AWS

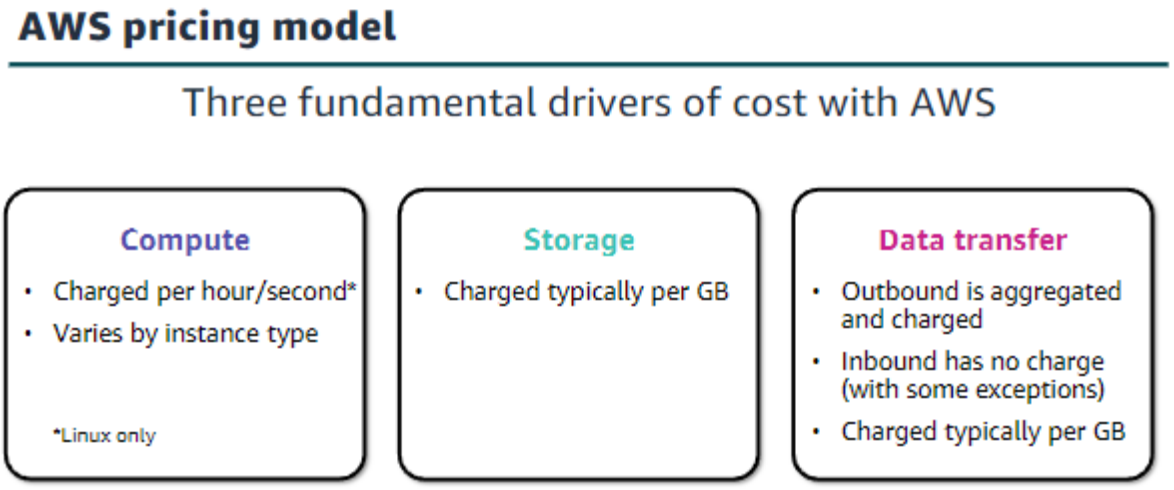
AWS

Created	@October 9, 2023 1:10 PM
Reviewed	

Amazon Web Services

Module 2: *Cloud Economics and Billing*

Section 1: Fundamentals of Pricing



There are three fundamental drivers of cost with AWS: **compute, storage, and outbound data transfer**.

There is no charge for inbound data transfer or for data transfer between other AWS services within the same AWS Region.

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.

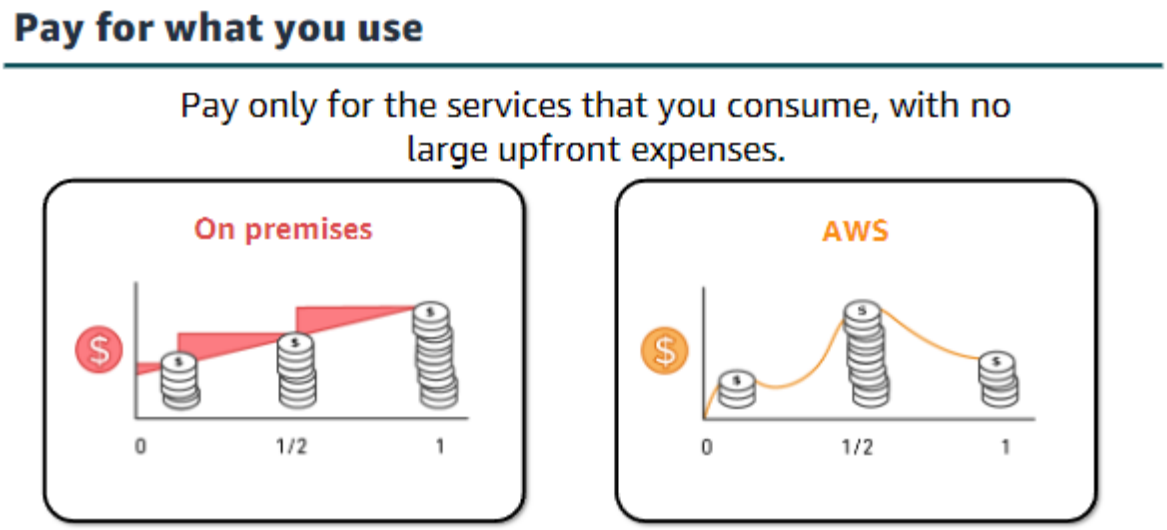
AWS offers a range of cloud computing services. For each service, you pay for exactly the amount of resources that you actually need. This utility-style pricing model includes:

•Pay for what you use

•Pay less when you reserve

•Pay less when you use more

•Pay even less as AWS grows



For certain services like AmazonElastic 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)

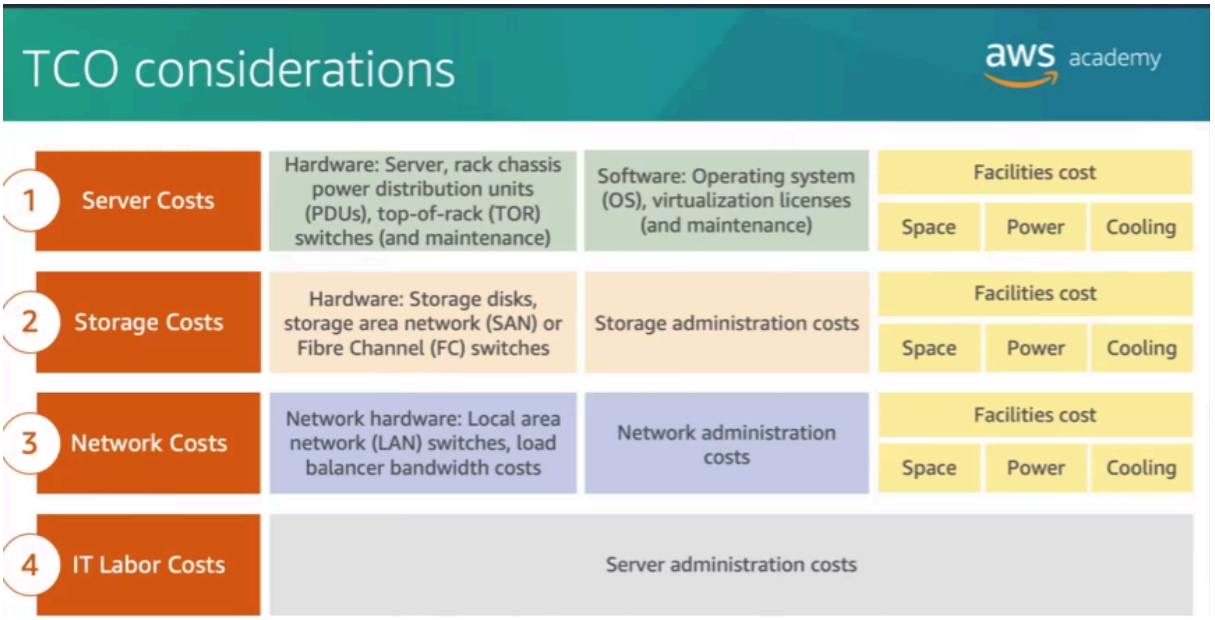
Section 2: Total Cost of Ownership

1. On-premises infrastructure is installed on company's own computers and servers.
2. Fixed costs = CapEx

3. CapEx includes facilities, hardware, licenses and maintenance staff. Scaling up can be expensive and time consuming. Scaling down does not reduce fixed costs.
4. Cloud Infra is purchased from a service provider who builds and maintains the facilities.
5. Pay for what you use.
6. Scaling up or down is simple.
7. Costs are easy to estimate.
8. How to choose best option?

   Compare on premises solution to cloud solution.
9. TCO is a financial estimate intended to help buyers and owners determine the direct and indirect cost of a product or system. Includes cost of the service + cost of owning the service.
10. Why use TCO?

    To compare cost of running and entire infrastructure environment on-premises or AWS. To budget and build the business case for moving to the cloud.

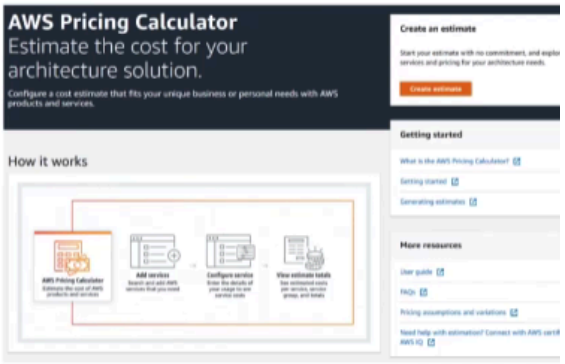


1. Cloud providers give transparent pricing based on different using method.

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



Access the AWS Pricing Calculator

Additional benefit considerations

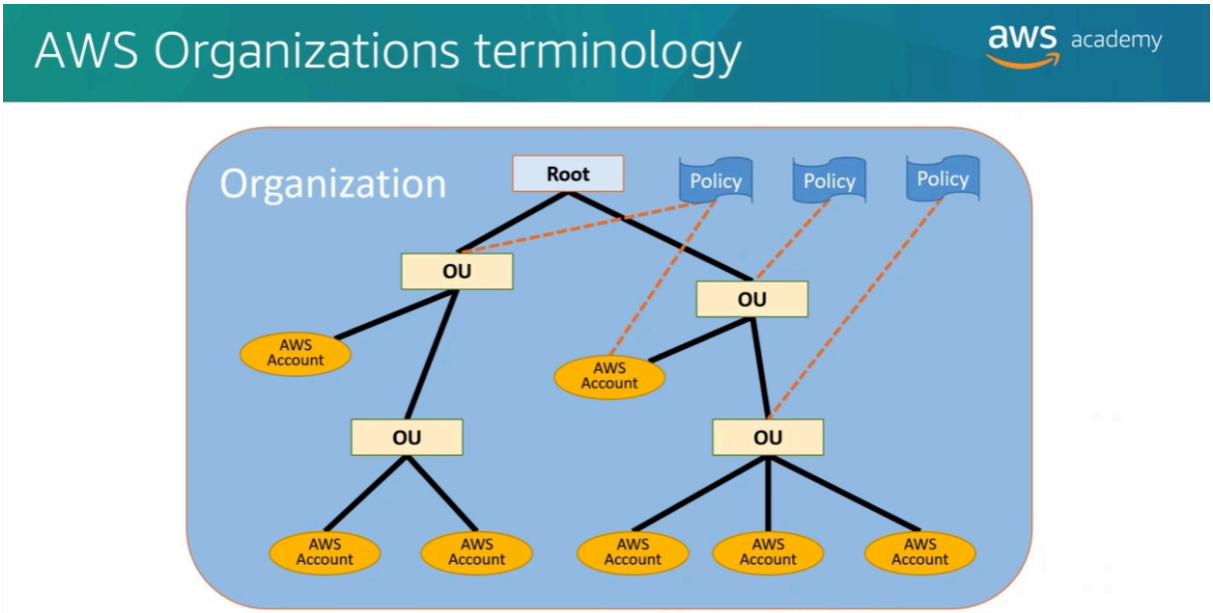
Hard benefits

- 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

Section 3: AWS Organizations



Section 4: AWS Billing and Cost Management

Section 5: AWS Billing Dashboard

Section 6: AWS Technical Support

For Account Assistance, the Support Concierge is a billing and account expert providing quick and efficient analysis. Addresses all non-tech billing and account level inquiries.

AWS Trusted Advisor is an automated service that acts like a customized cloud expert. Checks for opportunities to reduce expense and increase productivity.

Trusted Account Manager[TAM] are your primary point of contact. Only available via the Enterprise Support Plan.

AWS Support Plans

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

Case severity and response times

aws academy

	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

Student Guide [Notes] :

1. There are three fundamental drivers of cost with AWS: ***compute, storage,*** and ***outbound data transfer***.
2. There is no charge for inbound data transfer or for data transfer between other AWS services within the same AWS Region.
3. 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.

4. 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

AWS_Pricing_Overview.pdf

1. Pay for what you use:

    All AWS services are available on demand, require no long-term contracts, and have no complex licensing dependencies.

2. Pay less when you reserve:

    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)

3. Pay less by using more:

    You can get volume-based discounts and realize important savings as your usage increases. Amazon Simple Storage Service (Amazon S3), ***pricing is tiered, which means that you pay less per GB when you use more.***

    Data transfer in is always free.

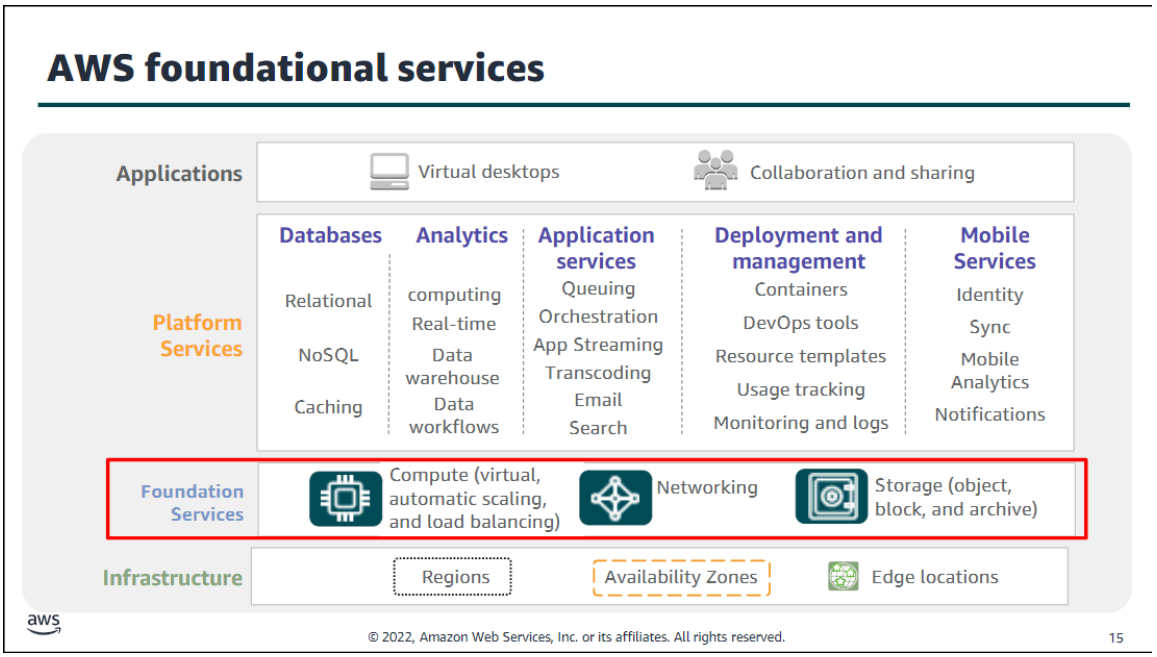
    Multiple storage services deliver lower storage costs based on your needs.

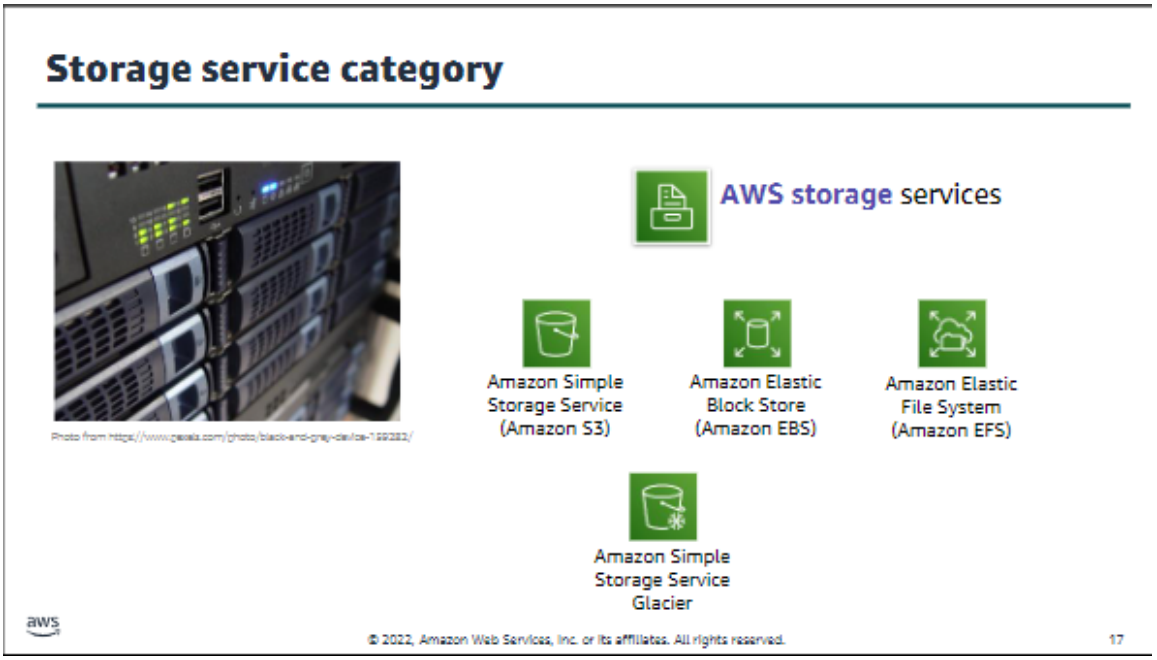
    AWS usage needs increase, you benefit from the economies of scale that enable you to increase adoption and keep costs under control.

Module 3: AWS Global Infrastructure Overview

Section 1: AWS Global Infrastructure

Section 2: AWS Services and Service Categories Overview



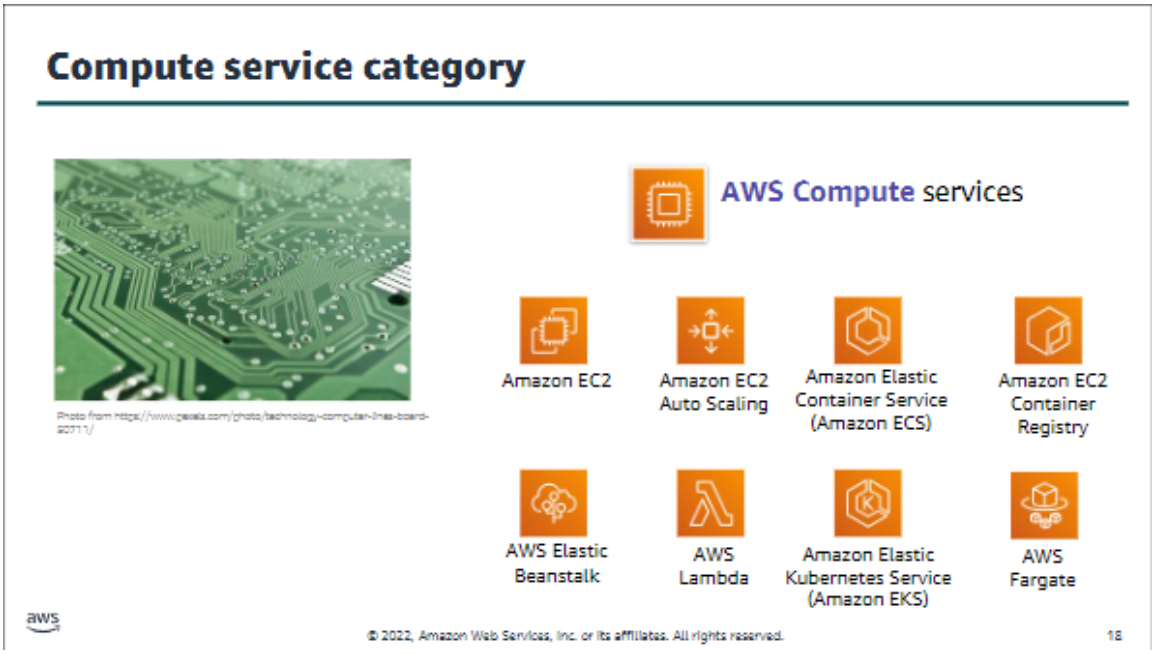


Amazon Simple Storage Service (Amazon S3) is an object storage service that offers scalability, data availability, security, and performance. Use it to store and protect any amount of data for websites, mobile apps, backup and restore, archive, enterprise applications, Internet of Things (IoT) devices,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



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.

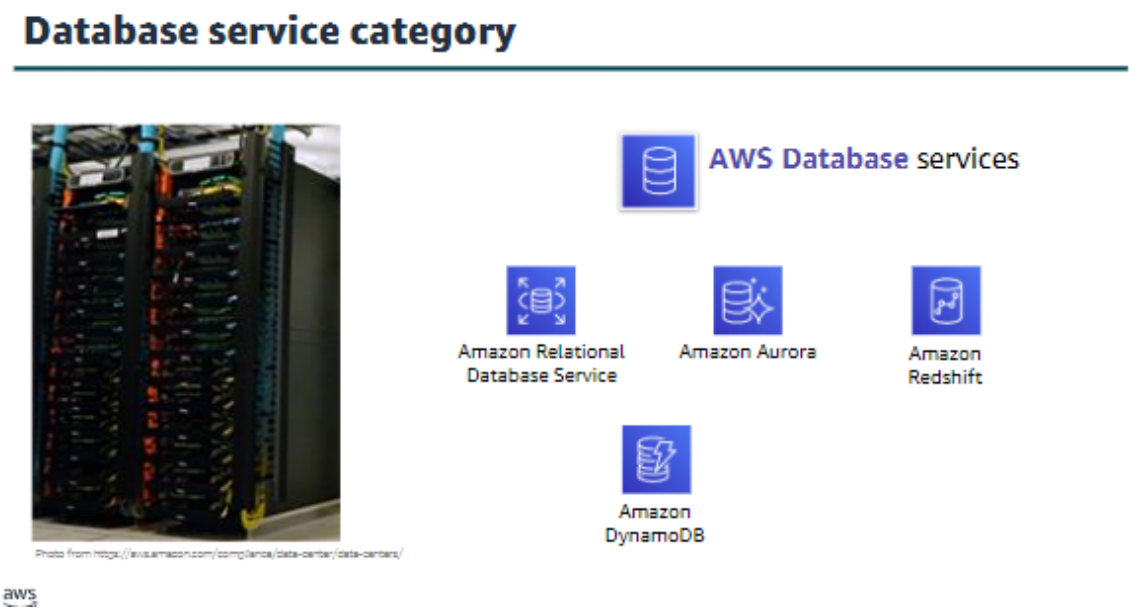
Amazon Elastic Container Registry (Amazon ECR) is a fully-managed Docker container registry that makes it easy for developers to store, manage, and deploy Docker container images.

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.

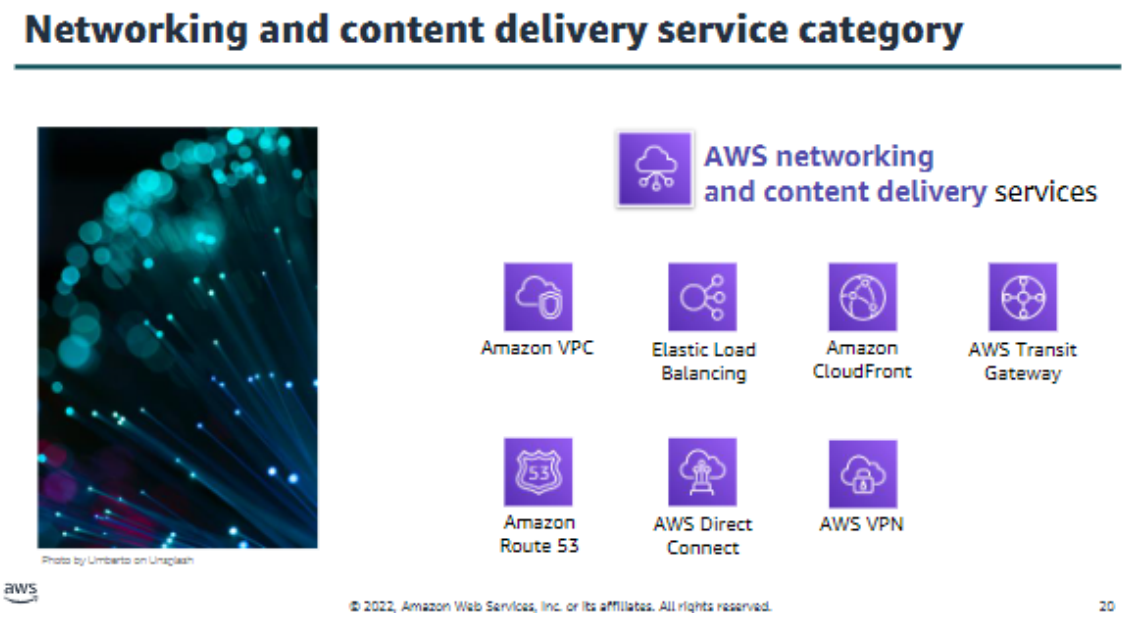


Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching, and backups.

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



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.

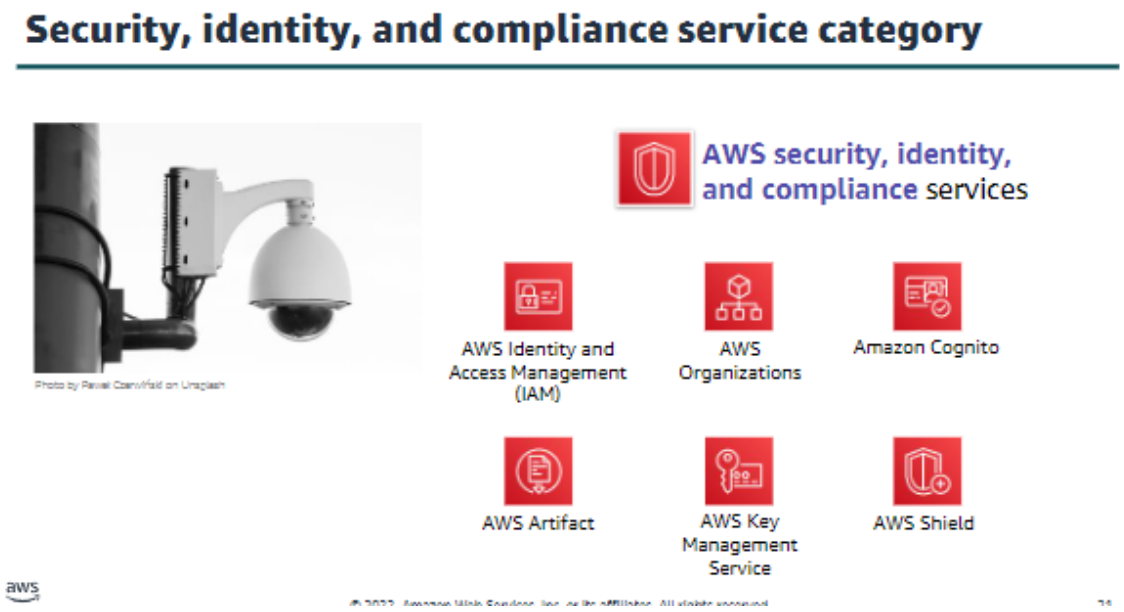
AWS Transit Gateway is a service that enables customers to connect their Amazon Virtual Private Clouds (VPCs) and their on-premises networks to a

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.



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.

A**WS 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

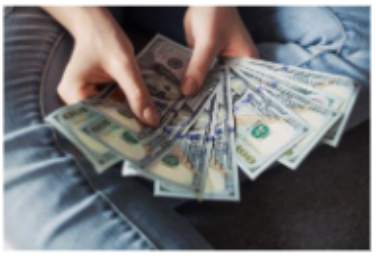
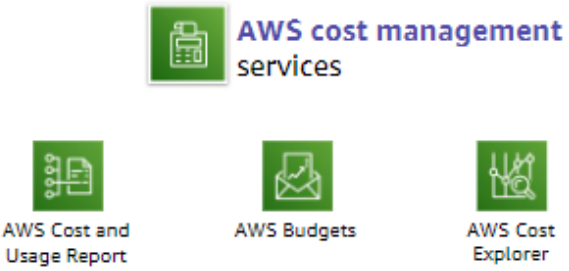


Photo by Alexander Mils on Unsplash



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 Maria Branco from Pexels

aws

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.

Hands on Activity

Question #1: Under which service category does the IAM service appear?

Answer: Security, Identity, & Compliance.

Question #2: Under which service category does the AmazonVPCservice 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?

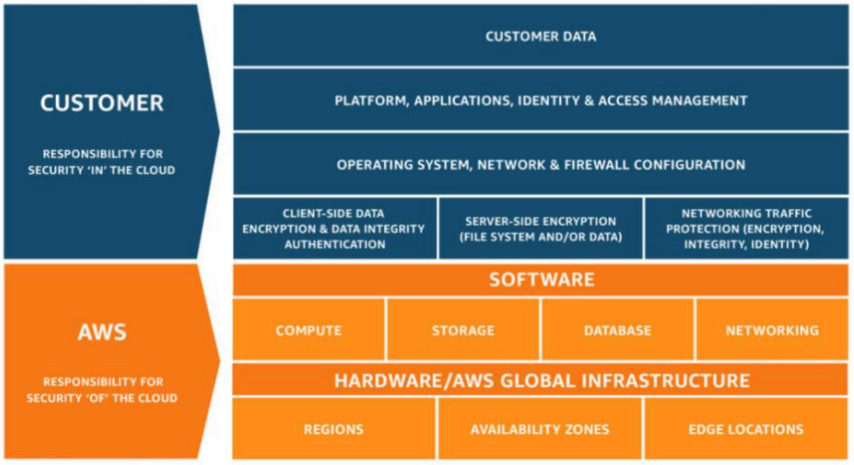
Answer: VPCs exist at the Region level.

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.

Module 4: AWS Cloud Security

AWS shared responsibility model



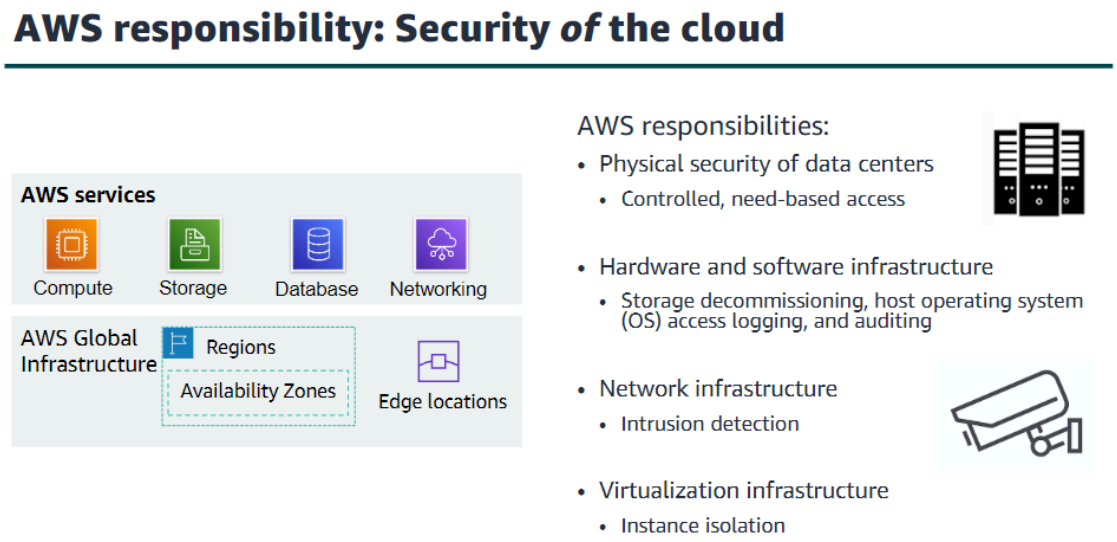
AWS operates, manages, and controls the components from the software virtualization layer down to the physical security of the facilities where AWS services operate. AWS is responsible

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).

Section 1: Shared Responsibility Model

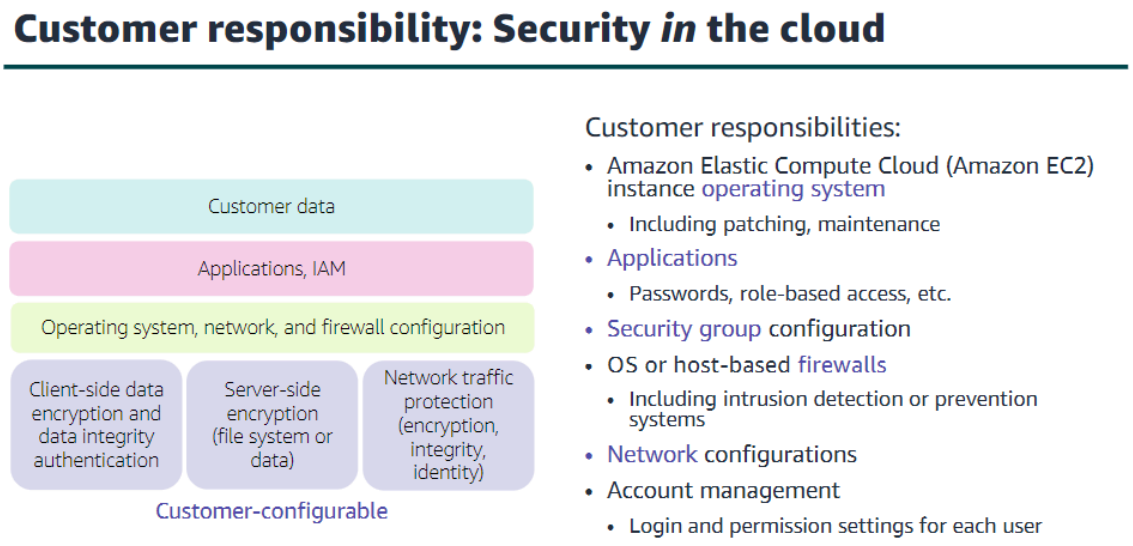


AWS is responsible for security of the cloud. But what does that mean? It means that AWS is responsible for protecting the global infrastructure that runs all the services that are offered in the AWS Cloud. The global infrastructure includes AWS Regions, Availability Zones, and edge locations.

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.



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.

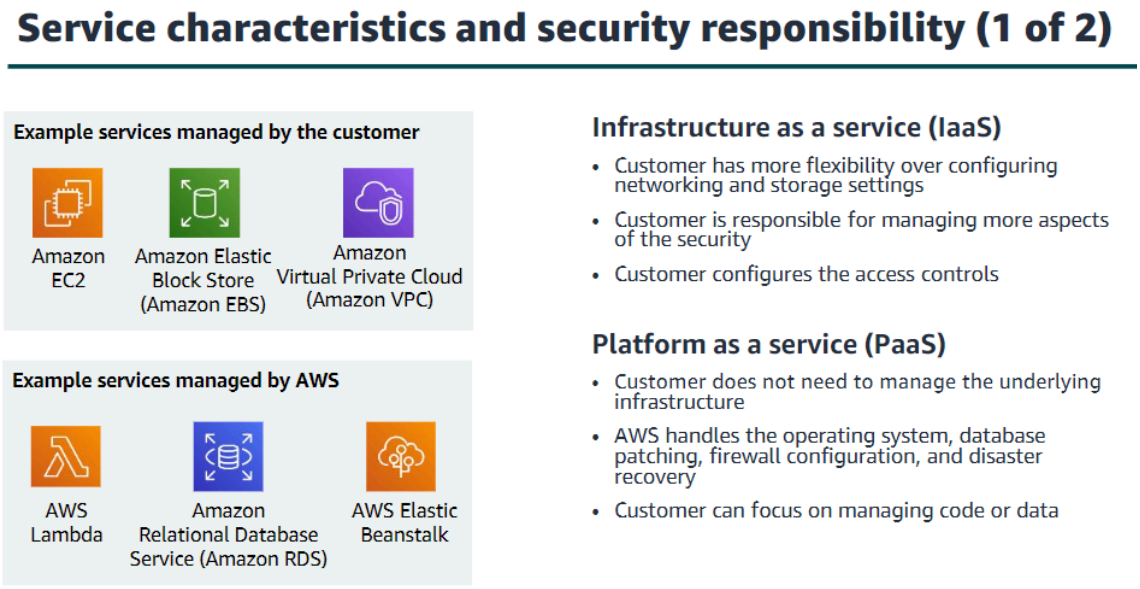
Customer responsibilities include selecting and securing any instance operating systems, securing the applications that are launched on AWS resources, security group configurations, firewall configurations, network configurations, and secure account management.

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.



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.

Example: 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.

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.

Example: 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.

Service characteristics and security responsibility (2 of 2)

Software as a service (SaaS)

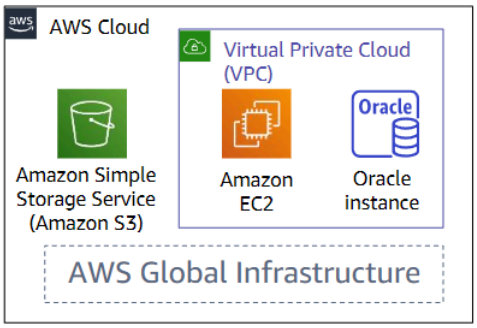
- Software is centrally hosted
- Licensed on a subscription model or pay-as-you-go basis.
- Services are typically accessed via web browser, mobile app, or application programming interface (API)
- Customers do not need to manage the infrastructure that supports the service

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. Examples:

- **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.
- **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.
- **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: Scenario 1 of 2 Answers

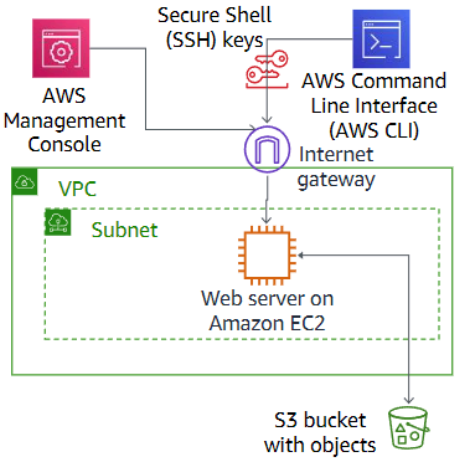
Consider this deployment. Who is responsible – AWS or the customer?



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

Activity: Scenario 2 of 2 Answers

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

Section 2: AWS IAM- Identity and Access Management

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)

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.

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. With IAM, you can manage **which** resources can be accessed by **who**, and **how** these resource scan be accessed.

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

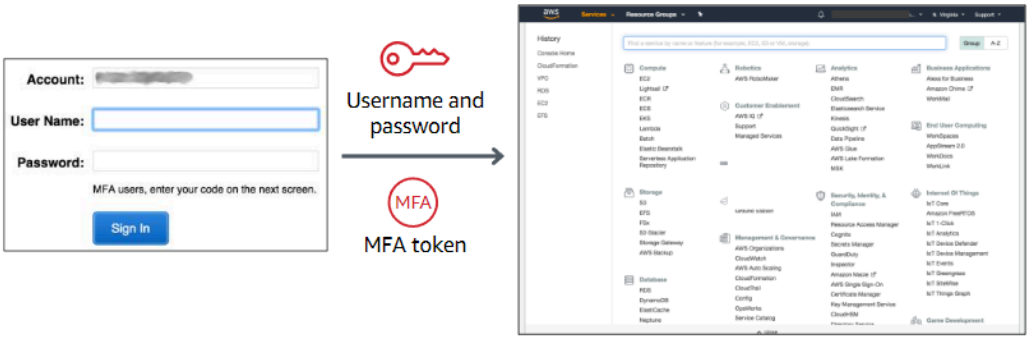
- Authenticate using:
  - 12-digit Account ID *or* alias
  - IAM user name
  - IAM password
- If enabled, **multi-factor authentication (MFA)** prompts for an authentication code.

AWS Management Console

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

IAM MFA

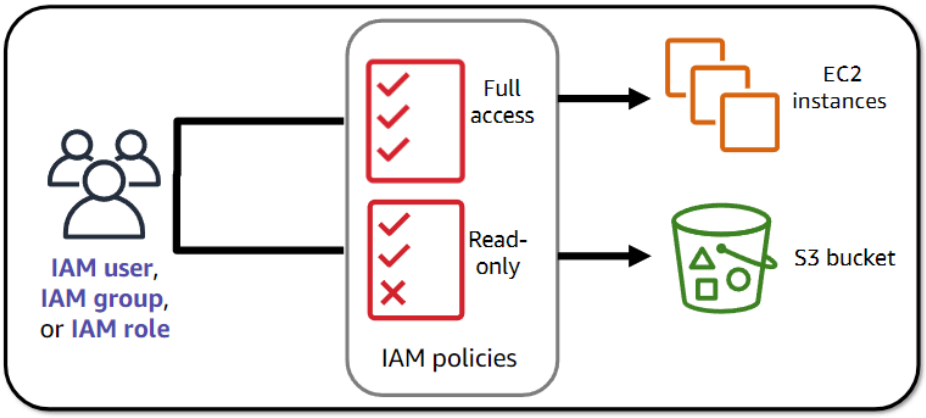
- MFA provides increased security.
- In addition to **username** and **password**, MFA requires a unique authentication code to access AWS services.



AWS Management Console

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.

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**.

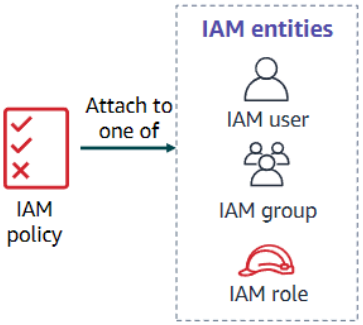
IAM permissions

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

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.

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)



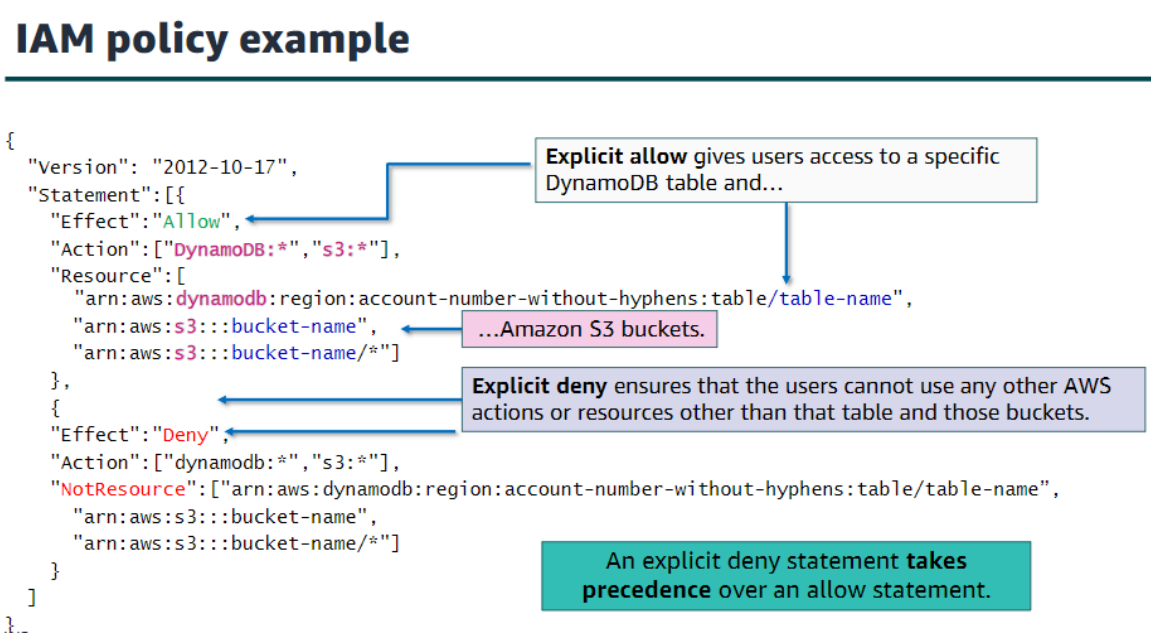
There are two types of IAM policies,

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:

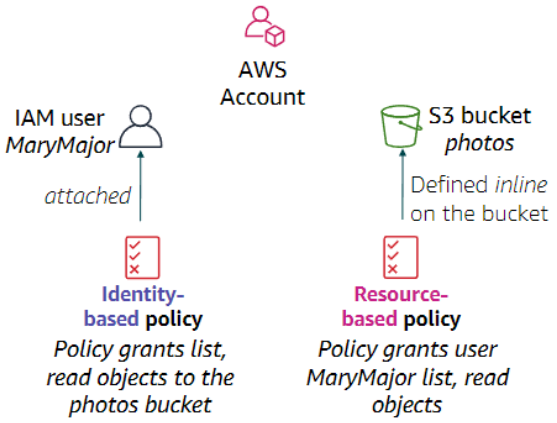
- **Managed policies** –Standalone identity-based policies that you can attach to multiple users, groups, and roles in your AWS account
- **Inline policies** –Policies that you create and manage, and that are embedded directly into a single user group or role.

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.



Resource-based policies

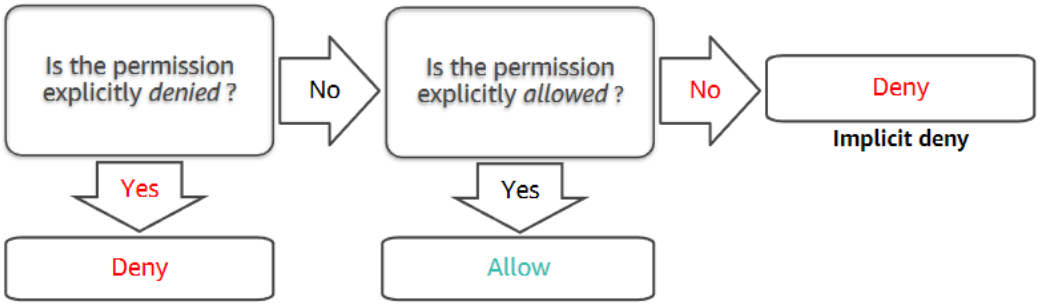
- *Identity-based policies* are attached to a user, group, or role
- **Resource-based policies** are attached to a resource (*not* to a user, group or role)
- Characteristics of resource-based policies –
  - Specifies who has access to the resource and what actions they can perform on it
  - The policies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



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.

IAM permissions

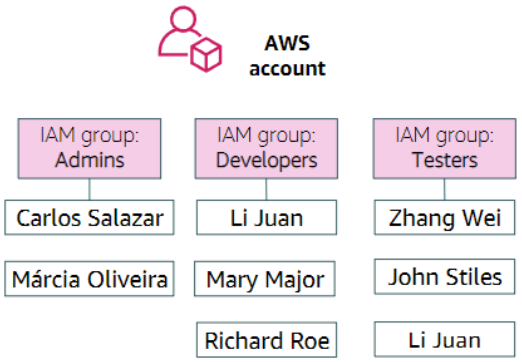
How IAM determines permissions:



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***.

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



IAM roles

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

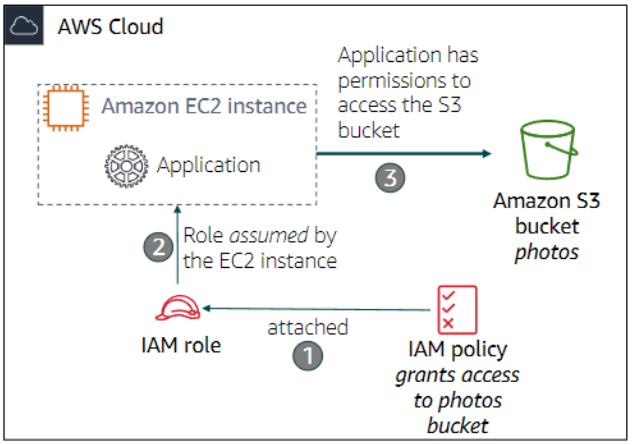
Example use of an IAM role

Scenario:

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

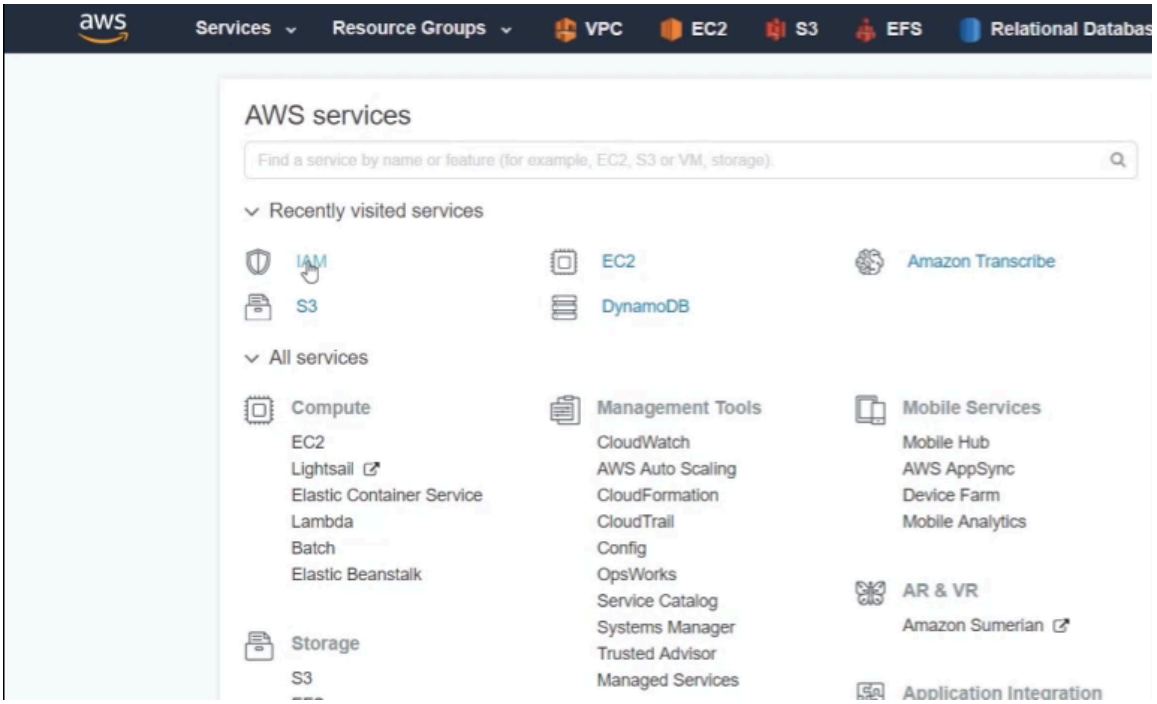
Solution:

- 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

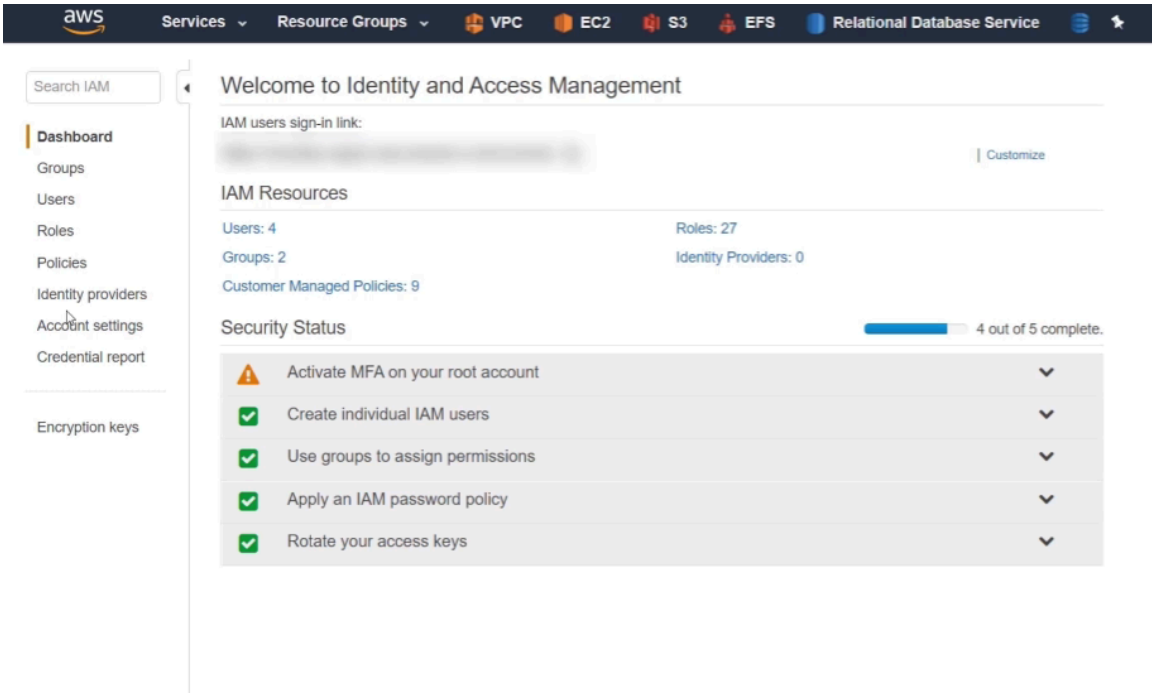


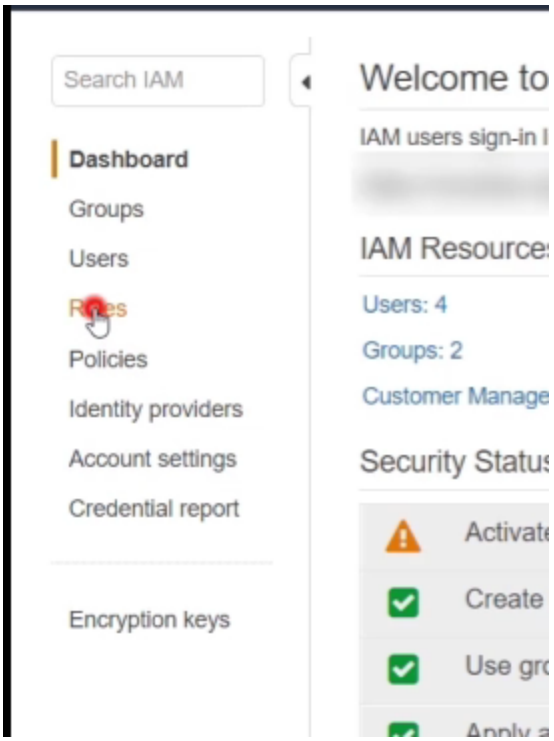
Section 3: Identity and Access Management Demonstration Module

