

# AWS Certified Solutions Architect Professional (SAP-C01) – Exam Guide

[Mike Gibbs MS, MBA, CCIE #7417, GPC-PCA, AWS-CSA](#)

[Elonzo Coleman MBA, PMP, AWS-CSA](#)

[www.gocloudcareers.com](#)

[www.youtube.com/c/GoCloudArchitects](#)

AWS Certified Solutions Architect Professional (SAP-C01) – Exam Guide by Mike Gibbs MS, MBA, CCIE #7417, GCP- PCA, AWS-CSA & Elonzo Coleman MBA, PMP

Port Saint Lucie, FL 34953

[www.gocloudcareers.com](http://www.gocloudcareers.com)

[www.youtube.com/c/GoCloudArchitects](http://www.youtube.com/c/GoCloudArchitects)

© 2021 Mike Gibbs & Elonzo Coleman

All rights reserved. No portion of this book may be reproduced in any form without permission from the publisher, except as permitted by U.S. copyright law. For permissions, contact:

[elitechcareers@gmail.com](mailto:elitechcareers@gmail.com)

## Disclaimer

**LIABILITY:** Mike Gibbs, Elonzo Coleman, Go Cloud Architects, Go Cloud Careers, and its organizations and agents specifically **DISCLAIM LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES** and assume no responsibility or liability for any loss or damage suffered by any person as a result of the use or misuse of these materials or any of the information or content on our website, blogs, and social media posts. Mike Gibbs, Elonzo Coleman, Go Cloud Architects, Go Cloud Careers, and its organizations and agents assume or undertake **NO LIABILITY** for any loss or damage suffered as a result of the use or misuse of any information or content or any reliance thereon.

## Contributing Authors

We would like to thank the following individuals for contributing to this book:

[Abigayle Marks, BS, AWS CSA-A](#)

[Adenike Lanpejo BS, AWS-CCP](#)

[Amarakanthan Navaratnam, B.Eng., CCNA, GCP-ACE](#)

Andrew Akwumakwuhie, BSc Computer Science, PMP-PMI, Azure Solution Architect Expert, CCNA, CCNP.

[Babita Sinha, PhD, AWS CSA-A, Azure Solutions Architect, Azure Data Engineer](#)

[Balwinder Kaur, BS, AWS CSA-A](#)

[Daniel Pike, Cloud Architect](#)

[Eva Tenya, BS, BA, CPhT, EMT.](#)

[Garvin Collie, BS, AWS CSA-P. Palo Alto Security Engineer, AWS Advanced Networking](#)

[Genet Simegn](#)

[Jawad Irshad - Multi-Cloud Architect, B.Sc., AWS-CSA](#)

[Joe Whyte BS, AWS CSA-A](#)

Kemi Thomson MS, MSc, PMP, AWS-CCP

[Kenya Carl, BS, PE](#)

[Kishore Devarakonda BS, AWS CCP](#)

[Kris Fernando, BS](#)

[Lamin Drammeh, MSc](#)

[Leopoldo Prates, MS, AWS CSA-A](#)

Mark Hema, Network Engineer

Marla Weatherspoon Johnson, BBA

Millicent Yirenkyi, BS, CSA-P

Mohammad Lal Mahmud Mia

Muili Lawal BS, Azure Administrator Associate

Nararaj Malkar, MS, VCP, IBM Cloud Professional Solutions Architect

Nathan Vontz

Nick Love, BS

Nikhil Annigeri, MS, Citrix CCA-V

Pierre-Charles Dussault, Cloud Architect

Pritpal Sajjan BS, Teradata Certified Master, Informatica Certified Admin, AWS CCP

Priyadarshini Gurunath MBA, GCP-PCA, AWS CSA, MC-ASE

Reginald Duncan, BS Computer Network Management

Robert Welch BS, BA, Fortinet Network Associate, AWS CCP

Roberto Rosales, CCNP, CCDP, AWS CSA-A, AWS ANS

Ryan Perez BA, AWS CSA-P

Shakir Thajudeen, NCC Advance Diploma In Computing

Simenesh Woldesenbet, BSc in Engineering in Biotechnology (focus in Bioinformatics). Senior Research Assistant, OHSU

William Olaleye

## Table of Contents

<b><u>Chapter 1 Introduction to Cloud Computing</u></b>	1
<u>The History of the Enterprise Network and Data Center</u>	1
<u>Types of Cloud Environments</u>	2
<u>The AWS Cloud and How It's Organized</u>	5
<u>Local Zones</u>	6
<u>Edge Locations</u>	7
<u>How to Access and Manage Resources in the Cloud</u>	7
<b><u>Chapter 2 Connecting to the Cloud</u></b>	9
<u>Direct Connect</u>	9
<u>Letter of Authorization</u>	12
<u>AWS Direct Connect Partners</u>	12
<u>Public and Private Virtual Interfaces (VIF)</u>	12
<u>Direct Connect Gateway</u>	12
<u>Link Aggregation Groups (LAG)</u>	13
<u>Bidirectional Forwarding Detection</u>	13
<u>Billing</u>	14
<u>Virtual Private Networks (VPN)</u>	14
<u>High-Availability Connections</u>	16
<b><u>Chapter 3 Storage Options on the AWS Cloud Platform</u></b>	17
<u>AWS Primary Storage Options</u>	17
<u>Amazon Simple Storage Service (S3)</u>	18
<u>Elastic Block Storage (EBS)</u>	28
<u>RAID</u>	31
<u>Elastic File System (EFS)</u>	35
<u>AWS Storage Gateway</u>	36
<u>Migrating Data to AWS</u>	37
<u>Storage for Collaboration</u>	40
<u>Amazon FSx for Windows</u>	40
<u>Amazon FSx for Lustre</u>	41
<b><u>Chapter 4 Computing on the AWS Platform (EC2)</u></b>	42
<u>Amazon Machine Images (AMI)</u>	43
<u>Autoscaling</u>	44
<u>Instance Purchasing Options</u>	44
<u>Tenancy Options</u>	46
<u>Securing EC2 Access</u>	46
<u>IP Addresses for EC2 Instances</u>	48
<u>Accessing EC2 Instances</u>	48

<b><u>Chapter 5 Introduction to Databases</u></b>	49
<u>What Is a Database?</u>	49
<u>Relational Databases</u>	50
<u>NoSQL Databases</u>	54
<u>Data Warehousing on AWS</u>	57
<u>Data Lakes</u>	59
<u>Database Storage Options</u>	60
<u>Backing Up the Database</u>	60
<u>Scaling the Database</u>	62
<u>AWS Glue</u>	65
<u>Database Migration Tools</u>	66
<u>Designing a High-Availability Database Architecture</u>	66
<b><u>Chapter 6 The AWS Virtual Private Cloud</u></b>	68
<u>The OSI Model</u>	68
<u>IP Addressing, Subnetting, and Supernetting</u>	69
<u>Routing Tables and Routing</u>	74
<u>Internet Gateways</u>	77
<u>NAT Instances</u>	78
<u>NAT Gateway</u>	79
<u>Elastic IP Addresses (EIPs)</u>	80
<u>Endpoints</u>	81
<u>VPC Peering</u>	86
<u>AWS CloudHub</u>	89
<u>AWS Transit Gateway</u>	90
<u>Network Access Control Lists (NACL)</u>	91
<u>Security Groups</u>	92
<b><u>Chapter 7 AWS Network Performance Optimizations</u></b>	94
<u>Placement Groups</u>	94
<u>Elastic Fabric Adapter and SR-IOV</u>	97
<u>Amazon Route 53</u>	97
<u>Load Balancers</u>	107
<b><u>Chapter 8 Security</u></b>	113
<u>AWS Shared Security Model</u>	113
<u>Principle of Least Privilege</u>	114
<u>Industry Compliance</u>	115
<u>Identity and Access Management</u>	115
<u>Identity Federations</u>	120
<u>Creating IAM Policies</u>	125
<u>Further Securing IAM with Multifactor Authentication</u>	129
<u>Multi-Account Strategies</u>	130
<u>Preventing Distributed Denial of Service Attacks</u>	131

<a href="#"><u>Amazon Web Application Firewall (WAF)</u></a>	132
<a href="#"><u>AWS Shield</u></a>	134
<a href="#"><u>AWS Service Catalog</u></a>	135
<a href="#"><u>AWS Systems Manager Parameter Store</u></a>	136
<a href="#"><u>AWS Macie</u></a>	137
<a href="#"><u>AWS GuardDuty</u></a>	137
<a href="#"><u>AWS Inspector</u></a>	137
 <a href="#"><b>Chapter 9 AWS Applications and Services</b></a>	
<a href="#"><u>AWS Simple Queuing Service (SQS)</u></a>	138
<a href="#"><u>AWS Simple Notification Service (SNS)</u></a>	142
<a href="#"><u>AWS Simple Workflow Service (SWF)</u></a>	143
<a href="#"><u>AWS Elastic Map Reduce (EMR)</u></a>	144
<a href="#"><u>AWS Kinesis</u></a>	145
<a href="#"><u>AWS Elastic Container Service (ECS)</u></a>	149
<a href="#"><u>AWS Elastic Kubernetes Service (EKS)</u></a>	152
<a href="#"><u>AWS Elastic Beanstalk</u></a>	153
<a href="#"><u>AWS CloudWatch</u></a>	154
<a href="#"><u>AWS Config</u></a>	155
<a href="#"><u>AWS CloudTrail</u></a>	157
<a href="#"><u>AWS CloudFront</u></a>	158
<a href="#"><u>AWS Lambda</u></a>	160
<a href="#"><u>AWS Lambda@Edge</u></a>	162
<a href="#"><u>AWS Step Functions</u></a>	163
<a href="#"><u>AWS Rekognition</u></a>	163
<a href="#"><u>AWS CloudFormation</u></a>	164
<a href="#"><u>AWS Certificate Manager (ACM)</u></a>	164
 <a href="#"><b>Chapter 10 DevOps</b></a>	
<a href="#"><u>AWS CodePipeline</u></a>	166
<a href="#"><u>AWS CodeCommit</u></a>	166
<a href="#"><u>AWS CodeBuild</u></a>	167
<a href="#"><u>AWS CodeDeploy</u></a>	167
 <a href="#"><b>Chapter 11 Cost Optimization</b></a>	
<a href="#"><u>Financial Differences Between Traditional Data Centers and Cloud Computing</u></a>	168
<a href="#"><u>Optimizing Technology Costs on the AWS Cloud</u></a>	168
<a href="#"><u>AWS Budgets</u></a>	171
<a href="#"><u>AWS Trusted Advisor</u></a>	172
 <a href="#"><b>Chapter 12 Building High-Availability Architectures</b></a>	
<a href="#"><u>What is Availability?</u></a>	174
<a href="#"><u>Building a High-Availability Network</u></a>	174

<b><u>Chapter 13 AWS Labs - Getting Hands-On</u></b>	<b>179</b>
<b><u>    Chapter 1 Labs</u></b>	
<u>Create an EC2 Instance and SSH Into That Instance</u>	180
<b><u>    Chapter 3 Labs</u></b>	
<u>Create an S3 Bucket</u>	191
<u>Create an EBS Volume</u>	200
<u>Create an EC2 Instance and Mount EBS Volume</u>	206
<u>Create an EFS File System</u>	212
<b><u>    Chapter 4 Labs</u></b>	
<u>Create an EC2 instance and Install LAMP Stack</u>	219
<u>Create an Amazon Machine Image (AMI)</u>	230
<u>Create a Security Group for an EC2 Instance</u>	235
<b><u>    Chapter 5 Labs</u></b>	
<u>Create an Amazon Aurora Database</u>	237
<u>Set Up ElastiCache for Database Caching</u>	245
<u>Create a Redshift Data Warehouse</u>	247
<u>Set Up ElastiCache with Aurora Serverless</u>	256
<u>Set Up Simple Queue Service for DB Queueing</u>	271
<u>Create a Table in Dynamo DB</u>	274
<u>Data Migrations Service &amp; Schema Conversion Tool</u>	277
<b><u>    Chapter 6 Labs</u></b>	
<u>Create a VPC</u>	282
<u>Create a Network Access Control List (NACL)</u>	285
<u>Create an NAT Gateway</u>	287
<u>Create an Elastic Network Interface (ENI)</u>	289
<u>Create an Internet Gateway</u>	292
<u>Set Up VPC Peering</u>	294
<b><u>    Chapter 7 Labs</u></b>	
<u>Create a Route 53 Hosted Zone</u>	307
<u>Create a Route 53 Health Check</u>	312
<u>Set Up an Application Load Balancer</u>	315
<u>Set Up a Cluster Placement Group</u>	328
<b><u>    Chapter 8 Labs</u></b>	
<u>Create an IAM User</u>	335
<u>Set Up MFA on the Root Account</u>	341
<u>Set Up Static Website Hosting with S3 and CloudFront</u>	344
<u>Set up WAF, Shield, and Cloud Front</u>	352

<u><a href="#">Set Up AWS Organizations</a></u>	363
<b><u>Chapter 9 Labs</u></b>	
<u><a href="#">Set UP ECS/EKS containers</a></u>	370
<u><a href="#">Create SNS Topic and Set Up Email Notifications</a></u>	385
<u><a href="#">Set Up Amazon Kinesis Streams</a></u>	395
<u><a href="#">Chapter 14 Passing the AWS Exam</a></u>	401
<u><a href="#">Chapter 15 Practice Exam</a></u>	403

## **Chapter 1 - Introduction to Cloud Computing**

### **The History of the Enterprise Network and Data Center**

Ever since computing resources provided a competitive advantage, organizations have been investing in their computing resources. Over time technology became not only a competitive advantage but also a necessary resource to be competitive in today's business environment. Since technology can bring extreme advances in productivity, sales, marketing, communication, and collaboration, organizations have invested more and more resources into technology. Organizations, therefore, built large and complex data centers and connected those data centers to an organization's users with specialized networking, security, and computing hardware resources. Enterprise data centers became huge networks, often requiring thousands of square feet of space; incredible amounts of power; cooling; hundreds, if not thousands, of servers, switches, routers; and many other technologies. Effectively, the enterprise and especially the global enterprise environments became massive networks—just like a cloud computing environment. The net result was a powerful private cloud environment.

Global enterprise data centers and high-speed networks work well. However, these networks come with a high cost and require a high level of expertise to manage these environments. Network and data center technology is not simple, and it requires a significant staff of expensive employees to design, operate, maintain, and fix these environments. Some large enterprise technology environments take billions of dollars to create and operate.

Global enterprise networks and data centers still have merit for high security, ultrahigh-performance environments requiring millisecond-level latency, and ultrahigh network and computing performance. An example environment that benefits from this traditional model is the global financial environment, where shaving milliseconds off network, server, and application performance can equate to a significant competitive advantage. However, for many customers, the costs of procuring and operating the equipment are just too costly.

Recent advances in network, virtualization, and processing power make the transition to cloud computing feasible.

### **Why Now Is the Optimal Time for Cloud Computing**

Let's first look at virtualization, as it is the key enabling technology of cloud computing. Server performance has increased dramatically in recent years. It is no longer necessary to have a server or multiple servers for every application. Since today's servers are so powerful, they can be partitioned into multiple logical servers in a physical server, reducing the need for so many servers. This reduces space, power, and cooling requirements. Additionally, virtualization makes it simple to move, add, or change server environments. Previously, any time you wanted to make a server change, you had to buy a new server, which could take weeks or months, install the operating system and all the applications' dependencies, and then find a time to

upgrade the server when it wasn't being used. This process was lengthy, as changes to any part of an IT environment can effect many other systems and users. With virtualization, to upgrade a server, you have to copy only the server file to another machine, and it's up and running.

Server virtualization became so critical in improving hardware resource utilization for computing that soon organizations explored moving the network to virtualized servers. Now routing, firewalling, and many other functions can be shifted to virtualized services with software-defined networking. For several years organizations have migrated their traditional data centers to virtualized enterprise data centers, and it has worked well. However, network speed (bandwidth) has made significant gains, while the cost of this high-performance networking has decreased substantially.

Therefore, it is now possible to move the data center to a cloud computing environment and still achieve high performance with lower total costs. With the ability to purchase multiple 10-gigabit-per-second links to AWS, it's now feasible to connect an organization to a cloud provider at almost the same speed as if the application is in the local data center, but with the benefits of a cloud computing environment.

## **Types of Cloud Environments**

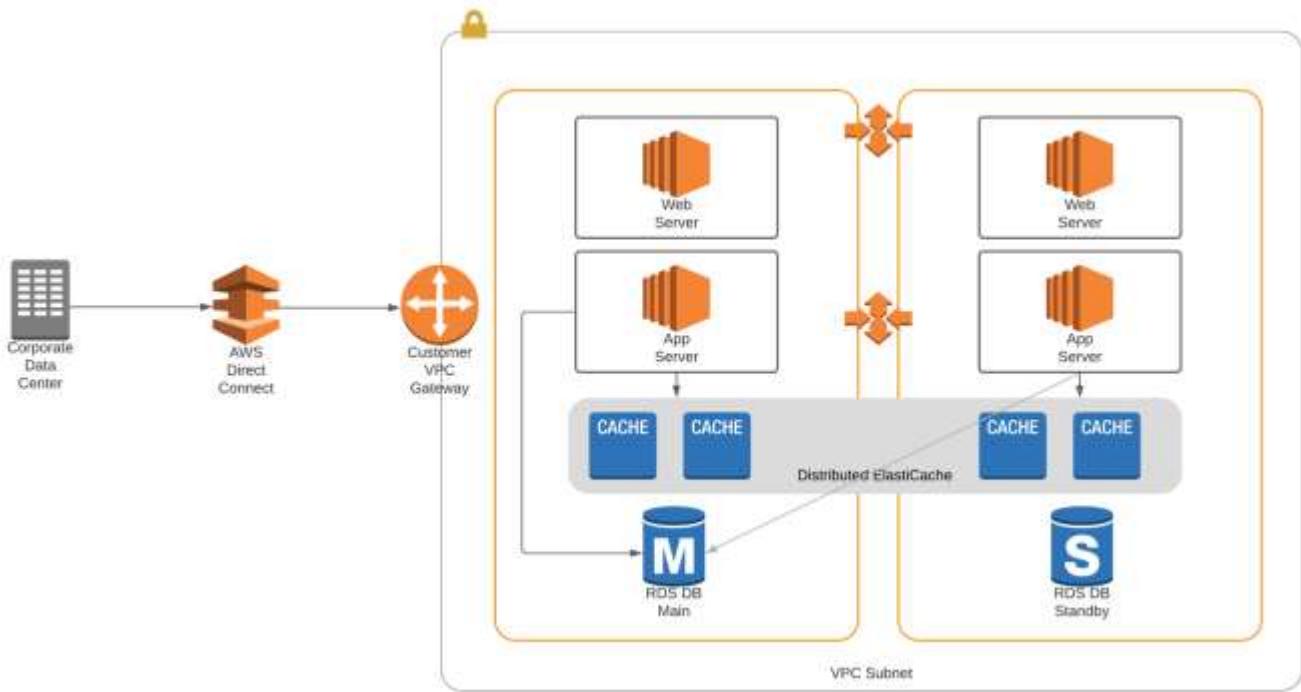
### **Hybrid Cloud**

A hybrid cloud combines a standard data center with outsourced cloud computing. For many organizations, the hybrid cloud is the perfect migration to the cloud. In a hybrid architecture, the organization can run its applications and systems in its local data center and offload part of the computing to the cloud. This provides an opportunity for the organization to leverage its investment in its current technology while moving to the cloud. Hybrid clouds provide an opportunity to learn and develop the optimal cloud or hybrid architecture.

Applications for hybrid cloud include:

- Disaster recovery – Run the organization computing locally, with a backup data center in the cloud.
- On-demand capacity – Prepare for spikes in application traffic by routing extra traffic to the cloud.
- High performance – Some applications benefit from the reduced latency and higher network capacity available on premises, and all other applications can be migrated to the cloud to reduce costs and increase flexibility.
- Specialized workloads – Move certain workflows to the cloud that require substantial development time, i.e., machine learning, rendering, transcoding.
- Backup – The cloud provides an excellent means to back up to a remote and secure location.

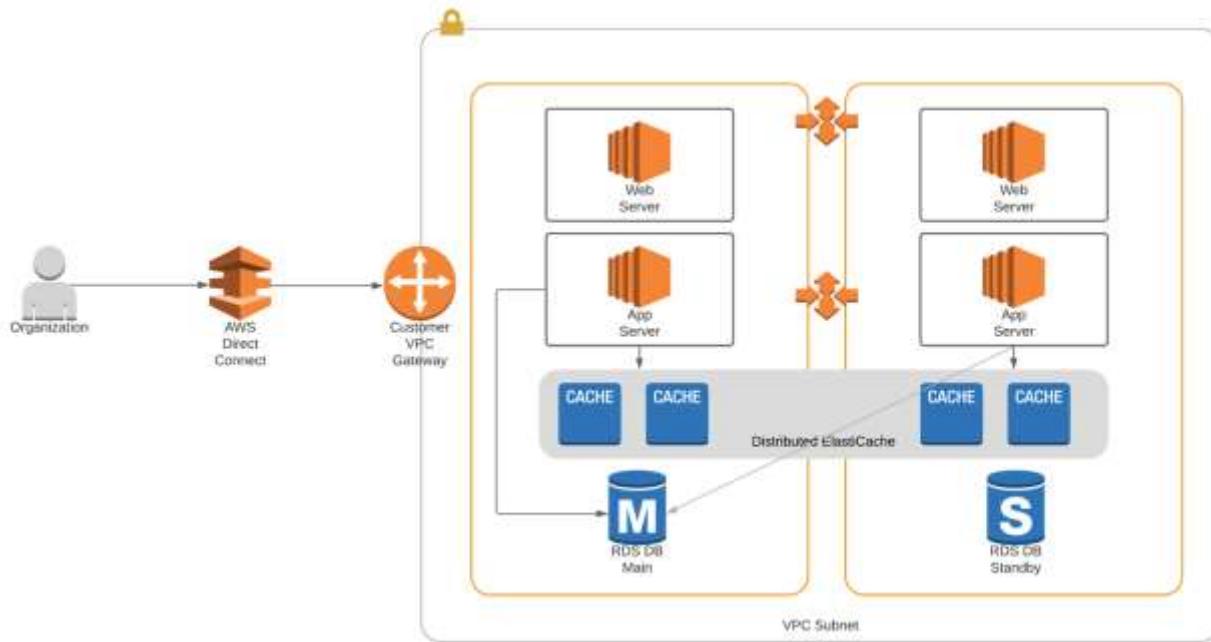
The diagram below shows an example of a hybrid cloud computing environment:



### Pure Cloud Computing Environment

In a pure cloud computing environment, all computing resources are in the cloud. This means servers, storage, applications, databases, and load balancers are all in the cloud. The organization is connected to the cloud with a direct connection or a VPN connection. The speed and reliability of the connection to the cloud provider will be the key determinant of the performance of this environment.

The diagram below shows an example of a pure cloud computing environment on the AWS platform:



A pure cloud computing environment has several advantages:

- Scalability – The cloud provides incredible scalability.
- Agility – Adding computing resources can occur in minutes versus weeks in a traditional environment.
- Pay-as-you-go pricing – Instead of purchasing equipment for maximum capacity, which may be idle 90 percent of the time, exactly what is needed is purchased when needed. This can provide tremendous savings.
- Professional management – Managing data centers is very complicated. Space, power, cooling, server management, database design, and many other components can easily overwhelm most IT organizations. With the cloud, most of these are managed for you by highly skilled individuals, which reduces the risk of configuration mistakes, security problems, and outages.
- Self-healing – Cloud computing can be set up with health checks that can remediate problems before they have a significant effect on users.
- Enhanced security – Most cloud organizations provide a highly secure environment. Most enterprises would not have access to this level of security due to the costs of the technology and the individuals to manage it.

## **Multi-Cloud Environment**

Organizations are now leveraging multiple cloud providers. This enables an organization to obtain the best benefits from each cloud provider. For example, cloud provider 1 has the best artificial intelligence/machine learning tools, and cloud provider 2 has the best infrastructure. Additionally, a multi-cloud environment protects an organization from an outage in a single cloud provider.

Multi-cloud environments also provide the following benefits:

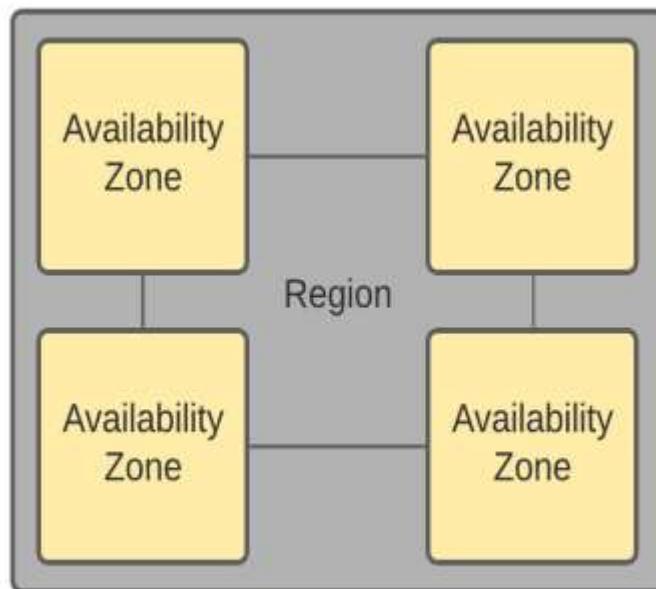
- High-availability – A single cloud provider is a single point of failure.
- Vendor lock-in – Being on multiple clouds protects the customer from a provider raising their rates, etc. It gives the customer negotiation power on pricing and services.

## **The AWS Cloud and How It's Organized**

The AWS cloud is organized into regions, availability zones, local zones, and edge locations. Each of these components performs a specific function in delivering cloud services to AWS clients.

The main components of the AWS cloud are **regions** and **availability zones**. A region is a large geographic area, i.e. a continent or part of a continent. An availability zone is simply a data center. A geographic region will have many data centers called availability zones.

Please see the diagram below:



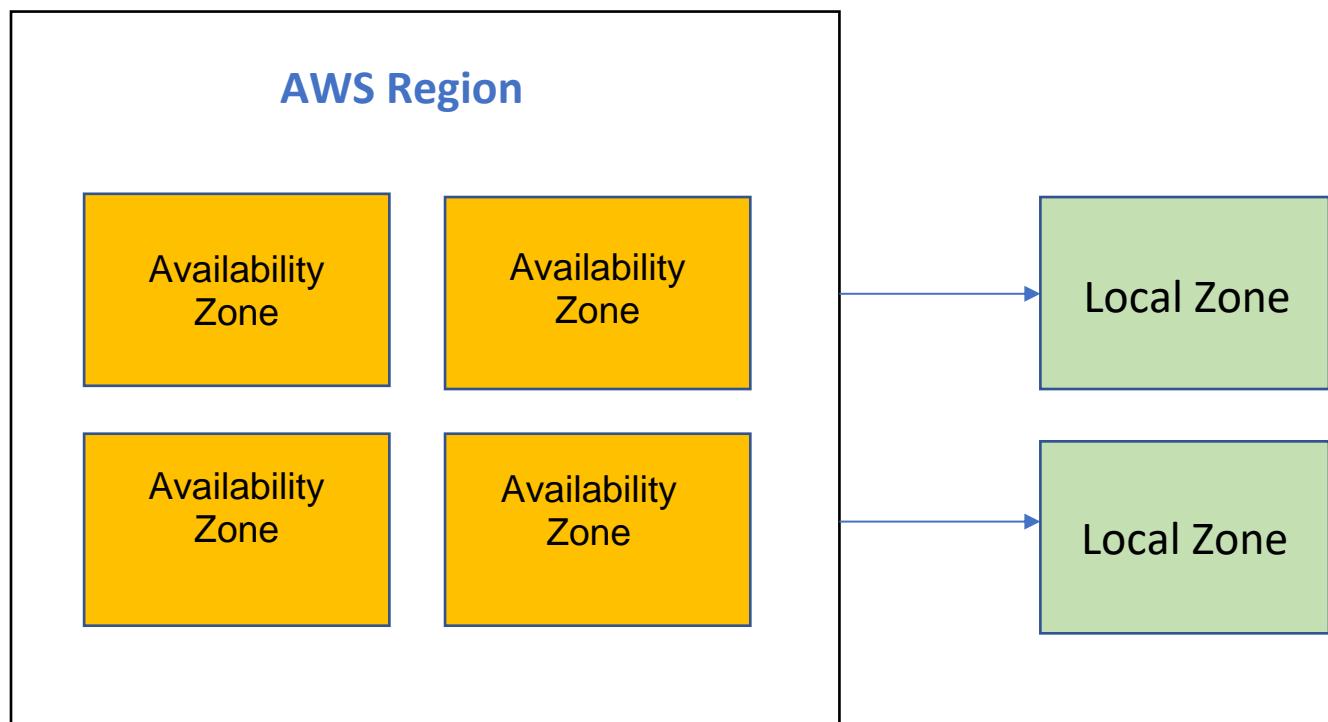
## Local Zones

Cloud computing often occurs at a significant distance from the customer's location. Sending and receiving data over a long distance can contribute to high levels of latency. High levels of latency can affect user experience as well as application performance. To that end, AWS offers the **local zone**. The local zone is an extension of the region. Effectively it's a data center closer to the user. Organizations place systems that require low latency in these local zones.

Things to note about local zones:

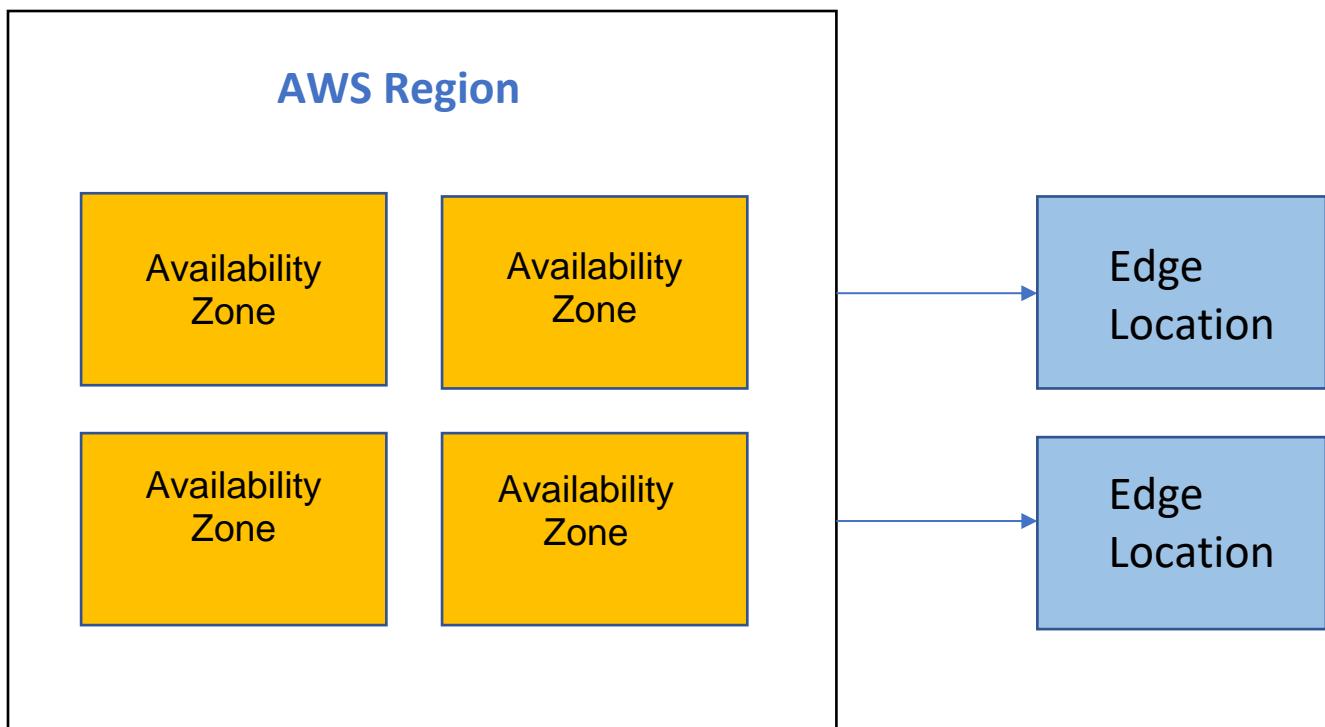
- To use a local zone, first opt into using local zones.
- Create a subnet in the local zone.
- Place systems in the local zone.

This diagram shows how the local zone is an extension of the region for low-latency computing:



## Edge Locations

Edge locations are used by the CloudFront content delivery network. Edge locations provide access to the CloudFront content delivery network in most major cities. Edge locations are used by CloudFront to improve internet performance and improve website security and scalability. CloudFront will be discussed in depth in the CloudFront section of this book.



## How to Access and Manage Resources in the Cloud

There are three ways to manage cloud computing resources on AWS. The methods to configure AWS cloud computing resources are the AWS Management Console, the command-line interface, and connecting via an API through the software development kit.

### **AWS Management Console**

The AWS Management Console is a simple-to-use, browser-based method to manage configurations. There are numerous options, with guidance and help functions. Most users will use this platform.

## **AWS CLI (Command Line Interface)**

The AWS Command Line Interface (CLI) enables you to manage computing resources via Linux commands and JavaScript Object Notation (JSON) scripting. This is efficient but requires more knowledge and training, and organizations need to know exactly what is needed and how to configure properly. With the CLI, there is no guidance as in the AWS Management Console.

## **AWS SDK (Software Development Kit)**

The AWS Software Development Kit (SDK) provides a highly effective method to modify and provision AWS resources on demand. This can be automated and can provide the ultimate scalability. This will require sophisticated knowledge to build the programming resources and is recommended for experts.

## **Chapter 2 - Connecting to the Cloud**

If an organization moves its computing environment to the cloud, then the connections to the cloud become critical. If the connection to the cloud fails, then the organization can no longer access cloud resources. The performance needs and an organization's dependency on IT will determine the connection requirements to the cloud.

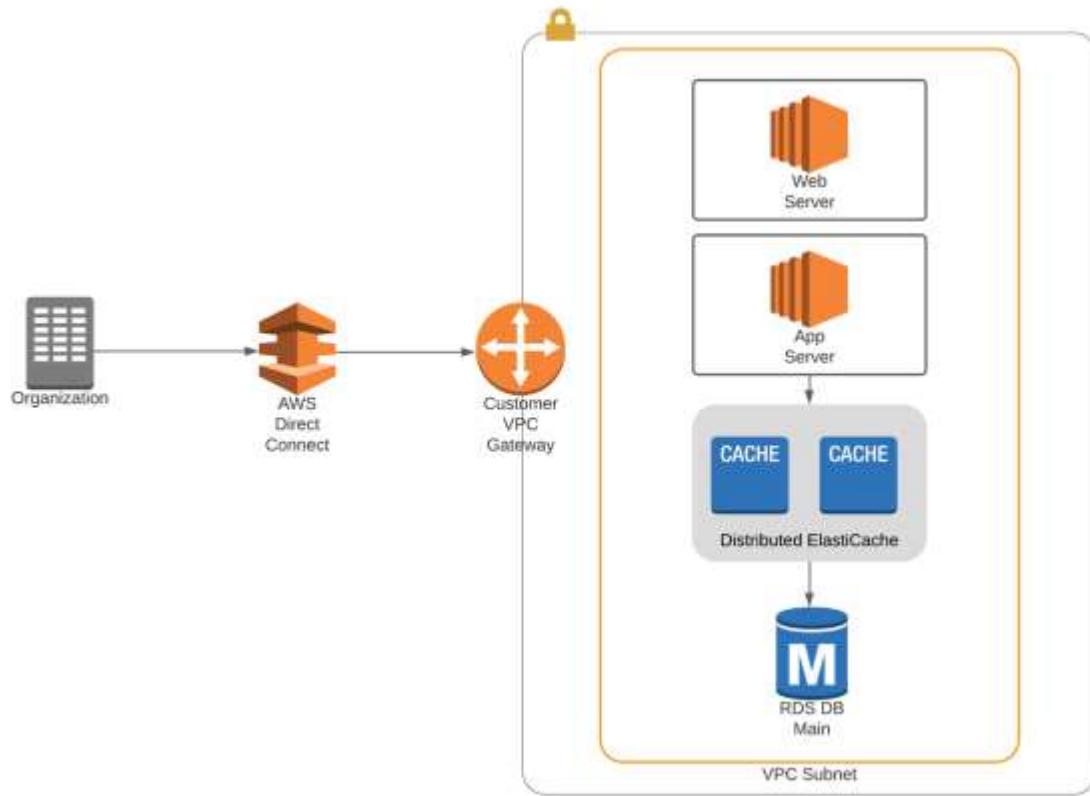
For most organizations, getting a "direct" connection to the cloud will be the preferred method. A direct connection is analogous to a private line in the networking world because it is effectively a wire that connects the organization to the cloud. This means guaranteed performance, bandwidth, and latency. As long as the connection is available, performance is excellent. This is unlike a VPN connection over the internet, where congestion anywhere on the internet can negatively affect performance.

Since network connections can fail, a direct connection is generally combined with a VPN backup over the internet. A VPN can send the data securely over the internet to AWS. A VPN provides data security via encryption and permits the transfer of routing information and the use of private address space. VPNs work by creating an IP security (IPsec) tunnel over the internet.

### **Direct Connect**

Organizations have two options when connecting to the cloud: a dedicated network connection or a VPN connection. An AWS direct connection is the equivalent of a private line between a customer's on-premises data center and the AWS network. A private line is analogous to a wire between the organization's data center and the AWS cloud. Since the private line is effectively a wire, there will be consistent speed and latency. The ability to get a guaranteed speed and consistent latency makes a direct connection the best option when an organization needs guaranteed or consistent network performance.

The diagram below shows a high-level logical view of the AWS direct connection.



### Key Underlying Technologies

The concept of a direct connection is based on Ethernet WAN (wide area network) technologies that provide dedicated point-to-point network connectivity between multiple locations. Key components include a fiber-optic router with a fiber-optic Ethernet port and, of course, a single-mode fiber-optic connection. Speeds available for the AWS Direct Connect include 1 Gbps, 10 Gbps, and 100 Gbps Ethernet. If additional bandwidth is desired, up to four connections can be bundled into a *link aggregation group*. The link aggregation group (LAG) will be covered in another section.

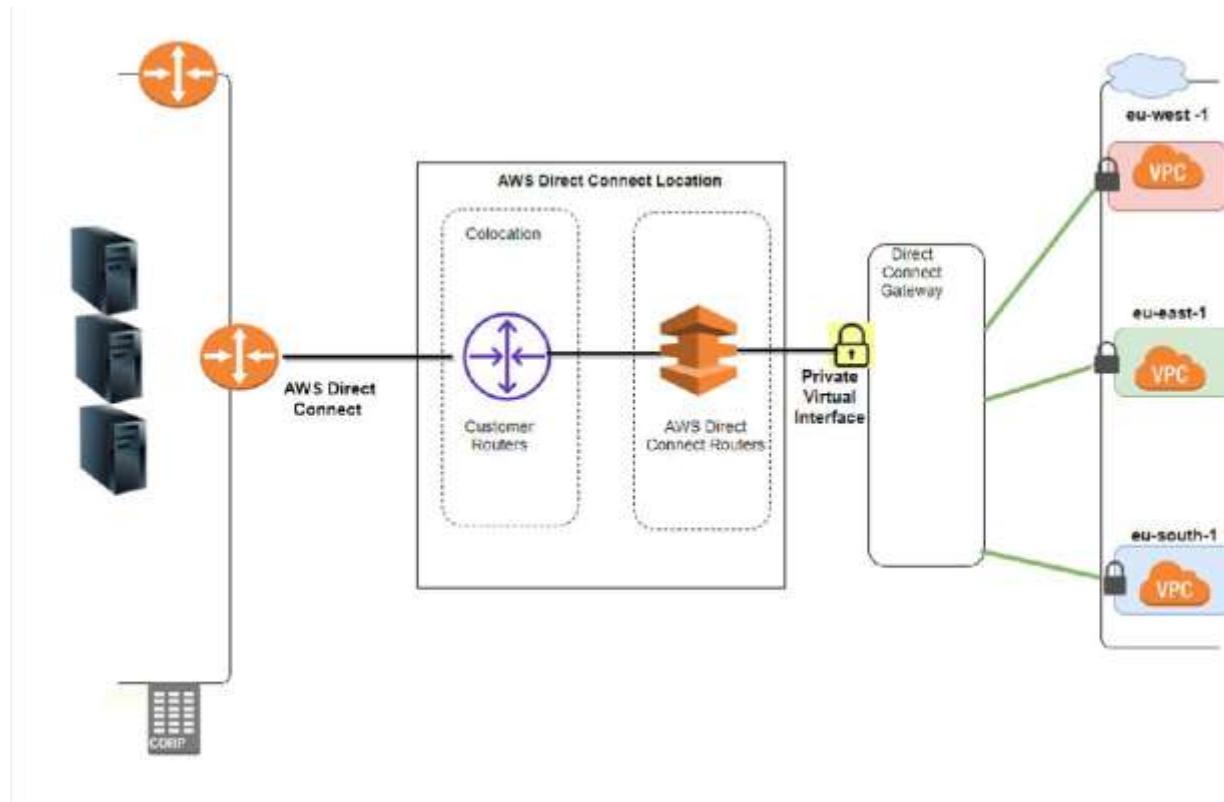
In addition, Bidirectional Forwarding Detection (BFD) is supported to help make sure traffic is only sent on healthy, fully active connections.

### How Does a Direct Connection Work?

On the surface, AWS Direct Connect gives the illusion of a wired connection, but several actions occur behind the scenes. The customer buys a connection from their data center to an AWS Direct Connect location. From there, a connection is made from the customer router in the direct connection location to the AWS switch. This occurs by connecting a cable between the

customer router and the AWS switch. This cable connection is called a *cross-connect*. Traffic is then backhauled over the AWS backbone network to the VPC. Customers can access the AWS Direct Connect console to build virtual interfaces when the network connection is established.

The diagram below shows the actual technology involved in a direct connection:



### Direct Connect Requirements

Direct Connect requires a virtual local area network (VLAN) tag on a direct connection. Each VLAN is tagged with an 802.1q tag. The 802.1q tag keeps customers' traffic separated on the layer 2 connections back to AWS.

All direct connection routers must support Border Gateway Protocol (BGP). BGP is used to build routing tables (network layer reachability information) which are used for traffic forwarding decisions. Additionally, the BGP implementation must support BGP Message Digest 5 (MD5) authentication via a customer authentication key or an AWS-generated key. It is recommended to use Bidirectional Forwarding Detection (BFD) so that traffic will fail over to a secondary circuit in the event of a failure.

## **Letter of Authorization**

To connect the customer network to the AWS network, a Letter of Authorization – Connecting Facility Assignment (LOA-CFA) must be granted. A LOA-CFA gives an internet service provider authority to act on behalf of a customer. An LOA-CFA application must be submitted via the management console, API, or CLI, identifying the Direct Connect region. Once complete, AWS will provision a switchport to the Direct Connect Location. Finally, the letter is submitted to the service provider to establish a cross-connect.

## **AWS Direct Connect Partners**

A Direct Connect Partner is an authorized service provider that provisions the direct connection from an on-premises data center to a customer's AWS Virtual Private Cloud (VPC). Direct Connect Partners can provide sub-Gbps connections if full-speed ports are not needed. Sub-Gbps connections drastically reduce the cost over full-speed ports. Once a port speed is determined, the Direct Connect Partner will establish a rate limiting policy. This is useful for organizations that only need partial speeds. In most cases a direct connect partner will be more expensive per GB of data transferred but cost less overall for organizations not needing full link speed.

## **Public and Private Virtual Interfaces (VIF)**

The direct connection offers access to both public and private interfaces. Public and private virtual interfaces (VIF) provide customers with the opportunity to connect to both public and private VIFs. A private VIF is used to securely access a VPC using private IP addresses, and a public VIF can access all public services using public IP addresses. AWS allows customers to connect to multiple VPCs in any AWS Region.

Public VIFs can be used to reach AWS services like S3, DynamoDB, and public subnets, whereas private VIFs are used to connect to virtual gateways (VGW). AWS allows customers to advertise up to one hundred routes over a BGP peering session but will not permit more than one thousand routes per customer. AWS will not readvertise routes, so customers can't become a transit ISP.

## **Direct Connect Gateway**

Direct Connect gateway (DCG) is a service built over top Direct Connect and provides the ability to connect VPCs in multiple availability zones (AZ) or Regions. Once connected to a DCG, a logical private connection is created between a customer's VPC in the selected AZ. Afterward, BGP peering is set up and connected to the gateway, and CIDR addresses will not overlap.

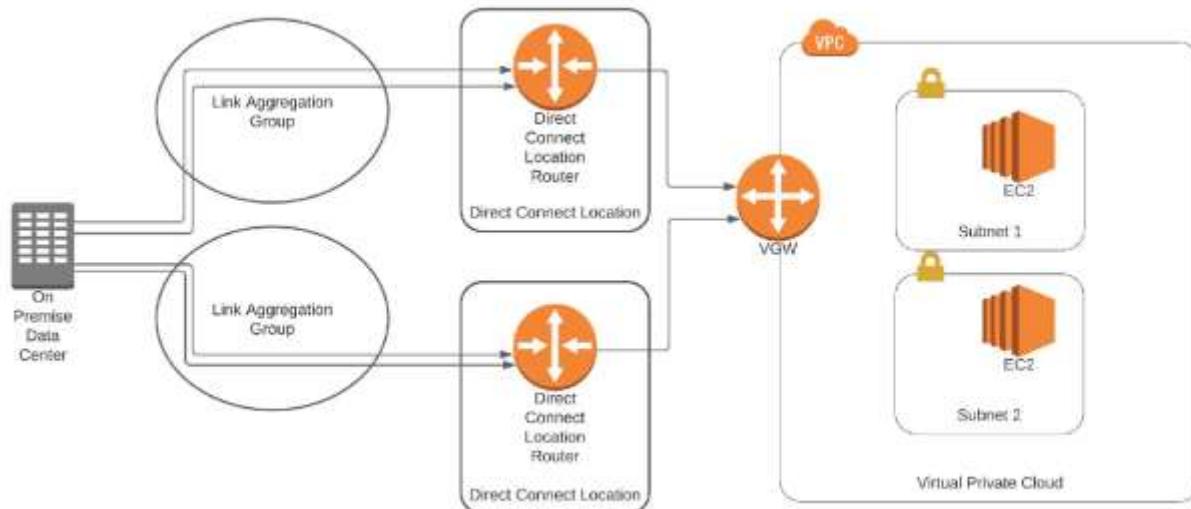
## Link Aggregation Groups (LAG)

Some organizations need high-speed access to the cloud, which can't be achieved through a single direct connection. **Link aggregation groups** (LAGs) allow customers to combine multiple Ethernet links into a single logical connection, increasing throughput and fault tolerance.

AWS supports LAGs that enable the combination of up to four 1 GB or 10 GB connections into a single logical port. All links must be the same type, speed, and latency. These requirements can best be achieved by using the same service provider.

By default, the LAG will be active for as long as the connections are available, adding speed and redundancy. However, this can be disabled if there is a need for a higher-performance backup solution. If an organization needs serious high-speed access, then it will likely need a redundant LAG connection.

An example of redundant LAGs can be seen below:



## Bidirectional Forwarding Detection

AWS supports Bidirectional Forwarding Detection (BFD), a network protocol that detects faults between two routers or switches within milliseconds or even microseconds. With fiber-optic network connections, there are two connections, a *send link* and a *receive link*. If the send link is up and the receive link is down, the network devices may not notice the down connection. BFD looks for any link failures (send or receive) and shuts down a non-fully functional link. It provides low-overhead detection of link failures, even on physical devices that do not support failure detection. BFD essentially notifies BGP of a broken connection, and BGP reroutes the traffic to a secondary (failover) circuit, if available.

## **Billing**

AWS Direct Connect has three separate charges that determine pricing: capacity, port hours, and data transfer out (DTO). There are no setup charges, and services can be canceled at any time. As with any other AWS service, data transfers into AWS via Direct Connect are free; however, services provided by AWS Direct Connect Partners may have additional terms or restrictions that apply.

Capacity is the maximum rate that data can be transported through a network connection.

Pricing of ports is dependent on per port hours consumed for each port type. Partial port hours consumed are billed as full hours and charged to the AWS account that owns the port. Hours will still be charged even if data is not passing through the port.

Data Transfer Out (DTO) is determined by the AWS Region and the AWS Direct Connect location. Consider the cost of the private circuit from the data center to the AWS location.

## **Virtual Private Networks (VPNs)**

The alternative to a direct connection is a virtual private network (VPN). A virtual private network is the creation of a secure connection inside of a public network. There are multiple types of VPN technologies: IPsec, L2TP, GRE, even multiprotocol BGP with virtual routing and forwarding. VPNs that are used to connect to AWS are based upon IPsec.

### **VPN Connection to AWS**

The simplest and cheapest means to connect to AWS is a VPN. A VPN provides a means to “tunnel” traffic over the internet in a secure manner. Modern VPNs use IPsec or SSL technologies. The VPN technology used to connect to AWS is an IPsec tunnel. Encryption is provided by IPsec, which provides a means to provide encryption (privacy), authentication (identifying the user), data authenticity (meaning the data has not been changed), and nonrepudiation (meaning, the user can't say they didn't send the message after the fact).

### **VPN Performance**

VPNs are inherently flexible, as all that is needed is internet access on both sides of the connection. However, the problem with VPN connections is that while the connection speed to the internet is guaranteed, there is no control of what happens on the internet. Because of this, there can be substantial performance degradation based upon the availability, routing, and congestion on the internet. VPN-only connections are ideal for remote workers and small branches of a few workers; where if they lose connectivity, there will not be significant costs to the organization.

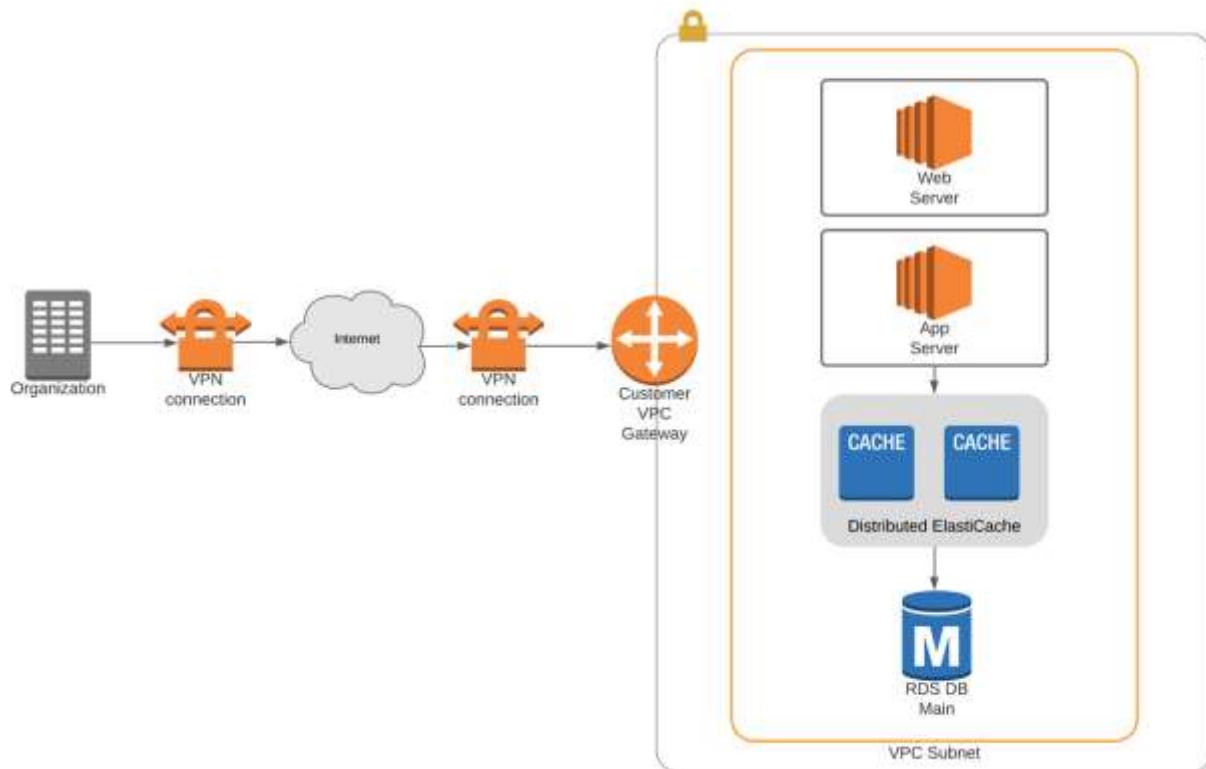
Advantages of using VPNs can be seen below:

- Cost – generally less expensive than direct connections
- Speed – can be set up in minutes
- Simplicity and flexibility – simple to create remote connections

Disadvantages of using VPNs for remote access:

- Lower performance than direct connections.
- Internet performance not consistent.
- Inconsistent bandwidth and latency.
- Dependent upon the routing of multiple ISPs.
- Once your data gets to the internet it is best effort – no delivery of message delivery.

The diagram below shows an example of a VPN connecting to the AWS platform.

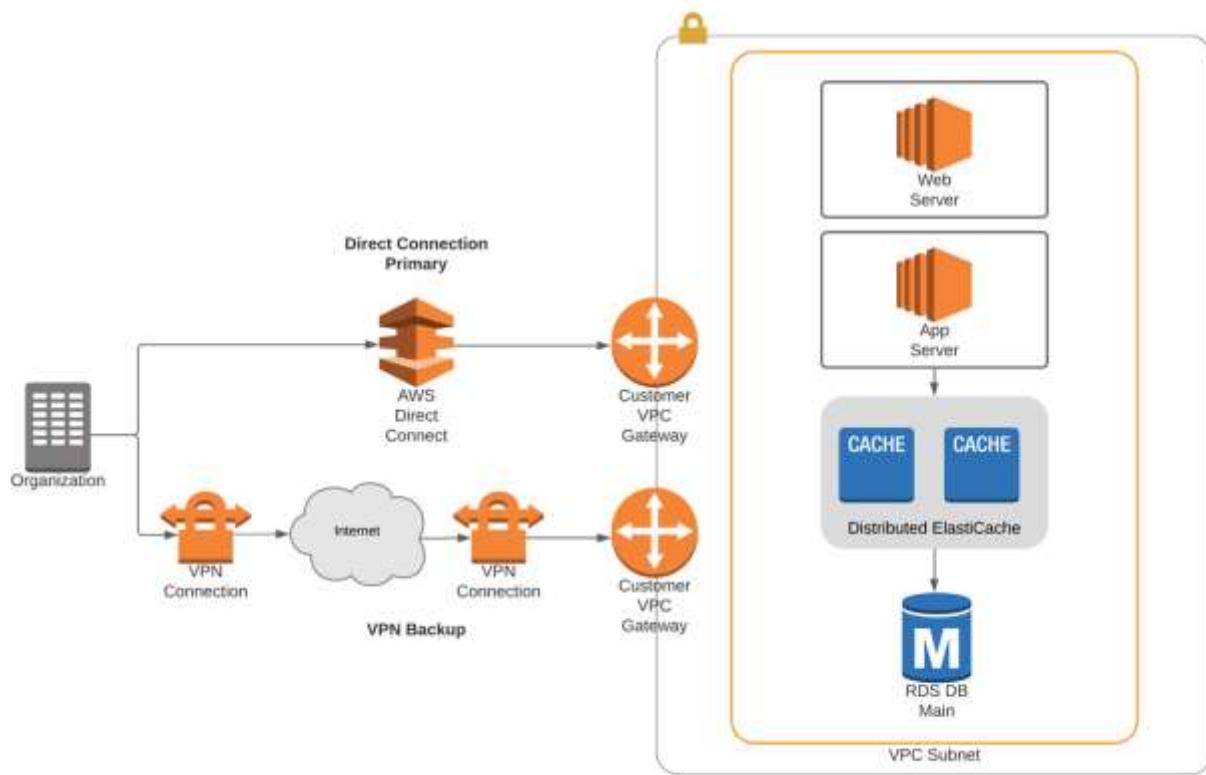


## High-Availability Connections

Connecting to the cloud with high availability is essential when an organization depends upon technology.

The highest-availability architectures will include at least two direct connections to the cloud. Ideally, each connection is with a separate service provider and a dedicated router, and each router is connected to different power sources. This configuration provides redundancy to the network connection, power failures, and the routers connecting to the cloud. For organizations that need 99.999 percent availability, this type of configuration is essential. For even higher availability, there can be a VPN connection as a backup to the other direct connections.

The diagram below shows an example of a high availability connection to the AWS platform.



## **High Availability at Lower Costs**

A lower cost means to achieve high availability is to have a dedicated connection and a VPN backup to the cloud. This will work well for most organizations, assuming they can tolerate reduced performance when using the backup environment.

## **Chapter 3 - Storage Options on the AWS Cloud Platform**

There are several storage options available to AWS cloud computing customers. These are AWS Simple Storage Service (S3), Elastic Block Store, Elastic File System, Storage Gateway, and WorkDocs.

In traditional data centers, there are three kinds of storage—block storage, object storage and file storage. Block storage is used to store data files on storage area networks (SANs) or cloud-based storage environments. It is excellent for computing situations where there is a need for fast, efficient, and reliable data transportation. File storage is stored on local systems, servers, or network file systems. Object storage is often used for backup, archival, and big data environments.

### **AWS Primary Storage Options**

In the AWS cloud environment, there are three types of storage: *block storage*, *object storage*, and *file storage*.

#### **Block Storage**

Block storage is a type of network storage that feels and functions like a hard drive in a computer. Block storage places data into blocks and then stores those blocks as separate pieces. Each block has a unique identifier. This type of storage places those blocks of data wherever they are most efficient. This enables incredible scalability and works well with numerous operating systems. Block storage is hierarchical, meaning files are stored in nested folders like a traditional desktop operating system.

#### **Object Storage**

Object-based storage differs from block storage. Object storage breaks data files into pieces called *objects*. It then stores those objects in a single place that can be used by multiple systems that have network access to the storage. Since each object will have a unique ID, it's easy for computing resources to access data on file-based storage. Additionally, each object has metadata or information about the data to make it easier to find when needed. This metadata allows for powerful search functions such as SQL queries to be run on the data. Object storage is “flat” or non-hierarchical, meaning all files are stored in a single folder or “bucket.” Due to the metadata, object storage is often used in big data environments and data lakes.

## **File Storage**

File storage is traditional storage. It can be used for a systems operating system and network file systems. Examples of this are NTFS-based volumes for Windows systems and NFS volumes for Linux/UNIX systems. These volumes can be mounted and directly accessed by numerous computing resources simultaneously.

## **AWS Object Storage**

The AWS platform provides an efficient platform for object storage with Amazon Simple Storage Service, otherwise known as S3.

### **Amazon Simple Storage Service (S3)**

Amazon S3 provides high-security, high-availability, durable, and scalable object-based storage. S3 has 99.99999999 percent durability and 99.99 percent availability. *Durability* refers to a file getting lost or deleted. Availability is the ability to access the system when you need access to your data. This means data stored on S3 is highly likely to be available when you need it.

Since S3 is object-based storage, it provides a perfect opportunity to store files, backups, and even static website hosting. Computing systems cannot boot from object-based storage block-based storage. Therefore, S3 cannot be used for the computing platform's operating system. Since block-based storage is effectively decoupled from the server's operating system, it has near limitless storage capabilities.<sup>1</sup>

S3 is typically used for these applications:

- Backup and archival for an organization's data
- Static website hosting
- Distribution of content, media, or software
- Disaster recovery planning
- Big data analytics
- Internet application hosting

Amazon S3 is organized into buckets. Each bucket is a container for files stored on S3. Buckets create a top-level name space, meaning they must be globally unique and can be accessed by their DNS name. An example of this address can be seen below:

<http://mybucketname.s3.amazonaws.com/file.html>

The diagram below shows the organizational structure of S3:



Since the buckets use DNS-type addressing, it is best to use names that follow standard DNS naming conventions. Bucket names can have up to sixty-three characters, including letters, numbers, hyphens, and periods. It's noteworthy that the path you use to access the files on S3 is not necessarily the location where the file is stored. The URL used to access your file is really a pointer to the database where your files are stored. S3 functions a lot like a database behind the scenes, which enables you to do incredible things with data stored on S3 like SQL queries. An organization can have up to one hundred buckets per account without requesting a bucket limit increase from AWS.

Buckets are placed in different geographic regions. When creating the bucket, to achieve the highest performance, place the bucket in a region that is closest. Additionally, this will decrease data transfer charges across the AWS network. For global enterprises, it may be necessary to place buckets in multiple geographic regions. AWS S3 provides a means for buckets to be replicated automatically between regions. This is called *cross-region replication*.

### S3 Is Object-Based Storage

S3 is used with many services within the AWS cloud platform. Files stored in S3 are called *objects*. AWS allows most objects to be stored in S3. Every object stored in an S3 bucket has a unique identifier, also known as a *key*. A key can be up to 1,024 bytes, which are comprised of Unicode characters and can include slashes, backslashes, dots, and dashes.

Single files can be as small as 0 bytes, all the way to 5 TB per file. This provides ultimate flexibility. Objects in S3 can have metadata (information about the data), which can make S3 extremely flexible and can assist with searching for data. The metadata is stored in a name-value pair environment, much like a database. Since metadata is placed in a name-value pair, S3 provides a SQL-based method to perform SQL-based queries of your data stored in S3. To promote scalability, AWS S3 uses an eventually consistent system. While this promotes scalability, it is possible that after you delete an object, it may be available for a short period.

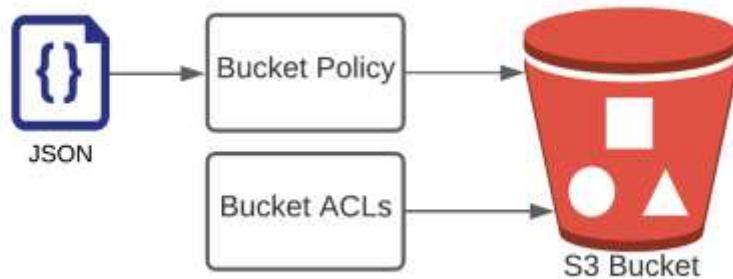
## Securing Data in S3

The security of your data is paramount, and S3 storage is no exception. S3 is secured via the bucket's policy and user policies. Both methods are written in JSON access-based policy language.

Bucket policies are generally the preferred method to secure your data in S3. Bucket policies allow for granular control of the security of your digital assets stored in S3. Bucket policies are based on IAM users and roles. S3 bucket policies are similar to the way Microsoft authenticates users and grants access to certain resources based upon the user's role, permissions, and groups in Active Directory.<sup>2</sup>

S3 also allows for using ACLs (Access Control Lists) to secure your data. However, the control afforded to your data via an ACL is much less sophisticated than with IAM policies. ACL-based permissions are essentially *read*, *write*, or *full control*.

The diagram below shows how ACL and bucket policies are used with AWS S3:



## **S3 Storage Tiers**

AWS S3 offers numerous storage classes to best suit an organization's availability and financial requirements. The main storage classes can be seen below:

- Amazon S3 Standard
- Amazon S3 Infrequent Access (Standard-IA)
- Amazon S3 Infrequent Access (Standard-IA) – One Zone
- Amazon S3 Intelligent-Tiering
- Amazon S3 Glacier

### **Amazon S3 Standard**

Amazon S3 Standard provides high-availability, high-durability, high-performance, and low-latency object storage. Given the performance of S3, it is well suited for storage of frequently accessed data. For most general-purpose storage requirements, S3 is an excellent choice.

### **Amazon S3 Infrequent Access (Standard-IA)**

Amazon S3 Infrequent Access provides the same high-availability, high-durability, high-performance, and low-latency object storage as standard S3. The major difference is the cost, which is substantially lower. However, with S3 Infrequent Access, you pay a retrieval fee every time data is accessed. This makes S3 extremely cost-effective for long-term storage of data not frequently accessed. However, access fees might make it cost-prohibitive for frequently accessed data.

### **Amazon S3 Infrequent Access (Standard-IA) – One Zone**

Amazon S3 Infrequent Access provides the same service as Amazon S3-IA but with reduced durability. This is great for backups of storage due to its low cost.

### **Amazon S3 Intelligent-Tiering**

Amazon S3 Intelligent-Tiering provides an excellent blend of S3 and S3-IA. Amazon keeps frequently accessed data on S3 Standard and effectively moves your infrequently accessed data to S3-IA, so you have the best and most cost-effective access to your data.

Additionally, S3 Intelligent-Tiering facilitates further cost optimization by enabling an organization to automatically archive information that was not accessed in a period. This semi-automated lifecycle policy can help an organization reduce its storage costs. The way this works is data that was not accessed in a period of 90 days will be moved to the archive access tier, which presents performance equivalent to the S3 Glacier storage class. Additionally, items that have not been accessed for a period of 180 days to the Deep Archive Access tier, which is equivalent to the S3 Deep Archive.

## Amazon S3 Glacier

Amazon Glacier offers secure, reliable, and very low-cost storage for data that does not require instant access. This storage class is perfect for data archival or backups. Access to data must be requested; after three to five hours, the data becomes available. Data can be accessed sooner by paying for expedited retrievals. Glacier also has a feature called Vault Lock. Vault Lock can be used for archives that require compliance controls, i.e., medical records. With Vault Lock, data cannot be modified but can be read when needed. Glacier, therefore, provides immutable data access, meaning it can't be changed while in Glacier.

## Managing Data in S3

This next section describes how to manage and protect your data on S3.

### S3 Lifecycle Management

The storage tiers provided by S3 provide an excellent means to have robust access to data with a variety of pricing models. S3 lifecycle management provides an effective means to automatically transition data to the best S3 tier for an organization's storage needs.

For example, let's say you need access to your data every day for thirty days, then infrequently access your data for the next thirty days, and you may never access your data again but want to maintain it for archival purposes. You can set up a lifecycle policy to automatically move your data to the optimal location.

You can store your data in S3, then after thirty days, have your data automatically moved to S3-IA, and after thirty days, the data can be moved to Glacier for archival purposes. This can be seen in the diagram below:



That's the power of lifecycle policies.

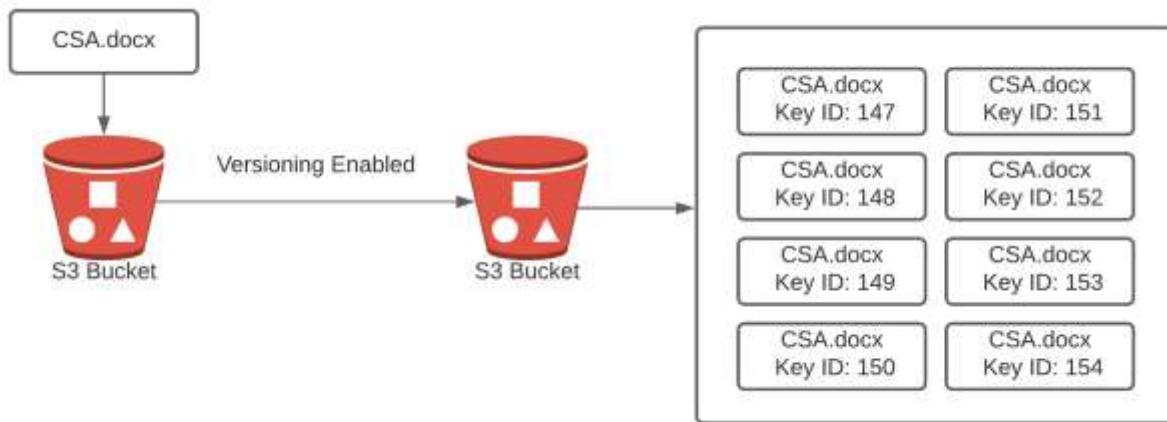
Lifecycle policies can be configured to be attached to the budget or specific objects as specified by an S3 prefix.

## S3 Versioning

To help secure S3 data from accidental deletion, *S3 versioning* is the AWS solution.

Amazon S3 versioning protects data against accidental or malicious deletion by keeping multiple versions of each object in the bucket, identified by a unique version ID. Therefore, multiple copies of objects are stored in S3 when versioning is enabled. Every time there is a change to the object, S3 will store another version of that object. Versioning allows users to preserve, retrieve, and restore every version of every object stored in their Amazon S3 bucket. If a user makes an accidental change or even maliciously deletes an object in your S3 bucket, you can restore the object to its original state simply by referencing the version ID along with the bucket and object key. Versioning is turned on at the bucket level. Once enabled, versioning cannot be removed from a bucket; it can be suspended only.

The diagram below shows how S3 versioning maintains a copy of all previous versions by using a Key ID:



## Multifactor Authentication Delete

To provide additional protection from data deletion, S3 supports multifactor authentication to delete an object from S3. When an attempt to delete an object from S3 is made, S3 will request an authentication code. The authentication code will be a one-time password that changes every few seconds. This one-time authentication code can be provided from a hardware key generator or a software-based authentication solution, i.e., Google Authenticator.

## Organizing Data in S3

As previously discussed, S3 storage is very similar to a database. Essentially the data is stored as a flat arrangement in a bucket. While this scales well, it is not necessarily the most organized structure for end users. S3 allows the user to specify a prefix and delimiter parameter so the

user can organize their data in what feels like a folder. Essentially the user could use a slash (/) or backslash (\) as a delimiter. This will make S3 storage look and feel like a traditional Windows or Linux file system organized by folders.<sup>3</sup> For example:

- mike/2020/awsvideos/storage/S3.mp4
- mike/2020/awsvideos/compute/ec2.mp4
- mike/2020/awsvideos/database/dynamo.mp4

## Encrypting Your Data

S3 supports a variety of encryption methods to enhance the security of your data. Generally, all data containing sensitive or organization proprietary data should be encrypted. Ideally, you encrypt your data on the way to S3, as well as when the data is stored (or resting) on S3. A simple way to encrypt data on the way to S3 is to use https, which uses SSL to encrypt your data on its way to the S3 bucket. Encrypting data on S3 can be performed using client-side encryption or server-side encryption.<sup>4, 5</sup>

### Client-Side Encryption

*Client-side encryption* means encrypting the data files prior to sending to AWS. This means the files are already encrypted when transferred to S3 and will stay encrypted when stored on S3. To encrypt files using client-side encryption, there are two options available to use. Files can be encrypted with a client-side master key or a client master key using the AWS key management system (KMS). When using client-side encryption, you maintain total control of the encryption process, including the encryption keys.

