



---

— iamneo —



Introduction to Cloud Computing

Cloud Computing Service Models

Cloud Deployment Models

## WHAT IS CLOUD COMPUTING?

1. Cloud Computing is the on-demand delivery of IT resources (like servers, storage, databases, networking, software) over the internet on a pay-as-you-go basis.

# WHAT IS CLOUD COMPUTING?

## FeatureDescription

**On-Demand Access-** Resources available anytime without manual provisioningScalability

Easily scale up or down based on usage

**Pay-as-you-go-Pay** only for what you use—no upfront investment

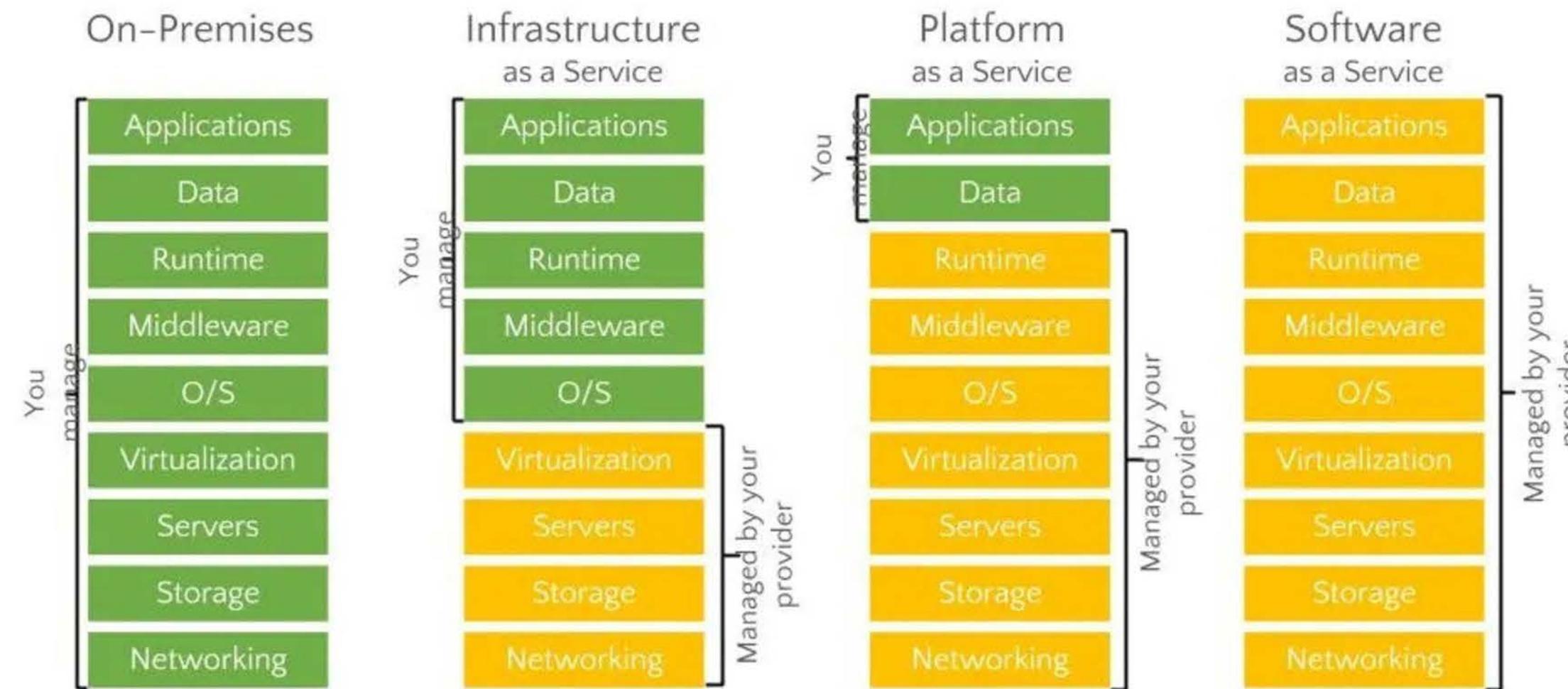
**Global Access-**Access from anywhere using internet

**Automatic Updates-**Software and hardware updates are managed by the provider

# CLOUD COMPUTING SERVICE MODELS

1. IAAS - Infrastructure As A Service
2. PAAS -Platform As A Service
3. SAAS - Software As A Service

# CLOUD COMPUTING SERVICE MODELS



## SOFTWARE AS A SERVICE

- Ready-to-use software delivered over the internet.
- Users don't manage infrastructure or platforms—just use the software.
- Accessible via web browsers with minimal setup.
- Example: Google Workspace (Gmail, Google Drive), Microsoft 365.

## PLATFORM AS A SERVICE

- Provides tools and environment to build, test, and deploy applications.
- Developers focus only on coding—no need to manage servers or OS.
- Scales automatically and supports continuous integration.
- Example: Heroku, Google App Engine.

## INFRASTRUCTURE AS A SERVICE

- Offers virtualized hardware resources like servers, storage, and networking.
- Users manage OS, applications, and runtime—provider handles hardware.
- Ideal for building custom platforms or hosting enterprise applications.
- Example: Amazon EC2, Microsoft Azure Virtual Machines.

# CLOUD COMPUTING MODELS



**VS**



Publically Shared  
Virtualised Resources

Supports multiple  
customers

Supports connectivity  
over the internet

Suited for less  
confidential information

Privately Shared  
Virtualised Resources

Cluster of dedicated  
customers

Connectivity over  
internet, fibre and private network

Suited for secured  
confidential information  
& core systems

## PUBLIC CLOUD

- **Definition:** Cloud services offered over the internet and shared among multiple users.
- **Advantages:** Cost-effective, easily scalable, no hardware maintenance.
- **Disadvantages:** Less control and more security concerns.
- **Examples:** Amazon Web Services (AWS), Microsoft Azure, Google Cloud

## PRIVATE CLOUD

- **Definition:** Cloud infrastructure dedicated to a single organization.
- **Advantages:** Higher security, control, and customization.
- **Disadvantages:** Expensive to set up and maintain.
- **Examples:** VMware vSphere, OpenStack private deployments.

## HYBRID CLOUD

- **Definition:** Combination of public and private cloud environments.
- **Advantages:** Flexibility to move workloads, better optimization.
- **Disadvantages:** Complex integration and security challenges.
- **Examples:** Microsoft Azure Stack, AWS Outposts.

## COMMUNITY CLOUD

- **Definition:** Shared cloud for a specific group of organizations with similar needs.
- **Advantages:** Cost and resource sharing, tailored to industry needs.
- **Disadvantages:** Limited availability, complex governance.
- **Examples:** Government or healthcare shared cloud services.

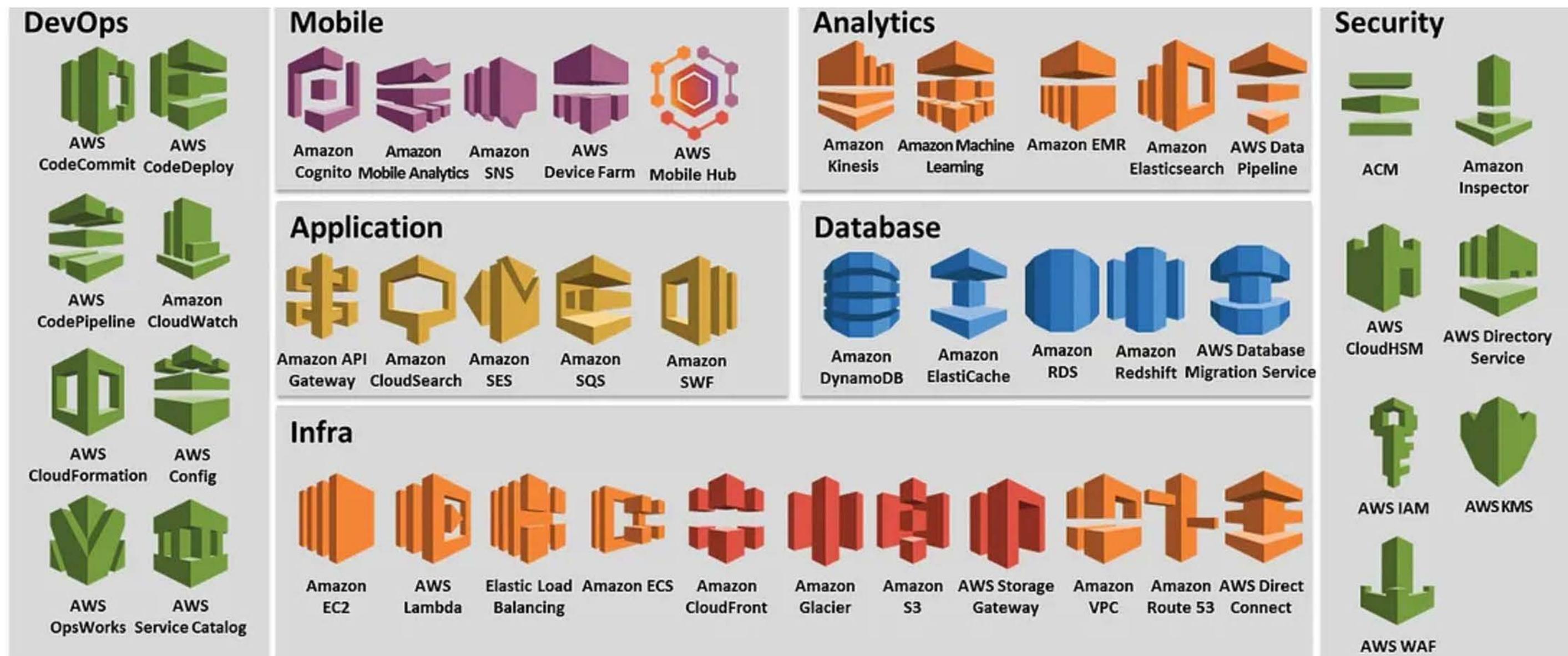
## AWS OVERVIEW AND HISTORY

- Amazon Web Services (AWS) is a cloud computing platform launched by Amazon in 2006.
- It started with services like S3 (storage) and EC2 (compute).
- AWS is the market leader in cloud computing, offering scalable, reliable, and low-cost infrastructure solutions.
- It supports businesses of all sizes—from startups to enterprises to government.

## AWS GLOBAL INFRASTRUCTURE

- **Region:** A geographical area (like us-east-1, ap-south-1) with multiple, isolated Availability Zones.
- **Availability Zone (AZ):** One or more physically separate data centers in a region with independent power, networking, and cooling.
- This model helps in disaster recovery, fault isolation, and better application performance.
- **Example:** Mumbai Region (ap-south-1) has 3 Availability Zones.

# AWS SERVICES



# AWS SERVICES

AWS offers 200+ fully featured services under several categories:

A	B
Category	Examples
Compute	EC2, Lambda, Elastic Beanstalk
Storage	S3, EBS, Glacier
Database	RDS, DynamoDB, Aurora
Networking	VPC, Route 53, CloudFront
Security	IAM, KMS, Shield, Cognito
Analytics	Athena, Redshift, EMR
AI/ML	SageMaker, Rekognition, Comprehend
DevOps Tools	CodeDeploy, CodePipeline, CloudFormation
Migration	Snowball, DMS, Migration Hub

iamneo



# Introduction to AWS

---



# Introduction to AWS

Welcome to the world of AWS, the most secure, flexible and scalable cloud computing platform. With AWS, you can build and deploy any application or software with ease.



# What is AWS?

Amazon Web Services (AWS) is a cloud computing platform that offers a wide range of services including computing power, storage, security, analytics, and more. It is a comprehensive platform that enables developers to build and deploy applications quickly and efficiently.

## Security



## Flexible



## Scalable

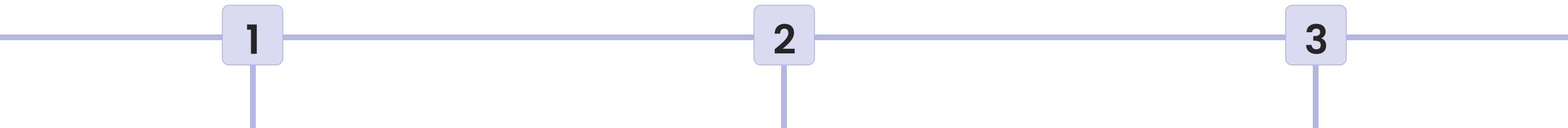


# Usage of AWS Cloud



# AWS Infrastructure

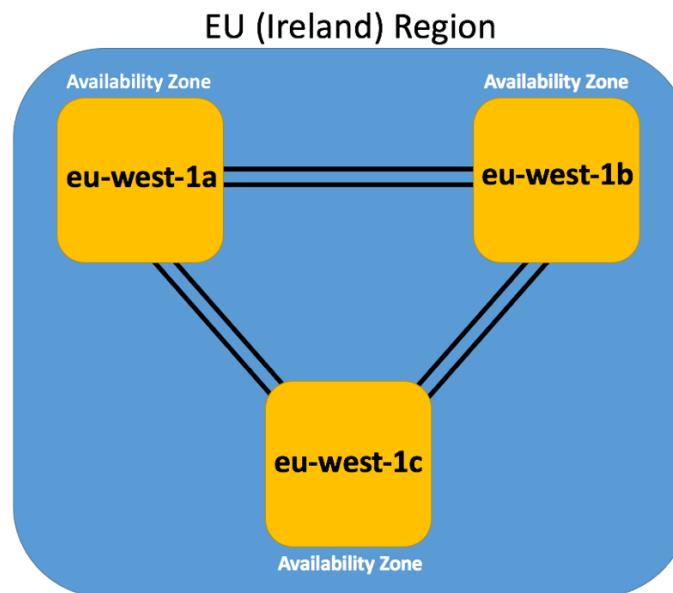
AWS has a global infrastructure comprising data centers, availability zones, and regions designed to provide high availability and durability for your applications and content.



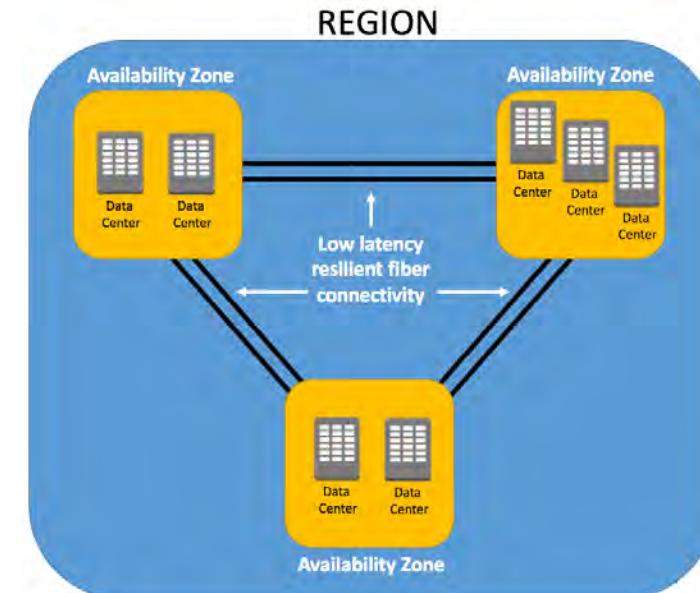
**Data Centers**



**Availability Zones**



**Regions**



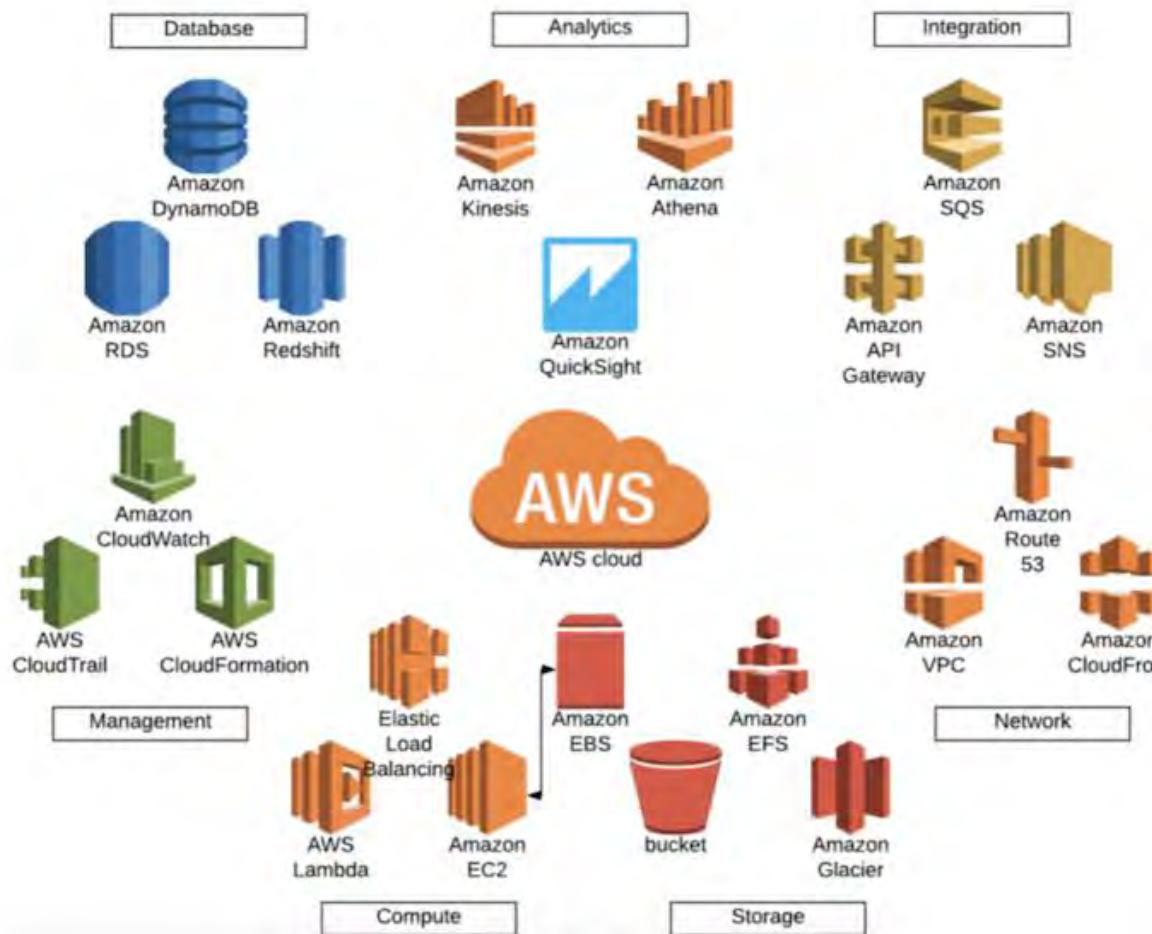
# Advantages of AWS

There are many advantages to using AWS, including cost savings, ease of use, innovation, and more.



# AWS Services Overview

With AWS, you can architect your applications using the most suitable architecture that meets your needs. Whether it's a simple web application or a complex distributed application with multiple tiers, AWS provides the tools and services you need to build and deploy it with ease.



# AWS Service Overview

Amazon Web Services (AWS) provides a comprehensive cloud computing platform that offers a wide range of services to meet the needs of customers. Some of the major services include:

## 1 Compute

Services like EC2, Lambda, and Elastic Beanstalk provide scalable computing resources for running applications and workloads.

## 2 Storage

Services like S3, EBS, and Glacier provide scalable and durable storage for data and applications.

## 3 Database

Services like RDS, DynamoDB, and Aurora provide managed database solutions for different types of data and workloads.

## 4 Analytics

Services like Redshift, Athena, and QuickSight provide tools for data processing, warehousing, and analysis.

# AWS Service Overview

---

## 5 Machine Learning

Services like SageMaker, DeepLens, and Rekognition provide pre-built models and tools for machine learning and AI applications.

## 6 Networking and Content Delivery

Services like VPC, CloudFront, and Route 53 provide tools for building and managing network infrastructure and content delivery networks.

## 7 Security, Identity, and Compliance

Services like IAM, Inspector, and GuardDuty provide tools for managing security, identity, and compliance in the cloud.

# AWS Pricing Model

One of the many benefits of AWS is the pay-as-you-go pricing model, which is designed to provide cost savings and flexibility for customers.

<b>Services</b>	<b>Pricing Model</b>
Amazon EC2	Pay per instance per hour
Amazon S3	Pay per GB stored per month
Amazon RDS	Pay per hour per instance
Amazon Lambda	Pay per 100ms of execution time

# Getting Started with AWS

---

Getting started with AWS is easy, with a range of resources and tools available to help you along the way. Here are some tips to help you get started:

## 1. Choose your platform

Choose the AWS platform that best meets your needs, whether it's EC2, S3, RDS, Lambda or something else.

## 2. Create your account

Sign up for an AWS account and provide your billing and payment information.

## 3. Launch your instance

Launch your instance and configure it as per your requirements.

iamneo



# AWS Cloud Benefits

---

# Agenda

---

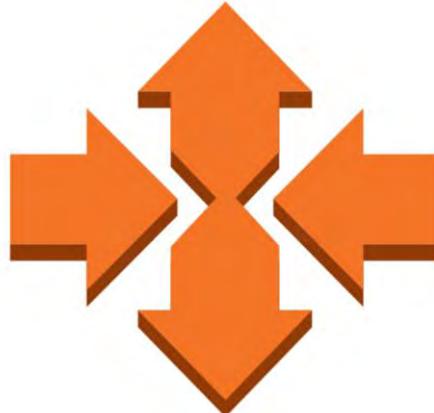
-  **Scalability of AWS Cloud**
-  **Cost-effectiveness of AWS Cloud**
-  **Reliability and Availability of AWS Cloud**
-  **Security of AWS Cloud**
-  **Flexibility of AWS Cloud**
-  **Benefits to Business and Operations**
-  **Cost Optimization**

# Scalability of AWS Cloud



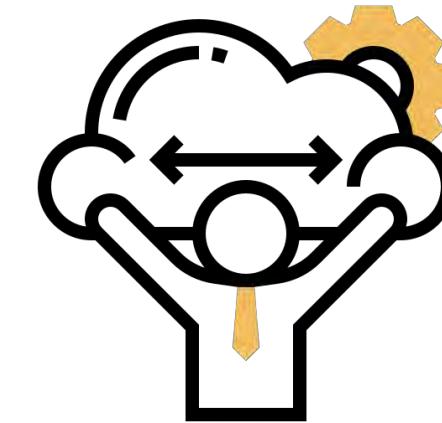
1

**Auto Scaling**



2

**Elasticity**



3

**Pre-Defined Services**

Choose from pre-defined services that automatically scale based on the demand they receive.

4

**Developer Tools**

Make scalability easier and more intuitive with AWS developer tools.

# Cost-effectiveness of AWS Cloud



designed by siffranek.com

## No Upfront Payments

You pay for what you use, with no upfront costs or long-term commitments.



## Simple Cost Management

Use AWS Cost Explorer to manage and optimize your cloud costs easily.

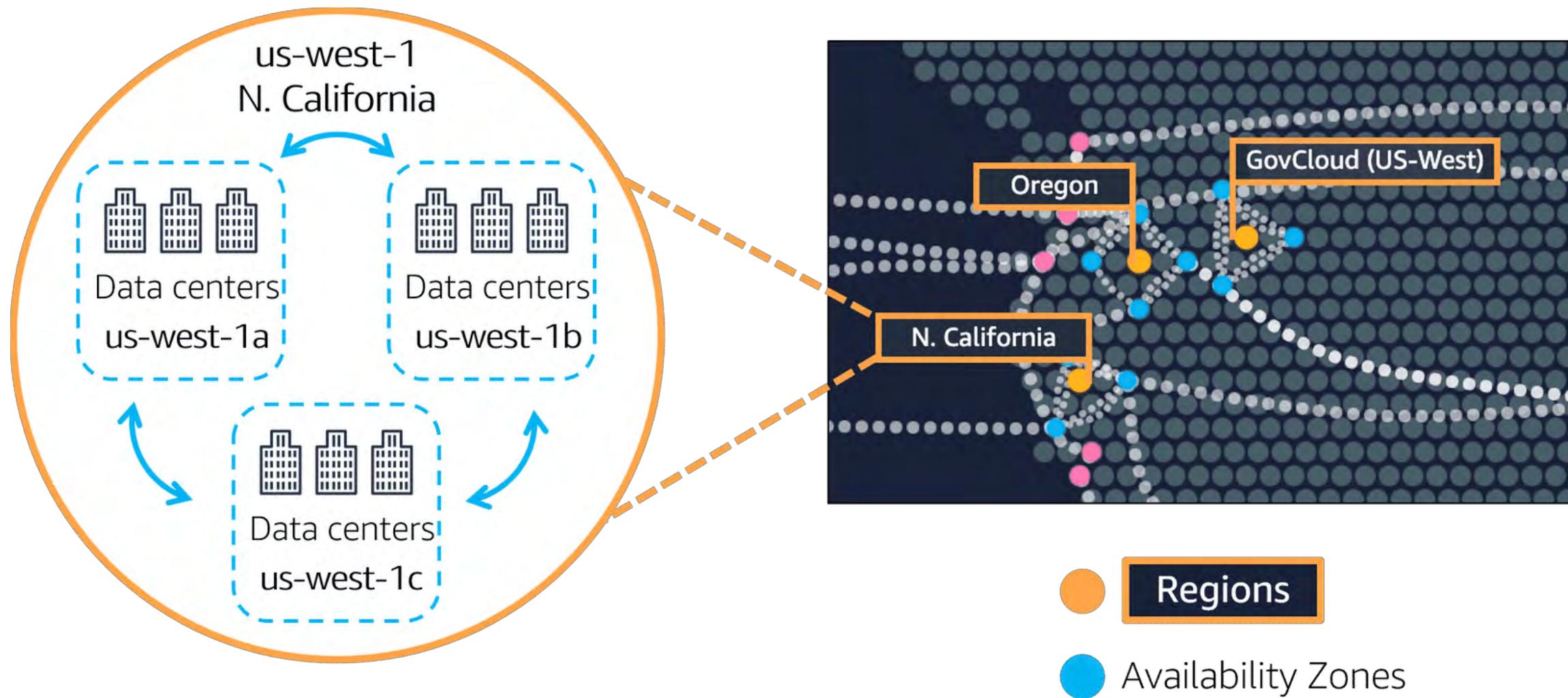


## Maximize Efficiency

Use resources only when needed to maximize utilization and efficiency.

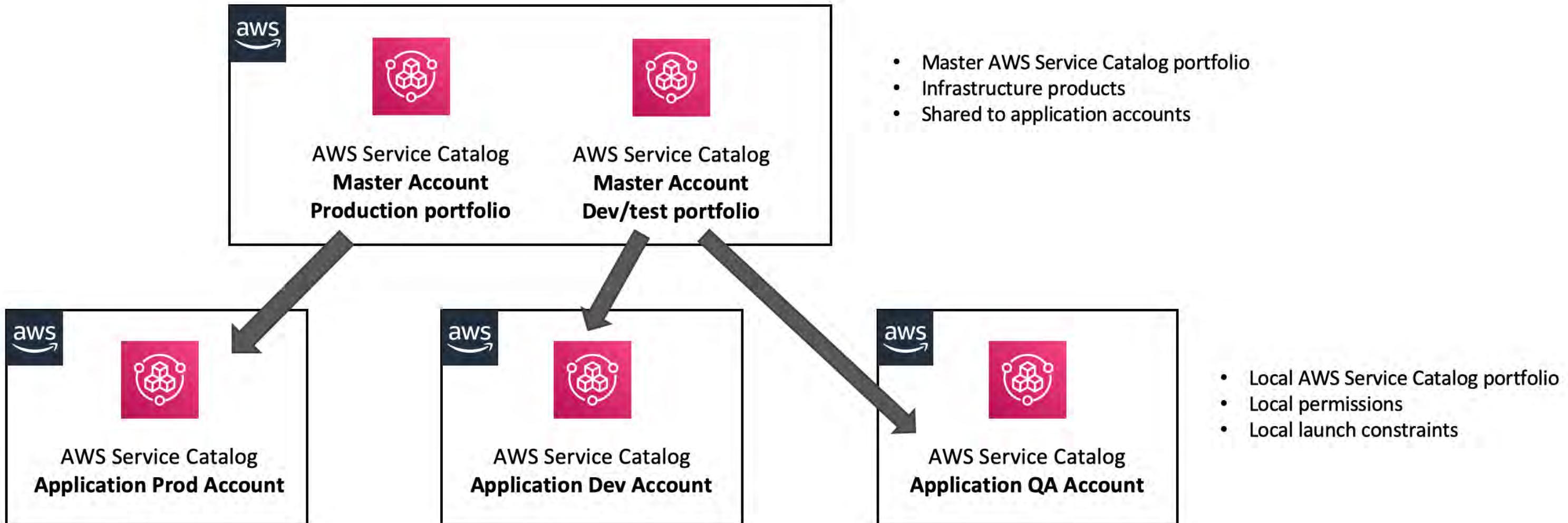
# Reliability and Availability of AWS Cloud

## Multiple Availability Zones



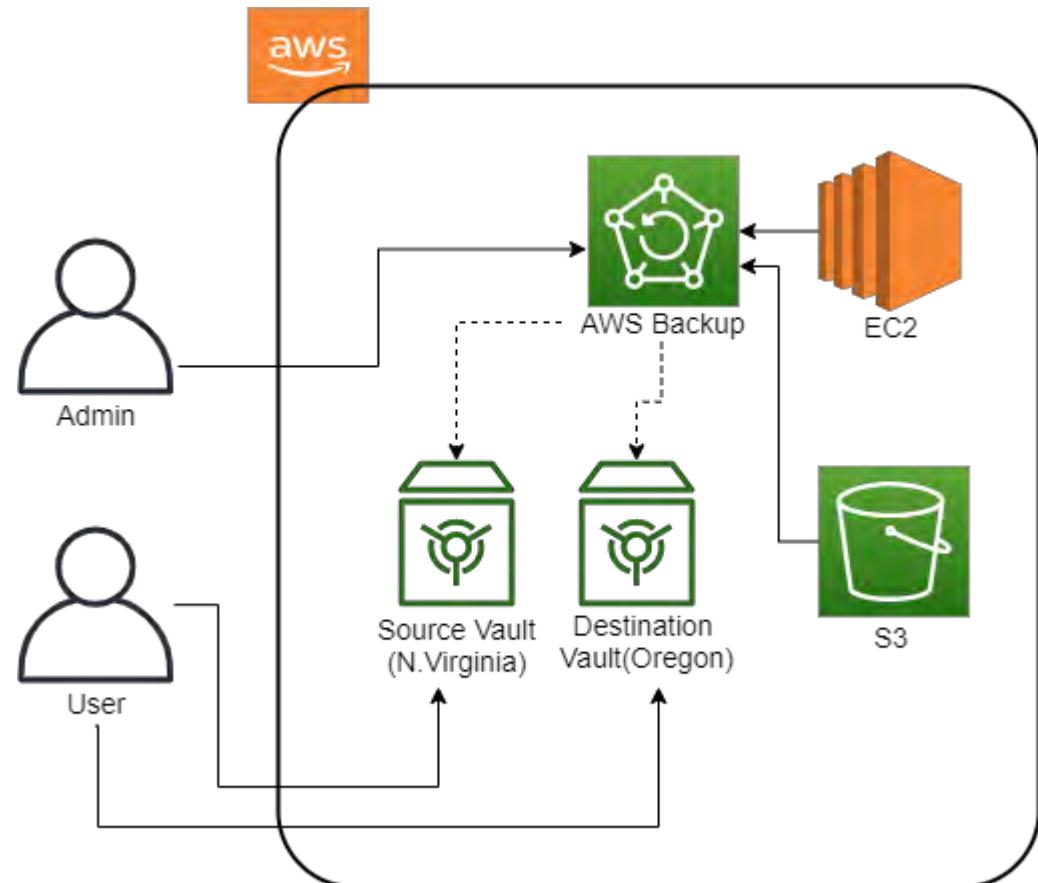
# Reliability and Availability of AWS Cloud

## Distributed Infrastructure



# Reliability and Availability of AWS Cloud

## Replication and Backups



## Service Level Agreement



# Security of AWS Cloud

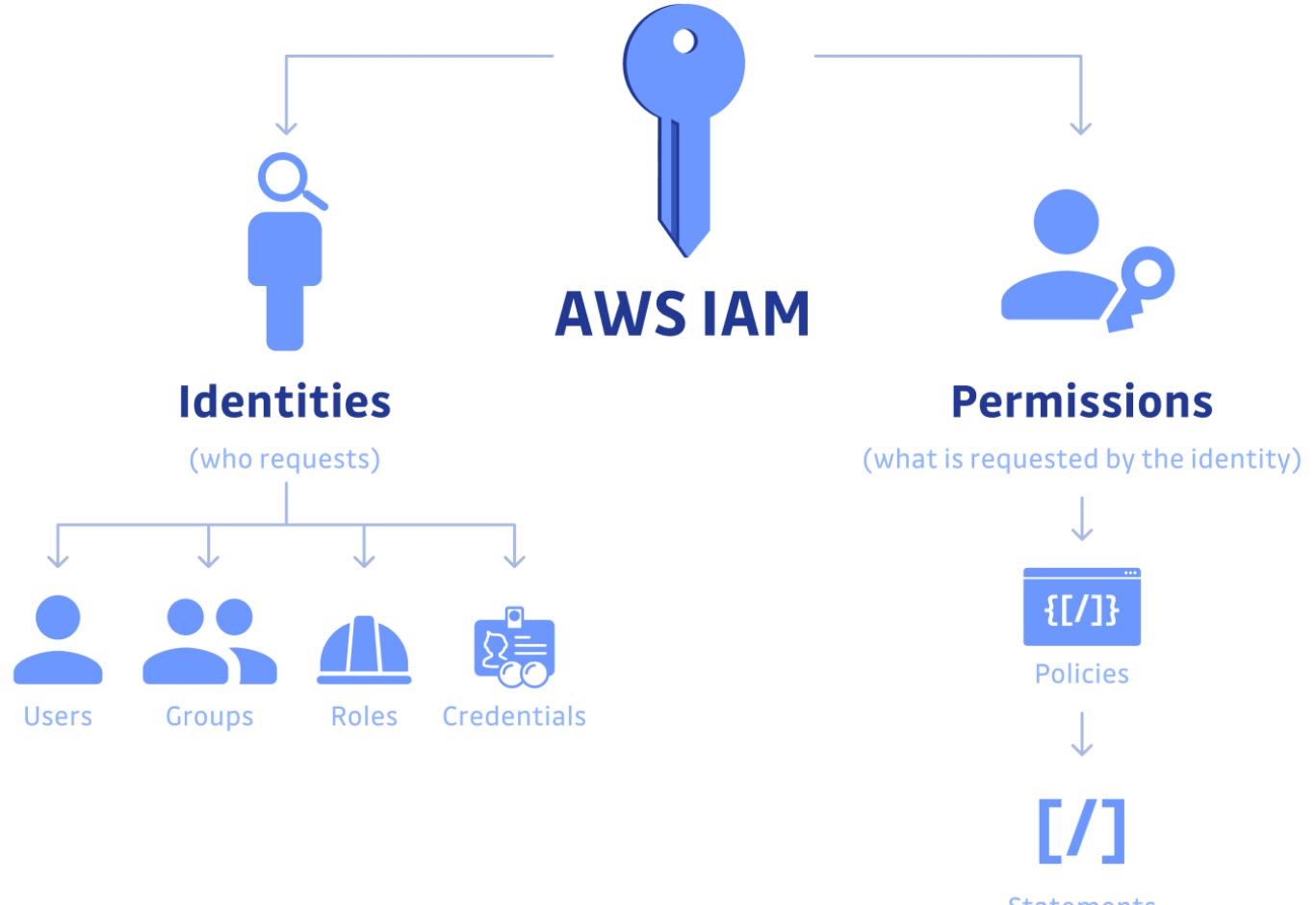
## Managed Security Services



## Data Encryption

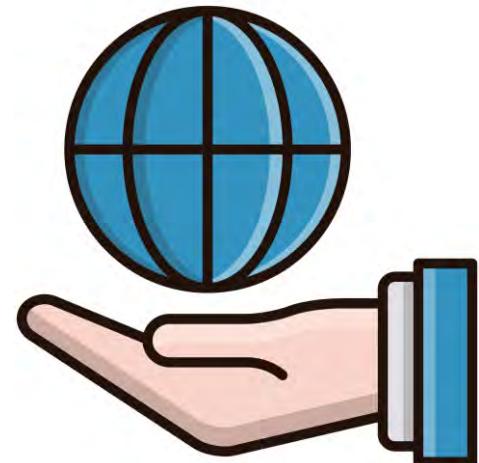


## Identity and Access Management

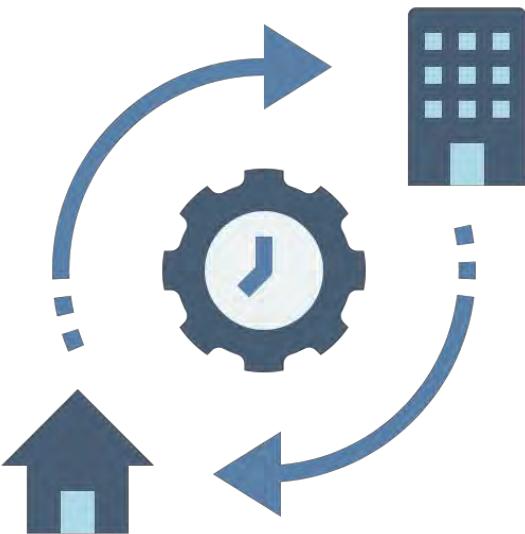


# Flexibility of AWS Cloud

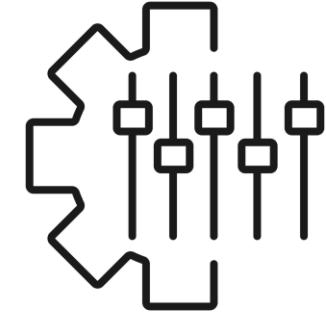
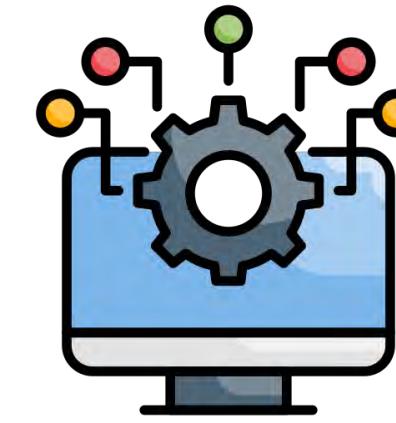
## Wide Range of Services



## Hybrid Environments



## Integration and Customization



## Open Standards and Platforms

Build and deploy on open standards and platforms, giving you more freedom in your technology choices.

# Flexibility of AWS Cloud

---

## Wide Range of Services

Choose from a wide range of services and solutions to meet your specific needs and requirements.

## Hybrid Environments

Combine AWS with on-premises solutions for a hybrid cloud and bridge the gap to the cloud at your own pace.

## Integration and Customization

Customize and integrate your solutions with a range of APIs and tools, including AWS Lambda, to suit your unique needs.

## Open Standards and Platforms

Build and deploy on open standards and platforms, giving you more freedom in your technology choices.

# Benefits to Business and Operations

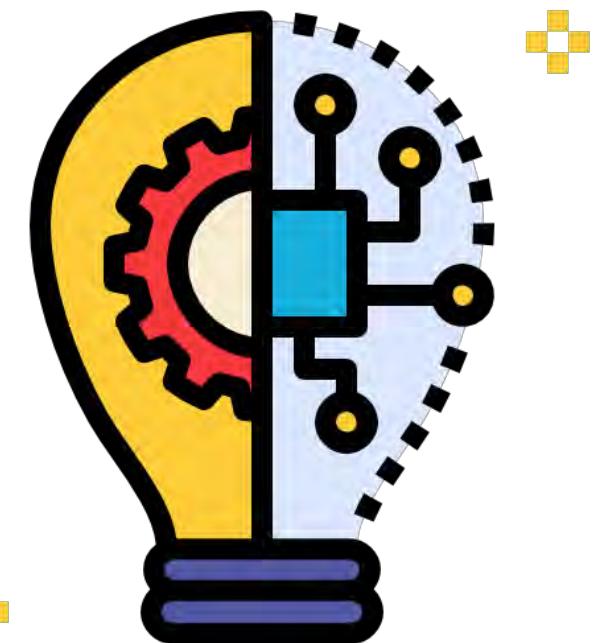
## Increased Agility

Build and deploy software faster, with higher quality, and with better alignment to your business needs.



## Improved Innovation

Use AWS tools and solutions to rapidly prototype, test, and experiment with new solutions and ideas.



## Better Resource Utilization

Scale resources up or down to meet demand, allowing you to better utilize resources and reduce waste.



# Cost Optimization



## Cut Costs

Use AWS Cost Explorer to identify cost-saving opportunities and eliminate unnecessary spending.



## Optimize Resources

Use AWS Trusted Advisor to optimize resource utilization and reduce costs by using the right resources for the right job.



## Reserved Instances

Save money by reserving instances when you know you'll need them in advance.

# Global Reach

## Wide Availability

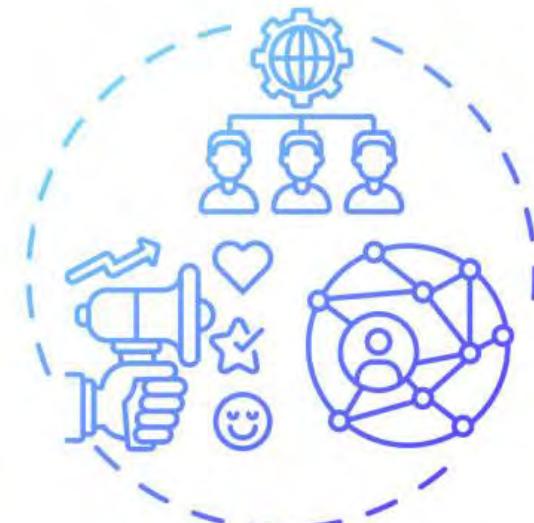
Deploy your applications to anywhere in the world with AWS's widespread infrastructure availability.

## High Performance

Have consistently high performance for applications delivered across the globe.

## Regulatory Requirements

Stay compliant with the various regulatory requirements across different countries.



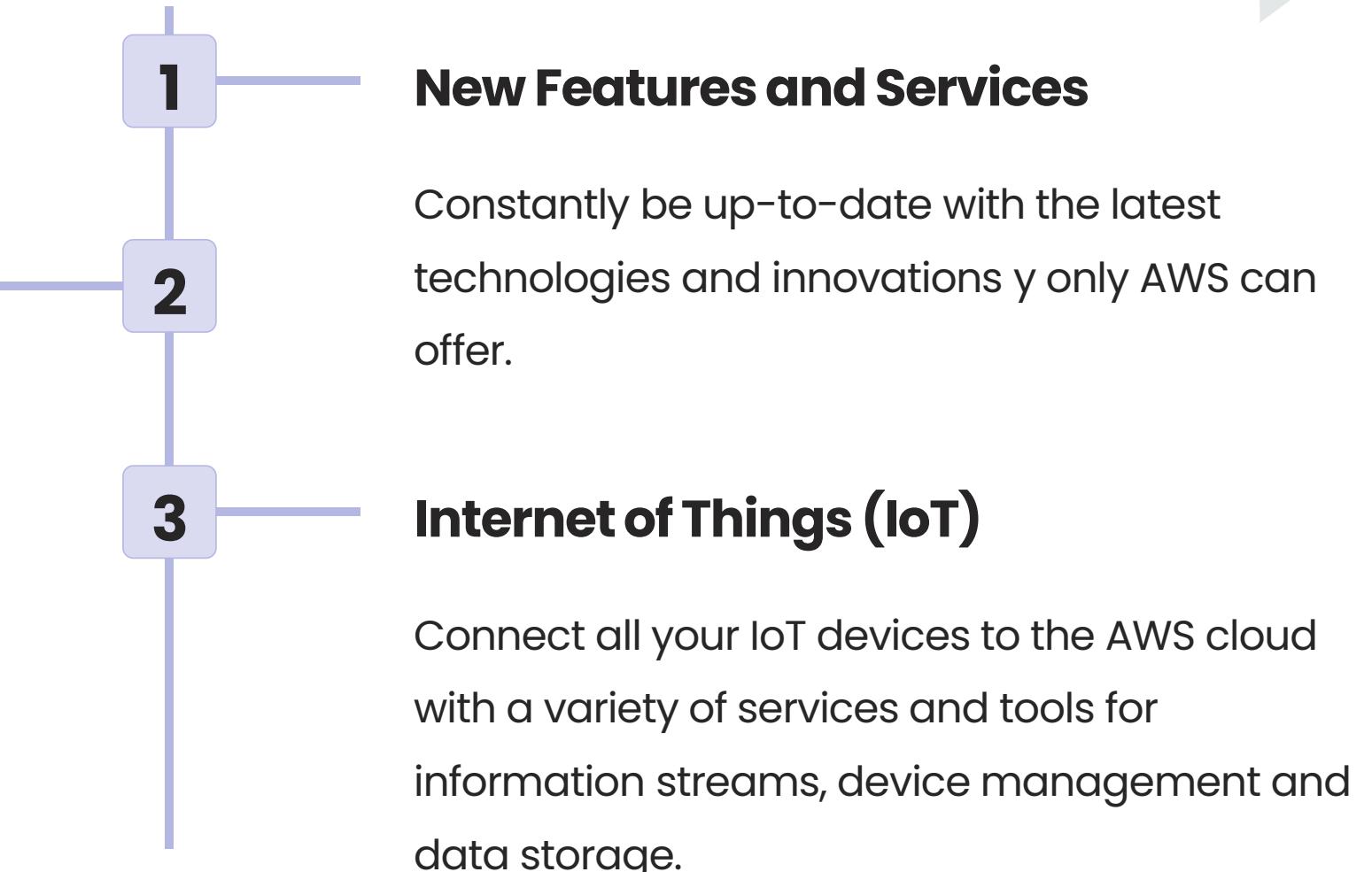
BETTER REACH

# Innovation

---

## Machine Learning Services

Get started on machine learning right away with pre-built models or train your own custom models with Amazon SageMaker.



# Case Studies and Success Stories

---

## Netflix

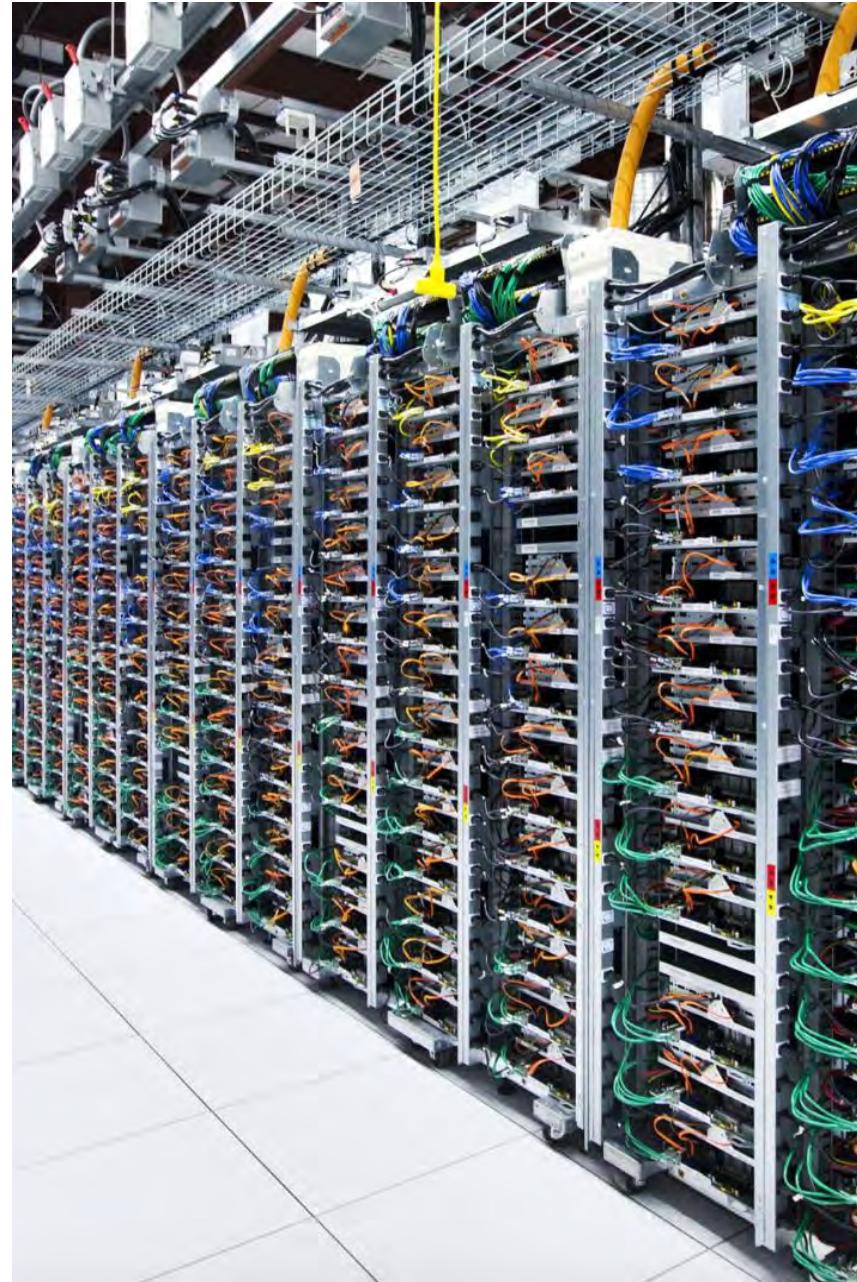


- Netflix runs its entire streaming service on Amazon Web Services (AWS). AWS provides Netflix with a scalable, reliable, and cost-effective infrastructure that allows them to deliver content to millions of users around the world.
- By using AWS, Netflix is able to quickly launch new features, process large amounts of data, and respond to changes in demand. AWS has helped Netflix reduce costs and increase agility, enabling them to focus on delivering high-quality content to their subscribers.



# AWS Regions and Availability Zones

---



# Introduction to AWS Infrastructure

Amazon Web Services is a secure, cost-effective, and reliable cloud service provider with a presence in over 190 countries. Get an overview of the global infrastructure of AWS, including Regions, Availability Zones, and Edge Locations.

# Regions, Availability Zones, and Edge Locations

## Regions



- AWS has 25 Regions globally, made up of geographically separated data centers.
- Each Region is a separate geographic area, designed to be isolated from every other Region.
- Resources aren't replicated across regions unless you do so specifically.

# Regions, Availability Zones, and Edge Locations

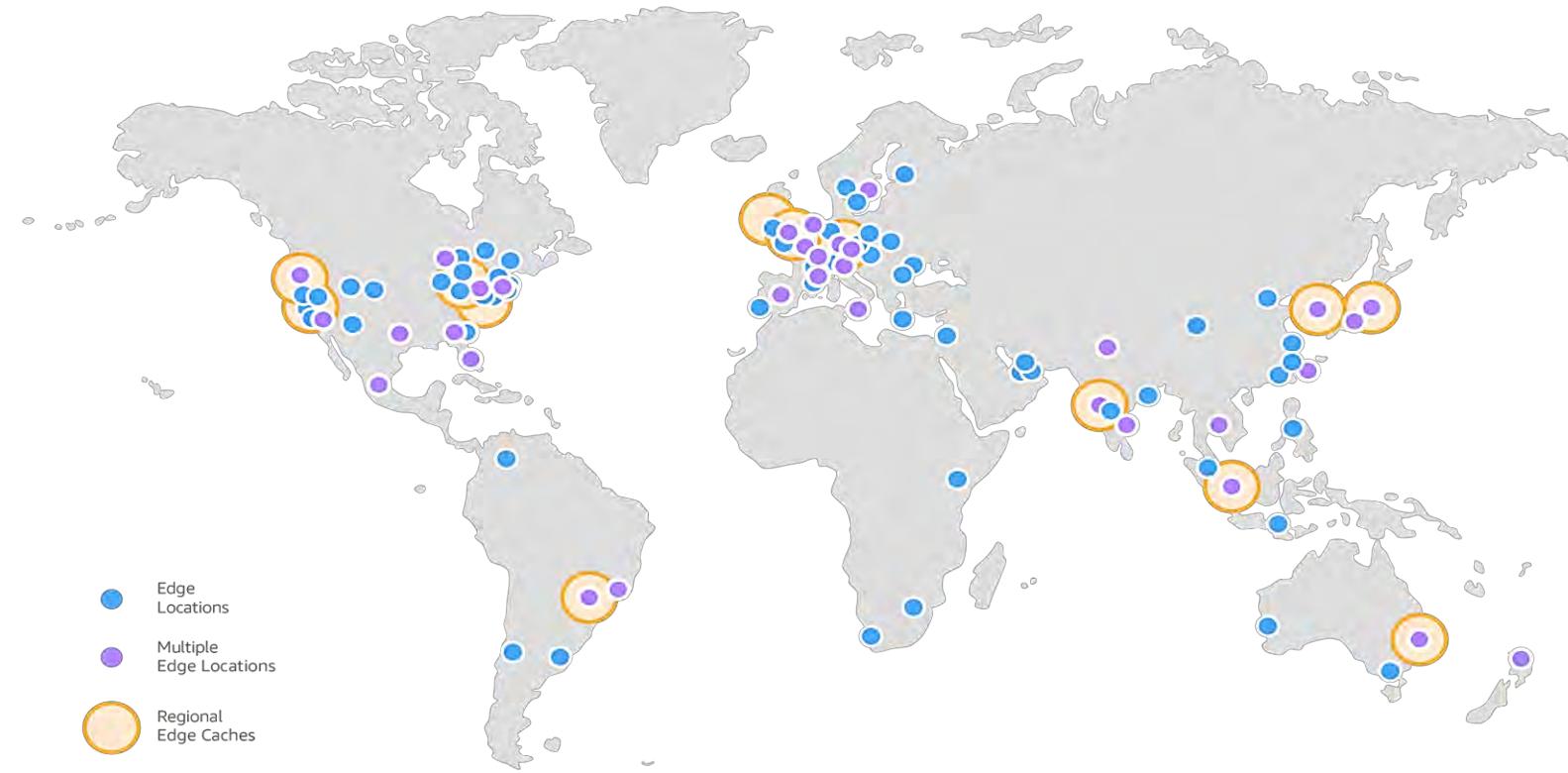
## Availability Zones



- Each Region is made up of two or more Availability Zones.
- An Availability Zone is simply a data center with redundant power, networking, and connectivity, located within the same Region.

# Regions, Availability Zones, and Edge Locations

## Edge Locations



- Edge Locations are endpoints for AWS CloudFront, which is a content delivery network (CDN) that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment.

# Benefits of using Regions and Availability Zones

---

- **Highly Available**
- **Scalable and Flexible**
- **Cost-Effective**
- **Secure**



# Understanding Regions

Discover everything you need to know about AWS regions. From their purpose to how to choose the right one, this presentation will take you on a journey through AWS' global infrastructure.

# What are AWS Regions?

---

## AWS infrastructure

- Regions are physical locations where AWS has a presence.
- This presence includes data centers and other AWS services.

## Region independence

- Regions operate independently from each other, which means that they have their own endpoints and individual availability zones.

# What are AWS Regions?

---

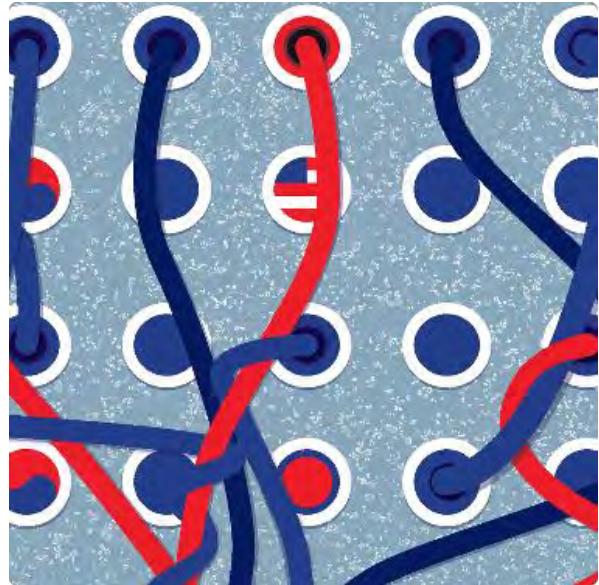
## Regional services

- Each region offers a specific range of services, which can vary depending on the region's location or local regulations.

## Global network

- AWS regions are connected through a global network that provides low latency and high throughput connections between regions and services.

# Why Choosing the Right Region is Crucial



**Latency**



**Legal requirements**

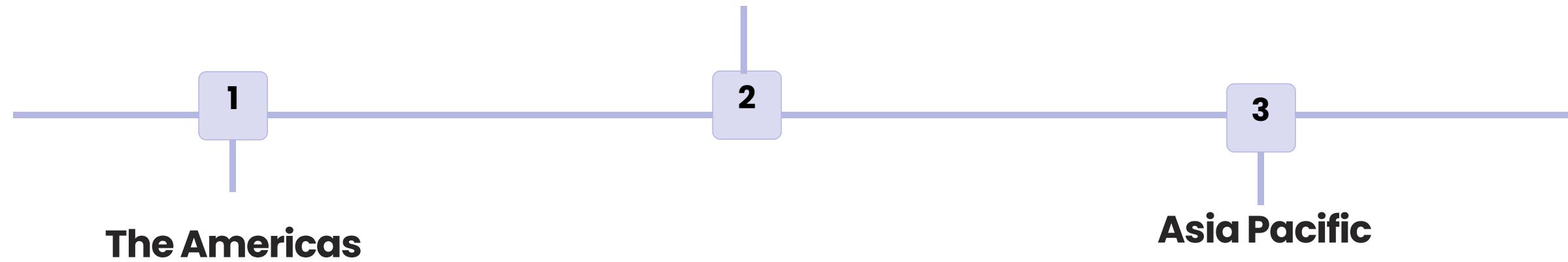


**Resilience**

# AWS Coverage Around the Globe

## EMEA

AWS has strong coverage in Europe, the Middle East, and Africa, with regions in Ireland, Frankfurt, London, and more.



## The Americas

AWS has multiple regions serving Canada, the United States, and South America, including Brazil and Argentina.

## Asia Pacific

AWS has several regions in the Asia Pacific area, including China, India, and Australia, and is expanding in the region with new facilities.

# Factors to Consider When Selecting a Region

---

1

**Workload location**

2

**Service availability**

3

**Regulation and compliance**

4

**Disaster recovery**

# Architecting for Resilience

## Resilience in the Cloud



**Cloud computing**

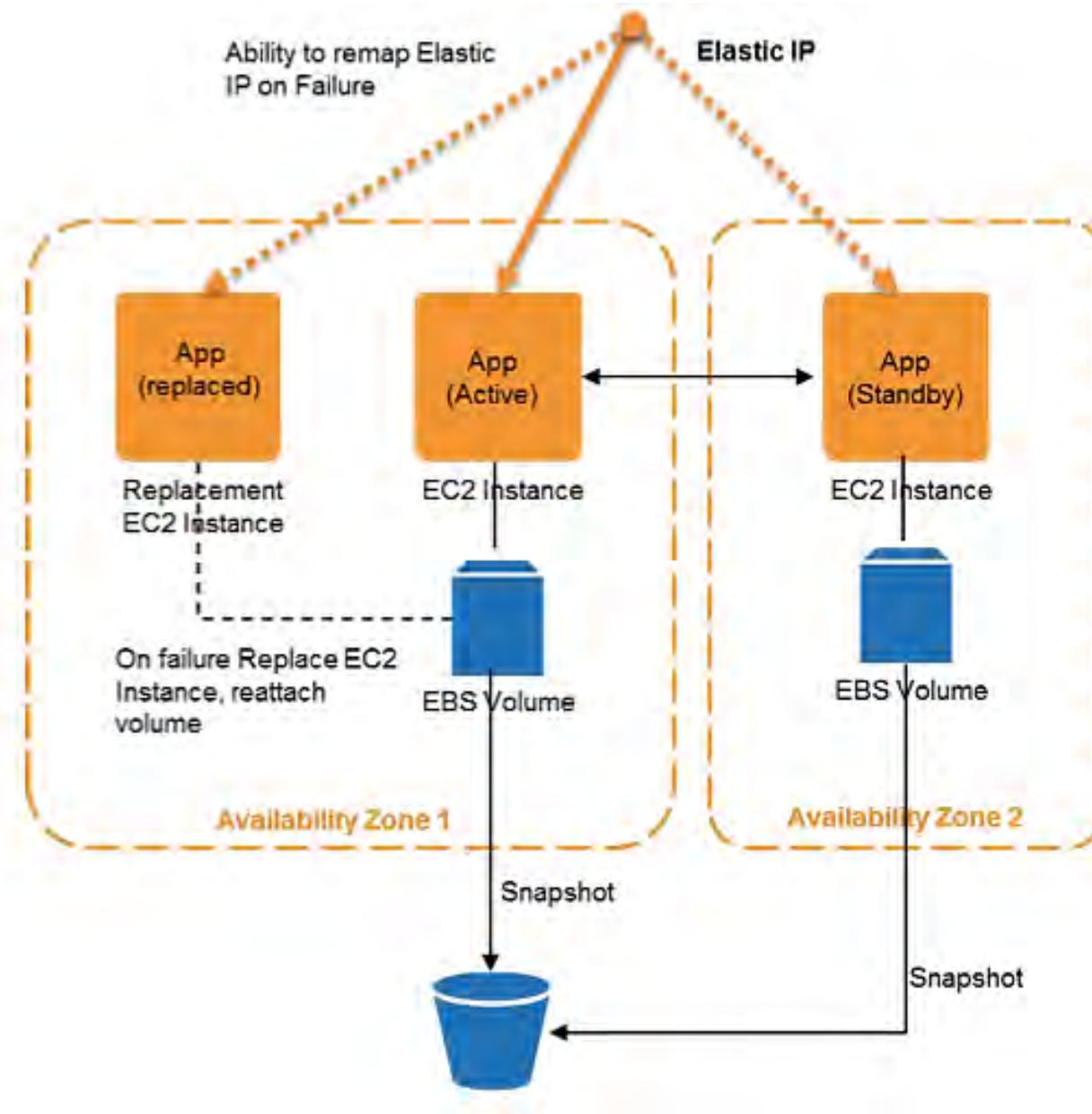


**Security**



**Connectivity**

# Fault-Tolerant Applications



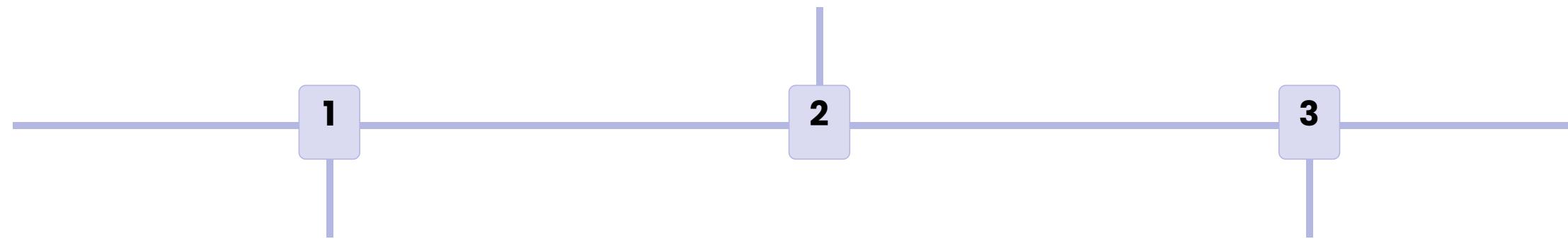
# Design Strategies for Fault-Tolerant Applications

Strategy	Description	Advantages
Decoupling	Components operate independently.	More resilient against individual component failure.
Redundancy	Multiple copies of components.	Protects against hardware or software failures.
Automated recovery	Automatically recover from failures.	Minimizes downtime and human error.

# Case Studies of Highly Available Architectures

## Streaming

Netflix's Chaos Monkey, which randomly shuts down components to test system resilience.



## DNS

Route 53 for global traffic routing with failover to a second region.

## Financial Services

JPMorgan Chase using AWS for their high performance, low-latency applications.

# Best Practices for Architecting for Resilience

---

## Testing

- Test for failures and unexpected behavior.

## Redundancy

- Have backups for critical components.

## Automation

- Automate where feasible for faster recovery times and fewer human errors.

# Edge Locations and Their Significance

---

## What are Edge Locations?

- Edge locations are servers that are geographically closer to the end-user, allowing for faster content delivery and lower latency.

