference is that virtual machines run directly on a hypervisor, but containers can run on any Linux OS if they have the appropriate kernel feature support and the Docker daemon is present. This makes containers very portable. Your laptop, your VM, your EC2 instance, and your bare metal server are all potential hosts where you can run a container.

**The right of the diagram has a virtual machine (VM)-based deployment.** Each of the three EC2 instances runs directly on the hypervisor that is provided by the AWS Global Infrastructure. Each EC2 instance runs a virtual machine. In this VM-based deployment, each of the three apps runs on its own VM, which provides process isolation.

**The left of the diagram has a container-based deployment.** There is only one EC2 instance that runs a virtual machine. The Docker engine is installed on the Linux guest OS of the EC2 instance, and there are three containers. In this container-based deployment, each app runs in its own container (which provides process isolation), but all the containers run on a single EC2 instance. The processes that run in the containers communicate directly to the kernel in the Linux guest OS and are largely unaware of their container silo. The Docker engine is present to manage how the containers run on the Linux guest OS, and it also provides essential management functions throughout the container lifecycle.

In an actual container-based deployment, a large EC2 instance could run hundreds of

containers.

# Amazon Elastic Container Service (Amazon ECS)



- Amazon Elastic Container Service (**Amazon ECS**) –
  - A highly scalable, fast, container management service
- Key benefits –
  - Orchestrates the running of Docker containers
  - Maintains and scales the fleet of nodes that run your containers
  - Removes the complexity of standing up the infrastructure
- Integrated with features that are familiar to Amazon EC2 service users –
  - Elastic Load Balancing
  - Amazon EC2 security groups
  - Amazon EBS volumes
  - IAM roles



Amazon Elastic  
Container Service

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

60

Given what you now know about containers, you might think that you could launch one or more Amazon EC2 instances, install Docker on each instance, and manage and run the Docker containers on those Amazon EC2 instances yourself. While that is an option, AWS provides a service called Amazon Elastic Container Service (Amazon ECS) that simplifies container management.

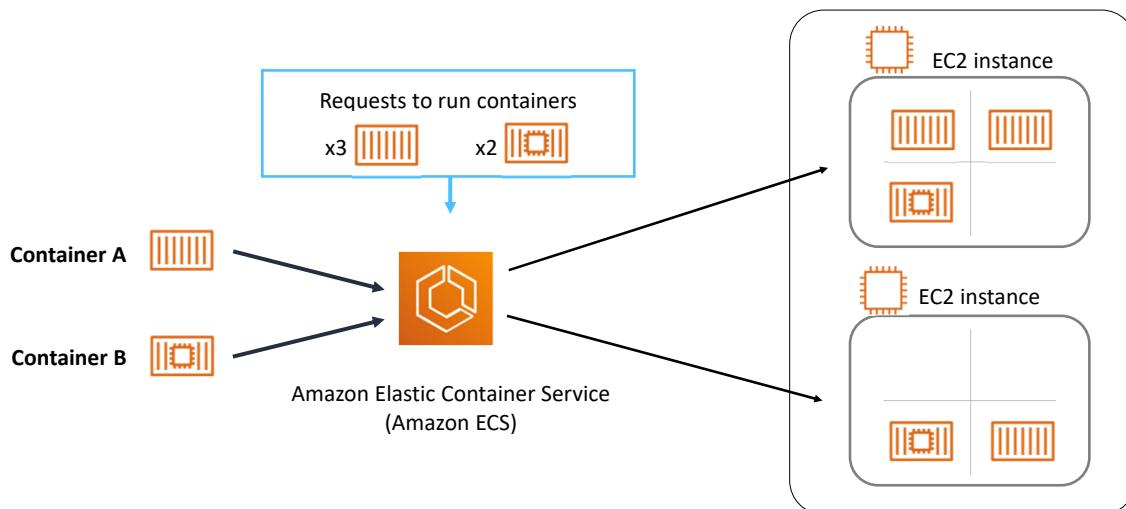
**Amazon Elastic Container Service (Amazon ECS)** is a highly scalable, high-performance container management service that supports Docker containers. Amazon ECS enables you to easily run applications on a managed cluster of Amazon EC2 instances.

Essential Amazon ECS features include the ability to:

- **Launch** up to tens of thousands of Docker containers in seconds
- **Monitor** container deployment
- **Manage** the state of the cluster that runs the containers
- **Schedule** containers by using a built-in scheduler or a third-party scheduler (for example, Apache Mesos or Blox)

Amazon ECS clusters can also use Spot Instances and Reserved Instances.

# Amazon ECS orchestrates containers



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

61

To prepare your application to run on Amazon ECS, you create a **task definition** which is a text file that **describes one or more containers**, up to a maximum of ten, that form your application. It can be thought of as a blueprint for your application. Task definitions specify parameters for your application, for example which containers to use, which ports should be opened for your application, and what data volumes should be used with the containers in the task.

A **task** is the instantiation of a task definition within a cluster. You can specify the number of tasks that will run on your cluster. The **Amazon ECS task scheduler** is responsible for placing tasks within your cluster. A task will run anywhere from one to ten containers, depending on the task definition you defined.

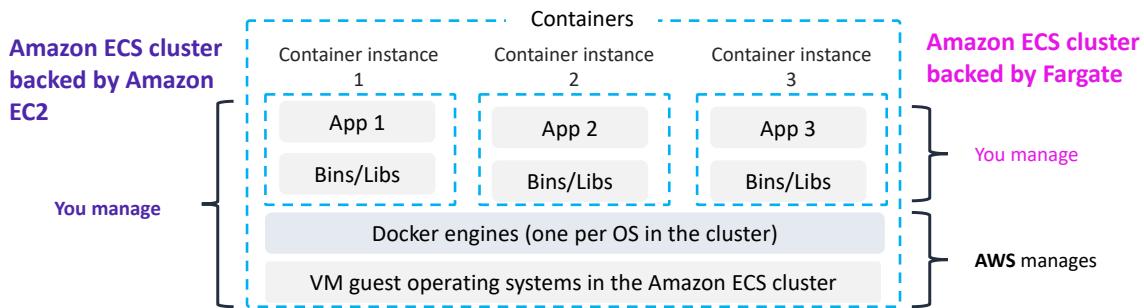
When Amazon ECS runs the containers that make up your task, it places them on an **ECS cluster**. The cluster (when you choose the EC2 launch type) consists of a group of EC2 instances each of which is running an **Amazon ECS container agent**.

Amazon ECS provides multiple scheduling strategies that will place containers across your clusters based on your resource needs (for example, CPU or RAM) and availability requirements.

# Amazon ECS cluster options



- **Key question:** Do **you** want to manage the Amazon ECS cluster that runs the containers?
  - If **yes**, create an **Amazon ECS cluster backed by Amazon EC2** (provides more granular control over infrastructure)
  - If **no**, create an **Amazon ECS cluster backed by AWS Fargate** (easier to maintain, focus on your applications)



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

62

When you create an Amazon ECS cluster, you have three options:

- A **Networking Only** cluster (powered by AWS Fargate)
- An **EC2 Linux + Networking** cluster
- An **EC2 Windows + Networking** cluster

If you choose one of the two **EC2 launch type** options, you will then be prompted to choose whether the cluster EC2 instances will run as On-Demand Instances or Spot Instances. In addition, you will need to specify many details about the EC2 instances that will make up your cluster—the same details that you must specify when you launch a stand alone EC2 instance. In this way, the EC2 launch type provides more granular control over the infrastructure that runs your container applications because you manage the EC2 instances that make up the cluster.

Amazon ECS keeps track of all the CPU, memory, and other resources in your cluster. Amazon ECS also finds the best server for your container on based on your specified resource requirements.

If you choose the networking-only **Fargate launch type**, then the cluster that will run your containers will be managed by AWS. With this option, you only need to package your application in containers, specify the CPU and memory requirements, define networking and IAM policies, and launch the application. You do not need to provision, configure, or scale the cluster. It removes the need to choose server types, decide when to scale your clusters, or optimize cluster packing. The Fargate option enables you to focus on designing

and building your applications.

# What is Kubernetes?



- Kubernetes is open source software for container orchestration.
  - Deploy and **manage containerized applications** *at scale*.
  - The same toolset can be used on premises and in the cloud.
- Complements Docker.
  - Docker enables you to run multiple containers on a single OS host.
  - Kubernetes **orchestrates** multiple Docker hosts (nodes).
- Automates –
  - Container provisioning.
  - Networking.
  - Load distribution.
  - Scaling.

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

63

**Kubernetes** is open source software for container orchestration. Kubernetes can work with many containerization technologies, including Docker. Because it is a popular open source project, a large community of developers and companies build extensions, integrations, and plugins that keep the software relevant, and new and in-demand features are added frequently.

Kubernetes enables you to deploy and manage **containerized applications** at scale. With Kubernetes, you can run any type of containerized application by using the same toolset in both on-premises data centers and the cloud. Kubernetes manages a **cluster** of compute instances (called **nodes**). It runs containers on the cluster, which are based on where compute resources are available and the resource requirements of each container. Containers are run in logical groupings called **pods**. You can run and scale one or many containers together as a pod. Each pod is given an IP address and a single Domain Name System (DNS) name, which Kubernetes uses to connect your services with each other and external traffic.

A key advantage of Kubernetes is that you can use it to run your containerized applications anywhere without needing to change your operational tooling. For example, applications can be moved from local on-premises development machines to production deployments in the cloud by using the same operational tooling.

## Amazon Elastic Kubernetes Service (Amazon EKS)



- Amazon Elastic Kubernetes Service ([Amazon EKS](#))
  - Enables you to run Kubernetes on AWS
  - Certified Kubernetes conformant (supports easy migration)
  - Supports Linux and Windows containers
  - Compatible with Kubernetes community tools and supports popular Kubernetes add-ons
- Use Amazon EKS to –
  - Manage clusters of Amazon EC2 compute instances
  - Run containers that are orchestrated by Kubernetes on those instances



Amazon Elastic  
Kubernetes Service

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

64

You might think that you could launch one or more Amazon EC2 instances, install Docker on each instance, install Kubernetes on the cluster, and manage and run Kubernetes yourself. While that is an option, AWS provides a service called Amazon Elastic Kubernetes Service (Amazon EKS) that simplifies the management of Kubernetes clusters.

**Amazon Elastic Kubernetes Service (Amazon EKS)** is a managed Kubernetes service that makes it easy for you to run Kubernetes on AWS without needing to install, operate, and maintain your own Kubernetes control plane. It is certified Kubernetes conformant, so existing applications that run on upstream Kubernetes are compatible with Amazon EKS.

Amazon EKS automatically manages the availability and scalability of the cluster nodes that are responsible for starting and stopping containers, scheduling containers on virtual machines, storing cluster data, and other tasks. It automatically detects and replaces unhealthy control plane nodes for each cluster. You can take advantage of the performance, scale, reliability, and availability of the AWS Cloud, which includes AWS networking and security services like Application Load Balancers for load distribution, IAM for role-based access control, and VPC for pod networking.

You may be wondering why Amazon offers both Amazon ECS and Amazon EKS, since they are both capable of orchestrating Docker containers. The reason that both services exist is to provide customers with flexible options. You can decide which option best matches your needs.

# Amazon Elastic Container Registry (Amazon ECR)



**Amazon ECR** is a fully managed Docker [container registry](#) that makes it easy for developers to store, manage, and deploy Docker container images.



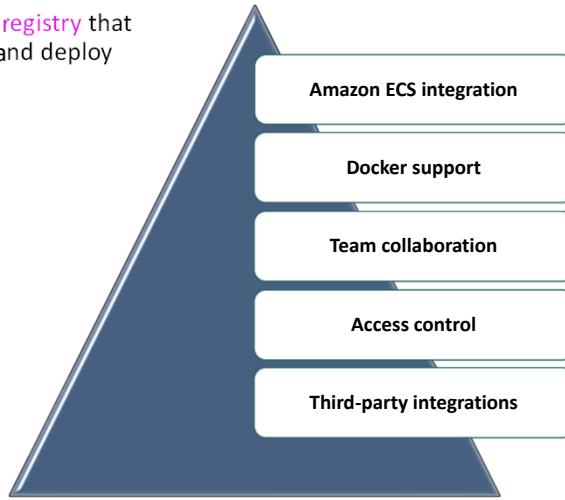
Amazon Elastic Container Registry



Image



Registry



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

65

**Amazon Elastic Container Registry (Amazon ECR)** is a fully managed Docker container registry that makes it easy for developers to store, manage, and deploy Docker container images. It is **integrated with Amazon ECS**, so you can store, run, and manage container images for applications that run on Amazon ECS. Specify the Amazon ECR repository in your task definition, and Amazon ECS will retrieve the appropriate images for your applications.

Amazon ECR supports Docker Registry HTTP API version 2, which enables you to interact with Amazon ECR by using Docker CLI commands or your preferred Docker tools. Thus, you can maintain your existing development workflow and access Amazon ECR from any Docker environment—whether it is in the cloud, on premises, or on your local machine.

You can transfer your container images to and from Amazon ECS via HTTPS. Your images are also automatically *encrypted* at rest using Amazon S3 server-side encryption.

It is also possible to use Amazon ECR images with **Amazon EKS**. See the [Using Amazon ECR Images with Amazon EKS](#) documentation for details.

## Section 4 key takeaways



66



- **Containers** can hold everything that an application needs to run.
- **Docker** is a software platform that packages software into containers.
  - A single application can span multiple containers.
- Amazon Elastic Container Service ([Amazon ECS](#)) orchestrates the running of Docker containers.
- **Kubernetes** is open source software for container orchestration.
- Amazon Elastic Kubernetes Service ([Amazon EKS](#)) enables you to run Kubernetes on AWS
- Amazon Elastic Container Registry ([Amazon ECR](#)) enables you to store, manage, and deploy your Docker containers.

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

Some key takeaways from this section include:

- Containers can hold everything that an application needs to run.
- Docker is a software platform that packages software into containers.
- A single application can span multiple containers.
- Amazon Elastic Container Service (Amazon ECS) orchestrates the running of Docker containers.
- Kubernetes is open source software for container orchestration.
- Amazon Elastic Kubernetes Service (Amazon EKS) enables you to run Kubernetes on AWS
- Amazon Elastic Container Registry (Amazon ECR) enables you to store, manage, and deploy your Docker containers.

Module 6: Compute

## Section 5: Introduction to AWS Lambda

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



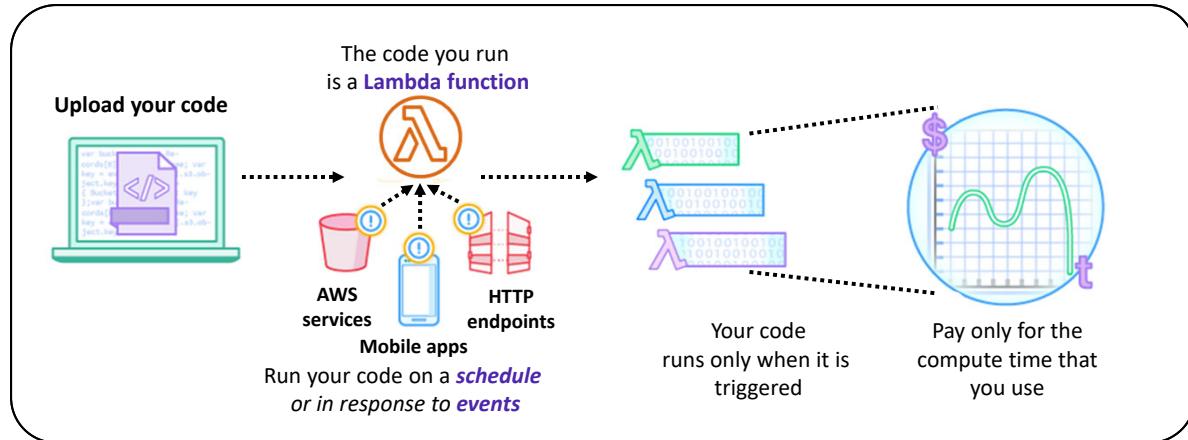
67

Introducing Section 5: Introduction to AWS Lambda.

# AWS Lambda: Run code without servers



AWS Lambda is a **serverless** compute service.



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

68

As you saw in the earlier sections of this module, AWS offers many compute options. For example, **Amazon EC2** provides virtual machines. As another example, **Amazon ECS** and **Amazon EKS** are container-based compute services.

However, there is another approach to compute that does not require you to provision or manage servers. This third approach is often referred to as **serverless computing**.

**AWS Lambda** is an event-driven, serverless compute service. Lambda enables you to run code without provisioning or managing servers.

You create a **Lambda function**, which is the AWS resource that contains the code that you upload. You then set the Lambda function to be triggered, either on a scheduled basis or in response to an event. Your code only runs when it is triggered.

You **pay only for the compute time you consume**—you are not charged when your code is not running.

## Benefits of Lambda



AWS  
Lambda



It supports multiple programming languages



Completely automated administration



Built-in fault tolerance



It supports the orchestration of multiple functions



Pay-per-use pricing

With Lambda, there are no new languages, tools, or frameworks to learn. Lambda **supports multiple programming languages**, including Java, Go, PowerShell, Node.js, C#, Python, and Ruby. Your code can use any library, either native or third-party.

Lambda **completely automates the administration**. It manages all the infrastructure to run your code on highly available, fault-tolerant infrastructure, which enables you to focus on building differentiated backend services. Lambda seamlessly deploys your code; does all the administration, maintenance, and security patches; and provides built-in logging and monitoring through Amazon CloudWatch.

Lambda provides **built-in fault tolerance**. It maintains compute capacity across multiple Availability Zones in each Region to help protect your code against individual machine failures or data center failures. There are no maintenance windows or scheduled downtimes.

You can **orchestrate multiple Lambda functions** for complex or long-running tasks by building workflows with AWS Step Functions. Use Step Functions to define workflows. These workflows trigger a collection of Lambda functions by using sequential, parallel,

branching, and error-handling steps. With Step Functions and Lambda, you can build stateful, long-running processes for applications and backends.

With Lambda, you **pay only for the requests that are served and the compute time that is required to run your code**. Billing is metered in increments of 100 milliseconds, which make it cost-effective and easy to scale automatically from a few requests per day to thousands of requests per second.

# AWS Lambda event sources

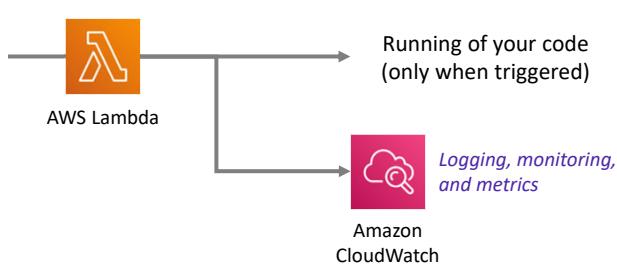


## Event sources



Configure other AWS services as **event sources** to invoke your function as shown here.

Alternatively, invoke a Lambda function from the Lambda console, AWS SDK, or AWS CLI.



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

70

An **event source** is an AWS service or a developer-created application that produces events that trigger an AWS Lambda function to run.

Some services publish events to Lambda by invoking the Lambda function directly. These services that invoke Lambda functions **asynchronously** include, but are not limited to, Amazon S3, Amazon Simple Notification Service (Amazon SNS), and Amazon CloudWatch Events.

Lambda can also poll resources in other services that do not publish events to Lambda. For example, Lambda can pull records from an **Amazon Simple Queue Service (Amazon SQS)** queue and run a Lambda function for each fetched message. Lambda can similarly read events from **Amazon DynamoDB**.

Some services, such as Elastic Load Balancing (Application Load Balancer) and Amazon API Gateway can **invoke your Lambda function directly**.

You can invoke Lambda functions directly with the Lambda console, the Lambda API, the AWS software development kit (SDK), the AWS CLI, and AWS toolkits. The direct invocation approach can be useful, such as when you are developing a mobile app and want the app to call Lambda functions. See the [Using Lambda with Other Services](#) documentation for

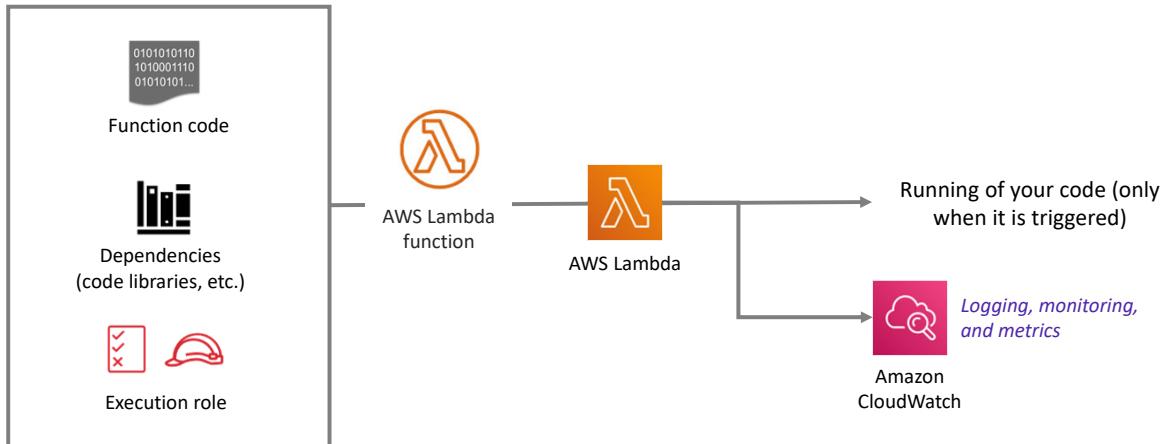
further details about all supported services.

**AWS Lambda automatically monitors Lambda functions by using Amazon CloudWatch.** To help you troubleshoot failures in a function, Lambda logs all requests that are handled by your function. It also **automatically stores logs that are generated by your code** through Amazon CloudWatch Logs.

# AWS Lambda function configuration



## Lambda function configuration



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

71

Remember that a Lambda function is the custom code that you write to process events, and that Lambda runs the Lambda function on your behalf.

When you use the AWS Management Console to create a **Lambda function**, you first give the function a name. Then, you specify:

- The **runtime environment** the function will use (for example, a version of Python or Node.js)
- An **execution role** (to grant IAM permission to the function so that it can interact with other AWS services as necessary)

Next, after you click **Create Function**, you configure the function. Configurations include:

- Add a **trigger** (specify one of the available **event sources** from the previous slide)
- Add your **function code** (use the provided code editor or upload a file that contains your code)
- Specify the **memory** in MB to allocate to your function (128 MB to 10,240 MB)
- Optionally specify environment variables, description, timeout, the specific virtual private cloud (VPC) to run the function in, tags you would like to use, and other settings.

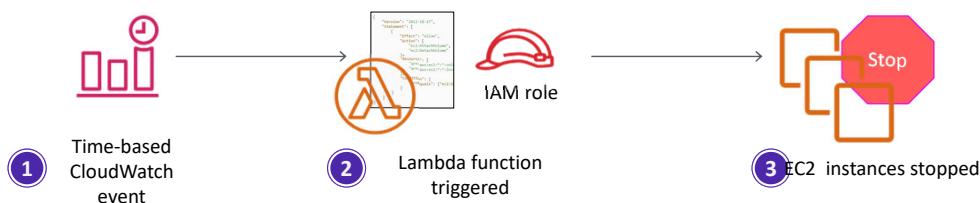
For more information, see [Configuring functions in the AWS Lambda console](#) in the AWS Documentation.

All of the above settings end up in a **Lambda deployment package** which is a ZIP archive that contains your function code and dependencies. When you use the Lambda console to author your function, the console manages the package for you. However, you need to create a deployment package if you use the Lambda API to manage functions.

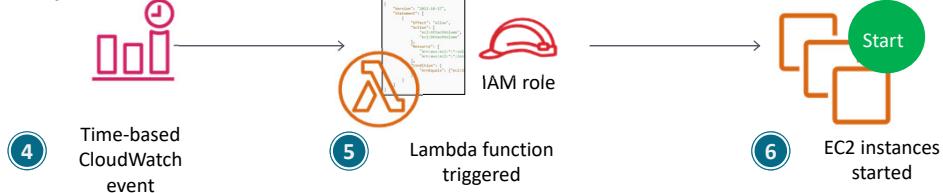
## Schedule-based Lambda function example: Start and stop EC2 instances



### Stop instances example



### Start instances example



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

72

Consider an example use case for a schedule-based Lambda function. Say that you are in a situation where you want to reduce your Amazon EC2 usage. You decide that you want to stop instances at a predefined time (for example, at night when no one is accessing them) and then you want to start the instances back up in the morning (before the workday starts).

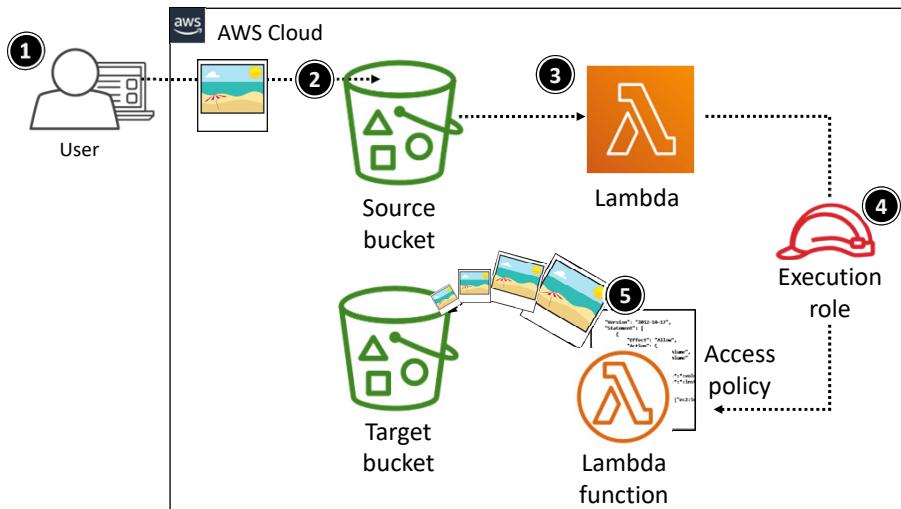
In this situation, you could configure **AWS Lambda** and **Amazon CloudWatch Events** to help you accomplish these actions automatically.

Here is what happens at each step in the example:

1. A CloudWatch event is scheduled to run a Lambda function to stop your EC2 instances at (for example) 22:00 GMT.
2. The Lambda function is triggered and runs with the IAM role that gives the function permission to stop the EC2 instances.
3. The EC2 instances enter the stopped state.
4. Later, at (for example) 05:00 AM GMT, a CloudWatch event is scheduled to run a Lambda function to start the EC2 instances.
5. The Lambda function is triggered and runs with the IAM role that gives it permission to start the EC2 instances.

6. The EC2 instances enter the running state.

## Event-based Lambda function example: Create thumbnail images



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

73

Now, consider an example use case for an event-based Lambda function. Suppose that you want to create a thumbnail for each image (.jpg or .png object) that is uploaded to an S3 bucket.

To build a solution, you can create a Lambda function that Amazon S3 invokes when objects are uploaded. Then, the Lambda function reads the image object from the source bucket and creates a thumbnail image in a target bucket. Here's how it works:

1. A user uploads an object to the source bucket in Amazon S3 (object-created event).
2. Amazon S3 detects the object-created event.
3. Amazon S3 publishes the object-created event to Lambda by invoking the Lambda function and passing event data.
4. Lambda runs the Lambda function by assuming the execution role that you specified when you created the Lambda function.
5. Based the event data that the Lambda function receives, it knows the source bucket name and object key name. The Lambda function reads the object and creates a thumbnail by using graphics libraries, and saves the thumbnail to the target bucket.

# AWS Lambda quotas



## Soft limits per Region:

- Concurrent executions = 1,000
- Function and layer storage = 75 GB

## Hard limits for individual functions:

- Maximum function memory allocation = 10,240 MB
- Function timeout = 15 minutes
- Deployment package size = 250 MB unzipped, including layers
- Container image code package size = 10 GB

Additional limits also exist. Details are in the [AWS Lambda quotas](#) documentation.

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

74

AWS Lambda does have some quotas that you should know about when you create and deploy Lambda functions.

AWS Lambda limits the amount of compute and storage resources that you can use to run and store functions. For example, as of this writing, the maximum memory allocation for a single Lambda function is 10,240 MB. It also has limits of 1,000 concurrent executions in a Region. Lambda functions can be configured to run up to 15 minutes per run. You can set the timeout to any value between 1 second and 15 minutes. If you are troubleshooting a Lambda deployment, keep these limits in mind.

There are limits on the **deployment package size** of a function (250 MB). A **layer** is a ZIP archive that contains libraries, a custom runtime, or other dependencies. With layers, you can use libraries in your function without needing to include them in your **deployment package**. Using layers can help avoid reaching the size limit for deployment package. Layers are also a good way to share code and data between Lambda functions.

For larger workloads that rely on sizable dependencies, such as machine learning or data intensive workloads, you can deploy your Lambda function to a container image up to 10 GB in size.

Limits are either soft or hard. **Soft limits** on an account can potentially be relaxed by

submitting a support ticket and providing justification for the request. **Hard limits** cannot be increased.

For the details on current AWS Lambda quotas, refer to the [AWS Lambda quotas](#) documentation.

## Section 5 key takeaways



75



- **Serverless computing** enables you to build and run applications and services without provisioning or managing servers.
- **AWS Lambda** is a serverless compute service that provides built-in fault tolerance and automatic scaling.
- An **event source** is an AWS service or developer-created application that triggers a Lambda function to run.
- The maximum memory allocation for a single Lambda function is 10,240 MB.
- The maximum run time for a Lambda function is 15 minutes.

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

Some key takeaways from this section of the module include:

- Serverless computing enables you to build and run applications and services without provisioning or managing servers.
- AWS Lambda is a serverless compute service that provides built-in fault tolerance and automatic scaling.
- An event source is an AWS service or developer-created application that triggers a Lambda function to run.
- The maximum memory allocation for a single Lambda function is 10,240 MB.
- The maximum run time for a Lambda function is 15 minutes.



## Activity: Create an AWS Lambda Stopinator Function

76

### To complete this activity:

- Go to the hands-on lab environment and launch the AWS Lambda activity.
- Follow the instructions that are provided in the hands-on lab environment.

Photo by Pixabay from Pexels.

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

In this hands-on activity, you will create a basic Lambda function that stops an EC2 instance.



## Activity debrief: key takeaways

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

77

The instructor will lead a conversation about the key takeaways from the activity after students have completed it.

Module 6: Compute

## Section 6: Introduction to AWS Elastic Beanstalk

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



Introducing Section 6: Introduction to AWS Elastic Beanstalk.



**AWS Elastic  
Beanstalk**

- An easy way to get **web applications** up and running
- A **managed service** that automatically handles –
  - Infrastructure provisioning and configuration
  - Deployment
  - Load balancing
  - Automatic scaling
  - Health monitoring
  - Analysis and debugging
  - Logging
- No additional charge for Elastic Beanstalk
  - Pay only for the underlying resources that are used

AWS Elastic Beanstalk is another AWS compute service option. It is a platform as a service (or PaaS) that facilitates the quick deployment, scaling, and management of your web applications and services.

You remain in control. The entire platform is already built, and you only need to upload your code. Choose your instance type, your database, set and adjust automatic scaling, update your application, access the server log files, and enable HTTPS on the load balancer.

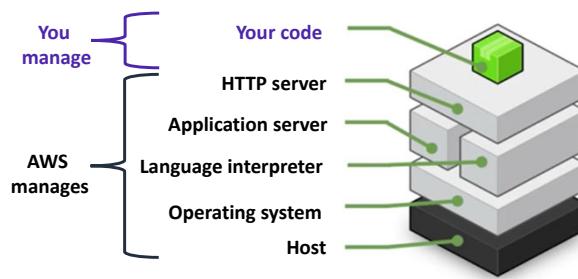
You upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning and load balancing to automatic scaling and monitoring application health. At the same time, you retain full control over the AWS resources that power your application, and you can access the underlying resources at any time.

There is no additional charge for AWS Elastic Beanstalk. You pay for the AWS resources (for example, EC2 instances or S3 buckets) you create to store and run your application. You only pay for what you use, as you use it. There are no minimum fees and no upfront commitments.

# AWS Elastic Beanstalk deployments



- It supports web applications written for common platforms
  - Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker
- You upload your code
  - Elastic Beanstalk automatically handles the deployment
  - Deploys on servers such as Apache, NGINX, Passenger, Puma, and Microsoft Internet Information Services (IIS)



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

80

AWS Elastic Beanstalk enables you to deploy your code through the AWS Management Console, the AWS Command Line Interface (AWS CLI), Visual Studio, and Eclipse. It provides all the application services that you need for your application. The only thing you must create is your code. Elastic Beanstalk is designed to make deploying your application a quick and easy process.

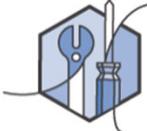
Elastic Beanstalk supports a broad range of platforms. Supported platforms include Docker, Go, Java, .NET, Node.js, PHP, Python, and Ruby.

AWS Elastic Beanstalk deploys your code on **Apache Tomcat** for Java applications; **Apache HTTP Server** for PHP and Python applications; **NGINX** or **Apache HTTP Server** for Node.js applications; **Passenger** or **Puma** for Ruby applications; and **Microsoft Internet Information Services (IIS)** for .NET applications, Java SE, Docker, and Go.

## Benefits of Elastic Beanstalk



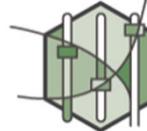
Fast and simple to start using



Developer productivity



Difficult to outgrow



Complete resource control

Elastic Beanstalk is **fast and simple to start using**. Use the AWS Management Console, a Git repository, or an integrated development environment (IDE) such as Eclipse or Visual Studio to upload your application. Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, automatic scaling, and monitoring application health.

You can improve your **developer productivity** by focusing on writing code instead of managing and configuring servers, databases, load balancers, firewalls, and networks. AWS updates the underlying platform that runs your application with patches and updates.

Elastic Beanstalk is **difficult to outgrow**. With Elastic Beanstalk, your application can handle peaks in workload or traffic while minimizing your costs. It automatically scales your application up or down based on your application's specific needs by using easily adjustable automatic scaling settings. You can use CPU utilization metrics to trigger automatic scaling actions.

You have the **freedom to select the AWS resources**—such as Amazon EC2 instance type—that are optimal for your application. Elastic Beanstalk enables you to retain full control over the AWS resources that power your application. If you decide that you want to take over some (or all) of the elements of your infrastructure, you can do so seamlessly by using the management capabilities that are provided by Elastic Beanstalk.



## Activity: AWS Elastic Beanstalk

82

### To complete this activity:

- Go to the hands-on lab environment and launch the AWS Elastic Beanstalk activity.
- Follow the instructions that are provided in the hands-on lab environment.

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

In this hands-on activity, you will gain an understanding of why you might want to use Elastic Beanstalk to deploy a web application on AWS.



## Activity debrief: Key takeaways



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

83

The instructor might choose to lead a conversation about the key takeaways from the activity after you have completed it.

## Section 6 key takeaways



84



- AWS Elastic Beanstalk enhances developer productivity.
  - Simplifies the process of deploying your application.
  - Reduces management complexity.
- Elastic Beanstalk supports Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker
- There is no charge for Elastic Beanstalk. Pay only for the AWS resources that you use.

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

Some key takeaways from this section of the module include:

- AWS Elastic Beanstalk enhances developer productivity.
  - Simplifies the process of deploying your application.
  - Reduces management complexity.
- Elastic Beanstalk supports Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker.
- There is no charge for Elastic Beanstalk. Pay only for the AWS resources you use.