Step 1: Go to AWS services dashboard and click IAM

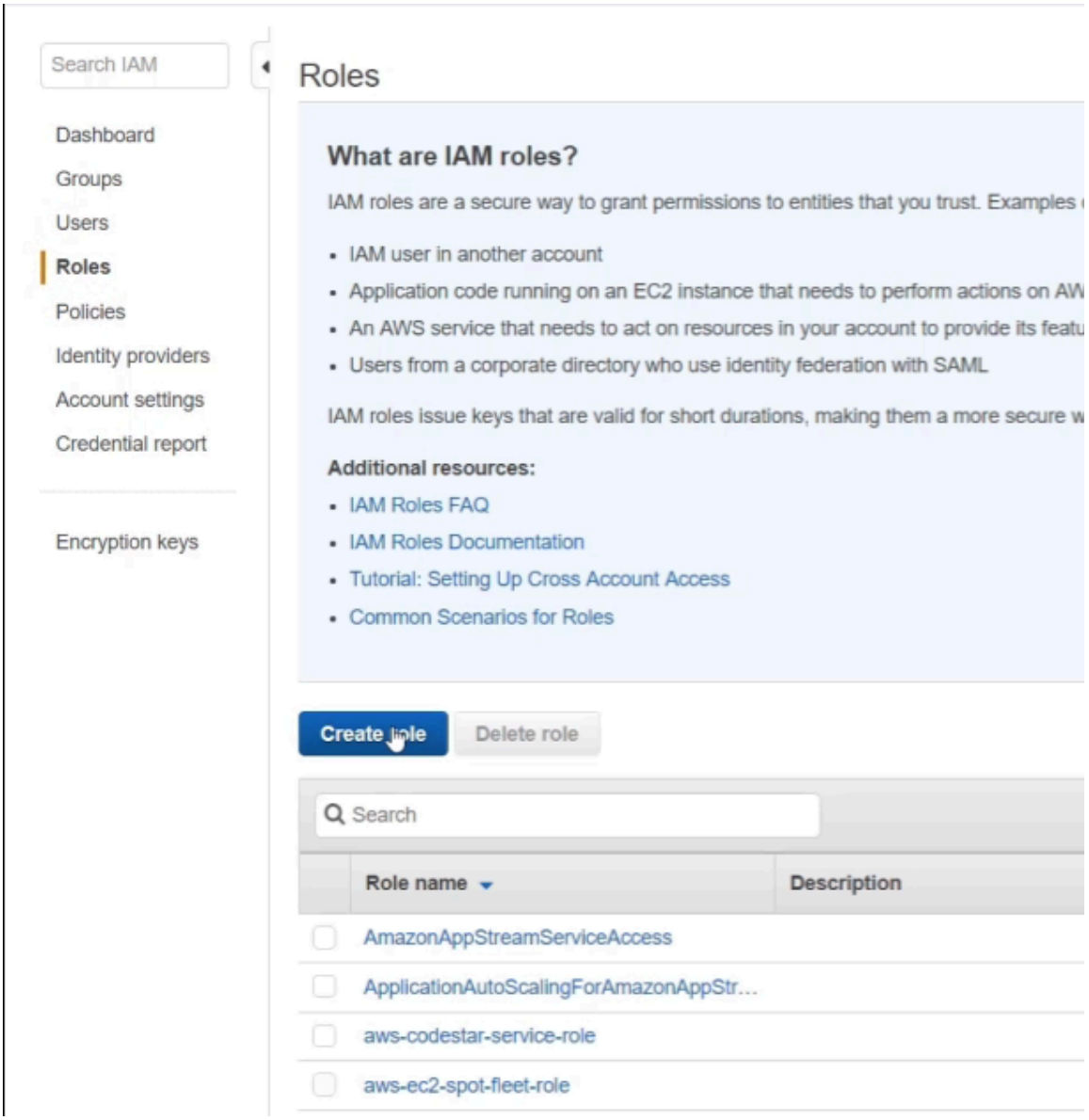


Step 2: To config a Role

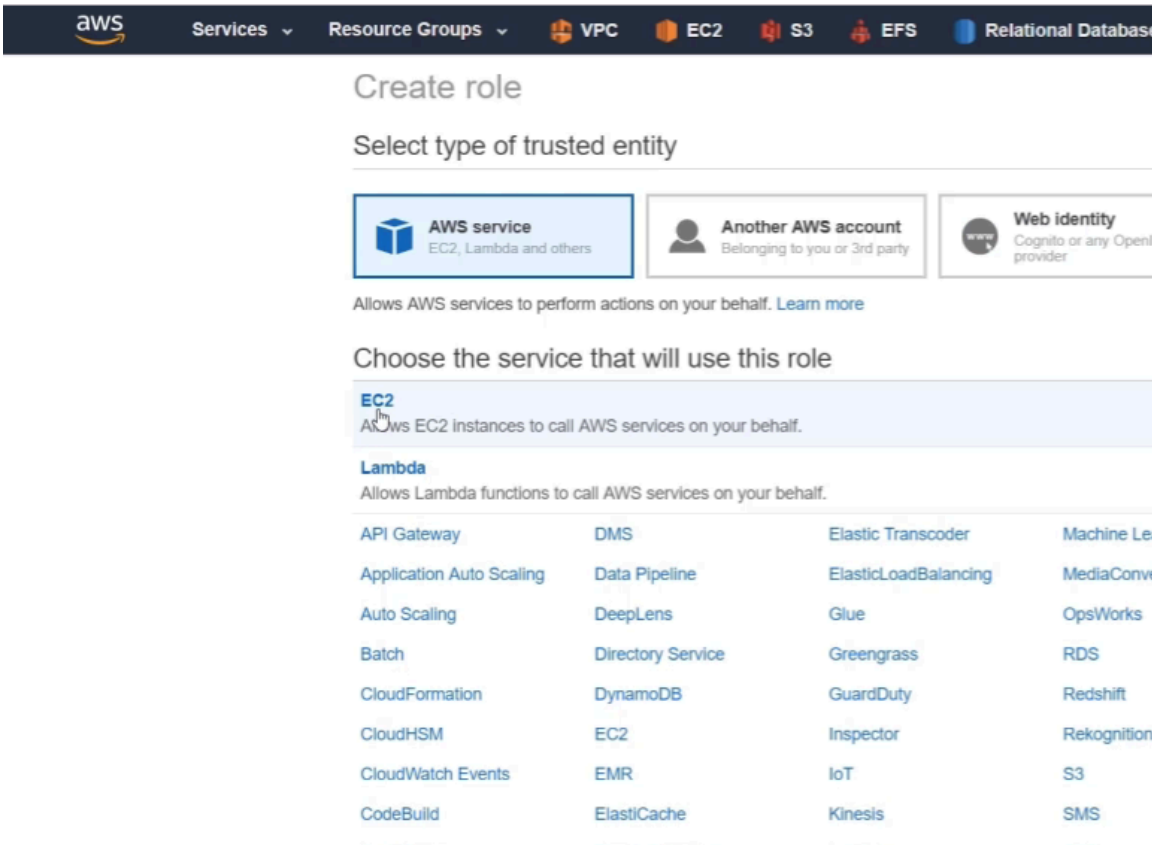




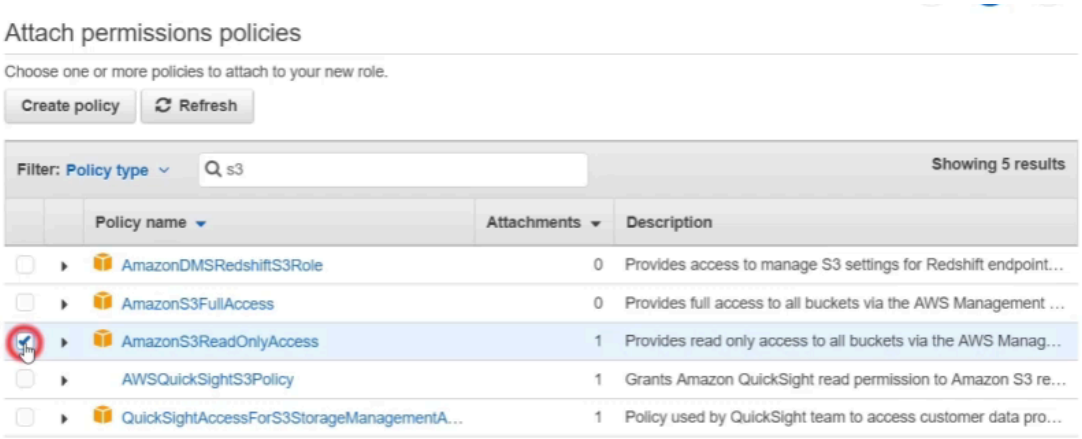
Step 3: Create a Role

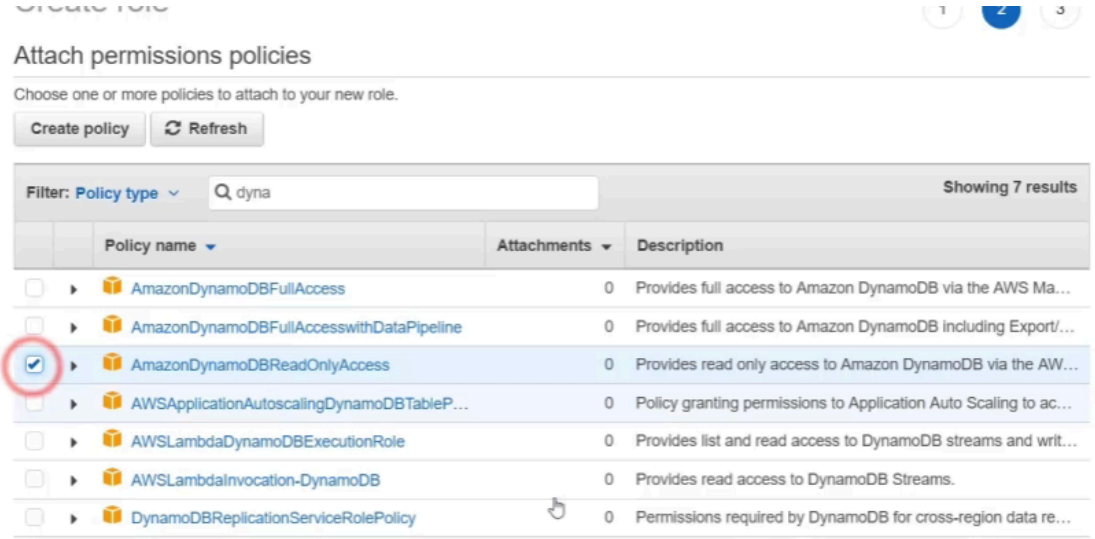


Step 4: For this instance we are choosing EC2

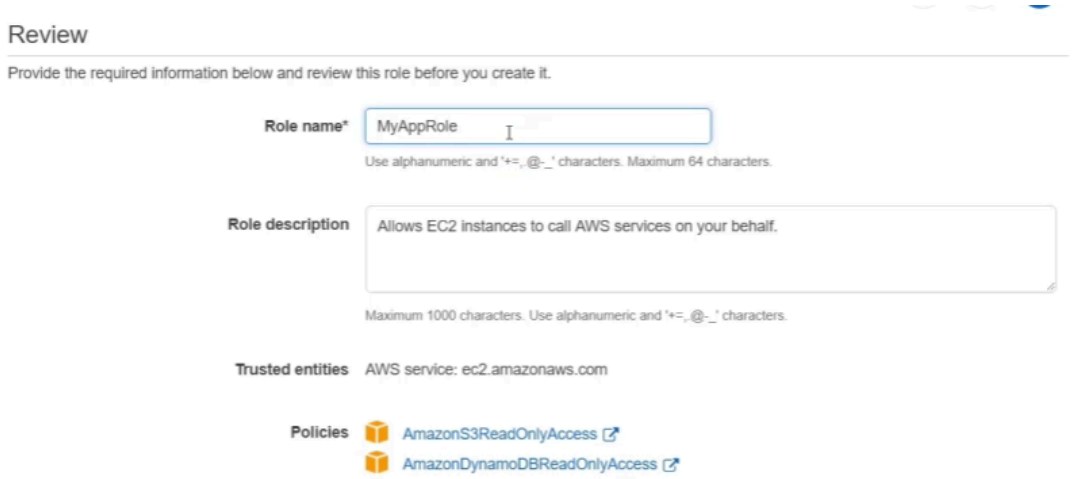


Step 5: To access S3 buckets. Click on next and set permissions to,

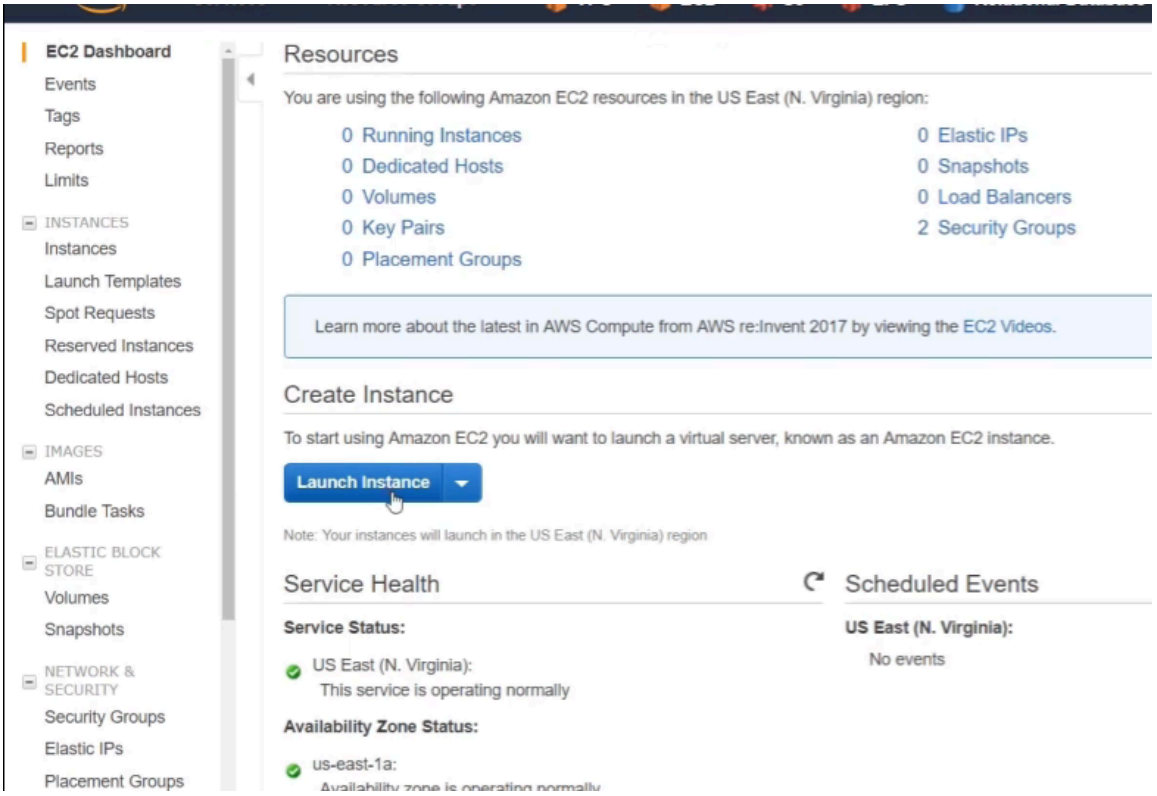




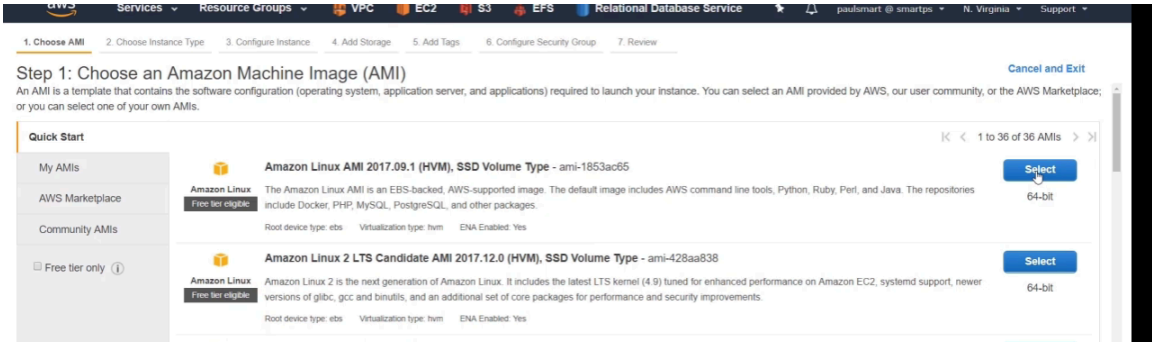
Step 6: Click on Review and give your role a name.

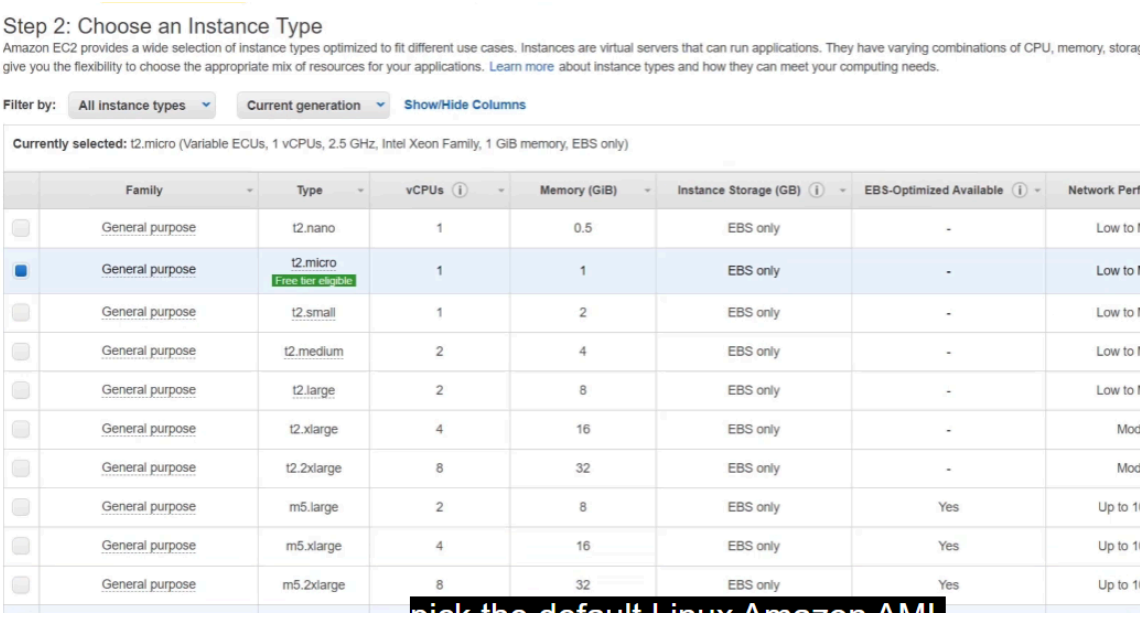


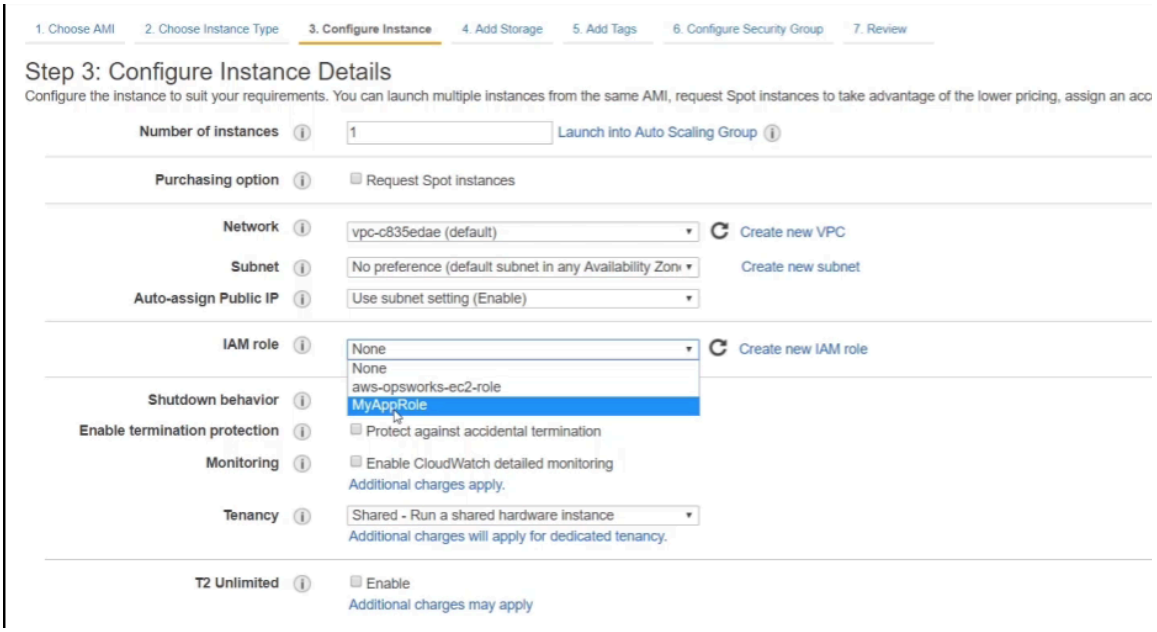
Step 7: Go to EC2 launch an instance



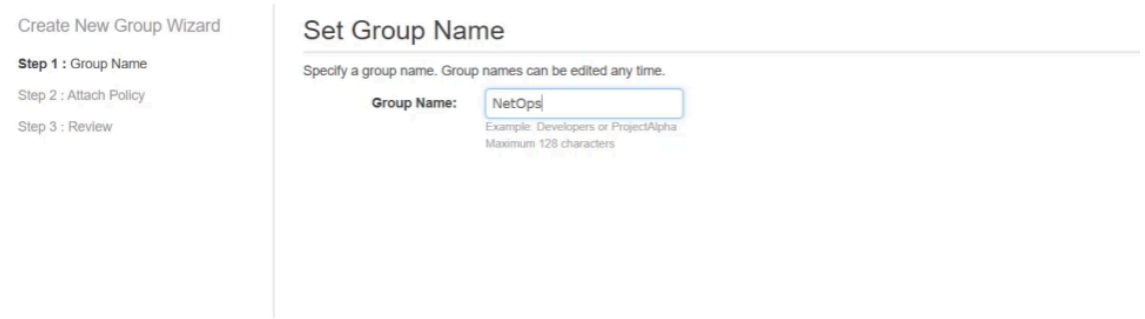
Now pick the default Linux AMI



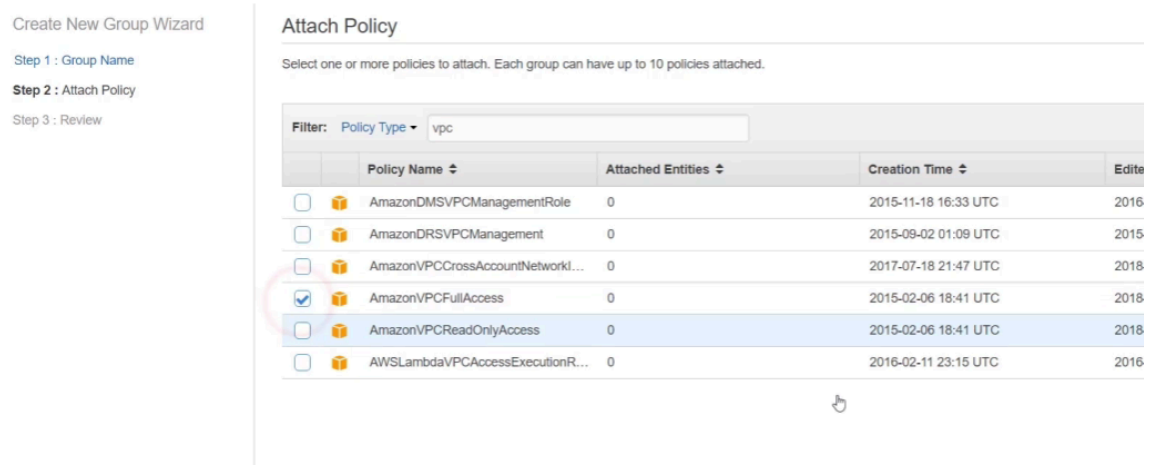




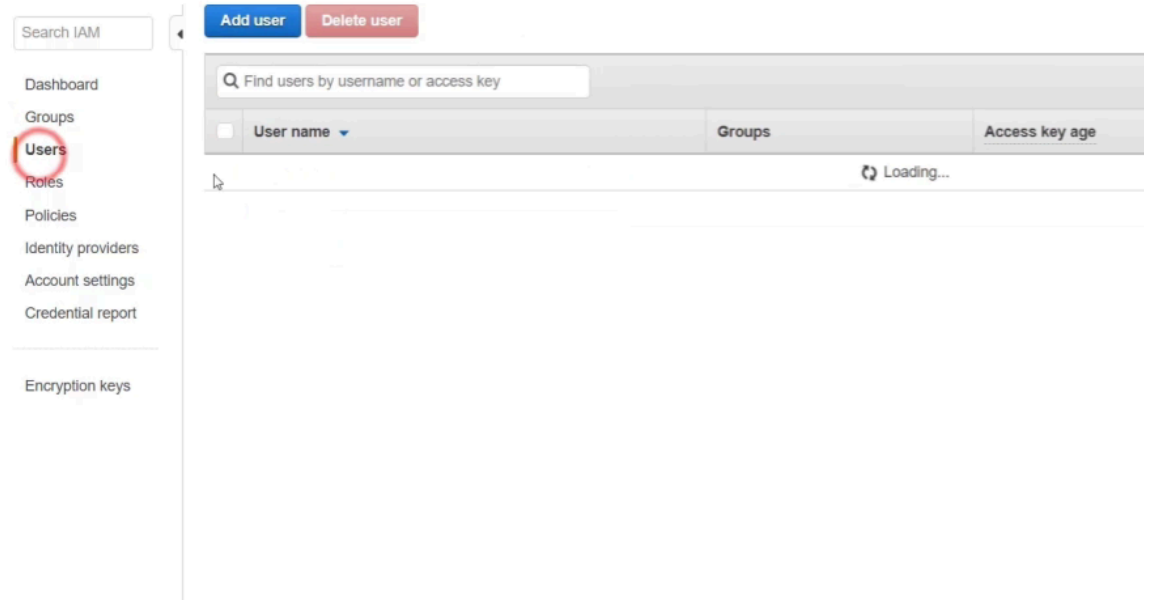
Now to create Groups, go to IAM, Create Group option,



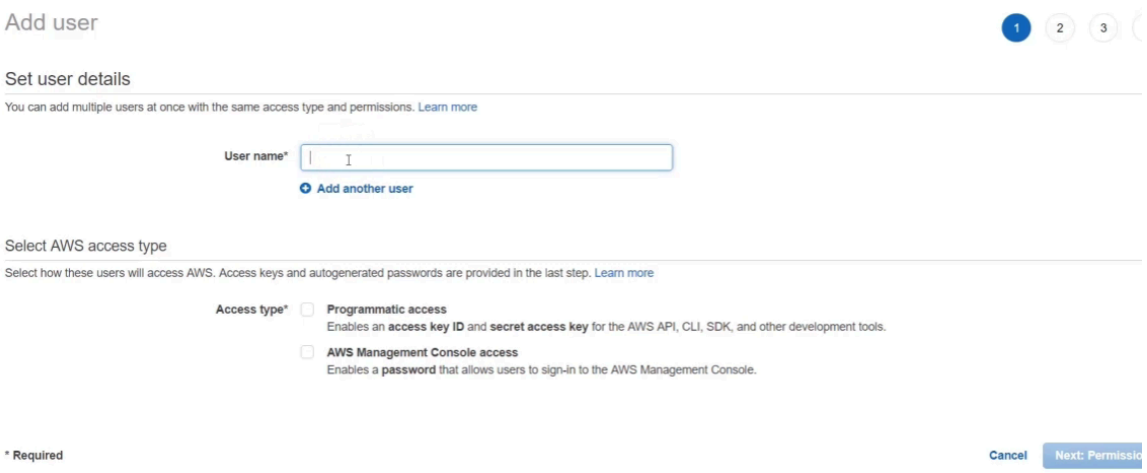
Attach a policy,

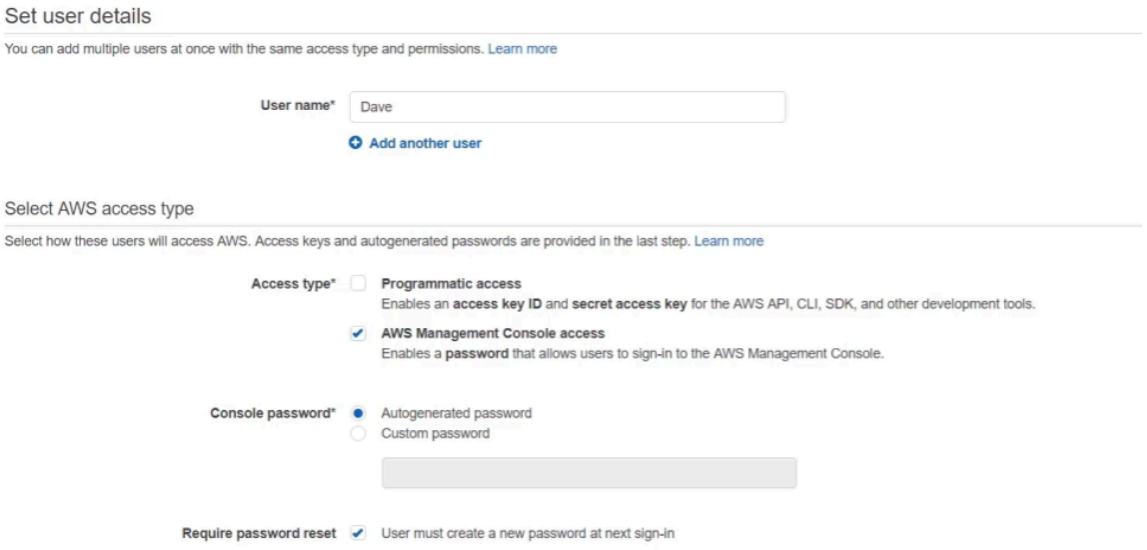


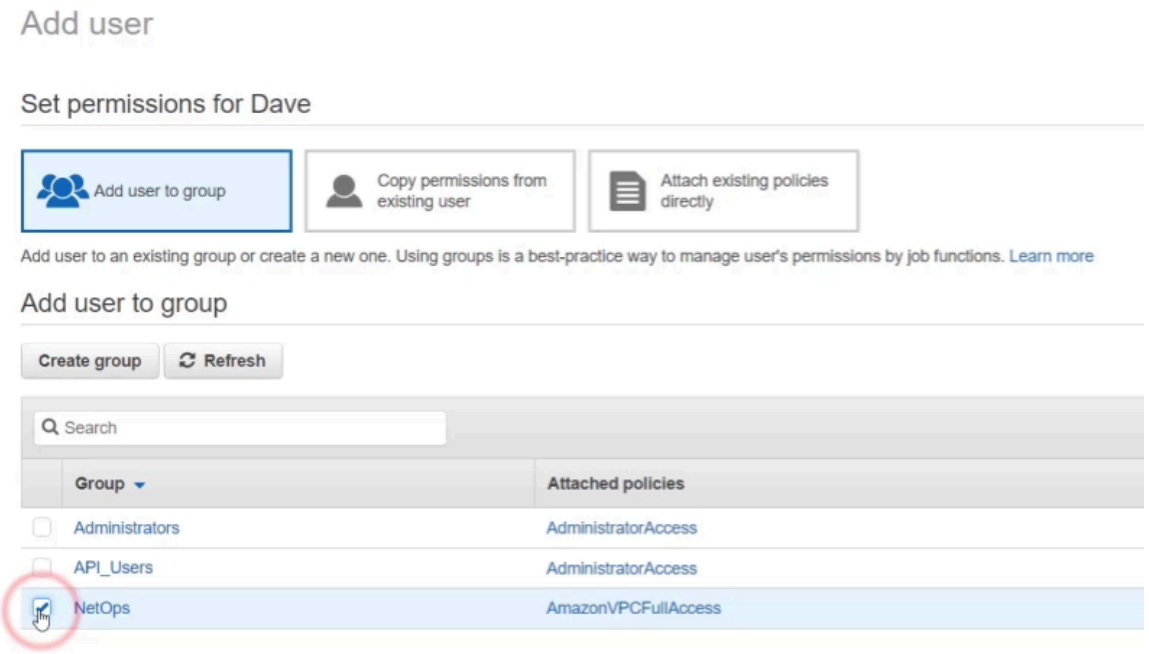
To create user,

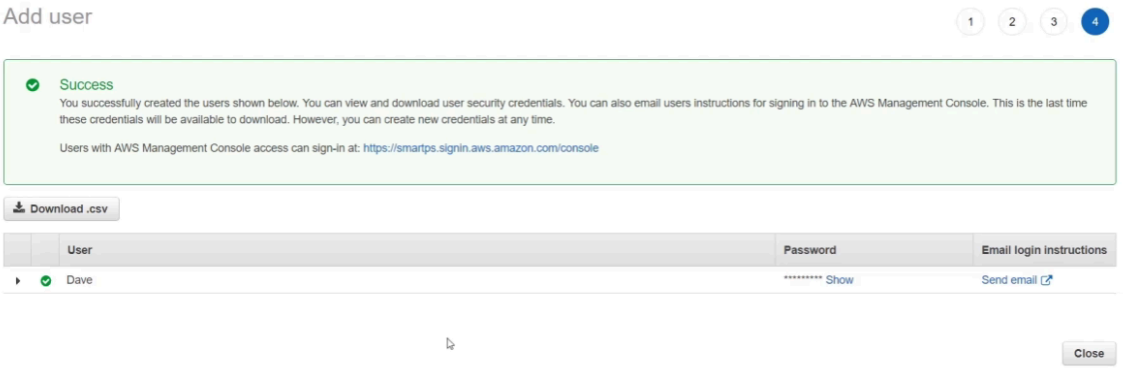


Add a user,


Add user
1
2
3
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.
* Required
Cancel
Next: Permissio

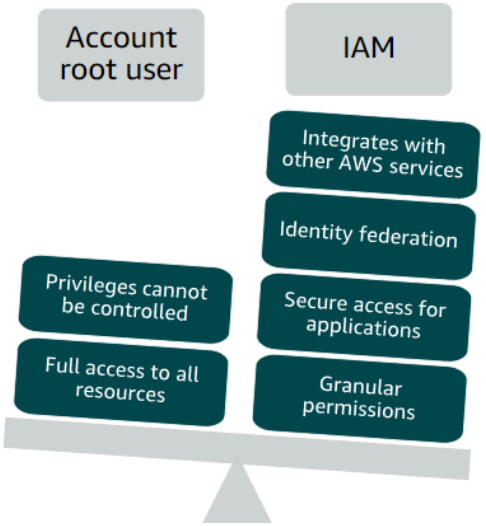

Set user details
You can add multiple users at once with the same access type and permissions. Learn more
User name*
Dave
Add another user
Select AWS access type
Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. Learn more
Access type*
Programmatic 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





Section 4: Securing new AWS Account

AWS account root user access versus IAM access



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

Securing a new AWS account: Account root user

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.

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.

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).

Securing a new AWS account: Billing reports

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

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

Securing AWS account

AWS Organizations

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

AWS Organizations: Service control policies

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

AWS Key Management Service (AWS KMS)

AWS Key Management Service (AWS KMS) features:

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

AWS Key Management Service (AWS KMS)

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.

Amazon Cognito

Amazon Cognito features:

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

Amazon Cognito 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.

AWS Shield

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

AWS Shield 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.

Section 5: Securing data on AWS

Encryption of data *at rest*

- **Encryption** encodes data with a secret key, which makes it unreadable
  - Only those who have the secret key can decode the data
  - AWS KMS can manage your secret keys

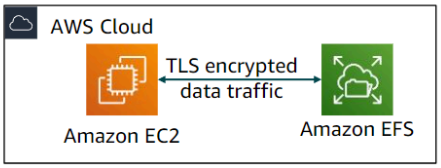
- AWS supports encryption of **data at rest**
  - Data at rest = Data stored physically (on disk or on tape)
  - You can encrypt data stored in any service that is supported by AWS KMS, including:
    - Amazon S3
    - Amazon EBS
    - Amazon Elastic File System (Amazon EFS)
    - Amazon RDS managed databases

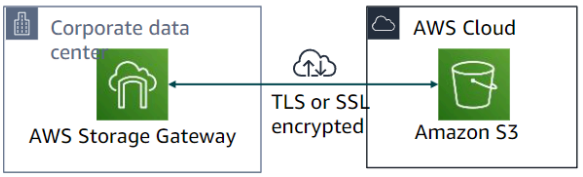
Data encryption is an essential tool to use when your objective is to protect digital data.

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

Encryption of data *in transit*

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





Data in transit refers to data that is moving across the network. Encryption of data in transit is accomplished by using Transport Layer Security (TLS)

1.2 with an open standard AES-256 cipher. TLS was formerly called Secure Sockets Layer (SSL)

AWS Certificate Manager is a service that enables you to provision, manage, and deploy SSL or TLS certificates for use with AWS services and your internal connected resources.

Securing Amazon S3 buckets and objects

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

- **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.
- **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.
- **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.

Section 6: Working to ensure compliance

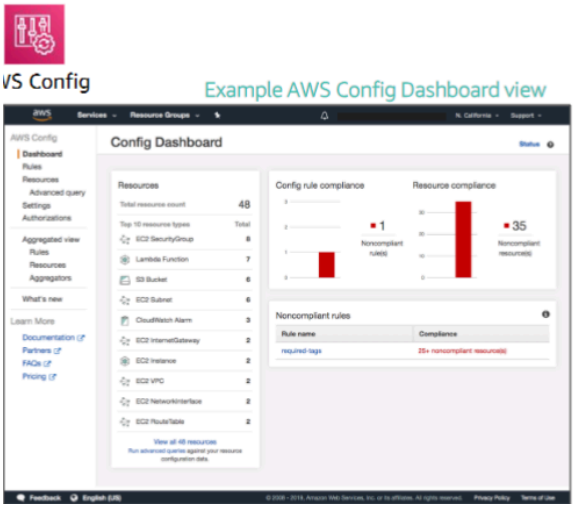
AWS compliance programs

- 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

As an example of a certificationfor which you can use AWS services to meet your compliance goals,consider the ISO/IEC 27001:2013certification. It specifies the requirements for establishing, implementing, maintaining,and continually improving an Information Security Management System.

AWS Config



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

AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. AWS Config continuously monitors and records your AWS resource configurations, and it enables you to automate the evaluation of recorded configurations against desired configurations.

This enables you to simplify compliance auditing, security analysis, change management, and operational troubleshooting. 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.**

AWS Artifact

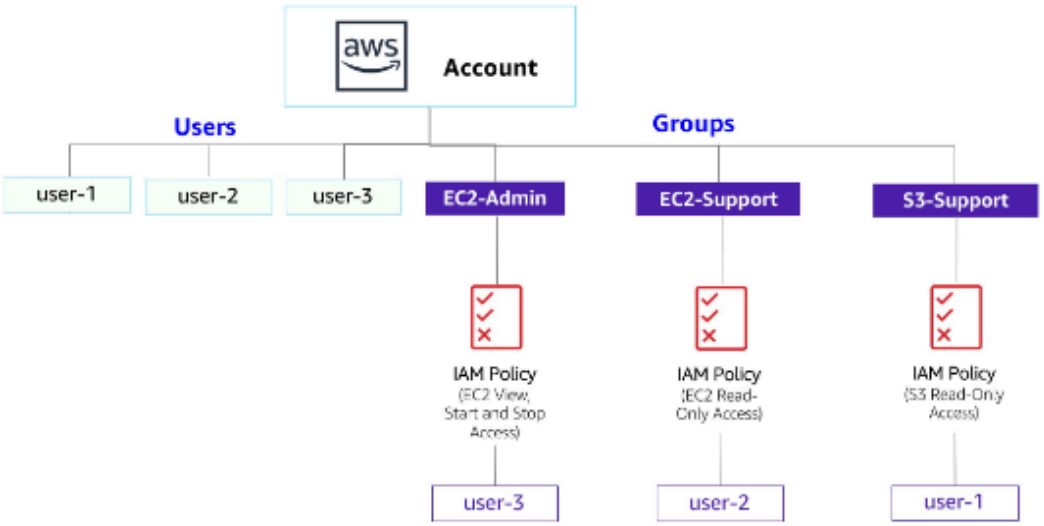
- **Is a resource for compliance-related information**
- Provide access to security and compliance reports, and select online agreements
- Can access example downloads:
  - AWS ISO certifications
  - Payment Card Industry (PCI) and Service Organization Control (SOC) reports
- Access AWS Artifact directly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

LAB ACTIVITY 1

Lab 1: Introduction to AWS IAM

AWS Identity and Access Management (IAM) is a web service that enables Amazon Web Services (AWS) customers to manage users and user permissions in AWS. With IAM, you can centrally manage **users**, **security credentials** such as access keys, and **permissions** that control which AWS resources users can access.

This lab will demonstrate:



- Exploring pre-created **IAM Users and Groups**
- Inspecting **IAM policies** as applied to the pre-created groups
- Following a **real-world scenario**, adding users to groups with specific capabilities enabled
- Locating and using the **IAM sign-in URL**

- **Experimenting** with the effects of policies on service access

Other AWS Services

During this lab, you may receive error messages when performing actions beyond the steps in this lab guide. These messages will not impact your ability to complete the lab.

AWS Identity and Access Management

AWS Identity and Access Management (IAM) can be used to:

- **Manage IAM Users and their access:** You can create Users and assign them individual security credentials (access keys, passwords, and multi-factor authentication devices). You can manage permissions to control which operations a User can perform.
- **Manage IAM Roles and their permissions:** An IAM Role is similar to a User, in that it is an AWS identity with permission policies that 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.
- **Manage federated users and their permissions:** You can enable *identity federation* to allow existing users in your enterprise to access the AWS Management Console, to call AWS APIs and to access resources, without the need to create an IAM User for each identity.

Accessing the AWS Management Console

1. At the top of these instructions, choose Start Lab to launch your lab.

   A Start Lab panel opens displaying the lab status. In the **Start Lab** dialog box that opens, note the AWS Region, as you will need to refer to it later in this lab.

2. Wait until you see the message "**Lab status: ready**", then click the **X** to close the Start Lab panel.

3. At the top of these instructions, choose AWS

   This will open the AWS Management Console in a new browser tab. The system will automatically log you in.

   Tip: If a new browser tab does not open, there will typically be a banner or icon at the top of your browser indicating that your browser is

preventing the site from opening pop-up windows. Click on the banner or icon and choose "Allow pop ups."

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time, to make it easier to follow the lab steps.

Task 1: Explore the Users and Groups

In this task, you will explore the Users and Groups that have already been created for you in IAM.

1. In the **AWS Management Console**, on the **Services** menu, select **IAM**.
2. In the navigation pane on the left, choose **Users**.

   The following IAM Users have been created for you:

   - user-1
   - user-2
   - user-3
3. Choose **user-1**.

   This will bring to a summary page for user-1. The **Permissions** tab will be displayed.
4. Notice that user-1 does not have any permissions.
5. Choose the **Groups** tab.

   user-1 also is not a member of any groups.
6. Choose the **Security credentials** tab.

   user-1 is assigned a **Console password**
7. In the navigation pane on the left, choose **User groups**.

   The following groups have already been created for you:

   - EC2-Admin
   - EC2-Support
   - S3-Support
8. Choose the **EC2-Support** group.

   This will bring you to the summary page for the **EC2-Support** group.

9. Choose the **Permissions** tab.

   This group has a Managed Policy associated with it, called **AmazonEC2ReadOnlyAccess**. Managed Policies are pre-built policies (built either by AWS or by your administrators) that can be attached to IAM Users and Groups. When the policy is updated, the changes to the policy are immediately apply against all Users and Groups that are attached to the policy.

10. Choose the plus (**+**) icon next to the AmazonEC2ReadOnlyAccess policy to view the policy details.

    Note: A policy defines what actions are allowed or denied for specific AWS resources.

    This policy is granting permission to List and Describe information about EC2, Elastic Load Balancing, CloudWatch and Auto Scaling. This ability to view resources, but not modify them, is ideal for assigning to a Support role.

    The basic structure of the statements in an IAM Policy is:

    - **Effect** says whether to *Allow* or *Deny* the permissions.
    - **Action** specifies the API calls that can be made against an AWS Service (eg *cloudwatch:ListMetrics*).
    - **Resource** defines the scope of entities covered by the policy rule (eg a specific Amazon S3 bucket or Amazon EC2 instance, or * which means *any resource*).

11. Choose the minus icon () to hide the policy details.
12. In the navigation pane on the left, choose **User groups**.
13. Choose the **S3-Support** group and then choose the **Permissions** tab.

    The S3-Support group has the **AmazonS3ReadOnlyAccess** policy attached.

14. Choose the plus (**+**) icon to view the policy details.

    This policy grants permissions to Get and List resources in Amazon S3.

15. Choose the minus icon () to hide the policy details.
16. In the navigation pane on the left, choose **User groups**.
17. Choose the **EC2-Admin** group and then choose the **Permissions** tab.

This Group is slightly different from the other two. Instead of a *Managed Policy*, it has an **Inline Policy**, which is a policy assigned to just one User or Group. Inline Policies are typically used to apply permissions for one-off situations.

18. Choose the plus (**+**) icon to view the policy details.

    This policy grants permission to view (Describe) information about Amazon EC2 and also the ability to Start and Stop instances.

19. Choose the minus icon () to hide the policy details.

Business Scenario

For the remainder of this lab, you will work with these Users and Groups to enable permissions supporting the following business scenario:

Your company is growing its use of Amazon Web Services, and is using many Amazon EC2 instances and a great deal of Amazon S3 storage. You wish to give access to new staff depending upon their job function:

User	In Group	Permissions
user-1	S3-Support	Read-Only access to Amazon S3
user-2	EC2-Support	Read-Only access to Amazon EC2
user-3	EC2-Admin	View, Start and Stop Amazon EC2 instances

Task 2: Add Users to Groups

You have recently hired **user-1** into a role where they will provide support for Amazon S3. You will add them to the **S3-Support** group so that they inherit the necessary permissions via the attached *AmazonS3ReadOnlyAccess* policy.

You can ignore any "not authorized" errors that appear during this task. They are caused by your lab account having limited permissions and will not impact your ability to complete the lab.

Add user-1 to the S3-Support Group

1. In the left navigation pane, choose **User groups**.
2. Choose the **S3-Support** group.
3. Choose the **Users** tab.

4. In the **Users** tab, choose **Add users**.
5. In the **Add Users to S3-Support** window, configure the following:
    - Select **user-1**.
    - At the bottom of the screen, choose **Add Users**.

    In the **Users** tab you will see that user-1 has been added to the group.

Add user-2 to the EC2-Support Group

You have hired **user-2** into a role where they will provide support for Amazon EC2.

1. Using similar steps to the ones above, add **user-2** to the **EC2-Support** group.

    user-2 should now be part of the **EC2-Support** group.

Add user-3 to the EC2-Admin Group

You have hired **user-3** as your Amazon EC2 administrator, who manage your EC2 instances.

1. Using similar steps to the ones above, add **user-3** to the **EC2-Admin** group.

    user-3 should now be part of the **EC2-Admin** group.
2. In the navigation pane on the left, choose **User groups**.

    Each Group should now have a **1** in the Users column for the number of Users in each Group.

    If you do not have a **1** beside each group, revisit the above instructions above to ensure that each user is assigned to a User group, as shown in the table in the Business Scenario section.

Task 3: Sign-In and Test Users

In this task, you will test the permissions of each IAM User.

1. In the navigation pane on the left, choose **Dashboard**.

    An **IAM users sign-in link** is displayed on the right. It will look similar to: *https://123456789012.signin.aws.amazon.com/console*

This link can be used to sign-in to the AWS Account you are currently using.

2. Copy the **Sign-in URL for IAM users in this account** to a text editor.
3. Open a private (Incognito) window.

   Mozilla Firefox

   - Choose the menu bars at the top-right of the screen
   - Select **New private window**

   Google Chrome

   - Choose the ellipsis at the top-right of the screen
   - Select **New Incognito Window**

   Microsoft Edge

   - Choose the ellipsis at the top-right of the screen
   - Choose **New InPrivate window**

   Microsoft Internet Explorer

   - Choose the **Tools** menu option
   - Choose **InPrivate Browsing**
4. Paste the **IAM users sign-in** link into the address bar of your private browser session and press **Enter**.

   Next, you will sign-in as **user-1**, who has been hired as your Amazon S3 storage support staff.
5. Sign-in with:

   - **IAM user name:** `user-1`
   - **Password:** `Lab-Password1`
6. In the **Services** menu, choose **S3**.
7. Choose the name of the bucket that exists in the account and browse the contents.

   Since your user is part of the **S3-Support** Group in IAM, they have permission to view a list of Amazon S3 buckets and the contents.

   Note: The bucket does not contain any objects.

Now, test whether they have access to Amazon EC2.

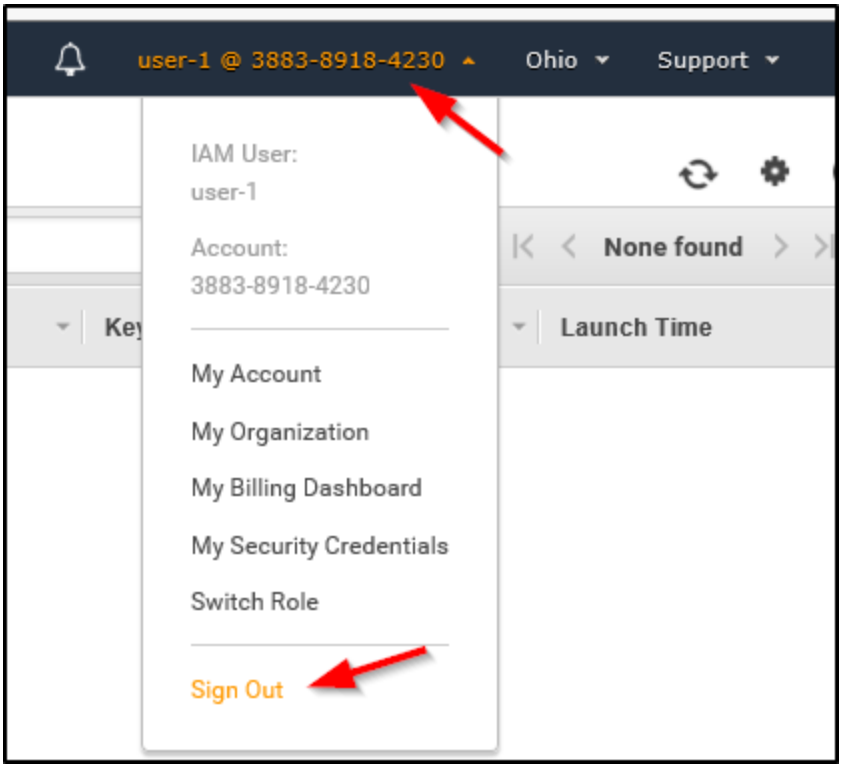
8. In the **Services** menu, choose **EC2**.
9. In the left navigation pane, choose **Instances**.

   You cannot see any instances. Instead, you see a message that states *You are not authorized to perform this operation*. This is because this user has not been granted any permissions to access Amazon EC2.

   You will now sign-in as **user-2**, who has been hired as your Amazon EC2 support person.

10. Sign user-1 out of the **AWS Management Console** by completing the following actions:

    - At the top of the screen, choose **user-1**
    - Choose **Sign Out**



11. Paste the **IAM users sign-in** link into your private browser tab's address bar and press **Enter**.

    Note: This link should be in your text editor.

12. Sign-in with:

    - **IAM user name:** `user-2`

- **Password:** `Lab-Password2`

13. In the **Services** menu, choose **EC2**.
14. In the navigation pane on the left, choose **Instances**.

    You are now able to see an Amazon EC2 instance because you have Read Only permissions. However, you will not be able to make any changes to Amazon EC2 resources.

    If you cannot see an Amazon EC2 instance, then your Region may be incorrect. In the top-right of the screen, pull-down the Region menu and select the region that you noted at the start of the lab (for example, **N. Virginia**).

    - Select the instance named *LabHost*.

15. In the **Instance state** menu above, select **Stop instance**.
16. In the **Stop Instance** window, select **Stop**.

    You will receive an error stating *You are not authorized to perform this operation*. This demonstrates that the policy only allows you to view information, without making changes.

17. Choose the X to close the *Failed to stop the instance* message.

    Next, check if user-2 can access Amazon S3.

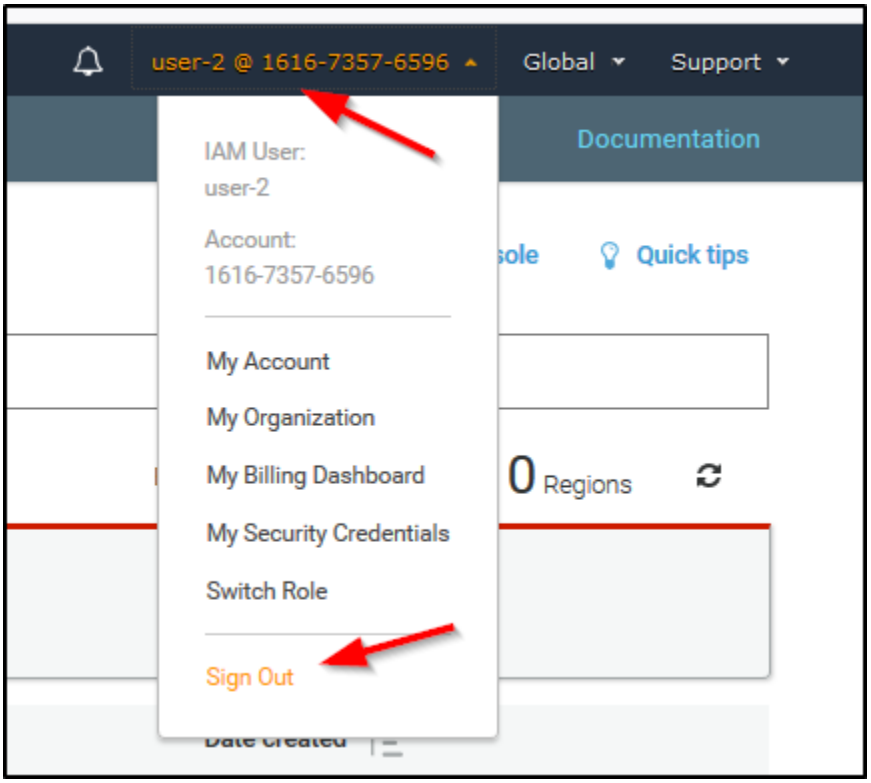
18. In the **Services**, choose **S3**.

    You will see the message **You don't have permissions to list buckets** because user-2 does not have permission to access Amazon S3.

    You will now sign-in as **user-3**, who has been hired as your Amazon EC2 administrator.

19. Sign user-2 out of the **AWS Management Console** by completing the following actions:

    - At the top of the screen, choose **user-2**
    - Choose **Sign Out**



20. Paste the **IAM users sign-in** link into your private window and press **Enter**.
21. Paste the sign-in link into the address bar of your private web browser tab again. If it is not in your clipboard, retrieve it from the text editor where you stored it earlier.
22. Sign-in with:
    - **IAM user name:** `user-3`
    - **Password:** `Lab-Password3`
23. In the **Services** menu, choose **EC2**.
24. In the navigation pane on the left, choose **Instances**.

    As an EC2 Administrator, you should now have permissions to Stop the Amazon EC2 instance.

    Select the instance named *LabHost* .

    If you cannot see an Amazon EC2 instance, then your Region may be incorrect. In the top-right of the screen, pull-down the Region menu and select the region that you noted at the start of the lab (for example, **N. Virginia**).
25. In the **Instance state** menu, choose **Stop instance**.

26. In the **Stop instance** window, choose **Stop**.

    The instance will enter the *stopping* state and will shutdown.

27. Close your private browser window.

Lab complete

1. Choose End Lab at the top of this page, and then select **Yes** to confirm that you want to end the lab.

   A panel indicates that *You may close this message box now...*

2. Select the **X** in the top-right corner to close the panel.

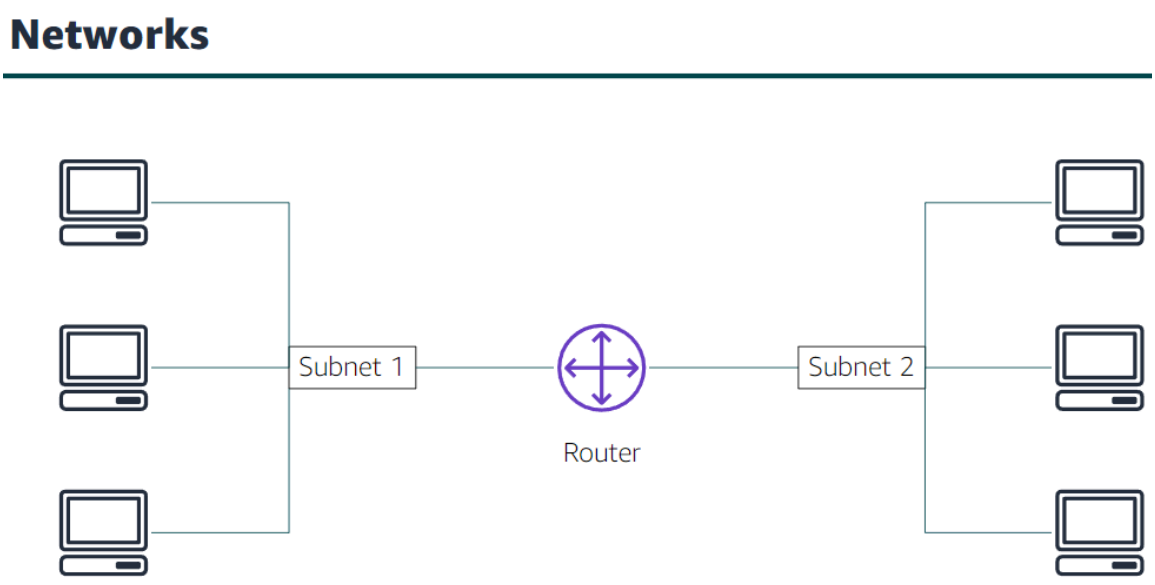
Conclusion

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

---

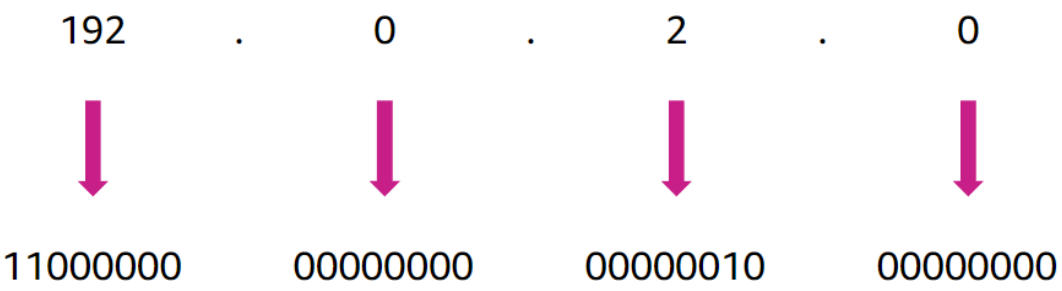
Module 5: Networking and Content Delivery

Section 1: Networking Basics



A computer network is two or more client machines that are connected together to share resources. A network can be logically partitioned into subnets.

IP addresses



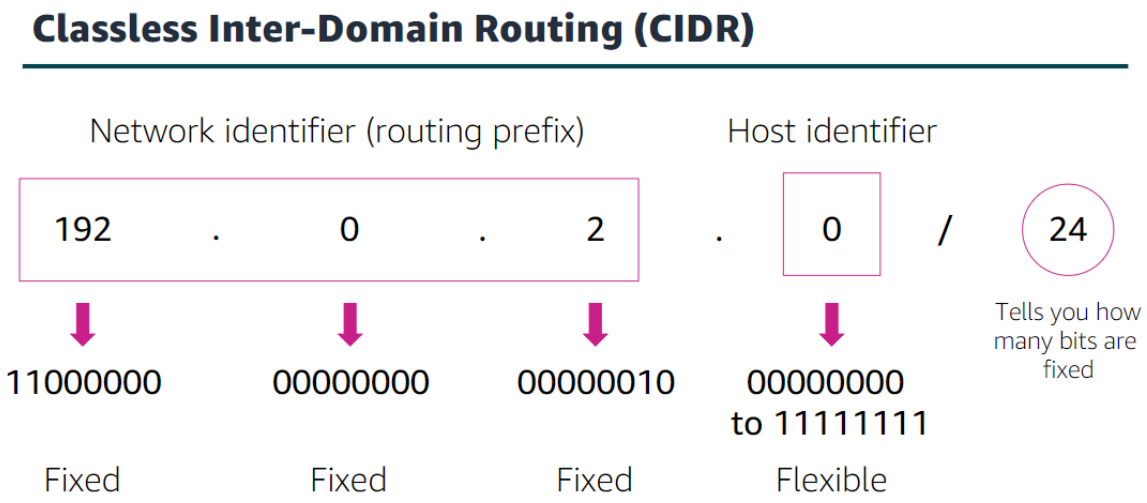
The combined total of the four numbers for an IP address is 32 bits in binary format.

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.

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.



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.

The last number (24) tells you that the first 24 bits must be fixed. The last 8 bits are flexible, which means that 28(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.

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	• Ensures that the application layer can read the data • Encryption	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)

The Open Systems Interconnection (OSI) model is a conceptual model that is usedto 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).

Section 2: Amazon VPC

Amazon 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

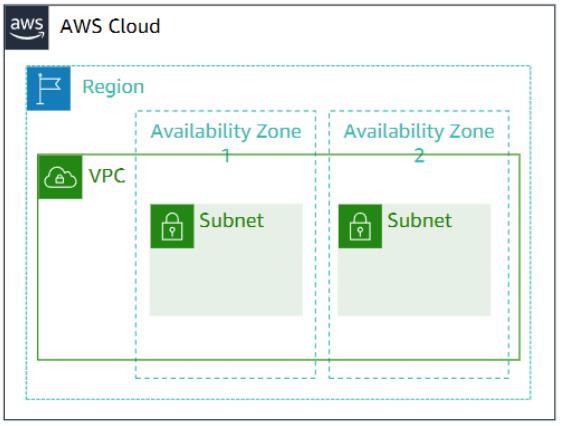
Amazon Virtual Private Cloud (Amazon VPC) is a service that lets you provision a logically isolated section of the AWS Cloud.

You can use both IPv4 and IPv6 in your VPC for secure access to resources and applications.

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

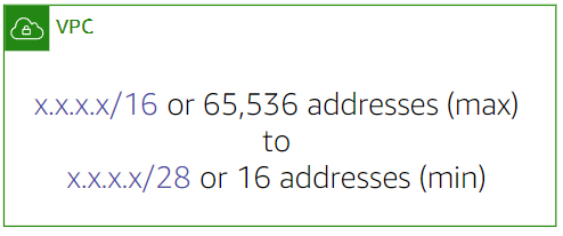


A VPC is a virtual network that is logically isolated from other virtual networks in the AWS Cloud. A VPC is dedicated to your account. VPCs belong to a single AWS Region and can span multiple Availability Zones.

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.



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

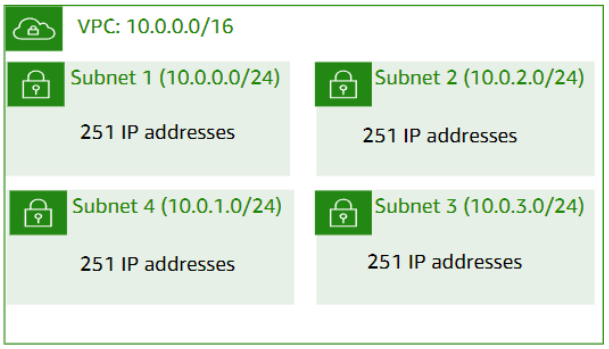
block might be as large as /16 (which is 216, or 65,536 addresses) or as small as /28 (which is 24, or 16 addresses).

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.

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.



ws

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

When you create a subnet, it requires its own CIDR block. 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

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.

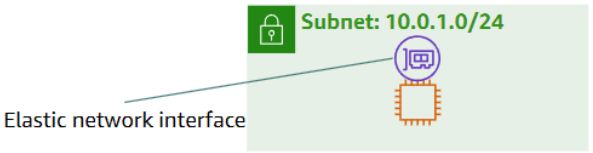
An Elastic IP address is a static and public IPv4 address that is designed for dynamic cloud computing.

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.

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.



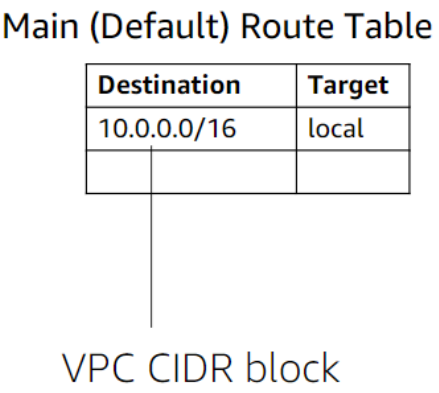
An elastic network interface is a virtual network interface that you can attach or detach from an instance in a VPC.

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.

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).



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.

The main route table is the route table is automatically assigned to your VPC. It controls the routing for all subnets that are not explicitly associated with any other route table.

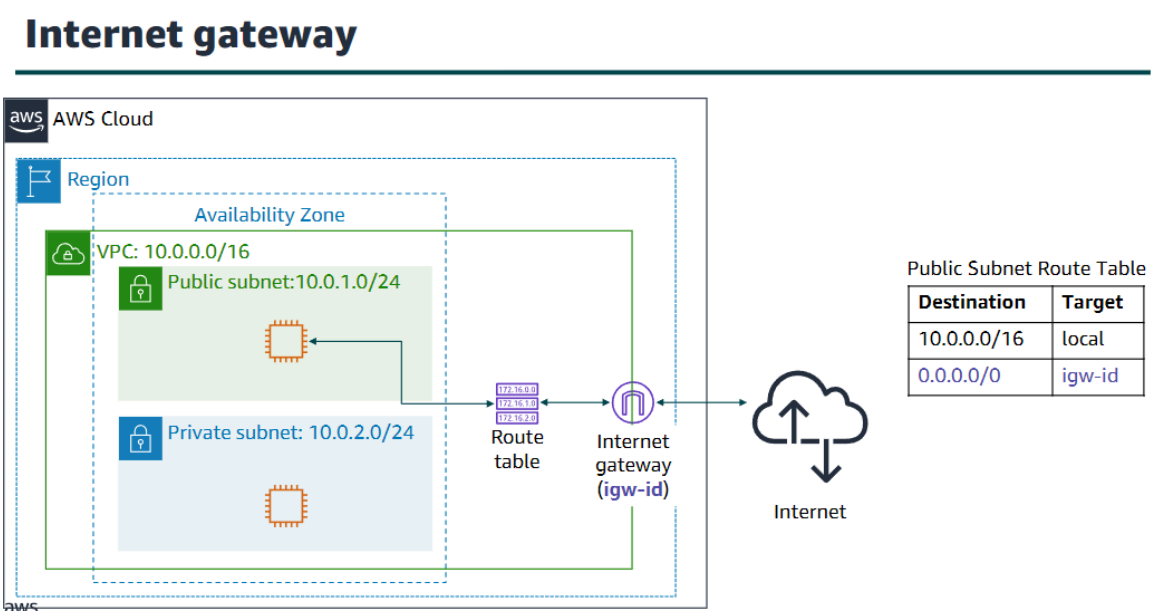
A subnet can be associated with only one route table at a time, but you can associate multiple subnets with the same route table.

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.