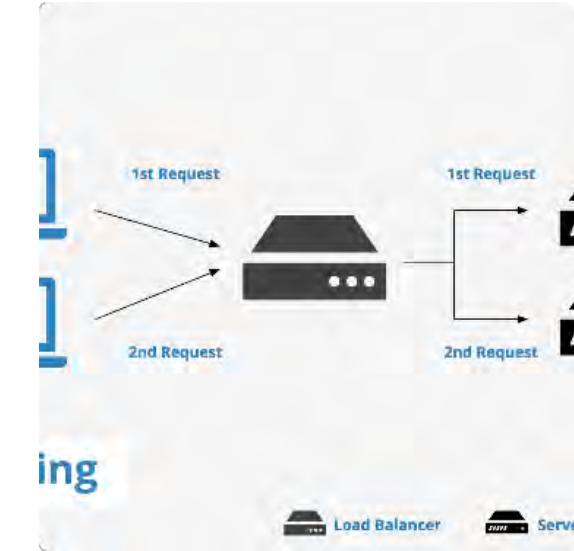
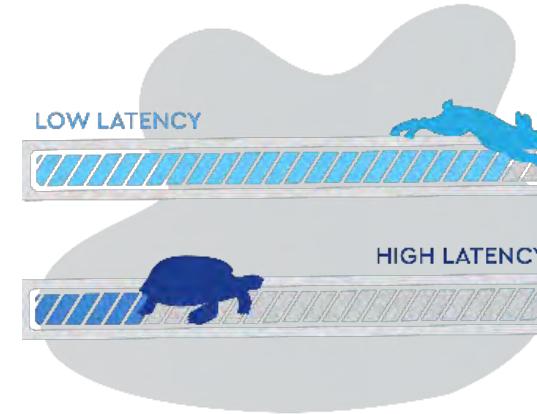
## The Purpose of Edge Locations

- Edge locations help to reduce latency and improve the user experience by caching content closer to the end-user.

# Enhanced Content Delivery and Reduced Latency



Content Delivery  
- A Beginner's



ing

## Content Delivery Networks (CDNs)

CDNs use edge locations to store cached versions of content, reducing the distance between the user and the server.

## Reduced Latency

Edge locations help to reduce latency by bringing computing closer to the end-user, bypassing the traditional cloud infrastructure.

## Load Balancing

Edge locations can also be used for load balancing, directing traffic to the most efficient server based on the user's location.

# Benefits of Edge Computing

---

- **Increased Security**
- **Improved Reliability**
- **Cost Savings**
- **Reduced latency/increased speed**
- **Increased productivity**

# Real-World Case Studies of Edge Computing

---

## Retail

A major retail company uses edge computing to improve their inventory management system in their physical stores.

- Edge computing allows for faster data processing of inventory data on local devices
- Eliminates the need for constant communication to a centralized server
- Reduces the risk of network congestion and device failure

## Manufacturing

A manufacturing company optimizes their production line with edge computing.

- Edge devices monitor machine performance and identify inefficiencies in real-time
- Critical data is processed locally, reducing the risk of system failure or loss of data
- Allows for predictive maintenance, reducing downtime and improving overall efficiency

# iamneo



Amazon Virtual  
Private Cloud (VPC)



## AWS VPC

# Introduction to AWS VPC

---

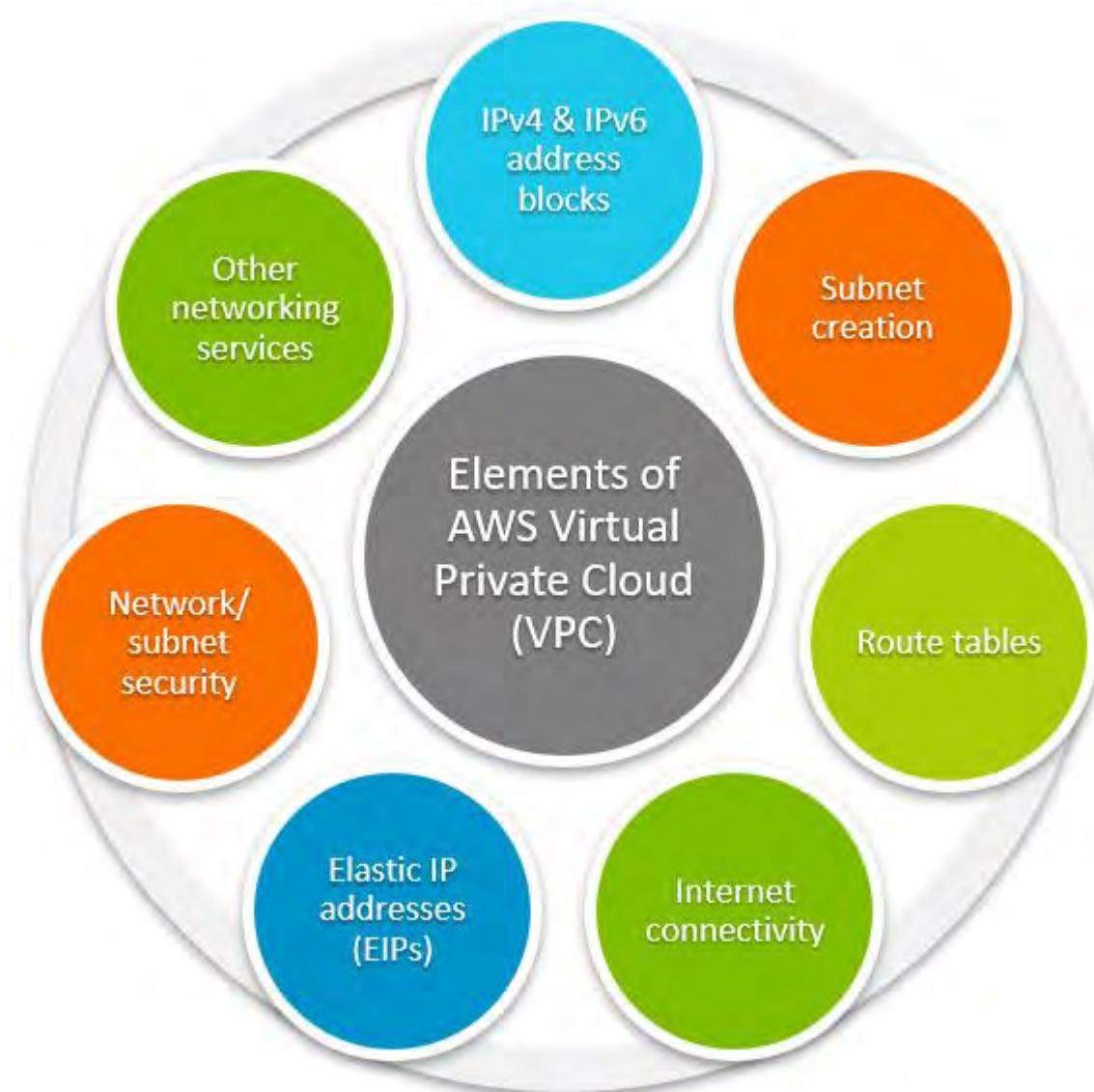
With Amazon Virtual Private Cloud (Amazon VPC), you can launch AWS resources in a logically isolated virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

## Benefits of Using VPC

- Improved Security 
- Better Control 
- Increased Flexibility 
- Cost Savings 



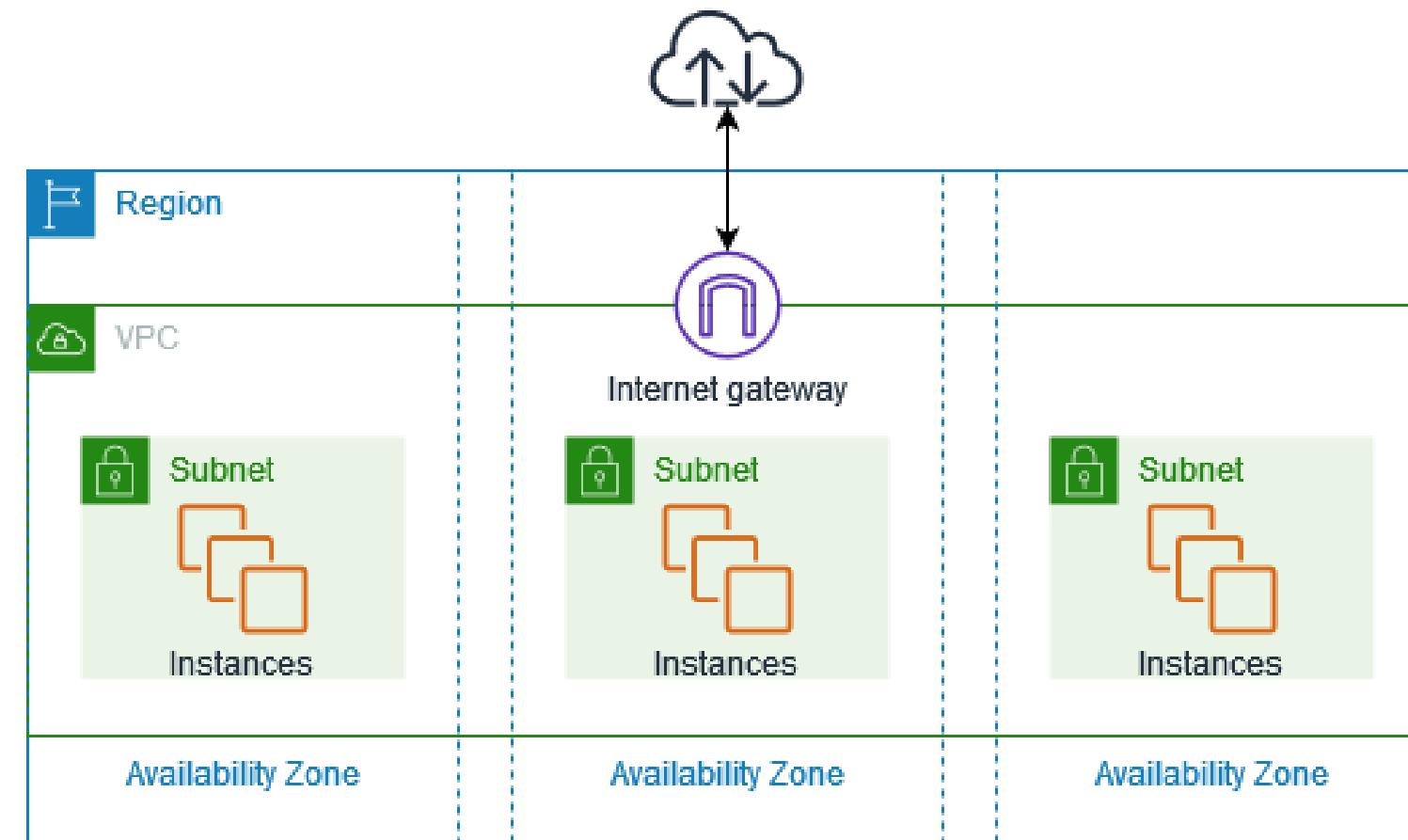
# Elements of AWS VPC



# Major components of a VPC

The following diagram shows an example VPC. The VPC has one subnet in each of the Availability Zones in the Region, EC2 instances in each subnet, and an internet gateway to allow communication between the resources in your VPC and the internet.

- **VPC CIDR blocks**
- **Subnet CIDR blocks**
- **Route Table**
- **Internet Gateway**

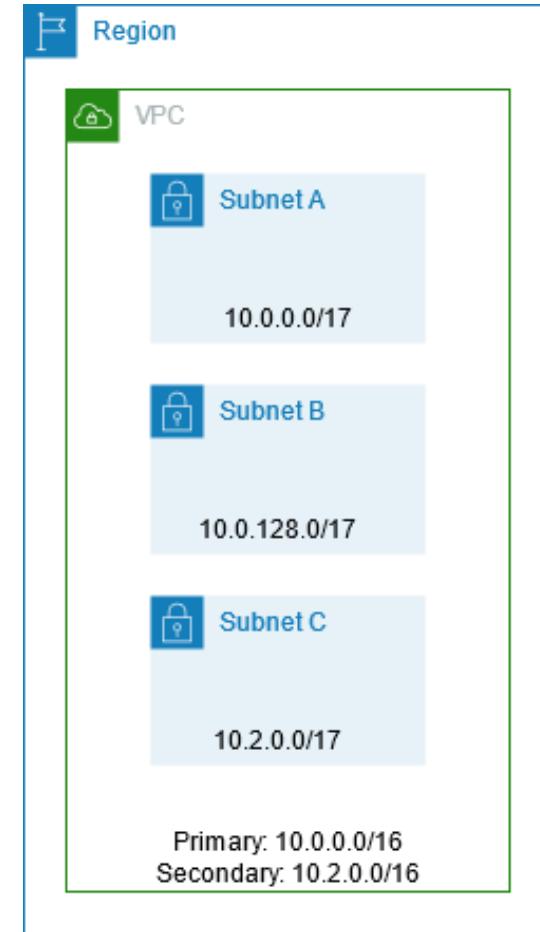


# VPC CIDR blocks

Amazon VPC supports IPv4 and IPv6 addressing. A VPC must have an IPv4 CIDR block associated with it. You can optionally associate multiple IPv4 CIDR blocks and multiple IPv6 CIDR blocks to your VPC.

## Example VPC CIDR blocks – IPv4

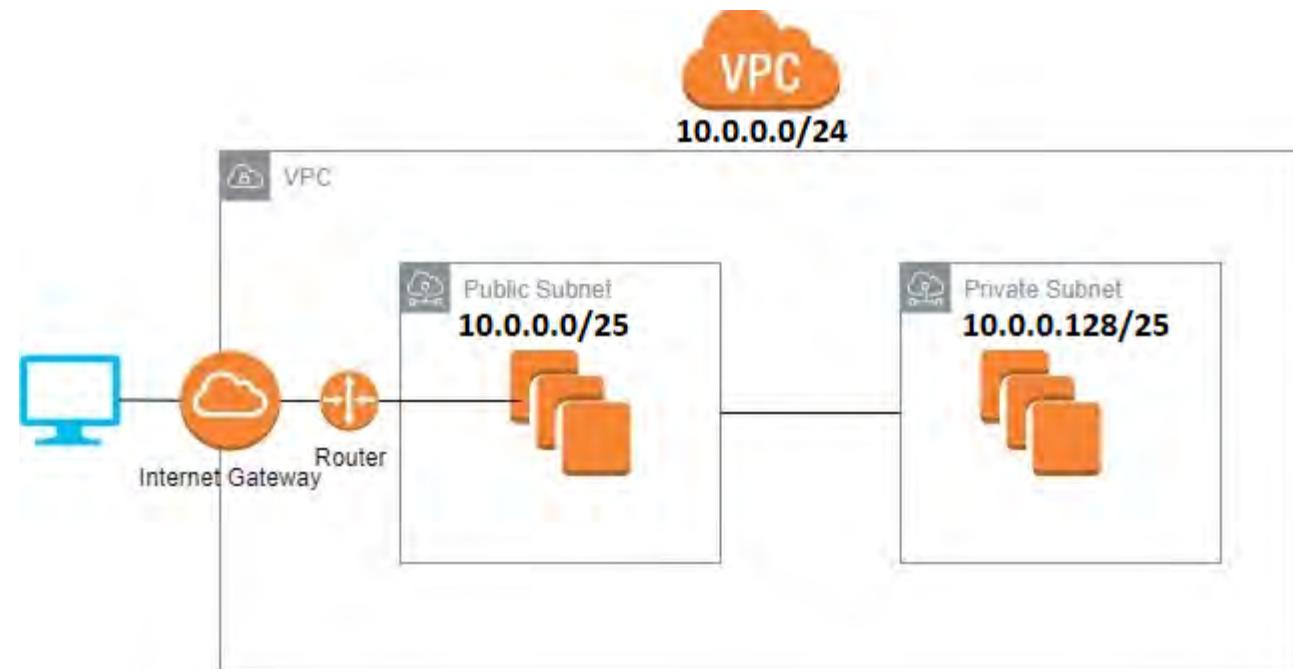
RFC 1918 range	Example CIDR block
10.0.0.0 – 10.255.255.255 (10/8 prefix)	10.0.0.0/16
172.16.0.0 – 172.31.255.255 (172.16/12 prefix)	172.31.0.0/16
192.168.0.0 – 192.168.255.255 (192.168/16 prefix)	192.168.0.0/20



# Subnet CIDR blocks

The CIDR block of a subnet can be the same as the CIDR block for the VPC (for a single subnet in the VPC), or a subset of the CIDR block for the VPC (to create multiple subnets in the VPC). The allowed block size is between a /28 netmask and /16 netmask. If you create more than one subnet in a VPC, the CIDR blocks of the subnets cannot overlap.

**Example:** if you create a VPC with CIDR block 10.0.0.0/24, it supports 256 IP addresses. You can break this CIDR block into two subnets, each supporting 128 IP addresses. One subnet uses CIDR block 10.0.0.0/25 (for addresses 10.0.0.0 – 10.0.0.127) and the other uses CIDR block 10.0.0.128/25 (for addresses 10.0.0.128 – 10.0.0.255).



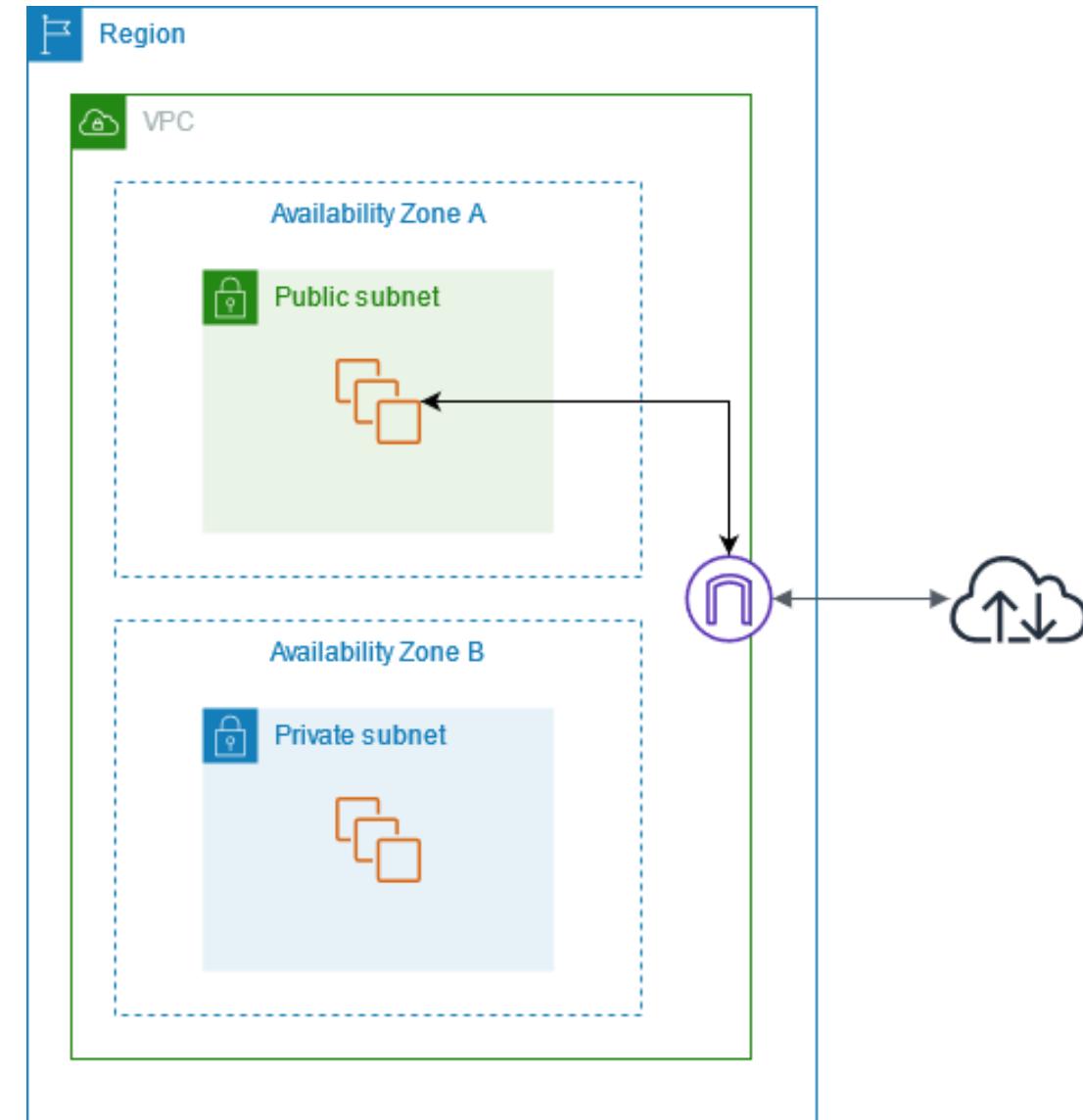
# VPC Internet Gateways

## Why use it?

With an Internet Gateway, you can enable communication between the instances in your VPC and the internet while keeping them secure.

## How to configure?

Create an Internet Gateway, Attach the internet gateway to a VPC, and then update the route table of the VPC subnet to point traffic to the internet gateway.



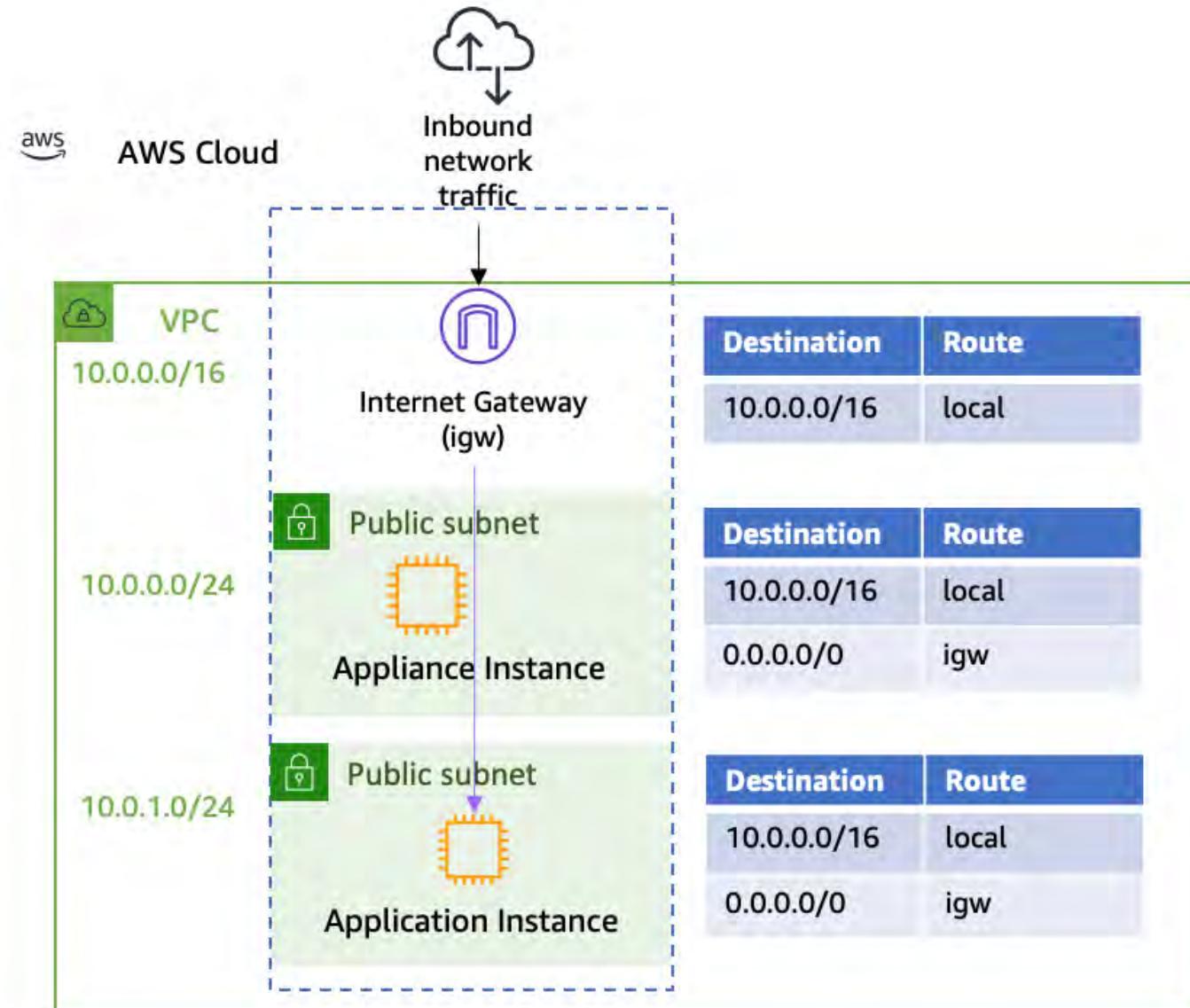
# VPC Route tables

## Subnets

Divide a VPC's IP address range into smaller CIDR blocks to utilize resources more efficiently.

## Route Tables

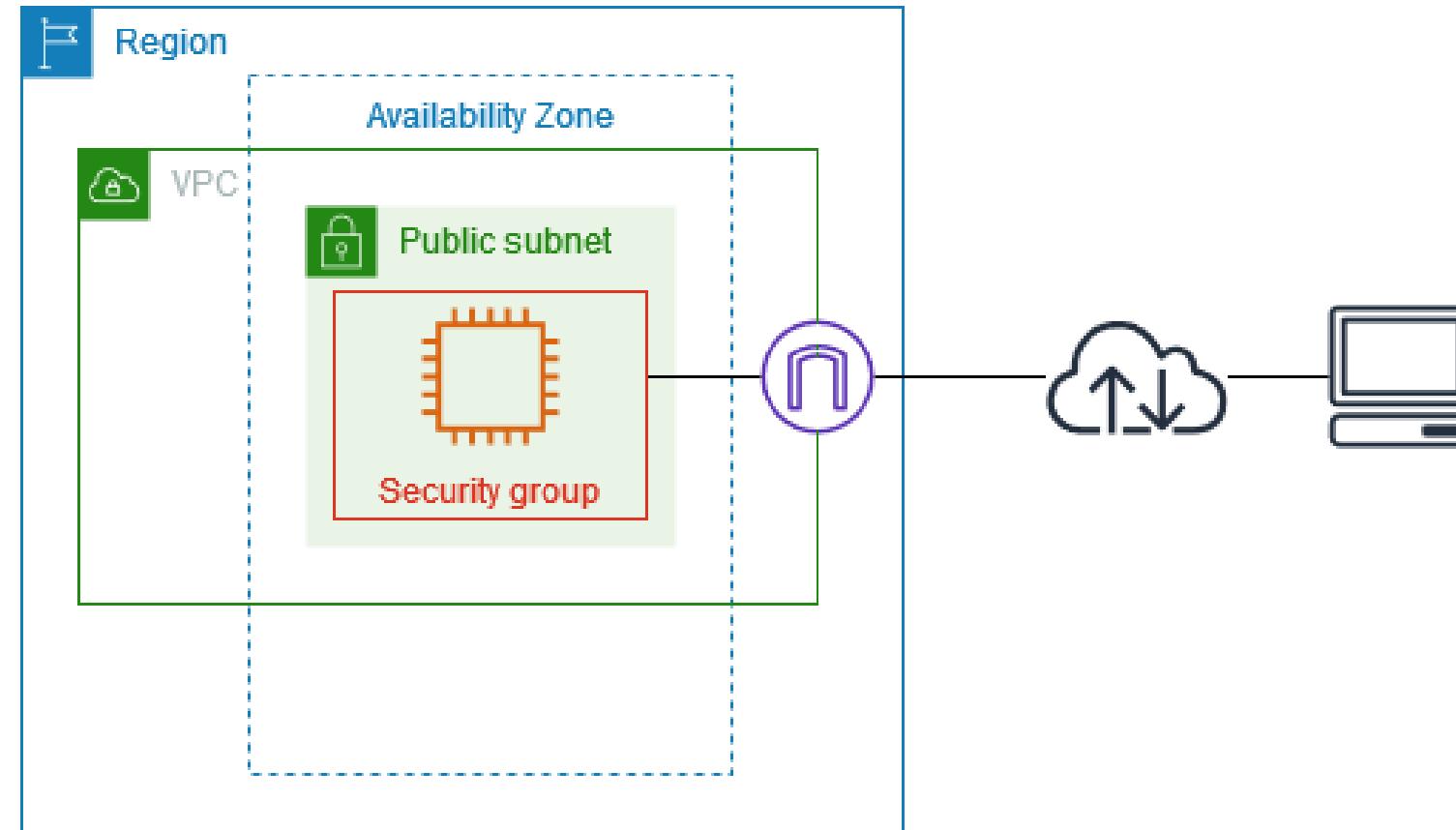
Define how traffic should be directed between subnets and to the internet. Routing can be customized based on the destination IP address.



# VPC with Public Subnet

## Public Subnet

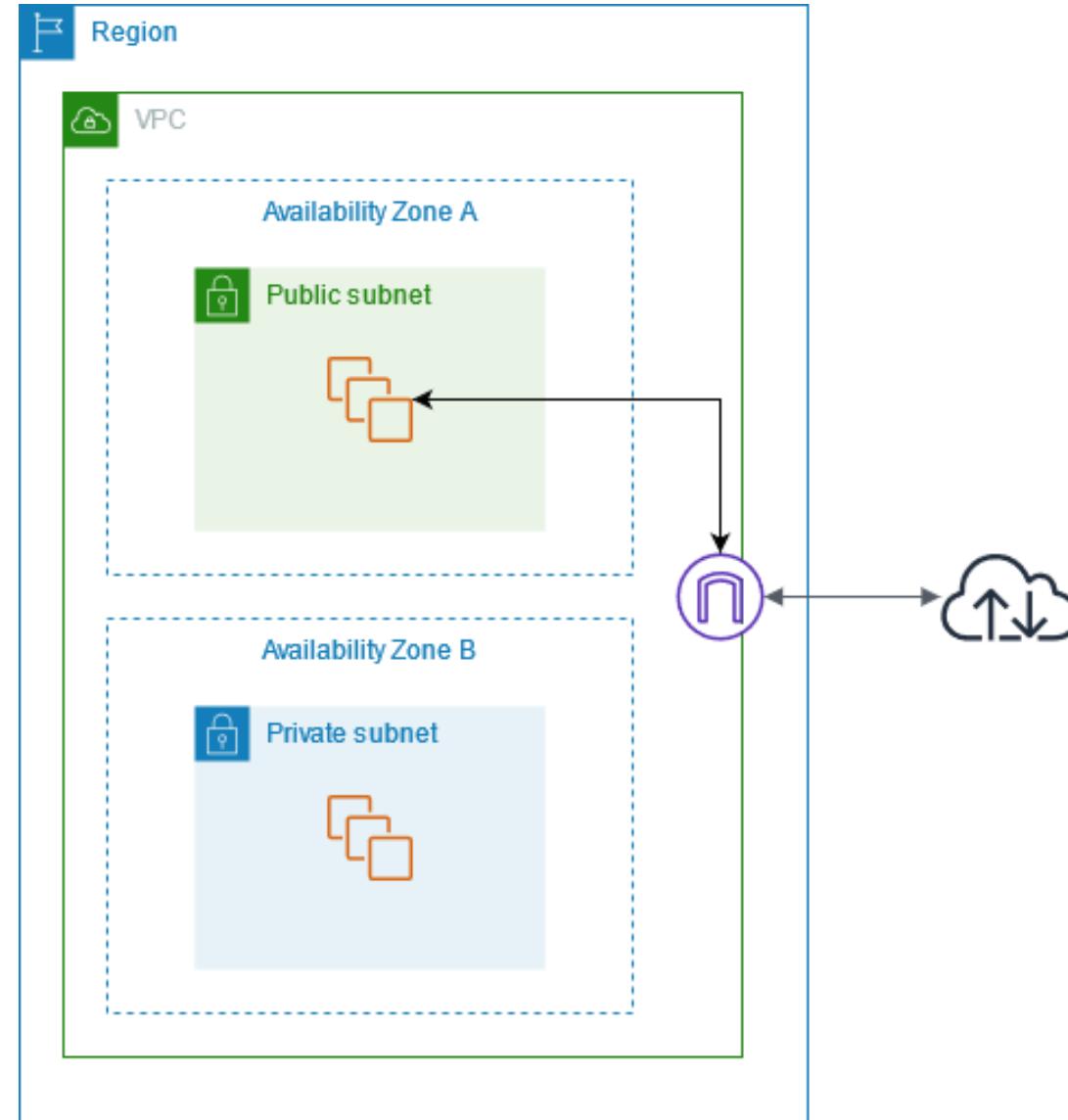
A public subnet is a subnet that is associated with a Route Table that has a route to an Internet Gateway (Igw). This route allows access from the Public Internet to the subnet.



# VPC with Private Subnet

## Private Subnet

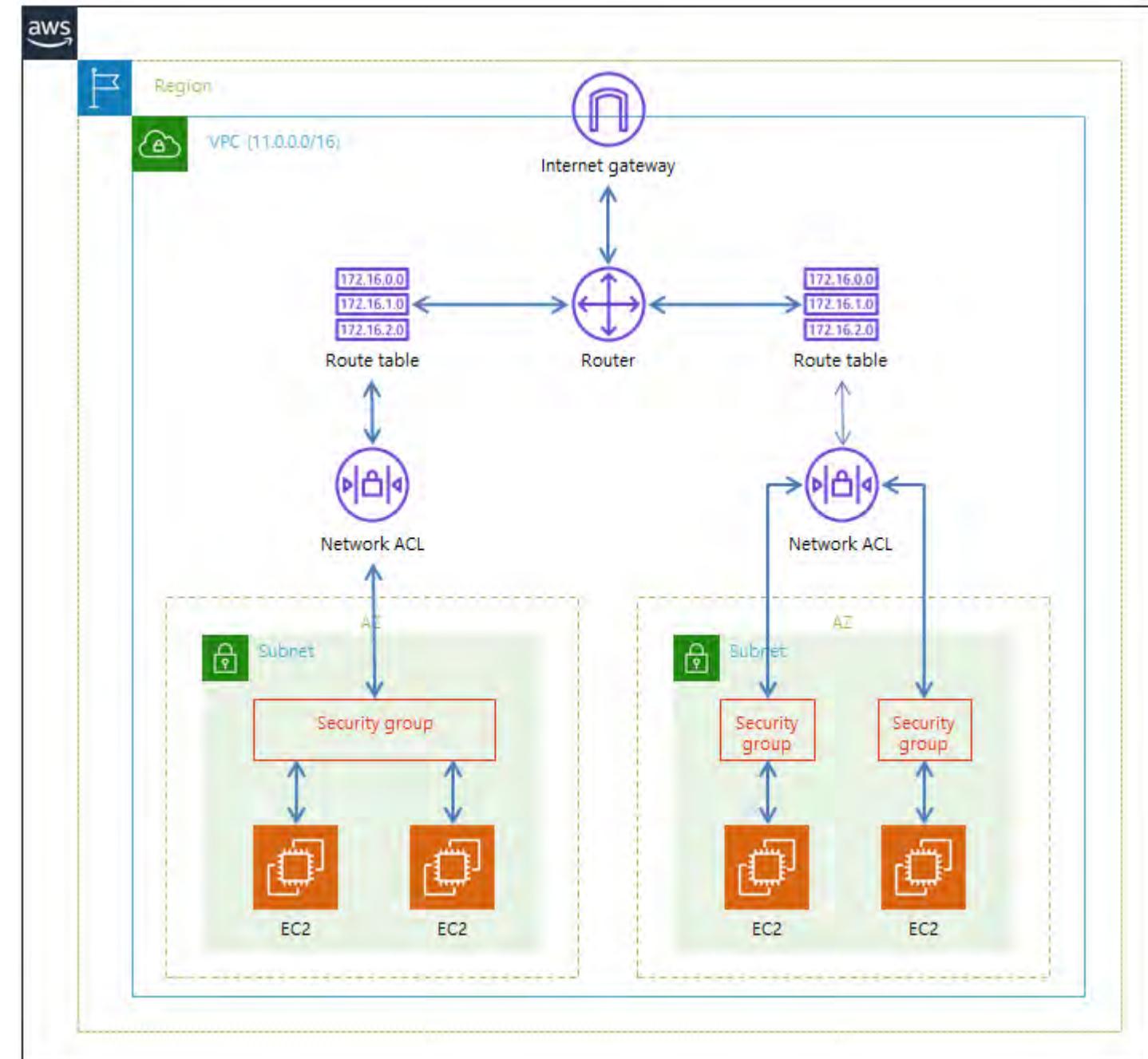
- A private subnet is a subnet that is associated with a route table that doesn't have a route to an internet gateway.
- Resources in private subnets cannot communicate with the public internet.
- AWS resources within the same VPC CIDR can communicate via their private IP addresses.



# Network ACL in AWS VPC

## Network ACL

- A network access control list (ACL) allows or denies specific inbound or outbound traffic at the subnet level.
- You can use the default network ACL for each VPC, or you can create a custom network ACLs for your VPCs, with rules that are similar to the rules for your security groups.
- This provides an additional layer of security to your VPC.
- There is no additional charge for using default network ACLs.



# Elastic IP Addresses

---

## 1    What are they?

Elastic IP addresses are static IPv4 addresses that you can allocate and associate with your AWS account. They allow you to mask the failure of an instance or software.

## 2    Why use them?

They provide the flexibility to mask an instance or software failure by quickly remapping the address to another instance in your account.

## 3    How to configure?

Allocate an elastic IP address, associate the IP address with your instances or network interfaces, and then update your domain name service (DNS) records.

# Use Cases for VPC

---

## **Development and Testing**

VPC allows you to create a sandbox environment for development and testing without affecting your production environment. You can easily create and destroy instances to test your applications.

## **Web Hosting**

VPC provides a scalable and secure environment for hosting your websites. It allows you to easily configure load balancing, auto scaling, and high availability for your applications.

## **Big Data Analytics**

VPC allows you to easily deploy and scale your big data analytics applications. You can securely connect to data sources and use AWS EMR and other tools to process and analyze your data.

# Introduction to AWS VPC Creation

---

Learn how to create a Virtual Private Cloud in AWS with this step-by-step guide. We'll cover everything from IP address ranges to VPC security best practices.

## Creating a VPC in AWS

### Pre-Setup

1. AWS Account Registration and Setup
2. AWS Console Access
3. Acquiring AWS Credentials

### Setup

1. VPC Creation
2. Creation of DHCP Options Sets
3. Creation of Subnets
4. Create Internet Gateway
5. Attach the Internet Gateway to VPC

### Post Setup

1. Launch an EC2 Instance within VPC
2. Assign Static IPs
3. Configuring Elastic IP Addresses
4. Configure Security Groups and Firewalls

# Enabling Internet Connectivity within VPC

---

- **Create a Public Subnet**

You can create a public subnet that has a route table entry that points to an Internet Gateway.

- **Create a Private Subnet**

Create a private subnet that has a route table entry that points to a NAT gateway.

- **Configure Security Groups**

Add an inbound rule to the instance to allow HTTP traffic and add an outbound rule to allow all traffic.

iamneo

ANY  
Questions?



Thankyou

iamneo



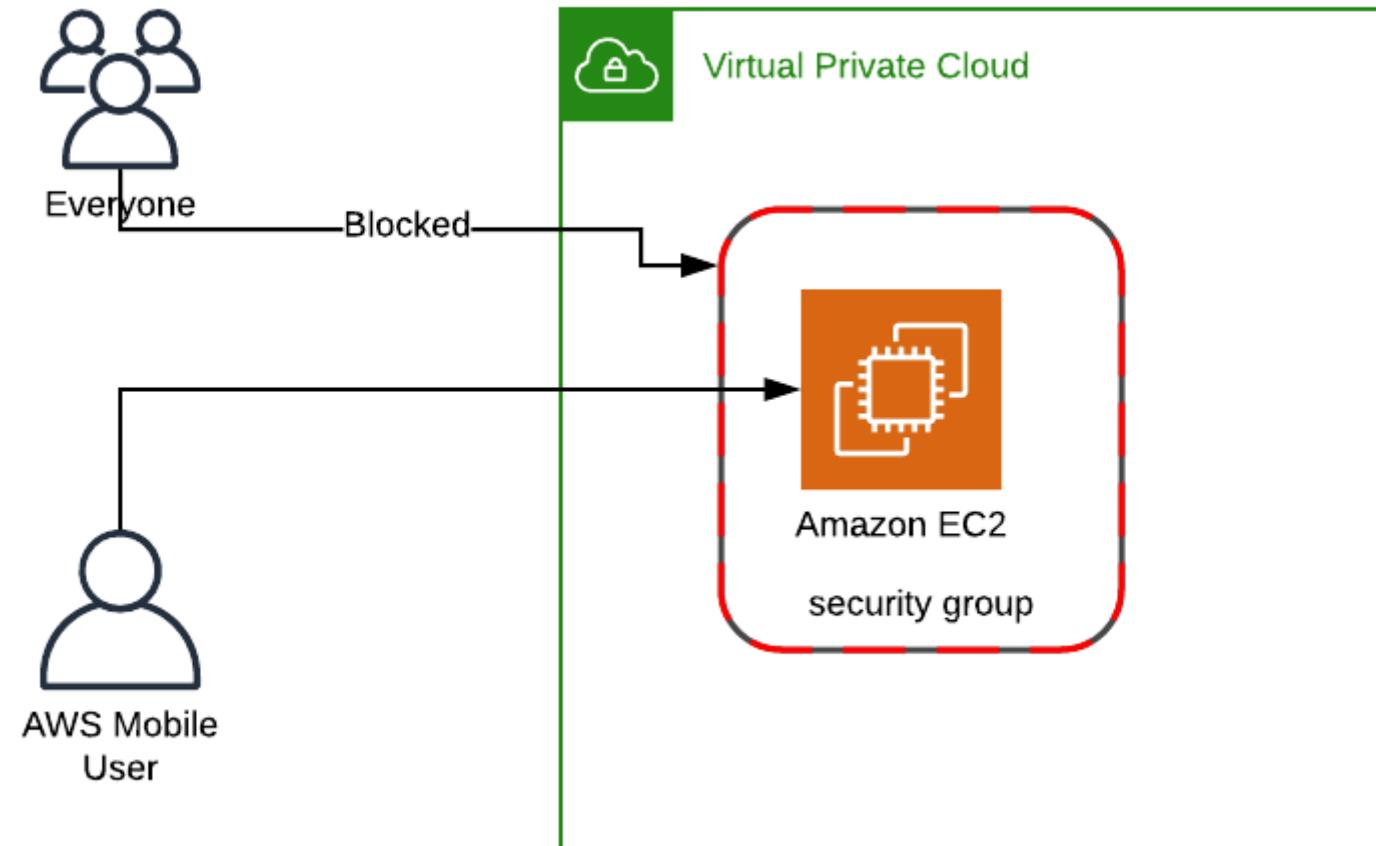
# AWS Security Groups

---

# Overview of AWS Security Groups

## Definition

AWS Security Groups are virtual firewalls that control incoming and outgoing traffic for EC2 instances. They act as a barrier between a user's instance and the internet, and can be used to filter traffic based on the rules set by the user.



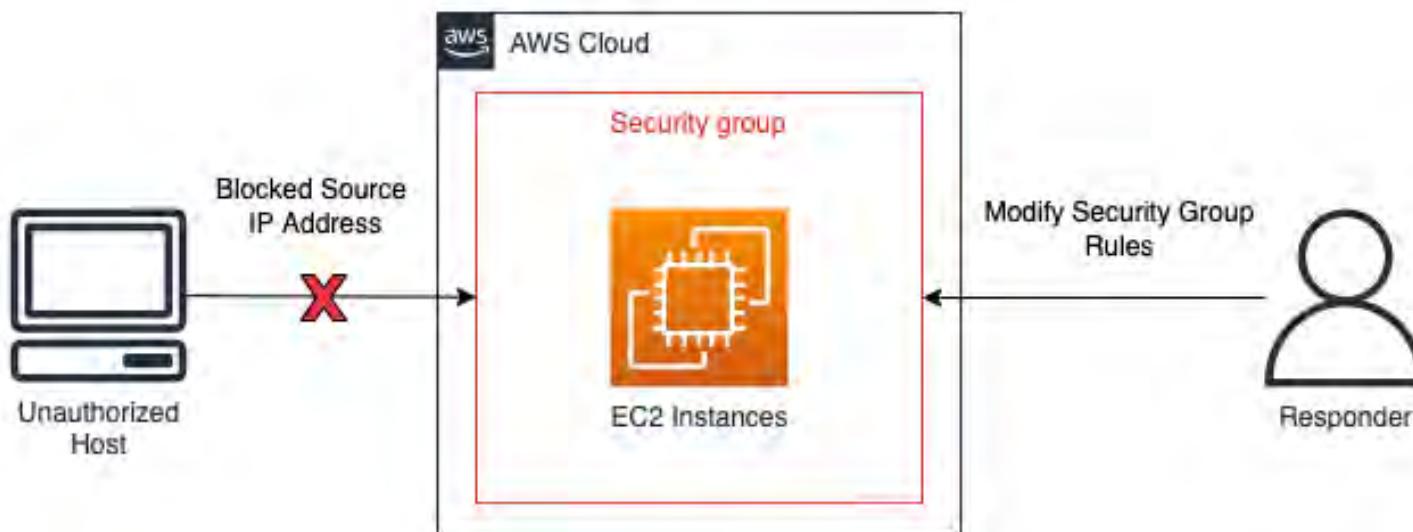
# Overview of AWS Security Groups

## Scope

- Security Groups operate at the instance level, not the subnet level.
- This means that each instance in a VPC can have its own Security Group, and that Security Groups cannot be shared across instances or subnets.

## Security Group Types

- There are two types of Security Groups: default and custom.
- Default Security Groups are created automatically and are associated with every instance launched in a VPC.
- Custom Security Groups are created by the user and can be associated with one or more instances in the VPC.



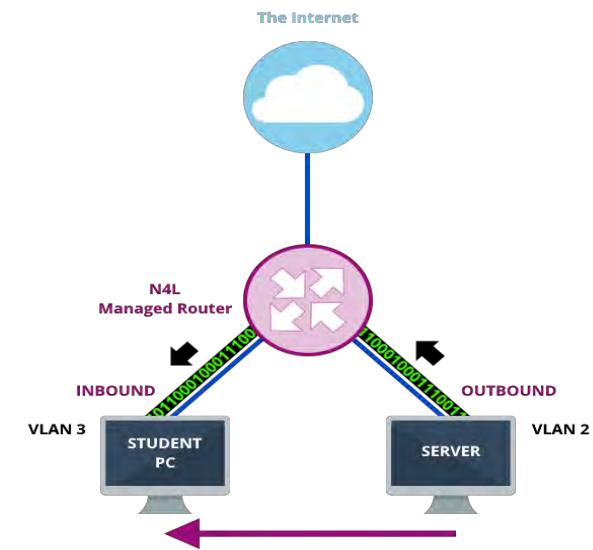
# How Security Groups Work

- Security Groups evaluate the traffic based on incoming and outgoing rules.
- Instances associate with Security Groups, and the rules apply to specific IP addresses.



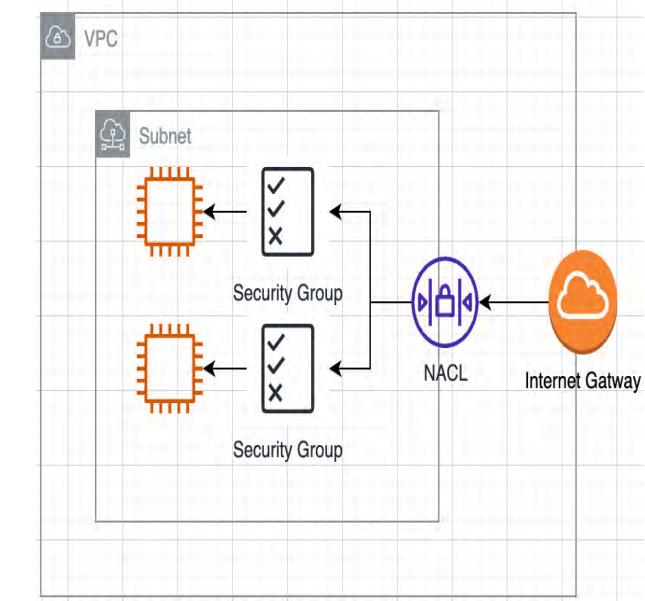
## Virtual Firewall

Security Groups act as a virtual firewall for instances, protecting them from malicious activity.



## Inbound and Outbound Rules

Security Groups use inbound and outbound rules to define the traffic that is allowed to go in and out of an instance.



## Architecture

Security Groups are part of the network architecture, allowing granular security management for your EC2 resources.

# Inbound and Outbound Rules

---

- 1 **Inbound Rules**

Inbound rules define the traffic allowed into an instance. They are stateful, meaning any outgoing traffic from the instance is automatically allowed, regardless of the Security Group outbound rules.
- 2 **Outbound Rules**

Outbound rules control the traffic allowed out of the instance. When you add a rule, add the destination IP address and port number that the instance wants to connect to.

# Creating and Configuring Security Groups

## Creating a Security Group

To create a Security Group, you need to understand your network requirements, instances, and protocols you use.

## Configuring Rules

To configure Security Group rules, you select the instance, protocol, port range, and IP address range.

## Launching Instances Using a Security Group

When launching an instance, you can associate it with one or more Security Groups. This allows for multiple firewalls to be applied to separate groups of instances.

# Best Practices for Using Security Groups

- Create separate Security Groups for different types of instances.
- Minimize the number of ports open to reduce the attack surface.
- Use a single Security Group for simplicity and manageability.



## Optimization

Optimize your Security Group rules by reviewing and modifying them periodically as your security needs evolve.



## Block All Traffic

Consider setting a default rule to block all traffic, and then open only the ports that you need.



## Backup

Backup your Security Group configuration regularly. You may need to restore it in the event of a disaster.

# Troubleshooting Security Group Issues

## Verify Security Group Rules

Ensure your Security Group has the correct rules in place, and that you're not blocking necessary traffic.

## Check Network Configuration Issues

Review your network configuration and make sure that your Security Group is associated with the right instances.

## Review Instance Log Files

Check instance log files and see if there are any errors or exceptions related to Security Group rules.

# Importance of Security Groups



## Cloud Security

As technology evolves, cloud security is becoming increasingly important for businesses to maintain their security posture. Utilizing AWS Security Groups is a crucial step towards ensuring your data center is secure.



## Don't Leave Your Data Unprotected

By leaving your data unprotected, you are putting your business and your customers at risk. Be proactive and start securing your AWS instances with Security Groups today!



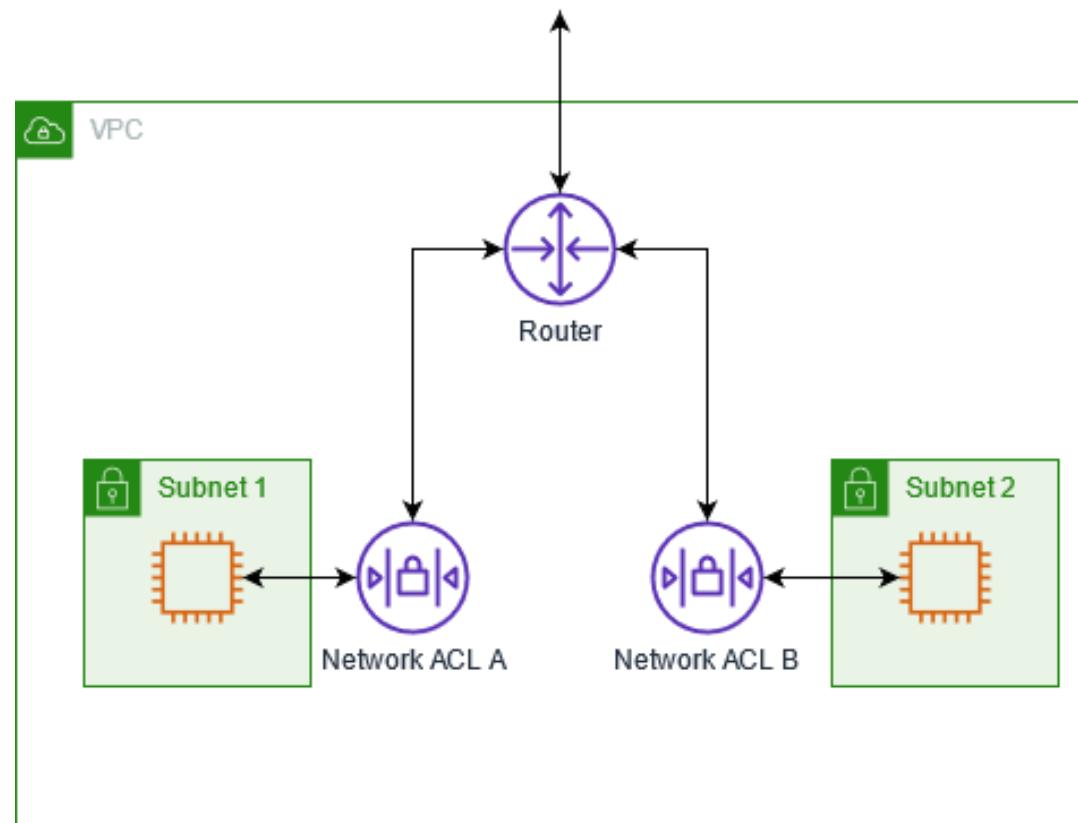
## Secure Your Success

By following best practices and configuring your security groups on AWS correctly, you can ensure the success of your business by protecting your data from potential cyber threats.

# Conclusion and Next Steps

---

- AWS Security Groups provide a powerful way to manage network security for your EC2 instances.
- By understanding the rules and configurations, and following best practices, you can ensure that your network is safe and secure.
- Next steps include reviewing your Security Group configuration, optimizing and backing up your rules regularly, and staying up-to-date with the latest AWS security features.



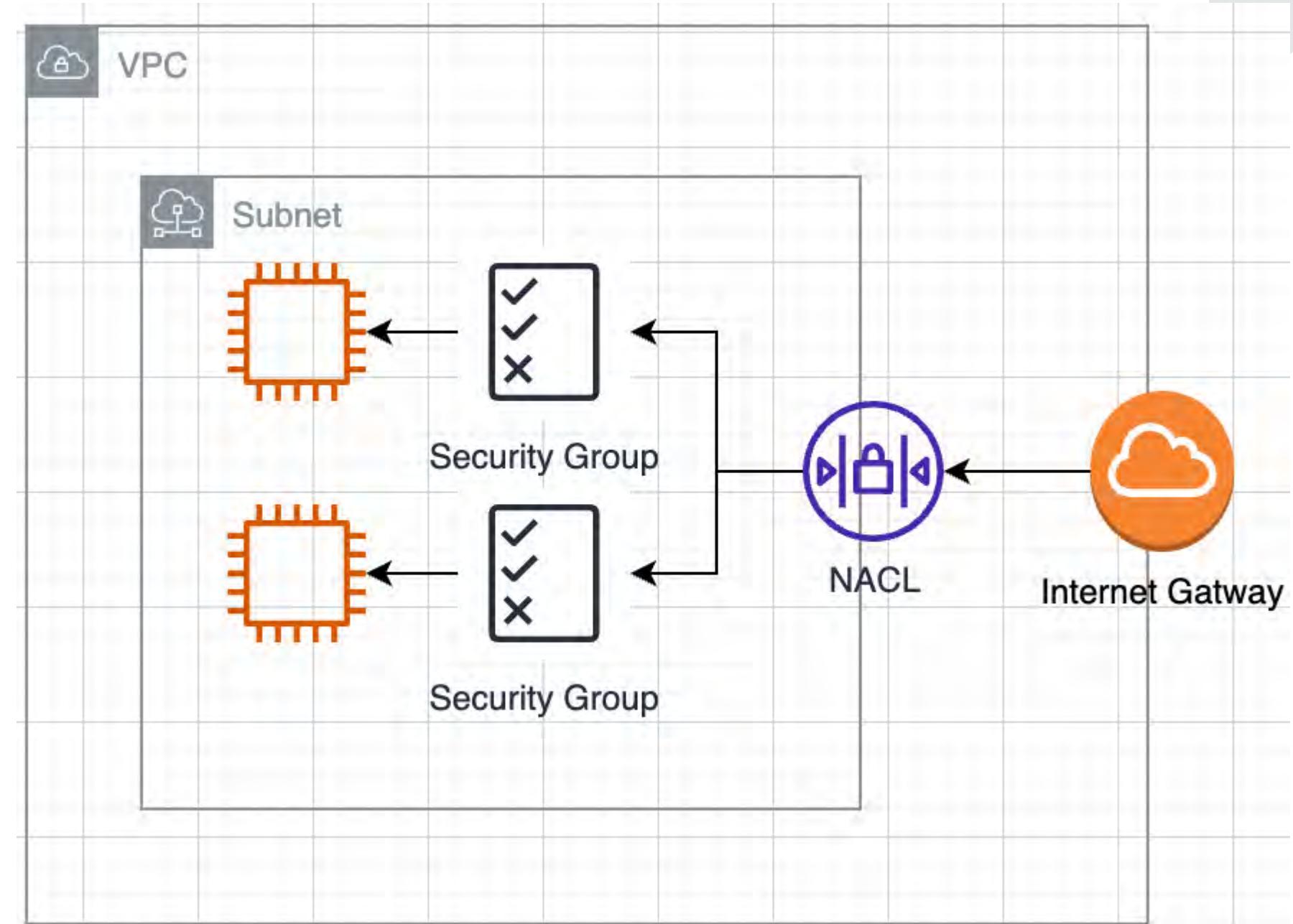
## AWS Network ACLs

# Overview of NACL

## What are Network ACLs?

Network ACLs, which stands for "Network Access Control Lists", are a virtual firewall that controls inbound and outbound traffic to and from your VPC subnets.

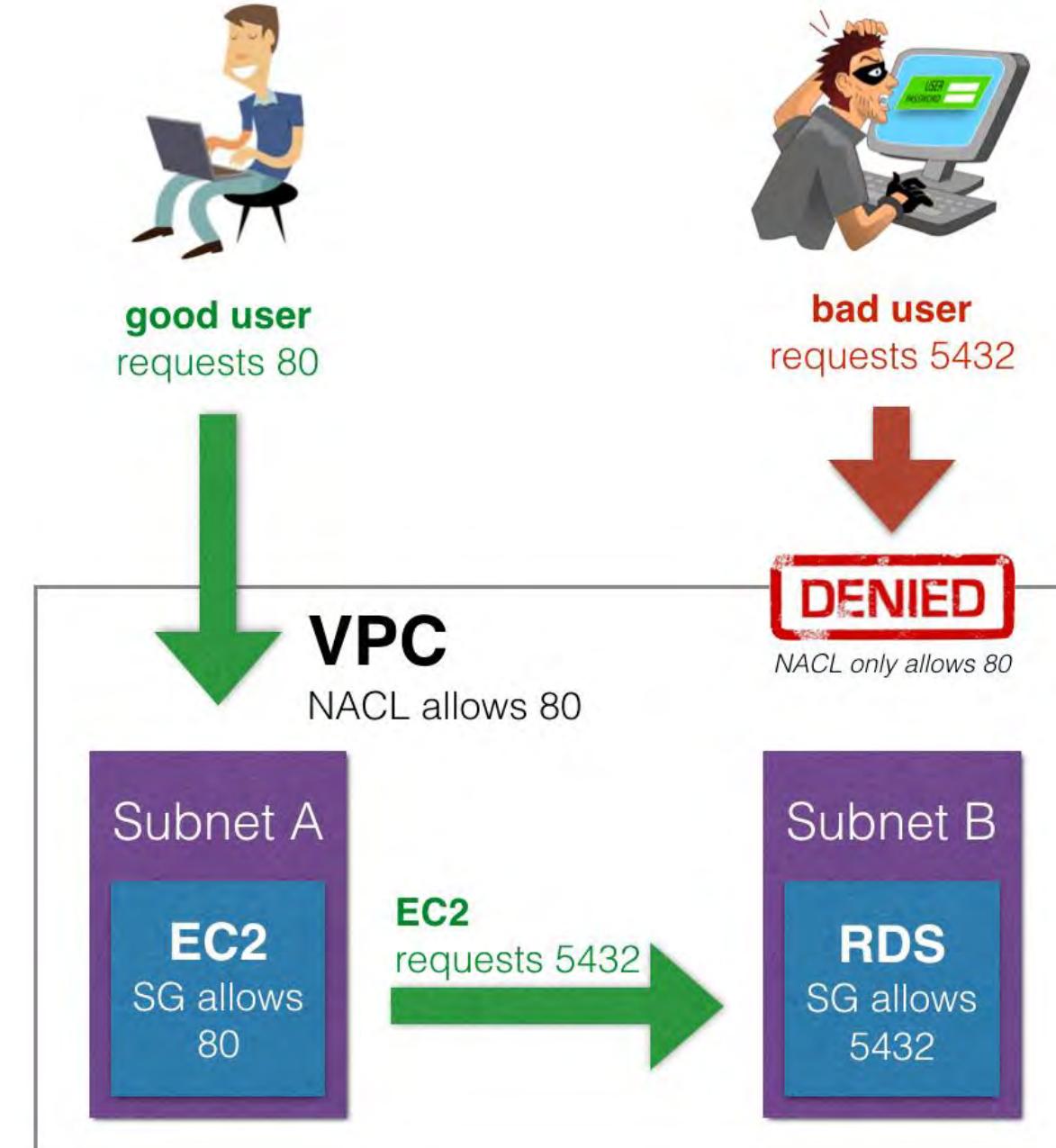
Think of them like a bouncer at a club who only lets in authorized guests.



# Usage of NACL

## When are NACLS used?

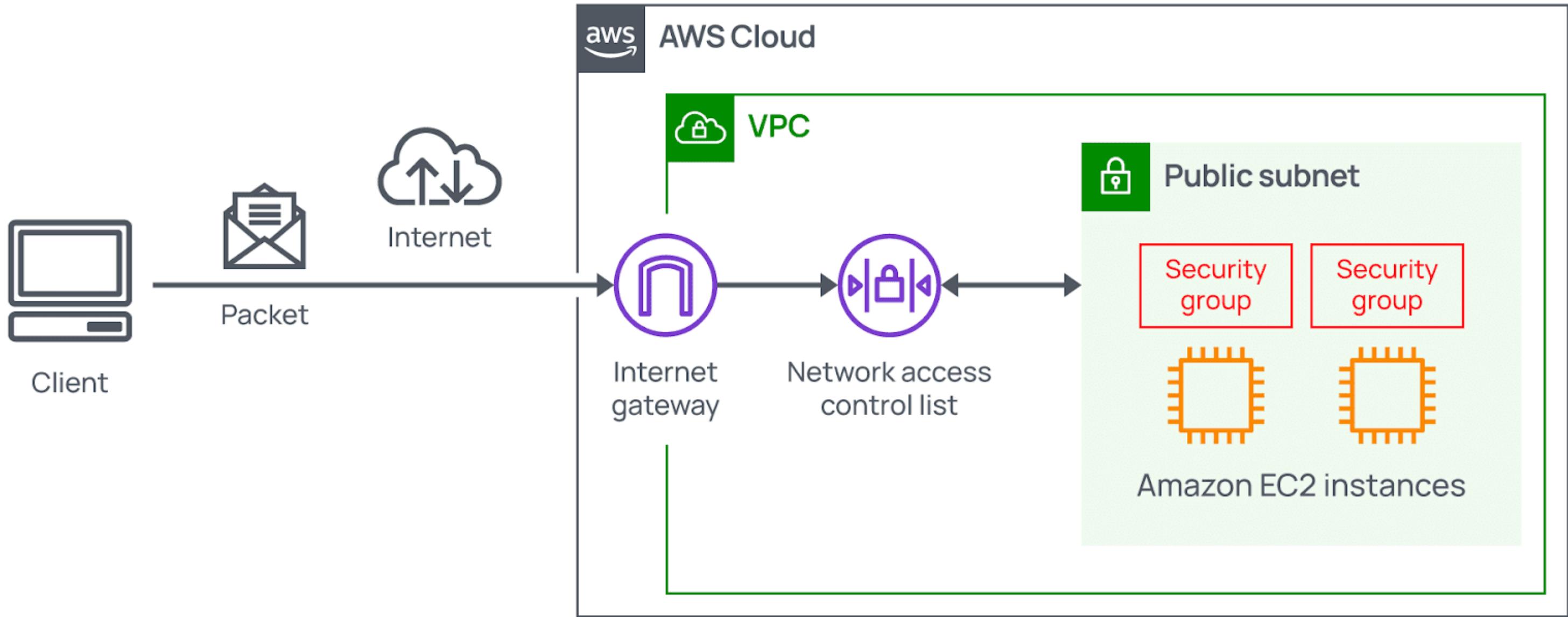
They are used to supplement the security provided by Security Groups and are the first layer of defense to protect your VPC subnets. This is important because you can never be too careful when it comes to security.



# Network ACL vs Security Group

NACL	SECURITY GROUP
Operates at the subnet level	Operates at the instance level
Supports allow rules and deny rules	Supports allow rules only
Is stateless: Return traffic must be explicitly allowed by rules	Is stateful: Return traffic is automatically allowed, regardless of any rules
Processes rules in number order when deciding whether to allow traffic	Evaluates all rules before deciding whether to allow traffic
Automatically applies to all instances in the subnets it's associated with (not subject to users to specifying the security group)	Applies to an instance only if someone specifies the security group when launching the instance, or associates the security group with the instance later on

# An example architecture of Network ACL



# Setting Up Network ACLs

1

## Create a new Network ACL

To create a new Network ACL, log in to the AWS Management Console and navigate to the VPC service. From there, select the "Network ACLs" option and click "Create Network ACL".

You will have to associate this ACL with a specific VPC subnet or group of subnets, so be sure to select the appropriate one during the creation process.

2

## Add inbound and outbound rules

Once you have created your new Network ACL, you will need to add inbound and outbound rules to control traffic flow. This is done by selecting the "Inbound Rules" or "Outbound Rules" tab and clicking "Edit".

From there, you can add rules to allow or deny specific traffic types or port ranges. Be sure to test your rules thoroughly to ensure they are working as expected.

3

## Review and save

Before you save your new Network ACL, be sure to double-check your rules for accuracy. Once you're ready, click "Save" to apply your new Network ACL to your VPC subnets.