### Server-Side Encryption

Alternatively, S3 supports *server-side encryption*. Server-side encryption is performed using S3 and KMS. Amazon S3 automatically will encrypt when storing and decrypt when accessing your data on S3. There are several methods to perform server-side encryption and key management. These options are discussed below.

#### SSE-KMS (Customer-Managed Keys with AWS KMS)

The SSE-KMS is a complete key management solution. The user manages their own master key, but the key management system manages the data key. This solution provides extra security by having separate permissions for using a customer-managed key, which provides added protection against unauthorized access of your data in Amazon S3. Additionally, SSE-KMS helps with auditing by providing a trail of how, by whom, and when your data was accessed.

## **SSE-S3 (AWS-Managed Keys)**

SSE-S3 is a fully integrated encryption solution for data in your S3 bucket. AWS performs all key management and provides secure storage of your encryption keys. Using SSE-S3, every object is encrypted with a unique encryption key. All object keys are then encrypted by a separate master key. AWS automatically generates new encryption keys and automatically rotates them monthly.

## **SSE-C (Customer-Provided Keys)**

SSE-C is used in environments when you want total autonomy over key management. When using SSE-C, AWS S3 will perform all encryption and decryption of your data, but you have total control of your encryption keys.

## **Tuning S3 for Your Needs**

There are a few tuning options to make S3 perform even better. They are covered below:

### **Presigned URLs**

All objects stored in S3 are private by default, meaning only the owner has access to the objects in their bucket. Therefore, to share a file with another user or organization, you need to provide access to the object. You can generate a presigned URL to share an S3-based object with others. When you generate the presigned URL, the owner's encryption key is used to sign the URL, which allows the receiver temporary access to the object.

You can use the following authentication options to generate a presigned URL. Each method has a different expiration to the authorization provided by the presigned URL. These options can be seen in the table below:

<b>Presigned URL Expiration</b>	
<b>Method</b>	<b>Expiration Time</b>
IAM Instance Profile	Up to 6 Hours
AWS Security Token Service	Up to 36 Hours
IAM User	Up to 7 Days
Temporary Token	When the Token Expires

### **Multipart Uploads**

AWS S3 allows for objects of up to 5 TB in size to be stored. However, when trying to upload large files, many things can go wrong and interrupt the file transmission. The largest file that can be uploaded as a single object is 5 GB. It is best practice to send files larger than 100 MB

as a multipart upload. In a multipart upload, the file is broken into pieces, and each piece is sent to S3. When all the pieces are received, S3 puts the pieces back together into a single file. This helps dramatically if anything goes wrong with the transmission; only a part of the file needs to be resent instead of the whole file. This improves both the speed and reliability of transferring large files to S3.

### **Range Gets**

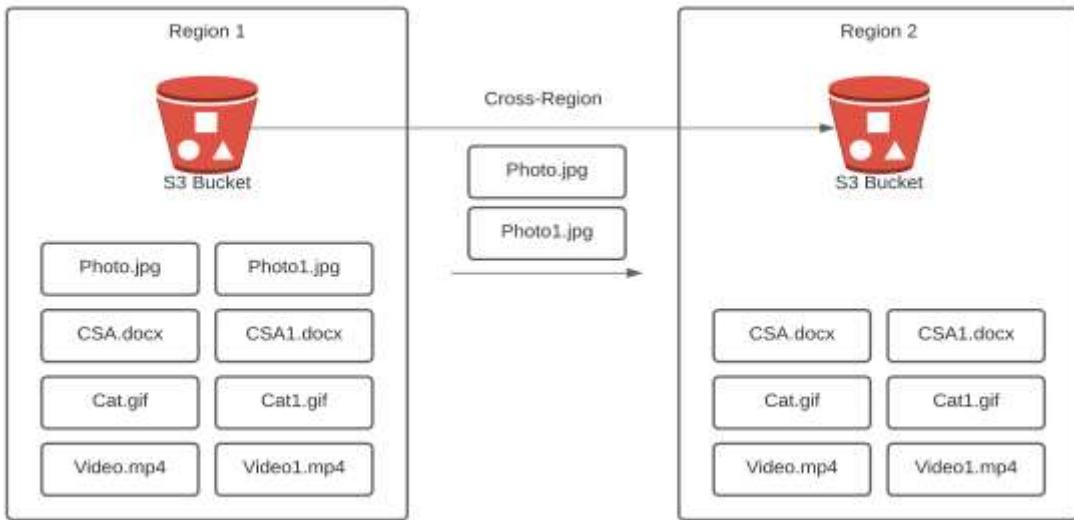
S3 supports large files. Sometimes you need access to some of the data in the file, but not the entire object. A *range get* allows you to request a portion of the file. This is highly useful when dealing with large files or when on a slow connection, i.e., mobile network.

### **Cross-Region Replication**

S3 cross-region replication automatically copies the objects in an S3 bucket to another region. Therefore, all S3 buckets will be synchronized. After cross-region replication is turned on, all new files will be copied to the region for which cross-region replication has been enabled. Objects in the bucket before turning on cross-region replication will need to be manually copied to the new bucket.

Cross-region replication is especially useful when hosting a global website's backend on S3. If website users experience latency due to the distance of the users to the website, the website can be hosted locally off a local S3 bucket. An example would be a US company with a website hosted in New York that suddenly gets a large customer base in Japan. The S3 bucket can be manually copied to a bucket in Asia. With cross-region replication enabled, any future changes to the website's main bucket in New York will be automatically copied to the S3 bucket in Asia. This can dramatically reduce latency and provide a disaster recovery backup of the main website.

The diagram below shows cross-region replication copying files from one region to another region:



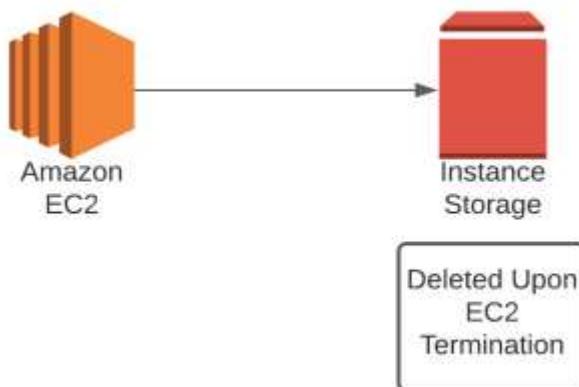
## Storage for Computing Resources

The next class of storage is storage directly available to computing platforms. These include instance storage, elastic block storage, and elastic file systems.

### Instance Storage

*Instance storage* is essentially a transient storage platform for an elastic computing instance. It is temporary in that when the instance is stopped, the volume is automatically deleted. This will be covered in much more depth when we discuss the EC2 platform.

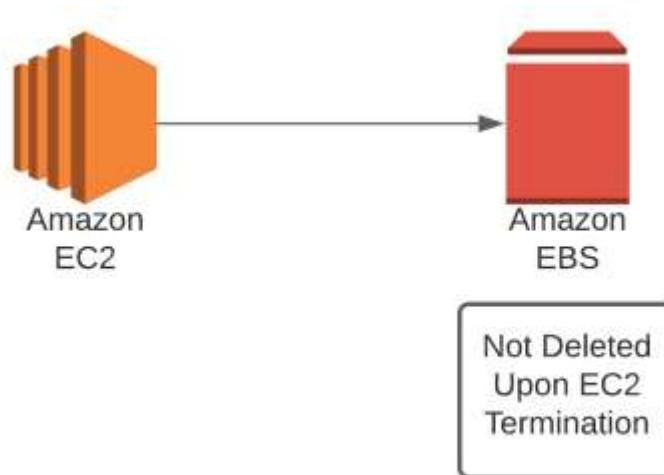
The diagram below shows how EC2 instance storage is used within the AWS platform. Note the instance storage is deleted upon EC2 termination.



## Elastic Block Storage (EBS)

Elastic block storage is a high-performance block storage platform for using EC2 instances and AWS databases. An EBS volume can be mounted and used by EC2 instances, and the data on an EBS volume is not deleted on an EC2 instance termination. EBS volumes are designed for high throughput and high-transaction workloads. This storage is ideal for databases, enterprise applications, containers, and big data applications. EBS is scalable and can scale to multiple petabytes of data. EBS is highly available, with availability of 99.999 percent, so it's perfect for mission-critical use. EBS functions like a virtual hard drive. EBS is automatically backed up to another availability zone to protect against any device failure.<sup>6</sup>

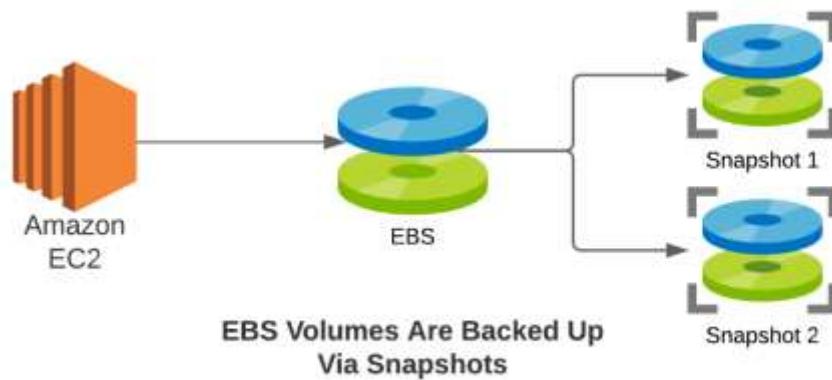
The diagram below shows how EBS instances are attached to an EC2 instance:



## **Backing Up EBS Volumes**

EBS volumes can be backed up to other regions in the form of EBS snapshots. An *EBS snapshot* is a point-in-time copy of your data. A snapshot can be shared with other users and can be copied to other regions. Snapshots can be versioned, and multiple versions can be maintained. Therefore, you can restore your data to any previous snapshots. EBS snapshots are incrementally backed up to optimize speed and storage space. Unlike with traditional incremental backups, you can delete some backups and still make a complete restore of current data stored on EBS volumes.<sup>7</sup>

The diagram below shows how EBS volumes are backed up using snapshots:



## EBS Volume Types

There are several types of EBS volumes available. Choosing the right volume type can make a substantial difference in the performance of your system. This next section describes the EBS volumes available and how to choose the right EBS volume type to meet the systems requirements. The four types of EBS volumes are EBS Provisioned IOPS (io1), EBS General Purpose SSD (gp2), EBS Throughput Optimized HDD (st1), and EBS Cold HDD (sc1).

### EBS Provisioned IOPS (io1)

EBS-provisioned IOPS is the highest-performance SSD storage available on the EBS platform. This platform enables the user to purchase guaranteed speed of read and write performance, in terms of inputs and outputs per second (IOPS). EBS-provisioned IOPS is optimal for applications that require high disk input and output performance (IO). It's designed for databases or other applications requiring low latency. There are three types of provisioned IOPS volumes. Each PIOPS version has different speeds.

EBS Provisioned IOPS has three options which are io1, io2, and io2 Block Express. Io2 Block Express is the highest performance SSD volume designed for business-critical, latency-sensitive, and transactional workloads. The throughput speeds provided by io2 Block Express volumes are up to 4,000 MB/second.

Volume Type	io2 Block Express	io2	io1
Volume Size	4 GB – 64 TB	4 GB – 16 TB	4 GB – 16 TB
Max IOPS/Volume	256,000	64,000	64,000
Max Throughput*/Volume	4,000 MB/s	1,000 MB/s	1,000 MB/s

Io2 Block Express is the highest performance SSD volume designed for business-critical, latency-sensitive, and transactional workloads. The throughput speeds provided by io2 Block Express volumes are up to 4,000 MB/second. Io1 and io2 are both pre-provisioned volumes but operate at a much lower speed than io2 Block Express, as you can see in the table above.

### **EBS General Purpose SSD (gp2)**

EBS general purpose SSD (gp2) is an SSD-based storage solution. It provides a good balance of price and performance. Gp2 is an excellent platform for a boot volume, as it has good performance and does not get erased upon EC2 instance shutdown. Gp2 is great for transactional workloads. It is great for dev and test environments that require low latency, at a much lower cost than PIOPS-based volumes used in production environments.

### **Throughput Optimized HDD (st1)**

EBS Throughput Optimized HDD (st1) is low-cost magnetic storage. St1 is designed for frequently accessed workloads. It supports a fairly significant throughput of 500 MB/second but with higher latency than SSD-based volumes. St1 is designed for situations that require a low-cost option to store substantial data. It works well with applications that have sequential read and writes to the disk.

### **EBS Cold HDD (sc1)**

EBS Cold HDD (sc1) is the lowest-cost option for EBS volumes. Sc1 is for workloads that do not require frequent disk access but still need a reliable storage platform.

The diagram below lists the speed and performance of the various EBS volumes.

Volume Type	SSD			HDD			
	General Purpose SSD		Provisioned IOPS SSD		Throughput Optimized HDD	Cold HDD	
	gp3	gp2	io2 Block Express	io2	io1	st1	sc1
Volume Size	1GB–16TB	4GB–16TB	4GB–64TB	4GB–16 TB	4GB–16 TB	125GB–16TB	125GB–16TB
Max IOPS/Volume	16,000	16,000	256,000	64,000	64,000	500	250
Max Throughput/Volume	1,000 MB/s	250 MB/s	4,000 MB/s	1,000 MB/s	1,000 MB/s	500 MB/s	250 MB/s
Performance Attribute	IOPS					MB/second	

\*Volume throughput is calculated as MB =  $1024^2$  bytes.

### Choosing the Correct EBS Volume Type

Choosing the correct EBS volume type can make the difference between optimal performance and performance problems within the system. The primary determination of volume selection is whether the application requires high throughput, low latency, or both.

For applications requiring high throughput and low latency, choose EBS-provisioned IOPS. Applications that require moderate throughput, but low latency will perform well on EBS General Purpose SSD volumes. For applications that require high throughput but are less sensitive to latency, EBS Throughput Optimized HDD are a great option, especially if the application performs sequential read and writes of data. Lastly, EBS Cold HDD is perfect for storing a large amount of infrequently accessed data that is not sensitive to higher latencies than SSD volumes.

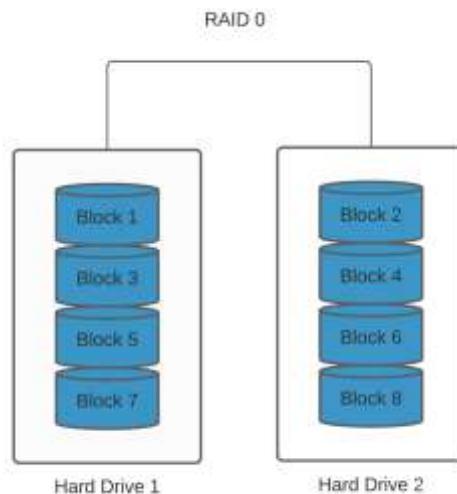
### RAID

When even higher throughput and lower latencies are required, RAID volumes can be configured. RAID volumes are an array of disks that can increase disk performance, capacity, and/or reliability. Since EBS volumes are virtual hard drives automatically backed up to another availability zone, EBS volumes will have even more flexibility than traditional RAID volumes. The next section will describe the primary RAID options.<sup>8</sup>

## RAID 0

RAID 0 is the fastest option for storage. RAID 0 places data on all hard drives in the array. Therefore, the speed and capacity are increased incrementally by every hard drive placed in the system. For example, a RAID array in RAID 0 with four hard drives each with 2 TB capacity will have a total capacity of 8 TB. If each disk has a 200 MB/second throughput, then four disks will provide effectively 800 MB/second of data throughput. The problem with RAID 0 is that there is no fault tolerance. If a single drive fails, then all data on the RAID array is lost. When RAID 0 volumes are used, an excellent backup strategy should be in place. RAID 0 is less of a problem with EBS volumes than traditional disks. Since AWS volumes are virtual disks that are automatically backed up to another availability zone, it is less risky than with traditional hard drives. Still, RAID 0 is often considered too risky to be used in a production environment due to lack of fault tolerance.

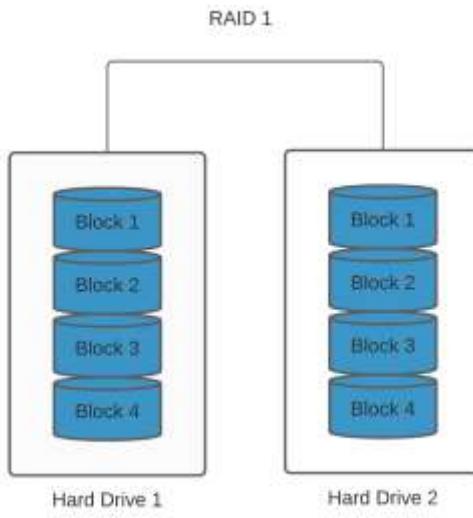
The diagram below shows an example of RAID 0:



## RAID 1

RAID 1 is a highly fault-tolerant form of RAID. RAID 1 is also known as *disk mirroring*. In RAID 1, a second disk or set of disks “mirrors,” or copies, the data from the primary drive to the secondary drive. This option does not increase speed or capacity. However, if a disk fails, all that is necessary is to “break” the mirror and failover to the other disk or disks in the array. This is an excellent option when rapid recovery of a hard drive failure is required. Generally, it’s one of the most expensive and lowest-performing RAID options, as it requires double the storage, one for the data and another drive or set of drives to back up the data in real time.

The diagram below shows an example of RAID 1:



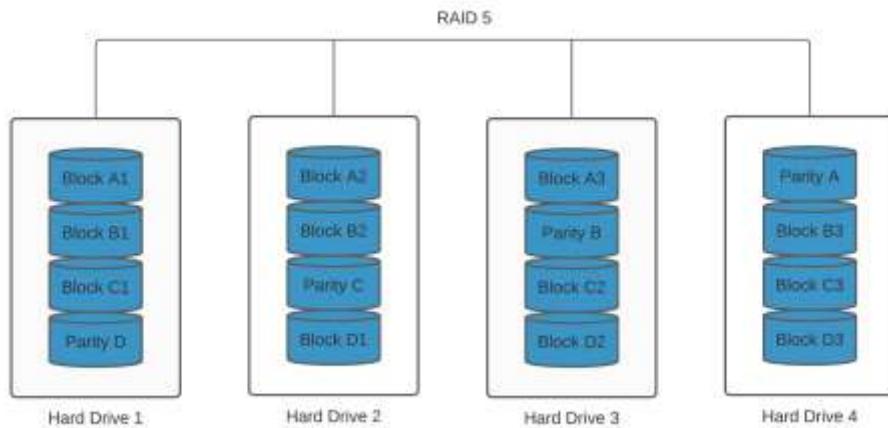
## RAID 5

RAID 5 is a very common form of RAID storage. It is not recommended by AWS but is widely used in enterprise computing, so we will describe it for informational purposes.

RAID 5 provides a good balance of speed and redundancy. A RAID 5 array requires at least three disks. Like RAID 0, the data is striped across all disks. Unlike RAID 0, parity or recovery information is also striped across the drives. Since parity information is striped across the drives, a percentage of the storage will be dedicated to writing parity data. This means with a three-disk RAID 5 array, there will only be the capacity of two disks. Additionally, there is overhead associated with writing parity data to the disks, which decreases potential disk performance. The good news is that if an individual disk fails, the data in the array can be rebuilt by installing a new disk, and the array copies the data to the new disk from the stored parity information.

RAID 5 provides greater speed than a single drive but is slower than RAID 0. Some version of RAID 5 is used in most enterprise computing environments.

The diagram below shows an example of RAID 5:

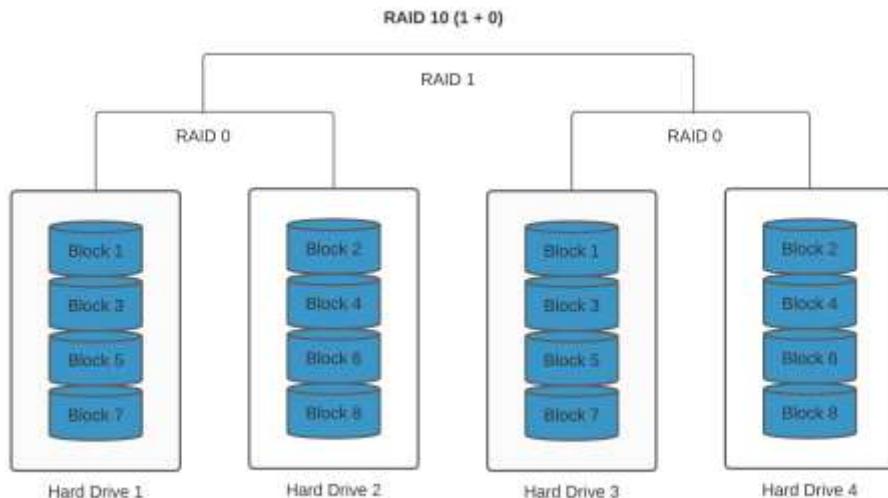


## RAID 10

RAID 10 is the highest-performing and highest-redundancy option available. RAID 10 is essentially a combination of RAID 0 and RAID 1. RAID 10 works by configuring two RAID 0 arrays and then configuring the second RAID array to mirror the primary RAID array. This enables the speed of RAID 0 with the redundancy of RAID 1.

The drawback of a RAID 10 array is it requires double the disks with no increase in storage capacity. So if ten disks are configured in RAID 0, another ten disks will be required to mirror the primary array. This makes it the highest-cost option for RAID arrays. This option is becoming increasingly popular, as higher disk speeds are increasingly needed, but with fault tolerance.

The diagram below shows an example of RAID 10:



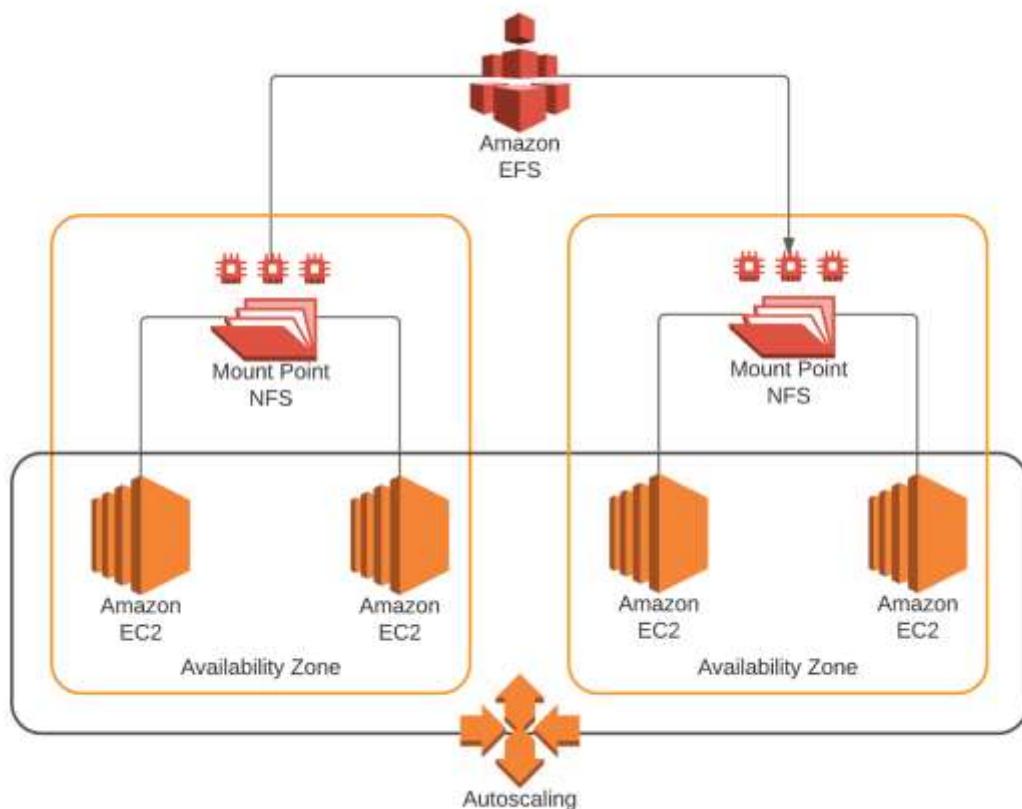
## Elastic File System (EFS)

The next storage for AWS is the Elastic File System (EFS). EFS is a high-performance, highly scalable file system for networked computers. It is essentially the AWS version of the network file system (NFS) originally invented by Sun Microsystems. Since EFS is a network file system, it can be accessed simultaneously by many computing instances. EFS is best used when a high-performance network file system is required. Think of corporate file shares or multiple servers needing access to the same files simultaneously.

There are two versions of EFS available. The two versions of EFS are standard and infrequent access. *Standard EFS* is the normal version and the highest performance option. *EFS Infrequent access* is a lower-cost option for files not accessed frequently. EFS has numerous benefits:

- Scalable – High throughput, high IOPS, high capacity, and low latency.
- Elastic – EFS will automatically adjust sizing to meet the required storage capacity.
- Pricing – Pay for what is used.
- POSIX compatible – This enables access from standard file servers both on premises and on the cloud. Works with traditional NFS-based file permissions and directories.

The diagram below shows how EFS is mounted and used by multiple EC2 instances on the AWS platform:



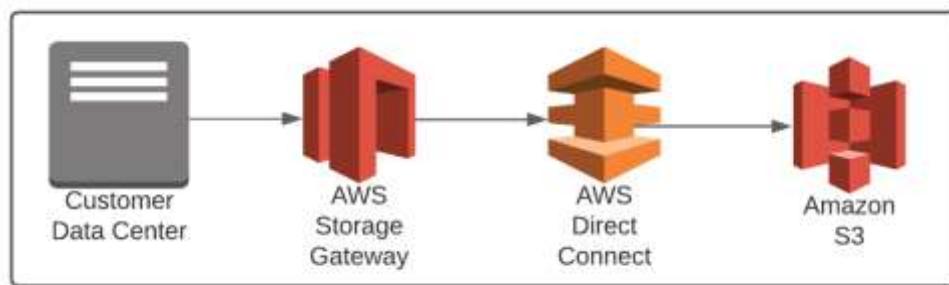
## AWS Storage Gateway

Connecting on-premises environments to the cloud and achieving optimal performance often requires some tuning. This is especially true when connecting servers to object-based storage on S3. A great option to connect the on-premises data center to S3-based storage is with a storage gateway. This makes the storage look and feel like a network file system to the on-premises computing systems.<sup>9</sup>

A storage gateway is a virtual machine that runs in the data center. It is a prebuilt virtual machine from AWS available in VMware, Hyper V, or KVM. This virtual machine acts as a virtual file server. The on-premises computing resources read and write to the storage gateway, which then synchronizes to S3. You can access the storage gateway with SMB or NFS based shares. SMB is optimal for Windows devices, while NFS is optimal for Linux machines.

Three types of storage gateways are available for different applications. They are *volume gateways*, both *cached volumes* and *stored volumes*; and *tape gateways*.

The diagram below shows an example of a storage gateway on the AWS platform:



### **Storage Gateway Cached Volume**

In a storage gateway cached volume, the organization's data is stored on S3. The cache volume maintains frequently accessed data locally on the volume gateway. This provides low-latency file access for frequently accessed files to the on-premises systems.

### **Storage Gateway Stored Volume**

In a storage gateway stored volume, the on-premises systems store your files to the storage gateway, just like any other file server. The storage gateway then asynchronously backs up your data to S3. The data is backed up to S3 via point-in-time snapshots. Since the data is on S3 via a snapshot, these snapshots can be used by EC2 instances should something happen to the on-premises data center. This provides an excellent and inexpensive offsite disaster recovery option.

## Tape Gateway

The tape gateway provides a cloud backup solution. It essentially replaces backup tapes used by some enterprise environments for deep archival purposes. Since it is virtual, there is no need to maintain physical tapes and the infrastructure to support a tape-based backup solution. With the tape gateway, data is copied to Glacier or Deep Archive.

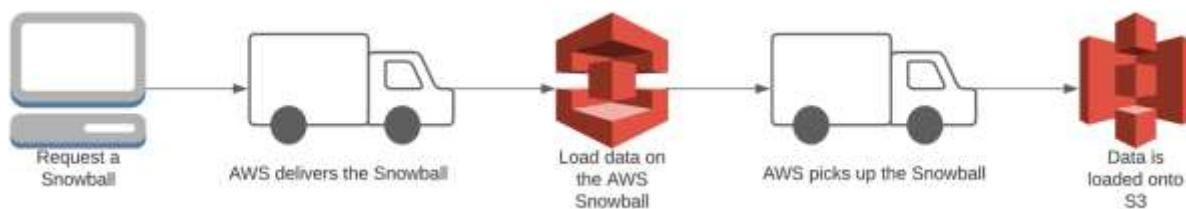
## Migrating Data to AWS

Moving an organization's data center to the cloud involves setting up the cloud infrastructure as well as moving an organization's data to AWS. Data can be copied to the cloud over a VPN or a direct connection. But if there is a tremendous amount of data or rapid timelines to move to the cloud, it might be necessary to move files to the cloud in a more efficient manner. Additionally, it might be cheaper to send data directly to S3, as opposed to adding additional high-performance, high-capacity direct connections to AWS. To that end, there are multiple methods of sending data directly to AWS and they are listed below.

### AWS Snowball

The AWS Snowball is a rugged computer with substantial storage that can be rented from AWS. AWS ships the Snowball to the customer. The customer places the Snowball on their network and copies the files they want to move to AWS. The files on the Snowball are encrypted. The organization then ships the Snowball to AWS. Once the Snowball is received by AWS, AWS employees move the data from the Snowball to the organization's cloud computing environment.

The diagram below shows an example of an AWS Snowball:



### AWS Snowball Edge

The Snowball Edge is a more powerful snowball that includes compute capabilities and a 100 terabyte (100TB) storage capacity. Snowball Edge can run Lambda functions, provide durable local storage on premises, and transfer files through NFS. Snowball Edge essentially allows for storage and compute power on the go.

Snowball Edge offers two types of configurations—*Storage Optimized* or *Compute Optimized*. Each configuration includes additional options such as increased compute functionality or an attached GPU.

The diagram/picture below shows an example of an AWS Snowball Edge device:



*Photo from AWS - AWS Snowcone - <https://aws.amazon.com/blogs/aws/aws-snowball-edge-more-storage-local-endpoints-lambda-functions/>*

The AWS Snowcone is a small data transfer device with a rugged exterior weighing roughly 4.5 lbs. (2.1 kg). Snowcone provides 8 TB of available storage that supports data transfer from on-premises Windows, Linux, and macOS systems, including file-based applications via Network File System (NFS).

The diagram/picture below shows an example of an AWS Snowcone:



*Photo from AWS - <https://aws.amazon.com/snow/>*

## AWS Import/Export Service

The import-export service is essentially a means to ship data to AWS. Essentially, you copy your data to a hard drive or hard drives. The organization then ships the hard drives to AWS. Once the hard drives are received by AWS, they move the data from the hard drives to your cloud computing environment.<sup>10</sup>

The diagram below shows how the AWS Import/Export service is used to quickly move large amounts of data to AWS:



## AWS Snowmobile

The AWS Snowmobile is a forty-five-foot-long semi-trailer truck that can securely transport vast amounts of data to AWS efficiently and inexpensively. This exabyte-scale data transfer service allows customers to deliver petabytes of data in weeks instead of months. Each truck has 100 petabytes (100 PB) of storage capacity and is protected by 24/7 video surveillance.

Once the Snowmobile is dispatched to your site, AWS personnel configure your network for file transfer to perform high-speed data migration using a removable network switch. After successful data transfer, the Snowmobile is taken back to AWS for importation to the cloud.

The diagram/picture below shows an example of an AWS Snowmobile:



## **Storage for Collaboration**

In recent years there has been a significant move to cloud storage for collaboration. This is especially useful for working on creative projects, documents, or presentations. AWS has created Amazon WorkDocs for this purpose.

### **Amazon WorkDocs**

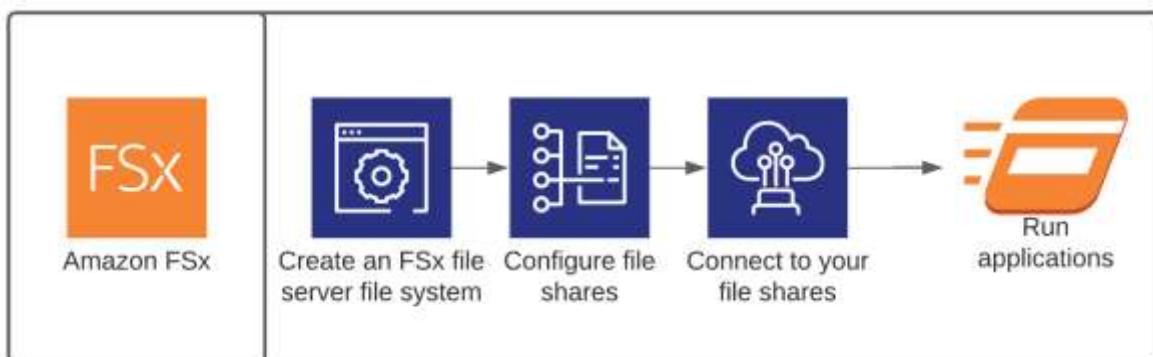
Amazon WorkDocs is a fully managed, secure content creation, storage, and collaboration service. It is similar in functionality to Google Drive or Dropbox. WorkDocs enables collaboration across projects and facilitates things like shared document editing. It is a simple solution with pay-as-you-go pricing. WorkDocs can be accessed with a web interface or with client software. Client software is available for Windows and macOS clients. This is secure storage and meets all main forms of compliance standards, i.e., HIPAA, PCI, and ISO.<sup>11</sup>

## **Amazon FSx for Windows**

Some organizations heavily utilize the Windows platform for many critical services. Organizations that need native windows file servers can use Amazon FSx. Amazon FSx's are hosted Microsoft Windows file servers. FSx is a Windows server; therefore it supports Microsoft features such as storage quotas and active directory integration. As a Windows file server, it uses the Server Message Block (SMB) protocol. Windows-based SMB file shares can be accessed by Windows, macOS, and Linux hosts.

FSx is a high-availability service with high-availability single and multiple availability zone options. FSx provides data protection through encryption, both in transit and at rest. Additionally, to protect against data loss, FSx provides fully managed backups.

The diagram below shows an example of AWS FSx being used for Windows hosts:



## **FSx for Lustre**

FSx for Lustre is high-performance storage based upon the Luster parallel file system. FSx for Lustre has sub millisecond latency. FSx for Lustre has excellent throughput—hundreds of gigabytes per second. Additionally, this is very-low-latency storage with millions of IOPS. FSx for Lustre can be used with S3 to allow the S3 objects to be accessed as if they were files in a network-attached storage (NAS) or elastic file system (EFS).

## **Chapter 4 - Computing on the AWS Platform (EC2)**

This chapter describes the computing options available within the AWS Platform. AWS provides numerous computing options that can be sized based upon an organization's needs. Sizing compute resources are similar to sizing virtual machines in a traditional data center. Servers should be configured based upon the CPU, memory, storage (size and performance), graphics processing unit (GPU), and network performance requirements.<sup>12</sup>

AWS's primary computing platform is called Elastic Compute Cloud (EC2). EC2 servers are virtual machines launched on the AWS platform. EC2 servers are called *instances*. EC2 instances are sized like any virtualized server environment. AWS has numerous options to meet an organization's needs. Instance types should be chosen based upon the need for the following computing options:

- CPU cores (virtual CPUs)
- Memory (DRAM)
- Storage (capacity and performance)
- Network performance

Instances are available in a wide range of sizes and configurations to provide a means to perfectly size a system based upon an organization's needs. AWS has a multitude of possible server configurations to assist the organization with properly sizing their computing instances.<sup>13</sup>

A summary of the EC2 instance types is below:

EC2 Instance Types	Specialty	Use Case
A1	Arm-Based Workloads	Web Servers
C5	Compute Optimized	Batch Processing, Media Transcoding
G3	GPU Based Workloads	Machine Learning
I3	High Speed Storage	Data Warehousing, High Performance Databases
M5	General Purpose	Databases
M6	General Purpose	Application Servers, Gaming Servers
R5	Memory Optimized	Caches, High Performance Databases
T3	Burstable Computing Platform	Web apps, Test Environments
X1	Lowest Pricing Per GB DRAM	Bid Data Processing Engines, In-Memory Databases

Every instance type can be ordered in various sizes. Check with AWS for sizing, as sizes are subject to change.

EC2 instances support Windows and Linux operating systems. An organization can build its own virtual machines or can start with a prebuilt virtual machine. A prebuilt virtual machine is called an Amazon Machine Image (AMI). It is important to note that AMIs will need a storage volume for the instance. The storage volume may be an instance volume or an EBS volume. It is essential to choose the correct storage volume type. Standard instance volumes will be deleted

upon instance reboot or termination. EBS volumes are persistent, and data will be available after reboot or instance termination.

## **Amazon Machine Images (AMI)**

Since the AMI is the basis for the virtual machine, it's important to choose the correct AMI. AMIs can be obtained from the following sources:

- **Published by AWS** – These images are prebuilt by Amazon for a variety of needs. They are available in a variety of operating systems.
- **AWS Marketplace** – These are prebuilt machines by AWS partners. These machines are generally created for specific uses (i.e., licensed software from a third party).
- **AMIs from existing instances** – These are generally customer-created AMIs. They are created from an existing server. They enable a full machine copy, with all applications and package dependencies installed.
- **AMIs from uploaded servers** – An AMI can be created from a physical server to virtual machine conversion. Additionally, an AMI can be imported from a virtual machine conversion (i.e., VMware or VirtualBox virtual machine).

A complete AMI has several components:

- Operating system from one of the four AMI sources.
- Launch permissions for the instance.
- A block device mapping of storage volumes in the system.

## **Automating the First Boot for EC2 Instances**

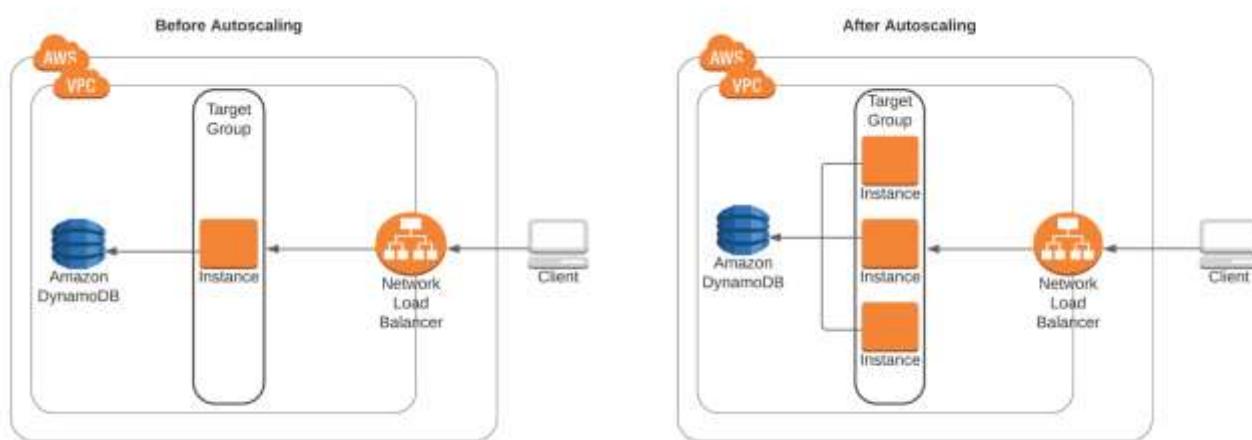
Starting with a full AMI is a great way to launch an EC2 instance. It is fast and efficient. However, the base AMI may not have all the necessary services installed and may need the latest software patches. There are two possible ways to take an AMI and configure it for an organization's use. The first option is to launch the EC2 instance with the premade AMI. After the instance boots, manually update the operating system and then install necessary services. Alternatively, the instance can be automatically updated upon boot with a bootstrap script. Bootstrap scripts for Linux can be as simple as a basic shell script. Windows systems can be bootstrapped with a power shell script.

## Autoscaling

One of the key drivers to the move to the cloud is *autoscaling*. In the traditional environment, servers are configured for peak demand plus a growth factor to meet increased demand over time. With traditional servers, capacity must always exceed demand, because if demand were to increase beyond the server's capacity, either the site would become unavailable, or business could be lost. Oversizing can become very expensive, causing the organization to pay for resources they may never use.

Autoscaling completely changes the computing paradigm. Autoscaling enables an organization to size its computing resources based upon average demand. If demand for computing resources increases, autoscaling can spin up another instance as needed. Therefore, an organization will have almost unlimited computing capacity while paying for only what is used. Many solutions architects consider autoscaling one of the best features of the cloud.

The diagram below shows how autoscaling increases compute instances to handle increased load:



## Instance Purchasing Options

AWS has numerous options for purchasing computing instances. Each type of instance is optimized for different computing requirements. The type of computing instances available can be seen below:

- On-demand instances
- Reserved instances
- Scheduled reserved instances
- Spot instances
- Dedicated hosts

## **On-Demand Instances**

*On-demand computing instances* are computing instances that are available when needed. On-demand instances are charged by either the second or hour of use and facilitate autoscaling. On-demand instances are optimal when reliable computing capacity is needed without complete knowledge of how long the application will be used or its capacity requirements.

## **Reserved Instances**

A *reserved instance* is an instance where an organization commits to purchase a specified compute capacity for a specified period of time. It is essentially a contract to purchase a computing instance and can be for one to three years. By purchasing instances based upon long-term use, the organization can receive substantial savings over on-demand pricing. Reserved instances are optimal when an organization knows how long the application will be used and its capacity requirements.

## **Scheduled Reserved Instances**

A *scheduled reserved instance* is a special type of reserved instance. This type of instance is optimal when you have a need for a specific amount of computing power on a scheduled basis. For example, if an organization has a mission-critical batch job that runs every Saturday and Sunday.

## **Spot Instances**

A *spot instance* is an instance that is pulled from unused AWS capacity. Spot instances are sold in an auction-like manner in which an organization places a bid. If the bid price is equal to or greater than the spot price, a spot instance is purchased. Spot instances are deeply discounted and can be a great option. The drawback of spot instances is that they can be terminated if the spot price goes above the price that was bid on the instances. Spot instances are ideal when an organization needs extra computing capacity at a great price for non-mission-critical use.

## **Dedicated Hosts**

A *dedicated host* is a dedicated server. Dedicated hosts are bare metal servers. A bare metal server is a physical server without an operating system or applications installed. With a dedicated host, the organization can install any operating system or application required. Dedicated hosts are optimal when an organization needs access to system-level information, such as actual CPU usage. Dedicated hosts are an excellent option when an application has a license that is dedicated to a physical machine.

## **Tenancy Options**

After the computing platform is chosen, it is necessary to determine the tenancy of the computing platform. The *tenancy*, or where the instances are located, can make a substantial impact on performance and availability. The tenancy options are:

- Shared tenancy
- Dedicated instances
- Dedicated hosts
- Placement groups

### **Shared Tenancy**

This is the standard tenancy for EC2 instances. With shared tenancy, the physical server hosted at AWS will contain virtual machines for several customers.

### **Dedicated Instance**

This is a server that is completely dedicated to a single customer. This server can house multiple virtual machines. All virtual machines on the dedicated instance belong to the single customer who owns the instance.

### **Dedicated Host**

As previously described, this is a bare metal server and is dedicated to a single customer.

### **Placement Groups**

AWS uses the concept of placement groups. Placement groups are where the computing instances are located. There are three options for placement groups. The options below are covered in the VPC section:

- Clustered
- Spread
- Partitioned

## **Securing EC2 Access**

Keeping an organization's technology secure requires an end-to-end security posture. EC2 instances are no exception, and they should be kept as secure as possible. Keeping the instance patched with the latest security updates and turning off unnecessary services is a major part of securing EC2 instances. AWS provides a security feature called *security groups*. Security groups can dramatically help in locking down EC2 services. Note that security groups are an essential

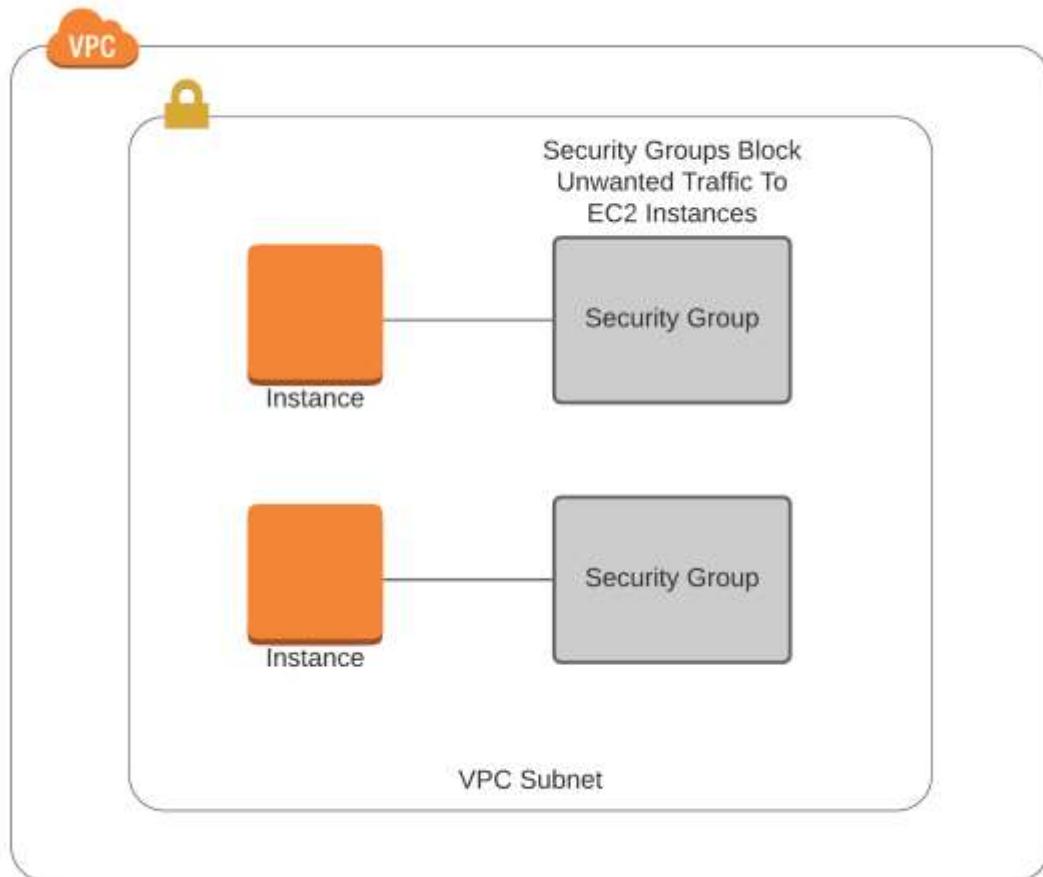
part of many AWS services. Security groups will be covered in depth in the security section of this book.

## Security Groups

A security group is a virtual firewall that lets an organization configure what network traffic is allowed into the server or AWS service. There are several key concepts with security groups:

- The default security policy is to deny traffic. An organization must explicitly permit desired traffic for the security group, or all traffic will be denied.
- Security groups include the source and destination IP address, protocol TCP/UDP, and port numbers.
- The security group is only as effective as its configuration. Be as specific as possible and allow only the traffic necessary into the system. Block all other traffic.

The diagram below shows how security groups protect compute instances by keeping unwanted traffic out of the server:



## **IP Addresses for EC2 Instances**

For an EC2 instance to function on a network, the instance must have an IP address. EC2 instances can have a private IP address, public IP address, or both. Instances also come with a fully qualified domain name assigned by AWS that can be used to connect to an EC2 instance. There are several key components to addressing EC2 instances:

- IP addresses are assigned to network interfaces and not computing systems.
- Depending upon the instance type, an instance can have multiple network interfaces.
- Each network interface must be on a different subnet, then other interfaces on the EC2 instance.
- Each IP address assigned to an interface must be unique within the VPC.
- IP addresses can be IPv4 or IPv6.
- All interfaces are automatically assigned an IPv6 globally unique address, which can be manually disabled.
- EC2 instances with public IP addresses with an internet gateway are reachable from the internet.
- EC2 instances with private IP addresses are not reachable from the internet unless a NAT instance and an internet gateway is used.
- EC2 instances with private IP addresses and a NAT gateway without an internet gateway will not be reachable from the internet but will be able to connect to the internet for operating system patches and other needed connectivity.

Note the section on VPCs discusses networking and IP addressing in much more depth.

## **Accessing EC2 Instances**

Accessing and managing EC2 instances are critically important to maintaining a high-availability architecture. EC2 instances can be accessed via the following methods:

- Directly from the EC2 console.
- Via Secure Shell (SSH) for Linux machines.
- Via Remote Desktop Protocol (RDP) for Windows systems.
- Some management can be configured over the API using the AWS SDK.

## Chapter 5 - Introduction to Databases

### What Is a Database?

A *database* is a repository of structured/unstructured information, or data, stored in a computer system. A database is usually controlled by a database management system (DBMS). The data and the DBMS, along with the applications that integrate with them, are referred to as a database.

Data in databases is commonly presented in rows and columns. This data can be managed by using SQL, which is a special language written to manipulate databases.

In the modern application environment, most web applications and many enterprise applications integrate with databases for information storage, management, and access to information. AWS has four forms of databases.

Databases have become such a critical component of modern information systems. But what exactly is a database and why are databases used? Databases are applications that allow for storage of large amounts of information. Databases can help an organization find key performance metrics from their data in order to make strategic business decisions.

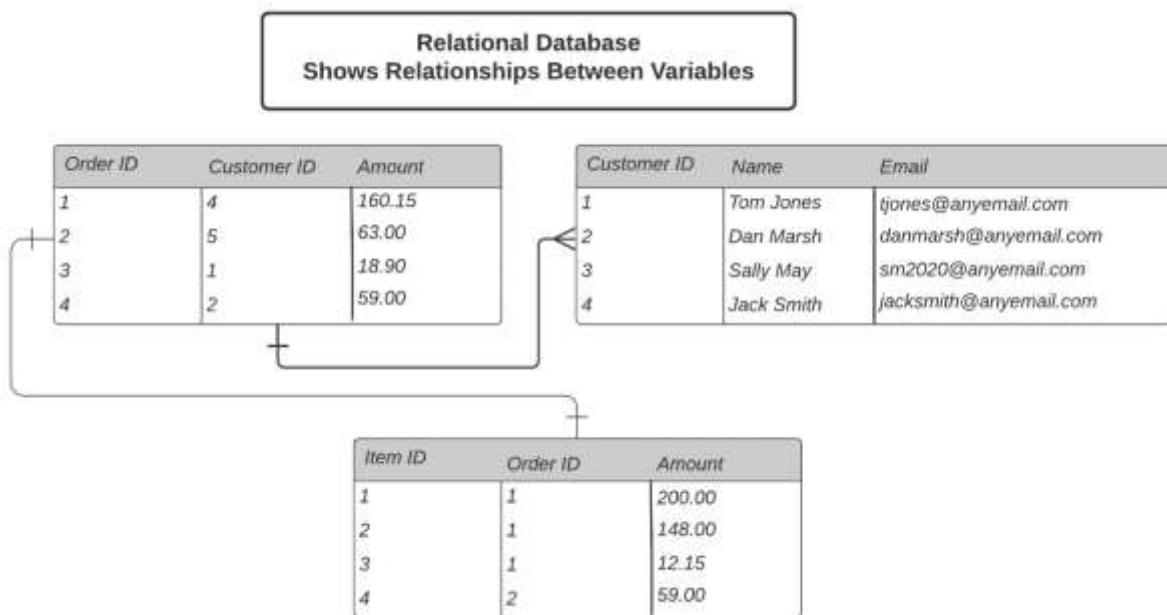
In the modern application environment, most web applications and many enterprise applications integrate with databases for information storage, management, and access to information. AWS has four forms of databases:

- Relational databases
  - Amazon Aurora
  - MariaDB
  - Microsoft SQL Server
  - MySQL
  - Oracle Database
  - PostgreSQL
- NoSQL databases
  - Amazon DynamoDB
  - Amazon DocumentDB
  - Cassandra
- Data warehousing databases
  - Amazon Redshift
- Data lakes – Data lakes allow an organization to store virtually any type of data at an almost unlimited scale. Unlike a database, a data lake can store data in its native format until it is needed by another application.

## Relational Databases

*Relational databases* are the most common type of database and are best for structured data. Relational databases help organizations find the relationships between different aspects of their business by showing how data is related to each other. Relational databases store data in a manner that is very similar to a spreadsheet, with columns and rows.<sup>13</sup>

The diagram below shows how relational databases help show the relationship between different variables:



With relational databases, as soon as data is written, it will be immediately available for query. The instant consistency is based upon relational databases following the ACID model.<sup>17</sup> More information on the ACID model can be seen below:

- Atomic – Transactions are all or nothing.
- Consistent – Data is consistent immediately after writing to the database.
- Isolated – Transactions do not affect each other.
- Durable – Data in the database will not be lost.

In the AWS environment, multiple relational databases are supported, including:

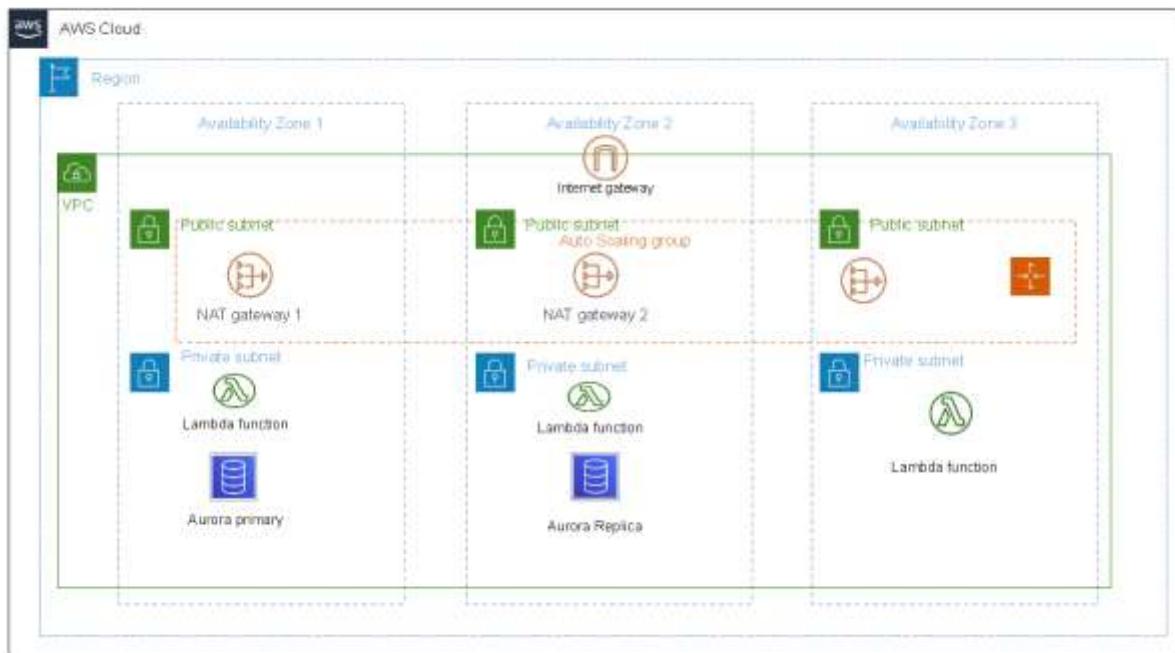
- Amazon Aurora
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle Database
- PostgreSQL

## Amazon Aurora

Amazon Aurora is an Amazon-branded relational database. It is a cloud-native relational database that combines much of the performance and feature set of traditional enterprise databases with the cost-effectiveness of open source databases.

Amazon Aurora is the Amazon-branded relational database service. It is a fully managed database service. Aurora is MySQL- and PostgreSQL-compatible database. Aurora is a high-performance database with speeds up to five times faster than MySQL and three times faster than PostgreSQL databases.

Amazon Aurora is a distributed, fault-tolerant database that autoscales up to ~100TB per database instance. Aurora is highly scalable and can leverage up to 15 read replicas. Additionally, Aurora provides point-in-time recovery, continuous backup to Amazon S3, and replication across three availability zones.



## MariaDB

MariaDB is an open-source relational database. It was created by the developers of the MySQL database. MariaDB has additional features and advanced functionality when compared to MySQL. MariaDB supports a larger connection pool and is comparatively faster than MySQL but doesn't support data masking and dynamic columns.

## **Microsoft SQL Server**

Microsoft SQL Server is the Microsoft-branded relational database solution. AWS supports multiple versions of Microsoft SQL:

- SQL Server 2008
- SQL Server 2012
- SQL Server 2014

Microsoft SQL Server enables organizations to bring their Windows-based workflows to the cloud. Microsoft SQL Server offers tools including the SQL Server Management Studio to help manage the infrastructure. Microsoft SQL Server supports high-availability clustering and failover options, but in a different manner than other relational databases. Please check Microsoft documentation for the most up-to-date information when configuring Microsoft SQL databases.

AWS supports four versions of the Microsoft SQL databases:

- Enterprise
- Express
- Standard
- Web

## **MySQL**

MySQL is one of the original open-source relational databases and has been around since 1995. MySQL is extremely popular and is used in a wide variety of web applications.

## **Oracle Databases**

Oracle is one of the most popular relational databases in the world. It has an extensive feature set and functionality. Unlike open-source databases, oracle databases are developed, licensed, and managed by Oracle. AWS RDS for Oracle supports multiple versions of the Oracle database:

- Standard
- Standard One
- Enterprise

Each of these supported versions of AWS RDS for Oracle has different performance, flexibility, and scalability options. AWS offers two licensing options with the AWS:

- **License included** – In this version the database is licensed to AWS. Two license options are available for this option:

- Standard Edition One
  - Standard Edition Two
- **Bring your own license** – You have a license for Oracle, and you host your database on AWS. This provides much more license flexibility and is available with these license options:
    - Standard
    - Enterprise
    - Standard Edition One
    - Standard Edition Two

## PostgreSQL Database

PostgreSQL is an open-source relational database. It has a very advanced feature set and enhanced functionality when compared to MySQL.

## Amazon RDS

This is an AWS-managed relational database service. You can use this service to manage and maintain your relational databases. It is not a database. It supports:

- MySQL
- Amazon Aurora
- MariaDB
- Oracle
- Microsoft SQL Server
- PostgreSQL

AWS databases run on virtual machines (EC2 instances). So, when creating an RDS database instance, you have to choose an instance type. Instance type are of similar size and name to an EC2 instance. Running on an EC2 instance gives RDS the ability to autoscale (scale vertically and scale horizontally on demand).

In addition to the built-in authentication of the supported databases, RDS can use AWS Identity and Access Management (IAM) to set permissions and control who can log into the database.

RDS works across availability zones (AZs) (multi-AZ). This is a good model for disaster recovery. You can have the primary database in one AZ and the standby in another AZ. If the primary database fails, you can fail over to the standby database in the other AZ.

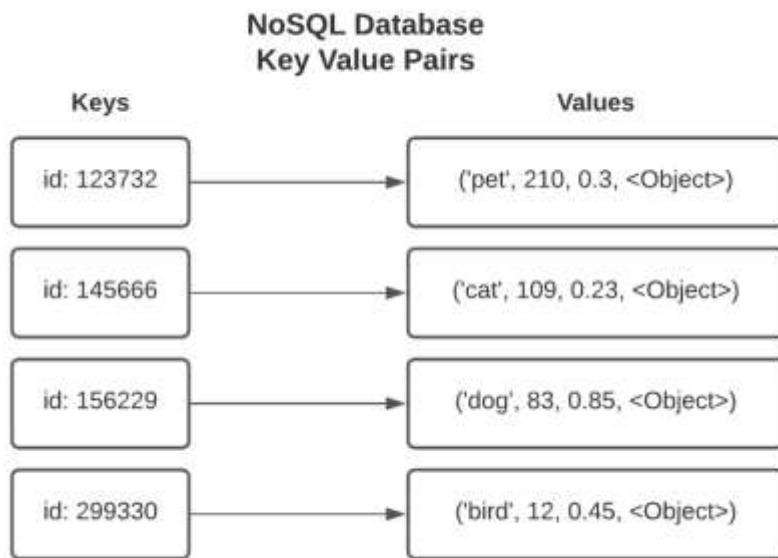
RDS supports synchronous replication to the standby. This means that data is copied over to the standby simultaneously as it is being written to the storage of the primary database. RDS also supports read replicas. Read replicas can span multiple regions. If there is a problem, you can manually promote read replicas to be a stand-alone writeable database.

## No SQL Databases

A NoSQL database stands for “not only SQL”. NoSQL databases facilitate enhanced flexibility and scalability by allowing a more flexible database schema. NoSQL databases can handle structured and nonstructured data. Being able to work with structured and unstructured data can allow NoSQL databases to scale far beyond relational databases.<sup>14</sup> NoSQL databases are optimal under the following circumstances:

- When you need to store large amounts of unstructured data.
- When the database schema may change.
- When you need flexibility.
- When an organization needs rapid deployment of the database.

The diagram below shows how NoSQL databases work with key value pairs.



## **DynamoDB**

Many enterprises require a database with near-unlimited scalability and flexibility beyond what can be achieved with a relational database. AWS offers a fully managed NoSQL database called DynamoDB for organizations that need the scalability and flexibility of a NoSQL database.

DynamoDB is a fully managed, high-availability NoSQL database service. DynamoDB has multiple advantages:

- Fully managed by AWS, so there is less management for the organization.
- Because it is serverless, there is near unlimited scalability, as the database is not bound to the capacity of a physical server or servers. Additionally, AWS manages servers, operating systems, and security.

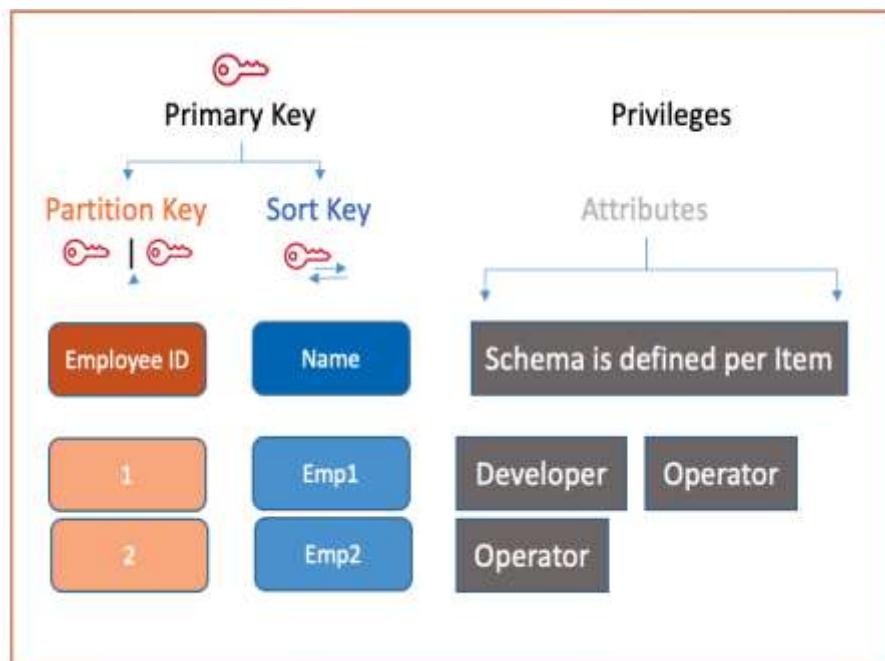
- High availability – By default, DynamoDB is placed in multiple availability zones.
- High-performance storage – DynamoDB uses high-performance SSD storage.
- Data protection – All data is encrypted by default in DynamoDB.
- Low latency – DynamoDB can be configured for sub-millisecond latency when used with DynamoDB Accelerator (which is an in-memory cache for DynamoDB).
- Backups – DynamoDB can be backed up with minimal or no effect on database performance.

### Key Things to Know about DynamoDB

DynamoDB has a very flexible schema, as it is not bound to the same table and column scheme used by relational databases. This flexibility allows for significant customization and scalability. DynamoDB works best with name/value pairs in the primary index. DynamoDB can also have secondary indexes, which allow applications to use additional query patterns over traditional SQL databases.

DynamoDB secondary indexes can be global or local. Global indexes can span across all database partitions. Secondary partitions have virtually unlimited capacity. The only real limitation is a key value cannot exceed 10 gigabytes (GB). Local secondary indexes have the same partition key as the base table.

The diagram below shows an example of the DynamoDB key value pair architecture:



DynamoDB, by default, does not follow the ACID model used by SQL databases, which helps increase its scalability.

DynamoDB by default uses the BASE model:

- Basically Available – The system should be available for queries.
- Soft State – Data in the database may change over time.
- Eventually Consistent – Writes to the database are eventually consistent. This means that a database read immediately after a write may not be available, but will be over time.

DynamoDB with the default configuration follows the eventually consistent model. However, DynamoDB can be configured to have strongly consistent reads if needed.

## **DynamoDB Pricing**

DynamoDB is priced based on throughput. To achieve the best performance and pricing, it is necessary to provision read and write capacity. Read and write capacity are provisioned prior to use and should be set to accommodate the organization's needs.

Autoscaling can also be used with DynamoDB. Autoscaling will watch the databases and scale up as needed, but it will not scale down. Since DynamoDB autoscaling does not scale back down, this method can lead to higher long-term costs.

## **When to Use DynamoDB**

DynamoDB is the optimal choice when near unlimited database scalability and low latency are required. Additionally, DynamoDB is an excellent choice when storing data from a large number of devices, such as Internet of Things (IoT).

Some common DynamoDB use cases are:

- Gaming applications
  - Storing game states
  - Player data stores
  - Leaderboards
- Financial applications
  - Storing a large number of user transactions
- E-commerce applications
  - Shopping carts
  - Inventory tracking
  - Customer profiles and accounts

## **Amazon DynamoDB Accelerator (DAX)**

Amazon DynamoDB Accelerator (DAX) is a managed service that provides high availability and high performance in memory cache for Amazon DynamoDB. DynamoDB Accelerator offers sub-microsecond latency. DynamoDB Accelerator can handle up about a million requests per second.

## **Document DB**

DocumentDB is a fully managed MongoDB-compatible database. DocumentDB is highly scalable and will autoscale up to 64 TB.

Key points to note about Document can be seen below:

- Fully managed
- MongoDB compatible
- Offers extreme performance
- Highly secure
- Industry compliant
- Highly available

## **Apache Cassandra**

Apache Cassandra is an open-source NoSQL distributed database. As a NoSQL database, it is extremely scalable. Apache Cassandra is designed for high performance and availability. In fact, Apache Cassandra is designed to be run across a large number of commodity servers, increasing performance and availability.

Key points of Apache Cassandra can be seen below:

- Distributed across multiple servers
- Highly scalable
- Fault tolerant
- Tunable consistency

## **Data Warehousing on AWS**

Data warehouses are designed to assist with business intelligence and analytics. Data warehouses are designed to perform analysis on large amounts of historical business data. A data warehouse is comprised of the following components:

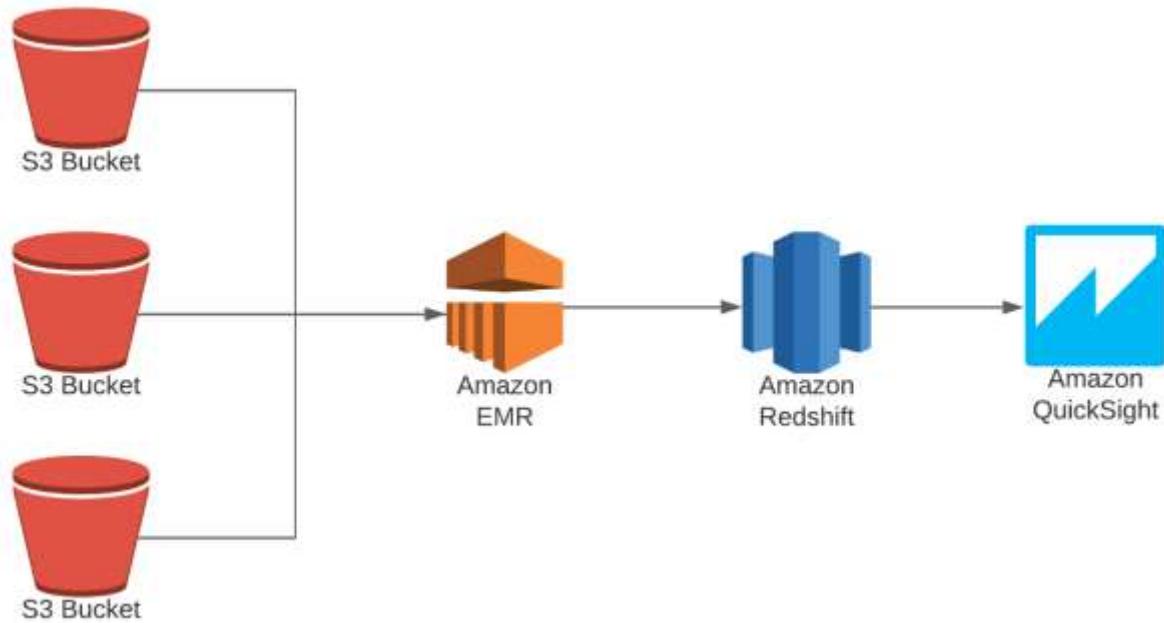
- Database to store data.
- Tools for visualizing the data.
- Tool for prepping and loading data.

## Redshift

Data warehousing is becoming a critical business tool. Data warehouses enable businesses to get actionable insights from their data. The AWS data warehousing solution is called Amazon Redshift. Amazon Redshift is a fast, powerful, and fully managed data warehouse solution. Amazon Redshift supports petabyte-scale data warehousing. Amazon Redshift is based upon PostgreSQL and supports SQL queries. Additionally, Redshift will work with many applications that perform SQL queries.<sup>22</sup>

Redshift is a highly scalable platform. The Redshift architecture is built around clusters of computing nodes. The primary node is considered a *leader node*, with supporting nodes called *compute nodes*. Compute nodes support the leader node. Queries are directed to the leader node.

The diagram below shows how data warehousing is performed with Amazon Redshift and other AWS services:



## Scaling Amazon Redshift Performance

Scaling Amazon Redshift is achieved by adding additional nodes. Amazon offers two types of nodes:

- Dense compute nodes – Dense compute nodes are based upon high-speed SSD RAID arrays.
- Dense storage nodes – Dense storage nodes are based upon magnetic disk RAID arrays.