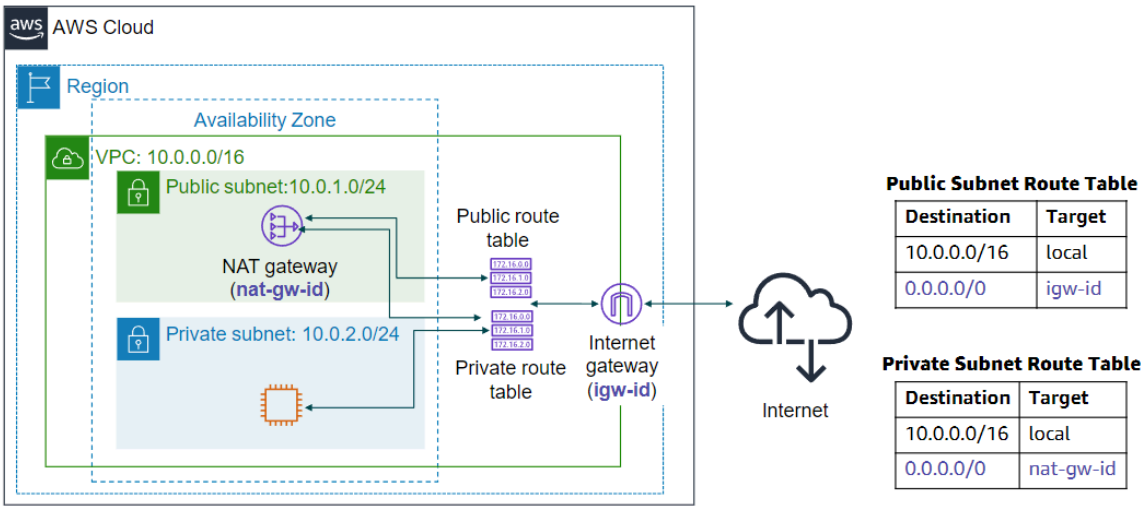
Section 3: VPC Networking



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).

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.

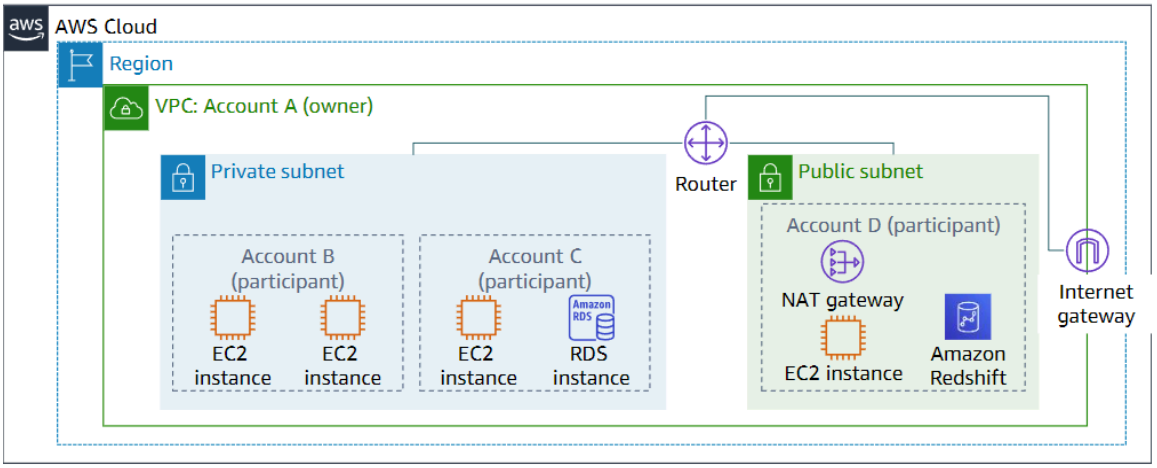
See the AWS Documentation for more information about

NAT gateways at https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html

NAT instances at https://docs.aws.amazon.com/vpc/latest/userguide/VPC_NAT_Instance.html

Differences between NAT gateways and NAT instances at https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-comparison.html

VPC sharing



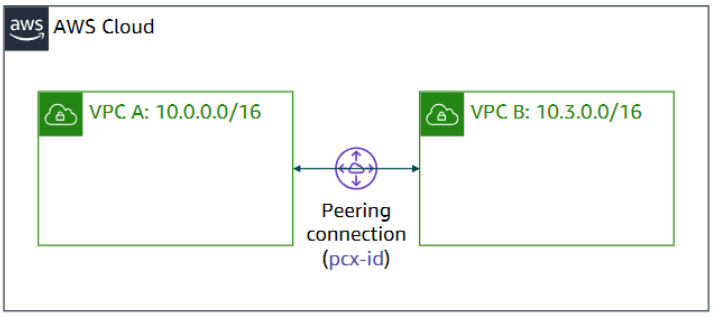
VPC sharing enables customers to share subnets with other AWS accounts in the same organization in AWS Organizations.

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 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:

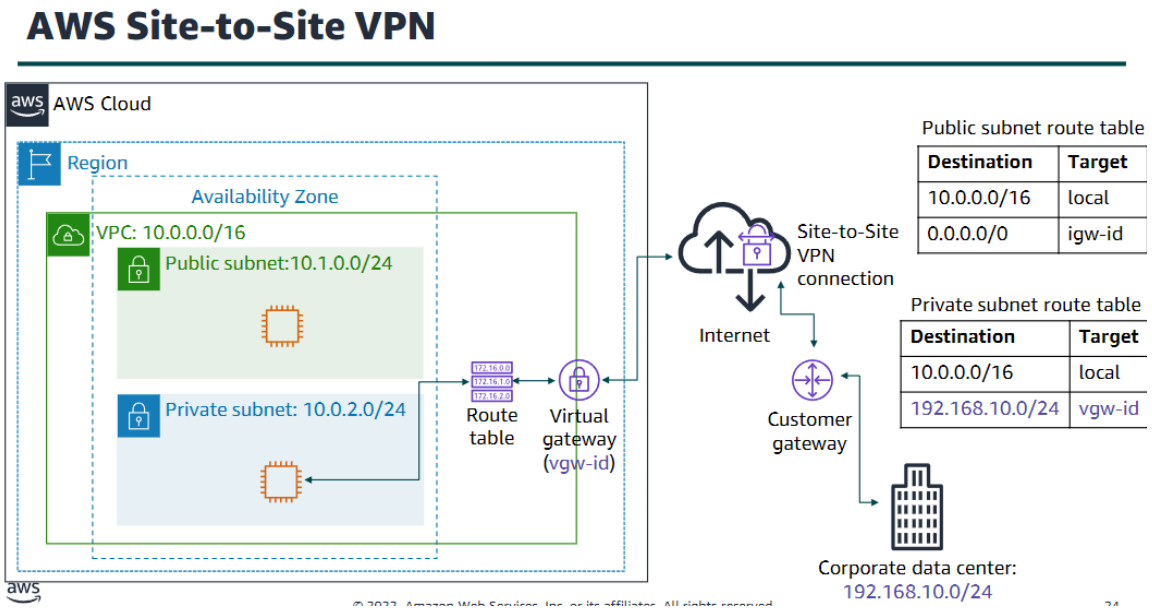
- IP spaces cannot overlap.
- Transitive peering is not supported.
- You can only have one peering resource between the same two VPCs.

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.

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.

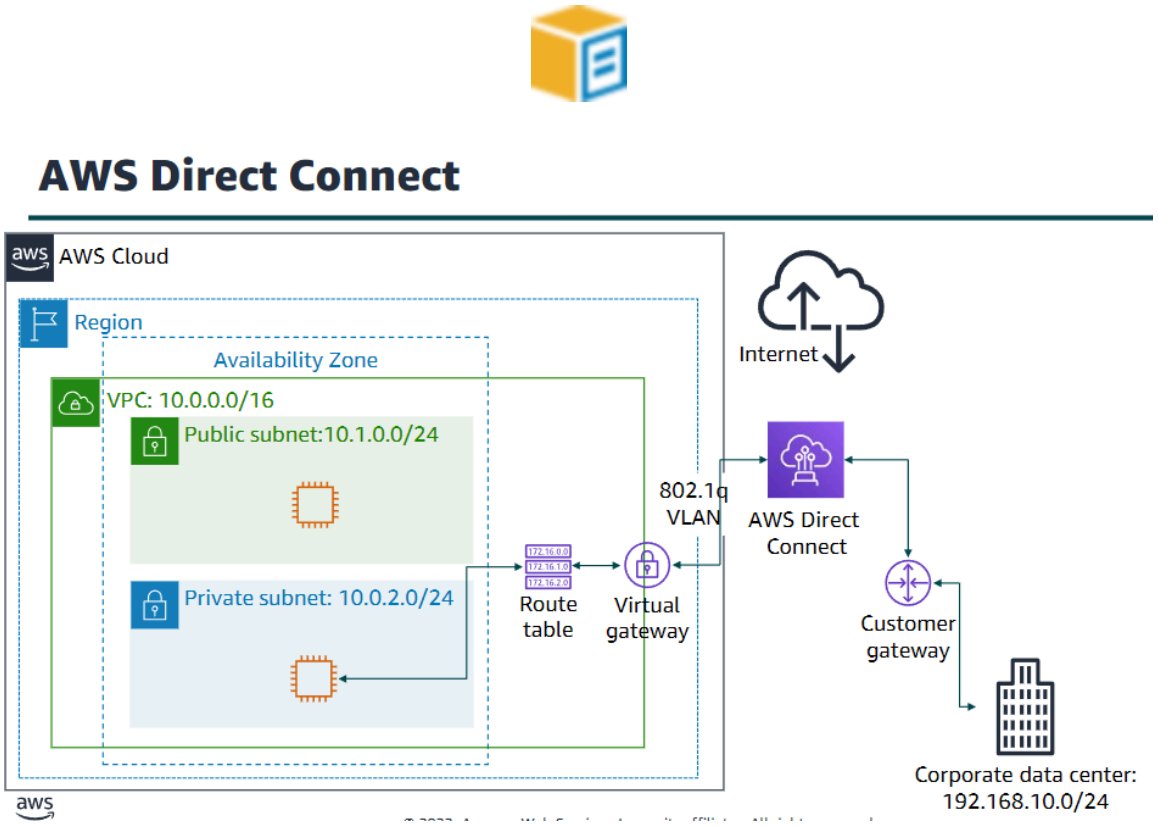
VPC peering has some restrictions:

- IP address ranges cannot overlap.
- Transitive peering is not supported.
    - For example, if VPC A is connected to VPC B and VPC A is connected to VPC C, 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.

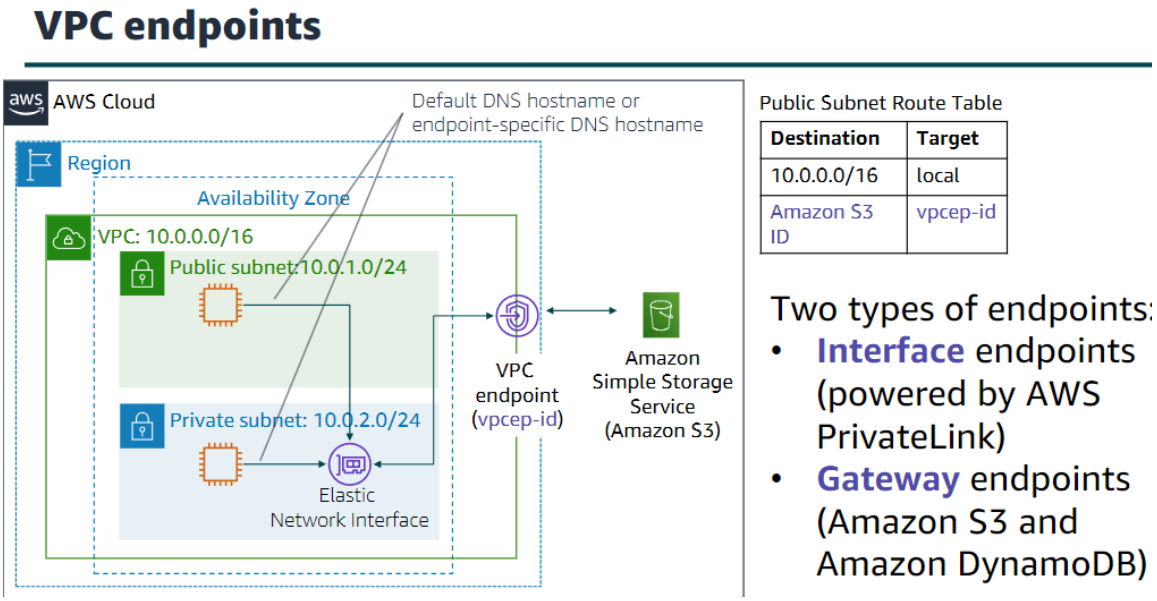


Instances that you launch into a VPC cannot communicate with a remote network. To connect your VPC to your remote network (that is, create a virtual private network or VPN connection), you:

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.



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. DX uses open standard 802.1q virtual local area networks (VLANs).



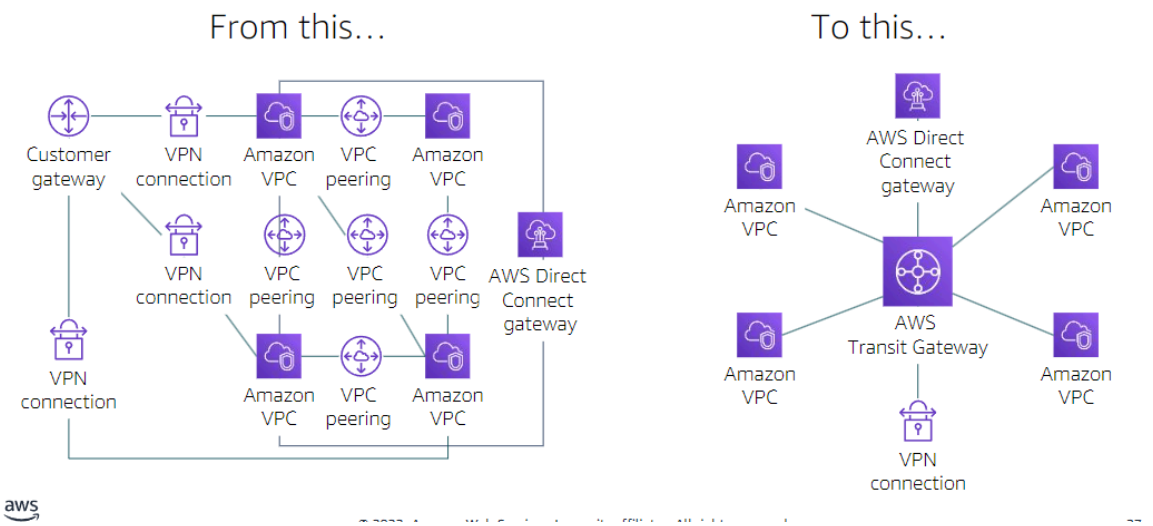
A VPC end point is a virtual device that enables you to privately connect your VPC to supported AWS services and VPC endpoint services that are powered by AWS Private Link. Connection to these services does not require an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection.

There are two types of VPC endpoints:

- An interface VPC endpoint (interface endpoint) enables connectivity to services powered by AWS Private Link. It can connect to AWS services, services hosted by other AWS customers and AWS Partner Network (APN) Partners, and supported AWS Marketplace APN Partner services. The service provider is the owner, and the service consumer creates the interface endpoint. Charges apply for creating and using the interface endpoint, including hourly usage rates and data processing rates.

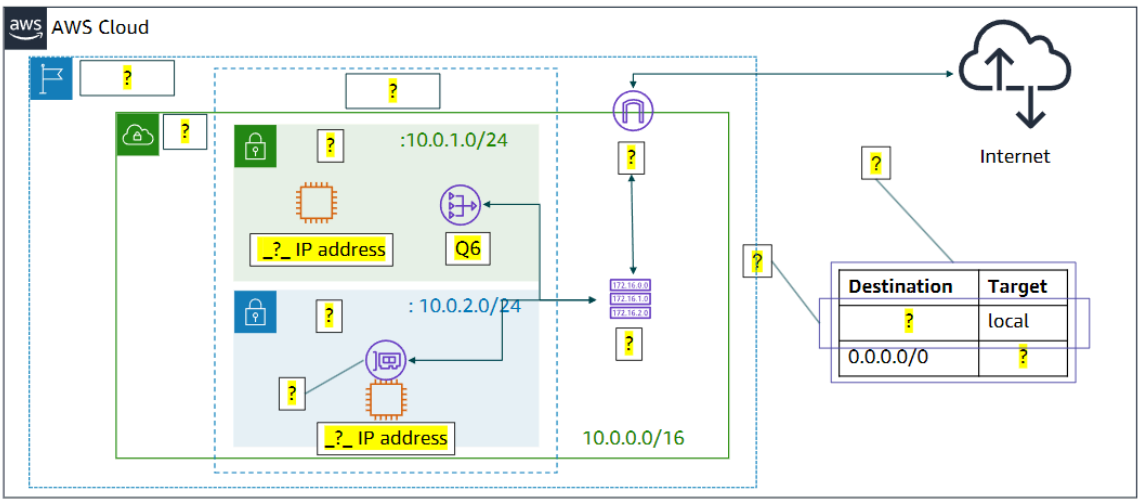
- Gateway endpoints incur no additional charges. Standard charges for data transfer and resource usage apply.

AWS Transit Gateway

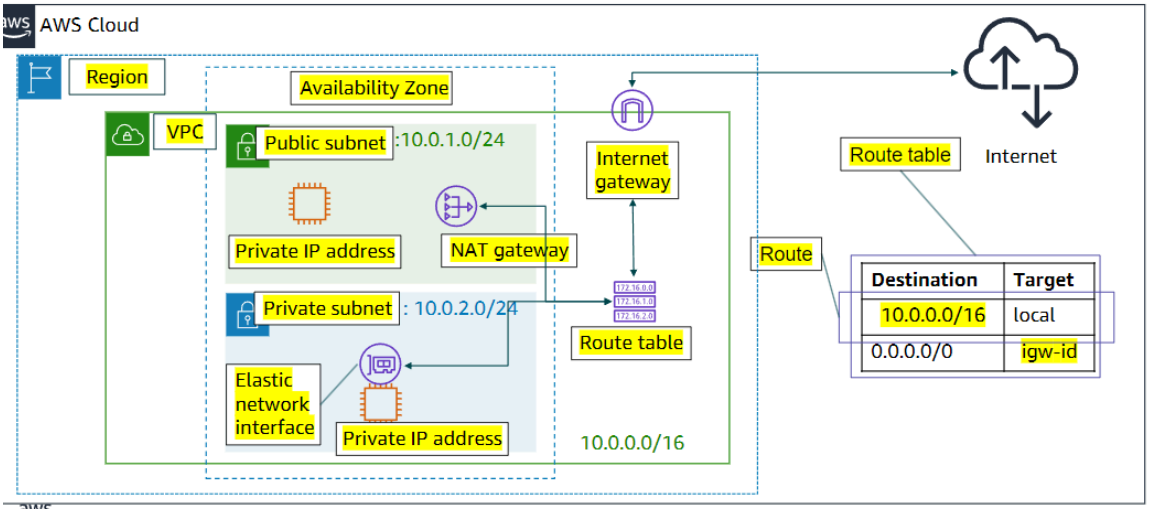


With AWS Transit Gateway, you only need to create and manage a single connection from the central gateway into each VPC, on-premises data center, or remote office across your network. A transit gateway acts as a hub that controls how traffic is routed among all the connected networks, which act like 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



Activity: Solution



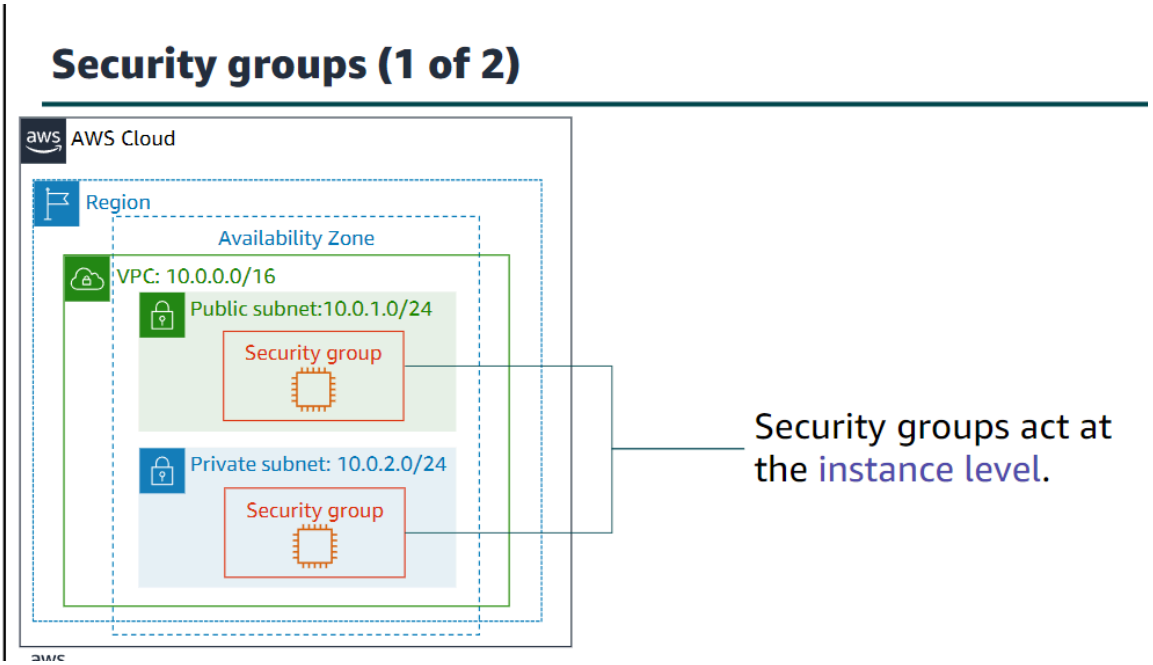
- Demo for Amazon VPC: https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-100-ACFNDS-20-EN/Module_5_VPC_Wizard+v2.0.mp4

https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-100-ACFNDS-20-EN/Module_5_VPC_Wizard+v2.0.mp4

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.

Section 4: VPC Security



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.

Security groups (2 of 2)

- 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.

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.

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.

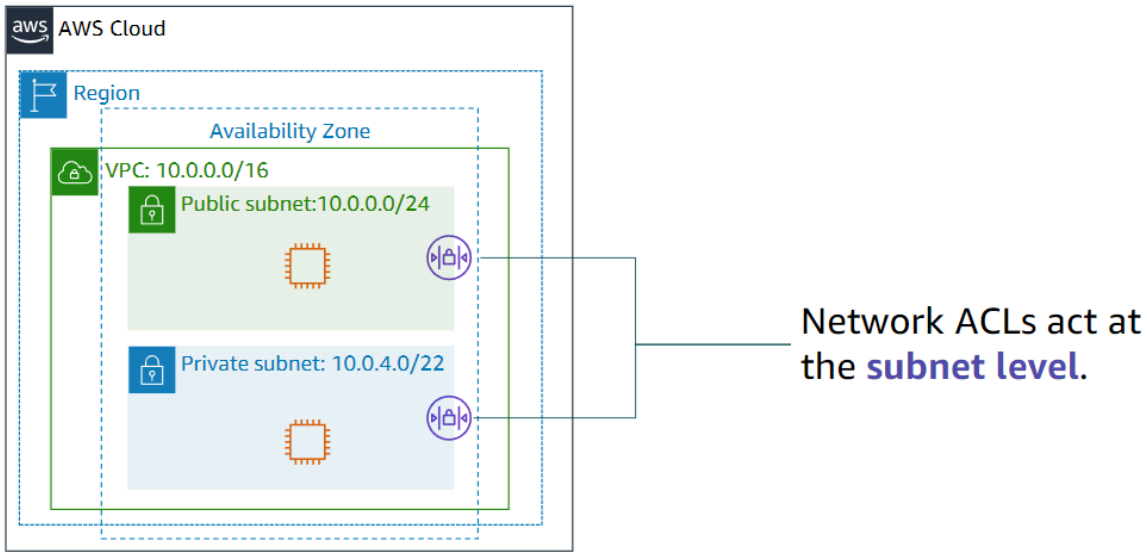
Custom security group examples

- 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

Network access control lists (network ACLs 1 of 2)



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.

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.

Network access control lists (network ACLs 2 of 2)

- 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

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

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.

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

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.

Section 5: Amazon Route 53

Amazon Route 53

Amazon Route 53

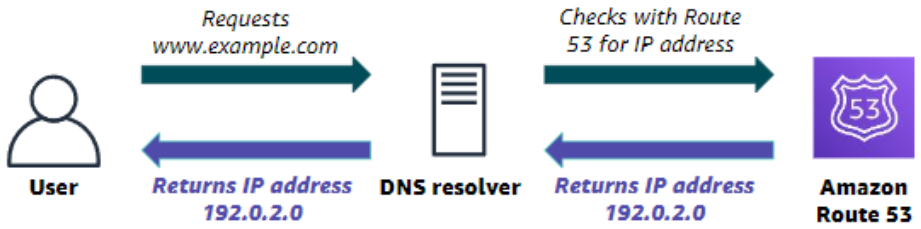
- Is a highly available and scalable Domain Name System (DNS) web service
- Is used to route end users to internet applications by translating names (like *www.example.com*) into numeric IP addresses (like *192.0.2.1*) that computers use to connect to each other
- Is fully compliant with IPv4 and IPv6
- 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

Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It is designed to give developers and businesses are liable and cost-effective way to route users to internet applications by translating names (like 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.

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 DNS resolution



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.

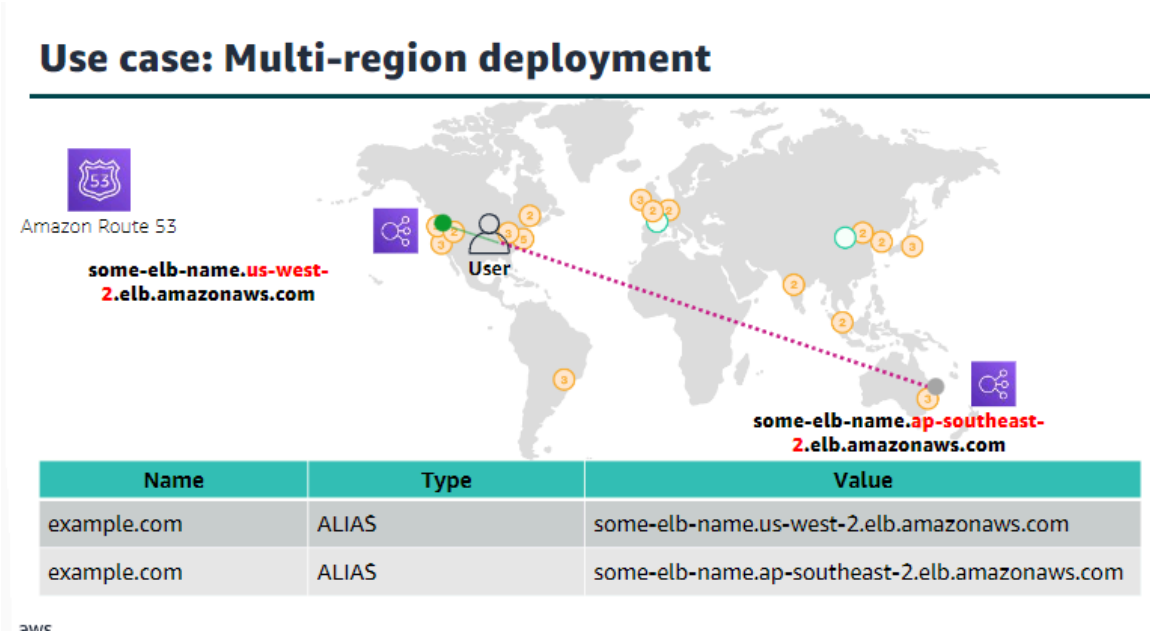
Latency routing (LBR) –Use when you have resources in multiple AWS Regions and you want to route traffic to the Region that provides the best latency. Latency routing works by routing your customers to the AWS endpoint (for example, Amazon EC2 instances, Elastic IP addresses, or load balancers) that 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.

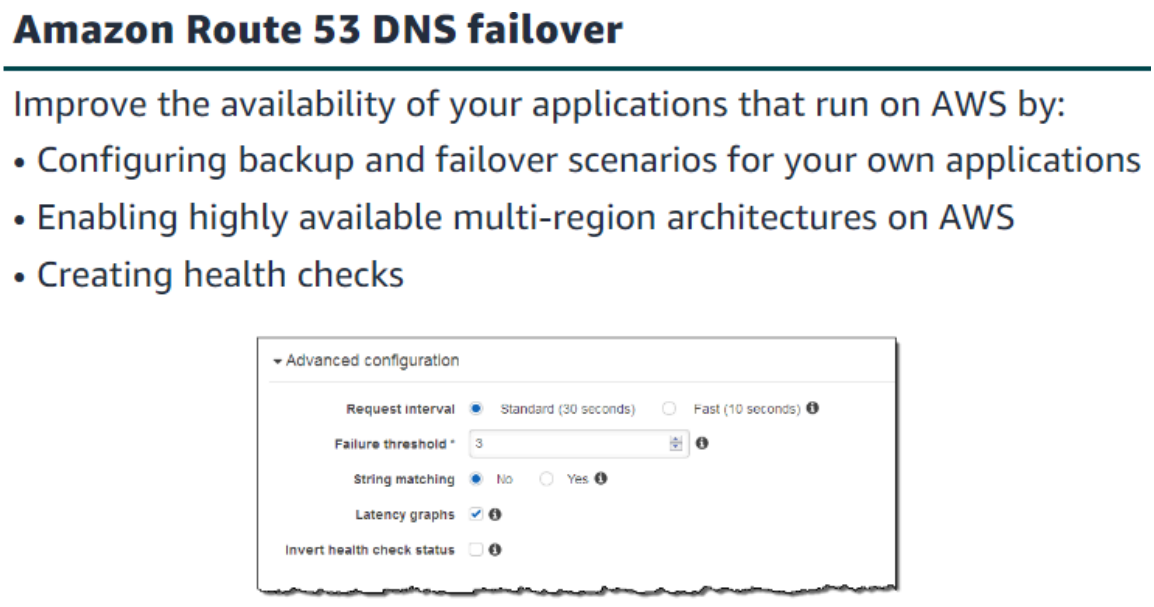
Multivalue answer routing–Use when you want Route53 to respond to DNS queries with up to eight healthy records that are selected at random. You can configure Amazon Route53 to return multiple values—such as IP addresses for your web servers—in response to DNS queries. You can specify multiple values for almost any record, but multivalueanswer routing also enables you to check the health of each resource so that Route53 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.



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.

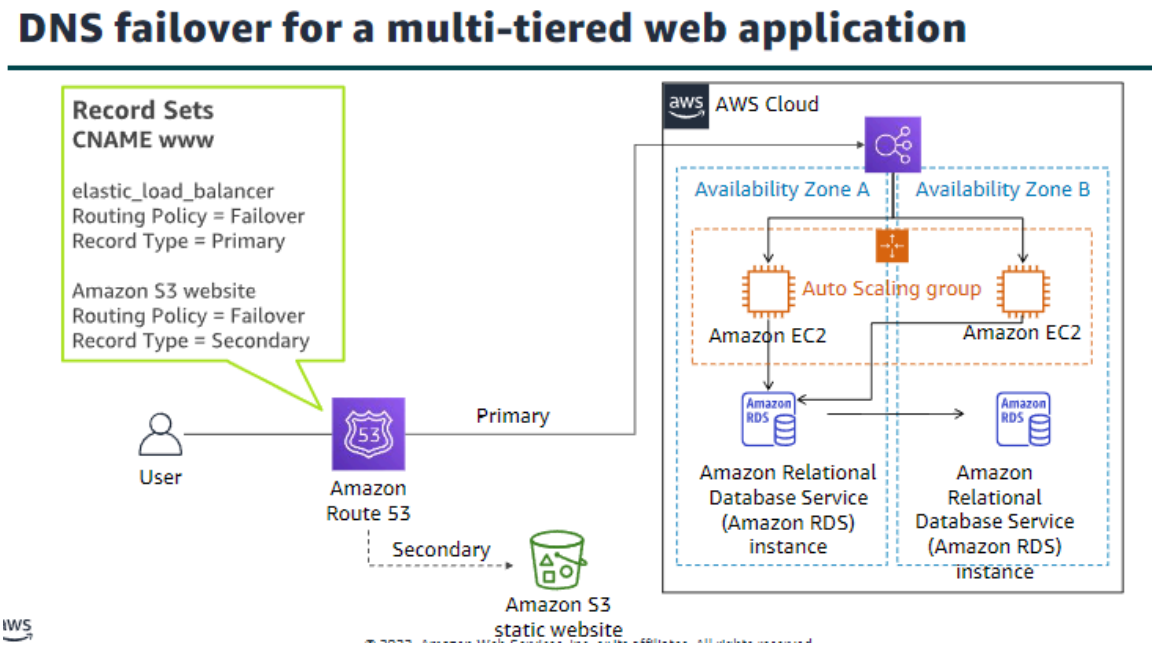
The benefits of multi-region deployment of Route 53 include:

- Latency-based routing to the Region
- Load balancing routing to the Availability Zone



Amazon Route 53 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.



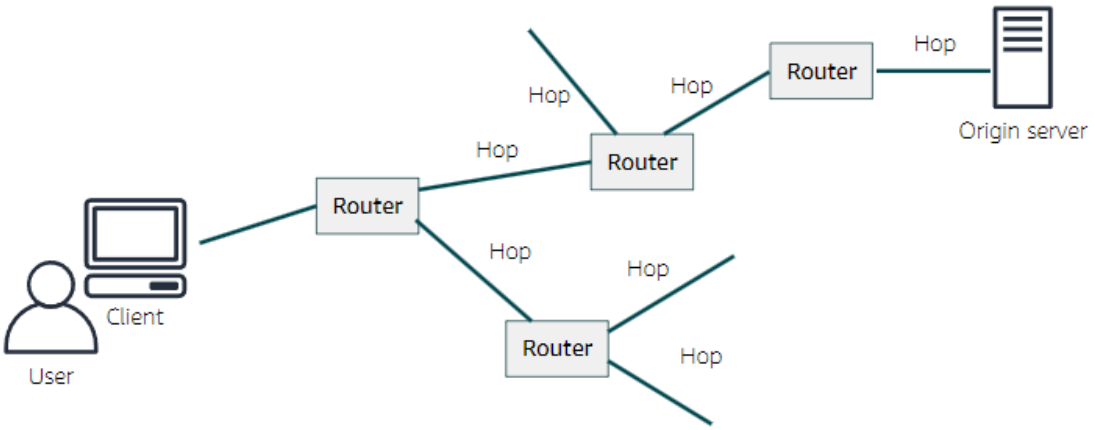
You can do the following tasks with Route 53 to ensure high availability:

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 6: Amazon CloudFront

Content delivery and network latency



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 contentdelivery 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

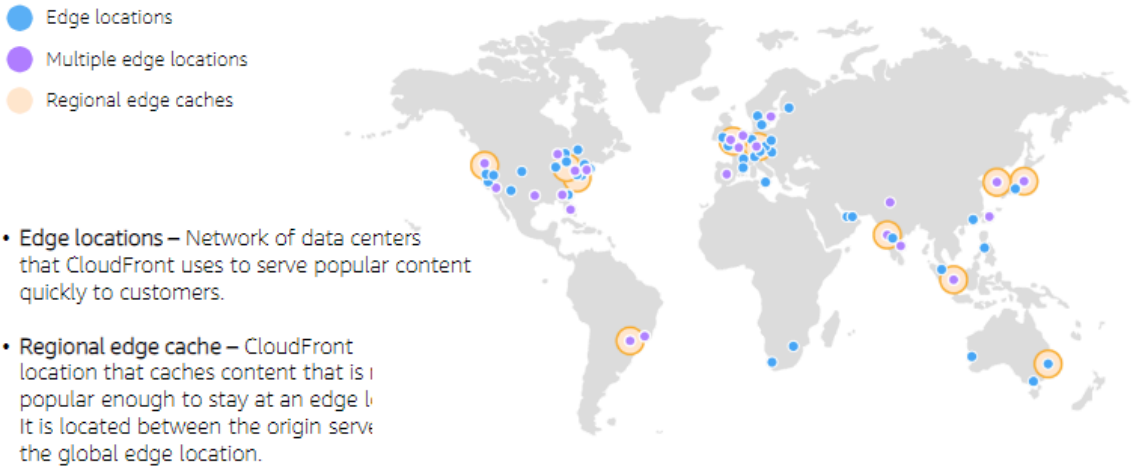
Amazon CloudFront

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.

Amazon CloudFront infrastructure



- **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 popular enough to stay at an edge l It is located between the origin serv the global edge location.

AmazonCloudFront 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.

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@Edgeso 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.

Amazon CloudFront pricing

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.

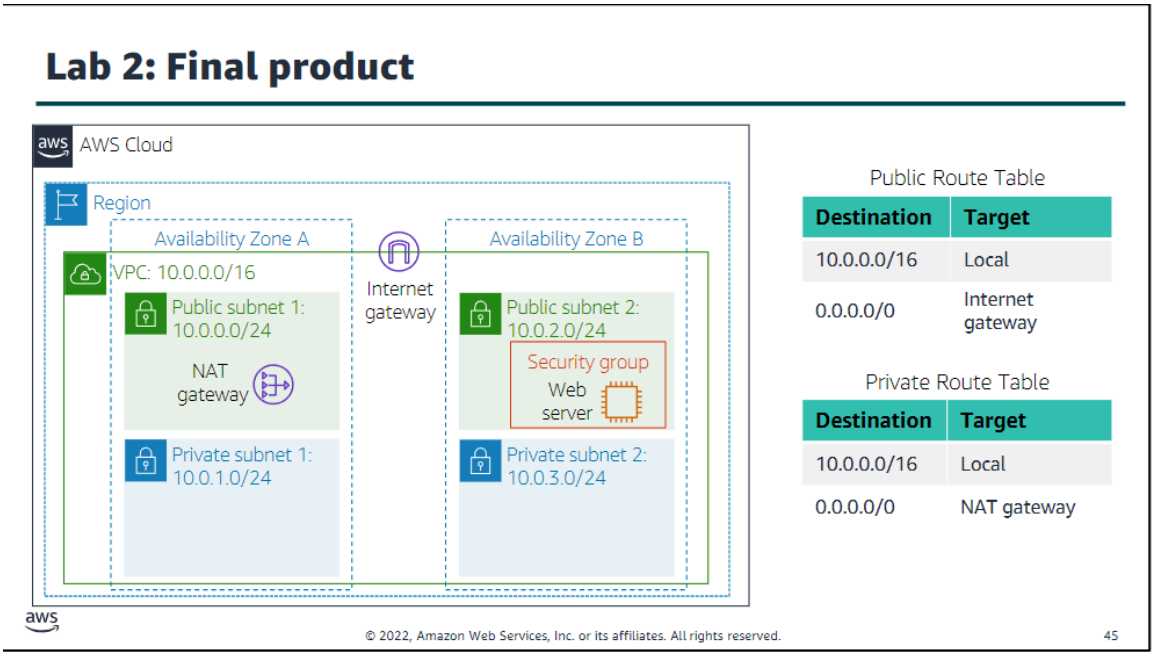
Amazon CloudFront charges are based on actual usage of the service in four areas:•Data transfer out–You are charged for the volume of data that is transferred out from Amazon CloudFront edge locations, measured in GB, to the internet or to your origin (both AWS origins and other origin servers).

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 day / 30 days) * $600 = $20.

Additional resources

- Amazon VPC Overview pag: https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html
- Amazon Virtual Private Cloud Connectivity Options whitepaper: https://docs.aws.amazon.com/whitepapers/latest/aws-vpc-connectivity-options/introduction.html
- One to Many: Evolving VPC Design AWS Architecture blog post: https://aws.amazon.com/blogs/architecture/one-to-many-evolving-vpc-design/
- Amazon VPC User Guide: https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html
- Amazon CloudFront overview page: https://aws.amazon.com/cloudfront/?nc=sn&loc=1

Lab Activity 2



This architecture diagram depicts what you create in the lab.

In this lab, you will use Amazon Virtual Private Cloud (VPC) to create your own VPC and add additional components to produce a customized network. You will also create a security group. You will then configure and customize an EC2 instance to run a web server and you will launch the EC2 instance to run in a subnet in the VPC.

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch Amazon Web Services (AWS) resources into a virtual network that you defined. This virtual network closely resembles a traditional network that you would operate in your own data center, with the benefits of using the scalable infrastructure of AWS. You can create a VPC that spans multiple Availability Zones.

After completing this lab, you should be able to do the following:

- Create a VPC.
- Create subnets.
- Configure a security group.
- Launch an EC2 instance into a VPC.

Duration

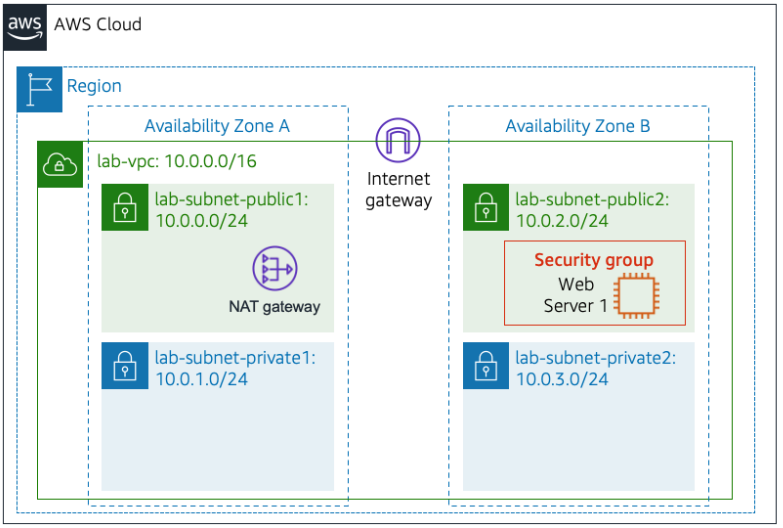
This lab takes approximately **30 minutes** to complete.

AWS service restrictions

In this lab environment, access to AWS services and service actions might be restricted to the ones that are needed to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that are described in this lab.

Scenario

In this lab you build the following infrastructure:



Public Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	Internet gateway

Private Route Tables

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	NAT gateway

Accessing the AWS Management Console

1. At the top of these instructions, choose Start Lab to launch your lab.

   A Start Lab panel opens displaying the lab status.

2. Wait until you see the message "**Lab status: ready**", then choose the **X** to close the Start Lab panel

3. At the top of these instructions, choose AWS

   This will open the AWS Management Console in a new browser tab. The system will automatically log you in.

   Tip: If a new browser tab does not open, there will typically be a banner or icon at the top of your browser indicating that your browser is preventing the site from opening pop-up windows. Choose on the banner or icon and choose "Allow pop ups."

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time, to make it easier to follow the lab steps.

Task 1: Create Your VPC

In this task, you will use the *VPC and more* option in the VPC console to create multiple resources, including a *VPC*, an *Internet Gateway*, a *public subnet* and a *private subnet* in a single Availability Zone, two *route tables*, and a *NAT Gateway*.

1. In the search box to the right of **Services**, search for and choose **VPC** to open the VPC console.
2. Begin creating a VPC.
    - In the top right of the screen, verify that **N. Virginia (us-east-1)** is the region.
    - Choose the **VPC dashboard** link which is also towards the top left of the console.
    - Next, choose **Create VPC**.

      Note: If you do not see a button with that name, choose the Launch VPC Wizard button instead.
3. Configure the VPC details in the *VPC settings* panel on the left:
    - Choose **VPC and more**.
    - Under **Name tag auto-generation**, keep *Auto-generate* selected, however change the value from project to `lab`.
    - Keep the **IPv4 CIDR block** set to 10.0.0.0/16
    - For **Number of Availability Zones**, choose **1**.
    - For **Number of *public* subnets**, keep the **1** setting.
    - For **Number of *private* subnets**, keep the **1** setting.
    - Expand the **Customize subnets CIDR blocks** section
        - Change **Public subnet CIDR block in us-east-1a** to `10.0.0.0/24`
        - Change **Private subnet CIDR block in us-east-1a** to `10.0.1.0/24`
    - Set **NAT gateways** to **In 1 AZ**.

- Set **VPC endpoints** to **None**.
- Keep both **DNS hostnames** and **DNS resolution** *enabled*.

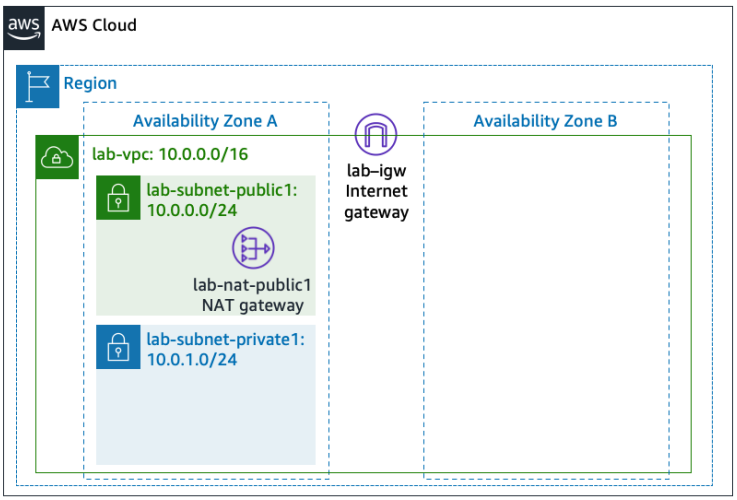
4. In the *Preview* panel on the right, confirm the settings you have configured.
   - **VPC:** `lab-vpc`
   - **Subnets**:
     - us-east-1a
       - ***Public*** **subnet name:** `lab-subnet-public1-us-east-1a`
       - ***Private*** **subnet name:** `lab-subnet-private1-us-east-1a`
   - **Route tables**
     - `lab-rtb-public`
     - `lab-rtb-private1-us-east-1a`
   - **Network connections**
     - `lab-igw`
     - `lab-nat-public1-us-east-1a`
5. At the bottom of the screen, choose **Create VPC**

   The VPC resources are created. The NAT Gateway will take a few minutes to activate.

   Please wait until *all* the resources are created before proceeding to the next step.
6. Once it is complete, choose **View VPC**

   The wizard has provisioned a VPC with a public subnet and a private subnet in one Availability Zone with route tables for each subnet. It also created an Internet Gateway and a NAT Gateway.

   To view the settings of these resources, browse through the VPC console links that display the resource details. For example, choose **Subnets** to view the subnet details and choose **Route tables** to view the route table details. The diagram below summarizes the VPC resources you have just created and how they are configured.



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

An *Internet gateway* is a VPC resource that allows communication between EC2 instances in your VPC and the Internet.

The `lab-subnet-public1-us-east-1a` public subnet has a CIDR of **10.0.0.0/24**, which means that it contains all IP addresses starting with **10.0.0.x**. The fact the route table associated with this public subnet routes 0.0.0.0/0 network traffic to the internet gateway is what makes it a public subnet.

A *NAT Gateway*, is a VPC resource used to provide internet connectivity to any EC2 instances running in *private* subnets in the VPC without those EC2 instances needing to have a direct connection to the internet gateway.

The `lab-subnet-private1-us-east-1a` private subnet has a CIDR of **10.0.1.0/24**, which means that it contains all IP addresses starting with **10.0.1.x**.

Task 2: Create Additional Subnets

In this task, you will create two additional subnets for the VPC in a second Availability Zone. Having subnets in multiple Availability Zones within a VPC is useful for deploying solutions that provide *High Availability*.

After creating a VPC as you have already done, you can still configure it further, for example, by adding more **subnets**. Each subnet you create resides entirely within one Availability Zone.

1. In the left navigation pane, choose **Subnets**.

   First, you will create a second *public* subnet.

2. Choose **Create subnet** then configure:

- **VPC ID: lab-vpc** (select from the menu).
- **Subnet name:** `lab-subnet-public2`
- **Availability Zone:** Select the *second* Availability Zone (for example, us-east-1b)
- **IPv4 CIDR block:** `10.0.2.0/24`

The subnet will have all IP addresses starting with **10.0.2.x**.

3. Choose **Create subnet**

   The second *public* subnet was created. You will now create a second *private* subnet.

4. Choose **Create subnet** then configure:

   - **VPC ID:** `lab-vpc`
   - **Subnet name:** `lab-subnet-private2`
   - **Availability Zone:** Select the *second* Availability Zone (for example, us-east-1b)
   - **IPv4 CIDR block:** `10.0.3.0/24`

   The subnet will have all IP addresses starting with **10.0.3.x**.

5. Choose **Create subnet**

   The second *private* subnet was created.

   You will now configure this new *private* subnet to route internet-bound traffic to the NAT Gateway so that resources in the second private subnet are able to connect to the Internet, while still keeping the resources private. This is done by configuring a *Route Table*.

   A *route table* contains a set of rules, called *routes*, that are used to determine where network traffic is directed. Each subnet in a VPC must be associated with a route table; the route table controls routing for the subnet.

6. In the left navigation pane, choose **Route tables**.
7. Select the **lab-rtb-private1-us-east-1a** route table.
8. In the lower pane, choose the **Routes** tab.

   Note that **Destination 0.0.0.0/0** is set to **Target nat-xxxxxxxx**. This means that traffic destined for the internet (0.0.0.0/0) will be sent to the

NAT Gateway. The NAT Gateway will then forward the traffic to the internet.

This route table is therefore being used to route traffic from private subnets.

9. Choose the **Subnet associations** tab.

   You created this route table in task 1 when you chose to create a VPC and multiple resources in the VPC. That action also created *lab-subnet-private-1* and associated that subnet with this route table.

   Now that you have created another private subnet, lab-subnet-private-2, you will associate this route table with that subnet as well.

10. In the Explicit subnet associations panel, choose **Edit subnet associations**
11. Leave **lab-subnet-private1-us-east-1a** selected, but also select **lab-subnet-private2**.
12. Choose **Save associations**

    You will now configure the Route Table that is used by the Public Subnets.

13. Select the **lab-rtb-public** route table (and deselect any other subnets).
14. In the lower pane, choose the **Routes** tab.

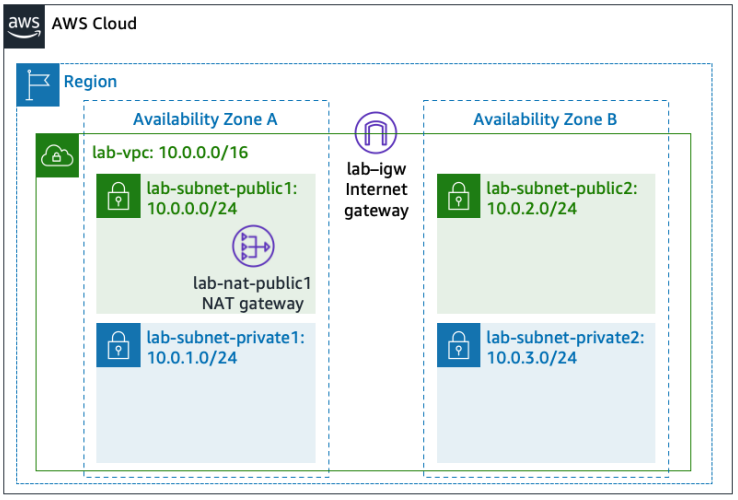
    Note that **Destination 0.0.0.0/0** is set to Target **igw-xxxxxxxx**, which is an Internet Gateway. This means that internet-bound traffic will be sent straight to the internet via this Internet Gateway.

    You will now associate this route table to the second public subnet you created.

15. Choose the **Subnet associations** tab.
16. In the Explicit subnet associations area, choose **Edit subnet associations**
17. Leave **lab-subnet-public1-us-east-1a** selected, but also select **lab-subnet-public2**.
18. Choose **Save associations**

    Your VPC now has public and private subnets configured in two Availability Zones. The route tables you created in task 1 have also been

updated to route network traffic for the two new subnets.



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

Task 3: Create a VPC Security Group

In this task, you will create a VPC security group, which acts as a virtual firewall. When you launch an instance, you associate one or more security groups with the instance. You can add rules to each security group that allow traffic to or from its associated instances.

1. In the left navigation pane, choose **Security groups**.
2. Choose **Create security group** and then configure:
   - **Security group name:** `Web Security Group`
   - **Description:** `Enable HTTP access`
   - **VPC:** choose the X to remove the currently selected VPC, then from the drop down list choose **lab-vpc**
3. In the **Inbound rules** pane, choose **Add rule**
4. Configure the following settings:
   - **Type:** *HTTP*
   - **Source:** *Anywhere-IPv4*
   - **Description:** `Permit web requests`
5. Scroll to the bottom of the page and choose **Create security group**

   You will use this security group in the next task when launching an Amazon EC2 instance.

Task 4: Launch a Web Server Instance

In this task, you will launch an Amazon EC2 instance into the new VPC. You will configure the instance to act as a web server.

1. In the search box to the right of **Services**, search for and choose **EC2** to open the EC2 console.
2. From the **Launch instance** menu choose **Launch instance**.
3. Name the instance:
    - Give it the name `Web Server 1`

      When you name your instance, AWS creates a tag and associates it with the instance. A tag is a key value pair. The key for this pair is ***Name***, and the value is the name you enter for your EC2 instance.
4. Choose an AMI from which to create the instance:
    - In the list of available *Quick Start* AMIs, keep the default **Amazon Linux** selected.
    - Also keep the default **Amazon Linux 2023 AMI** selected.

      The type of *Amazon Machine Image (AMI)* you choose determines the Operating System that will run on the EC2 instance that you launch.
5. Choose an Instance type:
    - In the *Instance type* panel, keep the default **t2.micro** selected.

      The *Instance Type* defines the hardware resources assigned to the instance.
6. Select the key pair to associate with the instance:
    - From the **Key pair name** menu, select **vockey**.

      The vockey key pair you selected will allow you to connect to this instance via SSH after it has launched. Although you will not need to do that in this lab, it is still required to identify an existing key pair, or create a new one, when you launch an instance.
7. Configure the Network settings:
    - Next to Network settings, choose **Edit**, then configure:
        - **Network:** *lab-vpc*

- **Subnet:** *lab-subnet-public2* (*not* Private!)
- **Auto-assign public IP:** *Enable*

- Next, you will configure the instance to use the *Web Security Group* that you created earlier.
  - Under Firewall (security groups), choose **Select existing security group**.
  - For **Common security groups**, select **Web Security Group**.

    This security group will permit HTTP access to the instance.

8. In the *Configure storage* section, keep the default settings.

   Note: The default settings specify that the *root volume* of the instance, which will host the Amazon Linux guest operating system that you specified earlier, will run on a general purpose SSD (*gp3*) hard drive that is 8 GiB in size. You could alternatively add more storage volumes, however that is not needed in this lab.

9. Configure a script to run on the instance when it launches:
   - Expand the **Advanced details** panel.
   - Scroll to the bottom of the page and then copy and paste the code shown below into the **User data** box:

     ```
     #!/bin/bash
     # Install Apache Web Server and PHP
     dnf install -y httpd wget php mariadb105-server
     # Download Lab files
     wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.co
     m/CUR-TF-100-ACCLFO-2/2-lab2-vpc/s3/lab-app.zip
     unzip lab-app.zip -d /var/www/html/
     # Turn on web server
     chkconfig httpd on
     service httpd start
     ```

     This script will run with root user permissions on the guest OS of the instance. It will run automatically when the instance launches for the first time. The script installs a web server, a database, and PHP libraries, and then it downloads and installs a PHP web application on the web server.

10. At the bottom of the **Summary** panel on the right side of the screen choose **Launch instance**

    You will see a Success message.

11. Choose **View all instances**

12. Wait until **Web Server 1** shows *2/2 checks passed* in the **Status check** column.

    This may take a few minutes. Choose the refresh icon at the top of the page every 30 seconds or so to more quickly become aware of the latest status of the instance.

    You will now connect to the web server running on the EC2 instance.

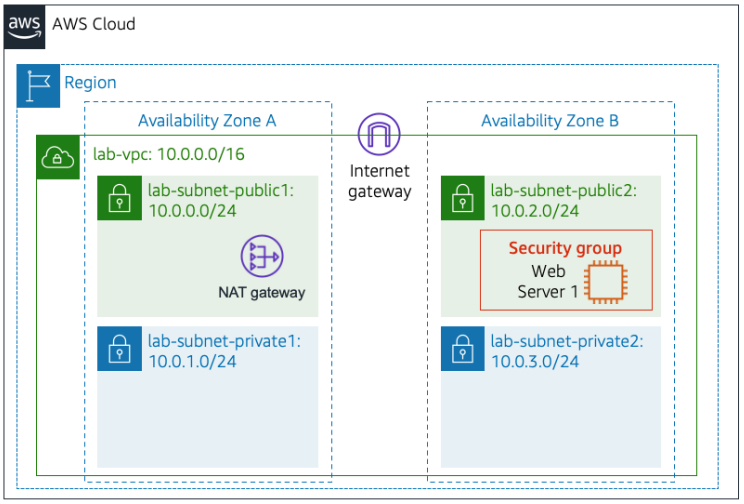
13. Select **Web Server 1**.

14. Copy the **Public IPv4 DNS** value shown in the **Details** tab at the bottom of the page.

15. Open a new web browser tab, paste the **Public DNS** value and press Enter.

    You should see a web page displaying the AWS logo and instance meta-data values.

    The complete architecture you deployed is:



Public Route Table

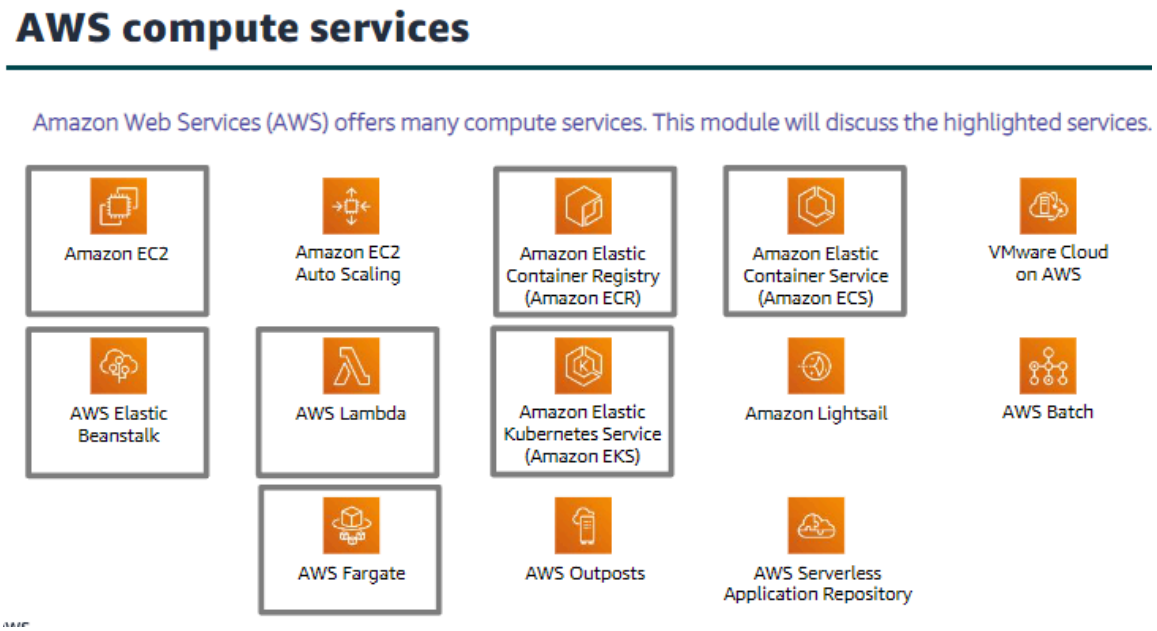
Destination	Target
10.0.0.0/16	local
0.0.0.0/0	Internet gateway

Private Route Tables

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	NAT gateway

Module 6: Compute

Section 1: Compute services and Overview



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 Beanstalkprovides 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 Lightsailprovides 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 Repositoryprovides a way to discover, deploy, and publish serverless applications.

Categorizing compute services

Services	Key Concepts	Characteristics	Ease of Use
• Amazon EC2	• Infrastructure as a service (IaaS) • Instance-based • Virtual machines	• Provision virtual machines that you can manage as you choose	A familiar concept to many IT professionals.
• AWS Lambda	• Serverless 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	• Container-based 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 web applications	• 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.