# Best Practices

---



## Tighten Inbound Rules

Block all traffic by default, and only allow specific traffic types and ports that are necessary for your application.

## Use Network ACLs in conjunction with Security Groups

While Security Groups are stateful, Network ACLs are stateless. Use them together for added security.



## Keep Network ACLs simple

The more rules, the more complex your network becomes. Keep your network ACLs as simple and straightforward as possible.

# Common Issues with NACL

---

## 1 Overlapping Rules

If rules overlap or conflict with each other, the most permissive rule takes precedence. Make sure your rules don't contradict themselves.

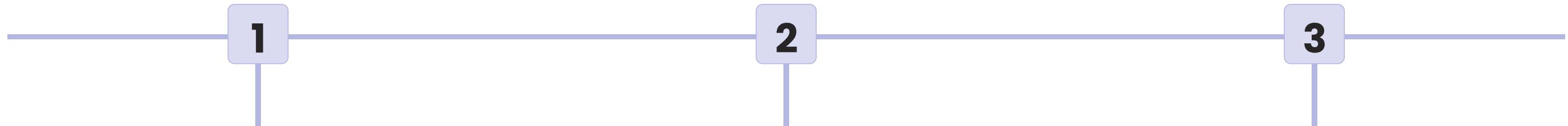
## 2 Confusing Statelessness

The stateless nature of Network ACLs can be confusing at first. Remember that each rule applies to each packet of traffic that matches, regardless of the traffic's originating connection state.

## 3 Limitations

Network ACLs can only filter traffic to and from subnet gateways. They cannot filter traffic between instances on the same subnet.

# Case Studies of NACL



## Software as a Service Company

A SaaS company used Network ACLs to maintain compliance with government regulations, blocking traffic from countries deemed high risk.

## Online Retailer

An online retailer used Network ACLs to restrict access to sensitive data such as customer payment information to specific staff and systems, minimizing their exposure to risk.

## Research Institution

A research institution used Network ACLs to filter out unwanted traffic to their research networks, using them to enforce policies that restrict access to specific resources such as high-performance computing clusters.

iamneo



Thankyou

---

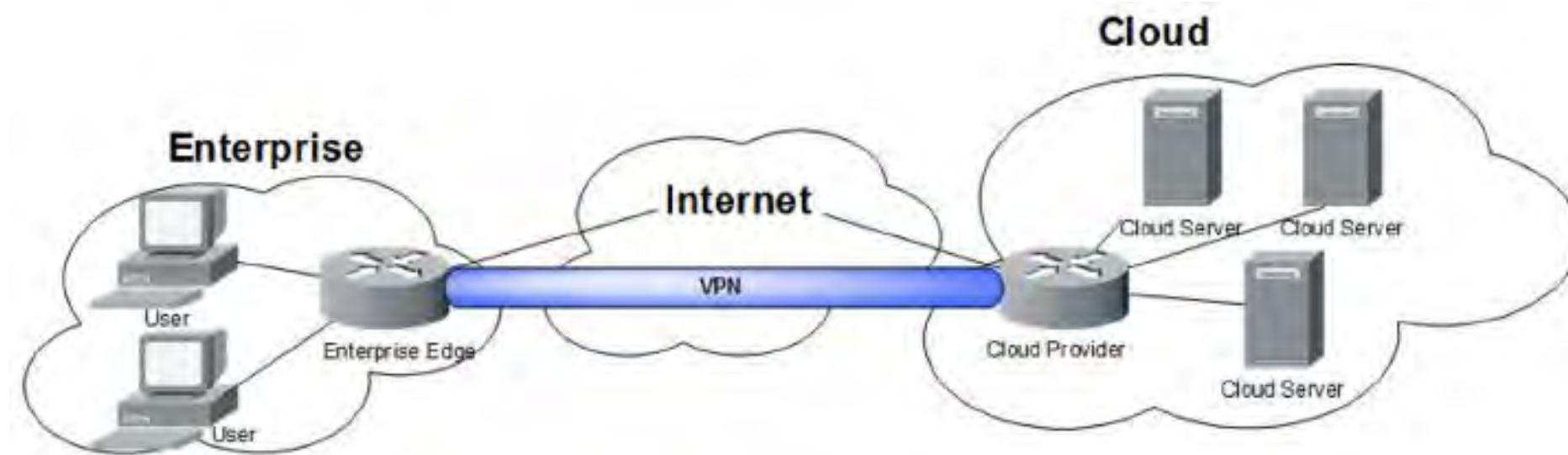


# AWS VPN and Direct Connect

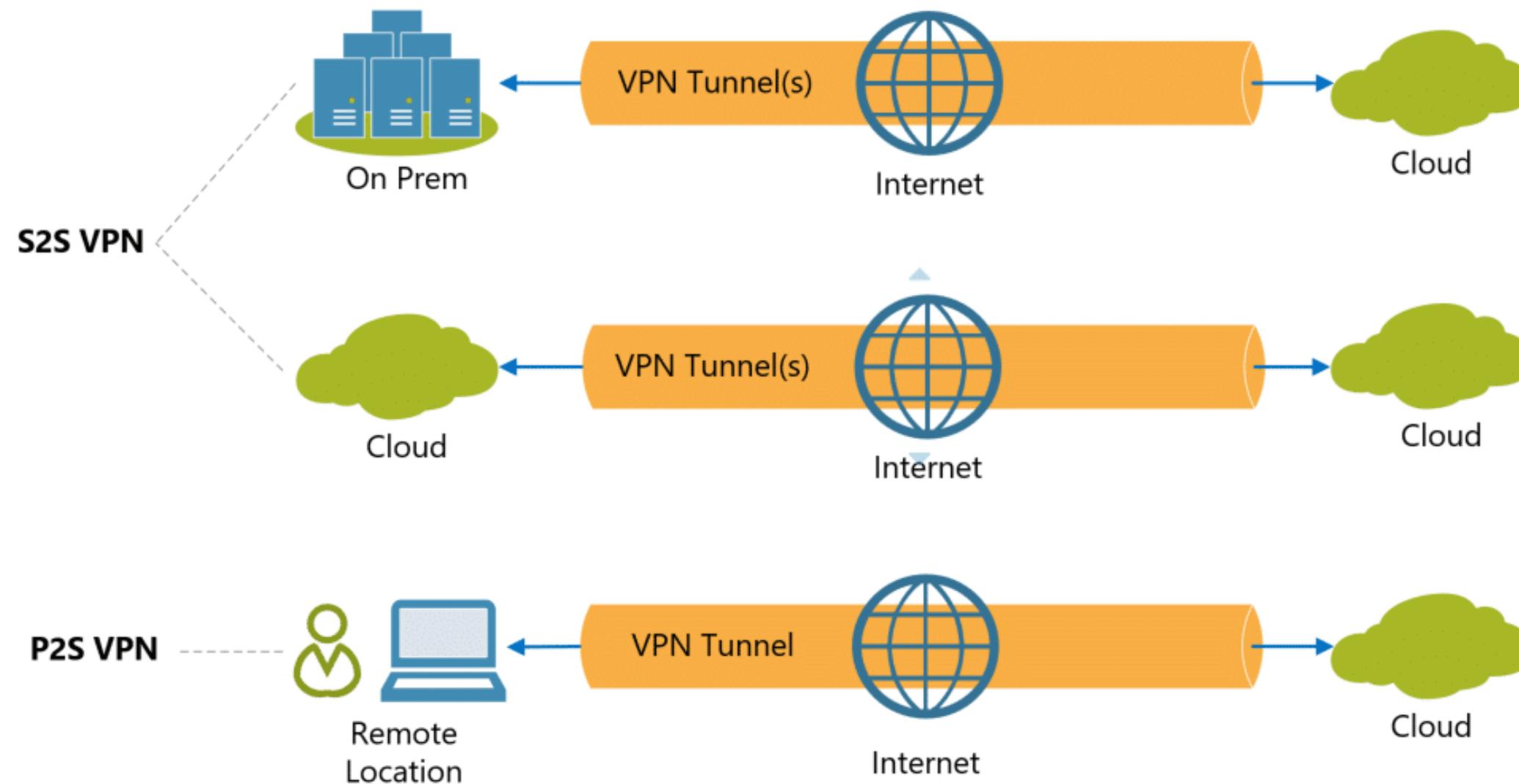
---

# Overview of Virtual Private Network

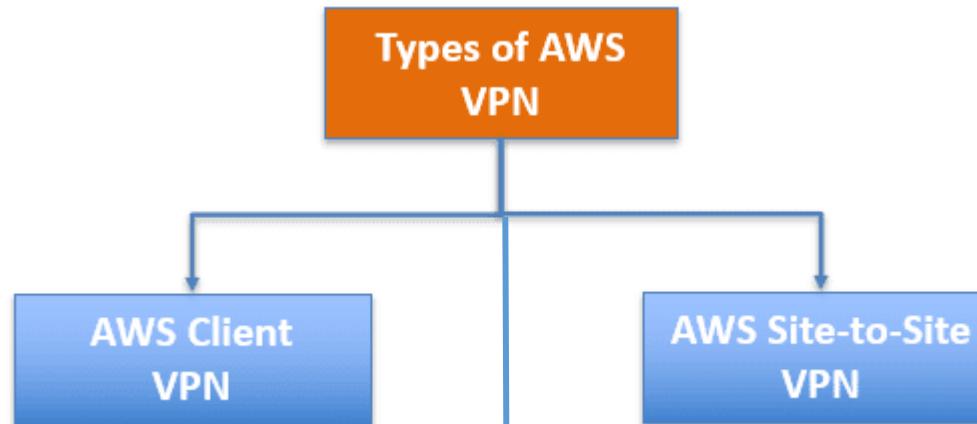
A **VPN connection** establishes a secure connection between you and the internet. Via the VPN, all your data traffic is routed through an encrypted virtual tunnel. This disguises your IP address when you use the internet, making its location invisible to everyone. A VPN connection is also secure against external attacks.



# Types of Virtual Private Network



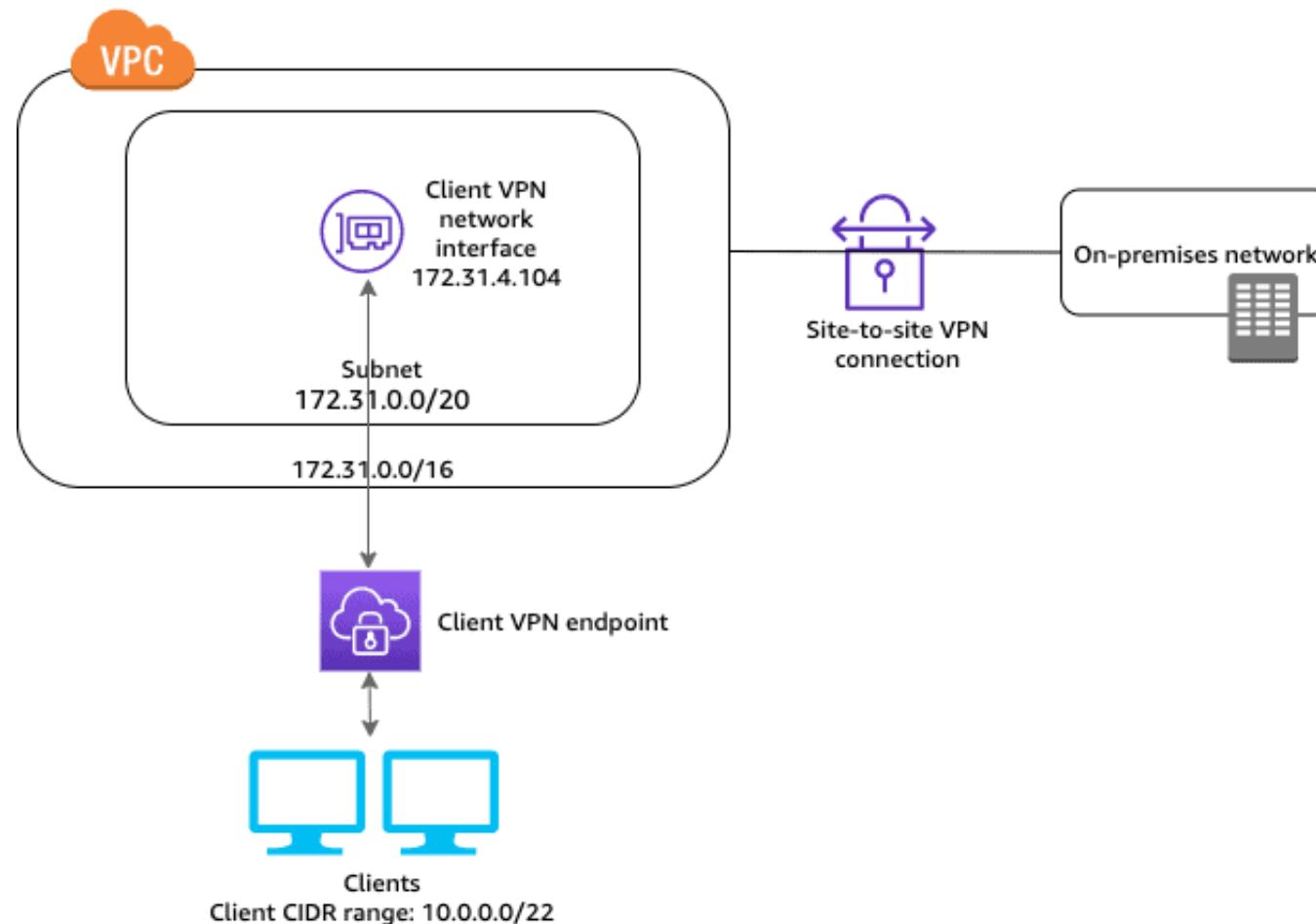
# Types of Virtual Private Network



- It is a fully managed remote access VPN solution that allows your distant employees to safely access resources on AWS as well as your on-premises network.
- Your users can access your applications in the same way before, during, and after the transfer to AWS.
- It is a fully managed service that uses IP Security (IPSec) tunnels to establish a secure link between your data centre or branch office and your AWS resources.
- You can connect to both your Amazon Virtual Private Clouds (VPC) and the AWS Transit Gateway when utilizing it, and two tunnels are used for each connection to increase redundancy.

# What is AWS VPN?

AWS Virtual Private Network (VPN) solutions connect your on-premises networks, distant offices, client devices, and the AWS global network in a secure manner. AWS Client VPN and AWS Site-to-Site VPN are the two services that make up this system. Each service offers a managed, scalable, and highly available cloud VPN solution to secure your network traffic.



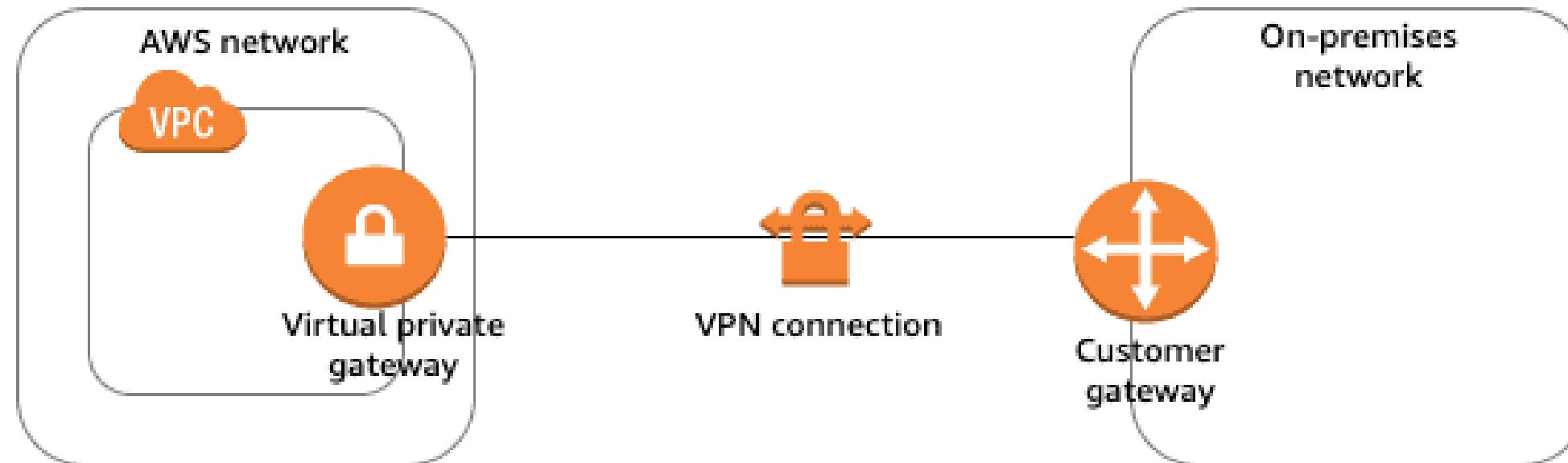
# Components of AWS VPN

## Virtual Private Gateway – VGW

- A virtual private gateway is the VPN concentrator on the AWS side of the VPN connection.

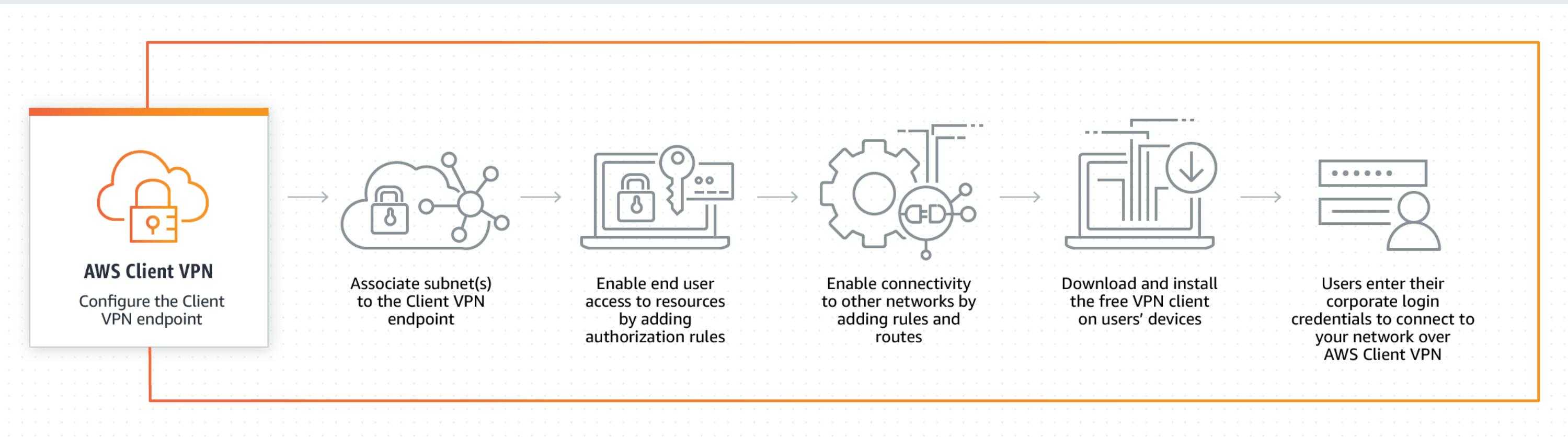
## Customer Gateway – CGW

- A customer gateway is a physical device or software application located on the customer side of the VPN connection.



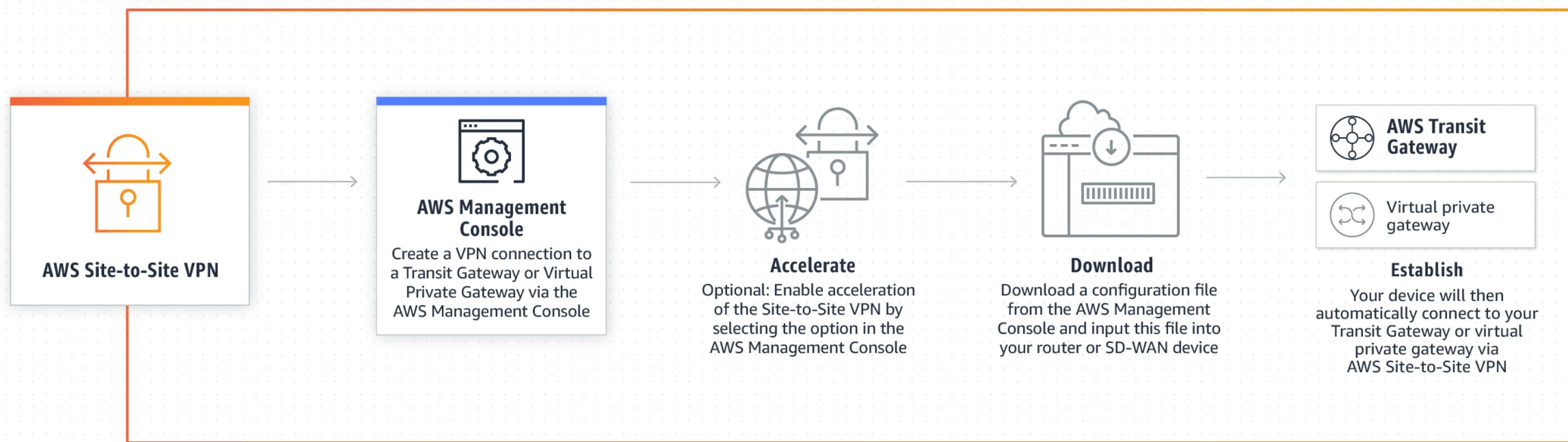
# How AWS Client VPN works?

- It automatically adjusts up or down dependent on demand because it is fully elastic.
- Your users can access your applications in the same way before, during, and after the transfer to AWS.
- The OpenVPN protocol is supported by AWS Client VPN, including the software client.

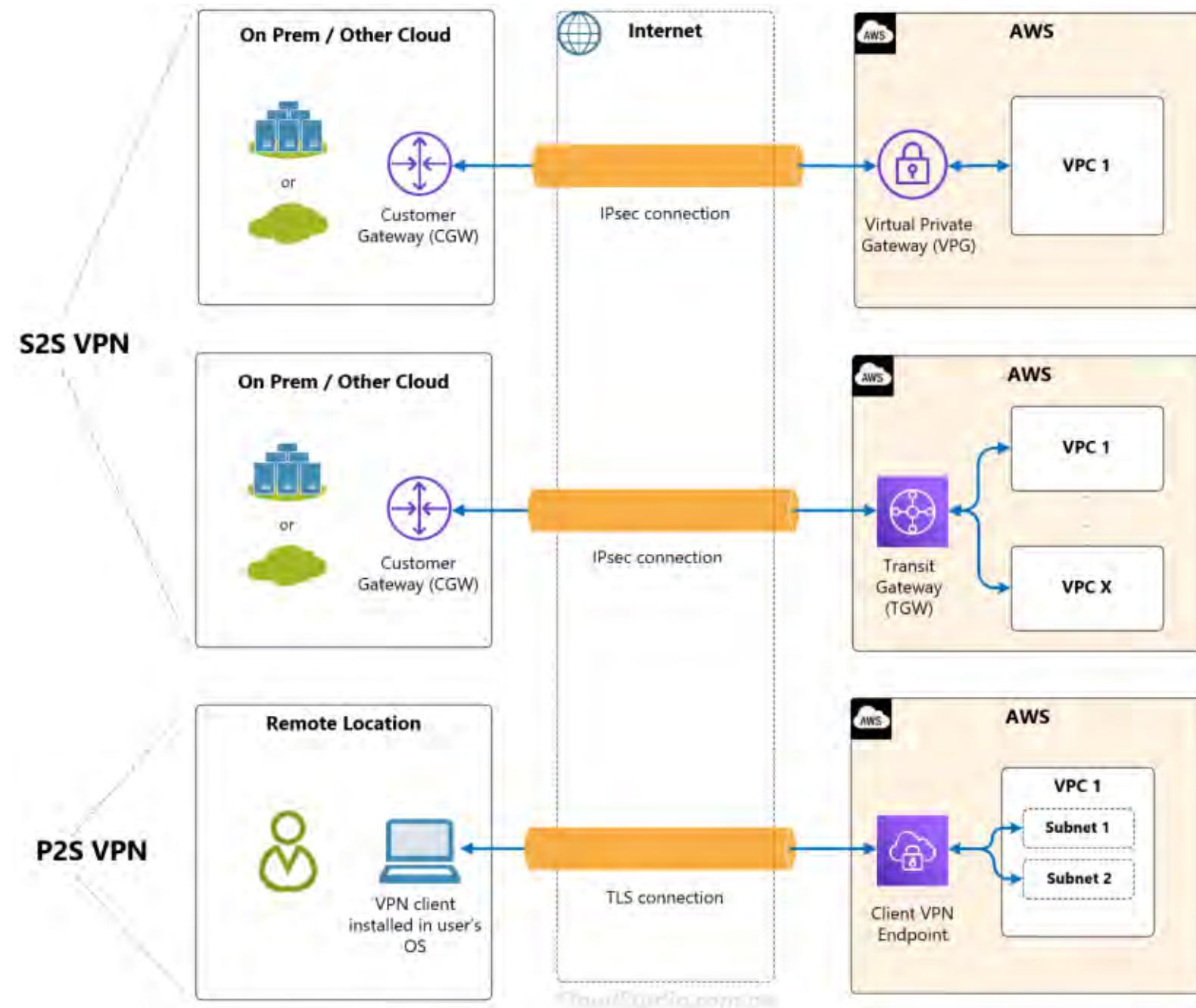


# How AWS Site-to-Site VPN works?

- The Accelerated Site-to-Site VPN option, which works with AWS Global Accelerator to dynamically route your traffic to the closest AWS network endpoint with the best speed, offers even better performance for internationally distributed applications.

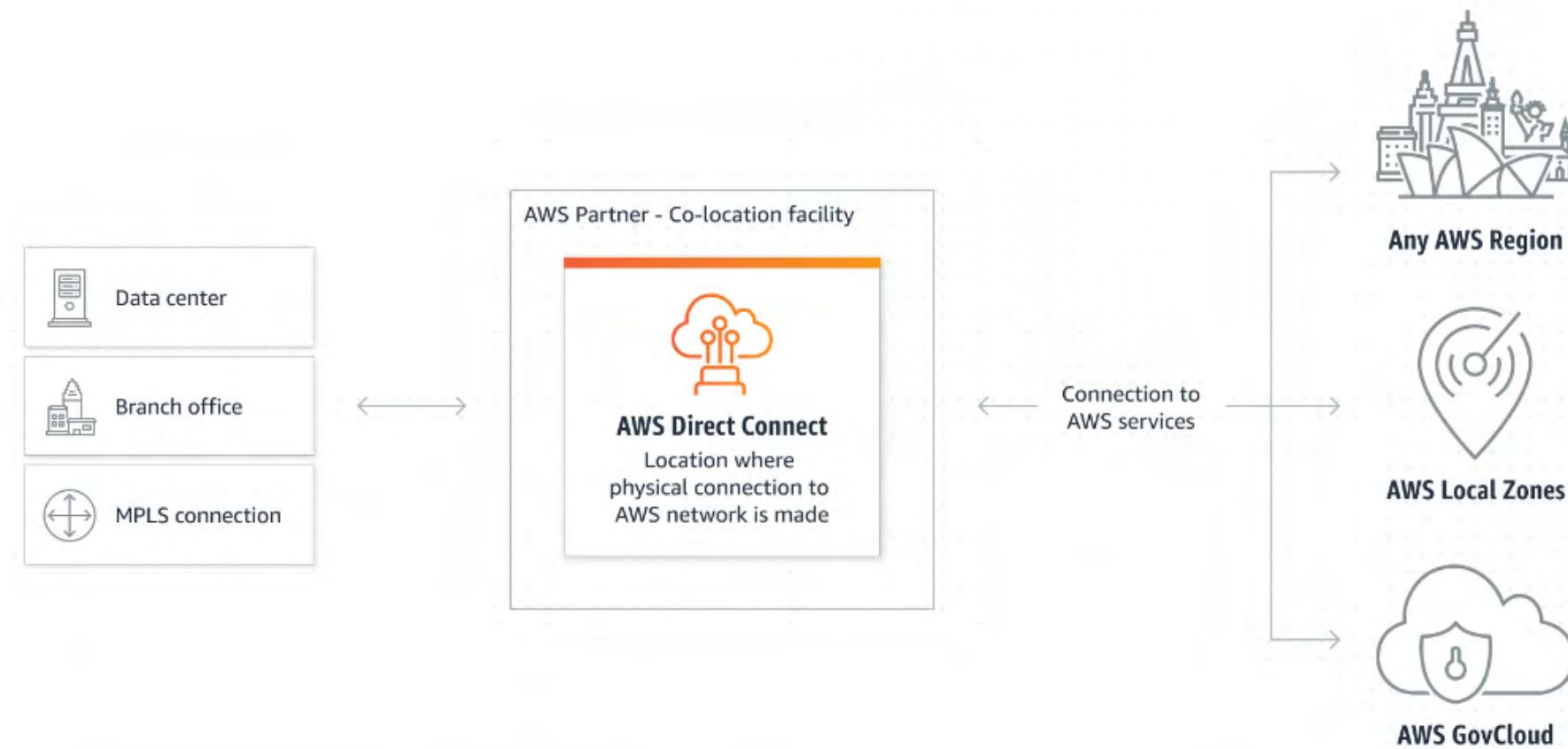


# Recall of all VPN options



# What is AWS Direct Connect?

AWS Direct Connect is a network service that provides dedicated secure network connections between your on-premises data center or office and AWS. You can use this service to connect to AWS resources in any region.



# Advantages of AWS Direct Connect

## Data Transfer

Move large amounts of data into and out of AWS with ease, reducing your network costs.

## Low Latency

Get faster, more consistent network performance and low latency connection to AWS.

## Hybrid Architecture

Integrate your existing IT infrastructure with AWS cloud services in a sturdy and secure environment.

## Secure

Connectivity is established over a private virtual interface, which is isolated from the internet.

## Reliable

Get consistent network performance with guaranteed bandwidths of up to 10 Gbps.

## Flexible

Choose from various options, such as dedicated ports and hosted connections, to meet your requirements.

## Scalable

Scale up your connectivity based on your business needs without requiring any redesign.

# Options for Direct Connect

---

There are two main options for establishing a Direct Connect connection:

- **Hosted connections:** You can use a hosted connection to connect to AWS resources over a Direct Connect connection provided by an AWS Direct Connect Partner. The partner provides the network infrastructure and assists with the connection setup and management.
- **Dedicated connections:** You can establish a dedicated connection between your network and AWS Direct Connect. This option gives you complete control over the connection and provides a private connection to AWS resources.

# How to Set Up AWS Direct Connect

- 1 Step 1: Determine your connectivity requirements**

Figure out which AWS services you need to access, the amount of traffic, and your network security requirements.
- 2 Step 2: Choose a Direct Connect location and Partner**

Select a Direct Connect location that meets your requirements and choose a Direct Connect Partner.
- 3 Step 3: Request a connection with AWS**

Make a request for a new Dedicated Connection or Hosted Connection and configure the Virtual Interface.
- 4 Step 4: Configure the physical connection**

Configure the physical connection from the Partner's network to your data center, office, or colocation environment.
- 5 Step 5: Monitor and test your connection**

Monitor your Dedicated Connection to ensure that it functions correctly and troubleshoot any issues that arise.

# Comparison of AWS VPN and Direct Connect

	AWS Site-to-Site VPN	AWS Direct Connect
Network	<p>Can reach 4 Gbps or less.</p> <p>Connected with shared and public networks, so the bandwidth and latency fluctuate.</p>	<p>Starts from 50 Mbps and expands to 100 Gbps.</p> <p>Network is not fluctuating and provides a consistent experience.</p>
Time to establish	<p>It is relatively easy to set up and faster to install than AWS Direct Connect.</p>	<p>Installation requires an experienced team, and setup is not as easy as AWS VPN.</p>
Pricing	<p>\$0.05 per connection hour. \$0.09 per GB of data transfer out(DTO).</p>	<p>\$0.02 to \$0.19 per GB of data transfer out(DTO). Port hour fees varies based on port speed.</p>
Security	<p>In AWS Site-to-Site VPN, the connection is encrypted via IPSec.</p>	<p>AWS Direct Connect does not encrypt your traffic in transit by default.</p>

# Use Cases for AWS Direct Connect

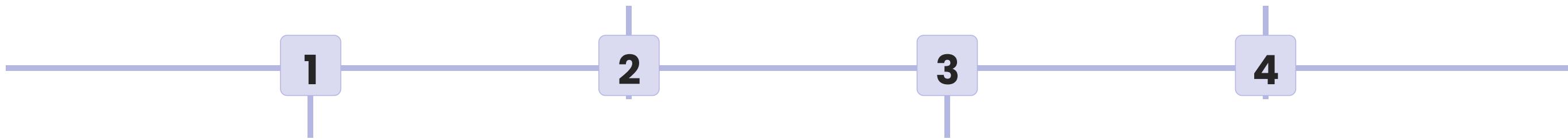
AWS Direct Connect can be used in various types of situations, such as:

## Big Data

Move large amounts of data into or out of AWS, with faster, more consistent network performance and improved security for your big data workloads.

## Media and Content

Streamline the delivery of video and other media to your audiences by using AWS Direct Connect to augment internet-based data transfer.



## Disaster Recovery

Easily establish a connection between your production environment and your DR environment hosted on AWS.

## Hybrid Cloud

Extend your existing data center infrastructure to the cloud seamlessly and securely.

iamneo



# AWS Route53

# AWS Route 53

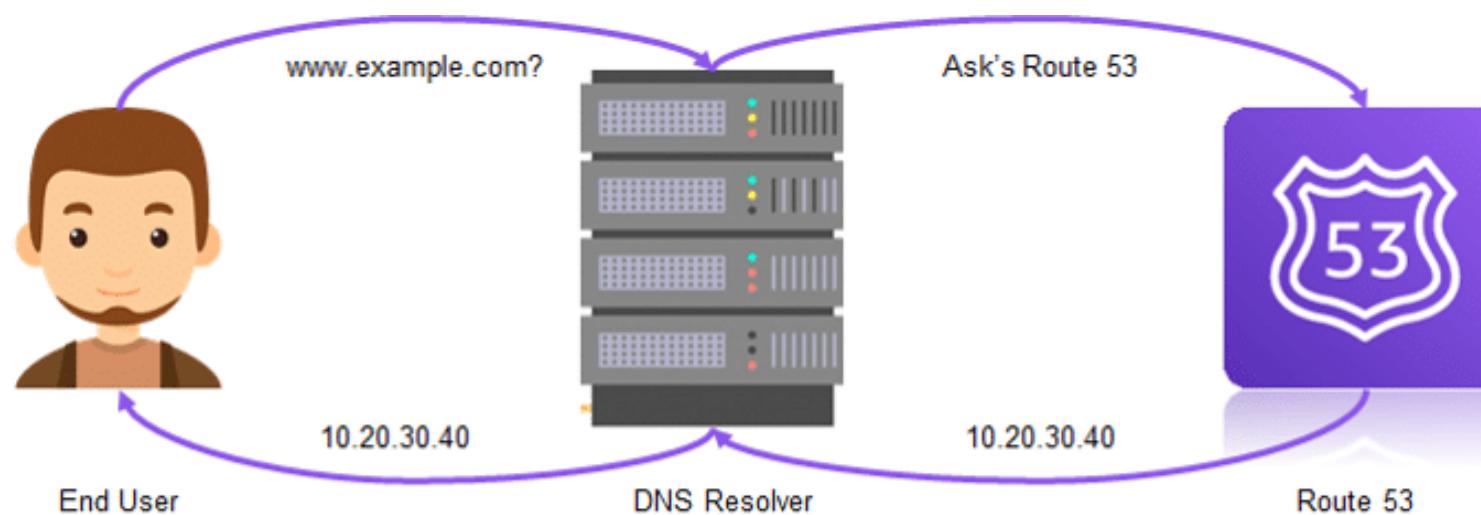
---



Learn all about AWS Route 53, the highly scalable and reliable Domain Name System (DNS) web service for routing internet traffic to your web apps.

# AWS Route 53

AWS Route 53 is a highly scalable Domain Name System (DNS) web service provided by Amazon Web Services. It is designed to give developers and businesses an extremely reliable and cost-effective way to route end users to Internet applications. With Route 53, you can register domain names and then route Internet traffic to the resources for your domain. It supports a variety of routing types, including failover, geolocation, weighted round-robin, latency-based routing, and more.



# Features of AWS Route 53



Amazon Route 53  
Benefits



High availability,  
reliability, and  
scalability



Security



Global network



Cost-effective

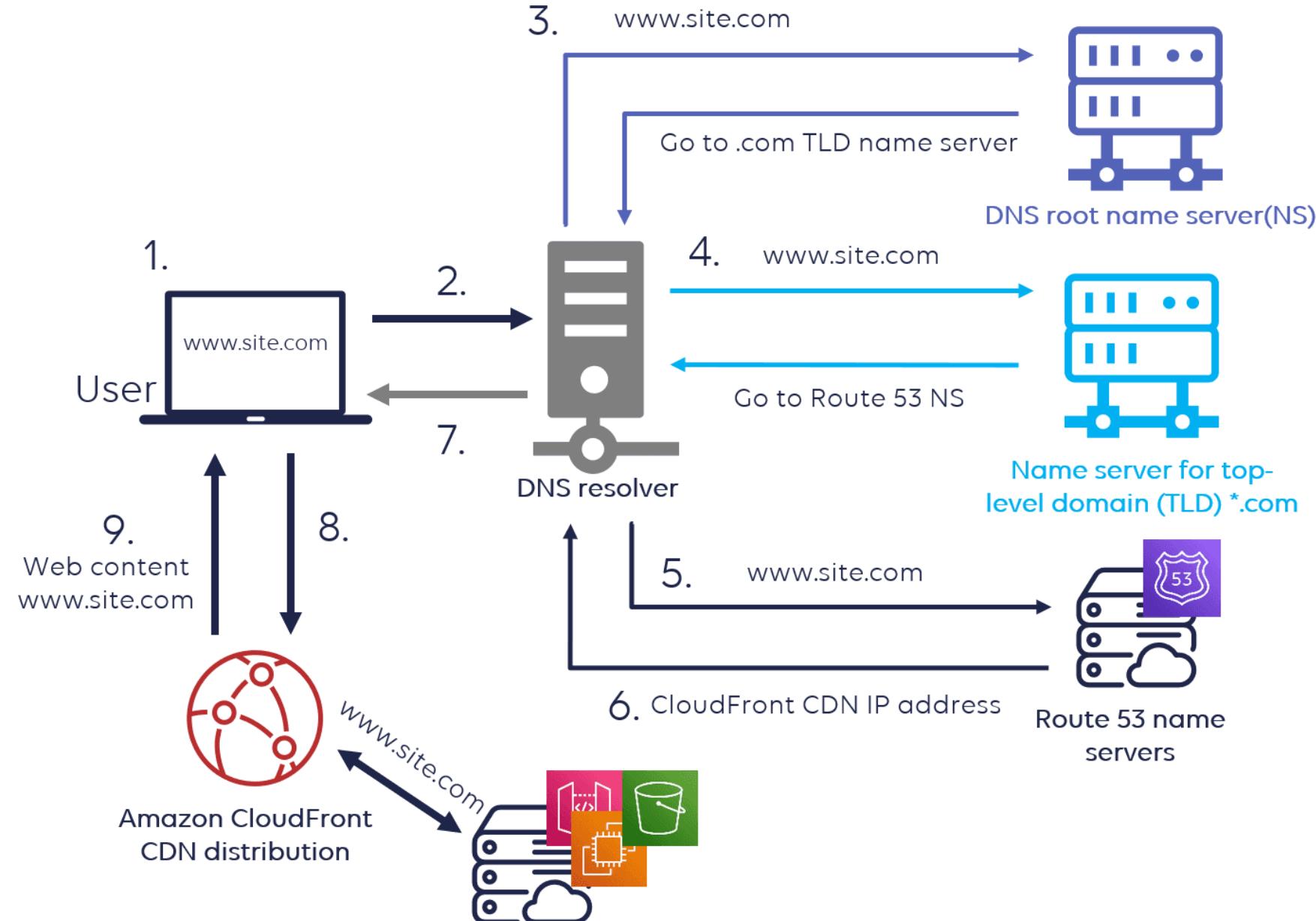


Integrated routing  
policies

# Benefits of AWS Route 53

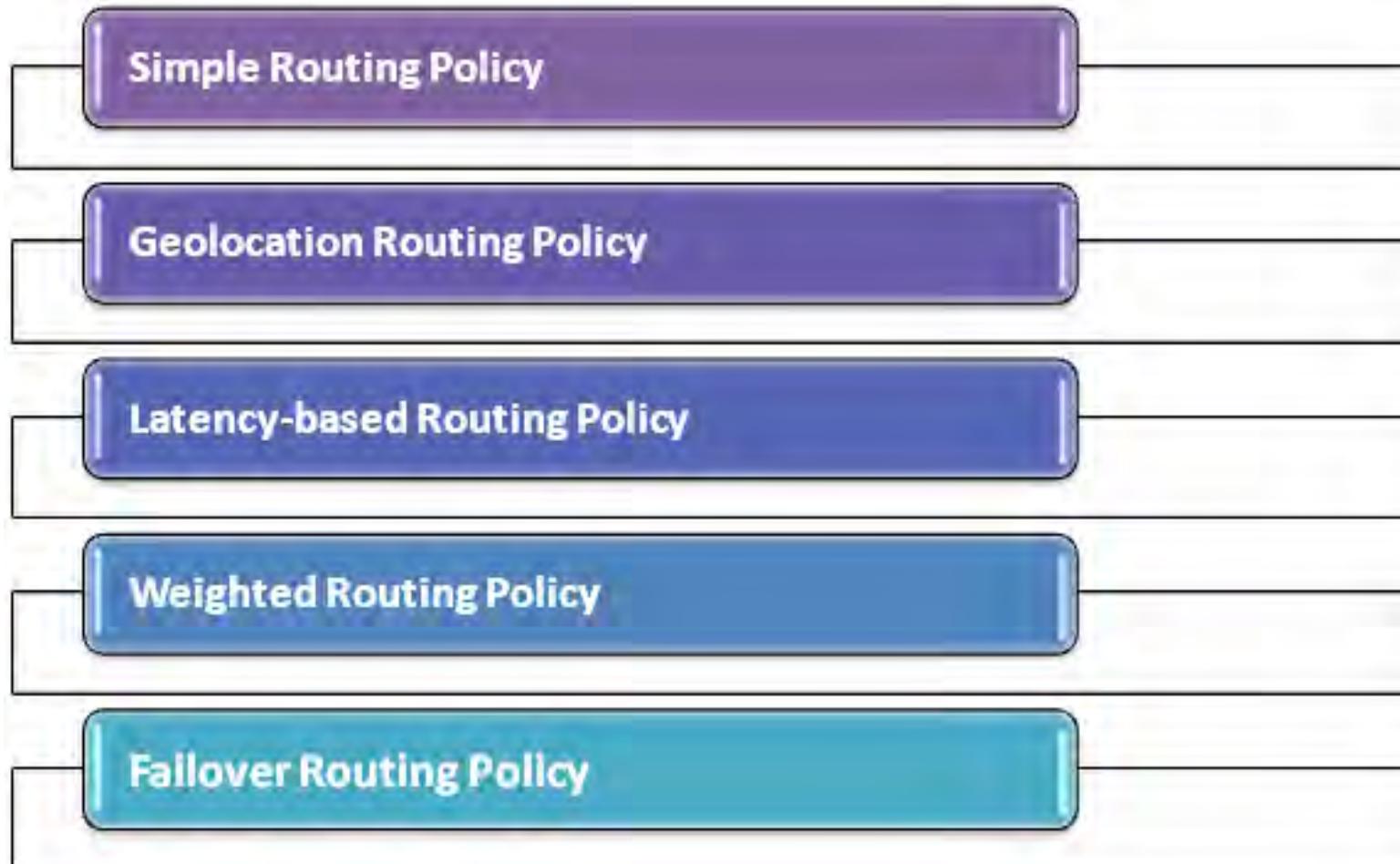


# How does AWS Route 53 work?



# Types of AWS Route 53 – routing policies

---



# Types of AWS Route 53 - routing policies

**Routing policy**

[Switch to quick create](#)

**Simple routing**  
Use if you want all of your clients to receive the same response(s).



**Weighted**  
Use when you have multiple resources that do the same job, and you want to specify the proportion of traffic that goes to each resource. For example: two or more EC2 instances.



**Geolocation**  
Use when you want to route traffic based on the location of your users.



**Latency**  
Use when you have resources in multiple AWS Regions and you want to route traffic to the Region that provides the best latency.



**Failover**  
Use to route traffic to a resource when the resource is healthy, or to a different resource when the first resource is unhealthy.



**Multivalue answer**  
Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random.



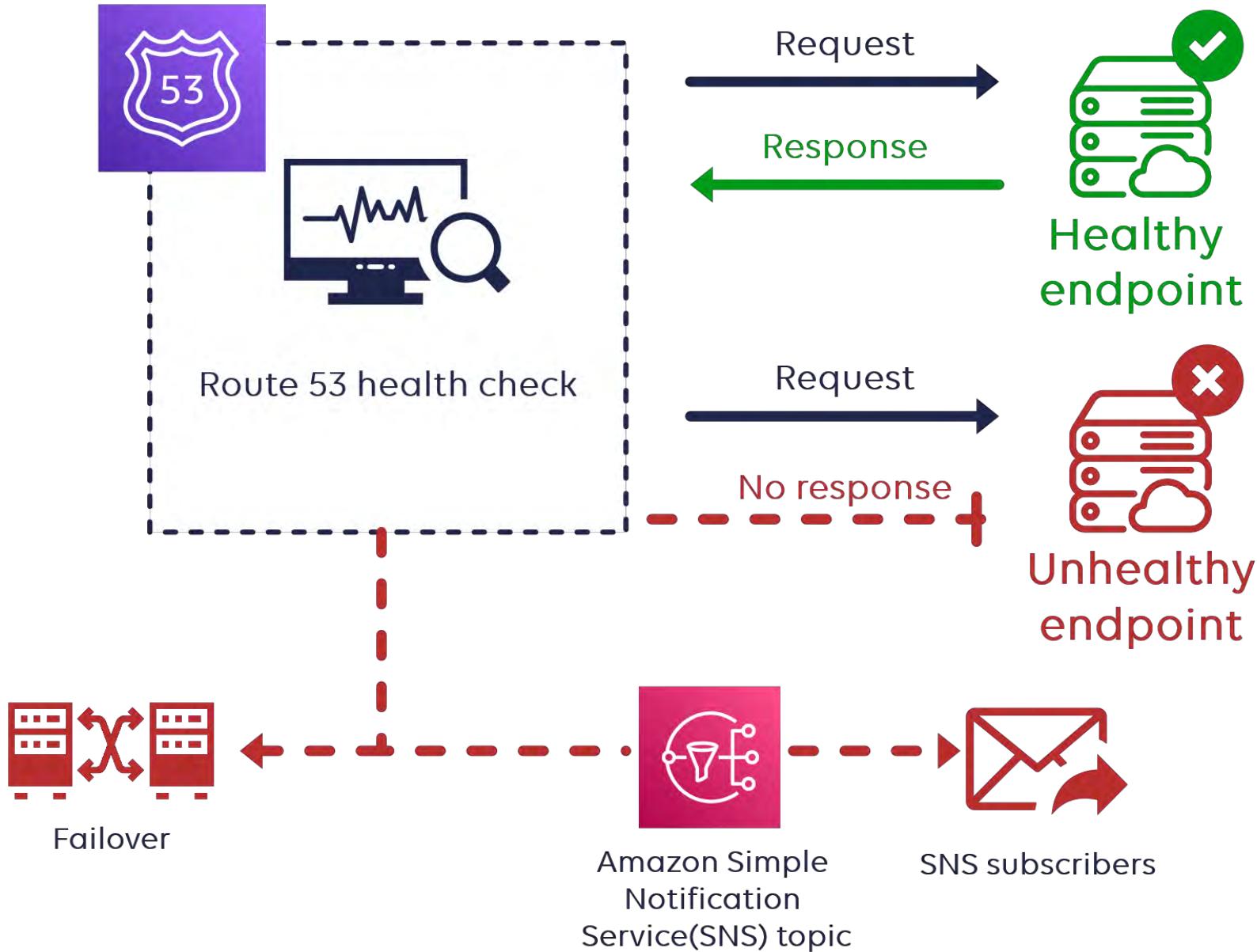
**IP-based**  
Use to route traffic to locations of IP address ranges in CIDR notation.



# What drives the popularity to AWS Route 53?



# Health Checks and Failover Routing



# Integrating with Other AWS Services

---



## AWS CloudFront

Route 53 integrates with Amazon's Content Delivery Network (CDN) service, CloudFront, to improve website speed and performance.



## AWS RDS

Easily route traffic to your Amazon Relational Database Service instances with Route 53 and take advantage of database scalability and reliability.



## AWS S3

Route 53 can route requests to your Amazon Simple Storage Service buckets, providing a global CDN for your website (when combined with CloudFront).

# Best Practices for AWS Route 53

---

## 1 Use geolocation routing

Take advantage of latency-based routing and route users to the endpoint nearest to them.

## 2 Use public hosted zones

Create public hosted zones for your internet-facing resources and private hosted zones for your internal resources.

## 3 Configure DNS failover

Minimize downtime by configuring DNS failover when your resources are unavailable. Monitor resource health with health checks.

iamneo

ANY  
Questions?



Thankyou

iamneo



# AWS CloudFront

---

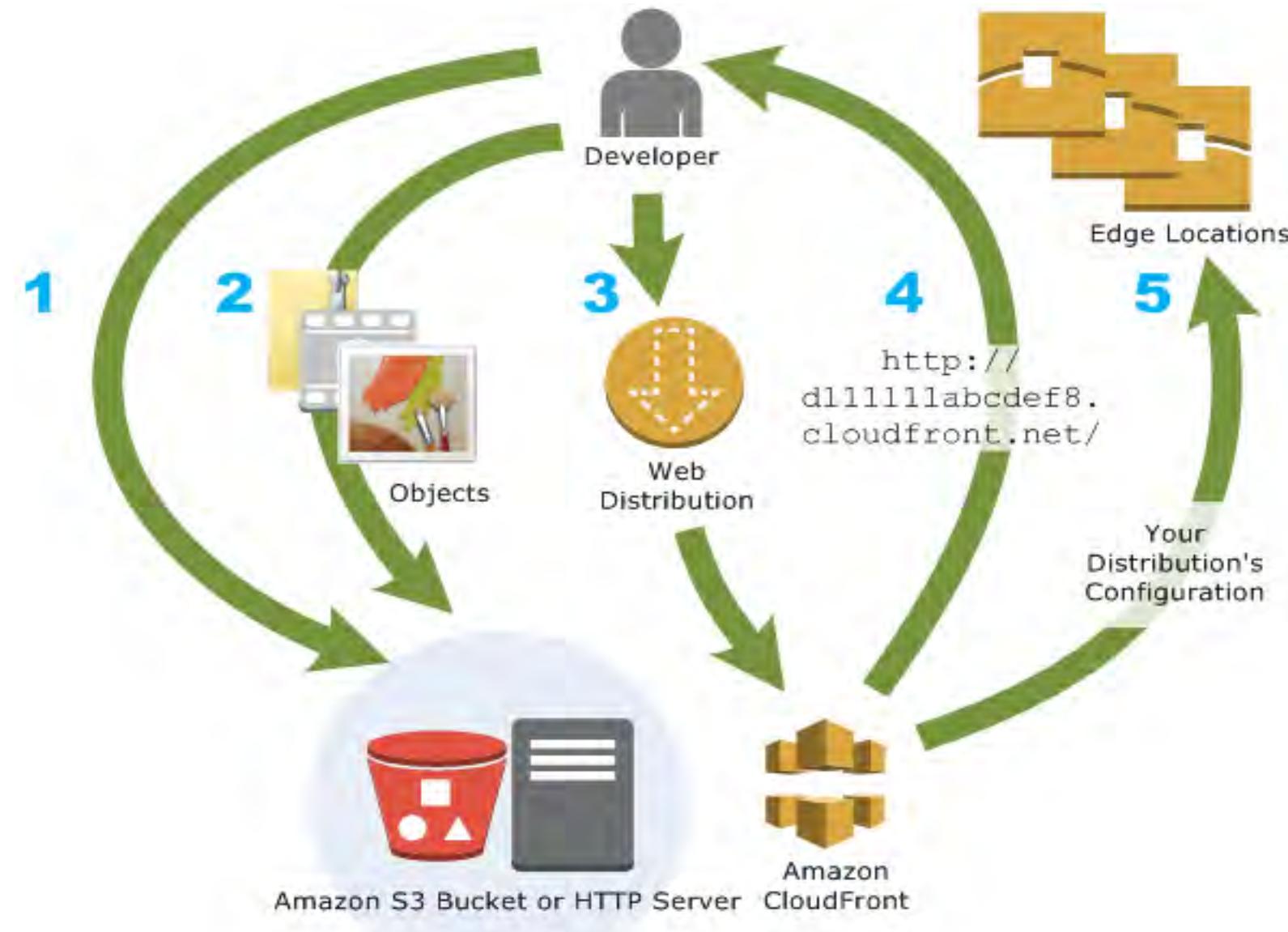
# Amazon CloudFront

---



Amazon CloudFront is a content delivery network that securely delivers data and files globally with low latency, high transfer speeds, and high availability.

# Amazon CloudFront



# Amazon CloudFront: A Global Content Delivery Network (CDN)

---

- ❑ Amazon CloudFront is a highly secure and scalable CDN that delivers data, videos, applications, and APIs to customers across the world with low latency and high transfer speeds.
- ❑ It operates by caching content at Edge locations, which are strategically placed around the world to ensure faster access and a better user experience.
- ❑ By using CloudFront, businesses can reduce load times, improve site performance, and save on bandwidth costs.
- ❑ CloudFront integrates with other Amazon Web Services (AWS) products, such as Amazon S3, Amazon EC2, and Elastic Load Balancing, to provide developers with an easy way to distribute content to end-users with a high degree of flexibility and control.

# Amazon CloudFront: A Global Content Delivery Network (CDN)

---

- It also offers a range of features, including SSL/TLS encryption, DDoS protection, and real-time logs and alerts, to ensure the security and availability of your content.
  
- Additionally, CloudFront supports a variety of content types, including static and dynamic content, and offers advanced customization options such as geo-restriction and signed URLs for secure content delivery.
  
- With no minimum usage commitments and a pay-as-you-go model, CloudFront is a cost-effective solution for businesses of all sizes.

# Benefits of using Amazon CloudFront

---

1

**Low Latency**

2

**Scalability**

3

**Security**

4

**Cost-Effective**

# Features of Amazon CloudFront

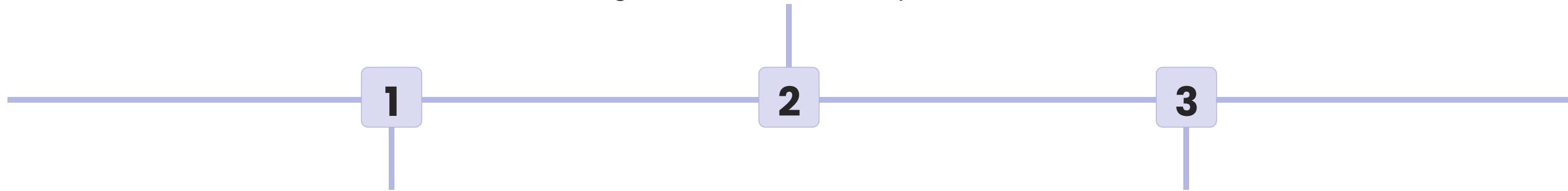
---

- **Distributed Architecture**
- **Origin Shield**
- **Customizable Behavior**
- **Real-time Logs**

# How to set up Amazon CloudFront

## Configure Settings

Configure various settings like security, cache control, and SSL certificate management to custom-fit your needs.



### Create a Distribution

Create a distribution by specifying the origin server and choosing the cache behavior for the content.

### Test and Deploy

Test your distribution and deploy it to your website, application, or software to start delivering content with Amazon CloudFront.

# Use cases of Amazon CloudFront



# **Content Delivery**



# Live Streaming



serverless

# Caching and Content Delivery Mechanisms in Amazon CloudFront

---

- ❑ Amazon CloudFront is a popular content delivery network that uses caching and content delivery mechanisms to improve the speed and efficiency of delivering content to users.
- ❑ Caching stores frequently accessed data in a location closer to the user, reducing the amount of time it takes to load the content.
- ❑ In Amazon CloudFront, you can choose from a variety of caching options, including edge caching, origin caching, and object caching.

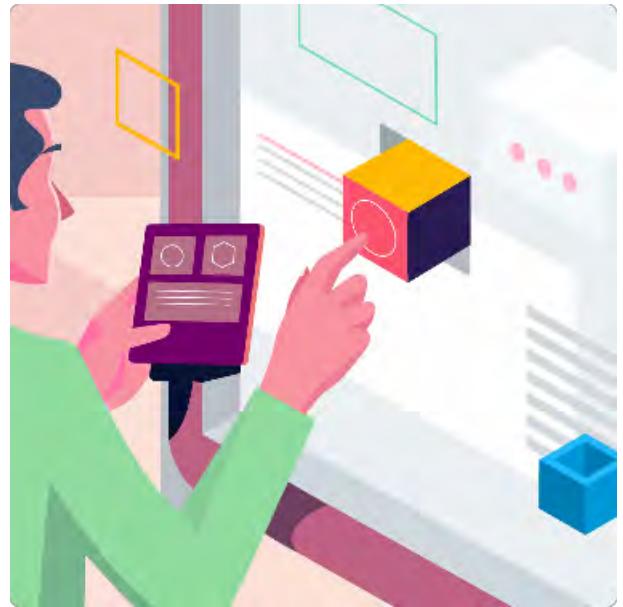
# Caching and Content Delivery Mechanisms in Amazon CloudFront

---

- ❑ Content delivery mechanisms distribute content across multiple servers, allowing for faster and more reliable access to the content.
- ❑ In Amazon CloudFront, your content is distributed to servers located all over the world, so users can access your content from the server that is closest to them.
- ❑ This helps to reduce latency and improve the overall performance of your website or application.

# Customizing Your Content Delivery

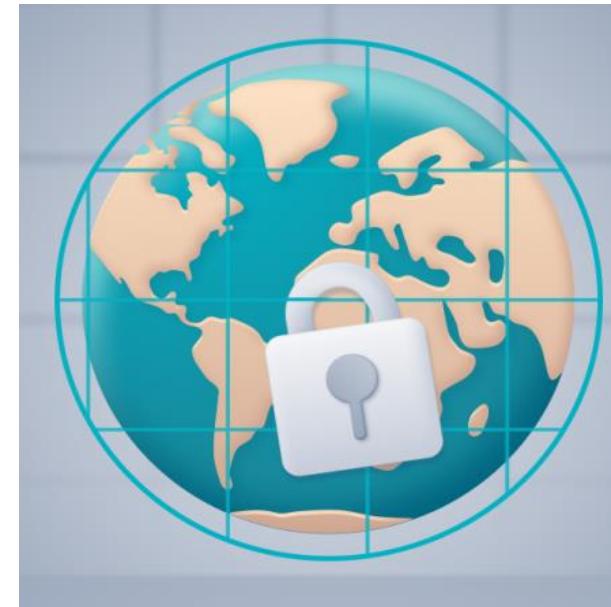
---



**Customization**



**Compression**



**Geo Restrictions**

# Maximizing Security and Protection

---

## Security

- CloudFront provides security at every level, from DDoS protection to network security, with advanced features such as AWS WAF, AWS Shield, and SSL/TLS security protocols.

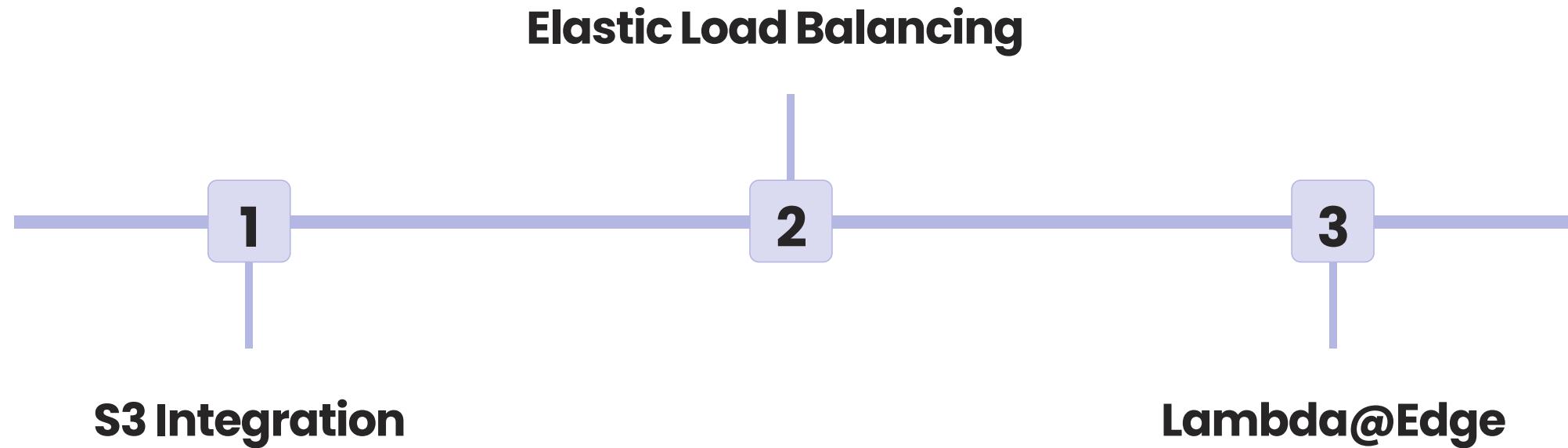
## Authentication

- CloudFront offers multiple tools to authenticate users and protect your content from unauthorized access, including signed cookies and URLs, token authentication, and CloudFront's origin access identity.

## Monitoring

- CloudFront comes with detailed logs and monitoring features to help you track your content delivery performance and troubleshoot issues.

# Integration with Other AWS Services



# Step-by-Step Guide to Creating a CloudFront Distribution

---

1. Create an Amazon S3 bucket or an HTTP(S) server to store your content.
2. Create a new CloudFront distribution.
3. Choose your origin server and configure your caching behavior.
4. Add alternate domain names and SSL certificates for secure content delivery.
5. Configure your distribution's security and privacy settings.
6. Create a CloudFront access identity (optional).
7. Create and configure CloudFront cache invalidation (optional).
8. Test your distribution settings and publish your CloudFront distribution.

# Configuring Origin Servers and Behaviors in Amazon CloudFront

---

- Amazon CloudFront allows you to configure origin servers and behaviors to control how content is delivered to your users.
- An origin server is the source of the content that CloudFront delivers to your users, and can be an Amazon S3 bucket, an Elastic Load Balancer, or a custom origin.
- To configure your origin servers and behaviors in Amazon CloudFront, you can create a distribution and specify the origin server and behaviors for that distribution.
- You can also create multiple origin servers and behaviors for a single distribution, allowing you to deliver different types of content from different sources.

# Configuring Origin Servers and Behaviors in Amazon CloudFront

---

- Behaviors control how CloudFront delivers your content, and can be used to specify caching settings, access restrictions, and other delivery options.
- You can create multiple behaviors for a single origin server, allowing you to customize the way your content is delivered based on its type or location.
- Overall, configuring origin servers and behaviors in Amazon CloudFront is a powerful way to optimize the delivery of your content and improve the user experience for your customers.

# Using Caching Options and Cache Invalidation

---

## 1 Caching Options

- CloudFront provides a variety of caching options, including dynamic, static, and streaming content delivery, with configurable TTLs and defaults.

## 2 Cache Invalidation

- CloudFront cache invalidation allows you to remove or update cached content in response to changes to your original content, ensuring that your end-users see the latest and most accurate content.

# Best Practices for Optimizing Content Delivery

---

## Content Optimization

- Optimize your content for CloudFront delivery by compressing images, minifying CSS and JavaScript, and reducing file sizes.
  - Optimize images using Amazon S3 and CloudFront integration.
  - Use CloudFront Lambda@Edge to compress, modify, or transform your content.

## Performance Tuning

- Tune your CloudFront distribution for maximum performance and minimal latency.
  - Choose your edge locations based on your customers' locations.
  - Optimize your caching behavior based on your content type and popularity.

# Best Practices for Optimizing Content Delivery

---

## Cost Optimization

- Reduce your CloudFront delivery costs without sacrificing performance.
  - Choose the right pricing model based on your usage pattern and content delivery needs.
  - Use CloudFront and S3 cost monitoring tools to optimize your costs and avoid unexpected charges.

# Comparison to other content delivery networks

---

## Cloudflare

Cloudflare offers free CDN plans, but they may compromise privacy by routing traffic through their servers, making it vulnerable to inspection.

## Azure CDN

Azure CDN integrates well with the Microsoft ecosystem, but its pricing model is complex, and configuring it can be challenging.