Generally speaking, SSD-based arrays have higher throughput than magnetic arrays. SSD-based arrays have much higher IOPS than magnetic-based RAID arrays and perform much better in applications that require high input/output (IO) performance.

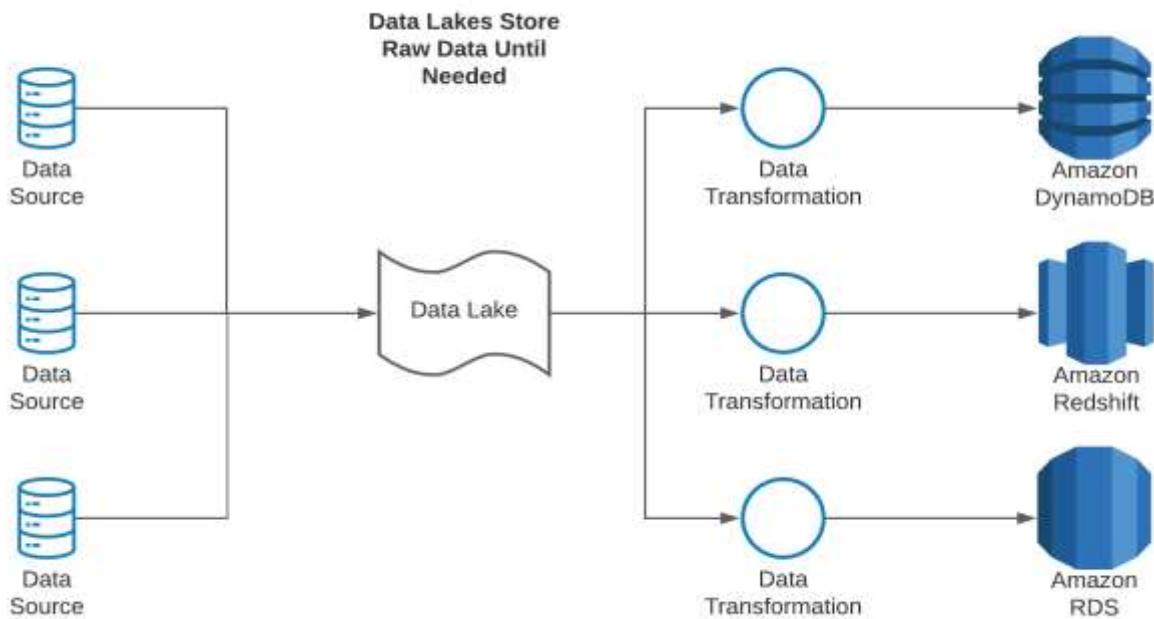
## **Data Lakes**

A data lake is not exactly a database, but it includes database elements, so it's included in the database section of this book.<sup>15, 16</sup>

### **What Is a Data Lake?**

A data lake is a repository that allows an organization to store structured and unstructured data in the same place. Data lakes allow an organization to store and analyze extremely large amounts of data.

The diagram below shows an example of a data lake on the AWS platform:



### **Benefits of a Data Lake**

Data lakes allow an organization to store virtually any type of data at an almost unlimited scale. Unlike a database, a data lake can store data in its native format until it is needed by another application. In a data lake, the data can be queried and searched for relevant data and solutions to current problems. Data lakes are highly adaptable and can be changed at any time to meet an organization's requirements.

## **Database Storage Options**

After determining the optimal database for an organization's needs, it is necessary to determine the proper storage options for the database. AWS databases are stored on EBS volumes, and the database storage options are:

- Previsioned IOPS (PIOPS)
  - Highest performance
  - Lowest latency
  - Highest throughput
- General Purpose SSD
  - High performance
  - Lower latency
  - Moderate throughput
- Magnetic storage
  - Moderate performance
  - Moderate throughput
  - Lowest cost
  - Highest latency
  - Designed for light IO requirements

## **Database Management and Optimizations**

There are many components to successfully architecting and scaling an enterprise-wide database. This section will cover the following topics:

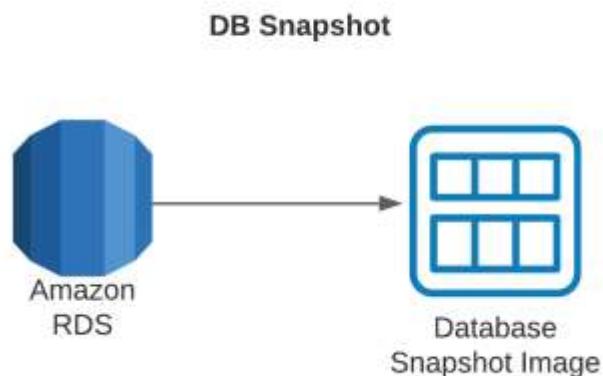
- Backing up the database.
- Scaling the database.
- Designing for high availability.
- Protection of data with encryption.
- Extraction, transforming, and loading tools (ETL).

## **Backing Up the Database**

Databases are automatically backed up by AWS. Database backups copy the entire server, not just the data stored on the database. Backups can be retained for up to thirty-five days, which is configurable from one to thirty-five days. Automated backups happen at a defined window each day. While the database is being backed up, it may be unavailable or have significantly degraded performance.<sup>23</sup>

Databases can also be backed up manually. Manual backups are in the form of a DB snapshot. DB snapshots are a point-in-time copy of the database's EBS volume. DB snapshots are maintained until manually deleted.

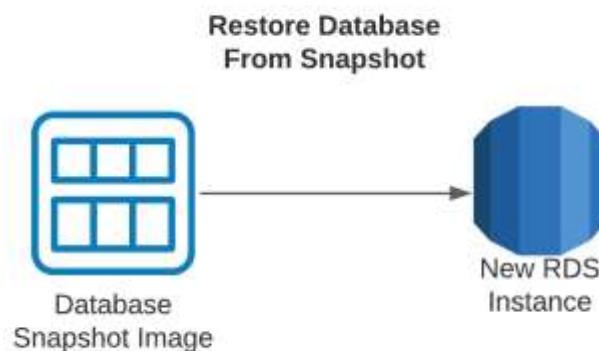
The diagram below shows how a database is manually backed up on the AWS platform:



### Restoring a Database Backup

Databases can easily be restored from backups. When an organization needs to restore a database, a new database instance is created. Since a new database is created, it will have a new IP address and DNS name. Therefore, if a database is restored, it may be necessary to update other applications with the new IP address or DNS name.

The diagram below shows how a database is restored from a snapshot image:



## **Scaling the Database**

Databases are mission-critical applications for many enterprises. As a mission-critical application, the database must scale to meet an organization's needs. There are several methods to increase the scalability of the database. Often it will take a combination of scaling methods to meet an organization's needs.

The first scaling method refers to scaling up versus scaling out. *Scaling up* refers to simply increasing the capacity of the server housing the database. At some point scaling up is not feasible, as a database can exceed the capacity of even the most powerful servers. Scaling out, by comparison, involves adding additional compute instances. Scaling out has two options: partitioning for NoSQL databases and read replicas for relational databases.

### **Scaling Out for Relational Databases**

Relational databases are scaled out by adding additional servers. With relational databases, the additional servers are called *read replicas*. A read replica is a read-only copy of the main database instance. Read replicas are synchronized in near real time. Read replicas are used to decrease the load on the main database server by sending read requests to read replicas as opposed to the primary server. This reduces CPU, memory, and disk IO on the main server. AWS supports up to five read replicas. Read replicas are helpful in the following scenarios:

- When there is a lot of read activity.
- To increase performance, but read replicas are for performance and not disaster recovery.
- When query traffic is slowing things down.
- For more capacity, as offloading read requests from the main database can save valuable resources.

### **Scaling Out for NoSQL Databases**

Partitioning the database involves chopping the database into multiple logical pieces called *shards*. The database has the intelligence to know how to route the data and the requests to the correct shard. Effectively, sharding breaks down the database into smaller, more manageable pieces. Partitioning the database is effective for NoSQL databases like DynamoDB and Cassandra.

### **Database Caching**

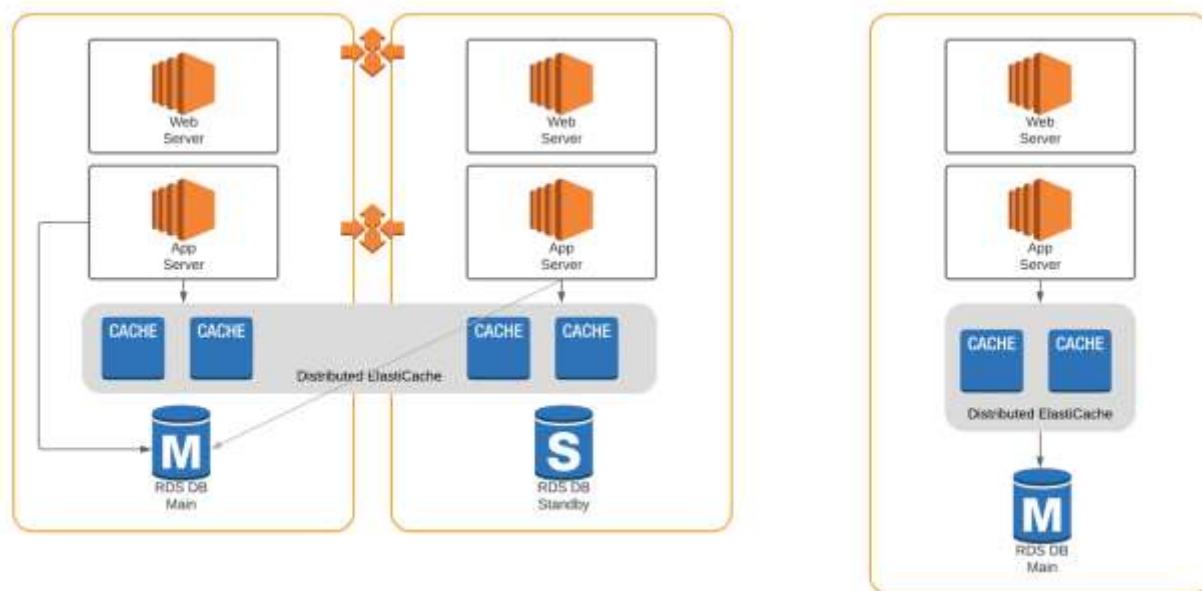
Database caching is another means to increase the scalability of a database. Caching is a method that takes frequently accessed information and places it in memory, so the request does not need to be forwarded to the database server.

Caching works by taking the first request for information to the database server. The server then responds to the request, and the cache temporarily stores the results of the request in memory. Future requests for the same information will be responded to by the cache and will not be sent to the database server. Future requests for information not stored in the cache will be sent to the server.

Caching data reduces requests to the server, freeing up server resources. To prevent stale data, the cache will not keep information in memory forever. Instead, the cache has a timeout to expire old data and make room for fresh data. This timeout, referred to as the *time to live* (TTL), can be configured based upon an organization's needs.

AWS supports two caching types. These are ElastiCache for Memcached and ElastiCache for Redis. Memcached is designed for simplicity. ElastiCache for Redis has a substantial feature set and functionality. Caching is an excellent method to increase scalability. Caching is beneficial only when there are frequent requests for the same information, or queries, as if all requests are for new information, they will all be sent to the main server, mitigating any benefit to the cache.

The diagram below shows an example of database caching on the AWS platform:



## Database Queueing

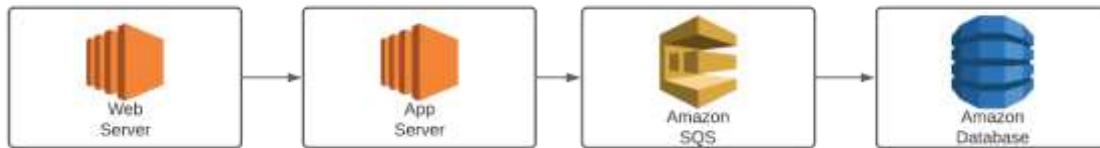
Database queueing can make a significant improvement in increasing the write performance of a database. Additionally, queueing can significantly help reduce CPU usage and other resources in the database. AWS has a queueing service called Simple Queue Service (SQS). Using SQS effectively decouples the database writes from the actual database.<sup>24</sup> The database works in the following manner:

1. Data is sent to the SQS queue.
2. The queue looks at the database.
3. If the database is free, the message is sent from the queue to the database and removed from the queue.
4. If the database is busy or unavailable, the message waits in the queue until the database is available.
5. When the database is available, the message is sent to the database and removed from the SQS queue.

AWS offers two versions of the SQS queue:

- **AWS standard SQS queue** – This is a simple queue to temporarily store messages prior to being written to the database. With Standard SQS queues, there is no guarantee of the order of messages leaving the queue. This is the default option.
- **First in, first out (FIFO)** – This option guarantees that the messages will exit the queue in the order that they were received

The diagram below shows an example of an SQS queue on the AWS platform:



### How Does SQS Help?

SQS helps by reducing write contention to the database. SQS helps with availability as well; if the database is temporarily down, messages can be retained in the queue. Since messages are retained in the queue until the database is ready, it can dramatically smooth CPU, memory, and disk IO performance. Additionally, the number of messages in the SQS queue can be used to scale out the database or computing service using the SQS queue.

### When to Use SQS

SQS is optimal to use in the following circumstances:

- To increase scalability when there are a lot of write requests to the system.
- To decrease the load on the database behind the SQS queue.
- When it's not known exactly how much performance is needed but the organization wants to be able to account for large spikes in traffic.
- When extra insurance is desired that critical messages won't be lost.
- When you want to decouple your application to increase availability, modularity, and scalability.

## AWS Glue

Databases provide an excellent means to store data. As explained in this chapter, there are many types of databases, and each has its strengths and weaknesses. It is often necessary to take data from one database and place it in another database.

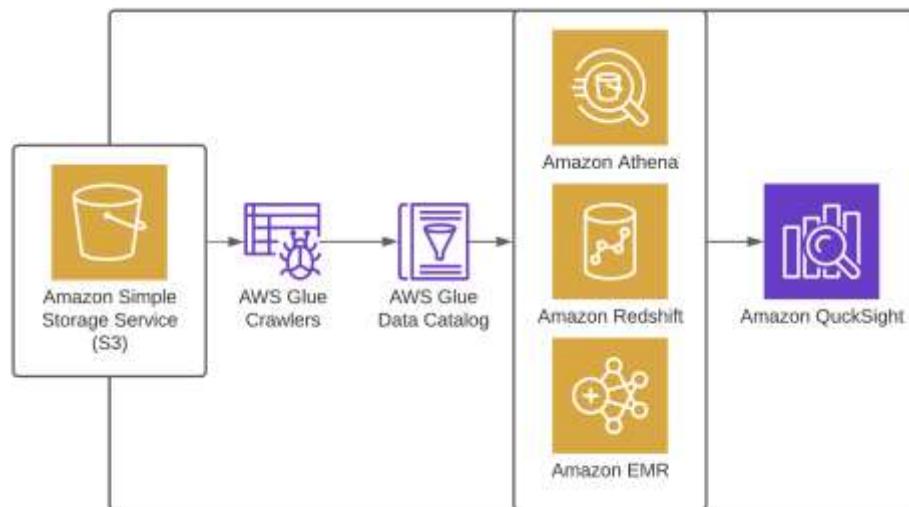
For example, an organization may use RDS for relational database purposes and have data stored on S3, and may want to start using Amazon Redshift. Taking the data from the RDS database and S3 may take some transformation prior to loading into the data warehouse. Transforming data is typically accomplished with extraction, transformation, and loading tools (ETL).

AWS has a fully managed and serverless ETL tool called Amazon Glue. Amazon Glue helps organizations by providing a service to take their data from all sources and load the data to the ultimate destination.<sup>25</sup>

### AWS Glue Highlights

- Point AWS Glue to the location of the data stored on AWS.
- Glue will automatically discover the data and create a metadata catalog.
- After the data is cataloged, it is searchable and queryable.
- The data can be queried directly or loaded into a database or data warehouse.

The diagram below shows an example of using AWS Glue:



## **AWS Database Migration Tools**

### **Moving the Database to the Cloud**

Migrating to the cloud often involves migrating a database from the data center to the cloud. Relational Databases can be migrated to the AWS platform using native tools such as Microsoft SQL Server Migration Assistant or AWS tools such as the *Database Migration Service* and the *Schema conversion tool*. These tools help simplify moving the database to the cloud.

Database migration falls under two main categories: homogeneous migration and heterogeneous migration. A homogeneous database migration involves moving to the same database (i.e., MySQL from the data center to a MySQL server on the cloud). A heterogeneous migration involves migrating to a different kind of database (i.e., MySQL in the data center and Amazon Aurora in the cloud).

### **Database Migration Service**

To assist customers in migrating their databases to the cloud, AWS provides the Database Migration Service (DMS). The DMS helps clients migrate their databases to the AWS platform with zero downtime. The DMS can capture live data; in fact the DMS can run while the database is fully operational. The net effect is a migration with zero downtime.

### **AWS Schema Conversion Tool**

To assist customers in migrating their databases to AWS, AWS provides the Schema Conversion Tool. The AWS SCT provides a graphical user interface to automatically convert the database schema of your source database (OLAP/OLTP) into a format compatible with your target Amazon RDS instance. The AWS SCT evaluates the current schema and migrates to the target database. The AWS SCT then provides a database migration assessment report of any necessary manual steps needed by the customer to complete the migration.

## **Designing a High-Availability Database Architecture**

Given the crucial nature of databases in the enterprise computing environment, making sure the database is available when needed is of utmost importance. The key to all high-availability designs is to avoid any single point of failure.

As a reminder, the AWS network is divided into regions and zones. Regions are large geographic areas, while an availability zone (AZ) is really a data center inside of a region. Regions may have many availability zones.

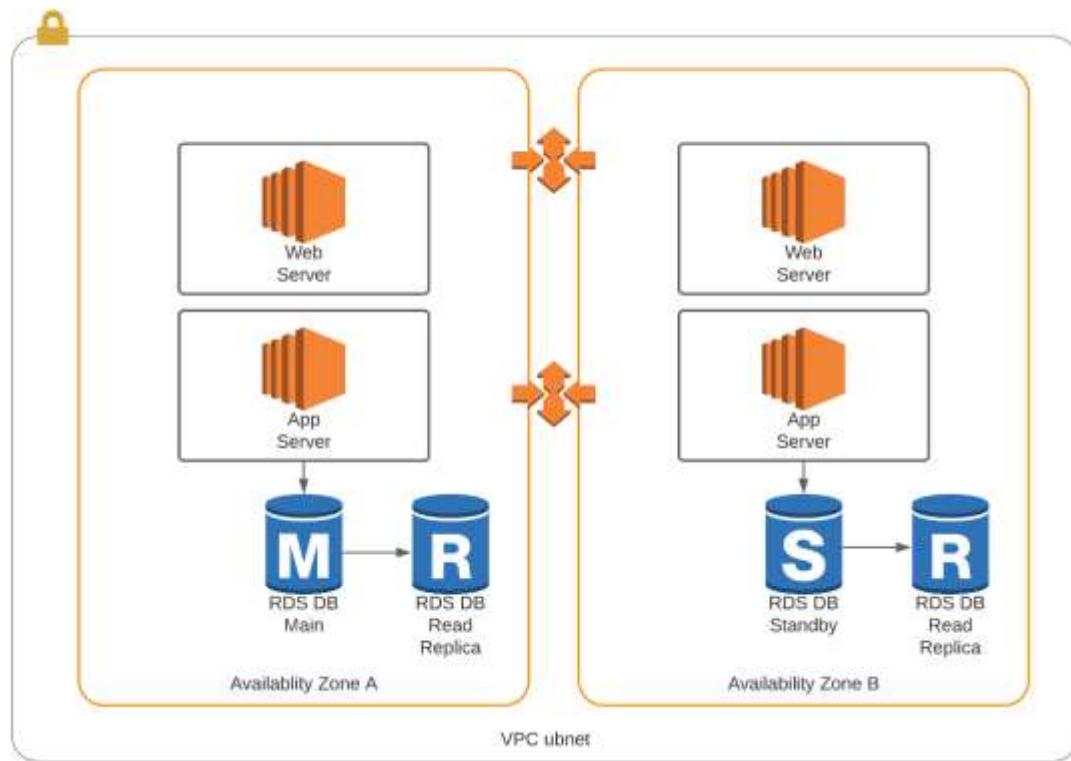
A high-ability database architecture will have database instances placed in multiple availability zones (multi-AZ). In a multi-AZ environment, there are multiple copies of the database, one in

every availability zone. It is important to note that multi-AZ environments do not increase database performance. Multi-AZ environments are for redundancy to enhance availability purposes.

In a multi-AZ environment, data from the primary (master) database is synchronously copied to the backup database in the other AZ. If the primary database were to fail, the database instance in the other AZ would take over. A failover to the backup database will be triggered in the following circumstances:

- The primary database instance fails.
- There is an outage in an availability zone.
- The database instance type is changed.
- The primary database is under maintenance (i.e., patching an operating system).
- A manual failover has been initiated (i.e., reboot with failover).

The diagram below shows a high-availability application and database architecture:

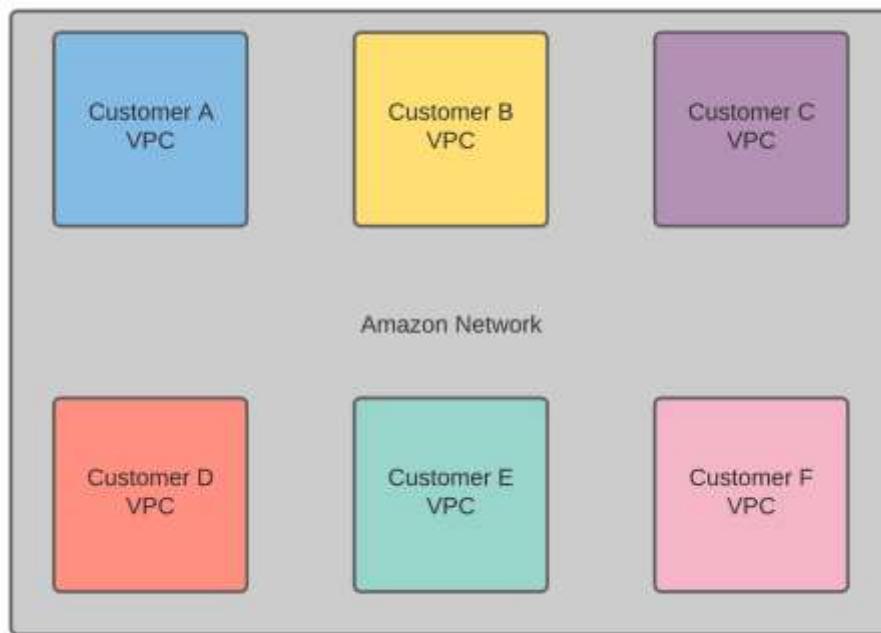


## Chapter 6 - The AWS Virtual Private Cloud (VPC)

### What Is the AWS VPC?

The AWS VPC is essentially a private virtual network inside the AWS network. While AWS is physically a shared network, each VPC is logically isolated from other AWS customers. The AWS network supports private and public addresses for each of its customers. Since AWS customers are logically separated, there is no contention for IP address space between VPCs. Each VPC will have its own routing table that is responsible for directing traffic. As with any network, a proper IP addressing scheme is essential for scalability.<sup>26</sup>

The diagram below shows an example of logically isolated customer VPCs on the AWS platform:



### The OSI Model

Throughout this book, especially the VPC section, we often reference the open systems interconnect (OSI) model. The OSI model divides network communications into seven layers. Each layer of this model has specific functions for network communication. Knowledge of the OSI model can be helpful in troubleshooting and understanding the components of network communication. The table below shows the seven layers of the OSI model and the associated functionality at each level.<sup>27</sup>

Operating Systems Interconnection Model (OSI)				
Layer	Number	Function	Examples	Name
Application	7	User Interface	HTTP, DNS, SSH	Data
Presentation	6	Presentation and Data Encryption	TLS	Data
Session	5	Controls Connection	Sockets	Data
Transport	4	Protocol Selection	TCP, UDP	Segments
Network	3	Logical Address	IP Address	Packets
Datalink	2	Hardware Connection	MAC Address	Frames
Physical layer	1	Physical Connection	Wire, Fiber	Bits

## IP Addressing

An IP address is a logical address assigned to a computing device that identifies that device on the network. An IP address is similar to an address on a home. For mail to be delivered, the house must have a unique address so that the postal service can identify the correct home and deliver the mail. Every address must be unique, even if the only thing that separates similar-looking addresses is the postal code. IP addresses are no different. For any device to talk to another device on an IP network, their addresses must be unique. There are two versions of IP addresses:

- **IPv4** – Original IP address used in networking that has been in use since the 1970s and is still the dominant address space.
- **IPv6** – Newer IP addressing model designed to overcome the limitations of IPv4.

IP addresses operate at layer 3 of the OSI model. IP addresses are logical addresses, in that they are assigned to a network interface. IP addresses are not hard coded, like a MAC address (layer 2), that is permanently assigned to an Ethernet interface.

IP addresses are 32-bit addresses, which was perfect when the internet was formed. However, the internet grew far beyond what anyone expected, and there were not enough public IP addresses. The Internet Engineering Task Force (IETF) came up with two solutions to the IP address space shortage.<sup>28, 29</sup> The first solution is to provide private IP address space as specified by the Request for Comments (RFC) 1918 and the second solution is IPv6 addresses. Private IP addresses are to be used on internal networks and are not globally routable. Private IP addresses are available in the following address space:

- 10.0.0.0/8
- 172.16.0.0/16 through 172.31.0.0/16
- 192.168.0.0/16

## IP Address Classes

IPv4 has five classes of IP address space. Address classes are legacy, and not really used in today's modern environment. IP address classes are covered for historical purposes and so the reader can better understand *classless interdomain routing* (CIDR). Classful addresses are not used anymore since there is a shortage of IP addresses. Instead, subnetting is used to optimize IP address usage and availability.

- Class A addresses
  - 1.0.0.0- 126.255.255.255/8
- Class B addresses
  - 128.0.0.0- 191.255.255.255/16
- Class C addresses
  - 192.0.0.0- 223.255.255.255/24
- Class D addresses (Multicast)
  - 224.0.0.0- 239.255.255.255
- Class E addresses (Experimental)
  - 240.0.0.0- 255.255.255.255

## Subnetting and Supernetting

Since there are a limited number of IP addresses, it's essential to use IP addresses carefully. One way to make use of an organization's IP address space is with subnetting. Subnetting is effectively taking an IP network and chopping it into smaller networks. Please see the graphic below:

Network	Subnet Mask	Effective Addresses	Effective AWS Addresses
192.168.1.0	255.255.255.0	253	
<b>Submitted To /28 Subnets</b>			
192.168.1.0	255.255.255.240	14	11
192.168.1.16	255.255.255.240	14	11
192.168.1.32	255.255.255.240	14	11
192.168.1.48	255.255.255.240	14	11
192.168.1.64	255.255.255.240	14	11
192.168.1.80	255.255.255.240	14	11
192.168.1.96	255.255.255.240	14	11
192.168.1.112	255.255.255.240	14	11
192.168.1.128	255.255.255.240	14	11
192.168.1.144	255.255.255.240	14	11
192.168.1.160	255.255.255.240	14	11
192.168.1.176	255.255.255.240	14	11
192.168.1.192	255.255.255.240	14	11
192.168.1.208	255.255.255.240	14	11
192.168.1.224	255.255.255.240	14	11
192.168.1.240	255.255.255.240	14	11

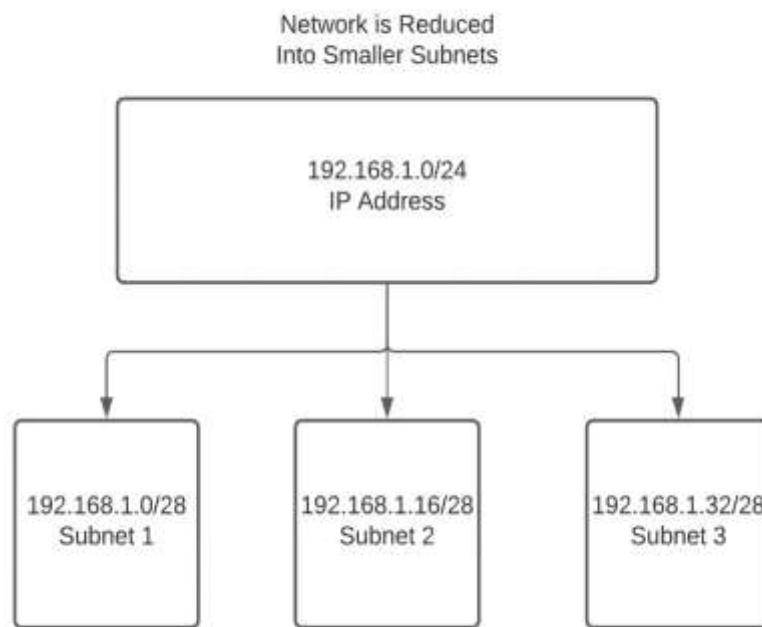
In this table, the 192.168.1.0/24 network has been submitted into sixteen /28 subnets.

Subnetting is critical for two reasons. The first reason is that every interface on a system needs to be on a different network or subnet. The second reason is there is a practical limitation of how many hosts can be on a subnet due to system broadcasts.

All interfaces need to be on a different subnet. Imagine a router with thirty interfaces. Each interface needs to communicate only with the router on the far end of the connection. In practicality, only two IP addresses are needed—one on each side of the connection. If the network 192.168.1.0/24 were attached to both sides of a WAN link, instead of only using two addresses, this link would use up all 253 addresses available on that subnet. By comparison, subnetting from a /24 to a /28 would allow for sixteen subnets available that would each support eleven hosts. Note that this is a reduction from the fourteen available addresses that would be present with a /28 subnet mask. This is because AWS reserves the first three IP addresses and the broadcast address space. An additional point to remember is that the smallest subnet supported by the AWS platform is a /28.

The second key reason that subnetting is essential and is related to constraining broadcast traffic. Host systems often identify each other by sending broadcast traffic to the local subnet. Broadcast traffic is different than traditional traffic. With traditional traffic, a system sends a message to another system. With broadcast traffic, when a host sends a broadcast, every host on the subnet sees and must process the broadcast. Additionally, network switches forward broadcast traffic out of every port except the port where the broadcast was sent. Broadcast traffic can easily overwhelm computing and network hardware. So, limiting the size of broadcast domains is essential. Limiting the reach of broadcasts is achieved by using smaller subnets and routing traffic between subnets.

The diagram below shows an example of a network being partitioned into smaller subnets:



## **Supernetting**

Supernets are the exact opposite of subnets. *Supernets* combine multiple smaller subnets into a larger network called a supernet. As we have previously stated, optimizing performance often involves reducing the size of broadcast domains with effective subnetting. Therefore, supernetting is generally used more to optimize routing than for addressing computer systems.

### **Supernetting to Optimize Routing**

Routers distribute traffic across the network by building a map of all available subnets. These subnets are subsets of the full class A, B, or C networks. Therefore, modern routing is called *classless interdomain routing* (CIDR). Routers often have memory and CPU limitations that limit the number of routes the router can support. As an example, AWS permits only one hundred routes in a VPC. To minimize the number of routes in the routing table while maintaining full reachability, it is often necessary to summarize routes. Route summarization is effectively taking several subnets and supernetting them into a single network. A full discussion of routing, switching, and IP addressing is beyond the scope of this book. For those interested in detailed routing and switching information, we recommend the book *Routing TCP/IP* by Jeff Doyle and Jennifer Dehaven Carroll.

### **How It Works**

Let's look at this section of a VPC routing table:

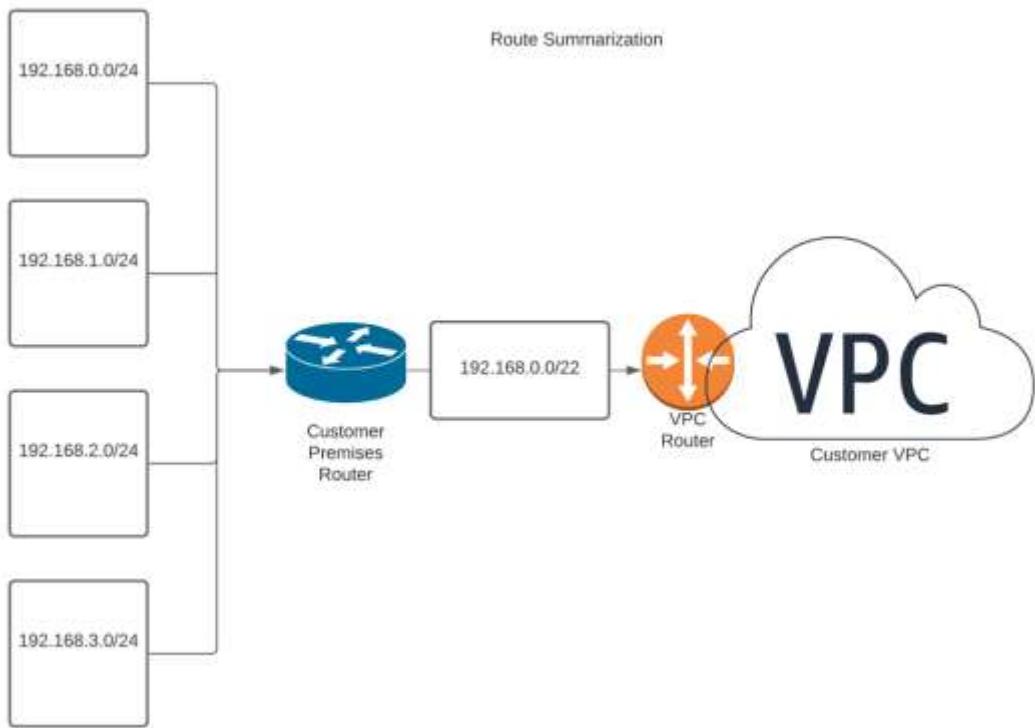
- Destination 192.168.0.0/24 target igw-123456789098765
- Destination 192.168.1.0/24 target igw-123456789098765
- Destination 192.168.2.0/24 target igw-123456789098765
- Destination 192.168.3.0/24 target igw-123456789098765

Instead of putting all these routes in a routing table, the subnets can be supernetted and summarized into the following route:

- Destination 192.168.0.0/22 target igw-123456789098765

As you can see, we have taken four routes and converted them into a single summarized route. This provides full reachability while saving valuable resources in the routing tables. With AWS allowing only one hundred routes, every route counts. So it's best to use an IP addressing scheme that can be summarized and to use route summarization whenever possible. This can also be used for traffic engineering on AWS. To learn more about traffic engineering using the BGP routing protocol that is supported by AWS, we recommend the book *Internet Routing Architectures* by Sam Halabi.

The diagram below shows an example of route summarization with supernetting:



## IPv6 Addresses

When IPv4 was invented, no one could have imagined the growth of the internet. Very quickly the internet would be out of addresses, and something more scalable would be needed. The Internet Engineering Task Force invented a new version of the internet protocol to overcome the weaknesses with IPv4. This new protocol was IPv6. To overcome the address shortage, the 32-bit binary address used with IPv4 was changed to a 128-bit hexadecimal address. This provides infinitely more address space and scalability. Realistically speaking, IPv6 address capacity is likely sufficient to provide every internet-enabled device an IP address.

While IPv6 is the future, IPv4 is still the main IP addressing scheme in use today. IPv6 addresses are becoming more popular. In modern times, most mobile phones have an IPv6 address. AWS automatically assigns an IPv6 address to every interface.

## Components of a VPC

Now that we have reviewed the basic elements of IP addressing, it's time to discuss the AWS-specific components of a VPC. The key components of a VPC can be seen below. This section will cover these VPC components in detail:

- VPC routing tables
- Internet gateways
- Egress-only internet gateways
- NAT instances and NAT gateways
- Elastic IP addresses (EIPs)
- VPC endpoints
- VPC peering connections
- Network access control lists
- Security groups

## Routing Tables and Routing

All VPCs effectively have a virtual router provided by AWS. The virtual router is used to direct traffic to its ultimate destination. Routers determine how to make traffic forwarding decisions based upon their routing tables. A sample routing table can be seen below:

Routing Table Example	Target	
172.16.1.0/24	Local	
192.168.0.0/16	pcx-123456	
192.168.1.0/24	pcx-654321	* most specific
0.0.0.0/0	igw-123456	

In the routing table above, it's clearly visible that the routing table has a destination subnet and a destination interface/gateway to forward traffic. The destination is referred to as the *target* in AWS. Note that if there are multiple paths to a destination, the most specific route will be chosen.

## How Routing Tables Work

Routers build a map of the network. The map of the network will show which interface to use to send traffic to its ultimate destination. Traffic will then be sent to the next router, which will have its map of the network. Packets are forwarded from router to router until they reach their ultimate destination.

The map of the network is called a *routing table*. Routing tables can be built statically or dynamically. Static routes are user configured, where dynamic routes are dynamically learned via a routing protocol. Static routes are ideal when there are very few paths to reach the ultimate destination. Dynamic routes are learned, which is excellent for large networks. Additionally, dynamic routing enables high availability, as the routers can reroute traffic to a backup path if needed. Dynamic routing can reroute traffic by detecting when a router or link is down and calculating an alternate path. Most, if not all, large enterprise networks use routing protocols as part of their high-availability architecture. There are essentially two kinds of routing protocols, and they can be seen below:

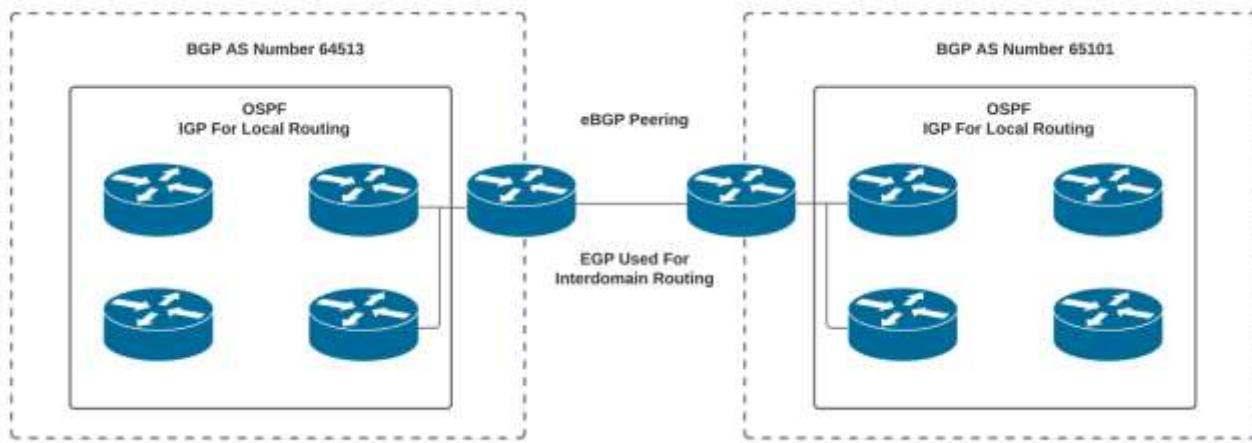
### **Interior Gateway Protocols (IGP)**

- Interior gateway protocols are used to exchange routing information inside of an organization.
- Interior gateway protocols provide a very detailed map of the organization's routes.
- Interior gateway protocols can detect outages and reroute traffic very quickly.
- Interior gateway protocols are tuned for performance at the expense of scalability.
- Some examples of interior gateway routing protocols include OSPF, IS-IS, and EIGRP.

### **Exterior Gateway Protocols (EGP)**

- Exterior gateway protocols are used to exchange routing information across organizations.
- Exterior gateway protocols provide extensive tuning tools, providing the means to engineer traffic and filter routes for scalability, security, and proper routing.
- Exterior gateway protocols are slower to reroute traffic, as they are designed for scalability and tunability.
- Border Gateway Protocol (BGP) is the exterior gateway protocol used by the internet and AWS.
- Exterior gateway protocols are tuned to be able to store an enormous number of routes (assuming the routers have sufficient memory and CPU capacity). At the time of this writing, the internet routing table has over eight hundred thousand routes.<sup>30</sup>

The diagram below shows how an IGP is used for internal routing and an EGP is used for interdomain routing:

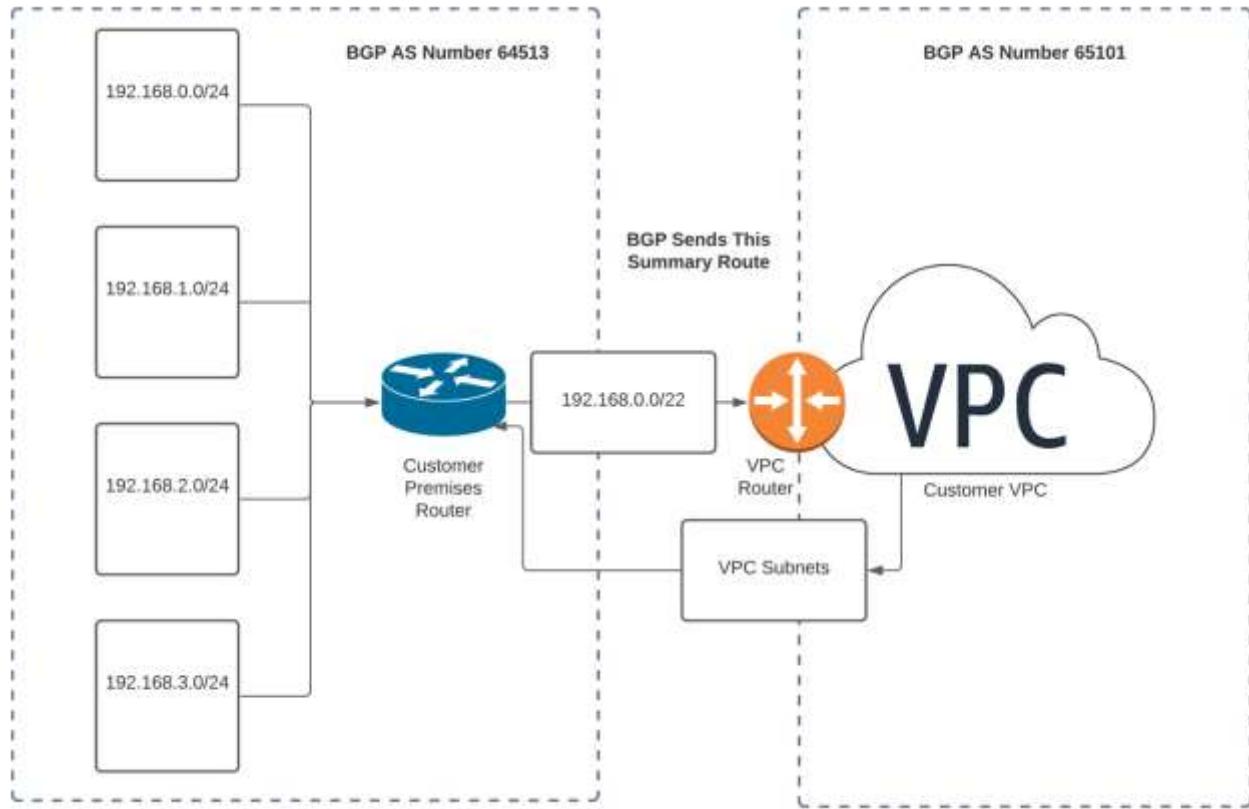


## Dynamic Routing with AWS

AWS supports connecting an organization to AWS with BGP. BGP is a highly tunable and scalable exterior gateway routing protocol. BGP runs on TCP port 179. It is essential when using BGP to connect to AWS that firewalls and network ACLs allow TCP port 179. BGP enables an organization to have multiple connections to the internet or AWS and load share across these connections. As with all BGP routing, an autonomous system number is required to identify an organization to the AWS network.

BGP is required when using a direct connection to connect to AWS. AWS supports some of the available BGP tuning options, like communities. AWS BGP implementation supports the well-known community “no export,” which prevents a VPC from becoming a transit autonomous system for the internet. (This means you won’t become an accidental internet service provider and have parts of the internet connecting through your VPC.) AWS has a light BGP implementation when compared to core internet routers. AWS supports a maximum of one hundred routes learned from BGP; therefore an IP addressing scheme capable of route summarization is necessary. AWS supports some of the BGP tuning options, such as weight, local preference, and Autonomous System path (AS path). Please see below for an example of a system connecting with BGP to share routing information via BGP to AWS.

The diagram below shows how an organization would use BGP to connect to AWS to share routing information:



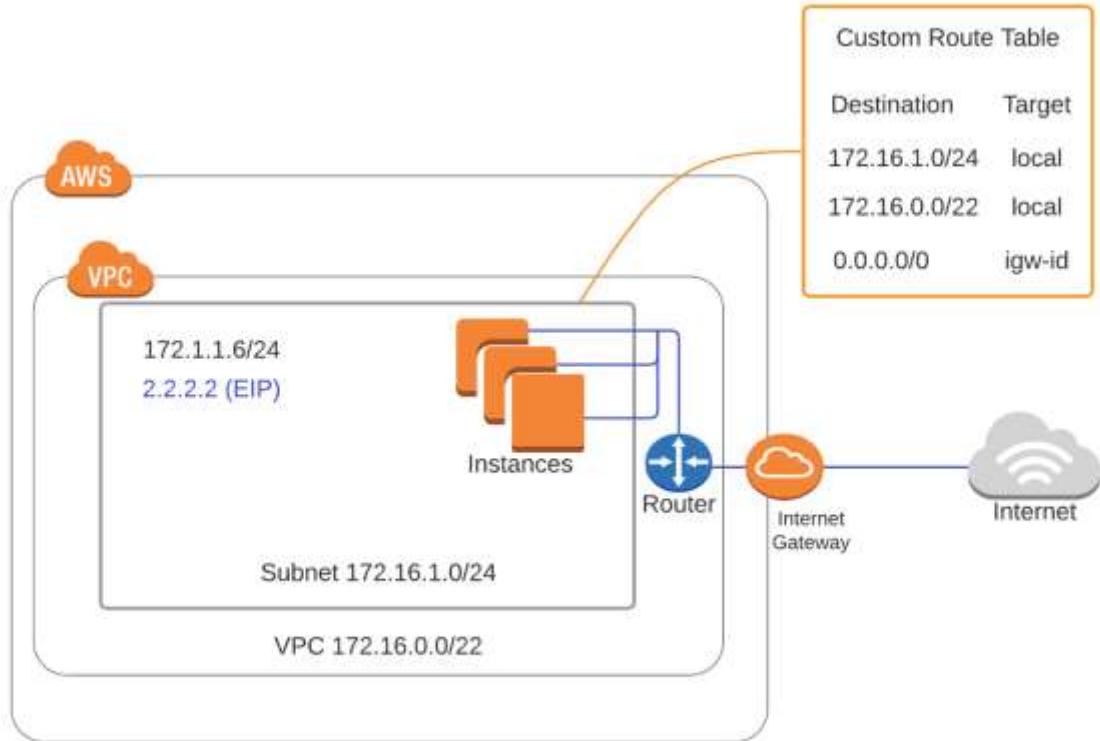
## Internet Gateways (IGW)

In order to connect to the internet, an internet connection and an internet gateway must be configured. The internet gateway is a router with an internet service provider connection. AWS provides internet service to VPC customers when the customer sets up an internet gateway. The AWS internet gateway is a high-availability, redundant internet router. When using the internet gateway, it will have a route to all internet destinations or a default route to an upstream provider. Additionally, the internet gateway will translate private IP addresses into a public address for internet connectivity.<sup>31</sup>

An internet gateway is created in the following manner:

1. Attach an IGW to the VPC.
2. Create a default route to send all internet-destined traffic to the internet gateway.
3. Assign a public IP address to the gateway.
4. Configure security. Systems will be reachable from the internet, so it is essential that systems are patched for security vulnerabilities and firewalls; network ACLs and security groups are configured.

The diagram below shows an example of an internet gateway on the AWS platform:



### Egress-Only Internet Gateways

Egress-only internet gateways allow internet connectivity to IPv6 systems. IPv6 does not really use private address space. So essentially all IPv6 addresses are unique and globally routable. Therefore, IPv6 systems do not need NAT to connect to the internet, as there is no need to translate an internal address into an external address.

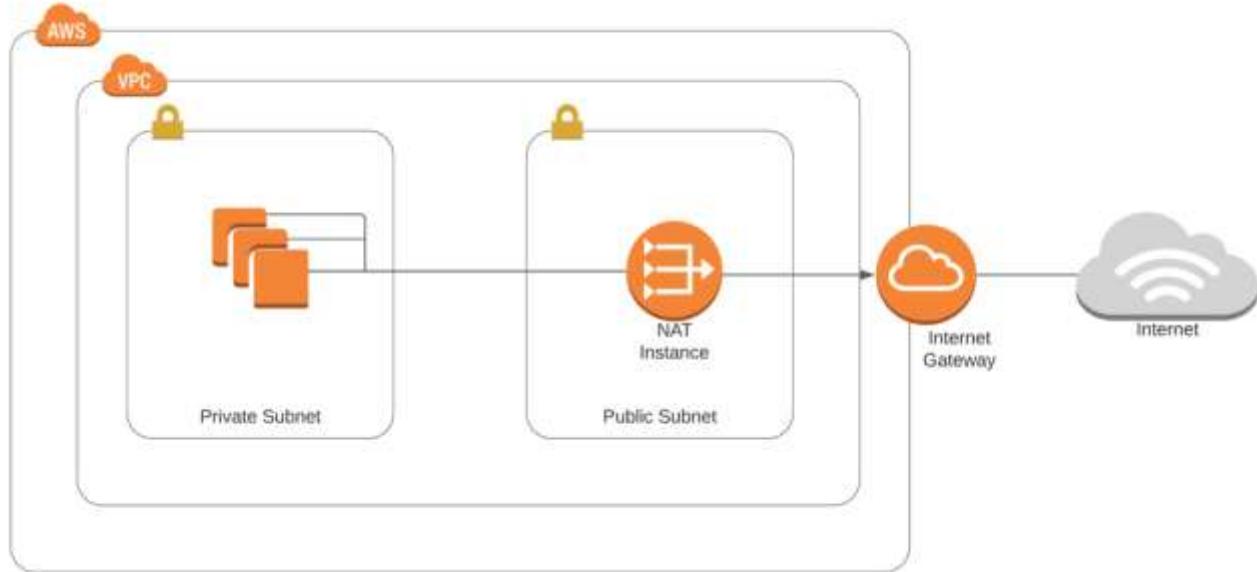
When using an egress-only internet gateway, systems will not be reachable from the internet. This type of internet gateway is stateful and allows traffic established by internal hosts to return to the AWS VPC. This allows internal systems to download software patches and upgrades from the internet while keeping outsiders from connecting into the AWS VPC from the internet. This is very similar in function to the NAT gateway without an internet gateway in IPv4.<sup>32</sup>

### NAT Instances

A *NAT instance* is a custom AWS virtual machine that translates private IP addresses into public IP addresses. The NAT instance is available as an AMI, and it runs on an EC2 instance. A NAT instance must be in a public subnet with a route to the internet gateway. This type of setup is used for egress only, meaning internal systems can connect to the internet, but systems on the

internet will not be able to connect to systems in the VPC. Additionally, the VPC routing table must have a default route to the internet gateway. This is really a legacy product and has largely been replaced with the AWS NAT gateway.<sup>33</sup>

The diagram below shows an example of a NAT instance on the AWS platform:

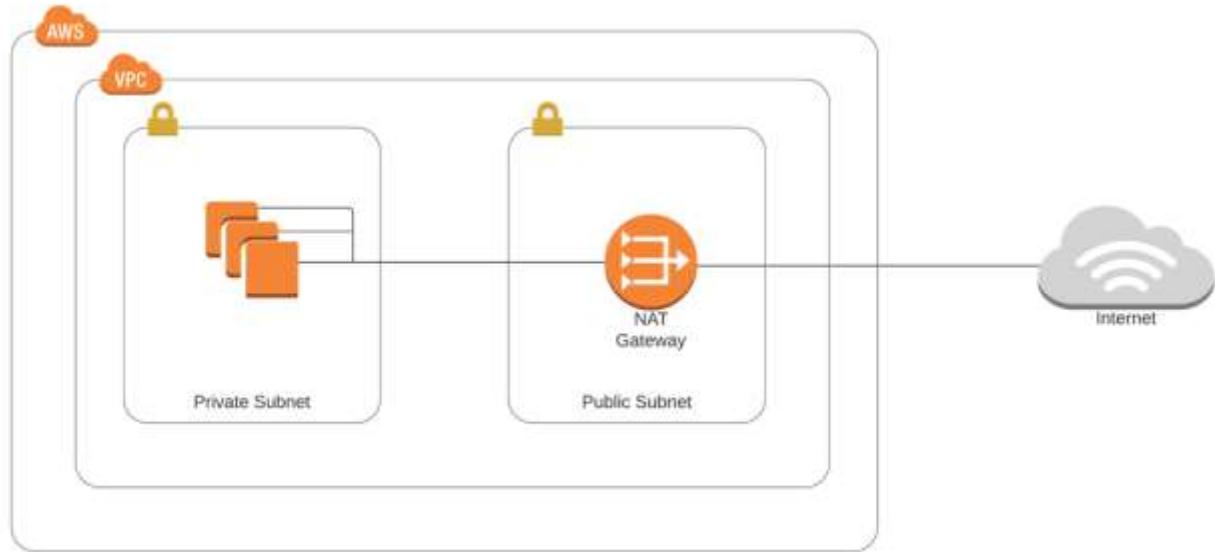


## NAT Gateway

A *NAT gateway* is a fully managed NAT service. The NAT gateway is highly available and redundant inside of an availability zone. A NAT gateway provides egress-only internet connectivity, similar to a NAT instance. This means that inbound connections from the internet will be refused, but internal hosts will be able to connect to the internet. A NAT gateway is optimal when hosts need to connect to the internet to download software patches but desire to keep their systems off the public internet for security reasons.<sup>34</sup>

The NAT gateway is configured in a public subnet. Then a default route must be created to send internet-destined traffic to the NAT gateway. Note that the NAT gateway will use an elastic IP for the life of the NAT gateway.

The diagram below shows an example of a NAT gateway on the AWS platform:



### **Elastic Network Interfaces (ENIs)**

A network interface is what connects a system to the network. AWS's version of this is called the Elastic network interface. *Elastic Network Interfaces* (ENIs) are virtual network interface cards, which are generally attached to EC2 instances. By default, there is an ENI created when you launch an EC2 instance. The default ENI is eth0. Systems can be placed on two subnets (dual-homed) or multiple subnets by creating multiple ENIs and attaching those ENIs to an EC2 instance. Multiple ENIs are commonly used for system management—for example, on a web server, the public side open will be open to http/https, and the private ENI will be open to port 22 for SSH.

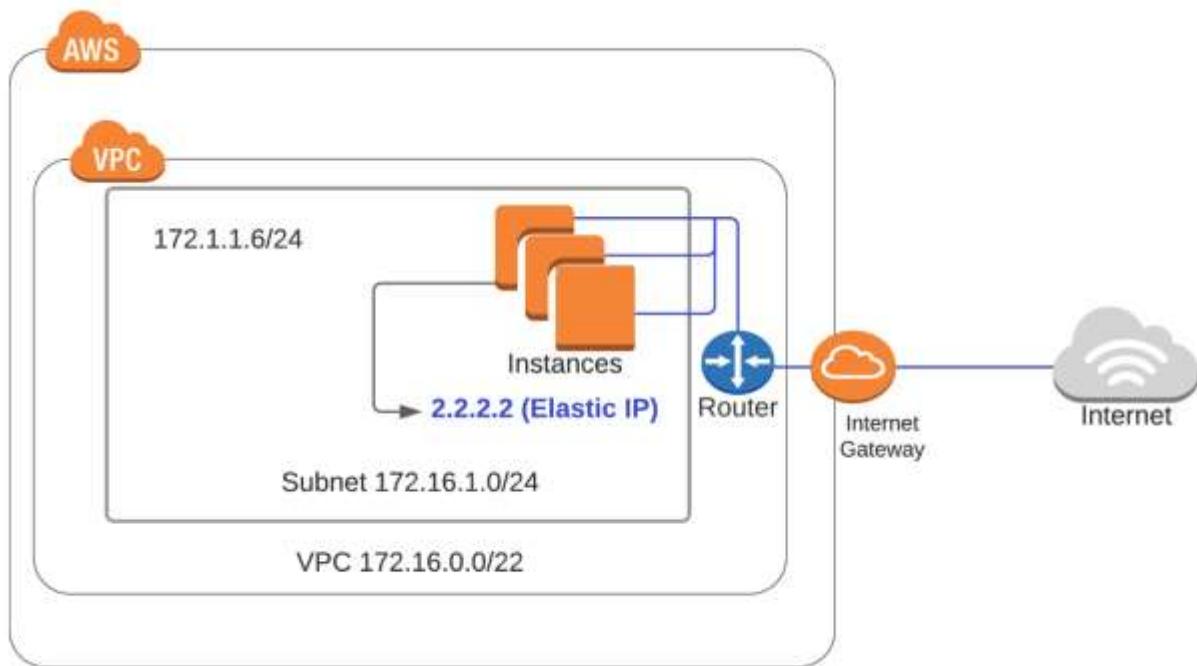
### **Elastic IP Addresses (EIPs)**

AWS maintains a pool of public IP addresses for their customers to use on the internet. These public IP addresses are called Elastic IP addresses. Elastic IP addresses work in the following manner:<sup>35</sup>

1. When an organization needs a public address, it sets up an Elastic IP address.
2. The EIP is taken from the AWS public address pool and dedicated to the customer's Elastic IP address.
3. The customer can keep this EIP as long as they are using the address.
4. When the customer no longer needs the EIP address, the customer closes the EIP, and the address is sent back to the AWS pool for future customer use.

An EIP can have a single public address that is mapped to multiple private IP addresses, with the main address being the primary address and the additional addresses being secondary addresses. Secondary IP addresses are useful during IP address migrations, as they allow for connectivity while IP addresses are changed. Secondary addresses are often used when an organization merges with another organization, and IP addresses need to be modified to allow for full connectivity. This often occurs when both organizations are using the same private IP address space.

The diagram below shows an example of using an Elastic IP address on the AWS platform:



## Endpoints

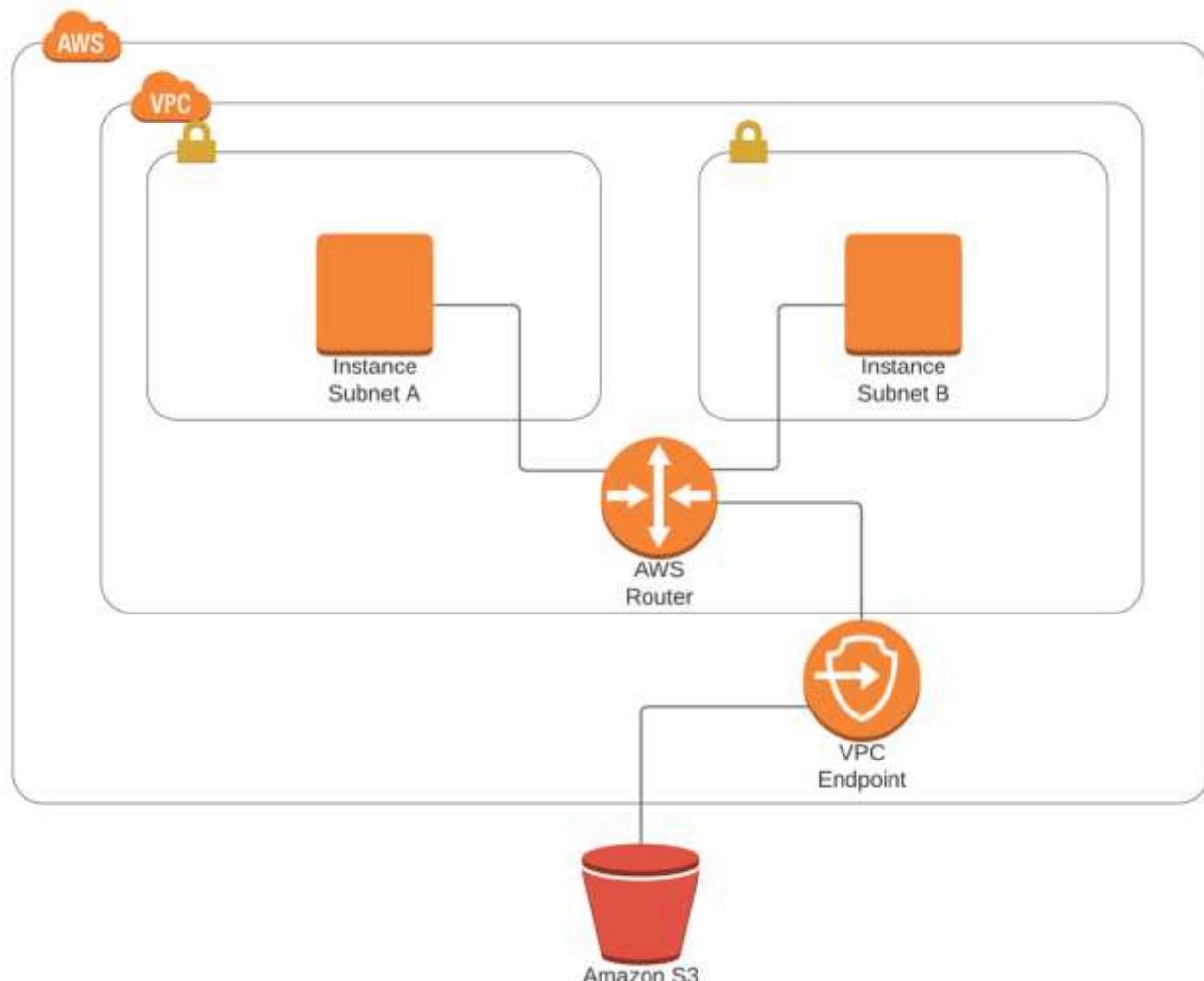
Internet gateways and NAT gateways offer a solution to connect a customer VPC to the public internet. While it's possible to connect to many AWS services over the public internet, it's not the preferred method. Connecting to the AWS services over the Amazon network can offer significant benefits, especially when connecting to other services or VPCs on the AWS platform. Connecting to another AWS service is performed with the creation of a VPC endpoint. When connecting with VPC endpoints, the connectivity is established over the AWS high-speed network backbone.<sup>36</sup> This can offer the following benefits:

- **Privacy and security** – Sending data over the AWS network is much more private and secure than the internet.

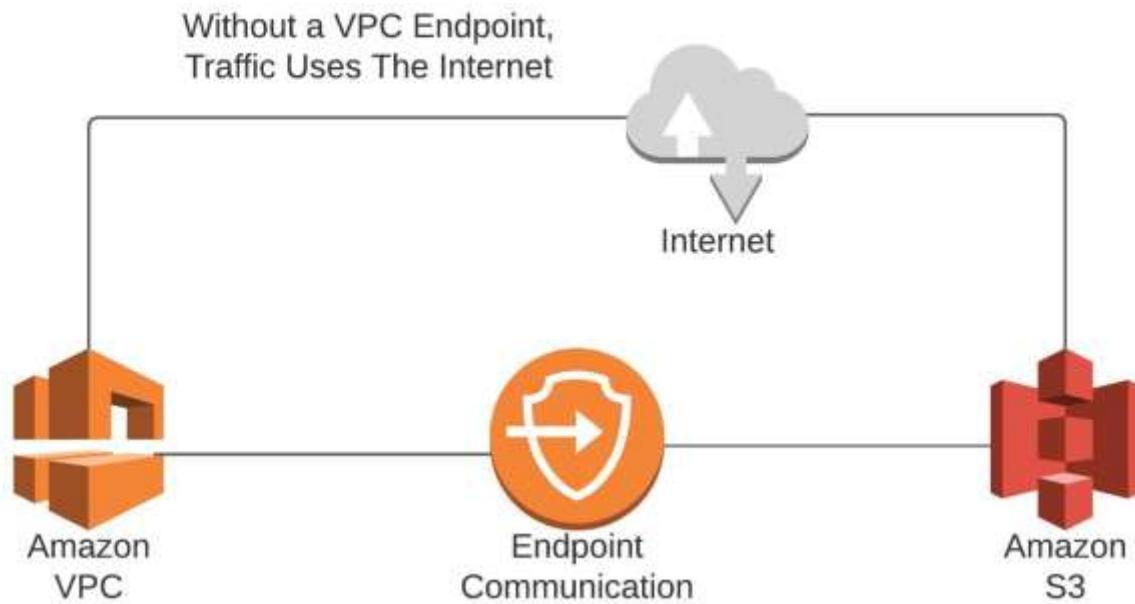
- **Performance** – Internet gateway speeds often lower performance than sending data directly over the AWS network.
  - The AWS network is fully managed by AWS. Therefore it can have lower latency and lower congestion than the public internet. The internet has no performance guarantees across autonomous systems.
- **Cost control** – AWS charges for internet use. Sending data over the AWS network will cost less than internet use.
- **Simplicity** – VPC endpoints do not require a public IP address, internet gateway, or NAT gateway.

The way to connect your organization's VPC to AWS without traversing the internet is with a VPC endpoint. VPC endpoints are virtual devices used for routing within the AWS network. There are two types of VPC endpoints: interface endpoints and gateway endpoints.

The diagram below shows an example of a VPC endpoint on the AWS platform:



Here is an example showing how without the end point, traffic would need to use the internet:



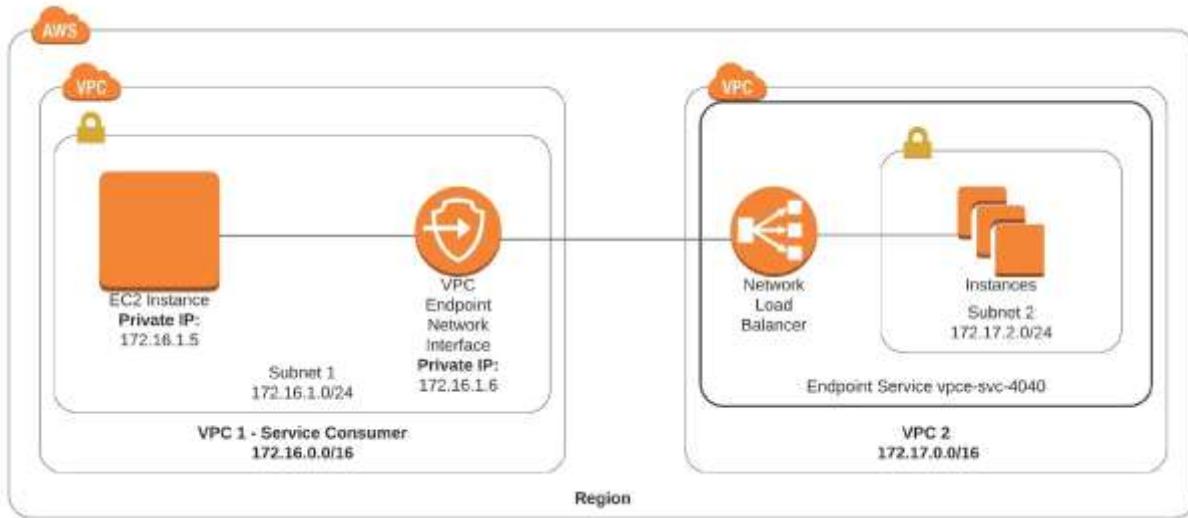
### Interface Endpoints

An *interface endpoint* is a means to connect to AWS services or external organizations or services. Certain AWS services are accessed via interface endpoints such as EC2 Systems Manager and Kinesis. External VPCs are also connected via interface endpoints. Interface endpoints automatically create an elastic network interface that uses a private address from the VPCs address pool. The interface endpoint serves as an entry point from your organization to supported services. Supported services include AWS services and other VPCs. Interface endpoints use the AWS PrivateLink service. The PrivateLink service restricts all access between VPCs or AWS services. Interface endpoints are compatible with most VPC services.<sup>37</sup>

Interface endpoints work differently than gateway endpoints. With a gateway endpoint, there is a route in the routing table created to access the external endpoint. Interface endpoints, on the other hand, are effectively a network interface. You can control the traffic destined to the endpoint with a security group.

After creation of the interface endpoint, AWS creates an ENI in your subnets that you specify to use the interface endpoint. After the creation of an endpoint, AWS will generate an AWS endpoint-specific DNS name so you can connect to the endpoint via its DNS name interface endpoints—private link.