Amazon EC2provides 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

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

Section 2: Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2)

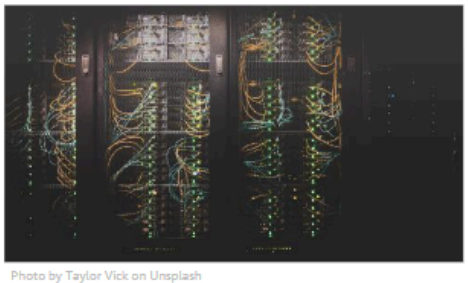
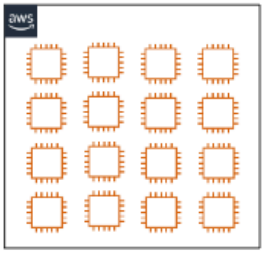


Photo by Taylor Vick on Unsplash

On-premises servers

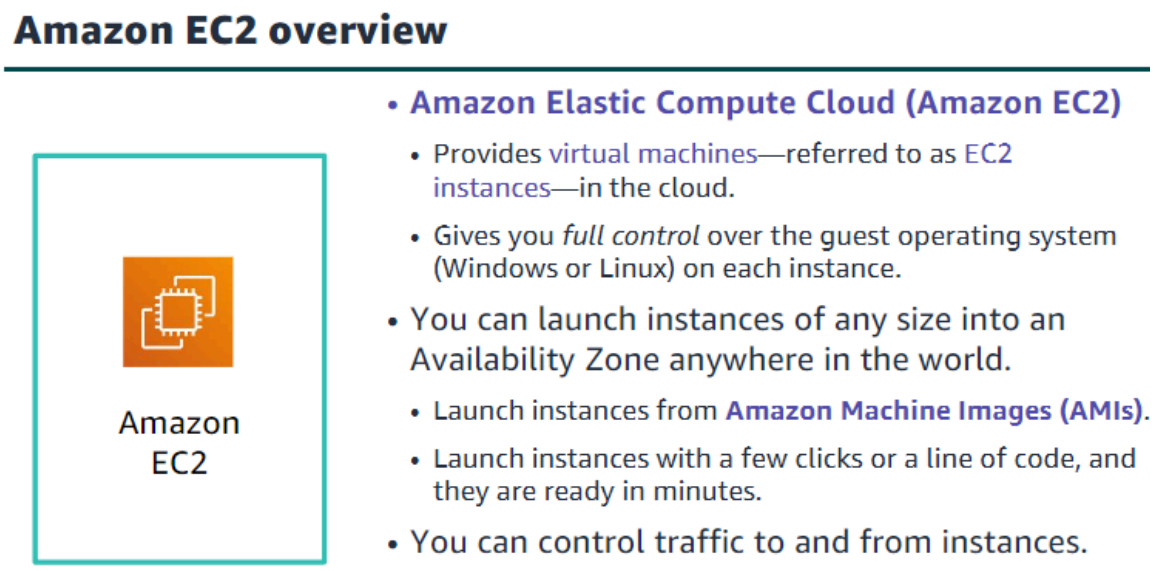
Example uses of Amazon EC2 instances

- ✓ Application server
- ✓ Web server
- ✓ Database server
- ✓ Game server
- ✓ Mail server
- ✓ Media server
- ✓ Catalog server
- ✓ File server
- ✓ Computing server
- ✓ Proxy server



Amazon EC2 instances

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



The EC2in 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

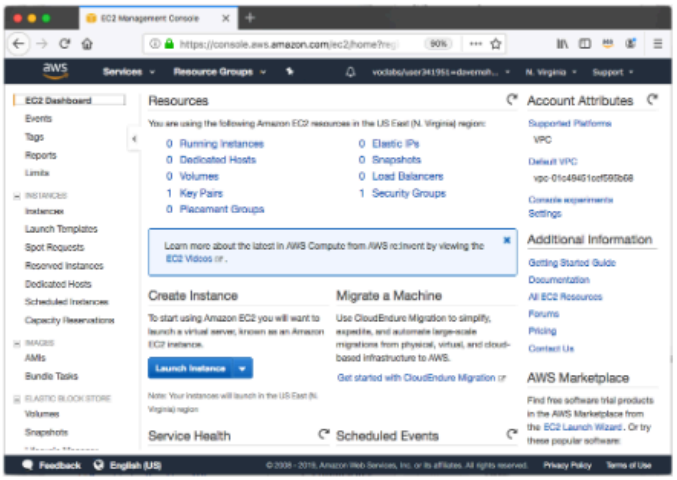
control over the Windows or Linux operating system that runs on the instance. Most server operating systems are supported, including: Windows 2008, 2012, 2016, and 2019, Red Hat, SuSE, Ubuntu, and Amazon Linux.An operating system that runs on a virtual machine is often called a guest operating system 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.

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.

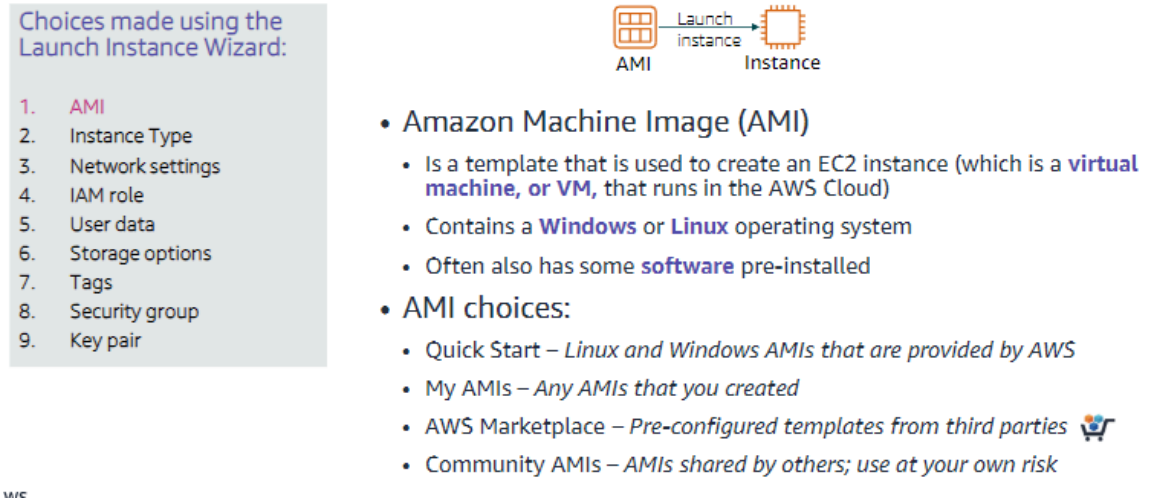
Launching an Amazon EC2 instance

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.

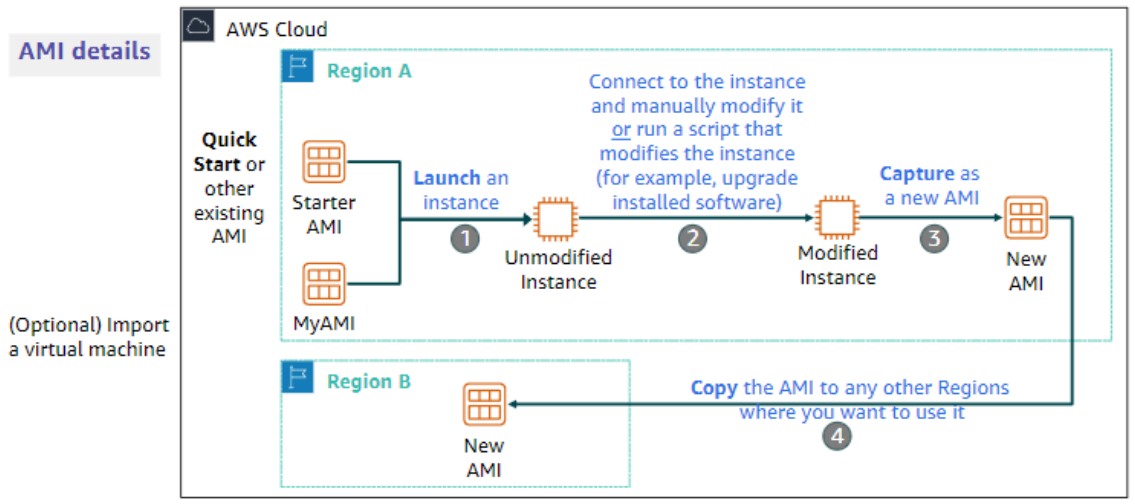


1. Select an AMI



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 importa 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.

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*

Amazon EC2 provides a selection of instance types that 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 naming

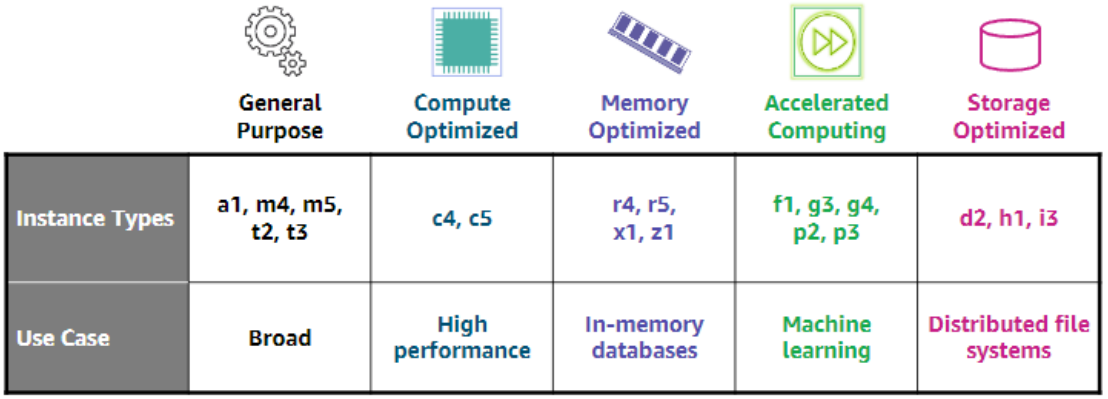
- Example: **t3.large**
  - T is the family name
  - 3 is the generation number
  - Large is the size

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

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



	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

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.•C5instances 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.•R5instances 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.

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.

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 useplacement groupsto influence the placement of a group of interdependentinstances 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.

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.

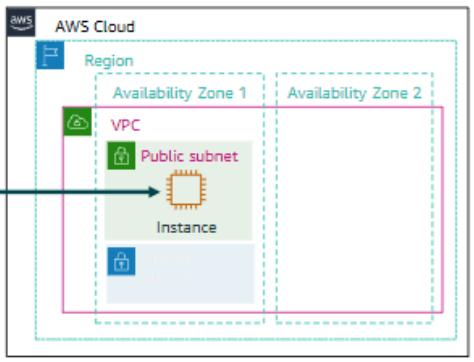
3. Specify network settings

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

Example: specify to deploy the instance here



IWS

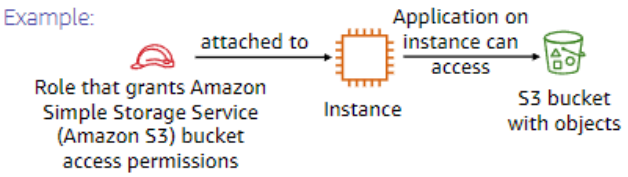
The choice of Regionmust be made before you start the Launch Instance Wizard. Verify that you are in the correct Region page of the Amazon EC2 console before you choose Launch Instance.When you launch an instance in a default VPC, AWS will assign it a public IP address by default. When you launch an 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.

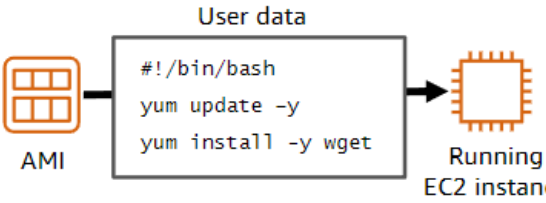


It is common to use EC2 instances to run an application that must make secure API calls to other AWS services. To support these use cases, AWS enables you to attach an AWS Identity and Access Management (IAM) role to an EC2 instance. Without this feature, you might be tempted to place AWS credentials 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 profileis 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.

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

When the EC2instance 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-initservice. On Windows instances, it is run by the EC2Configor EC2Launchutility. 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

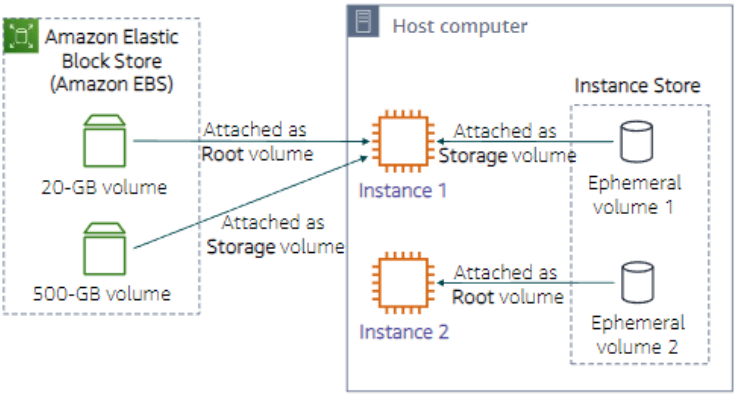
Amazon EC2 storage options

- **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 storageservice 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 EC2Instance Storeprovides 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)?



The Instance 1example 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 2example 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.

7. Add tags

Choices made by using the Launch Instance Wizard:

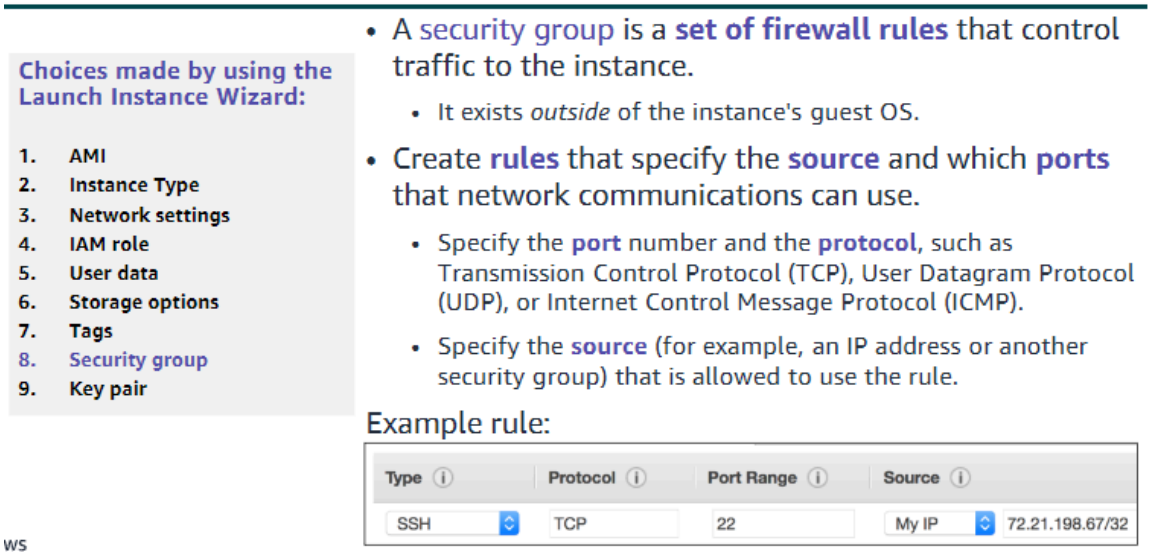
1. AMI
2. Instance Type
3. Network settings
4. IAM role
5. User data
6. Storage options
7. Tags
8. Security group
9. Key pair

- 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:

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



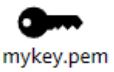
A security groupacts as a virtual firewall that controls network traffic for one or more instances. When you launch an instance, you can specify one or more security groups; otherwise, the default security group is used. You can add rulesto each security group. Rules allow traffic to or from its associated 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 sourcecan 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 outboundtraffic. You can remove the ruleand add outbound rules that only allow specific outbound traffic. If your security group has no outbound rules, no outboundtraffic that originates from your instance is allowed.

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.

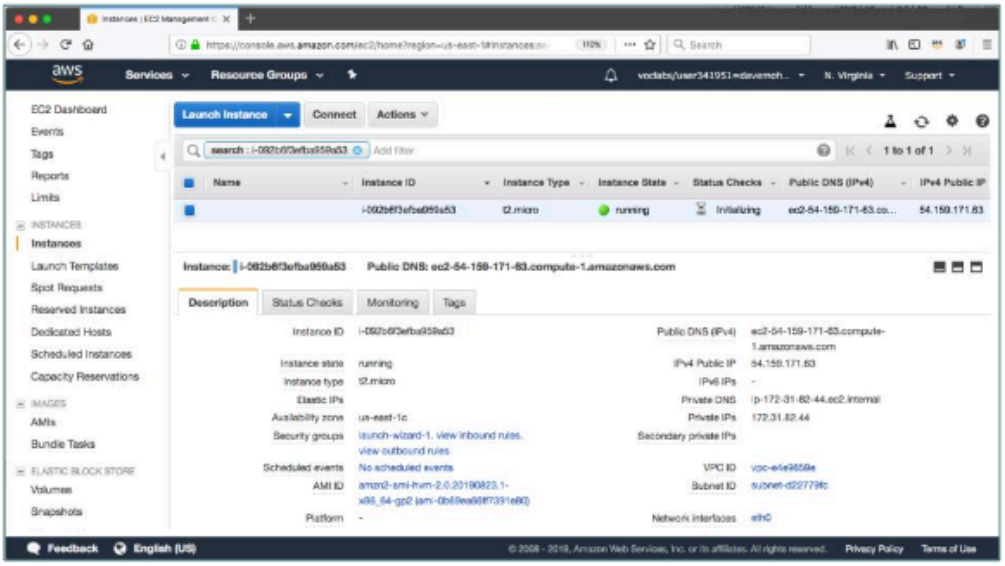


- It enables secure connections to the instance.

- 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.

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 akey 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 Windowsinstance, 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 Linuxinstances, 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



Another option: Launch an EC2 instance with the AWS Command Line Interface

- EC2 instances can also be created programmatically.

AWS Command Line Interface (AWS CLI)

- 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.

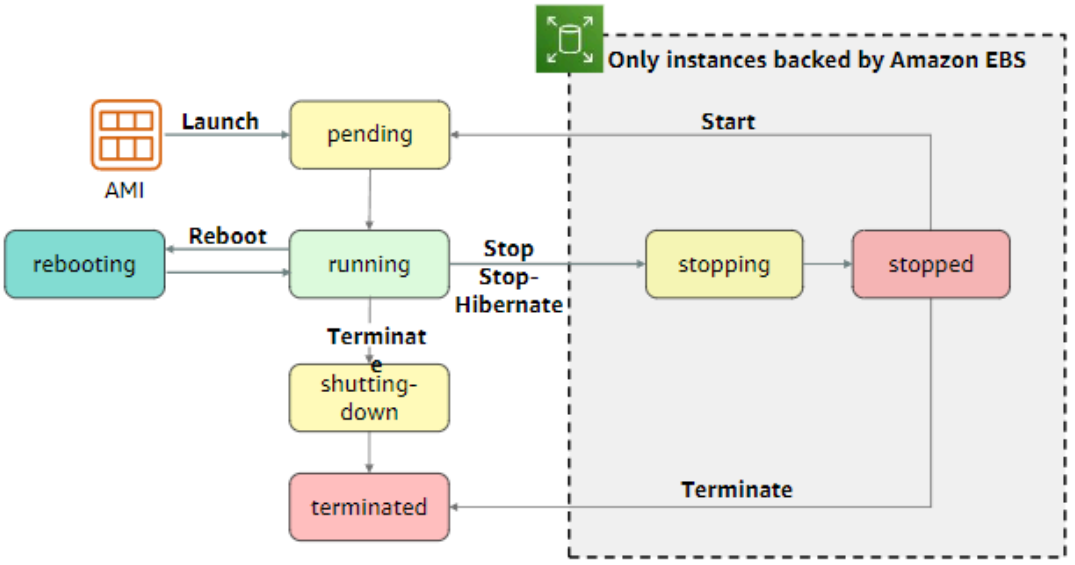
Example command:

```
aws ec2 run-instances \
--image-id ami-1a2b3c4d \
--count 1 \
--instance-type c3.large \
--key-name MyKeyPair \
--security-groups MySecurityGroup \
--region us-east-1
```

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

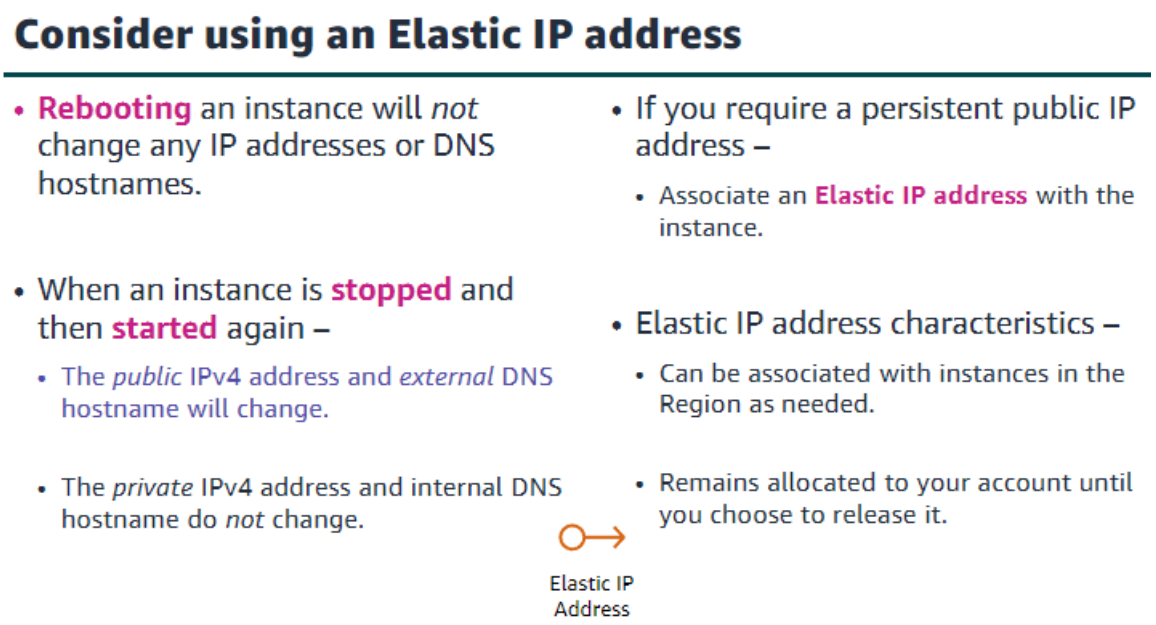
the Region where the AWS CLI will find the AMI and launch the EC2 instance.The command should successfully create an EC2 instance if:•The command is properly formed•The resources that the command needs already exist•You have sufficient permissions to run the command•You have sufficient capacity in the 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



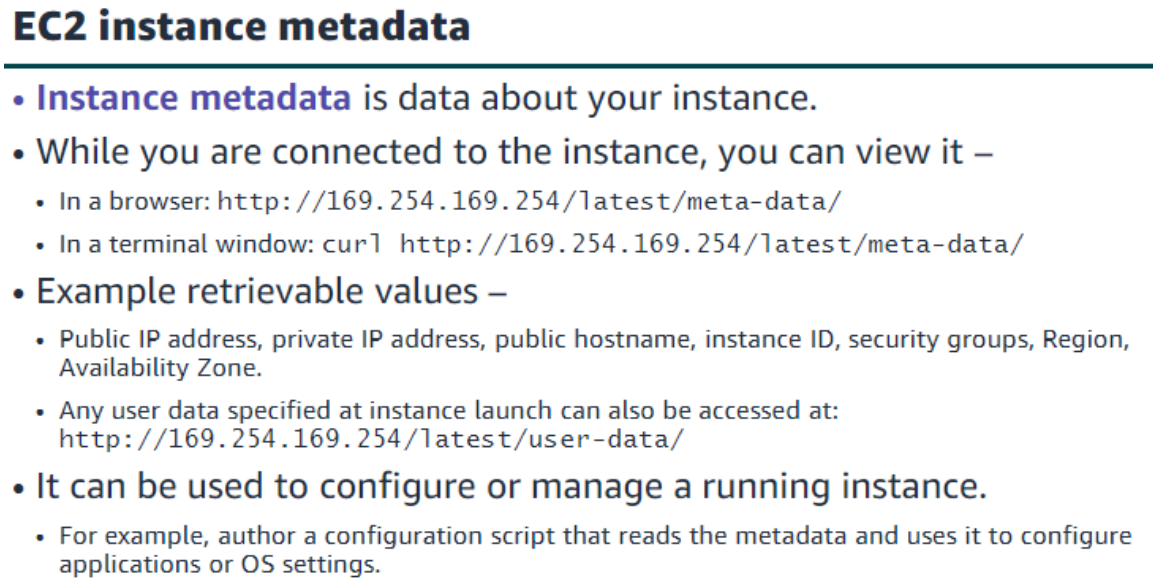
The arrows show actionsthat you can take and the boxes show the statethe instance will enter after that action. An instance can be in one of the following states:•Pending–When an instance is first launched from an AMI, or when you start a stopped instance, it enters the pending state when the instance 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.



A public IP addressis 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 beec2-203-0-113-25.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

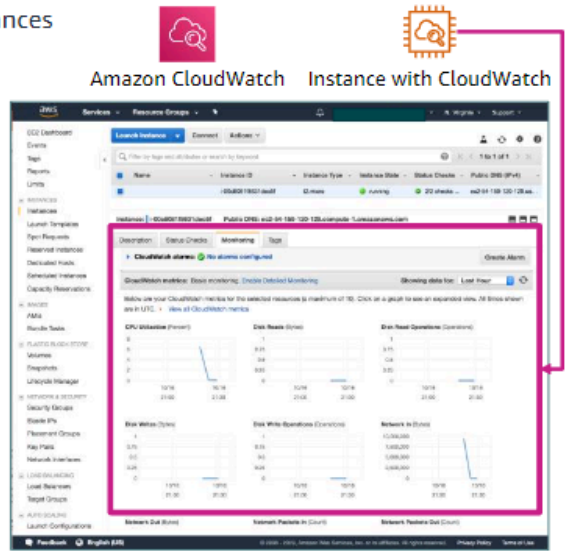
public (IPv4) internet addresses are a scarce public resource. However, this is a soft limit, and you can request a limit increase (which might be approved).



Instance metadata is data about your instance. You can view it while you are connected to the instance. To access it in a browser, go to thefollowing URL: http://169.254.169.254/latest/meta-data/. The data can also be read programmatically, such as from a terminal window thathas the cURL utility.In the terminal window,run curl http://169.254.169.254/latest/meta-data/to retrieve it. The IP address 169.254.169.254is 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

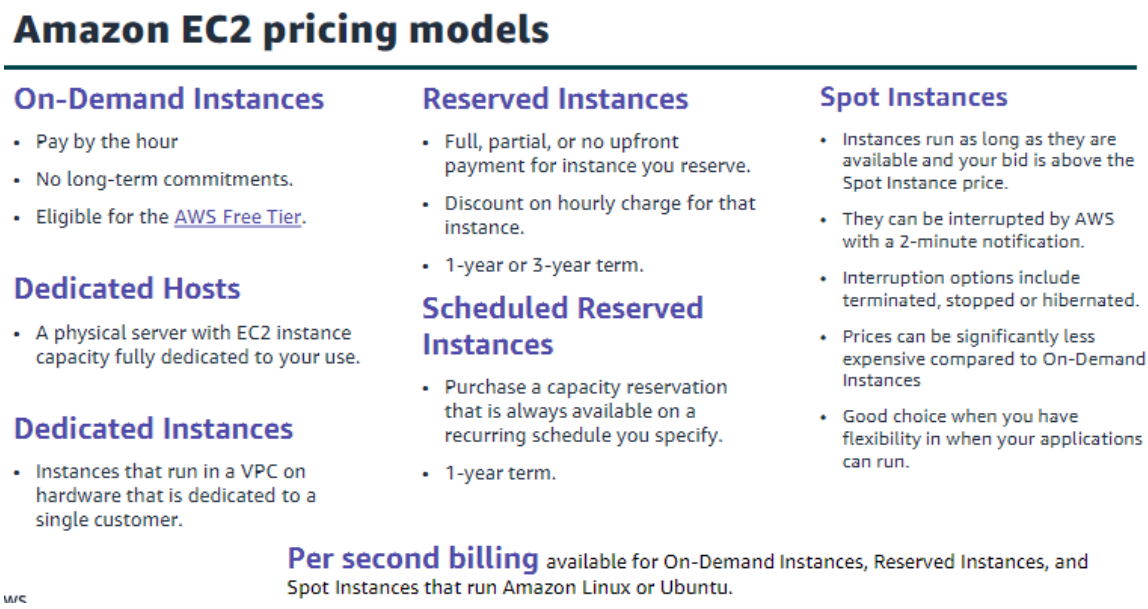
- Use **Amazon CloudWatch** to monitor EC2 instances
  - Provides near-real-time metrics
  - Provides charts in the Amazon EC2 console **Monitoring** tab that you can view
  - Maintains 15 months of historical data
- **Basic monitoring**
  - Default, no additional cost
  - Metric data sent to CloudWatch every 5 minutes
- **Detailed monitoring**
  - Fixed monthly rate for seven pre-selected metrics
  - Metric data delivered every 1 minute



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.

Section 3: Amazon EC2 Cost Optimization



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-DemandInstances are eligible for the AWS Free Tier (https://aws.amazon.com/free/). 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
• Low cost and flexibility	• Large scale, dynamic workload	• Predictability ensures compute capacity is available when needed	• Save money on licensing costs • Help meet compliance and regulatory requirements

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.

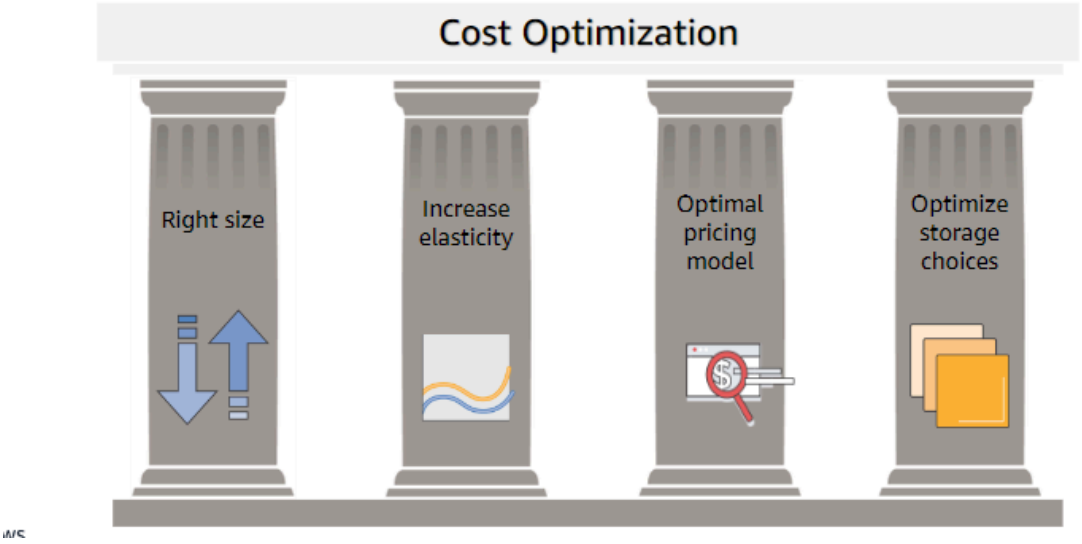
Amazon EC2 pricing models: Use cases

On-Demand Instances	Spot Instances	Reserved Instances	Dedicated Hosts
• Short-term, spiky, or unpredictable workloads • Application development or testing	• Applications with flexible start and end times • Applications only feasible at very low compute prices • Users with urgent computing needs for large amounts of additional capacity	• Steady state or predictable usage workloads • Applications that require reserved capacity, including disaster recovery • Users able to make upfront payments to reduce total computing costs even further	• Bring your own license (BYOL) • Compliance and regulatory restrictions • Usage and licensing tracking • Control instance placement

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

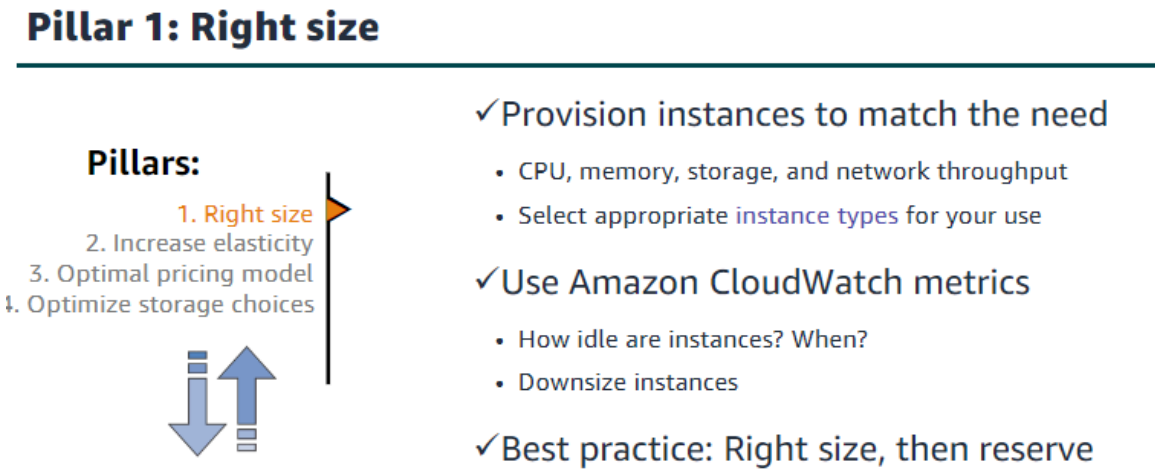
cases.Spot Instances are a good choice if your applications can tolerate interruption with a 2-minute warning notification. By default, instances are terminated, but you can configure them to stop or hibernate instead. Common use cases include fault-tolerant applications such as web servers, API backends, 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



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



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:•Selectthe cheapest instance available that still meets your performance requirements. •ReviewCPU, 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.•UseAmazon 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

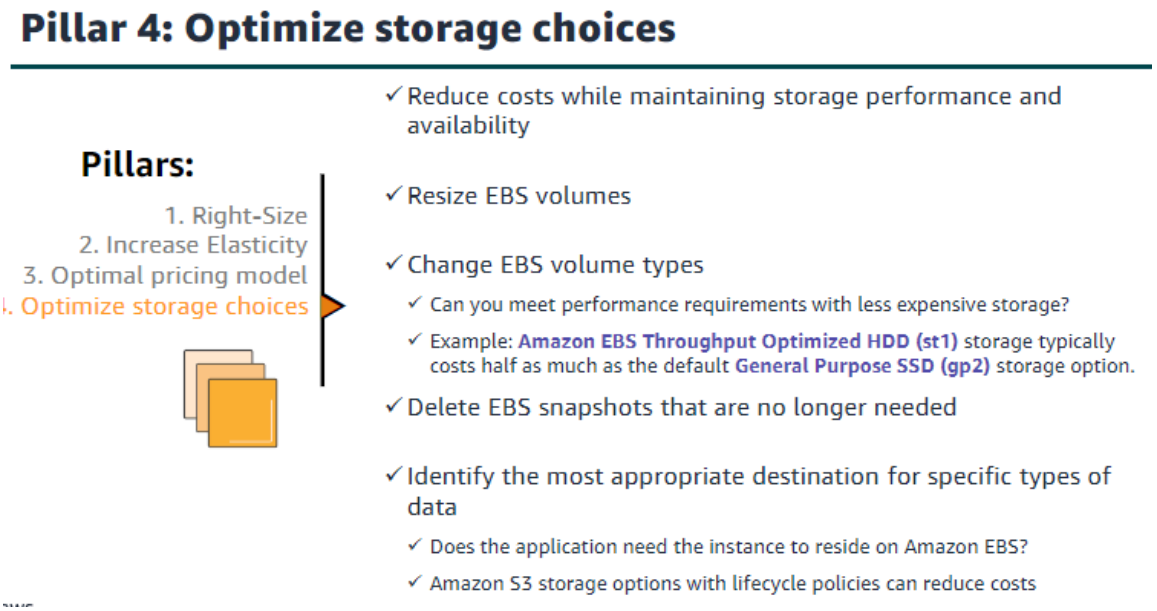
✓Optimize and *combine* purchase types

✓Examples:

- Use **On-Demand Instance** 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.



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.

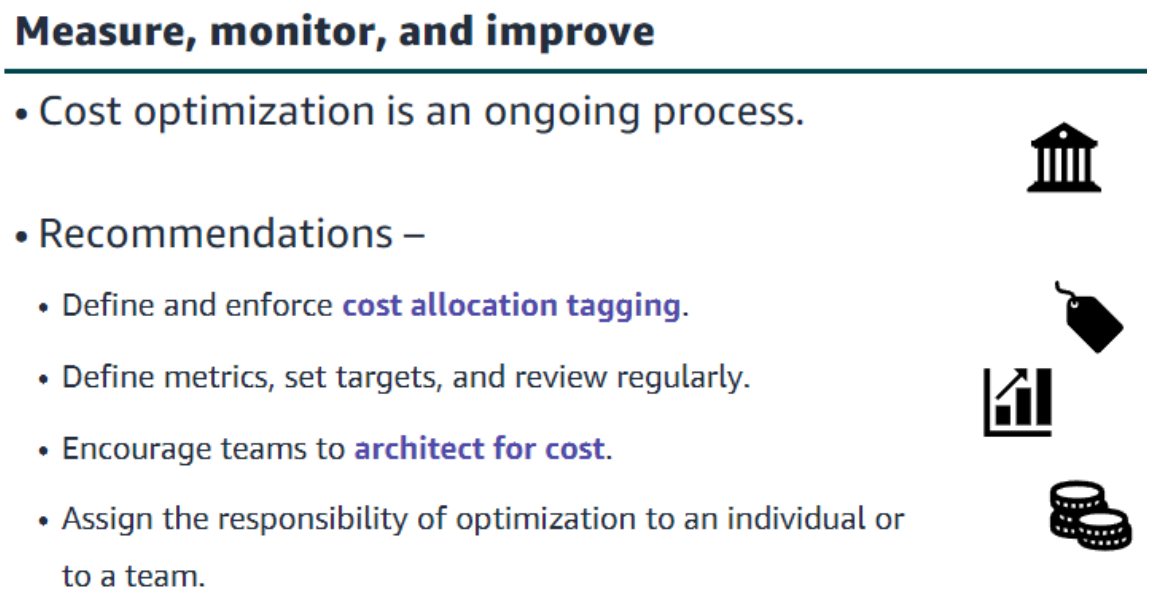
One way you can accomplish this is by resizing EBS volumes. For example, if you originally provisioned a 500-GB volume for an EC2 instance that will only need a maximum of 20 GB of storage space, you can reduce the size of the volume and save on costs.

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.

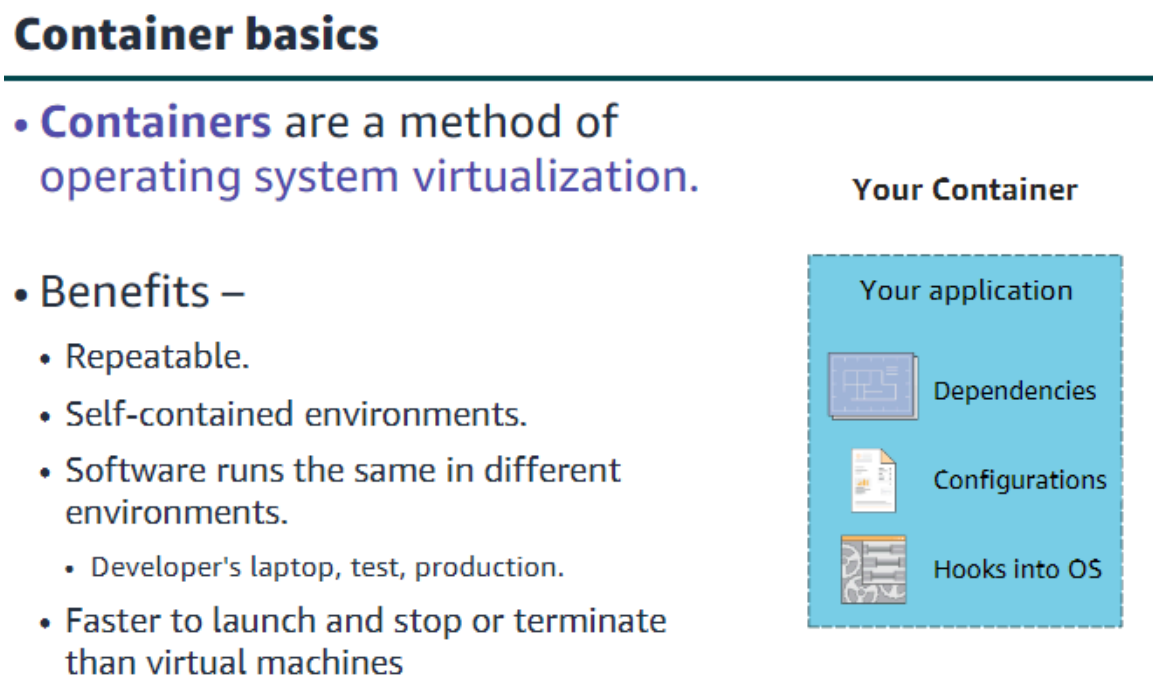
Finally, try to identify the most appropriate destination for specific types of data. Does your application need the data it uses to reside on Amazon EBS? Would the application run equally as well if it used Amazon S3 for storage instead? Configuring data lifecycle policies can also reduce costs. For example, you might automate the migration of older infrequently accessed data to cheaper storage locations, such as Amazon Simple Storage Service Glacier.



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.Tagginghelps 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 4: Container Services

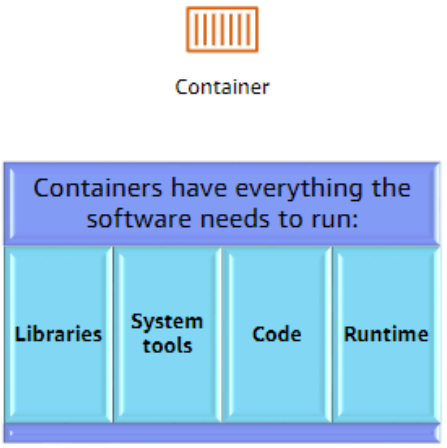


Containers are a method of operating system virtualization that enables you to run an application and its dependencies in resource-isolated processes. By using containers, you can easily package an application's code, configurations, and dependencies into easy-to-use building blocks that deliver environmental 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.

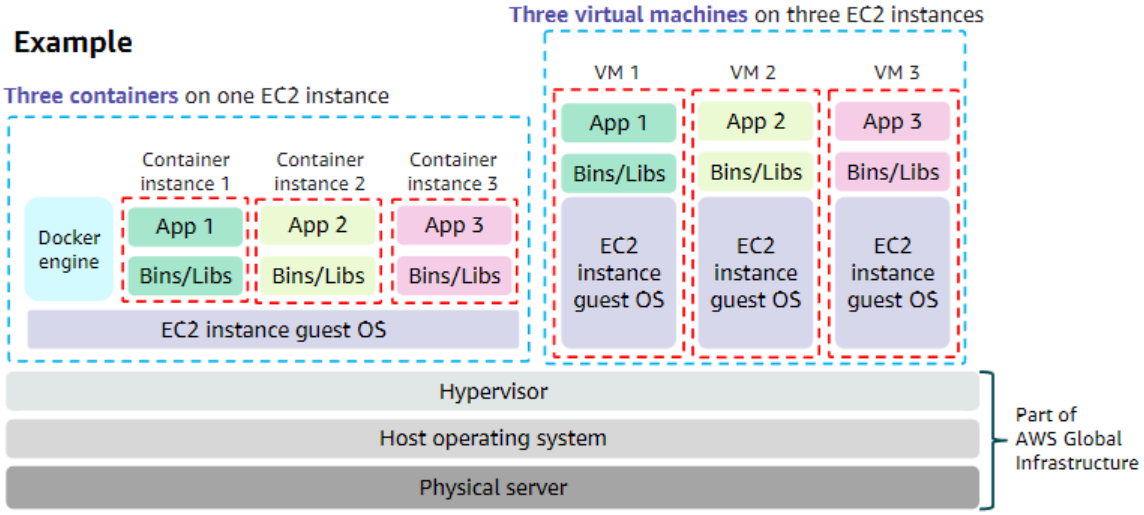
What is Docker?

- **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.



Dockeris 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



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.

Amazon Elastic Container Service (Amazon ECS)

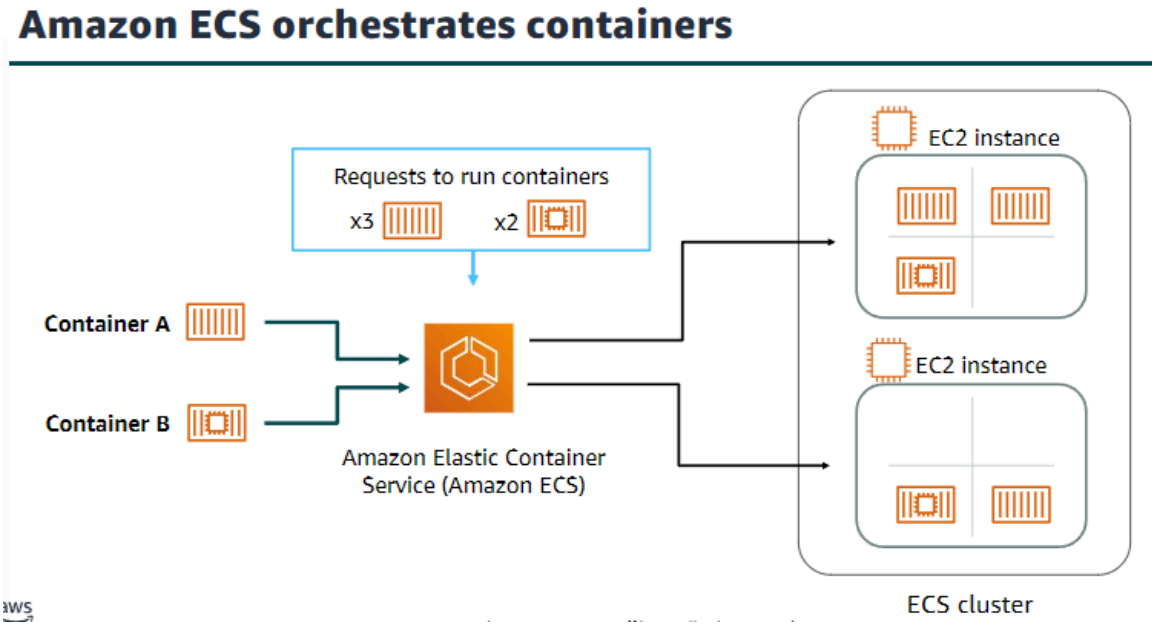
- Amazon Elastic Container Service (**Amazon ECS**) –
  - A highly scalable, fast, container management service

- 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 (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

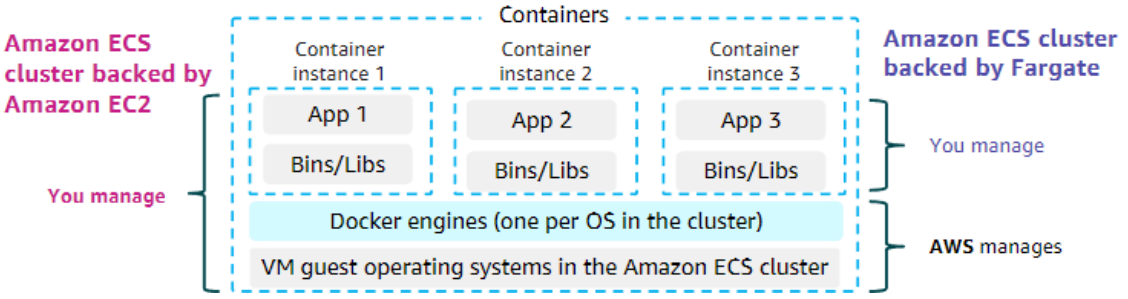
include the ability to: •Launchup to tens of thousands of Docker containers in seconds•Monitorcontainerdeployment•Managethe state of the cluster that runs the containers•Schedule containers byusing a built-in scheduler or a third-party scheduler (for example, Apache Mesosor Blox)Amazon ECS clusters can also use Spot Instances and Reserved Instances.



Create atask definitionwhich 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. Ataskis 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 scheduleris 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 ECScluster. 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)



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 lone 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.

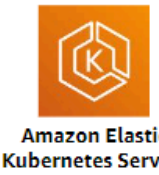
What is Kubernetes?

- 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.

Kubernetes is opensource software for container orchestration. Kubernetes can work with many containerization technologies, including Docker. Because it isapopular 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 applicationsat scale.With Kubernetes, you canrun any type of containerized application by using the same toolset in both on-premises data centers and the cloud. Kubernetes manages a clusterof 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. Containersare 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.

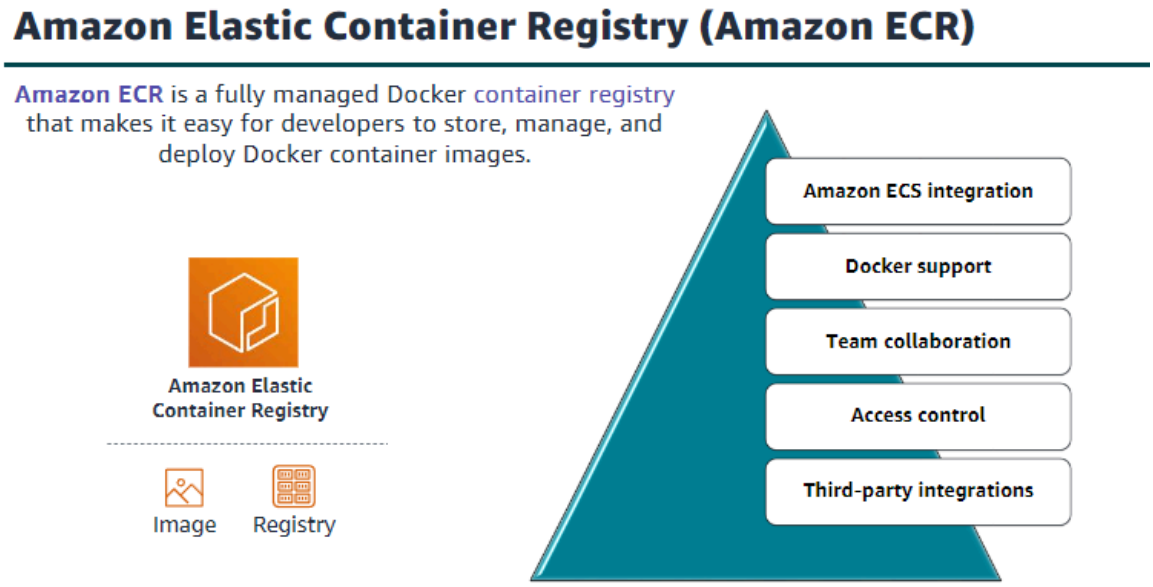
Amazon Elastic Kubernetes Service (Amazon EKS)

- 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

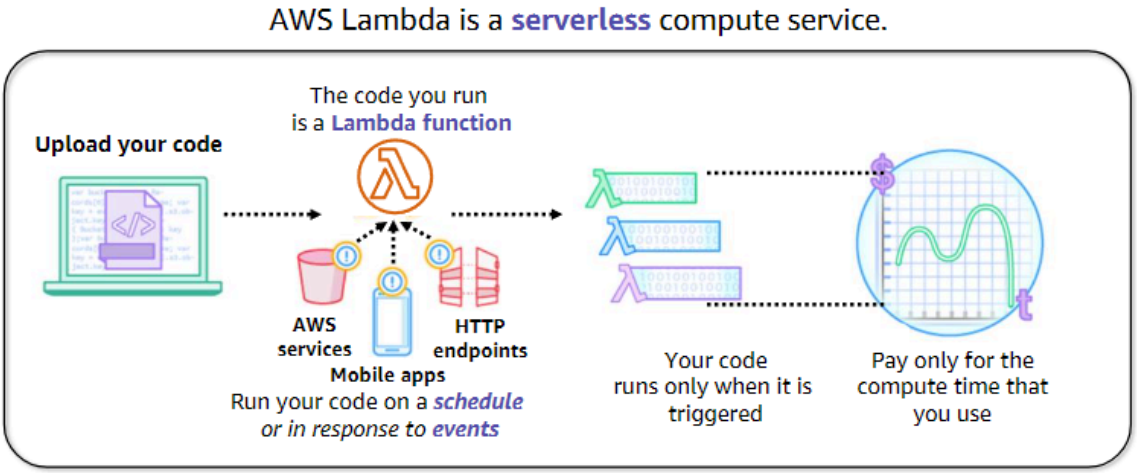
Amazon Elastic Kubernetes Service (Amazon EKS) is a managed Kubernetes service that makes it easy for you to run Kubernetes on AWS without needing to install, operate, and maintain your own Kubernetes control plane. It is certified Kubernetes conformant, so existing applications that run on upstream 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.



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.

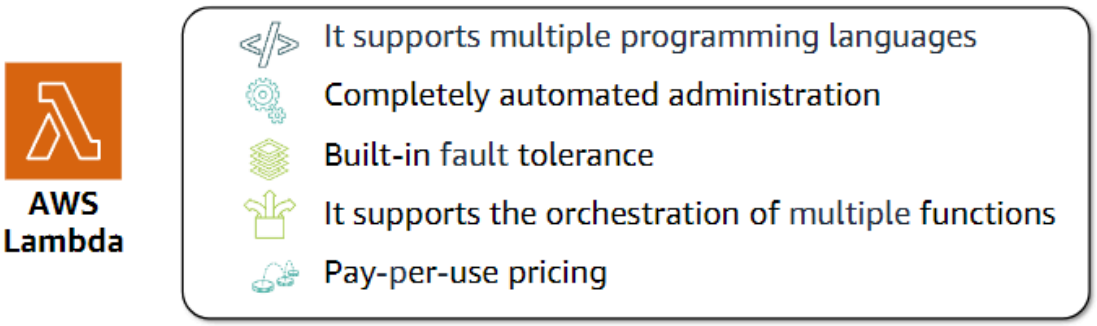
Section 5: Introduction to AWS Lambda

AWS Lambda: Run code without servers



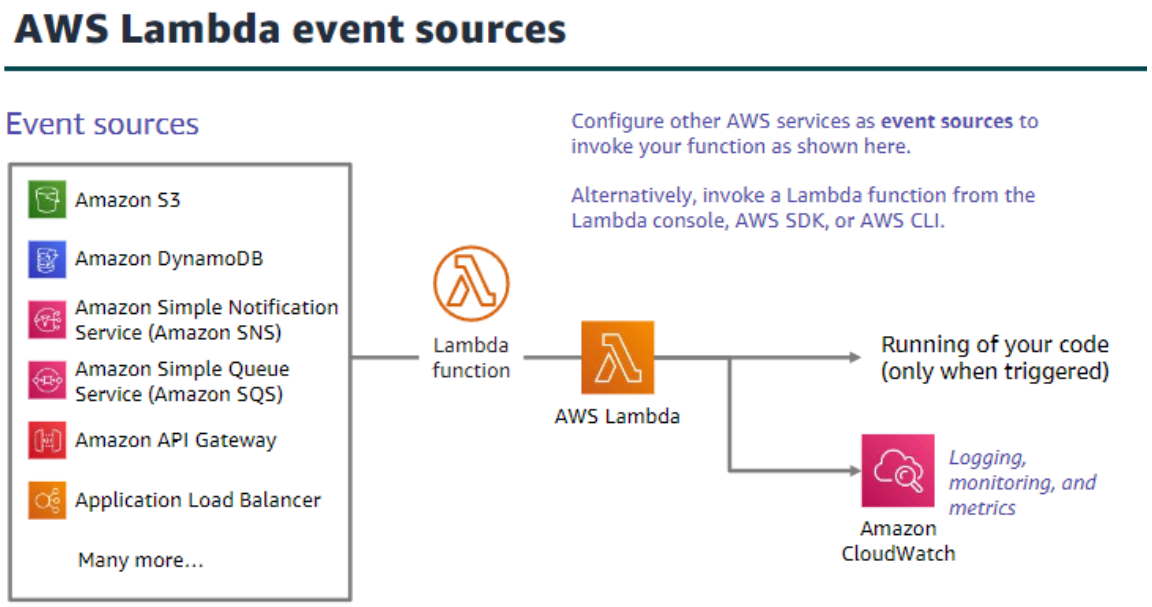
AWS Lambda is an event-driven, serverless compute service. Lambda enables you to run code without provisioning or managing servers. You create a Lambda function, which is the AWS resource that contains the code that you upload. You then set the Lambda function to be triggered, either on a scheduled basis 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



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

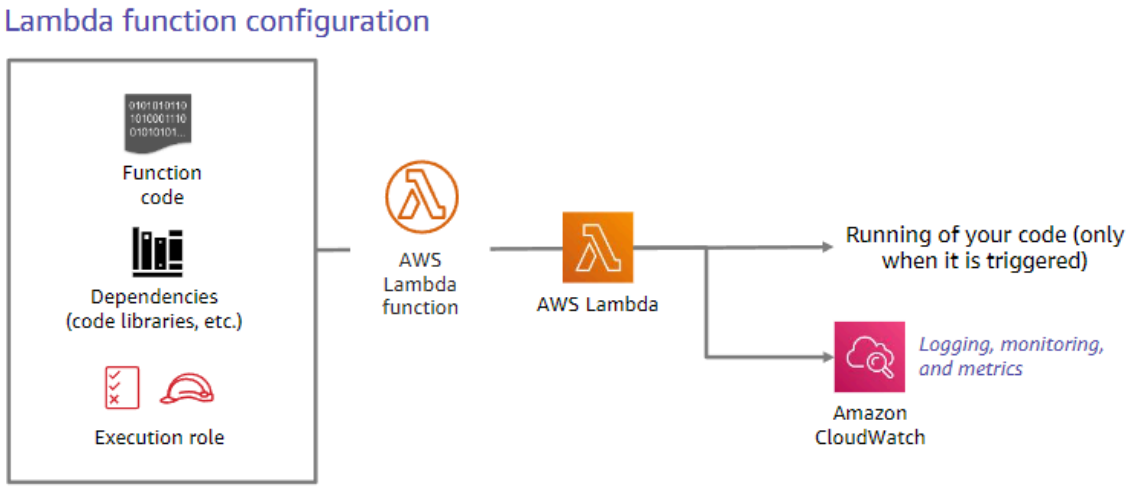
code; does all the administration, maintenance, and security patches; and provides built-in logging and monitoring throughAmazon CloudWatch.Lambda provides built-in fault tolerance. It maintains compute capacity across multiple Availability Zones in each Region to help protect your code against individual 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 withAWS 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.



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 asynchronouslyinclude, 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 at https://docs.aws.amazon.com/lambda/latest/dg/lambda-services.htmlfor 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.

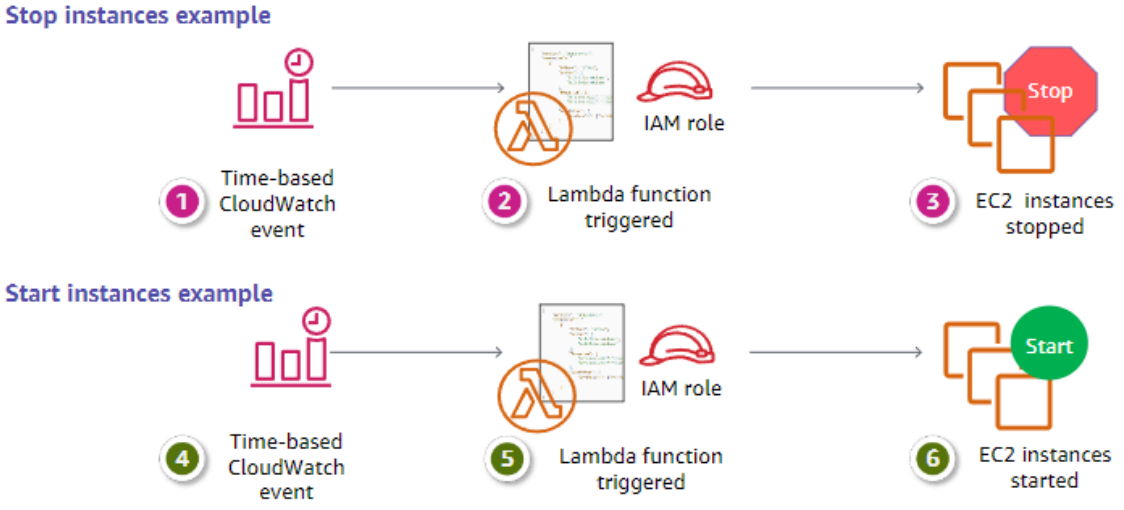
AWS Lambda function configuration



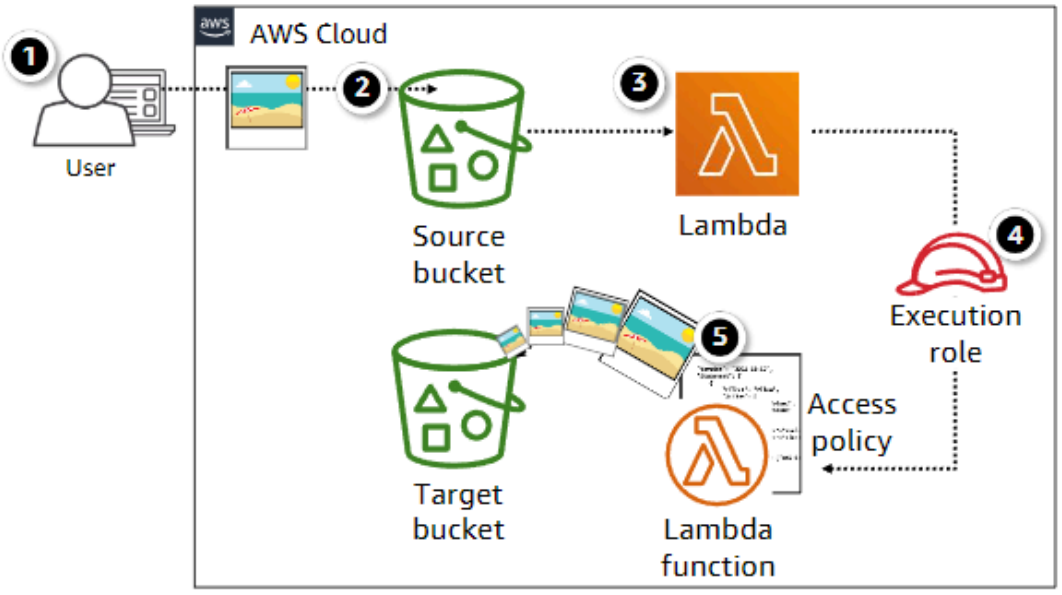
a Lambda function is the custom code that you write to process events, and that Lambda runs the Lambda function on your behalf. When you use the AWS Management Console to create a Lambda function, you first give the function a name. Then, you specify:•The runtime environment the function will use (for 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

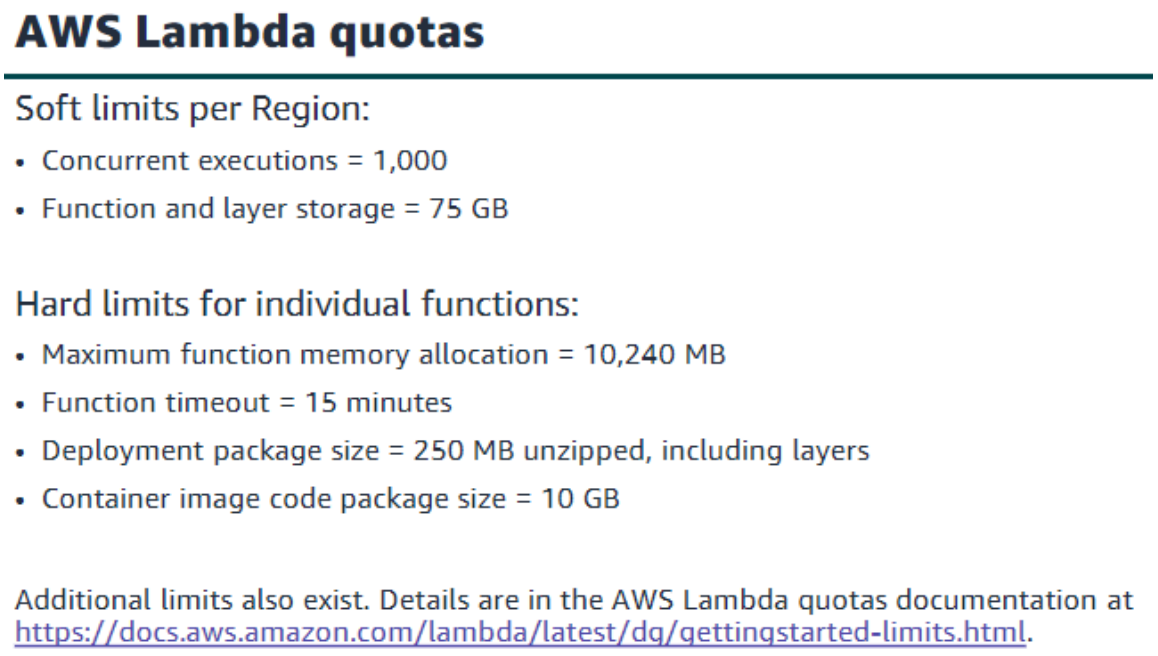
(use the provided code editor or upload a file that contains your code)•Specify the memoryin MB to allocate to your function (128 MB to 10,240 MB)•Optionally specify environment variables, description, timeout, the specific virtual private cloud (VPC) to run the function in, tags you would like to use, and other settings. For more information, see Configuring functions in the AWS Lambda console https://docs.aws.amazon.com/lambda/latest/dg/configuration-console.htmlin the AWS Documentation.