## Akamai

Akamai has a vast network of edge servers, but its prices can be high, and its documentation can be hard to navigate.

iamneo



# Amazon EC2



[www.iamneo.ai](http://www.iamneo.ai)

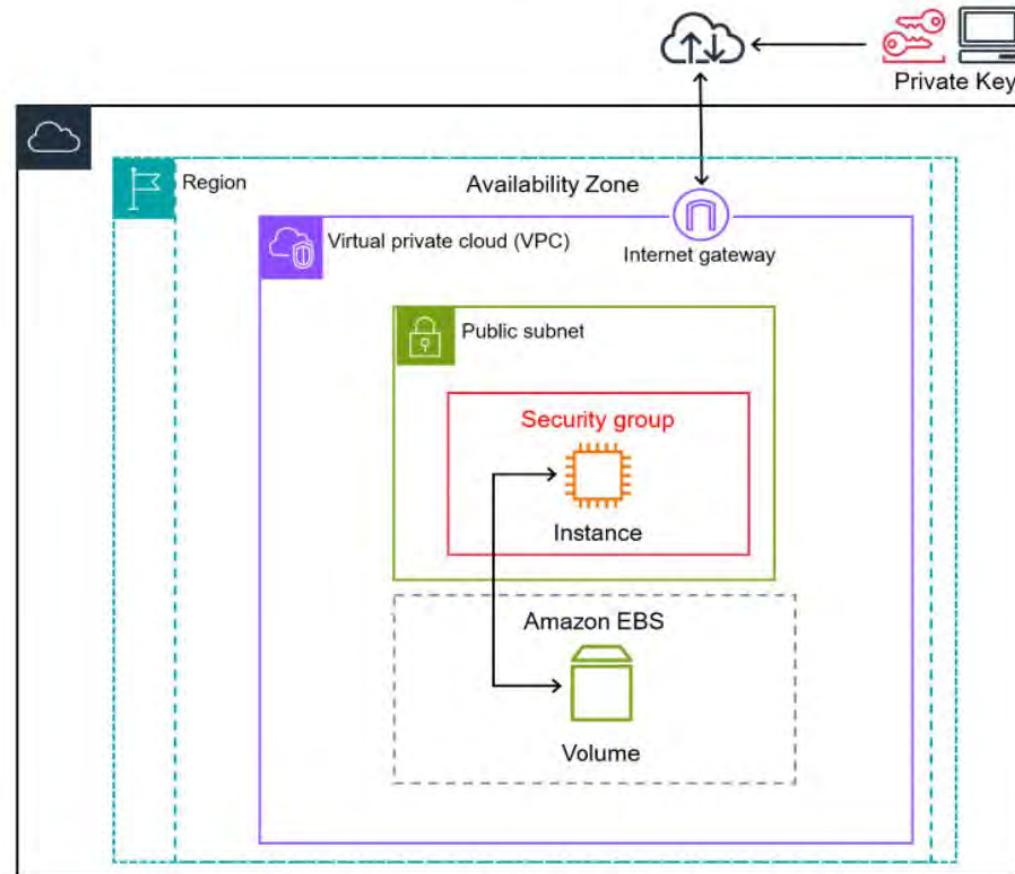
# Introduction to Amazon EC2



Amazon EC2

- Amazon Elastic Compute Cloud (Amazon EC2) is a web service that
- provides secure, resizable compute capacity in the cloud.
- Access reliable, scalable infrastructure on demand. Scale capacity within minutes with SLA commitment of 99.99% availability.  
Provide secure compute for your applications.

# Introduction to Amazon EC2



# Introduction to Amazon EC2

- 1 A computing powerhouse
- 2 Flexible and reliable
- 3 Affordable and cost-effective

# What can you do with Amazon EC2?

```
def db():
    if os.path.isfile(FILE_URL):
        db.create_all()

    if __name__ == "__main__":
        books = db.session.query(Book).all()
        render_template("index.html", books=books)

    @app.route("/edit", methods=["GET", "POST"])
    def edit(book_id):
        book_to_update = Book.query.get(book_id)
        book_to_update.rating = request.form["rating"]
        db.session.commit()
        return redirect(url_for("index"))

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

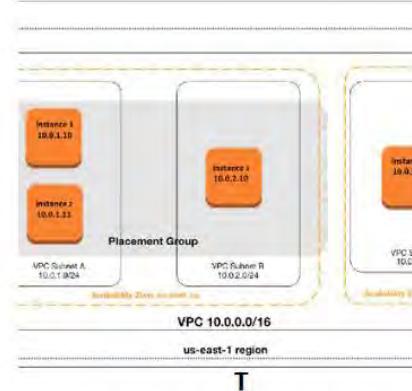
**Application Development**

Use EC2 as an environment to develop, test, and deploy applications, from simple web apps to complex enterprise solutions.



**Big Data**

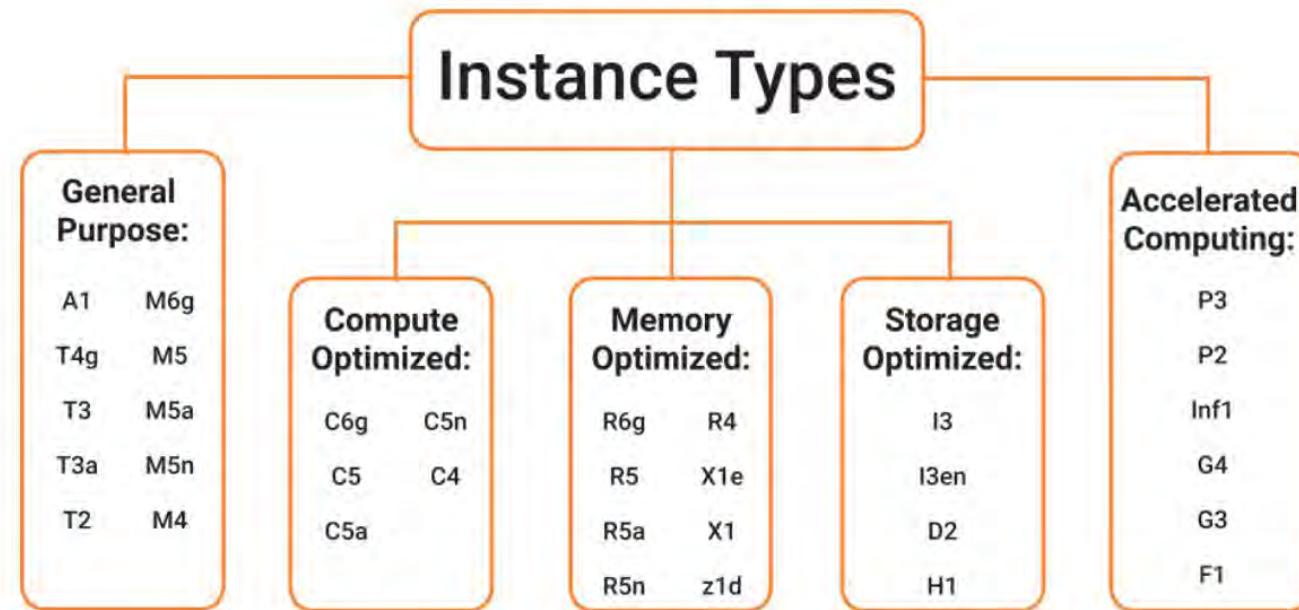
Run big data applications and workloads, including Hadoop and Spark, on EC2's powerful clusters of instances.



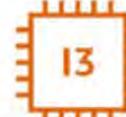
**Team Collaboration**

EC2 can be used for team collaboration and project management. Share instances and allow various group-level permissions to drive collaboration and productivity.

# Instance Types



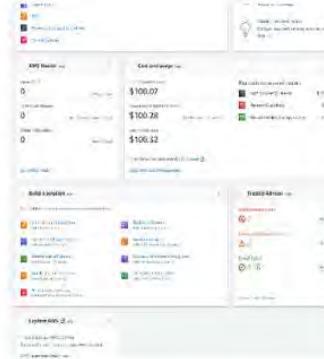
# Instance Types

General Purpose	Compute Optimised	Memory Optimised	Accelerated Computing	Storage Optimised
 ARM based core and custom silicon	 Compute - CPU intensive apps and DBs	 RAM - Memory intensive apps and DB's	 Processing optimised - Machine Learning	 High Disk Throughput - Big data clusters
 Tiny - Web servers and small DBs		 Xtreme RAM - For SAP/Spark	 Graphics Intensive - Video and streaming	 IOPS - NoSQL DBs
 Main - App servers and general purpose		 High Compute and High Memory - Gaming	 Field Programmable - Hardware acceleration	 Dense Storage - Data Warehousing

# Launching and Configuring EC2 Instances



# Introduction to AWS Management Console



## Sign Up or Log In

Create and log in to your AWS Management Console account. Follow the instructions to access the console.

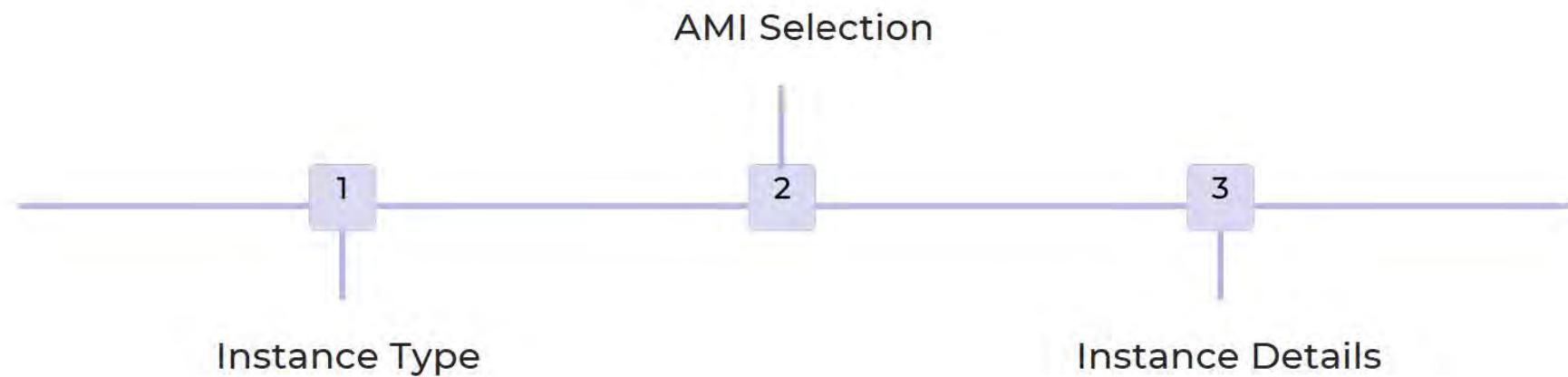
## AWS Dashboard

The dashboard is the home page of your AWS Management Console. Explore the various services AWS offers.

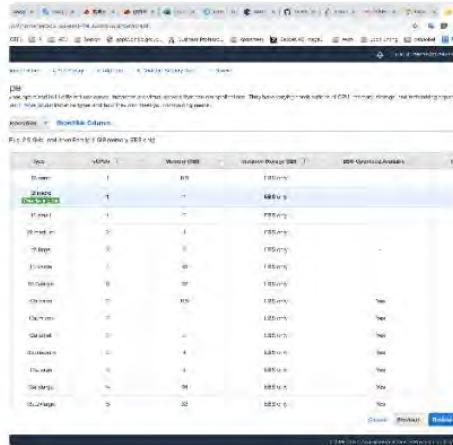
## Configuring Settings

Quickly access, configure, and customize your settings and preferences according to your needs.

# Launching an EC2 Instance



# Configuring Instance Details



## Instance Name

Name your instance and specify the purpose so you can easily identify it later on.

CPU	Architecture	Memory (MB)	Storage (GB)	Storage type	Network performance
I386, 808_64		64MB	-	-	Very Low
I386, 808_64		128	-	-	Low to Moderate
I386, 808_64		192	-	-	Low to Moderate
I386, 808_64		256	-	-	Low to Moderate
I386, 808_64		384	-	-	Low to Moderate
I386, 808_64		512	-	-	Low to Moderate
I386, 808_64		640	-	-	Low to Moderate
I386, 808_64		768	-	-	Low to Moderate
I386, 808_64		960	-	-	Low to Moderate
X86_64		128	-	-	Low to Moderate
X86_64		192	-	-	Low to Moderate
X86_64		256	-	-	Medium
X86_64		384	-	-	Medium
X86_64		512	-	-	Up to 2 Gbps
X86_64		640	-	-	Up to 5 Gbps
X86_64		768	-	-	Up to 8 Gbps

## Instance Type

Specify the details for the instance type.

# Configuring Instance Details

Port range	Protocol	Source
22	TCP	[REDACTED]/1
443	TCP	[REDACTED]/1
80	TCP	[REDACTED]/1
0 - 65535	TCP	0.0.0.0/0

Port range	Protocol	Destination
All	All	0.0.0.0/0

## Security Groups

Choose the security groups you want your instance to be associated with.



## Key Pairs

Choose the key pair to log in to your instance.

# Choosing Storage Options

## EBS Volume

Elastic Block Store (EBS) lets you store data separately from the instance. It can be easily attached or detached to the instance.

## Instance Store Volume

Instance Store Volumes work similarly to EBS, but the data is tied to the instance's lifecycle.

## Snapshot

Creating a snapshot ensures that you have a backup of your instance. Snapshots can be created on demand or scheduled.

# Setting Up Security Groups

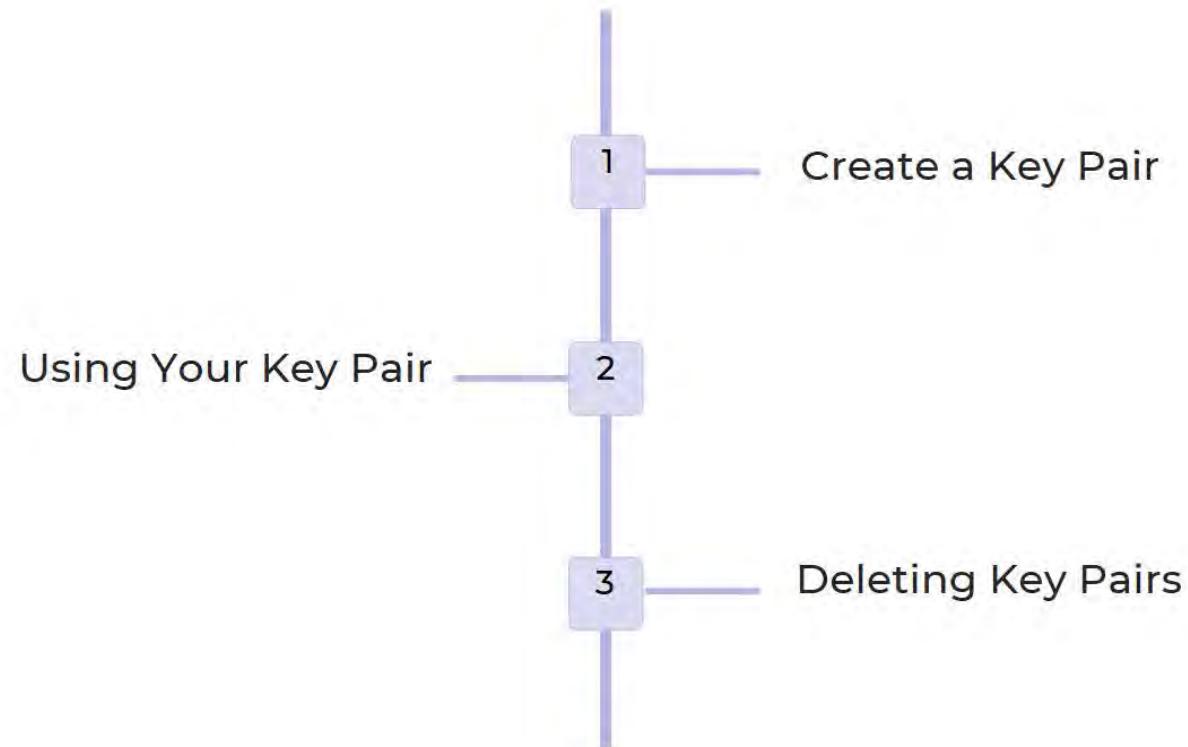
## Inbound Rules

- Control access to the instance
- Specify ports and protocols
- Allow unrestricted access

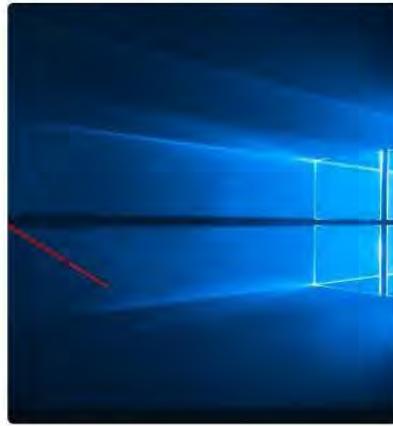
## Outbound Rules

- Control instance access to the internet
- Specify ports and protocols
- Allow unrestricted access

# Creating and Using Key Pairs



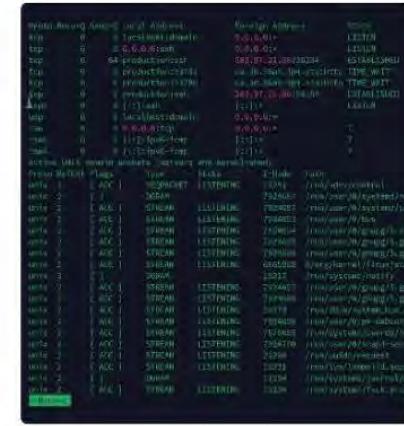
# Connecting to Instances using SSH or RDP



Windows -RDP



Mac -SSH



Linux -SSH

# Understanding EC2 Instance States and Lifecycle

## Running Instances

EC2 instances launched and running in a specific Availability Zone.

## Stopped Instances

EC2 instances that have been stopped and can be re-launched when needed.

## Terminated Instances

EC2 instances that have been terminated, and their data cannot be recovered.

# Managing EC2 Instances using AWS Management Console

Launch an Instance

Reboot and Terminate

Scaling and Load Balancing

Monitoring and Troubleshooting

# Managing EC2 Instances using CLI



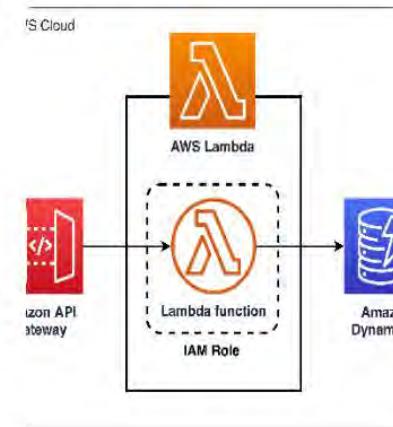
```
aws ec2 describe-instances --region us-east-1
{
    "Reservations": [
        {
            "Instances": [
                {
                    "InstanceId": "i-000000000000000000",
                    "ImageId": "ami-000000000000000000",
                    "InstanceType": "t2.micro",
                    "State": {
                        "Name": "running"
                    },
                    "PublicIpAddress": "54.122.122.122",
                    "PrivateIpAddress": "10.0.3.10",
                    "NetworkInterfaces": [
                        {
                            "Association": {
                                "PublicIp": "54.122.122.122",
                                "PrivateIp": "10.0.3.10"
                            }
                        }
                    ]
                }
            ]
        }
    ]
}
```

Command Line Interface