The diagram below shows an example of an interface endpoint:



## Gateway Endpoints

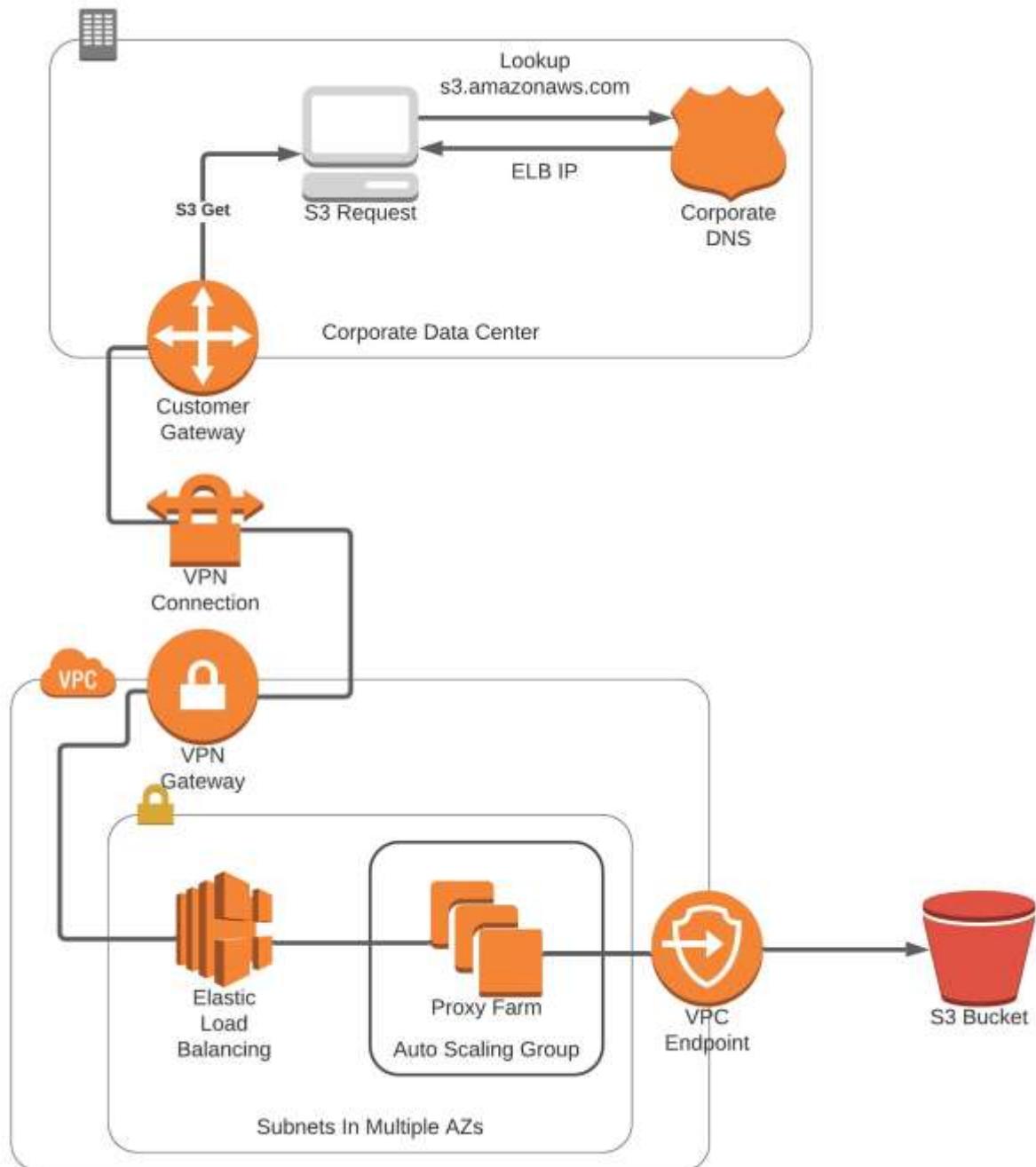
A gateway endpoint is a private endpoint that provides high-security access to an AWS service. It effectively places a route in the VPC's routing table for traffic destined to the AWS service. An example of a gateway endpoint is connecting Amazon S3 to DynamoDB.<sup>38</sup>

A gateway endpoint enables an organization to create multiple endpoints in a single VPC. Gateway endpoints allow private access from VPC to a public AWS service such as S3 or DynamoDB. For example, when an endpoint is created for S3, a prefix list and VPC endpoint are created for the VPC. The prefix list will adhere to this naming convention—pl-xxxxxxxx—and can be seen in the routing table. The routing table will show for route/prefix pl-xxxxxxxx. The route to the VPC endpoint will be in the routing table.

Gateway endpoints support IPv4 only. A gateway endpoint can only be linked to a single VPC. DNS resolution must be enabled in your VPC to ensure all the IP addresses maintained by AWS correctly resolved.

Gateway endpoints can be accessed over a direct connection or VPN location.

The graphic below shows how an organization can use a VPN connection to access an endpoint on the AWS cloud:



## Gateway Load Balancer Endpoints

A Gateway Load Balancer endpoint is a managed service used to deploy multiple network or security appliances. Gateway Load Balancer endpoints are used to securely exchange traffic across VPC boundaries. Gateway Load Balancer endpoints provide private connectivity between virtual appliances i.e., marketplace firewalls. You deploy the Gateway Load Balancer in the same VPC as you would use a network load balancer to load balance virtual appliances.

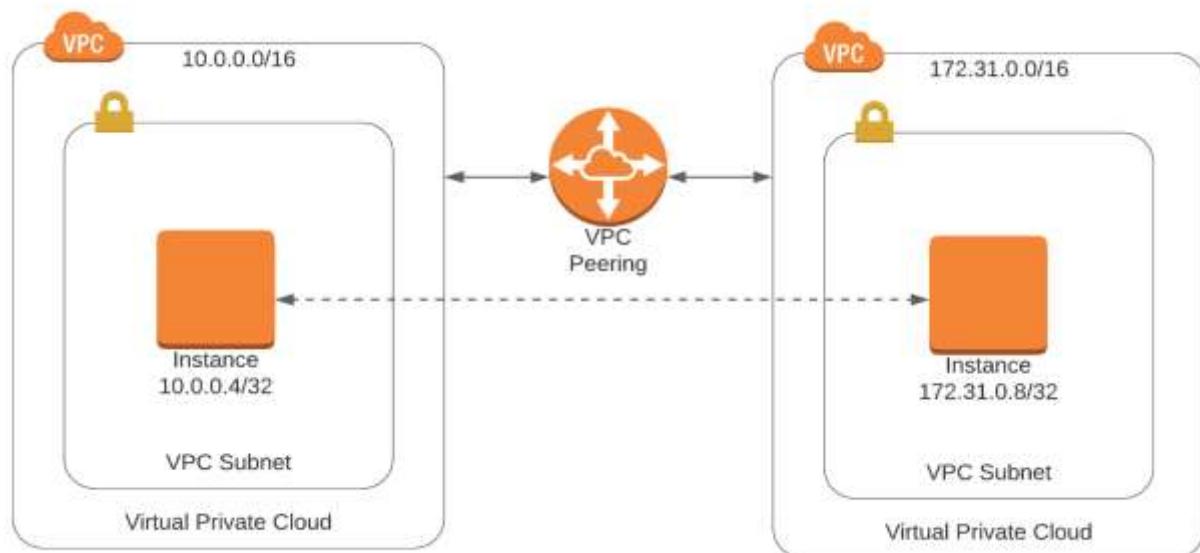
## VPC Peering

*VPC peering* is a technique to connect one or more VPCs without traversing the public internet. VPC peering also mitigates the need for direct or VPN connections between organizations that are hosted on the AWS network. VPC peering provides high-speed, high-availability connectivity by leveraging the AWS backbone for connectivity.<sup>39, 40</sup>

Some key things to know about VPC peering:

- VPC peering provides a nontransitive connection. This means that while VPC peering facilitates connectivity between VPCs, it does not facilitate routing traffic through a VPC to connect to another VPC.
- VPC peering uses the AWS network backbone, so there is a need for internet connections, internet gateways, NAT gateways, or public IP addresses.
- Inter-region VPC traffic is encrypted for data privacy.

The diagram below shows an example of a VPC peering on the AWS platform:

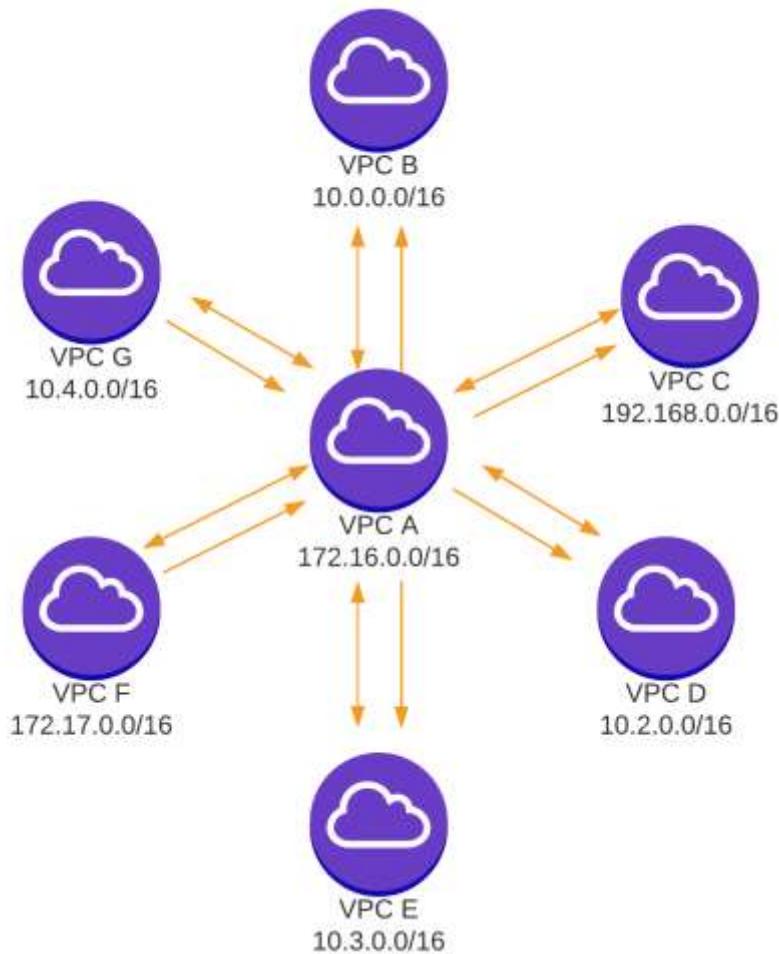


There are essentially two primary architectures for VPC peering: hub and spoke, and fully meshed.

### Hub and Spoke

In a hub-and-spoke environment, a hub is created with connections to all remote VPCs. This enables the hub to communicate with each remote VPC or spoke. However, since VPC peering is not transitive, VPCs will not be able to communicate with each other since communication is limited to hub-and-spoke VPCs.

The diagram below shows an example of hub-and-spoke VPC peering on the AWS platform:



## Fully Meshed

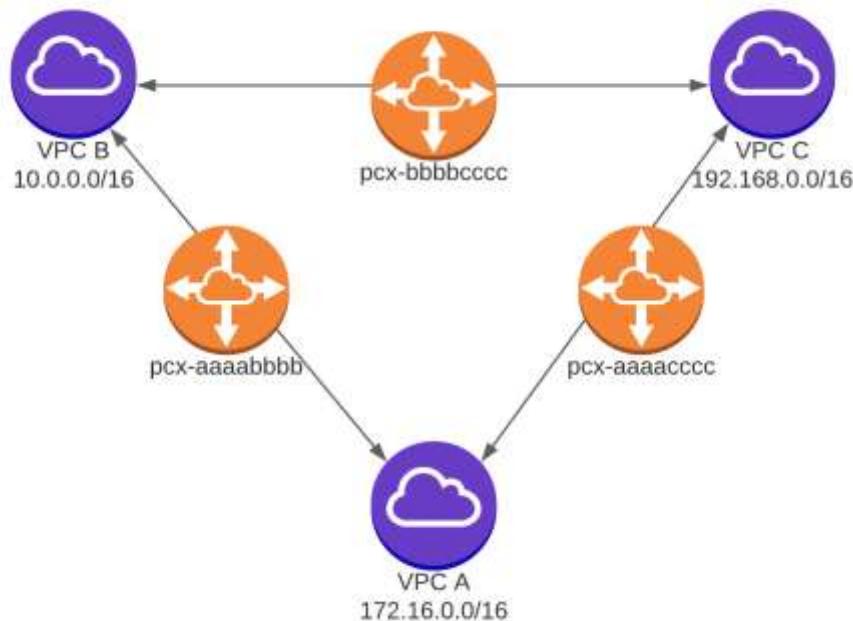
When all VPCs need to communicate with each other, every VPC will require peering to every other VPC. This is referred to as a *fully meshed environment*. Fully meshed architectures are required because VPC peering is not transitive. While fully meshed VPC peering works well, it is challenging from a scalability perspective. Scalability is challenged as the number of connections grows very rapidly as additional sites are added. The number of connections can be calculated with this formula:  $N*(N-1) / 2$ , where N is the number of nodes or VPCs.

Let's examine how fast VPC connections grow in the fully meshed environment.

- With three VPCs, six connections are required:
  - $3 * (3-1) / 2 = 6$  connections
- With ten VPCs, forty-five connections are required:
  - $10 * (10-1) / 2 = 45$  connections
- With twenty VPCs, only 190 connections are required:
  - $20 * (20-1) / 2 = 190$  connections
- With thirty VPCs, 435 connections are required:
  - $30 * (30-1) / 2 = 435$  connections

Fully meshed environments are excellent when a small number of VPCs require connectivity. However, fully meshed environments do not scale when a large number of VPCs require connectivity with each other. A solution to help create a more scalable VPC peering environment is with AWS CloudHub and Transit Gateway.

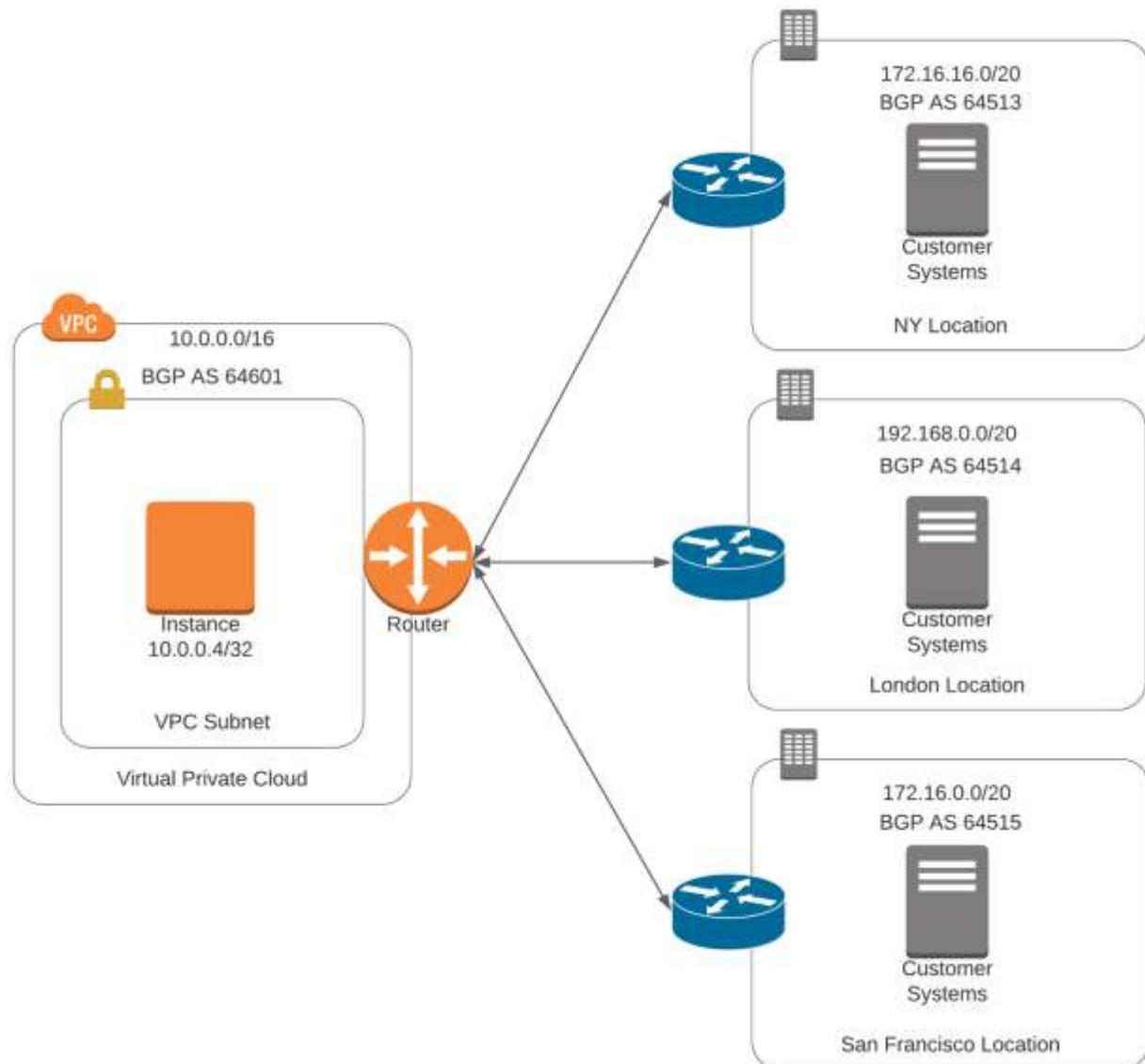
The diagram below shows an example of fully meshed VPC peering on the AWS platform:



## AWS CloudHub

When it's necessary to establish connectivity between a VPC and a large number of remote sites, CloudHub simplifies the process of VPC peering. CloudHub enables an organization to have transitive VPC connections in a hub-and-spoke environment. CloudHub uses BGP, specifically eBGP, to connect and share routing information across VPCs. Routing information is propagated via BGP, which provides network reachability to all remote locations or connected VPCs. Since all remote locations have knowledge of all subnets across organizations and VPCs, full communication across VPCs is established. Since BGP is used for route sharing, connectivity can be limited to only desired resources by using route filters, access lists, and firewalls.<sup>41</sup>

The diagram below shows an example of using CloudHub to simplify VPC peering on the AWS platform:



## Transit Gateway

Transit Gateway is a service designed to facilitate communication between multiple VPCs. It is very similar to CloudHub, but CloudHub only supports VPNs. Since the default behavior of the AWS VPC peering is nontransitive, Transit Gateway provides the ability to share routing across multiple VPCs.

Transit Gateway can share routes through both VPN connections and Direct Connect. Transit Gateway supports VPC peering across availability zones and regions. Transit Gateway ensures that all traffic traverses the AWS backbone and not the internet. Additionally, to further enhance security, all data remains encrypted, providing an additional layer of security. Transit Gateway supports near-unlimited scalability, allowing an organization to scale without having to worry about the capabilities of their Transit Gateway(s).

External resources can be connected to Transit Gateway through attachments. Transit Gateway supports the following **attachments**.

A connection to one or more:

1. VPCs
2. VPN connections
3. AWS Direct Connect gateways
4. AWS Transit Gateway Connect or third-party software-defined WAN appliances (SD-WAN)
5. Connections to other Transit Gateway(s) (Transit Gateway peering)

When sending a message across multiple VPCs, the VPCs must use the same Transit Gateway. This is because transit gateways do not share flow state information.

## **VPC Peering vs. PrivateLink**

Both VPC peering and PrivateLink can be used to connect two separate organizations (VPCs). However, there are some significant differences between VPC peering and PrivateLink. Knowing which of these services to choose is essential to deliver the best solution.

### **When VPC Peering Is Best**

VPC peering is best when an organization wants complete connectivity between organizations. VPC peering allows access to all the services an organization allows. VPC peering requires non-overlapping address space. So, if there is full communication between VPCs and non-overlapping IP address space, VPC peering is the best option.

## **When PrivateLink Is Best**

PrivateLink is much more scalable than VPC peering. VPC peering has a limitation of 125 connections. Therefore, if many connections are needed PrivateLink, is the best option. PrivateLink allows overlapping address space between organizations, as PrivateLink provides NAT by default. PrivateLink is inherently more secure than VPC peering. PrivateLink allows access to only one service. To further enhance security, PrivateLink connections are unidirectional, meaning if you want connections in and out of your VPC, you must enable bi-directional communication.

## **Enhancing the VPC Security Posture**

AWS provides numerous options to enhance the security of a VPC. Two fundamental options to enhance the security of a VPC include network access control lists and security groups.

### **Network Access Control List (NACL)**

The NACL is a means to enhance security by keeping unwanted traffic out of a subnet. NACLs block or permit traffic in a manner similar to an access control list on a router or a stateless firewall.<sup>42</sup> There are some key things to understand about NACLs in order to use them effectively:

- Rules are created to determine what traffic is allowed or denied.
- The accept or deny rules must be written in a specific order.
- The rules are processed in order. Therefore, if you explicitly deny something, it won't be possible to permit something that is denied by a previous statement.
- The order is determined by the number attached to the rule statement.
- Lower numbers in the rule statement are processed prior to higher numbers.
- Network ACLs have an implicit deny, so you must specify any traffic that you want to allow, or it will all be blocked when you add a network ACL. The allowed traffic must be sequenced before any rule that denies the desired traffic.
- NACLs are written in both inbound and outbound rules. Inbound rules determine what's allowed into a subnet; outbound rules determine what traffic is permitted to leave a subnet.
- Remember, NACLs are stateless, so inbound rules and outbound rules need to match. If SSL is allowed in, then it must be allowed back to the requester. NACLs are stateless; therefore, there is no means to allow return traffic like a stateful firewall.

The order of the rules in the NACL are important. Below are two examples of network ACLs, one with the proper technique, and the other with an incorrect technique.

## Proper NACL Structure – Do This!

### Inbound

Rule 110 Allow TCP Port 80 Source any

### Outbound

Rule 110 Allow TCP Port 80 Destination any

## Improper NACL Structure – Don't Do This!

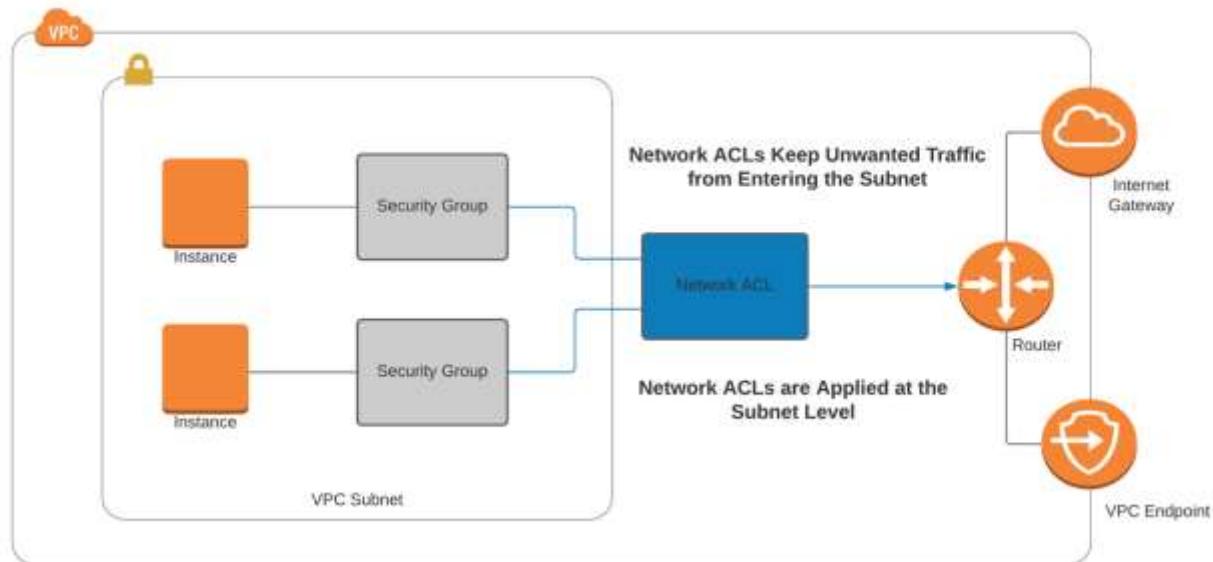
### Inbound

Rule 100 – Deny all traffic

Rule 110 Allow TCP Port 80 Source any

Note that in the above example with improper technique, all traffic is blocked by the first rule in the NACL; therefore all traffic is blocked. This reinforces the need to use the correct order in NACL rule statements.

The diagram below shows how network ACLs keep unwanted traffic out of the subnet:

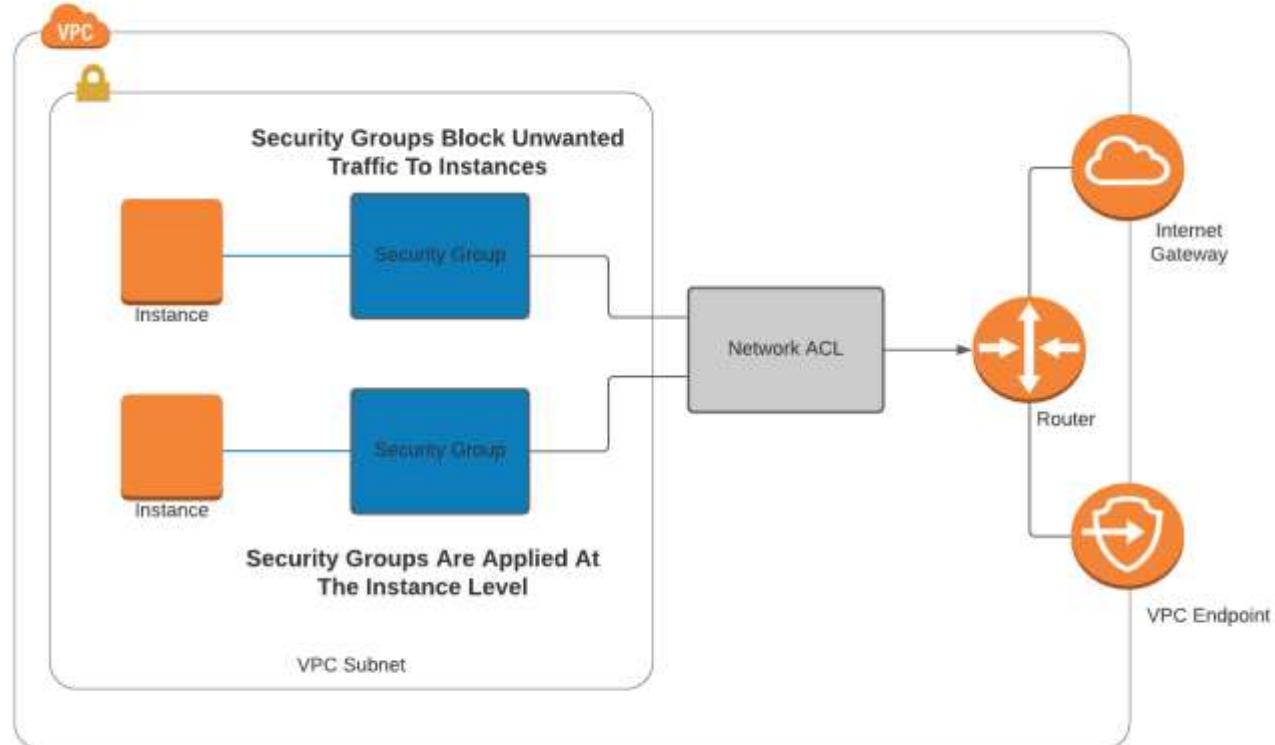


## Security Groups

A security group is essentially a stateful access control list (like a firewall) that is applied to a computing instance or AWS service. This is different than an NACL, which is applied to a subnet. Realistically speaking, a good security architecture will include NACLs at the subnet and security groups attached to the server.

Security groups have an implicit deny, so only permit statements are required. All that is necessary is configuring the permit statements to allow the desired traffic into the server. Since security groups are stateful, it is only necessary to permit inbound traffic, as outbound return traffic will be permitted. Security groups evaluate all rules prior to permitting or denying traffic, so the order of rules in a security group is not as critical as with NACLs.<sup>43</sup>

The diagram below shows how security groups keep unwanted traffic out of the instance:



## **Chapter 7 - AWS Network Performance Optimizations**

There are three components of AWS networking that can have a substantial impact on performance and availability: *placement groups*, *Route 53*, and *load balancers*.

### **Placement Groups**

*Placement groups* are simply where an organization places their equipment, such as servers (EC2 instances). Where an organization places its equipment can have a profound effect on performance and availability. There are three options for placement groups: *clustered*, *partitioned*, and *spread groups*.<sup>44</sup>

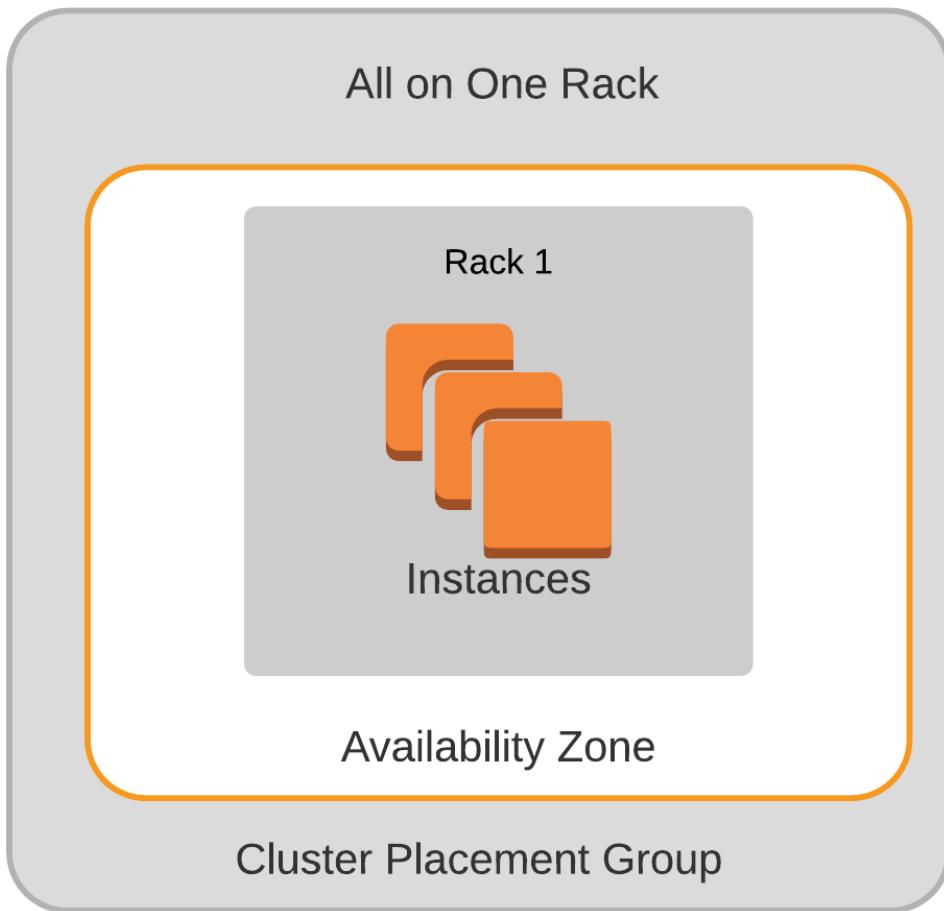
#### **Clustered Placement Groups**

A *clustered placement group* offers the best performance at the expense of availability. A clustered placement group means placing an organization's servers extremely close to each other to reduce latency and optimize performance.<sup>45</sup> Some key tenants of placement groups are as follows:

- Proximity – Instances are very close in physical proximity.
- Instances are often in the same rack.
- Instances are often on the same physical server.
- Since devices are often on the same rack, on the same network switch, and on the same server, this offers the absolute best network performance

Clustered placement groups have a major drawback when it comes to availability. Since everything is close together, often on the same server, rack, switch, and power source, there are many single points of failure when compared with other architectures. This architecture is perfect for applications that are not tolerant of latency.

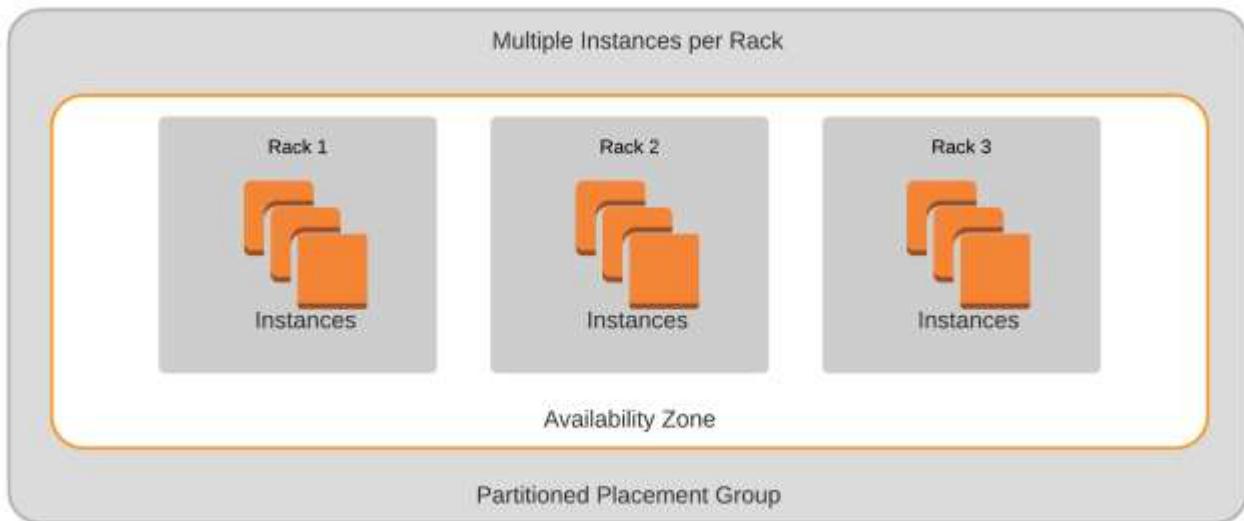
The diagram below shows an example of a clustered placement group:



### Partitioned Placement Groups

Partitioned placement groups provide a high level of performance, low latency, and higher availability than a clustered placement group. Partitioned placement groups place all components in a single AZ (data center). However, the instances are grouped into partitions and spread across data center racks. Spreading the load across a data center minimizes the risk that a single server, power outage, or network switch failure will bring the entire system down.<sup>46</sup>

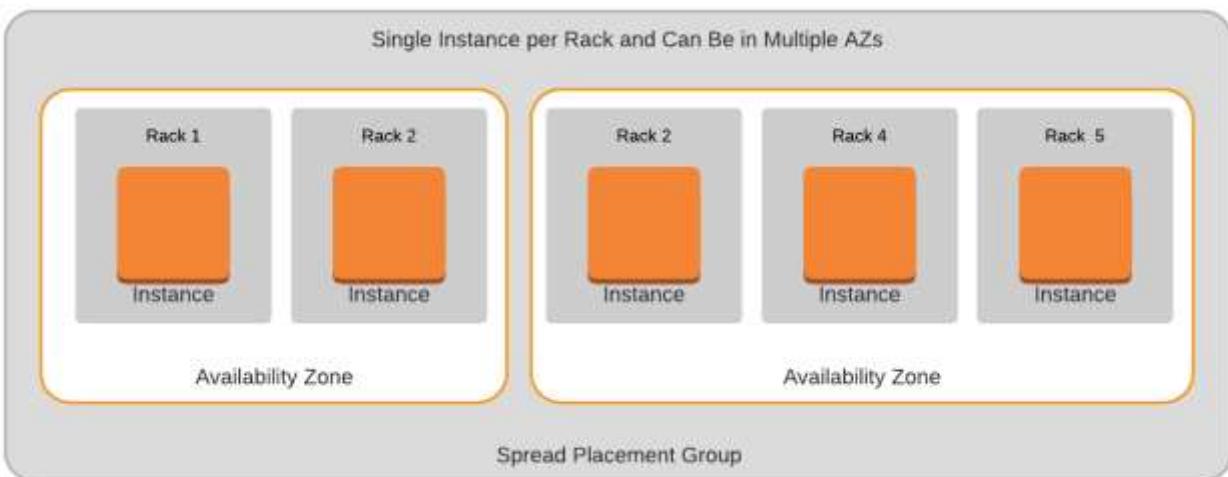
The diagram below shows an example of a partitioned placement group:



## Spread Placement Groups

A spread placement group is optimal when a high-availability design is required. In spread placement groups, instances are spread across hardware, racks, physical servers, power distribution units, and other system components. Groups can be spread across multiple availability zones. This design offers high availability, but with higher latency and lower network performance than clustered and partitioned placement groups.<sup>47</sup>

The diagram below shows an example of a spread placement group:



## **Elastic Fabric Adapter**

The Amazon Elastic Fabric Adapter (EFA) is a type of network interface offering high performance and throughput. The Elastic Network Adapter (ENA) was designed for high-performance computing (HPC) applications.

It is essentially a newer and faster form of a network interface, an interface that has been designed to support speeds of 100 Gbps.

The Elastic Fabric Adapter is a custom-built interface that bypasses the operating system to enhance the performance of inter-instance communications, which is critical to scaling of high-performance applications.

The Elastic Network Adapter can be attached to EC2 instances to accelerate high-performance computing and machine learning applications.

The Elastic Fabric Adapter works with the most commonly used interfaces, APIs, and libraries for inter-node communications. The Elastic Fabric Adapter can be used with little or no modifications in the organization's applications.

## **Single-Root Input/Output Virtualization (SR-IOV)**

Servers in the cloud are virtual machines. That means virtualized network cards, video cards, and other virtual hardware components. Under normal circumstances this works fine, but when an organization requires high performance, virtual hardware is often not enough.

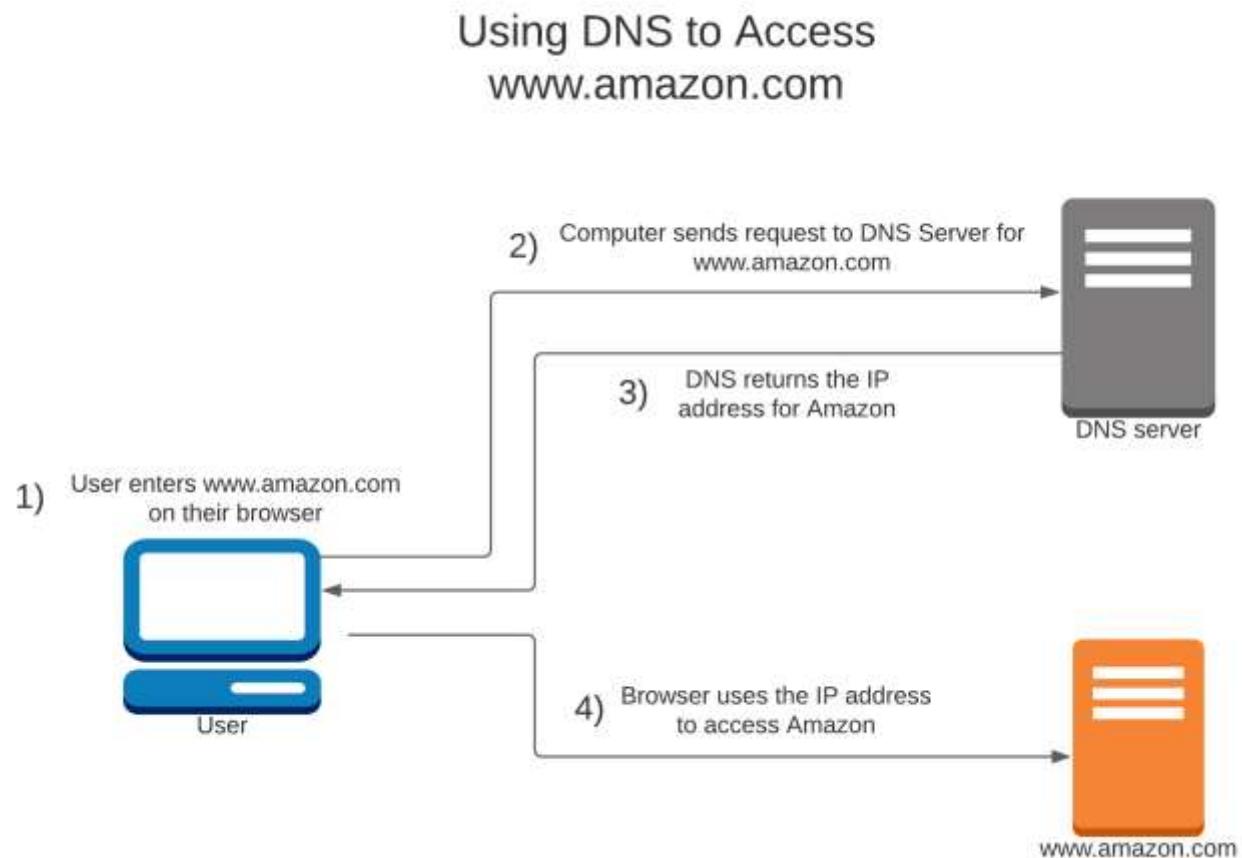
Often an organization needs the performance of actual hardware, a physical graphics processing unit or a physical network adapter. Enter the world of SR-IOV. *Single root I/O virtualization* is placing a physical network card in a virtual machine, essentially passing a PCIE card into the virtual machine.

By passing the physical network card into the virtual machine, the virtual machine will have full hardware performance. SR-IOV is a way to combine the flexibility of virtualization with bare metal server networking performance.

## **Amazon Route 53**

Every device on the network requires an IP address. While it's possible to connect directly to every system using just the device's IP address, it's infeasible to remember every system's IP address. Imagine remembering the IP address of every website on the internet. The solution that was developed to this challenge was the Domain Name System (DNS). The DNS maps a name with an address. Amazon Route 53 is Amazon's implementation of the DNS.<sup>48</sup>

The diagram below shows an example of DNS mapping a name to an IP address:



For example, it's very easy to remember [www.amazon.com](http://www.amazon.com). When we enter [www.amazon.com](http://www.amazon.com), our systems ask the DNS server what the IP address is for this website. Then our request is forwarded to the IP address of the Amazon website. You can see the name to address mapping on any UNIX/Linux and Windows systems with the nslookup command:

```
mgibbs@Michaels-Mac-Pro ~ % nslookup www.amazon.com
Server:     8.8.8
Address:   8.8.8#53

Non-authoritative answer:
www.amazon.com canonical name = tp.47cf2c8c9-frontier.amazon.com.
tp.47cf2c8c9-frontier.amazon.com canonical name = www.amazon.com.edgekey.net.
www.amazon.com.edgekey.net canonical name = e15316.e22.akamaiedge.net.
Name:      e15316.e22.akamaiedge.net
Address: 23.75.198.60
```

Some key points to know about AWS Route 53:

- Route 53 provides name to IP address mappings just like any other DNS platform.
- Route 53 is a high-availability platform for DNS services.
- Route 53 is highly scalable platform for DNS services and server health checks.
- AWS uses anycast services, for which there are multiple servers with the same address placed over the internet.
- Anycast provides extremely high availability and low latency. As a host, it will connect to the closest DNS server based upon its IP address. If a DNS server were to become unavailable, the host will connect to the next closest anycast address of the DNS server
- Route 53 supports most of the available DNS record types.
- Route 53 uses TCP and UDP port 53.
- Route 53 works with health checks and can be used to create a high-availability solution.
- Route 53 supports most DNS record types.
- Route 53 has numerous options to optimize a web-based environment.

AWS supports the following DNS record types:

<b>Amazon Route 53 Supported Record Types</b>
A (address record)
AAAA (IPv6 address record)
CNAME (canonical name record)
CAA (certification authority authorization)
MX (mail exchange record)
NAPTR (name authority pointer record)
NS (name server record)
SOA (start of authority record)
SPF (sender policy framework)
SRV (service locator)
TXT (text record)

While a deep understanding of DNS is beyond the scope of this book, there are some DNS records that are so foundational we recommend learning them. These key record types are as follows:

### A – Record

- The most fundamental DNS record.
- A record mapping a name to an IP address.
- The IPv6 equivalent is an AAAA record.

### CNAME – Record

- This is a record that maps a domain to another domain.
- It can map to another CNAME Record or an A record.
- CNAME records effectively redirect a request for one domain to another domain.
- i.e., map [www.a.com](http://www.a.com) to [www.b.com](http://www.b.com).

### NS – Record

- Identifies the DNS servers that are responsible for your DNS zone.
- These authoritative name servers propagate an organization's official DNS information to the DNS servers across the internet.
- NS records can be several entities.

## MX – Record

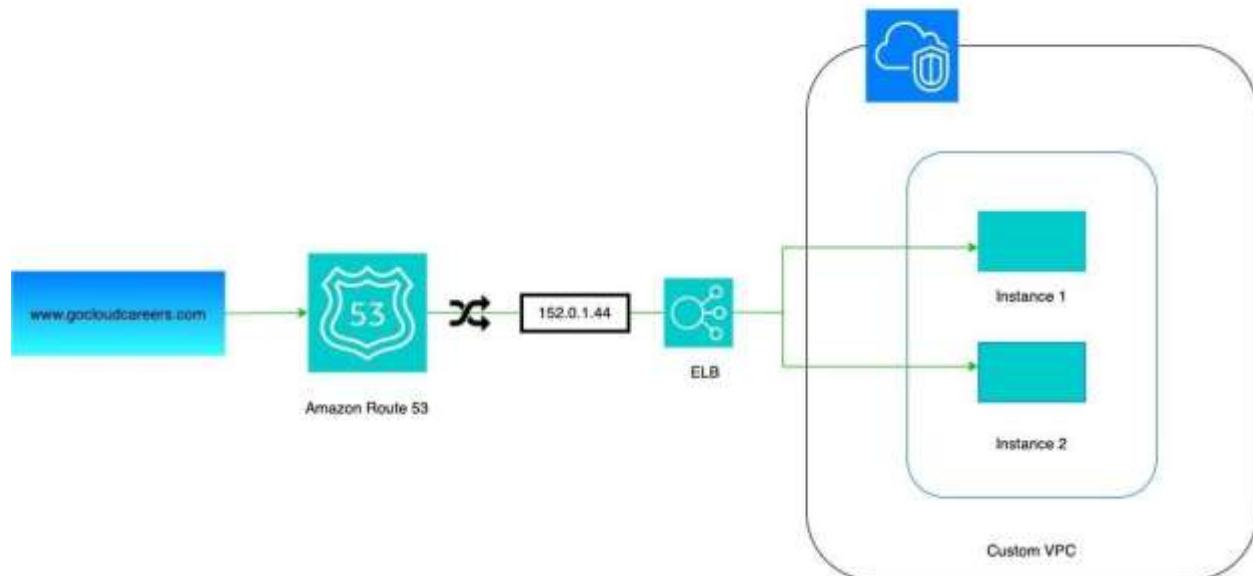
- An MX record specifies which mail servers can accept mail for your domain.
- MX records are necessary to be able to receive email.

## Policy Based Routing Options

AWS has several policy-based routing options for Route 53. These policies can be as simple as mapping a name to an IP address to finding the sever with the lowest latency. The AWS Route 53 policies and their functions are as follows:

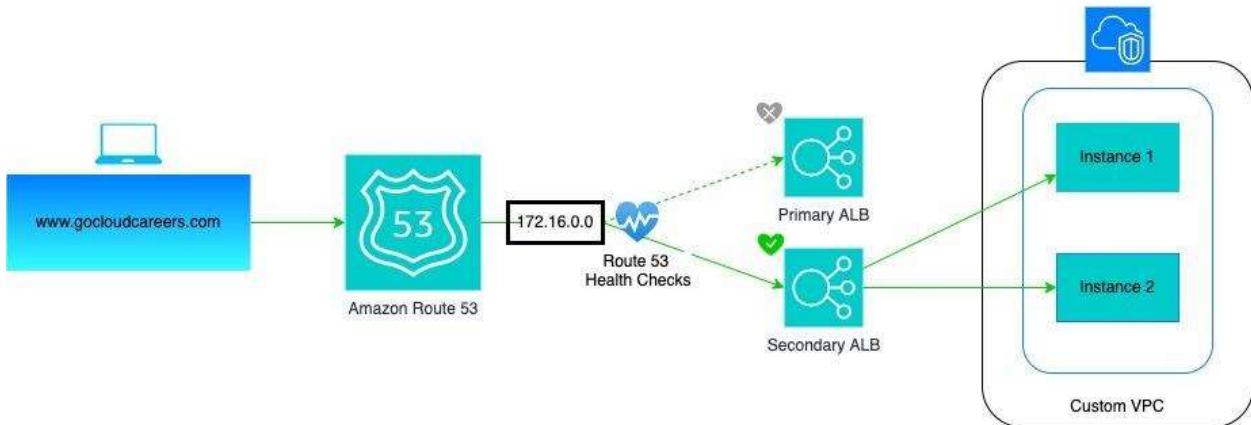
**Simple routing** – Basic DNS that maps a domain name to a single location. This is the default policy, which is perfect with a single server for a domain. An example of this policy can be seen below:

**Simple Routing Policy**

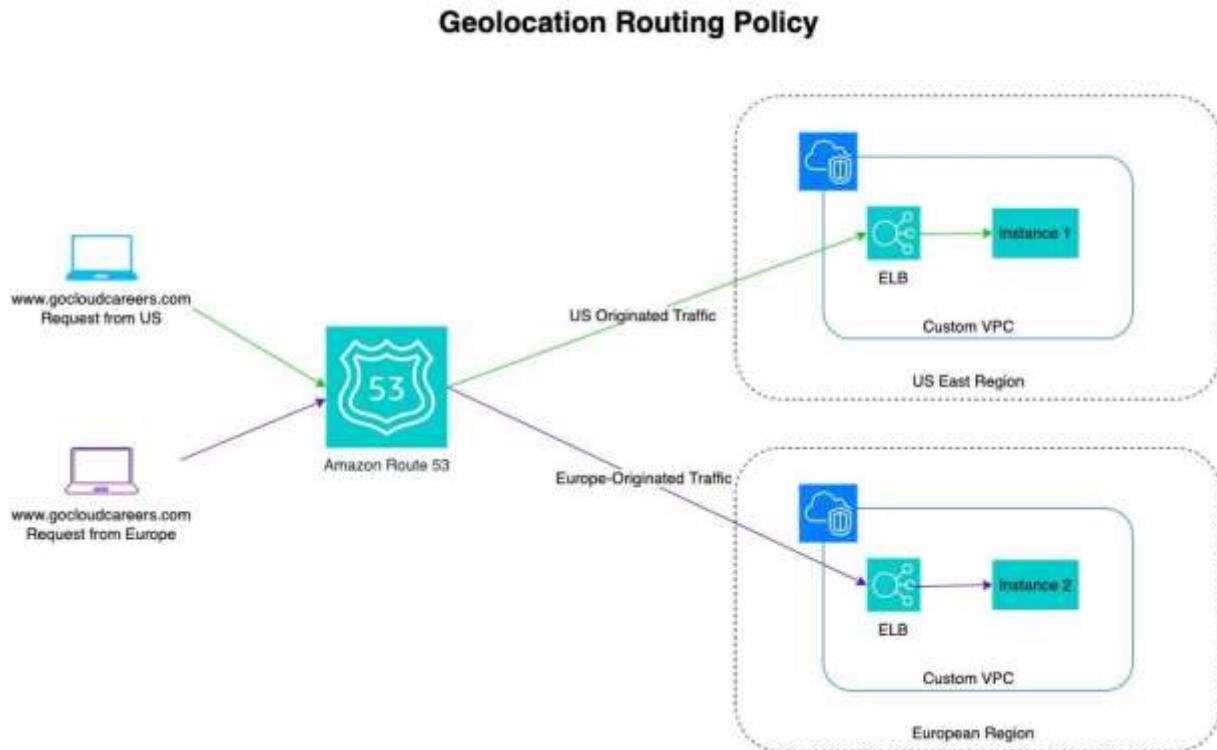


**Failover routing** – Sends the traffic to the main server. If that is not available, it sends traffic to a backup server. An example of this policy can be seen below:

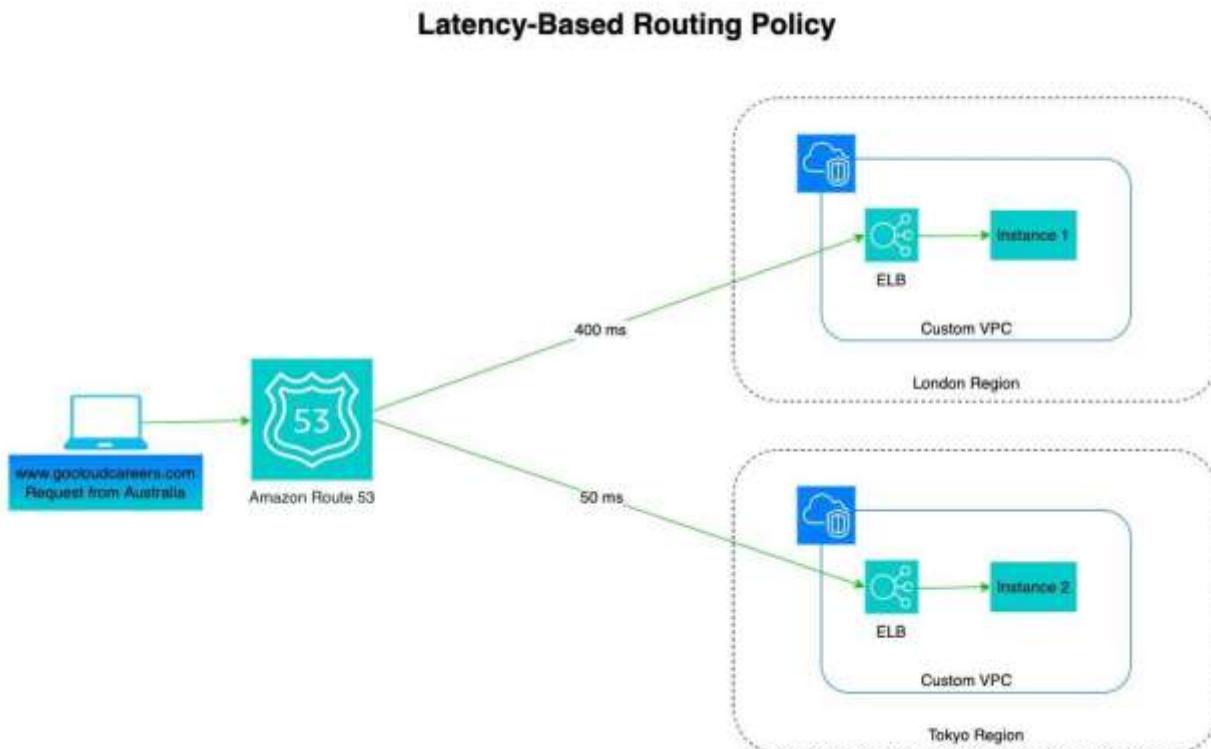
### Failover Routing Policy



**Geolocation routing** – Used when there are servers in several regions. To optimize performance, geolocation routing will look at the source IP address of the user (which will ultimately provide their location) and route them to the closest region so they have the best performance. An example of this policy can be seen below:

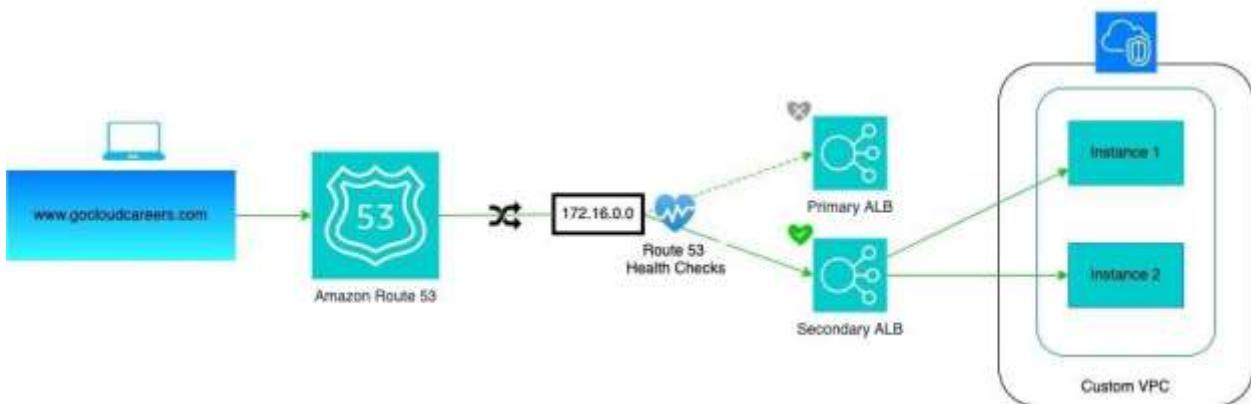


**Latency-based routing** – Will send to the server with the lowest latency to optimize performance. Ideal when the website is in multiple availability zones or regions. It provides the optimal experience to the user. An example of this policy can be seen below:



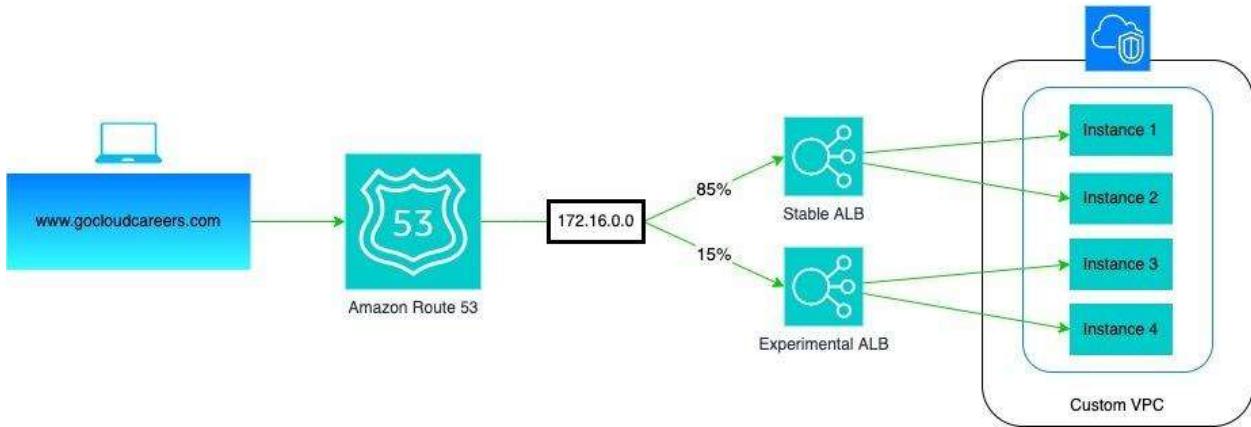
**Multivalue answer** – Route to any available server. An example of this policy can be seen below:

### Multi-value Answer Routing



**Weighted Routing** – Provides a means to share traffic between servers at a percentage you chose, i.e., 75 percent to server *a*, and 25 percent to server *b*. It is a great option to test an application’s functionality. Think of a CI/CD pipeline and blue-green deployments. Send most of the traffic to the old website for testing and a percentage to the new website. When the new site is tested, move the traffic to the new server. An example of this policy can be seen below:

### Weighted Routing Policy



**Geoproximity routing** – Used when an organization has servers in multiple availability zones. Geoproximity routing will send the requestor to the closest availability zone. There is also the ability to route traffic in a specific manner by using bias. Bias makes the geographic region larger or smaller.

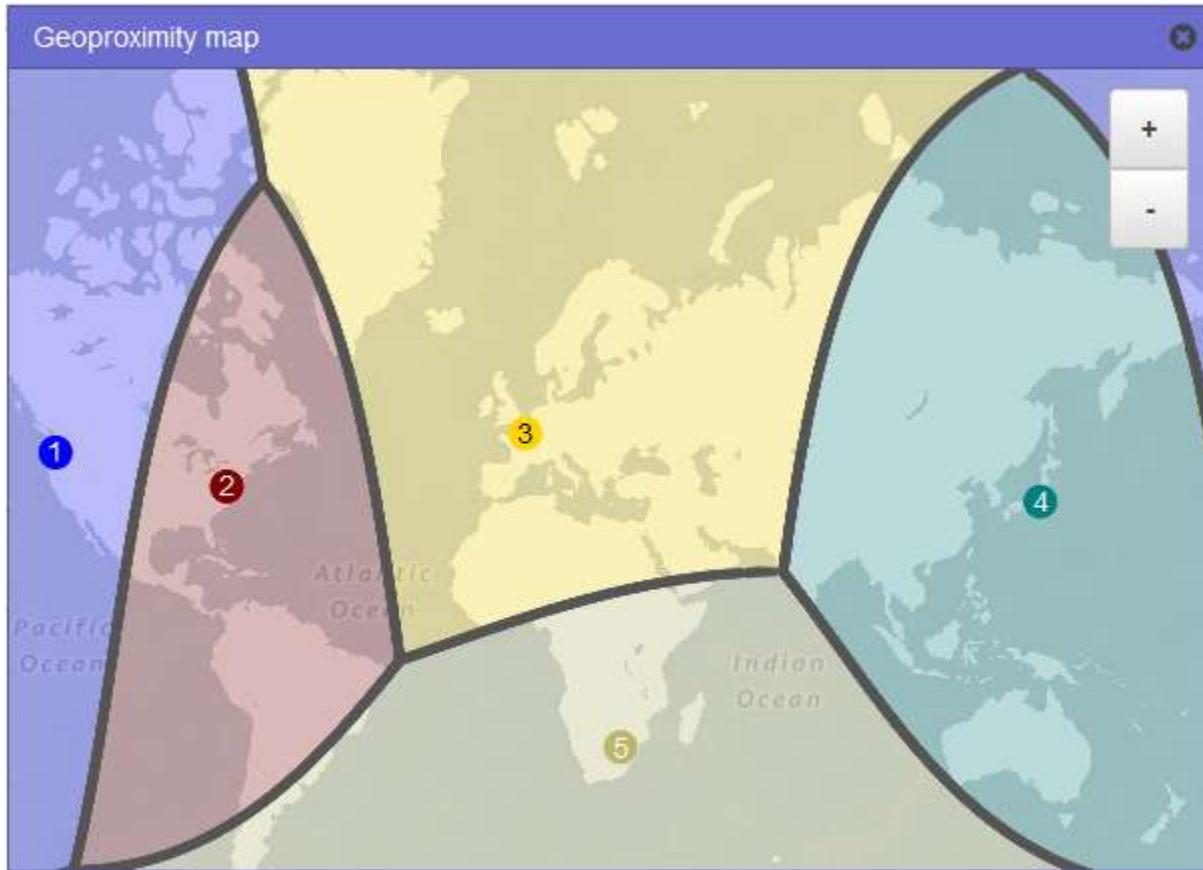


Image source <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html#routing-policy-geoproximity>

## Load Balancers

*Load balancers* are network devices that facilitate load sharing across servers. Load balancers can help greatly with scalability by allowing the application to be deployed across multiple servers. Load balancers can also increase availability by allowing multiple servers to be used simultaneously, removing single points of failure. Additionally, load balancers can use health checks to remove unhealthy servers from being used, which further enhances application availability.<sup>49</sup>

Since load balancers facilitate load sharing among servers, they can facilitate scaling out applications across multiple servers. Scaling out servers substantially increases performance. There are some key concepts when working with AWS load balancers. These key concepts are listeners, targets, target groups, and sticky sessions.

## Listeners

Load balancers have a *listener* which waits for connection requests. Application load balancers look at http, https requests, and ports 1–65535. Network load balancers look at protocol and port numbers, specifically TCP, UDP, TLS, and TCP\_UDP ports 1–65535.

## Targets

Load balancers distribute their traffic to a target. A *target* is a server, a container, or whatever is providing the service. Targets can be an EC2 instance or an IP address. When the target is an IP address, it must be from the private address space (RFC 1918) or the shared address space (RFC 6598):

- 10.0.0.0/8 (RFC 1918)
- 172.16.0.0/12 (RFC 1918)
- 192.168.0.0/16 (RFC 1918)
- 100.64.0.0/10 (RFC 6598) – shared address space

## Target Groups

*Target groups* enable the systems administrator to group multiple targets together. For example, a target group can be used to group multiple EC2 instances together.

## Sticky Sessions

A *sticky session* is keeping the session open between the client and a single server. By default, AWS network load balancers will keep a session active between the client and a single server. This is based on the way a network load balancer functions.

Application load balancers do not maintain a session between the client and server. However, the application load balancer can be set up to maintain a session between the client and the server. This occurs with enabling the application load balancer to use a cookie to track and manage the connection.

## Types of Load Balancers

There are essentially four kinds of load balancers: classic load balancers, network load balancers, application load balancers, and gateway load balancers.

## Network Load Balancers

*Network load balancers* operate at the transport layer (layer 4) of the OSI model. Network load balancers work with either the TCP or UDP transport protocol.

## **Application Load Balancers**

*Application load balancers* operate at the application layer (layer 7) of the OSI model. Application load balancers typically work with the http or https protocol.<sup>50</sup>

The AWS implementation of a load balancer is called an Elastic Load Balancer. There are four options for Elastic Load Balancers. These options are Application Load Balancers, Network Load Balancers, Gateway Load Balancers, and Classic Load Balancers. There are some key things to know about Elastic Load Balancers:

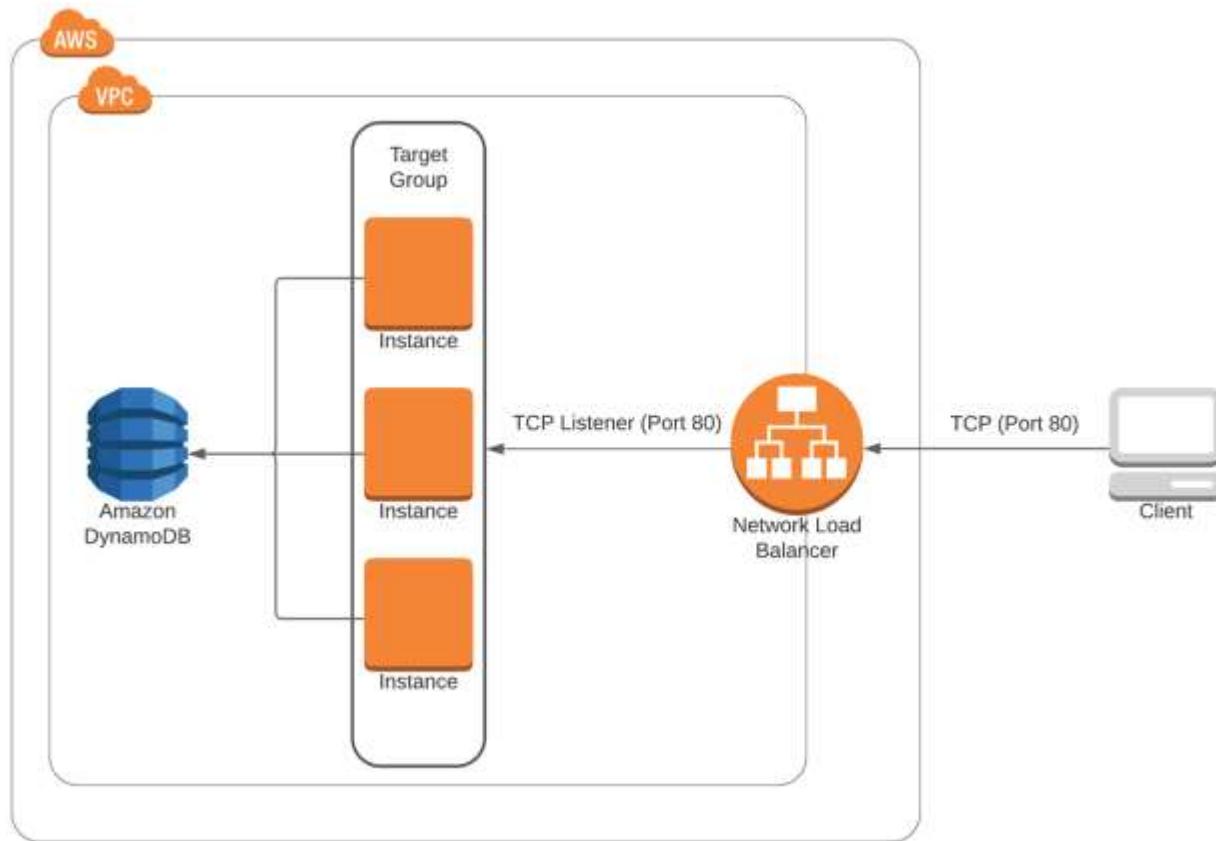
- Elastic Load Balancers automatically distribute traffic to multiple targets (i.e., EC2 instances, Lambda).
- Elastic Load Balancers can facilitate autoscaling. Load balancers can assist in the autoscaling of more instances if necessary to meet the application's performance needs.
- Elastic Load Balancers use an IP address, and if autoscaling occurs, multiple IP addresses will be used.
  - Plan your addressing scheme carefully so you don't run out of addresses.
  - Remember, AWS reserves the first four IP addresses (network and first three usable addresses) and the last (broadcast) IP address of the subnet.
- Elastic Load Balancers can load balance across availability zones.
- Elastic Load Balancers support health checks so nonfunctioning servers can be removed from use.
- Elastic Load Balancers can terminate SSL connections, which can reduce the load on servers.

## **Elastic Load Balancer – Network Load Balancer**

The Network Load Balancer operates at the transport layer (layer 4) of the OSI model. Network load balancers work with either the TCP or UDP transport protocol. Network Load Balancers perform routing based upon the destination port of the traffic they receive. Network Load Balancers are extremely fast and are an excellent option when ultimate speed is needed. Network Load Balancers can handle millions of requests per second and excel with rapidly changing traffic patterns.

The Elastic Load Balancer network version is stateful. This means that once a connection is established between the host and the server, the connection is maintained until the session is completed. The Network Load Balancer keeps state by maintaining sticky sessions that have a mapping of the source and destination of the connections.<sup>51</sup>

The diagram below shows an example of an Elastic Network Load Balancer:



### Elastic Load Balancer – Application Load Balancer

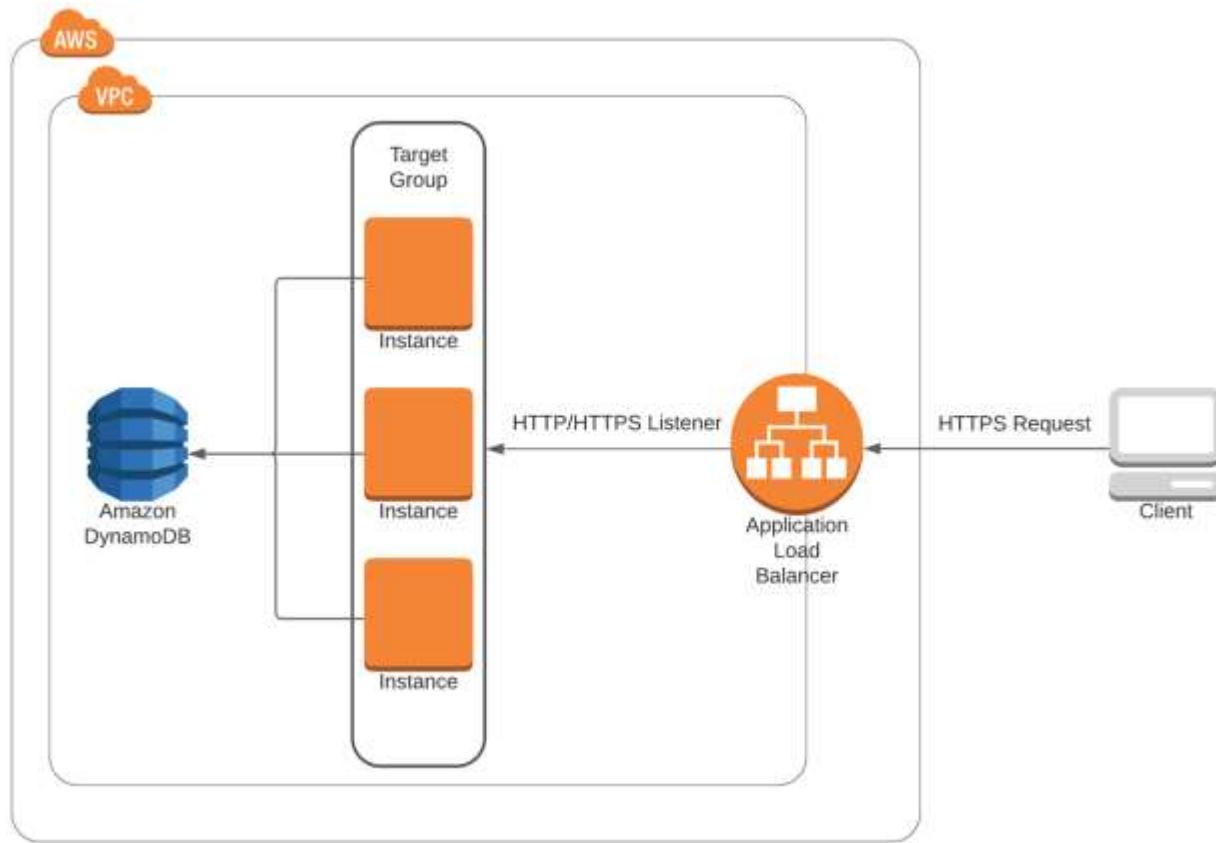
The Application Load Balancer operate at the application layer (layer 7) of the OSI model. Application load balancers typically work with the http or https protocol.<sup>50</sup> The application ELB can route traffic based upon many options and is ideal for web traffic. Application Load Balancers are an excellent means to load balance requests to microservices and container-based applications. Application ELBs can even load balance between a VPC and an on-premises data center.