Module 6: Compute

## Module wrap-up

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



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:

- Provide an overview of different AWS compute services in the cloud
- Demonstrate why to use Amazon Elastic Compute Cloud (Amazon EC2)
- Identify the functionality in the Amazon EC2 console
- Perform basic functions in Amazon EC2 to build a virtual computing environment
- Identify Amazon EC2 cost optimization elements
- Demonstrate when to use AWS Elastic Beanstalk
- Demonstrate when to use AWS Lambda
- Identify how to run containerized applications in a cluster of managed servers

In summary, in this module, you learned how to:

- Provide an overview of different AWS compute services in the cloud
- Demonstrate why to use Amazon Elastic Compute Cloud (Amazon EC2)
- Identify the functionality in the Amazon EC2 console
- Perform basic functions in Amazon EC2 to build a virtual computing environment
- Identify Amazon EC2 cost optimization elements
- Demonstrate when to use AWS Elastic Beanstalk
- Demonstrate when to use AWS Lambda
- Identify how to run containerized applications in a cluster of managed servers

# Complete the knowledge check



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

87

It is now time to complete the knowledge check for this module.

## Sample exam question



Which AWS service helps developers quickly deploy resources which can make use of different programming languages, such as .NET and Java?

- A. AWS CloudFormation
- B. AWS SQS
- C. AWS Elastic Beanstalk
- D. Amazon Elastic Compute Cloud (Amazon EC2)

Look at the answer choices and rule them out based on the keywords that were previously highlighted.

## Additional resources



- [Amazon EC2 Documentation](#)
- [Amazon EC2 Pricing](#)
- [Amazon ECS Workshop](#)
- [Running Containers on AWS](#)
- [Amazon EKS Workshop](#)
- [AWS Lambda Documentation](#)
- [AWS Elastic Beanstalk Documentation](#)
- [Cost Optimization Playbook](#)

Compute services on AWS is a large topic, and this module only provided an introduction to the subject. The following resources provide more detail:

- [Amazon EC2 Documentation](#)
- [Amazon EC2 Pricing](#)
- [Amazon ECS Workshop](#)
- [Running Containers on AWS](#)
- [Amazon EKS Workshop](#)
- [AWS Lambda Documentation](#)
- [AWS Elastic Beanstalk Documentation](#)
- [Cost Optimization Playbook](#)

# Thank you

© 2019 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections or feedback on the course, please email us at: [aws-course-feedback@amazon.com](mailto:aws-course-feedback@amazon.com). For all other questions, contact us at: <https://aws.amazon.com/contact-us/aws-training/>. All trademarks are the property of their owners.



Thank you for completing this module.

AWS Academy Cloud Foundations

# Module 7: Storage

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



Welcome to Module 7: Storage.

# Module overview



## Topics

- Amazon Elastic Block Store (Amazon EBS)
- Amazon Simple Storage Service (Amazon S3)
- Amazon Elastic File System (Amazon EFS)
- Amazon Simple Storage Service Glacier

## Demos

- Amazon EBS console
- Amazon S3 console
- Amazon EFS console
- Amazon S3 Glacier console

## Lab

- Working with Amazon EBS

## Activities

- Storage solution case study



## Knowledge check

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

2

Cloud storage is typically more reliable, scalable, and secure than traditional on-premises storage systems. Cloud storage is a critical component of cloud computing because it holds the information that applications use. Big data analytics, data warehouses, the Internet of Things (IoT), databases, and backup and archive applications all rely on some form of data storage architecture.

This module addresses the following topics:

- Amazon Elastic Block Store (Amazon EBS)
- Amazon Simple Storage Service (Amazon S3)
- Amazon Elastic File System (Amazon EFS)
- Amazon Simple Storage Service Glacier

This module includes four recorded demonstrations that show you how to use the AWS Management Console to create storage solutions.

This module includes a hands-on lab where you create an Amazon EBS volume, and then attach it to an Amazon Elastic Compute Cloud (Amazon EC2) instance. You also create a snapshot of your volume and then use the snapshot to create a new volume.

This module includes an activity that challenges you to determine the best storage solution for a business case.

Finally, you are asked to complete a knowledge check that tests your understanding of the key concepts in this module.

# Module objectives



After completing this module, you should be able to:

- Identify the different types of storage
- Explain Amazon S3
- Identify the functionality in Amazon S3
- Explain Amazon EBS
- Identify the functionality in Amazon EBS
- Perform functions in Amazon EBS to build an Amazon EC2 storage solution
- Explain Amazon EFS
- Identify the functionality in Amazon EFS
- Explain Amazon S3 Glacier
- Identify the functionality in Amazon S3 Glacier
- Differentiate between Amazon EBS, Amazon S3, Amazon EFS, and Amazon S3 Glacier

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

3

The goal of this module is to discover key concepts that relate to storage. You will learn about the different types of storage resources that are available and review the different pricing options so that you can understand how different choices affect your solution cost.

After completing this module, you should be able to:

- Identify the different types of storage
- Explain Amazon S3
- Identify the functionality in Amazon S3
- Explain Amazon EBS
- Identify the functionality in Amazon EBS
- Perform functions in Amazon EBS to build an Amazon EC2 storage solution
- Explain Amazon EFS
- Identify the functionality in Amazon EFS
- Explain Amazon S3 Glacier
- Identify the functionality in Amazon S3 Glacier
- Differentiate between Amazon EBS, Amazon S3, Amazon EFS, and Amazon S3 Glacier

# Core AWS services



Amazon Virtual  
Private Cloud  
(Amazon VPC)



Amazon Elastic  
Compute Cloud  
(Amazon EC2)



## Storage



AWS Identity and Access  
Management (IAM)



## Database

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

4

Storage is another AWS core service category. Some broad categories of storage include: instance store (ephemeral storage), Amazon EBS, Amazon EFS, Amazon S3, and Amazon S3 Glacier.

- Instance store, or *ephemeral storage*, is **temporary storage** that is added to your Amazon EC2 instance.
- Amazon EBS is **persistent, mountable storage** that can be mounted as a device to an Amazon EC2 instance. Amazon EBS can be mounted to an Amazon EC2 instance only within the same Availability Zone. Only one Amazon EC2 instance at a time can mount an Amazon EBS volume.
- Amazon EFS is a shared file system that multiple Amazon EC2 instances can mount at the same time.
- Amazon S3 is persistent storage where each file becomes an object and is available through a Uniform Resource Locator (URL); it can be accessed from anywhere.
- Amazon S3 Glacier is for cold storage for data that is not accessed frequently (for example, when you need long-term data storage for archival or compliance reasons).

Module 7: Storage

## Section 1: Amazon Elastic Block Store (Amazon EBS)

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



Introduce Section 1: Amazon Elastic Block Store (Amazon EBS).



## Amazon Elastic Block Store (Amazon EBS)

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

6

Amazon EBS provides persistent block storage volumes for use with Amazon EC2 instances. Persistent storage is any data storage device that retains data after power to that device is shut off. It is also sometimes called **non-volatile storage**.

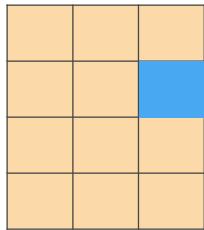
Each Amazon EBS volume is automatically replicated *within* its Availability Zone to protect you from component failure. It is designed for high availability and durability. Amazon EBS volumes provide the consistent and low-latency performance that is needed to run your workloads.

With Amazon EBS, you can scale your usage up or down within minutes, while paying a low price for only what you provision.

# AWS storage options: Block storage versus object storage

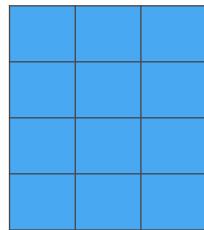


What if you want to change one character in a 1-GB file?



Block storage

Change one block (piece of the file)  
that contains the character



Object storage

Entire file must be updated

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

7

What happens if you want to change one character in a 1-GB file? With block storage, you change only the block that contains the character. With object storage, the entire file must be updated.

One critical difference between some storage types is whether they offer *block-level storage* or *object-level storage*.

This difference has a major effect on the throughput, latency, and cost of your storage solution. Block storage solutions are typically faster and use less bandwidth, but they can cost more than object-level storage.

Amazon EBS enables you to **create individual storage volumes** and **attach them** to an Amazon EC2 instance:

- Amazon EBS offers block-level storage.
- Volumes are automatically replicated within its Availability Zone.
- It can be backed up automatically to Amazon S3 through snapshots.
- Uses include –
  - Boot volumes and storage for Amazon Elastic Compute Cloud (Amazon EC2) instances
  - Data storage with a file system
  - Database hosts
  - Enterprise applications

Amazon EBS enables you to create individual storage volumes and attach them to an Amazon EC2 instance. Amazon EBS offers block-level storage, where its volumes are automatically replicated within its Availability Zone. Amazon EBS is designed to provide durable, detachable, block-level storage (which is like an external hard drive) for your Amazon EC2 instances. Because they are directly attached to the instances, they can provide low latency between where the data is stored and where it might be used on the instance.

For this reason, they can be used to run a database with an Amazon EC2 instance. Amazon EBS volumes are included as part of the backup of your instances into Amazon Machine Images (or AMIs). AMIs are stored in Amazon S3 and can be reused to create new Amazon EC2 instances later.

A backup of an Amazon EBS volume is called a *snapshot*. The first snapshot is called the *baseline snapshot*. Any other snapshot after the baseline captures only what is different from the previous snapshot.

Amazon EBS volumes uses include:

- Boot volumes and storage for Amazon EC2 instances
- Data storage with a file system
- Database hosts

- Enterprise applications

# Amazon EBS volume types



	Solid State Drives (SSD)		Hard Disk Drives (HDD)	
	General Purpose	Provisioned IOPS	Throughput-Optimized	Cold
Maximum Volume Size	16 TiB	16 TiB	16 TiB	16 TiB
Maximum IOPS/Volume	16,000	64,000	500	250
Maximum Throughput/Volume	250 MiB/s	1,000 MiB/s	500 MiB/s	250 MiB/s

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

9

Matching the correct technology to your workload is a best practice for reducing storage costs. Provisioned IOPS SSD-backed Amazon EBS volumes can give you the highest performance. However, if your application doesn't require or won't use performance that high, General Purpose SSD is usually sufficient. Only SSDs can be used as boot volumes for EC2 instances. The lower-cost options might be a solution for additional storage or use cases other than boot volumes.

To learn more, see [Amazon EBS volume types](#).

# Amazon EBS volume type use cases



Solid State Drives (SSD)		Hard Disk Drives (HDD)	
General Purpose	Provisioned IOPS	Throughput-Optimized	Cold
<ul style="list-style-type: none"><li>This type is recommended for most workloads</li><li>System boot volumes</li><li>Virtual desktops</li><li>Low-latency interactive applications</li><li>Development and test environments</li></ul>	<ul style="list-style-type: none"><li>Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/second of throughput per volume</li><li>Large database workloads</li></ul>	<ul style="list-style-type: none"><li>Streaming workloads that require consistent, fast throughput at a low price</li><li>Big data</li><li>Data warehouses</li><li>Log processing</li><li>It cannot be a boot volume</li></ul>	<ul style="list-style-type: none"><li>Throughput-oriented storage for large volumes of data that is infrequently accessed</li><li>Scenarios where the lowest storage cost is important</li><li>It cannot be a boot volume</li></ul>

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

10

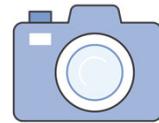
As mentioned previously an **Amazon EBS volume** is a durable, block-level storage device that you can attach to a single EC2 instance. You can use Amazon EBS volumes as primary storage for data that requires frequent updates, such as the system drive for an instance or storage for a database application. You can also use them for throughput-intensive applications that perform continuous disk scans. Amazon EBS volumes persist independently from the running life of an EC2 instance.

Use cases for EBS vary by the storage type used and whether you are using General Purpose or Provisioned IOPS.

# Amazon EBS features



- Snapshots –
  - Point-in-time snapshots
  - Recreate a new volume at any time
- Encryption –
  - Encrypted Amazon EBS volumes
  - No additional cost
- Elasticity –
  - Increase capacity
  - Change to different types



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

11

To provide an even higher level of data durability, Amazon EBS enables you to create **point-in-time snapshots** of your volumes, and you can re-create a new volume from a snapshot at any time. You can also share snapshots or even copy snapshots to different AWS Regions for even greater **disaster recovery (DR) protection**. For example, you can encrypt and share your snapshots from Virginia in the US to Tokyo, Japan.

You can also have encrypted Amazon EBS volumes at no additional cost, so the data that moves between the EC2 instance and the EBS volume inside AWS data centers is encrypted in transit.

As your company grows, the amount of data that is stored on your Amazon EBS volumes is also likely to grow. Amazon EBS volumes can increase capacity and change to different types, so you can change from hard disk drives (HDDs) to solid state drives (SSDs) or increase from a 50-GB volume to a 16-TB volume. For example, you can do this resize operation dynamically without needing to stop the instances.

## 1. Volumes –

- Amazon EBS volumes persist independently from the instance.
- All volume types are charged by the amount that is provisioned per month.

## 2. IOPS –

- General Purpose SSD:
  - Charged by the amount that you provision in GB per month until storage is released.
- Magnetic:
  - Charged by the number of requests to the volume.
- Provisioned IOPS SSD:
  - Charged by the amount that you provision in IOPS (multiplied by the percentage of days that you provision for the month).

When you begin to estimate the cost for Amazon EBS, you must consider the following:

1. **Volumes** – Volume storage for all Amazon EBS volume types is charged by the amount you provision in GB per month, until you release the storage.
2. **IOPS** – I/O is included in the price of General Purpose SSD volumes. However, for Amazon EBS magnetic volumes, I/O is charged by the number of requests that you make to your volume. With Provisioned IOPS SSD volumes, you are also charged by the amount you provision in IOPS (multiplied by the percentage of days that you provision for the month).

The pricing and provisioning of Amazon EBS are complex. In general, you pay for the size of the volume and its usage. To learn more about the full, highly complex pricing and provisioning concepts of Amazon EBS, see [Amazon EBS volume types](#).

### 3. **Snapshots –**

- Added cost of Amazon EBS snapshots to Amazon S3 is per GB-month of data stored.

### 4. **Data transfer –**

- Inbound data transfer is free.
- Outbound data transfer across Regions incurs charges.

3. **Snapshots –** Amazon EBS enables you to back up snapshots of your data to Amazon S3 for durable recovery. If you opt for Amazon EBS snapshots, the added cost is per GB-month of data stored.
4. **Data transfer –** When you copy Amazon EBS snapshots, you are charged for the data that is transferred across Regions. After the snapshot is copied, standard Amazon EBS snapshot charges apply for storage in the destination Region.

## Section 1 key takeaways



14

### Amazon EBS features:

- Persistent and customizable block storage for Amazon EC2
- HDD and SSD types
- Replicated in the same Availability Zone
- Easy and transparent encryption
- Elastic volumes
- Back up by using snapshots

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

Amazon EBS provides block-level storage volumes for use with Amazon EC2 instances. Amazon EBS volumes are off-instance storage that persists independently from the life of an instance. They are analogous to virtual disks in the cloud. Amazon EBS provides three volume types: General Purpose SSD, Provisioned IOPS SSD, and magnetic.

The three volume types differ in performance characteristics and cost, so you can choose the right storage performance and price for the needs of your applications.

Additional benefits include replication in the same Availability Zone, easy and transparent encryption, elastic volumes, and backup by using snapshots.

To learn more about Amazon EBS, see: [Elastic Block Store](#).

.

## Recorded demo: Amazon Elastic Block Store

15



### Set up demo

Amazon Elastic Block Store (EBS)



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

Now, take a moment to watch the [Elastic Block Store demo](#). The recording runs a little over 5 minutes, and it reinforces many of the concepts that were discussed in this section of the module.

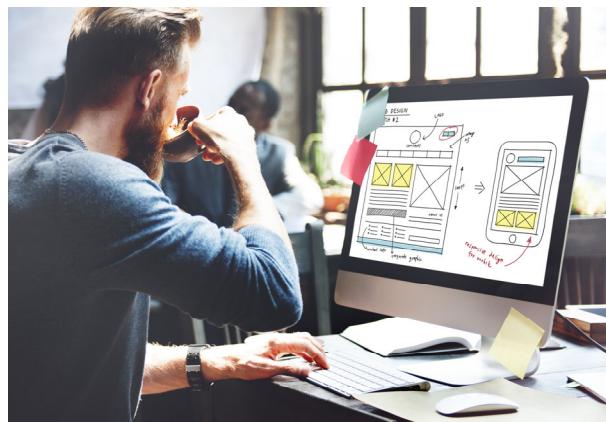
The demonstration shows how to configure the following resources by using the AWS Management Console. The demonstration shows how to:

- Create an Amazon General Purpose (SSD) EBS volume
- Attach the EBS volume to an EC2 instance

The demonstration also shows how to interact with the EBS volume using the Amazon Command Line Interface and how to mount the EBS volume to the EC2 instance.

## Lab 4: Working with Amazon EBS

16



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

You will now work on Lab 4: Working with Amazon EBS.

## Lab 4: Scenario



This lab is designed to show you how to create an Amazon EBS volume. After you create the volume, you will attach the volume to an Amazon EC2 instance, configure the instance to use a virtual disk, create a snapshot and then restore from the snapshot.



This lab is designed to show you how to create an Amazon EBS volume. After you create the volume, you will attach the volume to an Amazon EC2 instance, configure the instance to use a virtual disk, create a snapshot and then restore from the snapshot.

After completing this lab, you should be able to:

- Create an Amazon EBS volume
- Attach that volume to an instance
- Configure the instance to use the virtual disk
- Create an Amazon EBS snapshot
- Restore the snapshot

## Lab 4: Final product



### In this lab, you:

- Created an Amazon EBS volume
- Attached that volume to an instance
- Configured the instance to use the virtual disk
- Created an Amazon EBS snapshot
- Restored the snapshot



~ 30 minutes

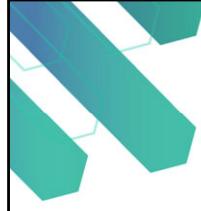


## Begin Lab 4: Working with Amazon EBS

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

19

It is now time to start the lab.



## Lab debrief: Key takeaways



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

20

In this lab, you:

- Created an Amazon EBS volume
- Attached the volume to an instance
- Configured the instance to use the virtual disk
- Created an Amazon EBS snapshot
- Restored the snapshot

Module 7: Storage

## Section 2: Amazon Simple Storage Service (Amazon S3)

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



Introduce Section 3: Amazon Simple Storage Service.

Companies need the ability to simply and securely collect, store, and analyze their data on a massive scale. Amazon S3 is object storage that is built to store and retrieve any amount of data from anywhere: websites and mobile apps, corporate applications, and data from Internet of Things (IoT) sensors or devices.



## Amazon Simple Storage Service (Amazon S3)

Amazon S3 is object-level storage, which means that if you want to change a part of a file, you must make the change and then re-upload the entire modified file. Amazon S3 stores data as objects within resources that are called **buckets**.

You will now learn more about Amazon S3.

# Amazon S3 overview



- Data is stored as objects in buckets
- Virtually unlimited storage
  - Single object is limited to 5 TB
- Designed for 11 9s of durability
- Granular access to bucket and objects

Amazon S3 is a managed cloud storage solution that is designed to scale seamlessly and provide 11 9s of durability. You can store virtually as many objects as you want in a bucket, and you can write, read, and delete objects in your bucket. Bucket names are universal and must be unique across all existing bucket names in Amazon S3. Objects can be up to 5 TB in size. By default, data in Amazon S3 is stored redundantly across multiple facilities and multiple devices in each facility.

The data that you store in Amazon S3 is not associated with any particular server, and you do not need manage any infrastructure yourself. You can put as many objects into Amazon S3 as you want. Amazon S3 holds trillions of objects and regularly peaks at millions of requests per second.

Objects can be almost any data file, such as images, videos, or server logs. Because Amazon S3 supports objects as large as several terabytes in size, you can even store database snapshots as objects. Amazon S3 also provides low-latency access to the data over the internet by Hypertext Transfer Protocol (HTTP) or Secure HTTP (HTTPS), so you can retrieve data anytime from anywhere. You can also access Amazon S3 privately through a virtual private cloud (VPC) endpoint. You get fine-grained control over who can access your data by using AWS Identity and Access Management (IAM) policies, Amazon S3 bucket policies, and even per-object access control lists.

By default, none of your data is shared publicly. You can also encrypt your data in transit and choose to enable server-side encryption on your objects.

You can access Amazon S3 through the web-based AWS Management Console; programmatically through the API and SDKs; or with third-party solutions, which use the API or the SDKs.

Amazon S3 includes event notifications that enable you to set up automatic notifications when certain events occur, such as when an object is uploaded to a bucket or deleted from a specific bucket. Those notifications can be sent to you, or they can be used to trigger other processes, such as AWS Lambda functions.

With storage class analysis, you can analyze storage access patterns and transition the right data to the right storage class. The Amazon S3 Analytics feature automatically identifies the optimal lifecycle policy to transition less frequently accessed storage to Amazon S3 Standard – Infrequent Access (Amazon S3 Standard-IA). You can configure a storage class analysis policy to monitor an entire bucket, a prefix, or an object tag.

When an infrequent access pattern is observed, you can easily create a new lifecycle age policy that is based on the results. Storage class analysis also provides daily visualizations of your storage usage in the AWS Management Console. You can export them to an Amazon S3 bucket to analyze by using the business intelligence (BI) tools of your choice, such as Amazon QuickSight.

## Amazon S3 storage classes



Amazon S3 offers a range of object-level storage classes that are designed for different use cases:

- Amazon S3 Standard
- Amazon S3 Intelligent-Tiering
- Amazon S3 Standard-Infrequent Access (Amazon S3 Standard-IA)
- Amazon S3 One Zone-Infrequent Access (Amazon S3 One Zone-IA)
- Amazon S3 Glacier
- Amazon S3 Glacier Deep Archive

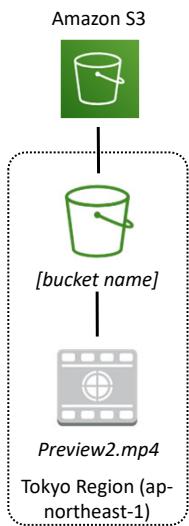
Amazon S3 offers a range of object-level storage classes that are designed for different use cases. These classes include:

- **Amazon S3 Standard** – Amazon S3 Standard is designed for high durability, availability, and performance object storage for frequently accessed data. Because it delivers low latency and high throughput, Amazon S3 Standard is appropriate for a variety of use cases, including cloud applications, dynamic websites, content distribution, mobile and gaming applications, and big data analytics.
- **Amazon S3 Intelligent-Tiering** – The Amazon S3 Intelligent-Tiering storage class is designed to optimize costs by automatically moving data to the most cost-effective access tier, without performance impact or operational overhead. For a small monthly monitoring and automation fee per object, Amazon S3 monitors access patterns of the objects in Amazon S3 Intelligent-Tiering, and moves the objects that have not been accessed for 30 consecutive days to the infrequent access tier. If an object in the infrequent access tier is accessed, it is automatically moved back to the frequent access tier. There are no retrieval fees when you use the Amazon S3 Intelligent-Tiering storage class, and no additional fees when objects are moved between access tiers. It works well for long-lived data with access patterns that are unknown or unpredictable.

- **Amazon S3 Standard-Infrequent Access (Amazon S3 Standard-IA)** – The Amazon S3 Standard-IA storage class is used for data that is accessed less frequently, but requires rapid access when needed. Amazon S3 Standard-IA is designed to provide the high durability, high throughput, and low latency of Amazon S3 Standard, with a low per-GB storage price and per-GB retrieval fee. This combination of low cost and high performance makes Amazon S3 Standard-IA good for long-term storage and backups, and as a data store for disaster recovery files.
- **Amazon S3 One Zone-Infrequent Access (Amazon S3 One Zone-IA)** – Amazon S3 One Zone-IA is for data that is accessed less frequently, but requires rapid access when needed. Unlike other Amazon S3 storage classes, which store data in a minimum of three Availability Zones, Amazon S3 One Zone-IA stores data in a single Availability Zone and it costs less than Amazon S3 Standard-IA. Amazon S3 One Zone-IA works well for customers who want a lower-cost option for infrequently accessed data, but do not require the availability and resilience of Amazon S3 Standard or Amazon S3 Standard-IA. It is a good choice for storing secondary backup copies of on-premises data or easily re-creatable data. You can also use it as cost-effective storage for data that is replicated from another AWS Region by using Amazon S3 Cross-Region Replication.
- **Amazon S3 Glacier** – Amazon S3 Glacier is a secure, durable, and low-cost storage class for data archiving. You can reliably store any amount of data at costs that are competitive with—or cheaper than—on-premises solutions. To keep costs low yet suitable for varying needs, Amazon S3 Glacier provides three retrieval options that range from a few minutes to hours. You can upload objects directly to Amazon S3 Glacier, or use Amazon S3 lifecycle policies to transfer data between any of the Amazon S3 storage classes for active data (Amazon S3 Standard, Amazon S3 Intelligent-Tiering, Amazon S3 Standard-IA, and Amazon S3 One Zone-IA) and Amazon S3 Glacier.
- **Amazon S3 Glacier Deep Archive** – Amazon S3 Glacier Deep Archive is the lowest-cost storage class for Amazon S3. It supports long-term retention and digital preservation for data that might be accessed once or twice in a year. It is designed for customers — particularly customers in highly regulated industries, such as financial services, healthcare, and public sectors — that retain datasets for 7–10 years (or more) to meet regulatory compliance requirements. Amazon S3 Glacier Deep Archive can also be used for backup and disaster recovery use cases. It is a cost-effective and easy-to-manage alternative to magnetic tape systems, whether these tape systems are on-premises libraries or off-premises services. Amazon S3 Glacier Deep Archive complements Amazon S3 Glacier, and it is also designed to provide 11 9s of durability. All objects that are stored in Amazon S3 Glacier Deep Archive are replicated and stored across at least three geographically dispersed Availability Zones, and these objects can be restored within 12 hours.

For more information see, [Amazon S3 storage classes](#).

## Amazon S3 bucket URLs (two styles)



To upload your data:

1. Create a **bucket** in an AWS Region.
2. Upload almost any number of **objects** to the bucket.

Bucket path-style URL endpoint:

`https://s3.ap-northeast-1.amazonaws.com/bucket-name`

Region code                              Bucket name

Bucket virtual hosted-style URL endpoint:

`https://bucket-name.s3-ap-northeast-1.amazonaws.com`

Bucket name                              Region code

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

25

To use Amazon S3 effectively, you must understand a few simple concepts. First, Amazon S3 stores data inside **buckets**. Buckets are essentially the prefix for a set of files, and must be uniquely named across all of Amazon S3 globally. Buckets are logical containers for objects. You can have one or more buckets in your account. You can control access for each bucket—who can create, delete, and list objects in the bucket. You can also view access logs for the bucket and its objects, and choose the geographical region where Amazon S3 stores the bucket and its contents.

To upload your data (such as photos, videos, or documents), create a bucket in an AWS Region, and then upload almost any number of objects to the bucket.

In the example, Amazon S3 was used to create a bucket in the Tokyo Region, which is identified within AWS formally by its Region code: *ap-northeast-1*

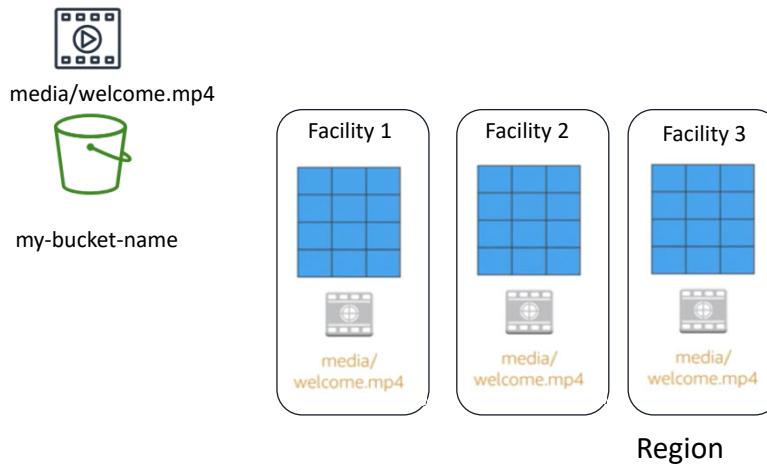
The URL for a bucket is structured like the examples. You can use two different URL styles to refer to buckets.

Amazon S3 refers to files as *objects*. As soon as you have a bucket, you can store almost any number of objects inside it. An object is composed of data and any metadata that describes that file, including a URL. To store an object in Amazon S3, you upload the file that you want to store to a bucket.

When you upload a file, you can set permissions on the data and any metadata.

In this example, the object *Preview2.mp4* is stored inside the bucket. The URL for the file includes the object name at the end.

## Data is redundantly stored in the Region



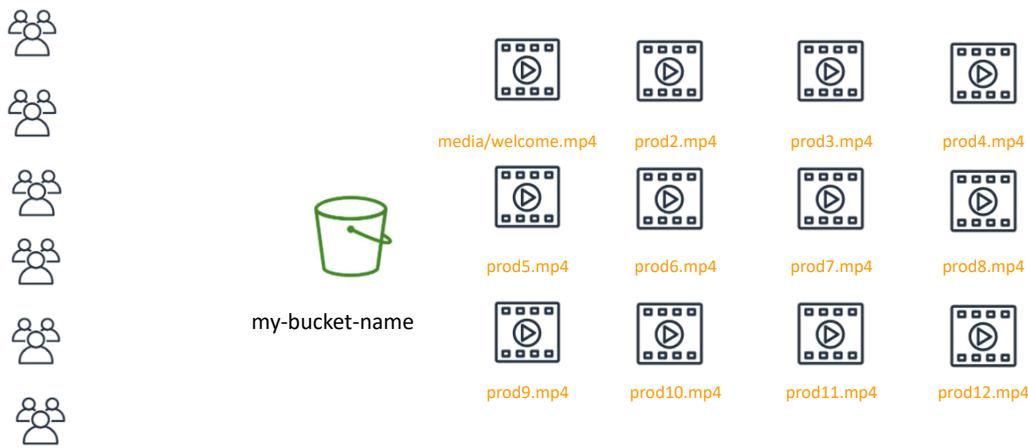
© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

26

When you create a bucket in Amazon S3, it is associated with a specific AWS Region. When you store data in the bucket, it is redundantly stored across multiple AWS facilities within your selected Region.

Amazon S3 is designed to durably store your data, even if there is concurrent data loss in two AWS facilities.

## Designed for seamless scaling



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

27

Amazon S3 automatically manages the storage behind your bucket while your data grows. You can get started immediately, and your data storage will grow with your application needs.

Amazon S3 also scales to handle a high volume of requests. You do not need to provision the storage or throughput, and you are billed only for what you use.

## Access the data anywhere



AWS Management  
Console



AWS Command Line  
Interface



SDK

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

28

You can access Amazon S3 through the console, AWS Command Line Interface (AWS CLI), or AWS SDK. You can also access the data in your bucket directly by using REST-based endpoints.

The endpoints support HTTP or HTTPS access. To support this type of URL-based access, Amazon S3 bucket names must be globally unique and Domain Name Server (DNS)-compliant.

Also, object keys should use characters that are safe for URLs.

## Common use cases



- Storing application assets
- Static web hosting
- Backup and disaster recovery (DR)
- Staging area for big data
- *Many more....*



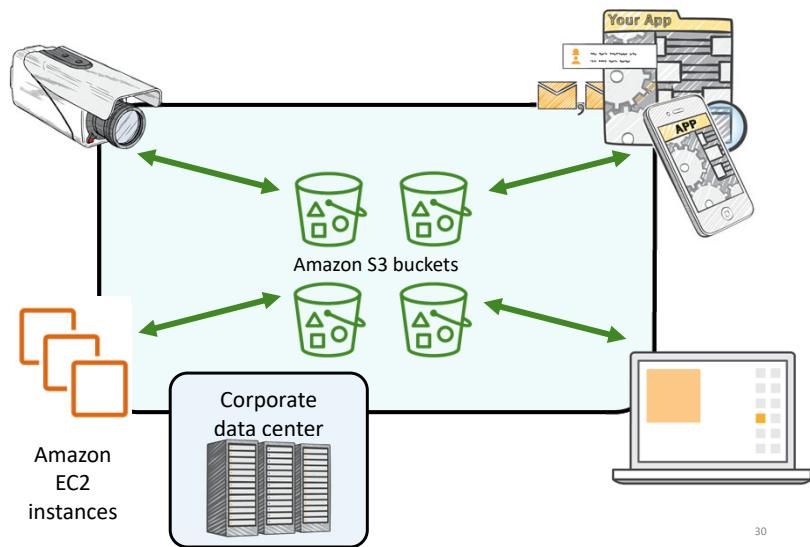
This flexibility to store a virtually unlimited amount of data—and to access that data from anywhere—means that Amazon S3 is suitable for a variety of scenarios. You will now consider some use cases for Amazon S3:

- As a location for any application data, Amazon S3 buckets provide a shared location for storing objects that any instances of your application can access—including applications on Amazon EC2 or even traditional servers. This feature can be useful for user-generated media files, server logs, or other files that your application must store in a common location. Also, because the content can be fetched directly over the internet, you can offload serving that content from your application and enable clients to directly fetch the data from Amazon S3 themselves.
- For static web hosting, Amazon S3 buckets can serve the static contents of your website, including HTML, CSS, JavaScript, and other files.
- The high durability of Amazon S3 makes it a good candidate for storing backups of your data. For greater availability and disaster recovery capability, Amazon S3 can even be configured to support cross-Region replication so that data in an Amazon S3 bucket in one Region can be automatically replicated to another Amazon S3 Region.

# Amazon S3 common scenarios



- Backup and storage
- Application hosting
- Media hosting
- Software delivery



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

30

**Backup and storage** – Provide data backup and storage services for others

**Application hosting** – Provide services that deploy, install, and manage web applications

**Media hosting** – Build a redundant, scalable, and highly available infrastructure that hosts video, photo, or music uploads and downloads

**Software delivery** – Host your software applications that customers can download

## Amazon S3 pricing



- Pay only for what you use, including –
  - GBs per month
  - Transfer OUT to other Regions
  - PUT, COPY, POST, LIST, and GET requests
- You do not pay for –
  - Transfers IN to Amazon S3
  - Transfers OUT from Amazon S3 to Amazon CloudFront or Amazon EC2 in the same Region

With Amazon S3, specific costs vary depending on the Region and the specific requests that were made. You pay only for what you use, including gigabytes per month; transfer out of other Regions; and PUT, COPY, POST, LIST, and GET requests.

As a general rule, you pay only for transfers that cross the boundary of your Region, which means you do not pay for transfers **in to** Amazon S3 or transfers out from Amazon S3 to Amazon CloudFront edge locations within that same Region.

To estimate Amazon S3 costs, consider the following:

**1. Storage class type –**

- Standard storage is designed for:
  - 11 9s of durability
  - Four 9s of availability
- S3 Standard-Infrequent Access (S-IA) is designed for:
  - 11 9s of durability
  - Three 9s of availability

**2. Amount of storage –**

- The number and size of objects

When you begin to estimate the costs of Amazon S3, you must consider the following:

**1. Storage class type –**

- **Standard storage** is designed to provide 11 9s of durability and four 9s of availability.
- **S3 Standard – Infrequent Access (S-IA)** is a storage option within Amazon S3 that you can use to reduce your costs by storing less frequently accessed data at slightly lower levels of redundancy than Amazon S3 standard storage. Standard – Infrequent Access is designed to provide the same 11 9s of durability as Amazon S3, with three 9s of availability in a given year. Each class has different rates.

**2. Amount of storage –** The number and size of objects stored in your Amazon S3 buckets.

### 3. Requests –

- The number and type of requests (**GET, PUT, COPY**)
- Type of requests:
  - Different rates for GET requests than other requests.

### 4. Data transfer –

- Pricing is based on the amount of data that is transferred out of the Amazon S3 Region
  - Data transfer in is free, but you incur charges for data that is transferred out.

### 3. Requests – Consider the number and type of requests. GET requests incur charges at different rates than other requests, such as PUT and COPY requests.

- **GET** – Retrieves an object from Amazon S3. You must have READ access to use this operation.
- **PUT** – Adds an object to a bucket. You must have WRITE permissions on a bucket to add an object to it.
- **COPY** – Creates a copy of an object that is already stored in Amazon S3. A COPY operation is the same as performing a GET and then a PUT.