```
#!/bin/bash
# Script to manage EC2 instances
# Usage: ./ec2_manager.sh [option] [args]
# Options:
#   -l: List all EC2 instances
#   -r: Run a command on all instances
#   -s: Stop all instances
#   -u: Start all instances
#   -d: Delete all instances
#   -h: Help
# Example usage: ./ec2_manager.sh -l
```

Automation using Shell Scripts



AWS Lambda Functions

# Managing EC2 Instances using SDKs

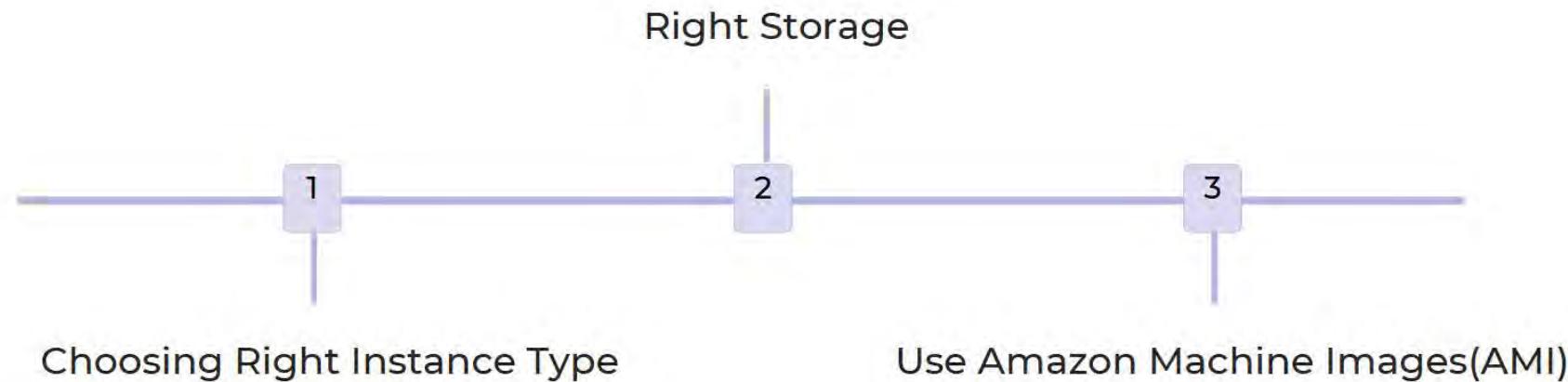
## Amazon SDKs

Use AWS SDKs to manage EC2 instances programmatically from your preferred programming languages such as Node.js, Java, Python, etc.

## AWS CloudFormation

Create your instances along with all required dependencies, security, network, and storage using AWS CloudformationStacks!

# Best Practices for Optimizing EC2 Instances for Cost



# Best Practices for Optimizing EC2 Instances for Cost

## Use AWS Compute Optimizer

Utilize available AWS Compute Optimizer tools to efficiently choose the most cost-effective instance type for your workload

## Reserved Instances

Save you up to 75% on EC2 instances and provide capacity reservation when you need it most.

## Spot Instances

Utilize the unused capacity of Amazon EC2 instances at highly reduced prices and perform cost-efficient batch processing or run other workloads with flexible start and end times.

iamneo



# AWS Load Balancer

---



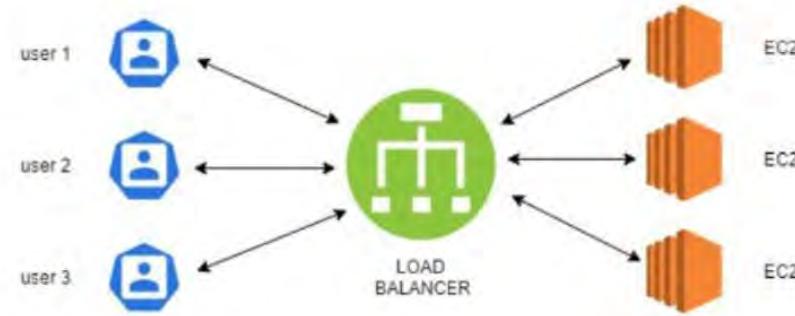
[www.iamneo.ai](http://www.iamneo.ai)

# AWS Load Balancer



Amazon Web Services (AWS) Load Balancer distributes traffic to multiple targets such as EC2 instances or containers, which increases the availability of your applications. In this presentation, we will explore different types of Load Balancers and how to create and configure them.

# Load Balancing



- Single point of access (DNS) to your application. Provide
- SSL termination (HTTPS) for your websites.
- Separate public traffic from private traffic.
- It is integrated with many AWS offerings / services.
- You can setup internal (private) or external (public) ELBs.
- AWS guarantees that it will be working AWS takes care of upgrades, maintenance, high availability.

# Types of AWS Load Balancers

Application Load Balancer

Network Load Balancer

Classic LoadBalancer

Gateway Load Balancer

# Types of AWS Load Balancers

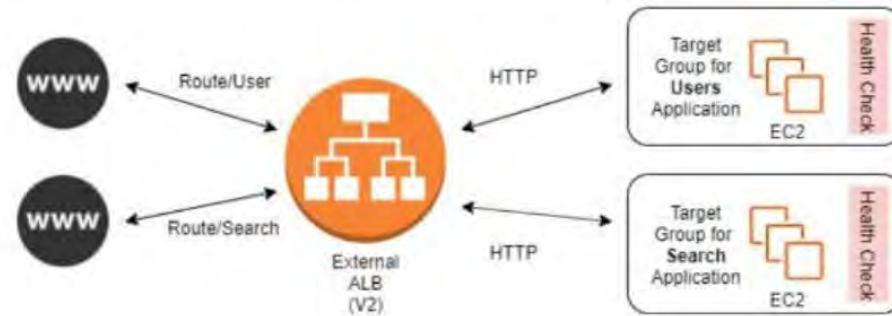
Application Load Balancer

Network Load Balancer

Classic LoadBalancer

Gateway Load Balancer

# Application Load Balancer (V2)



Application load balancers (Layer 7) allow you to do:

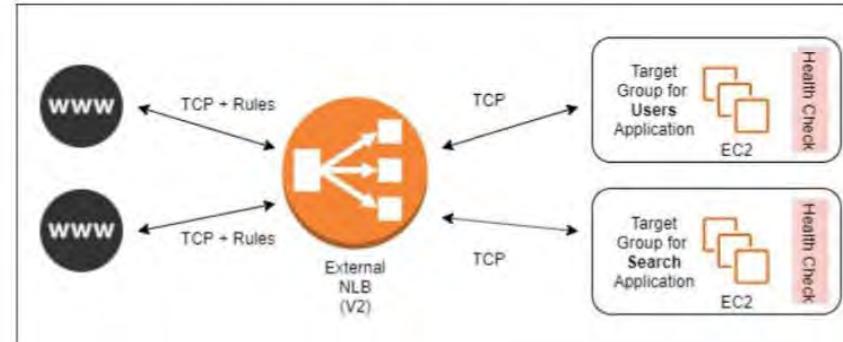
- Load balancing for multiple HTTP applications across machines (target groups).
- Load balancing for multiple applications running on the same machine(ex: containers).
- Load balancing based on route (path) in URL.
- Load balancing based on Hostname in URL.

# Application Load Balancer (V2)



- ALB is perfect for micro services & container-based applications.
- Port mapping feature to redirect to a dynamic port.
- Previously we used to create one Classic Load Balancer per application.
- That was very expensive and inefficient!
- Stickiness can be enabled at the target group level.
- ALB support HTTP/HTTPS & Web sockets protocols.
- The application servers don't see the IP of the client directly. Instead, it is inserted in the header X-Forwarded-For.

# Network Load Balancer (V2)



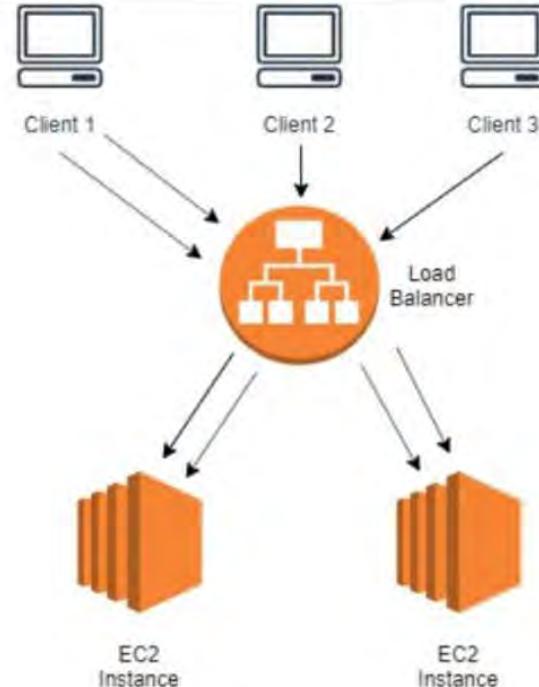
- Network load balancers (Layer 4) allow you to :
  - Forward TCP traffic to your instances
  - Handle millions of request per seconds
  - Support for static IP or elastic IP
  - Support Cross Zone Balancing.
- Network Load Balancers are mostly used for extreme performance and should not be the default load balancer.

# Important Notes:

---

- CLB, ALB & NLB support SSL certificates and provide SSL termination.
- All Load Balancers have health check capability.
- ALB is a great fit with ECS (Docker).
- All Load Balancers in AWS has a static host name.
- Do not change and use underlying IP.
- Network Load Balancer can directly see the client IP.
- 4xx errors are client induced errors.
- 5xx errors are application induced errors.
- 503 means at capacity or no registered target
- If the Load Balancer can't connect to your application, check your security groups.

# Load Balancer Stickiness



- Stickiness is nothing but the same client is always redirected to the same instance behind a load balancer.
- This works for Classic Load Balancers & Application Load Balancers.
- “Cookies” are used for stickiness and those cookies has an Expiry date which you can control.
- Enabling stickiness may bring imbalance to the load over the backend EC2 instances as it will stick to one instance and send traffic only to that instance for certain amount of time

# Steps for Creating and Configuring Load Balancers using AWS

Follow these steps to create and configure a load balancer in AWS:

1. Choose the appropriate type of load balancer for your use case, such as Application Load Balancer or Network Load Balancer.
2. Create a target group for your load balancer that specifies the resources you want to distribute traffic to, such as EC2 instances or containers.
3. Configure your load balancer listeners to specify the protocols and ports to use for incoming traffic.
4. Configure your load balancer routing to specify the rules for distributing traffic among your resources.
5. Set up health checks to ensure that your load balancer is sending traffic only to healthy resources.
6. Configure Auto Scaling to automatically adjust the number of resources in your target group based on traffic demand.
7. Manage your load balancer security groups to ensure that only authorized traffic is allowed.

# Configuring Listeners, Target Groups, and Routing Rules

Listeners, target groups, and routing rules are the building blocks of an AWS load balancer.

## Listeners

A listener is a process that checks for connection requests. Add a listener in the AWS Management Console by selecting your load balancer, going to the Listeners tab, and configuring the protocol and port.

## Target Groups

A target group is a group of resources that the load balancer distributes traffic to. Create a target group in the AWS Management Console by navigating to Target Groups, clicking Create target group, and specifying the target type, protocol, port, and VPC.

## Routing Rules

Routing rules determine how the load balancer distributes traffic among your target groups. Configure routing rules by selecting your listener, adding rules under the Rules tab, and setting conditions and actions to route traffic to specific target groups.

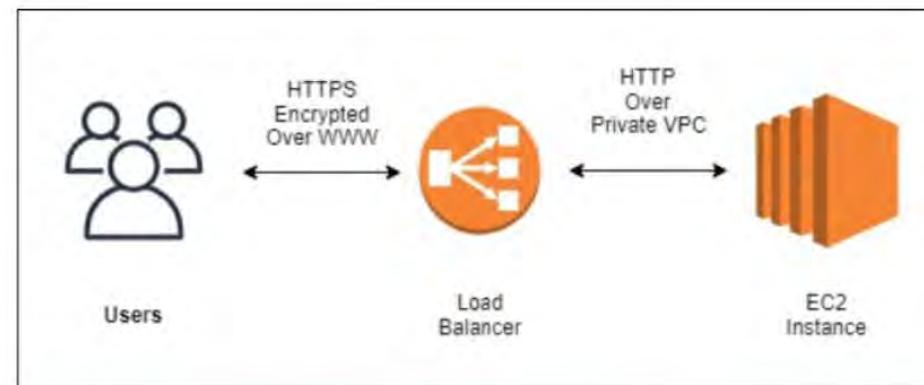
# Enabling Health Checks for Automatic Instance Failover

Health checks are a crucial component of an AWS load balancer, as they allow the load balancer to detect when a target is unhealthy and automatically failover to a healthy target. Here's how to enable health checks for your load balancer:

1. Create a target group for your load balancer that specifies the resources you want to distribute traffic to, such as EC2 instances or containers.
2. Configure health checks for the target group to ensure that the load balancer sends traffic only to healthy resources. Health checks can be configured to check the status of the instance, the status of the application running on the instance, or both.
3. Configure your load balancer listeners to specify the target group that the listener should forward traffic to.
4. Configure your load balancer routing rules to specify the actions that the load balancer should take for requests that match the conditions.

# Load Balancers SSL Certificates

- The load balancer uses an X.509 certificate (SSL/TLS server certificate)
- You can create/upload your own certificates alternatively
- HTTPS listener:
  - You must specify a default certificate
  - You can add an optional list of certs to support multiple domains
  - Clients can use SNI (Server Name Indication) to specify the hostname they reach



# Configuring Cross-Zone Load Balancing and Connection Draining

Cross-zone load balancing and connection draining are advanced features of an AWS load balancer that can improve the performance and reliability of your infrastructure. Here's how to configure them:

## Cross-Zone Load Balancing

To enable cross-zone load balancing for your load balancer:

1. Open the Amazon EC2 console and navigate to the load balancer that you want to configure.
2. Select the "Attributes" tab and click "Edit".
3. Set "Cross-Zone Load Balancing" to "Enabled".
4. Click "Save" to save your changes.

# Configuring Cross-Zone Load Balancing and Connection Draining

## Connection Draining

Connection draining is a mechanism that allows the load balancer to complete in-flight requests before terminating a target that has become unhealthy. To enable connection draining for your load balancer:

1. Open the Amazon EC2 console and navigate to the load balancer that you want to configure.
2. Select the "Attributes" tab and click "Edit".
3. Set "Connection Draining" to "Enabled".
4. Specify the amount of time, in seconds, that the load balancer should wait before terminating an unhealthy target.
5. Click "Save" to save your changes.

## Integrating with Auto Scaling for Dynamic Scaling of Instances Behind the Load Balancer

Auto Scaling is a powerful tool that allows you to automatically scale the number of instances in your infrastructure up or down based on demand. By integrating your load balancer with Auto Scaling, you can ensure that your infrastructure is always right-sized to handle your workloads. Here's how to configure Auto Scaling with your load balancer:

### Step 1: Create an Auto Scaling Group

1. Open the Amazon EC2 console and navigate to the Auto Scaling groups page.
2. Click "Create Auto Scaling group" and follow the on-screen instructions to create your Auto Scaling group.
3. Specify the desired capacity, minimum capacity, and maximum capacity for your group, as well as any other configuration options that you need.
4. Configure your scaling policies to define how your group should scale up and down based on demand.

## Integrating with Auto Scaling for Dynamic Scaling of Instances Behind the Load Balancer

### Step 2: Register Your Instances with the Load Balancer

To register your instances with your load balancer, you can use either the Amazon EC2 console or the command line interface. Here's how to do it using the console:

1. Open the Amazon EC2 console and navigate to the Instances page.
2. Select the instances that you want to register with your load balancer.
3. Click "Actions", then "Add to Load Balancer".
4. Select the load balancer that you want to register your instances with, then click "Add".

## Integrating with Auto Scaling for Dynamic Scaling of Instances Behind the Load Balancer

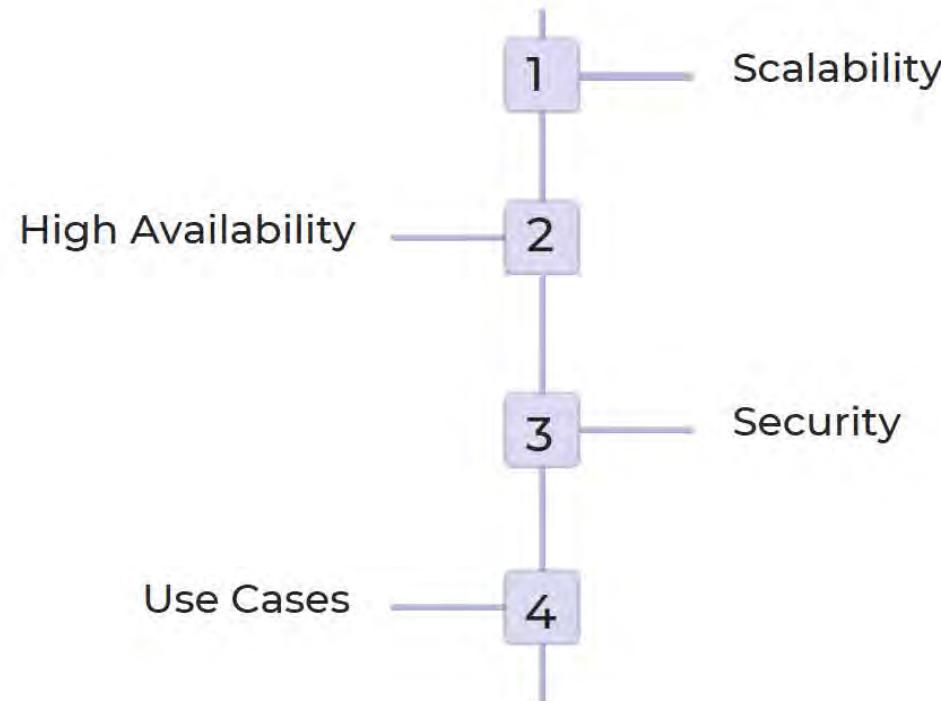
### Step 3: Configure Your Load Balancer to Use Your Auto Scaling Group

To configure your load balancer to use your Auto Scaling group, you need to update your target group. Here's how to do it:

1. Open the Amazon EC2 console and navigate to the target groups page.
2. Select the target group that you want to update.
3. Click "Edit", then select your Auto Scaling group from the "Registered" tab.
4. Click "Save" to save your changes.

By integrating your load balancer with Auto Scaling, you can ensure that your infrastructure is always optimized for your workloads. Your Auto Scaling group will automatically adjust the number of instances in your infrastructure based on demand, while your load balancer will distribute traffic evenly across all healthy instances, ensuring that your users receive a high-quality experience.

# Benefits and Use Cases of AWS Load Balancers



# Pricing of AWS Load Balancers

	Application Load Balancer	Network Load Balancer	Classic Load Balancer
Price per hour	\$0.0225	\$0.024	\$0.025
LCU Load Balancer	\$0.008	\$0.008	\$0.025
Capacity Units (LCU)	2,048 LCUs per hour	1 LCU per hour	1 LCU per hour

Load Balancer pricing is based on the number of hours and Load Balancer Capacity Units (LCUs) used per hour. Application Load Balancers are cheaper than Network and Classic Load Balancers.

# Troubleshooting Common Issues with AWS Load Balancers

## 1 High Latency

Check the network and application performance, validate the health of your targets, and explore options for scaling your instances.

## 2 HTTP 503 Errors

Ensure that the target instances are configured correctly, and check that they're receiving traffic from the Load Balancer. Switch to a healthy target group if needed.

## 3 SSL Certificate Issues

Ensure that the certificate is valid and review the security group rules to enable incoming traffic on appropriate ports.

# Best Practices for Implementing AWS Load Balancers



Security



Automation



Logging and Monitoring

iamneo



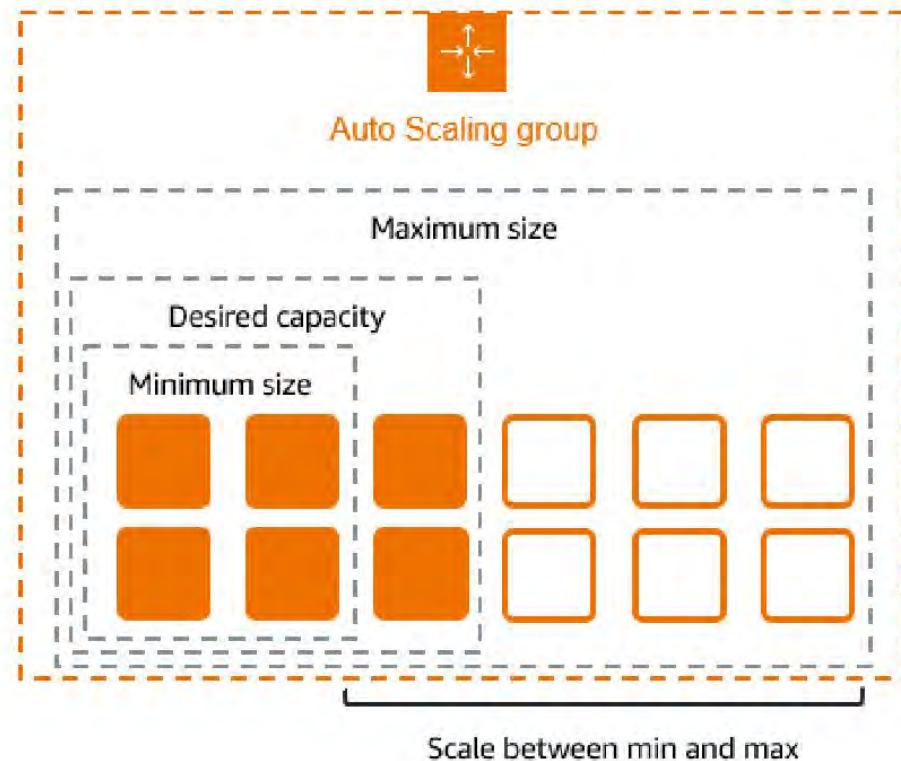
# AWS AutoScaling

# What is autoscaling and why is it important?

---

- Autoscaling is a feature of cloud computing that allows you to automatically adjust the number of computing resources in your system based on demand. This means that if your application suddenly becomes very popular and starts receiving a lot of traffic, autoscaling will automatically add more resources to handle the increased load.
- Autoscaling is important because it helps you ensure that your application is always available and responsive, even during times of high traffic.
- It also helps you save costs by only using the number of resources that you need, rather than paying for resources that are sitting idle.
- With autoscaling, you can achieve better performance, higher availability, and lower costs.

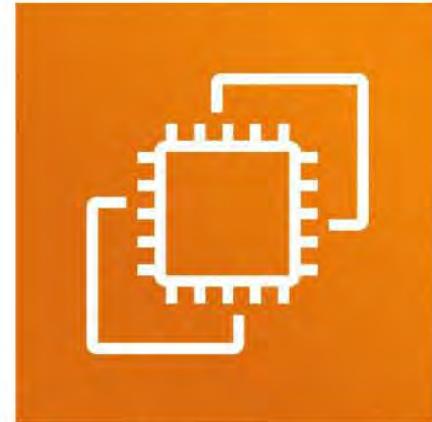
# What is autoscaling and why is it important?



# What is autoscaling and why is it important?



Scaling based on demand

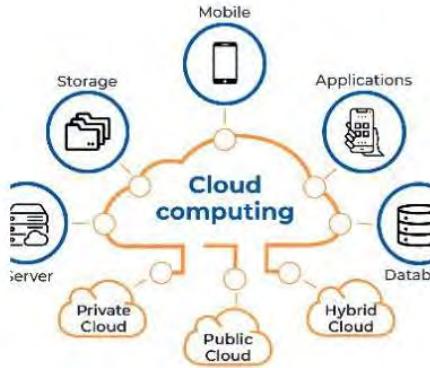


Multiple instance types supported



Integration with CloudWatch

# Why Autoscaling Matters?



Scalability and Flexibility



Cost Efficiency

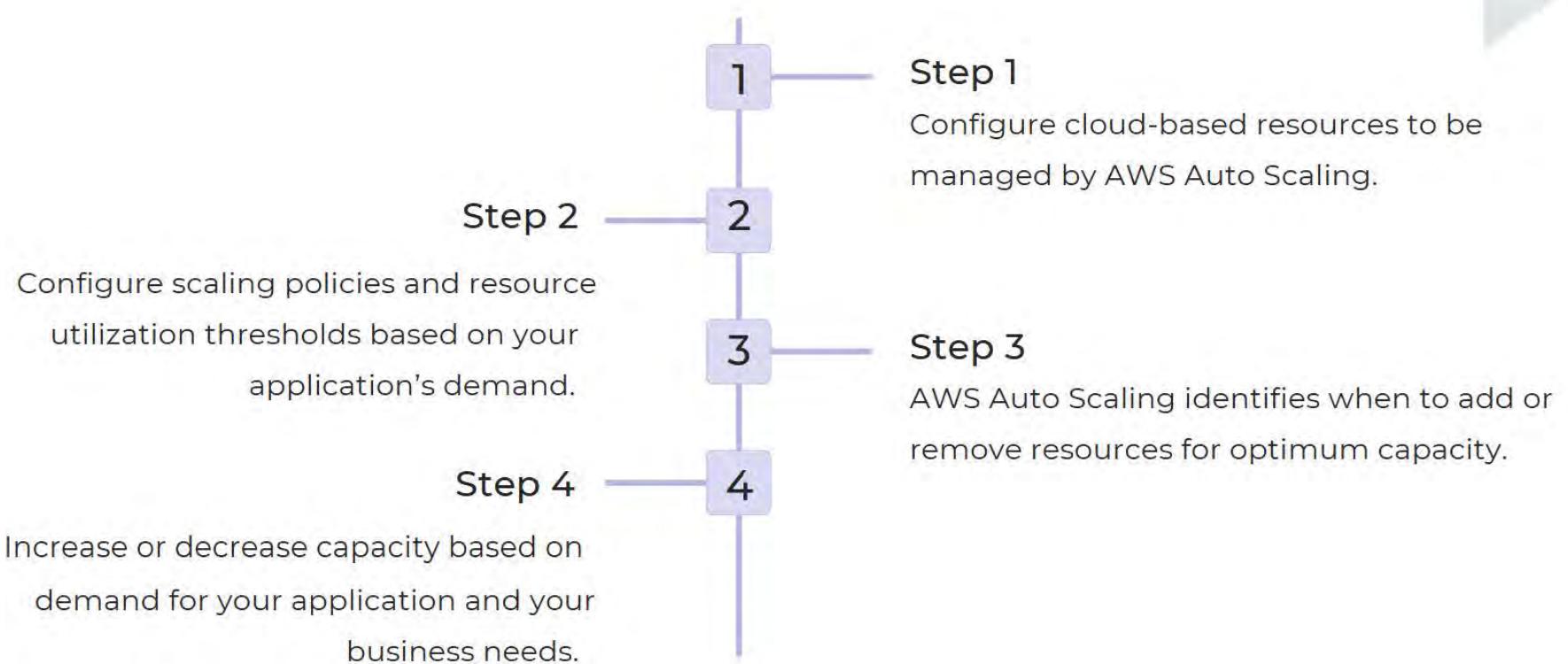


Performance and Reliability

# How does AWS Auto Scaling work?



# Implementation of AWS Auto Scaling



# Understanding the components of Auto Scaling

Auto Scaling is an AWS service that allows you to automatically adjust the capacity of your EC2 instances based on demand. There are three main components to Auto Scaling:

- **Auto Scaling groups:**A collection of EC2 instances that are created and managed together. You can specify the minimum and maximum number of instances in the group, and Auto Scaling will automatically adjust the number of instances based on demand.
- **Launch configurations:**A template that defines the settings for new instances that are launched by the Auto Scaling group. This includes the AMI, instance type, and security groups.
- **Scaling policies:**Rules that determine when and how to adjust the capacity of the Auto Scaling group. For example, you might create a scaling policy that adds 2 instances when CPU usage exceeds 80% for 5 minutes.

By using these components together, you can ensure that your application always has the right amount of capacity to handle traffic. You can also save money by only paying for the instances that you need.

# AutoScaling Groups

Defining Group Size Limits

Creating an AutoScaling Group

Configuring AutoScaling Triggers

Creating Multi-Zone Deployments

# Understanding Autoscaling Groups

---

- An autoscaling group is a collection of Amazon EC2 instances that are designed to work together to handle incoming traffic.

When you create an autoscaling group, you specify the minimum and maximum number of instances that should be running at any given time.

  - If the traffic to your application increases, the autoscaling group will automatically add more instances to handle the load. If the traffic decreases, the autoscaling group will remove some of the instances to save costs. This allows your application to handle variable levels of traffic without any manual intervention.

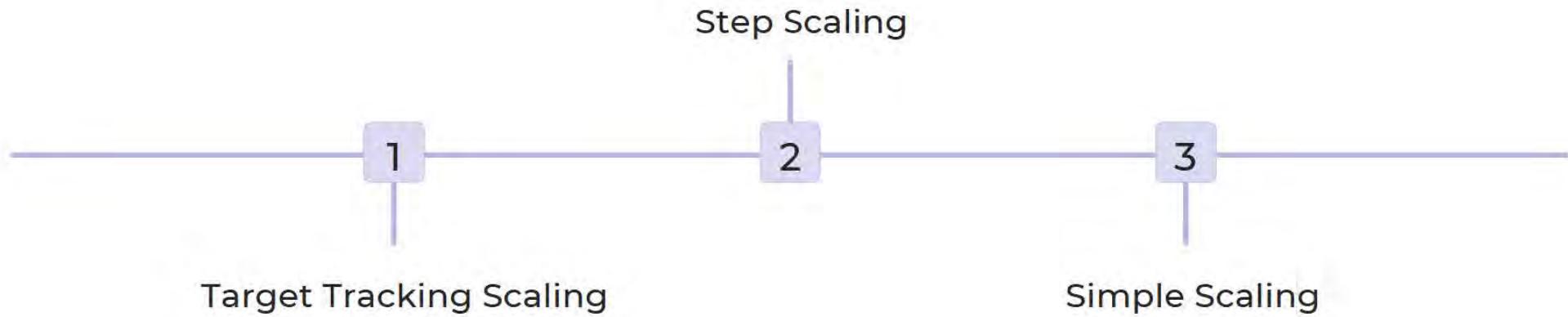
# Creating Auto Scaling groups and defining launch configurations

- Auto Scaling groups are a core component of AWS Auto Scaling. An Auto Scaling group contains a collection of Amazon EC2 instances that are created from a common Amazon Machine Image (AMI).
- The group automatically scales the number of instances up or down in response to changes in demand for the application. To create an Auto Scaling group, you'll need to:
  - 1.Create an Amazon Machine Image (AMI) that contains the software and configuration for your application.
  - 2.Create a launch configuration that describes the settings for the instances that will be launched by the Auto Scaling group. This includes the AMI ID, instance type, and security groups.
  - 3.Create an Auto Scaling group and specify the minimum and maximum number of instances that the group should maintain. You'll also need to specify the launch configuration that you created in step 2.

# Configuring Autoscaling Group Capacity

- When creating an autoscaling group, you need to specify the minimum, maximum, and desired capacity. The minimum capacity is the smallest number of instances that can be running at any time. The maximum capacity is the largest number of instances that can be running at any time. The desired capacity is the number of instances that should be running at any given time.
- When the autoscaling group launches, it will start with the desired capacity. If the traffic to your application increases and exceeds the desired capacity, the autoscaling group will automatically add more instances up to the maximum capacity. If the traffic decreases and goes below the desired capacity, the autoscaling group will remove some of the instances down to the minimum capacity.
- It's important to choose appropriate values for these parameters based on the expected traffic to your application. If the minimum capacity is too high, you'll be paying for instances that you don't need. If the maximum capacity is too low, your application won't be able to handle spikes in traffic.

# AutoScaling Policies



# Customizing AutoScaling with Lifecycle Hooks



Extending AWS Services



Terminate Protection



Amazon **EBS**

EBS Volume Creation

# Key Features and Benefits

---

Cost Optimization 

Quick and Accurate  
Scaling 

Improved Reliability 

# Use Cases

---



E-commerce applications



Mobile apps



Cloud infrastructure

# Challenges and Limitations

---

1 Application design

2 Scaling limitations

3 Costs

# Best Practices for AWS Auto Scaling

## Algorithm selection

Choose the algorithm that best meets the needs of your application.

## Application architecture

Consider implementing per-application scaling for optimal resource allocation.

## Monitoring and testing

Continuously monitor and test to ensure AWS Auto Scaling is working optimally with your applications.

## Scheduling

Use AWS Auto Scaling scheduled actions to proactively manage capacity.

# AutoScaling Case Studies: Examples of Successful Implementation

## Netflix

Netflix relies heavily on AutoScaling to stream content for millions of users worldwide. By dynamically allocating streaming resources as needed, Netflix ensures that users can watch their favorite shows and movies whenever they want.

## Nasa JPL

NASA's Jet Propulsion Laboratory (JPL) uses AutoScaling to process data and simulate workloads for various space missions. By scaling up or down based on demand, JPL ensures that its compute resources are optimized and cost-effective.

## Kajabi

Kajabi, an e-learning platform, uses predictive scaling to optimize compute resources during their peak hours. By predicting workload patterns and scaling in advance, they ensure students have smooth interaction with courses delivered on the platform.

iamneo



# Amazon Lambda

---



AWS Lambda

**Serverless Computing Made  
Easy with AWS Lambda**

# What is Serverless Computing?

---

## Cloud-based

Serverless computing enables developers to focus on writing code for specific tasks, rather than managing servers or infrastructure.

## Cost-effective

Because serverless computing only requires payment for actual usage, it's often more cost-effective than traditional hosting or infrastructure.

## Event-driven

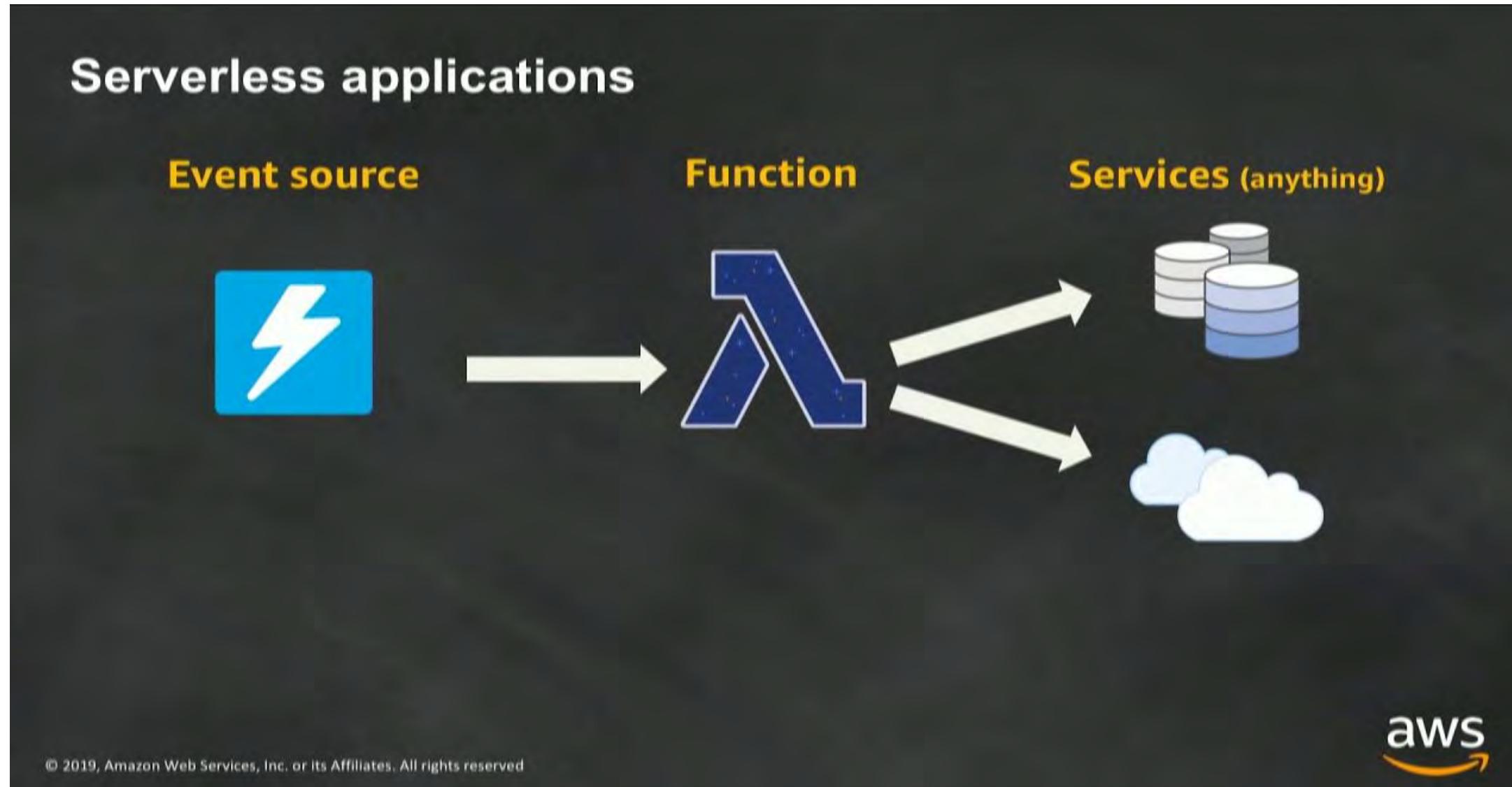