Schedule-based Lambda function example: Start and stop EC2 instances



Event-based Lambda function example: Create thumbnail images





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 layeris 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 deploymentpackage. 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 at https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-limits.html.

Section 6: Introduction to AWS Elastic Beanstalk

AWS Elastic Beanstalk

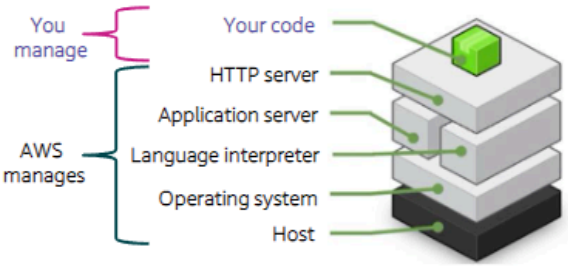
AWS Elastic Beanstalk

- 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)



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.

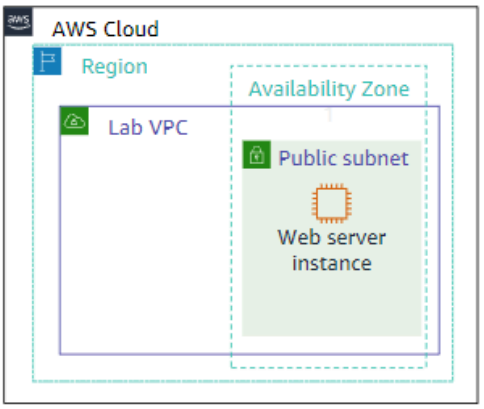
Additional resources

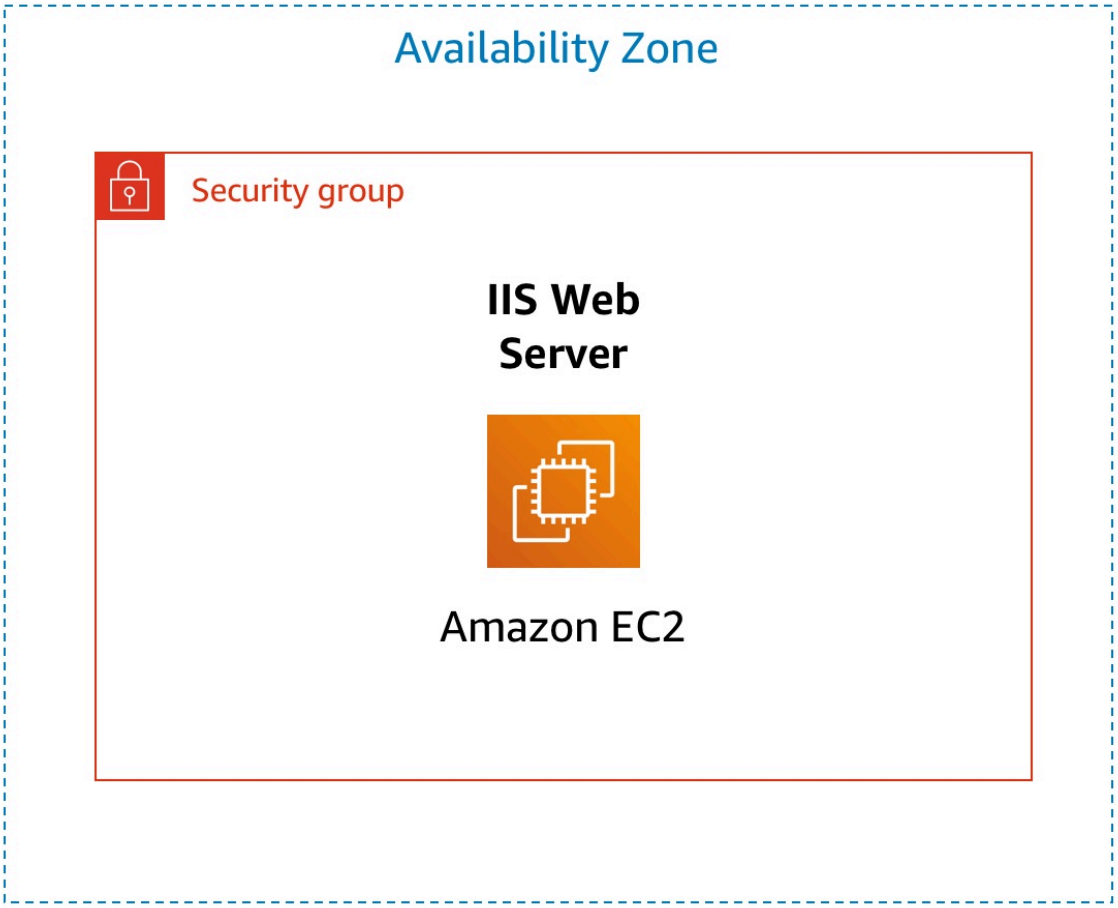
- Amazon EC2 Documentation: https://docs.aws.amazon.com/ec2/
- Amazon EC2 Pricing: https://aws.amazon.com/ec2/pricing/
- Amazon ECS Workshop: https://ecsworkshop.com/
- Running Containers on AWS: https://containersonaws.com/
- Amazon EKS Workshop: https://www.eksworkshop.com/
- AWS Lambda Documentation: https://docs.aws.amazon.com/lambda/
- AWS Elastic Beanstalk Documentation: https://docs.aws.amazon.com/elastic-beanstalk/
- Cost Optimization Playbook: https://d1.awsstatic.com/pricing/AWS_CO_Playbook_Final.pdf

Lab Activity 3:

Lab 3 scenario

In this lab, you will launch and configure your first virtual machine that runs on Amazon EC2.





This lab provides you with a basic overview of launching, resizing, managing, and monitoring an Amazon EC2 instance.

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make

web-scale cloud computing easier for developers.

Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.

Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios.

After completing this lab, you should be able to do the following:

- Launch a web server with termination protection enabled
- Monitor Your EC2 instance
- Modify the security group that your web server is using to allow HTTP access
- Resize your Amazon EC2 instance to scale
- Explore EC2 limits
- Test termination protection
- Terminate your EC2 instance

Duration

This lab takes approximately **35 minutes** to complete.

AWS service restrictions

In this lab environment, access to AWS services and service actions might be restricted to the ones that are needed to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that are described in this lab.

Accessing the AWS Management Console

1. At the top of these instructions, choose Start Lab to launch your lab.

   A Start Lab panel opens displaying the lab status.

2. Wait until you see the message "**Lab status: ready**", then choose the **X** to close the Start Lab panel.
3. At the top of these instructions, choose AWS

   This will open the AWS Management Console in a new browser tab. The system will automatically log you in.

   Tip: If a new browser tab does not open, there will typically be a banner or icon at the top of your browser indicating that your browser is preventing the site from opening pop-up windows. Choose on the banner or icon and choose "Allow pop ups."
4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time, to make it easier to follow the lab steps.

Task 1: Launch Your Amazon EC2 Instance

In this task, you will launch an Amazon EC2 instance with *termination protection*. Termination protection prevents you from accidentally terminating an EC2 instance. You will deploy your instance with a User Data script that will allow you to deploy a simple web server.

1. In the **AWS Management Console** choose **Services**, choose **Compute** and then choose **EC2**.

   Note: Verify that your EC2 console is currently managing resources in the **N. Virginia** (us-east-1) region. You can verify this by looking at the drop down menu at the top of the screen, to the left of your username. If it does not already indicate N. Virginia, choose the N. Virginia region from the region menu before proceeding to the next step.
2. Choose the Launch instance menu and select **Launch instance**.

Step 1: Name and tags

1. Give the instance the name `Web Server`.

   The Name you give this instance will be stored as a tag. Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type — you can quickly identify a specific resource based on the tags you have assigned to it. Each tag consists of

a Key and a Value, both of which you define. You can define multiple tags to associate with the instance if you want to.

In this case, the tag that will be created will consist of a *key* called `Name` with a *value* of `Web Server`

Step 2: Application and OS Images (Amazon Machine Image)

1. In the list of available *Quick Start* AMIs, keep the default **Amazon Linux** AMI selected.
2. Also keep the default **Amazon Linux 2023 AMI** selected.

   An **Amazon Machine Image (AMI)** provides the information required to launch an instance, which is a virtual server in the cloud. An AMI includes:

   - A template for the root volume for the instance (for example, an operating system or an application server with applications)
   - Launch permissions that control which AWS accounts can use the AMI to launch instances
   - A block device mapping that specifies the volumes to attach to the instance when it is launched

   The **Quick Start** list contains the most commonly-used AMIs. You can also create your own AMI or select an AMI from the AWS Marketplace, an online store where you can sell or buy software that runs on AWS.

Step 3: Instance type

1. In the *Instance type* panel, keep the default **t2.micro** selected.

   Amazon EC2 provides a wide selection of *instance types* optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more *instance sizes*, allowing you to scale your resources to the requirements of your target workload.

   The t2.micro instance type has 1 virtual CPU and 1 GiB of memory.

   Note: You may be restricted from using other instance types in this lab.

Step 4: Key pair (login)

1. For **Key pair name - *required***, choose **vockey**.

   Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. To ensure you will be able to log in to the guest OS of the instance you create, you identify an existing key pair or create a new key pair when launching the instance. Amazon EC2 then installs the key on the guest OS when the instance is launched. That way, when you attempt to login to the instance and you provide the private key, you will be authorized to connect to the instance.

   Note: In this lab you will not actually use the key pair you have specified to log into your instance.

Step 5: Network settings

1. Next to Network settings, choose **Edit**.
2. For **VPC**, select **Lab VPC**.

   The Lab VPC was created using an AWS CloudFormation template during the setup process of your lab. This VPC includes two public subnets in two different Availability Zones.

   Note: Keep the default subnet. This is the subnet in which the instance will run. Notice also that by default, the instance will be assigned a public IP address.
3. Under **Firewall (security groups)**, choose **Create security group** and configure:

   - **Security group name:** `Web Server security group`
   - **Description:** `Security group for my web server`

     A *security group* acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add *rules* to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group.
   - Under **Inbound security group rules**, notice that one rule exists. **Remove** this rule.

Step 6: Configure storage

1. In the *Configure storage* section, keep the default settings.

   Amazon EC2 stores data on a network-attached virtual disk called *Elastic Block Store*.

   You will launch the Amazon EC2 instance using a default 8 GiB disk volume. This will be your root volume (also known as a 'boot' volume).

Step 7: Advanced details

1. Expand **Advanced details**.
2. For **Termination protection**, select **Enable**.

   When an Amazon EC2 instance is no longer required, it can be *terminated*, which means that the instance is deleted and its resources are released. A terminated instance cannot be accessed again and the data that was on it cannot be recovered. If you want to prevent the instance from being accidentally terminated, you can enable *termination protection* for the instance, which prevents it from being terminated as long as this setting remains enabled.
3. Scroll to the bottom of the page and then copy and paste the code shown below into the **User data** box:

   ```
   #!/bin/bash
   dnf install -y httpd
   systemctl enable httpd
   systemctl start httpd
   echo '<html><h1>Hello From Your Web Server!</h1></html>' > /va
   r/www/html/index.html
   ```

   When you launch an instance, you can pass *user data* to the instance that can be used to perform automated installation and configuration tasks after the instance starts.

   Your instance is running Amazon Linux 2023. The *shell script* you have specified will run as the *root* guest OS user when the instance starts. The script will:

   - Install an Apache web server (httpd)
   - Configure the web server to automatically start on boot

- Run the Web server once it has finished installing
- Create a simple web page

Step 8: Launch the instance

1. At the bottom of the **Summary** panel on the right side of the screen choose Launch instance

   You will see a Success message.

2. Choose View all instances
   - In the Instances list, select **Web Server**.
   - Review the information displayed in the **Details** tab. It includes information about the instance type, security settings and network settings.

     The instance is assigned a *Public IPv4 DNS* that you can use to contact the instance from the Internet.

     To view more information, drag the window divider upwards.

     At first, the instance will appear in a *Pending* state, which means it is being launched. It will then change to *Initializing*, and finally to *Running*.

3. Wait for your instance to display the following:
   - **Instance State:** *Running*
   - **Status Checks:** *2/2 checks passed*

Congratulations! You have successfully launched your first Amazon EC2 instance.

Task 2: Monitor Your Instance

Monitoring is an important part of maintaining the reliability, availability, and performance of your Amazon Elastic Compute Cloud (Amazon EC2) instances and your AWS solutions.

1. Choose the **Status checks** tab.

   With instance status monitoring, you can quickly determine whether Amazon EC2 has detected any problems that might prevent your instances from running applications. Amazon EC2 performs automated

checks on every running EC2 instance to identify hardware and software issues.

Notice that both the **System reachability** and **Instance reachability** checks have passed.

2. Choose the **Monitoring** tab.

   This tab displays Amazon CloudWatch metrics for your instance. Currently, there are not many metrics to display because the instance was recently launched.

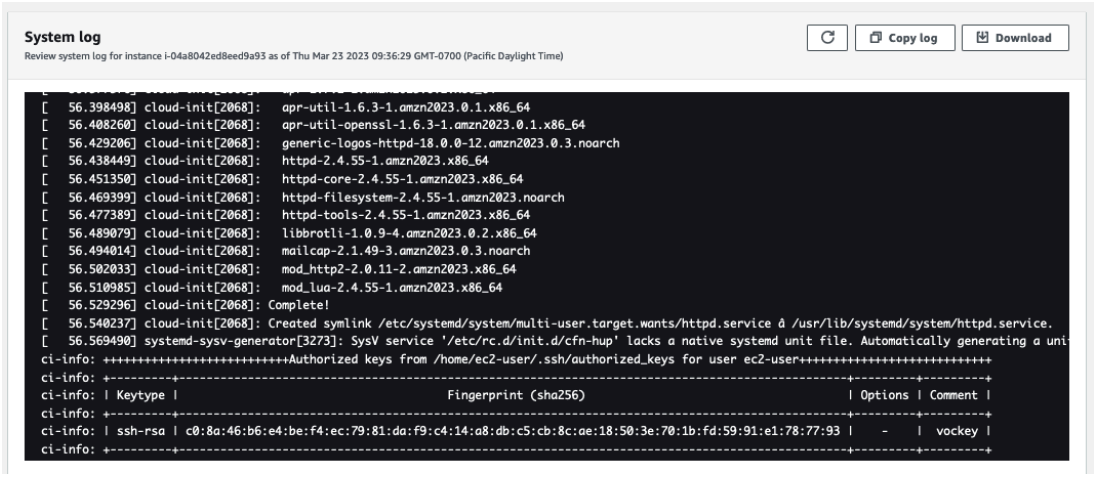
   You can choose the three dots icon in any graph and select **Enlarge** to see an expanded view of the chosen metric.

   Amazon EC2 sends metrics to Amazon CloudWatch for your EC2 instances. Basic (five-minute) monitoring is enabled by default. You can also enable detailed (one-minute) monitoring.

3. In the Actions menu towards the top of the console, select **Monitor and troubleshoot Get system log**.

   The System Log displays the console output of the instance, which is a valuable tool for problem diagnosis. It is especially useful for troubleshooting kernel problems and service configuration issues that could cause an instance to terminate or become unreachable before its SSH daemon can be started. If you do not see a system log, wait a few minutes and then try again.

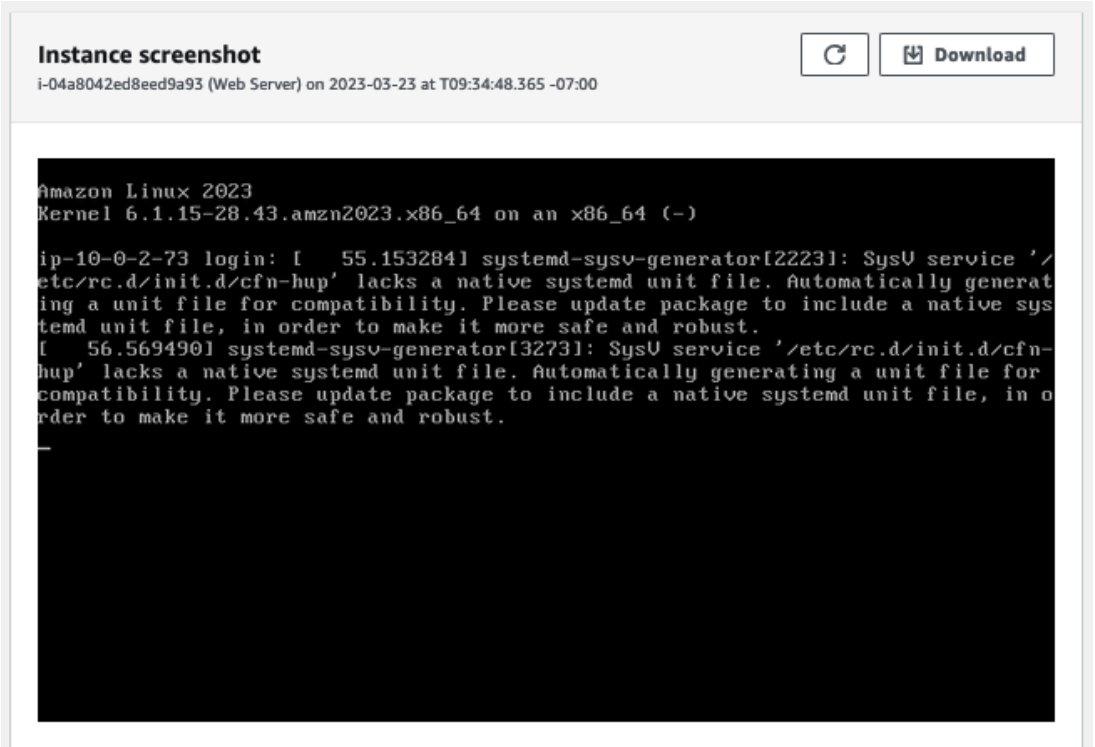
4. Scroll through the output and note that the HTTP package was installed from the **user data** that you added when you created the instance.



5. Choose **Cancel**.

6. Ensure **Web Server** is still selected. Then, in the Actions menu, select **Monitor and troubleshoot Get instance screenshot**.

   This shows you what your Amazon EC2 instance console would look like if a screen were attached to it.



   If you are unable to reach your instance via SSH or RDP, you can capture a screenshot of your instance and view it as an image. This provides visibility as to the status of the instance, and allows for quicker troubleshooting.

7. Choose **Cancel**.

   Congratulations! You have explored several ways to monitor your instance.

Task 3: Update Your Security Group and Access the Web Server

When you launched the EC2 instance, you provided a script that installed a web server and created a simple web page. In this task, you will access content from the web server.

1. Ensure **Web Server** is still selected. Choose the **Details** tab.

2. Copy the **Public IPv4 address** of your instance to your clipboard.
3. Open a new tab in your web browser, paste the IP address you just copied, then press **Enter**.

   Question: Are you able to access your web server? Why not?

   You are **not** currently able to access your web server because the *security group* is not permitting inbound traffic on port 80, which is used for HTTP web requests. This is a demonstration of using a security group as a firewall to restrict the network traffic that is allowed in and out of an instance.

   To correct this, you will now update the security group to permit web traffic on port 80.
4. Keep the browser tab open, but return to the **EC2 Console** tab.
5. In the left navigation pane, choose **Security Groups**.
6. Select **Web Server security group**.
7. Choose the **Inbound rules** tab.

   The security group currently has no inbound rules.
8. Choose Edit inbound rules, select Add rule and then configure:
   - **Type:** *HTTP*
   - **Source:** *Anywhere-IPv4*
   - Choose Save rules
9. Return to the web server tab that you previously opened and refresh the page.

   You should see the message *Hello From Your Web Server!*

   Congratulations! You have successfully modified your security group to permit HTTP traffic into your Amazon EC2 Instance.

Task 4: Resize Your Instance: Instance Type and EBS Volume

As your needs change, you might find that your instance is over-utilized (too small) or under-utilized (too large). If so, you can change the *instance type*. For example, if a *t2.micro* instance is too small for its workload, you

can change it to an *m5.medium* instance. Similarly, you can change the size of a disk.

Stop Your Instance

Before you can resize an instance, you must *stop* it.

When you stop an instance, it is shut down. There is no runtime charge for a stopped EC2 instance, but the storage charge for attached Amazon EBS volumes remains.

1. On the **EC2 Management Console**, in the left navigation pane, choose **Instances**.

   Web Server should already be selected.

2. In the Instance State menu, select **Stop instance**.

3. Choose Stop

   Your instance will perform a normal shutdown and then will stop running.

4. Wait for the **Instance state** to display: *Stopped*.

Change The Instance Type

1. In the Actions menu, select **Instance settings Change instance type**, then configure:

   - **Instance Type:** *t2.small*
   - Choose Apply

     When the instance is started again it will run as a *t2.small*, which has twice as much memory as a *t2.micro* instance. **NOTE**: You may be restricted from using other instance types in this lab.

Resize the EBS Volume

1. With the Web Server instance still selected, choose the **Storage** tab, select the name of the Volume ID, then select the checkbox next to the volume that displays.

2. In the Actions menu, select **Modify volume**.

   The disk volume currently has a size of 8 GiB. You will now increase the size of this disk.

3. Change the size to: `10` **NOTE**: You may be restricted from creating large Amazon EBS volumes in this lab.
4. Choose Modify
5. Choose Modify again to confirm and increase the size of the volume.

Start the Resized Instance

You will now start the instance again, which will now have more memory and more disk space.

1. In left navigation pane, choose **Instances**.
2. Select the **Web Server** instance.
3. In the Instance state menu, select **Start instance**.

   Congratulations! You have successfully resized your Amazon EC2 Instance. In this task you changed your instance type from *t2.micro* to *t2.small*. You also modified your root disk volume from 8 GiB to 10 GiB.

Task 5: Explore EC2 Limits

Amazon EC2 provides different resources that you can use. These resources include images, instances, volumes, and snapshots. When you create an AWS account, there are default limits on these resources on a per-region basis.

1. In the AWS Management Console, in the search box next to **Services**, search for and choose `Service Quotas`
2. Choose **AWS services** from the navigation menu and then in the AWS services *Find services* search bar, search for `ec2` and choose **Amazon Elastic Compute Cloud (Amazon EC2)**.
3. In the *Find quotas* search bar, search for `running on-demand`, but do not make a selection. Instead, observe the filtered list of service quotas that match the criteria.

   Notice that there are limits on the number and types of instances that can run in a region. For example, there is a limit on the number of *Running On-Demand Standard...* instances that you can launch in this region. When launching instances, the request must not cause your usage to exceed the instance limits currently defined in that region.

You can request an increase for many of these limits.

Task 6: Test Termination Protection

You can delete your instance when you no longer need it. This is referred to as *terminating* your instance. You cannot connect to or restart an instance after it has been terminated. In this task, you will learn how to use *termination protection*.

1. In the AWS Management Console, in the search box next to **Services**, search for and choose `EC2` to return to the EC2 console.
2. In left navigation pane, choose **Instances**.
3. Select the **Web Server** instance and in the Instance state menu, select **Terminate instance**.
4. Then choose Terminate

   Note that there is a message that says: *Failed to terminate the instance i-1234567xxx. The instance 'i-1234567xxx' may not be terminated. Modify its 'disableApiTermination' instance attribute and try again.*

   This is a safeguard to prevent the accidental termination of an instance. If you really want to terminate the instance, you will need to disable the termination protection.
5. In the Actions menu, select **Instance settings Change termination protection**.
6. Remove the check next to **Enable**.
7. Choose Save

   You can now terminate the instance.
8. Select the **Web Server** instance again and in the Instance state menu, select **Terminate instance**.
9. Choose Terminate

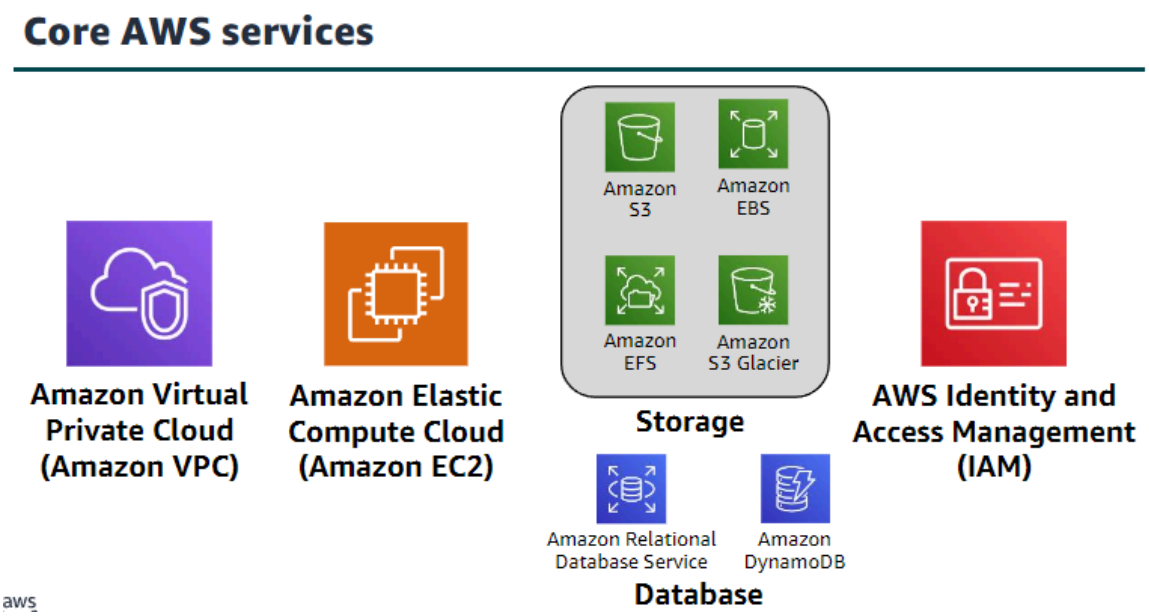
   Congratulations! You have successfully tested termination protection and terminated your instance.

Additional Resources

- Launch Your Instance

- Amazon EC2 Instance Types
- Amazon Machine Images (AMI)
- Amazon EC2 - User Data and Shell Scripts
- Amazon EC2 Root Device Volume
- Tagging Your Amazon EC2 Resources
- Security Groups
- Amazon EC2 Key Pairs
- Status Checks for Your Instances
- Getting Console Output and Rebooting Instances
- Amazon EC2 Metrics and Dimensions
- Resizing Your Instance
- Stop and Start Your Instance
- Amazon EC2 Service Limits
- Terminate Your Instance
- Termination Protection for an Instance

Module 7: Storage



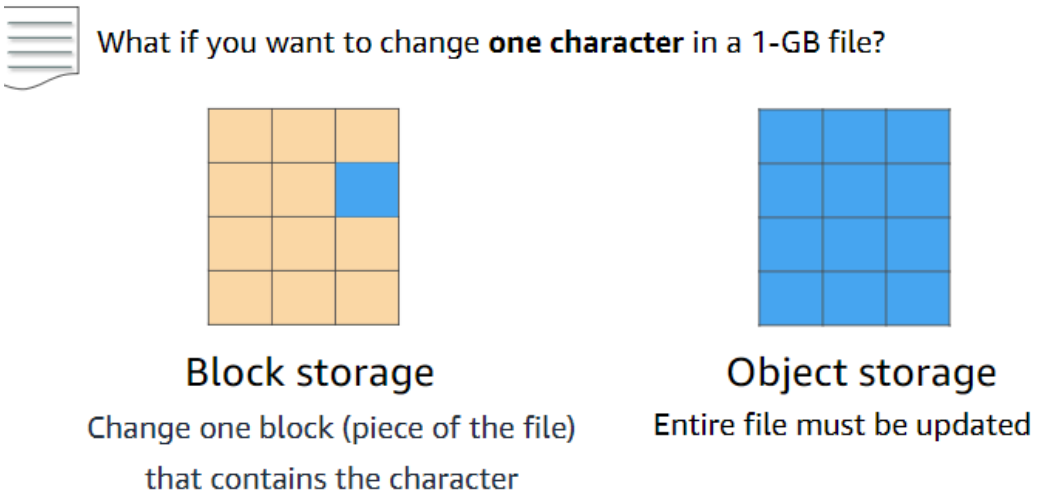
Storage is another AWS core service category. Some broad categories of storage include: instance store (ephemeral storage), Amazon EBS, Amazon EFS, Amazon S3, and Amazon S3 Glacier.•Instance store, or ephemeral storage, istemporary storagethat is added to your Amazon EC2 instance. •Amazon EBS is persistent, 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).

Section 1: Amazon Elastic Block Store(EBS)

Amazon Elastic Block Store
(Amazon EBS)

Amazon EBS provides persistent block storage volumes for use with Amazon EC2instances. Persistent storageis any datastoragedevice 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 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

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

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.

Amazon EBS volume type use cases

Solid State Drives (SSD)		Hard Disk Drives (HDD)	
General Purpose	Provisioned IOPS	Throughput-Optimized	Cold
• This type is recommended for most workloads	• Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/second of throughput per volume	• Streaming workloads that require consistent, fast throughput at a low price	• Throughput-oriented storage for large volumes of data that is infrequently accessed
• System boot volumes	• Large database workloads	• Big data	• Scenarios where the lowest storage cost is important
• Virtual desktops		• Data warehouses	• It cannot be a boot volume
• Low-latency interactive applications		• Log processing	
• Development and test environments		• It cannot be a boot volume	

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 of 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

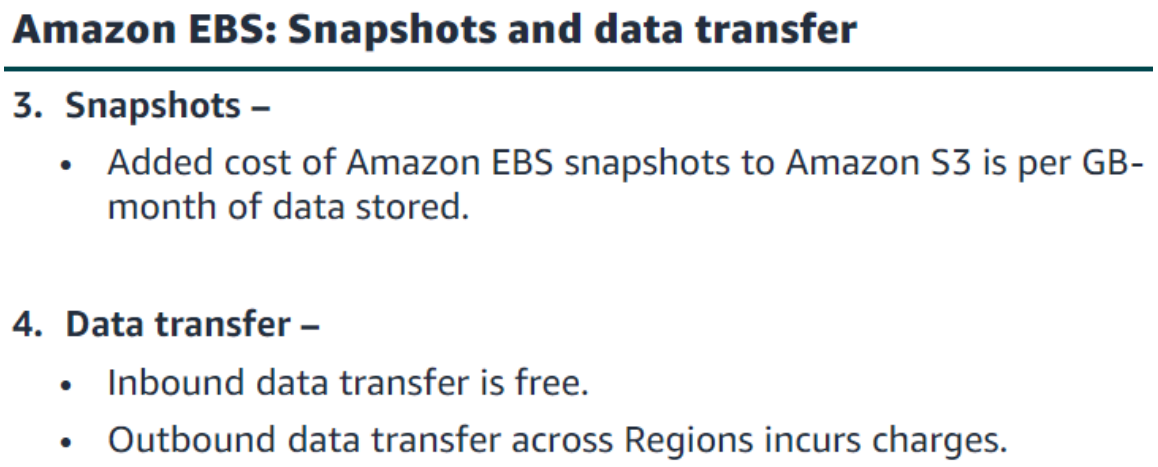
Amazon EBS enables you to create point-in-time snapshots of your volumes, and you can re-create a new volume from a snapshot at any time.

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.

Amazon EBS: Volumes, IOPS, and pricing

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).



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.

- **Demo Amazon EBS**

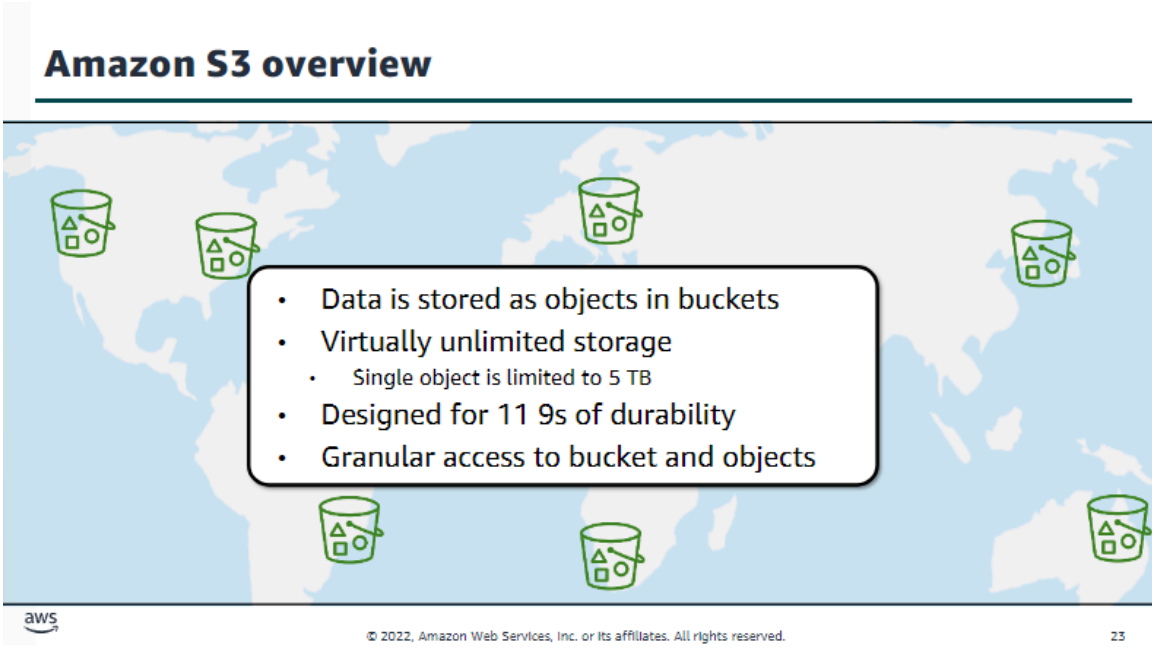
  https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-100-ACFNDS-20-EN/Module_7_EBS+v2.0.mp4

  https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-100-ACFNDS-20-EN/Module_7_EBS+v2.0.mp4

  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

Section 2: Amazon Simple Storage Service (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.



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 beuniqueacross all existingbucket namesinAmazon 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 Quick Sight.

Amazon S3 storage classes

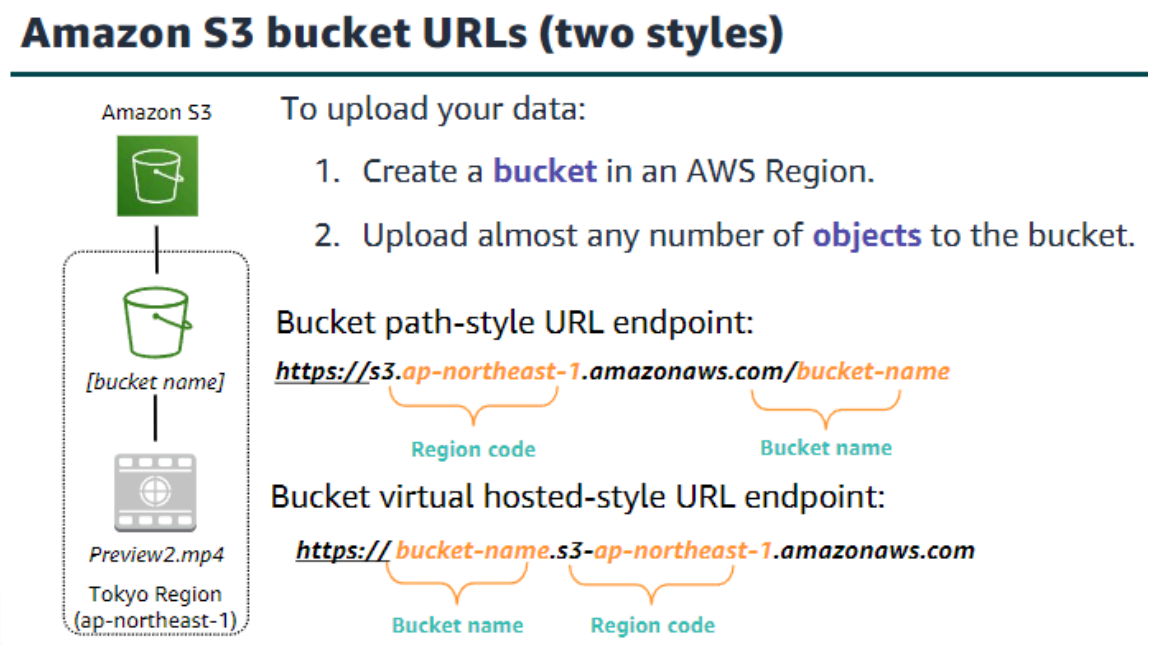
Amazon S3 offers a range of object-level storage classes that are designed for different use cases:

- 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

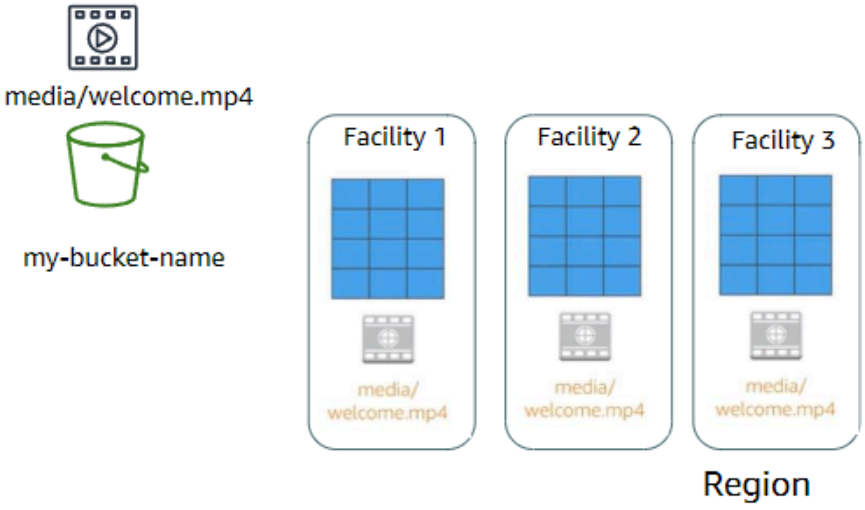
data (Amazon S3 Standard, Amazon S3 Intelligent-Tiering, Amazon S3 Standard-IA, and Amazon S3 One Zone-IA) and Amazon S3 Glacier.•Amazon S3 Glacier Deep Archive –Amazon S3 Glacier Deep Archive is the lowest-cost storage class for Amazon S3. It supports long-term retention and digital preservation for 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 about Amazon S3storage classes, see https://docs.aws.amazon.com/AmazonS3/latest/dev/storage-class-intro.html.



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

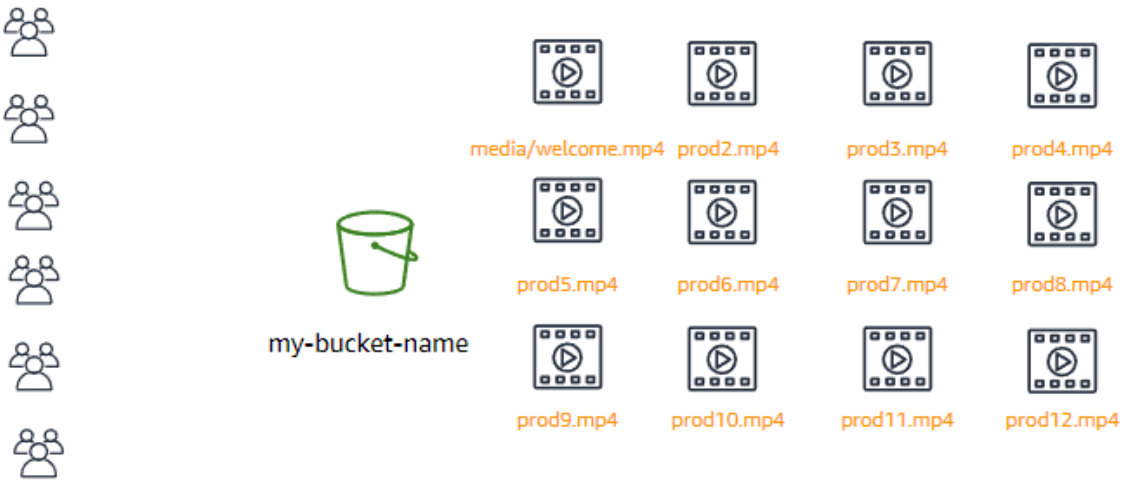
S3 globally. Buckets are logical containers for objects. You can have one or more buckets in your account. You can control access for each bucket—who can create, delete, and list objects in the bucket. You can also view access logs for the bucket and its objects, and choose the geographical region where 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-1The 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



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.

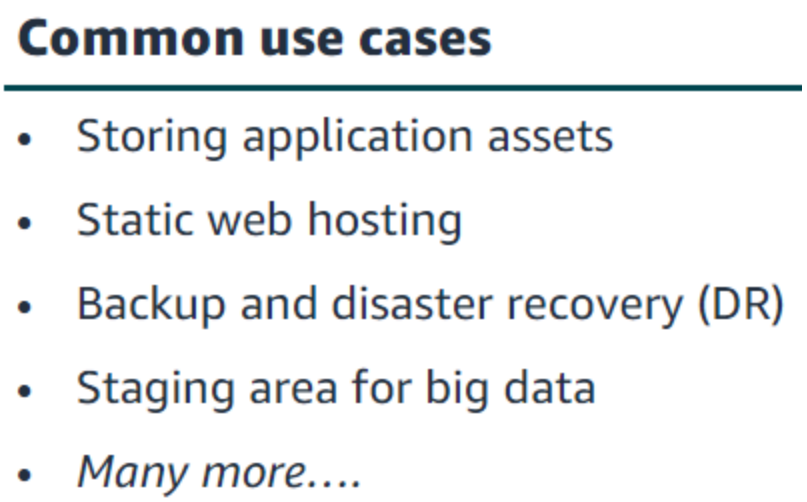
Designed for seamless scaling



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.

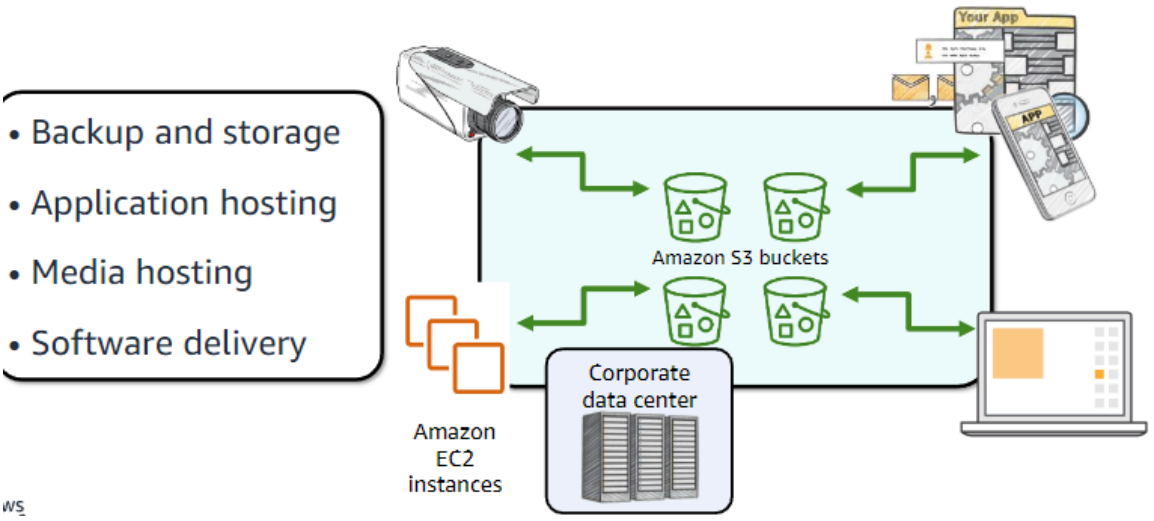
Access the data anywhere

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.



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 –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

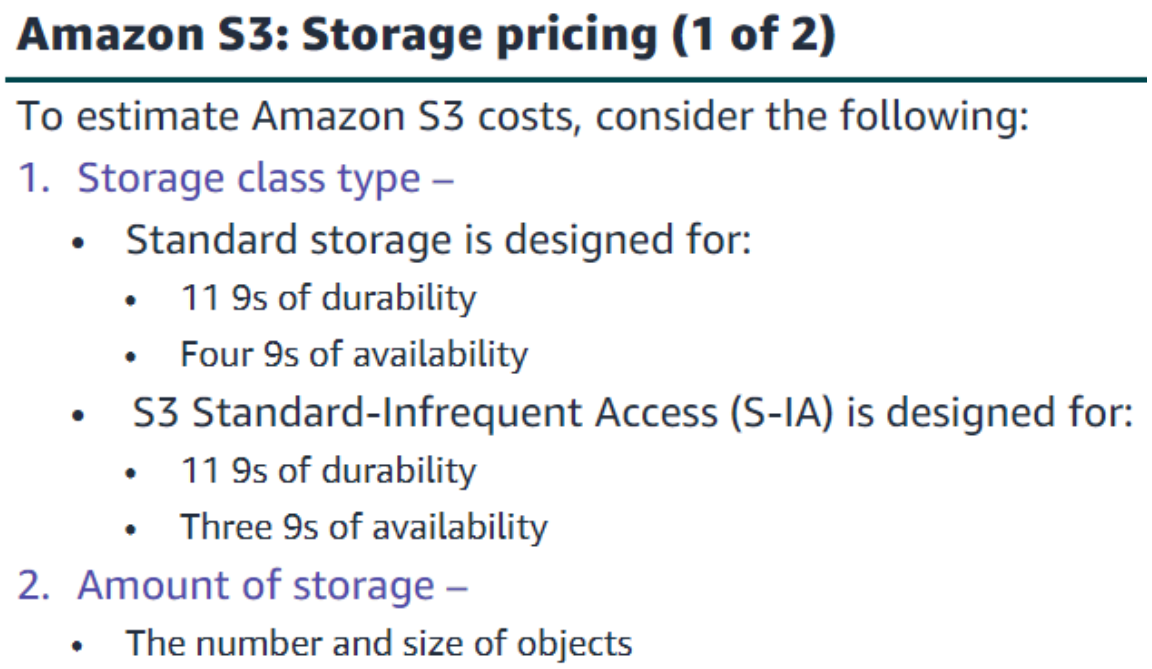
Software delivery –Host your software applications that customers can download.

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.



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.

Amazon S3: Storage pricing (2 of 2)

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.

- **Demo Amazon S3**

  https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-100-ACFNDS-20-EN/Module_7_S3+v2.0.mp4

  https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-100-ACFNDS-20-EN/Module_7_S3+v2.0.mp4

Section 3: Amazon Elastic File System(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)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.

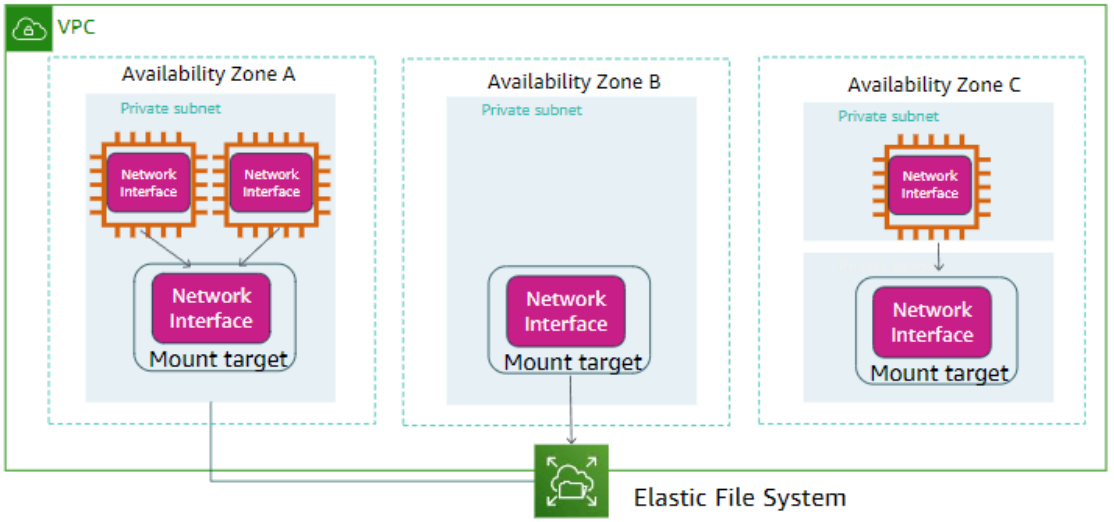
Amazon EFS features

- 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



ws

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.

Amazon EFS implementation

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.

Amazon EFS resources

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

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 stateYou 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.

- **Demo Amazon EFS**

  https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-100-ACFNDS-20-EN/Module_7_EFS+v2.0.mp4

Section 4: Amazon S3 Glacier

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.

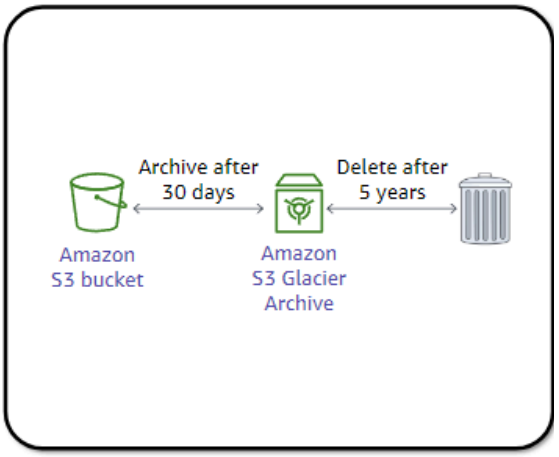
- Amazon S3 Glacier is designed to provide 11 9s of durability for objects.
- It supports the encryption of data in transit and at rest through Secure Sockets Layer (SSL) or Transport Layer Security (TLS).
- The Vault Lock feature enforces compliance through a policy.
- Extremely low-cost design works 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

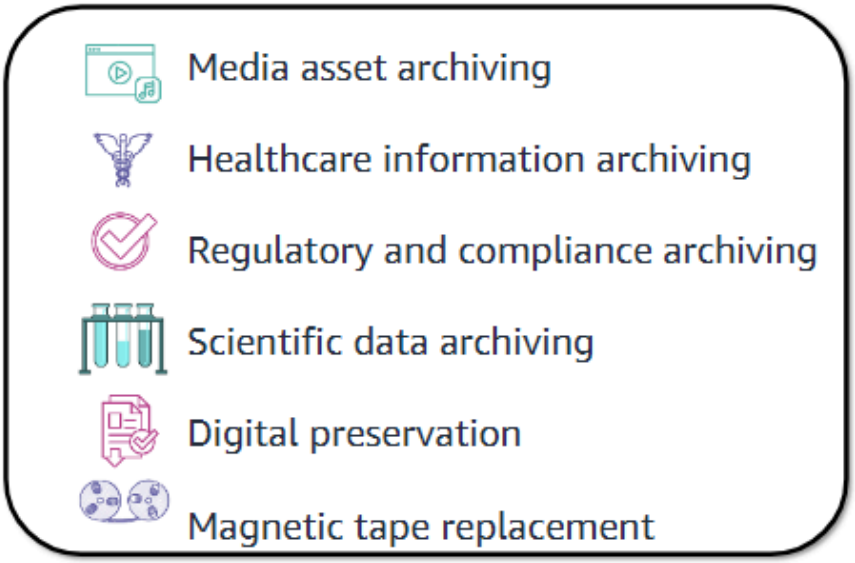
document) that you store in Amazon S3 Glacier. It is the base unit of storage in Amazon S3 Glacier. Each archive has its own unique ID and it can also have a description.•Vault–A container for storing archives. When you create a vault, you specify the vault name and the Region where you want to locate 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:•Expeditedretrievals are typically made available within 1–5 minutes (highest cost).•Standardretrievals typically complete within 3–5 hours (less time than expedited, more time than bulk).•Bulk retrievals typically complete within 5–12 hours (lowest cost).

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

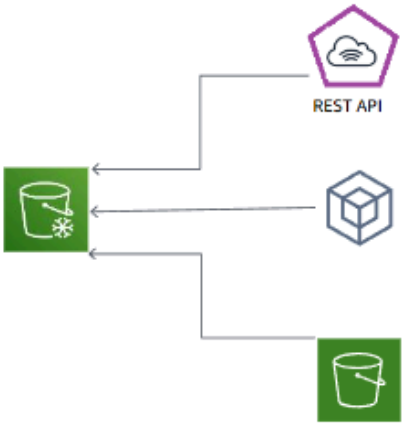


Amazon S3 Glacier use cases



Media asset archivingMedia 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 archivingTo 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 archivingMany 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 archivingResearch 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 preservationLibraries 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.

Using Amazon S3 Glacier



RESTful web services

Java or .NET SDKs

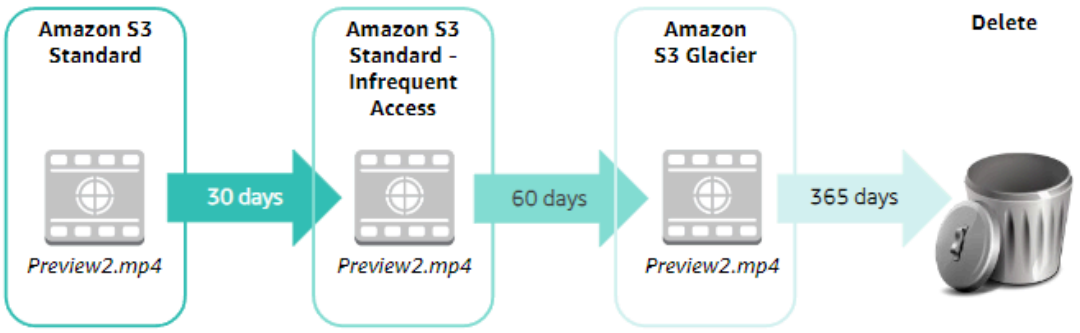
Amazon S3 with lifecycle policies

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.



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 about object lifecycle management, see http://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html

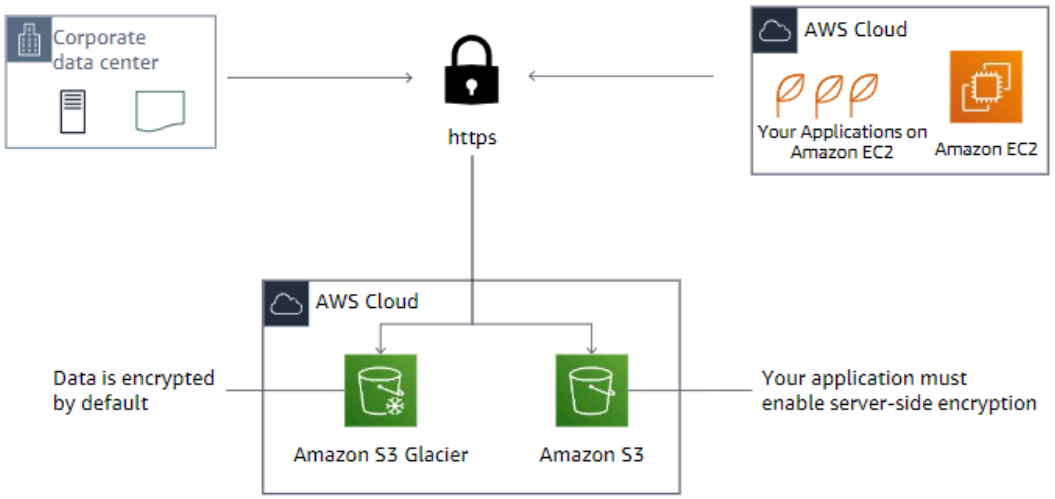
Storage comparison

	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

Amazon S3and 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 UPLOADand retrievaloperations.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



important difference between Amazon S3and Amazon S3Glacier 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 S3Glacier is encrypted by default. With Amazon S3, your application must initiate server-side encryption. You can accomplish server-side encryption in Amazon S3in several ways:•Server-side encryption with Amazon S3-managed encryption keys (SSE-S3) employs strong multi-factor encryption. Amazon S3encrypts each object with a unique key. As an additional safeguard, it encrypts the key with a main key that it regularly rotates. Amazon S3server-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 S3manages 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 S3objects. 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 S3buckets.

Security with Amazon S3 Glacier

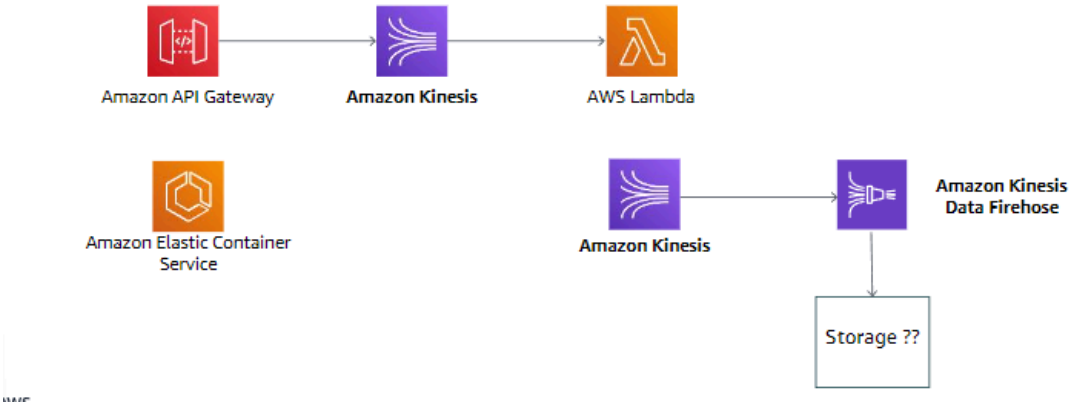
- **Demo Amazon S3 Glacier**

  https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-100-ACFNDS-20-EN/Module_7_S3_Glacier+v3.0.mp4

Activity: Storage Case Study

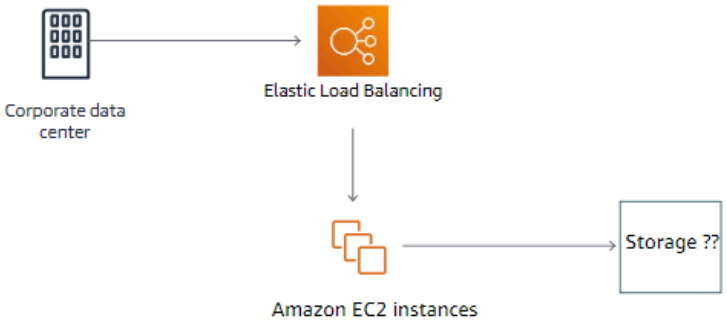
Storage case study activity (1 of 3)

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.