### 4. Data transfer – Consider the amount of data that is transferred out of the Amazon S3 Region. Remember that data transfer in is free, but there is a charge for data transfer out.

## Section 2 key takeaways



- Amazon S3 is a fully managed cloud storage service.
- You can store a virtually unlimited number of objects.
- You pay for only what you use.
- You can access Amazon S3 at any time from anywhere through a URL.
- Amazon S3 offers rich security controls.

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

You have completed an introduction to Amazon S3, including key features and some common use cases.

To learn more about Amazon S3, see [Amazon S3](#).

## Recorded demo: Amazon Simple Storage System

35



### Set up demo

Amazon S3

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

Now, take a moment to watch the [Amazon S3 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. The demonstration shows how to:

- Create an Amazon S3 bucket
- Upload files and create folders
- Change bucket settings

The demonstration also reviews some of the more commonly used settings for an S3 bucket.

Module 7: Storage

## Section 3: Amazon Elastic File System (Amazon EFS)

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



### Introduce Section 3: Amazon Elastic File System (Amazon EFS)

Amazon EFS implements storage for EC2 instances that multiple virtual machines can access at the same time. It is implemented as a shared file system that uses the Network File System (NFS) protocol.



## Amazon Elastic File System (Amazon EFS)

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

37

**Amazon Elastic File System (Amazon EFS)** provides simple, scalable, elastic file storage for use with AWS services and on-premises resources. It offers a simple interface that enables you to create and configure file systems quickly and easily.

Amazon EFS is built to dynamically scale on demand without disrupting applications—it will grow and shrink automatically as you add and remove files. It is designed so that your applications have the storage they need, when they need it.

## Amazon EFS features



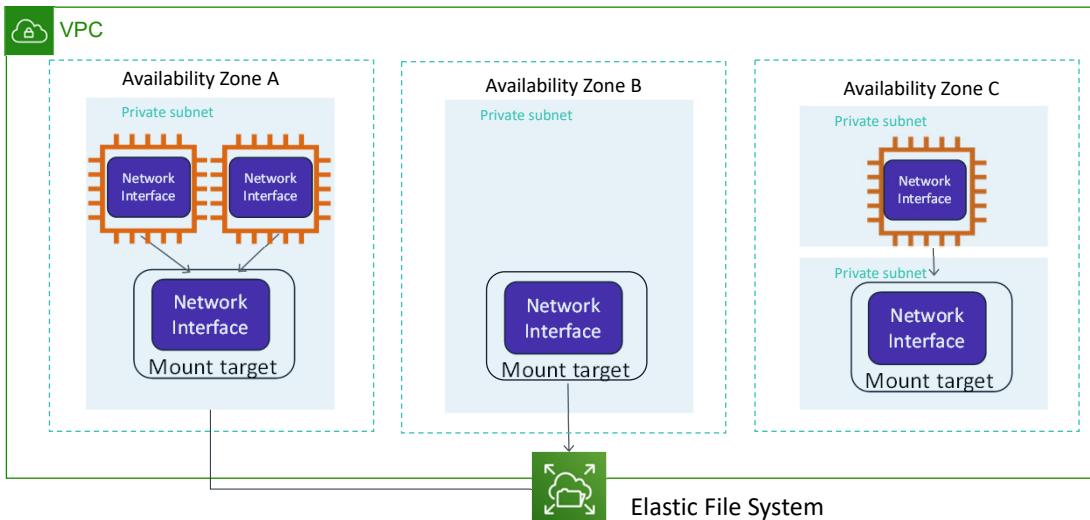
- File storage in the AWS Cloud
- Works well for big data and analytics, media processing workflows, content management, web serving, and home directories
- Petabyte-scale, low-latency file system
- Shared storage
- Elastic capacity
- Supports Network File System (NFS) versions 4.0 and 4.1 (NFSv4)
- Compatible with all Linux-based AMIs for Amazon EC2

Amazon EFS is a fully managed service that makes it easy to set up and scale file storage in the AWS Cloud. You can use Amazon EFS to build a file system for big data and analytics, media processing workflows, content management, web serving, and home directories.

You can create file systems that are accessible to Amazon EC2 instances through a file system interface (using standard operating system file I/O APIs). These file systems support full file system access semantics, such as strong consistency and file locking.

Amazon EFS file systems can automatically scale from gigabytes to petabytes of data without the need to provision storage. Thousands of Amazon EC2 instances can access an Amazon EFS file system at the same time, and Amazon EFS is designed to provide consistent performance to each Amazon EC2 instance. Amazon EFS is also designed to be highly durable and highly available. Amazon EFS requires no minimum fee or setup costs, and you pay only for the storage that you use.

# Amazon EFS architecture



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

39

Amazon EFS provides file storage in the cloud. With Amazon EFS, you can create a file system, mount the file system on an Amazon EC2 instance, and then read and write data from to and from your file system. You can mount an Amazon EFS file system in your VPC, through NFS versions 4.0 and 4.1 (NFSv4).

You can access your Amazon EFS file system concurrently from Amazon EC2 instances in your VPC, so applications that scale beyond a single connection can access a file system. Amazon EC2 instances that run in multiple Availability Zones within the same AWS Region can access the file system, so many users can access and share a common data source.

In the diagram, the VPC has three Availability Zones, and each Availability Zone has one mount target that was created in it. We recommend that you access the file system from a mount target within the same Availability Zone. One of the Availability Zones has two subnets. However, a mount target is created in only one of the subnets.

- ① Create your Amazon EC2 resources and launch your Amazon EC2 instance.
- ② Create your Amazon EFS file system.
- ③ Create your mount targets in the appropriate subnets.
- ④ Connect your Amazon EC2 instances to the mount targets.
- ⑤ Verify the resources and protection of your AWS account.

You must complete five steps to create and use your first Amazon EFS file system, mount it on an Amazon EC2 instance in your VPC, and test the end-to-end setup:

1. Create your Amazon EC2 resources and launch your instance. (Before you can launch and connect to an Amazon EC2 instance, you must create a key pair, unless you already have one.)
2. Create your Amazon EFS file system.
3. In the appropriate subnets, create your mount targets.
4. Next, connect to your Amazon EC2 instance and mount the Amazon EFS file system.
5. Finally, clean up your resources and protect your AWS account.

## File system

- Mount target
  - Subnet ID
  - Security groups
  - One or more per file system
  - Create in a VPC subnet
  - One per Availability Zone
  - Must be in the same VPC
- Tags
  - Key-value pairs



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

41

In Amazon EFS, a file system is the primary resource. Each file system has properties such as:

- ID
- Creation token
- Creation time
- File system size in bytes
- Number of mount targets that are created for the file system
- File system state

Amazon EFS also supports other resources to configure the primary resource. These resources include mount targets and tags.

**Mount target:** To access your file system, you must create mount targets in your VPC. Each mount target has the following properties:

- The mount target ID
- The subnet ID for the subnet where it was created
- The file system ID for the file system where it was created

- An IP address where the file system can be mounted
- The mount target state

You can use the IP address or the Domain Name System (DNS) name in your mount command.

**Tags:** To help organize your file systems, you can assign your own metadata to each of the file systems that you create. Each tag is a key-value pair.

Think of mount targets and tags as subresources that do not exist unless they are associated with a file system.

## Section 3 key takeaways



- Amazon EFS provides file storage over a network.
- Perfect for big data and analytics, media processing workflows, content management, web serving, and home directories.
- Fully managed service that eliminates storage administration tasks.
- Accessible from the console, an API, or the CLI.
- Scales up or down as files are added or removed and you pay for what you use.

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

You have completed an introduction to Amazon EFS, including key features and key resources. Amazon EFS provides file storage in the cloud that works well for big data and analytics, media processing workflows, content management, web serving, and home directories.

Amazon EFS scales up or down when files are added or removed, and you pay for only what you are using.

Amazon EFS is a fully managed service that is accessible from the console, an API, or the AWS CLI.

To learn more about Amazon EFS, see [Amazon EFS](#)

## Recorded demo: Amazon Elastic File System

43



### Set up demo

Amazon Elastic File System  
(Amazon EFS)

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

Now, take a moment to watch the [Amazon EFS demo](#). The recording runs a little over 6 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. The demonstration shows how to:

- Create an Elastic File System (EFS) implementation in a Virtual Private Cloud
- Attach the EFS
- Configure security and performance settings for the EFS implementation

The demonstration also reviews .how to get specific instructions for how to validate your EFS installation so you can connect to EC2 instances.

Module 7: Storage

## Section 4: Amazon S3 Glacier

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



### Introducing Section 4: Amazon S3 Glacier

Amazon S3 Glacier is a secure, durable, and extremely low-cost cloud storage service for data archiving and long-term backup.



## Amazon S3 Glacier

This section covers Amazon S3 Glacier.

## Amazon S3 Glacier review



Amazon S3 Glacier is a data archiving service that is designed for **security**, **durability**, and an **extremely low cost**.

- Amazon S3 Glacier is designed to provide 11 9s of durability for objects.
- It supports the encryption of data in transit and at rest through Secure Sockets Layer (SSL) or Transport Layer Security (TLS).
- The Vault Lock feature enforces compliance through a policy.
- Extremely low-cost design works well for long-term archiving.
  - Provides three options for access to archives—expedited, standard, and bulk—retrieval times range from a few minutes to several hours.

When you use Amazon S3 Glacier to archive data, you can store your data at an extremely low cost (even in comparison to Amazon S3), but you cannot retrieve your data immediately when you want it.

Data that is stored in Amazon S3 Glacier can take several hours to retrieve, which is why it works well for archiving.

There are three key Amazon S3 Glacier terms you should be familiar with:

- **Archive** – Any object (such as a photo, video, file, or document) that you store in Amazon S3 Glacier. It is the base unit of storage in Amazon S3 Glacier. Each archive has its own unique ID and it can also have a description.
- **Vault** – A container for storing archives. When you create a vault, you specify the vault name and the Region where you want to locate the vault.
- **Vault access policy** – Determine who can and cannot access the data that is stored in the vault, and what operations users can and cannot perform. One vault access permissions policy can be created for each vault to manage access permissions for that vault. You can also use a vault lock policy to make sure that a vault cannot be altered.

Each vault can have one vault access policy and one vault lock policy that are attached to it.

You have three options for retrieving data, each with varying access times and cost:

- **Expedited** retrievals are typically made available within 1–5 minutes (highest cost).
- **Standard** retrievals typically complete within 3–5 hours (less time than expedited, more time than bulk).
- **Bulk** retrievals typically complete within 5–12 hours (lowest cost).

You might compare these options to choosing the cost for shipping a package by using the most economical method for your needs.

# Amazon S3 Glacier



- Storage service for low-cost data archiving and long-term backup
- You can configure lifecycle archiving of Amazon S3 content to Amazon S3 Glacier
- Retrieval options –
  - Standard: 3–5 hours
  - Bulk: 5–12 hours
  - Expedited: 1–5 minutes



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

47

Amazon S3 Glacier's data archiving means that although you can store your data at an extremely low cost (even in comparison to Amazon S3), you cannot retrieve your data immediately when you want it.

Data stored in Amazon S3 Glacier can take several hours to retrieve.

You should be familiar with three key Amazon S3 Glacier terms:

- **Archive:** Any object such as a photo, video, file, or document that you store in Amazon S3 Glacier. It is the base unit of storage in Amazon S3 Glacier. Each archive has its own unique ID and can also have a description.
- **Vault:** A container for storing archives. When you create a vault, you specify the vault name and the region in which you would like to locate the vault.
- **Vault Access Policy:** Determine who can and cannot access the data stored in the vault and what operations users can and cannot perform. One vault access policy can be created for each vault to manage access permissions for that vault. You can also use a vault lock policy to make sure a vault cannot be altered. Each vault can have one vault access policy and one vault lock policy that is attached to it.

Three options are available for retrieving data with varying access times and cost: expedited, standard, and bulk retrievals. They are listed as follows:

- **Expedited** retrievals are typically made available within 1 – 5 minutes (highest cost).
- **Standard** retrievals typically complete within 3 – 5 hours (less than expedited, more than bulk).
- **Bulk** retrievals typically complete within 5 – 12 hours (lowest cost).

Compare it to choosing the cost to most economically ship a package.

-  Media asset archiving
-  Healthcare information archiving
-  Regulatory and compliance archiving
-  Scientific data archiving
-  Digital preservation
-  Magnetic tape replacement

## Media asset archiving

Media assets—such as video and news footage—require durable storage and can grow to many petabytes over time. Amazon S3 Glacier enables you to archive older media content affordably and then move it to Amazon S3 for distribution when it is needed.

## Healthcare information archiving

To meet regulatory requirements, hospital systems must retain petabytes of patient records—such as Low-Income Subsidy (LIS) information, picture archiving and communication system (PACS) data, or Electronic Health Records (EHR)—for decades. Amazon S3 Glacier can help you reliably archive patient record data securely at a very low cost.

## Regulatory and compliance archiving

Many enterprises, like those in financial services and healthcare, must retain regulatory and compliance archives for extended durations. Amazon S3 Glacier Vault Lock can help you set compliance controls so you can work towards meeting your compliance objectives, such as the U.S. Securities and Exchange Commission (SEC) Rule 17a-4(f).

## Scientific data archiving

Research organizations generate, analyze, and archive large amounts of data. By using Amazon S3 Glacier, you can reduce the complexities of hardware and facility management

and capacity planning.

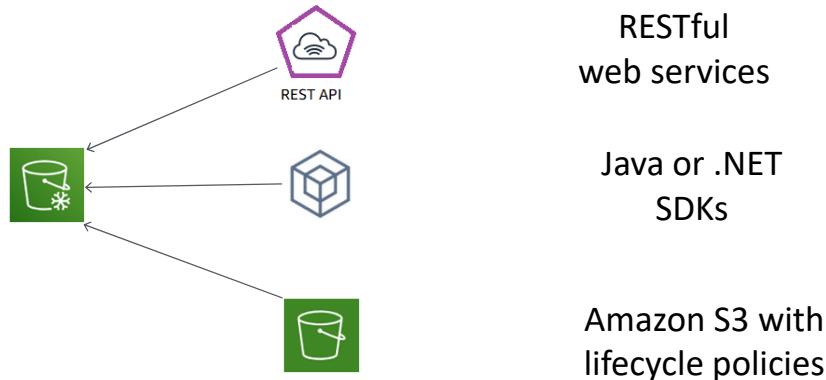
### **Digital preservation**

Libraries and government agencies must handle data integrity challenges in their digital preservation efforts. Unlike traditional systems—which can require laborious data verification and manual repair—Amazon S3 Glacier performs regular, systematic data integrity checks, and it is designed to be automatically self-healing.

### **Magnetic tape replacement**

On-premises or offsite tape libraries can lower storage costs, but they can require large upfront investments and specialized maintenance. Amazon S3 Glacier has no upfront cost and reduces the cost and burden of maintenance.

# Using Amazon S3 Glacier



To store and access data in Amazon S3 Glacier, you can use the AWS Management Console. However, only a few operations—such as creating and deleting vaults, and creating and managing archive policies—are available in the console.

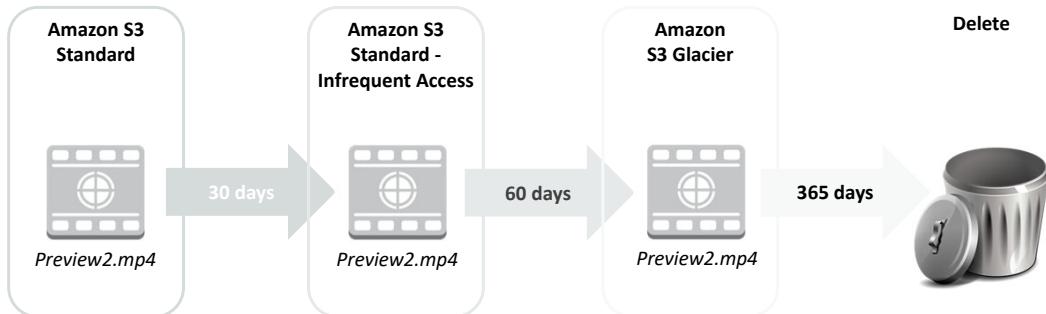
For almost all other operations and interactions with Amazon S3 Glacier, you must use either the Amazon S3 Glacier REST APIs, the AWS Java or .NET SDKs, or the AWS CLI.

You can also use lifecycle policies to archive data into Amazon S3 Glacier. Next, you will learn about lifecycle policies.

# Lifecycle policies



Amazon S3 lifecycle policies enable you to delete or move objects based on age.



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

50

You should automate the lifecycle of the data that you store in Amazon S3. By using lifecycle policies, you can cycle data at regular intervals between different Amazon S3 storage types. This automation reduces your overall cost, because you pay less for data as it becomes less important with time.

In addition to setting lifecycle rules per object, you can also set lifecycle rules per bucket.

Consider an example of a lifecycle policy that moves data as it ages from **Amazon S3 Standard** to **Amazon S3 Standard – Infrequent Access**, and finally, into **Amazon S3 Glacier** before it is deleted. Suppose that a user uploads a video to your application and your application generates a thumbnail preview of the video. This video preview is stored to Amazon S3 Standard, because it is likely that the user wants to access it right away.

Your usage data indicates that most thumbnail previews are not accessed after 30 days. Your lifecycle policy takes these previews and moves them to Amazon S3 – Infrequent Access after 30 days. After another 30 days elapse, the preview is unlikely to be accessed again. The preview is then moved to Amazon S3 Glacier, where it remains for 1 year. After 1 year, the preview is deleted. The important thing is that the lifecycle policy manages all this movement automatically.

To learn more, see [Object lifecycle management](#)

## Storage comparison



	Amazon S3	Amazon S3 Glacier
Data Volume	No limit	No limit
Average Latency	ms	minutes/hours
Item Size	5 TB maximum	40 TB maximum
Cost/GB per Month	Higher cost	Lower cost
Billed Requests	PUT, COPY, POST, LIST, and GET	UPLOAD and retrieval
Retrieval Pricing	¢ Per request	¢¢ Per request and per GB

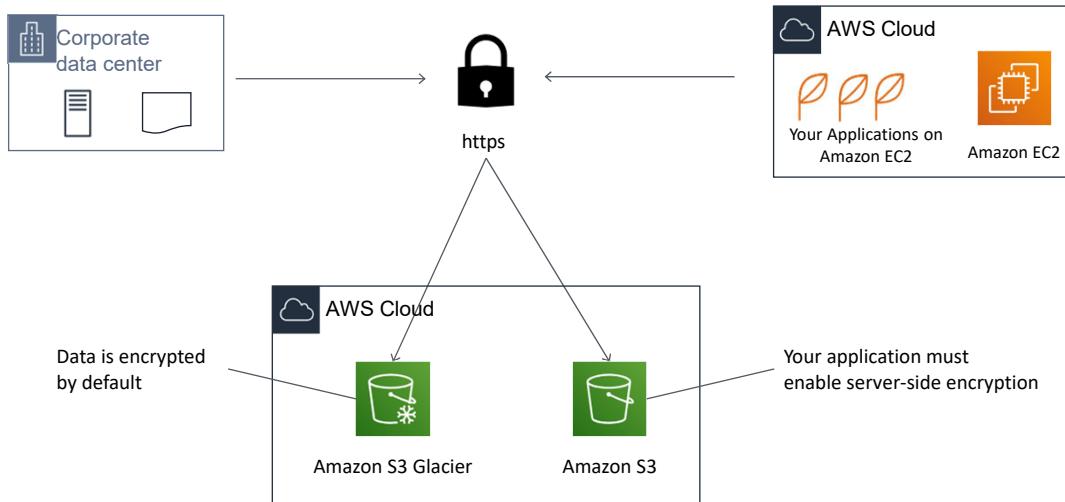
© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

51

While **Amazon S3** and **Amazon S3 Glacier** are both object storage solutions that enable you to store a virtually unlimited amount of data, they have some critical differences between them. The chart outlines some of these differences.

1. Be careful when you decide which storage solution is correct for your needs. These two services serve very different storage needs. Amazon S3 is designed for frequent, low-latency access to your data, but Amazon S3 Glacier is designed for low-cost, long-term storage of infrequently accessed data.
2. The maximum item size in Amazon S3 is 5 TB, but Amazon S3 Glacier can store items that are up to 40 TB.
3. Because Amazon S3 gives you faster access to your data, the storage cost per gigabyte is higher than it is with Amazon S3 Glacier.
4. While both services have per-request charges, Amazon S3 charges for **PUT, COPY, POST, LIST, GET** operations. In contrast, Amazon S3 Glacier charges for **UPLOAD and retrieval** operations.
5. Because Amazon S3 Glacier was designed for less-frequent access to data, it costs more for each retrieval request than Amazon S3.

# Server-side encryption



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

52

Another important difference between Amazon S3 and Amazon S3 Glacier is how data is encrypted. Server-side encryption is focused on protecting data at rest. With both solutions, you can securely transfer your data over HTTPS. Any data that is archived in Amazon S3 Glacier is encrypted by default. With Amazon S3, your application must initiate server-side encryption. You can accomplish server-side encryption in Amazon S3 in several ways:

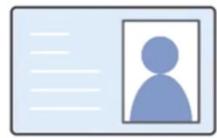
- Server-side encryption with **Amazon S3-managed encryption keys (SSE-S3)** employs strong multi-factor encryption. Amazon S3 encrypts each object with a unique key. As an additional safeguard, it encrypts the key with a main key that it regularly rotates. Amazon S3 server-side encryption uses one of the strongest block ciphers available, 256-bit Advanced Encryption Standard (AES-256), to encrypt your data.
- Using server-side encryption with **Customer-provided Encryption Keys (SSE-C)** enables you to set your own encryption keys. You include the encryption key as part of your request, and Amazon S3 manages both encryption (as it writes to disks), and decryption (when you access your objects).
- Using server-side encryption with **AWS Key Management Service (AWS KMS)** is a service that combines secure, highly available hardware and software to provide a key management system that is scaled for the cloud. AWS KMS uses **Customer Master Keys (CMKs)** to encrypt your Amazon S3 objects. You use AWS KMS through the **Encryption Keys** section in the IAM console. You can also access AWS KMS through the API to centrally create encryption keys, define the policies that control how keys can be used,

and audit key usage to prove that they are being used correctly. You can use these keys to protect your data in Amazon S3 buckets.

# Security with Amazon S3 Glacier



**Amazon S3  
Glacier**



**Control access with  
IAM**



**Amazon S3 Glacier encrypts  
your data with AES-256**



**Amazon S3 Glacier manages  
your keys for you**

By default, only you can access your data. You can enable and control access to your data in Amazon S3 Glacier by using IAM. You set up an IAM policy that specifies user access.

## Section 4 key takeaways



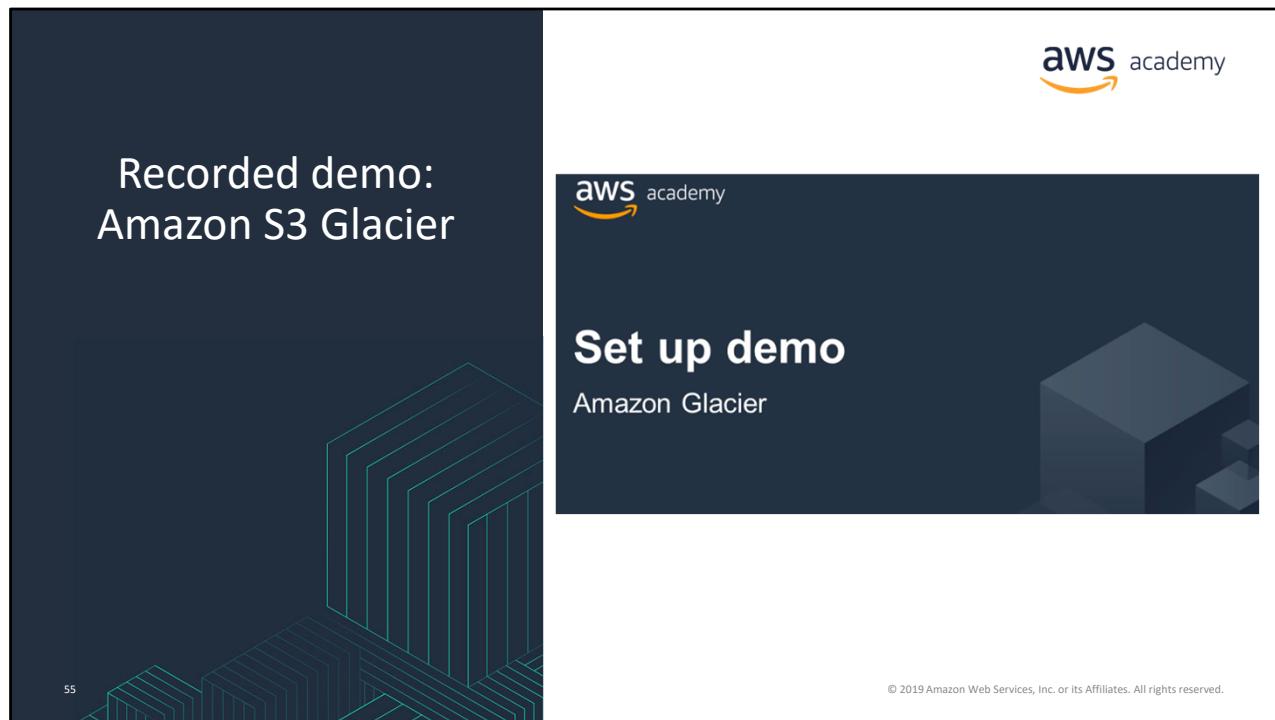
54

- Amazon S3 Glacier is a data archiving service that is designed for security, durability, and an extremely low cost.
- Amazon S3 Glacier pricing is based on Region.
- Its extremely low-cost design works well for long-term archiving.
- The service is designed to provide 11 9s of durability for objects.

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

You have completed an introduction to Amazon S3 Glacier, which included key differences between Amazon S3 and Amazon S3 Glacier.

To learn more about Amazon S3 Glacier, see [Glacier](#).



Now, take a moment to watch the [Amazon Glacier demo](#). The recording runs a little over 2 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. The demonstration shows how to:

- Create an Amazon Glacier vault
- Upload archived items to the vault using a third-party graphical interface tool
-

## Activity: Storage Case Studies



Photo by panumas nikhomkhai from Pexels.

56

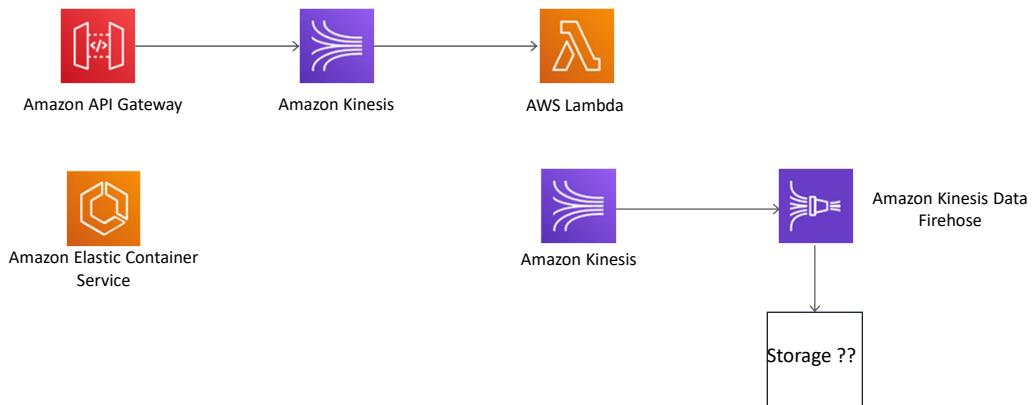
© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

In this educator-led activity, you will be asked to log in to the AWS Management Console. The activity instructions are on the next slide. You will be challenged to answer five questions. The educator will lead the class in a discussion of each question, and reveal the correct answers.

# Storage case study activity



**Case 1:** A data analytics company for travel sites must store billions of customer events per day. They use the data analytics services that are in the diagram. The following diagram illustrates their architecture.



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

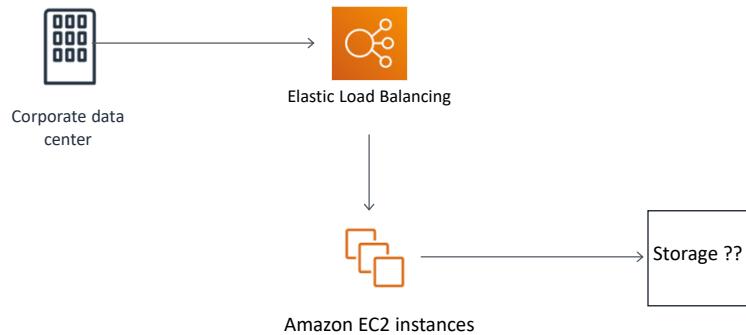
57

Break into groups of four or five people. Review the assigned case study. Create a presentation that describes the best storage solution for the organization that is described in your group's case. Your presentation should include the key factors that you considered when you selected the storage technology, and any factors that might change your recommendation.

## Storage case study activity



**Case 2:** A collaboration software company processes email for enterprise customers. They have more than 250 enterprise customers and more than half a million users. They must store petabytes of data for their customers. The following diagram illustrates their architecture.

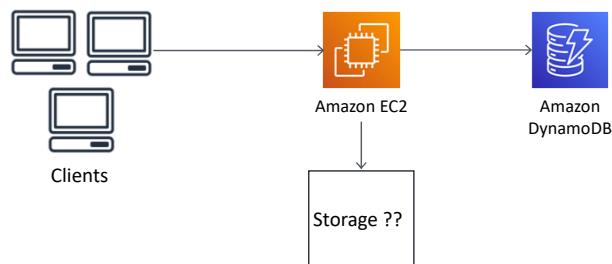


Break into groups of four or five people. Review the assigned case study. Create a presentation that describes the best storage solution for the organization that is described in your group's case. Your presentation should include the key factors that you considered when you selected the storage technology, and any factors that might change your recommendation.

## Storage case study activity



**Case 3:** A data protection company must be able to ingest and store large amounts of customer data and help their customers meet compliance requirements. They use Amazon EC2 for scalable compute and Amazon DynamoDB for duplicate data and metadata lookups. The following diagram illustrates their architecture.



Break into groups of four or five people. Review the assigned case study. Create a presentation that describes the best storage solution for the organization that is described in your group's case. Your presentation should include the key factors that you considered when you selected the storage technology, and any factors that might change your recommendation.

Module 7: Storage

## Module wrap-up

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



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

## Module summary



In summary, in this module, you learned how to:

- Identify the different types of storage
- Explain Amazon S3
- Identify the functionality in Amazon S3
- Explain Amazon EBS
- Identify the functionality in Amazon EBS
- Perform functions in Amazon EBS to build an Amazon EC2 storage solution
- Explain Amazon EFS
- Identify the functionality in Amazon EFS
- Explain Amazon S3 Glacier
- Identify the functionality in Amazon S3 Glacier
- Differentiate between Amazon EBS, Amazon S3, Amazon EFS, and Amazon S3 Glacier

In summary, in this module, you learned how to:

- Identify the different types of storage
- Explain Amazon S3
- Identify the functionality in Amazon S3
- Explain Amazon EBS
- Identify the functionality in Amazon EBS
- Perform functions in Amazon EBS to build an Amazon EC2 storage solution
- Explain Amazon EFS
- Identify the functionality in Amazon EFS
- Explain Amazon S3 Glacier
- Identify the functionality in Amazon S3 Glacier
- Differentiate between Amazon EBS, Amazon S3, Amazon EFS, and Amazon S3 Glacier

## Complete the knowledge check



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

62

Complete the knowledge check for this module.

## Sample exam question



A company wants to store data that is not frequently accessed. What is the best and cost-effective solution that should be considered?

- A. AWS Storage Gateway
- B. Amazon Simple Storage Service Glacier
- C. Amazon Elastic Block Store (Amazon EBS)
- D. Amazon Simple Storage Service (Amazon S3)

Look at the answer choices and rule them out based on the keywords that were previously highlighted.

## Additional resources



- [AWS Storage page](#)
- [Storage Overview](#)
- [Recovering files from an Amazon EBS volume backup](#)
- [Confused by AWS Storage Options? S3, EFS, EBS Explained](#)

If you want to learn more about the topics covered in this module, you might find the following additional resources helpful:

- [AWS Storage page](#)
- [Storage Overview](#)
- [Recovering files from an Amazon EBS volume backup](#)
- [Confused by AWS Storage Options? S3, EFS, EBS Explained](#)

# Thank you

© 2019 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections or feedback on the course, please email us at: [aws-course-feedback@amazon.com](mailto:aws-course-feedback@amazon.com). For all other questions, contact us at: <https://aws.amazon.com/contact-us/aws-training/>. All trademarks are the property of their owners.



Thanks for participating!

AWS Academy Cloud Foundations

# Module 8: Databases

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



Welcome to Module 8: Databases

# Module overview



## Topics

- Amazon Relational Database Service (Amazon RDS)
- Amazon DynamoDB
- Amazon Redshift
- Amazon Aurora

## Demos

- Amazon RDS console
- Amazon DynamoDB console

## Lab

- Lab 5: Build Your DB Server and Interact with Your DB Using an App

## Activity

- Database case studies



## Knowledge check

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

2

The business world is constantly changing and evolving. By accurately recording, updating, and tracking data on an efficient and regular basis, companies can use the immense potential from the insights that they obtain from their data. Database management systems are the crucial link for managing this data. Like other cloud services, cloud databases offer significant cost advantages over traditional database strategies.

In this module, you will learn about Amazon Relational Database Service (or Amazon RDS), Amazon DynamoDB, Amazon Redshift, and Amazon Aurora.

This module will address the following topics:

- Amazon Relational Database Service (Amazon RDS)
- Amazon DynamoDB
- Amazon Redshift
- Amazon Aurora

The module includes two recorded demonstrations that will show you how to access and interact with Amazon RDS and Amazon DynamoDB by using the AWS Management Console.

The module also includes a hands-on lab where you will set up an Amazon RDS database solution.

The module also includes an activity that challenges you to select the appropriate database service for a business case.

Finally, you will be asked to complete a knowledge check that will test your understanding of the key concepts that are covered in this module.

# Module objectives



After completing this module, you should be able to:

- Explain Amazon Relational Database Service (Amazon RDS)
- Identify the functionality in Amazon RDS
- Explain Amazon DynamoDB
- Identify the functionality in Amazon DynamoDB
- Explain Amazon Redshift
- Explain Amazon Aurora
- Perform tasks in an RDS database, such as launching, configuring, and interacting

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

3

In this module, you will learn about key concepts that are related to database solutions, including:

- Understanding the different database services in the cloud.
- Discovering the differences between unmanaged and managed database solutions.
- Understanding the differences between Structured Query Language (or SQL) and NoSQL databases.
- Comparing the availability differences of alternative database solutions.

The goal of this module is to help you understand the database resources that are available to power your solution. You will also review the different service features that are available, so you can begin to understand how different choices impact things like solution availability.

After completing this module, you should be able to:

- Explain Amazon Relational Database Service (Amazon RDS)
- Identify the functionality in Amazon RDS
- Explain Amazon DynamoDB
- Identify the functionality in Amazon DynamoDB
- Explain Amazon Redshift
- Explain Amazon Aurora
- Perform tasks in an RDS database, such as launching, configuring, and interacting

Module 8: Databases

## Section 1: Amazon Relational Database Service

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



Introducing Section 1: Amazon Relational Database Service.



## Amazon Relational Database Service (Amazon RDS)

Welcome to an introduction to the foundational database services that are available on Amazon Web Services (AWS). This module begins with Amazon Relational Database Service (Amazon RDS).

This section starts by reviewing the differences between a managed and unmanaged service in relation to Amazon RDS.

# Unmanaged versus managed services



## Unmanaged:

*Scaling, fault tolerance, and availability are managed by you.*



## Managed:

*Scaling, fault tolerance, and availability are typically built in to the service.*



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

6

AWS solutions typically fall into one of two categories: unmanaged or managed.