With serverless computing, code is executed only in response to events, such as data changes or user actions, making it highly efficient and scalable.

# Why AWS Lambda?

---

- **Flexible**
- **Scalable**
- **Cost-effective**
- **Integrative**

# How AWS Works



# How AWS Works



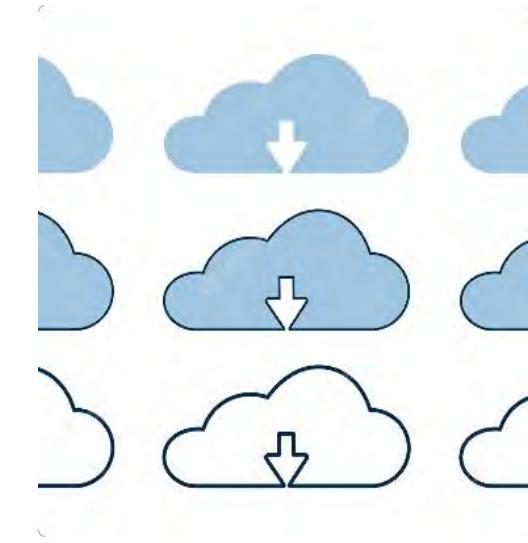
## Amazon Web Services

AWS is a cloud computing platform that provides a wide range of scalable services, including AWS Lambda, to businesses and individuals.



## Code and Triggers

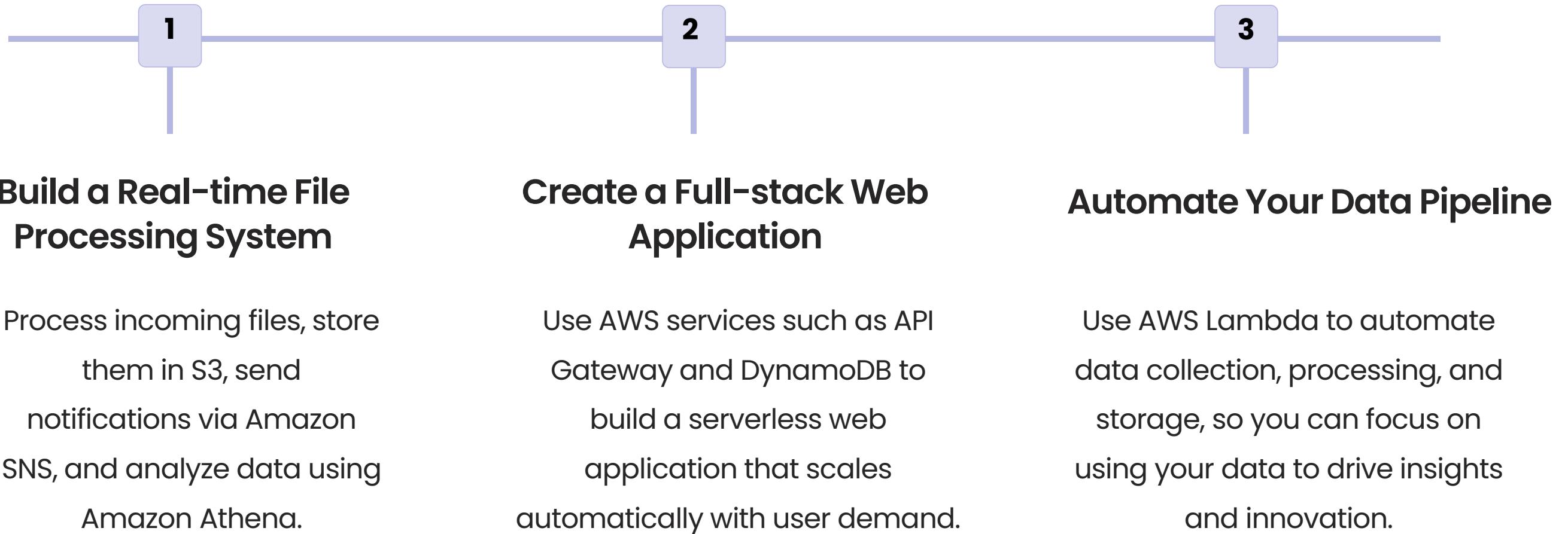
Developers create code in the appropriate language, set up triggers that launch the code, and rely on AWS to execute the code securely and efficiently.



## Leveraging the Cloud

With AWS Lambda, developers can leverage the power of the cloud, without worrying about server maintenance and setup.

# Use Cases for AWS Lambda



# Building Your First Lambda Function

---

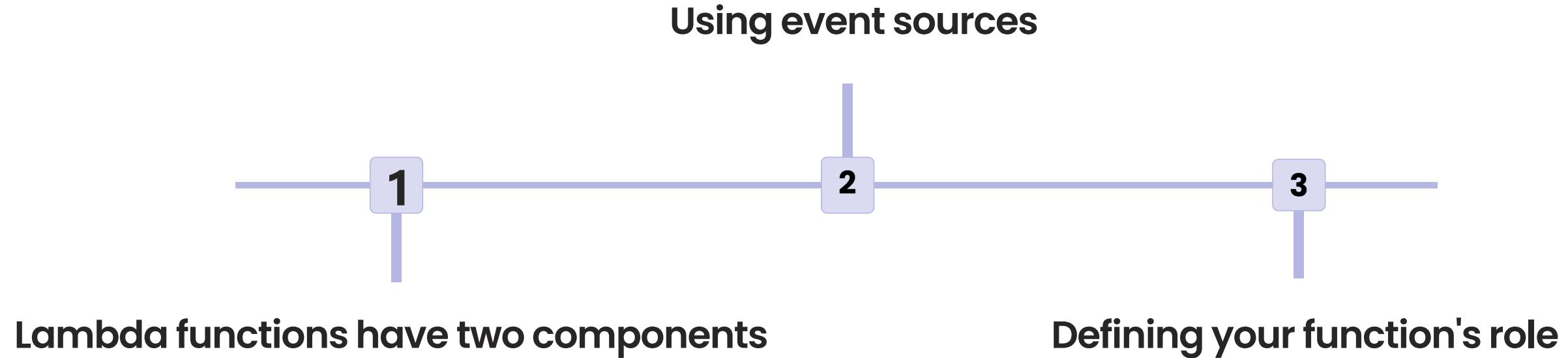
**Step 1: Choose a programming language**

**Step 2: Define a handler function**

**Step 3: Configure triggers**

**Step 4: Testing and debugging**

# Structuring Your Lambda Functions



# Triggering and Invoking Lambda Functions

## Manual Invocation

You can invoke a Lambda function manually using the AWS console.

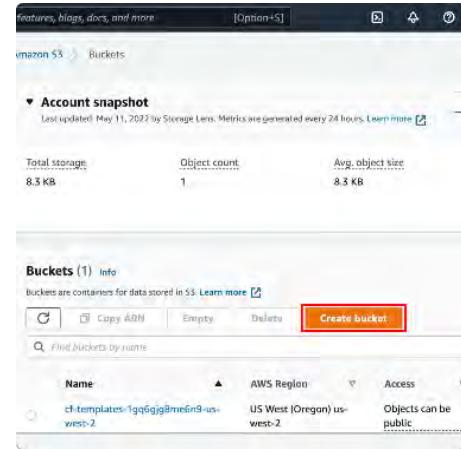
## API Gateway Trigger

Create an API Gateway and configure it to trigger your Lambda function based on HTTP requests.

## Event Trigger

Your function can be triggered based on events in other AWS resources, such as S3 object creation, Kinesis Data Streams, and DynamoDB.

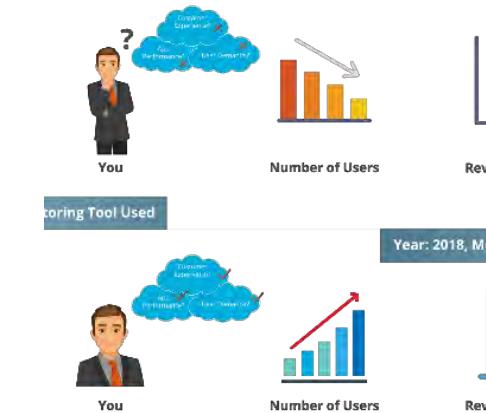
# Integrating AWS Lambda Functions with Other Services



Integrating with Amazon S3

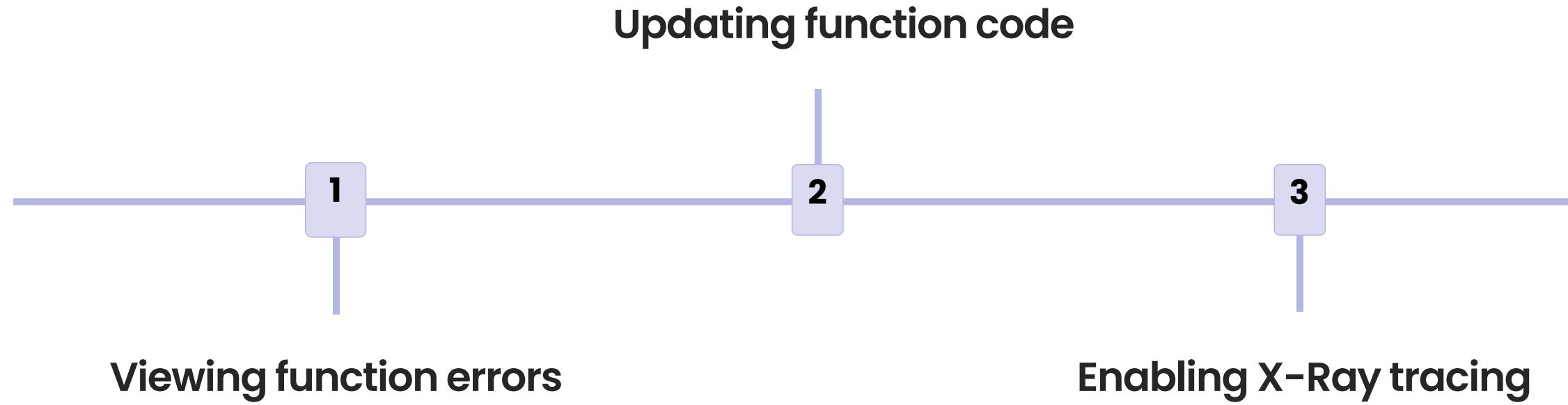


Integrating with Amazon Alexa



Managing and Monitoring Your Lambda Functions

# Managing and Monitoring Lambda Functions



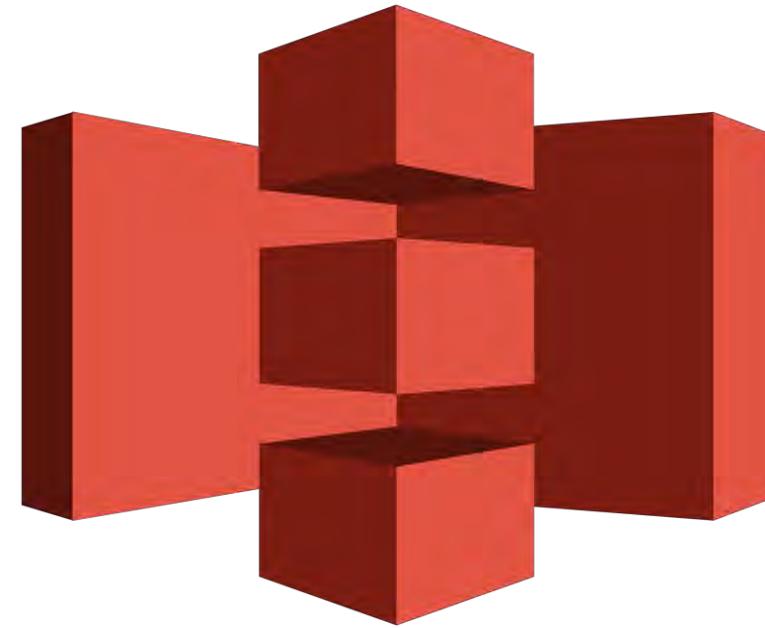
iamneo



# Amazon S3

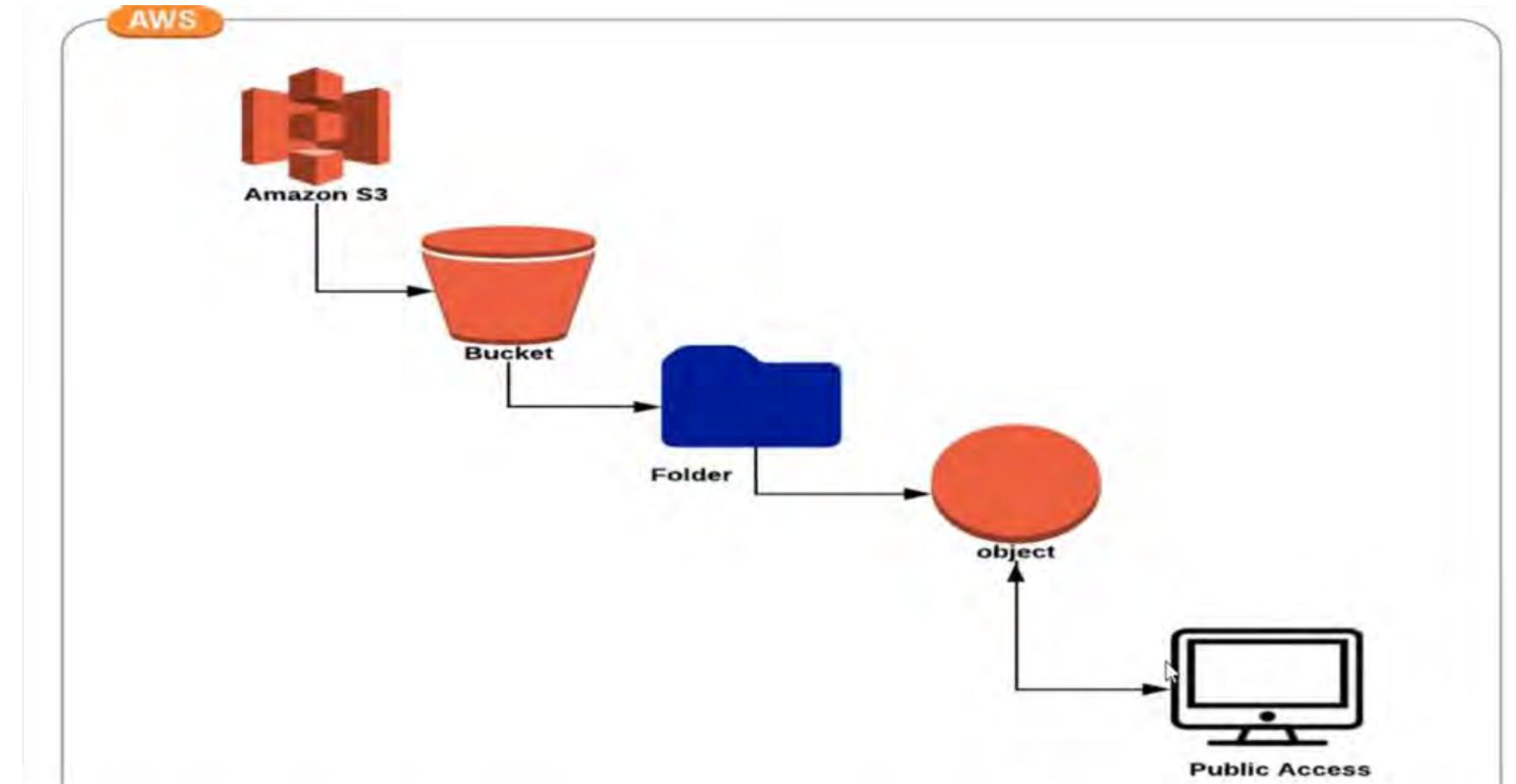
---

# Introduction to Amazon S3



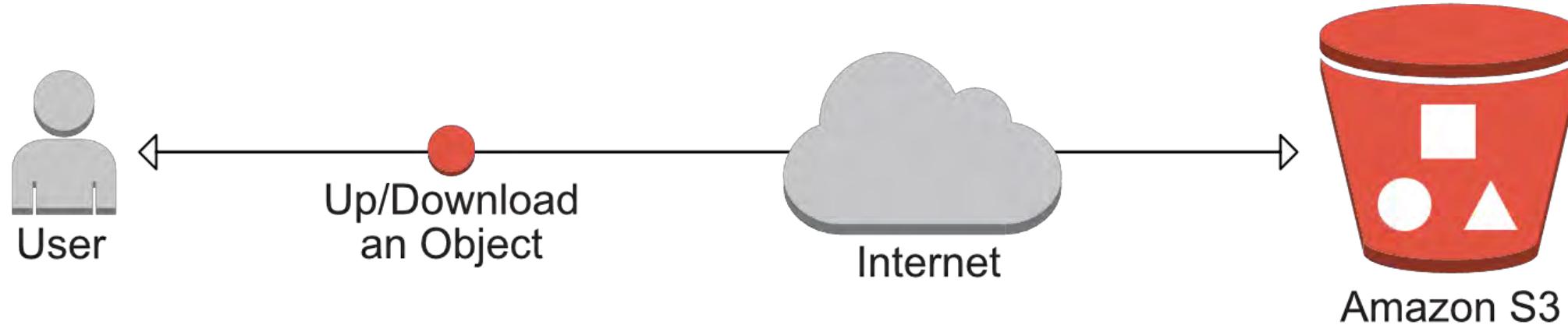
- Amazon Simple Storage Service (Amazon S3) is an object storage service offering industry-leading scalability, data availability, security, and performance.
- Customers of all sizes and industries can store and protect any amount of data for virtually any use case, such as data lakes, cloud-native applications, and mobile apps.
- With cost-effective storage classes and easy-to-use management features, you can optimize costs, organize data, and configure fine-tuned access controls to meet specific business, organizational, and compliance requirements.

# S3 Bucket Concepts



S3 buckets are the containers for storing objects. Buckets are accessed using a unique Amazon Resource Name (ARN).

# Object Storage



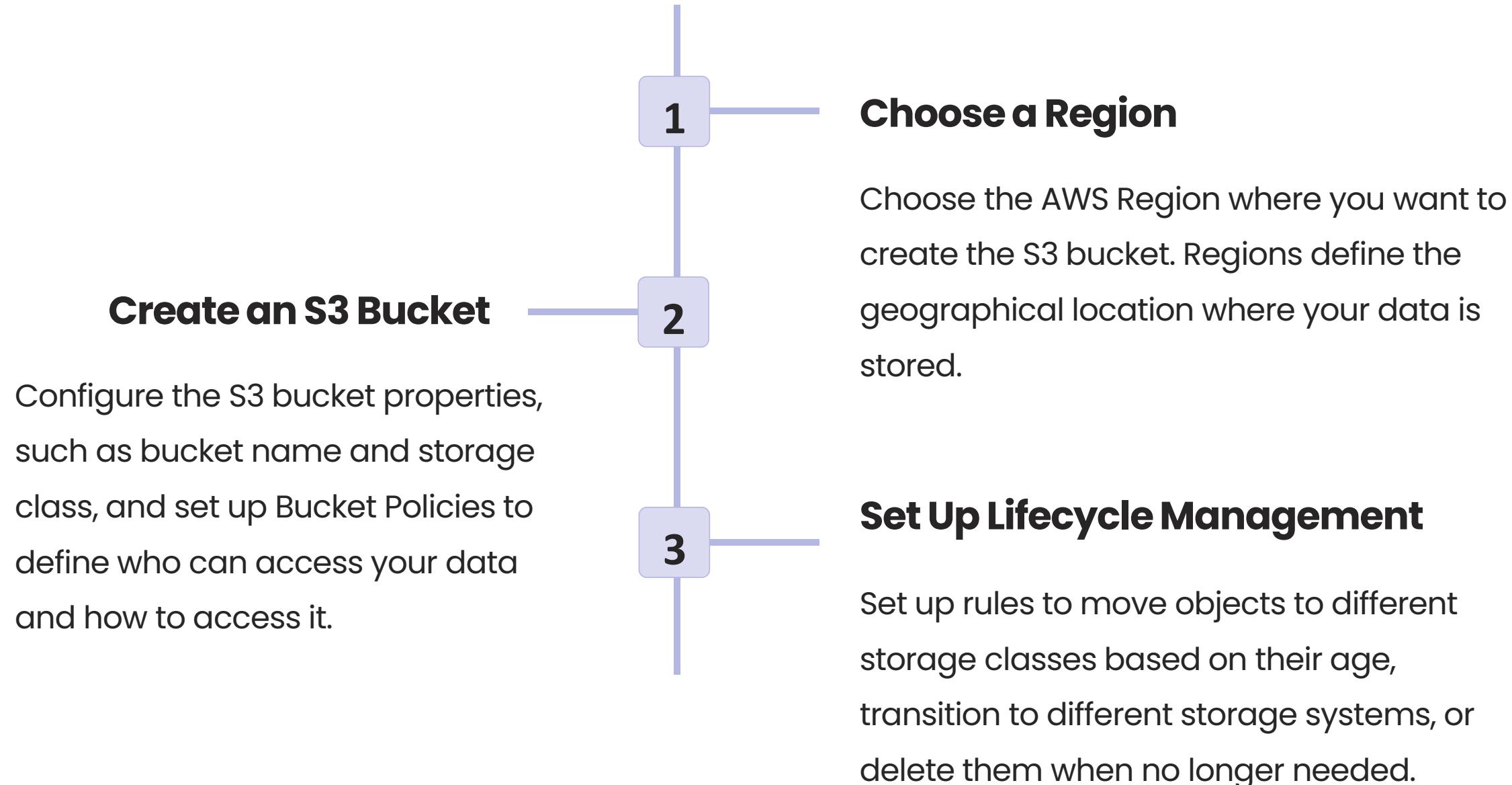
- Object storage is a technology that stores and manages data in an unstructured format called objects.
- Modern organizations create and analyze large volumes of unstructured data such as photos, videos, email, web pages, sensor data, and audio files.
- Cloud object storage systems distribute this data across multiple physical devices but allow users to access the content efficiently from a single, virtual storage repository.

# Key Features

---

- Amazon S3 has various features you can use to organize and manage your data in ways that support specific use cases, enable cost efficiencies, enforce security, and meet compliance requirements.
- Data is stored as objects within resources called “buckets”, and a single object can be up to 5 terabytes in size.
- S3 features include capabilities to append metadata tags to objects, move and store data across the S3 Storage Classes, configure and enforce data access controls, secure data against unauthorized users, run big data analytics, monitor data at the object and bucket levels.
- Objects can be accessed through S3 Access Points or directly through the bucket hostname.

# Basic Configuration of Amazon S3



# Creating S3 Buckets and Setting Bucket Policies

---

## Bucket Naming Rules

A bucket name must be unique, DNS-compliant, and follow specific naming rules, such as no uppercase letters or underscores, and between 3-63 characters long.

## Bucket Properties

You can set properties such as Location, Permissions, Versioning, Logging, and Tagging. Versioning enables you to store multiple versions of an object in the same bucket.

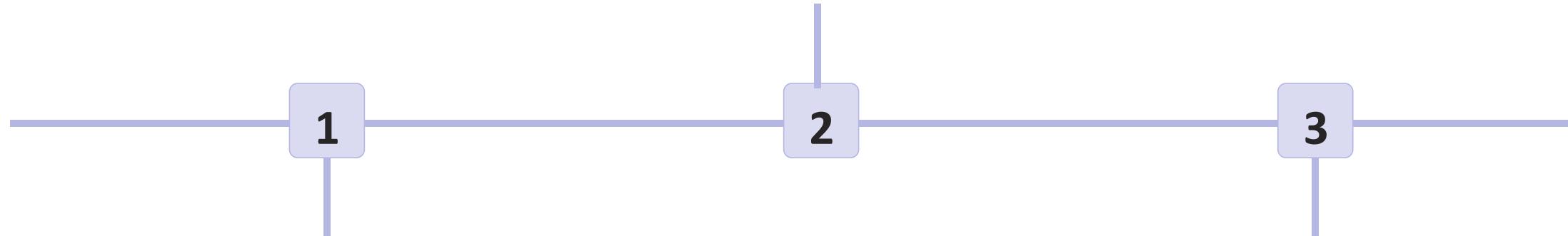
## Bucket Policies

You can create policies to define who can perform specific actions on the bucket, such as listing objects or uploading new objects. You can define users, groups, or roles and use specific conditions to limit access.

# Configuring S3 Object Permissions and Access Control

## S3 Object Tags

You can tag objects with metadata that enables you to categorize and search for objects based on their properties. Tags can be used to manage object lifecycle, to control access permissions, or for billing and cost management purposes.



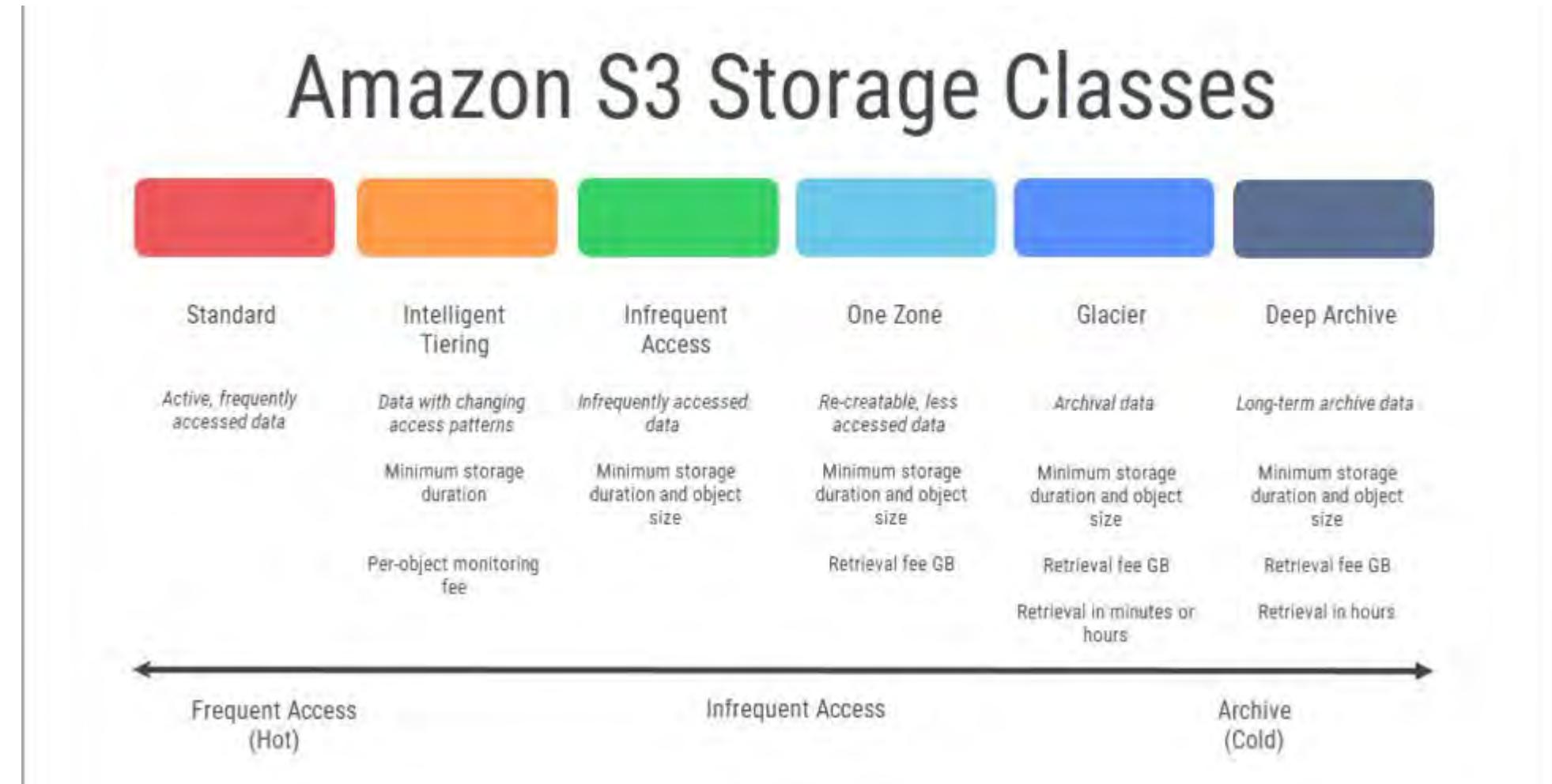
## Access Control Lists (ACLs)

ACLs provide finer-grained permissions to individual objects within the bucket. You can define permissions for specific users, groups, or public access. ACLs can be modified at any time on a per-object basis.

## Pre-Signed URLs

Pre-Signed URLs enable you to grant specific users temporary access to private objects in the bucket. You can set an expiration time and permissions for each URL.

# Understanding S3 Storage Classes



# Understanding S3 Storage Classes

---

## **Standard**

The default storage class. It is designed for frequently accessed data that requires low latency and high throughput

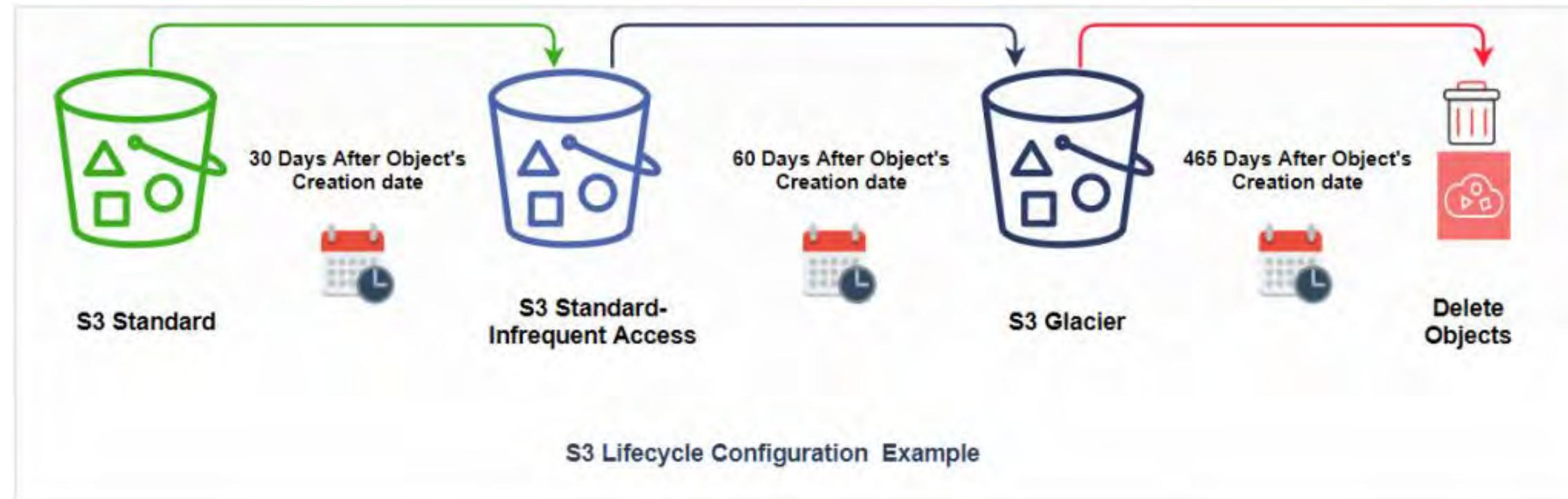
## **Intelligent-Tiering**

Designed for data with unknown or changing access patterns. It automatically moves objects between two access tiers based on changing access patterns.

## **Glacier**

Designed for data archiving and long-term backup. Data is stored for months or years and can take several hours to retrieve.

# Managing S3 Storage Lifecycle



# Lifecycle Management

---

## Lifecycle Configuration

You can create rules that automatically transition objects to different storage tiers based on their age.

Storage tiers include S3 Intelligent-Tiering, S3 Standard-Infrequent Access, and Glacier

## Delete Markers

When an object is deleted, it's not immediately removed but instead marked with a delete marker. You can restore an object before the expiration period if needed.

## S3 Inventory

S3 Inventory provides CSV or ORC files that list all your objects, their location, size, and metadata. You can use S3 Inventory to audit your objects, prepare for compliance audits, or analyze cost and usage trends.

iamneo



# Amazon EBS

---

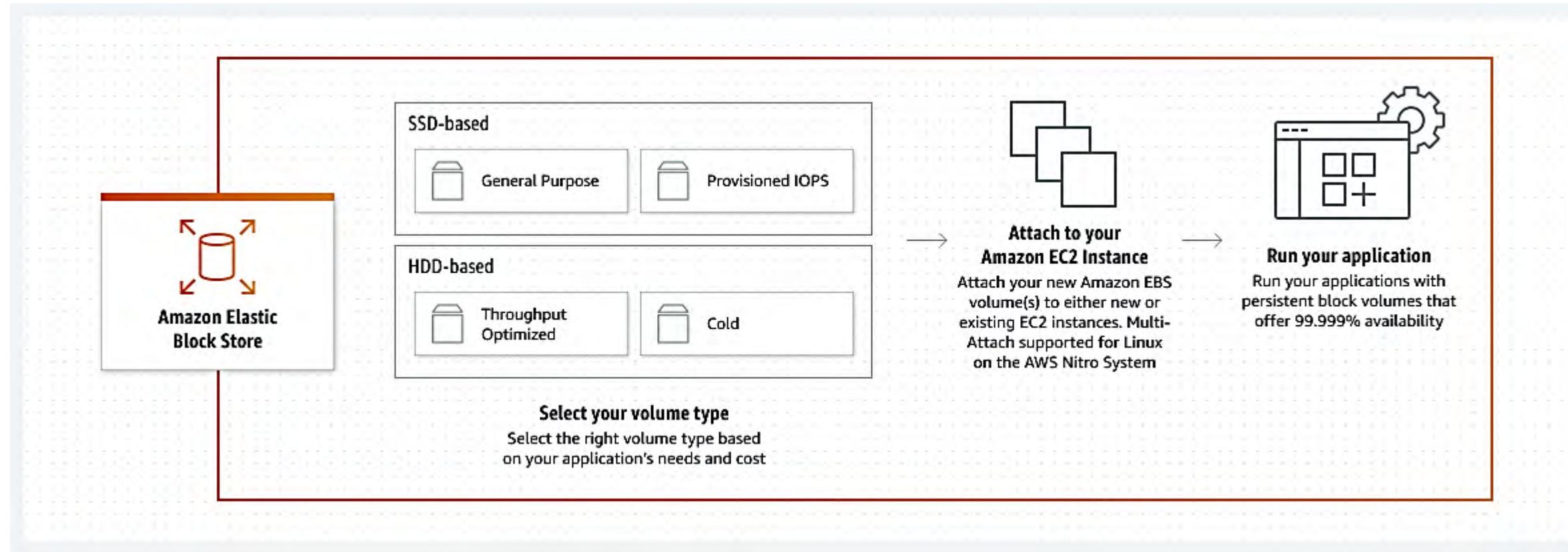
# Understanding S3 Storage Classes

---



- Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with EC2 instances. EBS volumes behave like raw, unformatted block devices.
- You can mount these volumes as devices on your instances. EBS volumes that are attached to an instance are exposed as storage volumes that persist independently from the life of the instance. You can create a file system on top of these volumes, or use them in any way you would use a block device.
- You can dynamically change the configuration of a volume attached to an instance.

# EBS Working



# EBS Volume Types



# EBS Volume Types Comparison

	General Purpose SSD		Provisioned IOPS SSD				
Volume type	gp3	gp2	io2 Block Express ‡	io2	io1		
Durability	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.999% durability (0.001% annual failure rate)	99.999% durability (0.001% annual failure rate)	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)		
Use cases	<ul style="list-style-type: none"><li>Low-latency interactive apps</li><li>Development and test environments</li></ul>		Workloads that require: <ul style="list-style-type: none"><li>Sub-millisecond latency</li><li>Sustained IOPS performance</li><li>More than 64,000 IOPS or 1,000 MiB/s of throughput</li></ul>	<ul style="list-style-type: none"><li>Workloads that require sustained IOPS performance or more than 16,000 IOPS</li><li>I/O-intensive database workloads</li></ul>			
Volume size	1 GiB - 16 TiB		4 GiB - 64 TiB	4 GiB - 16 TiB			
Max IOPS per volume (16 KiB I/O)	16,000		256,000	64,000 †			
Max throughput per volume	1,000 MiB/s	250 MiB/s *	4,000 MiB/s	1,000 MiB/s †			
Amazon EBS Multi-attach	Not supported		Supported				
Boot volume	Supported						

# EBS Use cases

---

## ➤ Block-Level Storage for EC2 Instances:

- The primary use case of AWS EBS is to provide block-level storage volumes for EC2 instances.
- EBS volumes act as durable, persistent storage that can be attached and detached from EC2 instances as needed.
- It enables data storage and retrieval for applications running on EC2 instances, offering flexibility and scalability.

## ➤ Database Storage:

- EBS volumes are commonly used for database storage, including both relational databases and NoSQL databases.
- EBS provides consistent, low-latency storage performance, making it well-suited for database workloads that require high I/O throughput and low latency access to data.

# EBS Use cases

---

## ➤ Application and Web Server Storage:

- EBS volumes are used as the primary storage for application and web servers hosted on EC2 instances.
- Application code, web content, log files, and other static or dynamic assets can be stored on EBS volumes, allowing for data persistence and easy management.

## ➤ Disaster Recovery and Backup:

- EBS snapshots enable efficient backup and disaster recovery solutions for EC2 instances and EBS volumes.
- Snapshots can be used to create point-in-time copies of EBS volumes, which can be stored in Amazon S3 for long-term durability and used to restore data in case of data loss or system failures.

# EBS Use cases

---

## ➤ Big Data Analytics:

- EBS volumes are commonly used for storing and processing large datasets in big data analytics scenarios.
- EBS provides the required performance and capacity to handle the massive volumes of data generated by analytics workloads, enabling efficient data processing with services like Amazon EMR (Elastic MapReduce) or self-managed analytics frameworks.

## ➤ Content Management and Media Workloads:

- EBS volumes can be used to store and manage content for content management systems (CMS) or media workloads.
- It provides fast and reliable storage for content repositories, media files, and other assets, allowing for efficient retrieval and distribution of content to end-users.

# EBS Use cases

---

## ➤ Development and Testing Environments:

- EBS volumes are commonly used in development and testing environments to store code repositories, development environments, and test data.
- EBS snapshots can be used to create consistent and reproducible copies of environments, simplifying the process of creating development and testing instances.

iamneo

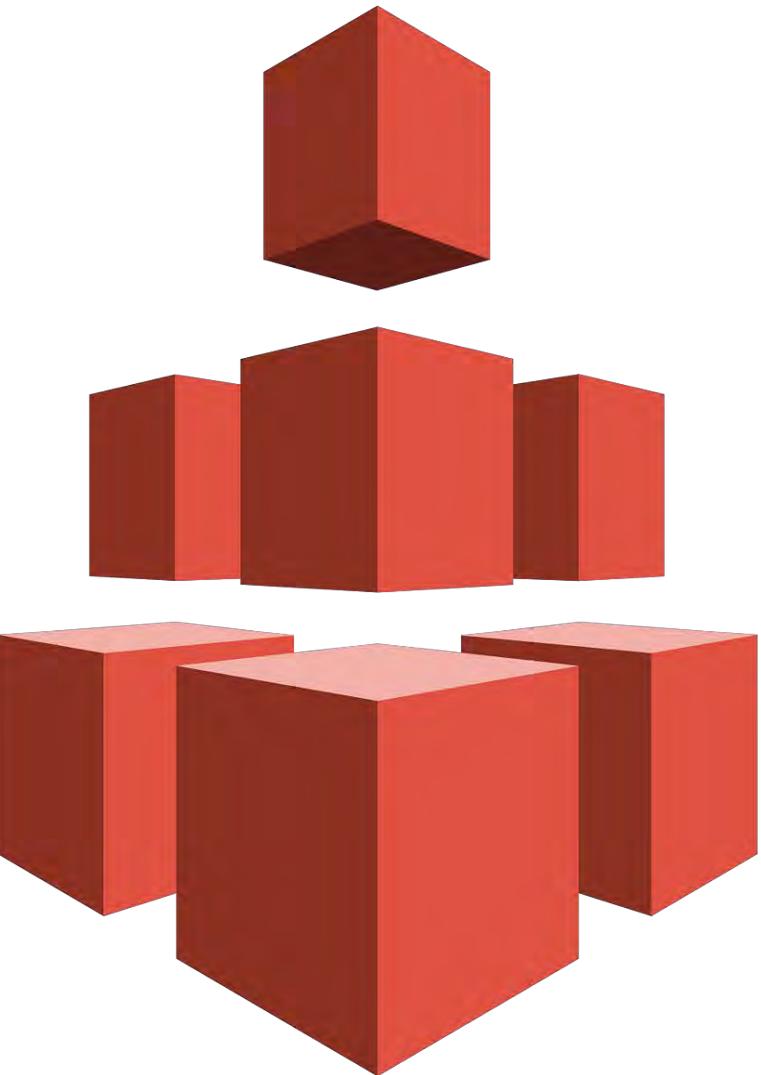


# Amazon EFS

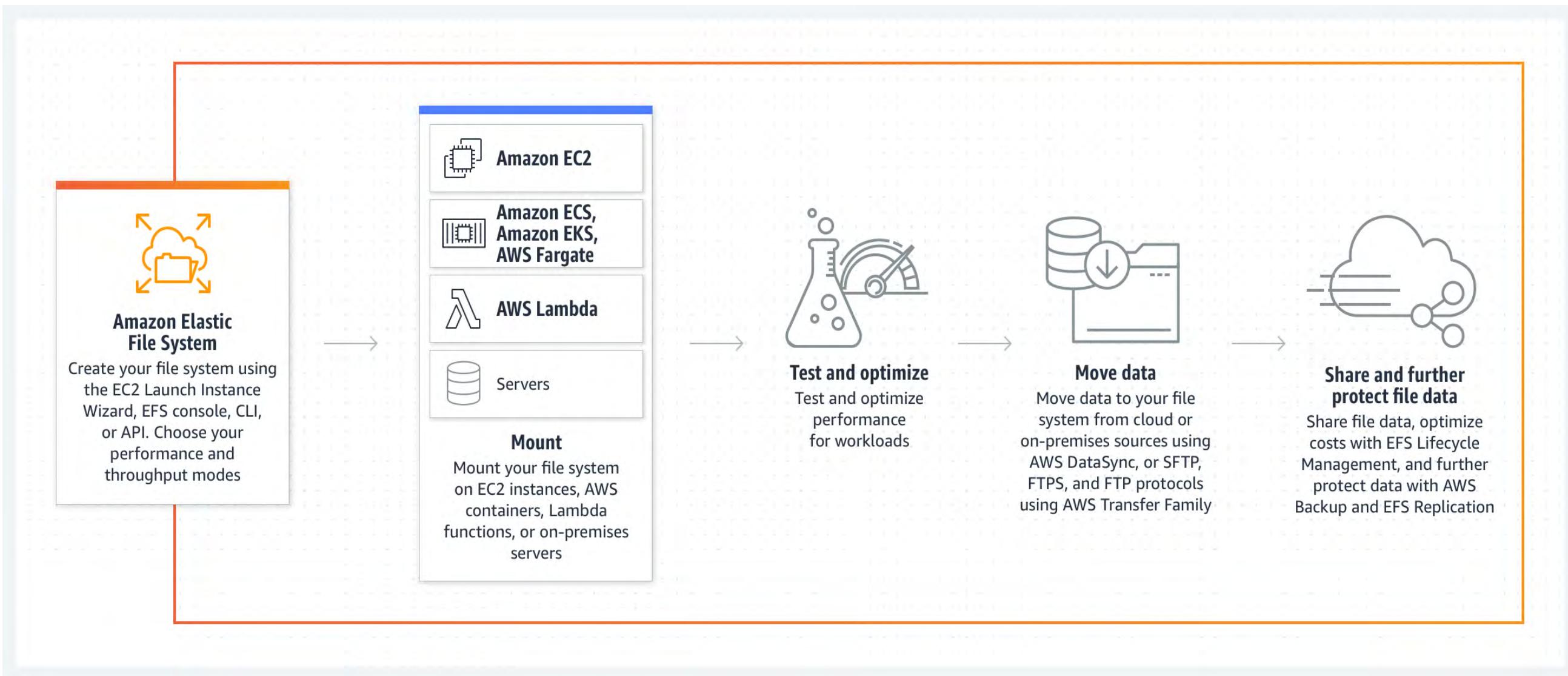
---

# Exploring Amazon EFS

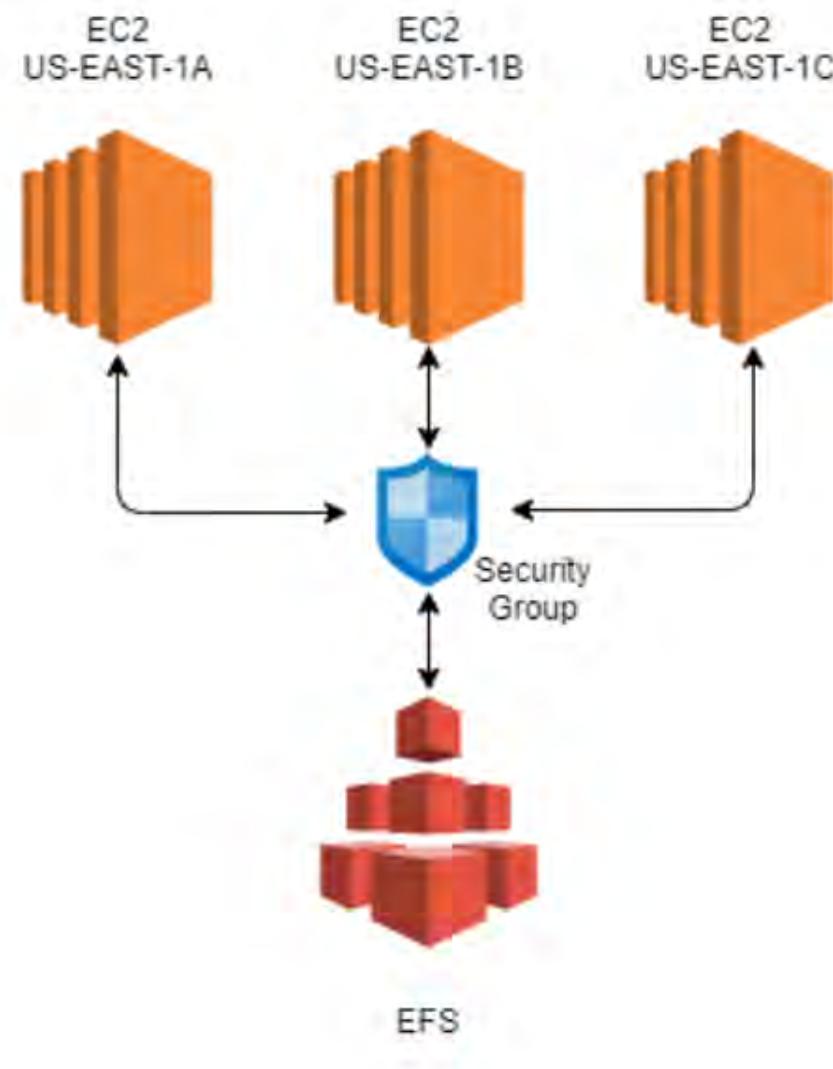
Amazon EFS (Elastic File System) is a scalable, secure, and fully-managed cloud-based file system with high availability and easy access to data from multiple instances. Let's dive deeper into its features, benefits, and potential use cases.



# How it Works



# Elastic File System (EFS)



- It is a Managed NFS that can be mounted on many EC2 instances within the same region but as many Availability zones you want.
- EFS works with EC2 instances in Multi-AZ.
- Highly available, Scalable, expensive (3 times more than Gp2), but you can pay for what you use.
- Use Cases: Content Management, web serving, Word press and much more.
- Uses NFSv4.1 protocol.

# Elastic File System (EFS)

---

- Uses security group to control access to EFS.
- Compatible with Linux Based AMI (Not Windows).
- Performance:
  - General Purpose (by default)
  - Max I/O – used when thousands of EC2 are using the EFS.
- File Sync to Sync from on-premise File system to EFS.
- Backup EFS-to-EFS (it is incremental, and you can choose frequency)
- Encryption at rest using KMS.

# The Many Types of Amazon EFS

## Standard

A general-purpose file system for Big Data, content repositories, web server logs, and more.

## Infrequent Access

A cost-optimized file system for infrequently accessed data, with lower storage prices. Ideal for backups, logs, and workflows.

## One Zone

A fully-managed file system with data stored in a single availability zone, useful for workloads that do not need multi-AZ resiliency.

## Lifecycle Management

An automated feature that moves your files between different storage classes based on the frequency of access, reducing costs and optimizing performance.

# Features That Make EFS Stand Out

---

**1 Easy to Use**

**2 High Performance**

**3 Secure and Compliant**

**4 Scalable and Flexible**

# Use Cases for Amazon EFS

---

## Big Data Analytics

EFS can store large volumes of data generated by data processing workflows, Big Data analytics, and machine learning models.

## Content Repositories

EFS provides a shared file system for content creation, document management, and collaborative work, accessible by all team members and devices.

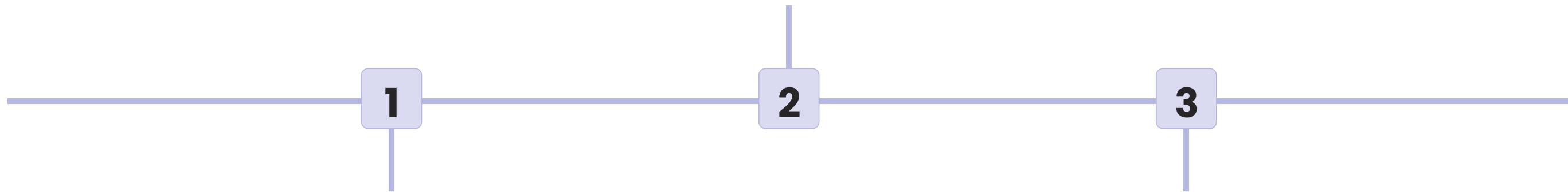
## Web Applications

EFS can support highly-available and scalable web applications, allowing file sharing across elastic deployments, caching layers, and fault-tolerant architectures.

# EFS Performance and Scalability

## Throughput Modes

EFS supports two throughput modes which you can switch dynamically: bursting and provisioned. Bursting mode offers up to 10Gbps throughput when a file system is idle, and it applies a bursting credit approach, while provisioned mode offers a predictable throughput level, ranging from 0.001 - 1024 MB/s, based on your setup.



## I/O Operations

EFS uses an optimized Linux file system driver to minimize write and metadata operations, and support sustained read/write performance at scale.

## Multi-AZ Resiliency

EFS offers data replication across multiple Availability Zones (AZs) within a region, ensuring high durability levels and preventing data loss in case of an outage.

# Managing Your Data with Amazon EFS

---

## Backing Up Your Data

- Use Lifecycle Management to move files to S3 buckets on a regular basis for long-term storage.
- Use EFS-to-EFS backup to copy file systems across regions or accounts for higher durability and disaster recovery capabilities.

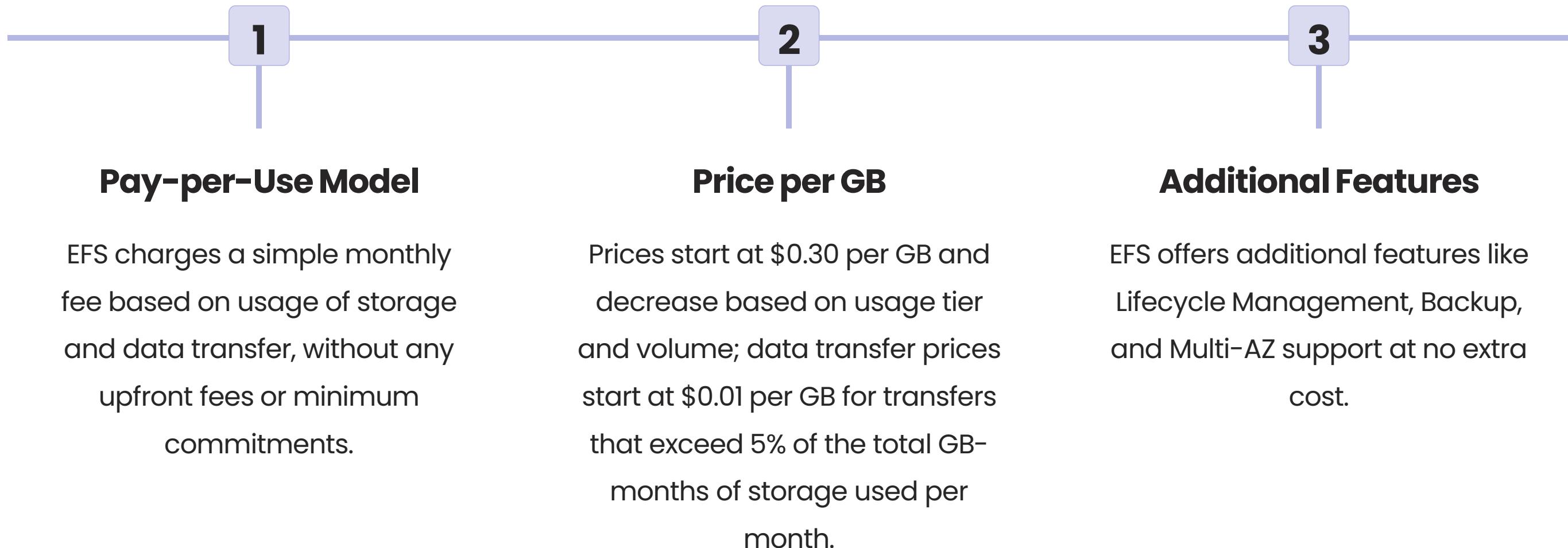
## Securing Your Data

- Encrypt data both in transit and at rest using AWS KMS or your own customer-managed CMK.
- Use IAM access controls to secure your file systems and manage user permissions.

## Monitoring Your Data

- Use Amazon CloudWatch metrics to monitor file system performance, I/O activities, and throughput utilization.
- Use Amazon CloudTrail logs to track file system events and changes for auditing and compliance purposes.

# Pricing and Billing for Amazon EFS

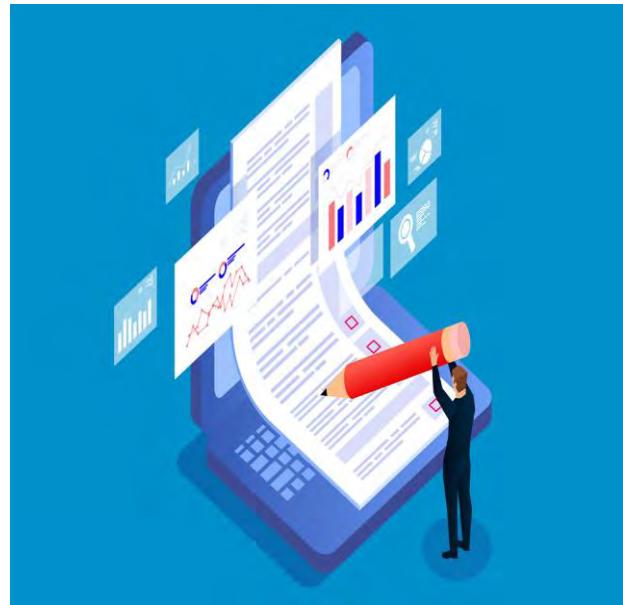


# Best Practices for Using Amazon EFS

---



**Data Management**



**Documentation and Expertise**



**Security and Compliance**

iamneo



# S3 Glacier

---

# Overview of S3 Glacier

---



Amazon S3 Glacier is a secure, durable, and cost-effective cloud storage service for data archiving and long-term backup. It is designed to deliver 99.99999999% durability and provide comprehensive data management.

# Storage Classes and Pricing

---

## S3 Glacier

- S3 Glacier is a low-cost storage service designed for data archiving. It's ideal for data that is seldom accessed but needs to be retained for a long period of time.
- However, retrieval times can range from minutes to hours, so it may not be the best choice for data that needs to be accessed frequently or quickly.
- One key feature of S3 Glacier is the ability to set up lifecycle policies, which automatically transition objects to Glacier from other S3 storage classes based on the age of the object.
- This can help reduce costs by moving less frequently accessed data to a lower-cost storage class, and by removing data that is no longer needed.

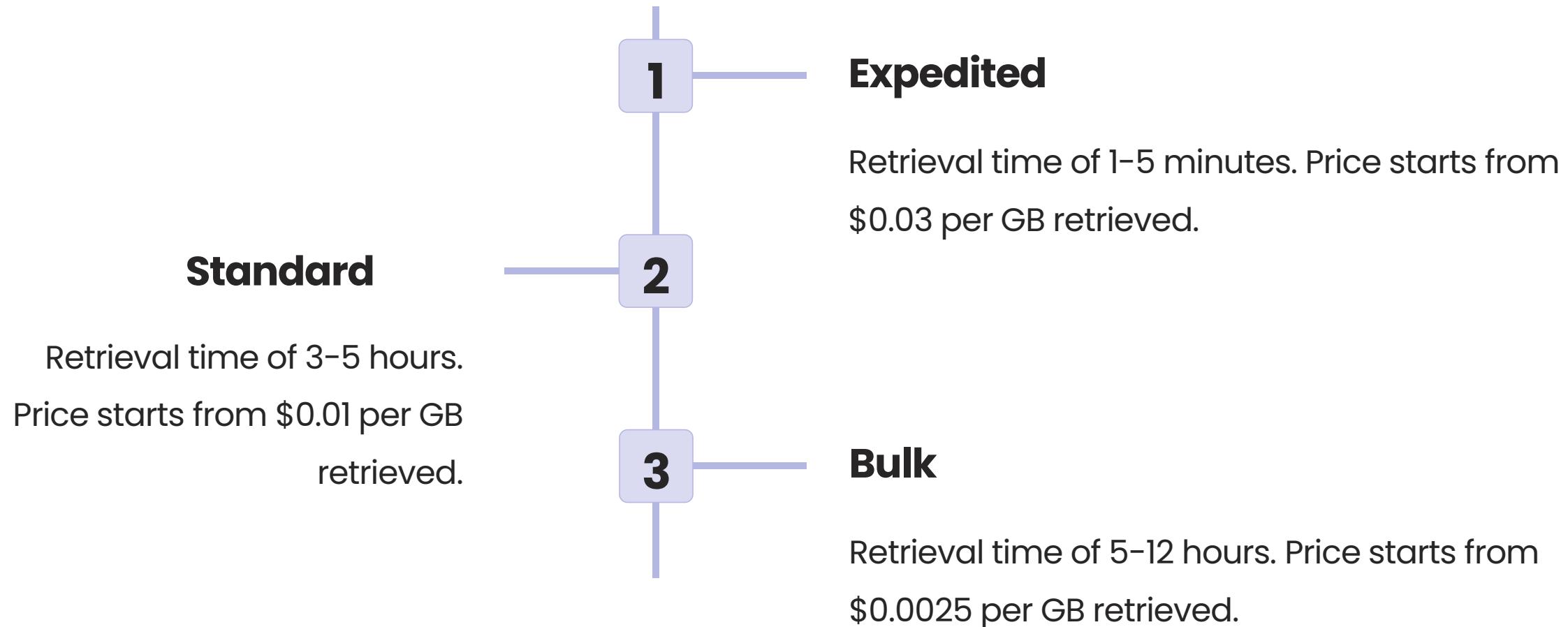
# Storage Classes and Pricing

---

## S3 Glacier Deep Archive

- S3 Glacier Deep Archive is the lowest-cost storage service available for long-term retention. It's designed for data that is rarely accessed and needs to be retained for up to several decades.
- The price per GB starts from \$0.00099/month, making it one of the most affordable storage options in the market
- One important thing to note is that S3 Glacier Deep Archive has a minimum storage duration of 180 days.
- This means that if you delete an object or remove it from the service before the 180 days are up, you will still be charged for the full 180-day duration.

# Retrieval Options and Costs



# Configuring S3 Glacier Vaults and Lifecycle Policies

---

1

## Creating a Glacier Vault

Use S3 console, command line tools, or AWS SDKs to create a Glacier vault.

2

## Lifecycle Policies

Automate the migration of objects between Amazon S3 and S3 Glacier, and between S3 Glacier and S3 Glacier Deep Archive using lifecycle policies.

3

## Notifications

Set up SNS notifications to get notified on vault and inventory events.

# Overview of S3 Glacier Deep Archive

## Cost-effective Archive Storage

S3 Glacier Deep Archive is the most cost-effective storage option for long-term data retention, backup, and archival.

## Retention Periods

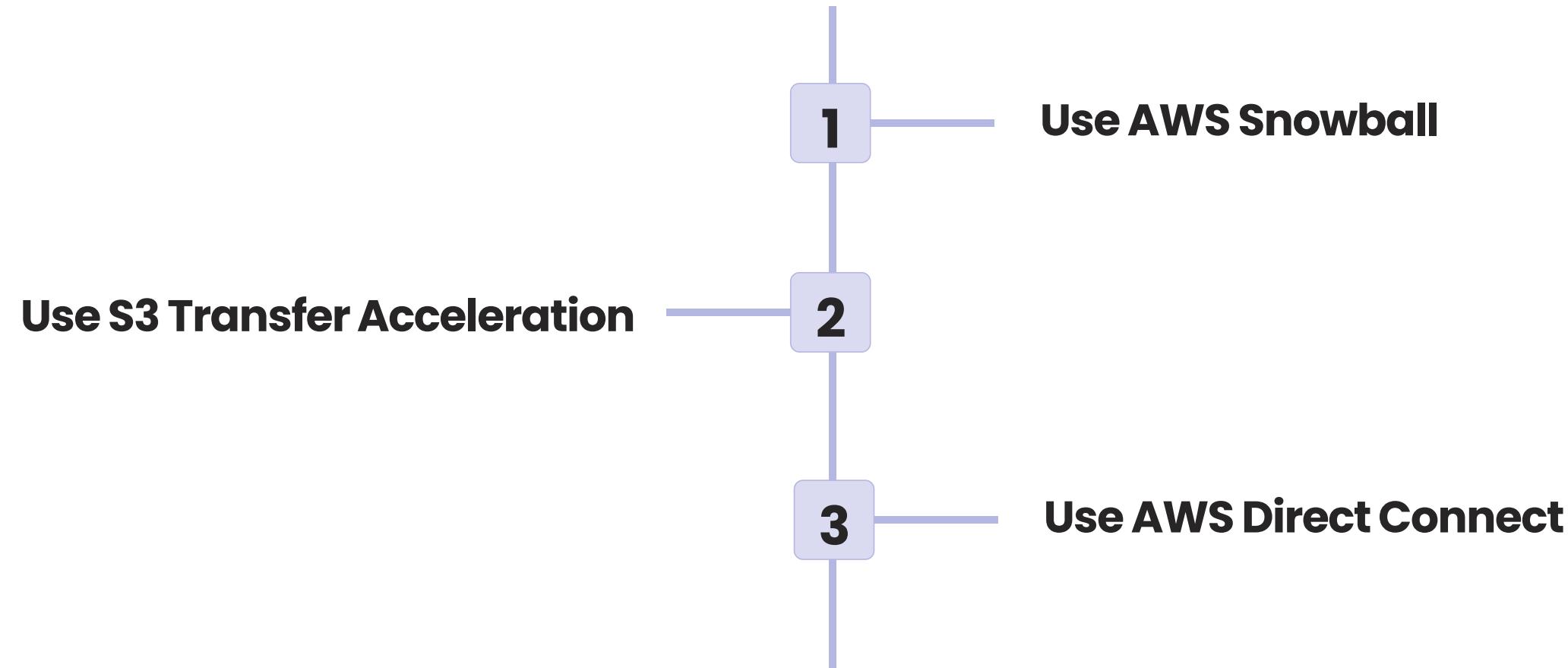
S3 Glacier Deep Archive provides retention periods ranging from seven years to 30 years.

## Secure Storage

S3 Glacier Deep Archive provides industry-standard AES-256 encryption for data at rest and in transit.

# Data Transfer

---



# Use Cases

---

1

## Data Archiving

Archive and manage data that may be rarely accessed, but still needs to be available for compliance.

2

## Backup and Disaster Recovery

Ensure reliable backup and recovery of critical business data in the cloud.

3

## Media Archives

Preserve and manage media assets, such as movies, audio files, and original recordings, for long-term storage.

# Benefits and Drawbacks

---

## Benefits

- Cost-effective
- Secure
- Reliable
- Scalable for all business sizes.
- Provides low-cost storage for rarely accessed but important data.

## Drawbacks

- Retrieving data can take a while depending on the retrieval option selected.
- Large object transfers can be challenging and can take time.

iamneo



# S3 Snowball

---

# Overview of AWS Snowball

---

## Hardware Device

- AWS Snowball is a physical device that organizations can use to migrate large amounts of their data to AWS.
- The device comes with a built-in E Ink shipping label, making it easy to track and manage the device during shipping.

## Data Migration

- Snowball enables organizations to migrate huge volumes of data to AWS with ease and confidence.
- It also provides tools to help you automate the data transfer process, saving you time and reducing the risk of error.

## Accelerated Transfer

- Snowball uses multiple network connections, helping to accelerate the data transfer process.
- This means you can transfer large amounts of data to AWS in less time than other data transfer methods.

# Overview of AWS Snowball

---

## Easier Integration

- Snowball integrates seamlessly with Amazon S3 and other AWS storage services to boost data transfer ease and consistency.
- This means you can easily integrate Snowball into your existing workflows and data management systems.

## Increased Security

- Snowball provides built-in security features to keep your data safe during the transfer process.
- The device is tamper-resistant and uses 256-bit encryption to secure your data.
- You also have the option to use your own encryption keys for added security.

# Benefits of using AWS Snowball

---

1

**Fast and Reliable Data Transfer**

2

**Cost-Effective Solution**

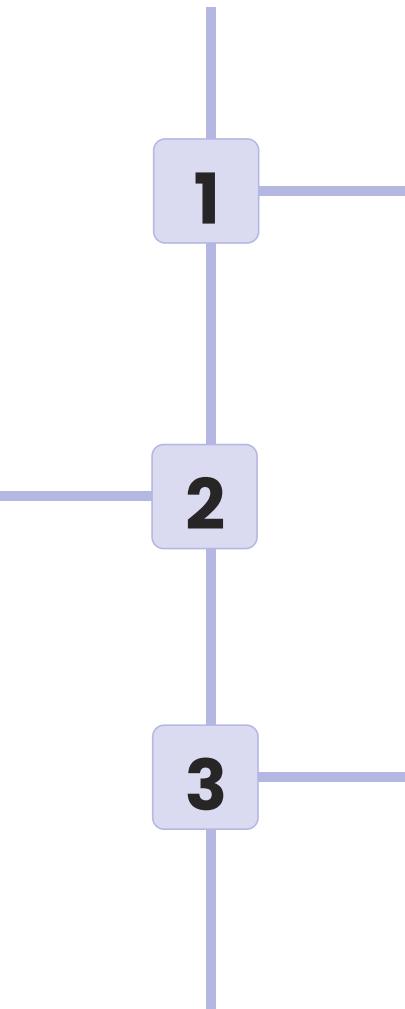
3

**Highly Secure**

# Use cases for AWS Snowball

## Backup and Disaster Recovery

Snowball enables companies to back up huge amounts of data to the cloud, ensuring safety from the risk of data loss and ensuring continuity of operations in the event of a disaster.