Application Load Balancers are stateful. Once a connection is established between the host and the server, the connection is maintained until the session is completed. The Application Load Balancer keeps state by maintaining sticky sessions that have a mapping of the source and destination addresses of the connections.

Application ELBs make forwarding decisions based upon the following factors:

- Domain name
- Path provided by the URL
- Elements in the http, https header
- Http method-based routing (i.e., put or get)
- Source address

The diagram below shows an example of an Elastic Application Load Balancer:



### Gateway Load Balancers

The AWS Gateway Load Balancer can be used to load balance third-party virtual appliances. Examples would include marketplace firewall solutions. The Gateway Load Balancer is used to provide redundancy—as there is no redundancy in a virtual appliance. Gateway Load Balancers are designed to replace the Network Load Balancer function when load balancing network devices. Gateway Load Balancers provide both layer 3 and layer 4 load balancing functions.

- Provide high availability by removing single points of failure with third-party appliances
- Can help increase network performance

## **Classic Load Balancers**

The AWS Classic Load Balancer can be network based or application based. This is a legacy platform and can work with both EC2-classic and VPCs.<sup>52</sup> Classic Load Balancers have been deprecated and will be retired by August 20, 2022. Classic Load Balancers are characterized by the following:

- Have autoscaling capabilities
- Can support single or multiple availability zones
- Can terminate SSL connections to reduce server load
- Are stateful by using sticky sessions
- Provide logs to analyze traffic flows and can be used with CloudTrail for auditing

## Chapter 8 - Security

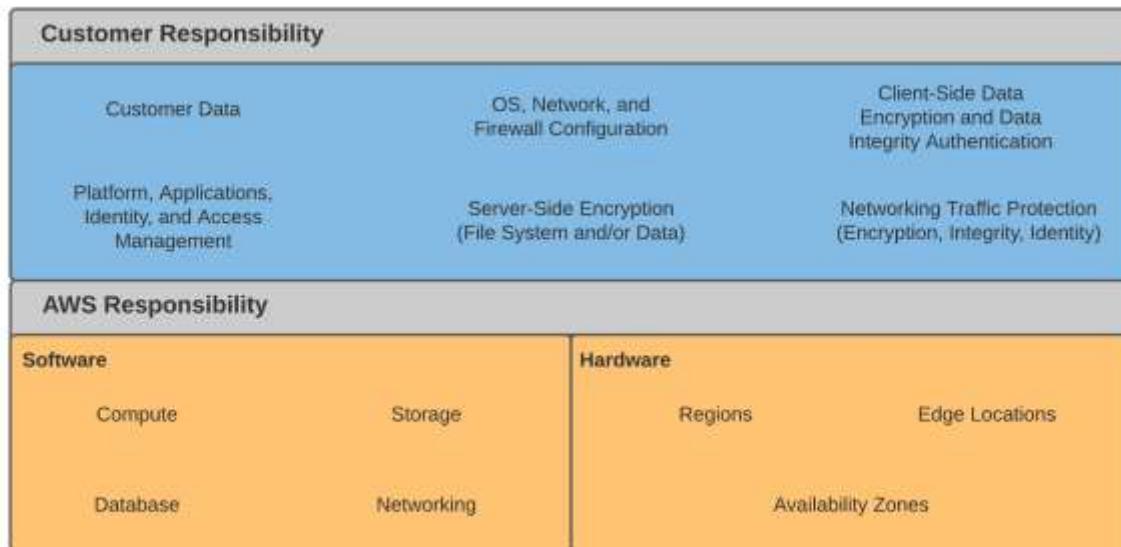
Security is a critical component for the success of any organization. A complete guide to full security architectures is beyond the scope of this book and should be explored by any organization that would face serious consequences if a security breach occurred. This chapter breaks down security into the following components:

- Who is responsible for what parts of the VPC
- Principle of least privilege
- Industry compliance
- Identity and access management
- Multiple account strategies
- Network ACLs, security groups, WAF
- Intrusion detection and prevention
- Distributed denial of service attacks and prevention
- Service catalogs
- Systems manager parameter store

### AWS Shared Security Model

While outsourcing the data center to the cloud can help with costs, agility, scalability, and even security, there is still a substantial amount of security that must be performed by the customer. When outsourcing an organization's data center to AWS, security and compliance responsibilities are shared between AWS and the customer. This is called the *shared security model*. In the shared security model, AWS maintains the security of the cloud, and the customer maintains the security of their VPC.<sup>53</sup>

The diagram below shows the AWS shared responsibility model:



## **Securing the Cloud**

AWS manages keeping the cloud secure. Keeping the cloud secure is really about managing the following functions:

- Principle of least privilege – The principle of least privilege is a best practice for limiting who from AWS can manage assets in the cloud by granting only the permissions required to complete a task.
- Physical security – Keeping the facility locked, keeping unauthorized users out of the AWS data centers.
- Security of the cloud – Keeping the cloud secure (firewalls, system patching, routing, IDS/IPS, change management)
- Follow industry regulations (accepted international security standards such as ISO 27001 and the National Institute of Standards and Technology (NIST), as Europe's General Data Protection Regulation (GDPR) and HIPAA).
- Keeping all AWS applications secure with patching and maintaining the underlying components of serverless applications offered by AWS.
- Keeping the AWS network secure with secure routing, VLANs, route filtering, firewalls, and intrusion prevention and detection (IDS/IPS).

## **Securing the VPC**

The customer is responsible for securing all aspects of their VPC. This means the customer is responsible for the following security components:

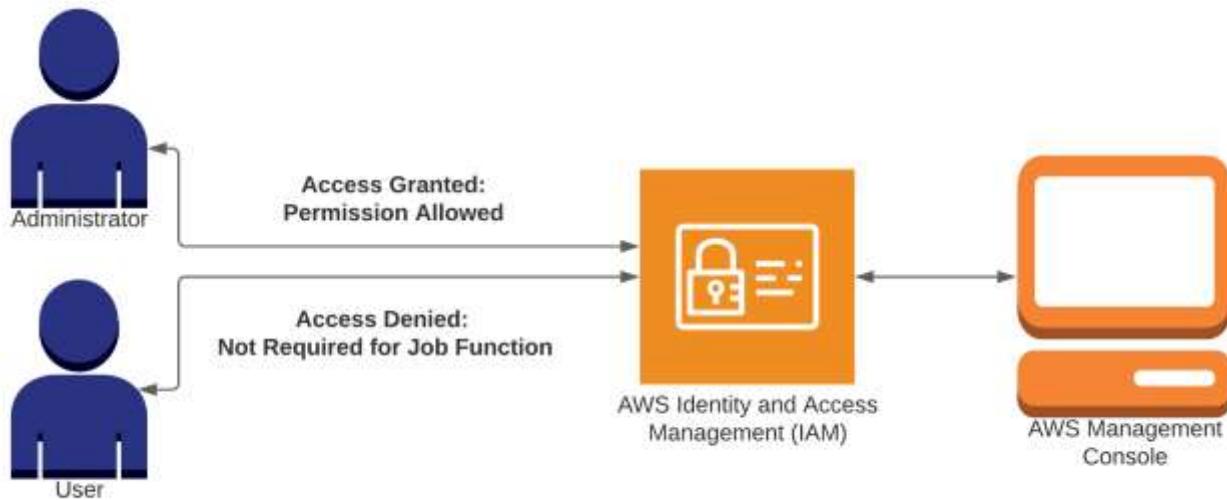
- Identity and access management – Determine who is allowed in the VPC and define their functions.
- Principle of least privilege – Grant the least privileges necessary for employees and partners to perform their functions effectively.
- Data security – Manage encryption.
- Maintenance of customer-designed applications.
- Management of the VPC routing tables.
- Managing traffic allowed into the VPC – Firewalls, NACLs, security groups.
- Maintenance of the operating systems and applications stored on EC2 compute instances.
- Physical security – Keep the devices that connect to the cloud secure from unauthorized users.

## **Principle of Least Privilege**

One of the most critical components of security is the principle of least privilege. The *principle of least privilege* is really about making sure individuals and systems using the cloud can access only the functions necessary to perform their role effectively. Granting more than the minimal

level of privileges can enable users or hackers to intentionally or accidentally damage the VPC. Additionally, privileges should be revoked when no longer needed, i.e., when an employee leaves the company.<sup>54</sup>

The diagram below shows the principle of least privilege by allowing access to the management console only to individuals who need access for their job function:



## Industry Compliance

Many industries throughout the world are highly regulated. Often these industries have a legal requirement that requires a level of security, data retention, and auditing policies. AWS supports many international compliance requirements.<sup>55</sup>

Some key compliance standards are as follows:

- PCI DSS – for payment cards
- ISO 9001, 27001, 27017, 27018
- Fed ramp
- HIPAA – US health care privacy

A full list can be seen at <https://aws.amazon.com/compliance/programs/>.

## Identity and Access Management

Identity and access management is a key component of any security architecture. Identity and access management is about identifying the user and giving the user access to the resources necessary to perform their functions.<sup>56</sup> Identity and access management is also referred to as AAA. The key components of AAA are as follows:

- Authentication – Identifying the user.
- Authorization – Determining if the user is allowed to access the resource.
- Accounting – The ability to see what the user has done.

AWS Identity and Access Management (IAM) enables granular management of access control of any organization's systems on AWS. IAM enables the ability to specify who has access to which services and resources and under what circumstances. IAM policies enable secure access to specific AWS services, i.e. APIs. This enables an organization to control access and use of their systems.

AWS divides IAM into users and roles. An *IAM user* is a person accessing the AWS cloud. Generally speaking, an *IAM role* is used by an AWS service to access another service, i.e., EC2 accessing DynamoDB.

AWS has some specific components of its IAM systems. AWS uses the concept of *principals*. In AWS a principal is an IAM entity that is permitted to access AWS resources. AWS further breaks down the principal concept into root users, IAM users, and roles.

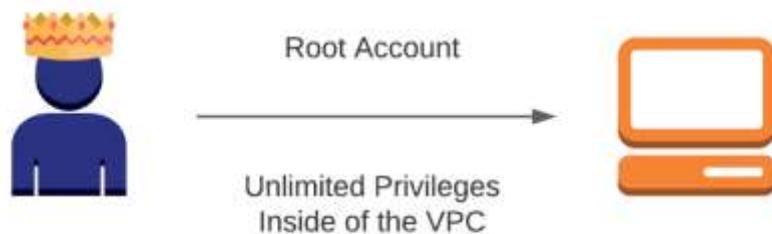
The diagram below shows the functions of authentication, authorization, and accounting:



## Root User

The root user is the person who created the AWS account. The root user has full system access. The root user can access the console and has programmatic access to AWS resources. Since the root user can do anything, including deletion of the VPC, it's best to use the root account to set up the VPC and then immediately create an IAM user with appropriate access to the VPC. This is similar in practice to not using the root account to log in to a UNIX or Linux system to prevent accidental system damage.

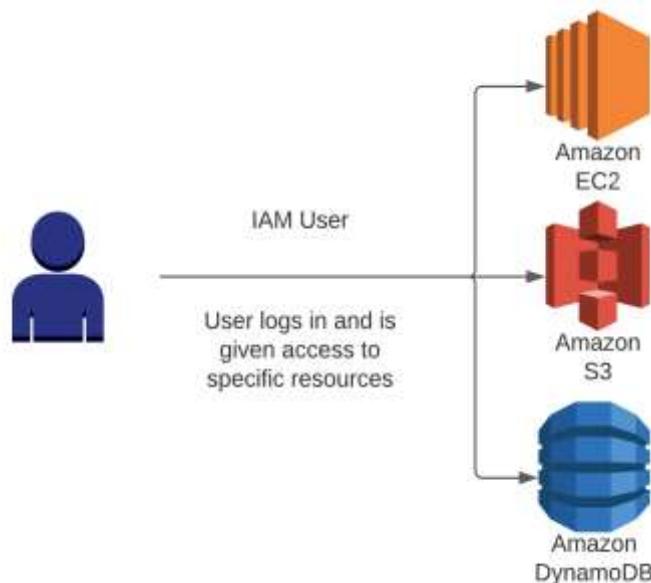
The diagram below shows an example of the root user privileges:



## IAM Users

IAM users are identities that have permissions to interact with AWS resources. IAM users are created by principals with administrative access. IAM users can be created with the AWS Management Console, CLI, or SDKs. IAM users are permanent unless deleted by an administrator.

The diagram below shows provides an IAM user accessing an AWS VPC:



## Roles and Security Tokens

Roles are used to provide access to AWS services. There are three types of roles in the AWS environment:

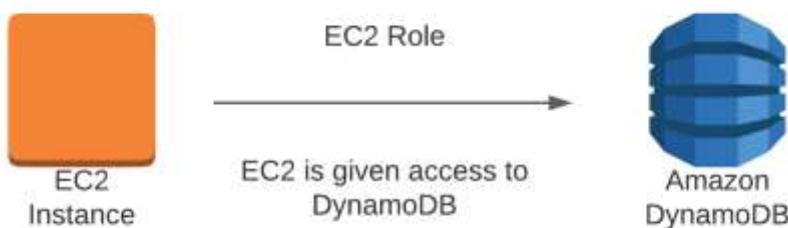
- EC2 roles
- Cross-account roles
- Identity federations

### EC2 Roles

*EC2 roles* enable EC2 computing instances to access AWS services, i.e., S3 and DynamoDB. To set up an EC2 Role, an IAM role is created and then applied to the EC2 instance. By creating the EC2 role, there is no need to store AWS credentials on the EC2 instance, which further enhances security.<sup>57</sup> Here is how EC2 roles work:

1. An EC2 role is created.
2. The EC2 role is applied to an EC2 instance.
3. When the EC2 instance attempts to access AWS services, a temporary token is provided to allow access.
4. The AWS service recognizes the tokens and grants access.
5. As temporary tokens expire, new tokens are generated frequently.
6. By rotating tokens, security is enhanced as no password (key) needs to be passed to the application.
7. This greatly enhances security. If an EC2 instance were to be hacked, no passwords would be given to the hacker. Since the tokens expire and are rotated frequently, even if hackers were to gain access to a token, it could not be used for long.

The diagram below shows an example of an EC2 instance accessing DynamoDB with an EC2 role:



## Cross-Account Roles

In the modern technology environment, it is frequently necessary for an organization to share resources with other business partners. To connect with organizations outside the VPC, cross-account roles are used. Connecting to other organizations can create significant business opportunities, but that connectivity also brings security challenges. The partner company may need access to certain resources but should not have access to any resource that could compromise the organization if lost or stolen. While it's always essential to provide access with the principle of least privilege, nowhere is it more critical than with connecting to external organizations.<sup>58, 59</sup> Therefore, be very strategic in assigning permissions to cross-account roles. Cross-account roles work in the following manner:

1. A role is created for the external user.
2. The external user connects to the AWS Secure Token Service (STS) and receives a temporary token.
3. The external user then provides the temporary token to AWS and is authorized to access the VPC.

The diagram below shows a cross-account role being used to access external VPCs.

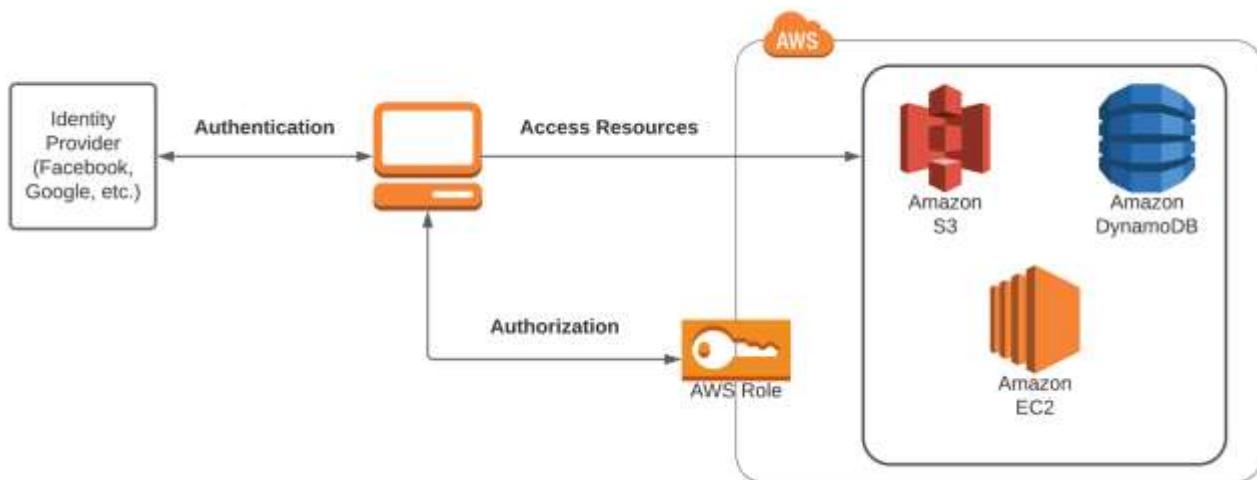


## Identity Federations

IAM is such a critical function for organizational security. As organizations grow in size and complexity, IAM can become challenging to manage. Often the best way to scale IAM systems is to connect (federate) with an identity provider. A VPC can connect to an identity provider and use its IAM database within AWS. Connections with an identity provider are built by building a trust relationship with the identity provider. After the trust relationship is established, a connection is made with OpenID connect (OIDC) or Security Assertion Markup Language 2.0 (SAML).<sup>60</sup>

Identity providers can be an organization's active directory or LDAP systems, or external providers such as Google, Amazon, Facebook, Twitter, or LinkedIn. AWS has three choices for authentication with identity providers: single sign-on, Federated IAM, and AWS Cognito.

The diagram below shows how identity federations work with the AWS platform.

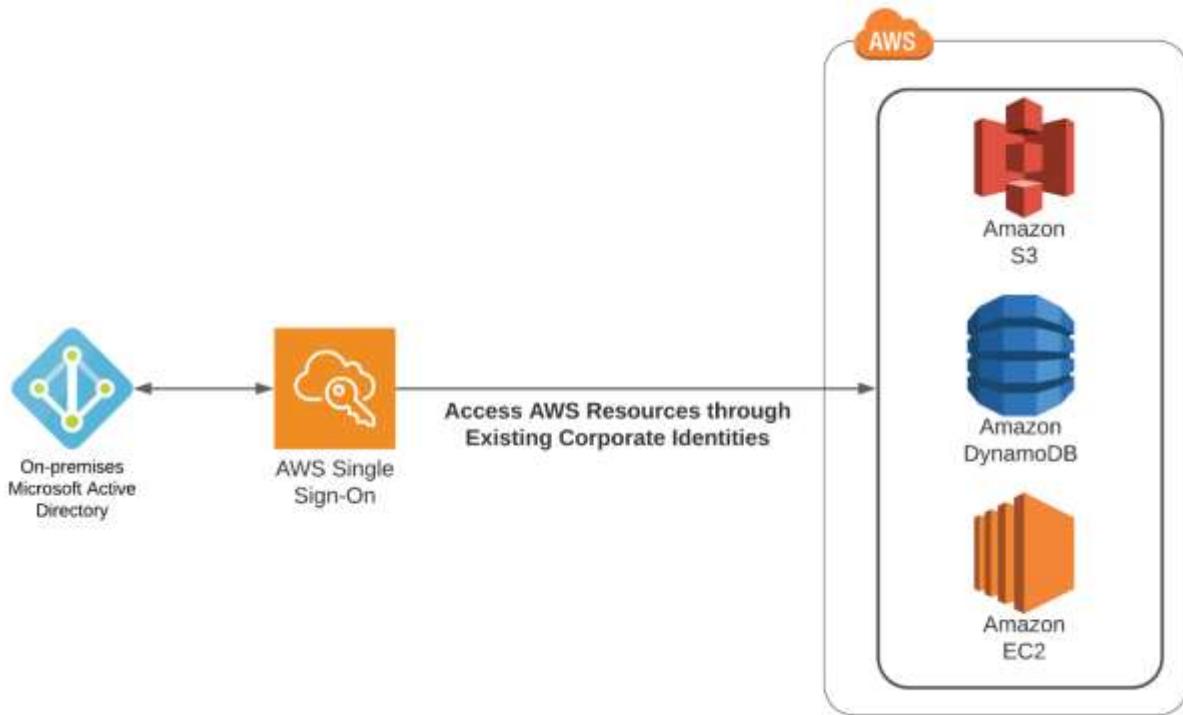


## **AWS Single Sign-On**

AWS Single Sign-On enables the user to authenticate once to the identity provider, and then they will not need to sign on to access AWS services.<sup>61</sup> It works in the following manner:

1. The user signs on to the identity provider.
2. The user is authenticated by the identity provider.
3. The identity provider determines what group (permissions) to give the user.
4. The user is given permissions and is authenticated and authorized to use AWS services.

The diagram below shows how Single Sign-On works with the AWS platform:



## Federated IAM

Federated IAM provides a means to authenticate with an external identity provider. Federated IAM enables significant and granular control over user functions. Federated IAM works in the following manner:

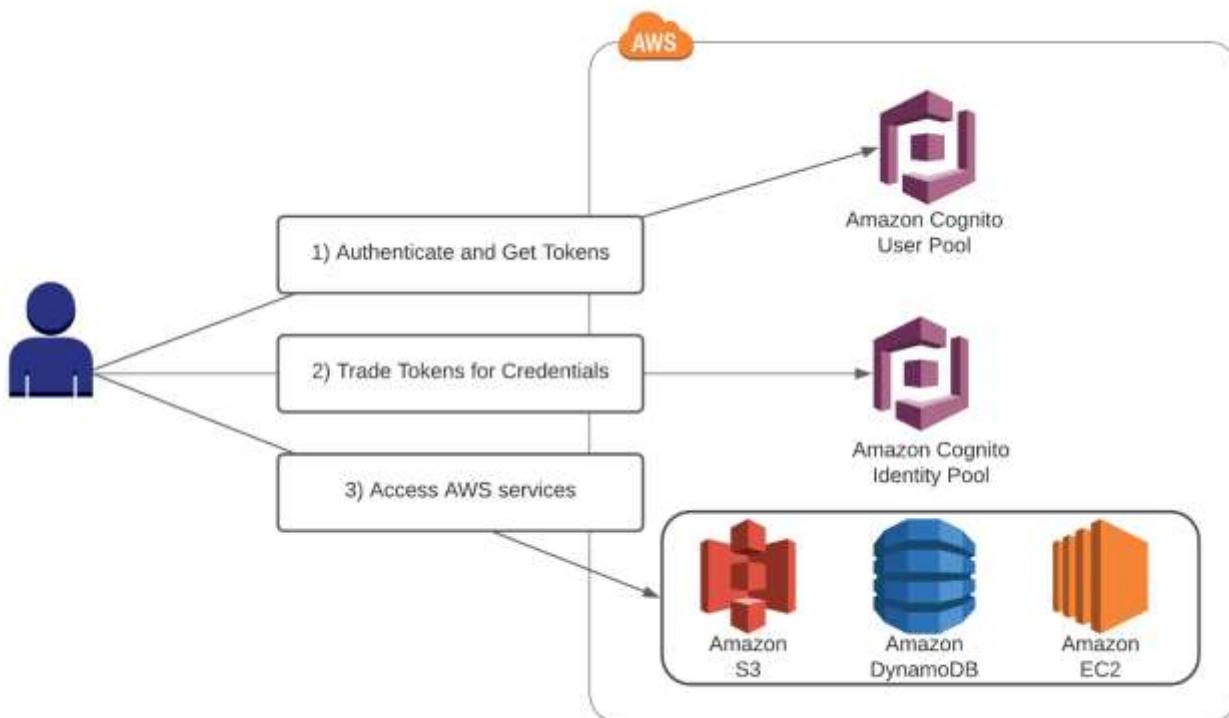
1. A user attempts authentication.
2. The request is forwarded to the identity provider.
3. The identity provider authenticates the users.
4. The identity provider determines the user's privileges.
5. The identity provider grants privileges based upon job role, the organization's cost center, and other factors.

## AWS Cognito

AWS Cognito is an identity and data synchronization service. AWS Cognito enables organizations to synchronize identity management and data across mobile devices. Cognito provides authentication, authorization, and user management for web and mobile apps. AWS Cognito users can sign in directly with a username and password, or with a third-party identity provider such as Facebook or Google.<sup>62</sup> AWS Cognito is simple and efficient. Cognito works in the following manner:

1. The user attempts to authenticate against Cognito.
2. Cognito authenticates the user.
3. Cognito provides a token for the user.
4. The user device trades token for credentials.
5. The credentials are then used to access AWS services.

The diagram below shows how AWS Cognito is used to authenticate mobile devices to access the AWS platform.

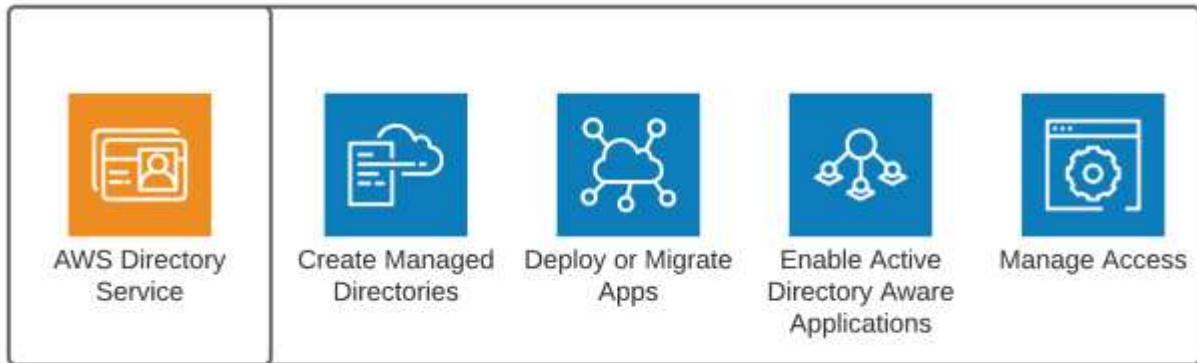


## AWS Directory Service

Another means to create a scalable IAM solution is with the AWS Directory Service. The AWS Directory Service provides hosted, dedicated tenant, Windows Active Directory (AD) servers. These are high-availability servers spread across two availability zones with the default configuration. The AD servers are actual Microsoft AD servers hosted by AWS. Being actual Microsoft AD servers, Microsoft-dependent workloads can function in the AWS VPC.<sup>63</sup>

AWS Directory Service can also be integrated with customers' on-premises Microsoft AD domain controllers. AWS Directory Service can also be used by AWS services such as EC2, RDS for SQL server, end-user computing, and AWS WorkSpaces for IAM functions. The hosted AD servers can also be used by EC2, RDS for SQL Server, AWS End User Computing, and AWS workspaces for IAM functions.

The diagram below shows how AWS Directory Service is used to facilitate Microsoft applications in the AWS environment.



## Authentication Process

Now that we have discussed the available IAM options, it is necessary to understand how the authentication process works under the different options.

### Username and Password

1. User logs in to the console with username and password.
2. AWS verifies the user's identity.
3. AWS provides an authorization based upon the user's privileges.

### Access Key

An *access key* is a combination of a twenty-character key ID and forty-character secret. The access key is used for authentication and facilitates connections to AWS via an API. This is generally performed with the Software Development Kit (SDK).

### Access Key and Security Token

When an IAM authentication needs to occur for an assumed role, a secure token is provided to the requesting application. The secure token, along with the access key, is used for authentication. This provides additional security over other methods.

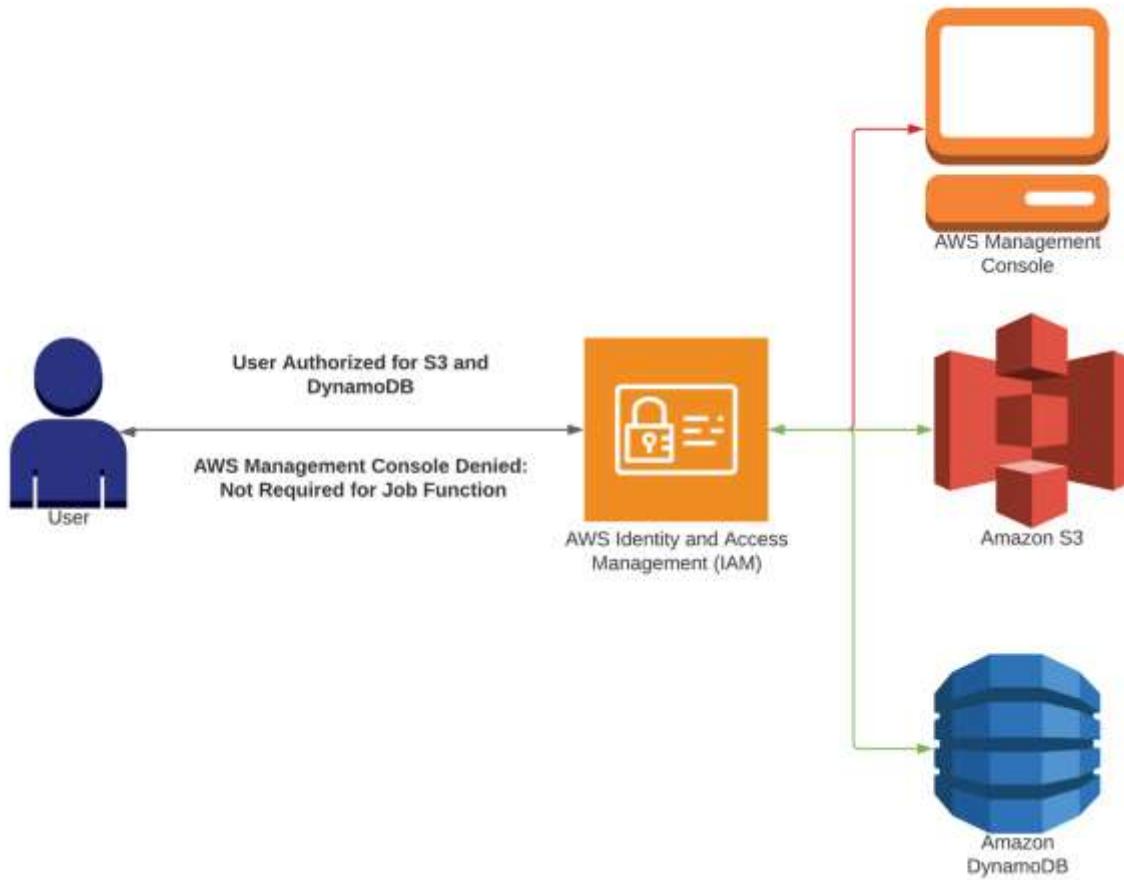
## Authorization

After authentication, it is necessary to authorize the user to perform whatever functions are necessary for their job function. To keep the VPC secure, the default policy is to deny access to all services. Authorization is really about granting permissions to necessary services that have been defined in the IAM policy.

IAM policy documents are written in JavaScript Object Notation (JSON). A policy document defines the following attributes:

- Effect – Allow or deny.
- Service – What service is being requested.
- Resource – What resource is being made available. This is the full Amazon Resource Name.
- Action – Determines the permissions of the user, i.e., read only, read/write.
- Condition – The condition component of IAM is optional. It enables very granular controls, i.e., to allow access from a specific IP subnet, time of day.

The diagram below shows the user authorization on the AWS platform:



## **Creating IAM Policies**

IAM policies determine who is allowed into the VPC and what actions they can perform. When creating an IAM policy, permissions can be applied to a specific resource or all resources. Providing access to specific resources is based upon the Amazon Resource Name (ARN). Providing access to all resources is accomplished with an asterisks (\*) wildcard. There are two types of policies available in AWS: AWS-managed policies and customer-managed policies.<sup>64</sup>

### **AWS Managed Policies**

*AWS-managed policies* are standalone policies created by AWS. These policies have several key attributes:

- Provide permission for services and functions within AWS.
- Are optimized for common use cases.
- Can be attached and moved to different entities and accounts in AWS.
- Can be based on job role to provide different levels of access.
- Have two major predefined roles: administrator access and power user.
- Administrator access provides full access to every service.
- Power users essentially have full access with the exception of IAM and organization management.

### **Customer-Managed Policies**

*Customer-managed policies* are managed by the customer for their account. These policies have several key attributes. They:

- Are not visible outside the customer's organization.
- Are custom made for the organization's specific needs.
- Can be attached to entities within the AWS account.

### **How to Create an IAM Policy**

There are several ways to create an IAM policy. To make an IAM policy from the console, perform the following steps:

1. Sign in to the IAM console from a user account with administrator privileges – <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose Policies.
3. You will see a list of AWS-managed policies. These are simple to use, updated by AWS as needed, and can help avoid configuration errors.
4. Alternatively, you choose to create a customer-managed policy starting with an AWS policy and then customize and use the policy generator. Or you can create one from scratch.

## Copying an AWS-Managed Policy

If an organization elects to create its own policy, the easiest method is to copy an AWS-managed policy and customize. This is the simplest method, and it helps to avoid configuration errors by starting with a known good configuration.

## AWS Policy Generator

The AWS Policy Generator is an easy-to-use questionnaire that will generate an IAM policy.<sup>65</sup> The policy generator works as follows:

1. Go to the policy generator page.
2. Answer the questions.
3. Assign permissions to specific resources. Multiple permissions can be created as statements.
4. A policy document is then created that can be edited.

The diagram below shows how the AWS Policy Generator can be used to create custom IAM policies:

The screenshot shows the AWS Policy Generator interface. It consists of three main sections:

- Step 1: Select Policy Type**: A dropdown menu is open, showing "Select Type of Policy" and "IAM Policy".
- Step 2: Add Statement(s)**: This section contains fields for "Effect" (Allow selected), "AWS Service" (Amazon S3 selected), "Actions" (SELECT\_ACTIONS selected), and "Amazon Resource Name (ARN)". There is also a note about ARN format and a "Add Conditions (Optional)" button.
- Step 3: Generate Policy**: A note states that a policy is a document written in the Access Policy Language that acts as a container for one or more statements. It includes a red link "Add one or more statements above to generate a policy".

At the bottom, there is a note about the AWS Policy Generator's purpose and a copyright notice for Amazon Web Services, LLC.

## Create an IAM Policy from Scratch

IAM policies can be generated from scratch in JSON format. When creating an IAM policy, it is essential to make sure to use the proper grammar and syntax. Creating an IAM policy from scratch is ideal for organizations that have individuals with JSON programming expertise and require significant policy customization.

### Sample IAM Policy

Now that we have discussed the methods to create an IAM policy, let's evaluate a sample policy. A sample policy can be seen below:

```
{
  "Version": "2020-09-01",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:volume/*",
        "arn:aws:ec2:*:*:instance/*"
      ],
      "Condition": {
        "ArnEquals": {"ec2:SourceInstanceARN": "arn:aws:ec2:*:*:instance/instance-id"}
      }
    }
  ]
}
```

Now let's evaluate this IAM policy.

1. The “version” and “date” explain when the policy was created.
2. The first string allows mounting and unmounting of a volume on EC2.
3. The next statement provides the locations of the volumes.
4. The third line specifies conditional elements that allow the policy to be in effect and is optional.

## Applying the IAM Policies

After the IAM policies are created, they need to be applied in order to function. IAM policies can be applied in several manners. Policies can be applied as a user policy, managed policy, or group policy.

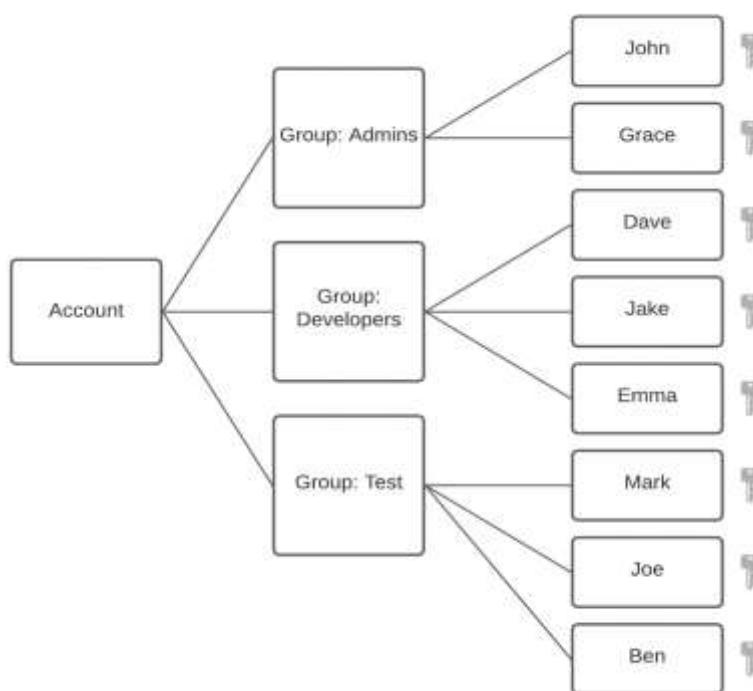
**User policy** – A user policy is applied to individual users. This works well but quickly becomes unscalable when many users exist in an organization. Imagine configuring individual policies for two hundred thousand employees.

**Managed policy** – With managed policies, a policy is created and exists independently of the user. The policy can then be attached to users or groups (collections of users). This method is effective and scalable.

**Group policy** – An organization can create a policy for a certain set of individuals. This is known as *role-based access control*. In role-based access control, users are put into groups.

For example, systems administrators all need the same access, so a policy is created for the group “systems administrators.” Then, a policy is applied to that group that provides access to the resources necessary to perform their jobs. This can be used for any group of users accounting, finance, etc. Users are then added to groups and inherit the policy from the group. This is a highly effective and scalable method for IAM policies.

The diagram below shows how group policies are used for IAM with the AWS platform:



## Further Securing IAM with Multifactor Authentication

Organizations looking for enhanced security around IAM can use *multifactor authentication* (MFA). MFA applies to the security concept of something you have and something you know. A perfect example is a debit card. To access money in your account, you need the card (something you have) and a pin number (something you know). This combination greatly enhances the security posture. MFA works in the following manner:

1. The organization sets up an authenticator app or device with a key.
2. The authenticator device creates a one-time password that changes every few seconds.
3. When the user logs in with their username and password, AWS will provide a challenge asking for the one-time password.
4. If the user provides the correct one-time password, they will be authenticated into the system.

This setup provides substantial security. Even if a hacker were to obtain the username and password, they would not be allowed into the system without knowing the constantly changing one-time passwords.

The diagram below shows how multifactor authentication is used in the AWS platform:



## Multi-Account Strategies

Another method to increase the security of the VPC is to partition the organization into multiple small accounts and share information between accounts. Each small account will be placed into a single billing unit called an *organization unit* (OU). Since all organization units are placed into a single billing organization, the organization can still benefit from volume discounts on the total services they consume.<sup>66</sup> Multi-account strategies are especially beneficial from a security perspective because of the following security enhancements:

- Isolation between organizational units.
- The ability to share only the necessary information between units.
- The ability to reduce the visibility of workloads between organizations.
- The ability to reduce the blast radius (meaning that if a problem happens in a single OU, it won't affect other OUs).
- The ability to truly compartmentalize data.

The diagram below shows an example of a multi-account strategy on the AWS platform:



## **Securing Access Between Organizational Units**

Access between organizational units is restricted by using a service control policy. Service Control Policies (SCP) provide IAM-like functionality across organizational units. SCPs allow the ability to “whitelist” or “blacklist” services or actions.

Service control policies are the heart of the control between organizational units. Service control policies help with the following:

- Access control – SCPs help manage access across the organization’s accounts using IAM.
- Encryption – Integration of AWS KMS helps to meet security requirements.
- Auditing – SCP integrated with CloudTrail. This provides the ability to track changes and discover any malicious activities.
- Automation – Ability to apply policies to the entire organization (i.e., HIPPA compliance for healthcare).

## **Restricting Network Access**

A strong security posture involves allowing traffic needed for business operations while keeping all unwanted traffic out of the network and services. Traffic filtering is accomplished with network ACLs, security groups, and firewalls. Network ACLs and security groups are covered in depth in the networking section of this book.

Firewalls are widely used in enterprise networking. Traditional firewalls keep unwanted traffic out of the network like a network ACL does, but firewalls are stateful. Since firewalls are stateful, they know the state of every connection that has passed through the firewall. Being stateful, firewall default policies block all incoming traffic but allow return traffic initiated by internal users that pass through the firewall. Modern firewalls can recognize common attack patterns and stop them by dynamically applying new firewall rules. AWS has a modern firewall solution that can be used with CloudFront, Application Load Balancers, and API gateways.

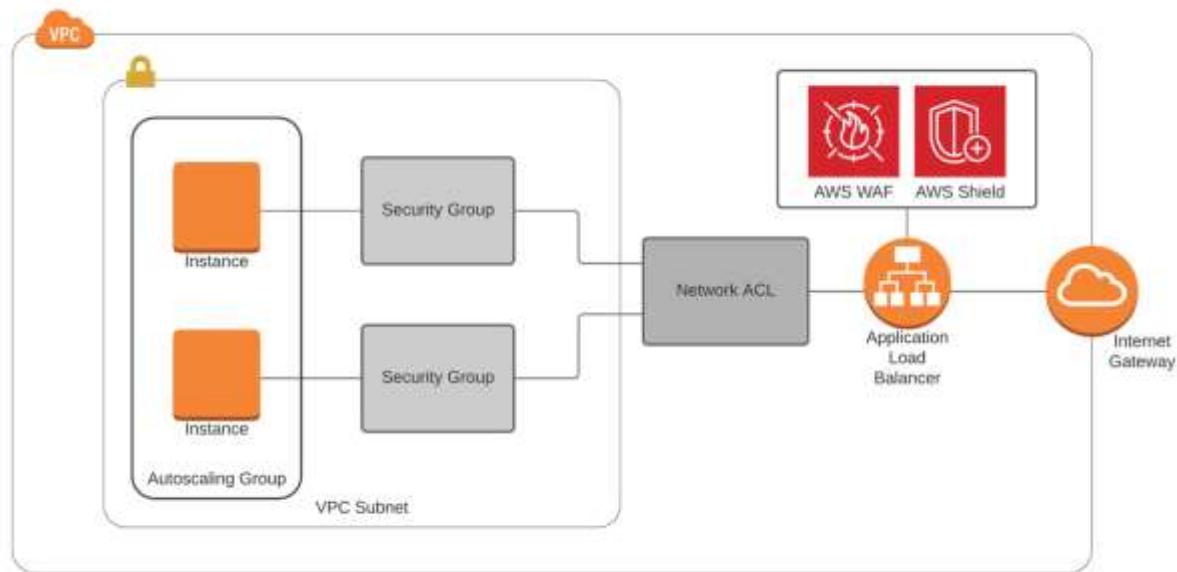
## **Preventing Distributed Denial of Service Attacks**

*Distributed denial of service* (DDoS) attacks are a common assault against an organization’s systems. A DDoS attack is designed to interrupt the normal function of a server, application, or network by overwhelming the service or its surrounding infrastructure. A DDoS attack is implemented by flooding traffic or server requests from multiple computers on the internet. Preventing a DDoS attack takes a full security posture. The key elements of DDoS prevention within AWS are as follows:

- Block unwanted traffic with network ACLs, which reduces the options the attacker can use to attack the network or server.
- Keep unwanted traffic out of servers and AWS services with security groups.

- Use a firewall. Adding AWS Web Application Firewall (WAF) can recognize common attacks and can dynamically apply policies to mitigate these attacks.
- Leverage AWS Shield, which provides enhanced DDoS protection. There are two versions of AWS Shield—Standard and Advanced. AWS Shield Standard is provided at no additional cost for organizations using AWS WAF. AWS Shield Advanced is an enhanced option at an additional cost that provides protection to EC2, ELB, CloudFront distributions, Route 53, and AWS Global Accelerator.
- Leverage autoscaling. Since the goal of a DDoS attack is to overwhelm the network or computing platform, autoscaling can help mitigate against DDoS attacks. During a DDoS attack, autoscaling can help increase compute capacity to offset the loss of computing capacity from the attacker

The diagram below shows how multiple network security measures are combined to thwart a distributed denial of service attack.



## Amazon Web Application Firewall (WAF)

AWS WAF is a web application firewall that protects against attacks. WAF monitors http(s) requests looking for exploits and can mitigate an attack if it occurs. WAF controls access to content based upon the firewall policy. WAF can assist with the protection of many AWS services, including CloudFront distributions, Amazon API Gateway, or application load balancers.

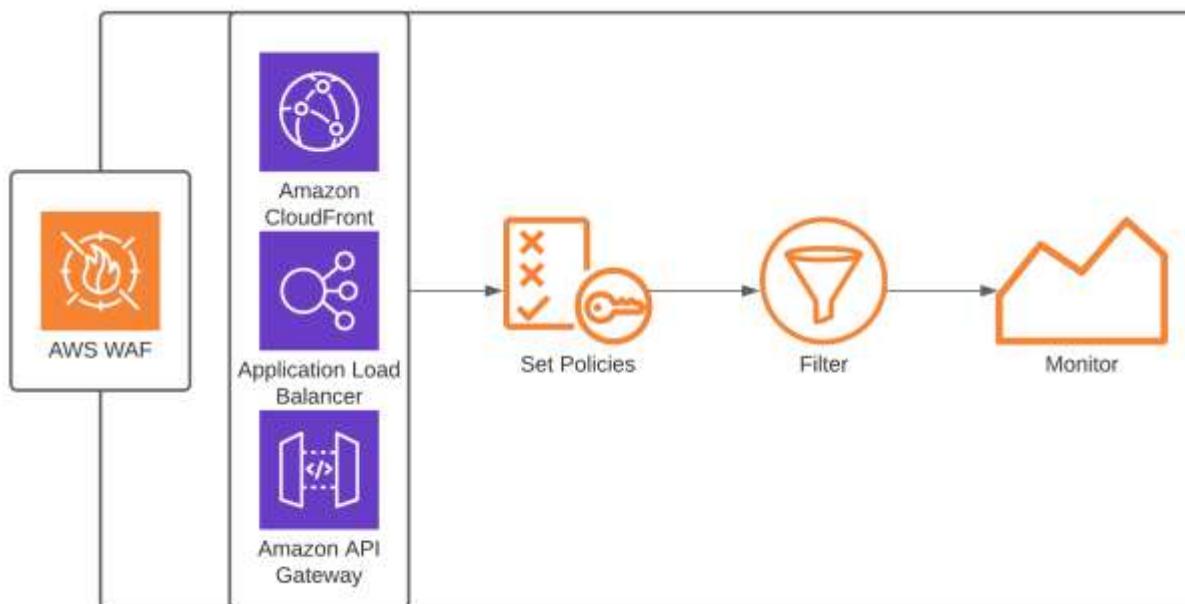
AWS WAF provides granular control to protect an organization's resources. It provides a means to control access through web ACLs, rules, or rules groups. Web access control lists (web ACLs) are lists that either allow or deny traffic depending upon the configured policy. Rules are

statements that allow or deny traffic based on the criteria in the policy. Rule groups are groups of individual rules that can be reused in other places.<sup>67</sup>

Setting up and using WAF is performed in the following manner:

1. Enable WAF on the application or device.
2. Create a policy that filters access to the application.
3. WAF analyzes the traffic depending on the policies created.
4. WAF will permit or deny the traffic depending on the traffic's adherence to the WAF policy.
5. If an attack occurs, new rules can be created to mitigate the attack.
6. WAF integrates with CloudWatch to provide increased visibility into network traffic and potential or actual attacks.

The diagram below shows how the AWS Web Application Firewall functions in the AWS environment.



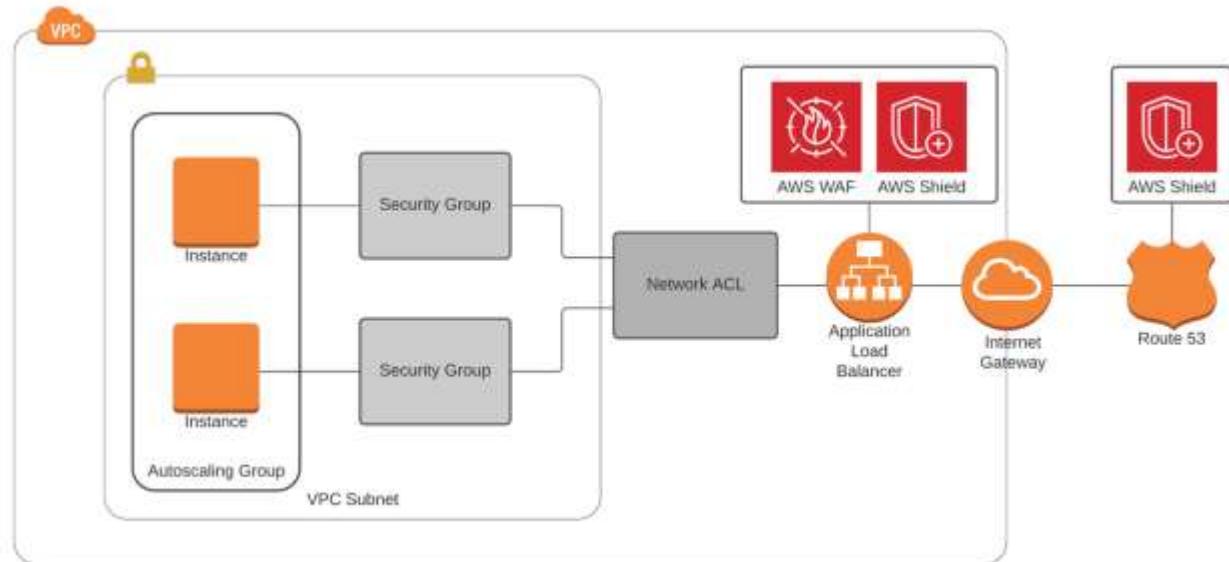
## AWS Shield

AWS Shield is an AWS service to protect against DDoS attacks. AWS Shield is available in two versions: AWS Shield Standard and AWS Shield Advanced.<sup>68</sup>

### **AWS Shield Standard**

AWS Shield Standard is a free DDoS protection service for AWS customers using WAF. AWS Shield Standard protects against the most common attacks. According to AWS, Shield Standard blocks against 96 percent of the most common attacks, including SYN/ACK floods, reflection attacks, and http slow reads. AWS Shield Standard works based on the logic contained in its policy.

The diagram below shows how AWS Shield is used to defend against distributed denial of service attacks:



### **AWS Shield Advanced**

AWS Shield Advanced is a paid DDoS protection service for AWS customers. It has many advantages and additional features above AWS Shield Standard. AWS Shield Advanced has a rich feature set and functionality, including:

- Additional protection for volumetric attacks by adding intelligent attack detection and mitigation tools.
- Dynamic solution that can look at traffic patterns to determine if an attack is occurring.
- Ability to detect an attack and can automatically deploy ACLs to mitigate the attack.
- Visibility and notification for layer 3/4/7 attacks.

- 24/7 access to a DDoS response team, assuming the customer is a member of the business or enterprise support.
- Protection for ELBs, EC2 instances, CloudFront distributions, Route 53, and AWS Global Accelerator.

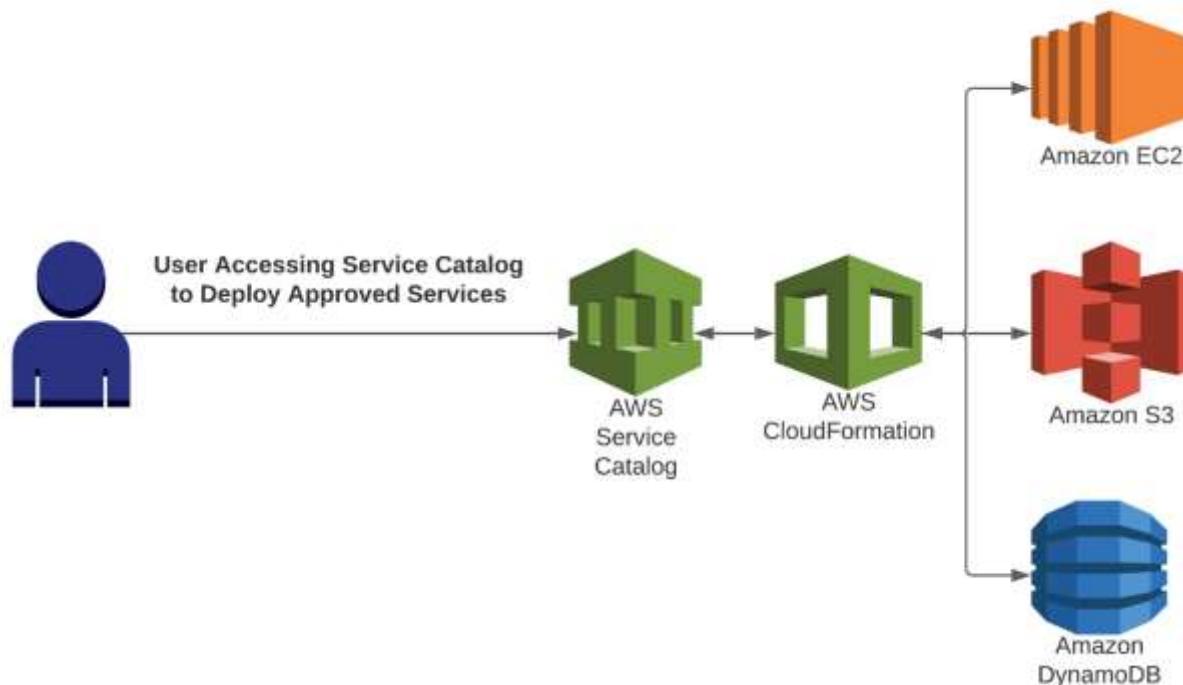
## AWS Service Catalog

As previously discussed, a full security posture includes physical security, access lists, security groups, firewalls, DDoS protection, IDS/IPS, and controlling and optimizing what is placed on the network. While there are many ways to control what is placed on the network, AWS makes it easier with the use of the AWS Service Catalog.

A service catalog is a means to create a list of approved services. The service catalog is defined by the customer, so they can allow what services they desire on the organization's network. The service catalog can include specific AMIs, servers, software, databases, and multitier application architectures. Therefore, the service catalog can help ensure compliance with corporate security standards by limiting what system admins can place on the network.

For example, the service catalog can be configured to allow only security-hardened AMIs on the network (fully patched, disabling unnecessary services, etc.). AWS Service Catalog simplifies deployments, as administrators can allow only approved and compliant services.<sup>69</sup>

The diagram below shows how the service catalog is used to select and deploy approved services.

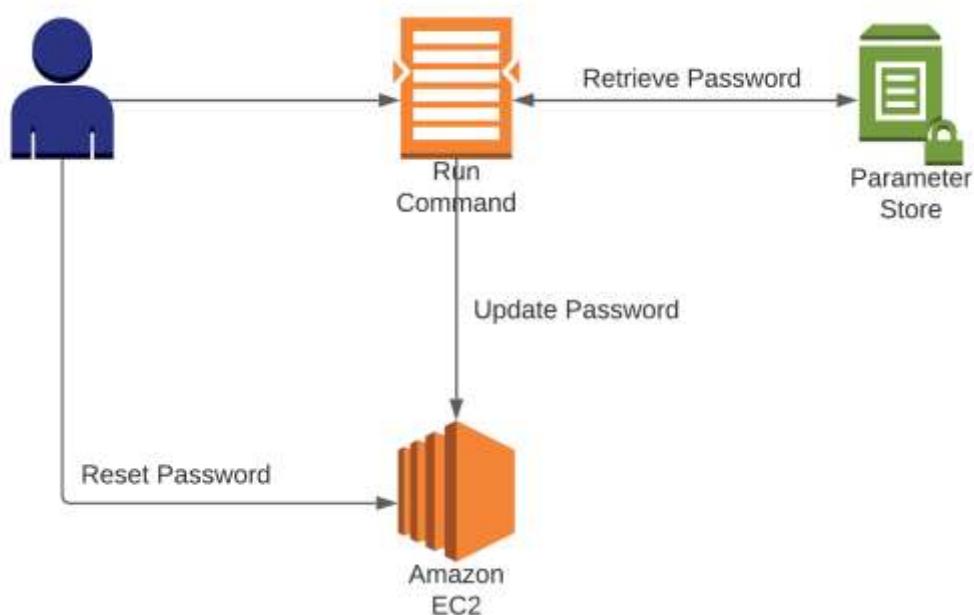


## AWS Systems Manager Parameter Store

Another component of an overall security posture is the secure management of passwords, database strings, and other critical components. Proper management of passwords is essential for a secure environment. If passwords are compromised, the organization would be open to attacks by hackers. Stolen passwords can lead to serious data loss and exploitation by hackers. AWS provides a solution for securely storing passwords, which is the AWS Systems Manager Parameter Store, which increases the security posture by separating an organization's code from their secret information.<sup>70, 71</sup>

The AWS Systems Manager Parameter Store provides secure, hierarchical storage for configuration data management and secrets management. It scans your managed instances and will report any policy violations detected. The AWS Systems Manager Parameter Store is a scalable, hosted, serverless environment optimized for storing passwords, database strings, license codes, and API keys. For extremely sensitive information, it's advisable to encrypt data in the store. It provides a means to encrypt sensitive data and provides an excellent means to track password use, as well as audit who has been accessing the system.

The diagram below shows how the Systems Manager Parameter Store can be used for secure password storage and maintenance.



## **Amazon Macie**

Amazon Macie is a fully managed service to enhance an organization's security. Amazon Macie uses machine learning and pattern matching to find and alert an organization if private data becomes publicly available. Most specifically, the alerts are based upon private information in S3 buckets becoming publicly available, or does not meet with compliance standards, i.e., unencrypted S3 bucket.

Macie provides an inventory of buckets that are encrypted, publicly accessible, and shared with other AWS accounts. Macie's findings can be searched and filtered through the AWS Management Console and relayed through the Amazon Event Bridge.

Macie integrates with event management systems, workflows, or AWS Step Functions to automate security remediation actions.

Amazon Macie, in combination with AWS Services, can help regulate for the General Data Privacy Regulation (GDPR), Health Insurance Portability, and Accountability Act (HIPAA).

## **Amazon GuardDuty**

Amazon GuardDuty is a threat-monitoring service to protect your AWS accounts, workloads, and data stored on AWS S3. GuardDuty monitors for malicious activity and provides detailed security findings. GuardDuty uses machine learning for pattern matching and detection of threats. GuardDuty can be set up to help remediate threats. Threats can be mitigated by the initiation of a Lambda function or by sending alerts to systems administrators.

GuardDuty protects against unusual data access on AWS S3, API requests from known malicious IP addresses, and compromised security credentials. GuardDuty provides detailed reporting that can be accessed in the AWS Management Console. Additionally, GuardDuty integrates with event management and workflow systems to help optimize an organization's security.

## **Amazon Inspector**

Amazon Inspector is a security service that provides an automated assessment of applications to uncover security vulnerabilities. Inspector looks for vulnerabilities such as deviations from security best practices, which are predefined rules that are regularly updated by AWS security personnel.

Once Inspector completes the assessment, the findings are placed in a report, ordered by level of severity. This report assists the organization in securing their systems by making systems administrators aware of known vulnerabilities. Some examples of the types of vulnerabilities identified by Inspector include checking to see if private systems are reachable from the internet, remote root login is enabled, or vulnerable software versions are installed.

## **Chapter 9 - AWS Applications and Services**

AWS has several key services that can be used to enhance VPC-based applications and services. This chapter will cover these key AWS services:

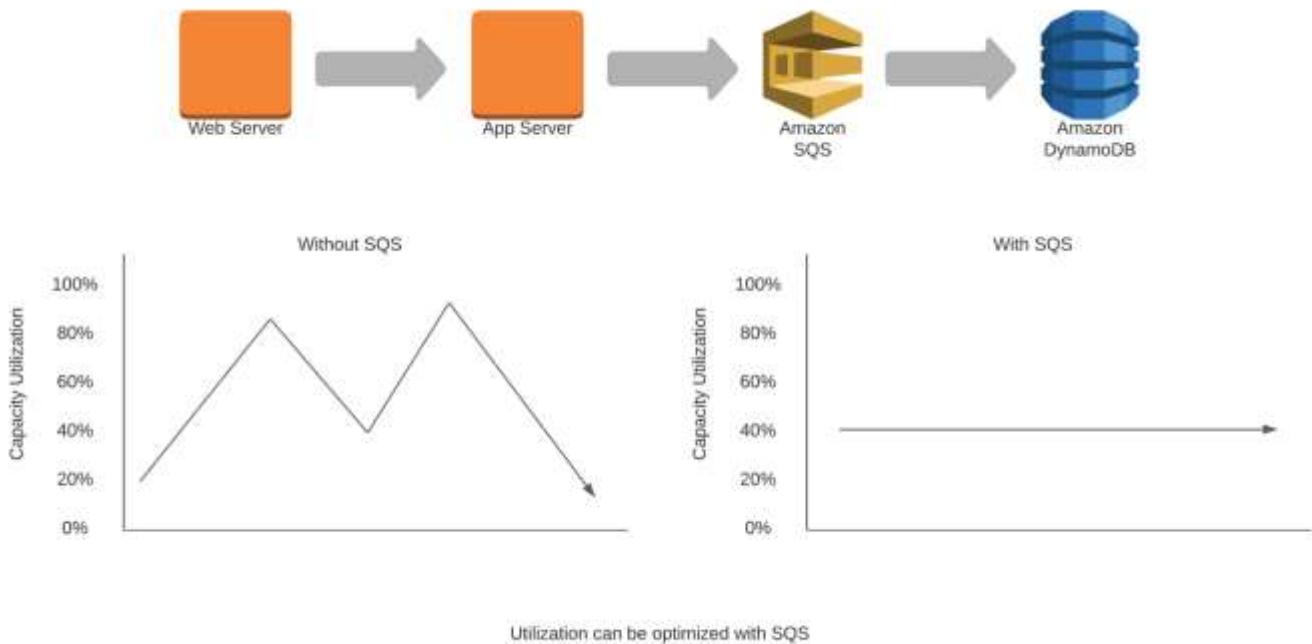
- Simple Queue Service (SQS)
- Simple Notification Service (SNS)
- Simple Workflow Service (SWF)
- Elastic Map Reduce (EMR)
- Kinesis
- Elastic Compute Service (ECS)
- Elastic Kubernetes Service (EKS)
- Elastic Beanstalk
- CloudWatch
- Config
- CloudTrail
- CloudFront
- Lambda
- Lambda@Edge
- Step Functions
- Rekognition
- CloudFormation
- AWS Certificate Manager (ACM)

### **AWS Simple Queue Service (SQS)**

To design a highly available and scalable application, it is sometimes necessary to decouple the components of an application's architecture. By decoupling the application's architecture, it is possible to minimize system bottlenecks and optimize the performance of the entire system. A great option for decoupling application architectures is the Amazon Simple Queue Service (SQS).<sup>72</sup>

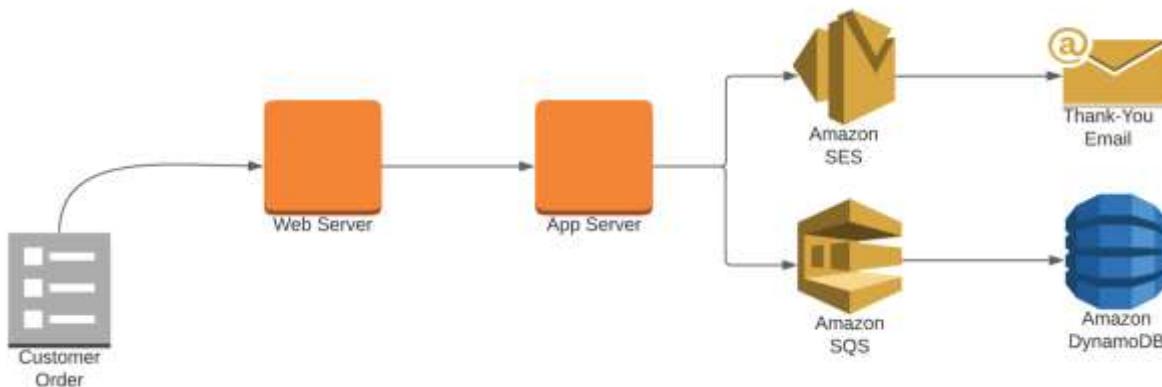
SQS is a message queuing service that provides temporary message storage prior to the message being transmitted to its ultimate destination. SQS enhances application availability by providing a way to retain messages when a part of the application's architecture is busy or unavailable. Since SQS can hold messages on the way to the destination, the organization can optimally size its architecture. If a traffic spike occurs, SQS will store and deliver the message when it's ready.

The diagram below shows how SQS can optimize performance and help promote scalability:



SQS is used to decouple application architecture components, which promotes scalability. SQS can further increase scalability and elasticity of services when SQS is used to autoscale instances based on the message's depth in the SQS queue. SQS can take the place of messaging middleware in multitiered applications. SQS is highly available, and multiple copies of every message are stored redundantly. SQS has integration with KMS to allow for end-to-end encryption. SQS is highly tunable transient storage. The SQS message queue retention period can be adjusted for up to fourteen days, with the default storage time being four days.

The diagram below shows how SQS queues are used to decouple application architecture components:



SQS can be configured for standard queues (the default), first in, first out (FIFO) queues, and dead-letter queues. The key attributes of these queue types are:

### **Standard Queues**

- Super-fast and support a nearly unlimited number of requests per second.
- Offers the fastest throughput and message delivery.
- Every message is delivered at least once.
- Best-effort delivery, and messages may be delivered out of order.

### **First In, First Out Queues**

- High throughput but much slower than standard queues.
- First in, first out delivery guarantees the messages will be processed once and in the order they are received.
- Since messages are sent in the order they are received, it is possible the FIFO queue will increase latency because all new messages will be waiting for the previous message to be processed.

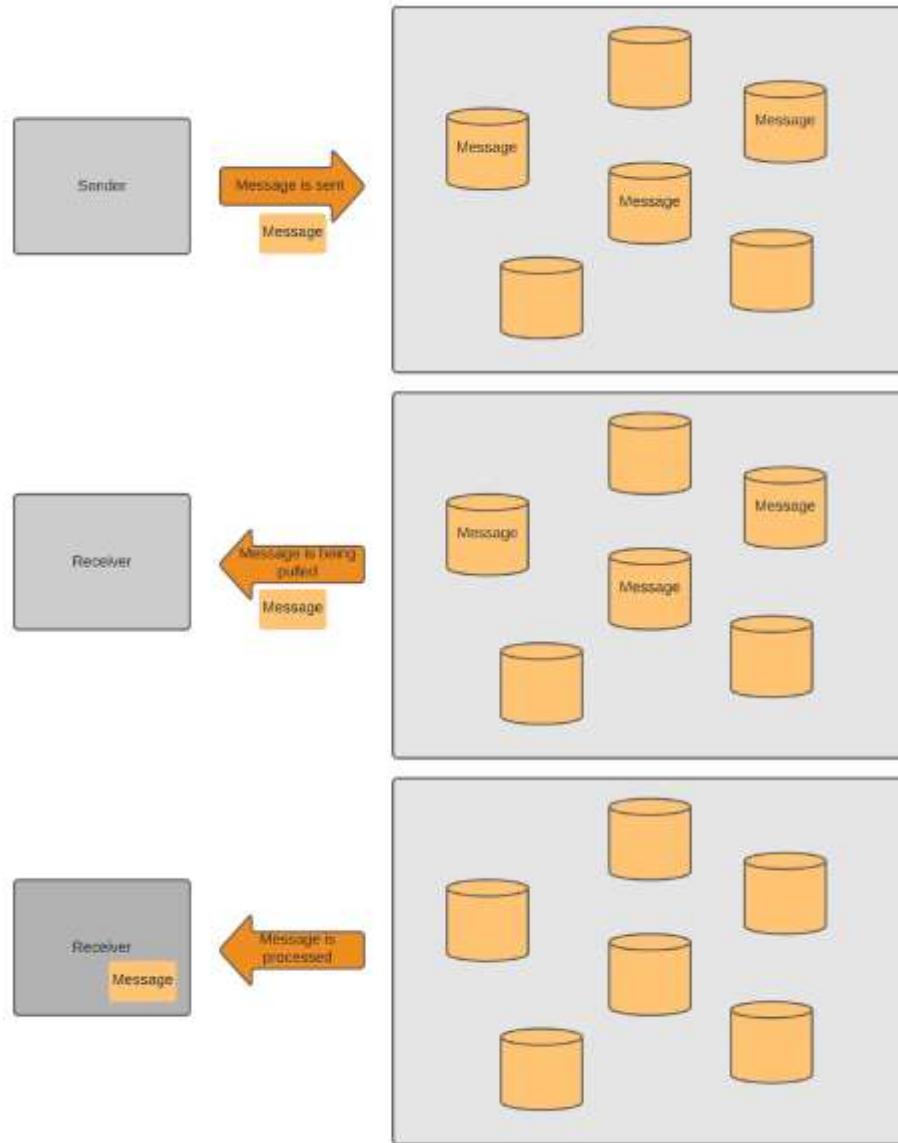
### **Dead-Letter Queues**

- Dead-letter queues (DLQ) can be set up to retain undeliverable messages if an error occurs in the system.

### **How SQS Works**

1. Messages are sent from the computing platform to the queue as a step to their ultimate destination.
2. After the message is inside the queue, it can be scheduled for delivery based upon the capacity of the ultimate destination for the message.
3. If the ultimate destination is busy, the message can stay in the queue until it is processed or times out. The message can stay in the queue for up to fourteen days based on the SQS configuration.
4. The message is pulled from the queue to be processed.
5. After the message is completely processed, the message is deleted from the queue.