Unmanaged services are typically provisioned in discrete portions as specified by the user. You must manage how the service responds to changes in load, errors, and situations where resources become unavailable. Say that you launch a web server on an Amazon Elastic Compute Cloud (Amazon EC2) instance. Because Amazon EC2 is an unmanaged solution, that web server will not scale to handle increased traffic load or replace unhealthy instances with healthy ones unless you specify that it use a scaling solution, such as AWS Automatic Scaling. The benefit to using an unmanaged service is that you have more fine-tuned control over how your solution handles changes in load, errors, and situations where resources become unavailable.

Managed services require the user to configure them. For example, you create an Amazon Simple Storage Service (Amazon S3) bucket and then set permissions for it. However, managed services typically require less configuration. Say that you have a static website that you host in a cloud-based storage solution, such as Amazon S3. The static website does not have a web server. However, because Amazon S3 is a managed solution, features such as scaling, fault-tolerance, and availability would be handled automatically and internally by Amazon S3.

Now, you will look at the challenges of running an unmanaged, standalone relational database. Then, you will learn how Amazon RDS addresses these challenges.

## Challenges of relational databases

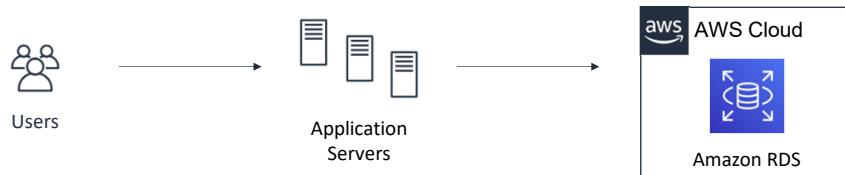


- Server maintenance and energy footprint
- Software installation and patches
- Database backups and high availability
- Limits on scalability
- Data security
- Operating system (OS) installation and patches



When you run your own relational database, you are responsible for several administrative tasks, such as server maintenance and energy footprint, software, installation and patching, and database backups. You are also responsible for ensuring high availability, planning for scalability, data security, and operating system (OS) installation and patching. All these tasks take resources from other items on your to-do list, and require expertise in several areas.

Managed service that sets up and operates a relational database in the cloud.

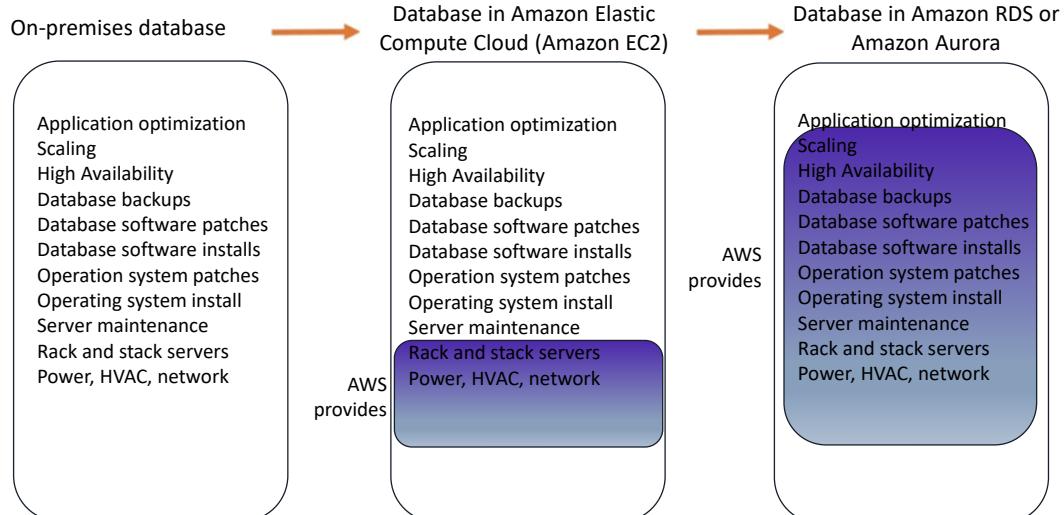


Amazon RDS is a managed service that sets up and operates a relational database in the cloud.

To address the challenges of running an unmanaged, standalone relational database, AWS provides a service that sets up, operates, and scales the relational database without any ongoing administration. Amazon RDS provides cost-efficient and resizable capacity, while automating time-consuming administrative tasks.

Amazon RDS enables you to focus on your application, so you can give applications the performance, high availability, security, and compatibility that they need. With Amazon RDS, your primary focus is your data and optimizing your application.

## From on-premises databases to Amazon RDS



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

9

### What does the term **managed services** mean?

When your database is on premises, the database administrator is responsible for everything. Database administration tasks include optimizing applications and queries; setting up the hardware; patching the hardware; setting up networking and power; and managing heating, ventilation, and air conditioning (HVAC).

If you move to a database that runs on an **Amazon Elastic Compute Cloud (Amazon EC2) instance**, you no longer need to manage the underlying hardware or handle data center operations. However, you are still responsible for patching the OS and handling all software and backup operations.

If you set up your database on **Amazon RDS or Amazon Aurora**, you reduce your administrative responsibilities. By moving to the cloud, you can automatically scale your database, enable high availability, manage backups, and perform patching. Thus, you can focus on what really matters most—optimizing your application.

# Managed services responsibilities



## You manage:

- Application optimization



## AWS manages:

- OS installation and patches
- Database software installation and patches
- Database backups
- High availability
- Scaling
- Power and racking and stacking servers
- Server maintenance



Amazon RDS

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

10

With Amazon RDS, you manage your application optimization. AWS manages installing and patching the operating system, installing and patching the database software, automatic backups, and high availability.

AWS also scales resources, manages power and servers, and performs maintenance.

Offloading these operations to the managed Amazon RDS service reduces your operational workload and the costs that are associated with your relational database. You will now go through a brief overview of the service and a few potential use cases.

# Amazon RDS DB instances



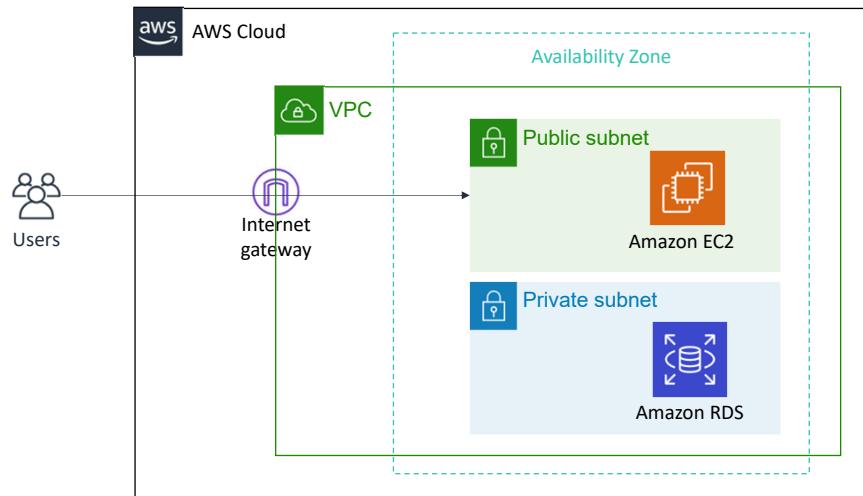
© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

11

The basic building block of Amazon RDS is the database instance. A **database instance** is an isolated database environment that can contain multiple user-created databases. It can be accessed by using the same tools and applications that you use with a standalone database instance. The resources in a database instance are determined by its database instance class, and the type of storage is dictated by the type of disks.

Database instances and storage differ in performance characteristics and price, which enable you to customize your performance and cost to the needs of your database. When you choose to create a database instance, you must first specify which database engine to run. Amazon RDS currently supports six databases: MySQL, Amazon Aurora, Microsoft SQL Server, PostgreSQL, MariaDB, and Oracle.

## Amazon RDS in a virtual private cloud (VPC)



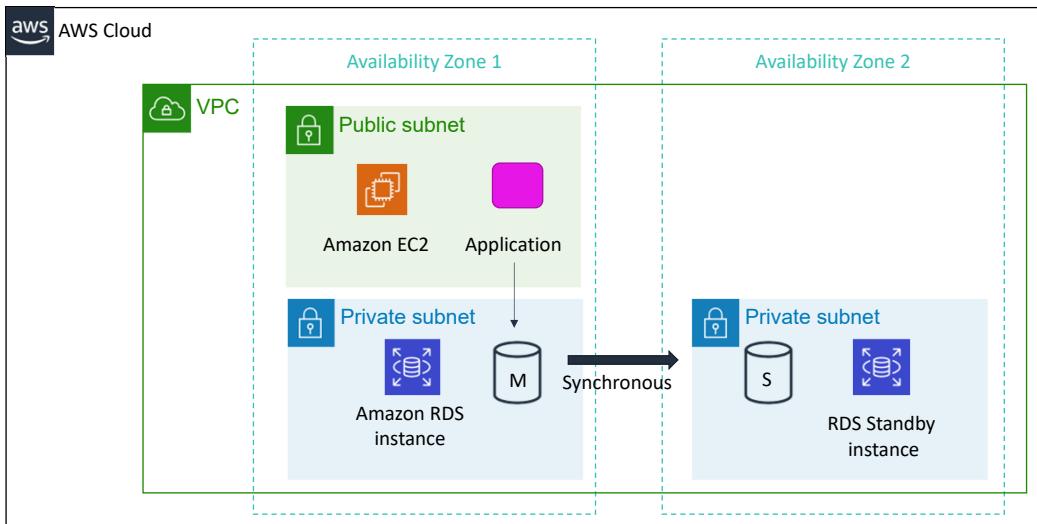
© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

12

You can run an instance by using **Amazon Virtual Private Cloud (Amazon VPC)**. When you use a virtual private cloud (VPC), you have control over your virtual networking environment.

You can select your own IP address range, create subnets, and configure routing and access control lists (ACLs). The basic functionality of Amazon RDS is the same whether or not it runs in a VPC. Usually, the database instance is isolated in a private subnet and is only made directly accessible to indicated application instances. Subnets in a VPC are associated with a single Availability Zone, so when you select the subnet, you are also choosing the Availability Zone (or physical location) for your database instance.

# High availability with Multi-AZ deployment



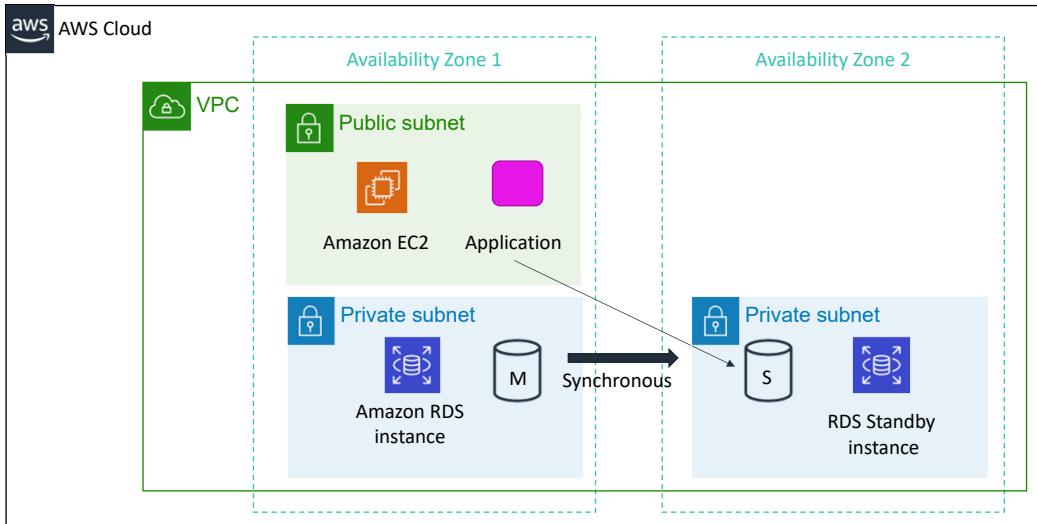
© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

13

One of the most powerful features of Amazon RDS is the ability to configure your database instance for high availability with a Multi-AZ deployment. After a Multi-AZ deployment is configured, Amazon RDS automatically generates a standby copy of the database instance in another Availability Zone within the same VPC. After seeding the database copy, transactions are synchronously replicated to the standby copy. Running a database instance in a Multi-AZ deployment can enhance availability during planned system maintenance, and it can help protect your databases against database instance failure and Availability Zone disruption.

# High availability with Multi-AZ deployment

## 2



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

14

Therefore, if the main database instance fails in a Multi-AZ deployment, Amazon RDS automatically brings the standby database instance online as the new main instance. The synchronous replication minimizes the potential for data loss. Because your applications reference the database by name by using the Amazon RDS Domain Name System (DNS) endpoint, you don't need to change anything in your application code to use the standby copy for failover.

# Amazon RDS read replicas

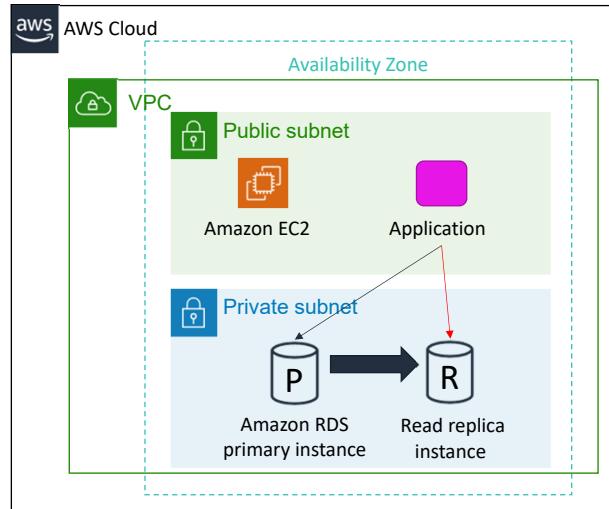


## Features

- Offers asynchronous replication
- Can be promoted to primary if needed

## Functionality

- Use for read-heavy database workloads
- Offload read queries



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

15

Amazon RDS also supports the creation of read replicas for MySQL, MariaDB, PostgreSQL, and Amazon Aurora. Updates that are made to the source database instance are asynchronously copied to the read replica instance. You can reduce the load on your source database instance by routing read queries from your applications to the read replica. Using read replicas, you can also scale out beyond the capacity constraints of a single database instance for read-heavy database workloads. Read replicas can also be promoted to become the primary database instance, but this requires manual action because of asynchronous replication.

Read replicas can be created in a different Region than the primary database. This feature can help satisfy disaster recovery requirements or reduce latency by directing reads to a read replica that is closer to the user.

<b>Web and mobile applications</b>	✓ High throughput ✓ Massive storage scalability ✓ High availability
<b>Ecommerce applications</b>	✓ Low-cost database ✓ Data security ✓ Fully managed solution
<b>Mobile and online games</b>	✓ Rapidly grow capacity ✓ Automatic scaling ✓ Database monitoring

Amazon RDS works well for web and mobile applications that need a database with high throughput, massive storage scalability, and high availability. Because Amazon RDS does not have any licensing constraints, it fits the variable usage pattern of these applications. For small and large ecommerce businesses, Amazon RDS provides a flexible, secure, and low-cost database solution for online sales and retailing. Mobile and online games require a database platform with high throughput and availability. Amazon RDS manages the database infrastructure, so game developers do not need to worry about provisioning, scaling, or monitoring database servers.

# When to Use Amazon RDS



## **Use Amazon RDS when your application requires:**

- Complex transactions or complex queries
- A medium to high query or write rate – Up to 30,000 IOPS (15,000 reads + 15,000 writes)
- No more than a single worker node or shard
- High durability

## **Do not use Amazon RDS when your application requires:**

- Massive read/write rates (for example, 150,000 write/second)
- Sharding due to high data size or throughput demands
- Simple GET or PUT requests and queries that a NoSQL database can handle
- Relational database management system (RDBMS) customization

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

17

## Use Amazon RDS when your application requires:

- Complex transactions or complex queries
- A medium to high query or write rate – up to 30,000 IOPS (15,000 reads + 15,000 writes)
- No more than a single worker node or shard
- High durability

## Do not use Amazon RDS when your application requires:

- Massive read/write rates (for example 150,000 writes per second)
- Sharding due to high data size or throughput demands
- Simple GET or PUT requests and queries that a NoSQL database can handle
- Or, relational database management system (RDBMS) customization

For circumstances when you should not use Amazon RDS, consider either using a NoSQL database solution (such as DynamoDB) or running your relational database engine on Amazon EC2 instances instead of Amazon RDS (which will provide you with more options for customizing your database).

# Amazon RDS: Clock-hour billing and database characteristics



## Clock-hour billing –

- Resources incur charges when running

## Database characteristics –

- Physical capacity of database:
  - Engine
  - Size
  - Memory class

When you begin to estimate the cost of Amazon RDS, you must consider the clock hours of service time, which are resources that incur charges when they are running (for example, from the time you launch a database instance until you terminate the instance).

Database characteristics should also be considered. The physical capacity of the database you choose will affect how much you are charged. Database characteristics vary depending on the database engine, size, and memory class.

### DB purchase type –

- On-Demand Instances
  - Compute capacity by the hour
- Reserved Instances
  - Low, one-time, upfront payment for database instances that are reserved with a 1-year or 3-year term

### Number of DB instances –

- Provision multiple DB instances to handle peak loads

Consider the database purchase type. When you use On-Demand Instances, you pay for compute capacity for each hour that your database instance runs, with no required minimum commitments. With Reserved Instances, you can make a low, one-time, upfront payment for each database instance you want to reserve for a 1-year or 3-year term.

Also, you must consider the number of database instances. With Amazon RDS, you can provision multiple database instances to handle peak loads.

## Provisioned storage –

- No charge
  - Backup storage of up to 100 percent of database storage for an active database
- Charge (*GB/month*)
  - Backup storage for terminated DB instances

## Additional storage –

- Charge (*GB/month*)
  - Backup storage in addition to provisioned storage

Consider provisioned storage. There is no additional charge for backup storage of up to 100 percent of your provisioned database storage for an active database instance. After the database instance is terminated, backup storage is billed per GB, per month.

Also consider the amount of backup storage in addition to the provisioned storage amount, which is billed per GB, per month.

# Amazon RDS: Deployment type and data transfer



## Requests –

- The number of input and output requests that are made to the database

## Deployment type—Storage and I/O charges vary, depending on whether you deploy to –

- Single Availability Zone
- Multiple Availability Zones

## Data transfer –

- No charge for inbound data transfer
- Tiered charges for outbound data transfer

Also consider the number of input and output requests that are made to the database.

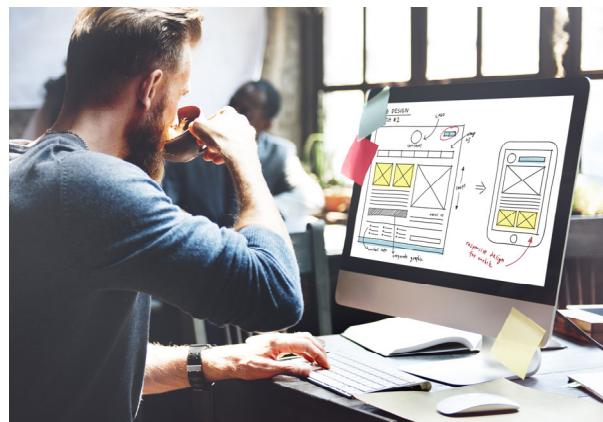
Consider the deployment type. You can deploy your DB instance to a single Availability Zone (which is analogous to a standalone data center) or to multiple Availability Zones (which is analogous to a secondary data center for enhanced availability and durability). Storage and I/O charges vary, depending on the number of Availability Zones that you deploy to.

Finally, consider data transfer. Inbound data transfer is free, and outbound data transfer costs are tiered.

Depending on the needs of your application, it's possible to optimize your costs for Amazon RDS database instances by purchasing Reserved Instances. To purchase Reserved Instances, you make a low, one-time payment for each instance that you want to reserve. As a result, you receive a significant discount on the hourly usage charge for that instance.

## Build Your DB Server and Interact with Your DB Using an App

22



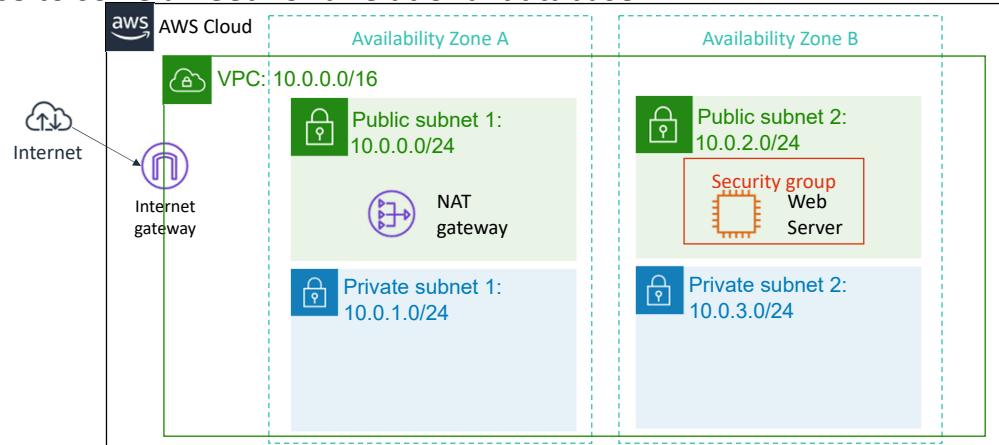
© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

You will now complete Lab 5: Build Your DB Server and Interact with Your DB Using an App.

## Lab 5: Scenario



This lab is designed to show you how to use an AWS managed database instance to solve a need for a relational database.



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

23

This lab is designed to show you how to use an AWS managed database instance to solve a need for a relational database. With Amazon RDS, you can set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, which enables you to focus on your applications and your business. Amazon RDS provides six familiar database engines to choose from: Amazon Aurora, Oracle, Microsoft SQL Server, PostgreSQL, MySQL, and MariaDB.

Amazon RDS Multi-AZ deployments provide enhanced availability and durability for DB instances, which make them a good fit for production database workloads. When you provision a Multi-AZ DB instance, Amazon RDS automatically creates a primary DB instance and synchronously replicates the data to a standby instance in a different Availability Zone.

After completing this lab, you should be able to:

- Launch an Amazon RDS DB instance with high availability.
- Configure the DB instance to permit connections from your web server.
- Open a web application and interact with your database.

## Lab 5: Tasks



Security group

Create a **VPC security group**.



Private subnet

Create a **DB subnet group**.



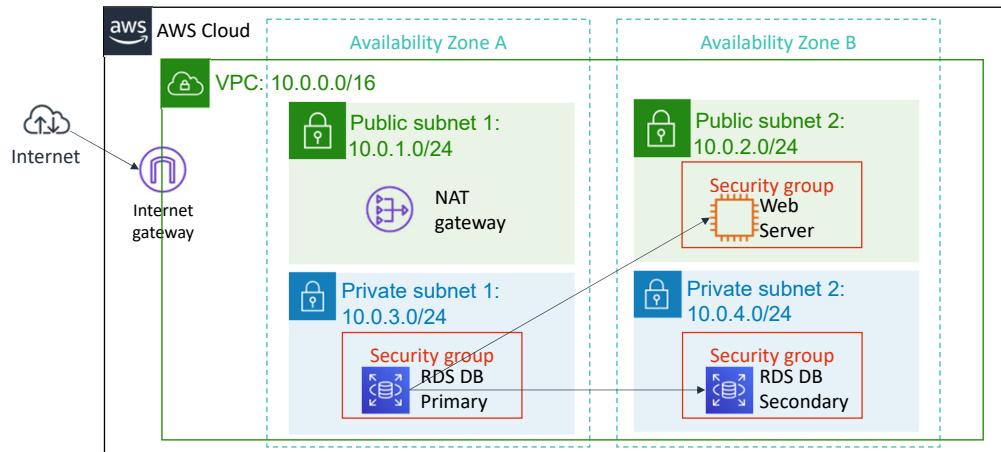
Amazon RDS

Create an **Amazon RDS DB instance** and interact with your database.

Your goal in completing this lab is to:

- Create a VPC security group.
- Create a DB subnet group.
- Create an Amazon RDS DB instance and interact with your database.

# Lab 5: Final product



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

25

In this lab, you:

- Launched an Amazon RDS DB instance with high availability.
- Configured the DB instance to permit connections from your web server.
- Opened a web application and interacted with your database



~ 30 minutes



**Begin Lab 5: Build your DB server and interact with your DB using an application**



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

26

It is now time to start the lab.



## Lab debrief: key takeaways

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

27

In this lab you:

- Created a VPC security group.
- Created a DB subnet group.
- Created an Amazon RDS DB instance
- Interacted with your database

## Recorded demo: Amazon RDS

28



Now, take a moment to watch the [Amazon RDS demo](#). The recording runs a little over 6minutes, 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 Amazon RDS installation running the Amazon Aurora database engine
- A security group to secure the database

The demonstration also shows how to validate that the database is operational.

## Section 1 key takeaways



29

- With Amazon RDS, you can set up, operate, and scale relational databases in the cloud.
- Features –
  - Managed service
  - Accessible via the console, AWS Command Line Interface (AWS CLI), or application programming interface (API) calls
  - Scalable (compute and storage)
  - Automated redundancy and backup are available
  - Supported database engines:
    - Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle, Microsoft SQL Server

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

Amazon RDS is a web service that makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks so you can focus on your applications and your business. Features include that it is a managed service, and that it can be accessed via the console, AWS Command Line Interface (AWS CLI), or application programming interface (API) calls. Amazon RDS is scalable for compute and storage, and automated redundancy and backup is available. Supported database engines include Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle, and Microsoft SQL Server.

Amazon RDS supports demanding database applications. You can choose between two solid state drive (SSD)-backed storage options: one option is optimized for high-performance Online Transactional Processing (OLTP) applications, and the other option works well for cost-effective, general-purpose use.

With Amazon RDS, you can scale your database's compute and storage resources with no downtime. Amazon RDS runs on the same highly reliable infrastructure that is used by other AWS services. It also enables you to run your database instances and Amazon VPC, which is designed to provide you with control and security.

Module 8: Databases

## Section 2: Amazon DynamoDB

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



Welcome to Section 2: Amazon DynamoDB.

## Relational versus non-relational databases



	Relational (SQL)	Non-Relational												
Data Storage	Rows and columns	Key-value, document, graph												
Schemas	Fixed	Dynamic												
Querying	Uses SQL	Focuses on collection of documents												
Scalability	Vertical	Horizontal												
Example	<table border="1"><thead><tr><th>ISBN</th><th>Title</th><th>Author</th><th>Format</th></tr></thead><tbody><tr><td>3111111223439</td><td>Withering Depths</td><td>Jackson, Mateo</td><td>Paperback</td></tr><tr><td>3122222223439</td><td>Wily Willy</td><td>Wang, Xiulan</td><td>Ebook</td></tr></tbody></table>	ISBN	Title	Author	Format	3111111223439	Withering Depths	Jackson, Mateo	Paperback	3122222223439	Wily Willy	Wang, Xiulan	Ebook	{ ISBN: 3111111223439, Title: "Withering Depths", Author: "Jackson, Mateo", Format: "Paperback" }
ISBN	Title	Author	Format											
3111111223439	Withering Depths	Jackson, Mateo	Paperback											
3122222223439	Wily Willy	Wang, Xiulan	Ebook											

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

31

With DynamoDB, this module transitions from relational databases to non-relational databases. Here is a review of the differences between these two types of databases:

- A **relational database** (RDB) works with structured data that is organized by tables, records, and columns. RDBs establish a well-defined relationship between database tables. RDBs use structured query language (SQL), which is a standard user application that provides a programming interface for database interaction. Relational databases might have difficulties scaling out horizontally or working with semistructured data, and might also require many joins for normalized data.
- A **non-relational database** is any database that does not follow the relational model that is provided by traditional relational database management systems (RDBMS). Non-relational databases have grown in popularity because they were designed to overcome the limitations of relational databases for handling the demands of variable structured data. Non-relational databases scale out horizontally, and they can work with unstructured and semistructured data.

Here is a look at what DynamoDB offers.

# What is Amazon DynamoDB?



Fast and flexible NoSQL database service for any scale



**Amazon DynamoDB**

- NoSQL database tables
- Virtually unlimited storage
- Items can have differing attributes
- Low-latency queries
- Scalable read/write throughput

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

32

DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit-millisecond latency at any scale.

Amazon manages all the underlying data infrastructure for this service and redundantly stores data across multiple facilities in a native US Region as part of the fault-tolerant architecture. With DynamoDB, you can create tables and items. You can add items to a table. The system automatically partitions your data and has table storage to meet workload requirements. There is no practical limit on the number of items that you can store in a table. For instance, some customers have production tables that contain billions of items.

One of the benefits of a NoSQL database is that items in the same table can have different attributes. This gives you the flexibility to add attributes as your application evolves. You can store newer format items side by side with older format items in the same table without needing to perform schema migrations.

As your application becomes more popular and as users continue to interact with it, your storage can grow with your application's needs. All the data in DynamoDB is stored on solid state drives (SSDs) and its simple query language enables consistent low-latency query performance. In addition to scaling storage, DynamoDB also enables you to provision the amount of read or write throughput that you need for your table. As the number of application users grows, DynamoDB tables can be scaled to handle the increased numbers of read/write requests with manual provisioning. Alternatively, you can enable automatic

scaling so that DynamoDB monitors the load on the table and automatically increases or decreases the provisioned throughput.

Some additional key features include global tables that enable you to automatically replicate across your choice of AWS Regions, encryption at rest, and item Time-to-Live (TTL).

# Amazon DynamoDB core components



- Tables, items, and attributes are the core DynamoDB components
- DynamoDB supports two different kinds of primary keys: Partition key and partition and sort key

The core DynamoDB components are tables, items, and attributes.

- A table is a collection of data.
- Items are a group of attributes that is uniquely identifiable among all the other items.
- Attributes are a fundamental data element, something that does not need to be broken down any further.

DynamoDB supports two different kinds of primary keys.

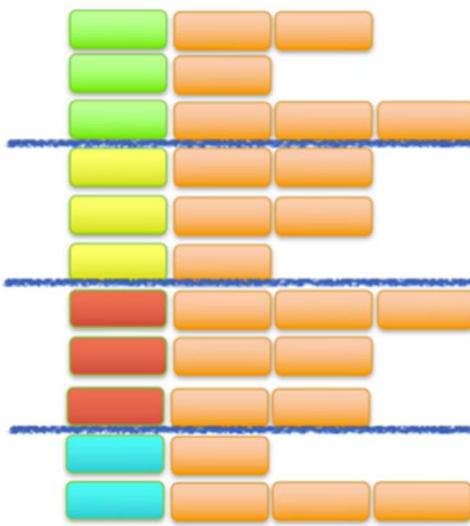
The **partition key** is a simple primary key, which is composed of one attribute called the **sort key**.

The partition key and sort key are also known as the **composite primary key**, which is composed of two attributes.

To learn more about how DynamoDB works, see:

[Table item attributes](#)

# Partitioning



As data grows, table partitioned by key

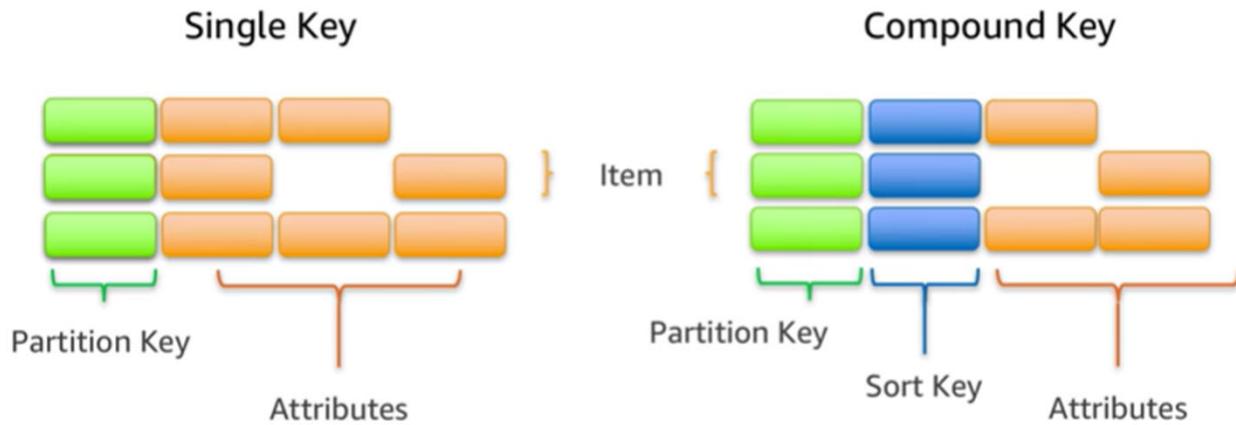
QUERY by Key to find items efficiently  
SCAN to find items by any attribute

As data grows, table data is partitioned and indexed by the primary key.

You can retrieve data from a DynamoDB table in two different ways:

- In the first method, the query operation takes advantage of partitioning to effectively locate items by using the primary key.
- The second method is via a scan, which enables you to locate items in the table by matching conditions on non-key attributes. The second method gives you the flexibility to locate items by other attributes. However, the operation is less efficient because DynamoDB will scan through all the items in the table to find the ones that match your criteria.

# Items in a table must have a key



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

35

To take full advantage of query operations and DynamoDB, it's important to think about the key that you use to uniquely identify items in the DynamoDB table. You can set up a simple primary key that is based on a single attribute of the data values with a uniform distribution, such as the **Globally Unique Identifier (GUID)** or other random identifiers.

For example, if you wanted to model a table with products, you could use some attributes like the product ID. Alternatively, you can specify a compound key, which is composed of a partition key and a secondary key. In this example, if you had a table with books, you might use the combination of author and title to uniquely identify table items. This method could be useful if you expect to frequently look at books by author because you could then use query.

## Section 2 key takeaways



36

### Amazon DynamoDB:

- Runs exclusively on SSDs.
- Supports document and key-value store models.
- Replicates your tables automatically across your choice of AWS Regions.
- Works well for mobile, web, gaming, adtech, and Internet of Things (IoT) applications.
- Is accessible via the console, the AWS CLI, and API calls.
- Provides consistent, single-digit millisecond latency at any scale.
- Has no limits on table size or throughput.

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

DynamoDB runs exclusively on SSDs, and it supports document and key-value store models.

DynamoDB works well for mobile, web, gaming, ad tech, and Internet of Things (IoT) applications.

It's accessible via the console, the AWS CLI, and API calls.

The ability to scale your tables in terms of both storage and provision throughput makes DynamoDB a good fit for structured data from the web, mobile, and IoT applications. For instance, you might have a large number of clients that continuously generate data and make large numbers of requests per second. In this case, the throughput scaling of DynamoDB enables consistent performance for your clients. DynamoDB is also used in latency-sensitive applications. The predictable query performance—even in large tables—makes it useful for cases where variable latency could cause significant impact to the user experience or to business goals, such as adtech or gaming.

The DynamoDB Global Tables feature reduces the work of replicating data between Regions and resolving update conflicts. It replicates your DynamoDB tables automatically across your choice of AWS Regions. Global Tables can help applications stay available and performant for business continuity.

## Recorded demo: Amazon DynamoDB

37



### Set up demo

Amazon DynamoDB

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

Now, take a moment to watch the [DynamoDB demo](#). The recording runs a little over 2 minutes, and it reinforces many of the concepts that were discussed in this section of the module.

The demonstration shows how to create a table running in Amazon DynamoDB by using the AWS Management Console. It also demonstrates how to interact with the table using the AWS Command Line Interface. The demonstration shows how you can query the table, and add data to the table.

.

# Amazon DynamoDB demonstration



## Amazon DynamoDB

Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale. Its flexible data model and reliable performance make it a great fit for mobile, web, gaming, ad-tech, IoT, and many other applications.

[Create table](#)

[Getting started guide](#)



Create tables



Add and query items



Monitor and manage tables

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

38

Review the demonstration: [Amazon DynamoDB console demo](#).