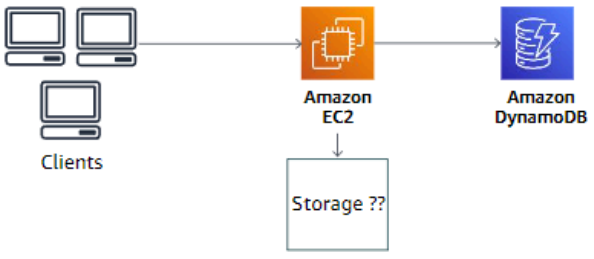
Storage case study activity (2 of 3)

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.



Storage case study activity (3 of 3)

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.



Additional resources

- AWS Storage page: https://aws.amazon.com/products/storage/
- Storage Overview: https://docs.aws.amazon.com/whitepapers/latest/aws-overview/storage-services.html
- Recovering files from an Amazon EBS volume backup: https://aws.amazon.com/blogs/compute/recovering-files-from-an-amazon-ebs-volume-backup/
- Confused by AWS Storage Options? S3, EFS, EBS Explained: https://dzone.com/articles/confused-by-aws-storage-options-s3-ebs-amp-efs-explained

Lab Activity 4: EBS

This lab focuses on Amazon Elastic Block Store (Amazon EBS), a key underlying storage mechanism for Amazon EC2 instances. In this lab, you will learn how to create an Amazon EBS volume, attach it to an instance, apply a file system to the volume, and then take a snapshot backup.

Topics covered

By the end of this lab, you will be able to:

- Create an Amazon EBS volume
- Attach and mount your volume to an EC2 instance
- Create a snapshot of your volume
- Create a new volume from your snapshot
- Attach and mount the new volume to your EC2 instance

Lab Pre-requisites

To successfully complete this lab, you should be familiar with basic Amazon EC2 usage and with basic Linux server administration. You should feel comfortable using the Linux command-line tools.

Duration

This lab will require approximately **30 minutes** to complete.

AWS service restrictions

In this lab environment, access to AWS services and service actions might be restricted to the ones that are needed to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that are described in this lab.

What is Amazon Elastic Block Store?

Amazon Elastic Block Store (Amazon EBS) offers persistent storage for Amazon EC2 instances. Amazon EBS volumes are network-attached and persist independently from the life of an instance. Amazon EBS volumes are highly available, highly reliable volumes that can be leveraged as an Amazon EC2 instances boot partition or attached to a running Amazon EC2 instance as a standard block device.

When used as a boot partition, Amazon EC2 instances can be stopped and subsequently restarted, enabling you to pay only for the storage resources used while maintaining your instance's state. Amazon EBS volumes offer greatly improved durability over local Amazon EC2 instance stores because Amazon EBS volumes are automatically replicated on the backend (in a single Availability Zone).

For those wanting even more durability, Amazon EBS provides the ability to create point-in-time consistent snapshots of your volumes that are then stored in Amazon Simple Storage Service (Amazon S3) and automatically replicated across multiple Availability Zones. These snapshots can be used as the starting point for new Amazon EBS volumes and can protect your data for long-term durability. You can also easily share these snapshots with co-workers and other AWS developers.

This lab guide explains basic concepts of Amazon EBS in a step-by-step fashion. However, it can only give a brief overview of Amazon EBS

concepts. For further information, see the Amazon EBS documentation.

Amazon EBS Volume Features

Amazon EBS volumes deliver the following features:

- **Persistent storage:** Volume lifetime is independent of any particular Amazon EC2 instance.
- **General purpose:** Amazon EBS volumes are raw, unformatted block devices that can be used from any operating system.
- **High performance:** Amazon EBS volumes are equal to or better than local Amazon EC2 drives.
- **High reliability:** Amazon EBS volumes have built-in redundancy within an Availability Zone.
- **Designed for resiliency:** The AFR (Annual Failure Rate) of Amazon EBS is between 0.1% and 1%.
- **Variable size:** Volume sizes range from 1 GB to 16 TB.
- **Easy to use:** Amazon EBS volumes can be easily created, attached, backed up, restored, and deleted.

Accessing the AWS Management Console

1. At the top of these instructions, click Start Lab to launch your lab.

   A Start Lab panel opens displaying the lab status.

2. Wait until you see the message "**Lab status: ready**", then click the **X** to close the Start Lab panel.

3. At the top of these instructions, click AWS

   This will open the AWS Management Console in a new browser tab. The system will automatically log you in.

   Tip: If a new browser tab does not open, there will typically be a banner or icon at the top of your browser indicating that your browser is preventing the site from opening pop-up windows. Choose the banner or icon and choose "Allow pop ups."

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time, to make it easier to follow the lab steps.

Task 1: Create a New EBS Volume

In this task, you will create and attach an Amazon EBS volume to a new Amazon EC2 instance.

1. In the **AWS Management Console**, on the **Services** menu, click **EC2**.
2. In the left navigation pane, choose **Instances**.

   An Amazon EC2 instance named **Lab** has already been launched for your lab.
3. Note the **Availability Zone** of the instance. It will look similar to *us-east-1a*.
4. In the left navigation pane, choose **Volumes**.

   You will see an existing volume that is being used by the Amazon EC2 instance. This volume has a size of 8 GiB, which makes it easy to distinguish from the volume you will create next, which will be 1 GiB in size.
5. Choose **Create volume** then configure:
   - **Volume Type:** *General Purpose SSD (gp2)*
   - **Size (GiB):** `1`. **NOTE**: You may be restricted from creating large volumes.
   - **Availability Zone:** Select the same availability zone as your EC2 instance.
   - Choose **Add Tag**
   - In the Tag Editor, enter:
     - **Key:** `Name`
     - **Value:** `My Volume`
6. Choose **Create Volume**

   Your new volume will appear in the list, and will move from the *Creating* state to the *Available* state. You may need to choose **refresh** to see your new volume.

Task 2: Attach the Volume to an Instance

You can now attach your new volume to the Amazon EC2 instance.

1. Select **My Volume**.
2. In the **Actions** menu, choose **Attach volume**.
3. Choose the **Instance** field, then select the instance that appears (Lab).

   Note that the **Device** field is set to */dev/sdf*. You will use this device identifier in a later task.
4. Choose **Attach volume**The volume state is now *In-use*.

Task 3: Connect to Your Amazon EC2 Instance

Windows Users: Using SSH to Connect

These instructions are for Windows users only.

If you are using macOS or Linux, skip to the next section.

1. Read through the three bullet points in this step before you start to complete the actions, because you will not be able see these instructions when the Details panel is open.
   - Choose the Details drop down menu above these instructions you are currently reading, and then choose Show. A Credentials window will open.
   - Choose the **Download PPK** button and save the **labsuser.ppk** file. Typically your browser will save it to the Downloads directory.
   - Then exit the Details panel by choosing the **X**.
2. Download needed software.
   - You will use **PuTTY** to SSH to Amazon EC2 instances. If you do not have PuTTY installed on your computer, download it here.
3. Open **putty.exe**
4. Configure PuTTY to not timeout:
   - Choose **Connection**
   - Set **Seconds between keepalives** to `30`

   This allows you to keep the PuTTY session open for a longer period of time.
5. Configure your PuTTY session:

- Choose **Session**
- **Host Name (or IP address):** Paste the *Public DNS or IPv4 address* of the Lab instance that you noted earlier.
- Back in PuTTY, in the **Connection** list, expand **SSH**
- Choose **Auth** and expand **Credentials**
- Under **Private key file for authentication:** Choose **Browse**
- Browse to the *labsuser.ppk* file that you downloaded, select it, and choose **Open**
- Choose **Open** again

6. To trust and connect to the host, choose **Accept**.
7. When prompted **login as**, enter: `ec2-user`

   This will connect you to the EC2 instance.
8. Windows Users: Choose here to skip ahead to the next task.

macOS and Linux Users

These instructions are for Mac/Linux users only. If you are a Windows user, skip ahead to the next task.

1. Read through all the instructions in this one step before you start to complete the actions, because you will not be able see these instructions when the Details panel is open.
   - Choose the Details drop down menu above these instructions you are currently reading, and then choose Show. A Credentials window will open.
   - Choose the **Download** button and save the **labsuser.pem** file.
   - Then exit the Details panel by choosing the **X**.
2. Open a terminal window, and change directory `cd` to the directory where the labsuser.pem file was downloaded.

   For example, run this command, if it was saved to your Downloads directory:

   ```
   cd ~/Downloads
   ```

3. Change the permissions on the key to be read only, by running this command:

```
chmod 400 labsuser.pem
```

4. Return to the AWS Management Console, and in the EC2 service, choose **Instances**.

   The **Lab** instance should be selected.

5. In the *Details* tab, copy the **Public IPv4 address** value.

6. Return to the terminal window and run this command (replace **<public-ip>** with the actual public IP address you copied):

```
ssh -i labsuser.pem ec2-user@<public-ip>
```

7. Type `yes` when prompted to allow a first connection to this remote SSH server.

   Because you are using a key pair for authentication, you will not be prompted for a password.

Task 4: Create and Configure Your File System

In this task, you will add the new volume to a Linux instance as an ext3 file system under the /mnt/data-store mount point.

If you are using PuTTY, you can paste text by right-clicking in the PuTTY window.

1. View the storage available on your instance:

```
df -h
```

   You should see output similar to:

```
Filesystem      Size  Used Avail Use% Mounted on2
devtmpfs        484M     0  484M   0% /dev3
tmpfs           492M     0  492M   0% /dev/shm4
tmpfs           492M  460K  491M   1% /run5
tmpfs           492M     0  492M   0% /sys/fs/cgroup6
/dev/xvda1      8.0G  1.5G  6.6G  19% /7
```

```
tmpfs           99M     0   99M   0% /run/user/08
tmpfs           99M     0   99M   0% /run/user/1000
```

This is showing the original 8GB disk volume. Your new volume is not yet shown.

2. Create an ext3 file system on the new volume:

```
sudo mkfs -t ext3 /dev/sdf
```

3. Create a directory for mounting the new storage volume:

```
sudo mkdir /mnt/data-store
```

4. Mount the new volume:

```
sudo mount /dev/sdf /mnt/data-store
```

To configure the Linux instance to mount this volume whenever the instance is started, you will need to add a line to */etc/fstab*.

```
echo "/dev/sdf   /mnt/data-store ext3 defaults,noatime 1 2" | sudo t
ee -a /etc/fstab
```

5. View the configuration file to see the setting on the last line:

```
cat /etc/fstab
```

6. View the available storage again:

```
df -h
```

The output will now contain an additional line - */dev/xvdf*:

```
Filesystem      Size  Used Avail Use% Mounted on2
devtmpfs        484M     0  484M   0% /dev3
tmpfs           492M     0  492M   0% /dev/shm4
tmpfs           492M  460K  491M   1% /run5
tmpfs           492M     0  492M   0% /sys/fs/cgroup6
```

```
/dev/xvda1      8.0G  1.5G  6.6G  19% /7
tmpfs           99M     0   99M   0% /run/user/08
tmpfs           99M     0   99M   0% /run/user/10009
/dev/xvdf       976M  1.3M  924M   1% /mnt/data-store
```

7. On your mounted volume, create a file and add some text to it.

```
sudo sh -c "echo some text has been written > /mnt/data-store/file.txt"
```

8. Verify that the text has been written to your volume.

```
cat /mnt/data-store/file.txt
```

Task 5: Create an Amazon EBS Snapshot

In this task, you will create a snapshot of your EBS volume.

You can create any number of point-in-time, consistent snapshots from Amazon EBS volumes at any time. Amazon EBS snapshots are stored in Amazon S3 with high durability. New Amazon EBS volumes can be created out of snapshots for cloning or restoring backups. Amazon EBS snapshots can also be easily shared among AWS users or copied over AWS regions.

1. In the **AWS Management Console**, choose **Volumes** and select **My Volume**.
2. In the **Actions** menu, select **Create snapshot**.
3. Choose **Add tag** then configure:
   - **Key:** `Name`
   - **Value:** `My Snapshot`
   - Choose **Create snapshot**
4. In the left navigation pane, choose **Snapshots**.

   Your snapshot is displayed. The status will first have a state of *Pending*, which means that the snapshot is being created. It will then change to a state of *Completed*.

Note: Only used storage blocks are copied to snapshots, so empty blocks do not occupy any snapshot storage space.

5. In your remote SSH session, delete the file that you created on your volume.

```
sudo rm /mnt/data-store/file.txt
```

6. Verify that the file has been deleted.

```
ls /mnt/data-store/
```

Your file has been deleted.

Task 6: Restore the Amazon EBS Snapshot

If you ever wish to retrieve data stored in a snapshot, you can **Restore** the snapshot to a new EBS volume.

Create a Volume Using Your Snapshot

1. In the **AWS Management Console**, select **My Snapshot**.
2. In the **Actions** menu, select **Create volume from snapshot**.
3. For **Availability Zone** Select the same availability zone that you used earlier.
4. Choose **Add tag** then configure:
   - **Key:** `Name`
   - **Value:** `Restored Volume`
   - Choose **Create volume**

   Note: When restoring a snapshot to a new volume, you can also modify the configuration, such as changing the volume type, size or Availability Zone.

Attach the Restored Volume to Your EC2 Instance

1. In the left navigation pane, choose **Volumes**.
2. Select **Restored Volume**.

3. In the **Actions** menu, select **Attach volume**.
4. Choose the **Instance** field, then select the (Lab) instance that appears.

   Note that the **Device** field is set to */dev/sdg*. You will use this device identifier in a later task.
5. Choose **Attach volume**

   The volume state is now *in-use*.

Mount the Restored Volume

1. Create a directory for mounting the new storage volume:

   ```
   sudo mkdir /mnt/data-store2
   ```
2. Mount the new volume:

   ```
   sudo mount /dev/sdg /mnt/data-store2
   ```
3. Verify that volume you mounted has the file that you created earlier.

   ```
   ls /mnt/data-store2/
   ```

   You should see file.txt.

Conclusion

Congratulations! You now have successfully:

- Created an Amazon EBS volume
- Attached the volume to an EC2 instance
- Created a file system on the volume
- Added a file to volume
- Created a snapshot of your volume
- Created a new volume from the snapshot
- Attached and mounted the new volume to your EC2 instance
- Verified that the file you created earlier was on the newly created volume

Module 8: Database

Section 1: Amazon Relational Database Service

Amazon Relational Database Service (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 into the service.

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.

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

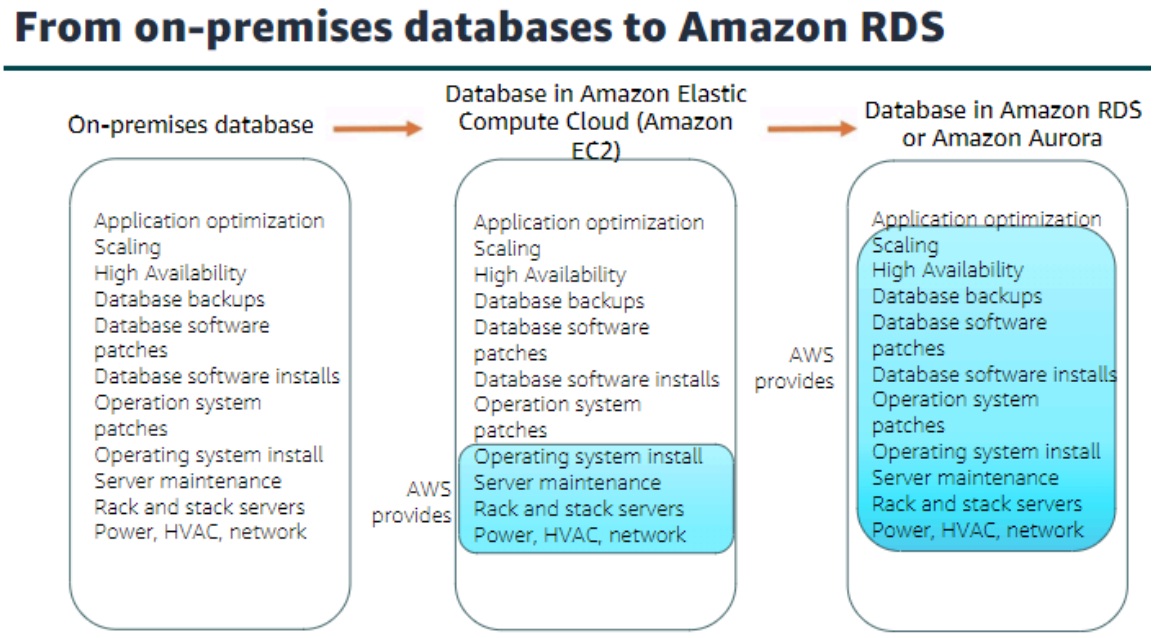
Amazon RDS

Managed service that sets up and operates a relational database in the cloud.

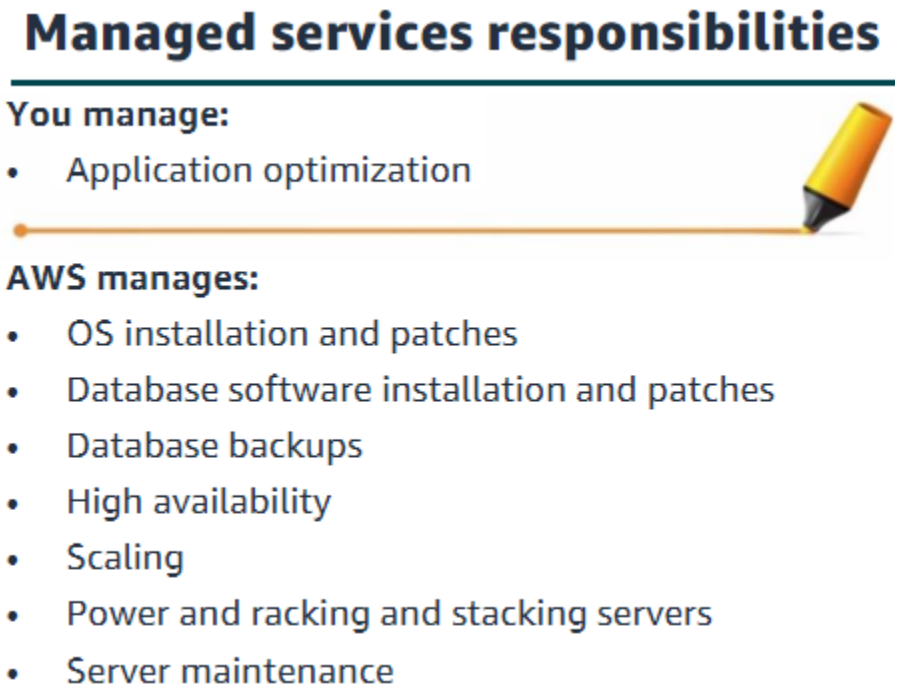
Amazon RDS is a managed service that sets up and operates a relational database in the cloud.

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.

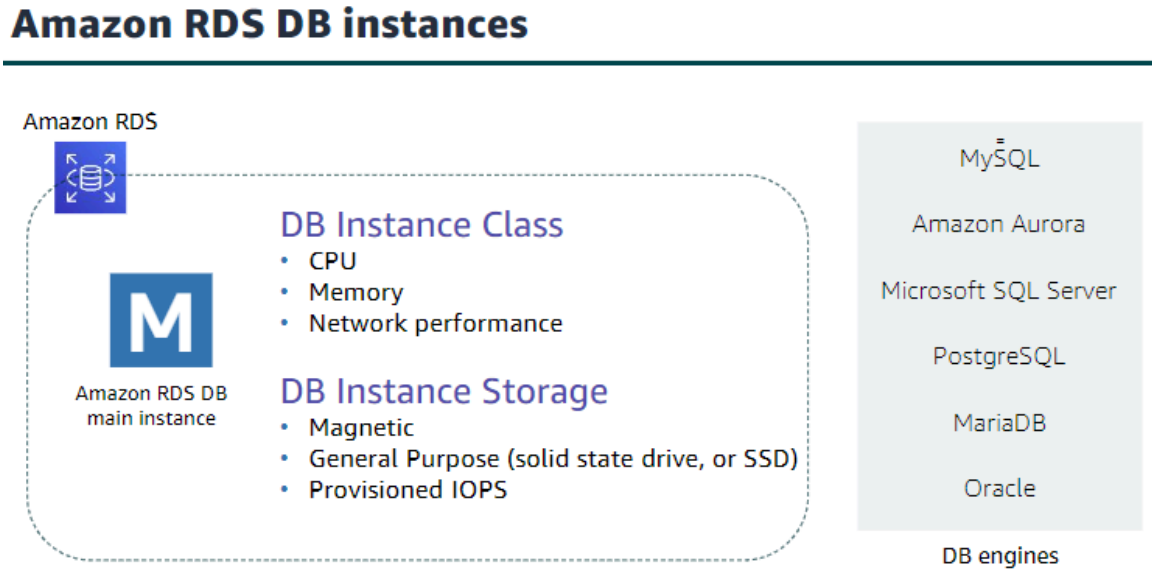
Amazon RDS enables you to focus on your application, so you can give applications the performance, high availability, security, and compatibility that they need. With Amazon RDS, your primary focus is your data and optimizing your application.



What does the term managed servicesmean?When your database is onpremises, the database administrator is responsible for everything.Database administration tasks include optimizingapplicationsand queries; setting up the hardware; patching the hardware;setting up networkingand power; andmanaging 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.



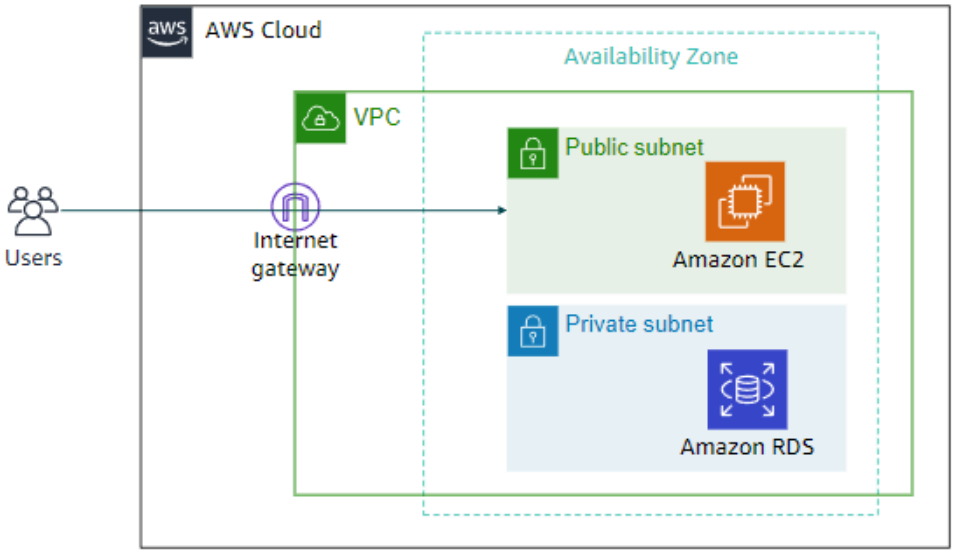
AWS also scales resources, manages power and servers, and performs maintenance. Offloading these operations to the managed Amazon RDS service reduces your operational workload and the costs that are associated with your relational database.



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.

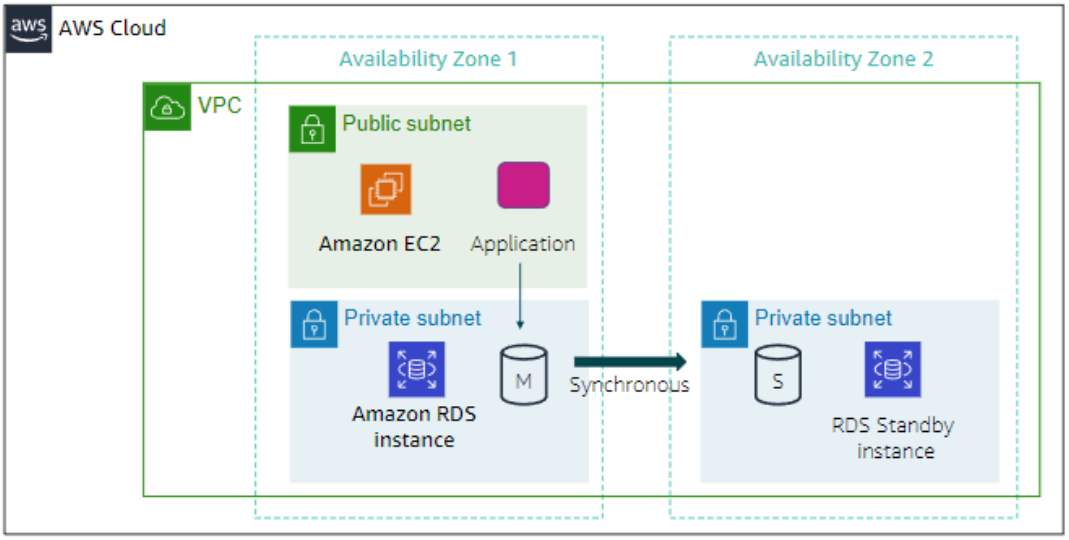
The resources in a database instance are determined by its database instance class, and the type of storage is dictated by the type of disks. Database instances and storage differ in performance characteristics and price, which enable you to customize your performance and cost to the needs of your database. 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)



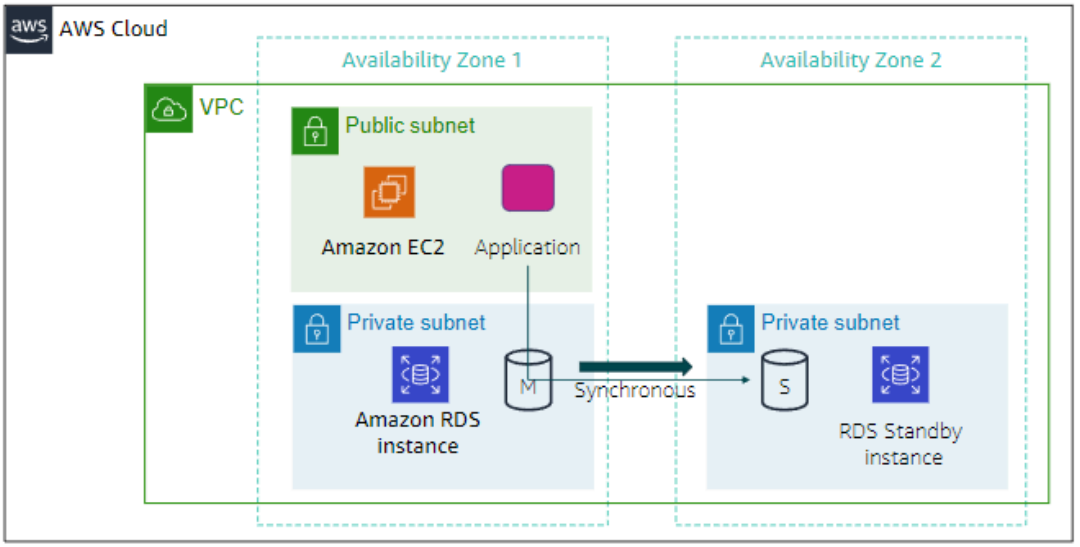
You can run an instance by using Amazon Virtual Private Cloud (Amazon VPC). When you use a virtual private cloud (VPC), you have control over your virtual networking environment. You can select your own IP address range, create subnets, and configure routing and access control lists (ACLs). The basic 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 (1 of 2)



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 aMulti-AZ deployment isconfigured, 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-AZdeployment 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 of 2)



Therefore, if the main database instance fails in a Multi-AZdeployment, Amazon RDS automatically brings the standby database instance online as the new main instance. Thesynchronous replicationminimizes 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.

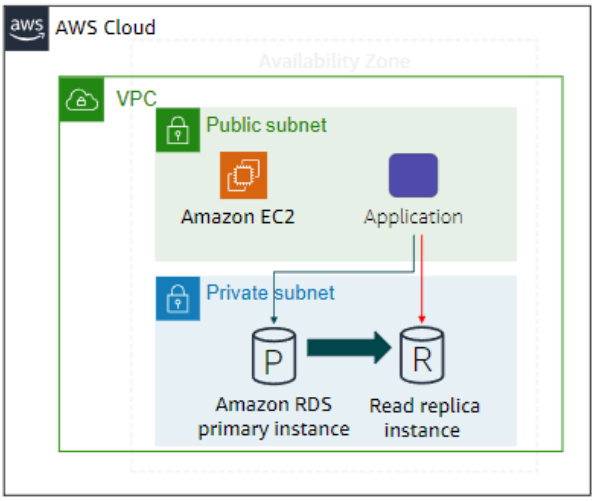
Amazon RDS read replicas

Features

- Offers asynchronous replication
- Can be promoted to primary if needed

Functionality

- Use for read-heavy database workloads
- Offload read queries



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 thisrequires 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.

Use cases

Web and mobile applications	✓ High throughput ✓ Massive storage scalability ✓ High availability
Ecommerce applications	✓ Low-cost database ✓ Data security ✓ Fully managed solution
Mobile and online games	✓ 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

Use Amazon RDS when your application requires: •Complex transactions or complex queries•A medium to high queryor write rate –up to 30,000 IOPS (15,000 reads + 15,000 writes)•No more than a single worker nodeor shard•High durabilityDo 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 GETor PUT requests and queries that a NoSQL database can handle•Or, relational database management system (RDBMS) customizationFor 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

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.

Amazon RDS: DB purchase type and multiple DB instances

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

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.

Amazon RDS: Storage

Provisioned storage –

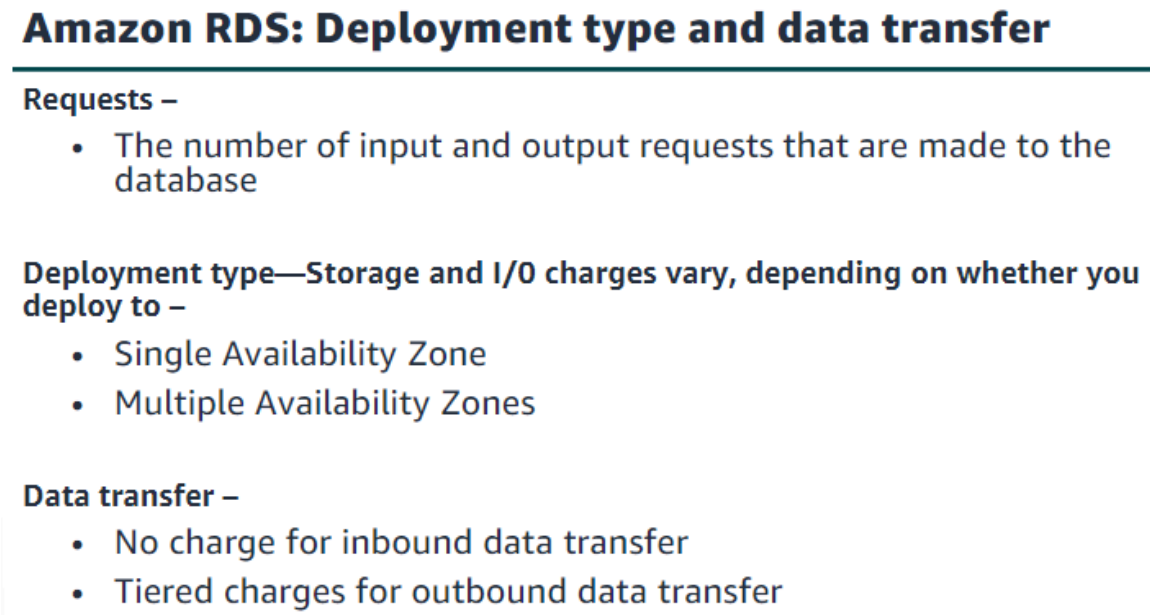
- No charge
  - Backup storage of up to 100 percent of database storage for an active database
- Charge (*GB/month*)
  - Backup storage for terminated DB instances

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 100percentof your provisioned database storage for an active database instance. After the database instance is terminated, backup

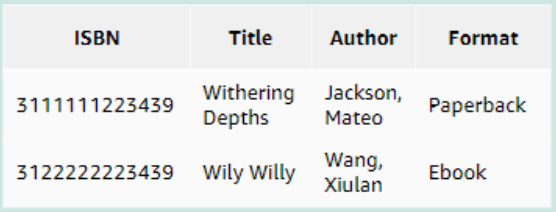
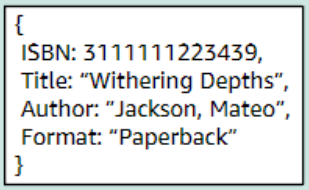
storage is billed per GB, per month. Also consider the amount of backup storage in addition to the provisioned storage amount, which is billed per GB, per month.



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.

Section 2: Amazon Dynamo DB

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		

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 semi-structured 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 semi-structured data.

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

DynamoDB is a fast and flexible NoSQL databaseservice 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

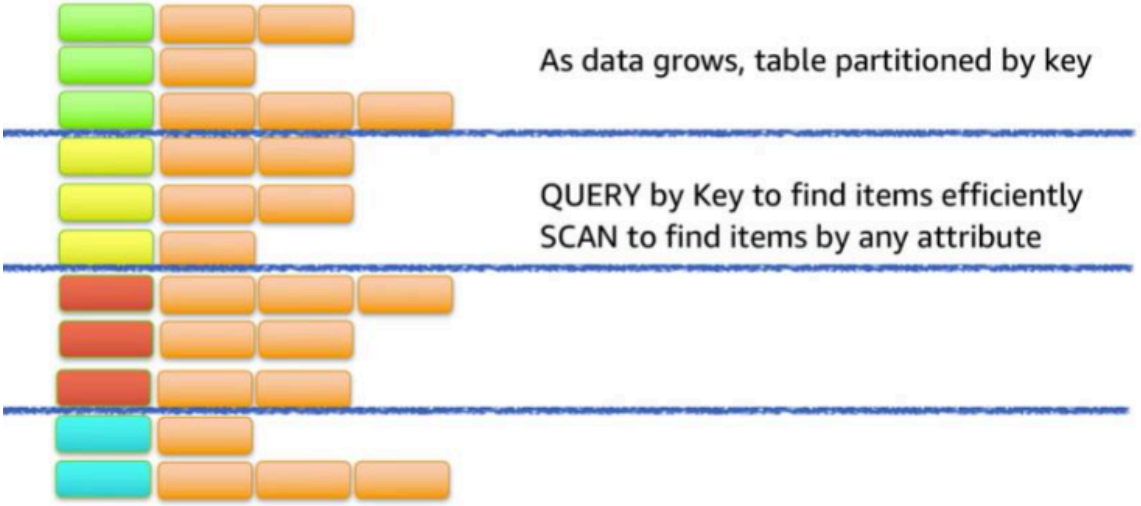
global tables that enable you to automatically replicate across your choice of AWS Regions, encryption at rest, and item Time-to-Live (TTL).

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.

Partitioning

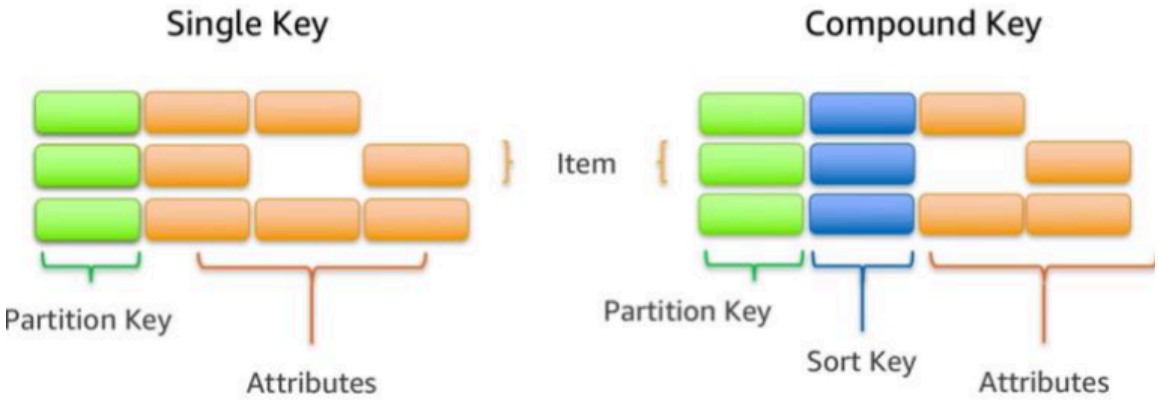


As data grows, table data is partitioned and indexed by the primary key. You can retrieve data from a DynamoDB table in two different ways:•In the first method, the query operation takes advantage of partitioning to effectively

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.

For accessibility:Partitioning allows large tables to be scanned and queried quickly. As data grows, table is partitioned by key. QUERY by Key to find items by any attribute. End of accessibility description.

Items in a table must have a key

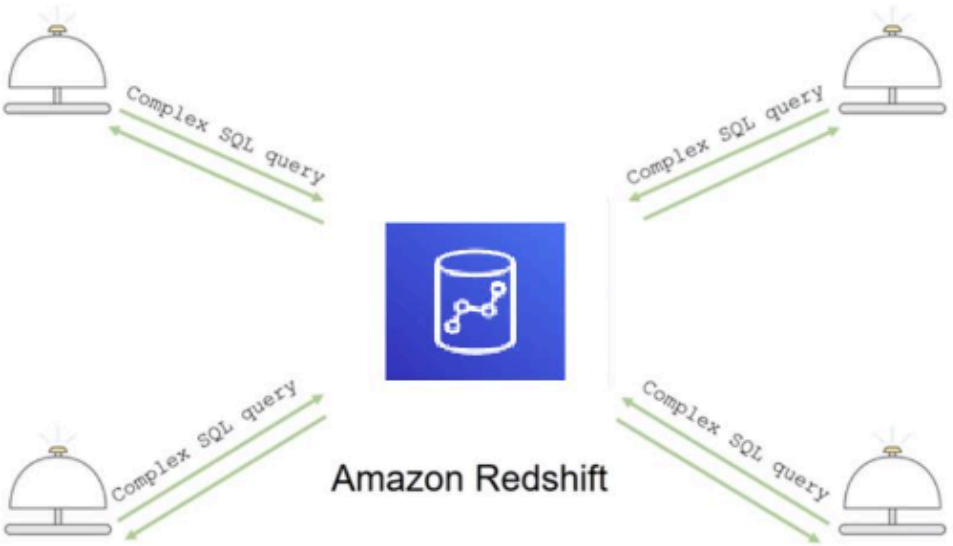


You can set up a simple primary key that is based on a single attribute of the data values with a uniform distribution, such as the Globally Unique Identifier (GUID) or other random identifiers. For example, if you wanted to model a table with products, you could use some attributeslike the product ID. 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, youmight 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 authorbecause you couldthen use query.For accessibility:The two different types of keys. A single key means the data is identified by an item in the data that uniquely identifies each record. A compound key is made up of a partition key and a second key that can be used for sorting data. End of accessibility description.

Section 3: Amazon RedShift

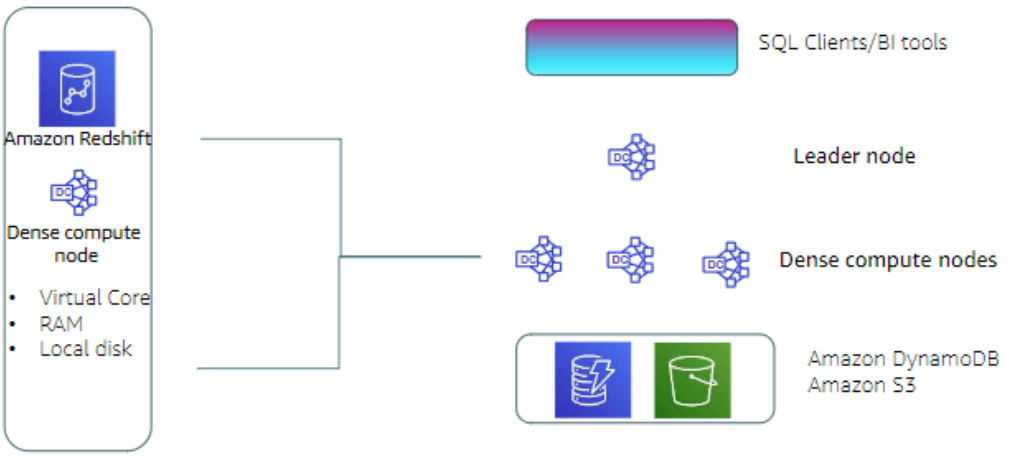
Amazon Redshift is a fast, fully managed data warehouse that makes it simple and cost-effective to analyze all your data by using standard SQL and your existing business intelligence (BI) tools.

Introduction to Amazon Redshift



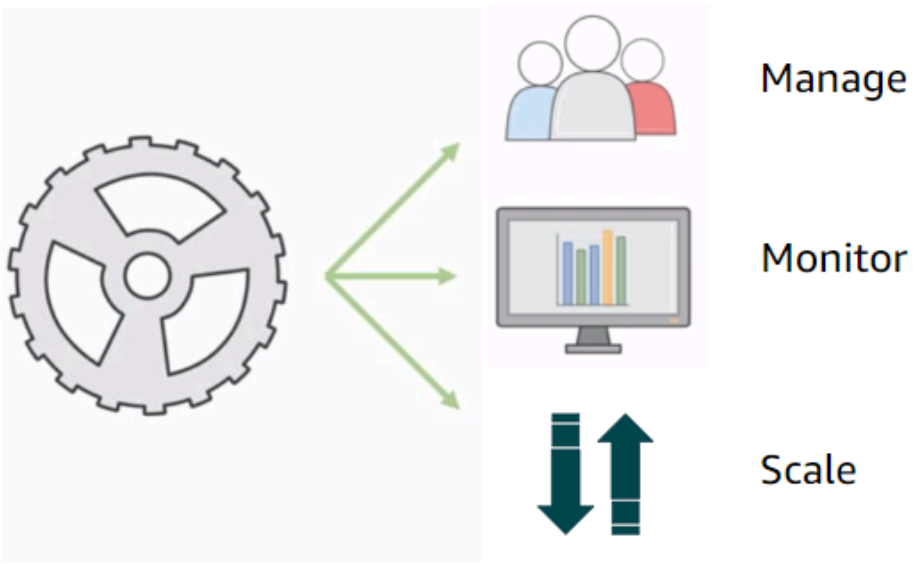
Analytics is important for businesses today, but building a data warehouse is complex and expensive. Data warehouses cantake months and significant financial resources to set up.Amazon Redshift is a fast and powerful, fullymanaged data warehouse that is simple and cost-effective to set up, use, and scale. 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.

Parallel processing architecture



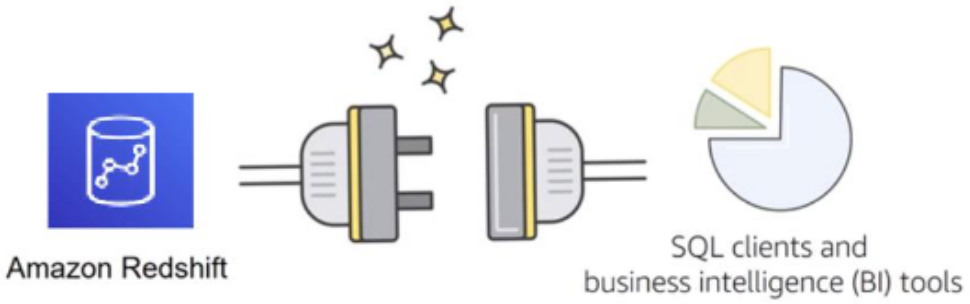
The leader node manages communications with client programs and all communication with compute nodes. It parses and develops plans to carry out database operations—specifically, the series of steps that are needed to obtain results for complex queries. The leader node compiles code for individual elements 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.Likeother 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 PartialUpfront Reserved Instance pricing). The Amazon Redshift Spectrum feature enables you to run queries against exabytes of data directly in Amazon S3.

Automation and scaling



It is straightforward to automate most of the common administrative tasks to manage, monitor, and scale your Amazon Redshift cluster—whichenables 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 builtin, and it is designed to provide strong encryption of your data both at rest and in transit.

Compatibility

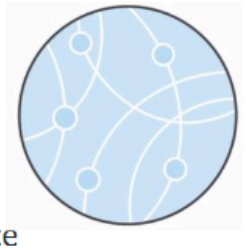


Finally, Amazon Redshift is compatible with the tools that you already know and use. Amazon Redshift supports standard SQL. It also provides high-performance Java Database Connectivity (JDBC) and Open Database

Connectivity (ODBC) connectors, which enable you to use the SQL clients and BI tools of your choice.

Amazon Redshift use cases (1 of 2)

- 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



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 toprocure the hardware and expertise that is needed to run these systems. With Amazon Redshift, smaller customers can quickly set upand useadata 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 enablescustomersto focus on querying and analyzing their data.

Amazon Redshift use cases (2 of 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 advantageof the scalable, easy-to-manage features that Amazon Redshift provides. Some customers use the Amazon Redshift to provide analytic capabilities to their applications. Some usersdeploy a cluster per customer,and use tagging to simplify and manage their service level agreements (SLAs) and billing. Amazon Redshift canhelp you reduce hardware and software costs.

Section 4: Amazon Aurora

Amazon Aurora

Amazon Aurora

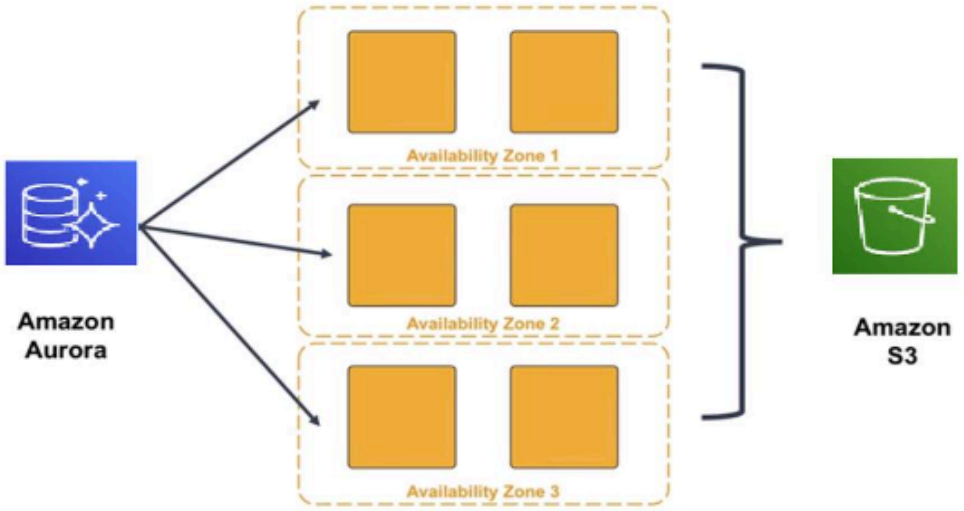
- Enterprise-class relational database
- Compatible with MySQL or PostgreSQL
- Automate time-consuming tasks (such as provisioning, patching, backup, recovery, failure detection, and repair).

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

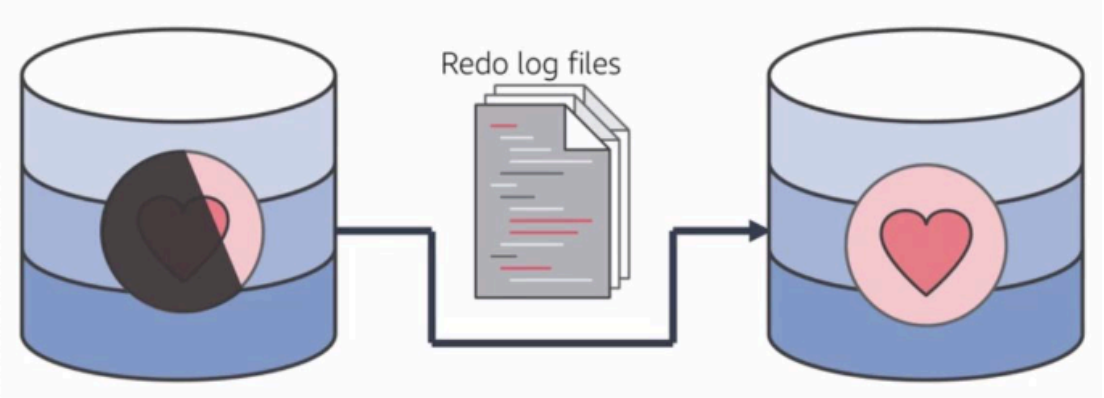
t is highly available and it offers a fast,distributedstorage subsystem. Amazon Aurora is straightforward to set up and uses SQL queries. It is designed to have drop-in compatibilitywith MySQL and PostgreSQL database enginesso that you can use most of your existing database tools with little or no change.Amazon 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 AWSDatabase 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



Why mightyou 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:itstores multiple copies of your data across multiple Availability Zones with continuous backups to Amazon S3. AmazonAurora can use upto 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



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.

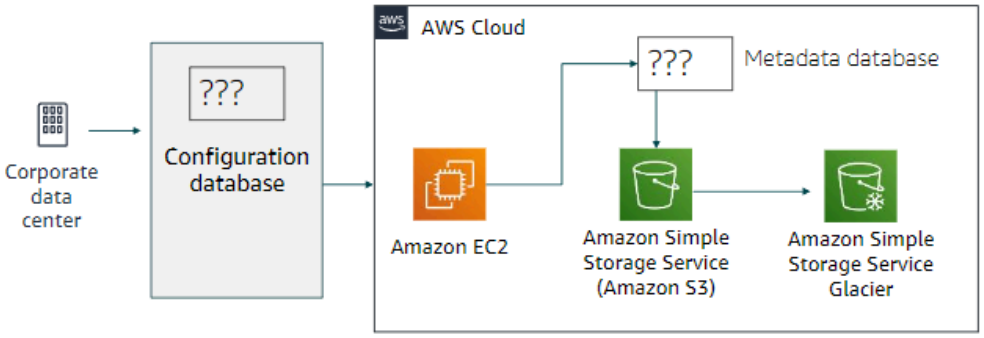
The right tool for the right job

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

Theseapplicationsneed databases to store terabytes to petabytes of new typesof data, provide access to the data with millisecond latency, process millions of requests persecond, and scale to support millions of usersanywherein 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 thatare builtfor your specific applicationuse cases.

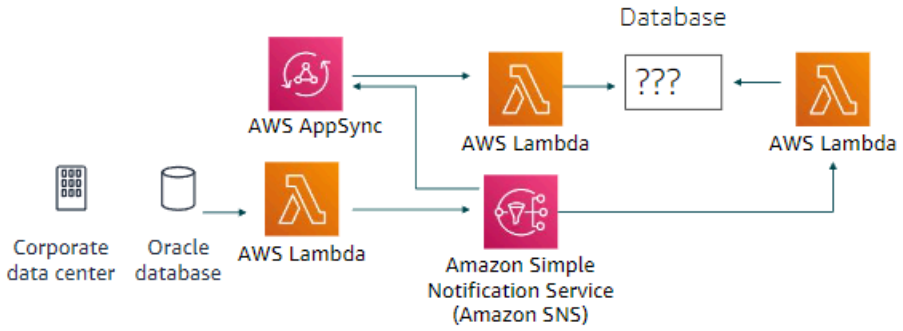
Database case study activity (1 of 3)

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.



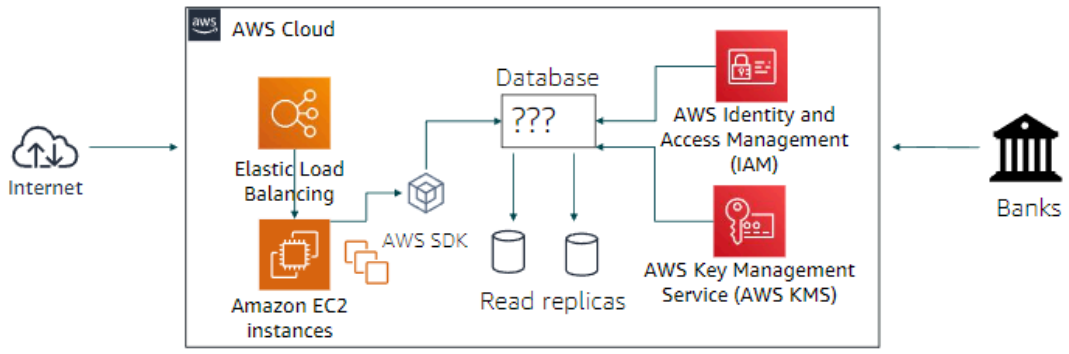
Database case study activity (2 of 3)

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.



Database case study activity 3

Case 3: An online payment processing company that processes over1 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.



Additional resources

- AWS Database page: https://aws.amazon.com/products/databases/
- Amazon RDS page: https://aws.amazon.com/rds/
- Overview of Amazon database services: https://docs.aws.amazon.com/whitepapers/latest/aws-overview/database.html
- Getting started with AWS databases: https://aws.amazon.com/products/databases/learn/

Lab Activity 5: Build a Database Server

Amazon Relational Database Service (Amazon RDS) 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, which allows you to focus on your applications and business. Amazon RDS provides you with six familiar database engines to choose from: Amazon Aurora, Oracle, Microsoft SQL Server, PostgreSQL, MySQL and MariaDB.

Objectives

After completing this lab, you can:

- Launch an Amazon RDS DB instance with high availability.
- Configure the DB instance to permit connections from your web server.

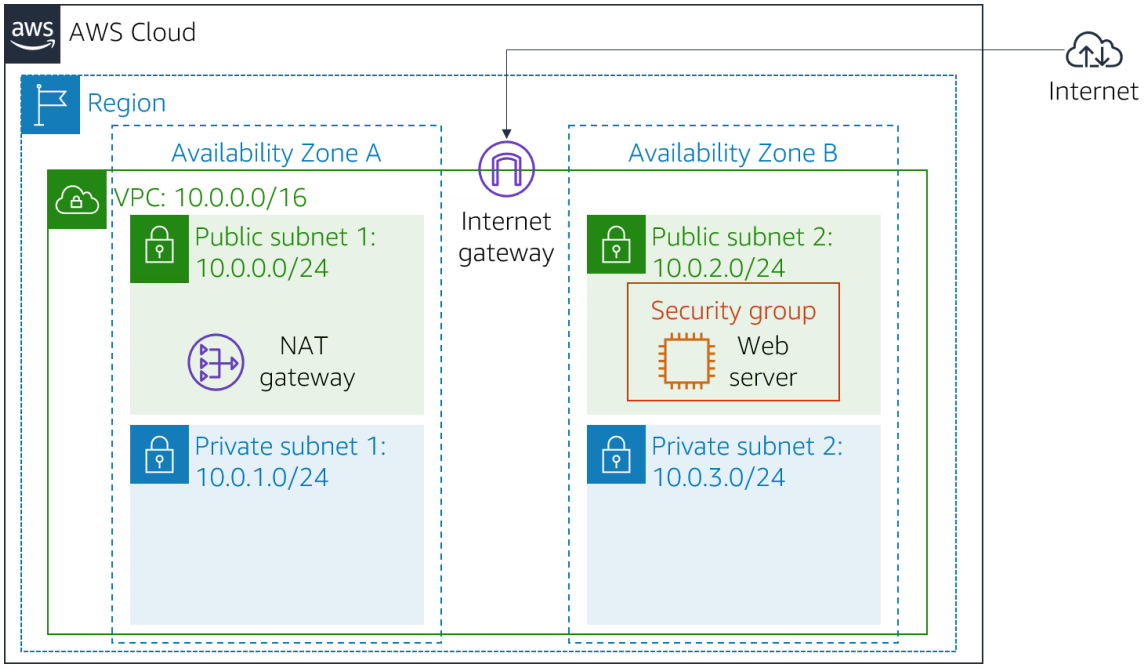
- Open a web application and interact with your database.

Duration

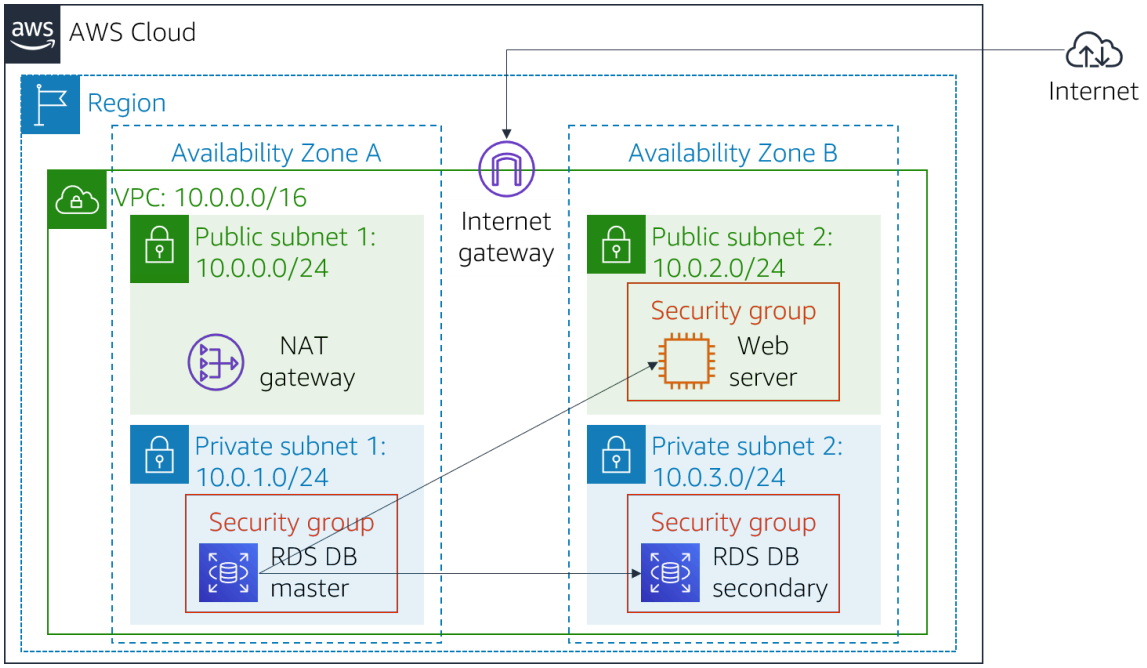
This lab takes approximately **30 minutes**.

Scenario

You start with the following infrastructure:



At the end of the lab, this is the infrastructure:



Accessing the AWS Management Console

1. At the top of these instructions, choose Start Lab to launch your lab.

   A Start Lab panel opens displaying the lab status.

2. Wait until you see the message "**Lab status: ready**", then choose the **X** to close the Start Lab panel.

3. At the top of these instructions, choose AWS

   This will open the AWS Management Console in a new browser tab. The system will automatically log you in.

   Tip: If a new browser tab does not open, there will typically be a banner or icon at the top of your browser indicating that your browser is preventing the site from opening pop-up windows. Choose on the banner or icon and choose "Allow pop ups."

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time, to make it easier to follow the lab steps.

Task 1: Create a Security Group for the RDS DB Instance

In this task, you will create a security group to allow your web server to access your RDS DB instance. The security group will be used when you launch the database instance.

1. In the **AWS Management Console**, on the Services menu, choose **VPC**.
2. In the left navigation pane, choose **Security Groups**.
3. Choose Create security group and then configure:

   - **Security group name:** `DB Security Group`
   - **Description:** `Permit access from Web Security Group`
   - **VPC:** *Lab VPC*

   You will now add a rule to the security group to permit inbound database requests.

4. In the **Inbound rules** pane, choose Add rule

The security group currently has no rules. You will add a rule to permit access from the *Web Security Group*.

5. Configure the following settings:
   - **Type:** *MySQL/Aurora (3306)*
   - **CIDR, IP, Security Group or Prefix List:** Type `sg` and then select *Web Security Group*.

   This configures the Database security group to permit inbound traffic on port 3306 from any EC2 instance that is associated with the *Web Security Group*.
6. Choose Create security group

   You will use this security group when launching the Amazon RDS database.

Task 2: Create a DB Subnet Group

In this task, you will create a *DB subnet group* that is used to tell RDS which subnets can be used for the database. Each DB subnet group requires subnets in at least two Availability Zones.

1. On the Services menu, choose **RDS**.
2. In the left navigation pane, choose **Subnet groups**.

   If the navigation pane is not visible, choose the menu icon in the top-left corner.
3. Choose Create DB Subnet Group then configure:
   - **Name:** `DB-Subnet-Group`
   - **Description:** `DB Subnet Group`
   - **VPC:** *Lab VPC*
4. Scroll down to the **Add Subnets** section.
5. Expand the list of values under **Availability Zones** and select the first two zones: **us-east-1a** and **us-east-1b**.
6. Expand the list of values under **Subnets** and select the subnets associated with the CIDR ranges **10.0.1.0/24** and **10.0.3.0/24**.

   These subnets should now be shown in the **Subnets selected** table.

7. Choose Create

   You will use this DB subnet group when creating the database in the next task.