## Content Distribution

Snowball allows content providers to distribute video content and large files to end-users quickly and seamlessly.

## Big Data Applications

Snowball provides the perfect infrastructure for big data applications by enabling organizations to quickly move large amounts of data from their data center to their AWS environment.

# How to request and set up an AWS Snowball

## Open AWS Management Console

Log into the AWS Management Console and choose Amazon Snowball to create a new job.

## Specify Job Details

Enter details about the job and the data transfer, including pick-up and drop-off addresses, destination S3 bucket, and other details.

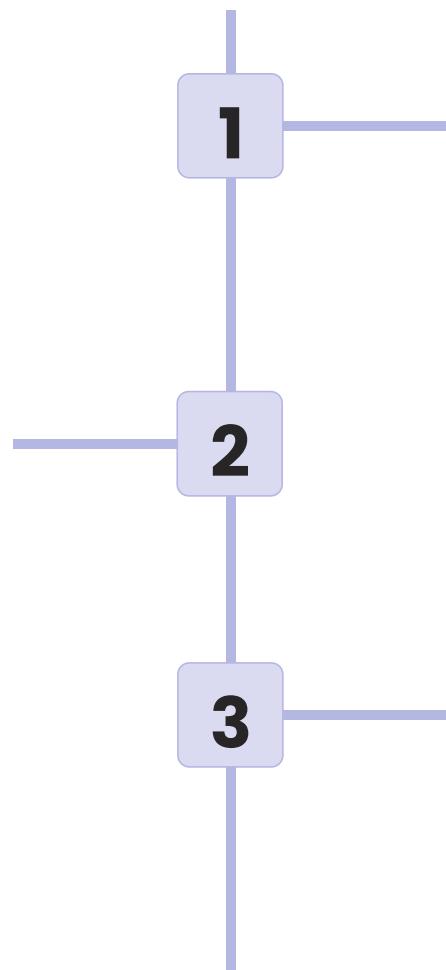
## Connect and Use the Snowball

Snowball gets shipped to the client, and then the data gets loaded onto the Snowball with the help of a user-friendly interface designed for novice users.

# How to transfer data with AWS Snowball

## Step 2. Load Data on the Device

Once you have created your export job, it's time to load your data onto the Snowball device. This is done using a user-friendly interface that is designed for novice users.



## Step 1. Create an Export Job

The first step is to use the AWS Management Console to create a new export job. This job contains information about the data you want to transfer.

## Step 3. Connect the Device and Create a Job

After your data is loaded onto the Snowball, it's time to connect the device to your AWS account and create an import job. Then, plug the Snowball into a power source and wait for the data to transfer.

# AWS Snowball pricing and limitations

**Data Transfer Cost:** \$0.03/GB

**Snowball Size Limit:** 80TB starting at \$300 + shipping costs

**Data Export:** Nothing

**Imported Data:** \$15 per Snowball transfer job

**Data Eras:** Nothing

**Snowball Capacity:** 80-100TB depending on the device version

# Conclusion and next steps

---

1

## Cost-Effective Data Migration

With AWS Snowball, organizations can transfer significant amounts of data to AWS in a simple yet cost-effective way.

2

## Flexible and Highly Secure

Snowball provides the perfect solution for transferring data, back up, and disaster recovery, as well as handling big data applications.

3

## Start Your Data Transfer Strategy Now

Sign up for AWS Snowball and make your data transfer easier and faster.

iamneo



# AWS Fundamentals and Security

---



[www.iamneo.ai](http://www.iamneo.ai)

# Overview of AWS

---

Amazon Web Services (AWS) is a cloud computing platform that offers a wide range of services including computing power, storage, security, analytics, and more. It is a comprehensive platform that enables developers to build and deploy applications quickly and efficiently.

**Fast**



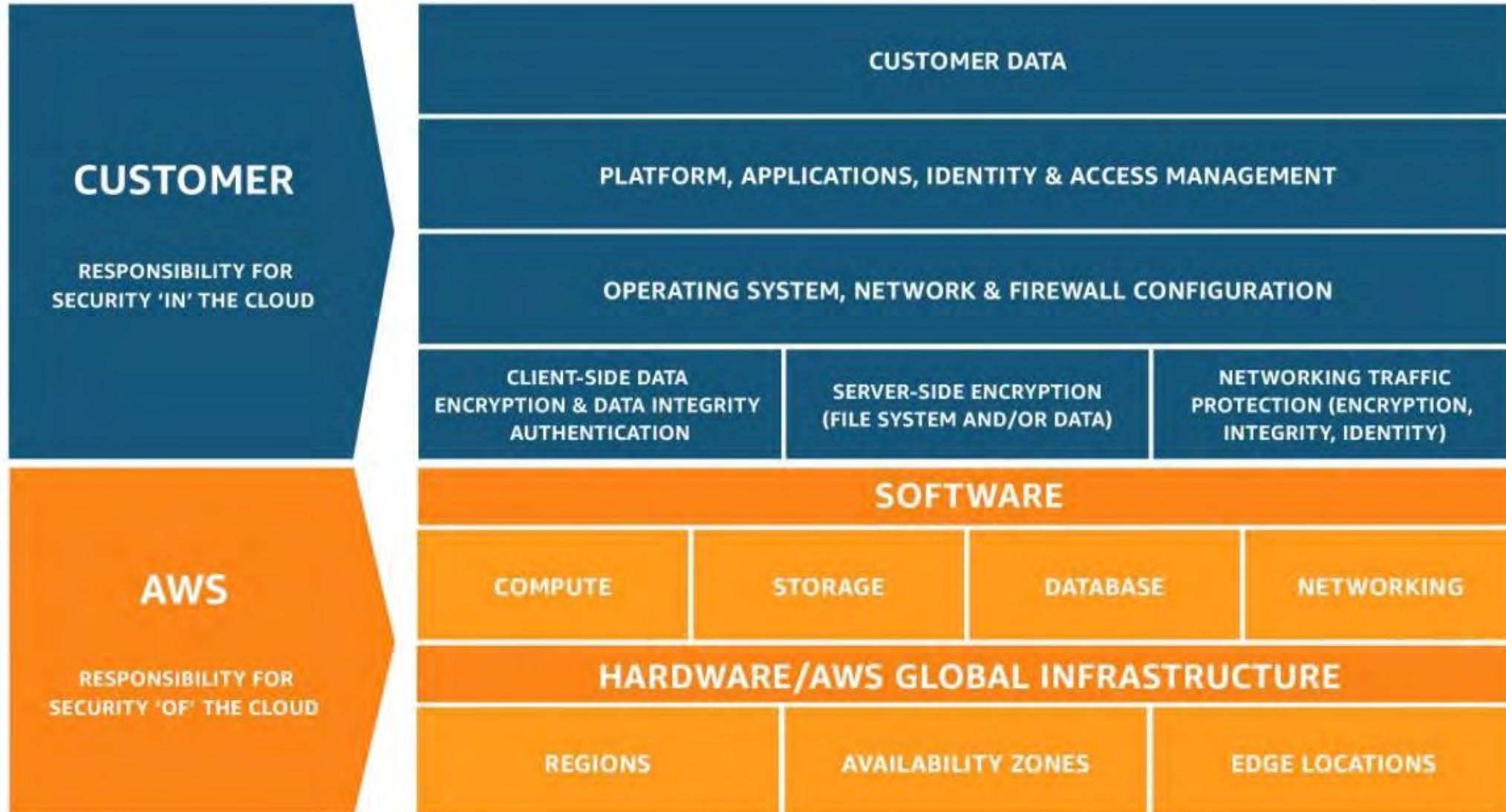
**Reliable**



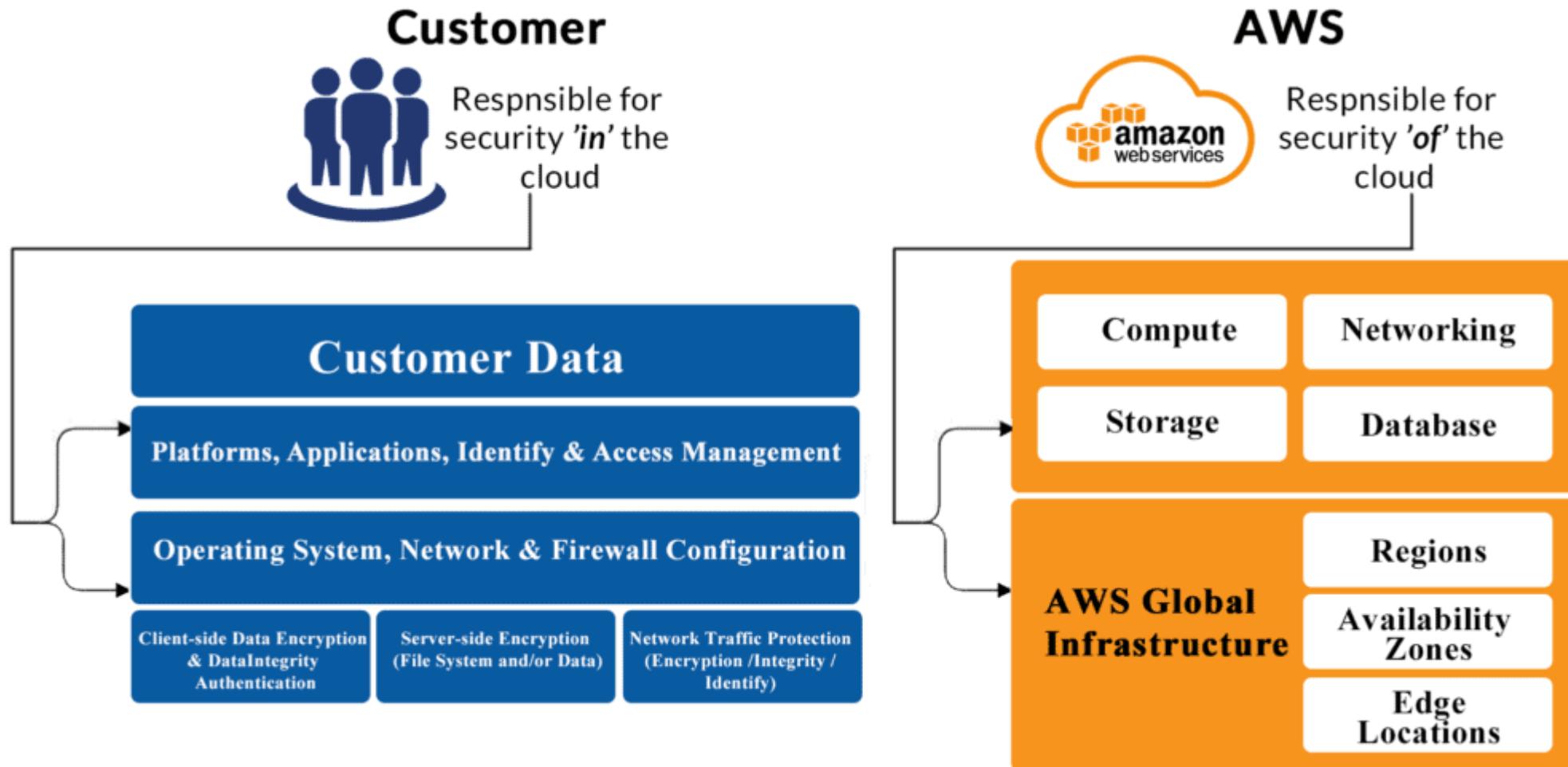
**Secure**



# What is Shared Responsibility Model?



# What is Shared Responsibility Model?



# AWS Security Controls

---

AWS provides a variety of security controls to help customers secure their data and information. These measures include access controls, encryption, threat detection, and security management tools.

## Access Controls



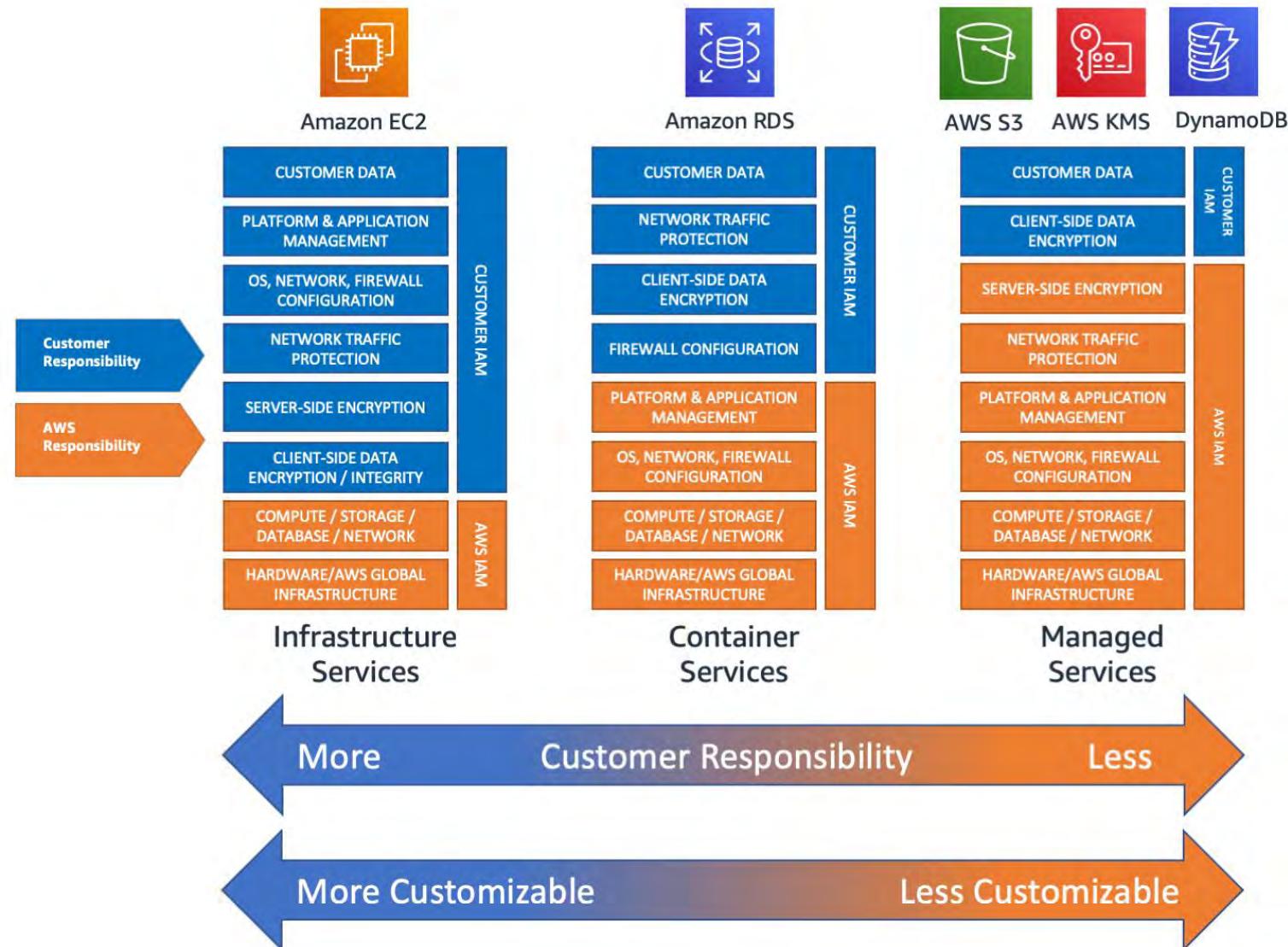
## Encryption



## Threat Detection



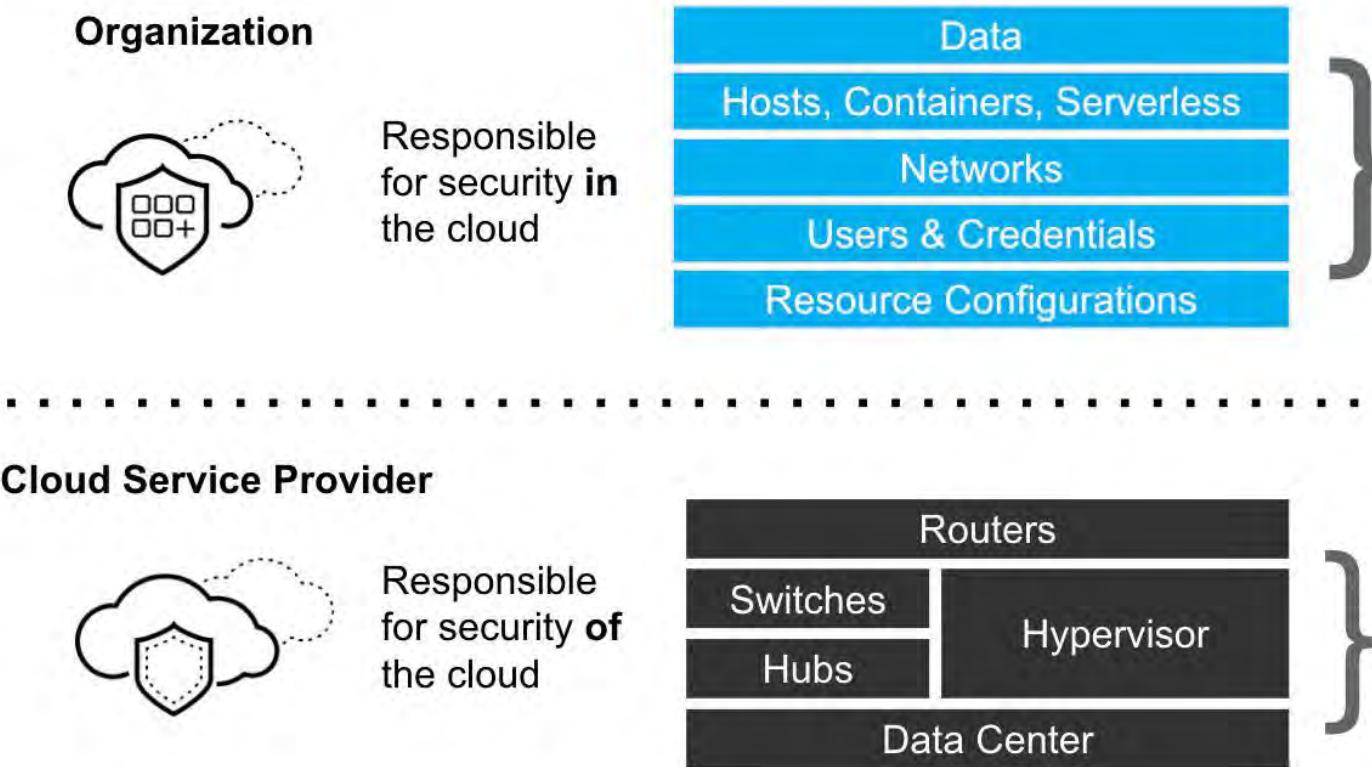
# Your Responsibility in Shared Responsibility Model



# Benefits of Shared Responsibility Model

The benefits of a shared responsibility model include:

- **1. AWS focus on Security, 2. Better Compliance, 3. Transparency and Collaboration**



# AWS Well-Architected Framework

As technology becomes more integrated into our daily lives, it is essential to understand security best practices. The AWS Well-Architected Framework provides a comprehensive guide to secure cloud computing.



# AWS Best Practices for Security

---

## Identity and Access Management



Users



Roles



Groups



Policies

## Encryption and Data Protection



## Network Security

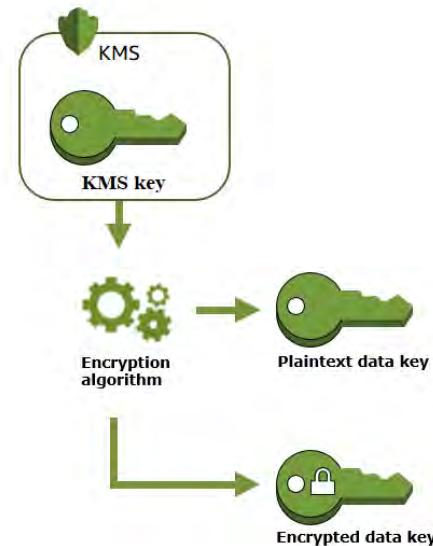


# Data Encryption in AWS

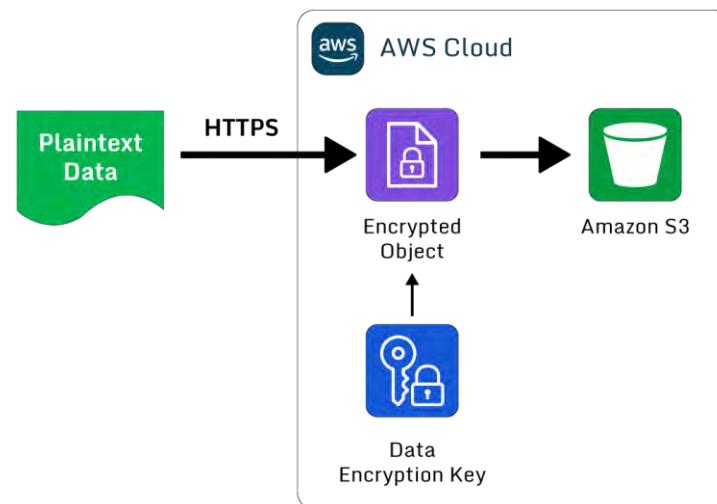
AWS offers a range of encryption options to secure your data, both in transit and at rest. Let's dive in!

## Encryption at Rest

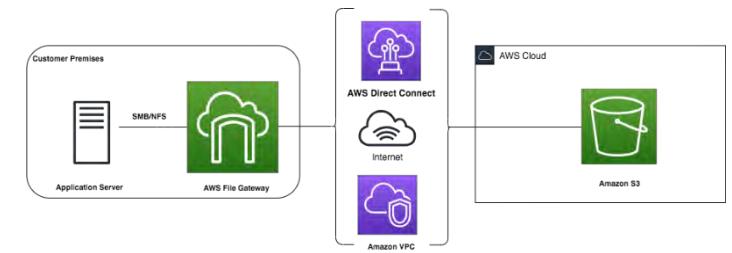
### AWS Key Management Service (KMS)



### Server-Side Encryption (SSE)



### AWS Storage Gateway



# AWS Key Management Service (KMS)

---

## Create Encryption Keys

Create, manage, and retain complete control over your encryption keys that protect your data stored in AWS.

## Manage Key Policies

Set and control access to your encryption keys according to your organizational compliance policies.

## Integrated with Other AWS Services

Use KMS keys for encrypting Amazon EBS volumes, Amazon S3 objects, Amazon RDS databases, and more.

# KMS and ACM Integration

---

## KMS Automated Certificate Management (ACM)

KMS ACM enables you to manage your SSL/TLS certificates and services that use HTTPS with AWS services that support ACM.

## SSL/TLS Key Management with KMS

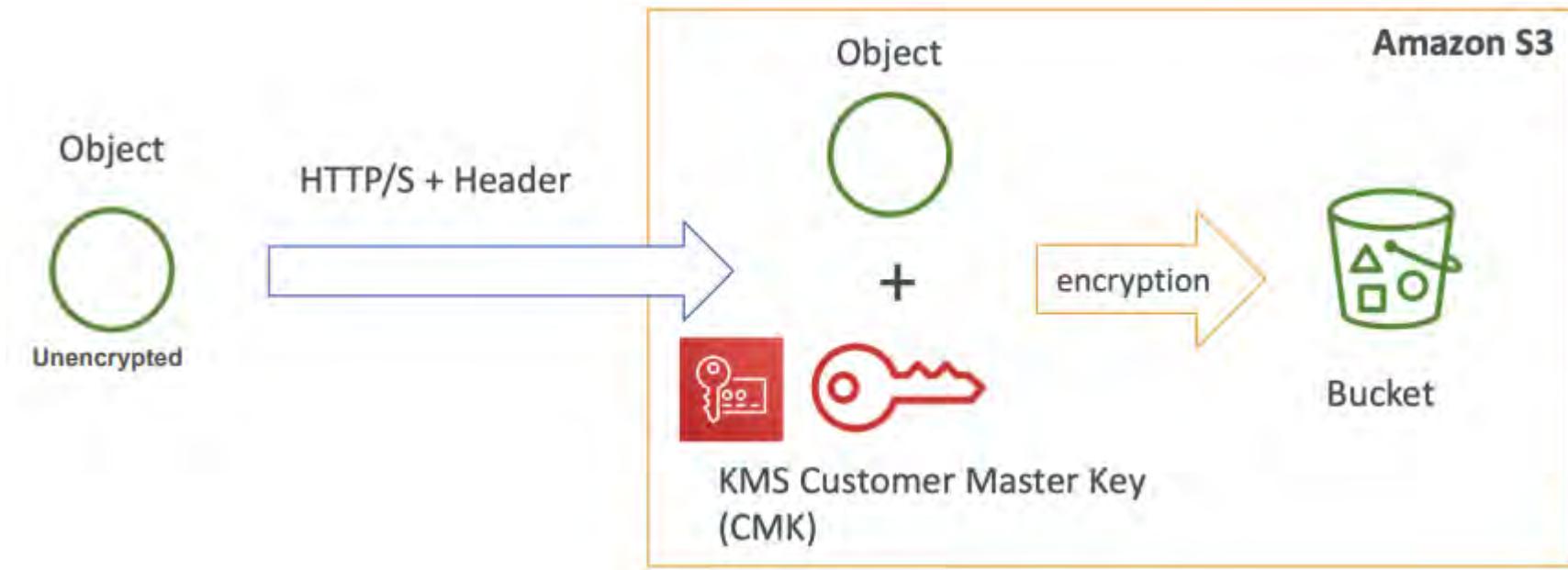
KMS enables you to control your keys and Certificates used for SSL/TLS, and automatically renew certificates

## Data Encryption Best Practices

- Design for Security
- Control Access to Encryption keys

# KMS key Practical Demonstration

Creating CMK and assigning to a user for encrypting S3 bucket



iamneo



# Amazon RDS

 [www.iamneo.ai](http://www.iamneo.ai)

# Introduction to Amazon RDS



- Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud.
- It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

# Benefits of RDS



# Benefits of RDS

---

- Availability
- Scalability
  - Vertical Scalability /Scaling Up
  - Horizontal Scalability /Scaling Out
- Performance
- Backup

# Creating RDS database instances using the AWS Management Console

---

- You can create a DB instance by using the AWS Management Console with **Easy create** enabled or not enabled.
- With **Easy create** enabled, you specify only the DB engine type, DB instance size, and DB instance identifier.
- **Easy create** uses the default setting for other configuration options.
- With **Easy create** not enabled, you specify more configuration options when you create a database, including ones for availability, security, backups, and maintenance.



# Creating RDS database instances using CLI

Alternatively, With AWS CLI, developers can create, edit, and delete Amazon RDS DB instances, snapshots, and security groups using the command line instead of the AWS Management Console.

To create a DB instance by using the AWS CLI, call the `create-db-instance` command with the following parameters:

- db-instance-identifier
- db-instance-class
- vpc-security-group-ids
- db-subnet-group
- engine
- master-username
- master-user-password
- allocated-storage
- backup-retention-period

# Understanding Amazon RDS Database Engines



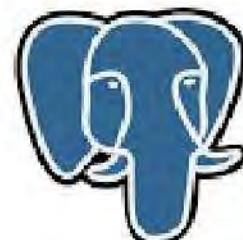
**Amazon RDS**

- MySQL is the world's most popular open-source relational database and Amazon RDS makes it easier to set up, operate, and scale MySQL deployments in the cloud.
- With Amazon RDS, you can deploy scalable MySQL servers in minutes with cost-efficient and resizable hardware capacity.
- Amazon RDS for MySQL frees you up to focus on application development by managing time-consuming database administration tasks, including backups, upgrades, software patching, performance improvements, monitoring, scaling, and replication.

# Understanding Amazon RDS Database Engines



Amazon RDS



PostgreSQL

- PostgreSQL has become the preferred open-source relational database for many enterprise developers and startups, powering leading business and mobile applications.
- Amazon RDS makes it easier to set up, operate, and scale PostgreSQL deployments on the cloud. With Amazon RDS, you can deploy scalable PostgreSQL deployments in minutes with cost-efficient and resizable hardware capacity.
- Amazon RDS manages complex and time-consuming administrative tasks

# Understanding Amazon RDS Database Engines



**ORACLE**

## **Amazon RDS**

- Amazon RDS for Oracle is a fully managed commercial database that makes it easy to set up, operate, and scale Oracle deployments in the cloud.
- Amazon RDS frees you up to focus on innovation and application development by managing time-consuming database administration tasks, including provisioning, backups, software patching, monitoring, and hardware scaling.
- You can run Amazon RDS for Oracle under two different licensing models :
  - “License Included”
  - “Bring-Your-Own-License (BYOL)”.

# Understanding Amazon RDS Database Engines



**Amazon RDS**



Microsoft  
**SQL Server**

- SQL Server is a relational database management system developed by Microsoft.
- Amazon RDS for SQL Server makes it easy to set up, operate, and scale SQL Server deployments in the cloud.
- With Amazon RDS, you can deploy multiple editions of SQL Server (2014, 2016, 2017 and 2019), including Express, Web, Standard, and Enterprise, in minutes with cost-efficient and re-sizeable compute capacity.

# Understanding Amazon RDS Database Engines



**Amazon RDS**



**MariaDB**

- MariaDB is a popular open-source relational database created by the original developers of MySQL.
- Amazon RDS makes it easy to set up, operate, and scale MariaDB server deployments in the cloud.
- With Amazon RDS, you can deploy scalable MariaDB cloud databases in minutes with cost-efficient and resizable hardware capacity.

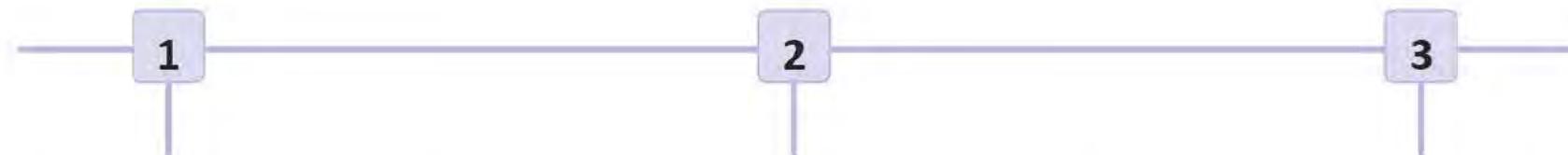
# Understanding Amazon RDS Database Engines



**Amazon** Aurora

- Amazon Aurora (Aurora) is a fully managed relational database engine that's compatible with MySQL and PostgreSQL.
- You already know how MySQL and PostgreSQL combine the speed and reliability of high-end commercial databases with the simplicity and cost-effectiveness of open-source databases.
- The code, tools, and applications you use today with your existing MySQL and PostgreSQL databases can be used with Aurora.

# Scaling and Monitoring Amazon RDS



## Scaling RDS Instances

Learn how to scale RDS instances for increased performance and capacity using manual or auto scaling techniques.

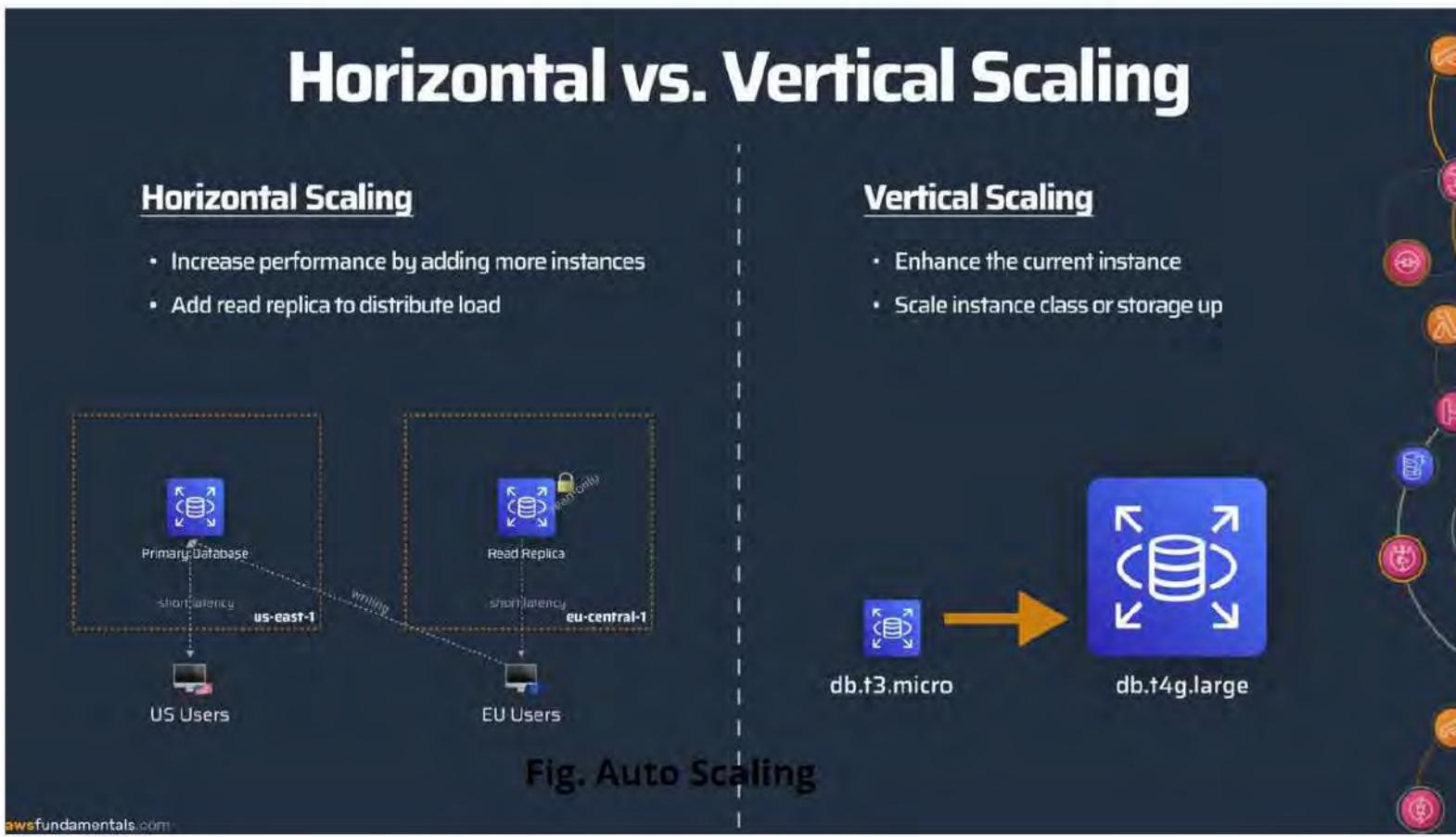
## Monitoring RDS performance

Measure the performance of your RDS instances using Amazon CloudWatch metrics and alarms and opt for automated monitoring as an alternative.

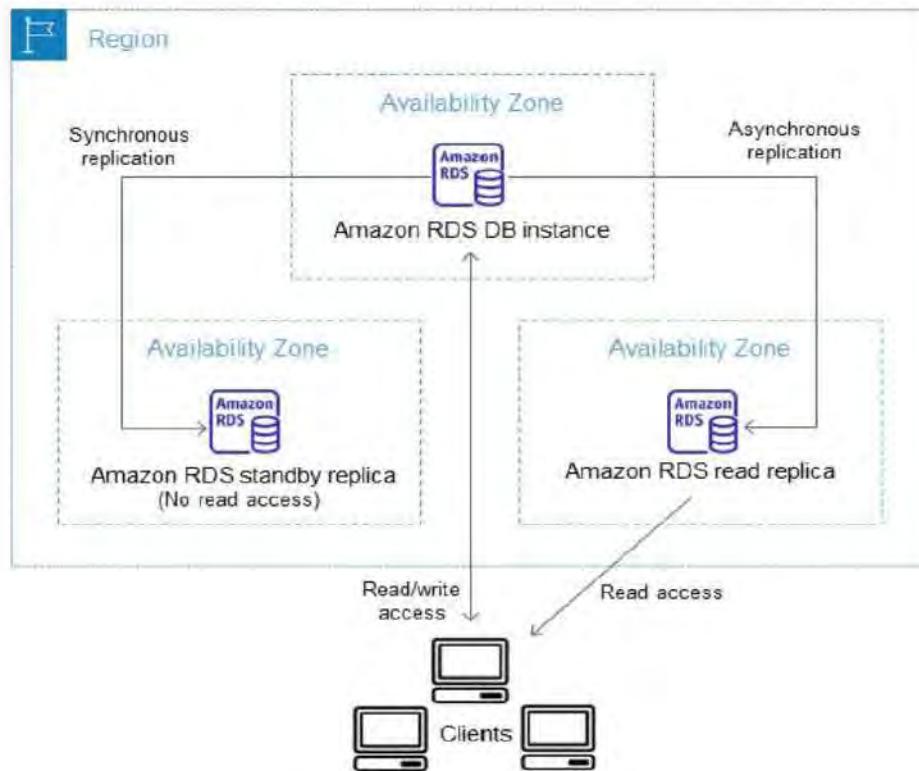
## Configuring auto scaling and read replicas

Learn how to configure auto scaling and read replicas for RDS instances for improved performance and durability.

# Auto Scaling for RDS Instances

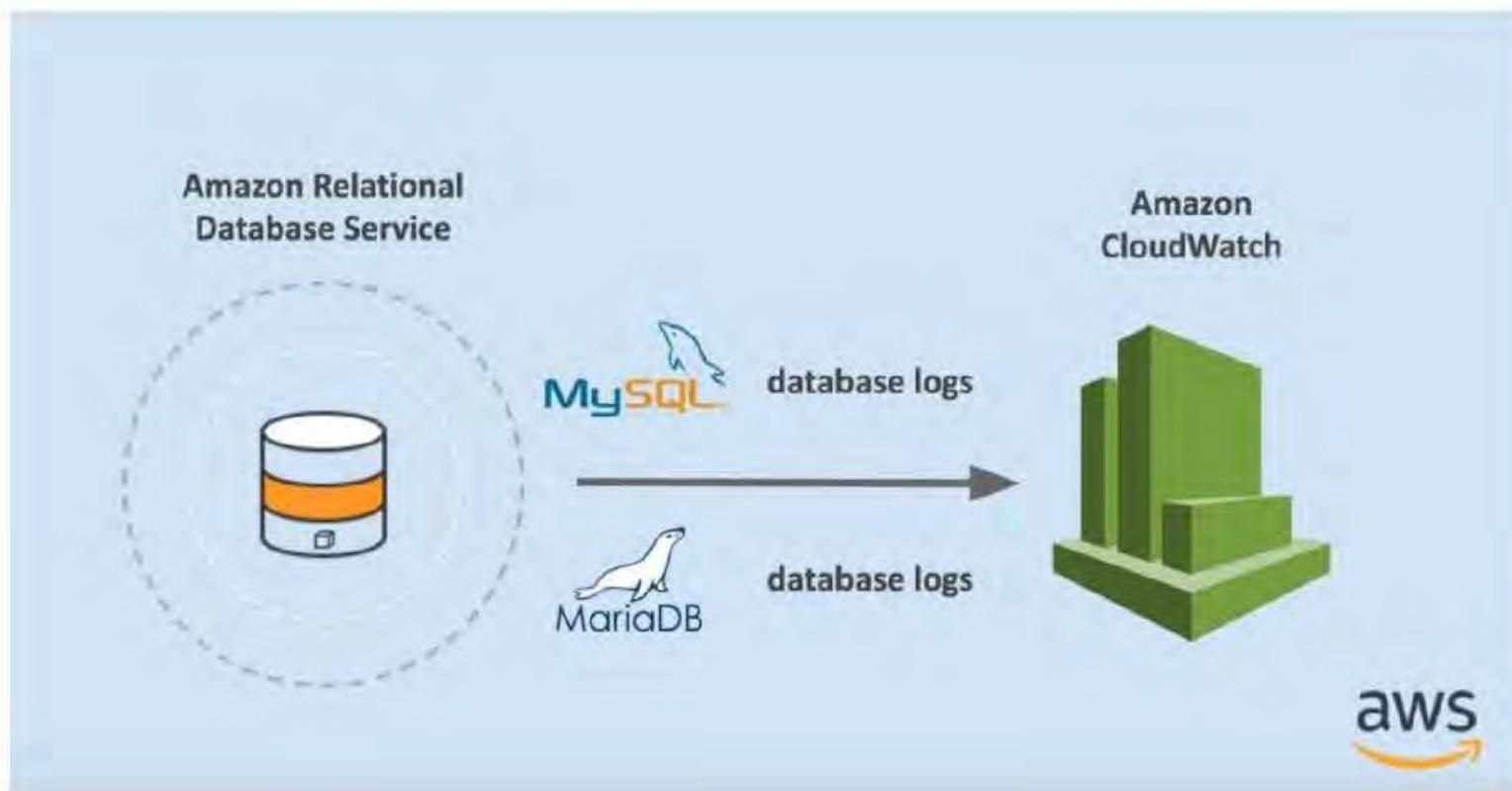


# Auto Read Replicas for RDS Instances



**Fig. Read Replicas**

# Understanding Amazon RDS Database Engines

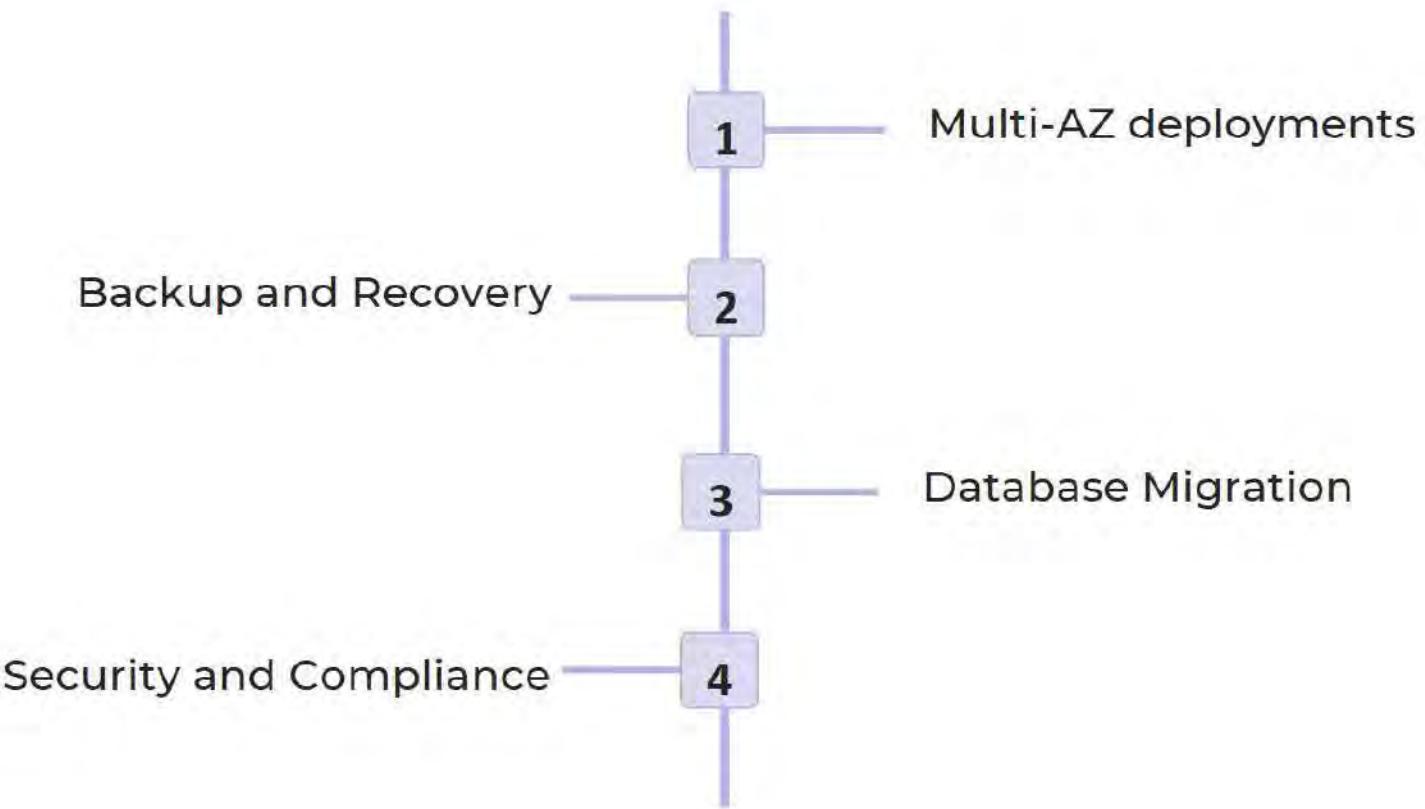


# **Understanding Amazon RDS Database Engines**

You can monitor DB instances using Amazon CloudWatch, which collects and processes raw data from Amazon RDS into readable, near real-time metrics. These statistics are recorded for a period of two weeks, so that you can access historical information and gain a better perspective on how your web application or service is performing.

By default, Amazon RDS metric data is automatically sent to CloudWatch in 1-minute periods.

# Utilizing RDS Features for High Availability and Durability



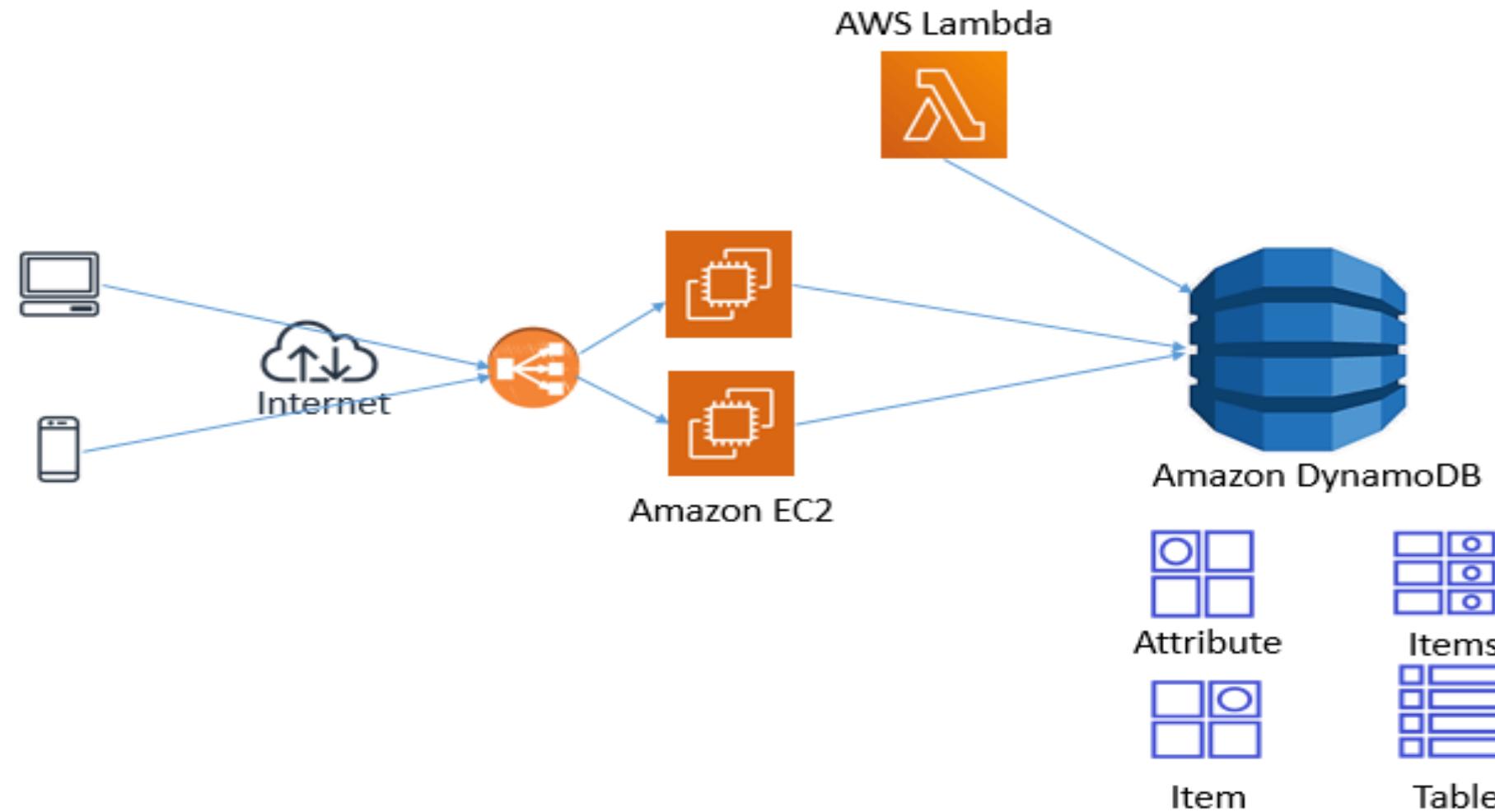
iamneo



# Amazon DynamoDB

---

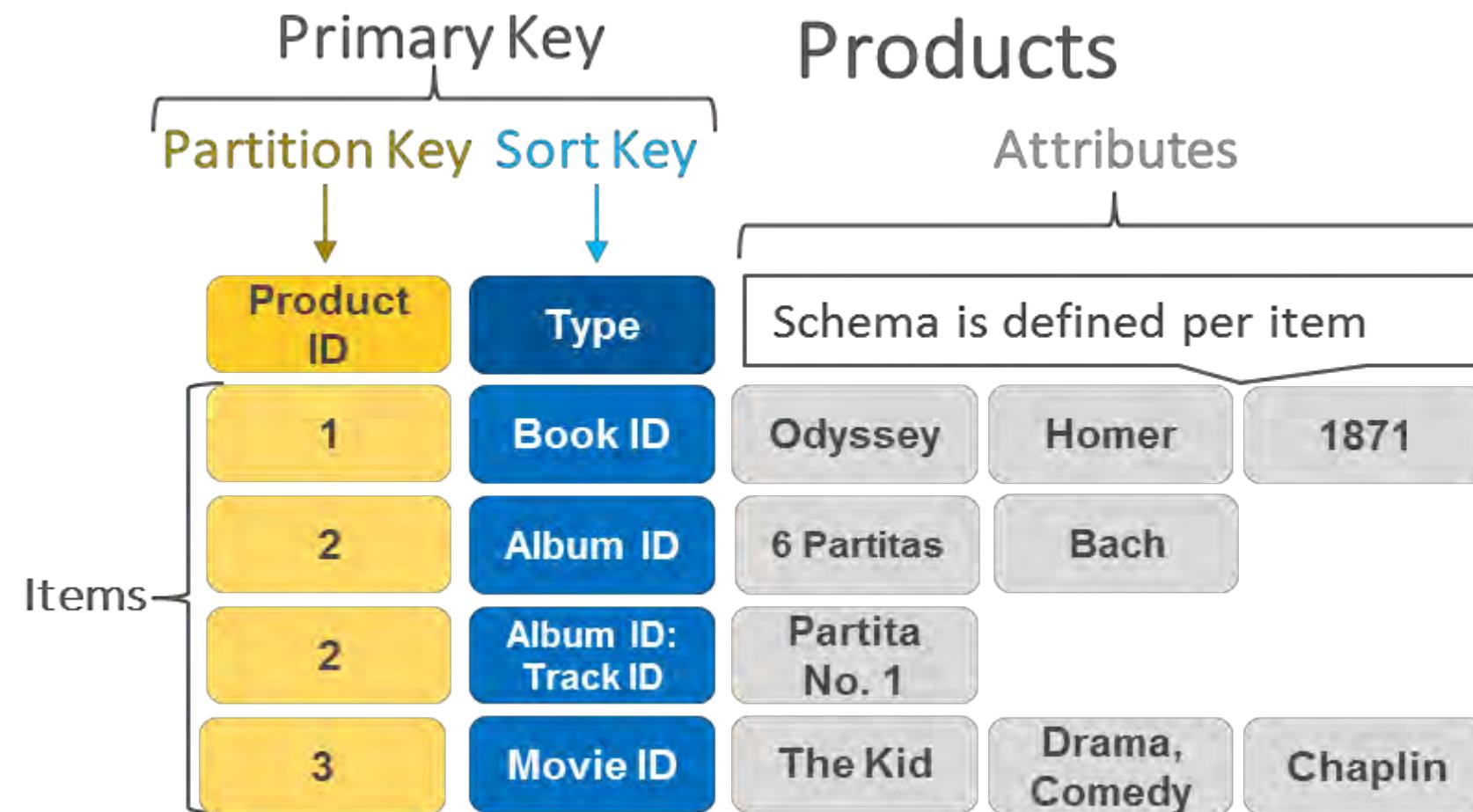
# Introduction to Amazon DynamoDB



# Introduction to Amazon DynamoDB

- AWS Dynamo DB is a No SQL Database which is built to support No SQL compatible database in cloud environment.
- AWS Dynamo DB is fully managed server less No SQL database service it means you do not need to take care of any server/infrastructure, AWS does take care it for you. Dynamo DB is highly scalable, available and durable database service. It is a key value pair database store.
- Dynamo DB can also store document.

# Key-Value Data Model and Benefits



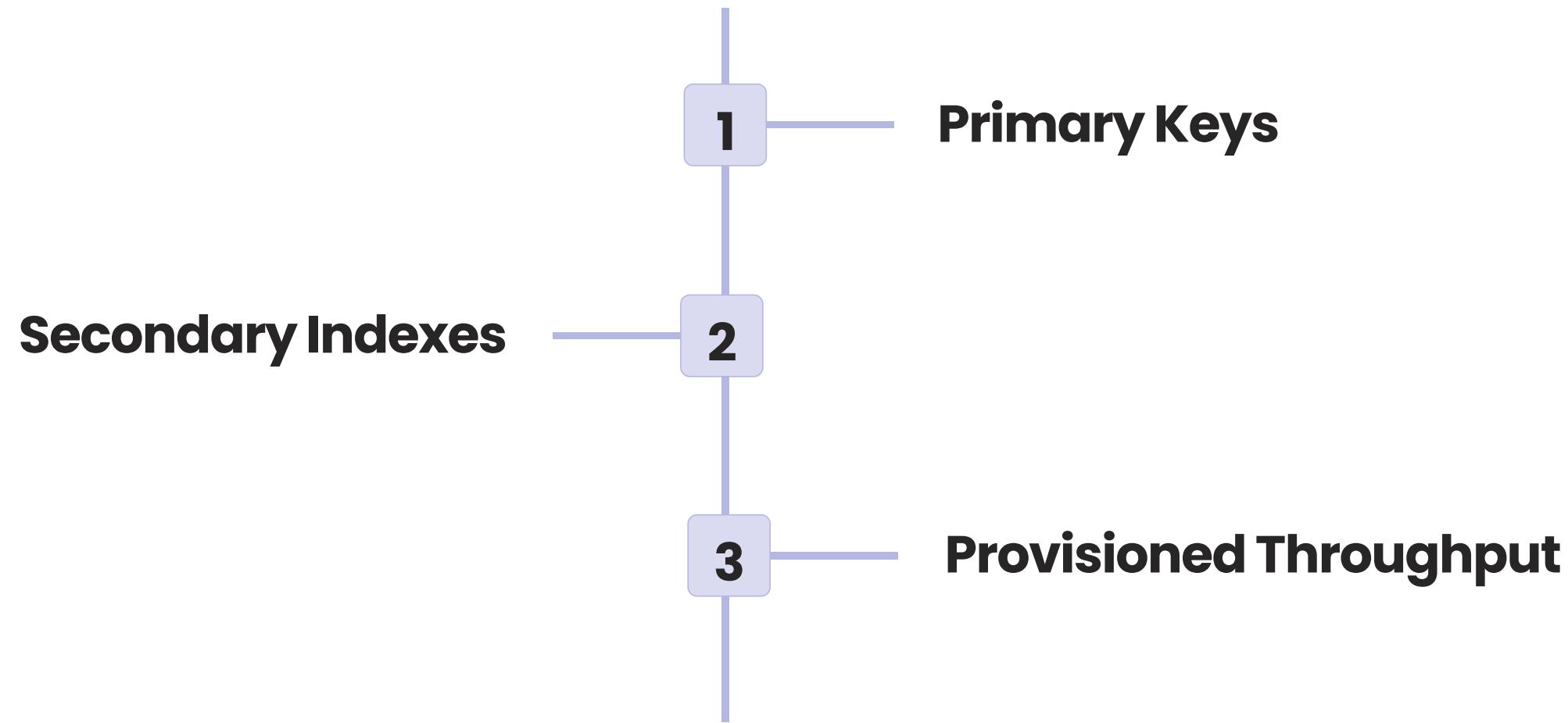
The primary key uniquely identifies each item in the table, and secondary indexes can be created.

# Benefits of DynamoDB

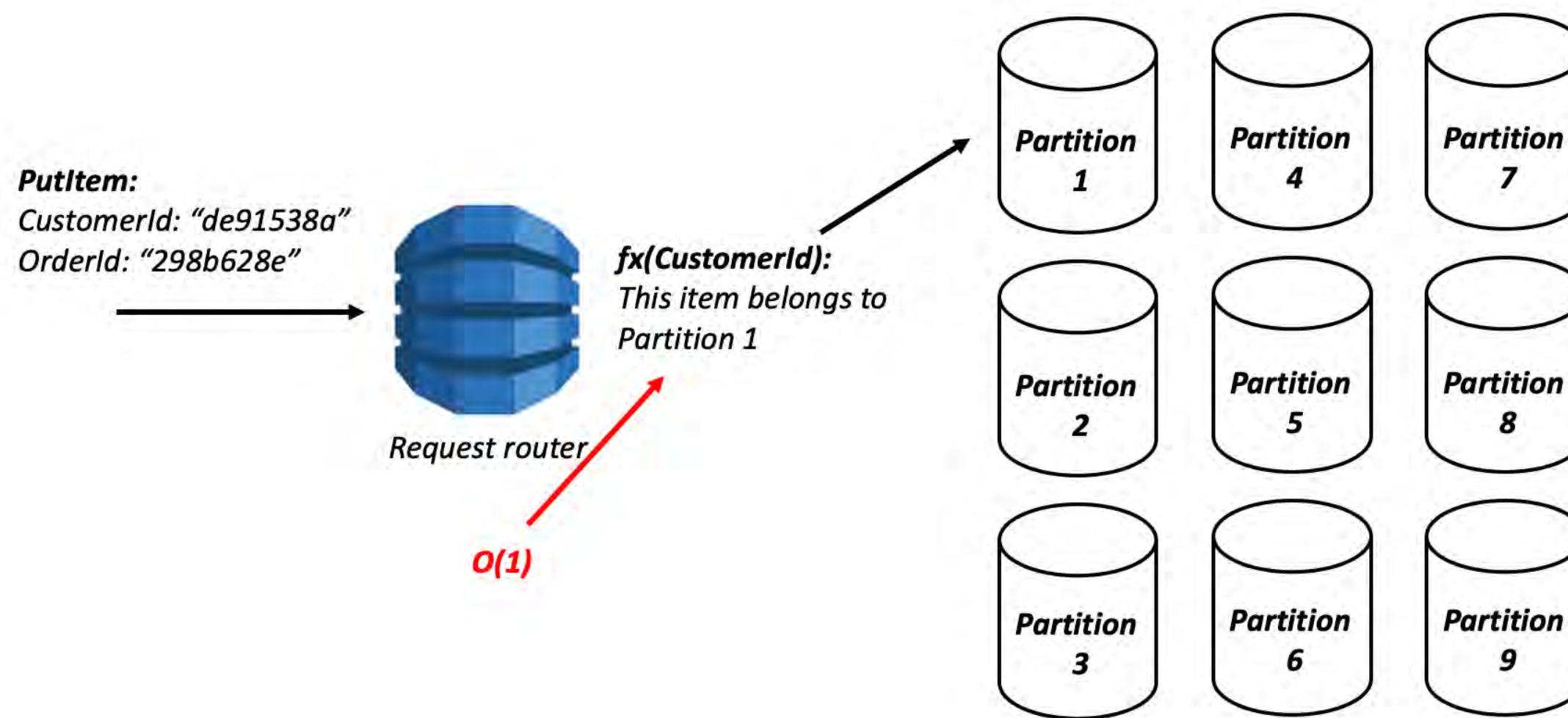
---

- High performance
- Fully managed
- On-demand pricing and automatic scaling
- Cost-effective
- Durable
- Highly available.

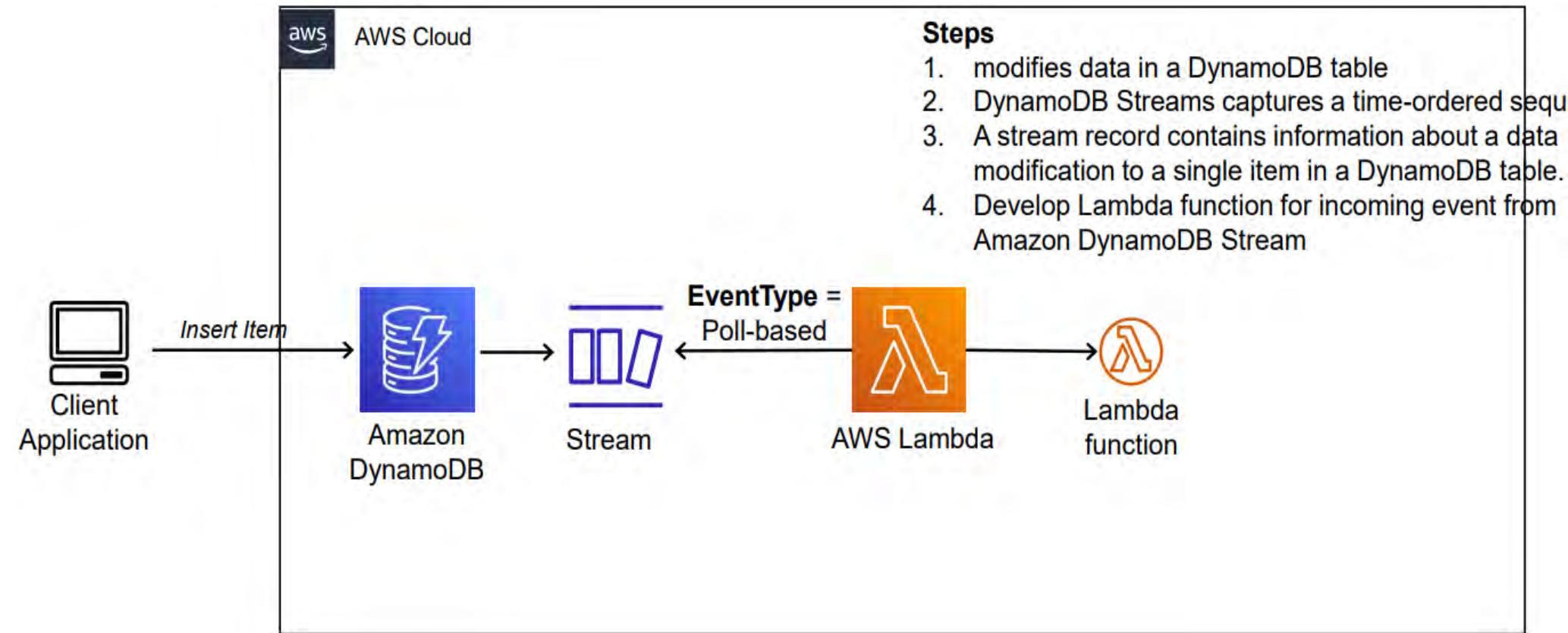
# Designing DynamoDB Tables



# Designing DynamoDB tables with proper primary keys



# Using DynamoDB Streams for capturing real-time data changes



DynamoDB Streams – Using AWS Lambda to Process DynamoDB Streams for Change Data Capture of DynamoDB Tables.

# Best Practices for Modeling Data in DynamoDB

---

Start with a clear understanding of your application

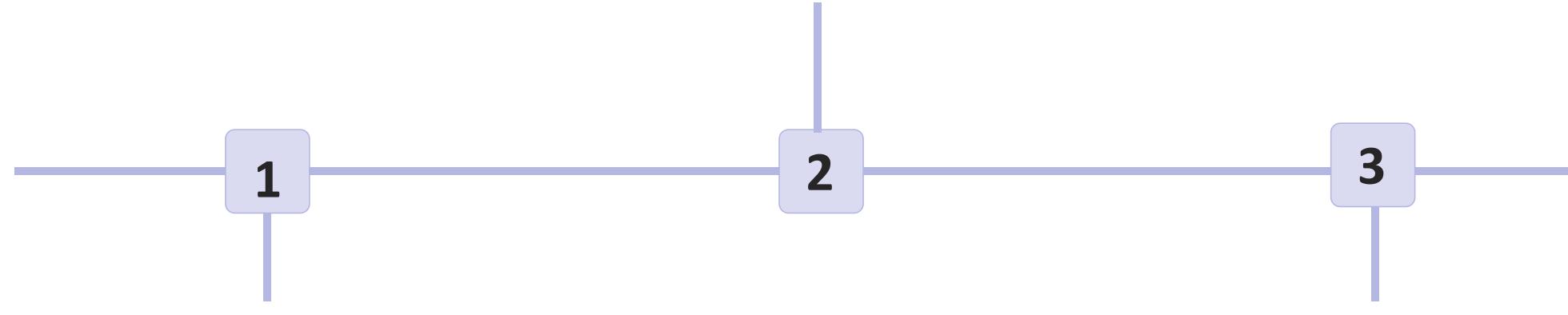
Use maximum capacity units efficiently

Optimize your table structure

Keep an eye on your costs

# Querying DynamoDB

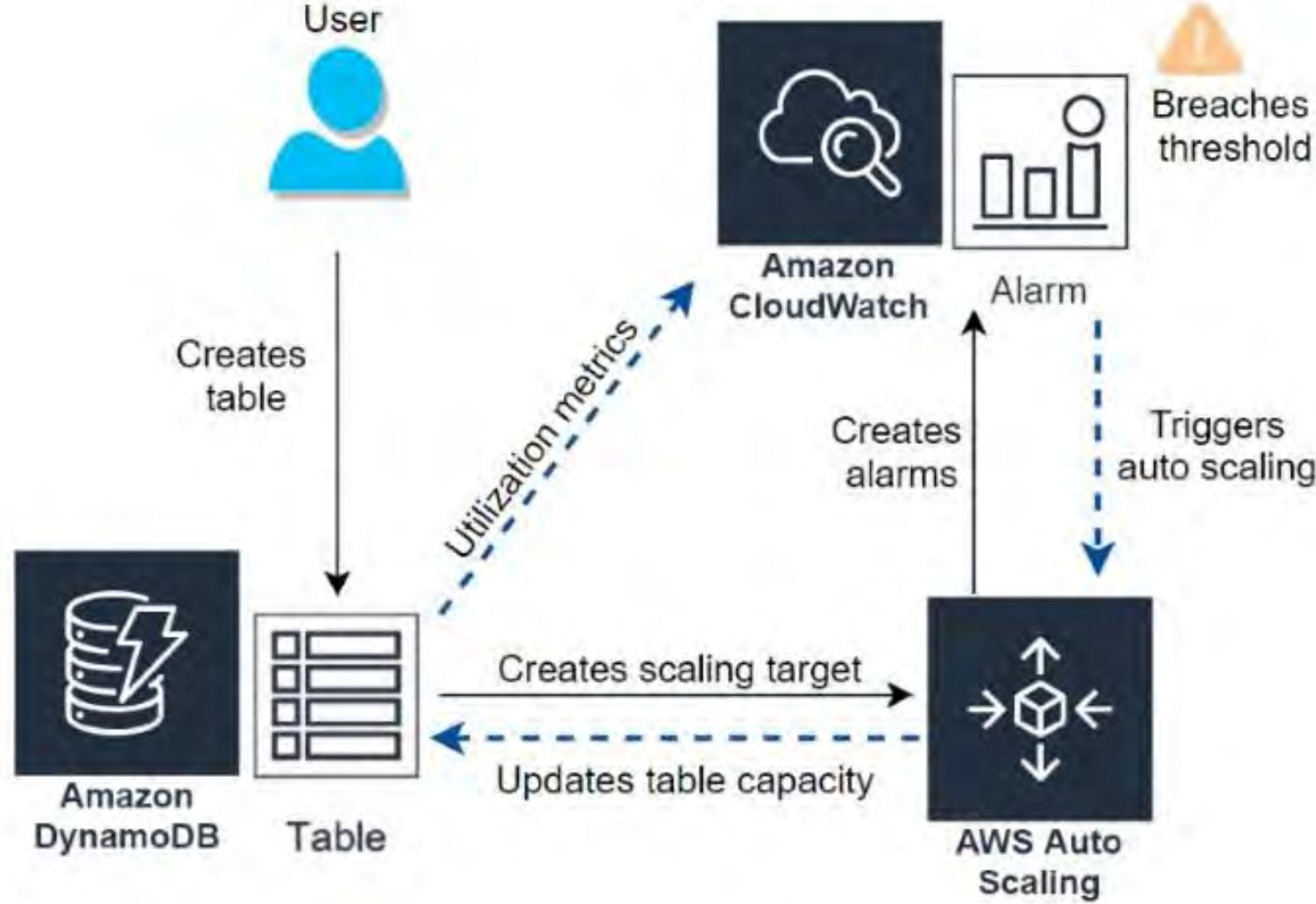
## Range-based Operations



**Key-based Operations**

**Filter Operations**

# Scaling DynamoDB



- **Partitioning**
- **Global Secondary Indexes**
- **Auto Scaling**

# Performing CRUD Operations in DynamoDB

---

## 1 Create

Use PutItem to add a new item to your table, specifying its partition key and any additional attributes.

## 2 Read

Use GetItem and BatchGetItem to retrieve specific items by their primary keys.

## 3 Update

Use UpdateItem to modify an existing item or create a new one if it doesn't exist, specifying the attributes to update or delete.

## 4 Delete

Use DeleteItem and BatchWriteItem to remove items from your table, specifying their primary keys.

# Querying Data in DynamoDB

---

- 1
- 2

## **Query Operations**

Use Query to retrieve specific items by their partition and sort key values, and filter by any additional attributes.

## **Scan Operations**

Use Scan to retrieve all items in a table or a specific set of items, and filter by any additional attributes.

# Configuring auto scaling for DynamoDB tables based on capacity needs

Create DynamoDB table Tutorial ?

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name\*  ⓘ

Primary key\* Partition key  String ⓘ

Add sort key

Table settings

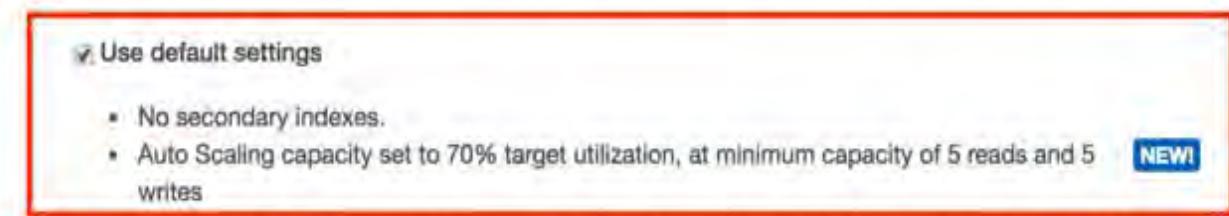
Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

Use default settings

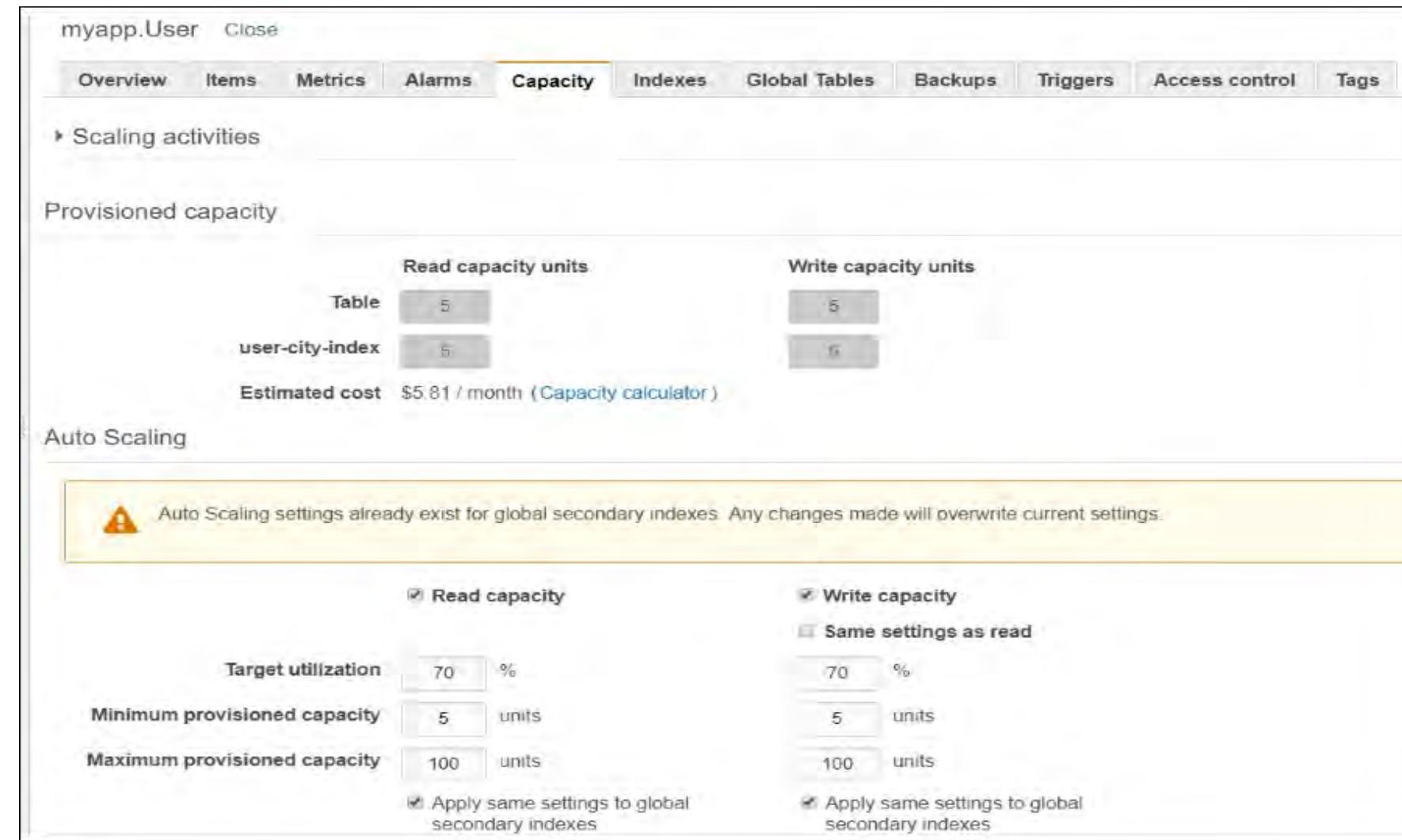
- No secondary indexes.
- Auto Scaling capacity set to 70% target utilization, at minimum capacity of 5 reads and 5 writes NEW!

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

Cancel Create

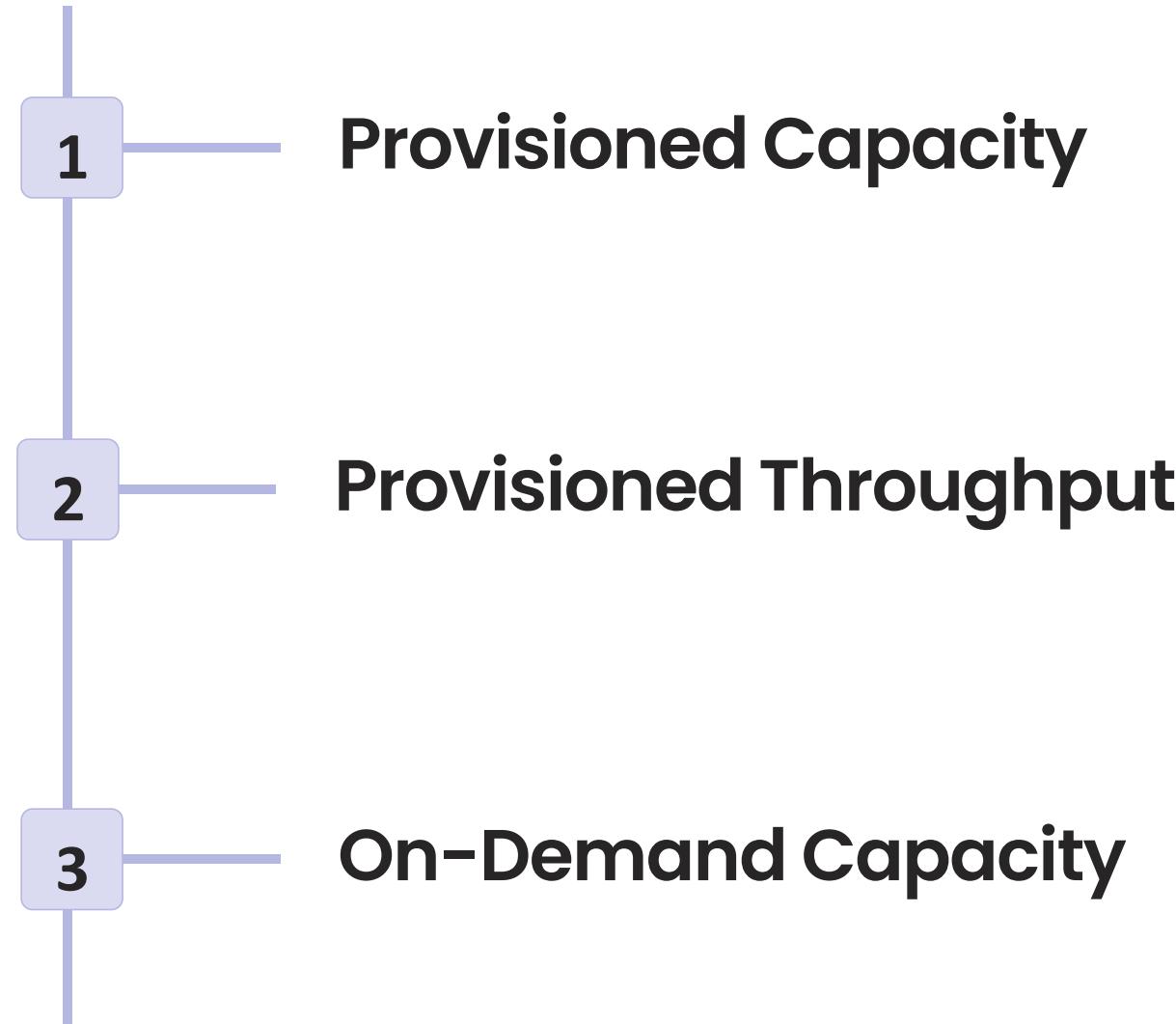


# Configuring auto scaling for DynamoDB tables based on capacity needs



# Capacity Needs and Scaling Strategies

---



# Utilizing Global Tables for Multi-Region Replication

---

## What are Global Tables?

Global Tables enable automatic replication of DynamoDB tables across multiple AWS Regions.

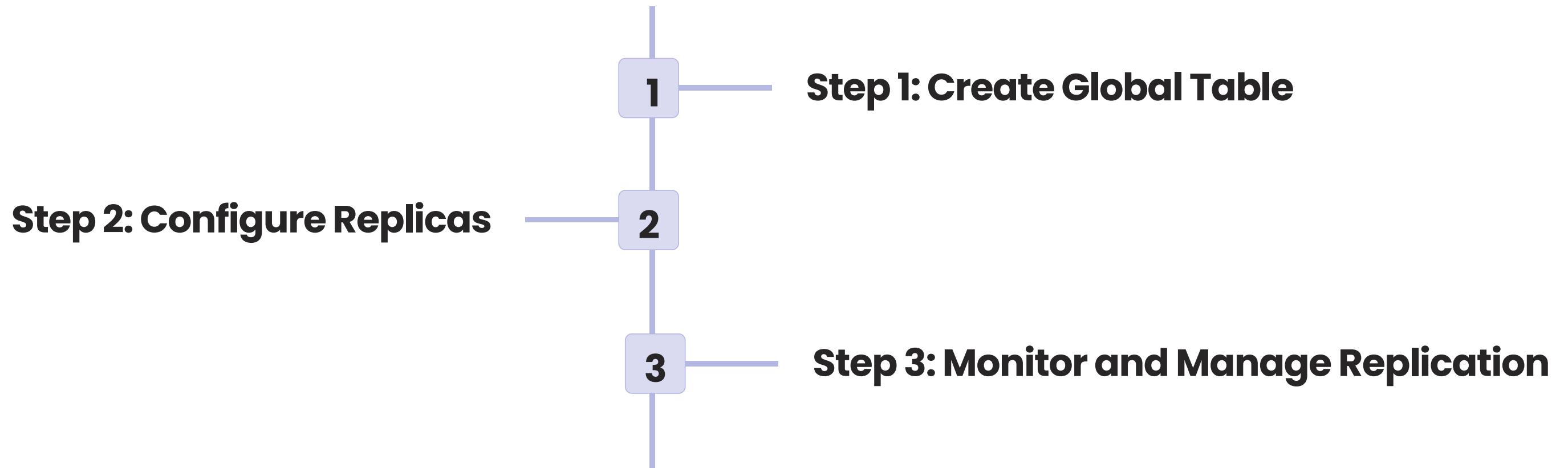
## Benefits of Multi-Region Replication

It helps prevent data loss from disasters, provides low-latency data access, and enables global reads and writes.

## Global Scalability with Global Tables

Leverage Global Tables for scalable, multi-region access to your DynamoDB tables.

# How to Configure Global Tables



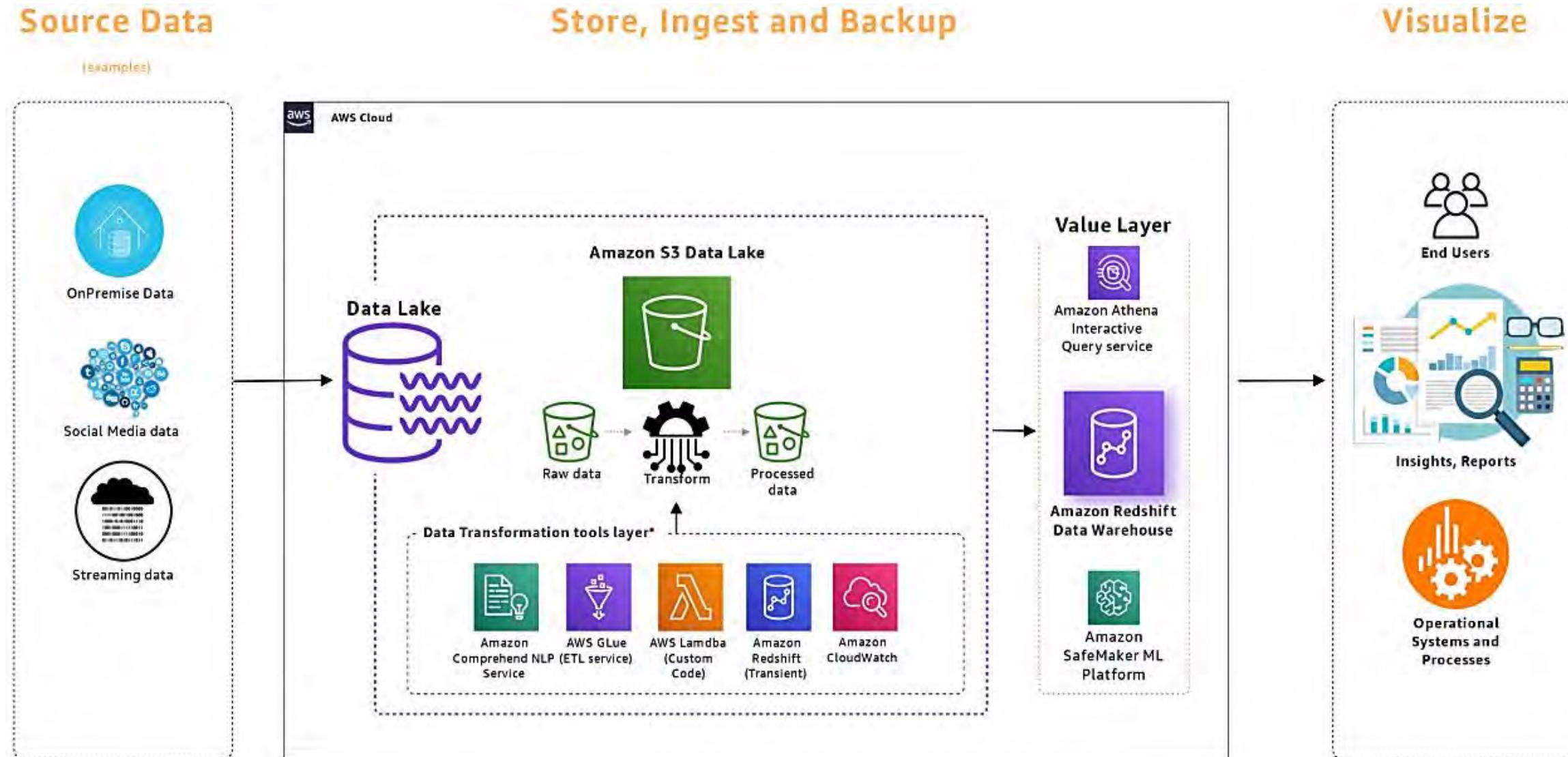
iamneo



# Amazon Redshift

---

# Introduction to Amazon Redshift



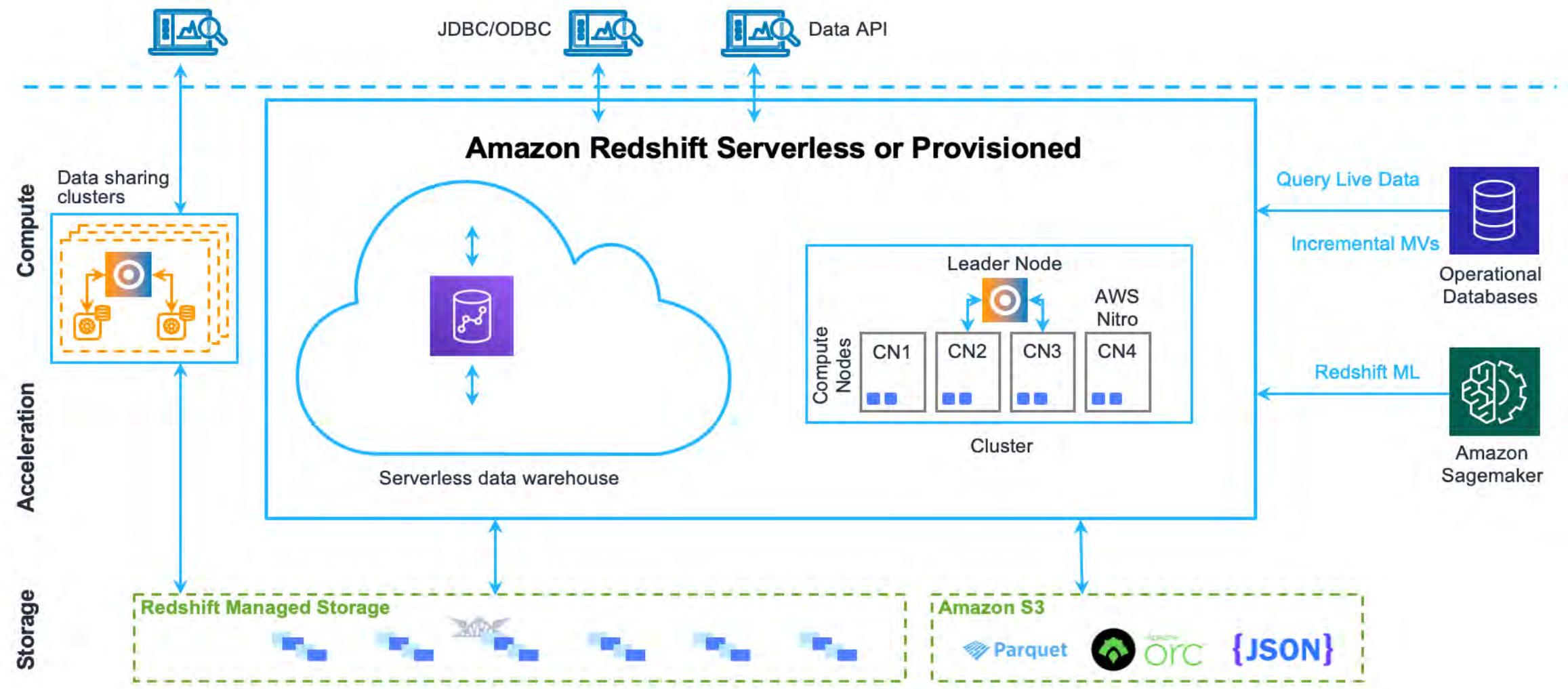
# Introduction to Amazon Redshift

---

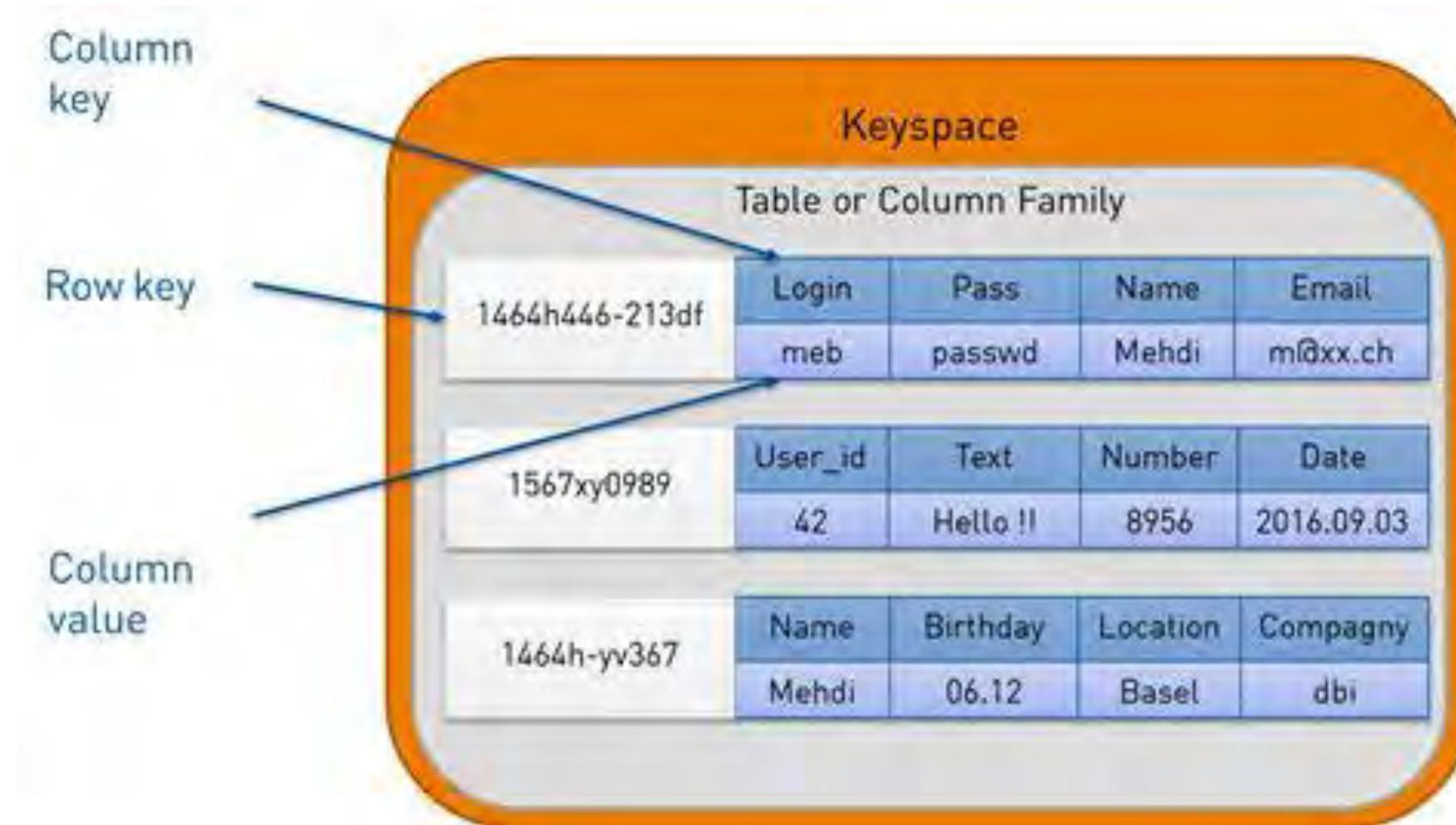
- Amazon Redshift is a powerful cloud data warehousing service equipped with the latest technology to store big data and analytics.
- It's a fully managed service that runs complex analytic queries on large data sets... and it's super easy to use!
- Amazon Redshift Serverless lets you access and analyze data without the usual configurations of a provisioned data warehouse.
- Resources are automatically provisioned, and data warehouse capacity is intelligently scaled to deliver fast performance for even the most demanding and unpredictable workloads.

# Fully managed data warehouse service

This figure introduces the elements of the Amazon Redshift data warehouse architecture



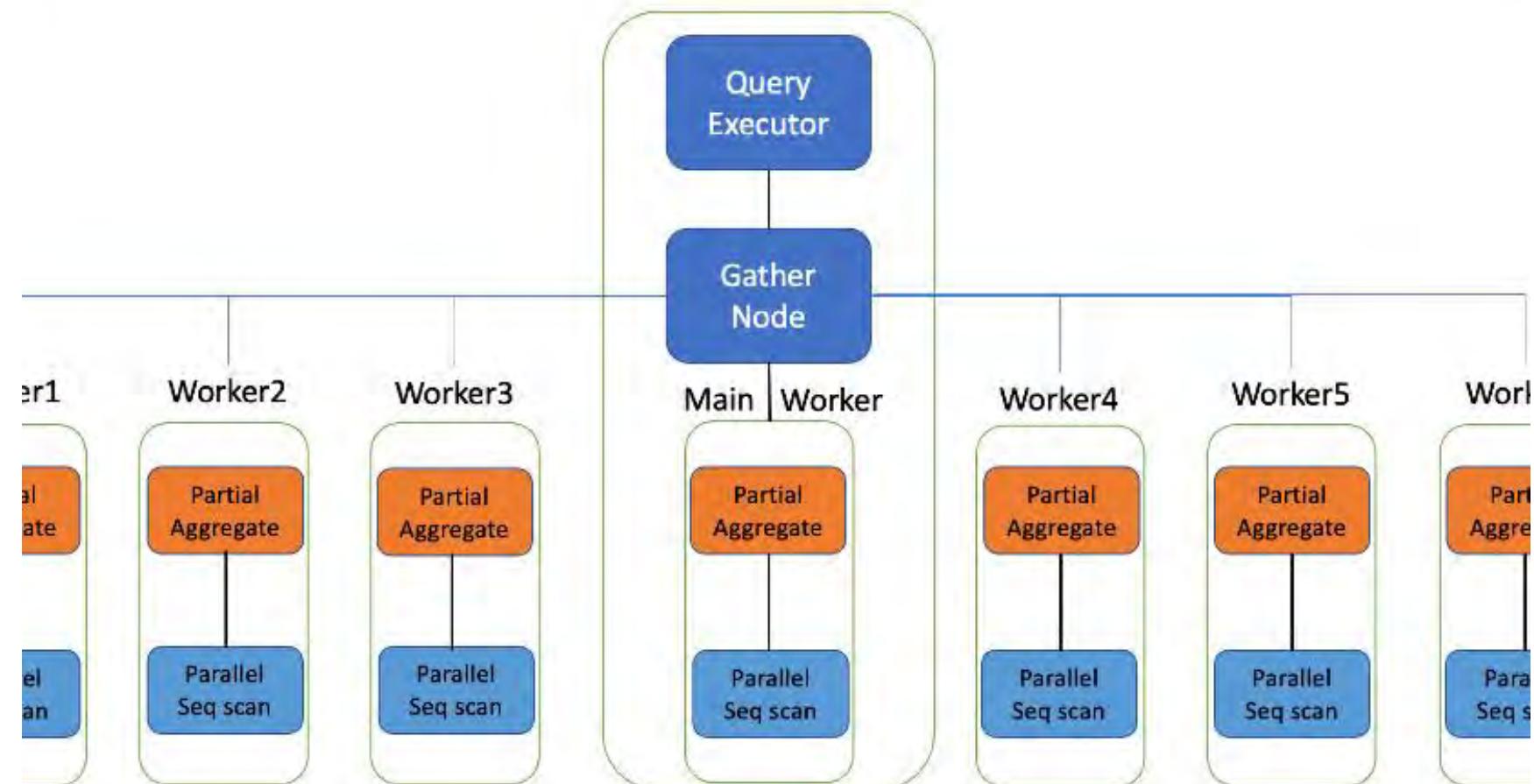
# Columnar Storage and Parallel Query Processing



## Columnar Storage

Redshift uses a columnar storage technique that provides high-speed performance and efficiency by only selecting the columns it needs to perform the query.

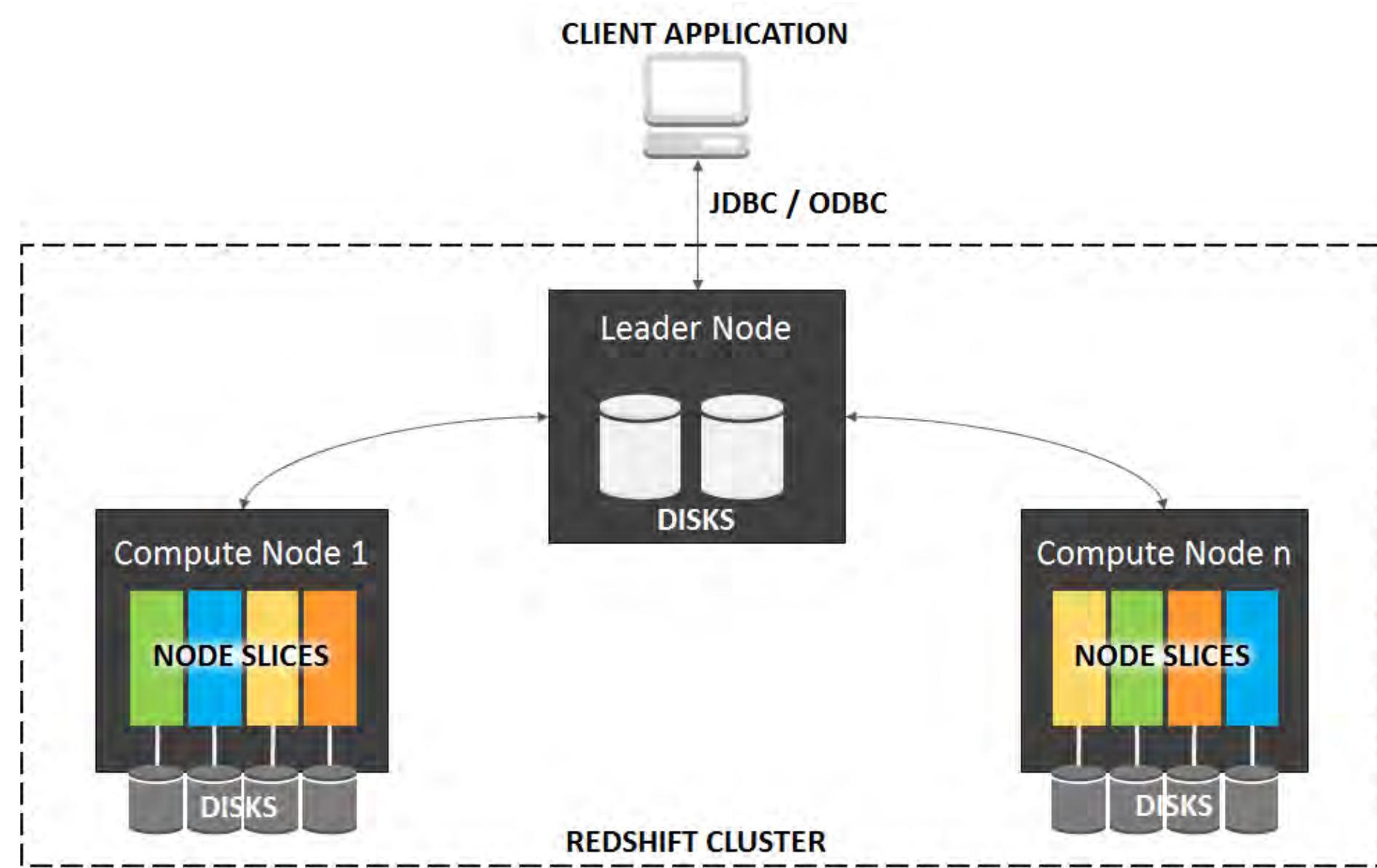
# Columnar Storage and Parallel Query Processing



## Parallel Query Processing

Redshift utilizes parallel processing to execute complex queries involving multiple nodes to quickly provide results in seconds.

# Creating and Configuring Amazon Redshift Clusters



# Creating and Configuring Amazon Redshift Clusters

---

Some of the essential concepts and terminologies that you need to keep in mind when working with Amazon Redshift:

- **Clusters**
  - Leader node
  - Compute node
- **Node slices**
- **Databases**

# Creating a Cluster

---

1

## Step-by-Step Guide

Creating an Amazon Redshift cluster can be done with just a few clicks. By following the AWS Management Console's step-by-step guide, you can have your cluster up and running in no time!

2

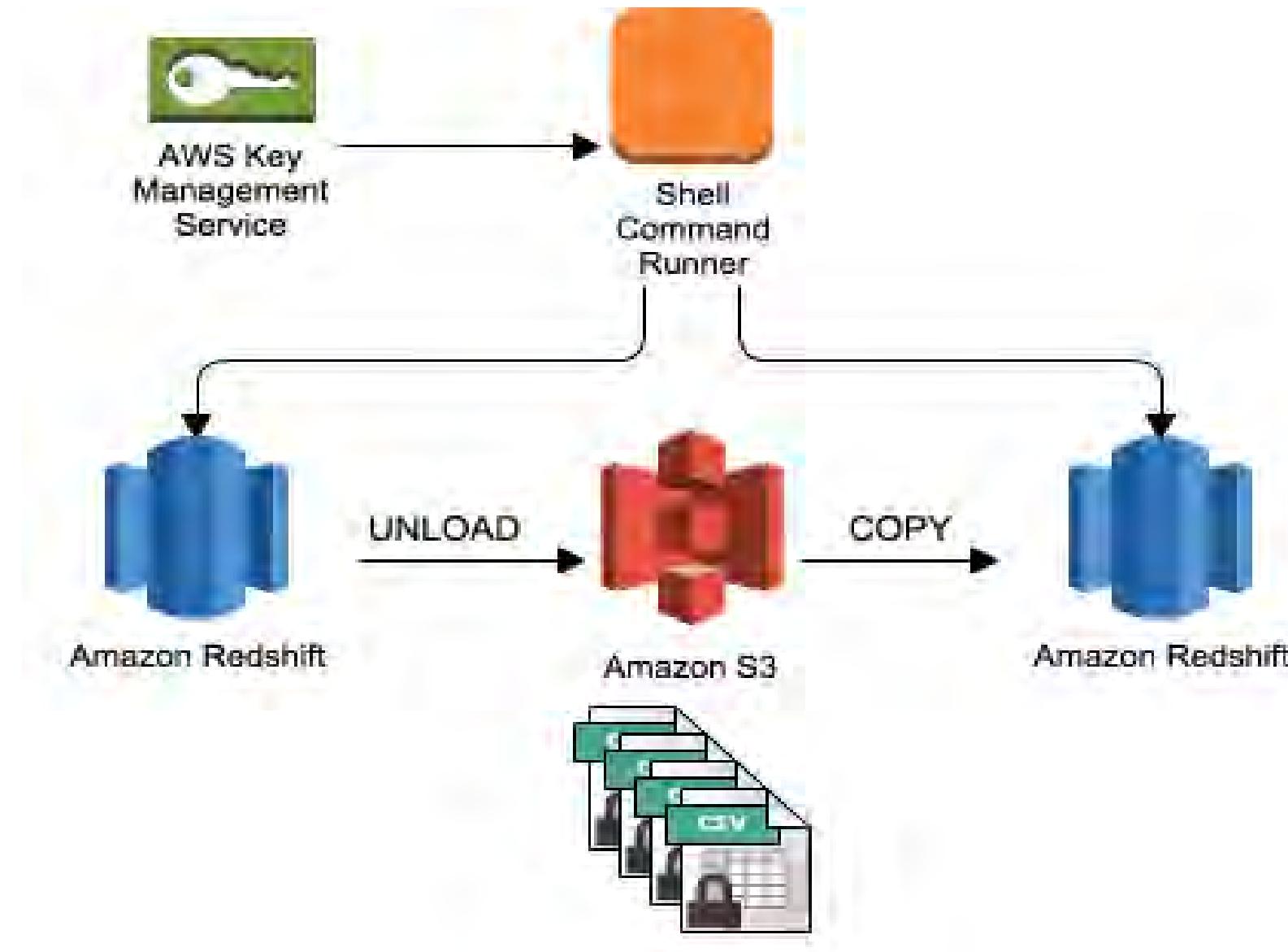
## Cli or SDK

For more complex production environments, the AWS Management Console offers a Command Line Interface (CLI), as well as Software Development Kits (SDKs), for scripting Amazon Redshift service tasks.

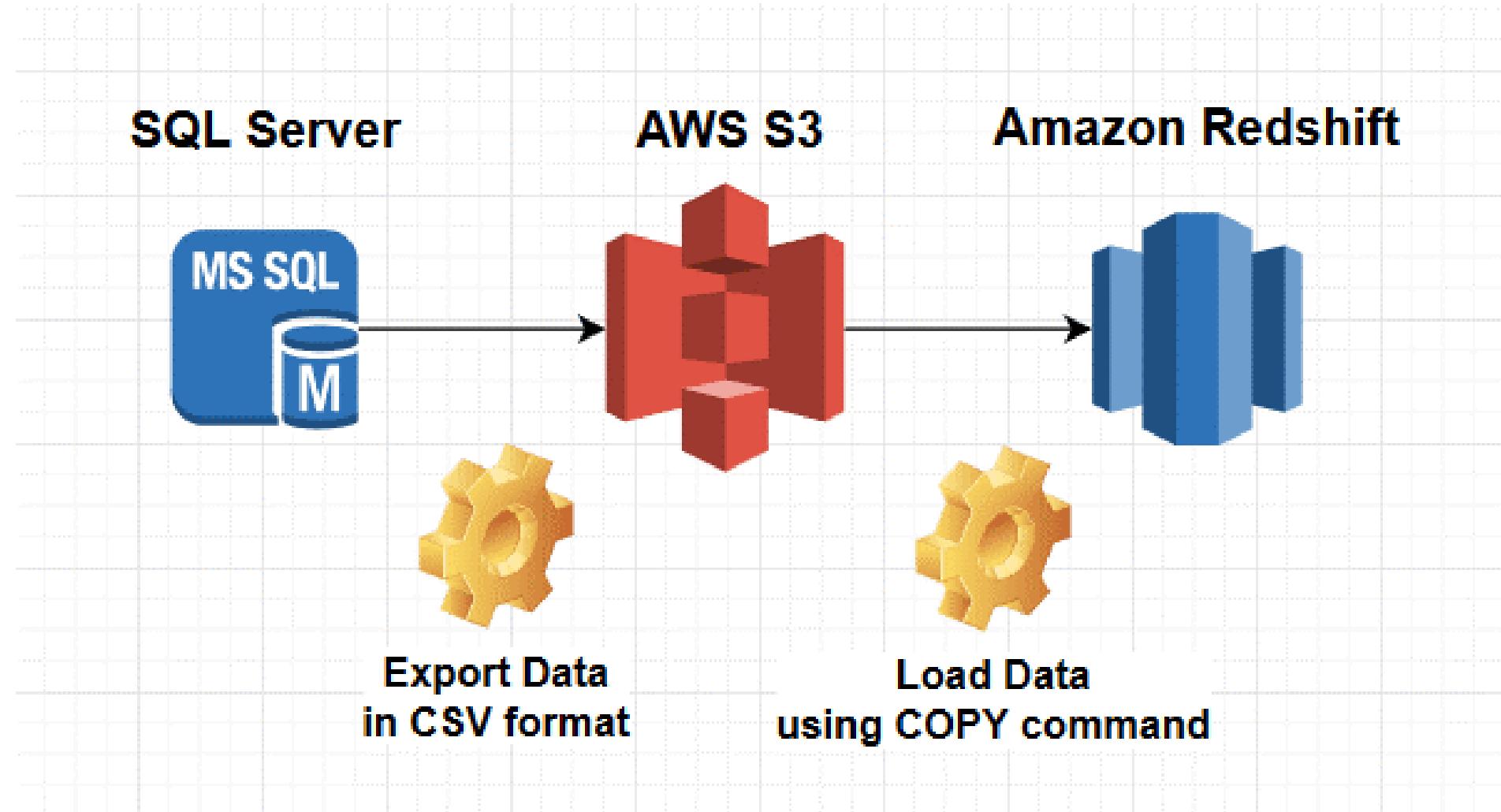
# Configuring cluster node types

Node size	vCPU	RAM (GiB)	Managed storage quota per node
ra3.xlplus	4	32	32 TB
ra3.4xlarge	12	96	128 TB
ra3.16xlarge	48	384	128 TB

# Configuring cluster Security



# Loading data in Redshift using COPY commands



The **Redshift COPY Command** is a very powerful and flexible interface to load data to Redshift from other sources. That said, it does have its share of limitations, specifically when it comes to enforcing data types and handling duplicate rows.

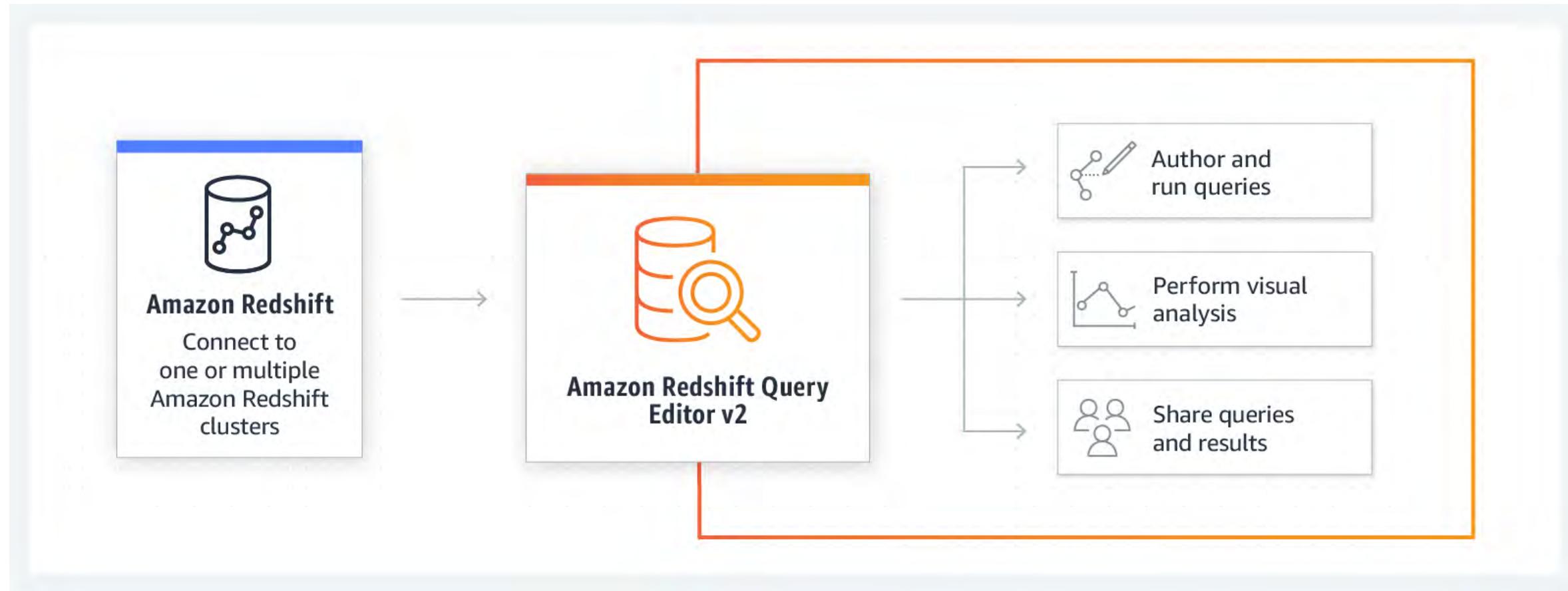
# Loading and managing data in Redshift using UNLOAD commands



## Methods to Unload Data from Amazon Redshift to S3

- Method 1: Unload Data from Amazon Redshift to S3 using the UNLOAD command
- Method 2: Unload Data from Amazon Redshift to S3 in Amazon Parquet Format

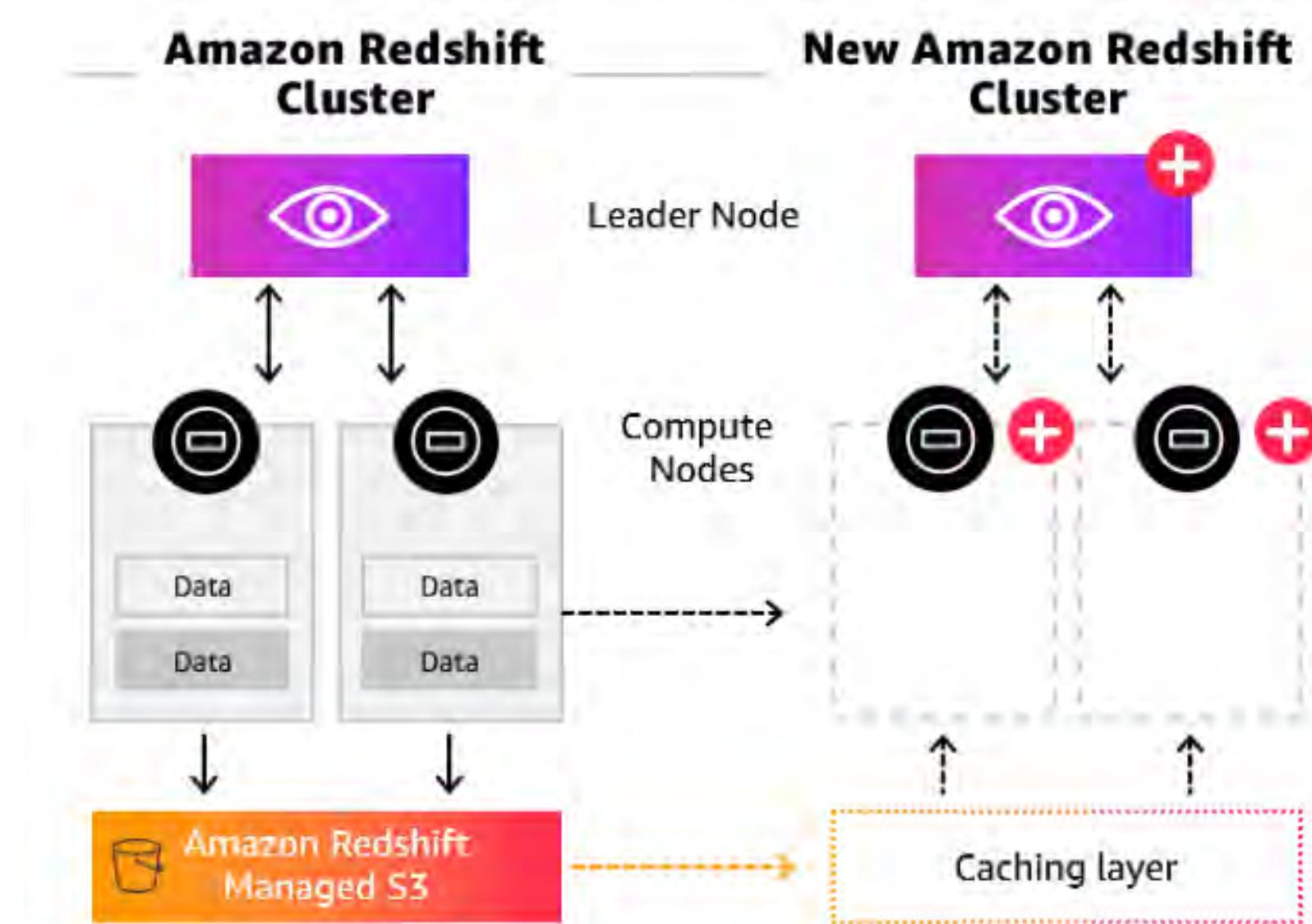
# Querying in Amazon Redshift



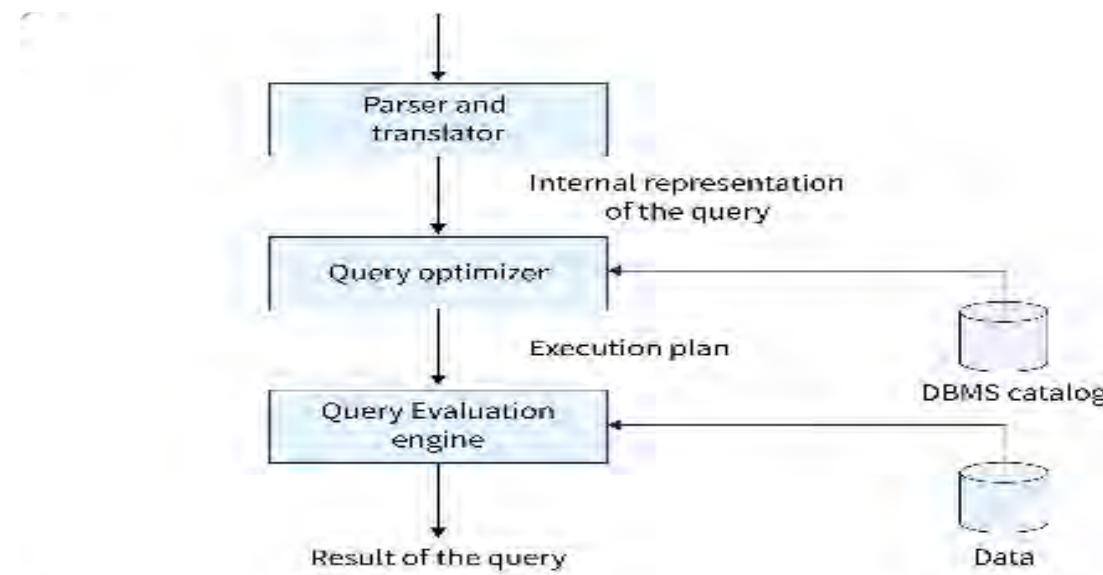
Amazon Redshift Query Editor v2.0 is a web-based analyst workbench for you to explore, share, and collaborate on data with your teams in SQL through a common interface.

# Optimizing Performance in Amazon Redshift

The following diagram is an architectural illustration of how Automatic table optimization works:

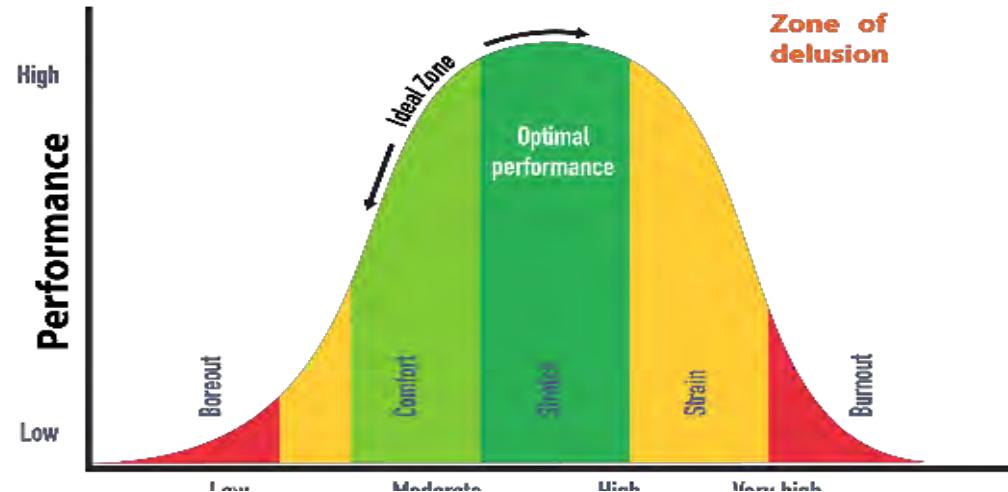


# Query optimizer

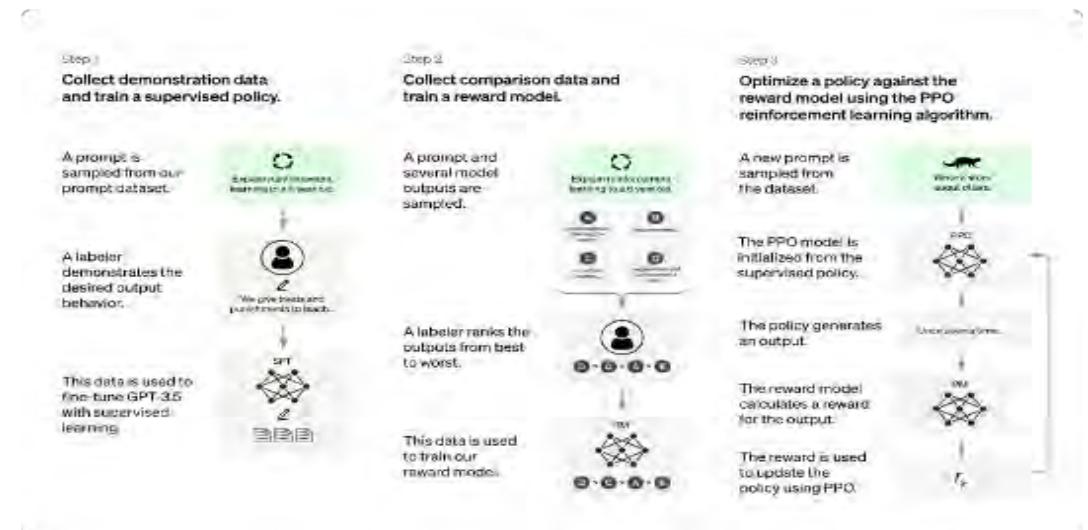


- The Amazon Redshift query run engine incorporates a query optimizer that is MPP-aware and also takes advantage of the columnar-oriented data storage.
- The Amazon Redshift query optimizer implements significant enhancements and extensions for processing complex analytic queries that often include multi-table joins, subqueries, and aggregation.

# Utilizing Redshift query tuning performance



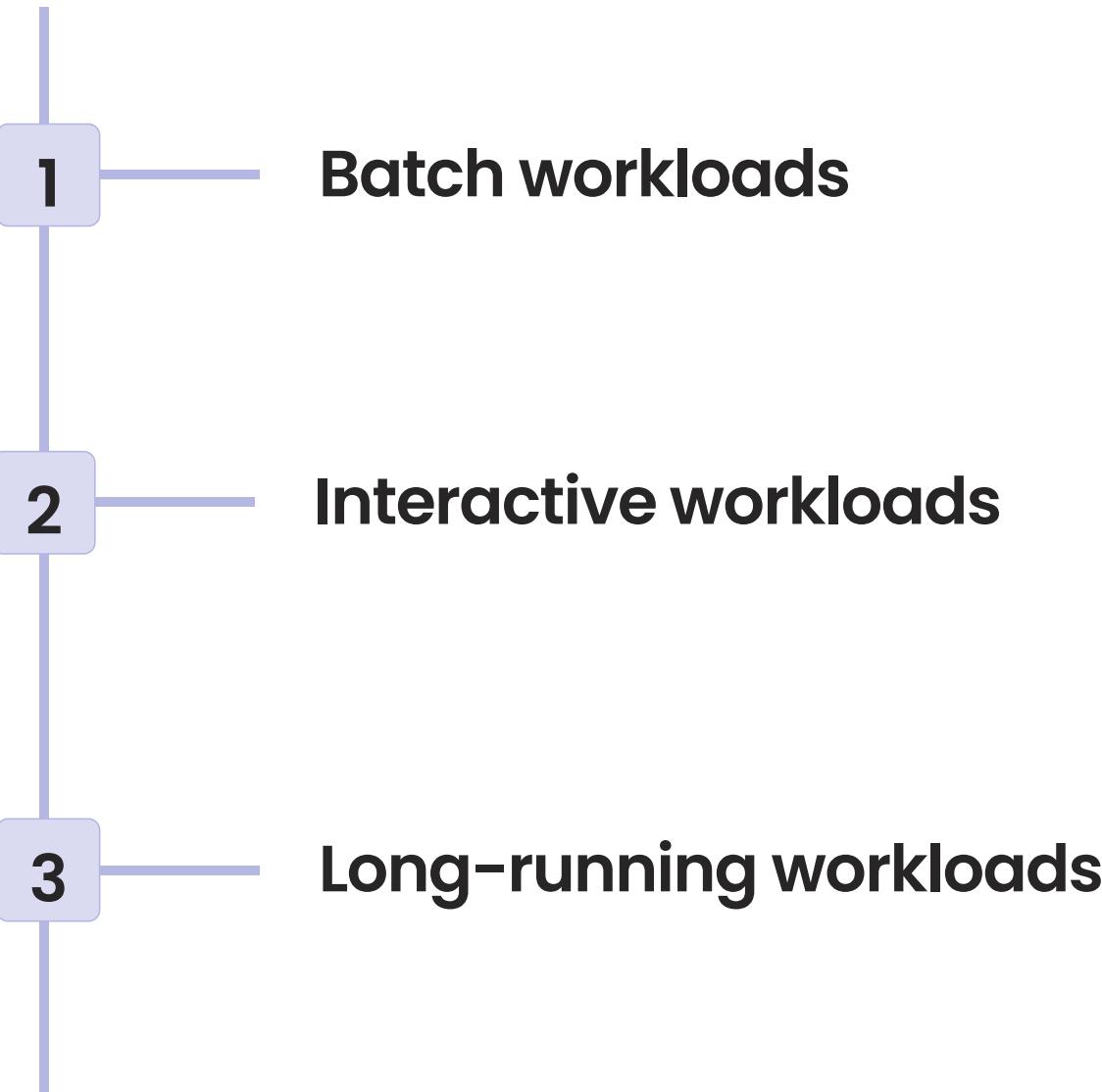
- Use Redshift performance tuning features to optimize performance, such as WLM, concurrency scaling, and workload management.



- Learn how to use advanced techniques for fine-tuning Redshift performance, such as workload management, parallel query execution, and optimizing disk utilization.

# Types of workloads in Redshift

---



# Workload management when dealing with large datasets

---

1

Granularity of queries

2

Query queuing

3

Workload management via WLM

# Scaling concurrency in Redshift

---

Maximize cluster size

Use appropriate distribution keys

Be mindful of query complexity

Avoid hotspots in your data

# Benefits of effective workload management and concurrency scaling in Redshift

---

- Improved query performance
- Reduced costs
- Improved data quality

iamneo



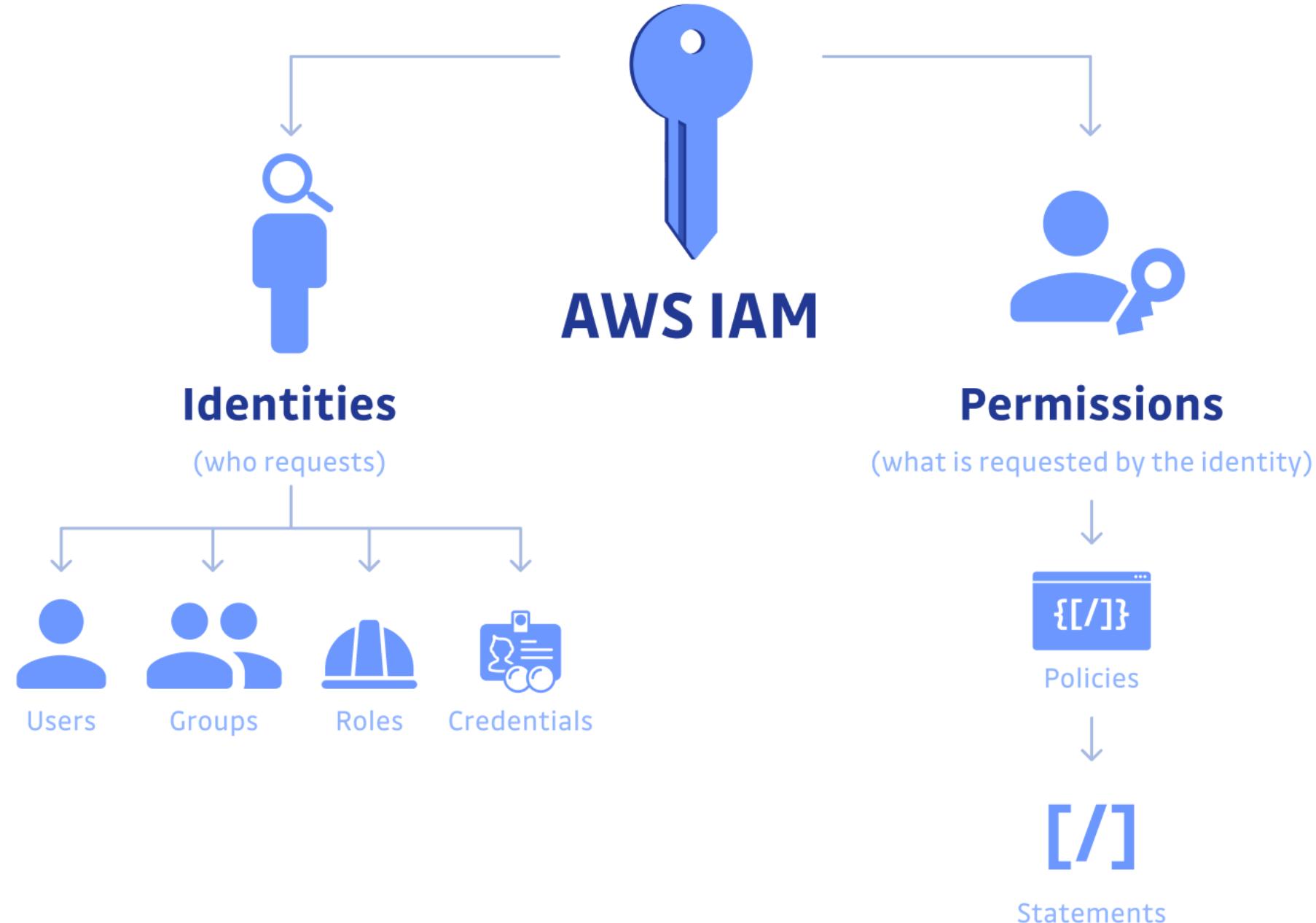
# IAM and MFA

---

# Agenda

- IAM Overview
- Roles of IAM in AWS Security
- Simple and Secure
- Control Access
- Centralize Management

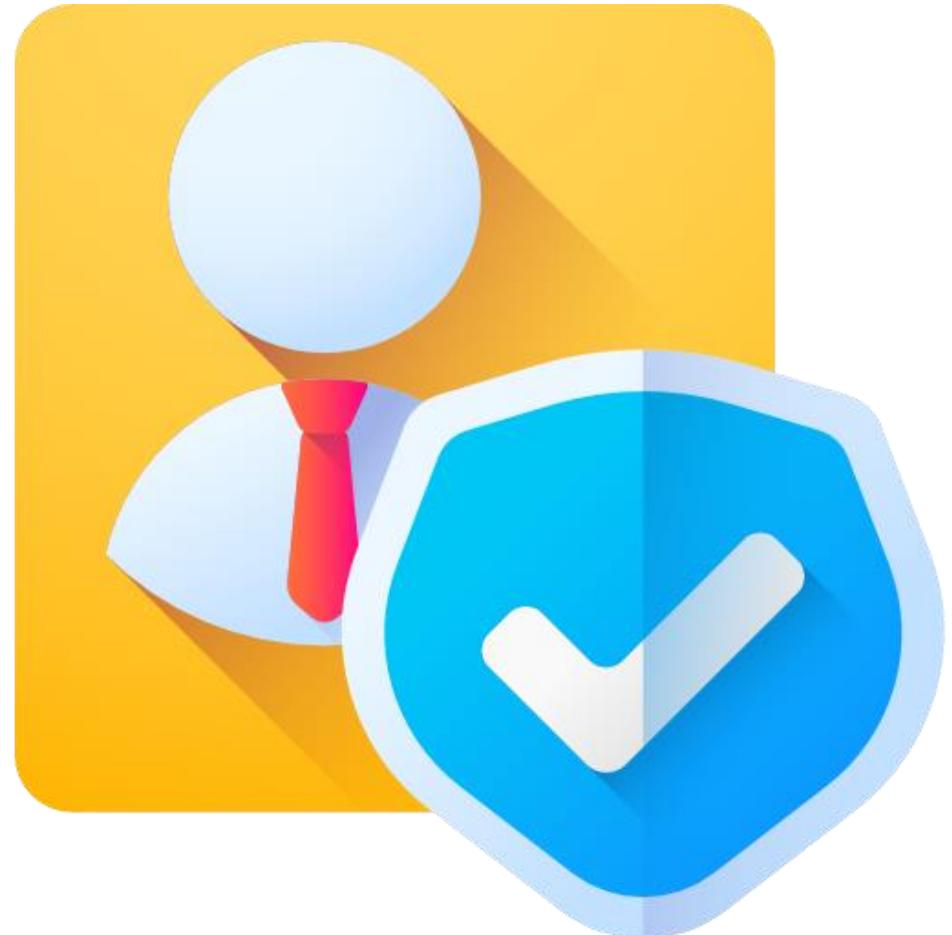