You can access this recorded demonstration in the learning management system.

Module 8: Databases

## Section 3: Amazon Redshift

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



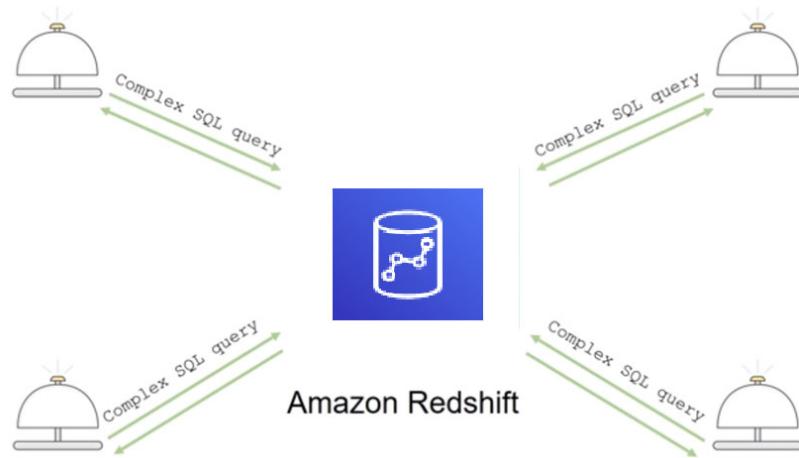
Welcome to Section 3: Amazon Redshift.



Amazon Redshift

Amazon Redshift is a fast, fully managed data warehouse that makes it simple and cost-effective to analyze all your data by using standard SQL and your existing business intelligence (BI) tools. Here is a look at Amazon Redshift and how you can use it for analytic applications.

# Introduction to Amazon Redshift



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

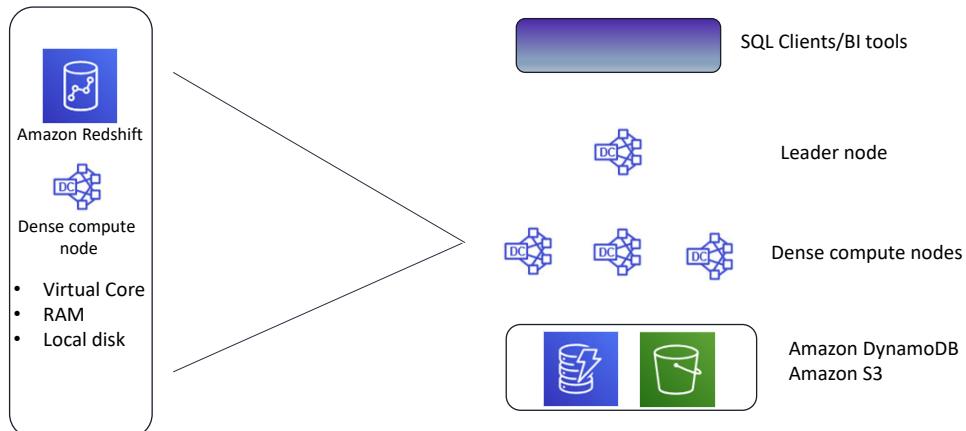
41

Analytics is important for businesses today, but building a data warehouse is complex and expensive. Data warehouses can take months and significant financial resources to set up.

Amazon Redshift is a fast and powerful, fully managed data warehouse that is simple and cost-effective to set up, use, and scale. It enables you to run complex analytic queries against petabytes of structured data by using sophisticated query optimization, columnar storage on high-performance local disks, and massively parallel data processing. Most results come back in seconds.

You will next review a slightly more detailed exploration of key Amazon Redshift features and some common use cases.

# Parallel processing architecture



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

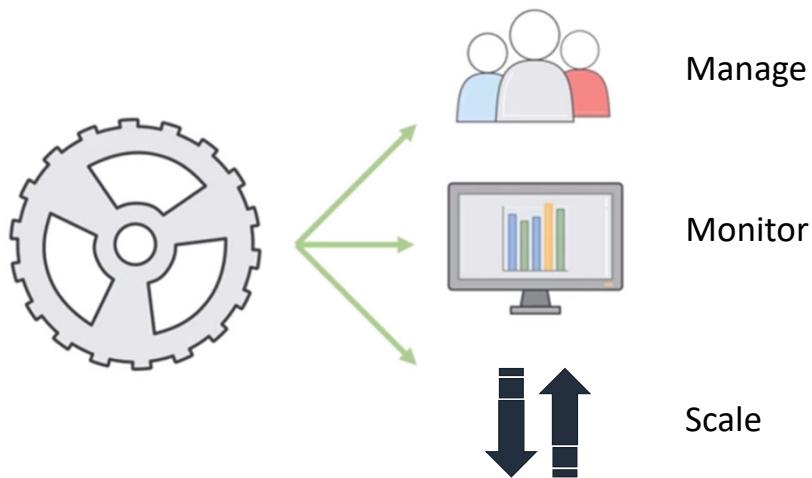
42

The leader node manages communications with client programs and all communication with compute nodes. It parses and develops plans to carry out database operations—specifically, the series of steps that are needed to obtain results for complex queries. The leader node compiles code for individual elements of the plan and assigns the code to individual compute nodes. The compute nodes run the compiled code and send intermediate results back to the leader node for final aggregation.

Like other AWS services, you only pay for what you use. You can get started for as little as 25 cents per hour and, at scale, Amazon Redshift can deliver storage and processing for approximately \$1,000 dollars per terabyte per year (with 3-Year Partial Upfront Reserved Instance pricing).

The Amazon Redshift Spectrum feature enables you to run queries against exabytes of data directly in Amazon S3.

## Automation and scaling



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

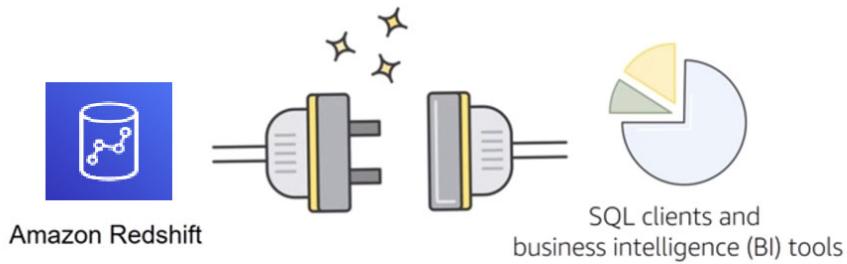
43

It is straightforward to automate most of the common administrative tasks to manage, monitor, and scale your Amazon Redshift cluster—which enables you to focus on your data and your business.

Scalability is intrinsic in Amazon Redshift. Your cluster can be scaled up and down as your needs change with a few clicks in the console.

Security is the highest priority for AWS. With Amazon Redshift, security is built in, and it is designed to provide strong encryption of your data both at rest and in transit.

# Compatibility



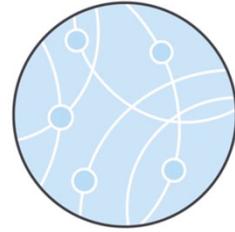
Finally, Amazon Redshift is compatible with the tools that you already know and use. Amazon Redshift supports standard SQL. It also provides high-performance Java Database Connectivity (JDBC) and Open Database Connectivity (ODBC) connectors, which enable you to use the SQL clients and BI tools of your choice.

Next, you will review some common Amazon Redshift use cases.

# Amazon Redshift use cases



- Enterprise data warehouse (EDW)
  - Migrate at a pace that customers are comfortable with
  - Experiment without large upfront cost or commitment
  - Respond faster to business needs
- Big data
  - Low price point for small customers
  - Managed service for ease of deployment and maintenance
  - Focus more on data and less on database management



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

45

This slide discusses some Amazon Redshift use cases.

Many customers migrate their traditional enterprise data warehouses to Amazon Redshift with the primary goal of agility. Customers can start at whatever scale they want and experiment with their data without needing to rely on complicated processes with their IT departments to procure and prepare their software.

Big data customers have one thing in common: massive amounts of data that stretch their existing systems to a breaking point. Smaller customers might not have the resources to procure the hardware and expertise that is needed to run these systems. With Amazon Redshift, smaller customers can quickly set up and use a data warehouse at a comparatively low price point.

As a managed service, Amazon Redshift handles many of the deployment and ongoing maintenance tasks that often require a database administrator. This enables customers to focus on querying and analyzing their data.

## Amazon Redshift use cases 2



- Software as a service (SaaS)
  - Scale the data warehouse capacity as demand grows
  - Add analytic functionality to applications
  - Reduce hardware and software costs



Software as a service (SaaS) customers can take advantage of the scalable, easy-to-manage features that Amazon Redshift provides. Some customers use the Amazon Redshift to provide analytic capabilities to their applications. Some users deploy a cluster per customer, and use tagging to simplify and manage their service level agreements (SLAs) and billing. Amazon Redshift can help you reduce hardware and software costs.

## Section 3 key takeaways



47



### Amazon Redshift features:

- Fast, fully managed data warehouse service
- Easily scale with no downtime
- Columnar storage and parallel processing architectures
- Automatically and continuously monitors cluster
- Encryption is built in

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

In summary, Amazon Redshift is a fast, fully managed data warehouse service. As a business grows, you can easily scale with no downtime by adding more nodes. Amazon Redshift automatically adds the nodes to your cluster and redistributes the data for maximum performance.

Amazon Redshift is designed to consistently deliver high performance. Amazon Redshift uses columnar storage and a massively parallel processing architecture. These features parallelize and distribute data and queries across multiple nodes. Amazon Redshift also automatically monitors your cluster and backs up your data so that you can easily restore if needed. Encryption is built in—you only need to enable it.

To learn more about Amazon Redshift, see:  
<https://aws.amazon.com/redshift/>.

Module 8: Databases

## Section 4: Amazon Aurora

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



Introducing Section 4: Amazon Aurora.

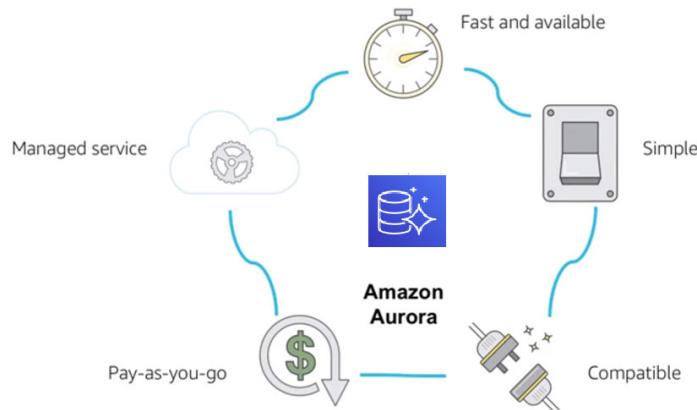


Amazon Aurora

- Enterprise-class relational database
- Compatible with MySQL or PostgreSQL
- Automate time-consuming tasks (such as provisioning, patching, backup, recovery, failure detection, and repair).

Amazon Aurora is a MySQL- and PostgreSQL-compatible relational database that is built for the cloud. It combines the performance and availability of high-end commercial databases with the simplicity and cost-effectiveness of open-source databases. Using Amazon Aurora can reduce your database costs while improving the reliability and availability of the database. As a fully managed service, Aurora is designed to automate time-consuming tasks like provisioning, patching, backup, recovery, failure detection, and repair.

# Amazon Aurora service benefits



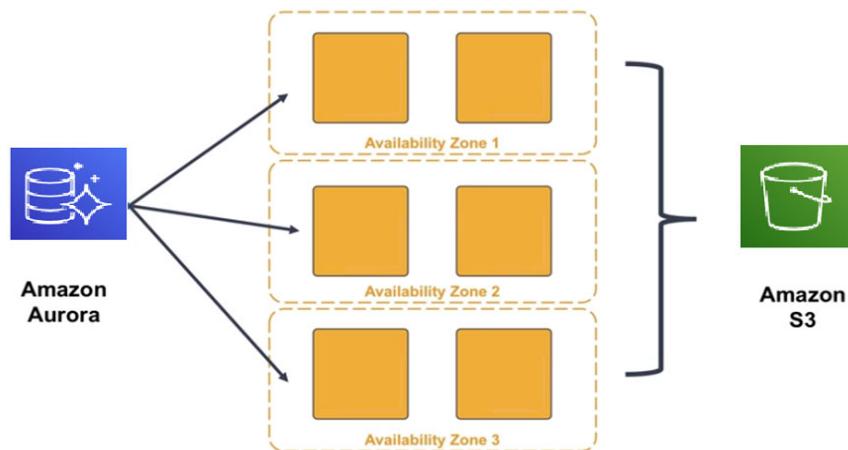
© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

50

This slide covers some of the benefits of Amazon Aurora. It is highly available and it offers a fast, distributed storage subsystem. Amazon Aurora is straightforward to set up and uses SQL queries. It is designed to have drop-in compatibility with MySQL and PostgreSQL database engines so that you can use most of your existing database tools with little or no change.

Amazon Aurora is a pay-as-you-go service, which means that you only pay for the services and features that you use. It's a managed service that integrates with features such as AWS Database Migration Service (AWS DMS) and the AWS Schema Conversion Tool. These features are designed to help you move your dataset into Amazon Aurora.

## High availability



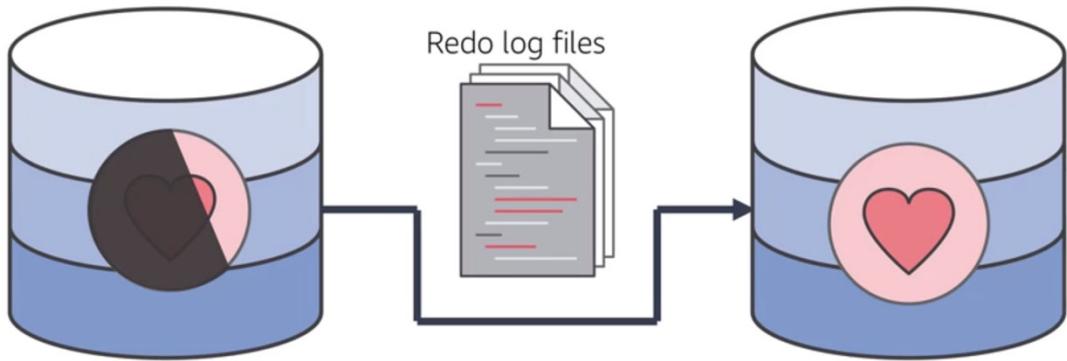
© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

51

Why might you use Amazon Aurora over other options, like SQL with Amazon RDS? Most of that decision involves the high availability and resilient design that Amazon Aurora offers.

Amazon Aurora is designed to be highly available: it stores multiple copies of your data across multiple Availability Zones with continuous backups to Amazon S3. Amazon Aurora can use up to 15 read replicas can be used to reduce the possibility of losing your data. Additionally, Amazon Aurora is designed for instant crash recovery if your primary database becomes unhealthy.

## Resilient design



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

52

After a database crash, Amazon Aurora does not need to replay the redo log from the last database checkpoint. Instead, it performs this on every read operation. This reduces the restart time after a database crash to less than 60 seconds in most cases.

With Amazon Aurora, the buffer cache is moved out of the database process, which makes it available immediately at restart. This reduces the need for you to throttle access until the cache is repopulated to avoid brownouts.

## Section 4 key takeaways



53

### Amazon Aurora features:

- High performance and scalability
- High availability and durability
- Multiple levels of security
- Compatible with MySQL and PostgreSQL
- Fully managed

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

In summary, Amazon Aurora is a highly available, performant, and cost-effective managed relational database.

Aurora offers a distributed, high-performance storage subsystem. Using Amazon Aurora can reduce your database costs while improving the reliability of the database.

Aurora is also designed to be highly available. It has fault-tolerant and self-healing storage built for the cloud. Aurora replicates multiple copies of your data across multiple Availability Zones, and it continuously backs up your data to Amazon S3.

Multiple levels of security are available, including network isolation by using Amazon VPC; encryption at rest by using keys that you create and control through AWS Key Management Service (AWS KMS); and encryption of data in transit by using Secure Sockets Layer (SSL).

The Amazon Aurora database engine is compatible with existing MySQL and PostgreSQL open source databases, and adds compatibility for new releases regularly.

Finally, Amazon Aurora is fully managed by Amazon RDS. Aurora automates database management tasks, such as hardware provisioning, software patching, setup, configuration, or backups.

To learn more about Amazon Aurora, see:

[Aurora](#)

# The right tool for the right job



## What are my requirements?

Enterprise-class relational database

Amazon RDS

Fast and flexible NoSQL database service for any scale

Amazon DynamoDB

Operating system access or application features that are not supported by AWS database services

Databases on Amazon EC2

Specific case-driven requirements (machine learning, data warehouse, graphs)

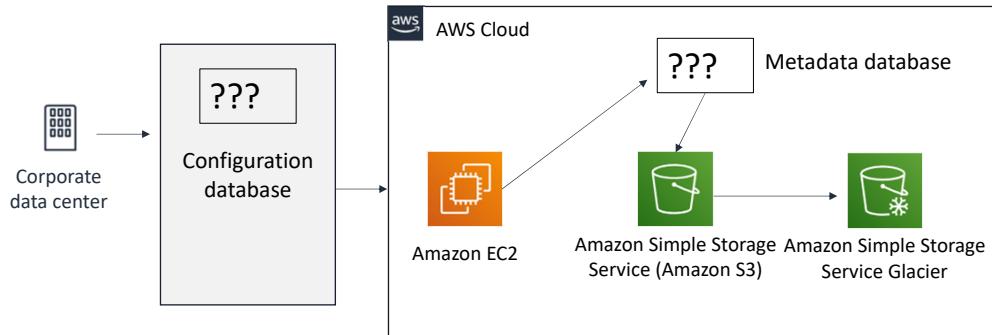
AWS purpose-built database services

As you saw in this module, the cloud continues to drive down the cost of storage and compute. A new generation of applications has emerged, which created a new set of requirements for databases. These applications need databases to store terabytes to petabytes of new types of data, provide access to the data with millisecond latency, process millions of requests per second, and scale to support millions of users anywhere in the world. To support these requirements, you need both relational and non-relational databases that are purpose-built to handle the specific needs of your applications. AWS offers a broad range of databases that are built for your specific application use cases.

# Database case study activity 1



Case 1: A data protection and management company that provides services to enterprises. They must provide database services for over 55 petabytes of data. They have two types of data that require a database solution. First, they need a relational database store for configuration data. Second, they need a store for unstructured metadata to support a de-duplication service. After the data is de-duplicated, it is stored in Amazon S3 for quick retrieval, and eventually moved to Amazon S3 Glacier for long-term storage. The following diagram illustrates their architecture.



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

55

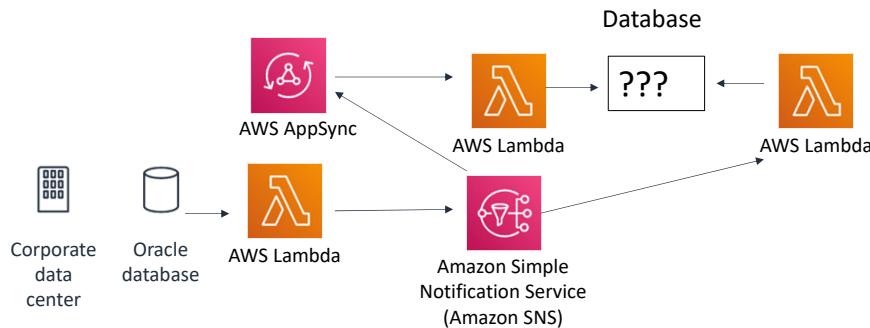
In this activity, you will review one of three business scenarios that were taken from actual AWS customers. Break into groups of four or five.

Review the assigned case study. Create a presentation that describes the best database solution for the organization that is described in your group's case. Your presentation should include the key factors that you considered when you selected the database technology, in addition to any factors that could change your recommendation.

## Database case study activity 2



Case 2: A commercial shipping company that uses an on-premises legacy data management system. They must migrate to a serverless ecosystem while they continue to use their existing database system, which is based on Oracle. They are also in the process of decomposing their highly structured relational data into semistructured data. The following diagram illustrates their architecture.

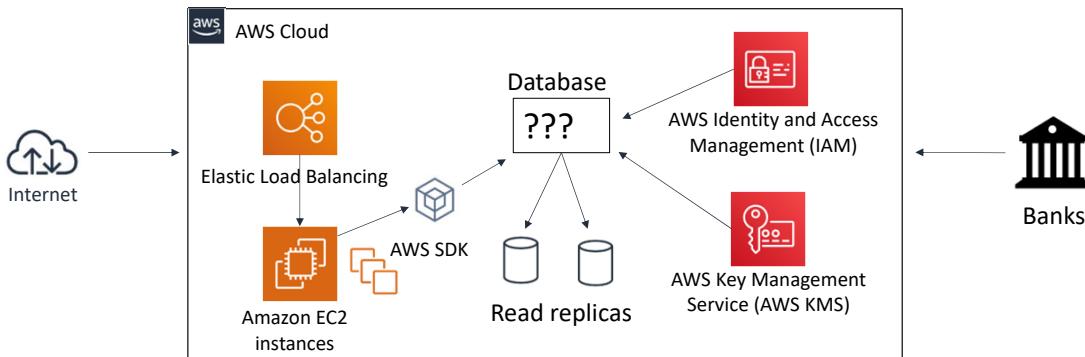


Review the assigned case study. Create a presentation that describes the best database solution for the organization that is described in your group's case. Your presentation should include the key factors that you considered when you selected the database technology, in addition to any factors that could change your recommendation.

## Database case study activity 3



Case 3: An online payment processing company that processes over 1 million transactions per day. They must provide services to ecommerce customers who offer flash sales (sales that offer greatly reduced prices for a limited time), where demand can increase by 30 times in a short time period. They use IAM and AWS KMS to authenticate transactions with financial institutions. They need high throughput for these peak loads. The following diagram illustrates their architecture.



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

57

Review the assigned case study. Create a presentation that describes the best database solution for the organization that is described in your group's case. Your presentation should include the key factors that you considered when you selected the database technology, in addition to any factors that could change your recommendation.

Module 8: Databases

## Module wrap-up

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



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:

- Explain Amazon Relational Database Service (Amazon RDS)
- Identify the functionality in Amazon RDS
- Explain Amazon DynamoDB
- Identify the functionality in Amazon DynamoDB
- Explain Amazon Redshift
- Explain Amazon Aurora
- Perform tasks in an RDS database, such as launching, configuring, and interacting

In summary, in this module, you learn how to:

- Explain Amazon Relational Database Service (Amazon RDS)
- Identify the functionality in Amazon RDS
- Explain Amazon DynamoDB
- Identify the functionality in Amazon DynamoDB
- Explain Amazon Redshift
- Explain Amazon Aurora
- Perform tasks in an RDS database, such as launching, configuring, and interacting

## Complete the knowledge check



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

60

The instructor might choose to lead a conversation about the key takeaways from the lab after you complete it.

## Sample exam question



Which of the following is a fully-managed NoSQL database service?

- A. Amazon Relational Database Service (Amazon RDS)
- B. Amazon DynamoDB
- C. Amazon Aurora
- D. Amazon Redshift

Look at the answer choices, and rule them out based on the keywords that were previously highlighted.

## Additional resources



- [AWS Database page](#)
- [Amazon RDS page](#)
- [Overview of Amazon database services](#)
- [Getting started with AWS databases](#)

If you want to learn more about the topics covered in this module, you might find the following additional resources helpful:

- [AWS Database page](#)
- [Amazon RDS page](#)
- [Overview of Amazon database services](#)
- [Getting started with AWS databases](#)

# Thank you

© 2019 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections or feedback on the course, please email us at: [aws-course-feedback@amazon.com](mailto:aws-course-feedback@amazon.com). For all other questions, contact us at: <https://aws.amazon.com/contact-us/aws-training/>. All trademarks are the property of their owners.



Thanks for participating!

AWS Academy Cloud Foundations

# Module 9: Cloud Architecture

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



Module 9: Cloud Architecture

# Module overview



## Topics

- AWS Well-Architected Framework
- Reliability and high availability
- AWS Trusted Advisor

## Activities

- AWS Well-Architected Framework Design Principles
- Interpret AWS Trusted Advisor Recommendations



## Knowledge check

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

2

This module will address the following topics:

- AWS Well-Architected Framework
- Reliability and high availability
- AWS Trusted Advisor

The module also includes two activities. In one activity, you will be challenged to review an architecture and evaluate it against the AWS Well-Architected Framework design principles. In the second activity, you will gain experience interpreting AWS Trusted Advisor recommendations.

Finally, you will be asked to complete a knowledge check that will test your understanding of key concepts covered in this module.

## Module objectives



After completing this module, you should be able to:

- Describe the AWS Well-Architected Framework, including the six pillars
- Identify the design principles of the AWS Well-Architected Framework
- Explain the importance of reliability and high availability
- Identify how AWS Trusted Advisor helps customers
- Interpret AWS Trusted Advisor recommendations

After completing this module, you should be able to:

- Describe the AWS Well-Architected Framework, including the six pillars
- Identify the design principles of the AWS Well-Architected Framework
- Explain the importance of reliability and high availability
- Identify how AWS Trusted Advisor helps customers
- Interpret AWS Trusted Advisor recommendations

Module 9: Cloud Architecture

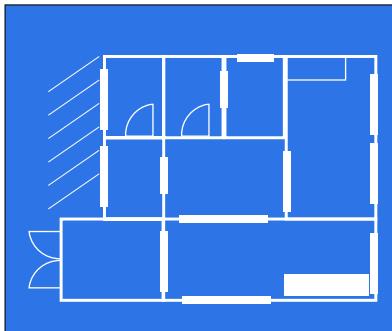
## Section 1: AWS Well-Architected Framework

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



## Section 1: AWS Well-Architected Framework

# Architecture: designing and building

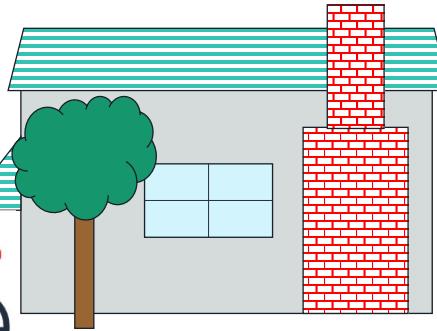


Structure design

Customer  
(Decision maker)



Architect



Completed structure

Building crew  
(Delivery team)

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

5

Architecture is the art and science of designing and building large structures. Large systems require architects to manage their size and complexity.

Cloud architects:

- Engage with decision makers to identify the business goal and the capabilities that need improvement.
- Ensure alignment between technology deliverables of a solution and the business goals.
- Work with delivery teams that are implementing the solution to ensure that the technology features are appropriate.

Having well-architected systems greatly increases the likelihood of business success.

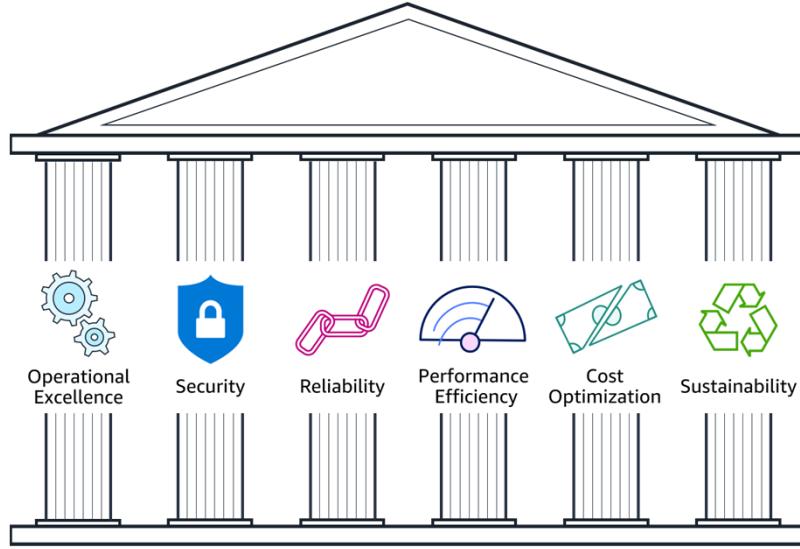
# What is the AWS Well-Architected Framework?



- A guide for designing infrastructures that are:
  - ✓ Secure
  - ✓ High-performing
  - ✓ Resilient
  - ✓ Efficient
- A consistent approach to evaluating and implementing cloud architectures
- A way to provide best practices that were developed through lessons learned by reviewing customer architectures

The AWS Well-Architected Framework is a guide that is designed to help you build the most secure, high-performing, resilient, and efficient infrastructure possible for your cloud applications and workloads. It provides a set of foundational questions and best practices that can help you evaluate and implement your cloud architectures. AWS developed the Well-Architected Framework after reviewing thousands of customer architectures on AWS.

# Pillars of the AWS Well-Architected Framework



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

7

The AWS Well-Architected Framework is organized into six pillars: operational excellence, security, reliability, performance efficiency, cost optimization, and sustainability. The first five pillars have been part of the framework since the framework's introduction in 2015. The sustainability pillar was added as the sixth pillar in 2021 to help organizations learn how to minimize the environmental impacts of running cloud workloads.

The remainder of this module focuses on the first five pillars (operational excellence, security, reliability, performance efficiency, cost optimization) and leads you through a review of an example architecture against each pillar's design principles.

For more about the sustainability pillar, refer to the sustainability pillar section within the Well-Architected Framework documentation:

<https://docs.aws.amazon.com/wellarchitected/latest/sustainability-pillar/sustainability-pillar.html>.

# Pillar organization



<b>Best practice area</b>	<b>Identity and Access Management</b>
<b>Question text</b>	<b>SEC 1: How do you manage credentials and authentication?</b>
<b>Question context</b>	Credential and authentication mechanisms include passwords, tokens, and keys that grant access directly or indirectly in your workload. Protect credentials with appropriate mechanisms to help reduce the risk of accidental or malicious use.
<b>Best practices</b>	<p>Best practices:</p> <ul style="list-style-type: none"><li>• Define requirements for identity and access management</li><li>• Secure AWS account root user</li><li>• Enforce use of multi-factor authentication</li><li>• Automate enforcement of access controls</li><li>• Integrate with centralized federation provider</li><li>• Enforce password requirements</li><li>• Rotate credentials regularly</li><li>• Audit credentials periodically</li></ul>

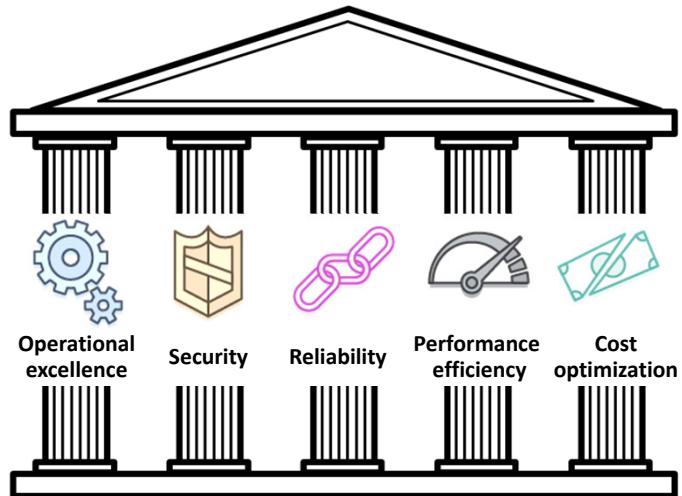
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

8

Each pillar includes a set of design principles and best practice areas. Each best practice area aligns to questions a reviewer should ask when designing an architecture. The questions for each pillar are part of the Well-Architected Framework Appendix.

# Introduction to the AWS Well-Architected Framework Design Principles Activity

9



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

As you go through the rest of this section, you will be prompted to review the architecture of a fictitious company using the AWS Well-Architected Framework design principles for each of the following five pillars: operational excellence, security, reliability, performance efficiency, and cost optimization.

## AnyCompany background



- AnyCompany Corporation: “*Cityscapes you can stand over*”
- Founded in 2008 by John Doe
- Sells 3D-printed cityscapes
- About to apply for investment
- Has asked **you** to perform a review of their platform as part of their due diligence
- Cloud native

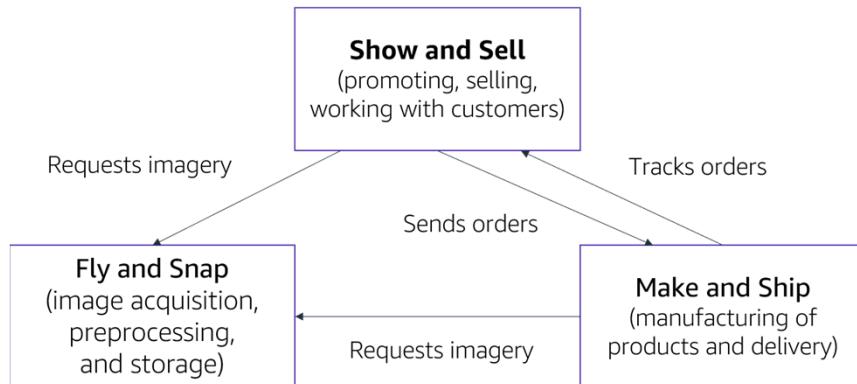
Here's the background of the company whose architecture you will be reviewing:

AnyCompany Corporation was founded in 2008 by John Doe. It sells high-quality three-dimensional (3D) printed cityscapes of neighborhoods that enable you to see individual buildings and trees. The cityscapes are printed in color, with brickwork, roofs, gardens, and even cars in their correct coloration.

The company is about to apply for private investment to fund their growth until their initial public offering (IPO). John and the board have asked you to perform an independent review of their technology platform to make sure that it will pass due diligence.

John was interested in using cloud computing from the start. In 2008, he created an account with AWS and spun up his first Amazon Elastic Compute Cloud (Amazon EC2) instance. Over the years, the architecture of the AnyCompany platform has evolved. John now has a team of five technologists who write and operate all the technology in the organization. John still writes core code for extracting structure from motion, but he has given the AWS account root user credentials to the rest of his team to manage.

## AnyCompany background (continued)

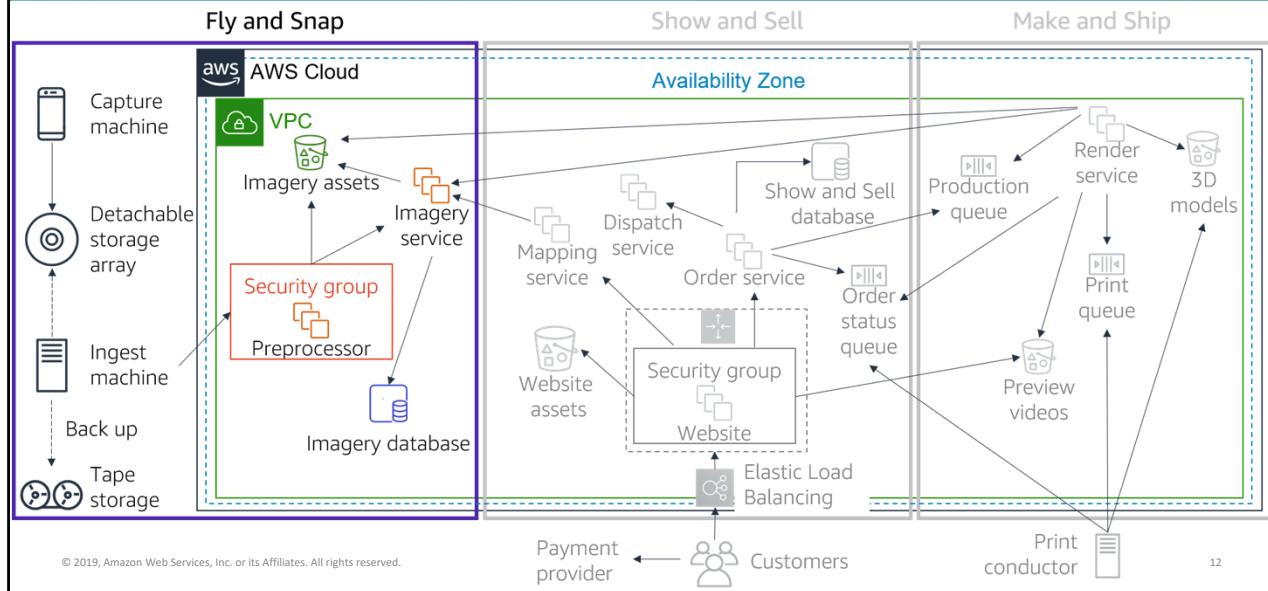


AnyCompany Corporation has three main departments:

- Fly and Snap – image acquisition, preprocessing, and storage
- Show and Sell – promoting, selling, and working with customers
- Make and Ship – manufacturing of products and delivery

The high-level design for the AnyCompany platform looks like the organizational structure of the company.