The diagram below shows how SQS is used to reliably deliver messages to their ultimate destination:



## When to Use SQS

SQS can be extremely beneficial in designing a highly-available and scalable architecture. Some key situations where SQS can make a notable difference are as follows:

- With capacity planning and application scalability.
- To make sure messages (orders) are not lost if part of the system is overloaded (i.e., database on multilayered application).
- For cost optimization, as it offers the ability to right-size the instances supporting an application.

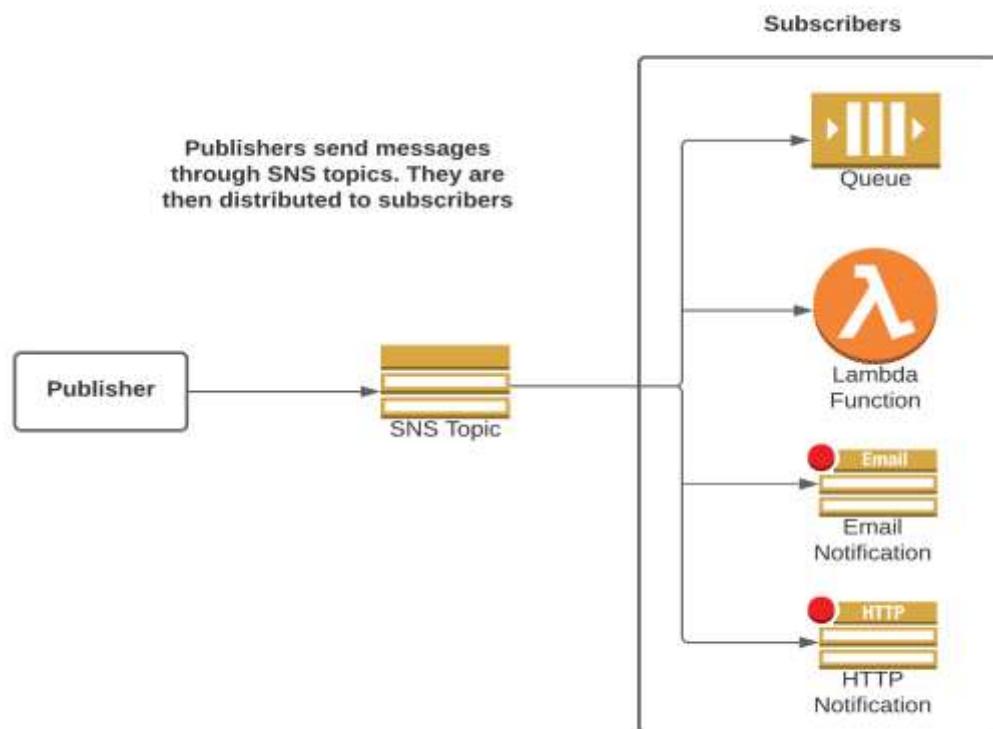
- For autoscaling, with its ability to trigger autoscaling based on queue depth, as opposed to a less direct metric, i.e., CPU utilization.
- To support an application's ability to handle large spikes in traffic without having to scale or make changes to the platform
- To handle increased traffic destined for databases, often without the need to increase write capacity on the database.

## AWS Simple Notification Service (SNS)

Amazon Simple Notification Service is a managed messaging service to deliver messages between systems, or between systems and individuals. SNS is used to decouple messages between microservice applications. SNS is also used to send SMS, email and push messages to devices.<sup>73</sup>

SNS facilitates communication between senders and recipients via a publish-subscribe model. The publish-subscribe (pub-sub) messaging model enables notifications to be delivered to clients using a push mechanism. Push notifications notify clients of message updates. SNS consists of publishers and subscribers. Publishers communicate by sending a message to a topic. A subscriber subscribes to a topic and receives messages that have been published to the topic. It functions like an email list — you subscribe to the list and receive messages the sender (publisher) sends to the list.

The diagram below shows how the SNS publisher/subscriber model works for message delivery:



SNS is a highly available platform that by default runs across multiple availability zones. SNS can be used to fan out messages to a large number of subscriber systems or customer endpoints. Endpoints can be many things; some examples are SQS queues and Lambda functions. SNS allows message filtering through policies so that only desired notifications are received. SNS encrypts messages immediately to protect from unauthorized access.

SNS can be used in a variety of situations. Some common SNS use cases can be seen below:

### **Application and System Alerts**

- SNS can send a notification when a predefined event occurs (i.e., a limit is passed).
- For example: When a CPU's utilization goes over 80 percent, notify system administrators.

### **Email and Text Messages**

- SNS can push notifications to people's emails and/or send them text messages.
- For example: a company's CEO is going to be on TV, and a broadcast link is sent to employees' emails and phones.

### **Mobile Notifications**

- SNS can send push notifications directly to mobile applications.
- For example: notify a customer of a flash sale on your app.

## **AWS Simple Workflow Service (SWF)**

SWF is a workflow management solution. SWF enables the coordination of tasks across distributed application components. SWF enables you to create a workflow of tasks that take multiple steps for completion. SWF then coordinates the execution of tasks across the platform. Normally it would take substantial application development to coordinate tasks across multiple systems. As SWF is a prebuilt workflow management solution, all that's necessary is to tell SWF the necessary workflow steps, and SWF manages all coordination of steps until completion.<sup>74</sup>

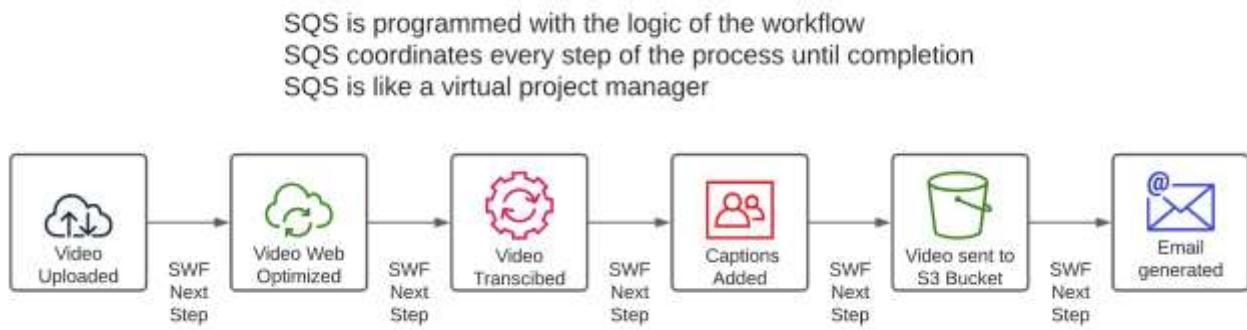
SWF manages workflows using workers that carry out the steps. These workers are programmed to perform, process, and confirm the completion of each step. Workers can be deployed using EC2 or Lambda, or on a local system.

SWF controls the flow of tasks using *deciders* to keep track of the workflow. The decider receives decision tasks from SWF and then determines and schedules the next step to complete the task.

Let's explore a common multistep workflow that would benefit from SWF:

1. Video is uploaded.
2. Video is processed and converted to an optimized format.
3. After formatting, the video is transcribed.
4. After transcription, the video's transcriptions are added as subtitles.
5. After final processing, the video is stored on a server.
6. After the video is stored on the server, the user gets a notification that their video is ready for download.

The diagram below shows how SWF can be used in the above multistep workflow.



Without SWF, it would be necessary to develop software to manage the coordination of all steps. With SWF, all that is necessary is to configure the workflow in SWF. SWF will manage the process from the time the video is uploaded until the time the user gets notified that the video is ready for download.

## AWS Elastic Map Reduce (EMR)

Elastic Map Reduce is an AWS application for processing and analyzing large amounts of data. EMR acts as a managed cluster that manages big data frameworks. Typically, big data management requires specialty servers. These specialty servers must be provisioned and require operating system installation, application installation, and configuration. AWS EMR provides an alternative to this by offering a prebuilt system that facilitates big data analysis and processing without the need for server management, application installation, and configuration.<sup>75</sup>

EMR is a managed service that takes a lot of complexity out of big data management. EMR uses tools such as Apache Spark, Apache Hive, Apache HBase, Apache Flink, Apache Hudi, and Presto. EMR takes advantage of these tools in order to offer greater performance than traditional solutions, at a lower overall cost.

The diagram below shows how AWS EMR is used to analyze large amounts of data:



## **AWS Kinesis**

Amazon Kinesis is an AWS service for collecting, processing, and analyzing streaming data. Amazon Kinesis can collect and analyze streaming data in real time from sources including video, audio, logs, website clickstreams, and Internet of Things (IoT) devices. Unlike traditional environments where you collect, store, and then analyze the data, Kinesis can do this in real time. By analyzing data in real time, the organization can receive a competitive advantage by not having to wait for data storage and processing to obtain actionable insights.<sup>76</sup>

### **Why Use Kinesis?**

Kinesis is ideal for situations when large amounts of streaming data need to be rapidly moved and processed. Some application examples include the following:

- Weather sensors located across the globe that report current conditions every five minutes.
- A fleet of airplanes sending information about their status every few minutes.

These applications generate a large volume of streaming information, and Kinesis makes managing these types of data streams in real time feasible.

### **Kinesis Platforms**

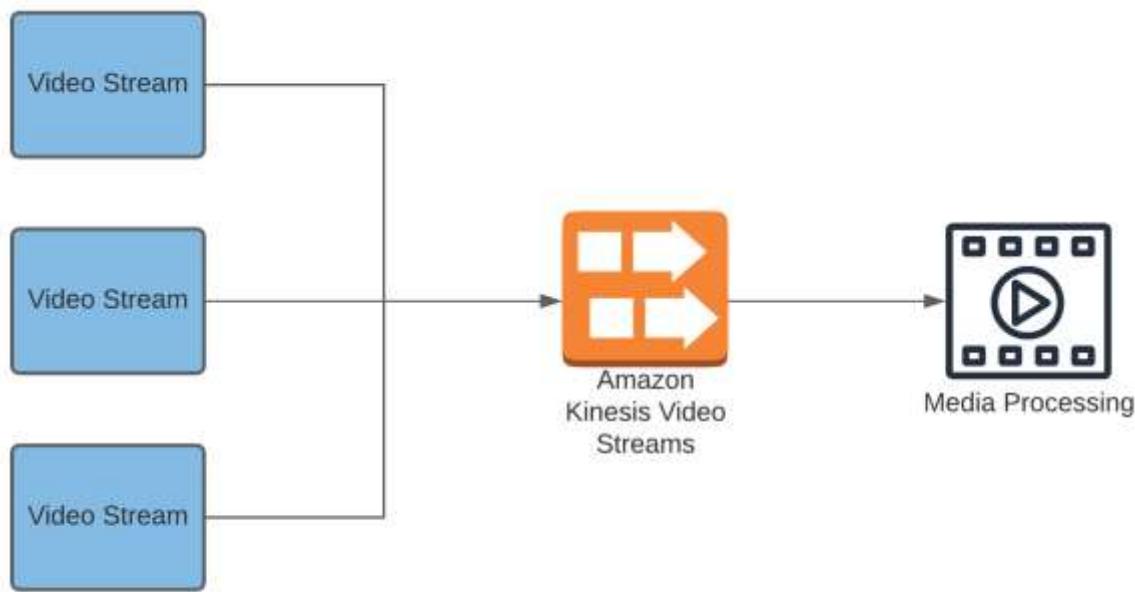
There are four Kinesis platforms:

- Kinesis Video Streams
- Kinesis Data Streams
- Kinesis Data Firehose
- Kinesis Data Analytics

## AWS Kinesis Video Streams

Kinesis Video Streams is a Kinesis application specifically for video data. Kinesis Video Streams enables the collection of videos from multiple sources, as well as providing ingestion, storage, and indexing of multiple streams. The videos obtained by Kinesis streams can be sent for media processing or used by machine learning applications.

The diagram below shows how AWS Kinesis Video Streams is used to capture large amounts of real-time streaming data.



## AWS Kinesis Data Streams

Kinesis Data Streams is highly scalable platform for real-time data. Kinesis Data Streams can capture gigabytes per second of data from hundreds or thousands of sources. This includes financial transactions, location tracking, database event streams, and other streaming data. Kinesis Data Streams can ingest an organization's data and provide meaningful insights using business intelligence tools.

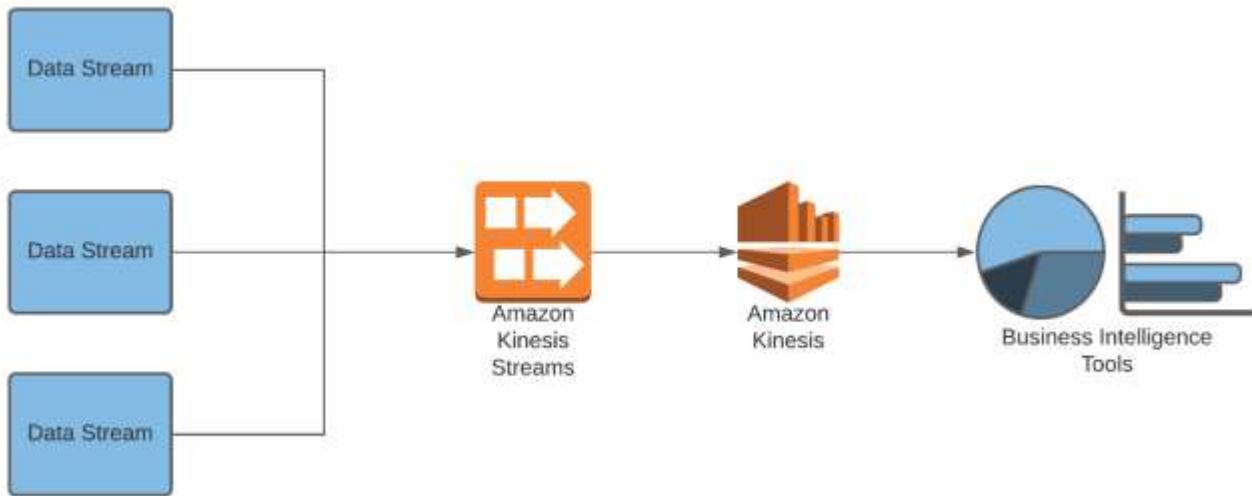
Kinesis Data Streams can be used in a variety of situations. Some common use cases are:

- Large event data collection.
- Real-time data analytics.
- Capturing gaming data.
- Capturing mobile data.

## How It Works

1. Streaming data is captured by AWS Kinesis Streams.
2. Streaming data is then sent for processing via EC2 and/or Kinesis Data Analytics.
3. After data is processed, the data can be sent to business intelligence tools.

The diagram below shows how AWS Kinesis Data Streams are used to capture large amounts of real-time streaming data.



## AWS Kinesis Data Firehose

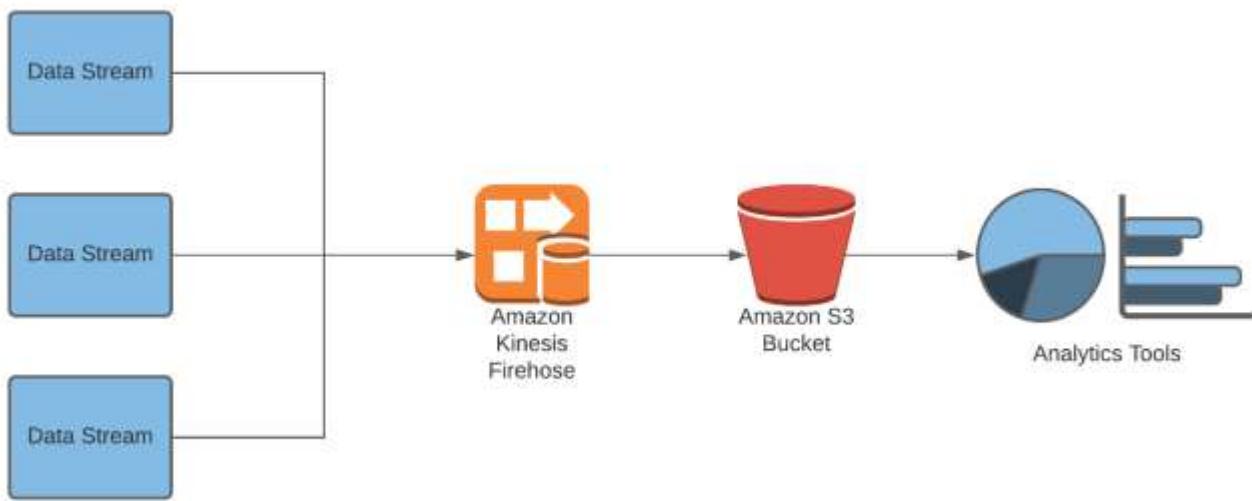
Amazon Kinesis Data Firehose is a managed service to load streaming data into data stores, data lakes, and data analytics services. Amazon Kinesis Data Firehose can capture streaming data and put it in S3 and Redshift, as well as other services. Amazon Kinesis Data Firehose scales to match the throughput of your data and supports autoscaling and data monitoring.

Amazon Kinesis Data Firehose pricing is based on throughput. Throughput is based on the number of shards. A *shard* is a throughput unit, with a capacity of 1 megabit per second. During setup, the administrator configures the capacity by the number of shards. Shards are increased as capacity requirements increase. A policy can be set up to autoscale the number of shards based upon utilization—up to ten shards per region, per account can be created. If additional shards are required, they can be obtained by contacting AWS support.

## How It Works

1. Streaming data is captured by AWS Kinesis Data Firehose.
2. Streaming data is sent for storage, i.e., S3.
3. Stored data can be analyzed with analytics tools.

The diagram below shows how AWS Kinesis Data Firehose is used to capture large amounts of real-time streaming data and store the data in S3.



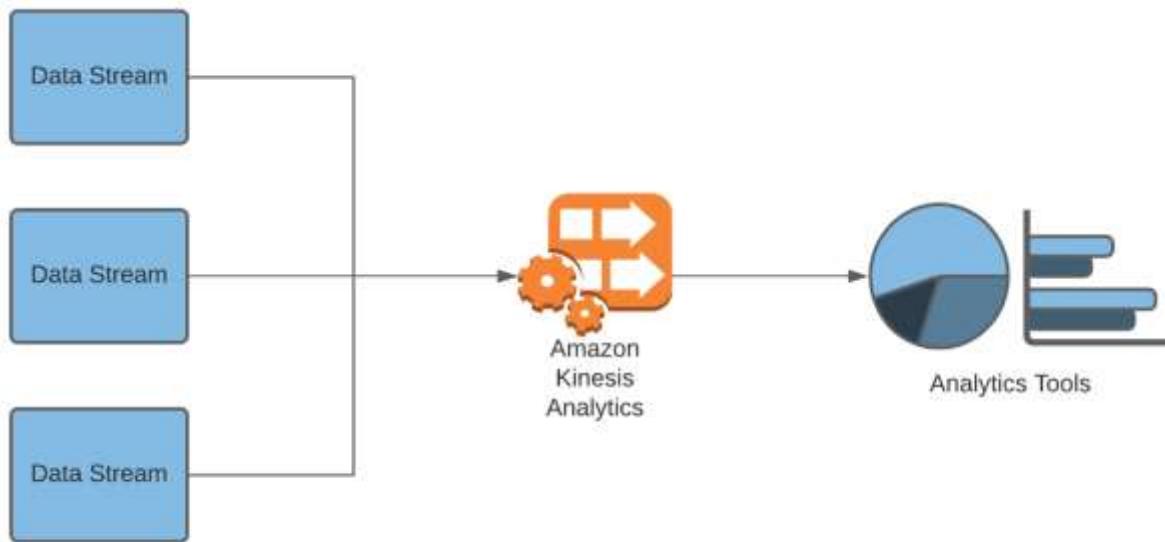
## AWS Kinesis Data Analytics

Kinesis Data Analytics is a managed service to transform and analyze streaming data in real time. Kinesis Data Analytics uses Apache Flink to process data streams. Kinesis Data Analytics can autoscale to meet an organization's needs. Kinesis Data Analytics can be queried with standard SQL queries.

## How It Works

1. Streaming data is captured by Amazon Kinesis Data Streams, Firehouse, Elasticsearch, S3, DynamoDB, and other data sources.
2. Amazon Kinesis Data Analytics analyzes data in real time.
3. Amazon Kinesis Data Analytics sends processed data to analytics tools.

The diagram below shows how AWS Kinesis Data Analytics is used to capture and process large amounts of data in real time.

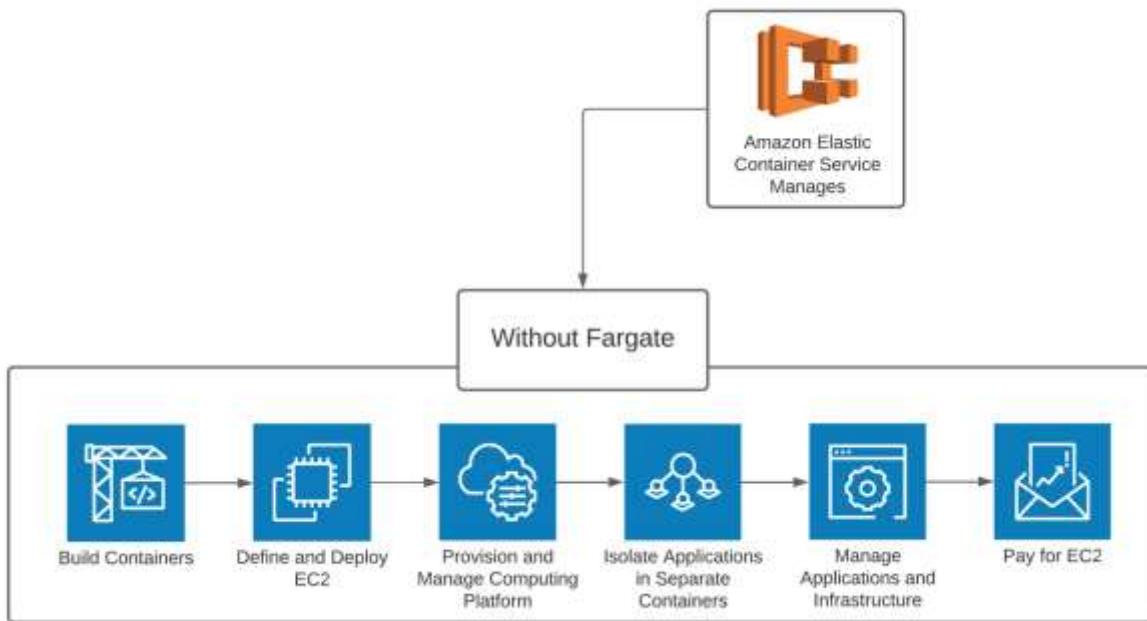


## AWS Elastic Container Service (ECS)

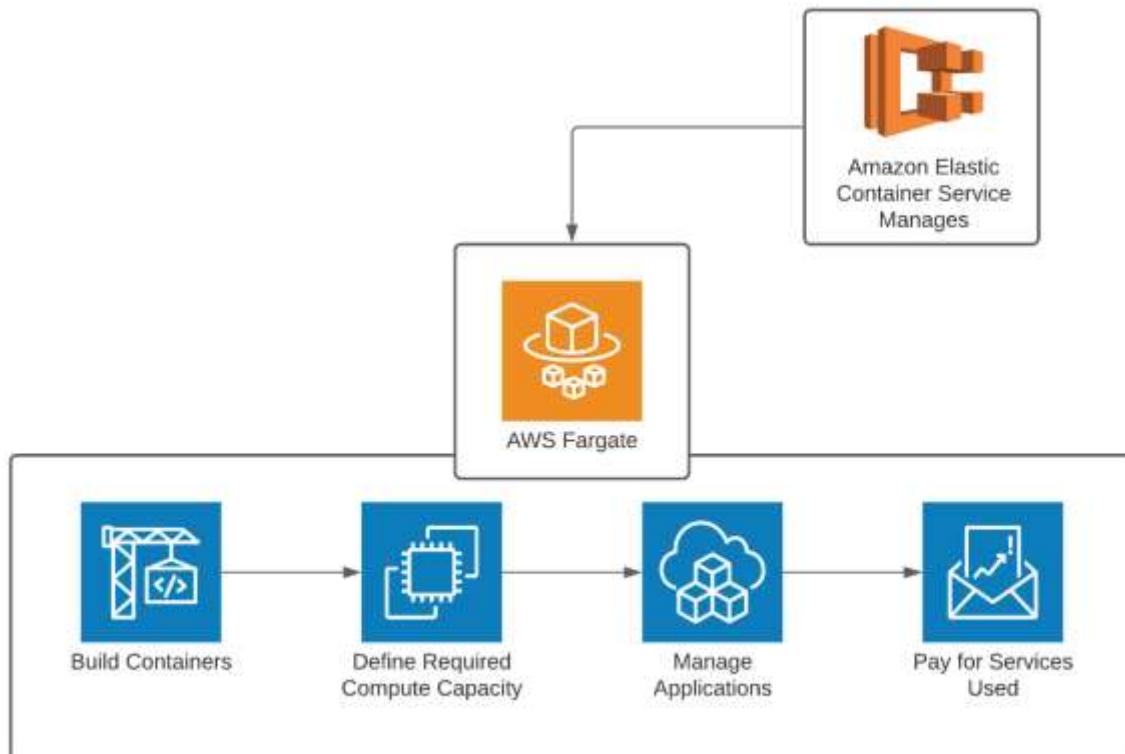
AWS Elastic Container Service (ECS) is a fully managed container management service. ECS is a high-availability (99.99 percent) and high-security container management solution. ECS is deployed in a VPC, which facilitates using AWS security features like network ACLs and security groups. ECS works to manage containers on EC2 or AWS Fargate.

ECS is often used with AWS Fargate, which is a serverless computing engine for containers. This enables ECS and Fargate to create a completely serverless container platform. ECS is used with Fargate, where ECS provides orchestration and management of the Fargate container service. When using ECS with Fargate, there is no need to configure computing instances, install, and manage operating systems, or manage computing instances. Since Fargate is serverless, there is almost limitless scalability.

The diagram below shows how AWS ECS is used to manage containers on the EC2 platform:



The diagram below shows how AWS ECS is used to manage Fargate containers:

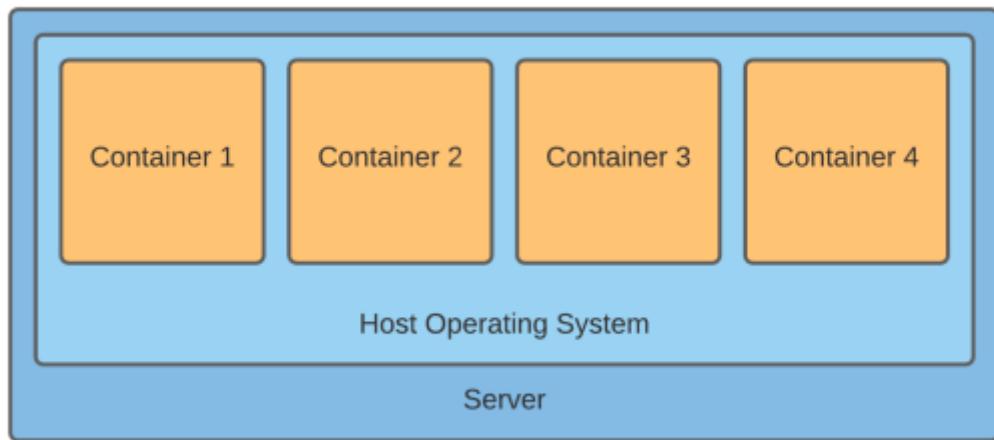


## What Is a Container?

A container is a lightweight, self-contained software package that acts as a modern version of a virtual machine. Virtual machines have a lot of overhead, as they require an entire copy of an operating system to run. By comparison, a container contains just enough of an operating system for the application in the container to run. Since the container runs only a small percentage of operating system packages, a container requires much less memory and CPU resources than a virtual machine. Therefore, many more containers can run at higher performance than virtual machines on a server.<sup>77, 78</sup>

Container images are logically isolated from each other, promoting a secure environment. The host of the containers can support only clients that use the host's operating system. This means Linux containers are hosted on a Linux host and Windows containers are hosted on a Windows host. With the new Windows Subsystem for Linux, Windows systems can now support Linux containers. The reason containers must be on the same operating system is that containers depend on packages from the host operating system.

The diagram below shows how a server can support numerous containers in a logically isolated manner:



## Choosing Between EC2 and Fargate for Containers

As discussed, containers can be hosted on EC2 instances or Fargate. Both approaches have their merit. Choosing the right host is dependent upon an organization's requirements. Please see the list below for guidance on choosing the best container approach:

- If complete control over the device hosting the container is needed, choose EC2.
- If there are specialized requirements and specific customization options required, choose EC2.

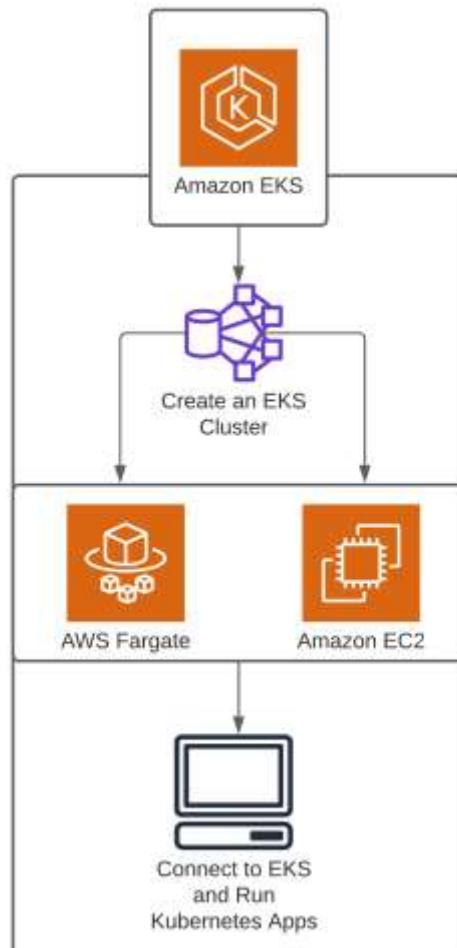
- If a highly scalable platform with minimal management overhead is desired, choose Fargate.
- If near limitless scalability is required, choose Fargate.
- Overall, Fargate is likely a better solution for most customers' needs.

## AWS Elastic Kubernetes Service (EKS)

AWS Elastic Kubernetes Service (EKS) is a fully managed Kubernetes container management service. EKS is a full Kubernetes service, which means Kubernetes containers can be moved to EKS without modification. Kubernetes is an open-source container management service and is the industry standard for containerized applications.<sup>79</sup>

EKS is extremely similar to ECS, but it uses the Kubernetes container platform. Like ECS, EKS is often used with AWS Fargate, creating a completely serverless container platform. EKS can also be used with EC2 in the same manner that ECS can be used with EC2.

The diagram below shows how AWS EKS is used to manage Kubernetes containers:



## **AWS Elastic Beanstalk**

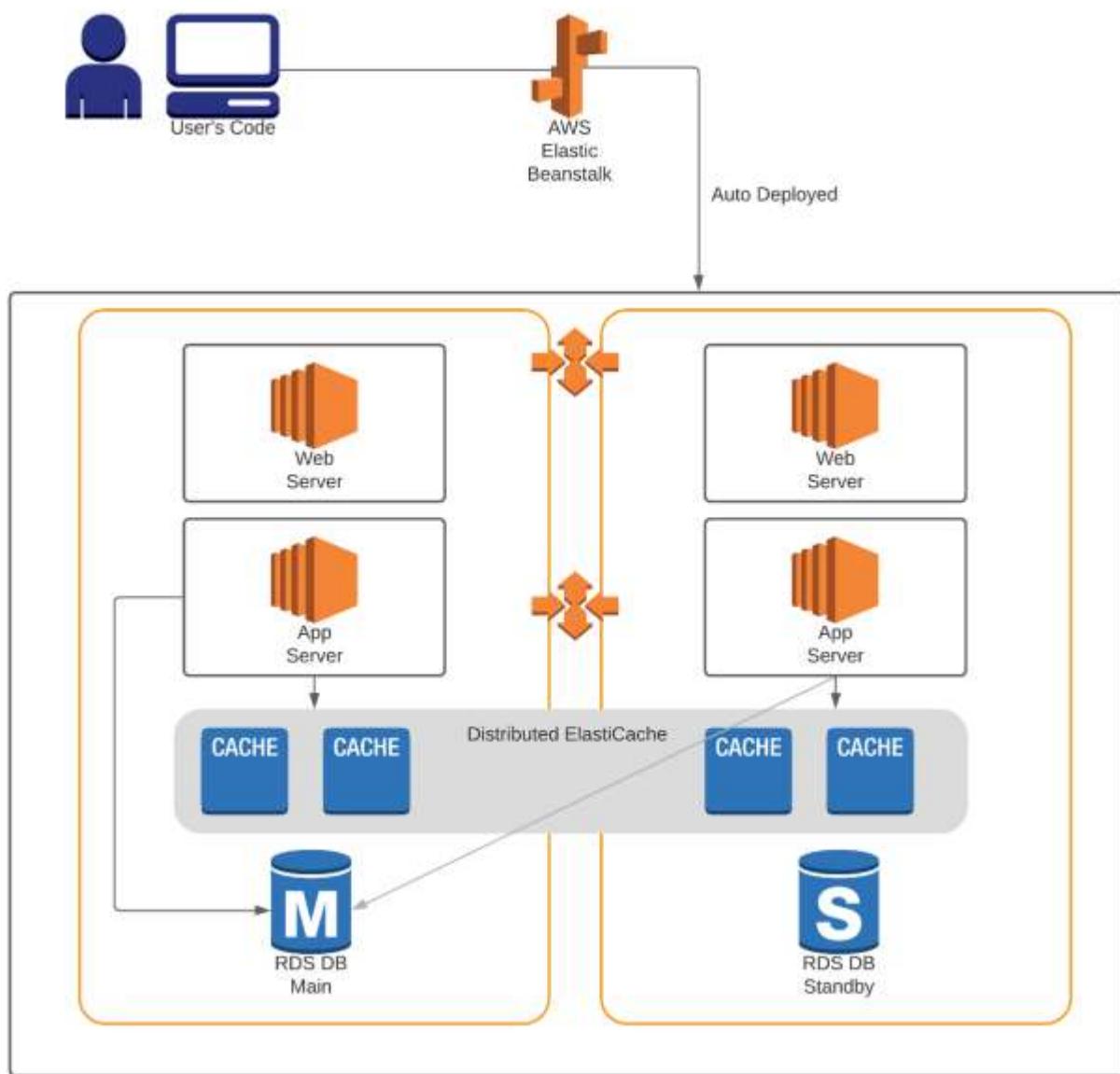
AWS Elastic Beanstalk is a service for provisioning, deploying, and scaling web applications and services. When using Elastic Beanstalk, the administrator simply uploads the code, and Elastic Beanstalk automatically deploys the necessary infrastructure (EC2, containers, load balancers). All infrastructure deployed by Elastic Beanstalk is autoscaling, so it grows with customer requirements. Additionally, infrastructure deployed from Elastic Beanstalk is automatically load balanced.<sup>80</sup>

Elastic Beanstalk provides the necessary tools for web deployment and automatically applies them to the customer, enabling the customer to focus on code development and not infrastructure management. Elastic Beanstalk supports the following programming languages:

- Go
- Java
- .NET
- Node.js
- PHP
- Python
- Ruby

Elastic Beanstalk provisions and manages the environment while allowing the administrator to manage the environment if desired after the computing platform is deployed. Elastic Beanstalk monitors application health and is integrated with CloudWatch logs for performance monitoring.

The diagram below shows AWS Elastic Beanstalk automatically deploying the application environment based upon inputting the organization's code:



## AWS CloudWatch

CloudWatch is a monitoring service to monitor AWS resources and applications deployed on AWS. CloudWatch provides metrics to monitor performance and troubleshoot issues.<sup>81, 82</sup>

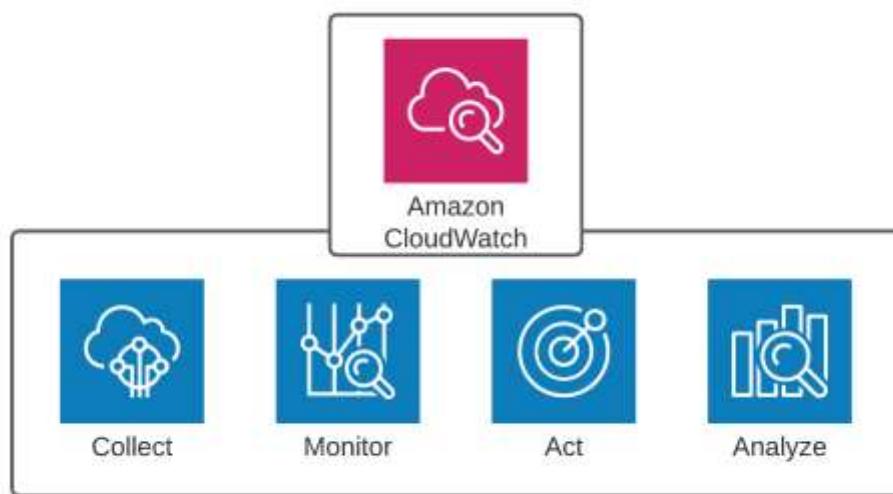
CloudWatch can work with built-in metrics and custom metrics. CloudWatch has several built-in default metrics, which include CPU utilization, disk utilization, and network utilization.

CloudWatch custom metrics can be set up to monitor factors critical to the application's performance, such as memory utilization, API performance, or other metrics.

CloudWatch has a notification service that notifies the organization when certain metrics have been reached. Organizations can set custom metrics and alert notifications. CloudWatch events can also be used to trigger autoscaling, Lambda functions, SNS notifications, actions on containers, and many other functions.

AWS CloudWatch is available in two versions for EC2 instances: basic monitoring and advanced. Basic monitoring automatically provides information every five minutes at no charge. Detailed monitoring provides information every one minute at an additional cost. When using detailed monitoring, it must be enabled at the EC2 instance. As pricing is subject to change, please reference the AWS website for current pricing information for CloudWatch services at <https://aws.amazon.com/cloudwatch/pricing/>.

The diagram below shows AWS CloudWatch can be used for VPC monitoring and optimization:



## AWS Config

AWS Config is a service that enables assessment, auditing, and evaluation of configurations in AWS. AWS Config provides a means to see what changes were made and who made these changes in the VPC. AWS Config provides monitoring of any changes that have occurred. When a change is made, AWS config can send an SNS alert to systems administrators.

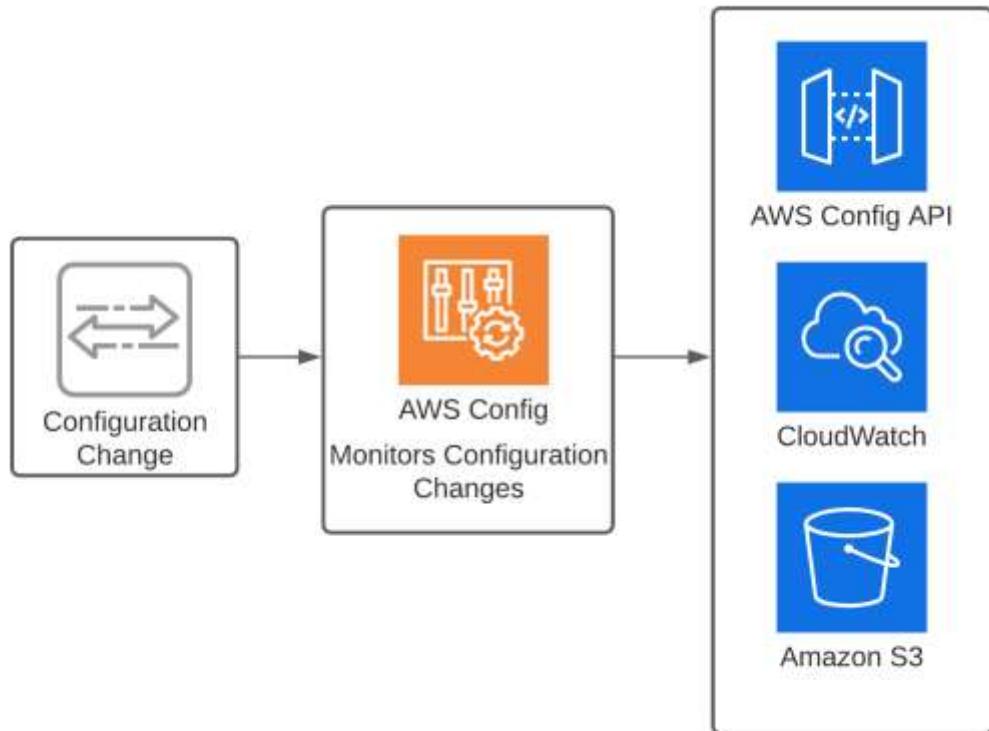
AWS Config also provides constant monitoring of configurations and checks configurations against an organization's policies. If a change is made that violates the organization's policy, an SNS alert is sent, and a CloudWatch event will occur.

AWS Config provides a means to assist with change management. Config can track relationships between resources, so if changes are made, it will be easy to determine which systems will be affected by the changes. AWS Config can help with troubleshooting, as it can integrate with CloudTrail and track changes made. In this manner, if a configuration change causes a problem, AWS config will show which changes should be reverted to go back to a fully functional environment.

## How It Works

1. A configuration change is made.
2. AWS Config notes the change and records the change in a consistent format.
3. AWS Config will then check the change against an organization's policies.
4. AWS Config will notify the services that can notify the system administrator of configuration changes. Notifications can be sent as a CloudWatch event, SNS, or other AWS service.

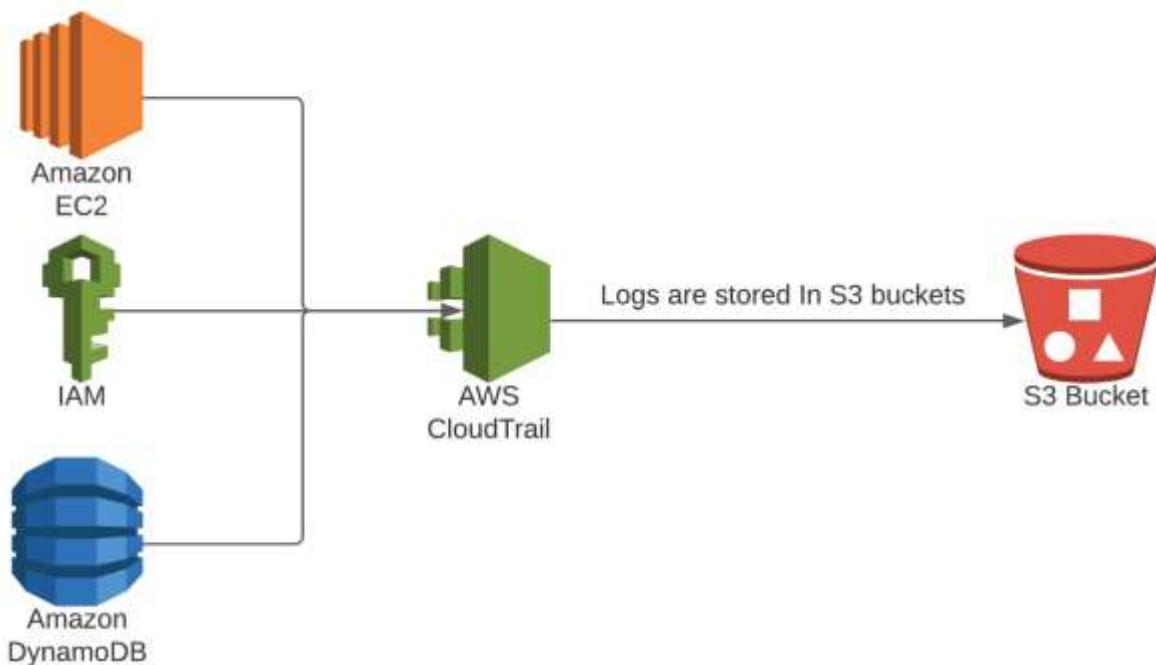
The diagram below shows how AWS Config monitors for configuration changes:



## AWS CloudTrail

CloudTrail is an AWS service that assists with the auditing process. CloudTrail provides an audit log that supports risk management and compliance endeavors. CloudTrail provides a means to track changes made to an AWS account by a user, role, or AWS service.<sup>83</sup>

The diagram below shows how AWS CloudTrail is used for logging and auditing of an organization's VPC:



CloudTrail is enabled when the AWS account is created. To start using CloudTrail, create a trail with the CloudWatch console, CLI, or CloudTrail API. CloudTrail records events, and these events are visible in the CloudTrail console under Event History. The CloudTrail event history shows events that have occurred in the last ninety days. Additionally, CloudTrail can be configured to store logs in an S3 bucket for long-term storage.

Two types of CloudTrail trails can be created, and they can be seen below:

### **Local Trail**

A *local trail* is tied to a single region trail. CloudTrail logs are put into a single bucket. This is the default option when CloudTrail is configured by the CLI or API.

## A Trail that Applies to All Regions

A CloudTrail can be set up to monitor all regions. This provides the most comprehensive logging and auditing options available. This provides a record of all events that occur inside an organization's infrastructure. This type of trail can help correlate events across an organization's global infrastructure and provide insight on fixing problems.

## CloudTrail and Compliance Audits

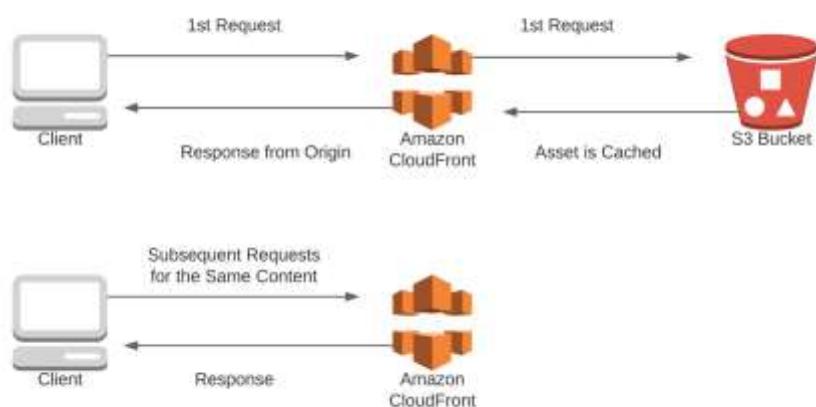
Many industries are highly regulated and have specialized data storage, protection, privacy, and auditing requirements. CloudTrail can be very beneficial in highly regulated environments, as CloudTrail can help show how the organization is adhering to legal requirements.

## AWS CloudFront

AWS CloudFront is an Amazon-branded content delivery network. CloudFront can dramatically improve web hosting and is integrated with numerous AWS services. Effectively, CloudFront is a network of caching servers spread throughout the world. When a request is made to a webpage, its location is determined, and the web request is sent to the closest CloudFront server. Local CloudFront servers cache website content and speed the delivery to remote locations throughout the world.<sup>84, 85, 86</sup> The caching server works in the following manner:

1. The web request is sent to the CloudFront caching server.
2. If the website has been requested prior to the cache timeout, the content is sent straight to the user.
3. If the website data is not stored on the cache, the cache reaches out to the original website.
4. When the data is received, the information is stored on the cache until the cache's expiration, and the data is sent to the requestor.

The diagram below shows how CloudFront caching can be used to improve the scalability and performance of web applications:



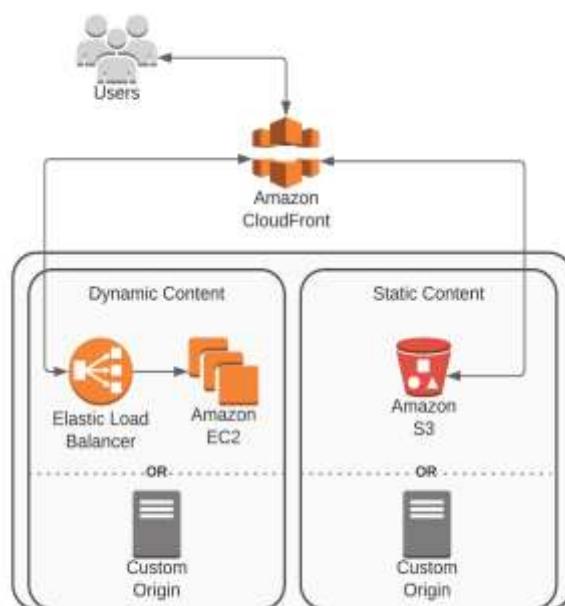
Caching can assist with website scalability and performance by offloading frequent requests to the cache instead of the actual website. Caching is very helpful for frequently requested content. If the content is very dynamic and user requests are all for new data, then the caching server will not help to improve performance or scalability.

CloudFront integrates with numerous AWS services, including S3, EC2, Elastic Load Balancers, and Route 53. CloudFront is typically used as a front end to static websites stored on S3. CloudFront can also be a front end to an EC2-based website as long as an Elastic Load Balancer is part of the architecture.

CloudFront can help website performance through the following mechanisms:

- Cached content – The request does not need to go to the web server because it is cached.
- Global reach – There are over 217 points of presence for CloudFront. So CloudFront can get content much closer to the user's location.
- Routing efficiency – When CloudFront is used, requests that go to the original source (i.e., S3 bucket) traverse the AWS backbone and not the public internet. Therefore, performance can be enhanced, as AWS can manage their network for lower latency than when traversing an unknown number of internet service providers.
- Persistent connections – CloudFront maintains connections to the source. This minimizes the number of connections required on the web server, which reduces server load.

The diagram below shows how AWS CloudFront can be used to enhance the performance of a web application with static and dynamic content:



CloudFront can also make a significant impact on an organization's security.

### **CloudFront Integrates with Web Application Firewall (WAF)**

- WAF adds firewalling capabilities to protect against common web attacks.

### **CloudFront Can Help Prevent Distributed Denial of Service Attacks**

- CloudFront distributes requests through multiple points of presence.
- CloudFront forwards only legitimate http/https requests to the server that aren't already in the cache. This means the attacker cannot launch a DDoS by sending a large number of invalid requests to the server.
- AWS Shield Standard is included with CloudFront to provide additional layers of DDoS protection.

### **CloudFront Can Provide Encryption in Transit**

- CloudFront can enforce SSL/TLS protocols.
- CloudFront integrates with the AWS Certificate Manager.
- CloudFront supports Server Name Identification (SNI) as well as custom certificates.

### **Tuning CloudFront**

CloudFront is highly tunable to meet an organization's needs. CloudFront can be modified by changing the *time to live* (TTL) for objects in the cache. The minimum, maximum, and default TTL for objects are configurable options. If problems occur in the cache, it is possible to clear the cache. Clearing the cache is performed via the API or with the command line with the following command structure:

- `aws cloudfront create-invalidation --distribution-id distribution_ID --paths "/*".`

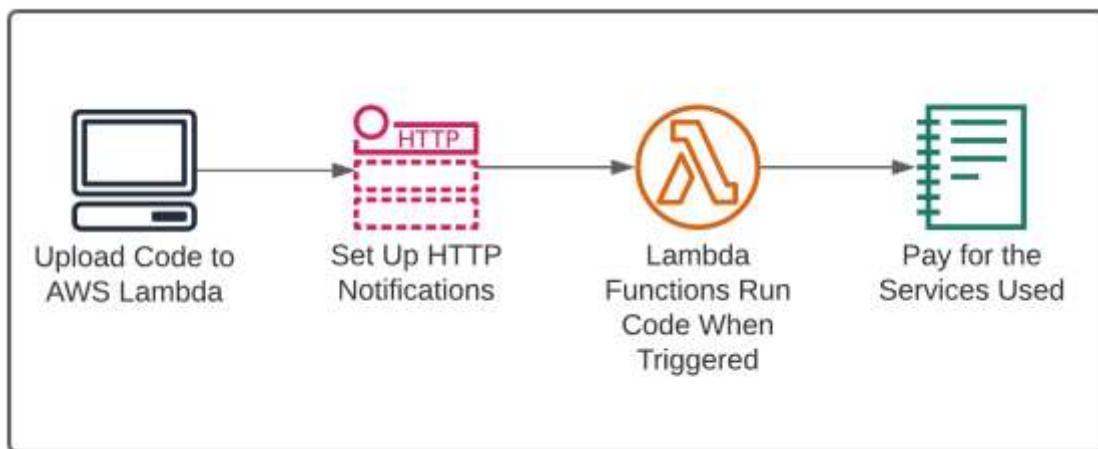
### **AWS Lambda**

AWS Lambda is a serverless computing service to enable automation across an organization's infrastructure. AWS Lambda is useful in many situations where automation can increase the efficiency of technology by decreasing manual intervention. Some examples of automation with the Lambda platform include processing data across multiple systems, patching operating systems, and remediation of security events.<sup>87</sup>

Using Lambda functions is much simpler than deploying custom automation applications. To get started using Lambda, just upload the code for the Lambda function. Since Lambda is serverless, there is no need to manage servers and operating systems. Lambda supports the following programming languages:

- C#
- Go
- Java
- Node.js
- Python
- PowerShell
- Ruby

The diagram below shows how Lambda functions are deployed on the AWS platform:

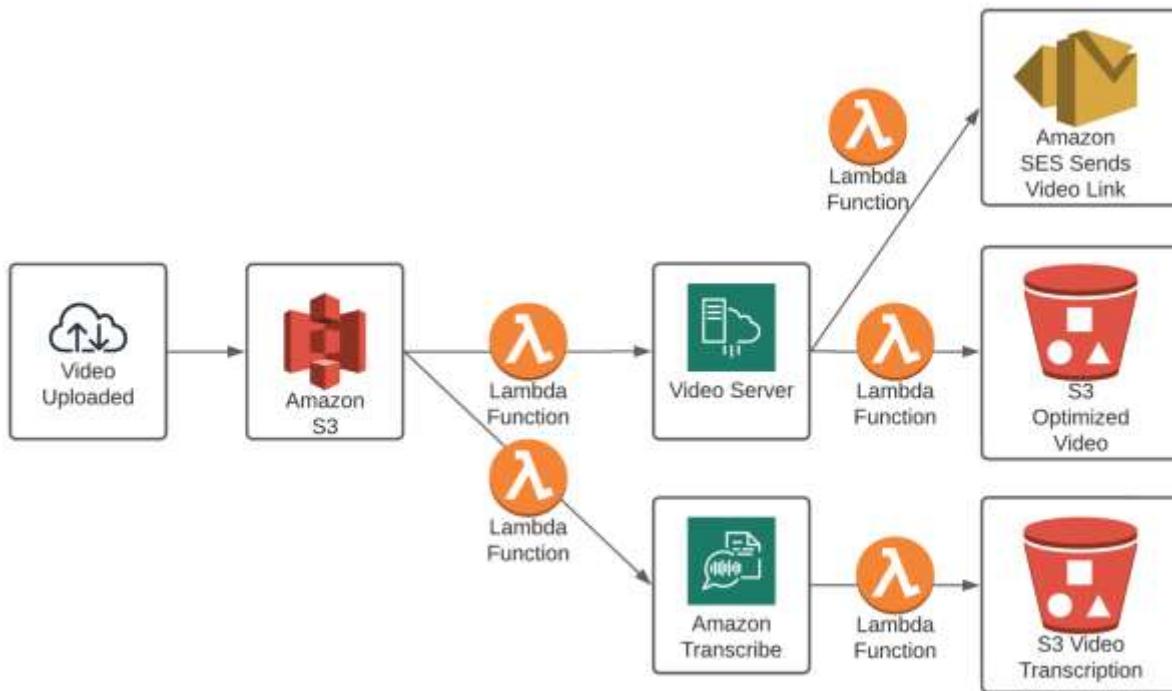


Lambda is stateless, meaning that once the function is performed, the function has been completed. If additional functions are needed, it will be necessary to set up additional Lambda functions. Lambda functions can be run in response to events in a VPC. A sample video optimization using Lambda can be seen below:

In this example, there is a video-processing application that optimizes and transcribes a video after being uploaded into S3 using Lambda functions.

1. A new video is uploaded into S3.
2. When S3 detects a new video, a Lambda function is triggered to inform the transcription application that a new video is ready to be transcribed.
3. The video gets transcribed.
4. After the video is transcribed, a Lambda function triggers AWS SNS/SES.
5. An email is then sent to the customer, telling them their video is ready for download.

The diagram below shows how AWS Lambda can be used to automate workflows on the AWS platform:

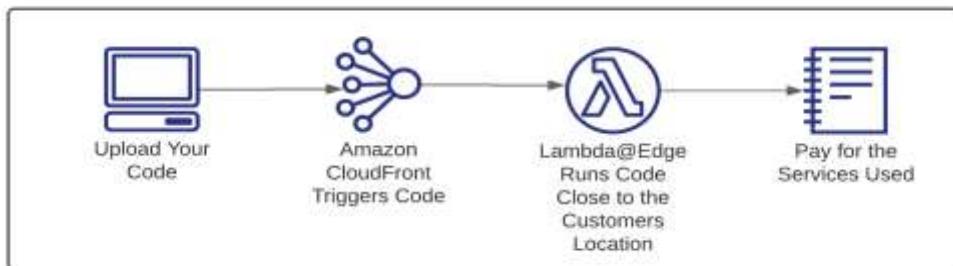


## AWS Lambda@Edge

AWS Lambda@Edge is a serverless CloudFront feature. Lambda@Edge works with the CloudFront content delivery network. Lambda@Edge enables the content to be closer to the customer and achieve higher performance. Additionally, Lambda@Edge allows for running Lambda functions closer to the user. Setting up is a matter of the following:

1. Upload code to a Lambda function.
2. Set up the Lambda function to be triggered by CloudFront.
3. The Lambda@Edge code is run where your users are located.

The diagram below shows how AWS Lambda@Edge can be used to perform functions close to the customer's location:



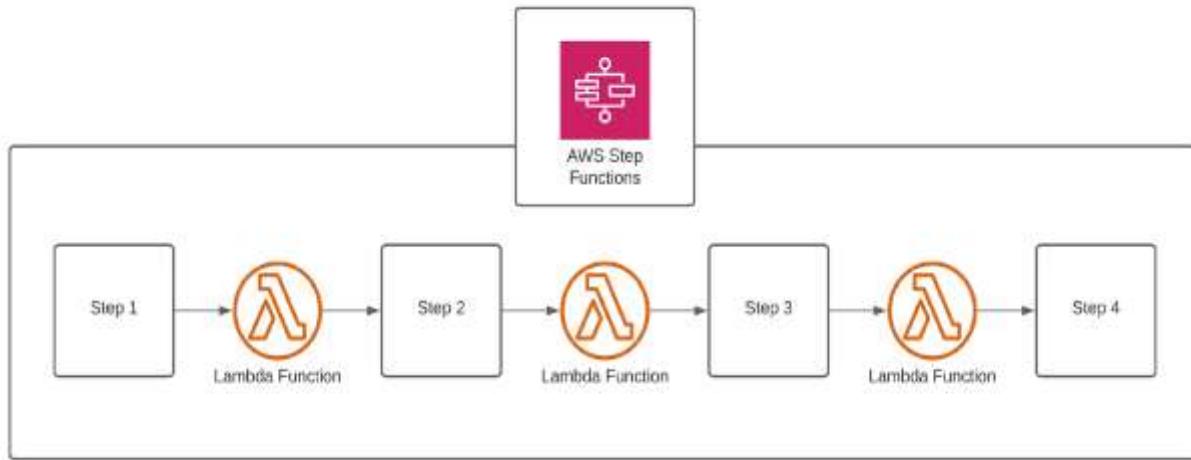
## AWS Step Functions

AWS Step Functions provide a means to sequence multiple Lambda functions. AWS Step Functions are serverless, just like AWS Lambda. Step Functions help facilitate a multiple-step workflow. For example, if a file is uploaded to S3, a Lambda function can send the file to an EC2 instance for processing. After processing is completed, a Lambda function can send the processed data to DynamoDB.<sup>88</sup>

Step Functions are simple to design. The process to set up AWS Step Functions is as follows:

1. Design the steps of the application.
2. Create the individual Lambda functions.
3. Configure the workflow in the Step Functions.
4. Connect the workflow components to individual tasks specified in the Lambda functions.
5. Execute the Step Functions in normal use.
6. Optimize and evolve the Lambda and Step Functions as needs change.

The diagram below shows how AWS Step Functions can be used to automate multistep workflows:



## AWS Rekognition

Rekognition is a way to analyze videos and images using machine learning. Rekognition can be helpful in analyzing video content.<sup>89</sup> Some examples of using AWS Rekognition are:

- Identification of individuals in a video.
- Analysis of emotional state based upon facial expressions.
- Identification of unwanted content in videos.
- Ability to search a video for a certain person.
- Labeling of images, logos, or objects found inside of a video.
- Detecting anomalies and unwanted content within an organization's videos.

## AWS CloudFormation

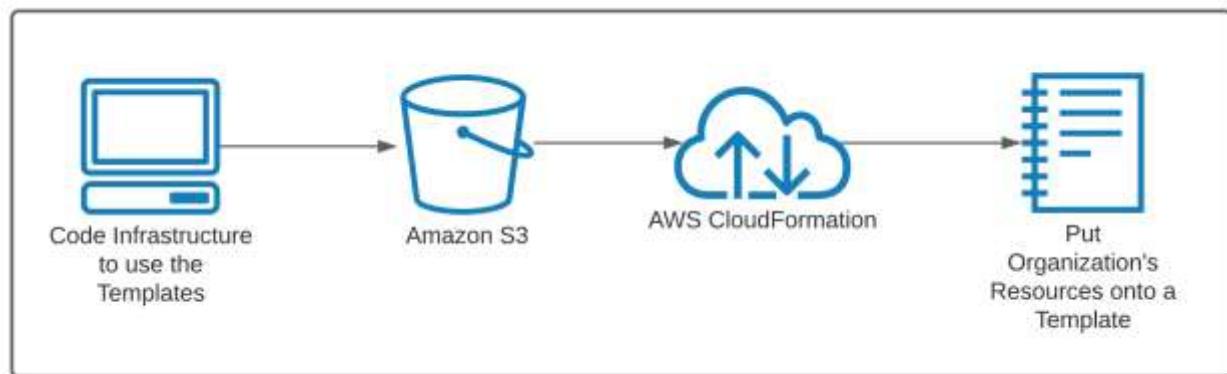
CloudFormation is a means to template known good configurations of an organization's services. For example, if an organization has a common application that requires the configuration of several servers with specific patches, a CloudFormation template can make sure all new servers are properly configured. CloudFormation therefore helps you provision applications in a safe and repeatable manner.

CloudFormation templates can be made with simple text files or via supported programming languages. AWS Cloud Formation templates are available through a multitude of options. CloudFront can deploy your templates across your infrastructure by rebuilding applications or building new ones.<sup>90</sup>

### How It Works

1. Develop the code for the organization's infrastructure.
2. The code can be made from a template or from scratch in either JSON or YAML format.
3. Store your code either locally or on S3.
4. Use CloudFormation with the customized code either with the CloudFormation console, CLI, or API.
5. CloudFront will then provision your systems.

The diagram below shows an example of CloudFormation used to template an organization's services.

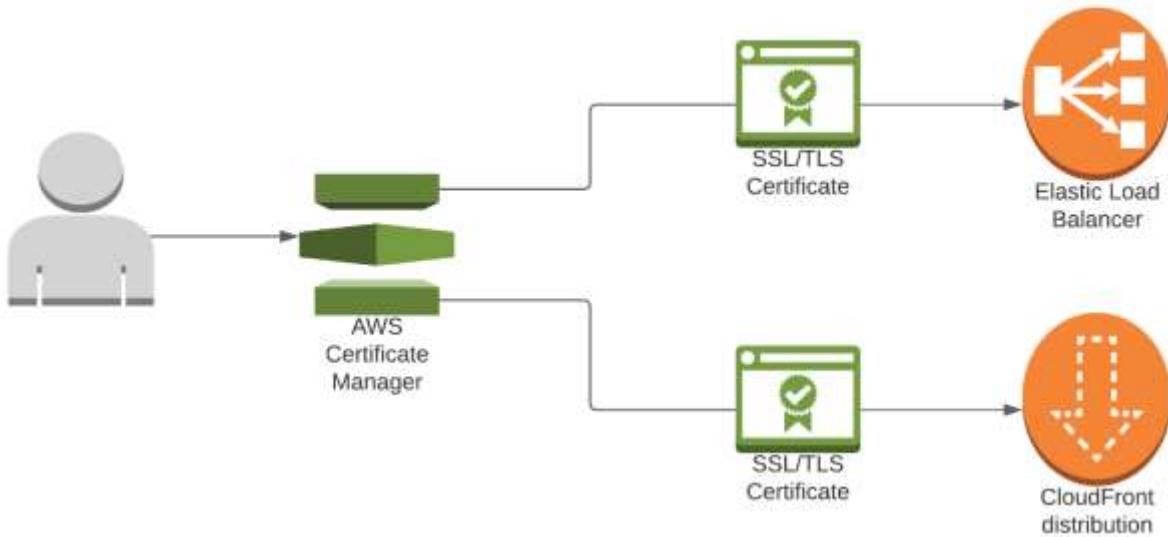


## AWS Certificate Manager (ACM)

AWS Certificate Manager is a service for SSL/TLS certificates. AWS Certificate Manager makes it easy to provision, manage, and deploy certificates either publicly or privately. It allows users to deploy certificates on AWS resources quickly and effectively. It provides free public and private certificates to AWS services such as ELBs and API Gateways.

AWS Certificate Manager provides a means to obtain certificates to websites, promoting safe and secure connections. AWS Certificate Manager is a platform to manage all of your certificates in the AWS cloud centrally.<sup>91, 92</sup> There are two options when deploying certificates: the default certificate manager and ACM private CA.

The diagram below shows how the certificate manager can be used to obtain SSL/TLS certificates:



### **Default AWS Certificate Manager (ACM)**

AWS Certificate Manager is for customers who want encryption and security using TLS. The certificates are deployed through ELBs, CloudFront, and API Gateways to make communication secure.

### **ACM Private CA**

Private CA is for communication within the organization. Private CA can issue certificates for users, computers, applications, services, servers, and more devices throughout the organization. Private CA certificates are for use internally and not on the internet. Private certificates also come at an additional cost.

## **Chapter 10 - DevOps**

DevOps is a hybrid of software development and operations combined. DevOps is a set of practices designed to increase software release speed and quality. Additionally, DevOps practices are used to automate deployments in the form of infrastructure as code. DevOps is a collaborative process designed to deliver code changes more frequently and reliably by breaking the cycle into manageable segments.

DevOps uses a continuous integration and continuous delivery (CI/CD) method, which refers to automating all stages of the software development process. By enforcing automation in all phases of software development, companies can better enhance business goals and customer needs.

DevOps is not architecture. But cloud architects and solutions architects should be familiar with DevOps practices. This section will cover AWS DevOps tools.

AWS provides native CI/CD tools, including AWS CodePipeline, AWS CodeCommit, AWS CodeBuild, and AWS CodeDeploy.

### **AWS CodePipeline**

AWS CodePipeline is a continuous delivery service that quickly automates the software release process without upfront costs or long-term commitments. CodePipeline can integrate with third-party services like GitHub and many Amazon services, including Amazon S3, AWS Code Commit, and Amazon ECR.

The software release process can be modeled using the CLI, CloudFormation, and SDKs. CodePipeline can deploy software faster, increasing cost savings that come from a more efficient workflow.

### **AWS CodeCommit**

AWS CodeCommit is a managed source code management (SCM) service that securely hosts Git repositories. AWS CodeCommit provides a highly available and scalable architecture that supports all Git tools and plugins. Source code is encrypted in transit and at rest.

SCM is a critical component of the DevOps lifecycle, enabling teams to seamlessly track and improve code modifications during the entire development process. Using a fully-managed repository improves performance by eliminating the need to build the underlying infrastructure.

## **AWS CodeBuild**

AWS CodeBuild is a predesigned build environment where customers can test code with continuous scaling (on a pay-per-minute basis). Compatible programming languages and automation tools include but are not limited to Python, Go, Apache Maven, and Gradle. With the help of continuous integration tools, the creation of applications is faster and more consistent (or reliable). Sustainable software development improves quality and reduces risk.

AWS CodeBuild can be used on the AWS CodePipeline console to run tests and produce software packages. AWS CodeBuild can be used on the AWS Command Line Interface (AWS CLI) or the AWS SDKs.

## **AWS CodeDeploy**

AWS CodeDeploy is a service that helps automate code and software deployments quickly by eliminating errors from manual operations. CodeDeploy can be used on premises and with Amazon EC2, AWS Fargate, and AWS Lambda. Additionally, CodeDeploy can scale to match software deployment needs.

CodeDeploy can track the status of application deployments and provide detailed reports via the AWS Management Console or the AWS Command Line Interface (CLI). Push notifications can be created to view live updates of deployments.

These are the AWS native CI/CD tools. There are other tools used by DevOps engineers. A comprehensive list of DevOps tools and what those tools perform can be seen at the link below:

<https://www.gentelli.com/thought-leadership/insights/devops-tools>

## **Chapter 11 - Cost Optimization**

### **Financial Differences Between Traditional Data Centers and Cloud Computing**

Migrating from the data center to the cloud can have a profound effect on an organization's technology costs. In most scenarios a move to the cloud will have a lower total cost of ownership than with traditional data centers. This is because traditional data centers have high costs to purchase equipment, build data centers, and manage staff of the data center and IT systems. These are heavy capital expenditures (CAPEX) with a moderate degree of operational expenses (OPEX). The list below shows the typical capital and operational expenses with traditional data centers. With a traditional data center, the organization purchases the following equipment (CAPEX):

- Physical servers
- Routers
- Switches
- Firewalls
- Load balancers
- Racks
- Power distribution units
- Generators
- Uninterruptible power supplies (UPS)
- Data center cooling

Additionally, there are moderate OPEX costs associated with traditional data centers. The primary OPEX costs are:

- Large IT staff
- Electric bills
- WAN connections
- Internet connections

### **Optimizing Technology Costs on the AWS Cloud**

Moving to the cloud changes the cost structure completely. In the cloud computing environment, there is minimal to purchase, so CAPEX is very low. However, the organization pays every time cloud services are used, and usage can get quite expensive. Therefore, with cloud computing, while CAPEX is low, OPEX is high. Generally speaking, moving to the cloud will have a lower total cost of ownership than with traditional data centers. Furthermore, the better the cloud architecture is designed, the lower the total costs for cloud computing. There are five steps to lowering the cost of cloud computing:

## **Step One**

- Provision only the resources needed, as you pay for all resources used.
- Monitor your systems to get insights into the proper size of compute and network resources.