Task 3: Create an Amazon RDS DB Instance

In this task, you will configure and launch a Multi-AZ Amazon RDS for MySQL database instance.

Amazon RDS ***Multi-AZ*** deployments provide enhanced availability and durability for Database (DB) instances, making them a natural 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 (AZ).

1. In the left navigation pane, choose **Databases**.
2. Choose Create database

   If you see **Switch to the new database creation flow** at the top of the screen, please choose it.
3. Select **MySQL** under **Engine Options**.
4. Under **Templates** choose **Dev/Test**.
5. Under **Availability and durability** choose **Multi-AZ DB instance**.
6. Under **Settings**, configure:
   - **DB instance identifier:** `lab-db`
   - **Master username:** `main`
   - **Master password:** `lab-password`
   - **Confirm password:** `lab-password`
7. Under **DB instance class**, configure:
   - Select **Burstable classes (includes t classes)**.
   - Select *db.t3.micro*
8. Under **Storage**, configure:
   - **Storage type:** *General Purpose (SSD)*

- **Allocated storage:** *20*

9. Under **Connectivity**, configure:
   - **Virtual Private Cloud (VPC):** *Lab VPC*
10. Under **Existing VPC security groups**, from the dropdown list:
    - Choose *DB Security Group*.
    - Deselect *default*.
11. Expand **Additional configuration**, then configure:
    - **Initial database name:** `lab`
    - Uncheck **Enable automatic backups**.
    - Uncheck **Enable encryption**
    - Uncheck **Enable Enhanced monitoring**.

    This will turn off backups, which is not normally recommended, but will make the database deploy faster for this lab.
12. Choose Create database

    Your database will now be launched.

    If you receive an error that mentions "not authorized to perform: iam:CreateRole", make sure you unchecked *Enable Enhanced monitoring* in the previous step.
13. Choose **lab-db** (choose the link itself).

    You will now need to wait **approximately 4 minutes** for the database to be available. The deployment process is deploying a database in two different Availability zones.

    While you are waiting, you might want to review the Amazon RDS FAQs or grab a cup of coffee.
14. Wait until **Info** changes to **Modifying** or **Available**.
15. Scroll down to the **Connectivity & security** section and copy the **Endpoint** field.

    It will look similar to: *lab-db.cggq8lhnxvnv.us-west-2.rds.amazonaws.com*
16. Paste the Endpoint value into a text editor. You will use it later in the lab.

Task 4: Interact with Your Database

In this task, you will open a web application running on your web server and configure it to use the database.

1. To copy the **WebServer** IP address, choose on the Details drop down menu above these instructions, and then choose Show.
2. Open a new web browser tab, paste the *WebServer* IP address and press Enter.

   The web application will be displayed, showing information about the EC2 instance.
3. Choose the **RDS** link at the top of the page.

   You will now configure the application to connect to your database.
4. Configure the following settings:
   - **Endpoint:** Paste the Endpoint you copied to a text editor earlier
   - **Database:** `lab`
   - **Username:** `main`
   - **Password:** `lab-password`
   - Choose **Submit**

   A message will appear explaining that the application is running a command to copy information to the database. After a few seconds the application will display an **Address Book**.

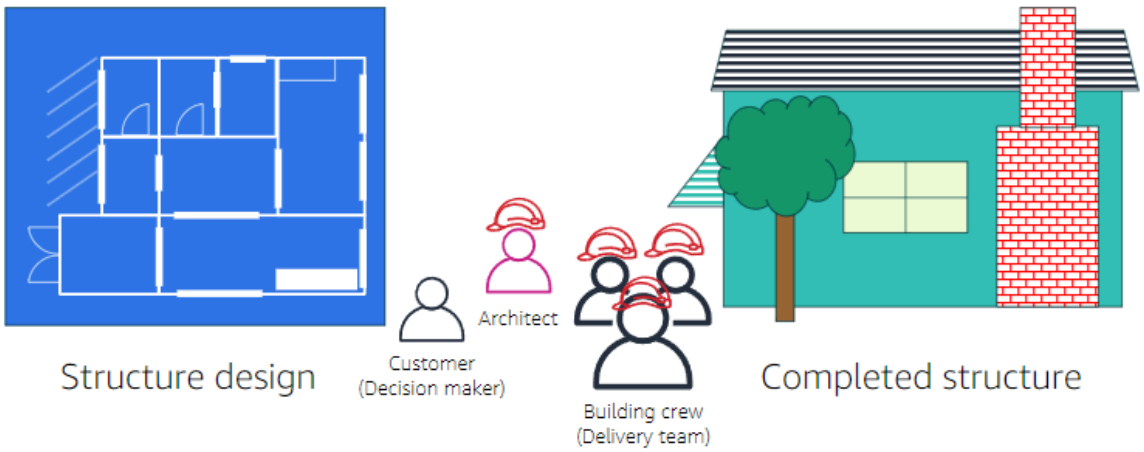
   The Address Book application is using the RDS database to store information.
5. Test the web application by adding, editing and removing contacts.

   The data is being persisted to the database and is automatically replicating to the second Availability Zone.

Module 9: Cloud Architecture

Section 1: **AWS Well-Architected Framework Design Principles**

Architecture: designing and building



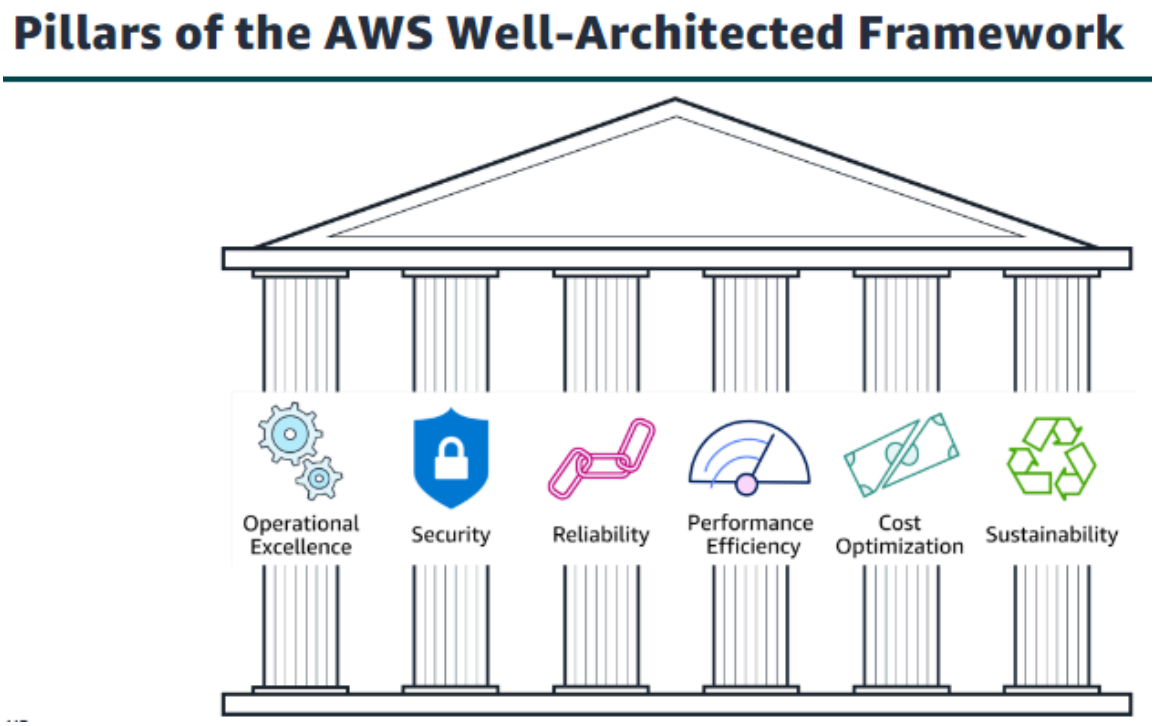
Architecture is the art and science of designing and building large structures. Large systems require architects to manage their size and complexity. Cloud architects:•Engage with decision makers to identify the business goal and the capabilities that need improvement. •Ensure alignment between technology 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.



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 accessibility:The pillars include operational excellence, security, reliability, performance efficiency, cost optimization, and sustainability. End of accessibility description.

Pillar organization

Best practice area — **Identity and Access Management**

Question text — **SEC 1: How do you manage credentials and authentication?**

Question context — 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.

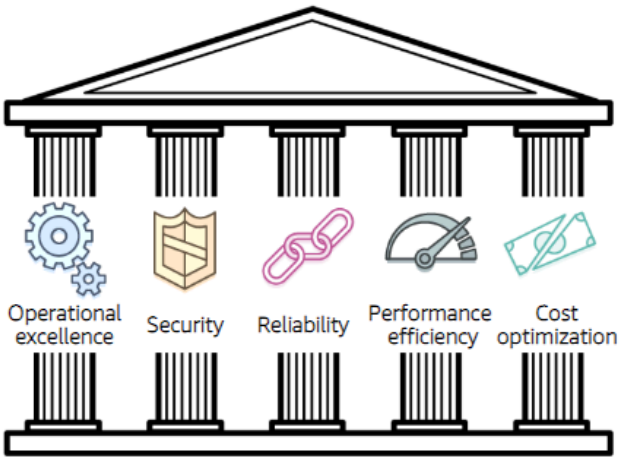
Best practices

Best practices:

- Define requirements for identity and access management
- Secure AWS account root user
- Enforce use of multi-factor authentication
- Automate enforcement of access controls
- Integrate with centralized federation provider
- Enforce password requirements
- Rotate credentials regularly
- Audit credentials periodically

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

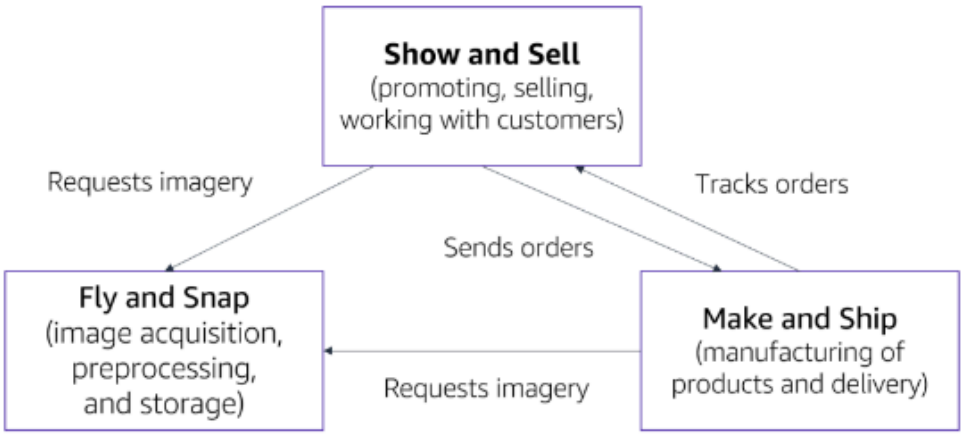


AnyCompany background

- 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

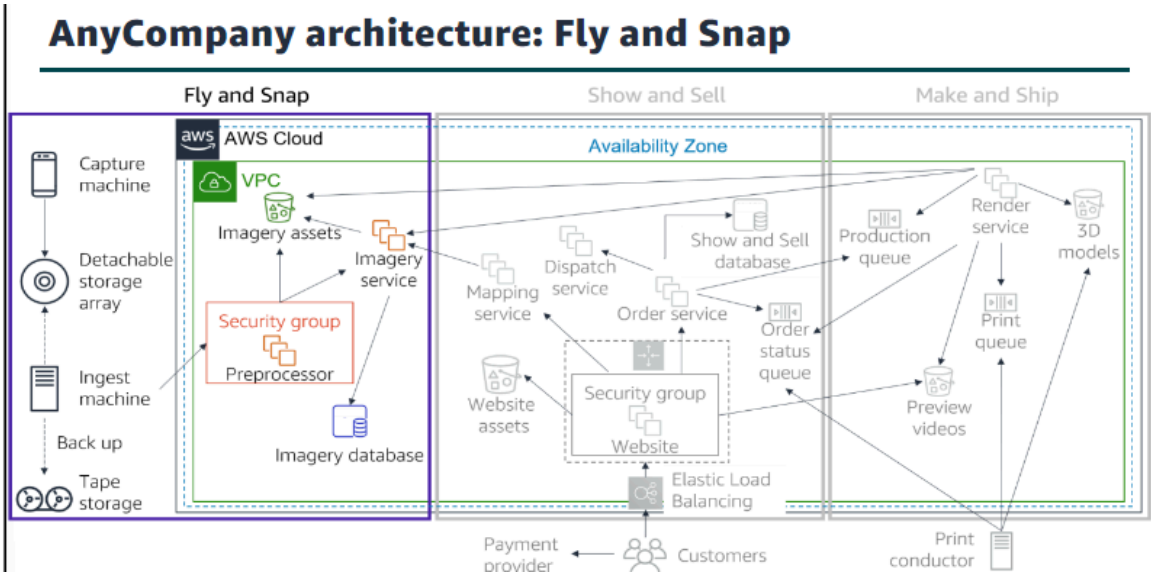
AnyCompanyCorporation was founded in 2008 by JohnDoe. It sells high-quality three-dimensional (3D) printed cityscapes of neighborhoods that enable you to see individual buildings and trees. The cityscapes are printed in color, with brickwork, roofs, gardens, and even cars in their correct coloration.The 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 AnyCompanyplatform 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)



AnyCompanyCorporation 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 deliveryThe high-level design for the AnyCompanyplatform looks like the organizational structure of the company.

For accessibility:High-level design of mappahood platform: Show and Sell (promoting, selling, working with customers) sends orders and requests imagery. Make and Ship (manufacturing and delivery) tracks orders from Show and Sell and requests imagery from Fly and Snap (acquisition, preprocessing, and storage. End of accessibility description.



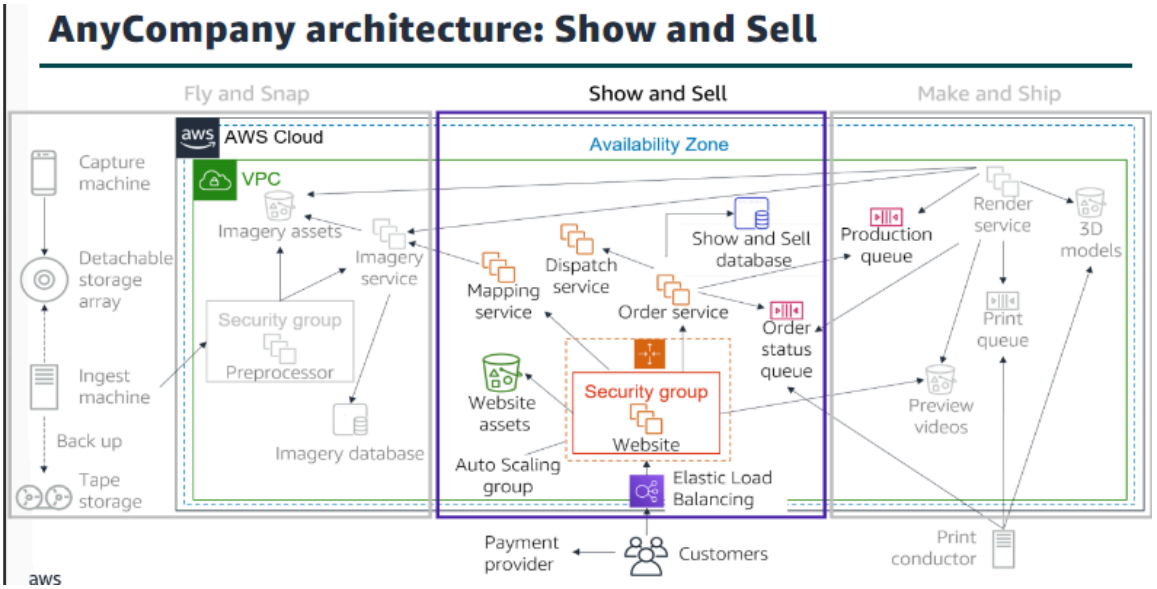
Fly and Snap

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 machinethat has an external storage array. The Capture machine is also connected to the aircraft's flight system and continuously captures navigation data—suchas 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.

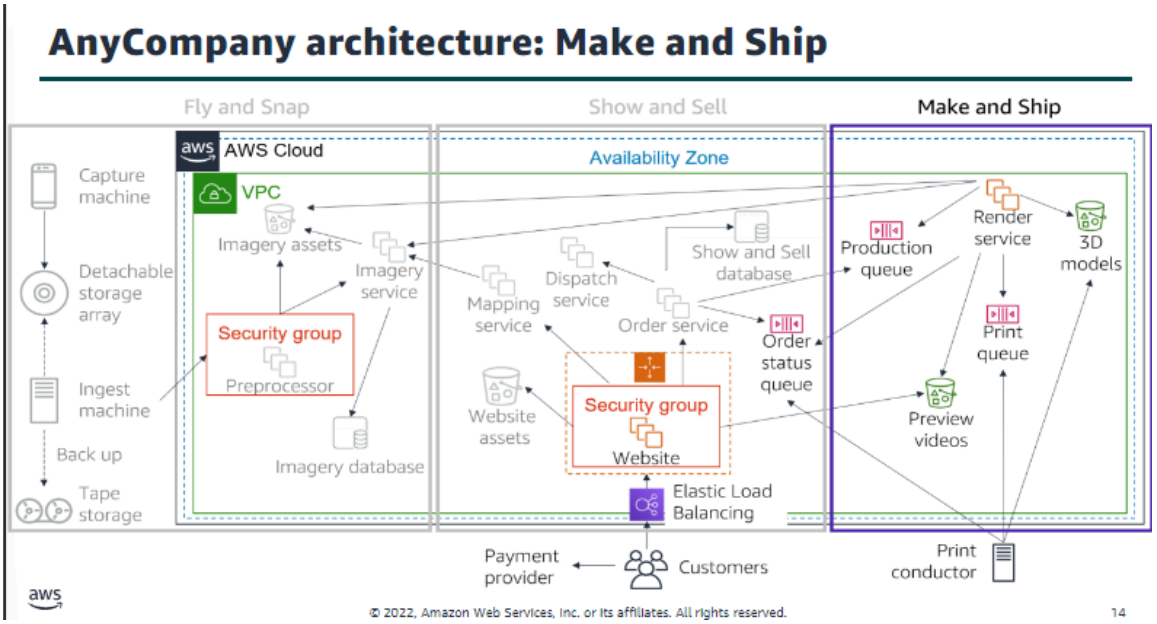
For accessibility:Fly and snap architecture. A capture machine and an ingest machine sends imagery to a detachable storage array. The ingest machine also backs up to tape storage. End of accessibility description.



Show and Sell

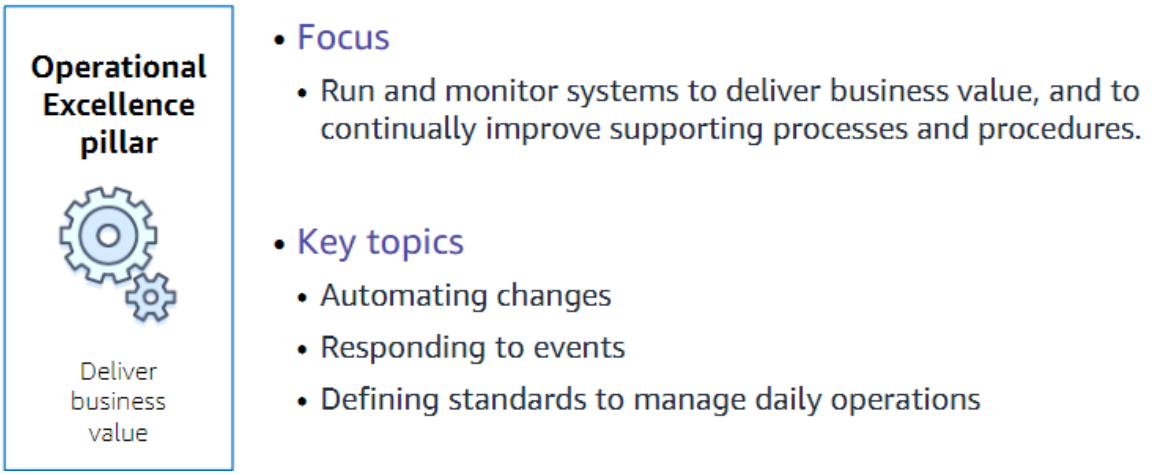
When customers visit the AnyCompanywebsite,they can see images and videos of the physical product. These images are in a variety of formats (forexample, alarge-scale, walk-around map). The websiteuses Elastic Load Balancingwith Hypertext Transfer Protocol Secure (HTTPS), and an Auto Scaling groupof EC2 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 servicecorrelates 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. AnyCompanydoes not process or store any credit card information.After the websitereceives payment confirmation, it instructs the Order serviceto push the order to production. Orders (including customer details) are recorded in the Show and Sell database, which is anRDBMSthat is based in Amazon EC2. To initiate a video preview or full print of an order, the Orders serviceplaces a message on the Production queue, which allows the Render serviceto indicate when a preview video is available. The Order servicealso reads from the Order status queueand 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 thirdparty through the broker Dispatch service.



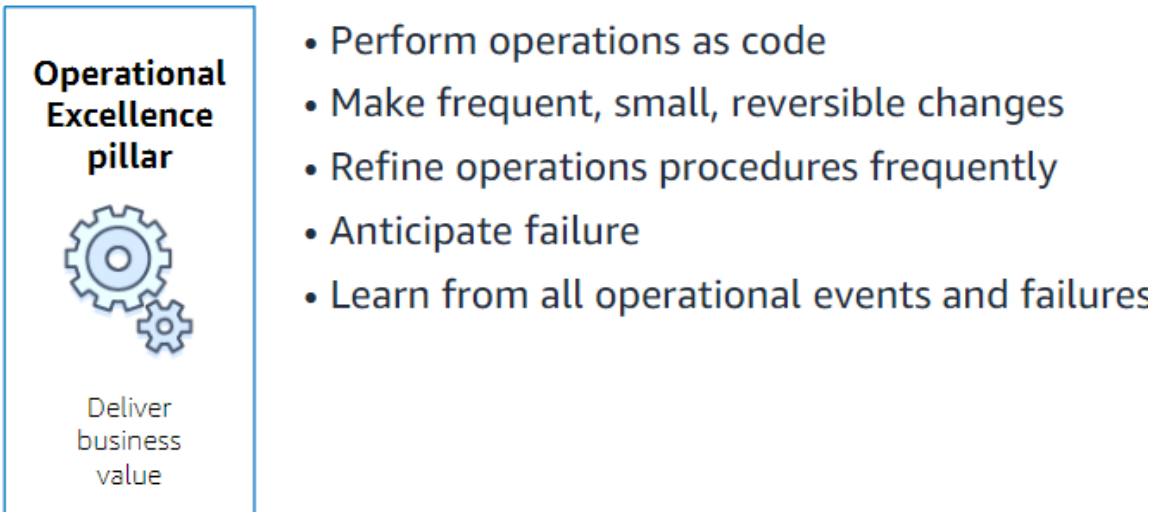
Make and ShipAnyCompanyhas proprietary technology that enables it to generate 3D models from a combination of photographs and video (extracting structure from motion).The Render serviceis a fleet of g2.2xlarge instances. The Render servicetakes orders from the Production queueand generates the 3D models that are stored in an S3 bucket. The Render servicealso uses the 3D models to create flyby videos so that customers can preview their orders on the AnyCompanywebsite. 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 queuewith 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 AnyCompanywebsite, 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 conductormachine takes orders from the Print queueand sends them to the next available printer. The Print conductorsends order updates to the Order status queue. The Print conductorsends a final update when the order has been completed, passed quality assurance, and is ready for dispatch.

Operational Excellence pillar – deliver business value



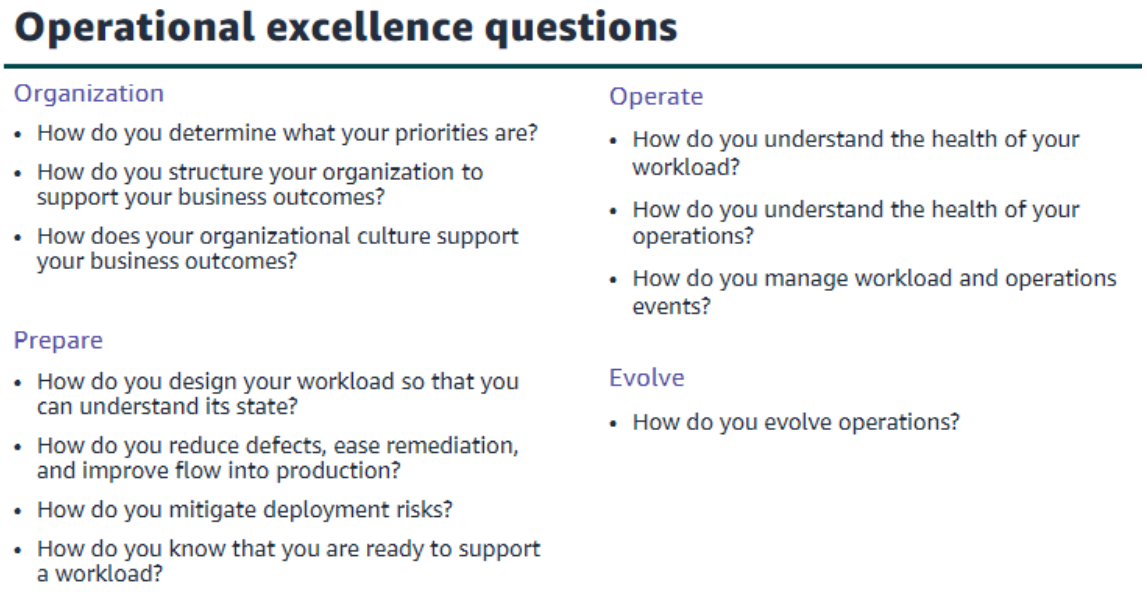
The Operational Excellence pillar focuses on the ability to run and monitor systems to deliver business value,and to continually improve supporting processes and procedures. Key topics include: automating changes, responding to events, and defining standards to manage daily operations.

Operational excellence design principles



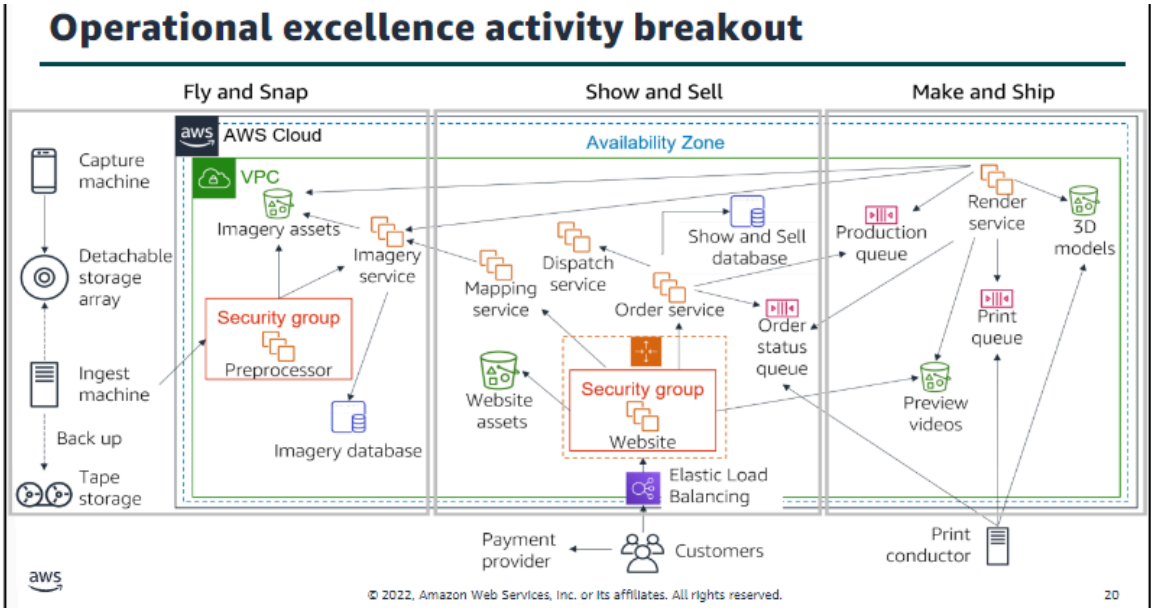
There are five design principles for operational excellence in the cloud:•Perform operations as code–Define your entire workload (that is, applications and infrastructure) as code and update it with code. Implement operations procedures as code and configure them to automatically trigger in response to events. By performing operations as code, you limit human

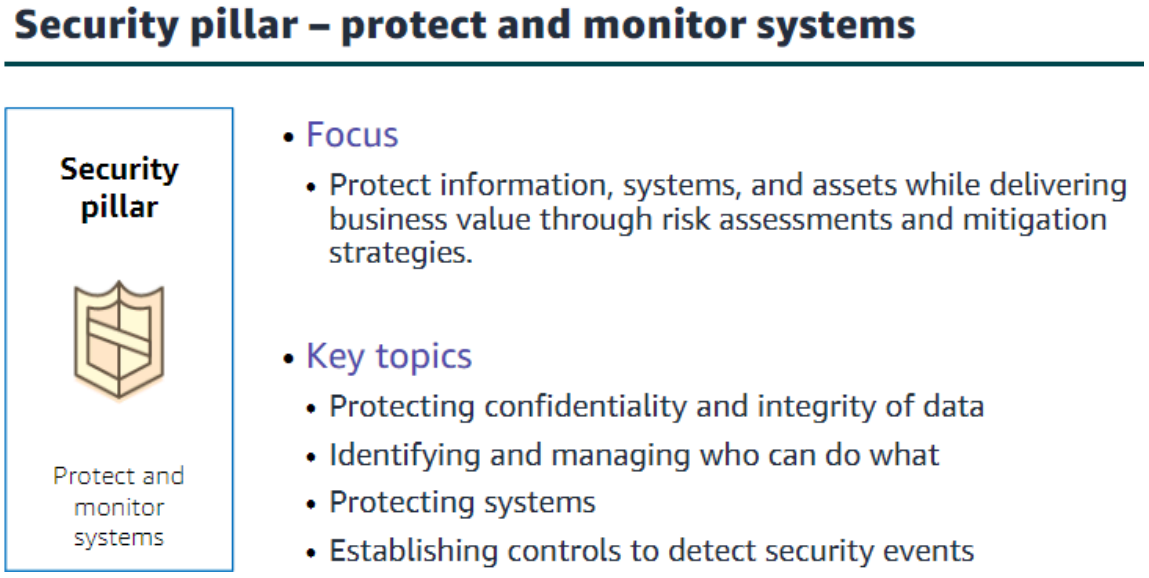
error and enable consistent responses to events.•Make frequent, small, reversible changes–Design workloads to enable components to be updated regularly. Make changes in small increments that can be reversed if they fail (without affecting customers when possible).•Refine operations procedures frequently–Look 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.



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

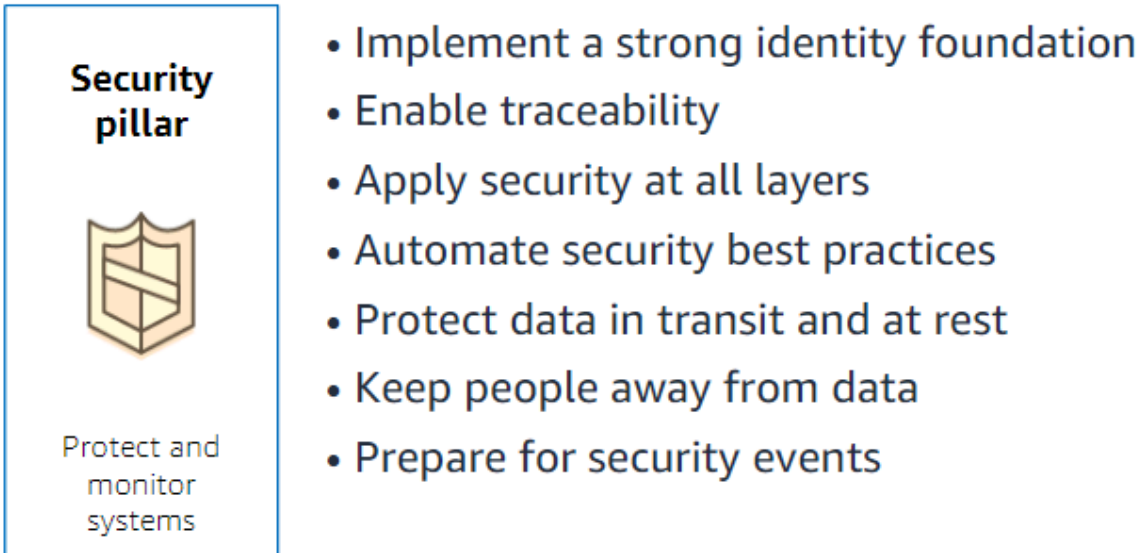
needs, change over time, it's important to design operations that evolve in response to change and to incorporate lessons learned through their performance.





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.

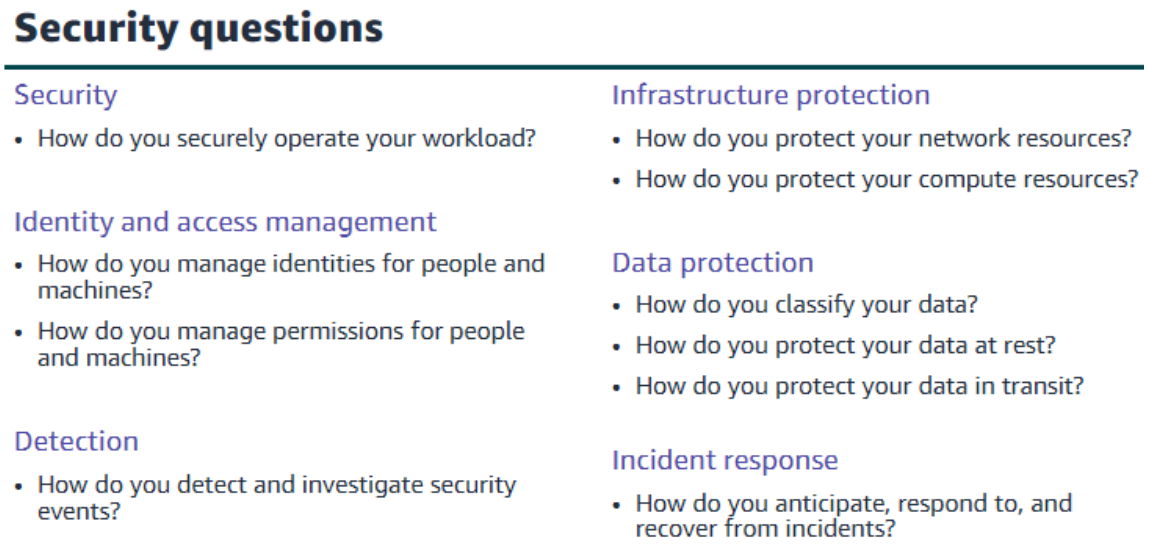
Security design principles



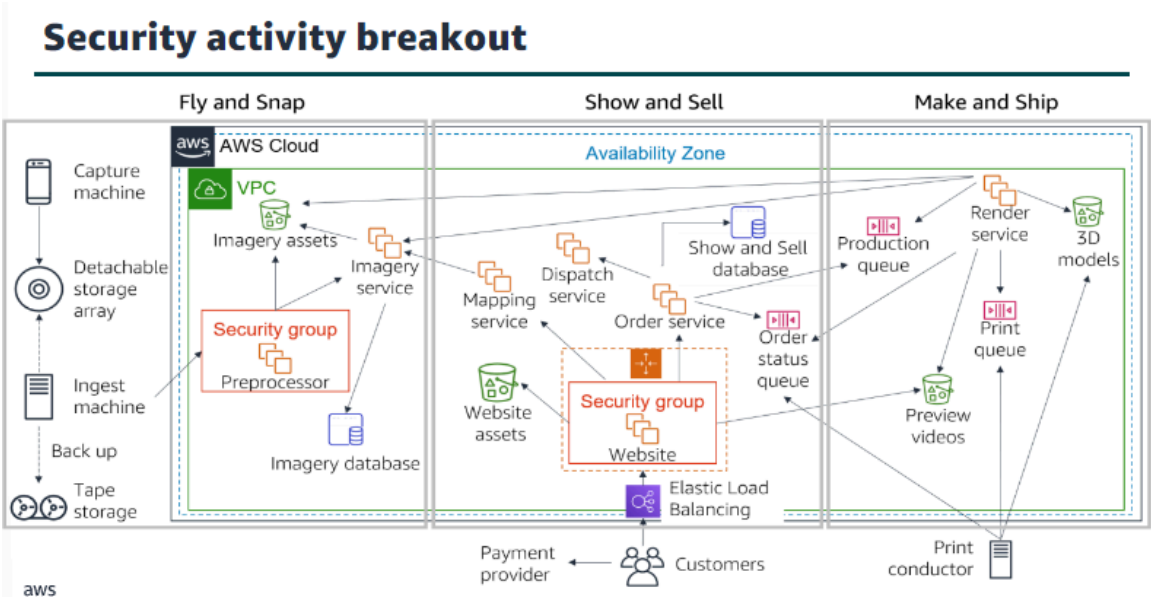
There are seven design principlesthat can improve security:

- 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 defenseindepth and apply security controls to all layers of your architecture (forexample,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.



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.



AWS_Well_Architectured_Framework_2020.pdf

Reliability pillar – recover from failure and mitigate disruption

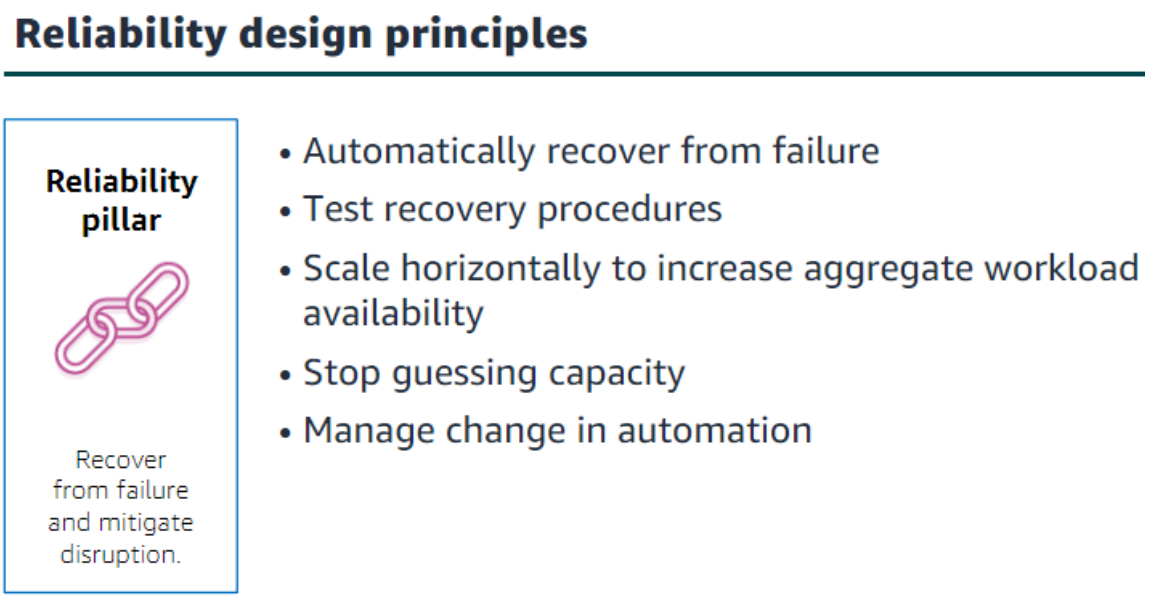
Reliability pillar

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

TheReliability 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.



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

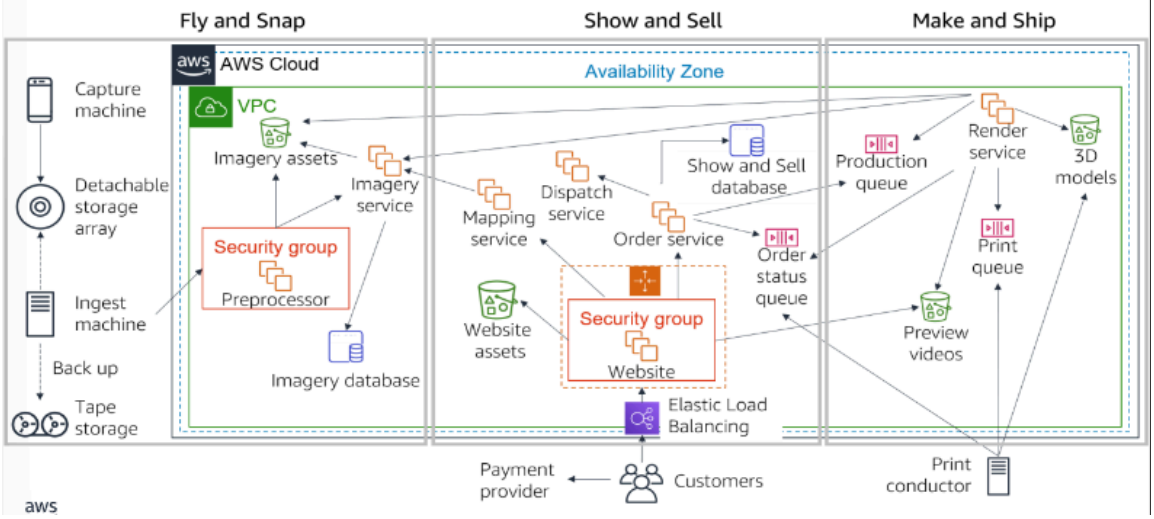
- How do you monitor workload resources?
- How do you design your workload to adapt to changes in demand?
- How do you implement change?

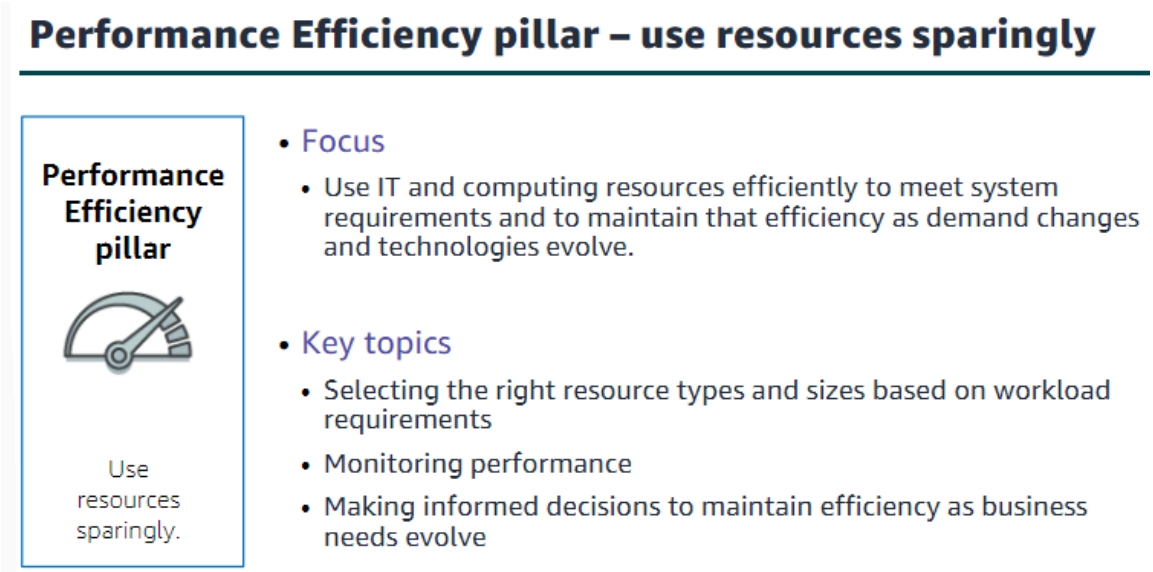
Failure management

- 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?

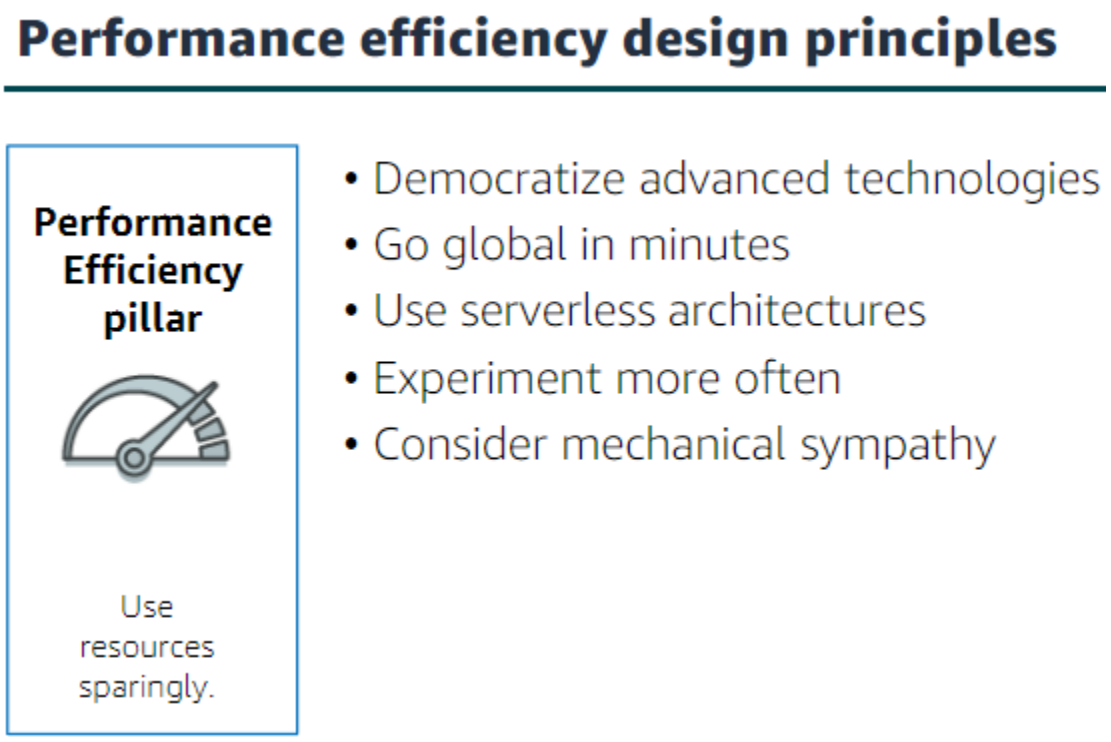
The foundational questions for reliability fall under four best practice areas: foundations, workload architecture, change management, and failure management. To achieve reliability, a system must have both a well-planned foundation and monitoring in place. It must have mechanisms for handling changes in demand or requirements. The system should be designed to detect failure and automatically heal itself.

Activity breakout





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.



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 selectapproaches fordatabases 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

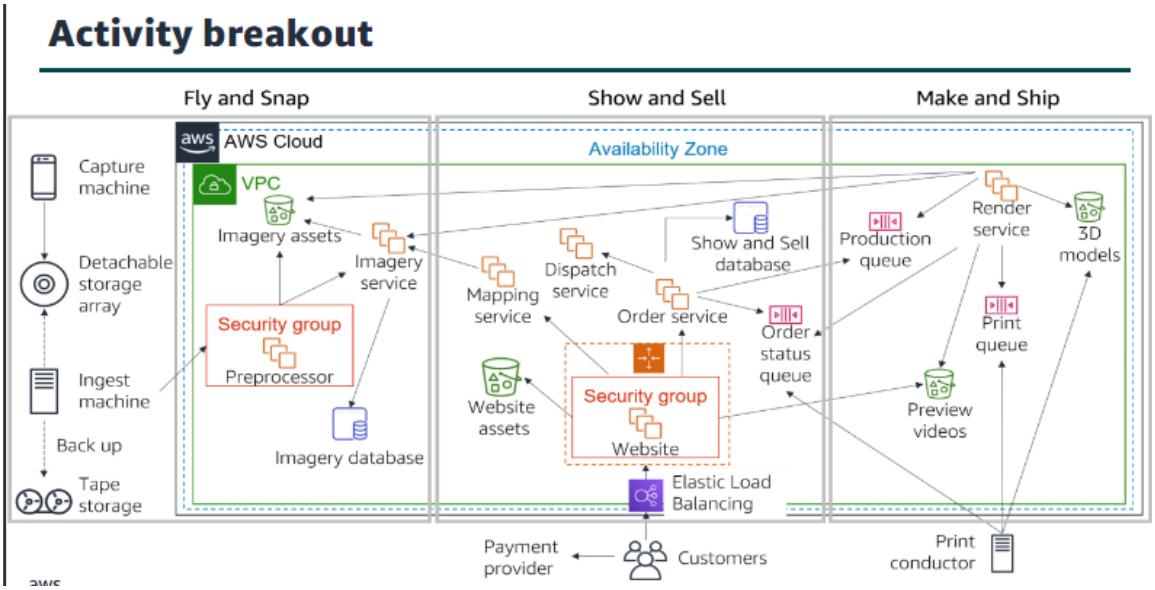
- How do you monitor your resources to ensure they are performing?

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 remediatethem. Finally, use tradeoffs in your architecture to improve performance, such as using compression, using caching, or relaxing consistency requirements.



Cost Optimization pillar – eliminate unneeded expense

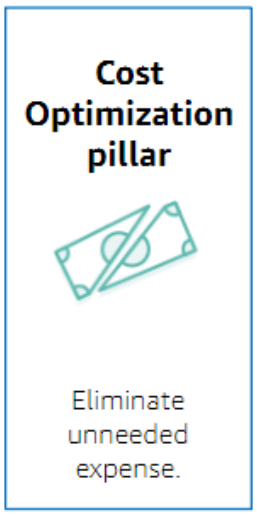
Cost Optimization pillar

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



- 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 meansthat 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

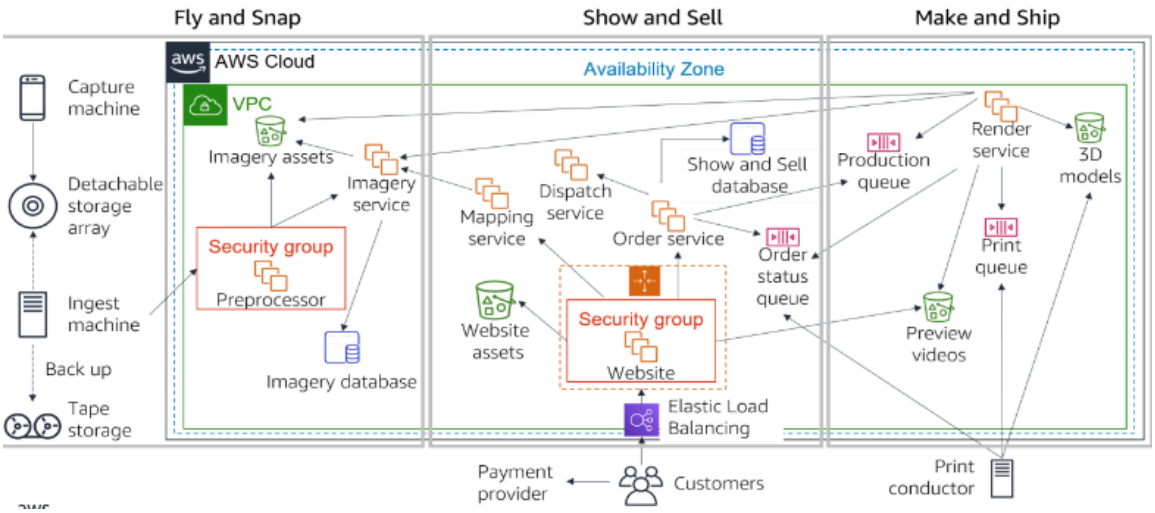
- How do you manage demand and supply resources?

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



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 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.

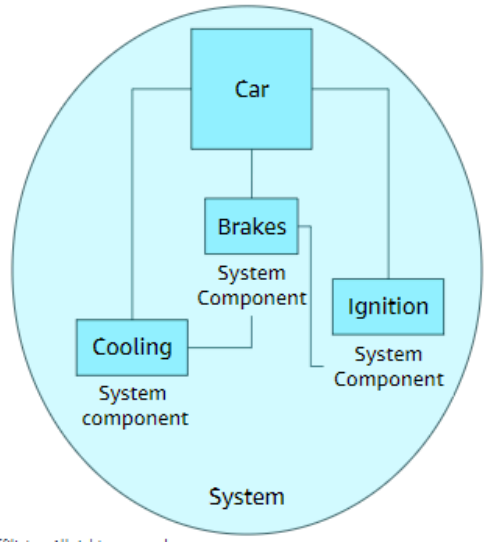
The AWS Well-Architected Tool provides a consistent process for you to review and measure your cloud architectures. You can use the results that the tool provides to identify next steps for improvement, drive architectural decisions, and bring architecture considerations into your corporate governance process.

Section 2: Reliability and Availability

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.

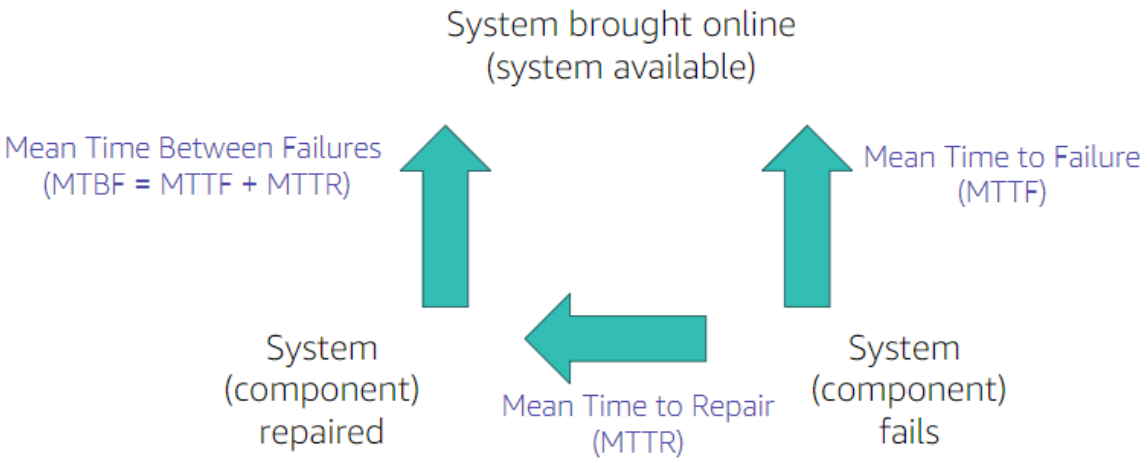
Reliability

- 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



Reliabilityis 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



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

Formally, availabilityis 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.

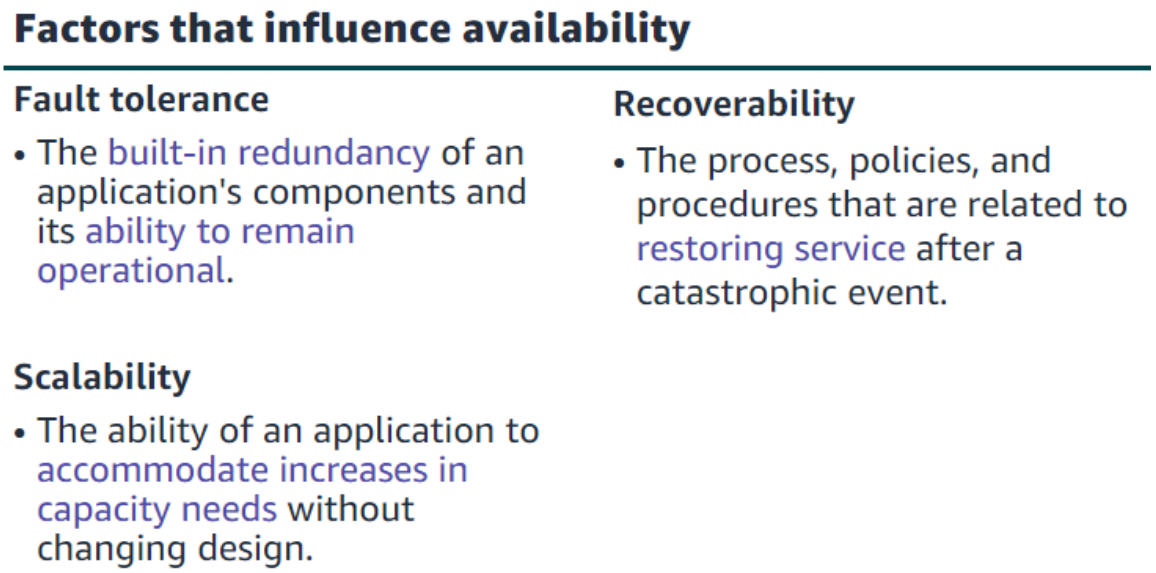
A highly availablesystem is one that can withstand some measure of degradation while still remaining available. In a highly available system, downtime is minimized as much as possible and minimal human intervention is required. A highly available system can be viewed as a set of system-wide, shared resources 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.

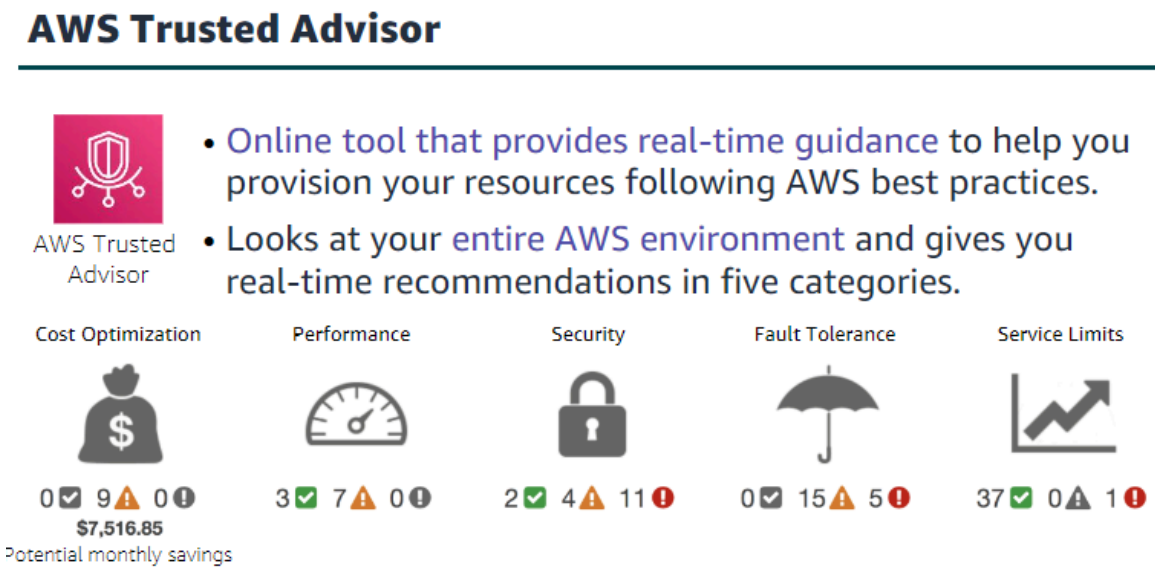
For accessibility: Availability tiers with max disruption per year and application categories. The tiers range from availabilities of 99 percent to 99.999 percent. End of accessibility description.



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 tolerancerefers to the built-in redundancy of an application's components and the ability of the application to remain operationaleven 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. •Scalabilityis 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.•Recoverabilityis 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.

Section 3: AWS Trusted Advisor



AWS Trusted Advisor is an online tool that provides real-time guidance to help you provision your resources following AWS best practices. AWSTrusted 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 80percentof 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.

Activity: Interpret AWS Trusted Advisor 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.

Activity: Recommendation #3

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	WebServerSG	sg-xxxxxxx1 (vpc-xxxxxxx1)	tcp	22	Red	0.0.0.0/0
us-west-2	DatabaseServerSG	sg-xxxxxxx2 (vpc-xxxxxxx2)	tcp	8080	Red	0.0.0.0/0

Security groups information including Region, security group name, security group ID, protocol, port, status, and IP range from 2 sample security groups. The status of both is red.