# AnyCompany architecture: Fly and Snap



## Fly and Snap

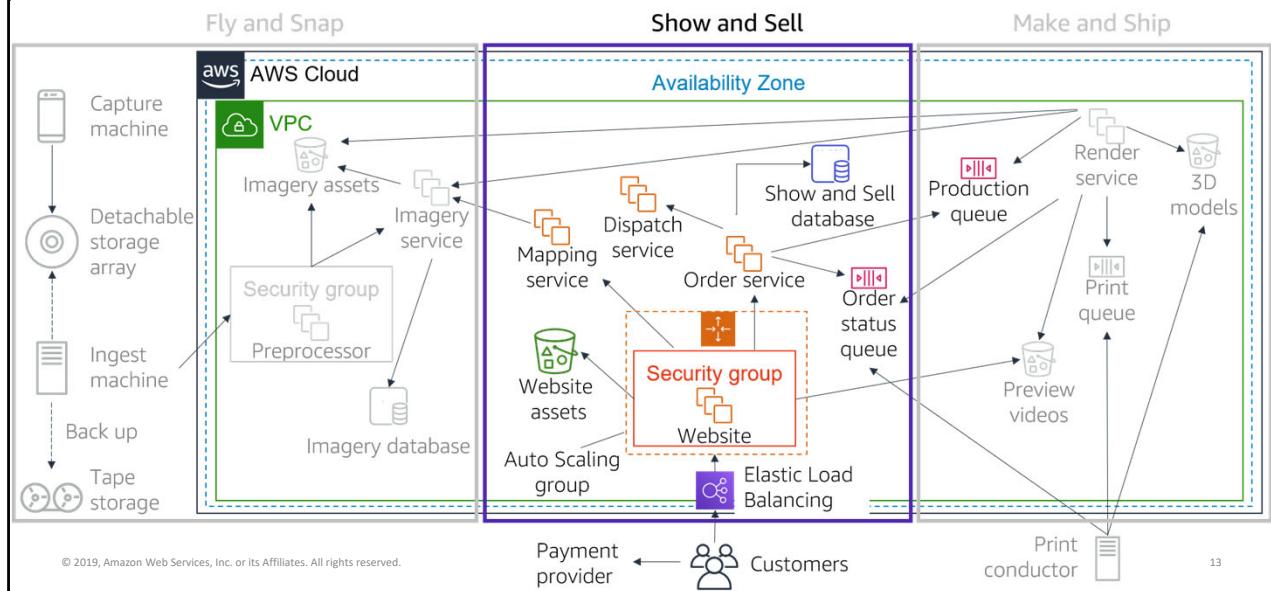
Multiple devices (currently, camera and video cameras) are mounted on lightweight aircraft that capture imagery of major cities, including famous locations, on a scheduled basis. Each device generates imagery assets that are time-stamped with a clock that is synchronized with the aircraft's clock. The imagery assets are streamed to the onboard **Capture machine** that has an external **storage array**. The Capture machine is also connected to the aircraft's flight system and continuously captures navigation data—such as global positioning system (GPS) data, compass readings, and elevation.

When it returns to base, the storage array is disconnected and taken into an ingest bay. Here, the storage array is connected to an **Ingest machine**. The Ingest machine creates a compressed archive of the storage array and uses file transfer protocol (FTP) to send it to an EC2 instance **Preprocessor machine**. After the storage array has been processed, the archive is written to **tape** (for backup). The storage array is then cleared and ready for the next flight. Tapes are held offsite by a third-party backup provider.

The Preprocessor machine periodically processes new datasets that have been uploaded to it. It extracts all the imagery assets and stores them in an **Amazon Simple Storage Service (Amazon S3)** bucket. It notifies the Imagery service about the files and provides it with the flight information. The **Imagery service** uses the flight information to compute a 3D orientation and location for every moment of the flight, which it correlates to the imagery file timestamps. This information is stored in a **relational database management system**.

**(RDBMS)** that is based in Amazon EC2, with links to the imagery assets in Amazon S3.

# AnyCompany architecture: Show and Sell



## Show and Sell

When customers visit the AnyCompany **website**, they can see images and videos of the physical product. These images are in a variety of formats (for example, a large-scale, walk-around map). The **website** uses **Elastic Load Balancing** with Hypertext Transfer Protocol Secure (HTTPS), and an **Auto Scaling group** of EC2 instances that run a content management system. Static website assets are stored in an **S3 bucket**.

Customers can select a location on a map and see a video preview of their cityscape. Customers can also choose the physical size of the map, choose the color scheme (available in white, monochrome, or full color), and have the option to place light-emitting diode (LED) holes in the map to build illuminated maps. The **Mapping service** correlates the map location input from the website with the **Imagery service** to confirm if imagery is available for that location.

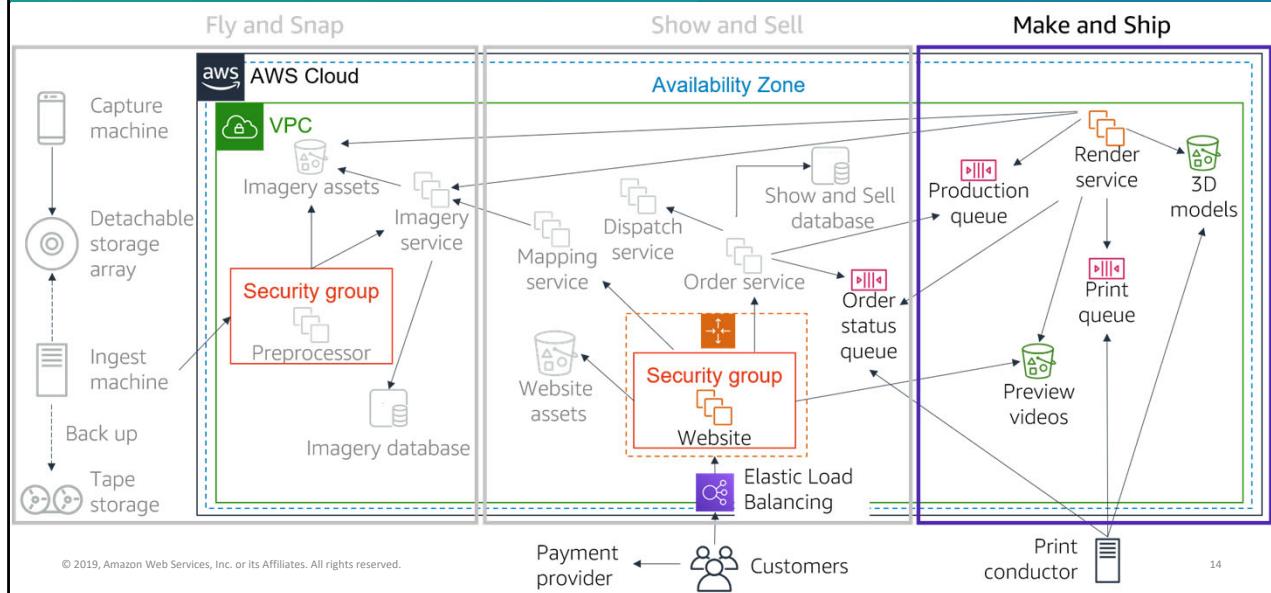
If the customers are happy with the preview, they can order their cityscape. Customers pay by credit card. Credit card orders are processed by a certified third-party payment card industry (PCI)-compliant provider. AnyCompany does not process or store any credit card information.

After the **website** receives payment confirmation, it instructs the **Order service** to push the order to production. Orders (including customer details) are recorded in the **Show and Sell**

**database**, which is an RDBMS that is based in Amazon EC2.

To initiate a video preview or full print of an order, the **Orders service** places a message on the **Production queue**, which allows the **Render service** to indicate when a preview video is available. The **Order service** also reads from the **Order status queue** and records status changes in the **Show and Sell database**. Customers can track their order through manufacturing and see when it has been dispatched, which is handled by a third party through the broker Dispatch service.

# AnyCompany architecture: Make and Ship



## Make and Ship

AnyCompany has proprietary technology that enables it to generate 3D models from a combination of photographs and video (extracting structure from motion).

The **Render service** is a fleet of g2.2xlarge instances. The **Render service** takes orders from the **Production queue** and generates the 3D models that are stored in an **S3 bucket**. The **Render service** also uses the 3D models to create flyby videos so that customers can preview their orders on the AnyCompany **website**. These videos are stored in a separate **S3 bucket**. Once a year, the team deletes old previews. However, models are kept in case they are needed for future projects.

After a customer places an order, a message is placed in the **Print queue** with a link to the 3D model. At each stage of the Make and Ship process, order status updates are posted to the **Order status queue**. This queue is consumed by the AnyCompany **website**, which shows the order history.

The Make and Ship team has four 3D printers that print high-resolution and detailed color-control models. An on-premises **Print conductor** machine takes orders from the **Print queue** and sends them to the next available printer. The **Print conductor** sends order updates to the **Order status queue**. The **Print conductor** sends a final update when the order has been completed, passed quality assurance, and is ready for dispatch.

## Activity overview



- Break into small groups.
- You will learn about each of the pillars. At the end of each pillar, there is a set of questions from the AWS Well-Architected Framework for you to work through with your group. Use these Framework questions to guide your review of the AnyCompany architecture.
- For each Well-Architected Framework question, answer the following questions about the AnyCompany architecture:
  - What is the CURRENT STATE (what is AnyCompany doing now)?
  - What is the FUTURE STATE (what do you think AnyCompany should be doing?)
- Agree on the top improvement that AnyCompany should make to its architecture for each set of Well-Architected Framework questions.
- Hint: There are no right or wrong answers.

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

15

For this activity, you will break into small groups. As you learn about each pillar, your group will work through a set of questions from the AWS Well-Architected Framework. You will use these Well-Architected Framework questions to guide your review of the AnyCompany architecture.

For each Well-Architected Framework question, your group will answer the following questions about the AnyCompany architecture:

- What is the CURRENT STATE (what is AnyCompany doing now)?
- What is the FUTURE STATE (what do you think AnyCompany should be doing)?

Your team must then agree on the top improvement that AnyCompany should make based on the answers to these three questions.

Note that there are no right or wrong answers. The AWS Well-Architected Framework questions are there to prompt discussion.

For prescriptive guidance on implementation, see the details for each pillar on the Well-Architected Framework website:

<https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html>. All of the questions for each pillar are part of the Well-Architected Framework Appendix.



## Operational Excellence pillar

Operational Excellence pillar

# Operational Excellence pillar



## Operational Excellence pillar



Deliver business value

- **Focus**

- Run and monitor systems to deliver business value, and to continually improve supporting processes and procedures.

- **Key topics**

- Automating changes
- Responding to events
- Defining standards to manage daily operations

The *Operational Excellence pillar* focuses on the ability to run and monitor systems to deliver business value, and to continually improve supporting processes and procedures. Key topics include: automating changes, responding to events, and defining standards to manage daily operations.

# Operational excellence design principles



## Operational Excellence pillar



Deliver business value

- Perform operations as code
- Make frequent, small, reversible changes
- Refine operations procedures frequently
- Anticipate failure
- Learn from all operational events and failures

There are five design principles for operational excellence in the cloud:

- *Perform operations as code* – Define your entire workload (that is, applications and infrastructure) as code and update it with code. Implement operations procedures as code and configure them to automatically trigger in response to events. By performing operations as code, you limit human error and enable consistent responses to events.
- *Make frequent, small, reversible changes* – Design workloads to enable components to be updated regularly. Make changes in small increments that can be reversed if they fail (without affecting customers when possible).
- *Refine operations procedures frequently* – Look for opportunities to improve operations procedures. Evolve your procedures appropriately as your workloads evolve. Set up regular game days to review all procedures, validate their effectiveness, and ensure that teams are familiar with them.
- *Anticipate failure* – Identify potential sources of failure so that they can be removed or mitigated. Test failure scenarios and validate your understanding of their impact. Test your response procedures to ensure that they are effective and that teams know how to run them. Set up regular game days to test workloads and team responses to simulated events.
- *Learn from all operational failures* – Drive improvement through lessons learned from all operational events and failures. Share what is learned across teams and through the entire organization.

# Operational excellence questions



## Organization

- How do you determine what your priorities are?
- How do you structure your organization to support your business outcomes?
- How does your organizational culture support your business outcomes?

## Prepare

- How do you design your workload so that you can understand its state?
- How do you reduce defects, ease remediation, and improve flow into production?
- How do you mitigate deployment risks?
- How do you know that you are ready to support a workload?

## Operate

- How do you understand the health of your workload?
- How do you understand the health of your operations?
- How do you manage workload and operations events?

## Evolve

- How do you evolve operations?

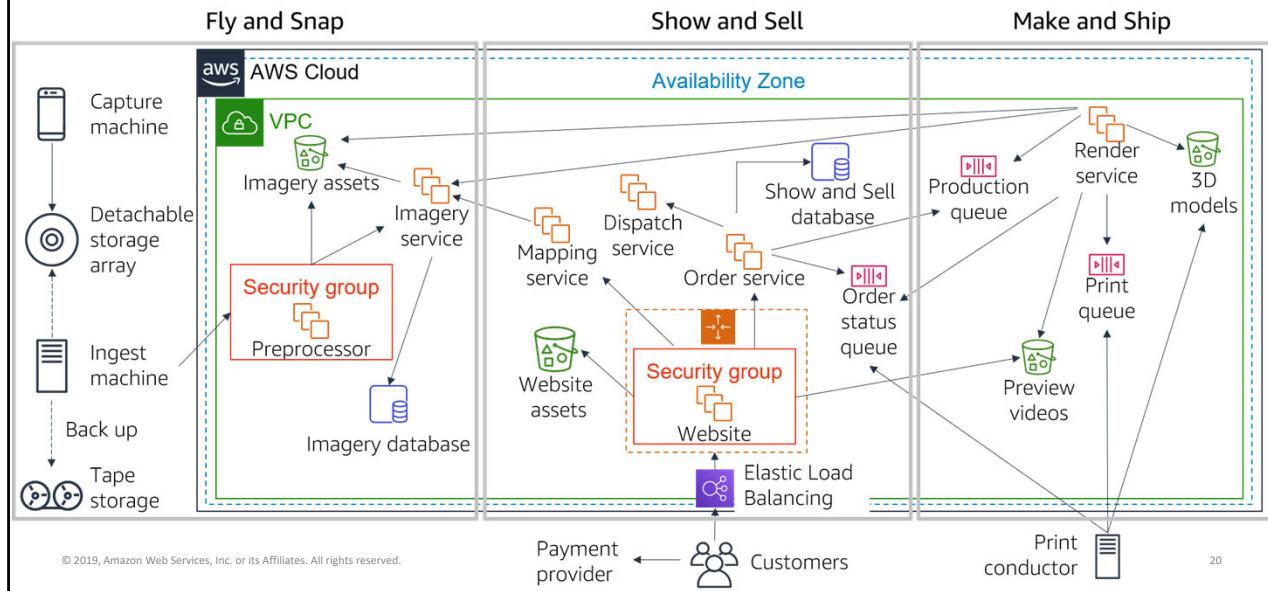
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

19

The foundational questions for operational excellence fall under three best practice areas: organization, prepare, operate, and evolve.

Operations teams must understand business and customer needs so they can effectively and efficiently support business outcomes. Operations teams create and use procedures to respond to operational events and validate the effectiveness of procedures to support business needs. Operations teams collect metrics that are used to measure the achievement of desired business outcomes. As business context, business priorities, and customer needs, change over time, it's important to design operations that evolve in response to change and to incorporate lessons learned through their performance.

# Activity breakout



Here is the entire AnyCompany architecture for you to consult as you complete the activity. Refer to the notes for the AnyCompany background and architecture slides to help you with this exercise. You might also want to refer to the Appendix in the [AWS Well-Architected Framework](#).

1. Review the following three operational excellence questions from the AWS Well-Architected Framework:
  - OPS 4: How do you design your workload so that you can understand its state?
  - OPS 6: How do you mitigate deployment risk?
  - OPS 7: How do you know that you are ready to support a workload?
2. For each Well-Architected Framework question, answer what is the current state of the AnyCompany architecture and what is the final state.
3. Agree on the top improvement that AnyCompany should make.



## Security pillar

Security pillar

# Security pillar



## Security pillar



Protect and monitor systems

- **Focus**

- Protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies.

- **Key topics**

- Protecting confidentiality and integrity of data
- Identifying and managing who can do what
- Protecting systems
- Establishing controls to detect security events

The *Security pillar* focuses on the ability to protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies. Key topics include: protecting confidentiality and integrity of data, identifying and managing who can do what (or privilege management), protecting systems, and establishing controls to detect security events.

# Security design principles



## Security pillar



Protect and monitor systems

- Implement a strong identity foundation
- Enable traceability
- Apply security at all layers
- Automate security best practices
- Protect data in transit and at rest
- Keep people away from data
- Prepare for security events

There are seven design principles that can improve security:

- *Implement a strong identity foundation* – Implement the principle of least privilege and enforce separation of duties with appropriate authorization for each interaction with your AWS resources. Centralize privilege management and reduce or even eliminate reliance on long-term credentials.
- *Enable traceability* – Monitor, alert, and audit actions and changes to your environment in real time. Integrate logs and metrics with systems to automatically respond and take action.
- *Apply security at all layers* – Apply defense in depth and apply security controls to all layers of your architecture (for example, edge network, virtual private cloud, subnet, and load balancer; and every instance, operating system, and application).
- *Automate security best practices* – Automate security mechanisms to improve your ability to securely scale more rapidly and cost effectively. Create secure architectures and implement controls that are defined and managed as code in version-controlled templates.
- *Protect data in transit and at rest* – Classify your data into sensitivity levels and use mechanisms such as encryption, tokenization, and access control where appropriate.
- *Keep people away from data* – To reduce the risk of loss or modification of sensitive data due to human error, create mechanisms and tools to reduce or eliminate the need for direct access or manual processing of data.
- *Prepare for security events* – Have an incident management process that aligns with organizational requirements. Run incident response simulations and use tools with automation to increase your speed of detection, investigation, and recovery.

# Security questions



## Security

- How do you securely operate your workload?

## Identity and access management

- How do you manage identities for people and machines?
- How do you manage permissions for people and machines?

## Detection

- How do you detect and investigate security events?

## Infrastructure protection

- How do you protect your network resources?
- How do you protect your compute resources?

## Data protection

- How do you classify your data?
- How do you protect your data at rest?
- How do you protect your data in transit?

## Incident response

- How do you anticipate, respond to, and recover from incidents?

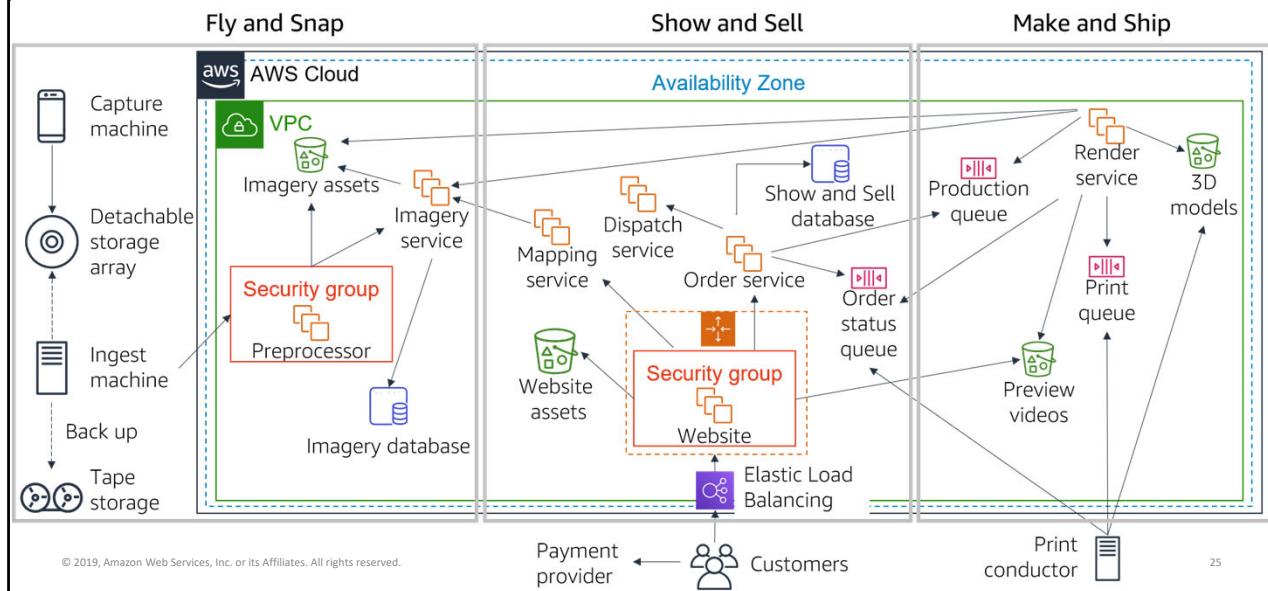
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

24

The foundational questions for security fall under six best practice areas: security, identity and access management, detection, infrastructure protection, data protection, and incident response.

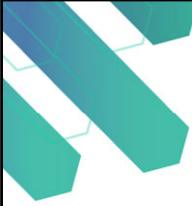
Before you architect any system, you must put security practices in place. You must be able to control who can do what. In addition, you must be able to identify security incidents, protect your systems and services, and maintain the confidentiality and integrity of data through data protection. You should have a well-defined and practiced process for responding to security incidents. These tools and techniques are important because they support objectives such as preventing financial loss or complying with regulatory obligations.

# Activity breakout



Here is the entire AnyCompany architecture for you to consult as you complete the activity. Refer to the notes for the AnyCompany background and architecture slides to help you with this exercise. You might also want to refer to the Appendix in the [AWS Well-Architected Framework](#).

1. Review the following three security questions from the AWS Well-Architected Framework:
  - SEC 1: How do you securely operate your workload?
  - SEC 4: How do you detect and investigate security events?
  - SEC 6: How do you protect your compute resources?
2. For each Well-Architected Framework question, answer what is the current state of the AnyCompany architecture and what is the final state.
3. Agree on the top improvement that AnyCompany should make.



## Reliability pillar

Reliability pillar

# Reliability pillar



## Reliability pillar



Recover from failure and mitigate disruption.

- **Focus**

- Ensure a workload performs its intended function correctly and consistently when it's expected to.

- **Key topics**

- Designing distributed systems
- Recovery planning
- Handling change

The *Reliability pillar* focuses on ensuring a workload performs its intended function correctly and consistently when it's expected to. A resilient workload quickly recovers from failures to meet business and customer demand. Key topics include: designing distributed systems, recovery planning, and handling change.

# Reliability design principles



## Reliability pillar



Recover from failure and mitigate disruption.

- Automatically recover from failure
- Test recovery procedures
- Scale horizontally to increase aggregate workload availability
- Stop guessing capacity
- Manage change in automation

There are five design principles that can increase reliability:

- *Automatically recover from failure* – Monitor systems for key performance indicators and configure your systems to trigger an automated recovery when a threshold is breached. This practice enables automatic notification and failure-tracking, and for automated recovery processes that work around or repair the failure.
- *Test recovery procedures* – Test how your systems fail and validate your recovery procedures. Use automation to simulate different failures or to recreate scenarios that led to failures before. This practice can expose failure pathways that you can test and rectify before a real failure scenario.
- *Scale horizontally to increase aggregate workload availability* – Replace one large resource with multiple, smaller resources and distribute requests across these smaller resources to reduce the impact of a single point of failure on the overall system.
- *Stop guessing capacity* – Monitor demand and system usage, and automate the addition or removal of resources to maintain the optimal level for satisfying demand.
- *Manage change in automation* – Use automation to make changes to infrastructure and manage changes in automation.

# Reliability questions



## Foundations

- How do you manage service quotas and constraints?
- How do you plan your network topology?

## Workload architecture

- How do you design your workload service architecture?
- How do you design interactions in a distributed system to prevent failure?
- How do you design interactions in a distributed system to mitigate or withstand failures?

## Change management

- How do you monitor workload resources?
- How do you design your workload to adapt to changes in demand?
- How do you implement change?

## Failure management

- How do you back up data?
- How do you use fault isolation to protect your workload?
- How do you design your workload to withstand component failures?
- How do you test reliability?
- How do you plan for disaster recovery?

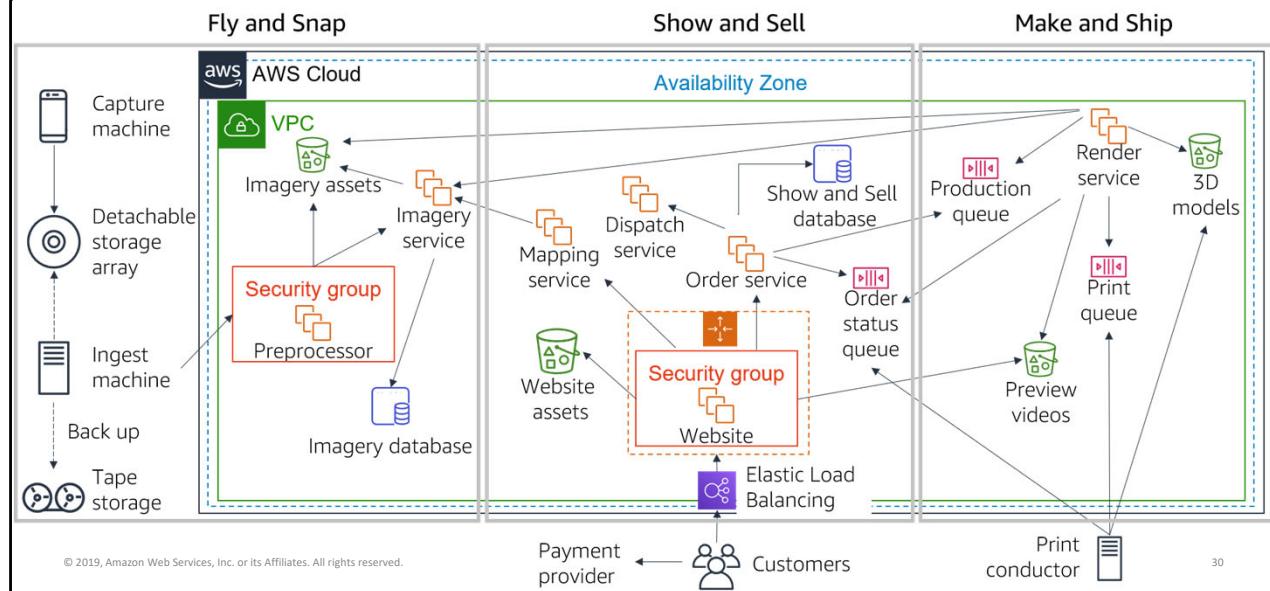
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

29

The foundational questions for reliability fall under four best practice areas: foundations, workload architecture, change management, and failure management.

To achieve reliability, a system must have both a well-planned foundation and monitoring in place. It must have mechanisms for handling changes in demand or requirements. The system should be designed to detect failure and automatically heal itself.

# Activity breakout



Here is the entire AnyCompany architecture for you to consult as you complete the activity. Refer to the notes for the AnyCompany background and architecture slides to help you with this exercise. You might also want to refer to the Appendix in the [AWS Well-Architected Framework](#).

1. Review the following three reliability questions from the AWS Well-Architected Framework:
  - REL 2: How do you plan your network topology?
  - REL 7: How do you design your system to adapt to changes in demand?
  - REL 9: How do you back up data?
2. For each Well-Architected Framework question, answer what is the current state of the AnyCompany architecture and what is the final state.
3. Agree on the top improvement that AnyCompany should make.



## Performance Efficiency pillar

Performance Efficiency pillar

# Performance Efficiency pillar



## Performance Efficiency pillar



Use resources sparingly.

- **Focus**

- Use IT and computing resources efficiently to meet system requirements and to maintain that efficiency as demand changes and technologies evolve.

- **Key topics**

- Selecting the right resource types and sizes based on workload requirements
- Monitoring performance
- Making informed decisions to maintain efficiency as business needs evolve

The *Performance Efficiency pillar* focuses on the ability to use IT and computing resources efficiently to meet system requirements, and to maintain that efficiency as demand changes or technologies evolve. Key topics include: selecting the right resource types and sizes based on workload requirements, monitoring performance, and making informed decisions to maintain efficiency as business needs evolve.

# Performance efficiency design principles



## Performance Efficiency pillar



Use resources sparingly.

- Democratize advanced technologies
- Go global in minutes
- Use serverless architectures
- Experiment more often
- Consider mechanical sympathy

There are five design principles that can improve performance efficiency:

- *Democratize advanced technologies* – Consume technologies as a service. For example, technologies such as NoSQL databases, media transcoding, and machine learning require expertise that is not evenly dispersed across the technical community. In the cloud, these technologies become services that teams can consume. Consuming technologies enables teams to focus on product development instead of resource provisioning and management.
- *Go global in minutes* – Deploy systems in multiple AWS Regions to provide lower latency and a better customer experience at minimal cost.
- *Use serverless architectures* – Serverless architectures remove the operational burden of running and maintaining servers to carry out traditional compute activities. Serverless architectures can also lower transactional costs because managed services operate at cloud scale.
- *Experiment more often* – Perform comparative testing of different types of instances, storage, or configurations.
- *Consider mechanical sympathy* – Use the technology approach that aligns best to what you are trying to achieve. For example, consider your data access patterns when you select approaches for databases or storage.

# Performance efficiency questions



## Selection

- How do you select the best performing architecture?
- How do you select your compute solution?
- How do you select your storage solution?
- How do you select your database solution?
- How do you configure your networking solution?

## Review

- How do you evolve your workload to take advantage of new releases?

## Monitoring

- How do you monitor your resources to ensure they are performing?

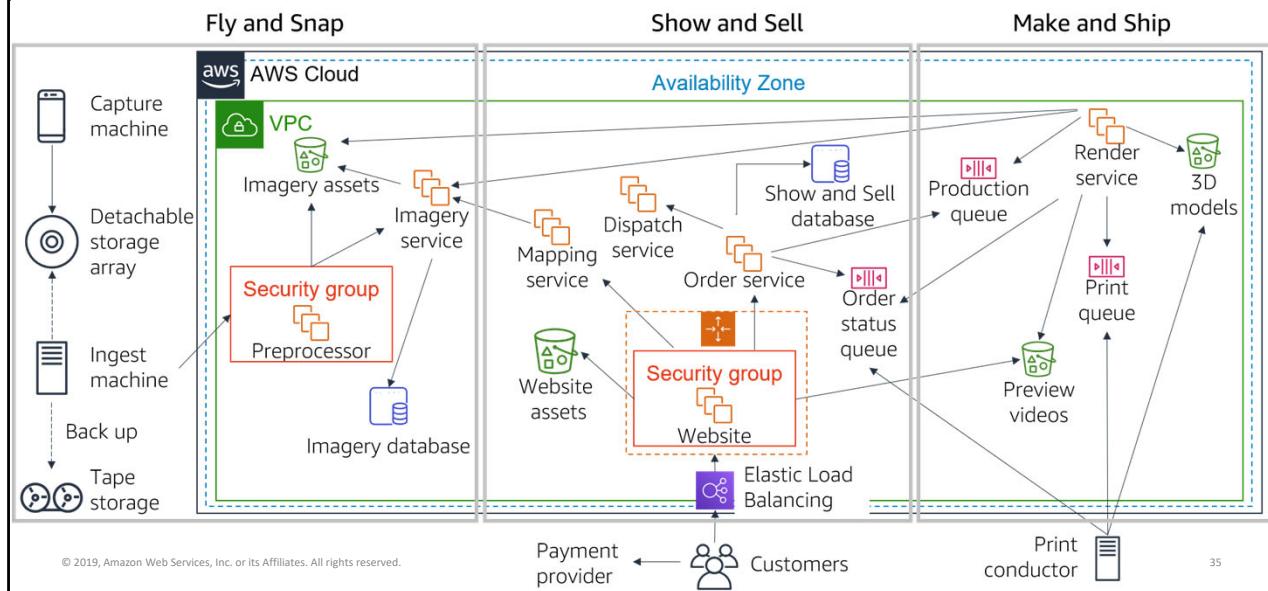
## Tradeoffs

- How do you use tradeoffs to improve performance?

The foundational questions for performance efficiency fall under four best practice areas: selection, review, monitoring, and tradeoffs.

Use data to design and build a high-performance architecture. Gather data on all aspects of the architecture, from the high-level design to the selection and configuration of resource types. Review your choices periodically to ensure that you are taking advantage of new AWS services. Perform monitoring so that you are aware of any deviance from expected performance and can take prompt action to remediate them. Finally, use tradeoffs in your architecture to improve performance, such as using compression, using caching, or relaxing consistency requirements.

# Activity breakout



Here is the entire AnyCompany architecture for you to consult as you complete the activity. Refer to the notes for the AnyCompany background and architecture slides to help you with this exercise. You might also want to refer to the Appendix in the [AWS Well-Architected Framework](#).

1. Review the following three performance efficiency questions from the AWS Well-Architected Framework:
  - **PERF 1:** How do you select the best performing architecture?
  - **PERF 2:** How do you select your compute solution?
  - **PERF 4:** How do you select your database solution?
2. For each Well-Architected Framework question, answer what is the current state of the AnyCompany architecture and what is the final state.
3. Agree on the top improvement that AnyCompany should make.



## Cost Optimization pillar

Cost Optimization pillar

# Cost Optimization pillar



## Cost Optimization pillar



Eliminate unneeded expense.

- **Focus**

- Avoid unnecessary costs.

- **Key topics**

- Understanding and controlling where money is being spent
- Selecting the most appropriate and right number of resource types
- Analyzing spend over time
- Scaling to meeting business needs without overspending

The *Cost Optimization pillar* focuses on the ability to avoid unnecessary costs. Key topics include: understanding and controlling where money is being spent, selecting the most appropriate and right number of resource types, analyzing spend over time, and scaling to meeting business needs without overspending.

# Cost optimization design principles



## Cost Optimization pillar



Eliminate unneeded expense.

- Implement Cloud Financial Management
- Adopt a consumption model
- Measure overall efficiency
- Stop spending money on undifferentiated heavy lifting
- Analyze and attribute expenditure

There are five design principles that can optimize costs:

- *Implement Cloud Financial Management* – To achieve financial success and accelerate business value realization in the cloud, you need to invest in cloud financial management and cost optimization. You need to build capability through knowledge building, programs, resources, and processes to become a cost-efficient organization.
- *Adopt a consumption model* – Pay only for the computing resources that you require. Increase or decrease usage depending on business requirements, not by using elaborate forecasting.
- *Measure overall efficiency* – Measure the business output of the workload and the costs that are associated with delivering it. Use this measure to know the gains that you make from increasing output and reducing costs.
- *Stop spending money on undifferentiated heavy lifting* – AWS does the heavy lifting of racking, stacking, and powering servers, which means that you can focus on your customers and business projects instead of the IT infrastructure.
- *Analyze and attribute expenditure* – The cloud makes it easier to accurately identify system usage and costs, and attribute IT costs to individual workload owners. Having this capability helps you measure return on investment (ROI) and gives workload owners an opportunity to optimize their resources and reduce costs.

# Cost optimization questions



## Practice cloud financial management

- How do you implement cloud financial management?