## **Step Two**

- Properly size resources.
- Plan and size equipment based upon average use and not peak usage. Cloud computing allows autoscaling, so you don't need to overprovision in advance as in a traditional data center.
- Leverage means to decouple systems in the architecture when possible. For example, an SQS queue can dramatically decrease the spikes in the system, allowing for less expensive resources to be used in the architecture.

## **Step Three**

- Purchase the right computing platform. Know when it's best to use On-Demand Instances, Reserved Instances, and Spot Instances, as they can have a dramatic effect on costs. These options are discussed below:

### **On-Demand Instances**

- Are the most expensive at a pure pricing level.
- Provide instant access to computing power.
- Are ideal when you don't know the exact amount of computing power required but need flexible options.
- Are highly reliable, in that on-demand instances won't be terminated like a spot instance when AWS pricing changes.
- Promote scalability by facilitating autoscaling.
- Ideal for situations when you have a temporary application or when you don't know how long the application will be used.

### **Spot Instances**

- Are the lowest-cost option for computing power within the AWS platform.
- Let the customer bid for unused computing power in AWS.
- Have constant pricing changes based upon AWS capacity and current bids by other organizations using the AWS platform.
- Are not for critical workloads, as they can be shut down by AWS if the price for spot instances changes.
- Are optimal for batch jobs that are not critical and have a means to restart the processing if the system is shut down.

## **Reserved Instances**

- Offer discounted service when an organization makes a guaranteed purchase of computing capacity for a period of time.
- Provide pricing based on a contract—the longer the contract, the greater the discount.
- Are ideal for an application with a known capacity and a known duration for which the organization will use the computing platform.

There are three types of reserved instances.

### **Standard Reserved Instances**

- Are the lowest-cost option.
- Are optimal for a long-running application.

### **Convertible Reserved Instances**

- Convertible reserved instances are reserved instances with the flexibility to change the size of computing instances.
- With convertible reserved instances, an organization purchases a computing platform based upon need. If the organization needs to resize its computing instances, it has the flexibility to change.
- Convertible reserved instances offer flexibility but with higher costs than standard reserved instances.

### **Scheduled Reserved Instances**

- Scheduled reserved instances are reserved instances for computing platforms that are used by organizations with frequent and periodic needs for computing power.
- Scheduled reserved instances are optimal for critical workloads that happen periodically, for example, a batch job that needs to run uninterrupted every weekend for forty-eight hours straight.
- Scheduled reserved instances cost more than standard reserved instances, but they enable discounted pricing for periodic workloads.

To optimize costs for computing instances, purchase the computing platform that will be most cost-effective based on the organization's needs. Costs can be best optimized by using a combination of on-demand, reserved, and spot instances based on an organization's needs.

## **Step Four**

- Leverage managed services and serverless options to minimize time and costs spent managing computing instances.

## **Step Five**

Step five is about managing data transfer costs.

- AWS charges for data sent between regions. Being mindful of cross-region data charges can make a big difference in an organization's costs. S3 cross-region replication can assist with data transfer costs when there is a large amount of data being requested across regions.
- Leverage CloudFront to reduce data transfer costs between regions, as content will be cached and served locally.
- Use the right connection to AWS. Data in and out of a VPC over a direct connection can be lower cost than VPN if large amounts of data are being transferred.

## **AWS Budgets**

Another means to control costs is to create a budget and stick to that budget. AWS budgets help the organization stick to a budget with budget notifications.<sup>93</sup>

### **How AWS Budgets Work**

- The organization creates a budget and sets custom alerts.
- The budget is created in the AWS Management Console or within the AWS Billing Console.
- When an organization gets close to exceeding its budget, an alert is sent.

This can ensure that an organization adheres to its budget. Additionally, the budget alarms can help an organization plan for future optimizations of their network. For example, an organization may find that due to use, reserved instances may enable large cost savings. This enables the organization to optimize cloud computing expenses.

The diagram below shows how AWS budgets can be used to help an organization manage its AWS cloud computing expenses.



## **AWS Trusted Advisor**

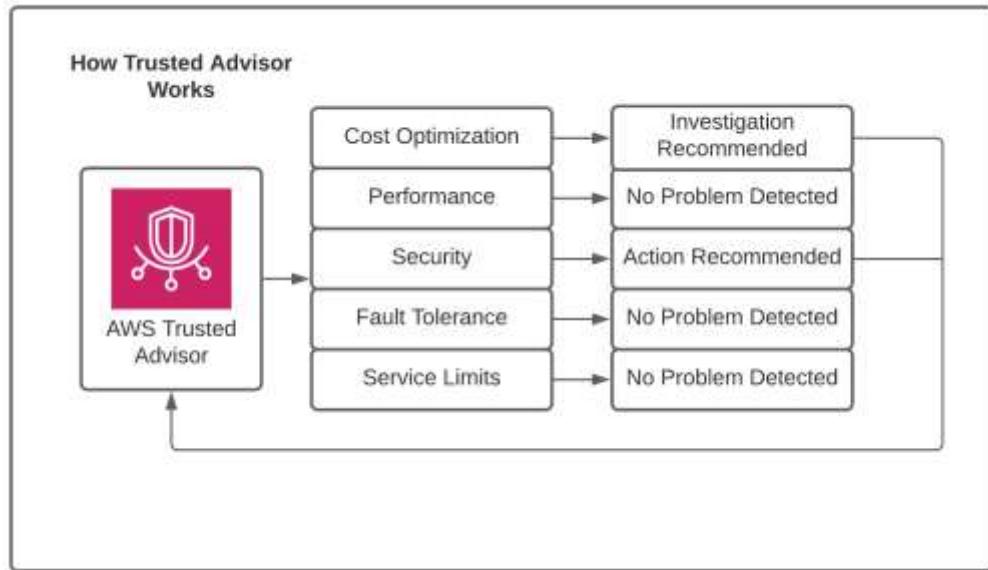
AWS Trusted Advisor is an online tool to help an organization optimize its spending on the AWS platform.<sup>94</sup>

### **How AWS Trusted Advisor Works**

1. AWS Trusted Advisor scans and evaluates an organization's infrastructure.
2. AWS Trusted Advisor compares an organization's infrastructure to AWS best practices and provides recommendations. These recommendations can improve performance, security, availability, and system costs.
3. The organization evaluates the Trusted Advisor recommendations.
4. The organization implements the appropriate recommendations from AWS Trusted Advisor.
5. This should reduce costs and/or improve system performance.

There are two versions of Trusted Advisor for clients, the Developer Support Plans and Business Support Plans. Organizations with AWS Basic and Developer Support Plans have access to six security checks and fifty service limit checks with Trusted Advisor. Organizations with Business Support Plans or AWS Enterprise Support Plans receive fifteen Trusted Advisor checks (fourteen cost optimization, seventeen security, twenty-four fault tolerance, ten performance, and fifty service limits).

The diagram below shows how AWS Trusted Advisor is used to help an organization optimize its AWS infrastructure:



## **Chapter 12 - Building High-Availability Architectures**

### **What Is Availability?**

Availability refers to the service being available for use when you need it. A high-availability infrastructure is highly likely to be ready when needed. Designing for high availability can become extremely expensive, depending upon the availability required.<sup>95</sup>

In general, there are four levels of availability:

- 99.0 percent
- 99.9 percent
- 99.99 percent
- 99.999 percent

Most people would consider levels of availability of 99.9 percent or greater to be a highly available network. Many organizations require much higher levels of availability. For example, service providers, banks, health care organizations, and other organizations that are completely dependent upon technology—with serious consequences if the systems are not operational—require 99.999 percent or greater availability.

The diagram below shows the typical availability metrics and associated downtime per year:

<b>Availability Level</b>	<b>Maximum Downtime Per Year</b>
99.000%	3.65 days
99.900%	8.76 hours
99.990%	52.6 minutes
99.999%	5.25 minutes

### **Building a High-Availability Network**

Building a high-availability network is based upon the key tenant of no single points of failure. This means complete redundancy is required in all aspects of the computing environment. Necessary redundant services are as follows:

- Redundant power
- Redundant cooling for servers
- Redundant network connections
- Redundant service providers
- Redundant routers
- Redundant switches

- Redundant servers
- Redundant load balancers
- Redundant DNS
- Redundant storage
- Redundant locations
- Redundant applications, i.e., databases

## **Change Management**

A strong change management program is required for high-availability systems. Configuration changes can have a major impact on system performance, especially if configuration mistakes occur. Therefore, prior to making any changes across an organization's systems, all stakeholders need to be notified of prospective changes. All stakeholders need to evaluate that any changes made will not affect the systems they manage. Additionally, all stakeholders need to agree on a time for configuration changes. Configuration changes should be made at a time when the system is minimally used—ideally a time when user access and system utilization are at their lowest levels.

## **High Availability in the Cloud**

Building a high-availability system in the cloud is much simpler than with the traditional data center. This is because AWS maintains the key elements of high availability in the architecture, including:

- Redundant power
- Redundant cooling
- Redundant connections to the internet and across the backbone
- Redundant routers and switches

Since AWS natively performs many of the parts of a high-availability architecture, only a subset of redundancy is required, and those elements can be seen below. Whenever possible, place the platform in multiple availability zones. These elements include the following:

- EC2 compute instances
- Databases
- Elastic Load Balancers
- DNS or Route 53
- NAT gateways
- Storage

Another key component of high-availability design is the connections from the organization to the AWS VPC. For most clients this will include a direct connection to AWS and a VPN backup. Organizations requiring even higher levels of availability and performance might have a primary direct connection, a backup direction connection, and a VPN backup to the direct connections.

Ideally all connections to AWS are across multiple service providers, so that if a service provider were to have an outage, the backup connections would remain intact. Furthermore, on the customer end, redundant connections should be placed on redundant routers.

## **High Availability Requires High Security**

High-availability environments require a strong security posture. A strong security posture is required because if security is compromised, it can have a major impact on system performance and availability. Key components of high security for high-availability architecture are below:

### **Principle of Least Privilege**

- Allow users access to only the systems they need to perform their job.
- Use strong IAM policies to limit the users to the minimum services necessary.

### **Containing Problems**

- Limit blast radius of an application by using AWS Organizations.

### **Keep Unwanted Traffic Out**

- Keep unwanted network traffic out using network ACLs.
- Configure services with security groups so that only desired traffic is allowed.
- Use Amazon WAF for firewall, AWS Shield for DDoS, and IDS/IPS for intrusion protection.

### **Physical Security**

- Equipment accessing the AWS network should be secured to prevent unauthorized access.

### **Passwords and Authentication**

- Allow only strong passwords to be used and rotate them frequently.
- Use multifactor authentication.
- Use temporary passwords or tokens whenever possible.

### **Data Privacy**

- Encrypt data to ensure its privacy from unauthorized users.
- Encrypt data both at rest and in transit.

## **Logging and Monitoring**

There should be constant monitoring of the systems for the following:

- System alerts
- Security breaches
- Usage
- Performance

## **Other Essential Security Components**

- Disable all unused and unnecessary services on systems.
- Allow only approved services to be used.
- Template known configurations with CloudFormation to be sure new systems meet security requirements.

## **Additional Means to Increase Availability**

There are some additional methods for increasing availability. Using systems and services that are designed to enhance availability is one method. DNS and load balancers with health checks are purposely designed to enhance availability and performance. Elasticity and autoscaling features help to make sure applications are always usable and don't become unstable with high demand.

Availability can also be increased by decoupling applications using services like SQS. Decoupling components of the architecture can promote system integrity and availability by enabling functionality when one or more parts of the system are unavailable. For example, adding an SQS queue in front of a database can keep the system functioning and not lose messages during a database outage. Additionally, use services designed to lower overall system load so the systems will be available and not busy when needed. An example is using redundant caching in front of web servers.



## **Chapter 13 - AWS Labs – Getting Hands-On**

In this section we showcase some hands-on training. Labs will follow the flow of the chapters in this book whenever possible.

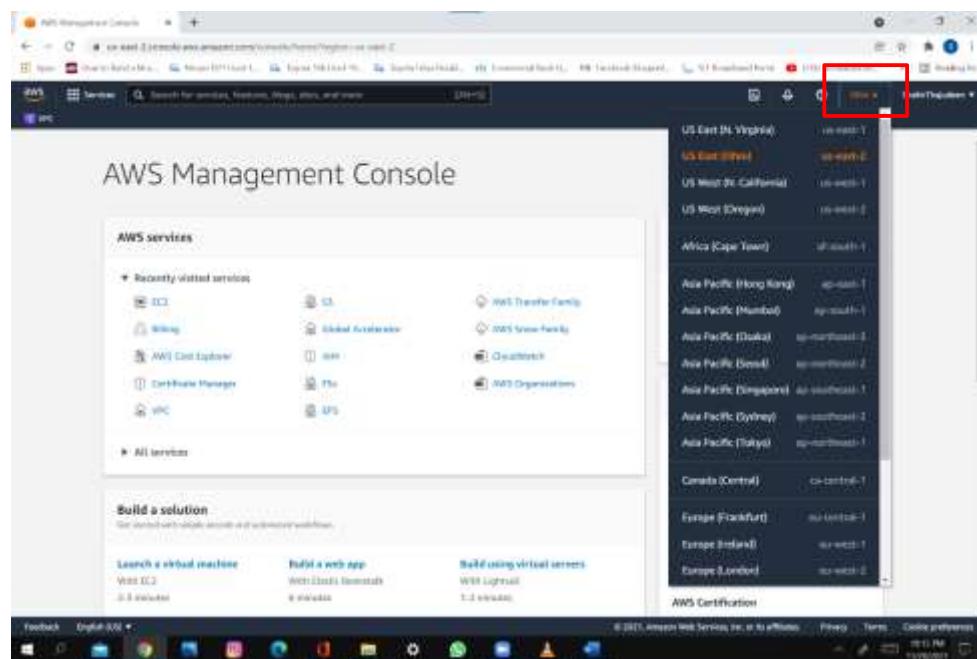
## Chapter 1 Labs

Creating a virtual machine (EC2 instance) and remotely accessing that virtual machine with SSH.

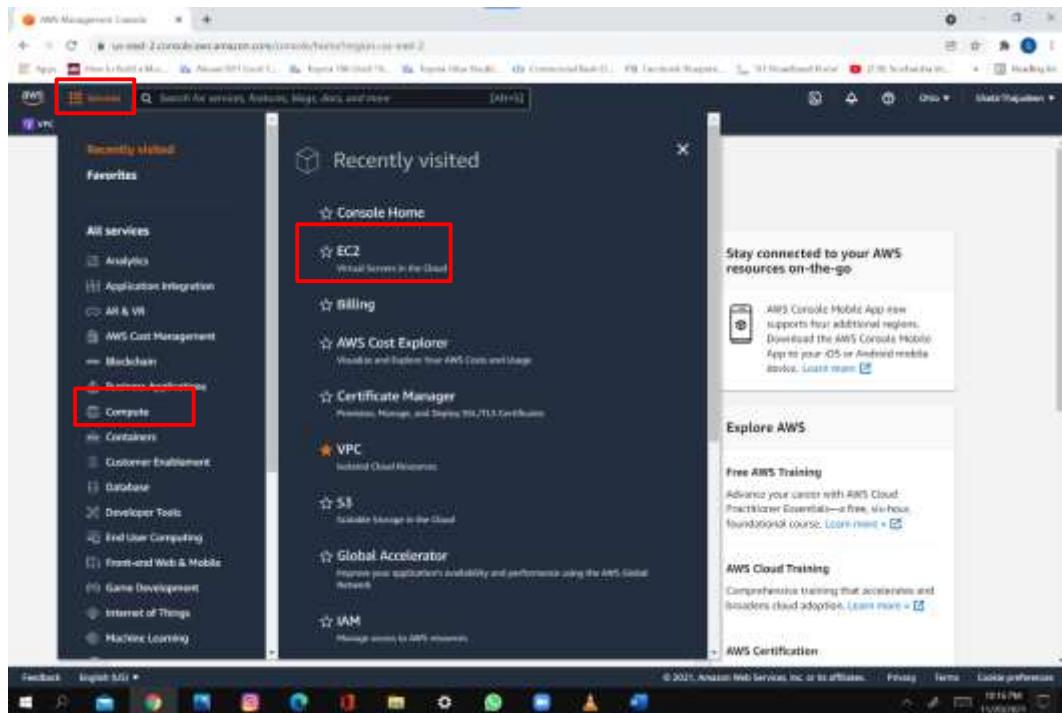
### Lab: Create an EC2 Instance

The objective of this lab is to create a virtual machine, which is called an EC2 instance in AWS cloud.

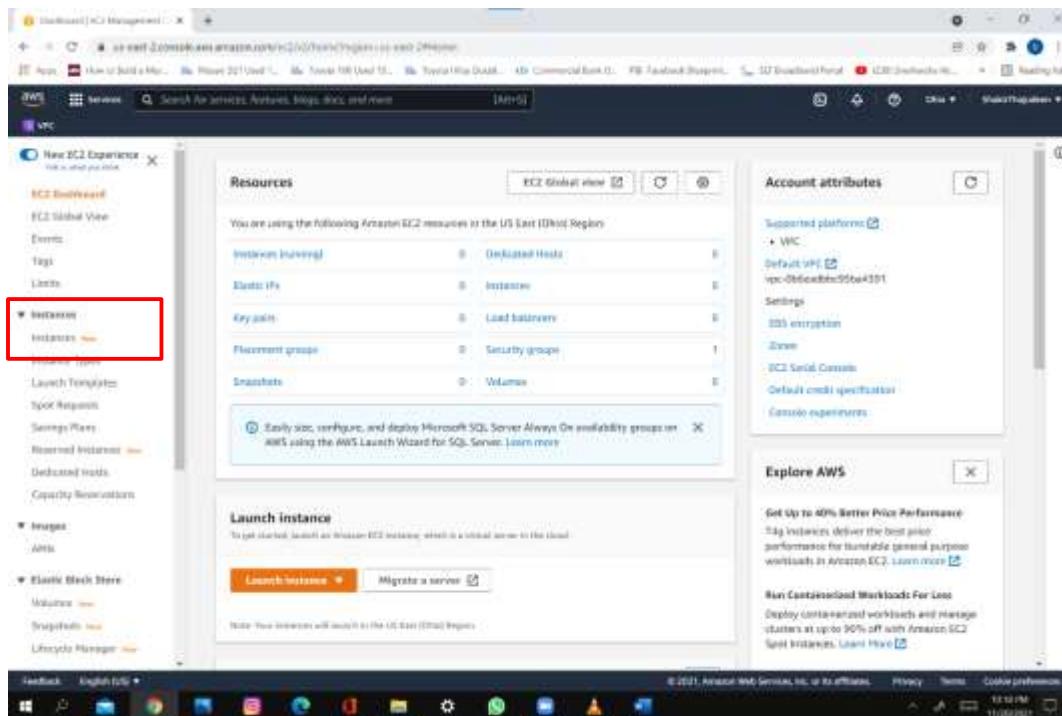
Log in to the **AWS Management Console** and select the region where the EC2 instance should be created:



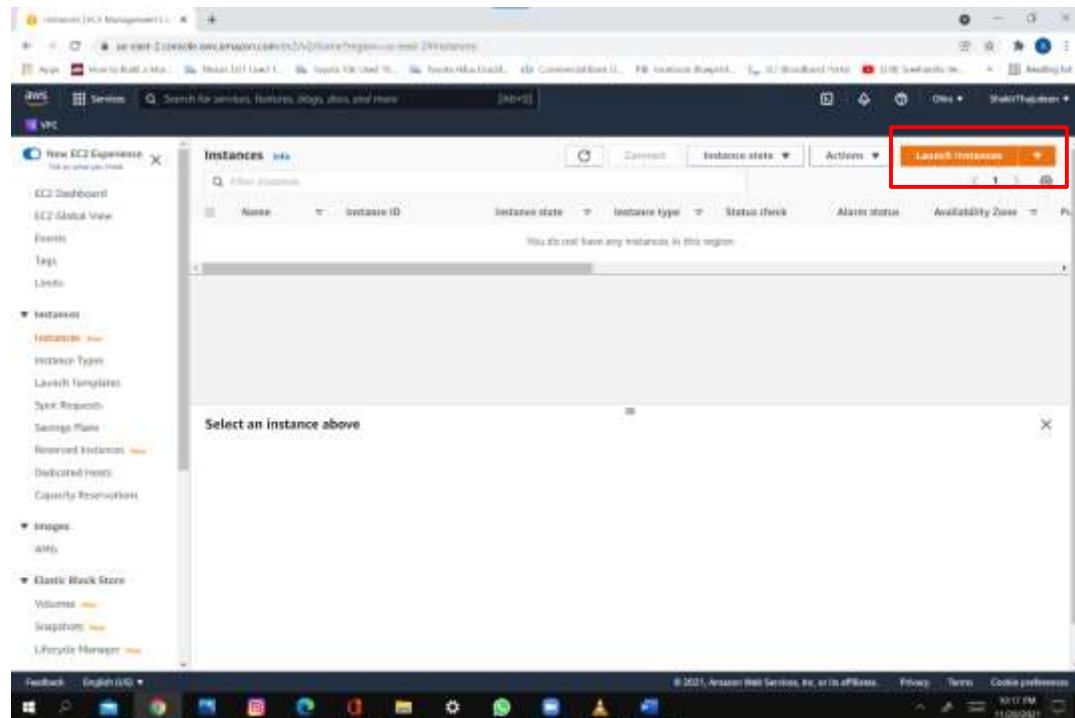
Click on **Services**, then **Compute**, and finally click on **EC2**:



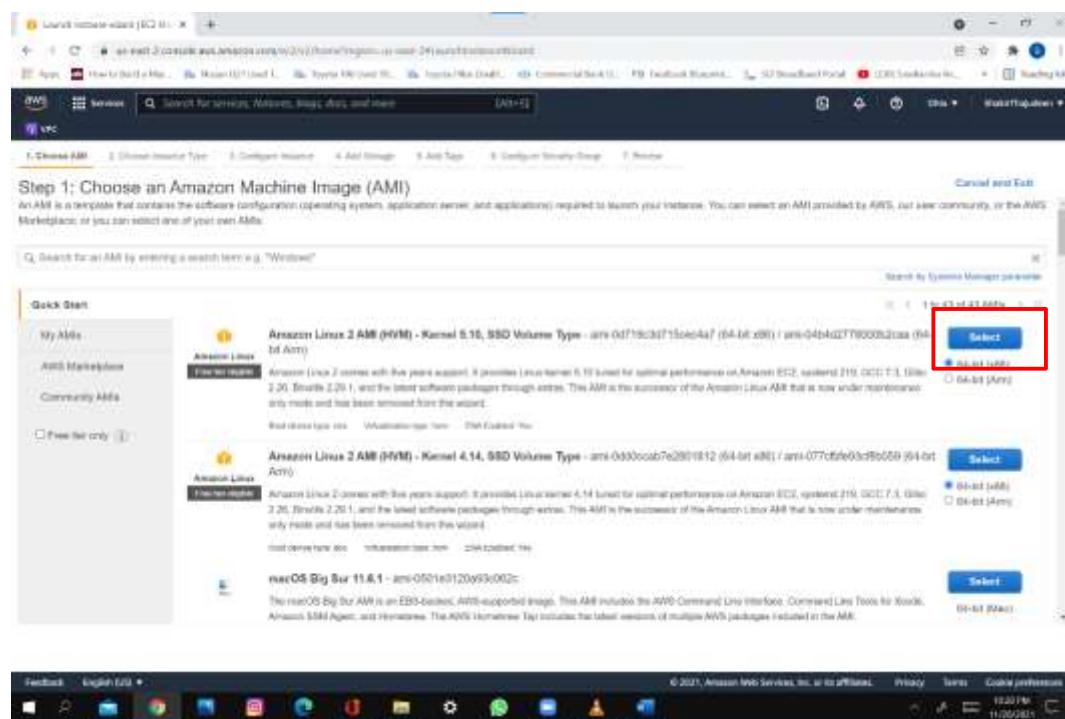
In the **EC2 Dashboard** , select **Instances** under **Instances**:



Click on **Launch Instances**:



Select an AWS Machine Image (AMI):



Select the **t2.micro** instance type and click **Next**:

The screenshot shows the AWS Launch Wizard interface. The title bar says "Launch instance wizard (EC2) Step 2: Choose an Instance Type". Below it, a sub-header says "Step 2: Choose an Instance Type". A note states: "Amazon EC2 provides a wide selection of instance types optimized for different use cases. Instances are virtual servers that run your applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your application. Learn more about instance types and how they can meet your computing needs." A table lists various instance types with their details. The **t2.micro** row is selected and highlighted with a red box. At the bottom right of the table, there are buttons: "Cancel", "Previous", "Review and Launch", and "Next: Configure Instance Details". The "Review and Launch" button is blue, while the others are grey. The "Next: Configure Instance Details" button is also highlighted with a red box.

Configure instance details and click **Next**:

The screenshot shows the AWS Launch Wizard interface. The title bar says "Launch instance wizard (EC2) Step 3: Configure Instance Details". Below it, a sub-header says "Step 3: Configure Instance Details". The page contains several configuration sections: Network (subnet: vpc-0f9e6d9a, auto-assign public IP: enabled), Placement group (no placement group), Capacity Reservation (none), Domain join directory (no directory), IAM role (none), Shutdown behavior (stop), Stop / Hibernate behavior (stop), Enable termination protection (disabled), Monitoring (disabled), Tenancy (shared - Run on shared hardware instance), and Elastic I/O (disabled). At the bottom right, there are buttons: "Cancel", "Previous", "Review and Launch", and "Next: Add Storage". The "Next: Add Storage" button is highlighted with a red box.

Select Storage or just click **Next** to use the default settings:

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and manage store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not increase stored volumes. Learn more about storage options in Amazon EBS.

Volume Type	Device	Size (GB)	Volume Type	Throughput (Mbps)	Delete on Termination	Encryption
Root	/dev/sda1	8	General Purpose SSD (gp2)	100 / 300	No	Not Encrypted

Add New Volume

Note: Free tier eligible instances can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about this usage fee eligibility and usage restrictions.

Shared file systems

You currently don't have any file systems on this instance. Select "Add file system" button below to add a file system.

Add file system

Cancel Previous Review and Launch Next: Add Tags

Add tags and click **Next**:

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can't be applied to volumes, instances or hosts.

Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EBS resources.

Key	Value
AWS:instance:iamArn	arn:aws:iam::123456789012:root

This resource currently has no tags.

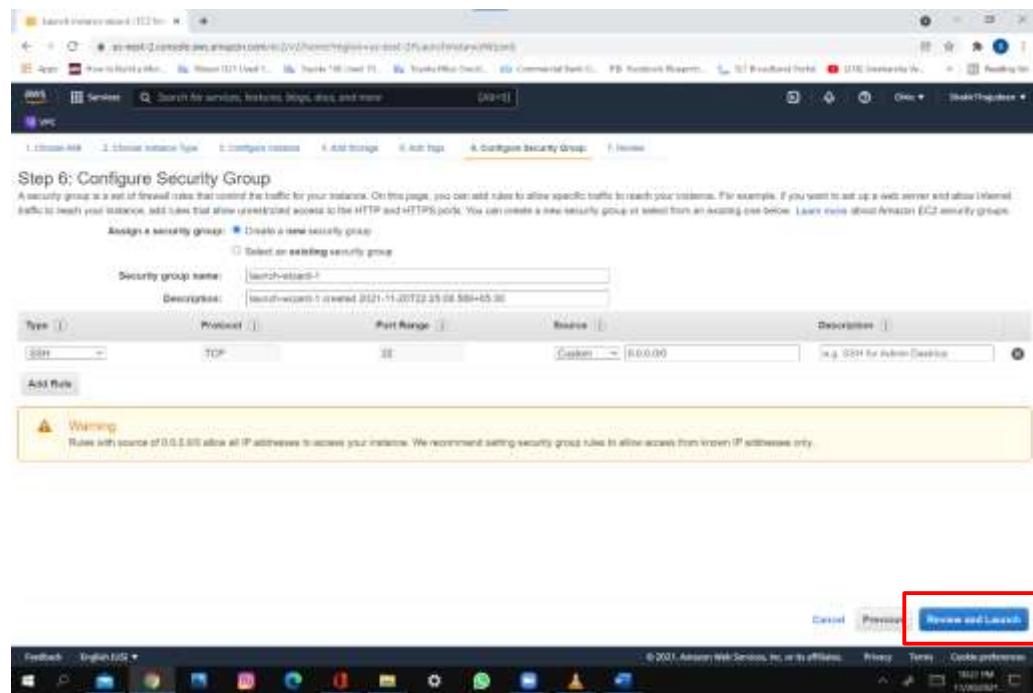
Choose the **Add tag** button or [click](#) to edit a Name tag.

Make sure your IAM policy includes permissions to create tags.

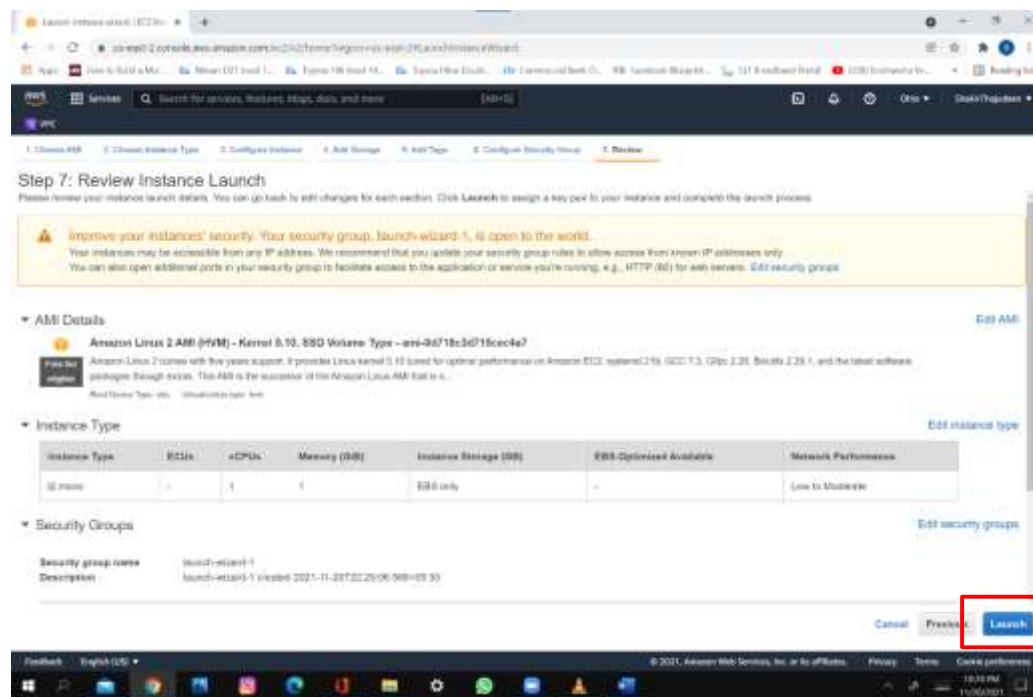
Add Tag (Up to 50 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Groups

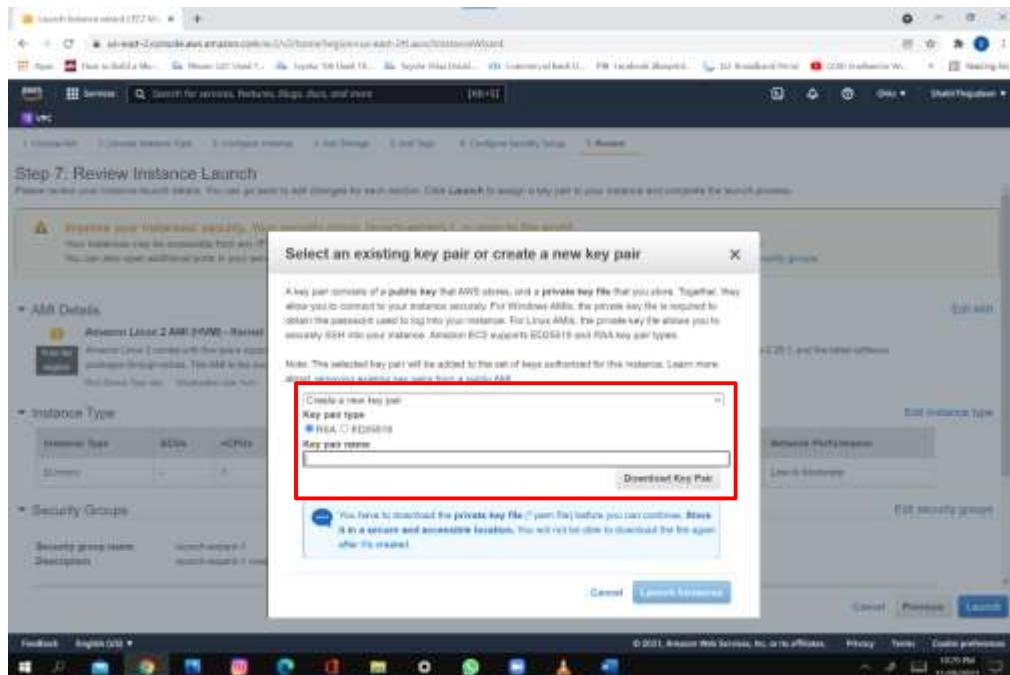
Configure security group and click **Review and Launch**:



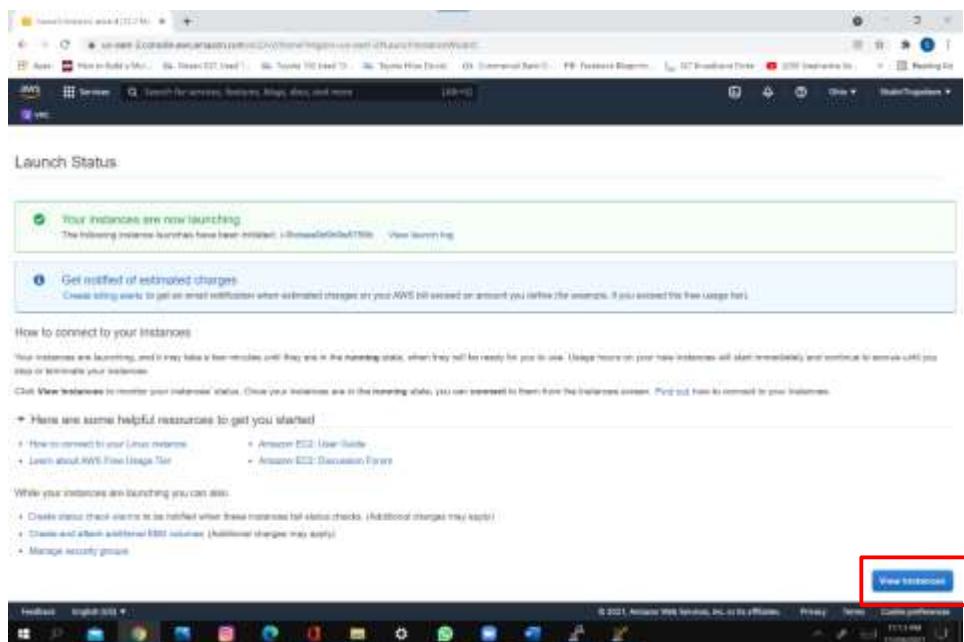
Review the instance configuration. When ready, click **Launch**:



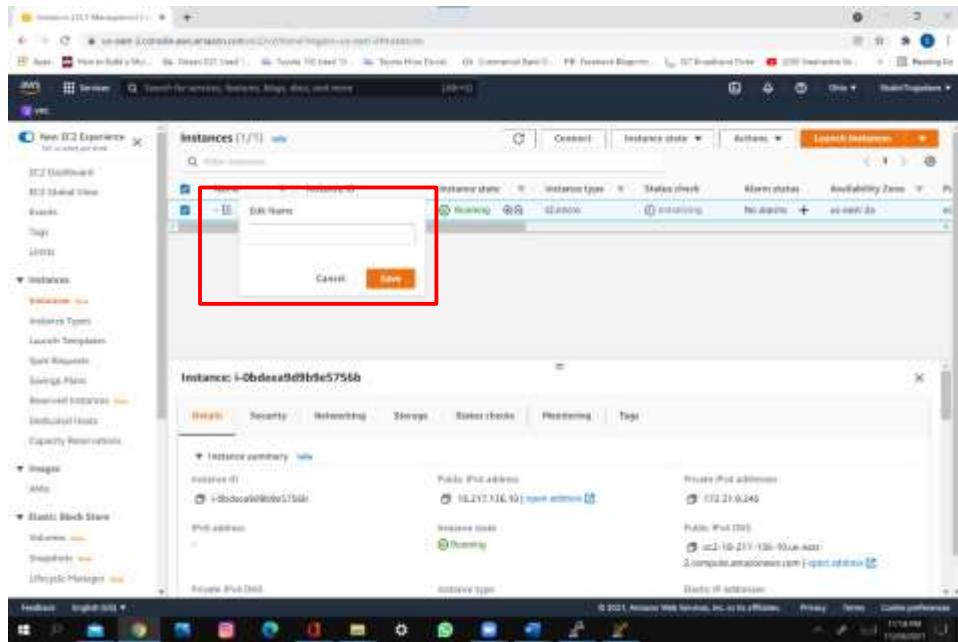
Create a new key pair and download the file:



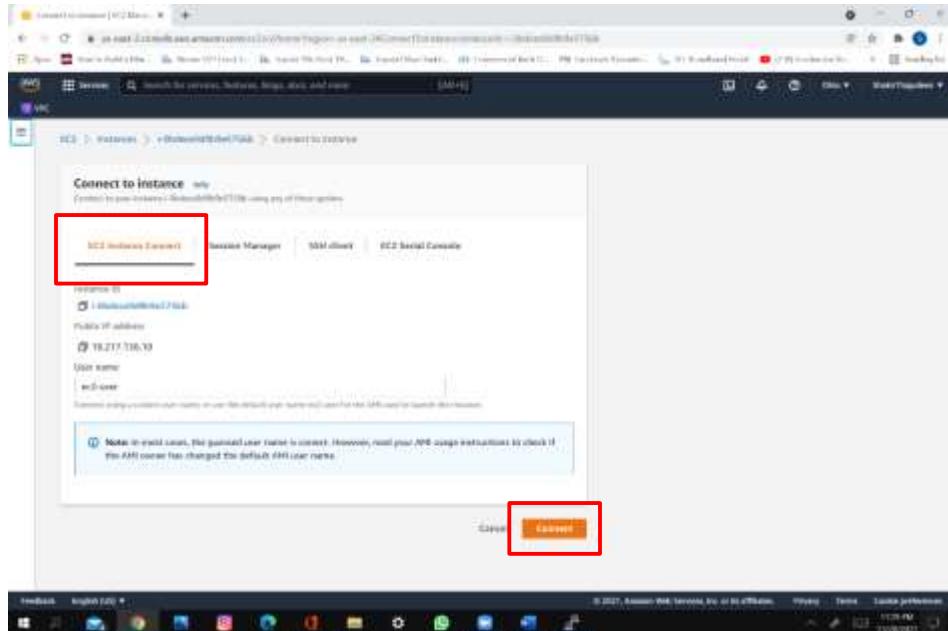
Launch status. Click View Instances:



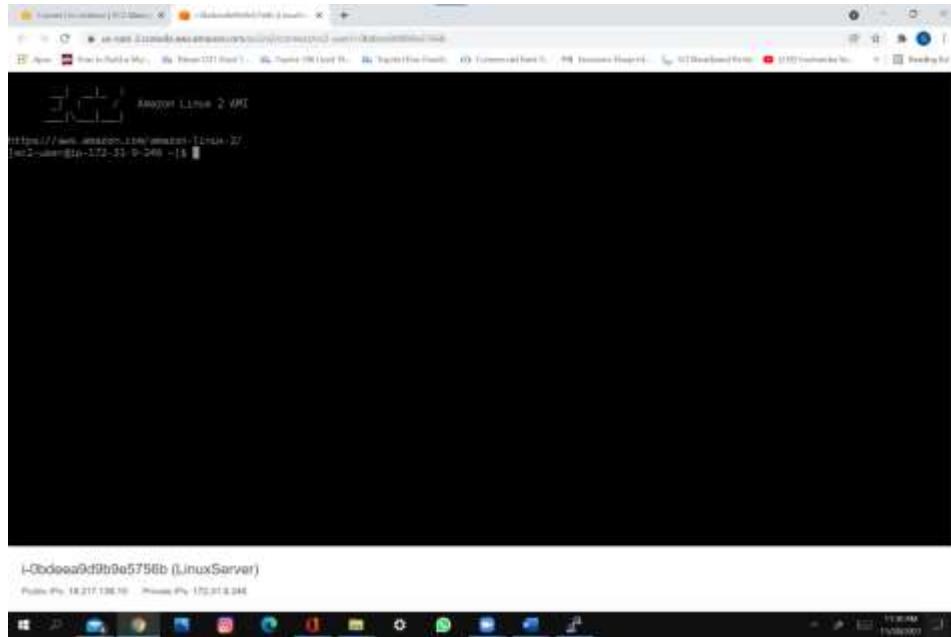
Launch instance – status. You can edit the instance name:



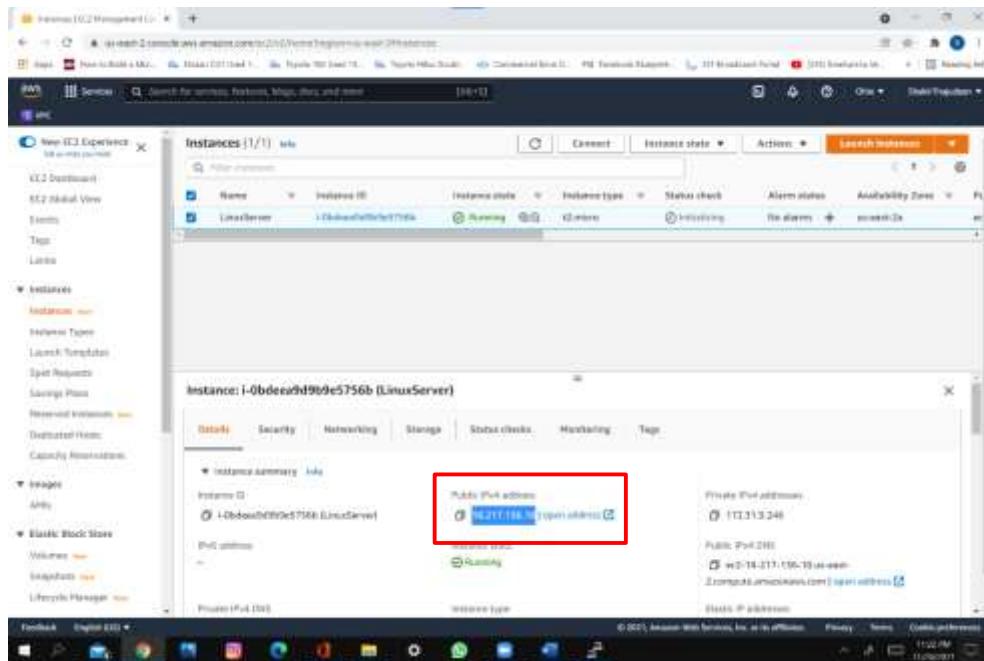
Select the instance, then click **Connect**. Select **EC2 Instance Connect** and click **Next**:



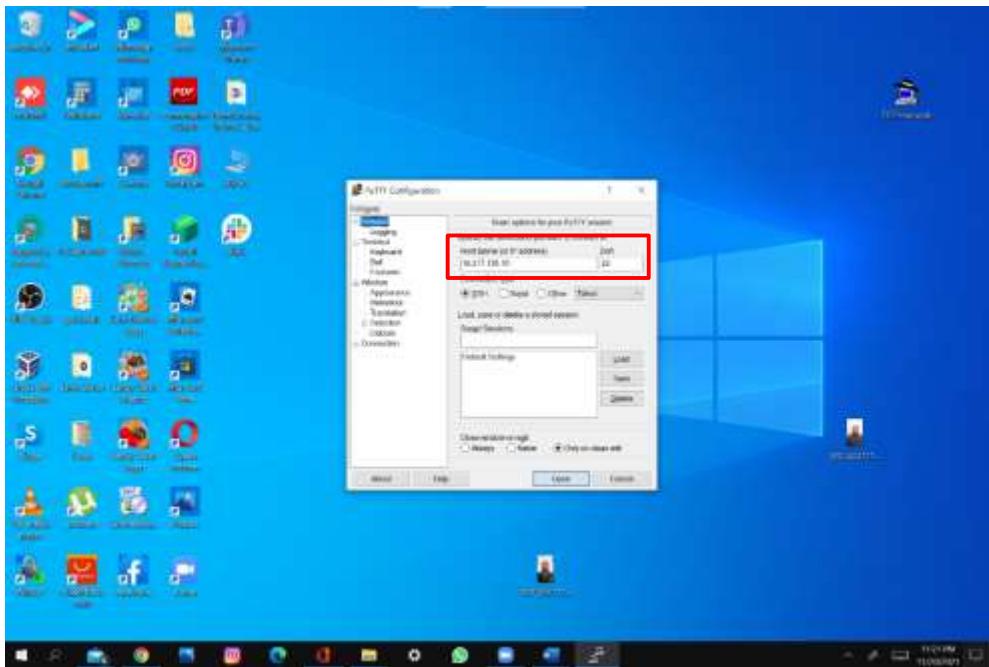
A new browser tab will open with the EC2's Command Line Interface (CLI):



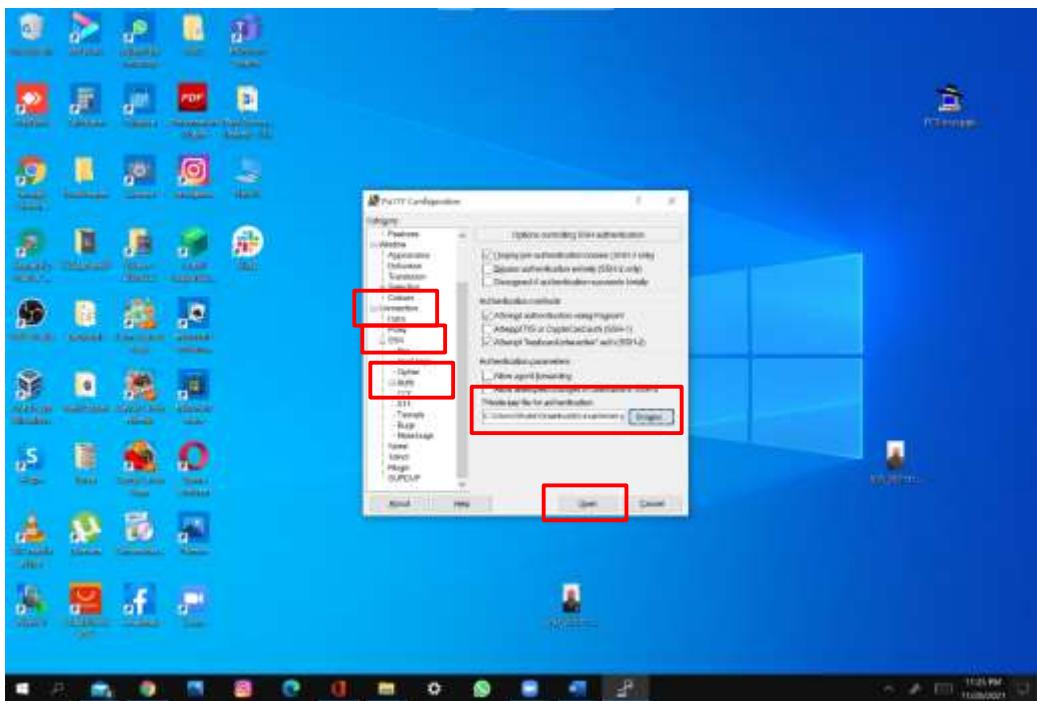
To connect to a LINUX EC2 instance from Windows, an SSH client like PuTTY is needed. From the EC2 Instance details page, copy its public IP address:



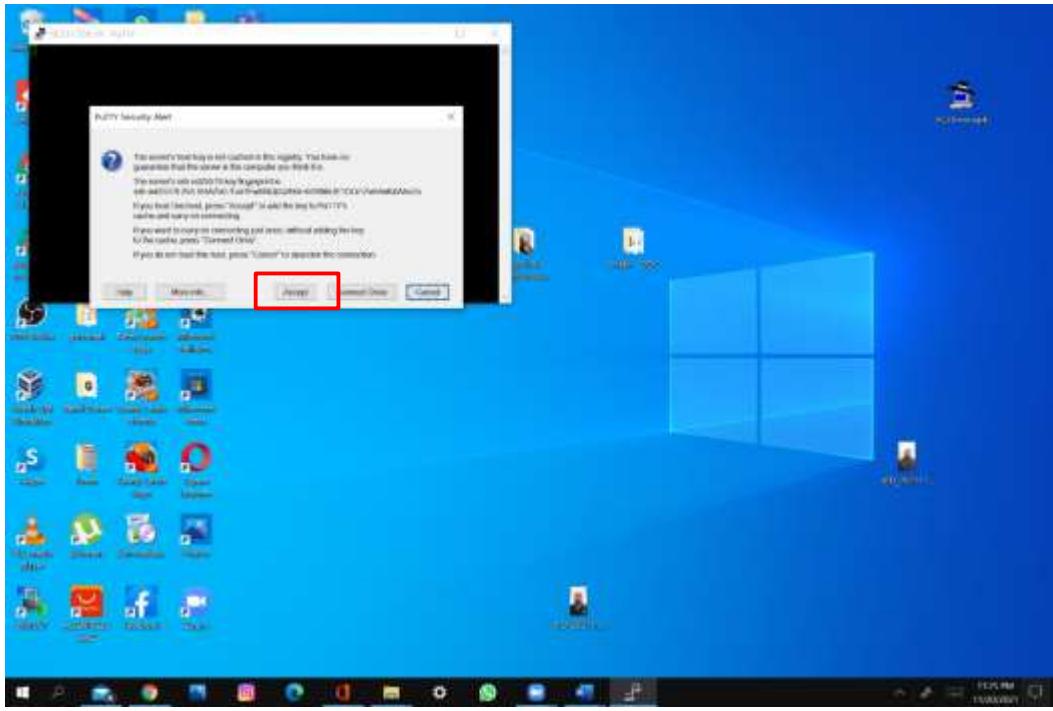
Paste that address in PuTTY's **Host Name (or IP address)** field:



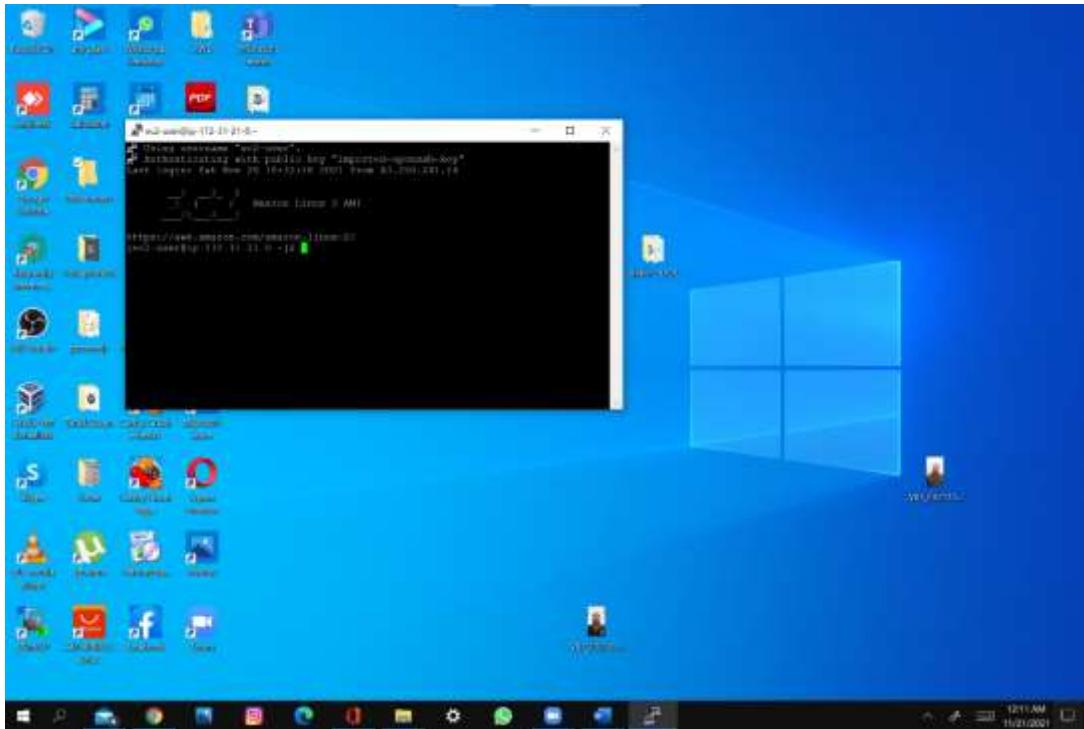
From the left panel, select **Connection**, then select **SSH**, and finally select **Auth**. Click **Browse** and locate the downloaded putty .ppk key. Click **Open**:



A PuTTY Security Alert will pop up. Accept it:



The access to the EC2 instance CLI is ready:



## Chapter 3 Labs

### Lab: Create an S3 Bucket

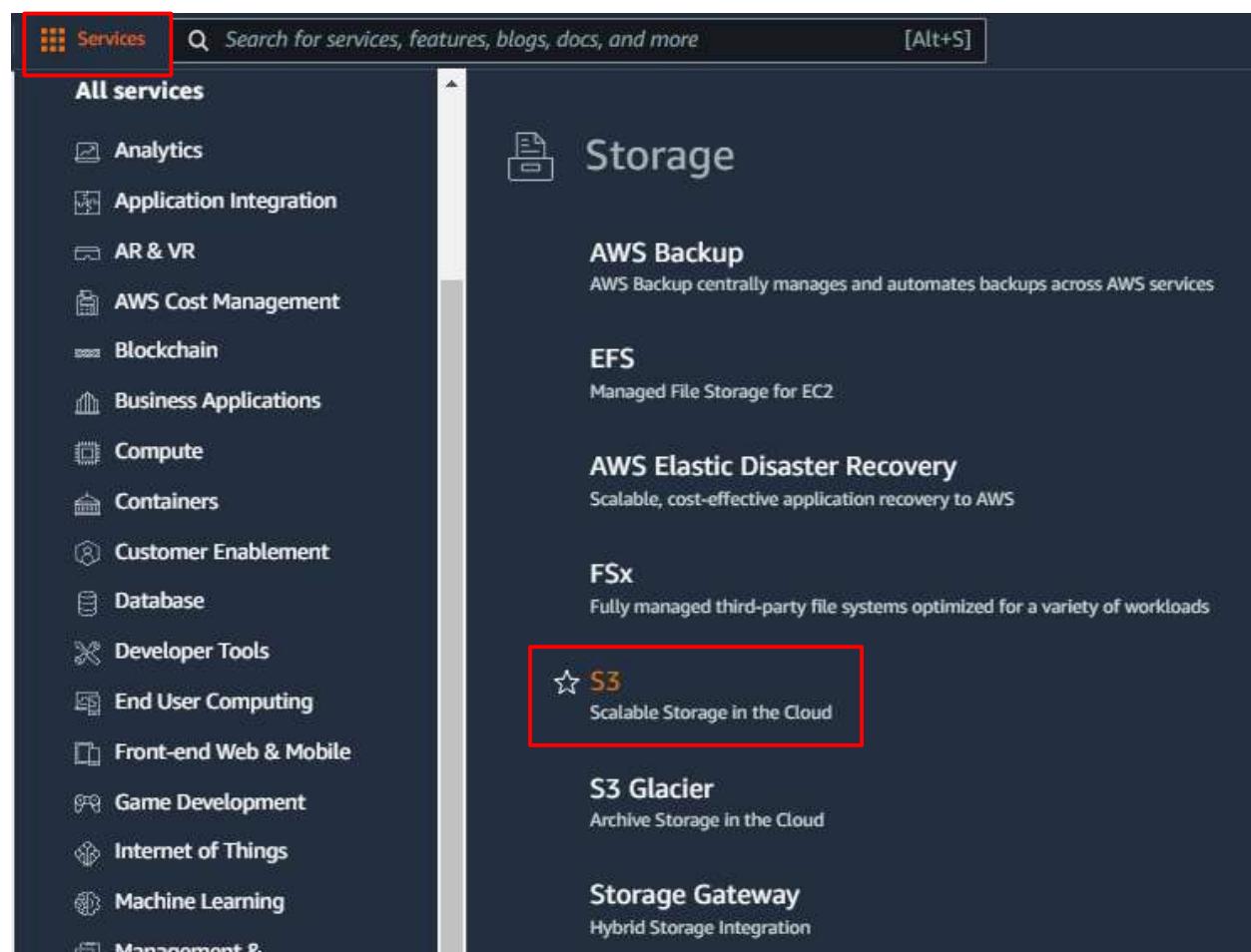
The objective of this lab is to create an S3 bucket.

There are three ways to create an Amazon S3 bucket, via the management console, CLI, or API.

Remember bucket names should be unique for each AWS partition, and once a bucket is created, bucket name or region cannot be changed.

#### **Create an S3 bucket using the S3 console**

Sign in to the AWS Management Console and click on **Services**. The S3 option is found under the **Storage** section in the AWS console:



Click **Create bucket**:



The Create bucket wizard opens.

Enter a name for the Bucket, select the region in which the bucket should be created:

Amazon S3 > Create bucket

## Create bucket Info

Buckets are containers for data stored in S3. Learn more [↗](#)

### General configuration

Bucket name  Bucket name must be unique and must not contain spaces or uppercase letters. See rules for bucket naming [↗](#)

AWS Region  ▼

Copy settings from existing bucket - optional  
Only the bucket settings in the following configuration are copied.

The bucket name is visible in the URLs that point to the objects in the bucket. So avoid including sensitive information in the bucket name.

Choose a Region that minimizes latency and costs and addresses regulatory requirements.

Choose the Block Public Access settings that you want to apply to the bucket:

### Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

#### **Block all public access**

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

By default, **Block all public access** is enabled. This can be changed for specific case types. Block Public Access settings that you enable for the bucket are also enabled for all access points that you create on the bucket.

For this lab we will not use versioning nor encryption, so leave the default settings selected.

From **Object Lock**, select **Disable**.

If Object Lock is required, some permissions need to be addressed:

- s3:CreateBucket,
- s3:PutBucketVersioning, and
- s3:PutBucketObjectLockConfiguration.

## Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

### Bucket Versioning

- Disable
- Enable

## Tags (0) - *optional*

Track storage cost or other criteria by tagging your bucket. [Learn more](#)

No tags associated with this bucket.

[Add tag](#)

## Default encryption

Automatically encrypt new objects stored in this bucket. [Learn more](#)

### Server-side encryption

- Disable
- Enable

## ▼ Advanced settings

### Object Lock

Store objects using a write-once-read-many (WORM) model to help you prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. [Learn more](#)

- Disable
- Enable

Permanently allows objects in this bucket to be locked. Additional Object Lock configuration is required in bucket details after bucket creation to protect objects in this bucket from being deleted or overwritten.

i Object Lock works only in versioned buckets. Enabling Object Lock automatically enables Bucket Versioning.

Once all the options are selected, click **Create bucket**.

Once the bucket is created, click on it to show its details:

The screenshot shows the Amazon S3 console interface. At the top, the path 'Amazon S3 > uniquenameinawspartition' is visible. Below it, the bucket name 'uniquenameinawspartition' is displayed with an 'Info' link. A navigation bar at the top of the main content area includes tabs for 'Objects' (which is selected), 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The main content area is titled 'Objects (0)' and contains a message stating 'Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant'. Below this message are several action buttons: 'Copy' (with icon), 'Copy S3 URI', 'Copy URL', 'Download', 'Open' (with icon), 'Delete', 'Actions' (with a dropdown arrow), 'Create folder', and 'Upload' (highlighted in orange). A search bar labeled 'Find objects by prefix' is present. A table header row is shown with columns for 'Name', 'Type', and 'Last modified'. A message 'No objects' and the sub-message 'You don't have any objects in this buck' are displayed. Finally, a large orange 'Upload' button is located at the bottom right.

To upload a file, click on the **Upload** button. Then click **Add files** to locate the file or files to be uploaded:

**Upload** Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more [?](#)

Drag and drop files and folders you want to upload here, or choose **Add files**, or **Add folders**.

**Files and folders (0)**  
All files and folders in this table will be uploaded.

< 1 >

	Name	Folder	Type	Size
No files or folders You have not chosen any files or folders to upload.				

**Destination**

Destination  
`s3://uniqueusernameinawspartition`

▶ **Destination details**  
Bucket settings that impact new objects stored in the specified destination.

▶ **Permissions**  
Grant public access and access to other AWS accounts.

▶ **Properties**  
Specify storage class, encryption settings, tags, and more.

Properties like *storage class*, *encryption*, *object permissions*, *tags*, and *metadata* can be configured on a file-by-file basis before they are uploaded to the S3 bucket.

Under Permissions, leave the default options selected:

▼ **Permissions**  
Grant public access and access to other AWS accounts.

**Access control list (ACL)**  
Grant basic read/write permissions to other AWS accounts. [Learn more](#) 

 AWS recommends using S3 bucket policies or IAM policies for access control. [Learn more](#) 

**Access control list (ACL)**

Choose from predefined ACLs  
 Specify individual ACL permissions

**Predefined ACLs**

Private (recommended)  
Only the object owner will have read and write access.

Grant public-read access  
Anyone in the world will be able to access the specified objects. The object owner will have read and write access. [Learn more](#) 

Under **Properties**, selected the **Standard** Storage class:

▼ **Properties**  
Specify storage class, encryption settings, tags, and more.

**Storage class**  
Amazon S3 offers a range of storage classes designed for different use cases. [Learn more](#) or see [Amazon S3 pricing](#)

Storage class	Designed for	Availability Zones	Min storage duration	T	C
<input checked="" type="radio"/> Standard	Frequently accessed data (more than once a month) with milliseconds access	≥ 3	-	-	-
<input type="radio"/> Intelligent-Tiering	Data with changing or unknown access patterns	≥ 3	-	-	-
<input type="radio"/> Standard-IA	Infrequently accessed data (once a month) with milliseconds access	≥ 3	30 days	1	
<input type="radio"/> One Zone-IA	Recreatable, infrequently accessed data (once a month) stored in a single Availability Zone with milliseconds access	1	30 days	1	
<input type="radio"/> Glacier	Long-lived archive data accessed once a year with retrieval of minutes to hours	≥ 3	90 days	-	-
<input type="radio"/> Glacier Deep Archive	Long-lived archive data accessed less than once a year with retrieval of hours	≥ 3	180 days	-	-
<input type="radio"/> Reduced redundancy	Noncritical, frequently accessed data with milliseconds access (not recommended as S3 Standard is more cost effective)	≥ 3	-	-	-

Under **Server-side encryption settings**, select **Do not specify an encryption key**:

**Server-side encryption settings**

Server-side encryption protects data at rest. [Learn more](#)

**Server-side encryption**

Do not specify an encryption key  
 Specify an encryption key

**⚠** If your bucket policy requires encrypted uploads, you must specify an encryption key or your upload will fail.

ⓘ Since default encryption is disabled for this bucket, no encryption settings will be applied to the objects when storing them in Amazon S3.

---

**Tags - optional**

Track storage cost or other criteria by tagging your objects. [Learn more](#)

No tags associated with this resource.

**Add tag**

---

**Metadata - optional**

Metadata is optional information provided as a name-value (key-value) pair. [Learn more](#)

No metadata associated with this resource.

**Add metadata**

When all the configurations are done, click on the **Upload** button.

## Lab: Create an Elastic Block Store (EBS) Volume

The objective of this lab is to create an empty Elastic Block Store (EBS) volume.

From the management console, select the Region (top right of the screen) where you want to create your resources. Then select EC2 under the Compute service group.

The screenshot shows the AWS Management Console homepage. At the top, there's a search bar and a navigation bar with links for Home, Services, and Help. A red box highlights the 'Region' dropdown menu in the top right corner. Below the header, the main content area is titled 'AWS Management Console'. It features a sidebar on the left with 'AWS services' grouped into categories like Compute, Containers, Storage, and others. A red box highlights the 'Compute' section. The main content area lists various AWS services under three main groups: Management & Governance, Security, Identity, & Compliance, and Compute. On the right side, there are sections for 'Stay connected to your AWS resources on-the-go', 'Explore AWS', '7 Reasons to get AWS Certified', 'AWS Training', 'AWS Free Digital Training', and 'AWS Certification Resources'. At the bottom, there are links for 'Feedback', 'English (US)', and copyright information.

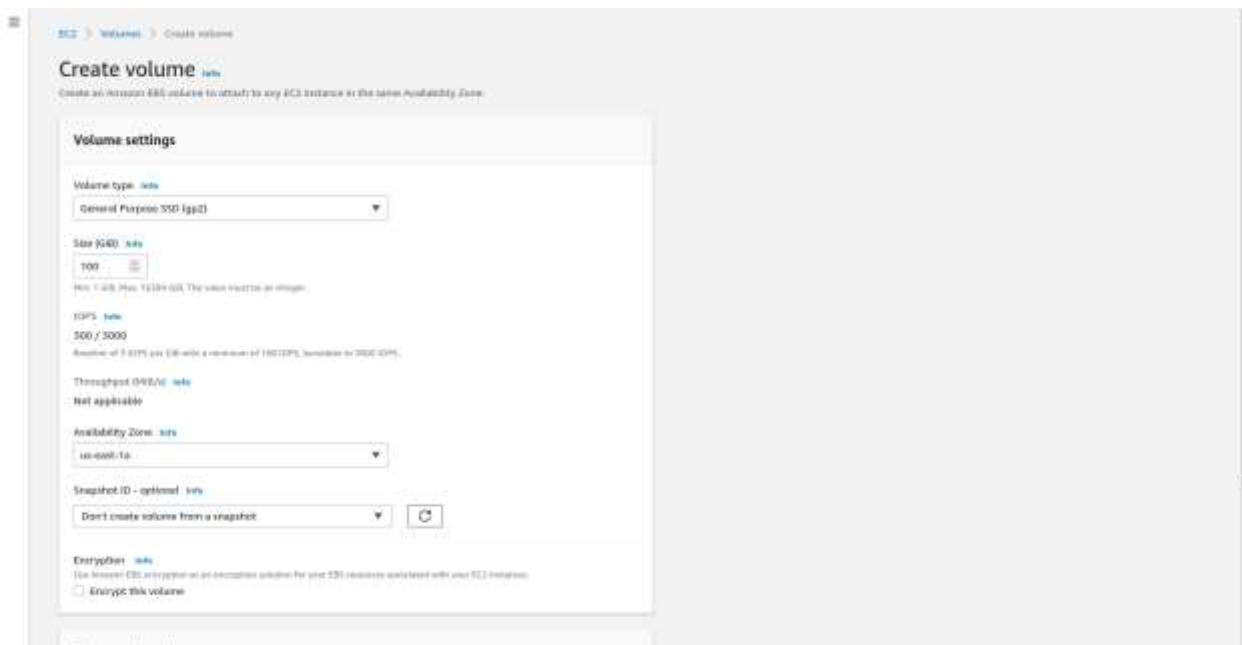
In the EC2 dashboard, select Volumes under Elastic Block Store:

The screenshot shows the AWS EC2 Dashboard. On the left sidebar, under the 'Elastic Block Store' section, the 'Volumes' option is highlighted with a red box. The main content area displays the 'Resources' section with various metrics: Instances (0), Dedicated Hosts (0), Elastic IPs (0), Instances (1), Key pairs (2), Load balancers (0), Placement groups (0), Security groups (0), Snapshots (0), and Volumes (2). A callout box provides instructions for launching Microsoft SQL Server Always On availability groups. The 'Service health' section shows the US East (N. Virginia) region with a green status indicating it's operating normally. The 'Zones' section lists five availability zones: us-east-1a, us-east-1c, us-east-1d, us-east-1e, and us-east-1f. On the right side, there's an 'Account attributes' panel listing supported platforms (VPC, Direct VPC, spot-advertise, Semigroup, EBS encryption, S3, VPC serial console, Default credit specification, Compute environments) and an 'Explore AWS' panel with links to Graviton challenges, ML Inference savings, and AWS Graviton2 instances.

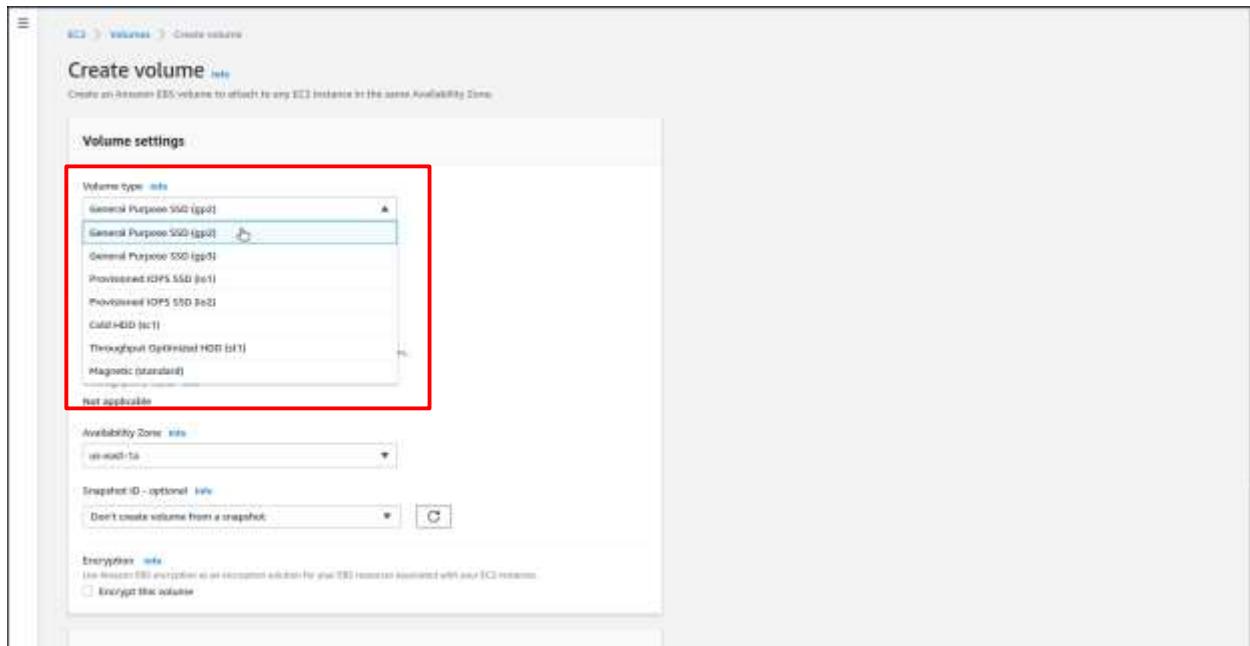
Click on the Create volume button to start the creation process:

The screenshot shows the 'Volumes' page within the EC2 dashboard. The left sidebar has the 'Volumes' option selected. The main area displays a table of existing volumes, with one row highlighted. The 'Actions' column for this row contains a red box around the 'Create volume' button. Below the table, a large text input field labeled 'Select a volume source' is visible.