Activity: Recommendation #4

Amazon EBS Snapshots

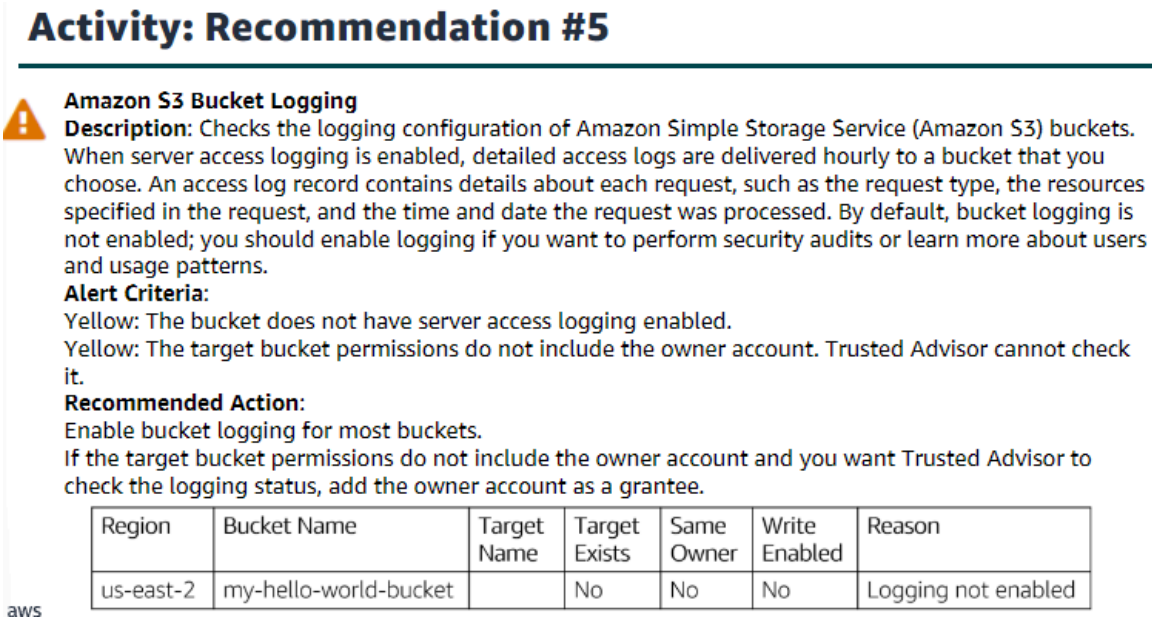
Description: Checks the age of the snapshots for your Amazon Elastic Block Store (Amazon EBS) volumes (available or in-use). Even though Amazon EBS volumes are replicated, failures can occur. Snapshots are persisted to Amazon Simple Storage Service (Amazon S3) for durable storage and point-in-time recovery.

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

Amazon EBS volume information for Region us-east-1 including volume ID, volume name, volume attachment, status (listed as red), and reason (listed as no snapshot).



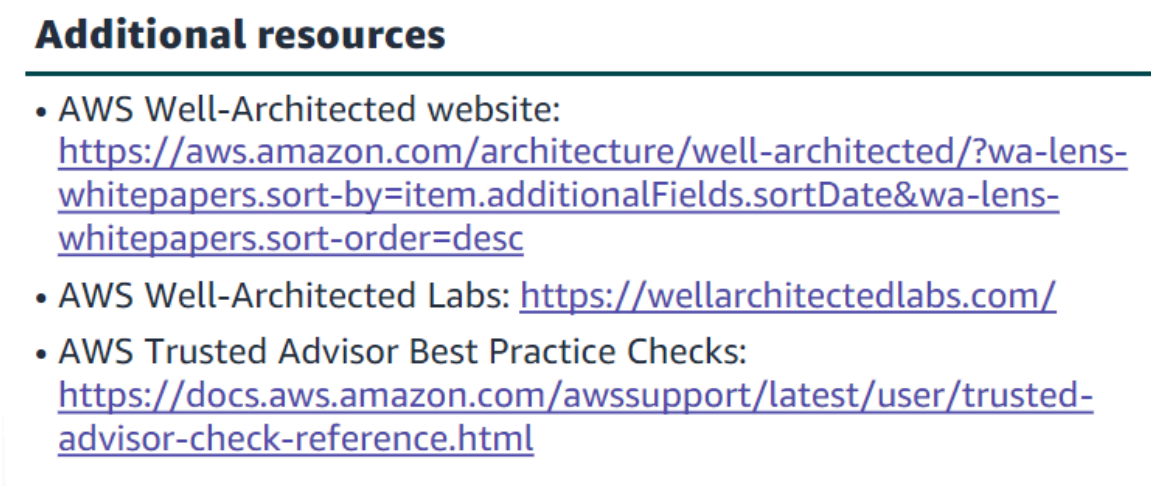
Activity: Recommendation #5

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

aws

Amazon S3bucket logging information for Region us-east-2 including bucket name, target exists (no), same owner (no), write enabled (no), and reason (logging not enabled).



Additional resources

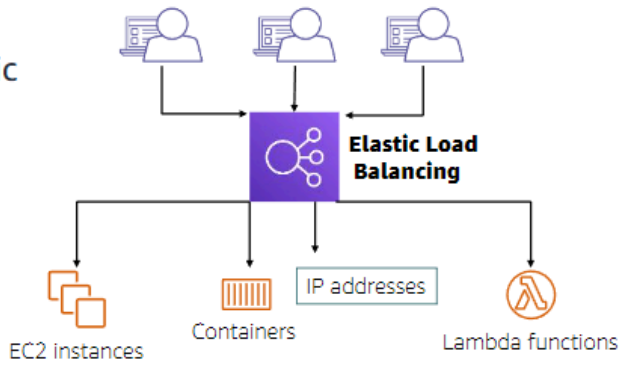
- AWS Well-Architected website: https://aws.amazon.com/architecture/well-architected/?wa-lens-whitepapers.sort-by=item.additionalFields.sortDate&wa-lens-whitepapers.sort-order=desc
- AWS Well-Architected Labs: https://wellarchitectedlabs.com/
- AWS Trusted Advisor Best Practice Checks: https://docs.aws.amazon.com/awssupport/latest/user/trusted-advisor-check-reference.html

Module 10: Automatic Scaling and Modeling

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.



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.

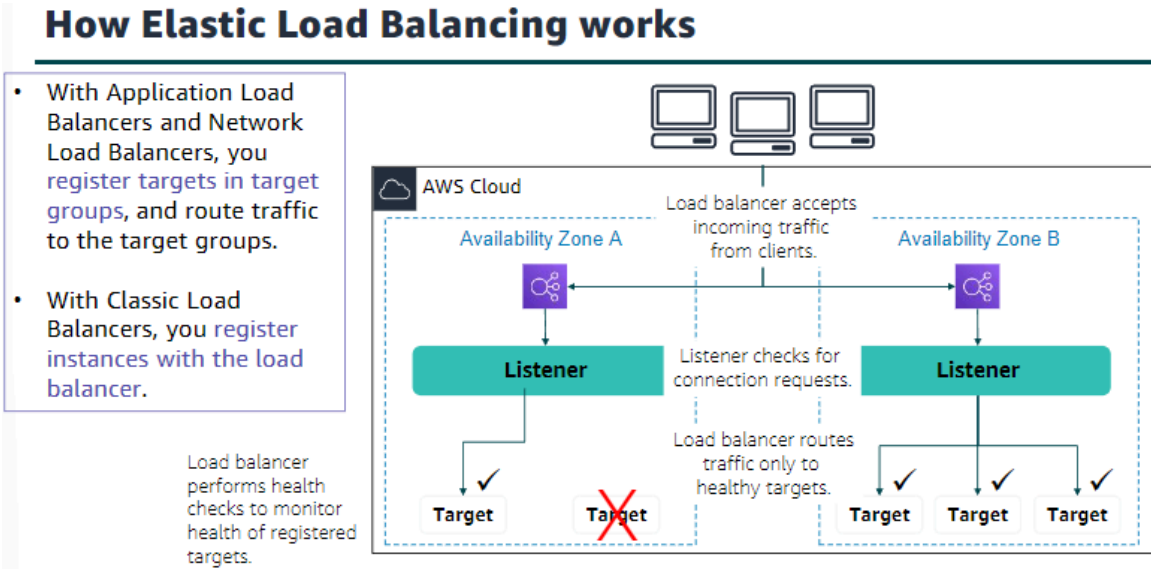
Elastic Load Balancing is an AWS service that distributes incoming application or network traffic across multiple targets—such as Amazon Elastic Compute Cloud (Amazon EC2) instances, containers, internet protocol (IP) addresses, and Lambda functions—in a single Availability Zone or across multiple Availability 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

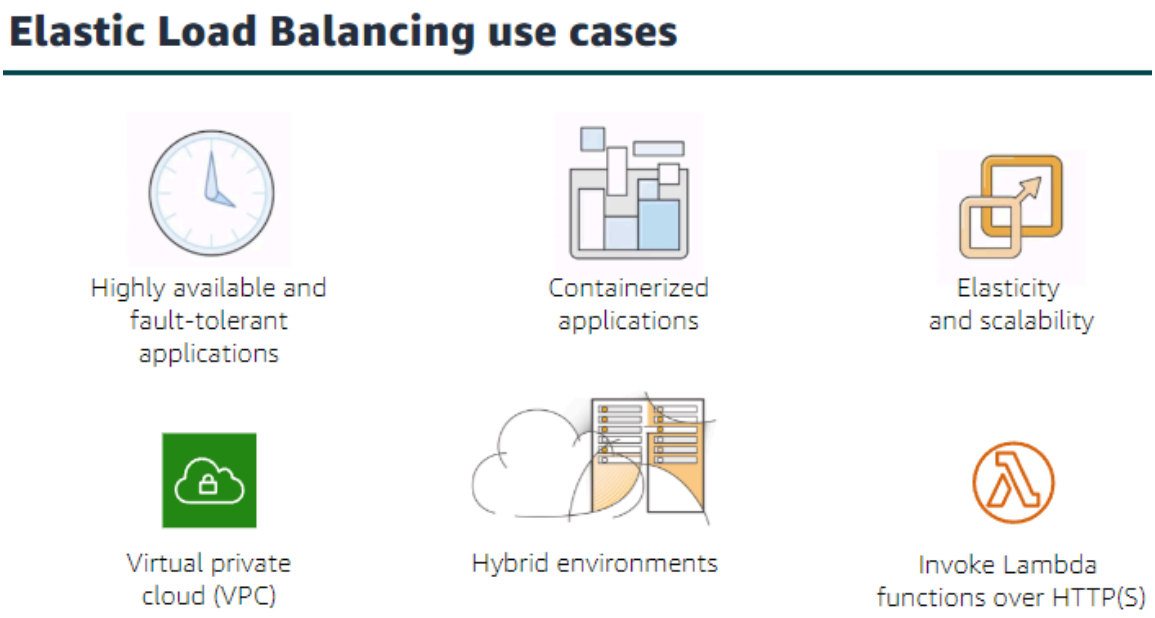
Application Load Balancer	Network Load Balancer	Classic Load Balancer (Previous Generation)
• Load balancing of HTTP and HTTPS traffic	• Load balancing of TCP, UDP, and TLS traffic where extreme performance is required	• Load balancing of HTTP, HTTPS, TCP, and SSL traffic
• Routes traffic to targets based on content of request • Provides advanced request routing targeted at the delivery of modern application architectures, including microservices and containers	• Routes traffic to targets based on IP protocol data • Can handle millions of requests per second while maintaining ultra-low latencies • Is optimized to handle sudden and volatile traffic patterns	• Load balancing across multiple EC2 instances
• Operates at the application layer (OSI model layer 7)	• Operates at the transport layer (OSI model layer 4)	• Operates at both the application and transport layers.

Elastic Load Balancing is available in three types:

- An Application Load Balanceroperates 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 Balanceroperates at the network transport level (OSImodel layer 4), routing connections to targets—EC2instances, 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 Balancerprovides basic load balancing across multiple EC2instances, 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.



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.



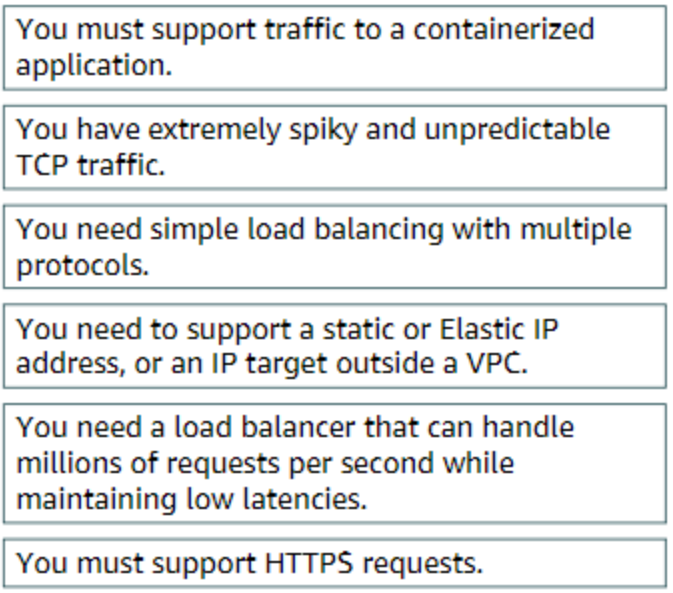
There are many reasons to use a load balancer: •Achieve high availability and better fault tolerance for your applications–Elastic Load Balancing balances traffic across healthy targets in multiple Availability Zones. If one or more of your targets in a single Availability Zone are unhealthy, Elastic 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.•Enablehybrid 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 resourcesbyusing 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.

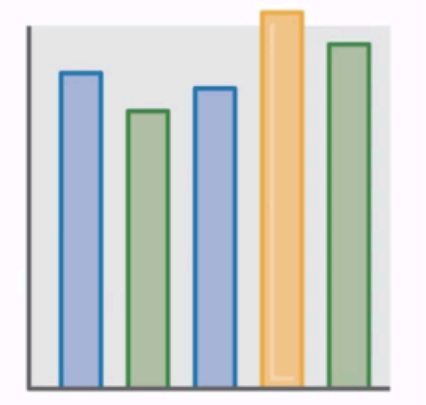
Activity: Elastic Load Balancing



Activity: Elastic Load Balancing Answers

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

Load balancer monitoring

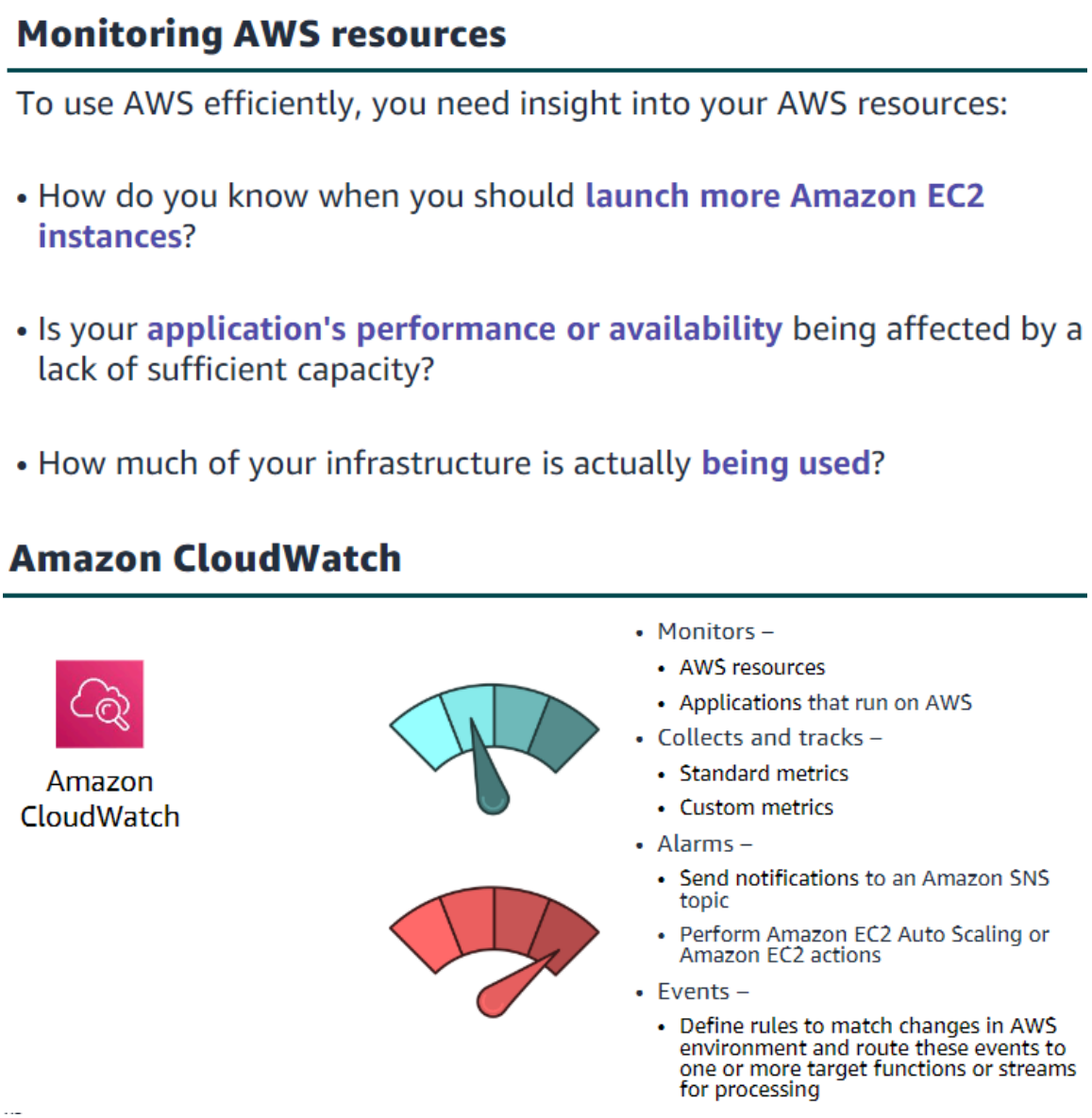


- **Amazon CloudWatch metrics** – Used to verify that the system is performing as expected and creates an alarm to initiate an action if a metric goes outside an acceptable range.
- **Access logs** – Capture detailed information about requests sent to your load balancer.
- **AWS CloudTrail logs** – Capture the who, what, when, and where of API interactions in AWS services.

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 timeseries 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.

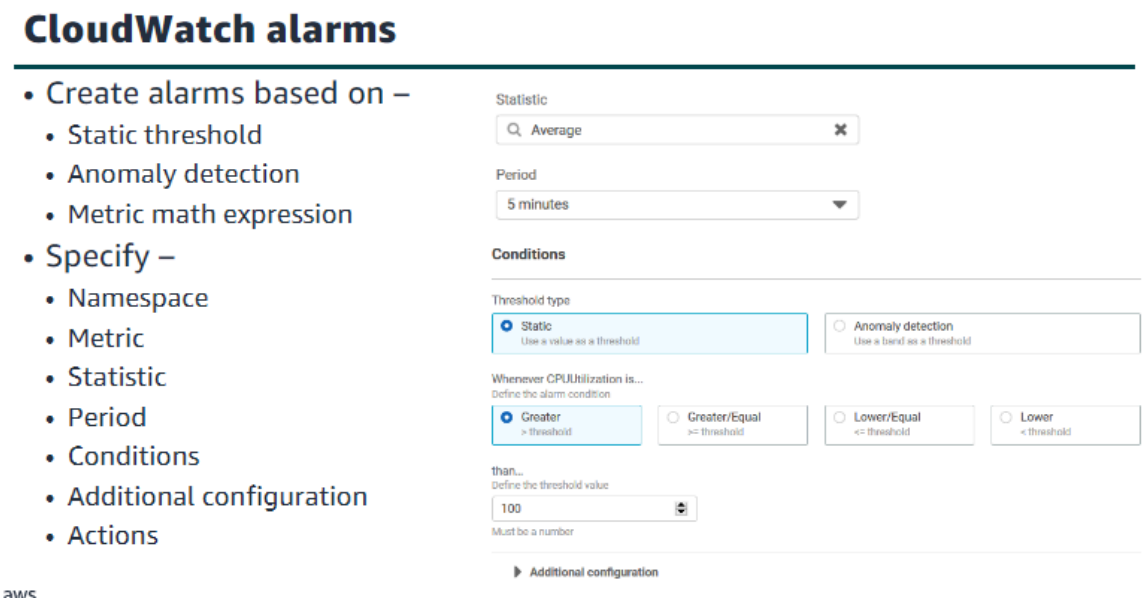
Section 2: Amazon CloudWatch



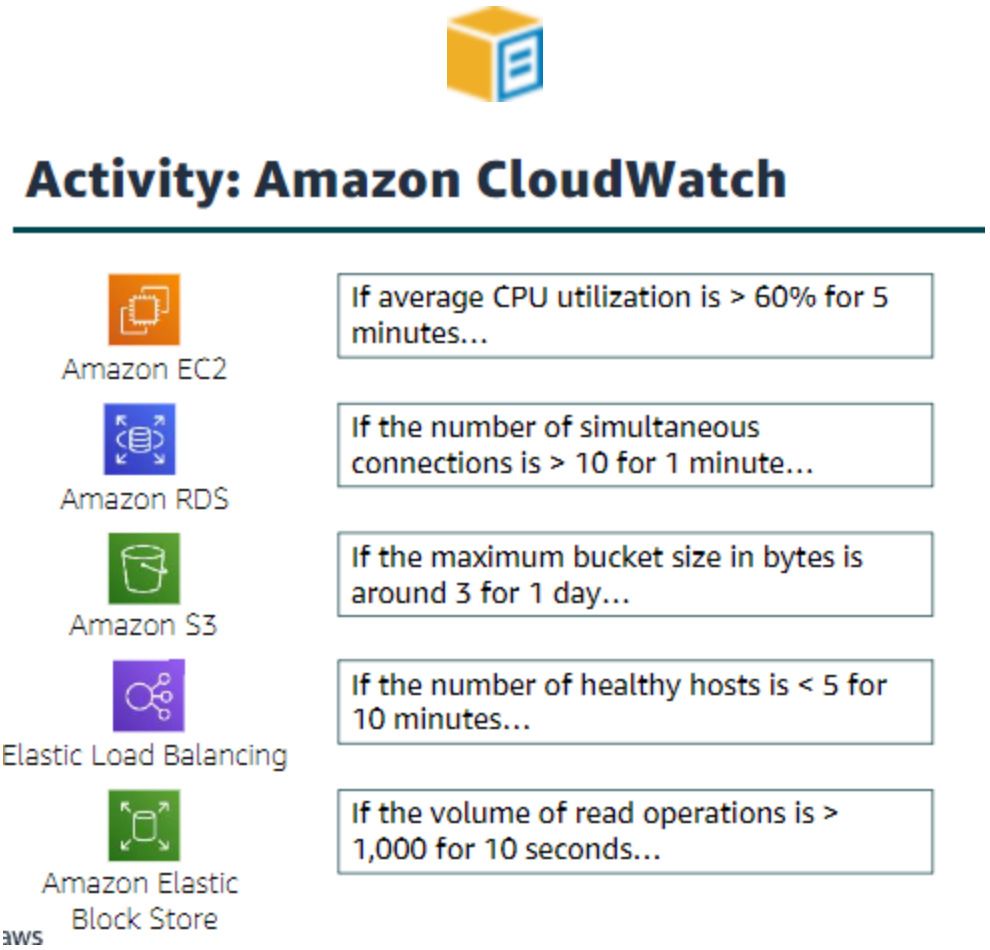
Amazon CloudWatch is a monitoring and observability service that is built for DevOps engineers, developers, site reliability engineers (SRE), and IT managers. CloudWatch monitors your AWS resources (and the

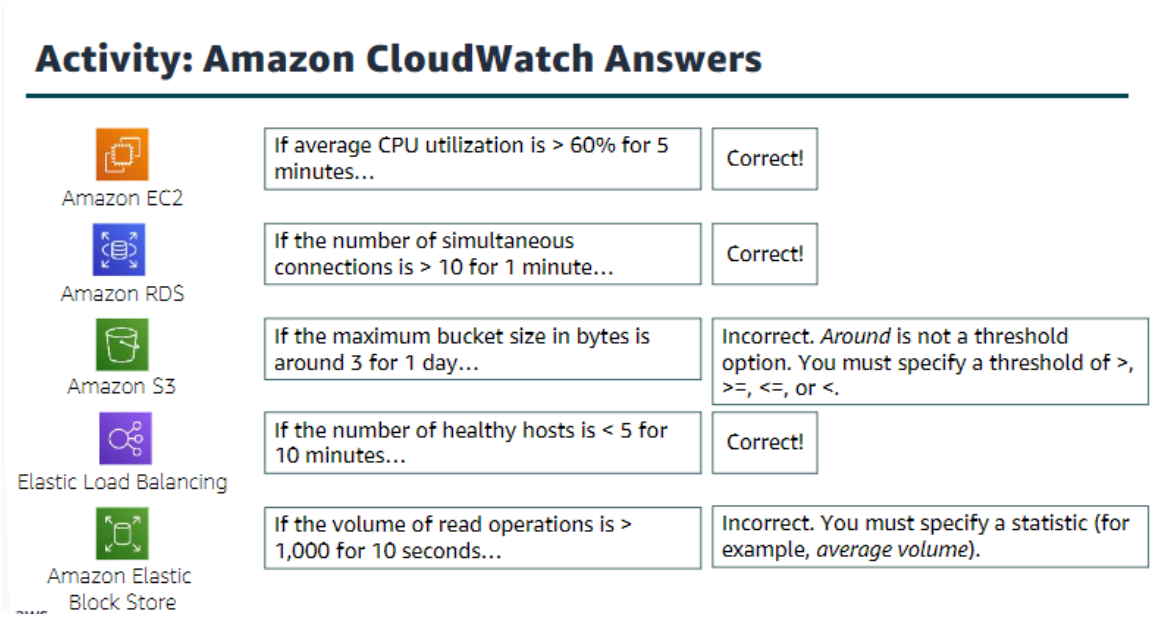
applicationsthat you run on AWS) in real time. You can use CloudWatch to collect and track metrics, which are variables that you can measure for your resources and applications. You can create an alarm to monitor any Amazon CloudWatch metric in your account and use the alarm to automatically send a notification 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.



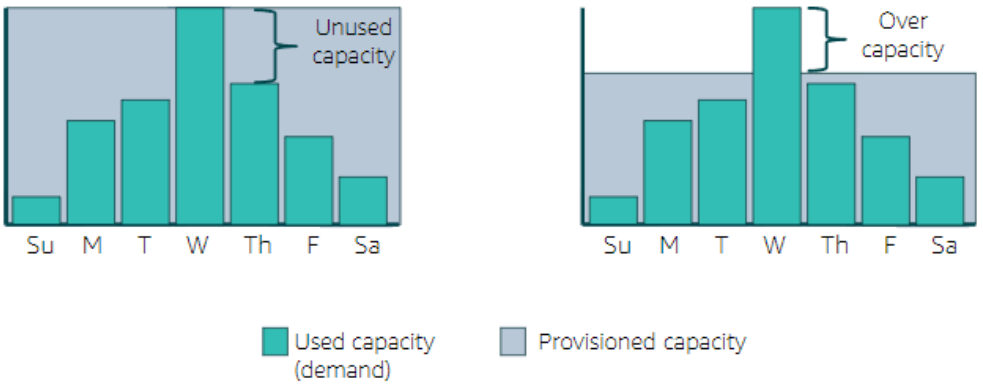
You can create a CloudWatch alarm that watches a single CloudWatch metric or the result of a math expression based on CloudWatch metrics. You can create a CloudWatch alarm based on a static threshold, anomaly detection, or a metric math expression.When you create an alarm based on a static threshold, 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 Lowerthan 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.





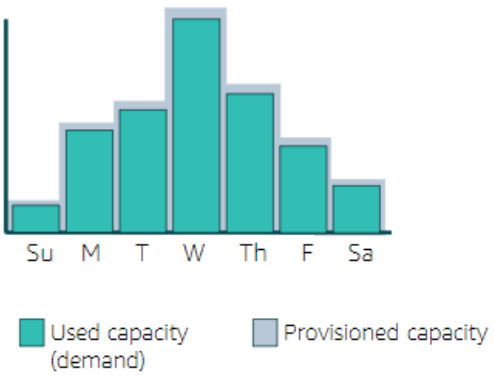
Section 3: Amazon EC2 Auto Scaling

Why is scaling important?



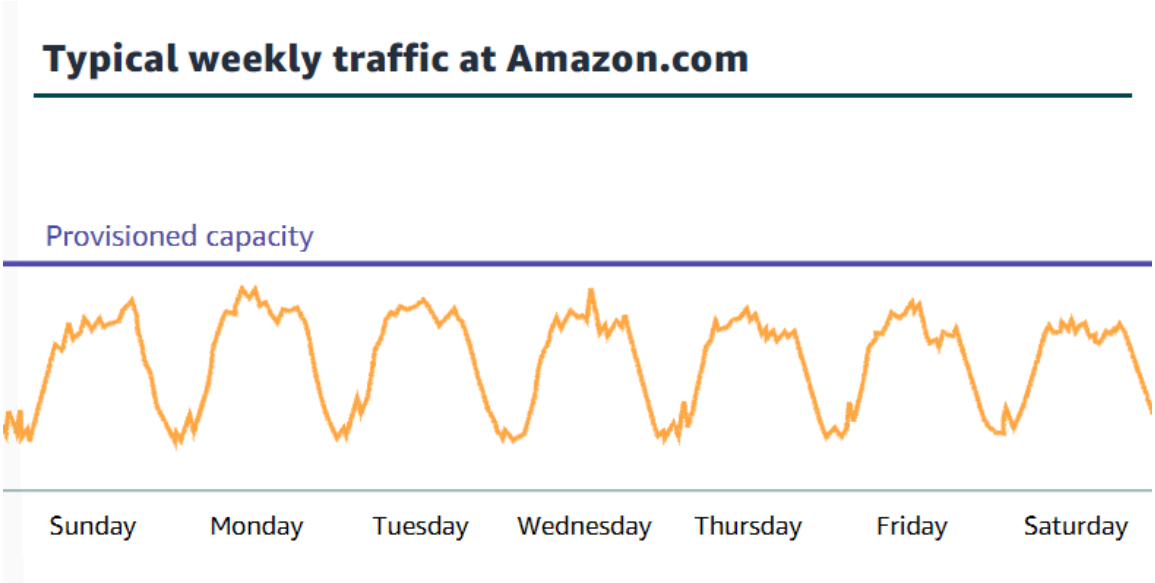
Scaling is the ability to increase or decrease the compute capacity of your application. To understand why scaling is important, consider this example of a workload that has varying resource requirements. In this example, the most resource capacity is required on Wednesday, and the least resource capacity is required on Sunday. One option is to allocate more than enough capacity so youcan 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

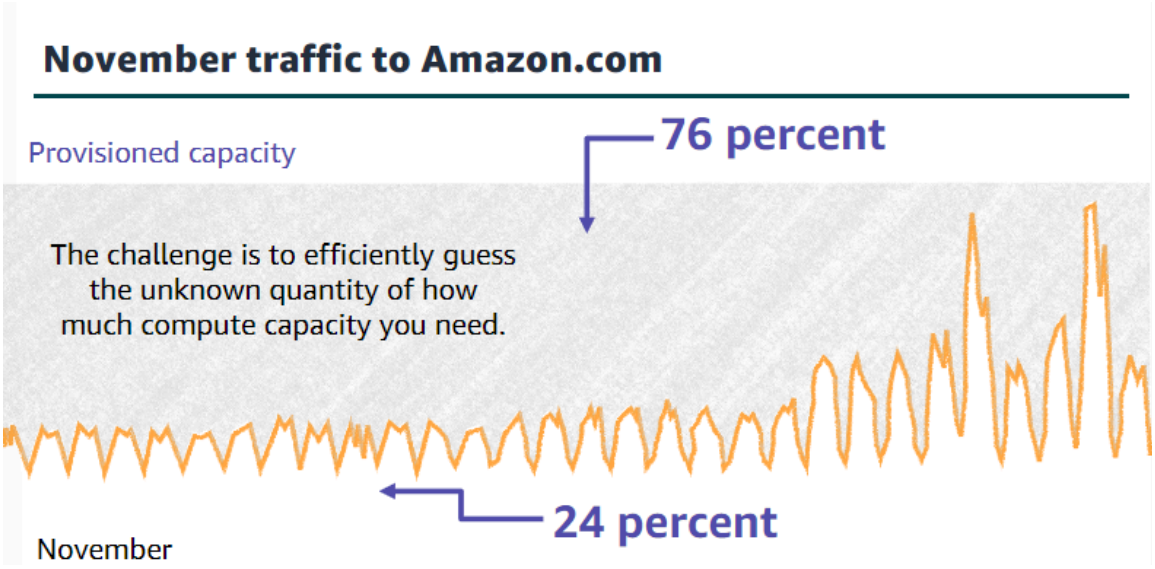


- Helps you maintain application availability
- Enables you to automatically add or remove EC2 instances according to conditions that you define
- Detects impaired EC2 instances and unhealthy applications, and replaces the instances without your intervention
- Provides several scaling options – Manual, 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.

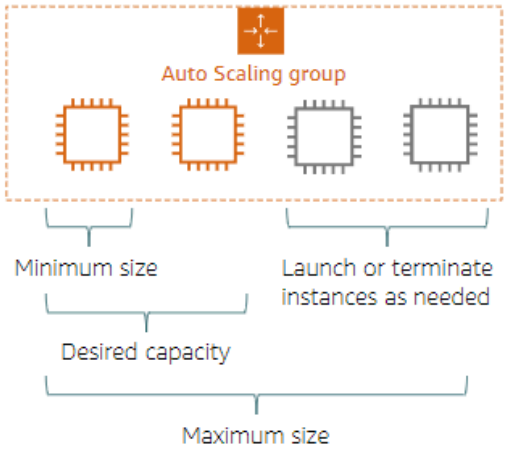


Automatic scaling is useful for predictable workloads—for example, the weekly traffic at the retail company Amazon.com.



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

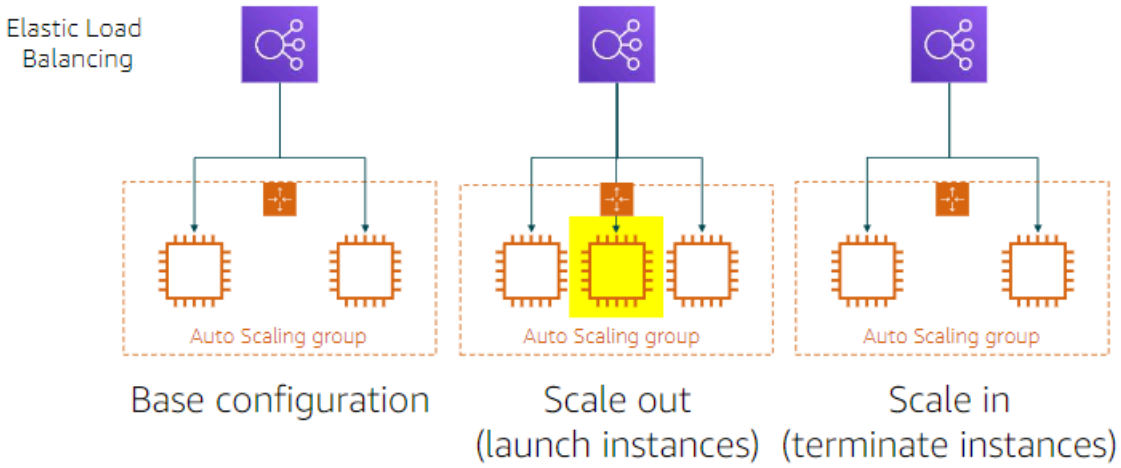


An **Auto Scaling group** is a collection of EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management.

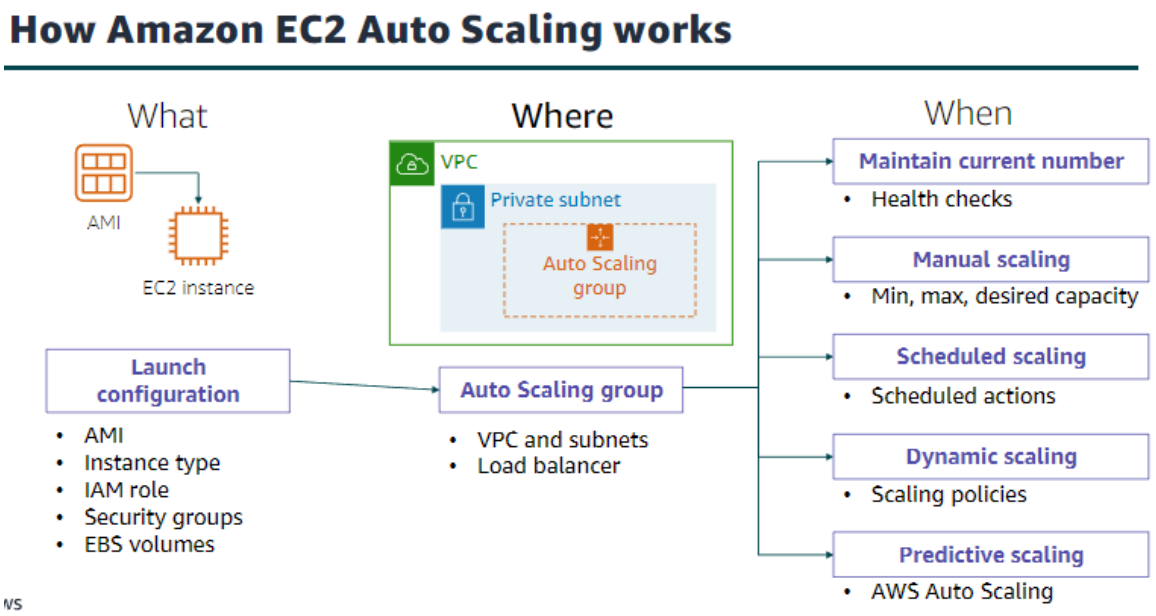
An Auto Scaling group is a collection of Amazon EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management. The size of an Auto Scaling group depends on the number of

instances you set as the desired capacity. You can adjust its size to meet demand, either manually or by using automatic scaling. See more about Auto Scaling Groups at https://docs.aws.amazon.com/autoscaling/ec2/userguide/auto-scaling-groups.html.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 ofyour 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 instanceswithin your minimum and maximum number of instances, based on the criteria that you specify.

Scaling out versus scaling in



With Amazon EC2 Auto Scaling, launching instances is referred to as scaling out, and terminating instances is referred to as scaling in.

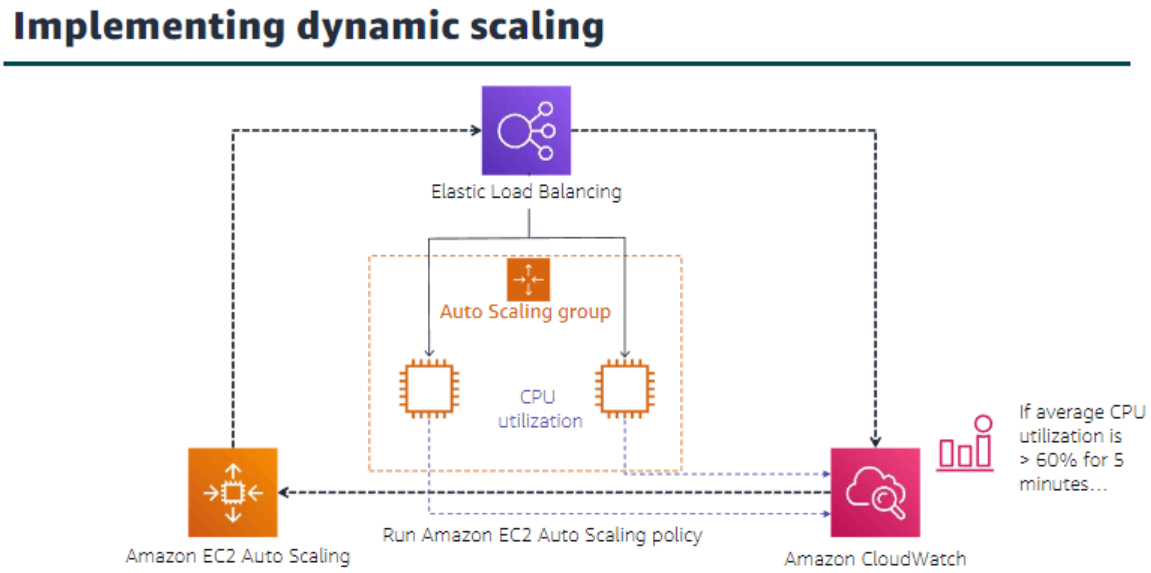


To launch EC2 instances, an Auto Scaling group uses a launch configuration, which is an instance configuration template. You can think of a launch configuration as whatyou are scaling. When you create a launch configuration, you specify information for the instances. The information you specify includes 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. See more information on the launch configuration at https://docs.aws.amazon.com/autoscaling/ec2/userguide/launch-configurations.html.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 whenyou 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

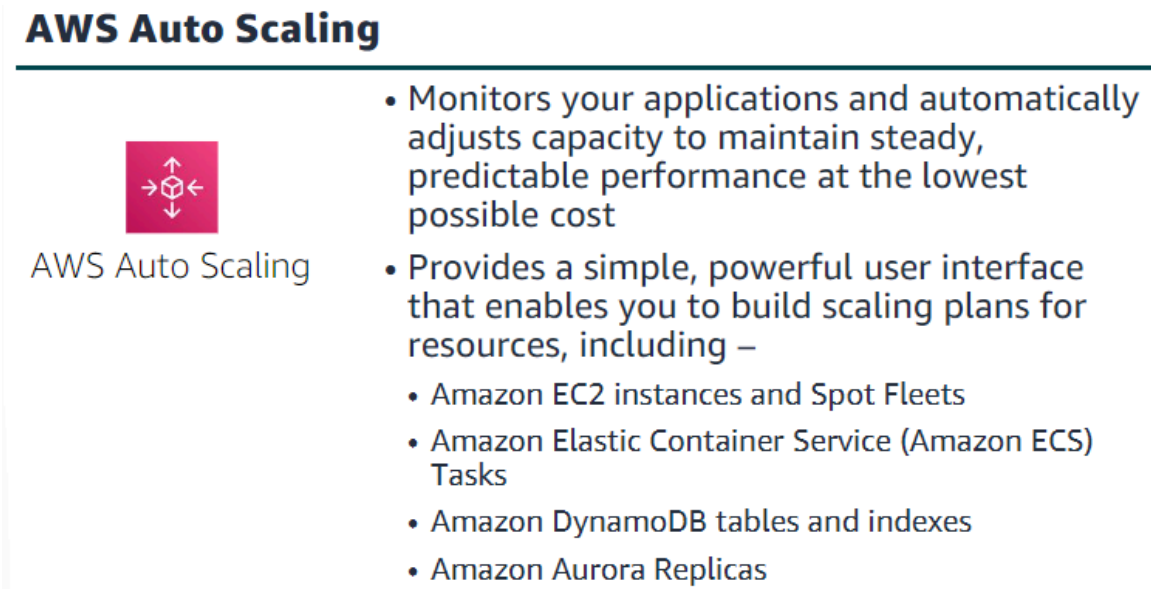
the maximum, minimum, or desired capacity of your Auto Scaling group. See more on manual scaling at https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-manual-scaling.html.•Scheduled scaling–With scheduled scaling, scaling actions are performed automatically as a function of date and time. This is useful for predictable workloads when you know exactly when to increase or decrease the number of instances in your group. For example, saythat every week,the traffic to your web application starts to increase on Wednesday, remains high on Thursday, and starts to decrease on Friday. See more on scheduled scaling at https://docs.aws.amazon.com/autoscaling/ec2/userguide/ec2-auto-scaling-scheduled-scaling.html. 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. •–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. More on dynamic scaling at https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scale-based-on-demand.html. 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. See policy types at https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scale-based-on-demand.html#as-scaling-types.•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 learningmodels 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. See more on predictive scaling at https://aws.amazon.com/blogs/aws/new-predictive-scaling-for-ec2-powered-by-machine-learning/. To learn more about these options, see Scaling the Size of Your Auto Scaling Group in the AWS Documentation at https://docs.aws.amazon.com/autoscaling/ec2/userguide/scale-your-group.html



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 policiesif the average CPU utilization across the fleet goes above 60 percent for 5 minutes.•AmazonEC2Auto 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 EC2Auto Scalingmakes 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 applicationhandles customer demand.



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.

Lab Activity 6: Elastic Load Balancing

This lab walks you through using the Elastic Load Balancing (ELB) and Auto Scaling services to load balance and automatically scale your infrastructure.

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve fault tolerance in your applications by seamlessly providing the required amount of load balancing capacity needed to route application traffic.

Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity out or in automatically according to conditions you define. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 instances. Auto Scaling can

also automatically increase the number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs. Auto Scaling is well suited to applications that have stable demand patterns or that experience hourly, daily, or weekly variability in usage.

Objectives

After completing this lab, you can:

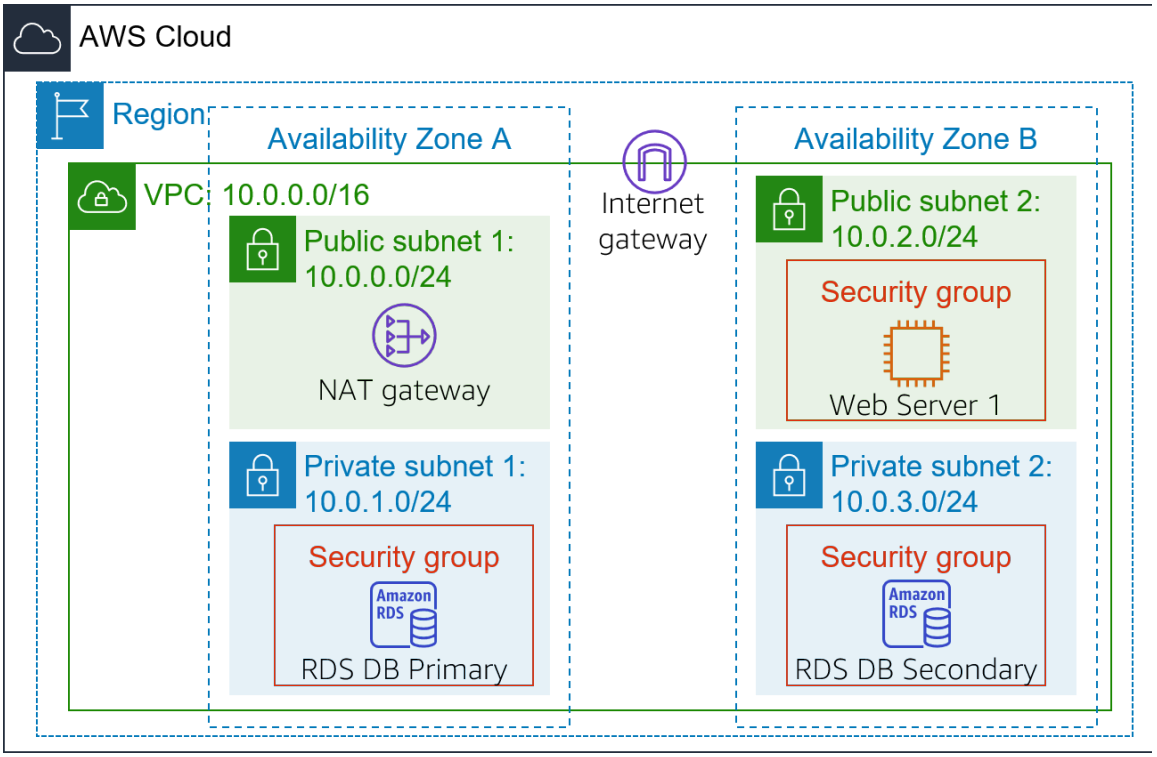
- Create an Amazon Machine Image (AMI) from a running instance.
- Create a load balancer.
- Create a launch template and an Auto Scaling group.
- Automatically scale new instances
- Create Amazon CloudWatch alarms and monitor performance of your infrastructure.

Duration

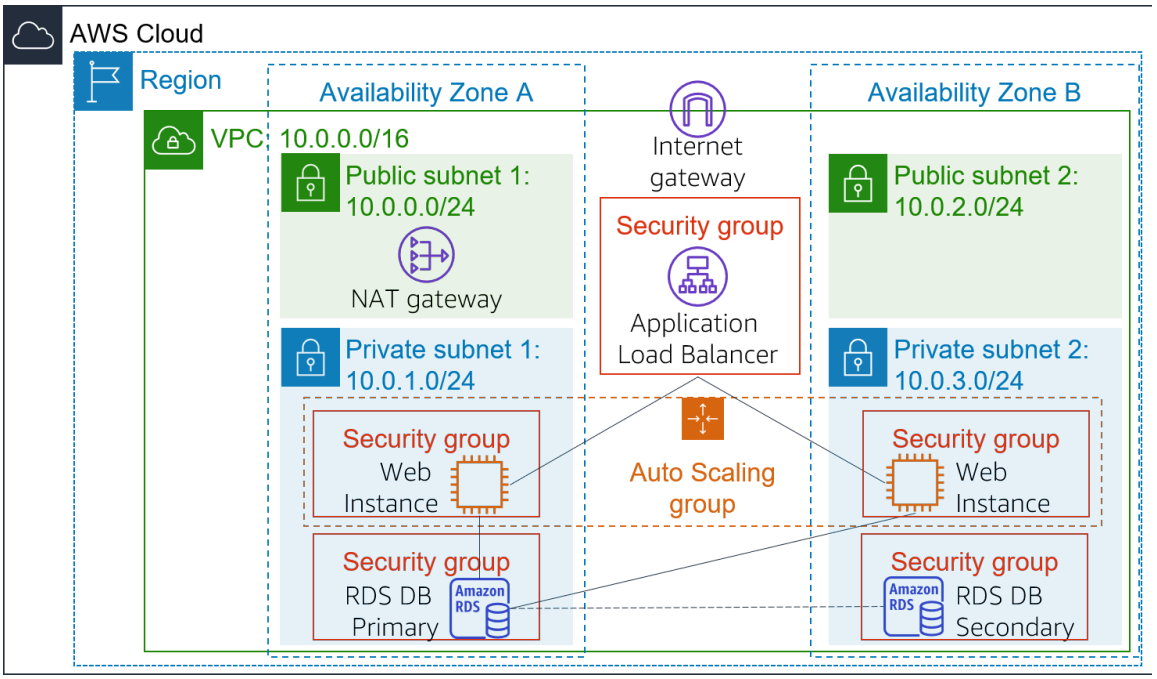
This lab takes approximately **30 minutes**.

Scenario

You start with the following infrastructure:



The final state of the infrastructure is:



Accessing the AWS Management Console

1. At the top of these instructions, choose Start Lab to launch your lab.

   A Start Lab panel opens displaying the lab status.

2. Wait until you see the message "**Lab status: in creation**", then choose the **X** to close the Start Lab panel.

   Note: It may take approximately 10 minutes or longer for the lab status to change to ready.

3. At the top of these instructions, choose AWS

   This will open the AWS Management Console in a new browser tab. The system will automatically log you in.

   Tip: If a new browser tab does not open, there will typically be a banner or icon at the top of your browser indicating that your browser is preventing the site from opening pop-up windows. Choose the banner or icon and choose "Allow pop ups."

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time, to make it easier to follow the lab steps.

Task 1: Create an AMI for Auto Scaling

In this task, you will create an AMI from the existing *Web Server 1*. This will save the contents of the boot disk so that new instances can be launched with identical content.

1. In the **AWS Management Console**, in the search box next to **Services** , search for and select **EC2**.
2. In the left navigation pane, choose **Instances**.

   First, you will confirm that the instance is running.
3. Wait until the **Status Checks** for **Web Server 1** displays *2/2 checks passed*. If necessary, choose refresh to update the status.

   You will now create an AMI based upon this instance.
4. Select **Web Server 1**.
5. In the **Actions** menu, choose **Image and templates** > **Create image**, then configure:
   - **Image name:** `WebServerAMI`
   - **Image description:** `Lab AMI for Web Server`
6. Choose **Create image**

   A confirmation banner displays the **AMI ID** for your new AMI.

   You will use this AMI when launching the Auto Scaling group later in the lab.

Task 2: Create a Load Balancer

In this task, you will create a load balancer that can balance traffic across multiple EC2 instances and Availability Zones.

1. In the left navigation pane, choose **Target Groups**.

   Analysis: *Target Groups* define where to *send* traffic that comes into the Load Balancer. The Application Load Balancer can send traffic to multiple Target Groups based upon the URL of the incoming request, such as having requests from mobile apps going to a different set of servers. Your web application will use only one Target Group.
   - Choose **Create target group**

- Choose a target type: **Instances**
- **Target group name**, enter: `LabGroup`
- Select **Lab VPC** from the **VPC** drop-down menu.

2. Choose **Next**. The **Register targets** screen appears.

   Note: *Targets* are the individual instances that will respond to requests from the Load Balancer.

   You do not have any web application instances yet, so you can skip this step.

3. Review the settings and choose **Create target group**
4. In the left navigation pane, choose **Load Balancers**.
5. At the top of the screen, choose **Create load balancer**.

   Several different types of load balancer are displayed. You will be using an *Application Load Balancer* that operates at the request level (layer 7), routing traffic to targets — EC2 instances, containers, IP addresses and Lambda functions — based on the content of the request. For more information, see: Comparison of Load Balancers
6. Under **Application Load Balancer**, choose **Create**
7. Under **Load balancer name**, enter: `LabELB`
8. Scroll down to the **Network mapping** section, then:

   - For **VPC**, choose **Lab VPC**

     You will now specify which *subnets* the Load Balancer should use. The load balancer will be internet facing, so you will select both Public Subnets.

   - Choose the **first** displayed Availability Zone, then select **Public Subnet 1** from the Subnet drop down menu that displays beneath it.
   - Choose the **second** displayed Availability Zone, then select **Public Subnet 2** from the Subnet drop down menu that displays beneath it.

     You should now have two subnets selected: **Public Subnet 1** and **Public Subnet 2**.

9. In the **Security groups** section:

- Choose the Security groups drop down menu and select **Web Security Group**
- Below the drop down menu, choose the **X** next to the default security group to remove it.

  The **Web Security Group** security group should now be the only one that appears.

10. For the Listener HTTP:80 row, set the Default action to forward to **LabGroup**.
11. Scroll to the bottom and choose **Create load balancer**

    The load balancer is successfully created.

    - Choose **View load balancer**

      The load balancer will show a state of *provisioning*. There is no need to wait until it is ready. Please continue with the next task.

Task 3: Create a Launch Template and an Auto Scaling Group

In this task, you will create a *launch template* for your Auto Scaling group. A launch template is a template that an Auto Scaling group uses to launch EC2 instances. When you create a launch template, you specify information for the instances such as the AMI, the instance type, a key pair, and security group.

1. In the left navigation pane, choose **Launch Templates**.
2. Choose **Create launch template**
3. Configure the launch template settings and create it:
   - **Launch template name:** `LabConfig`
   - Under **Auto Scaling guidance**, select *Provide guidance to help me set up a template that I can use with EC2 Auto Scaling*
   - In the Application and OS Images (Amazon Machine Image) area, choose *My AMIs*.
   - **Amazon Machine Image (AMI)**: choose *Web Server AMI*
   - **Instance type:** choose *t2.micro*

- **Key pair name**: choose *vockey*
- **Firewall (security groups)**: choose *Select existing security group*
- **Security groups**: choose *Web Security Group*
- Scroll down to the **Advanced details** area and expand it.
- Scroll down to the **Detailed CloudWatch monitoring** setting. Select *Enable*

  Note: This will allow Auto Scaling to react quickly to changing utilization.
- Choose **Create launch template**

  Next, you will create an Auto Scaling group that uses this launch template.

4. In the Success dialog, choose the **LabConfig** launch template.
5. From the **Actions** menu, choose *Create Auto Scaling group*
6. Configure the details in Step 1 (Choose launch template or configuration):
   - **Auto Scaling group name:** `Lab Auto Scaling Group`
   - **Launch template**: confirm that the *LabConfig* template you just created is selected.
   - Choose **Next**
7. Configure the details in Step 2 (Choose instance launch options):
   - **VPC**: choose *Lab VPC*
   - **Availability Zones and subnets**: Choose *Private Subnet 1* and then choose *Private Subnet 2*.
   - Choose **Next**
8. Configure the details in Step 3 (Configure advanced options):
   - Choose **Attach to an existing load balancer**
     - **Existing load balancer target groups**: select *LabGroup*.
   - In the **Additional settings** pane:
     - Select **Enable group metrics collection within CloudWatch**

This will capture metrics at 1-minute intervals, which allows Auto Scaling to react quickly to changing usage patterns.

- Choose **Next**

9. Configure the details in Step 4 (Configure group size and scaling policies - optional):
   - Under **Group size**, configure:
     - **Desired capacity:** 2
     - **Minimum capacity:** 2
     - **Maximum capacity:** 6

       This will allow Auto Scaling to automatically add/remove instances, always keeping between 2 and 6 instances running.
   - Under **Scaling policies**, choose *Target tracking scaling policy* and configure:
     - **Scaling policy name:** `LabScalingPolicy`
     - **Metric type:** *Average CPU Utilization*
     - **Target value:** `60`

       This tells Auto Scaling to maintain an *average* CPU utilization *across all instances* at 60%. Auto Scaling will automatically add or remove capacity as required to keep the metric at, or close to, the specified target value. It adjusts to fluctuations in the metric due to a fluctuating load pattern.
   - Choose **Next**
10. Configure the details in Step 5 (Add notifications - optional):

    Auto Scaling can send a notification when a scaling event takes place. You will use the default settings.

    - Choose **Next**
11. Configure the details in Step 6 (Add tags - optional):

    Tags applied to the Auto Scaling group will be automatically propagated to the instances that are launched.

    - Choose **Add tag** and Configure the following:
      - **Key:** `Name`

    - **Value:** Lab Instance
  - Choose **Next**

12. Configure the details in Step 6 (Review):
    - Review the details of your Auto Scaling group
    - Choose **Create Auto Scaling group**

      Your Auto Scaling group will initially show an instance count of zero, but new instances will be launched to reach the **Desired** count of 2 instances.

Task 4: Verify that Load Balancing is Working

In this task, you will verify that Load Balancing is working correctly.

1. In the left navigation pane, choose **Instances**.

   You should see two new instances named **Lab Instance**. These were launched by Auto Scaling.

   If the instances or names are not displayed, wait 30 seconds and choose refresh in the top-right.

   Next, you will confirm that the new instances have passed their Health Check.

2. In the left navigation pane, choose **Target Groups**.
3. Select *LabGroup*
4. Choose the **Targets** tab.

   Two target instances named Lab Instance should be listed in the target group.

5. Wait until the **Status** of both instances transitions to *healthy*.

   Choose Refresh in the upper-right to check for updates if necessary.

   Healthy indicates that an instance has passed the Load Balancer's health check. This means that the Load Balancer will send traffic to the instance.

   You can now access the Auto Scaling group via the Load Balancer.

6. In the left navigation pane, choose **Load Balancers**.

7. Select the *LabELB* load balancer.
8. In the Details pane, copy the **DNS name** of the load balancer, making sure to omit "(A Record)".

   It should look similar to: *LabELB-1998580470.us-west-2.elb.amazonaws.com*
9. Open a new web browser tab, paste the DNS Name you just copied, and press Enter.

   The application should appear in your browser. This indicates that the Load Balancer received the request, sent it to one of the EC2 instances, then passed back the result.

Task 5: Test Auto Scaling

You created an Auto Scaling group with a minimum of two instances and a maximum of six instances. Currently two instances are running because the minimum size is two and the group is currently not under any load. You will now increase the load to cause Auto Scaling to add additional instances.

1. Return to the AWS Management Console, but do not close the application tab — you will return to it soon.
2. in the search box next to **Services** , search for and select **CloudWatch**.
3. In the left navigation pane, choose **All alarms**.

   Two alarms will be displayed. These were created automatically by the Auto Scaling group. They will automatically keep the average CPU load close to 60% while also staying within the limitation of having two to six instances.

   Note: Please follow these steps only if you do not see the alarms in 60 seconds.

   - On the **Services** menu, choose **EC2**.
   - In the left navigation pane, choose **Auto Scaling Groups**.
   - Select **Lab Auto Scaling Group**.
   - In the bottom half of the page, choose the **Automatic Scaling** tab.
   - Select **LabScalingPolicy**.
   - Choose **Actions** and **Edit**.

- Change the **Target Value** to `50`.
- Choose **Update**
- On the **Services** menu, choose **CloudWatch**.
- In the left navigation pane, choose **All alarms** and verify you see two alarms.

4. Choose the **OK** alarm, which has *AlarmHigh* in its name.

   If no alarm is showing **OK**, wait a minute then choose refresh in the top-right until the alarm status changes.

   The **OK** indicates that the alarm has *not* been triggered. It is the alarm for **CPU Utilization > 60**, which will add instances when average CPU is high. The chart should show very low levels of CPU at the moment.

   You will now tell the application to perform calculations that should raise the CPU level.

5. Return to the browser tab with the web application.
6. Choose **Load Test** beside the AWS logo.

   This will cause the application to generate high loads. The browser page will automatically refresh so that all instances in the Auto Scaling group will generate load. Do not close this tab.

7. Return to browser tab with the **CloudWatch** console.

   In less than 5 minutes, the **AlarmLow** alarm should change to **OK** and the **AlarmHigh** alarm status should change to *In alarm*.

   You can choose Refresh in the top-right every 60 seconds to update the display.

   You should see the **AlarmHigh** chart indicating an increasing CPU percentage. Once it crosses the 60% line for more than 3 minutes, it will trigger Auto Scaling to add additional instances.

8. Wait until the **AlarmHigh** alarm enters the *In alarm* state.

   You can now view the additional instance(s) that were launched.

9. In the search box next to **Services** , search for and select **EC2**.
10. In the left navigation pane, choose **Instances**.

More than two instances labeled **Lab Instance** should now be running. The new instance(s) were created by Auto Scaling in response to the CloudWatch alarm.

Task 6: Terminate Web Server 1

In this task, you will terminate *Web Server 1*. This instance was used to create the AMI used by your Auto Scaling group, but it is no longer needed.

1. Select **Web Server 1** (and ensure it is the only instance selected).
2. In the **Instance state** menu, choose **Instance State** > **Terminate Instance**.
3. Choose **Terminate**