## Expenditure and usage awareness

- How do you govern usage?
- How do you monitor usage and cost?
- How do you decommission resources?

## Cost-effective resources

- How do you evaluate cost when you select services?
- How do you meet cost targets when you select resource type, size, and number?
- How do you use pricing models to reduce cost?
- How do you plan for data transfer changes?

## Manage demand and supply resources

- How do you manage demand and supply resources?

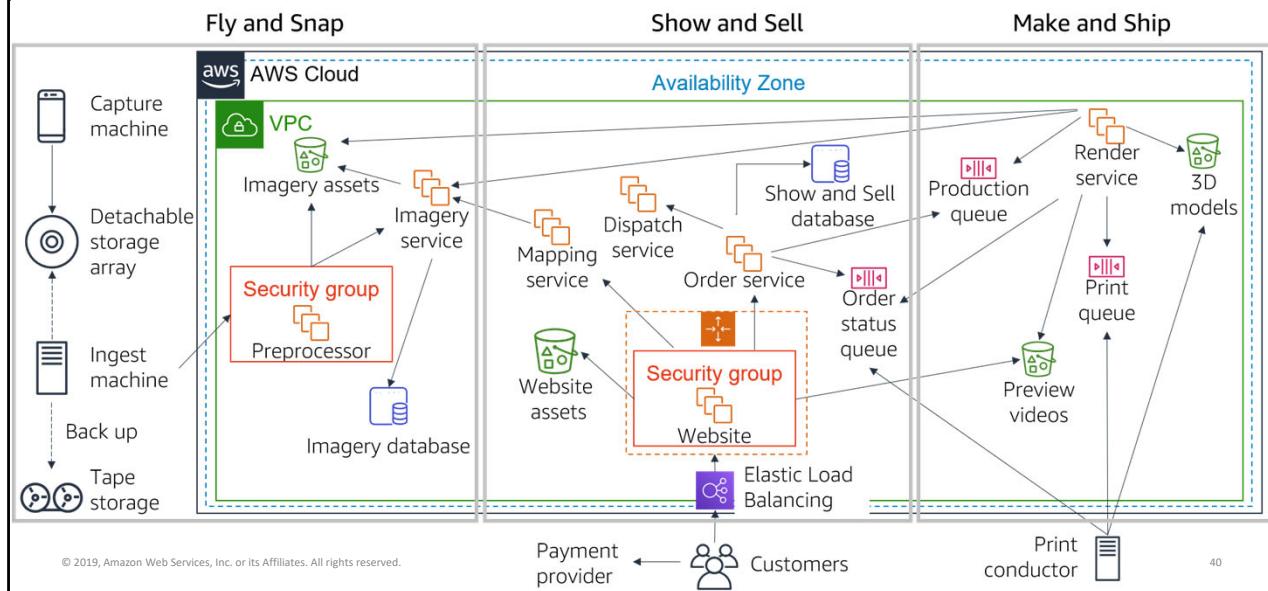
## Optimize over time

- How do you evaluate new services?

The foundational questions for cost optimization fall under five best practice areas: practice cloud financial management, expenditure and usage awareness, cost-effective resources, manage demand and supply resources, and optimize over time.

Similar to the other pillars, there are tradeoffs to consider when evaluating cost. For example, you may choose to prioritize for speed—going to market quickly, shipping new features, or simply meeting a deadline—instead of investing in upfront cost optimization. As another example, designing an application for a higher level of availability typically costs more. You should identify your true application needs and use empirical data to inform your architectural design decisions. Perform benchmarking to establish the most cost-optimal workload over time.

# Activity breakout



Here is the entire AnyCompany architecture for you to consult as you complete the activity. Refer to the notes for the AnyCompany background and architecture slides to help you with this exercise. You might also want to refer to the Appendix in the [AWS Well-Architected Framework](#).

1. Review the following three cost optimization questions from the AWS Well-Architected Framework:
  - COST 2: How do you govern usage?
  - COST 6: How do you meet cost targets when you select resource type, size, and number?
  - COST 7: How do you use pricing models to reduce cost?
2. For each Well-Architected Framework question, answer what is the current state of the AnyCompany architecture and what is the final state.
3. Agree on the top improvement that AnyCompany should make.

## The AWS Well-Architected Tool



- Helps you review the state of your workloads and compares them to the latest AWS architectural best practices
- Gives you access to knowledge and best practices used by AWS architects, whenever you need it
- Delivers an action plan with step-by-step guidance on how to build better workloads for the cloud
- Provides a consistent process for you to review and measure your cloud architectures

The activity that you just completed is similar to how you would use the AWS Well-Architected Tool.

The AWS Well-Architected Tool helps you review the state of your workloads and compare them to the latest AWS architectural best practices. It gives you access to knowledge and best practices used by AWS architects, whenever you need it.

This tool is available in the AWS Management Console. You define your workload and answer a series of questions in the areas of operational excellence, security, reliability, performance efficiency, and cost optimization (as defined in the AWS Well-Architected Framework). The AWS Well-Architected Tool then delivers an action plan with step-by-step guidance on how to improve your workload for the cloud.

The AWS Well-Architected Tool provides a consistent process for you to review and measure your cloud architectures. You can use the results that the tool provides to identify next steps for improvement, drive architectural decisions, and bring architecture considerations into your corporate governance process.

## Section 1 key takeaways



- The AWS Well-Architected Framework provides a **consistent approach** to evaluate cloud architectures **and guidance** to help implement designs.
- The AWS Well-Architected Framework documents a **set of design principles** and **best practices** that enable you to understand if a specific architecture aligns well with cloud best practices.
- The AWS Well-Architected Framework is organized into **six pillars**.
- Each pillar includes its own set of **design principles and best practices**.

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

Some key takeaways from this section of the module include:

- The AWS Well-Architected Framework provides a consistent approach to evaluate cloud architectures and guidance to help implement designs.
- The AWS Well-Architected Framework documents a set of design principles and best practices that enable you to understand if a specific architecture aligns well with cloud best practices.
- The AWS Well-Architected Framework is organized into six pillars.
- Each pillar includes its own set of design principles and best practices.

Module 9: Cloud Architecture

## Section 2: Reliability and availability

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



## Section 2: Reliability and availability

***“Everything fails, all the time.”***

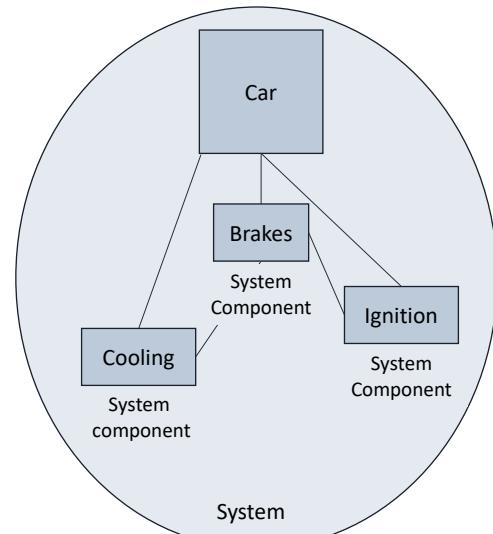
Werner Vogels, CTO, Amazon.com

In the words of Werner Vogels, Amazon’s CTO, “Everything fails, all the time.” One of the best practices that is identified in the AWS Well-Architected Framework is to plan for failure (or application or workload downtime). One way to do that is to architect your applications and workloads to withstand failure. There are two important factors that cloud architects consider when designing architectures to withstand failure: reliability and availability.

# Reliability



- A measure of your system's **ability to provide functionality** when desired by the user.
- **System** includes all system components: hardware, firmware, and software.
- **Probability** that your entire system will function as intended for a specified period.
- **Mean time between failures (MTBF)** = total time in service/number of failures



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

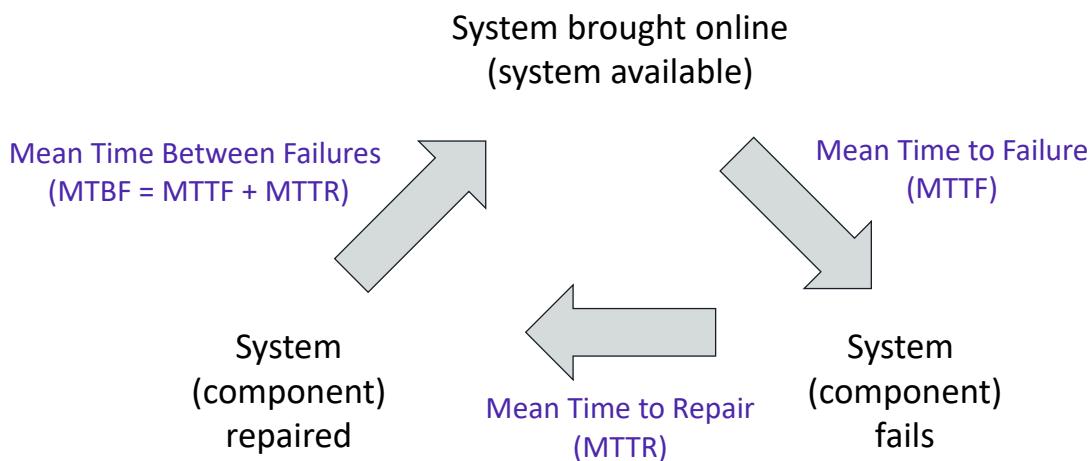
45

*Reliability* is a measure of your system's ability to provide functionality when desired by the user. Because "everything fails, all the time," you should think of reliability in statistical terms. Reliability is the probability that an entire system will function as intended for a specified period. Note that a system includes all system components, such as hardware, firmware, and software. Failure of system components impacts the availability of the system.

To understand reliability, it is helpful to consider the familiar example of a car. The car is the system. Each of the car's components (for example, cooling, ignition, and brakes) must work together in order for the car to work properly. If you try to start the car and the ignition fails, you cannot drive anywhere—the car is not available. If the ignition fails repeatedly, your car is not considered reliable.

A common way to measure reliability is to use statistical measurements, such as Mean Time Between Failures (MTBF). MTBF is the total time in service over the number of failures.

# Understanding reliability metrics



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

46

Say that you have an application that you bring online Monday at noon. The application is said to be *available*. It functions normally until it fails Friday at noon. Therefore, the time to failure (or the length of time the application is available) is 96 hours. You spend from Friday at noon until Monday at noon diagnosing why the application failed and repairing it, at which point you bring the application back online. Therefore, the time to repair is 72 hours.

Then, it happens again: the application fails on Friday at noon, you spend from Friday at noon until Monday at noon repairing it, and you bring it online on Monday at noon.

Say this failure-repair-restore cycle happens *every week*. You can now calculate the average of these numbers. In this example, your mean time to failure (MTTF) is 96 hours, and your mean time to repair (MTTR) is 72 hours. Your mean time between failures (MTBF) is 168 hours (or 1 week), which is the sum of MTTF and MTTR.

# Availability



- Normal operation time / total time
- A percentage of uptime (for example, 99.9 percent) over time (for example, 1 year)
- Number of 9s – Five 9s means 99.999 percent availability

As you just learned, failure of system components impacts the availability of the system.

Formally, *availability* is the percentage of time that a system is operating normally or correctly performing the operations expected of it (or normal operation time over total time). Availability is reduced anytime the application isn't operating normally, including both scheduled and unscheduled interruptions.

Availability is also defined as the percentage of uptime (that is, length of time that a system is online between failures) over a period of time (commonly 1 year).

A common shorthand when referring to availability is *number of 9s*. For example, *five 9s* means 99.999 percent availability.

## High availability



- System can withstand some measure of degradation while still remaining available.
- Downtime is minimized.
- Minimal human intervention is required.



A *highly available* system is one that can withstand some measure of degradation while still remaining available. In a highly available system, downtime is minimized as much as possible and minimal human intervention is required.

A highly available system can be viewed as a set of system-wide, shared resources that cooperate to guarantee essential services. High availability combines software with open-standard hardware to minimize downtime by quickly restoring essential services when a system, component, or application fails. Services are restored rapidly, often in less than 1 minute.

# Availability tiers



Availability	Max Disruption (per year)	Application Category
99%	3 days 15 hours	Batch processing, data extraction, transfer, and load jobs
99.9%	8 hours 45 minutes	Internal tools like knowledge management, project tracking
99.95%	4 hours 22 minutes	Online commerce, point of sale
99.99%	52 minutes	Video delivery, broadcast systems
99.999%	5 minutes	ATM transactions, telecommunications systems

Availability requirements vary. The length of disruption that is acceptable depends on the type of application. Here is a table of common application availability design goals and the maximum length of disruption that can occur within a year while still meeting the goal. The table contains examples of the types of applications that are common at each availability tier.

# Factors that influence availability



## Fault tolerance

- The **built-in redundancy** of an application's components and its ability to remain operational.

## Recoverability

- The process, policies, and procedures that are related to **restoring service** after a catastrophic event.

## Scalability

- The ability of an application to **accommodate increases in capacity needs** without changing design.

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

50

Though events that might disrupt an application's availability cannot always be predicted, you can build availability into your architecture design. There are three factors that determine the overall availability of your application:

- *Fault tolerance* refers to the **built-in redundancy** of an application's components and the **ability of the application to remain operational** even if some of its components fail. Fault tolerance relies on specialized hardware to detect failure in a system component (such as a processor, memory board, power supply, I/O subsystem, or storage subsystem) and instantaneously switch to a redundant hardware component. The fault-tolerant model does not address software failures, which are the most common reason for downtime.
- *Scalability* is the ability of your application to accommodate increases in capacity needs, remain available, and perform within your required standards. It does not guarantee availability, but it contributes to your application's availability.
- *Recoverability* is the ability to restore service quickly and without lost data if a disaster

makes your components unavailable, or it destroys data.

Keep in mind that improving availability usually leads to increased cost. When you consider how to make your environment more available, it's important to balance the cost of the improvement with the benefit to your users.

Do you want to ensure that your application is always alive or reachable, or do you want to ensure that it is servicing requests within an acceptable level of performance?

## Section 2 key takeaways



- **Reliability** is a measure of your system's ability to provide functionality when desired by the user, and it can be measured in terms of MTBF.
- **Availability** is the percentage of time that a system is operating normally or correctly performing the operations expected of it (or normal operation time over total time).
- Three factors that influence the availability of your applications are **fault tolerance**, **scalability**, and **recoverability**.
- You can design your workloads and applications to be **highly available**, but there is a cost tradeoff to consider.

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

Some key takeaways from this section of the module include:

- Reliability is a measure of your system's ability to provide functionality when desired by the user, and it can be measured in terms of MTBF.
- Availability is the percentage of time that a system is operating normally or correctly performing the operations expected of it (or normal operation time over total time).
- Three factors that influence the availability of your applications are fault tolerance, scalability, and recoverability.
- You can design your workloads and applications to be highly available, but there is a cost tradeoff to consider.

Module 9: Cloud Architecture

## Section 3: AWS Trusted Advisor

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



### Section 3: AWS Trusted Advisor

As you have learned so far, you can use the AWS Well-Architected Framework as you design your architectures to understand potential risks in your architecture, identify areas that need improvement, and drive architectural decisions. In this section, you will learn about AWS Trusted Advisor, which is a tool that you can use to review your AWS environment as soon as you start implementing your architectures.

# AWS Trusted Advisor



AWS Trusted Advisor

- Online tool that provides real-time guidance to help you provision your resources following AWS best practices.
- Looks at your entire AWS environment and gives you real-time recommendations in five categories.

Cost Optimization



0 ✓ 9 ⚠ 0 !  
\$7,516.85

Performance



3 ✓ 7 ⚠ 0 !

Security



2 ✓ 4 ⚠ 11 !

Fault Tolerance



0 ✓ 15 ⚠ 5 !

Service Limits



37 ✓ 0 ⚠ 1 !

Potential monthly savings

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

53

AWS Trusted Advisor is an online tool that provides real-time guidance to help you provision your resources following AWS best practices.

AWS Trusted Advisor looks at your entire AWS environment and gives you recommendations in five categories:

- *Cost Optimization* – AWS Trusted Advisor looks at your resource use and makes recommendations to help you optimize cost by eliminating unused and idle resources, or by making commitments to reserved capacity.
- *Performance* – Improve the performance of your service by checking your service limits, ensuring you take advantage of provisioned throughput, and monitoring for overutilized instances.
- *Security* – Improve the security of your application by closing gaps, enabling various AWS security features, and examining your permissions.
- *Fault Tolerance* – Increase the availability and redundancy of your AWS application by taking advantage of automatic scaling, health checks, Multi-AZ deployments, and backup capabilities.
- *Service Limits* – AWS Trusted Advisor checks for service usage that is more than 80 percent of the service limit. Values are based on a snapshot, so your current usage might differ. Limit and usage data can take up to 24 hours to reflect any changes.

For a detailed description of the information that AWS Trusted Advisor provides, see [AWS Trusted Advisor Best Practice Checks](#).

# Activity: Interpret AWS Trusted Advisor recommendations



## Trusted Advisor Dashboard

### Cost Optimization



9 ✓ 0 ▲ 0 !  
\$0.00

Potential monthly savings

### Performance



9 ✓ 1 ▲ 0 !

### Security



13 ✓ 2 ▲ 2 !

### Fault Tolerance



14 ✓ 2 ▲ 1 !

### Service Limits



48 ✓ 0 ▲ 0 !

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

54

You have a friend who used AWS Trusted Advisor for the first time. She is trying to interpret its recommendations to improve her cloud environment and needs your help. This is her dashboard. While everything looks OK in the cost optimization and service limit categories, you notice that there are a few recommendations that you should review to help her improve her security, performance, and fault tolerance.

Help your friend interpret the following recommendations.

# Activity: Recommendation #1



## MFA on Root Account

**Description:** Checks the root account and warns if multi-factor authentication (MFA) is not enabled. For increased security, we recommend that you protect your account by using MFA, which requires a user to enter a unique authentication code from their MFA hardware or virtual device when interacting with the AWS console and associated websites.

**Alert Criteria:** MFA is not enabled on the root account.

**Recommended Action:** Log in to your root account and activate an MFA device.

For this recommendation, answer these questions:

- What is the status?
- What is the problem?
- What specific environment details are you given?
- What is the best practice?
- What is the recommended action?

## Activity: Recommendation #2



### IAM Password Policy

**Description:** Checks the password policy for your account and warns when a password policy is not enabled, or if password content requirements have not been enabled. Password content requirements increase the overall security of your AWS environment by enforcing the creation of strong user passwords. When you create or change a password policy, the change is enforced immediately for new users but does not require existing users to change their passwords.

**Alert Criteria:** A password policy is enabled, but at least one content requirement is not enabled.

**Recommended Action:** If some content requirements are not enabled, consider enabling them. If no password policy is enabled, create and configure one. See [Setting an Account Password Policy for IAM Users](#).

For this recommendation, answer these questions:

- What is the status?
- What is the problem?
- What specific environment details are you given?
- What is the best practice?
- What is the recommended action?

# Activity: Recommendation #3



## 1 Security Groups – Unrestricted Access

**Description:** Checks security groups for rules that allow unrestricted access to a resource. Unrestricted access increases opportunities for malicious activity (hacking, denial-of-service attacks, loss of data).

**Alert Criteria:** A security group rule has a source IP address with a /0 suffix for ports other than 25, 80, or 443.)

**Recommended Action:** Restrict access to only those IP addresses that require it. To restrict access to a specific IP address, set the suffix to /32 (for example, 192.0.2.10/32). Be sure to delete overly permissive rules after creating rules that are more restrictive.

Region	Security Group Name	Security Group ID	Protocol	Port	Status	IP Range
us-east-1	WebServersSG	sg-xxxxxxxx1 (vpc-xxxxxxxx1)	tcp	22	Red	0.0.0.0/0
us-west-2	DatabaseServerSG	sg-xxxxxxxx2 (vpc-xxxxxxxx2)	tcp	8080	Red	0.0.0.0/0

For this recommendation, answer these questions:

- What is the status?
- What is the problem?
- What specific environment details are you given?
- What is the best practice?
- What is the recommended action?

## Activity: Recommendation #4



### Amazon EBS Snapshots

**Description:** Checks the age of the snapshots for your Amazon Elastic Block Store (Amazon EBS) volumes (available or in-use). Even though Amazon EBS volumes are replicated, failures can occur. Snapshots are persisted to Amazon Simple Storage Service (Amazon S3) for durable storage and point-in-time recovery.

**Alert Criteria:**

- Yellow: The most recent volume snapshot is between 7 and 30 days old.
- Red: The most recent volume snapshot is more than 30 days old.
- Red: The volume does not have a snapshot.

**Recommended Action:** Create weekly or monthly snapshots of your volumes

Region	Volume ID	Volume Name	Snapshot ID	Snapshot Name	Snapshot Age	Volume Attachment	Status	Reason
us-east-1	vol-xxxxxxxx	My-EBS-Volume				/dev/...	Red	No snapshot

For this recommendation, answer these questions:

- What is the status?
- What is the problem?
- What specific environment details are you given?
- What is the best practice?
- What is the recommended action?

## Activity: Recommendation #5



### Amazon S3 Bucket Logging

**Description:** Checks the logging configuration of Amazon Simple Storage Service (Amazon S3) buckets. When server access logging is enabled, detailed access logs are delivered hourly to a bucket that you choose. An access log record contains details about each request, such as the request type, the resources specified in the request, and the time and date the request was processed. By default, bucket logging is not enabled; you should enable logging if you want to perform security audits or learn more about users and usage patterns.

**Alert Criteria:**

Yellow: The bucket does not have server access logging enabled.

Yellow: The target bucket permissions do not include the owner account. Trusted Advisor cannot check it.

**Recommended Action:**

Enable bucket logging for most buckets.

If the target bucket permissions do not include the owner account and you want Trusted Advisor to check the logging status, add the owner account as a grantee.

Region	Bucket Name	Target Name	Target Exists	Same Owner	Write Enabled	Reason
us-east-2	my-hello-world-bucket		No	No	No	Logging not enabled

For this recommendation, answer these questions:

- What is the status?
- What is the problem?
- What specific environment details are you given?
- What is the best practice?
- What is the recommended action?

## Section 3 key takeaways



- AWS Trusted Advisor is an online tool that provides real-time guidance to help you provision your resources by following AWS best practices.
- AWS Trusted Advisor looks at your [entire AWS environment](#) and gives you real-time recommendations in five categories.
- You can use AWS Trusted Advisor to help you optimize your AWS environment as soon as you start implementing your architecture designs.

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

Some key takeaways from this section of the module include:

- AWS Trusted Advisor is an online tool that provides real-time guidance to help you provision your resources by following AWS best practices.
- AWS Trusted Advisor looks at your entire AWS environment and gives you real-time recommendations in five categories.
- You can use AWS Trusted Advisor to help you optimize your AWS environment as soon as you start implementing your architecture designs.

Module 9: Cloud Architecture

## Module wrap-up

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



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

## Module summary



In summary, in this module you learned how to:

- Describe the AWS Well-Architected Framework, including the six pillars
- Identify the design principles of the AWS Well-Architected Framework
- Explain the importance of reliability and high availability
- Identify how AWS Trusted Advisor helps customers
- Interpret AWS Trusted Advisor recommendations

In summary, in this module you learned how to:

- Describe the AWS Well-Architected Framework, including the six pillars
- Identify the design principles of the AWS Well-Architected Framework
- Explain the importance of reliability and high availability
- Identify how AWS Trusted Advisor helps customers
- Interpret AWS Trusted Advisor recommendations

## Complete the knowledge check



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

63

Now, complete the knowledge check.

## Sample exam question



A SysOps engineer working at a company wants to protect their data in transit and at rest. What services could they use to protect their data?

- A. Elastic Load Balancing
- B. Amazon Elastic Block Store (Amazon EBS)
- C. Amazon Simple Storage Service (Amazon S3)
- D. All of the above

Look at the answer choices and rule them out based on the keywords that were previously highlighted.

## Additional resources



- [AWS Well-Architected website](#)
- [AWS Well-Architected Labs](#)
- [AWS Trusted Advisor Best Practice Checks](#)

If you want to learn more about the topics covered in this module, you might find the following additional resources helpful:

- [AWS Well-Architected website](#)
- [AWS Well-Architected Labs](#)
- [AWS Trusted Advisor Best Practice Checks](#)

# Thank You

© 2019 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections or feedback on the course, please email us at: [aws-course-feedback@amazon.com](mailto:aws-course-feedback@amazon.com). For all other questions, contact us at: <https://aws.amazon.com/contact-us/aws-training/>. All trademarks are the property of their owners.



Thanks for participating!

AWS Academy Cloud Foundations

# Module 10: Automatic Scaling and Monitoring

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



Welcome to Module 10: Automatic Scaling and Monitoring

# Module overview



## Topics

- Elastic Load Balancing
- Amazon CloudWatch
- Amazon EC2 Auto Scaling

## Activities

- Elastic Load Balancing activity
- Amazon CloudWatch activity

## Lab

- Scale and Load Balance Your Architecture



## Knowledge check

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

2

This module will address the following topics:

- Elastic Load Balancing
- Amazon CloudWatch
- Amazon EC2 Auto Scaling

The module also includes two activities. One activity will challenge you to indicate Elastic Load Balancing use cases. The other activity will challenge you to identify Amazon CloudWatch examples.

The module also includes a hands-on lab where you will use Amazon EC2 Auto Scaling, Elastic Load Balancing, and Amazon CloudWatch together to create a dynamically scalable architecture.

Finally, you will be asked to complete a knowledge check that will test your understanding of key concepts that are covered in this module.

## Module objectives



After completing this module, you should be able to:

- Indicate how to distribute traffic across Amazon Elastic Compute Cloud (Amazon EC2) instances by using Elastic Load Balancing
- Identify how Amazon CloudWatch enables you to monitor AWS resources and applications in real time
- Explain how Amazon EC2 Auto Scaling launches and releases servers in response to workload changes
- Perform scaling and load balancing tasks to improve an architecture

After completing this module, you should be able to:

- Indicate how to distribute traffic across Amazon Elastic Compute Cloud (Amazon EC2) instances by using Elastic Load Balancing
- Identify how Amazon CloudWatch enables you to monitor AWS resources and applications in real time
- Explain how Amazon EC2 Auto Scaling launches and releases servers in response to workload changes
- Perform scaling and load balancing tasks to improve an architecture

**Module 10: Automatic Scaling and Monitoring**

## Section 1: Elastic Load Balancing

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

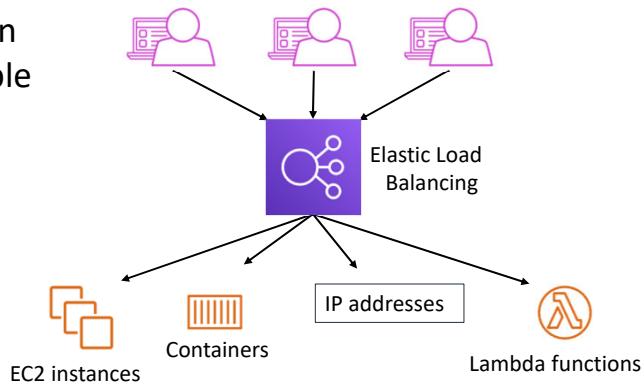


## Section 1: Elastic Load Balancing

# Elastic Load Balancing



- Distributes incoming application or network traffic across multiple targets in a single Availability Zone or across multiple Availability Zones.
- Scales your load balancer as traffic to your application changes over time.



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

5

Modern high-traffic websites must serve hundreds of thousands—if not millions—of concurrent requests from users or clients, and then return the correct text, images, video, or application data in a fast and reliable manner. Additional servers are generally required to meet these high volumes.

Elastic Load Balancing is an AWS service that distributes incoming application or network traffic across multiple targets—such as Amazon Elastic Compute Cloud (Amazon EC2) instances, containers, internet protocol (IP) addresses, and Lambda functions—in a single Availability Zone or across multiple Availability Zones. Elastic Load Balancing scales your load balancer as traffic to your application changes over time. It can automatically scale to most workloads.

# Types of load balancers



Application Load Balancer	Network Load Balancer	Classic Load Balancer (Previous Generation)
<ul style="list-style-type: none"><li>Load balancing of HTTP and HTTPS traffic</li><li>Routes traffic to targets based on content of request</li><li>Provides advanced request routing targeted at the delivery of modern application architectures, including microservices and containers</li><li>Operates at the application layer (OSI model layer 7)</li></ul>	<ul style="list-style-type: none"><li>Load balancing of TCP, UDP, and TLS traffic where extreme performance is required</li><li>Routes traffic to targets based on IP protocol data</li><li>Can handle millions of requests per second while maintaining ultra-low latencies</li><li>Is optimized to handle sudden and volatile traffic patterns</li><li>Operates at the transport layer (OSI model layer 4)</li></ul>	<ul style="list-style-type: none"><li>Load balancing of HTTP, HTTPS, TCP, and SSL traffic</li><li>Load balancing across multiple EC2 instances</li><li>Operates at both the application and transport layers.</li></ul>

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

6

Elastic Load Balancing is available in three types:

- An *Application Load Balancer* operates at the application level (Open Systems Interconnection, or OSI, model layer 7). It routes traffic to targets—Amazon Elastic Compute Cloud (Amazon EC2) instances, containers, Internet Protocol (IP) addresses, and Lambda functions—based on the content of the request. It is ideal for advanced load balancing of Hypertext Transfer Protocol (HTTP) and Secure HTTP (HTTPS) traffic. An Application Load Balancer provides advanced request routing that is targeted at delivery of modern application architectures, including microservices and container-based applications. An Application Load Balancer simplifies and improves the security of your application by ensuring that the latest Secure Sockets Layer/Transport Layer Security (SSL/TLS) ciphers and protocols are used at all times.
- A *Network Load Balancer* operates at the network transport level (OSI model layer 4), routing connections to targets—EC2 instances, microservices, and containers—based on IP protocol data. It works well for load balancing both Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) traffic. A Network Load Balancer is capable of handling millions of requests per second while maintaining ultra-low latencies. A

Network Load Balancer is optimized to handle sudden and volatile network traffic patterns.

- A *Classic Load Balancer* provides basic load balancing across multiple EC2 instances, and it operates at both the application level and network transport level. A Classic Load Balancer supports the load balancing of applications that use HTTP, HTTPS, TCP, and SSL. The Classic Load Balancer is an older implementation. When possible, AWS recommends that you use a dedicated Application Load Balancer or Network Load Balancer.

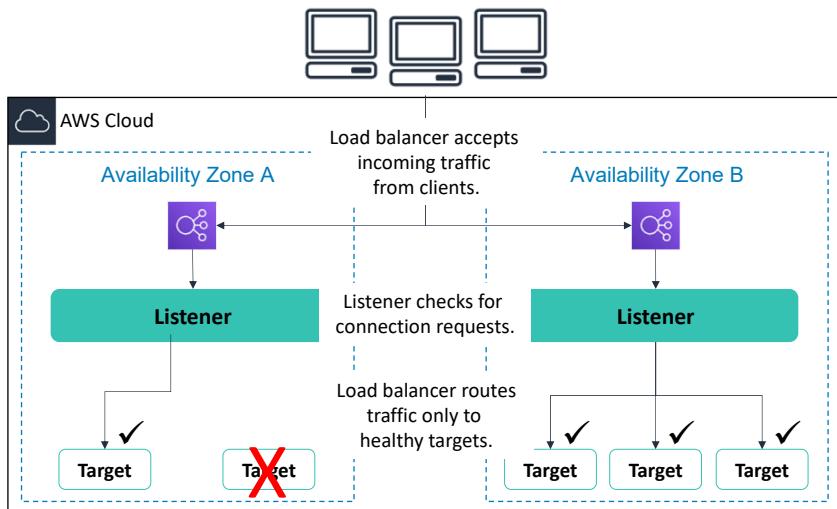
To learn more about the differences between the three types of load balancers, see *Product comparisons* on the Elastic Load Balancing [Features page](#).

# How Elastic Load Balancing works



- With Application Load Balancers and Network Load Balancers, you register targets in target groups, and route traffic to the target groups.
- With Classic Load Balancers, you register instances with the load balancer.

Load balancer performs health checks to monitor health of registered targets.



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

7

A load balancer accepts incoming traffic from clients and routes requests to its registered targets (such as EC2 instances) in one or more Availability Zones.

You configure your load balancer to accept incoming traffic by specifying one or more *listeners*. A listener is a process that checks for connection requests. It is configured with a protocol and port number for connections from clients to the load balancer. Similarly, it is configured with a protocol and port number for connections from the load balancer to the targets.

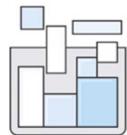
You can also configure your load balancer to perform *health checks*, which are used to monitor the health of the registered targets so that the load balancer only sends requests to the healthy instances. When the load balancer detects an unhealthy target, it stops routing traffic to that target. It then resumes routing traffic to that target when it detects that the target is healthy again.

There is a key difference in how the load balancer types are configured. With Application Load Balancers and Network Load Balancers, you register targets in *target groups*, and route traffic to the target groups. With Classic Load Balancers, you register instances with the load balancer.

# Elastic Load Balancing use cases



Highly available and fault-tolerant applications



Containerized applications



Elasticity and scalability



Virtual private cloud (VPC)



Hybrid environments



Invoke Lambda functions over HTTP(S)

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

8

There are many reasons to use a load balancer:

- *Achieve high availability and better fault tolerance for your applications* – Elastic Load Balancing balances traffic across healthy targets in multiple Availability Zones. If one or more of your targets in a single Availability Zone are unhealthy, Elastic Load Balancing will route traffic to healthy targets in other Availability Zones. After the targets return to a healthy state, load balancing will automatically resume traffic to them.
- *Automatically load balance your containerized applications* – With enhanced container support for Elastic Load Balancing, you can now load balance across multiple ports on the same EC2 instance. You can also take advantage of deep integration with Amazon Elastic Container Service (Amazon ECS), which provides a fully-managed container offering. You only need to register a service with a load balancer, and Amazon ECS transparently manages the registration and de-registration of Docker containers. The load balancer automatically detects the port and dynamically reconfigures itself.
- *Automatically scale your applications* – Elastic Load Balancing works with Amazon CloudWatch and Amazon EC2 Auto Scaling to help you scale your applications to the demands of your customers. Amazon CloudWatch alarms can trigger auto scaling for your EC2 instance fleet when the latency of any one of your EC2 instances exceeds a preconfigured threshold. Amazon EC2 Auto Scaling then provisions new instances and your applications will be ready to serve the next customer request. The load balancer

will register the EC2 instance and direct traffic to it as needed.

- *Use Elastic Load Balancing in your virtual private cloud (VPC)* – You can use Elastic Load Balancing to create a public entry point into your VPC, or to route request traffic between tiers of your application within your VPC. You can assign security groups to your load balancer to control which ports are open to a list of allowed sources. Because Elastic Load Balancing works with your VPC, all your existing network access control lists (network ACLs) and routing tables continue to provide additional network controls. When you create a load balancer in your VPC, you can specify whether the load balancer is public (default) or internal. If you select internal, you do not need to have an internet gateway to reach the load balancer, and the private IP addresses of the load balancer will be used in the load balancer's Domain Name System (DNS) record.
- *Enable hybrid load balancing* – Elastic Load Balancing enables you to load balance across AWS and on-premises resources by using the same load balancer. For example, if you must distribute application traffic across both AWS and on-premises resources, you can register all the resources to the same target group and associate the target group with a load balancer. Alternatively, you can use DNS-based weighted load balancing across AWS and on-premises resources by using two load balancers, with one load balancer for AWS and other load balancer for on-premises resources. You can also use hybrid load balancing to benefit separate applications where one application is in a VPC and the other application is in an on-premises location. Put the VPC targets in one target group and the on-premises targets in another target group, and then use content-based routing to route traffic to each target group.
- *Invoking Lambda functions over HTTP(S)* – Elastic Load Balancing supports invoking Lambda functions to serve HTTP(S) requests. This enables users to access serverless applications from any HTTP client, including web browsers. You can register Lambda functions as targets and use the support for content-based routing rules in Application Load Balancers to route requests to different Lambda functions. You can use an Application Load Balancer as a common HTTP endpoint for applications that use servers and serverless computing. You can build an entire website by using Lambda functions, or combine EC2 instances, containers, on-premises servers, and Lambda functions to build applications.