# Users, Groups, Roles, and Policies in IAM



# IAM Best Practices

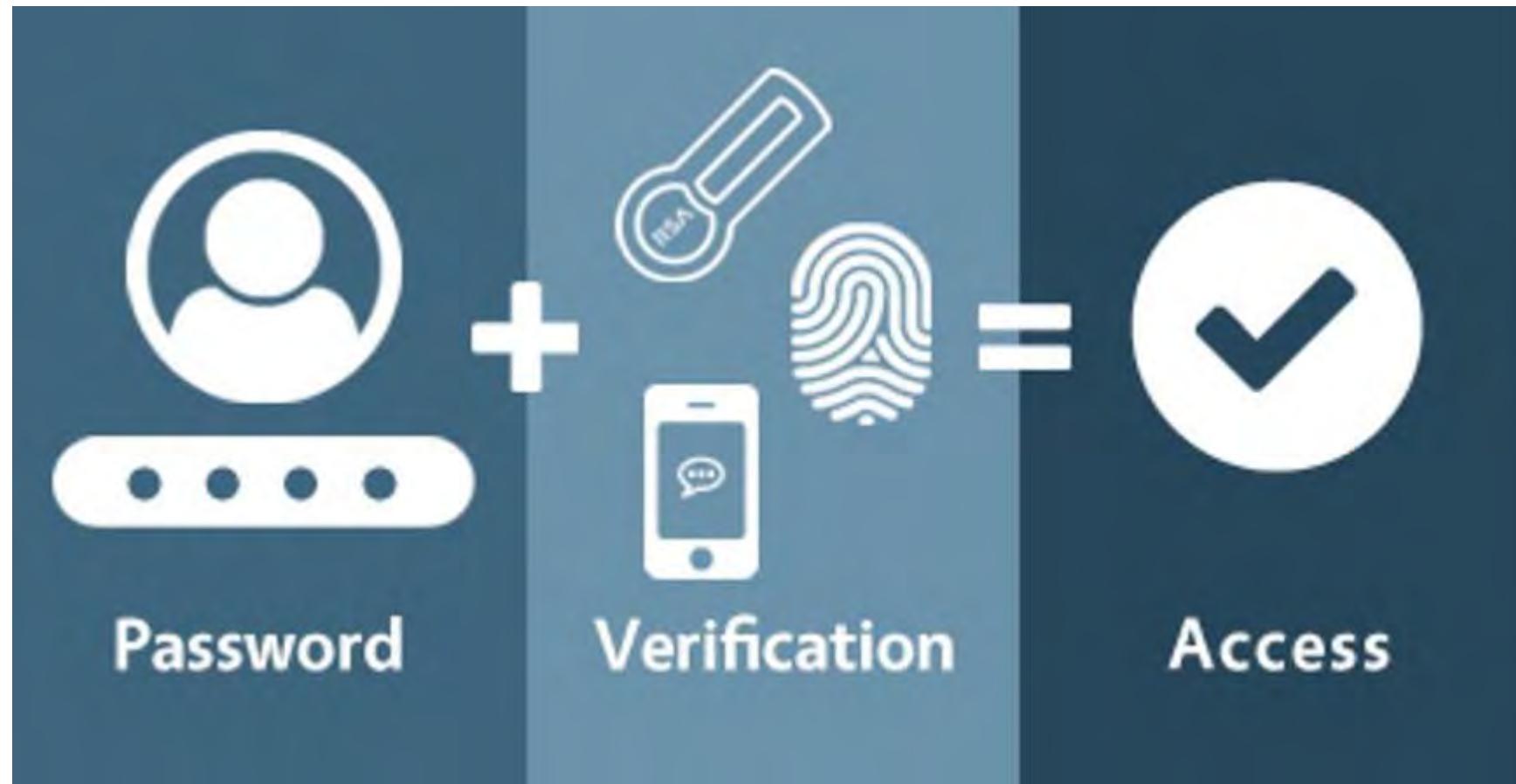
---

- Use temporary credentials
- Use IAM Access Analyzer
- Require multi-factor authentication (MFA)
- Set permissions guardrails across multiple accounts
- Rotate access keys regularly for use cases that require long-term credentials
- Safeguard your root user credentials and don't use them for everyday tasks
- Use permissions boundaries to delegate permissions management within an account.
- Use strong and unique passwords for IAM users and root accounts, and avoid sharing or reusing passwords.



# Multi-Factor Authentication (MFA)

MFA is to enhance account security for IAM users and root accounts. MFA adds an extra layer of security to protect user accounts from unauthorized access and data breaches, which can lead to financial losses and reputation damage. Implementation of MFA has become increasingly important in today's digital age.

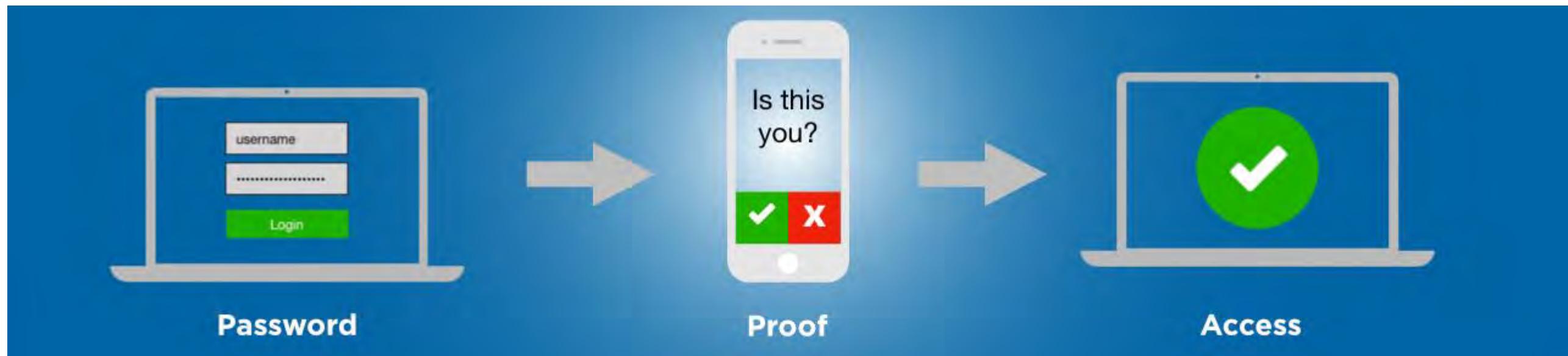


# The Importance of MFA

**Phishing attacks:** MFA helps to prevent unauthorized access to user accounts, even if the user's password has been compromised in a phishing attack.

**Stolen credentials:** Even if a hacker manages to steal a user's credentials through malware or other means, MFA can help to prevent access to user accounts.

**Ransomware:** Even if a hacker manages to steal a user's credentials through malware or other means, MFA can help to prevent access to user accounts.



# Types of MFA in AWS IAM

Select an MFA device to use, in addition to your username and password, whenever you need to authenticate.



## Authenticator app

Authenticate using a code generated by an app installed on your mobile device or computer.



## Security Key

Authenticate using a code generated by touching a YubiKey or other supported FIDO security key.



## Hardware TOTP token

Authenticate using a code displayed on a hardware Time-based one-time password (TOTP) token.

# MFA Authenticator app

## Set up your authenticator app

A virtual MFA device is an application running on your device that you can configure by scanning a QR code.

1

Install a compatible application such as Google Authenticator, Duo Mobile, or Authy app on your mobile device or computer.

[See a list of compatible applications](#)

2

Show QR code

Open your authenticator app, choose **Show QR code** on this page, then use the app to scan the code. Alternatively, you can type a secret key. [Show secret key](#)

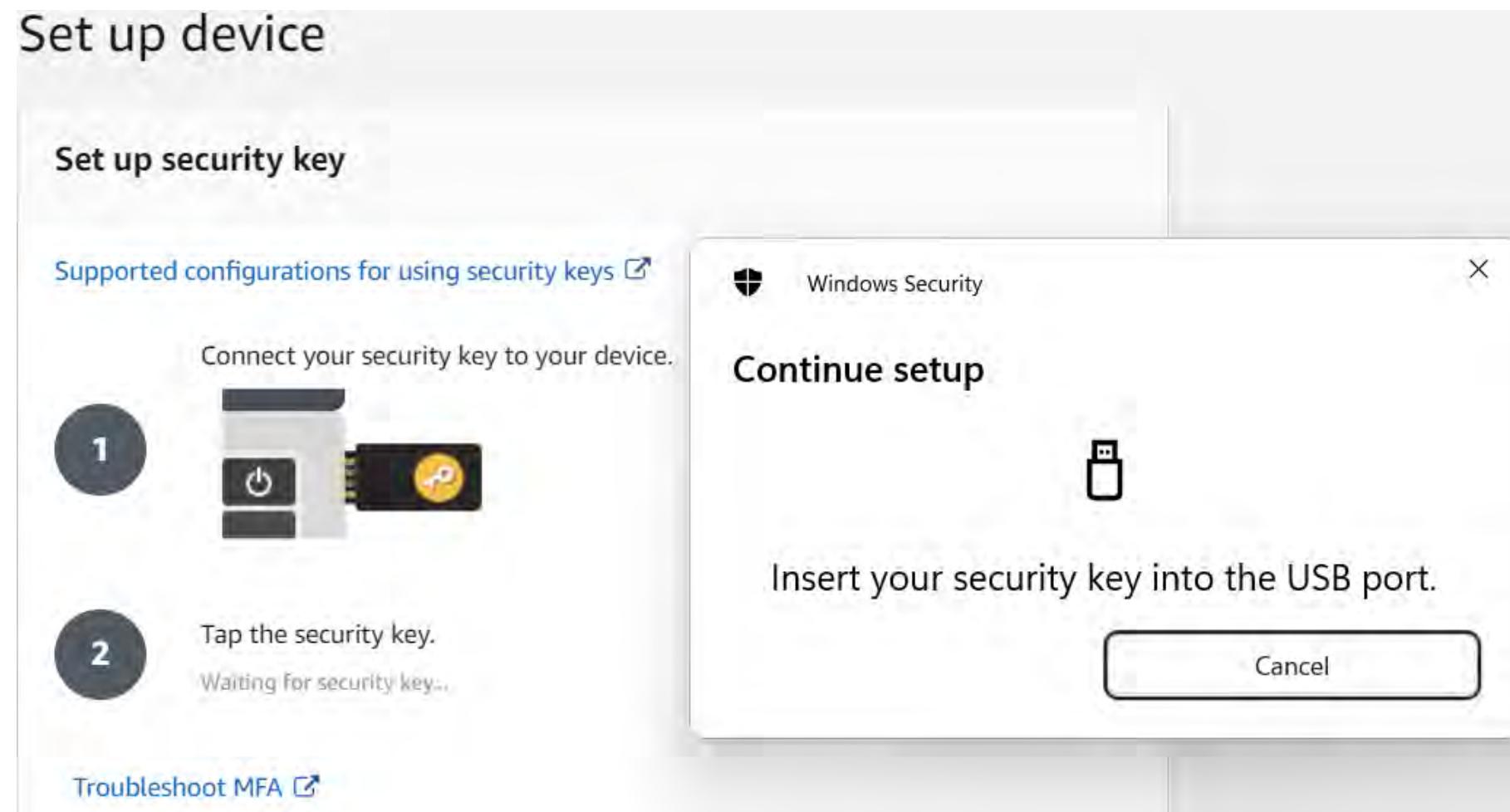
3

MFA code 1

MFA code 2

# Security Keys

Security keys are USB devices that provide an extra layer of security by requiring physical access to a device to log in.



# Hardware MFA Devices

A hardware key fob token is a small, physical device that generates Time-based one-time password (TOTP) token that is required for login.

Set up device

**Set up hardware MFA device**

For more information about using a hardware MFA device, see the [IAM User Guide](#)

- 1 Enter the device serial number located on the back of the device.
- 2 Press the button on the front of the device and enter the 6-digit number that appears.
- 3 Wait 30 seconds and then press the button again. Enter the second number.

[Cancel](#) [Previous](#) [Add MFA](#)

# Configuring MFA for IAM Users

Implementing MFA for IAM users can provide an additional layer of security for their AWS account access.

Step	Description
1	Login to AWS Console
2	Select IAM from the list of services
3	Click on "Users" from the left navigation menu
4	Select the user to whom MFA has to be added
5	Click on the "Security Credentials" tab
6	Click "Activate MFA" and follow the prompts to configure and set up MFA

# Configuring MFA for Root Accounts

MFA for the AWS root account provides an additional level of security and helps to prevent unauthorized access to the account.

- 1** **Steps to Configure MFA for Root Account:**  
Login to the AWS Management Console, go to the Account Settings page, click on "Security Credentials", select "Activate MFA", and follow the prompts to set up MFA.
- 2** **Additional Security Benefits**  
MFA for the AWS root account can help to prevent accidental or deliberate changes to critical account settings, which could result in detrimental business impact.
- 3** **Importance of protecting root account**  
Root account access should be limited to only a few trusted individuals to reduce the risk of a security breach.

iamneo



Thankyou

---



# AWS Monitoring and logging

---

# Monitoring with AWS CloudWatch

---



Amazon CloudWatch

Welcome to an overview of AWS CloudWatch.  
Learn how it can provide you a unified view of  
your AWS resources and applications.

# Features and Uses of CloudWatch

---

CloudWatch is a monitoring service for AWS resources and the applications you run on them.

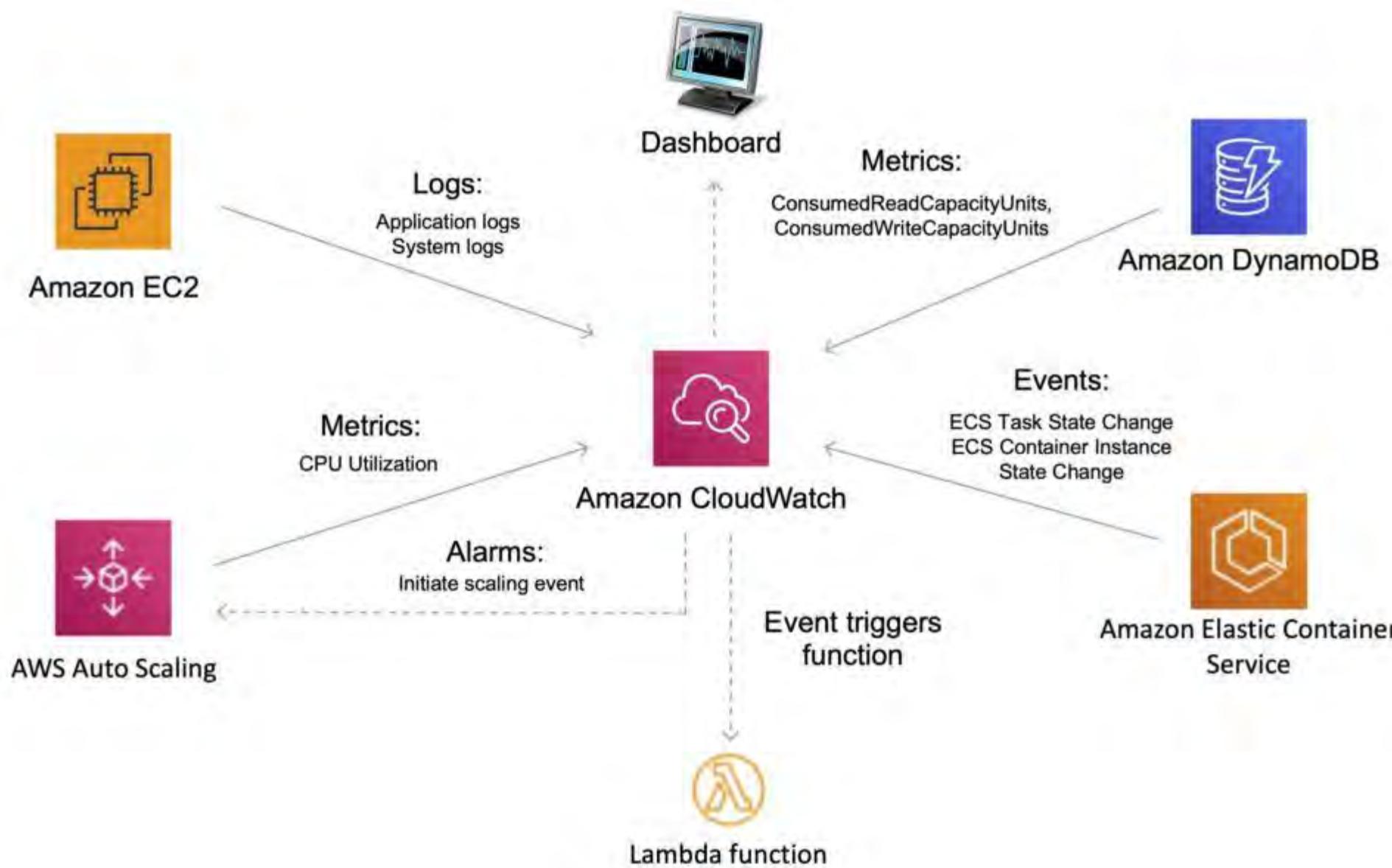
**Unified Dashboard**

**Monitor Serverless Applications**

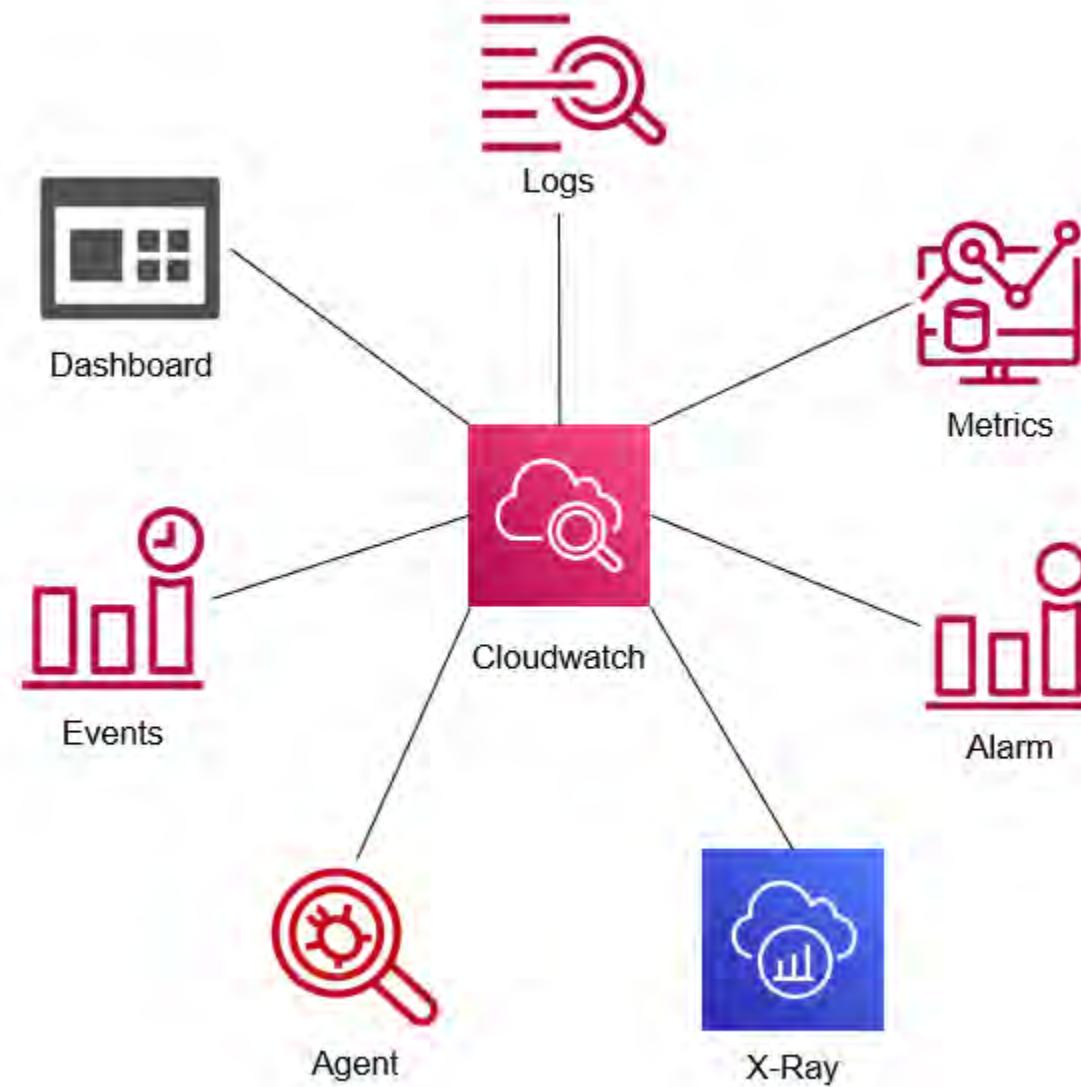
**Performance Monitoring**

**Logs and Analysis**

# Features and Uses of CloudWatch

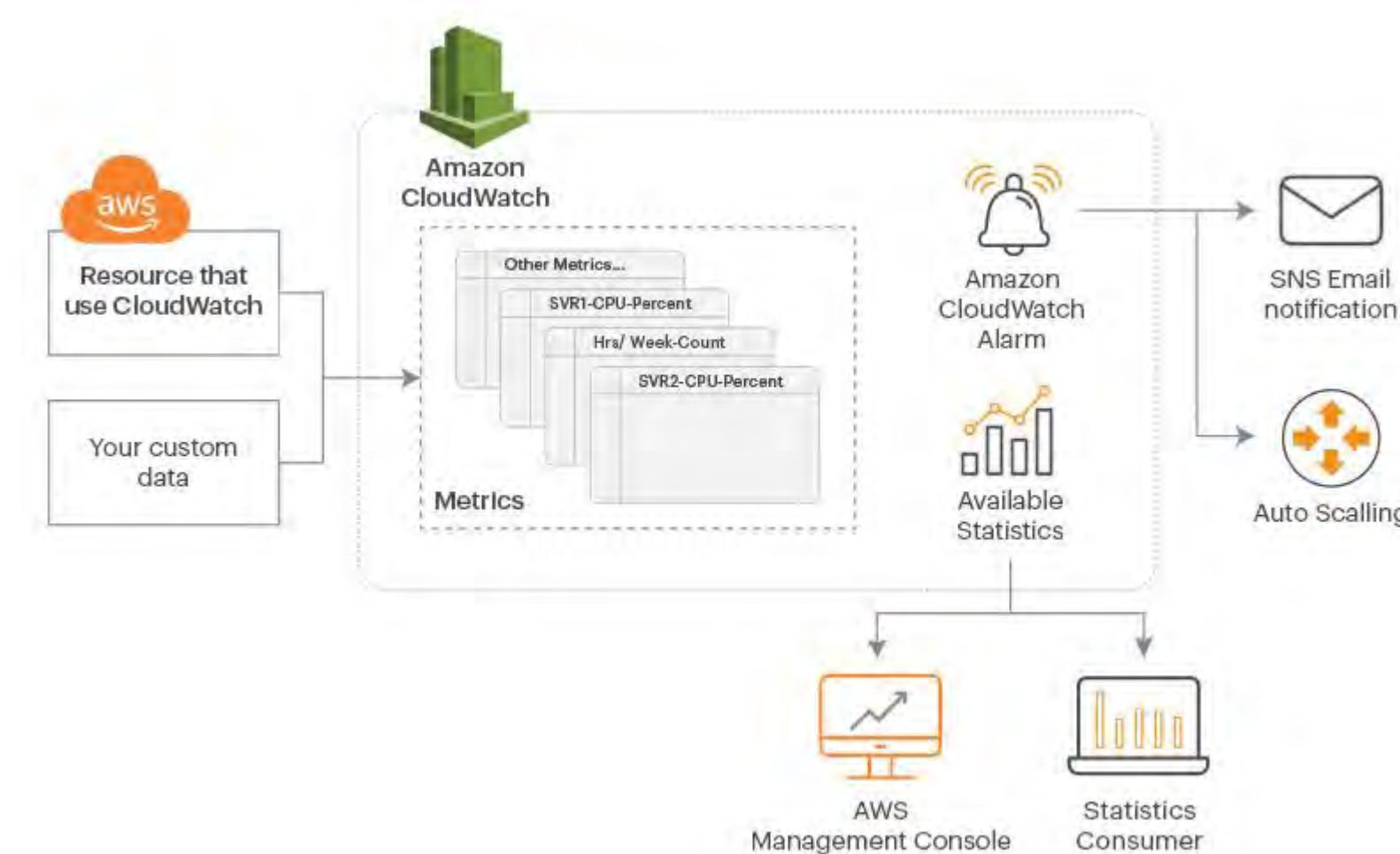


# Components of Cloudwatch



# Metrics in CloudWatch

CloudWatch provides a variety of metrics that you can use to monitor your AWS resources and applications.



# Metrics in CloudWatch

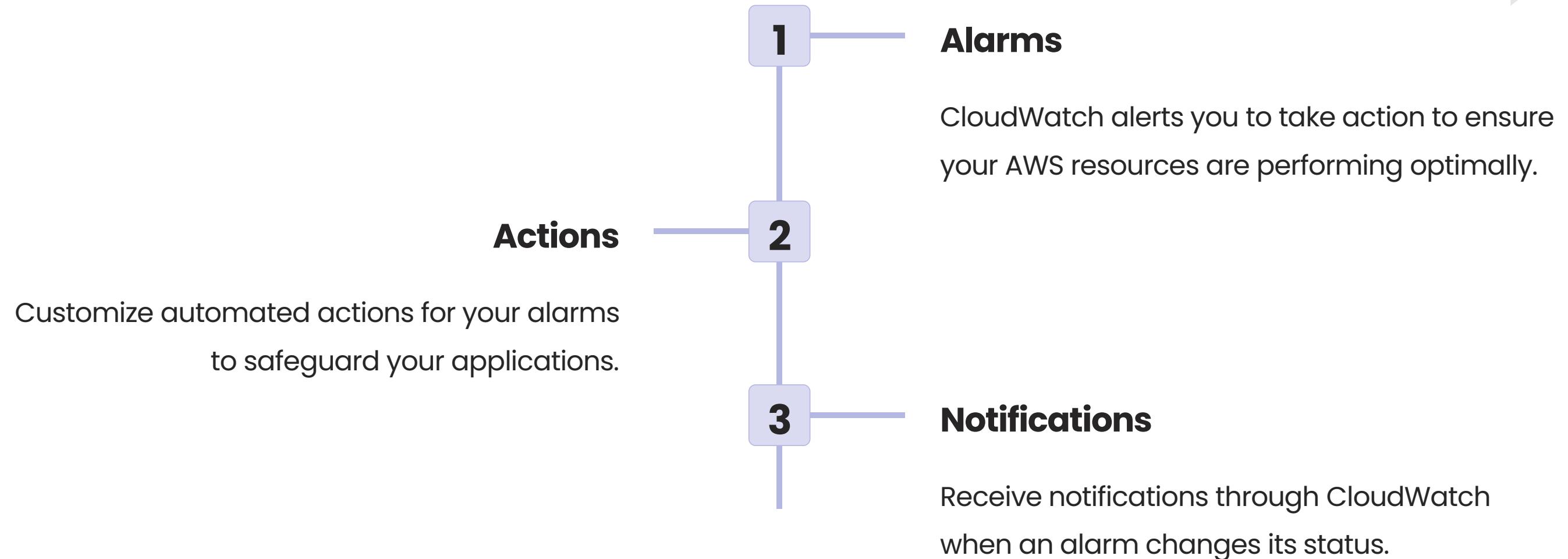
---

**Custom Metrics**

**Predefined Metrics**

**High-Resolution Metrics**

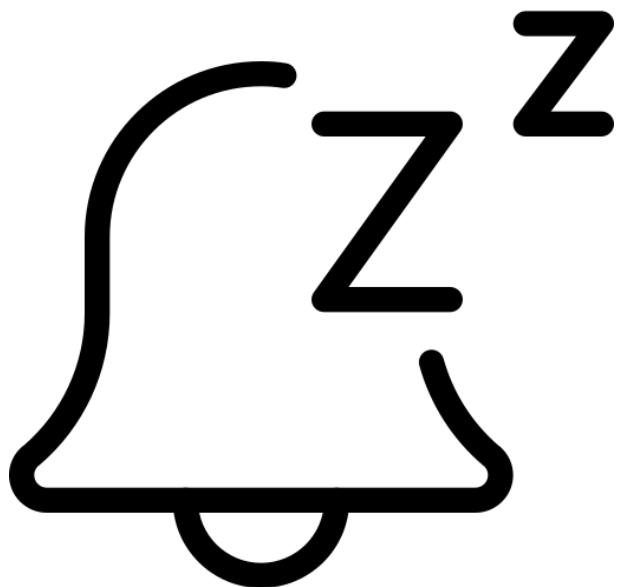
# Alarms and Notifications



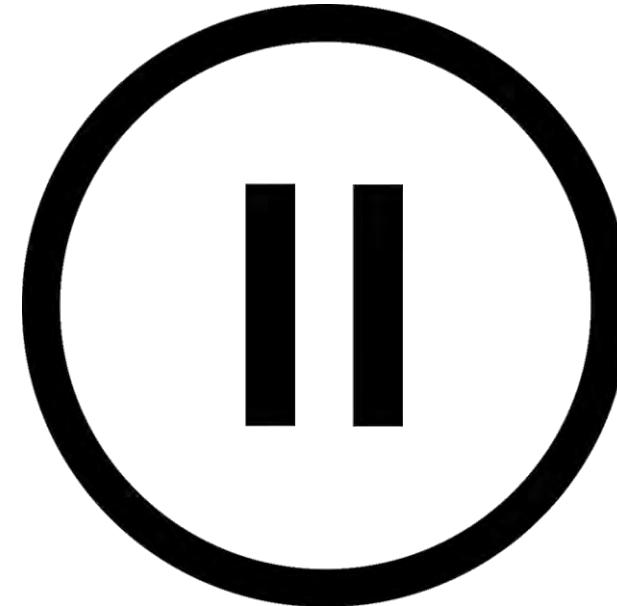
# Creating and Managing CloudWatch Alarms



**Create an Alarm**



**Snooze an Alarm**



**Suspend an Alarm**

# Logs and Log Groups in CloudWatch

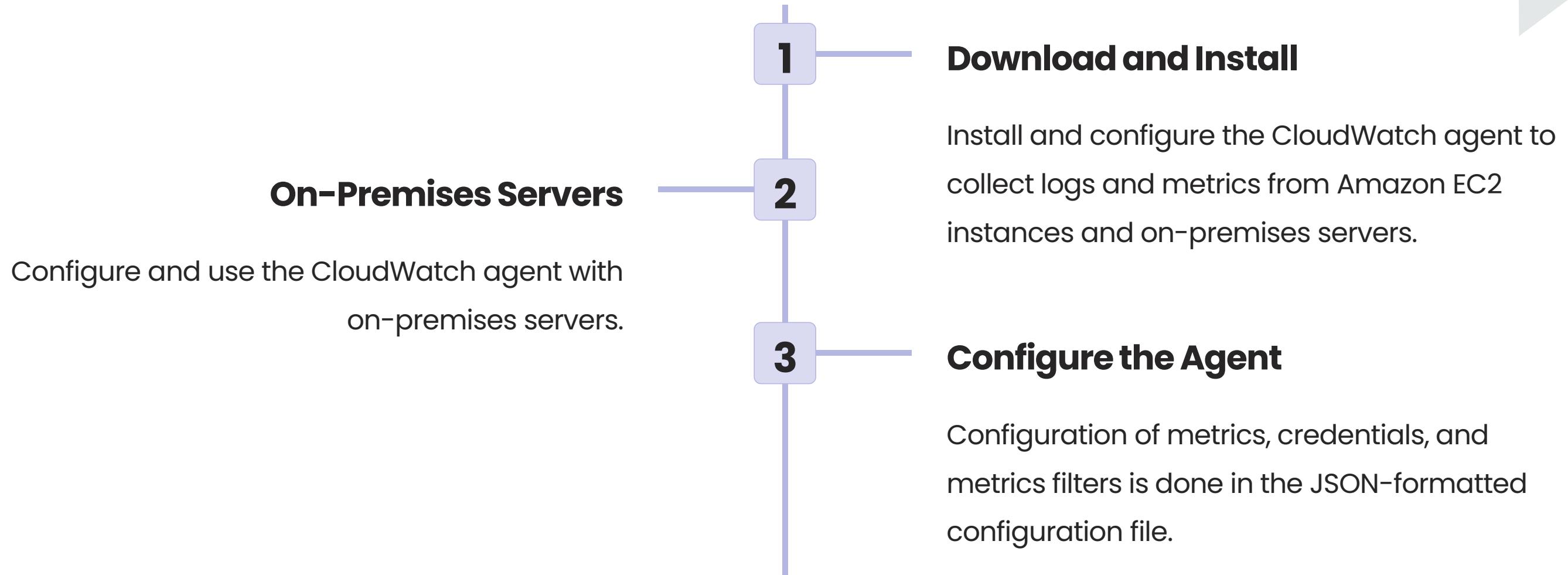
---

**Create Log Groups**

**Log Stream**

**Centralized Log Management**

# CloudWatch Agent: Installation and Configuration



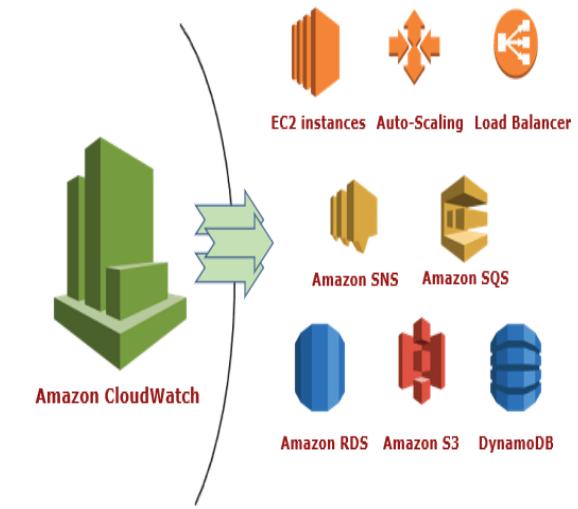
# Best practices for using CloudWatch



**Data Collection**



**Alerting**



**Architecture**

iamneo



# AWS CloudTrail

---

# Overview of AWS CloudTrail

## What is AWS CloudTrail?

AWS CloudTrail is a service that records user activity and API events in your AWS account.

## Why AWS CloudTrail?

CloudTrail provides a history of AWS API calls for your account, including calls made via the AWS Management Console, SDKs, command line tools, and other AWS services.

## How does AWS CloudTrail work?

CloudTrail captures API calls and delivers the resulting log files to an Amazon S3 bucket that you specify.

# Benefits of CloudTrail for AWS customers



**Improved Security**



**Cost Management**

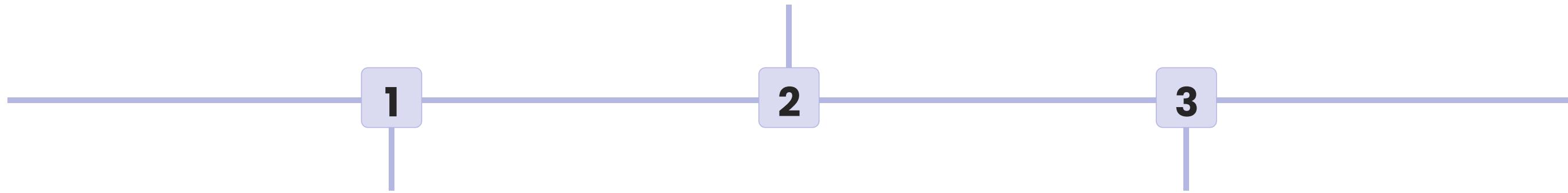


**Operational Insights**

# How CloudTrail works

## CloudWatch Integration

You can configure CloudTrail to create CloudWatch Metrics to help identify trends and get insights into your AWS environment.



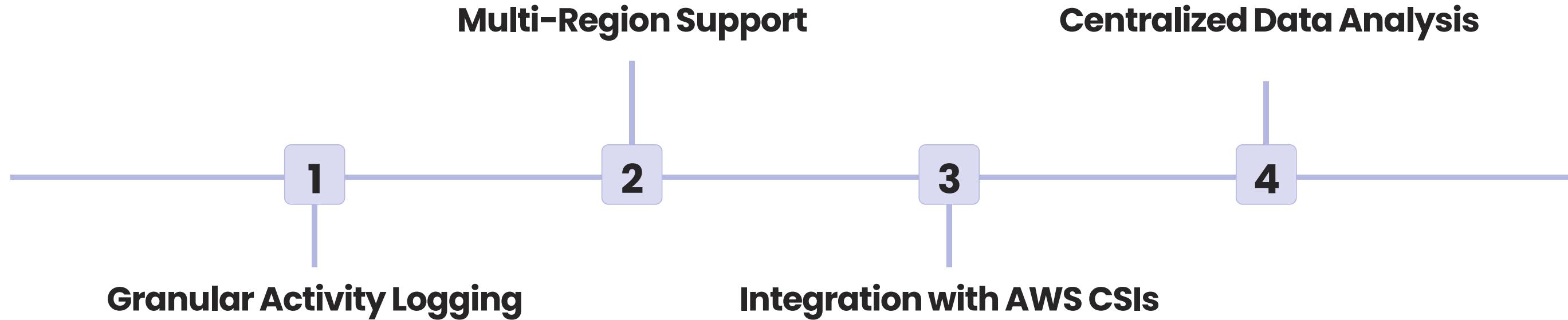
### Log File Delivery

CloudTrail delivers log files to an Amazon S3 bucket specified by you.

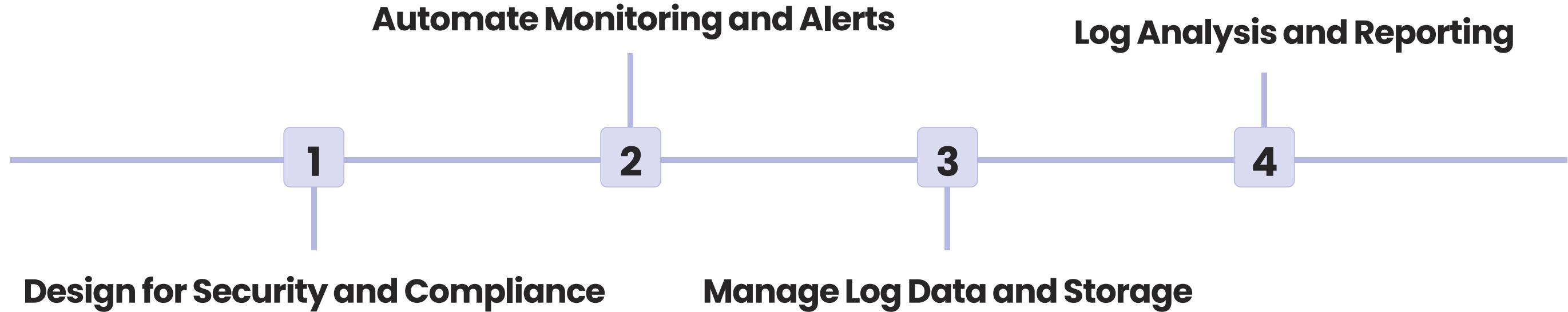
### Event Notifications

You can set up Amazon SNS notifications based on certain criteria within CloudTrail. These notifications can be received via email, SMS, or other supported transports.

# Key Features of AWS CloudTrail



# Usage of implementing CloudTrail



# Retention and encryption of CloudTrail logs

## Log File Retention

You can specify how long CloudTrail should keep your logs, from 1 to 7 years. This retention period can help you meet your regulatory and compliance requirements.

## Encryption Options

You can encrypt your log files using AWS Key Management Service (KMS) or server-side encryption with Amazon S3-managed encryption keys (SSE-S3).

# Real-world use cases of CloudTrail

---

- 1 **Investigating Unauthorized Access**
- 2 **Enabling Compliance Audits**
- 3 **Tracking Resource Changes**
- 4 **Identifying Root Cause of Issues**

iamneo



# AWS Introduction - Session 3

---

# Agenda

---

- AWS Economics
- Design Principles
- AWS Account Creation



## AWS Economics: Lowering Your Costs in the Cloud

Learn how to leverage AWS pricing models and the Free Tier to save money. We'll cover On-demand, Reserved, and Spot instances, as well as strategies for cost optimization and TCO analysis.



# AWS Economics

---

**Different Pricing Models**

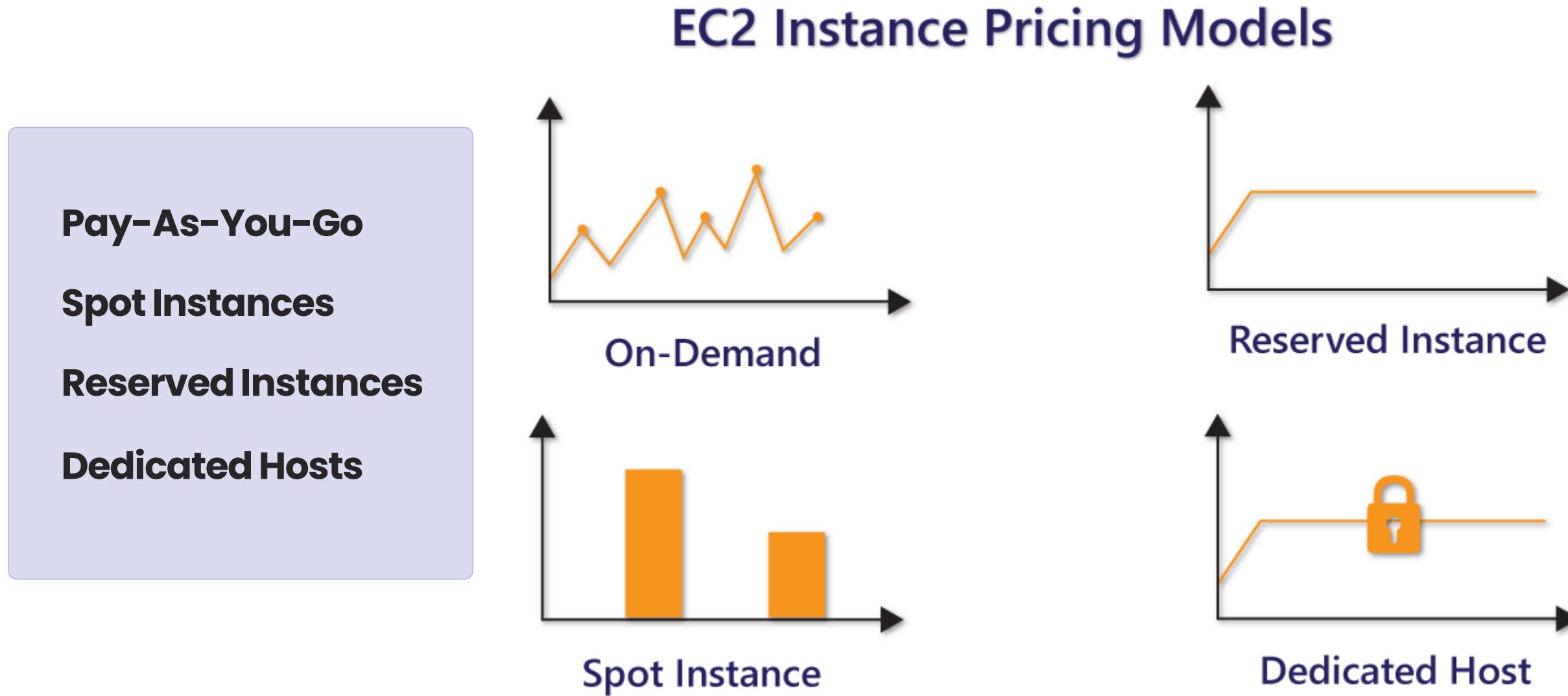
**Cost Optimization Techniques**

**Total Cost of Ownership (TCO) Analysis**

**Understanding Billing and Pricing**



# AWS Pricing Model



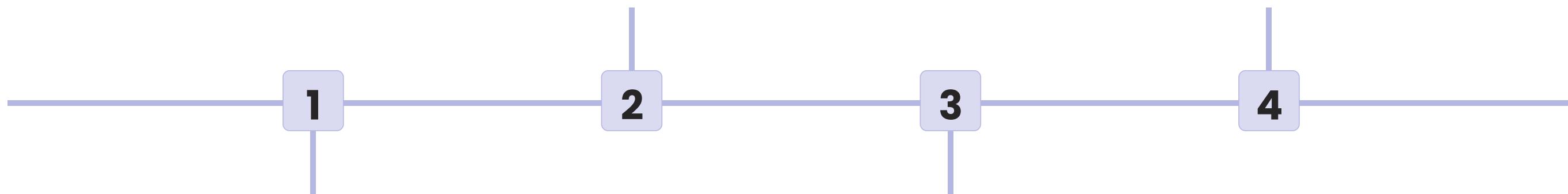
# Factors Affecting AWS Costs

## Storage

The size and type of storage you use impacts your costs, as well as how frequently you access or move data.

## Network Bandwidth

Your network bandwidth usage can impact your costs, depending on how much traffic you generate.



## Instance Type

The cost of your infrastructure depends on the type of instance you choose and the demand in the market.

## Data Transfer

The amount of data transferred between regions and availability zones can significantly impact your bills.

# Optimizing Infrastructure for Cost Savings



*Optimization of Cloud Consumption*

Right sizing



## Right-Sizing Instances

Selecting the appropriate instance type based on your needs can save you up to 50% on costs.

## Monitoring and Measuring Resource Utilization

Identifying underutilized resources can reduce your costs by up to 40%.



## Using Cost Optimization Tools

Tools like AWS Cost Explorer and AWS Budgets can help you analyze and control your costs.

# Cost Management Tools in AWS



## AWS Cost Explorer

Analyze your costs, usage, and trends to identify areas for optimization.



## AWS Budgets

Set custom cost and usage budgets to monitor and optimize your costs in real-time.



## AWS Trusted Advisor

Get tailored cost optimization recommendations based on your usage patterns and requirements.

# Total Cost of Ownership (TCO) Analysis

## Direct Costs

Hardware, software, and personnel costs.

## Indirect Costs

Environmental costs, providers, and governance.

## Opportunity Costs

What you miss by not going cloud, like innovation and agility.

## Risk Costs

Lack of agility, security, and compliance risks.

# AWS Design Principles

Learn about the principles that underpin a successful AWS cloud architecture, from scalability to automation.



# Scalability



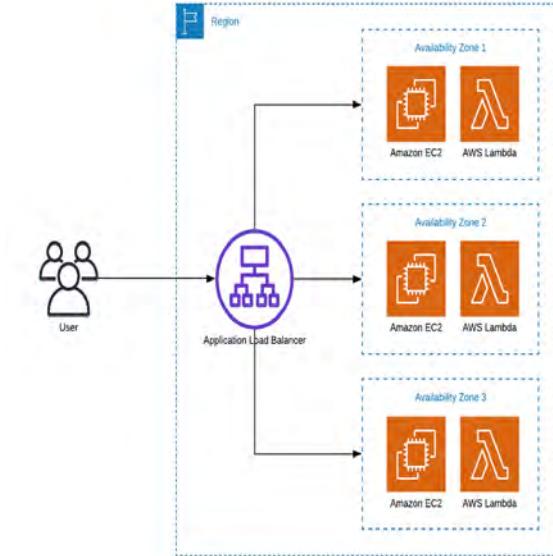
## Start Small 🏠

Design for scalability by starting small and gradually increasing your capacity as your workload grows.



## Cloud Resources 🌐

Take advantage of AWS's elastic resources to scale up and down automatically, reducing infrastructure costs.



## Load Balancing 🚒

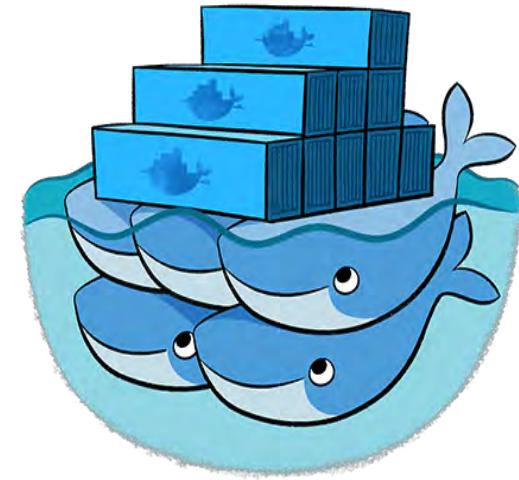
Route incoming traffic to multiple targets to improve your application's availability, scalability and redundancy.

# Services, not Servers



## Maintenance Challenges

Using servers might make it difficult to stay up-to-date with security patches, low-level system troubles and unpredictable growth.



## Containerization

Containers are an excellent abstraction for packaging, scaling, and distributing applications, and to reduce infrastructure diversity.



## Microservices

Decompose applications into microservices that you can develop, deploy, and scale independently, thus targeting different teams.

# Security

---

## Secure Access

Granting the minimum necessary privileges to your staff will reduce the risk of sensitive data exposure.

## Regular Audits

Conduct regular security audits to verify that your infrastructure is up-to-date with the latest security patches.

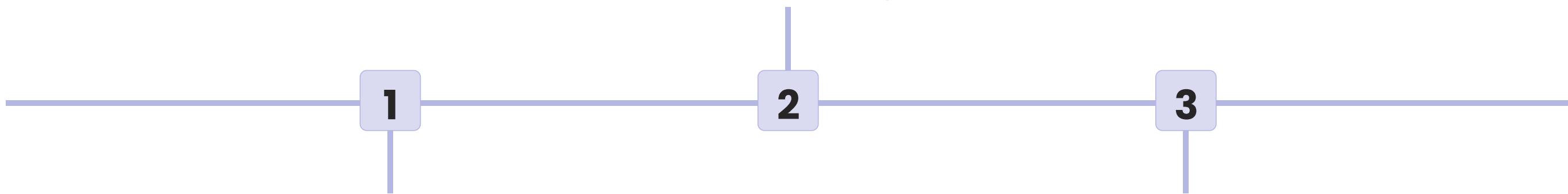
## Encryption

Using regularly-rotated encryption keys will protect sensitive data from security breaches in transit and at rest.

# Automation

## DevSecOps

By shifting security left in the development process through infrastructure-as-code, and auditing of cloud resources will increase automation and enhance security.



## Continuous Integration & Deployment

CI/CD operations that are automated continuously will reduce the probability of deployment errors, improve consistency throughout teams and providing quicker release cycles.

## Orchestration

Build automated workflows that handle orchestration of cloud resource management tasks reducing downtime, scaling and auto-healing capabilities.



## Creating Your AWS Account

Discover the benefits of creating an AWS account and how to navigate the process, as well as how to set up billing and support resources available to you.



# Getting Started with AWS: Creating Your Account

## Types of Accounts

Choose between Basic, Developer, and Enterprise accounts, each with different levels of support and access to features.

## Step-by-Step Guide

We'll walk through the account creation process, from selecting your plan to configuring settings.

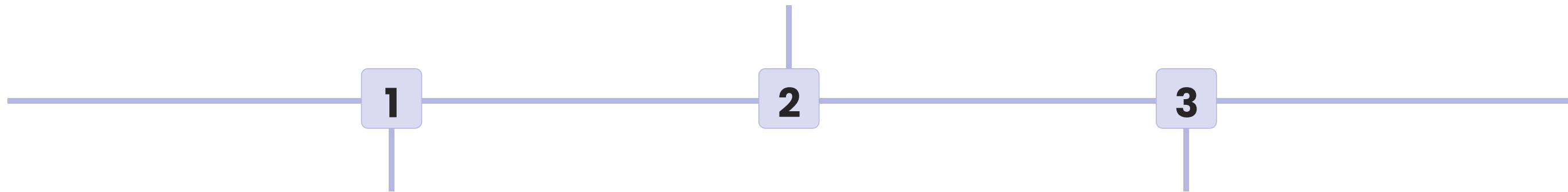
## Settings and Configurations

Create policies, configure user access, and set up billing alerts and notifications to ensure a smooth experience.

# Step-by-Step Account Creation Process

## Adding IAM Users

Learn how to add IAM users to your AWS account, including how to create custom user groups and set user policies based on their roles.



### Creating an Account

Discover the step-by-step process to create an AWS account, including how to choose your account type, provide billing information, and more.

### Setting Up Billing and Payments

Discover how to set up billing and payments, as well as how to monitor your spending and estimate future costs.



# Thank You

iamneo



# Amazon Billing

---

# Introduction to AWS Billing

- The AWS Billing console allows you to easily understand your AWS spending, view and pay invoices, manage billing preferences and tax settings, and access additional Cloud Financial Management services.
- Quickly evaluate whether your monthly spend is in line with prior periods, forecast, or budget, and investigate and take corrective actions in a timely manner.
- Managing your AWS costs and usage doesn't have to be difficult.
- The AWS Billing Console is your one-stop-shop for accessing your billing, payments, and cost management information and capabilities.

# Cost Management console

The screenshot shows the AWS Management Console homepage. At the top, there is a dark navigation bar with the AWS logo, a "Services" dropdown, "Resource Groups" dropdown, "AWS Cost Explorer" icon, "AWS Budgets" icon, a notification bell icon, and user information ("Admin/erincarl-Isengard @ kulj... Oregon Support"). Below the navigation bar, the title "AWS Management Console" is displayed. On the left, there is a sidebar titled "AWS services" with a "Find Services" search bar containing the text "Billing". Below the search bar, under "Recently visited services", there are links for "Billing" (which is highlighted with a green border), "AWS Cost Explorer", "Trusted Advisor", and "EC2". To the right of the sidebar, there are two main sections: "Access resources on the go" (with a mobile phone icon and text about the AWS Console Mobile App) and "Explore AWS" (with a link to "EC2 Spot Instances" and its description). The background of the main content area has a faint watermark of the AWS logo.

# The Importance of Monitoring Costs

---

1

## **Costs can add up quickly**

Without proper monitoring, the costs of AWS services can quickly add up, causing unexpected charges and putting a strain on your budget.

2

## **Resource allocation**

Monitoring costs can also help you allocate resources more effectively, ensuring you're using your budget wisely.

3

## **Optimizing services**

By identifying areas of high cost, you can make adjustments to optimize your AWS services and reduce waste.

# AWS Account Setup and Billing Configuration

---

- **Creating an AWS account**

We'll walk through the process of creating a new AWS account and setting up billing preferences.

- **Configuring billing alerts and notifications**

Setting up billing alerts and notifications can help you stay on top of your spending and avoid unexpected charges.

- **Introduction to billing console**

Learn how to navigate the AWS billing console and view detailed billing reports to help manage costs.

- **Understanding the AWS Free Tier and its limitations**

We'll provide an overview of the AWS Free Tier, including its limitations and how to monitor usage to avoid unexpected fees.

# AWS Free Tier and its limitations

**Filter by:**

[Clear all filters](#)

**Tier Type**

- Featured
- 12 Months Free
- Always Free
- Trials

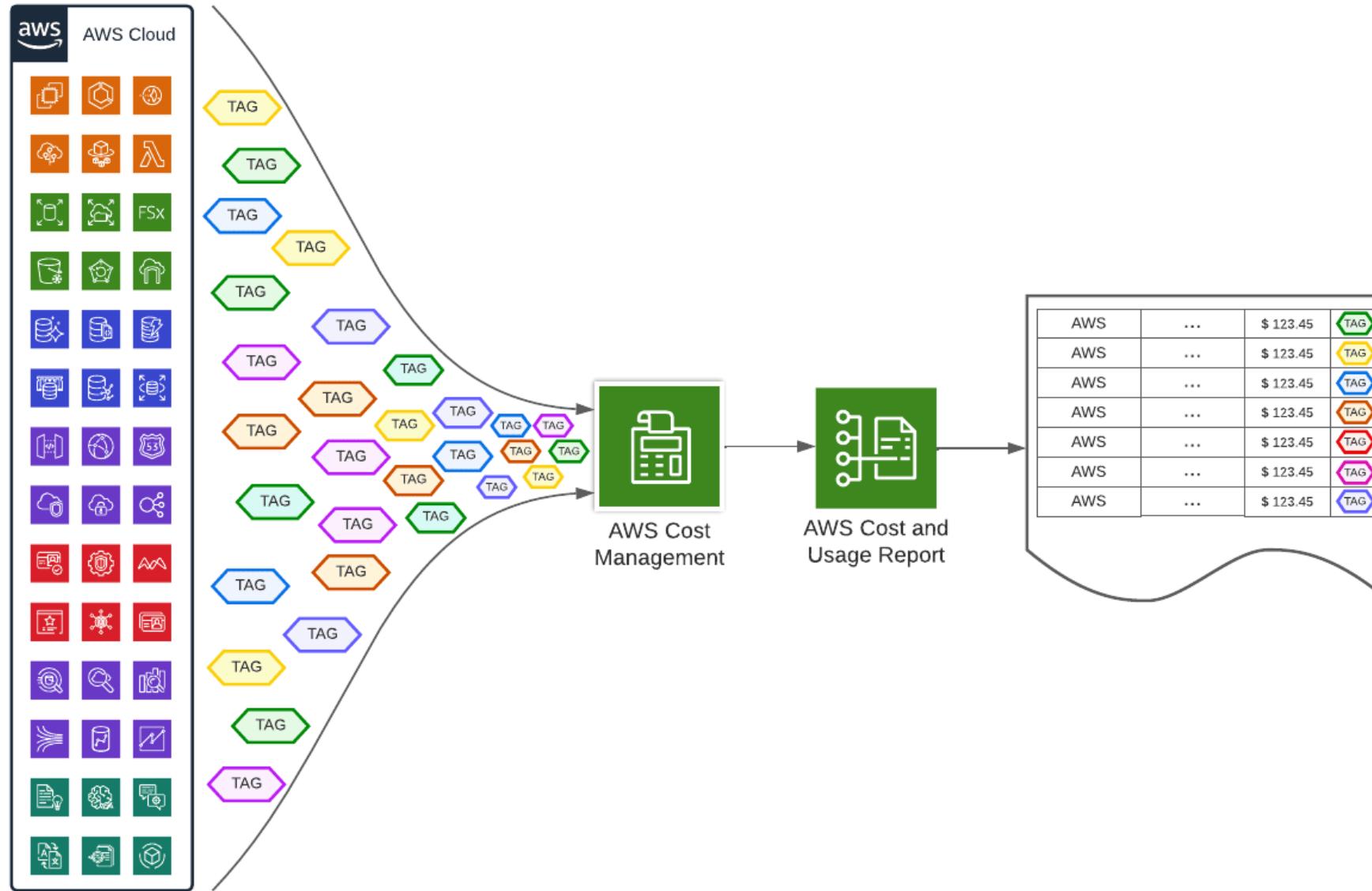
**Product Categories**

- Analytics
- Application Integration
- AR & VR
- Business Productivity
- Compute
- Customer Engagement
- Database
- Developer tools
- End User Computing
- Front-End Web & Mobile
- Game Tech
- Internet of Things
- Machine Learning
- Management & Governance

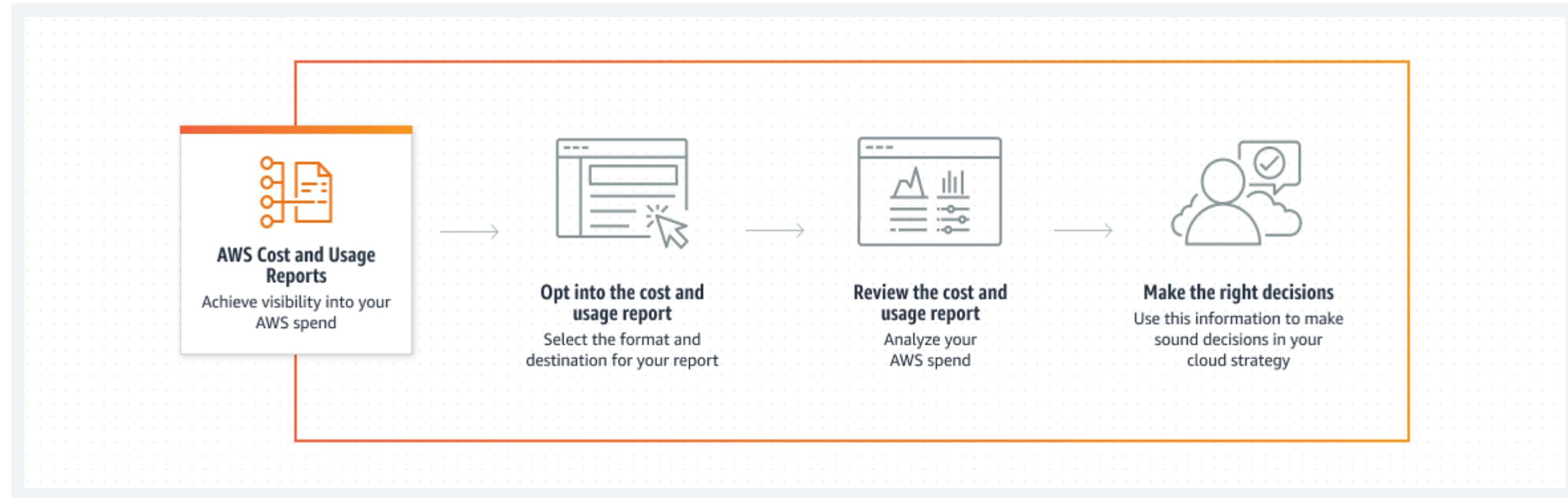
*Search free tier products*

<b>COMPUTE</b> Free Tier      12 MONTHS FREE <b>Amazon EC2</b> <b>750 Hours</b> per month Resizable compute capacity in the Cloud.  <small>View Details</small>	<b>STORAGE</b> Free Tier      12 MONTHS FREE <b>Amazon S3</b> <b>5 GB</b> of standard storage Secure, durable, and scalable object storage infrastructure.  <small>View Details</small>	<b>DATABASE</b> Free Tier      12 MONTHS FREE <b>Amazon RDS</b> <b>750 Hours</b> per month of db.t2.micro database usage (applicable DB engines) Managed Relational Database Service for MySQL, PostgreSQL, MariaDB, Oracle BYOL, or SQL Server.  <small>View Details</small>
<b>DATABASE</b> Free Tier      ALWAYS FREE <b>Amazon DynamoDB</b> <b>25 GB</b> of storage	<b>MACHINE LEARNING</b> Free Tier      FREE TRIAL <b>Amazon SageMaker</b> <b>250 Hours</b> per month of t2.medium notebook usage for	<b>COMPUTE</b> Free Tier      ALWAYS FREE <b>AWS Lambda</b> <b>1 Million</b> free requests per month

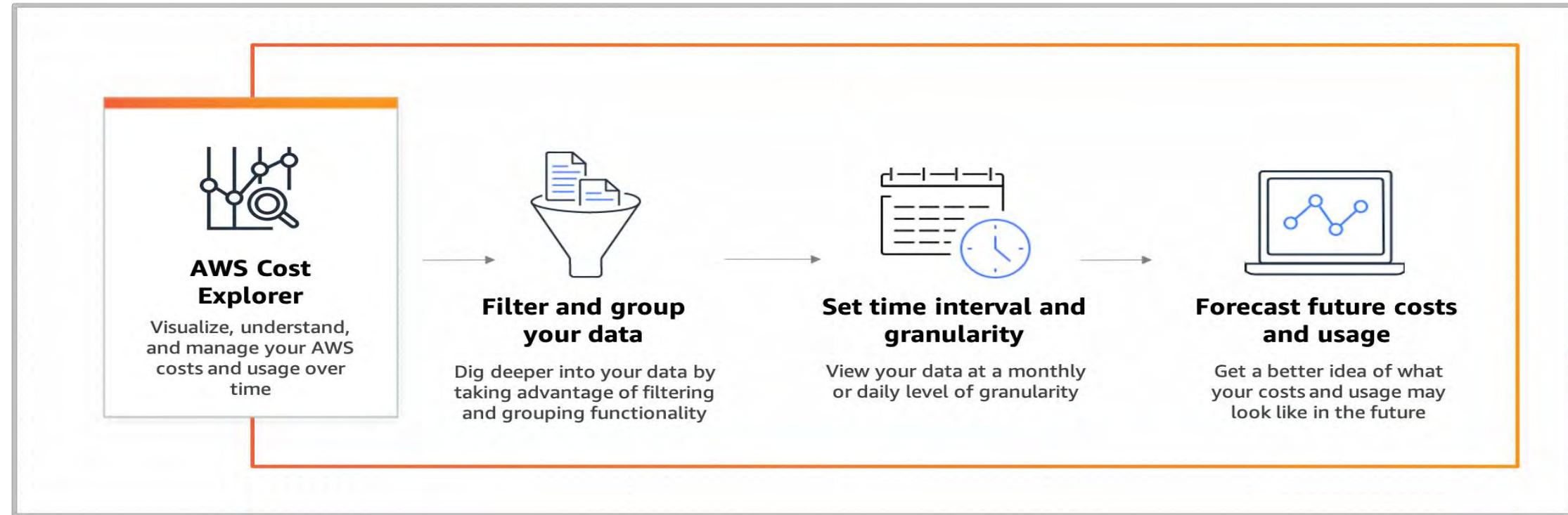
# Cost Allocation Tags and Billing Reports



# Cost and Usage Reports



# Cost visualization tools in the AWS



- AWS Cost Explorer has an easy-to-use interface that lets you visualize, understand, and manage your AWS costs and usage over time. Get started quickly by creating custom reports that analyze cost and usage data.
- Analyze your data at a high level or dive deeper into your cost and usage data to identify trends, pinpoint cost drivers, and detect anomalies.

# Pricing Models

## AWS Pricing Models

### Free Tier

- ❖ Free
- ❖ Opportunity to try new services
- ❖ Suitable for trials and testing
- ❖ Easy to Set Up
- ❖ Impractical for production grade use



### On-Demand

- ❖ No Commitment
- ❖ No Upfront Costs
- ❖ Highly Flexible
- ❖ Easy to Set Up
- ❖ Suitable for Short Term Projects
- ❖ Most Expensive Option



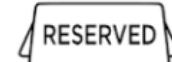
### Spot Instance

- ❖ No Commitment
- ❖ No Upfront Costs
- ❖ Limited Flexibility
- ❖ Can be Terminated with little notice
- ❖ Suitable for Fault Tolerant Apps
- ❖ Cheapest Option



### Reserved Instance

- ❖ 1 or 3 year Commitment
- ❖ Upfront Cost Option
- ❖ Limited Flexibility
- ❖ Suitable for Predictable apps
- ❖ Cheaper than On-Demand



### Savings Plan

- ❖ 1 or 3 year Commitment
- ❖ Upfront Cost Option
- ❖ Flexible
- ❖ Predictable Costs
- ❖ Easy to work with
- ❖ Cheaper than On-Demand

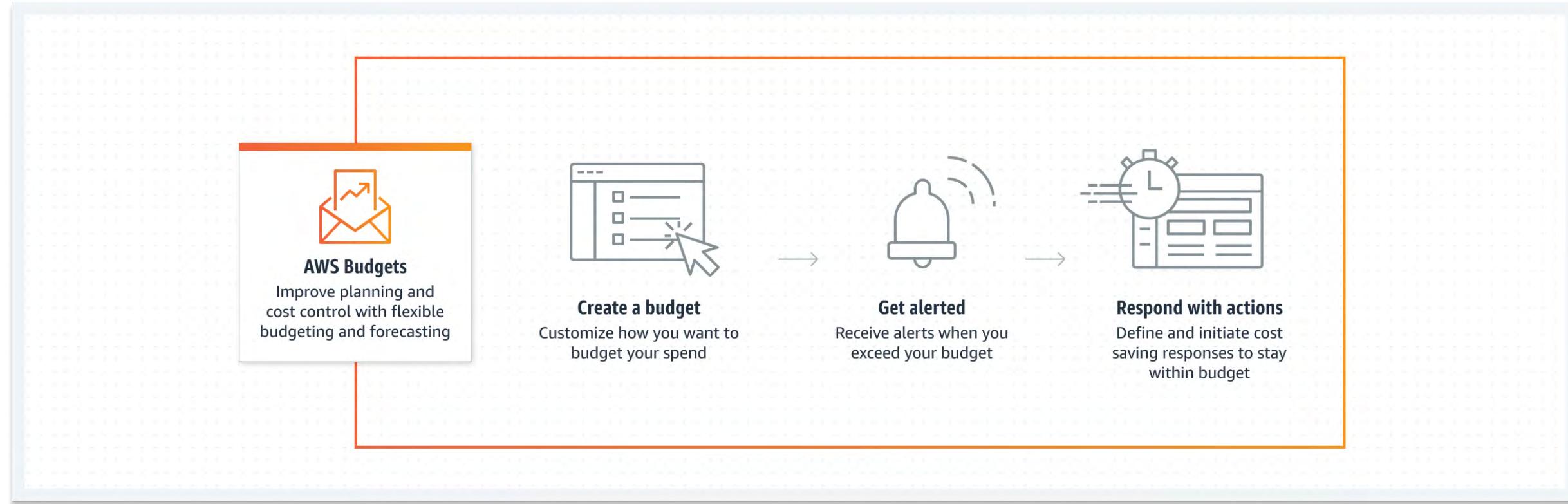


# AWS Pricing Calculator

---

- AWS Pricing Calculator is a web-based planning tool that you can use to create estimates for your AWS use cases.
- You can use it to model your solutions before building them, explore the AWS service price points, and review the calculations behind your estimates.
- You can use it to help you plan how you spend, find cost saving opportunities, and make informed decisions when using Amazon Web Services.

# AWS Budgets



With AWS Budgets, set custom budgets to track your costs and usage, and respond quickly to alerts received from email or SNS notifications if you exceed your threshold.

# AWS Budgets Use cases

---

- **Monitor costs and usage**

Set your preferred budget period to daily, monthly, quarterly, or annually, and create specific budget limits.

- **Create scheduled reports**

Stay informed on how actual or forecasted costs and usage progress toward your budget threshold.

- **Respond to thresholds**

Set up custom actions to run automatically or through an approval process when a budget target is exceeded.