# Activity: Elastic Load Balancing



You must support traffic to a containerized application.

Application Load Balancer

You have extremely spiky and unpredictable TCP traffic.

Network Load Balancer

You need simple load balancing with multiple protocols.

Classic Load Balancer

You need to support a static or Elastic IP address, or an IP target outside a VPC.

Network Load Balancer

You need a load balancer that can handle millions of requests per second while maintaining low latencies.

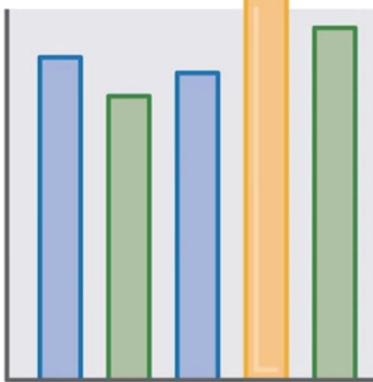
Network Load Balancer

You must support HTTPS requests.

Application Load Balancer

For this activity, name the load balancer you would use for the given scenario.

# Load balancer monitoring



- **Amazon CloudWatch metrics** – Used to verify that the system is performing as expected and creates an alarm to initiate an action if a metric goes outside an acceptable range.
- **Access logs** – Capture detailed information about requests sent to your load balancer.
- **AWS CloudTrail logs** – Capture the who, what, when, and where of API interactions in AWS services.

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

10

You can use the following features to monitor your load balancers, analyze traffic patterns, and troubleshoot issues with your load balancers and targets:

- *Amazon CloudWatch metrics* – Elastic Load Balancing publishes data points to Amazon CloudWatch for your load balancers and your targets. CloudWatch enables you to retrieve statistics about those data points as an ordered set of time series data, known as metrics. You can use metrics to verify that your system is performing as expected. For example, you can create a CloudWatch alarm to monitor a specified metric and initiate an action (such as sending a notification to an email address) if the metric goes outside what you consider an acceptable range.
- *Access logs* – You can use access logs to capture detailed information about the requests that were made to your load balancer and store them as log files in Amazon Simple Storage Service (Amazon S3). You can use these access logs to analyze traffic patterns and to troubleshoot issues with your targets or backend applications.
- *AWS CloudTrail logs* – You can use AWS CloudTrail to capture detailed information about the calls that were made to the Elastic Load Balancing application programming interface (API) and store them as log files in Amazon S3. You can use these CloudTrail logs to determine who made the call, what calls were made, when the call was made,

the source IP address of where the call came from, and so on.

## Section 1 key takeaways



- Elastic Load Balancing distributes incoming application or network traffic across multiple targets in one or more Availability Zones.
- Elastic Load Balancing supports three types of load balancers:
  - Application Load Balancer
  - Network Load Balancer
  - Classic Load Balancer
- ELB offers instance health checks, security, and monitoring.

Some key takeaways from this section of the module include:

- Elastic Load Balancing distributes incoming application or network traffic across multiple targets (such as Amazon EC2 instances, containers, IP addresses, and Lambda functions) in one or more Availability Zones.
- Elastic Load Balancing supports three types of load balancers:
  - Application Load Balancer
  - Network Load Balancer
  - Classic Load Balancer
- Elastic Load Balancing offers several monitoring tools for continuous monitoring and logging for auditing and analytics.

**Module 10: Automatic Scaling and Monitoring**

## Section 2: Amazon CloudWatch

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



## Section 2: Amazon CloudWatch

# Monitoring AWS resources



To use AWS efficiently, you need insight into your AWS resources:

- How do you know when you should **launch more Amazon EC2 instances?**
- Is your **application's performance or availability** being affected by a lack of sufficient capacity?
- How much of your infrastructure is actually **being used?**

To use AWS efficiently, you need insight into your AWS resources.

For example, you might want to know:

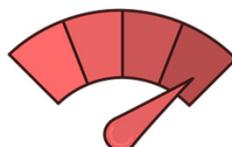
- When you should launch more Amazon EC2 instances?
- If your application's performance or availability is being affected by a lack of sufficient capacity?
- How much of your infrastructure is actually being used?

How do you capture this information?

# Amazon CloudWatch



Amazon  
CloudWatch



- Monitors –
  - AWS resources
  - Applications that run on AWS
- Collects and tracks –
  - Standard metrics
  - Custom metrics
- Alarms –
  - Send notifications to an Amazon SNS topic
  - Perform Amazon EC2 Auto Scaling or Amazon EC2 actions
- Events –
  - Define rules to match changes in AWS environment and route these events to one or more target functions or streams for processing

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

14

You can capture this information with Amazon CloudWatch.

Amazon CloudWatch is a monitoring and observability service that is built for DevOps engineers, developers, site reliability engineers (SRE), and IT managers. CloudWatch monitors your AWS resources (and the applications that you run on AWS) in real time. You can use CloudWatch to collect and track metrics, which are variables that you can measure for your resources and applications.

You can create an alarm to monitor any Amazon CloudWatch metric in your account and use the alarm to automatically send a notification to an Amazon Simple Notification Service (Amazon SNS) topic or perform an Amazon EC2 Auto Scaling or Amazon EC2 action. For example, you can create alarms on the CPU utilization of an EC2 instance, Elastic Load Balancing request latency, Amazon DynamoDB table throughput, Amazon Simple Queue Service (Amazon SQS) queue length, or even the charges on your AWS bill. You can also create an alarm on custom metrics that are specific to your custom applications or infrastructure.

You can also use Amazon CloudWatch Events to define rules that match incoming events (or changes in your AWS environment) and route them to targets for processing. Targets can include Amazon EC2 instances, AWS Lambda functions, Kinesis streams, Amazon ECS tasks, Step Functions state machines, Amazon SNS topics, Amazon SQS queues, and built-in targets. CloudWatch Events becomes aware of operational changes as they occur. CloudWatch Events responds to these operational changes and takes corrective action as necessary, by sending messages to respond to the environment, activating functions,

making changes, and capturing state information.

With CloudWatch, you gain system-wide visibility into resource utilization, application performance, and operational health. There is no upfront commitment or minimum fee; you simply pay for what you use. You are charged at the end of the month for what you use.

# CloudWatch alarms



- Create alarms based on –
  - Static threshold
  - Anomaly detection
  - Metric math expression
- Specify –
  - Namespace
  - Metric
  - Statistic
  - Period
  - Conditions
  - Additional configuration
  - Actions

Statistic

Period

5 minutes

Conditions

Threshold type

Static  
Use a value as a threshold

Anomaly detection  
Use a band as a threshold

Whenever CPUUtilization is...

Define the alarm condition

Greater

Greater/Equal

Lower/Equal

Lower

than...  
Define the threshold value

100

Must be a number

► Additional configuration

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

15

You can create a CloudWatch alarm that watches a single CloudWatch metric or the result of a math expression based on CloudWatch metrics. You can create a CloudWatch alarm based on a static threshold, anomaly detection, or a metric math expression.

When you create an alarm based on a static threshold, you choose a CloudWatch metric for the alarm to watch and the threshold for that metric. The alarm goes to ALARM state when the metric breaches the threshold for a specified number of evaluation periods.

For an alarm based on a static threshold, you must specify the:

- *Namespace* – A namespace contains the CloudWatch metric that you want, for example, AWS/EC2.
- *Metric* – A metric is the variable you want to measure, for example, *CPU Utilization*.
- *Statistic* – A statistic can be an average, sum, minimum, maximum, sample count, a predefined percentile, or a custom percentile.
- *Period* – A period is the evaluation period for the alarm. When the alarm is evaluated, each period is aggregated into one data point.
- *Conditions* – When you specify the conditions for a static threshold, you specify whenever the metric is *Greater*, *Greater or Equal*, *Lower or Equal*, or *Lower* than the threshold value, and you also specify the threshold value.
- *Additional configuration information* – This includes the number of data points within the evaluation period that must be breached to trigger the alarm, and how CloudWatch should treat missing data when it evaluates the alarm.
- *Actions* – You can choose to send a notification to an Amazon SNS topic, or to perform

an Amazon EC2 Auto Scaling action or Amazon EC2 action.

For more information on creating CloudWatch alarms, see the topics under [Using Alarms](#) in the AWS Documentation.

## Activity: Amazon CloudWatch



 Amazon EC2	If average CPU utilization is > 60% for 5 minutes...	Correct!
 Amazon RDS	If the number of simultaneous connections is > 10 for 1 minute...	Correct!
 Amazon S3	If the maximum bucket size in bytes is around 3 for 1 day...	Incorrect. <i>Around</i> is not a threshold option. You must specify a threshold of <i>&gt;</i> , <i>&gt;=</i> , <i>&lt;=</i> , or <i>&lt;</i> .
 Elastic Load Balancing	If the number of healthy hosts is < 5 for 10 minutes...	Correct!
 Amazon Elastic Block Store	If the volume of read operations is > 1,000 for 10 seconds...	Incorrect. You must specify a statistic (for example, <i>average volume</i> ).

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

16

For this activity, see if you can identify which are correct CloudWatch alarms. For the ones that are incorrect, see if you can identify the error.

## Section 2 key takeaways



17



- Amazon CloudWatch helps you monitor your AWS resources—and the applications that you run on AWS—in real time.
- CloudWatch enables you to –
  - Collect and track standard and custom metrics.
  - Set alarms to automatically send notifications to SNS topics, or perform Amazon EC2 Auto Scaling or Amazon EC2 actions.
  - Define rules that match changes in your AWS environment and route these events to targets for processing.

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

Some key takeaways from this section of the module include:

- Amazon CloudWatch helps you monitor your AWS resources—and the applications that you run on AWS—in real time.
- CloudWatch enables you to –
  - Collect and track standard and custom metrics.
  - Set alarms to automatically send notifications to SNS topics or perform Amazon EC2 Auto Scaling or Amazon EC2 actions based on the value of the metric or expression relative to a threshold over a number of time periods.
  - Define rules that match changes in your AWS environment and route these events to targets for processing.

**Module 10: Automatic Scaling and Monitoring**

## Section 3: Amazon EC2 Auto Scaling

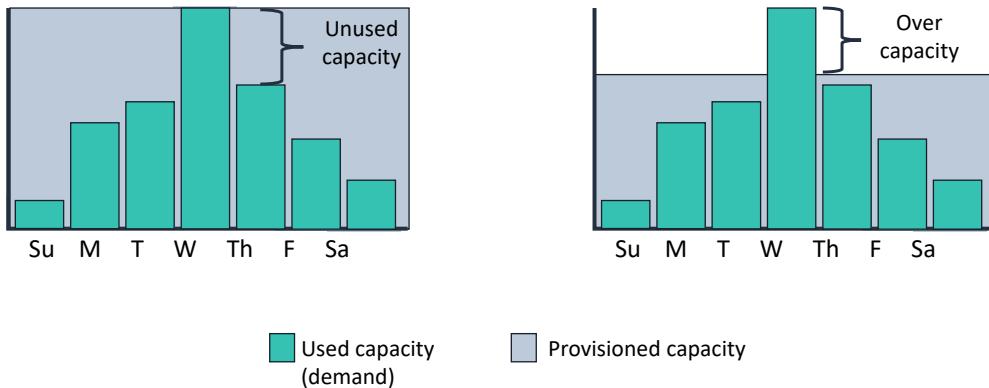
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



### Section 3: Amazon EC2 Auto Scaling

When you run your applications on AWS, you want to ensure that your architecture can scale to handle changes in demand. In this section, you will learn how to automatically scale your EC2 instances with Amazon EC2 Auto Scaling.

# Why is scaling important?



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

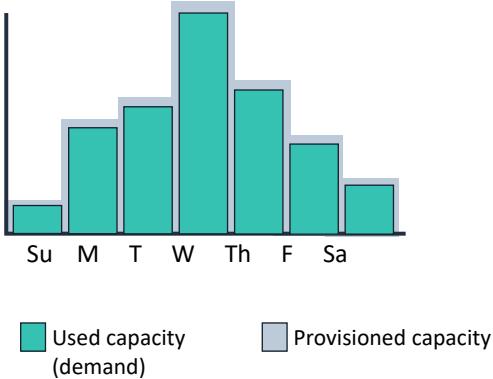
19

Scaling is the ability to increase or decrease the compute capacity of your application. To understand why scaling is important, consider this example of a workload that has varying resource requirements. In this example, the most resource capacity is required on Wednesday, and the least resource capacity is required on Sunday.

One option is to allocate more than enough capacity so you can always meet your highest demand—in this case, Wednesday. However, this situation means that you are running resources that will be underutilized most days of the week. With this option, your costs are not optimized.

Another option is to allocate less capacity to reduce costs. This situation means that you are under capacity on certain days. If you don't solve your capacity problem, your application could underperform or potentially even become unavailable for users.

# Amazon EC2 Auto Scaling



- Helps you maintain application availability
- Enables you to automatically add or remove EC2 instances according to conditions that you define
- Detects impaired EC2 instances and unhealthy applications, and replaces the instances without your intervention
- Provides several scaling options – Manual, scheduled, dynamic or on-demand, and predictive

In the cloud, because computing power is a programmatic resource, you can take a flexible approach to scaling. Amazon EC2 Auto Scaling is an AWS service that helps you maintain application availability and enables you to automatically add or remove EC2 instances according to conditions you define. You can use the fleet management features of EC2 Auto Scaling to maintain the health and availability of your fleet.

Amazon EC2 Auto Scaling provides several ways to adjust scaling to best meet the needs of your applications. You can add or remove EC2 instances manually, on a schedule, in response to changing demand, or in combination with AWS Auto Scaling for predictive scaling. Dynamic scaling and predictive scaling can be used together to scale faster.

To learn more about Amazon EC2 Auto Scaling, see the [Amazon EC2 Auto Scaling](#) product page.

## Typical weekly traffic at Amazon.com



Provisioned capacity



Sunday      Monday      Tuesday      Wednesday      Thursday      Friday      Saturday

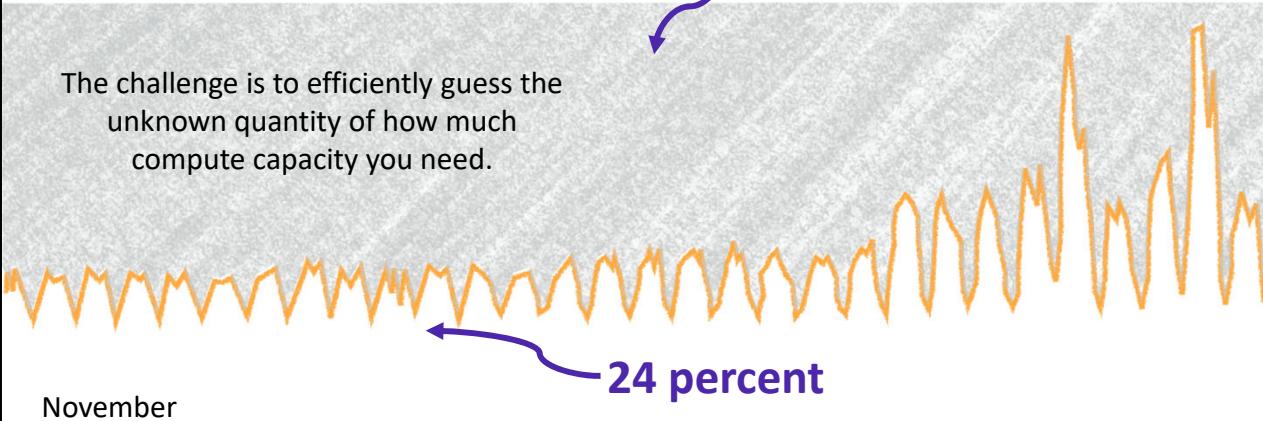
Automatic scaling is useful for predictable workloads—for example, the weekly traffic at the retail company Amazon.com.

## November traffic to Amazon.com



Provisioned capacity

The challenge is to efficiently guess the unknown quantity of how much compute capacity you need.

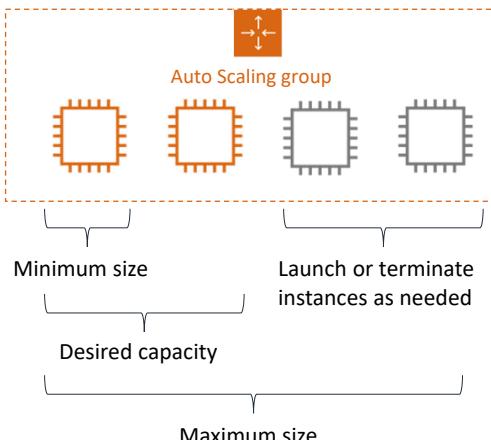


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

22

Automatic scaling is also useful for dynamic on-demand scaling. Amazon.com experiences a seasonal peak in traffic in November (on Black Friday and Cyber Monday, which are days at the end of November when US retailers hold major sales). If Amazon provisions a fixed capacity to accommodate the highest use, 76 percent of the resources are idle for most of the year. Capacity scaling is necessary to support the fluctuating demands for service. Without scaling, the servers could crash due to saturation, and the business would lose customer confidence.

# Auto Scaling groups



An **Auto Scaling group** is a collection of EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management.

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

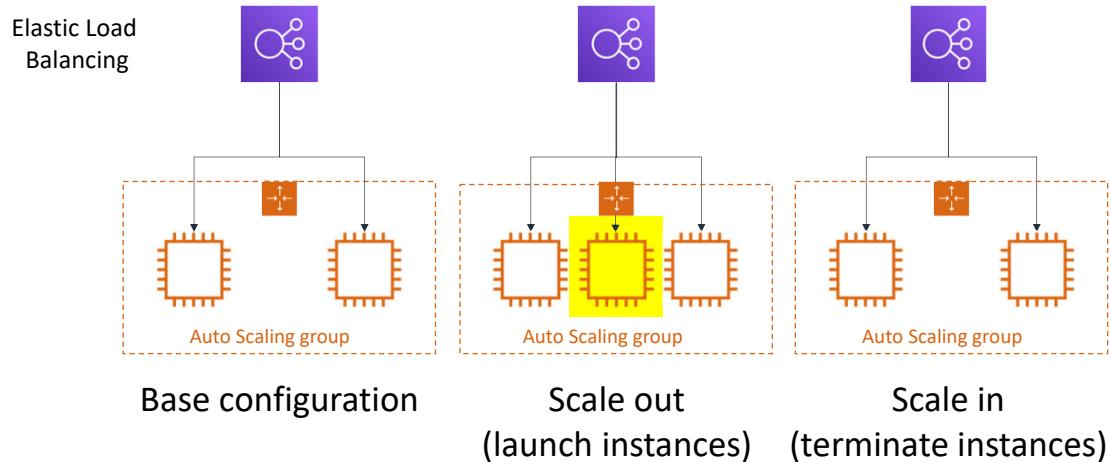
23

An [Auto Scaling group](#) is a collection of Amazon EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management. The size of an Auto Scaling group depends on the number of instances you set as the *desired capacity*. You can adjust its size to meet demand, either manually or by using automatic scaling.

You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling is designed to prevent your group from going below this size. You can specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling is designed to prevent your group from going above this size. If you specify the desired capacity, either when you create the group or at any time afterwards, Amazon EC2 Auto Scaling is designed to adjust the size of your group so it has the specified number of instances. If you specify scaling policies, then Amazon EC2 Auto Scaling can launch or terminate instances as demand on your application increases or decreases.

For example, this Auto Scaling group has a minimum size of one instance, a desired capacity of two instances, and a maximum size of four instances. The scaling policies that you define adjust the number of instances within your minimum and maximum number of instances, based on the criteria that you specify.

# Scaling out versus scaling in



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

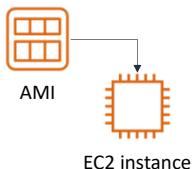
24

With Amazon EC2 Auto Scaling, launching instances is referred to as *scaling out*, and terminating instances is referred to as *scaling in*.

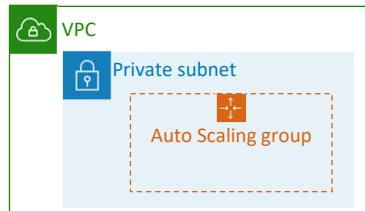
# How Amazon EC2 Auto Scaling works



## What



## Where



### Launch configuration

- AMI
- Instance type
- IAM role
- Security groups
- EBS volumes

### Auto Scaling group

- VPC and subnets
- Load balancer

## When

### Maintain current number

- Health checks

### Manual scaling

- Min, max, desired capacity

### Scheduled scaling

- Scheduled actions

### Dynamic scaling

- Scaling policies

### Predictive scaling

- AWS Auto Scaling

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

25

To launch EC2 instances, an Auto Scaling group uses a [launch configuration](#), which is an instance configuration template. You can think of a launch configuration as *what* you are scaling. When you create a launch configuration, you specify information for the instances. The information you specify includes the ID of the Amazon Machine Image (AMI), the instance type, AWS Identity and Access Management (IAM) role, additional storage, one or more security groups, and any Amazon Elastic Block Store (Amazon EBS) volumes.

You define the minimum and maximum number of instances and desired capacity of your Auto Scaling group. Then, you launch it into a subnet within a VPC (you can think of this as *where* you are scaling). Amazon EC2 Auto Scaling integrates with Elastic Load Balancing to enable you to attach one or more load balancers to an existing Auto Scaling group. After you attach the load balancer, it automatically registers the instances in the group and distributes incoming traffic across the instances.

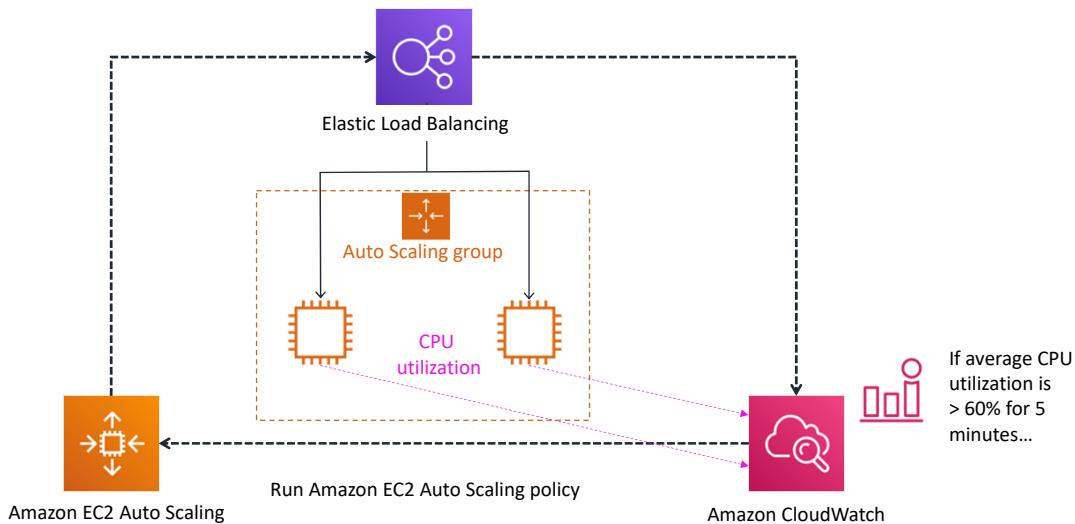
Finally, you specify *when* you want the scaling event to occur. You have many scaling options:

- *Maintain current instance levels at all times* – You can configure your Auto Scaling group to maintain a specified number of running instances at all times. To maintain the current instance levels, Amazon EC2 Auto Scaling performs a periodic health check on running instances in an Auto Scaling group. When Amazon EC2 Auto Scaling finds an unhealthy instance, it terminates that instance and launches a new one.

- [Manual scaling](#) – With manual scaling, you specify only the change in the maximum, minimum, or desired capacity of your Auto Scaling group.
- [Scheduled scaling](#) – With scheduled scaling, scaling actions are performed automatically as a function of date and time. This is useful for predictable workloads when you know exactly when to increase or decrease the number of instances in your group. For example, say that every week, the traffic to your web application starts to increase on Wednesday, remains high on Thursday, and starts to decrease on Friday. You can plan your scaling actions based on the predictable traffic patterns of your web application. To implement scheduled scaling, you create a [scheduled action](#).
- [Dynamic, on-demand scaling](#) – A more advanced way to scale your resources enables you to define parameters that control the scaling process. For example, you have a web application that currently runs on two instances and you want the CPU utilization of the Auto Scaling group to stay close to 50 percent when the load on the application changes. This option is useful for scaling in response to changing conditions, when you don't know when those conditions will change. Dynamic scaling gives you extra capacity to handle traffic spikes without maintaining an excessive amount of idle resources. You can configure your Auto Scaling group to scale automatically to meet this need. The [scaling policy type](#) determines how the scaling action is performed. You can use Amazon EC2 Auto Scaling with Amazon CloudWatch to trigger the scaling policy in response to an alarm.
- [Predictive scaling](#) – You can use Amazon EC2 Auto Scaling with AWS Auto Scaling to implement predictive scaling, where your capacity scales based on predicted demand. Predictive scaling uses data that is collected from your actual EC2 usage, and the data is further informed by billions of data points that are drawn from our own observations. AWS then uses well-trained machine learning models to predict your expected traffic (and EC2 usage), including daily and weekly patterns. The model needs at least 1 day of historical data to start making predictions. It is re-evaluated every 24 hours to create a forecast for the next 48 hours. The prediction process produces a scaling plan that can drive one or more groups of automatically scaled EC2 instances.

To learn more about these options, see [Scaling the Size of Your Auto Scaling Group](#) in the AWS Documentation.

# Implementing dynamic scaling



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

26

One common configuration for implementing dynamic scaling is to create a CloudWatch alarm that is based on performance information from your EC2 instances or load balancer. When a performance threshold is breached, a CloudWatch alarm triggers an automatic scaling event that either scales out or scales in EC2 instances in the Auto Scaling group.

To understand how it works, consider this example:

- You create an Amazon CloudWatch alarm to monitor CPU utilization across your fleet of EC2 instances and run automatic scaling policies if the average CPU utilization across the fleet goes above 60 percent for 5 minutes.
- Amazon EC2 Auto Scaling instantiates a new EC2 instance into your Auto Scaling group based on the launch configuration that you create.
- After the new instance is added, Amazon EC2 Auto Scaling makes a call to Elastic Load Balancing to register the new EC2 instance in that Auto Scaling group.
- Elastic Load Balancing then performs the required health checks and starts distributing traffic to that instance. Elastic Load Balancing routes traffic between EC2 instances and feeds metrics to Amazon CloudWatch.

Amazon CloudWatch, Amazon EC2 Auto Scaling, and Elastic Load Balancing work well individually. Together, however, they become more powerful and increase the control and flexibility over how your application handles customer demand.

# AWS Auto Scaling



## AWS Auto Scaling

- Monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost
- Provides a simple, powerful user interface that enables you to build scaling plans for resources, including –
  - Amazon EC2 instances and Spot Fleets
  - Amazon Elastic Container Service (Amazon ECS) Tasks
  - Amazon DynamoDB tables and indexes
  - Amazon Aurora Replicas

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

27

So far, you learned about scaling EC2 instances with Amazon EC2 Auto Scaling. You also learned that you can use Amazon EC2 Auto Scaling with AWS Auto Scaling to perform predictive scaling.

AWS Auto Scaling is a separate service that monitors your applications. It automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. The service provides a simple, powerful user interface that enables you to build scaling plans for resources, including:

- Amazon EC2 instances and Spot Fleets
- Amazon Elastic Container Service (Amazon ECS) tasks
- Amazon DynamoDB tables and indexes
- Amazon Aurora Replicas

If you are already using Amazon EC2 Auto Scaling to dynamically scale your EC2 instances, you can now use it with AWS Auto Scaling to scale additional resources for other AWS services.

To learn more information about AWS Auto Scaling, see [AWS Auto Scaling](#).

## Section 3 key takeaways



28



- Scaling enables you to respond quickly to changes in resource needs.
- Amazon EC2 Auto Scaling maintains application availability by automatically adding or removing EC2 instances.
- An Auto Scaling group is a collection of EC2 instances.
- A launch configuration is an instance configuration template.
- Dynamic scaling uses Amazon EC2 Auto Scaling, CloudWatch, and Elastic Load Balancing.
- AWS Auto Scaling is a separate service from Amazon EC2 Auto Scaling.

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

Some key takeaways from this section of the module include:

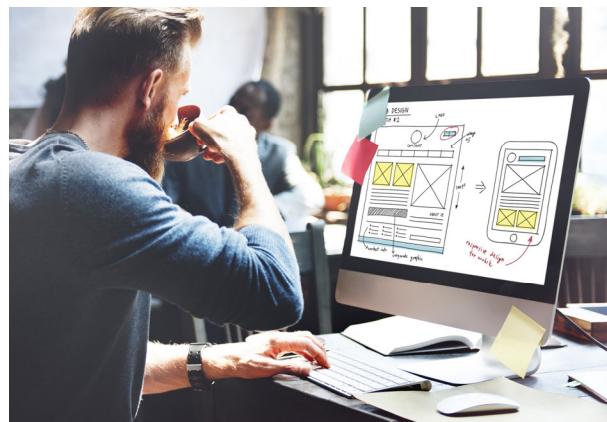
- Scaling enables you to respond quickly to changes in resource needs.
- Amazon EC2 Auto Scaling helps you maintain application availability, and enables you to automatically add or remove EC2 instances according to your workloads.
- An Auto Scaling group is a collection of EC2 instances.
- A launch configuration is an instance configuration template.
- You can implement dynamic scaling with Amazon EC2 Auto Scaling, Amazon CloudWatch, and Elastic Load Balancing.

AWS Auto Scaling is a separate service that monitors your applications, and it automatically adjusts capacity for the following resources:

- Amazon EC2 instances and Spot Fleets
- Amazon ECS tasks
- Amazon DynamoDB tables and indexes
- Amazon Aurora Replicas

## Lab 6: Scale and Load Balance Your Architecture

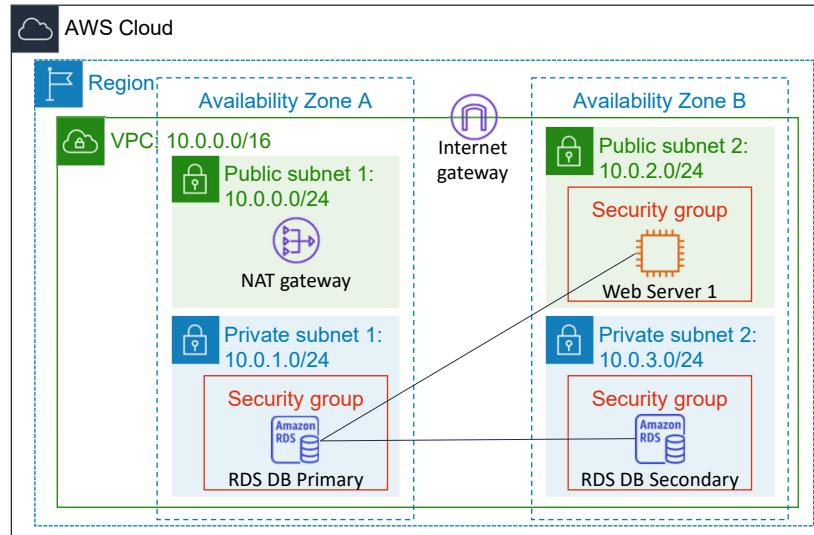
29



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

You will now complete Lab 6: Scale and Load Balance Your Architecture.

## Lab 6: Scenario



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

30

In this lab, you will use Elastic Load Balancing and Amazon EC2 Auto Scaling to load balance and scale your infrastructure. You will start with the given infrastructure.

## Lab 6: Tasks

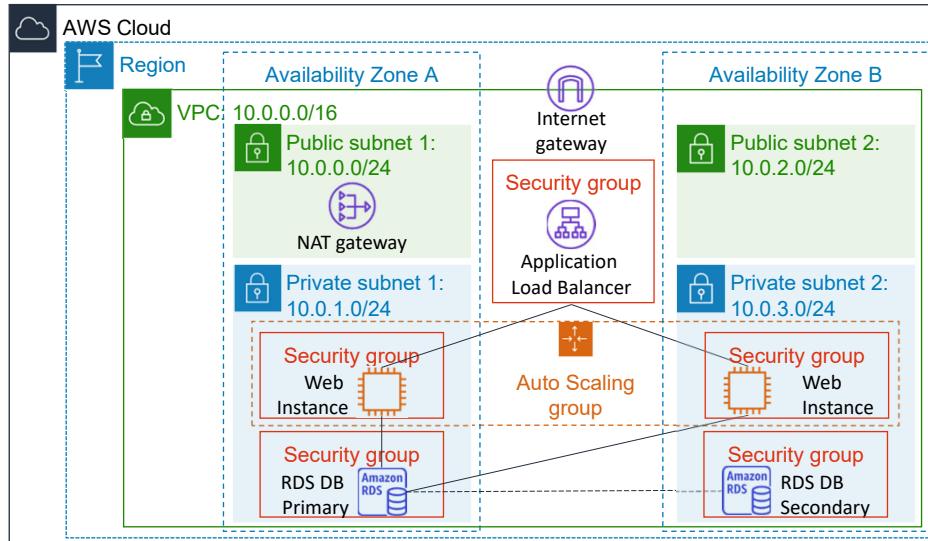


- Create an Amazon Machine Image (AMI) from a running instance.
- Create an Application Load Balancer.
- Create a launch configuration and an Auto Scaling group.
- Automatically scale new instances within a private subnet.
- Create Amazon CloudWatch alarms and monitor performance of your infrastructure.

In this lab, you will complete the following tasks:

- Create an Amazon Machine Image (AMI) from a running instance.
- Create an Application Load Balancer.
- Create a launch configuration and an Auto Scaling group.
- Automatically scale new instances within a private subnet.
- Create Amazon CloudWatch alarms and monitor performance of your infrastructure.

# Lab 6: Final product



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

32

The diagram summarizes what you will have built after you complete the lab.



~ 30 minutes



## Begin Lab 6: Scale and Load Balance Your Architecture

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

33

It is now time to start the lab. It should take you approximately 30 minutes to complete the lab.



## Lab debrief: Key takeaways



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

34

### In this lab, you:

- Created an Amazon Machine Image (AMI) from a running instance.
- Created a load balancer.
- Created a launch configuration and an Auto Scaling group.
- Automatically scaled new instances within a private subnet.
- Created Amazon CloudWatch alarms and monitored performance of your infrastructure.

**Module 10: Automatic Scaling and Monitoring**

## Module wrap-up

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



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:

- Indicate how to distribute traffic across Amazon Elastic Compute Cloud (Amazon EC2) instances using Elastic Load Balancing.
- Identify how Amazon CloudWatch enables you to monitor AWS resources and applications in real time.
- Explain how Amazon EC2 Auto Scaling launches and releases servers in response to workload changes.
- Perform scaling and load balancing tasks to improve an architecture.

In summary, in this module you learned how to:

- Indicate how to distribute traffic across Amazon Elastic Compute Cloud (Amazon EC2) instances using Elastic Load Balancing.
- Identify how Amazon CloudWatch enables you to monitor AWS resources and applications in real time
- Explain how Amazon EC2 Auto Scaling launches and releases servers in response to workload changes.
- Perform scaling and load balancing tasks to improve an architecture.

## Complete the knowledge check



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

37

Complete the knowledge check for this module.

## Sample exam question



Which service would you use to send alerts based on Amazon CloudWatch alarms?

- A. Amazon Simple Notification Service
- B. AWS CloudTrail
- C. AWS Trusted Advisor
- D. Amazon Route 53

Look at the answer choices and rule them out based on the keywords that were previously highlighted.

# Thank you

© 2019 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections or feedback on the course, please email us at: [aws-course-feedback@amazon.com](mailto:aws-course-feedback@amazon.com). For all other questions, contact us at: <https://aws.amazon.com/contact-us/aws-training/>. All trademarks are the property of their owners.



Thanks for participating!