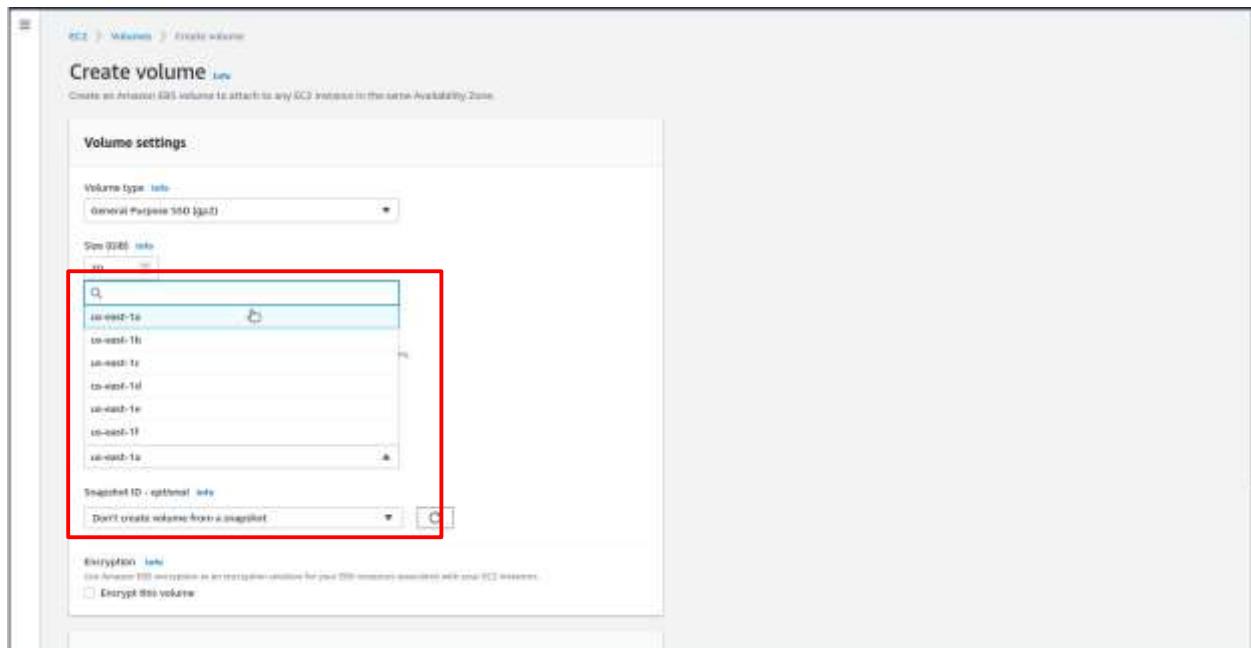
The new screen shows the options for the volume creation:



For this lab, select a General Purpose SSD (gp2) volume type:



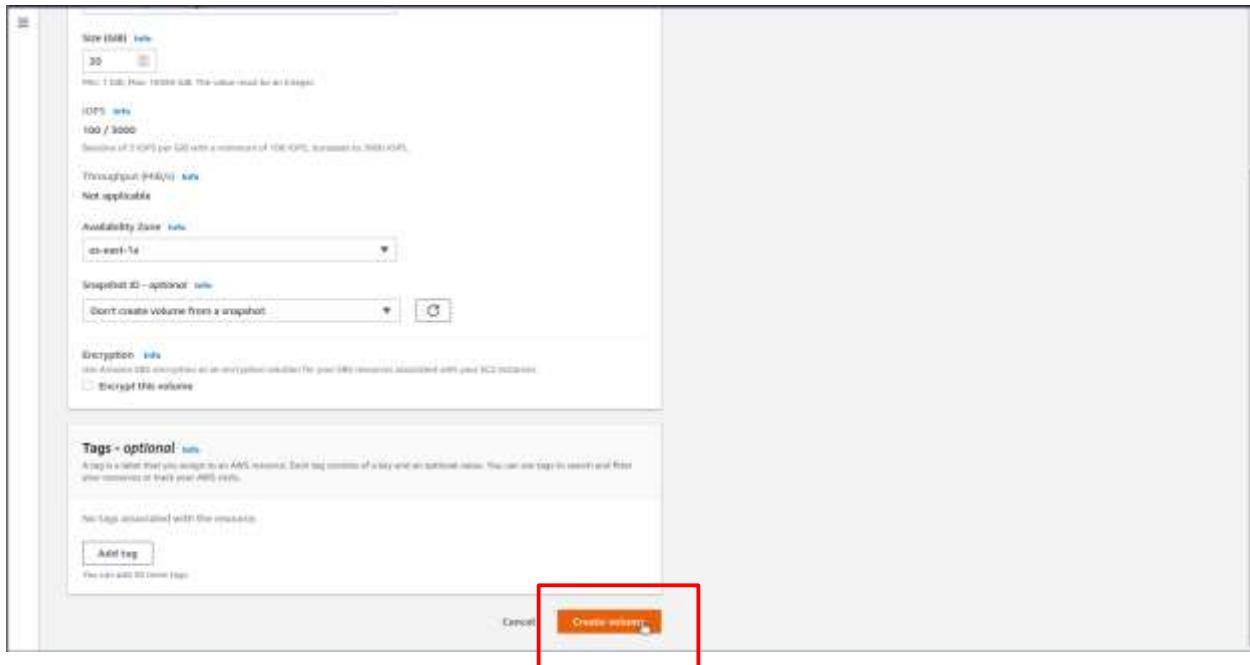
Set the size of the volume to 20GB and select one Availability Zone from the selection box:



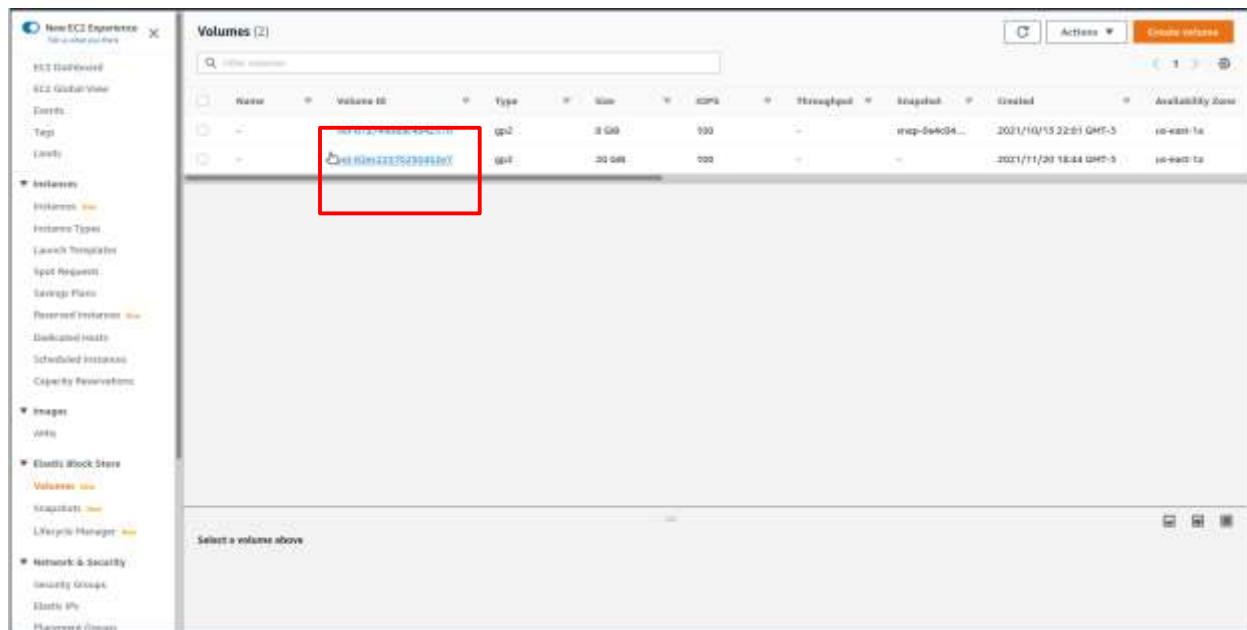
It is important to note that volumes can only be attached to EC2 instances that belong to the same Availability Zone.

In this lab, we are creating an empty volume, so leave the option Snapshot ID as *Don't create volume from a snapshot*. Leave the Encryption option unchecked, since we will not deal with encryption in this lab.

When all the settings are done, click on the Create volume button:



The window will update and show the new volume. Click on the name of the volume to show its details:



## Volume details:

The screenshot shows the AWS EC2 Volume Details page for volume `vol-02ec22570250452e7`. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Logs, Instances, Volumes, Volume Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, and Elastic Block Store. The main content area displays the following details:

Volume ID	Size	Type	Volume status
<code>vol-02ec22570250452e7</code>	20 GiB	gp3	Okay
Volume state	IOPS	Throughput	Encryption
Available	100	-	Not encrypted
KMS key ID	KMS key alias	XMLEasy-ARN	Tags
-	-	-	-
Availability Zone	Created	HDD-Attach enabled	Attached instances
<code>us-east-1a</code>	<code>Sat Nov 20 2021 10:44:24 GMT-0300 (Brasília Standard Time)</code>	No	-
Details: AMI			

Below the details, there are tabs for Status checks, Monitoring, and Tags. The Status checks section shows:

Volume status	Availability Zone
Okay	<code>us-east-1a</code>

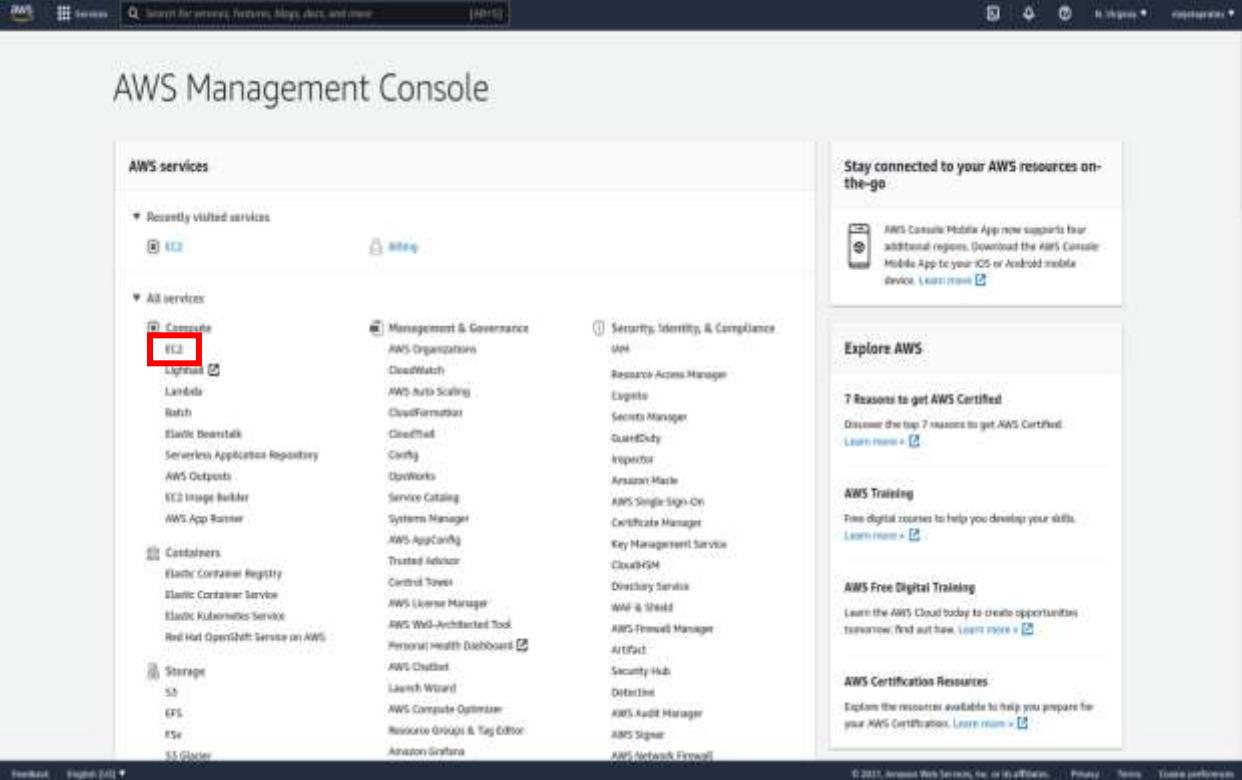
The Monitoring and Tags sections are currently empty.

The empty EBS volume is ready.

## Lab: Create an EC2 Instance and Mount an EBS Volume

The objective of this lab is to attach an Elastic Block Store (EBS) volume to an EC2 instance (virtual machine).

First create the instance. From the management console, select the **Region** (top right of the screen) where the resources were created. Select **EC2** under the **Compute** service group, from the **AWS Management Console**:



The screenshot shows the AWS Management Console interface. At the top, there's a search bar and a navigation bar with links like 'AWS Home', 'Services', 'Logs', 'Metrics', 'CloudWatch Metrics', 'CloudWatch Logs', and 'AWS Lambda'. Below the search bar, the title 'AWS Management Console' is displayed. On the left, the 'AWS services' sidebar is open, showing categories like 'Recently visited services' (with 'EC2' selected), 'All services' (with 'Compute' expanded), 'Containers', 'Storage', and 'AWS Glue'. Under 'Compute', 'EC2' is highlighted with a red box. The main content area shows three columns of services: 'Management & Governance' (AWS Organizations, CloudWatch, AWS Auto Scaling, CloudFormation, CloudWatch Metrics, Config, OpsWorks, Service Catalog, Systems Manager, AWS AppConfig, Trusted Advisor, Control Tower, AWS License Manager, AWS Web-Architecture Tool, Personal Health Dashboard, AWS Chatbot, Launch Wizard, AWS Compute Optimizer, Resource Groups & Tag Editor, Amazon Graviton), 'Security, Identity, & Compliance' (IAM, Resource Access Manager, Secrets Manager, GuardDuty, Inspector, Amazon Macie, AWS Single Sign-On, Certificate Manager, Key Management Service, CloudHSM, Directory Service, WAF & Shield, AWS Firewall Manager, AWS IAM API, Security Hub, Detective, AWS Audit Manager, AWS Signer, AWS Network Firewall), and 'Explore AWS' (Stay connected to your AWS resources on-the-go, AWS Certified, AWS Training, AWS Free Digital Training, AWS Certification Resources). At the bottom of the page, there are links for 'Feedback', 'English (US)', and copyright information: '© 2021, Amazon Web Services, Inc. or its affiliates.' followed by 'Privacy', 'Terms', and 'Cookie preferences'.

Using the left-side panel from the EC2 dashboard, select **Volumes** under **Elastic Block Store** to access the volumes dashboard:

The screenshot shows the AWS EC2 Dashboard. On the left sidebar, under the 'Elastic Block Store' section, the 'Volumes' option is selected. The main content area displays the 'Resources' section, which lists various Amazon EC2 resources in the US East (N. Virginia) Region. The 'Volumes' section shows 2 items. A callout box provides instructions to 'Ready, size, configure, and deploy Microsoft SQL Server Always On availability group on AWS using the AWS Launch Wizard for SQL Server.' To the right, there's an 'Account attributes' panel listing supported platforms (AWS Lambda), default VPC, and settings like EBS encryption, S3 encryption, and EBS KMS. Below that is an 'Explore AWS' panel with links to Graviton challenges, ML Inference discounts, and Graviton2-powered EC2 instances.

Check the selection box of the volume to show its details. Notice that the **Volume state** shows **Available**:

This screenshot shows the 'Volumes (1/2)' page in the AWS Management Console. It lists two volumes: one with Volume ID 'vol-07a7440d1ca423f0' and another with 'vol-02ec22570250452e7'. The second volume is selected. The 'Details' tab is active, showing detailed information for the selected volume. The 'Volume state' field is highlighted with a red box and shows the value 'Available'. Other details include Volume ID, Size (20 GiB), Type (gp2), IOPS (100), Throughput (—), Snapshot ID ('snap-014404...'), Creation Time ('2021/10/13 22:01 GMT-2'), Availability Zone ('us-east-1a'), and Volume status ('Okay'). The 'Encryption' field shows 'Not encrypted' and the 'Snapshot' field shows 'Snapshot'. The 'Attached Instances' field is currently empty.

Click on the Actions button and select Attach volume:

The screenshot shows the AWS EC2 Volumes page. On the left is a navigation sidebar with various EC2-related options like Dashboard, Global View, Events, Tags, Details, Instances, Instance Types, Launch Templates, Spot Requests, Launch Plans, Reserved Instances, Dedicated Instances, Scheduled Instances, Capacity Reservations, Images, AMIs, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, and Network Interfaces. The 'Volumes' section is selected. The main area displays two volumes in a table:

Name	Volume ID	Type	IOPS	Throughput	Snapshot	Create
vol-03ec22570250452e7	gp2	4000	100	-	snap-04d4...	2021/11/29
vol-02ec22570250452e7	gp2	3000	100	-	-	2021/11/29

Below the table, a message says "Volume ID: vol-02ec22570250452e7". There are tabs for Details, Status details, Monitoring, and Tags. The Details tab is active. The volume information includes:

- Volume ID: vol-02ec22570250452e7
- Size: 20 GiB
- Type: gp2
- Volume State: Available
- IOPS: 900
- Throughput: -
- KMS Key ID: -
- KMS Key Alias: -
- KMS Key ARN: -
- Encryption: Not encrypted
- Volume status: Okay
- Availability Zone: us-east-1a
- Created: Sat Nov 29 2021 18:44:24 GMT-0500 (Standard Time)
- Multi-attach enabled: No
- Attached instances: -
- Tags: -

Using the selection box, select the EC2 instance where the volume should be attached. Click on the Attach volume button:

The screenshot shows the 'Attach volume' dialog box. At the top, it says "EC2 > Volumes > vol-02ec22570250452e7 > Attach volume". Below that is a sub-header "Attach volume". A note says "Attach a volume to an instance to use it as you would a regular physical hard disk drive." The dialog has a "Basic details" section with fields for Volume ID (selected), Availability Zone (us-east-1a), and Instance ID. The Instance ID field contains the value "i-037579318b4c9b5f" and has a red box around it. At the bottom are "Cancel" and "Attach volume" buttons, with an arrow pointing to the "Attach volume" button.

The updated window shows the volumes. Select the volume to show its details. Notice that the **Volume state** changed to **In-use**:

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot	Created	Availability Zone
vol-02ec22570230452e7	gp2	20 GiB	100	-	100	snap-0440d4...	2021/11/10 13:22:01 GMT-5	us-east-1a
vol-02ec22570230452e8	gp2	20 GiB	100	-	100	snap-0333fb...	2021/11/10 10:44 GMT-5	us-east-1a
vol-02ec22570230452e9	gp2	20 GiB	100	-	100	snap-0333fb...	2021/11/10 22:47 GMT-5	us-east-1a

Using the left-side panel, select **Instances** under **Instances**. Check the selection box of the instance, then select the **Storage** tab:

Instance	Instance ID	InstanceState	InstanceType	StatusCheck	AlarmStatus	AvailabilityZone	PublicIpAddress	PrivateIpAddress
QueryTime	i-0b597a11730b7c5b2	running	t2.micro	-	No alarms	us-east-1a	-	-
	i-0b597a11730b7c5b2	running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-3-236-159-19.us...	123.134.19

The EC2 instance has two EBS volumes attached to it.

The newly attached volume can be formatted and mounted to the EC2 instance, but this is not in the scope of this lab.

## Removing a volume from an EC2 instance

The root volume of an EC2 instance is where the operating system is installed. Detaching the root volume should be done only under special circumstances. Otherwise it will make the instance unusable.

Detaching an additional volume that was previously attached to the instance can be done while the instance is running, but the mount point of the volume could be lost. If the volume is not needed in that instance anymore, it is a safe procedure.

Using the left-side panel, open the **Volumes** dashboard. Select the volume you want to detach from the instance:

The screenshot shows the AWS EC2 Volumes dashboard. On the left sidebar, under the 'instances' section, the 'Volumes' link is highlighted with a red box. In the main content area, a table titled 'Volumes (1/1)' lists one volume. The volume ID 'vol-033c251145039bed' is highlighted with a red box. The volume details are shown in the expanded view below the table, including its status as 'In use' and its association with instance 'i-0b957e1179885362'.

Name	Volume ID	Type	IOPS	Throughput	Snapshot	Created	Availability Zone
	vol-033c251145039bed	GP2	20 IOPS	100	snap-00a86...	2021/11/02 22:47 (GMT-5)	us-east-1a

Volume ID: vol-033c251145039bed

Details Status Details Monitoring Tag

Details

Volume ID	Size	Type	Volume status
vol-033c251145039bed	20 GiB	gp2	OK

Volume state	IOPS	Throughput	Encryption
In use	100	-	Not encrypted

EBS key ID	SSD key ARN	SSD key ARN	Snapshot
-	-	-	snap-020060ccca62803f

Availability Zone	Created	Multi-attach enabled	Attached instances
us-east-1a	Oct 20 2021 22:47:12 GMT-0300 (Brasil Standard Time)	No	i-0b957e1179885362

Click on the Actions button and select Detach volume:

The screenshot shows the AWS EC2 Volumes page. On the left, there's a navigation sidebar with various EC2-related options like Instances, Images, and Block Storage. The main area displays a table of volumes. One volume is selected, showing its details in a modal window. The 'Actions' button in the top right of the modal is highlighted with a red arrow. A red box highlights the 'Detach volume' option in the dropdown menu that appears when the button is clicked.

**Volume Details (Selected):**

- Volume ID: vol-035c2511458398ed
- Status: In use
- Type: gp2
- Size: 20 GiB
- IOPS: 100
- Throughput: -
- Snapshot: snap-03466...
- Create Time: 2021/11/20

**Actions (Dropdown):**

- Modify volume
- Create snapshot
- Create snapshot lifecycle policy
- Detach volume
- Force detach volume
- Manage auto-enabled QoS
- Manage tags

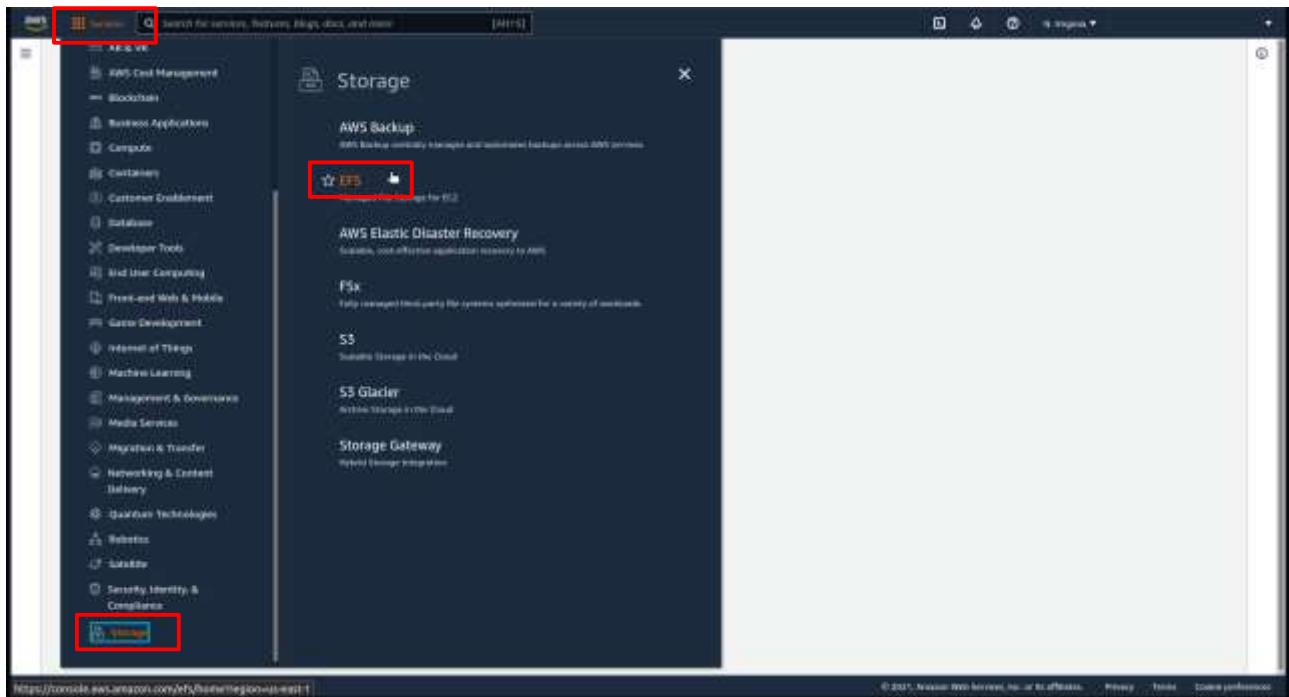
Confirm the detachment.

The EC2 instance has only the Root volume now.

## Lab: Create an Elastic File System (EFS)

The objective of this lab is to create an EFS File System that can be accessed by Linux and Unix systems.

From the management console, select the **Region** (top right of the screen) where the file system should be created. Click on the Services button on the top left of the screen to show all services groups. Select **Storage** from the list; then select **EFS**:



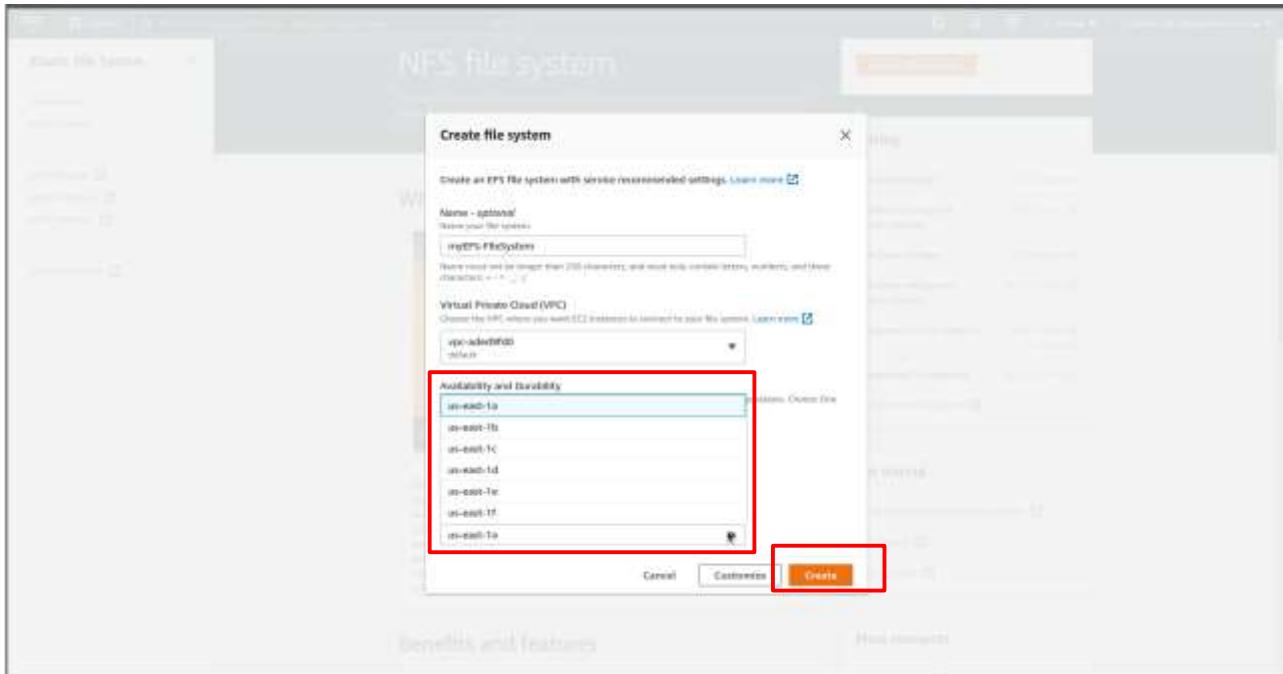
Click on the **Create file system** button:

The screenshot shows the Amazon EFS landing page. On the left, there's a sidebar with options like 'File systems', 'Access points', 'AWS backup', 'AWS DataSync', and 'AWS Transfer'. The main content area features the title 'Amazon Elastic File System' and a sub-section 'Scalable, elastic, cloud-native NFS file system'. Below this is a video thumbnail titled 'Amazon Elastic File System - Scalable, Elastic...'. To the right of the video, the 'Create file system' button is highlighted with a red box. Further down, there's a 'Pricing' section with various storage and access options, and a 'Get started' section.

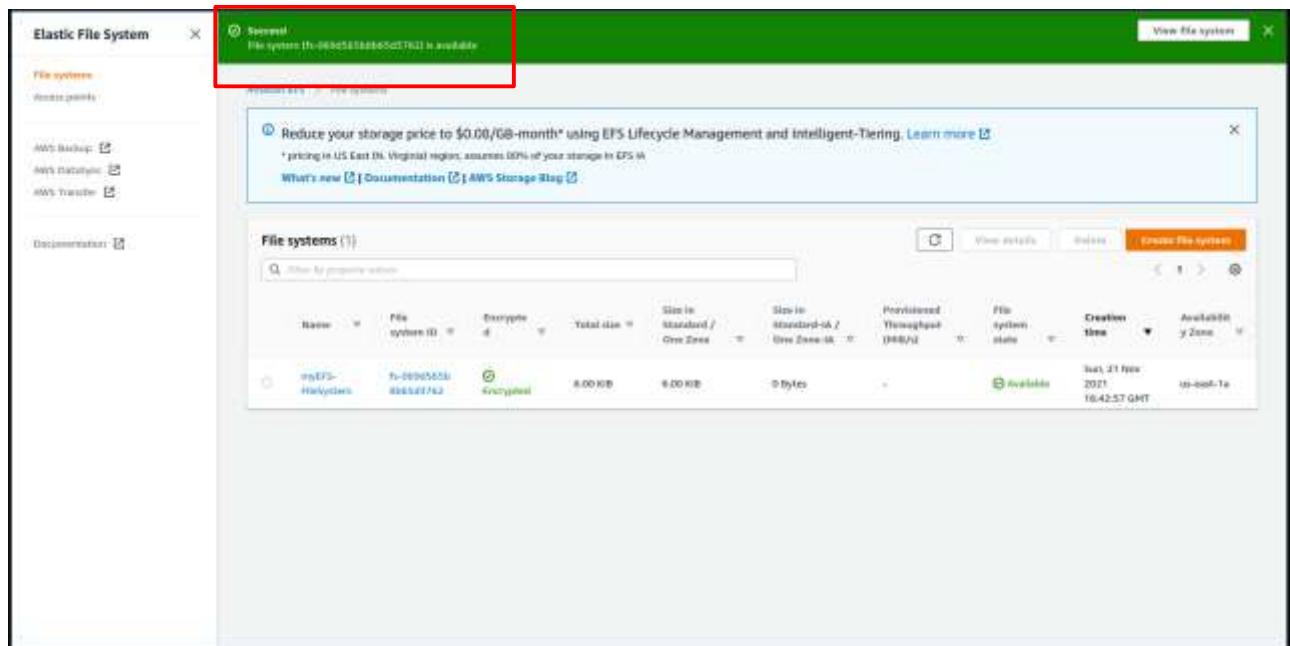
Enter a name for the file system, select the VPC, and choose the redundancy. For this lab we will choose One Zone:

This screenshot shows the 'Create file system' dialog box. It includes fields for 'Name' (set to 'myefs-fs'), 'Virtual Private Cloud (VPC)' (set to 'vpc-adm01'), and 'Availability and Durability' (with 'One Zone' selected). The 'One Zone' option is highlighted with a red box. At the bottom, there are 'Cancel', 'Customize', and 'Create' buttons.

Select the Availability Zone, and click on the **Create** button:



It could take some time. Then the window will update and show the status of the file system:



Select the file system and click on the **View details** button:

Screenshot of the AWS Elastic File System (EFS) console. The left sidebar shows navigation links: 'File systems', 'Access points', 'AWS Backup', 'AWS DataSync', and 'AWS Transfer'. The main area displays a table titled 'File systems (1)'. The table has columns: Name, File system ID, Encrypted, Total size, Size in Standard / One Zone, Size in Standard IA / One Zone IA, Provisioned Throughput (Mbps), File system state, Creation time, and Availability Zone. The single entry in the table is 'myEFS-FileSystem' (File system ID: fs-069d565b8b65d3762, Encrypted: Yes, Total size: 0.00 KiB, Size in Standard / One Zone: 0.00 KiB, Size in Standard IA / One Zone IA: 0 bytes, Provisioned Throughput: 0 Mbps, File system state: Available, Creation time: Sat, 21 Nov 2021 18:42:57 GMT, Availability Zone: us-east-1a). A red box surrounds the 'myEFS-FileSystem' row, and another red box surrounds the 'View details' button in the table header.

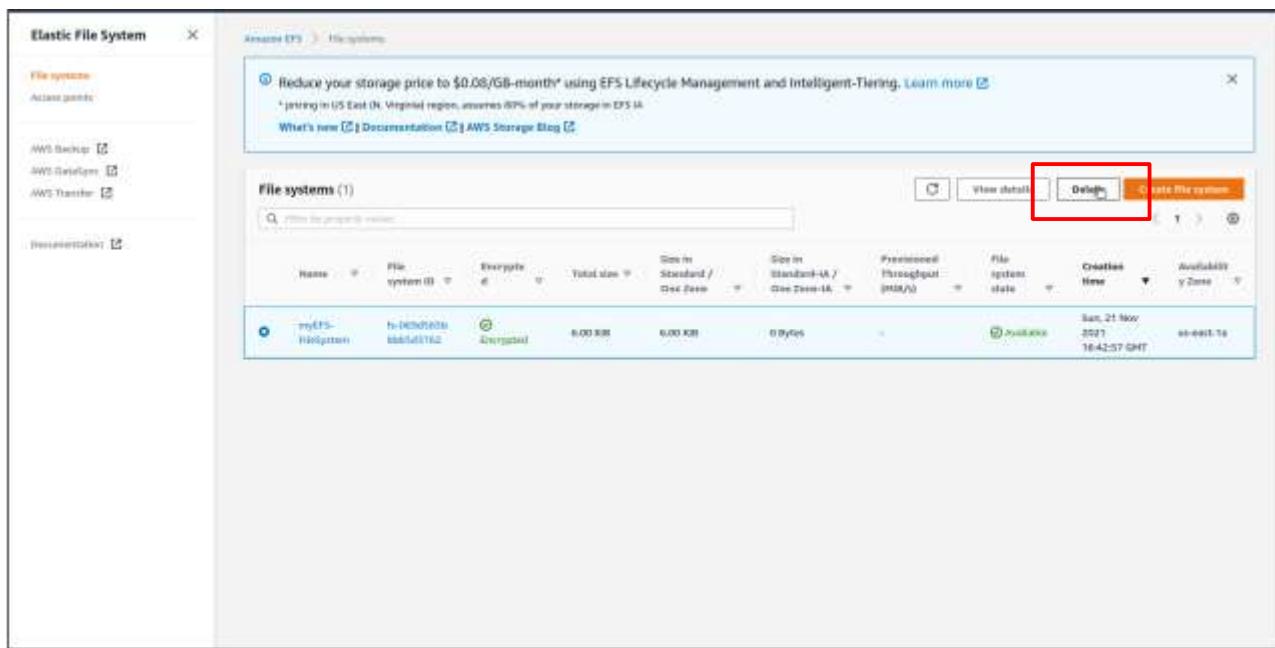
File system details:

Screenshot of the 'myEFS-FileSystem' (fs-069d565b8b65d3762) details page. The left sidebar shows navigation links: 'File systems', 'Access points', 'AWS Backup', 'AWS DataSync', and 'AWS Transfer'. The main area has tabs: 'General', 'Metered size', 'Monitoring', 'Tags', 'File system policy', 'Access points', and 'Network'. The 'General' tab is selected. It displays information: Performance mode (General Purpose), Throughput mode (Throughput), Lifecycle management (Transition in 10 days since last access, Transition out of 10 on first access), and Availability zone (us-east-1a). The 'Metered size' tab is selected. It displays: Total size (0.00 KiB), Storage in Standard / One Zone (8.00 KiB / 100%), and Storage in Standard IA / One Zone IA (0 bytes / 0%). A legend indicates: Blue for Storage in Standard / One Zone and Orange for Storage in Standard IA / One Zone IA.

## Removing an EFS File System

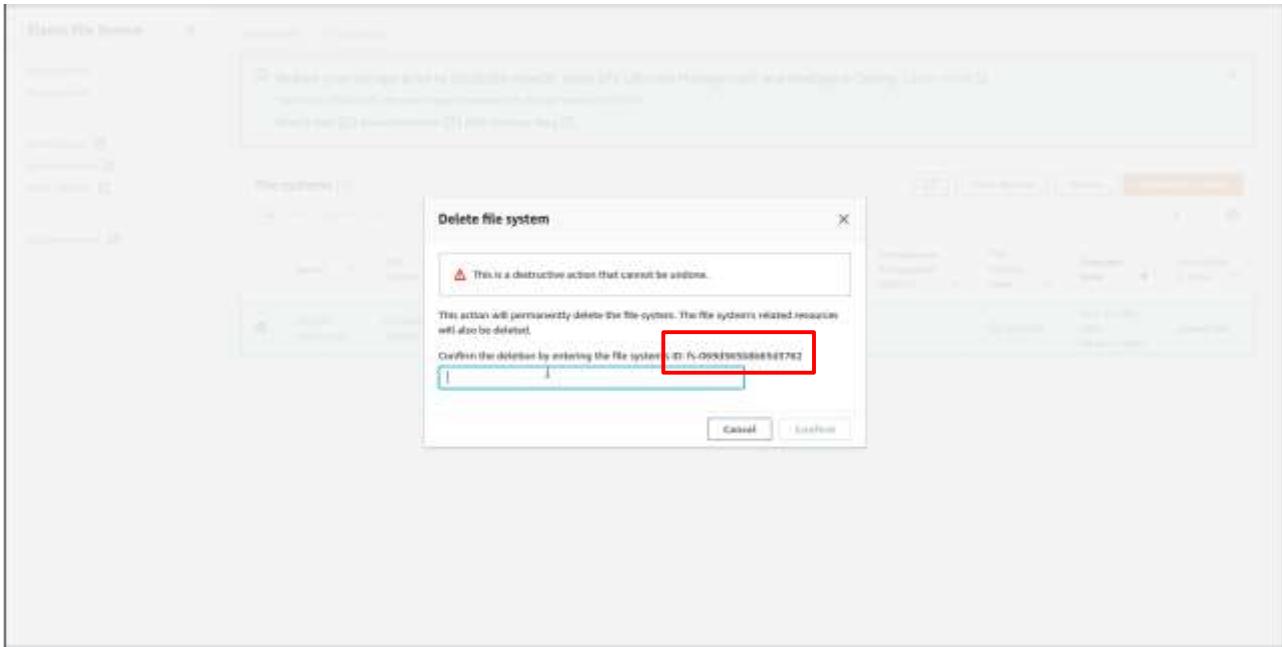
Before removing an EFS file system, it must be unmounted from the EC2 instances. This is done from inside the EC2 instance Command Line Interface – CLI.

To remove an EFS File System, select the file system. Then click the **Delete** button:

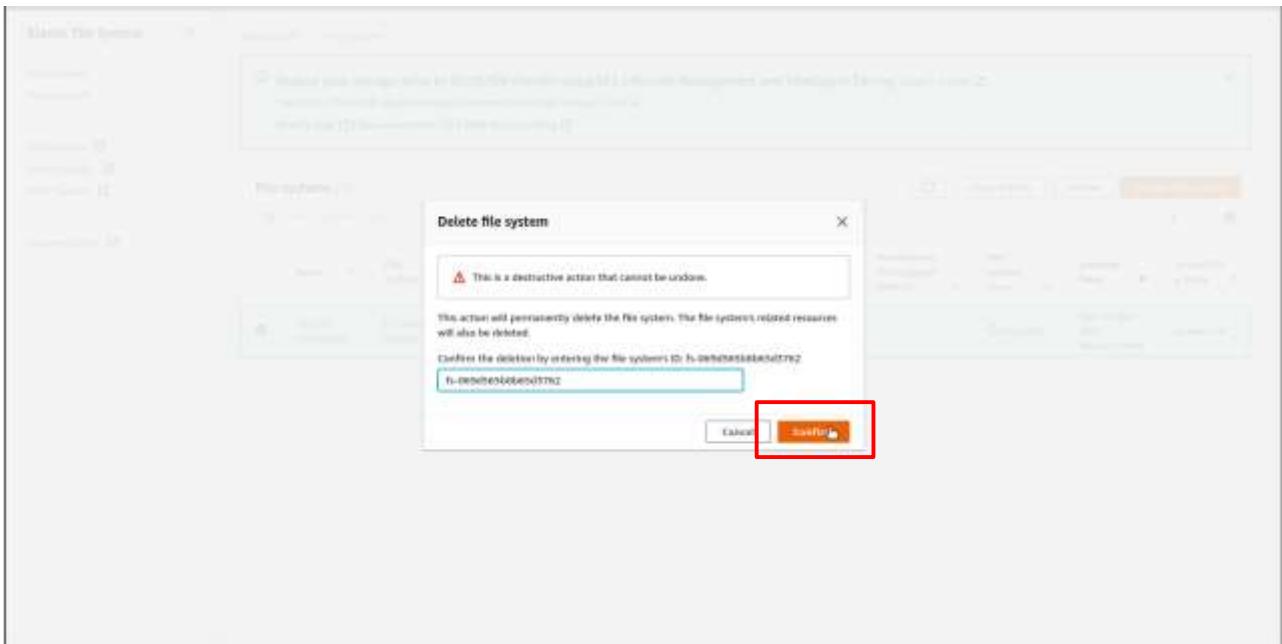


The screenshot shows the AWS EFS console with the 'File systems' list. The table has columns: Name, File system ID, Encrypted, Total size, Size in Standard / One Zone, Size in Standard-IA / One Zone-IA, Provisioned Throughput (MB/s), File system state, Creation time, and Availability Zone. A single row is selected, showing 'myefs-test' as the name, 'fs-0c90f600' as the file system ID, 'Unencrypted' as the encryption status, '0.00 GB' as the total size, '0.00 GB' as the size in Standard storage, '0.00 GB' as the size in Standard-IA storage, '0 Bytes' as the provisioned throughput, 'available' as the file system state, 'Sun, 21 Nov 2021 16:42:57 GMT' as the creation time, and 'us-east-1' as the availability zone. The 'Delete' button in the table header is highlighted with a red box.

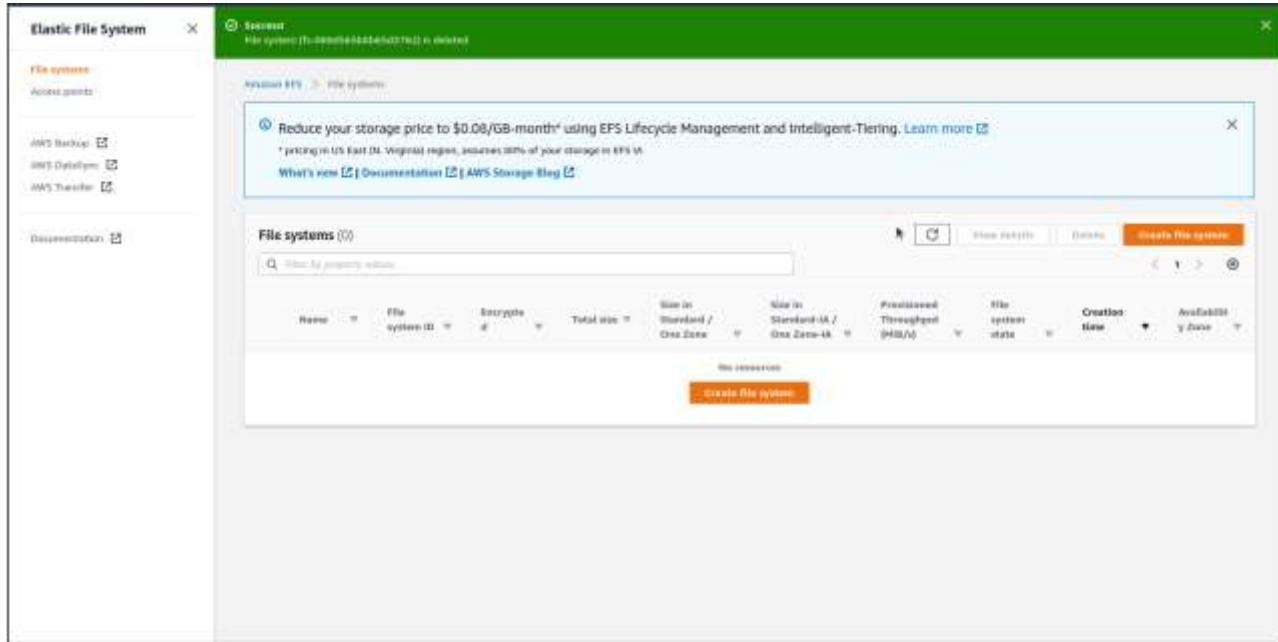
It will show the ID of the file system. Copy that and paste into the box:



After pasting the ID in the box, click on the Confirm button:



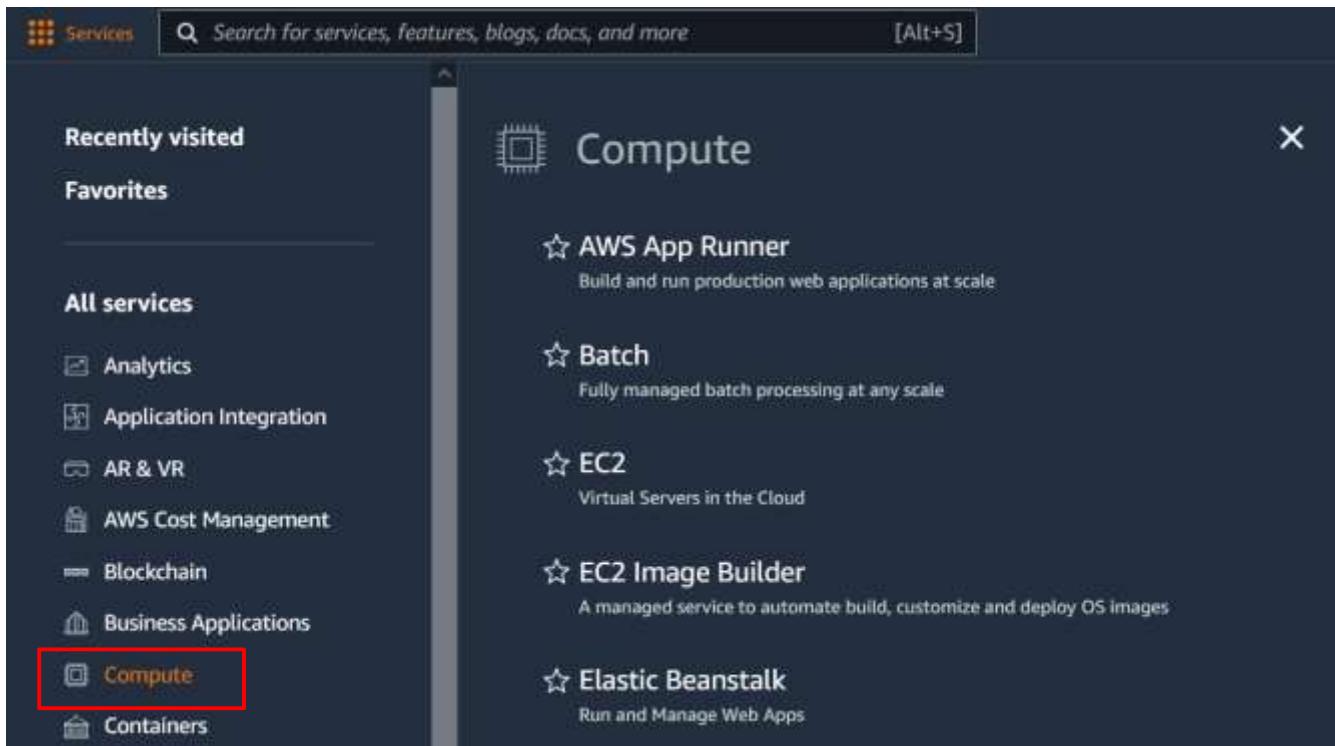
After some time, a green banner will appear on the top of the window informing the file system has been successfully deleted:



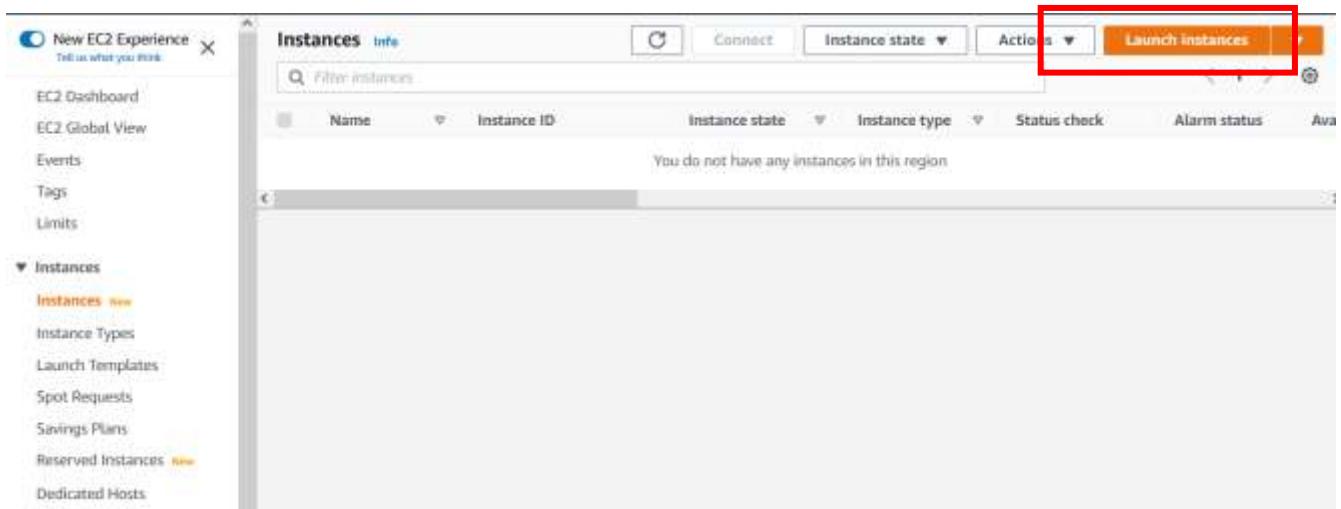
## Chapter 4 Labs

### Lab: Launch a Linux EC2 Instance and Set Up Linux, Apache, MySQL, PHP (LAMP) Stack

Log in to the AWS Management Console. Go to Services > Compute > EC2



Click on **Launch instance**:



Select Amazon Linux 2:

A screenshot of the 'Step 1: Choose an Amazon Machine Image (AMI)' wizard. The top navigation bar includes steps 1 through 7. On the left, a 'Quick Start' sidebar lists 'My AMIs', 'AWS Marketplace', and 'Community AMIs'. The main area shows a search bar and a list of AMIs. One item, 'Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-0d718c3d715dec4a7 (64-bit x86) / ami-d4fb4d2778000b2caa (64-bit Arm)', is selected and highlighted with a red box. To the right, there are 'Select' and 'Cancel and Exit' buttons, along with radio buttons for '64-bit (x86)' and '64-bit (Arm)'.

Select a free tier eligible instance type; click Review and Launch.

1. Choose AMI    2. Choose Instance Type    3. Configure Instance    4. Add Storage    5. Add Tags    6. Configure Security Group    7. Review

### Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance families ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2.micro (~ ECUs, 1 vCPUs, 2.5 GHz, ~ 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro	1	1	EBS only	-	Low to Moderate	Yes
	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
	t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel Previous **Review and Launch** Next: Configure Instance Details

Click on “6. Configure Security Group,” and add SSH, http, and https rules to the policy. Click Review.

1. Choose AMI   2. Choose Instance Type   3. Configure Instance   4. Add Storage   5. Add Tags   **6. Configure Security Group**   7. Review

### Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group:  Create a new security group  Select an existing security group

Security group name: launch-wizard-1

Description: launch-wizard-1 created 2021-11-20T15:48:32.167-06:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Anywhere	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom	e.g. SSH for Admin Desktop

Add Rule

Click Launch.

▼ AMI Details Edit AMI

 **Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-0d718c3d715cec4a7**

Free tier eligible

Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is no longer supported.

Root Device Type: ebs Virtualization type: hvm

▼ Instance Type Edit instance type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

► Security Groups Edit security groups

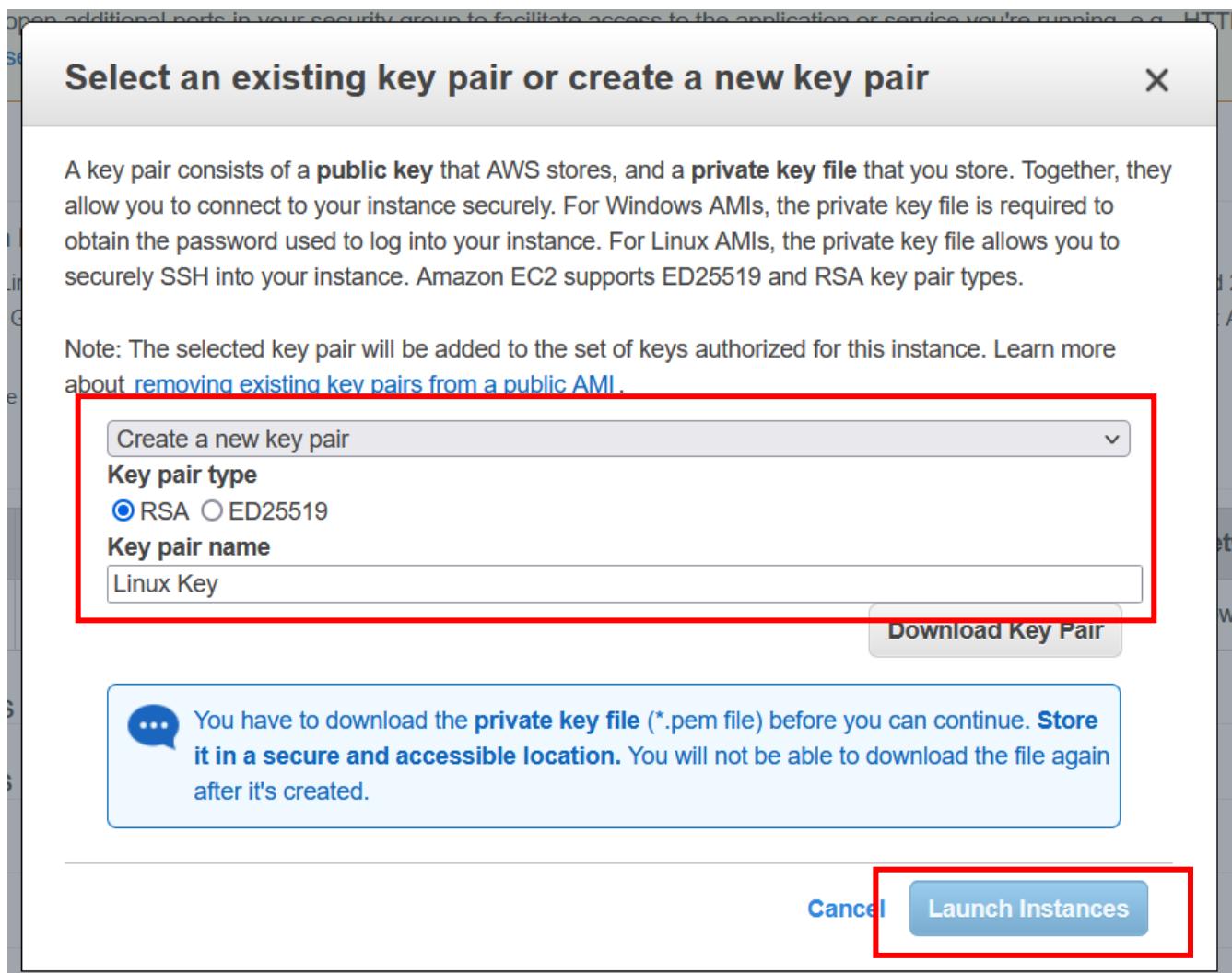
► Instance Details Edit instance details

► Storage Edit storage

► Tags Edit tags

Cancel Previous Launch

Create a new security key pair and download the key. Make sure you save it! Then click Launch Instance.



On the main EC2 page, click on the Instance ID, then click Connect, and click Connect again.

Instances (1) <a href="#">Info</a>		<a href="#">C</a>	<a href="#">Connect</a>	<a href="#">Instance state ▾</a>	<a href="#">Actions</a>
<a href="#">Filter instances</a>					
<input type="checkbox"/>	Name ▾	Instance ID	Instance state	Instance type	Status check
<input type="checkbox"/>	-	i-0ea5f935541f81eb7	<a href="#">Running</a>	t2.micro	<a href="#">2/2 checks passed</a>

EC2 > Instances > i-0ea5f935541f81eb7		
<b>Instance summary for i-0ea5f935541f81eb7</b>	<a href="#">Info</a>	<a href="#">Edit</a> <a href="#">Connect</a> <a href="#">Instances</a>
Updated less than a minute ago		
Instance ID	Public IPv4 address	Private IPv4 address
<a href="#">i-0ea5f935541f81eb7</a>	<a href="#">18.188.233.207   open address</a>	<a href="#">172.31.24.11</a>
IPv6 address	Instance state	Public IPv4 DNS
-	<a href="#">Running</a>	<a href="#">ec2-18-188-24-11.us-east-2.compute.internal</a>
Private IPv4 DNS	Instance type	Elastic IP addresses
<a href="#">ip-172-31-24-11.us-east-2.compute.internal</a>	t2.micro	-
VPC ID	AWS Compute Optimizer findings	IAM Role:

### Connect to instance [Info](#)

Connect to your instance i-0ea5f935541f81eb7 using any of these options

---

[EC2 Instance Connect](#) [Session Manager](#) [SSH client](#) [EC2 Serial Console](#)

---

Instance ID  
[i-0ea5f935541f81eb7](#)

Public IP address  
[18.188.233.207](#)

User name

Connect using a custom user name, or use the default user name ec2-user for the AMI used to launch the instance.

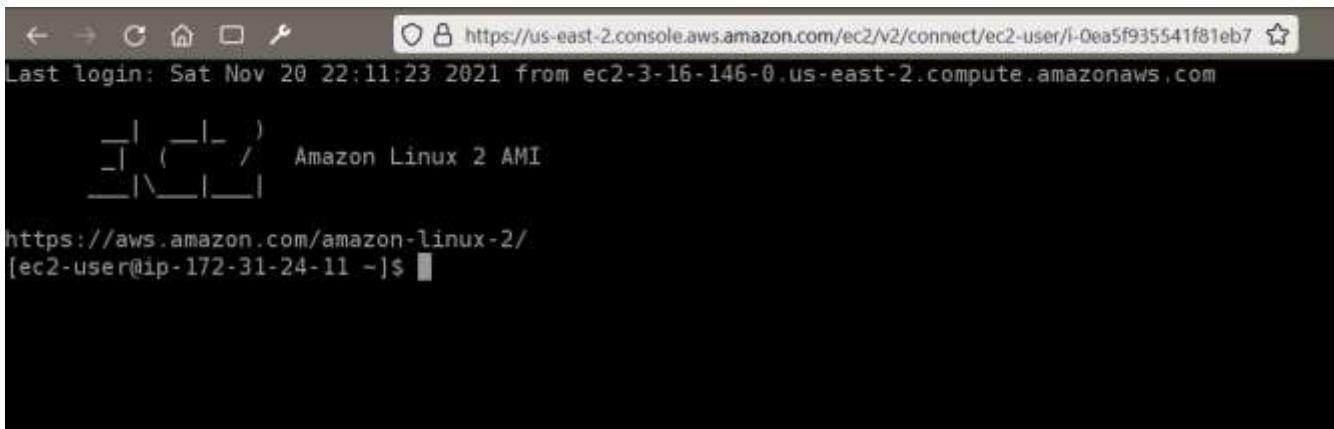
**Note:** In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

---

[Cancel](#) [Connect](#)

Congratulations! You've created an EC2 instance.

Go to the Linux command line.



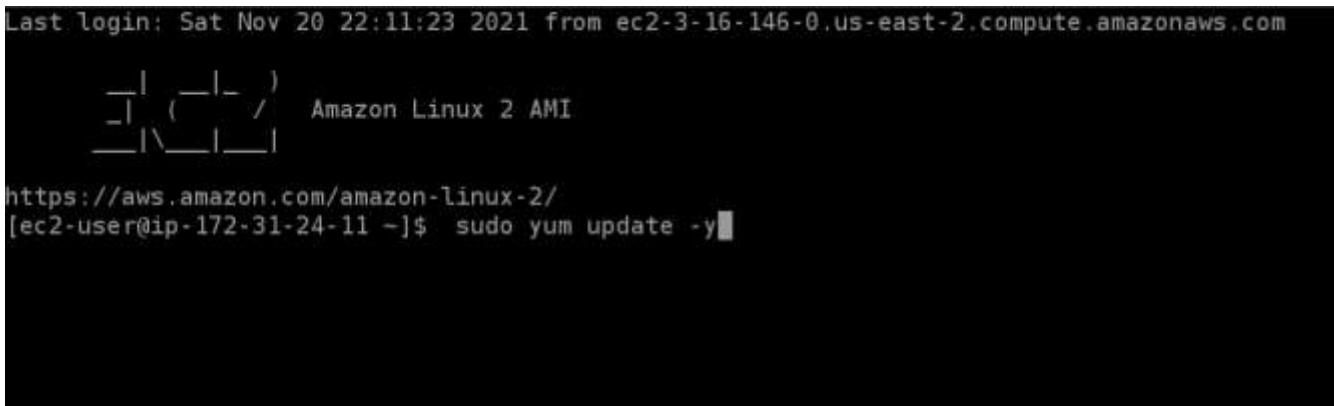
The screenshot shows the AWS Lambda function configuration page. At the top, there's a header with the function name and a 'Create Function' button. Below the header, there are sections for 'Environment Variables' and 'Code'. Under 'Environment Variables', there is one entry: 'AWS\_LAMBDA\_FUNCTION\_NAME' with the value 'lambda-function'. In the 'Code' section, there is a 'Edit' button next to a code editor window. The code editor contains the following Python code:

```
def lambda_handler(event, context):
    # Add your code here
    pass
```

Make sure all software packages are up to date.

Type the following at the command prompt:

```
sudo yum update -y
```



The screenshot shows the AWS Lambda function configuration page. At the top, there's a header with the function name and a 'Create Function' button. Below the header, there are sections for 'Environment Variables' and 'Code'. Under 'Environment Variables', there is one entry: 'AWS\_LAMBDA\_FUNCTION\_NAME' with the value 'lambda-function'. In the 'Code' section, there is a 'Edit' button next to a code editor window. The code editor contains the following Python code:

```
def lambda_handler(event, context):
    # Add your code here
    pass
```

Install MariaDB (Substituted for MySQL) and PHP Amazon Linux Extras repositories.

Type the following at the command prompt:

```
sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
```



```
[ec2-user@ip-172-31-24-11 ~]$ sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
```

Install the Apache, MariaDB, and PHP software packages.

Type the following at the command prompt:

```
yum install -y httpd mariadb-server
```



```
[ec2-user@ip-172-31-24-11 ~]$ sudo yum install -y httpd mariadb-server
```

Start the Apache web server.

Type the following at the command prompt:

sudo systemctl start httpd

```
[ec2-user@ip-172-31-24-11 ~]$ sudo systemctl start httpd
```

Set Apache to automatically start after a reboot

Type the following at the command prompt:

sudo systemctl enable httpd

```
[ec2-user@ip-172-31-24-11 ~]$ sudo systemctl enable httpd
```

Create an index.html file for the homepage of the website.

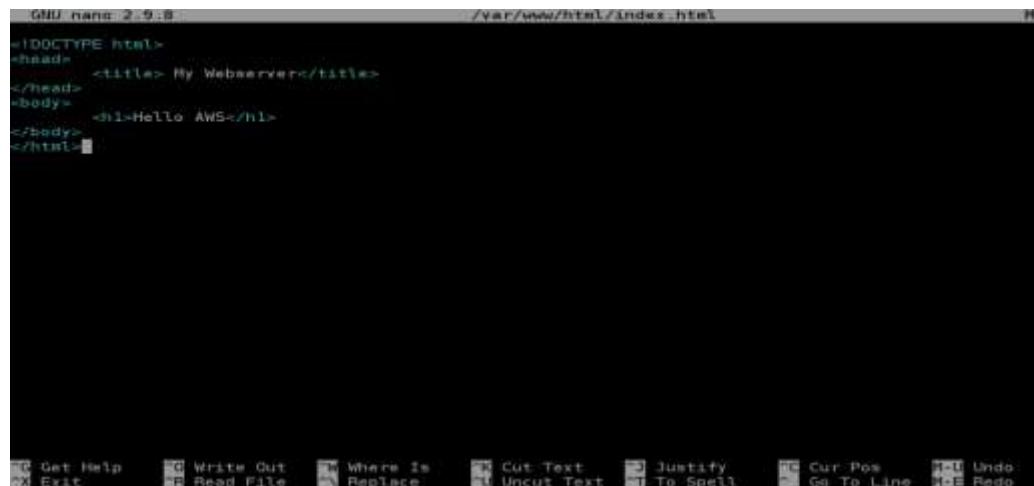
Type the following at the command prompt:

sudo nano /var/www/html/index.html

```
[ec2-user@ip-172-31-24-11 ~]$ sudo nano /var/www/html/index.html
```

Insert the following and save the file with Control+O. Exit nano with Control+X.

```
<!DOCTYPE html>
<head>
    <title>My Web server</title>
</head>
<body>
    <h1>Hello AWS</h1>
</body>
</html>
```



The screenshot shows a terminal window with the nano text editor open. The file being edited is located at /var/www/html/index.html. The code inside the file is:

```
GNU nano 2.9.8
<!DOCTYPE html>
<head>
    <title> My Webserver</title>
</head>
<body>
    <h1>Hello AWS</h1>
</body>
</html>
```

At the bottom of the screen, there is a menu bar with various options: Get Help, Write Out, Where Is, Cut Text, Justify, Cur Pos, Undo, Exit, Read File, Replace, Uncut Text, To Spell, Go To Line, and Redo.

Navigate to the public DNS record for your server.

The screenshot shows the AWS EC2 Instances page. The top navigation bar includes 'EC2 > Instances > i-0ea5f935541f81eb7'. Below the navigation is a header with 'Instance summary for i-0ea5f935541f81eb7' and 'Info' buttons. The main content area displays various instance details in a grid format:

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0ea5f935541f81eb7	18.188.233.207   <a href="#">open address</a>	172.31.24.11
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-18-188-233-207.us-east-2.compute.amazonaws.com   <a href="#">open address</a>
Private IPv4 DNS	Instance type	Elastic IP addresses
ip-172-31-24-11.us-east-2.compute.internal	t2.micro	-
VPC ID	AWS Compute Optimizer finding	IAM Role
vpc-074561f609493c230	Opt-in to AWS Compute Optimizer for recommendations.   <a href="#">Learn more</a>	-
Subnet ID		
subnet-0244cd4b6adf7e416		

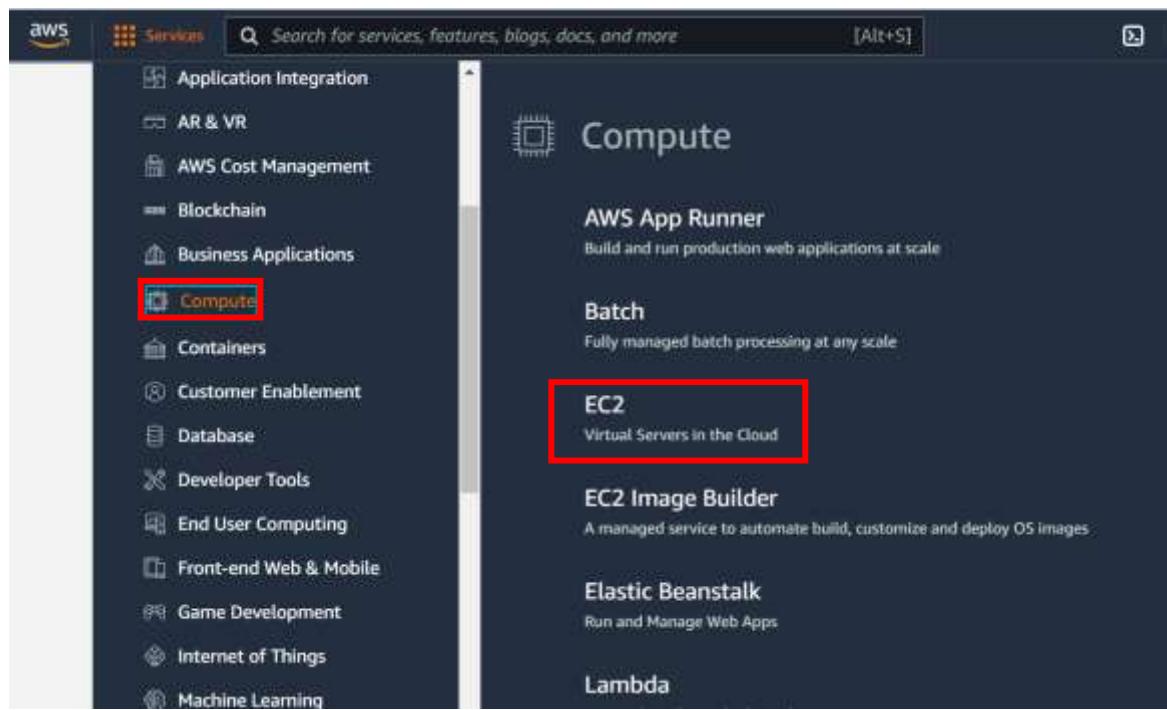
Congratulations, you've set up a LAMP server on an EC2 instance.



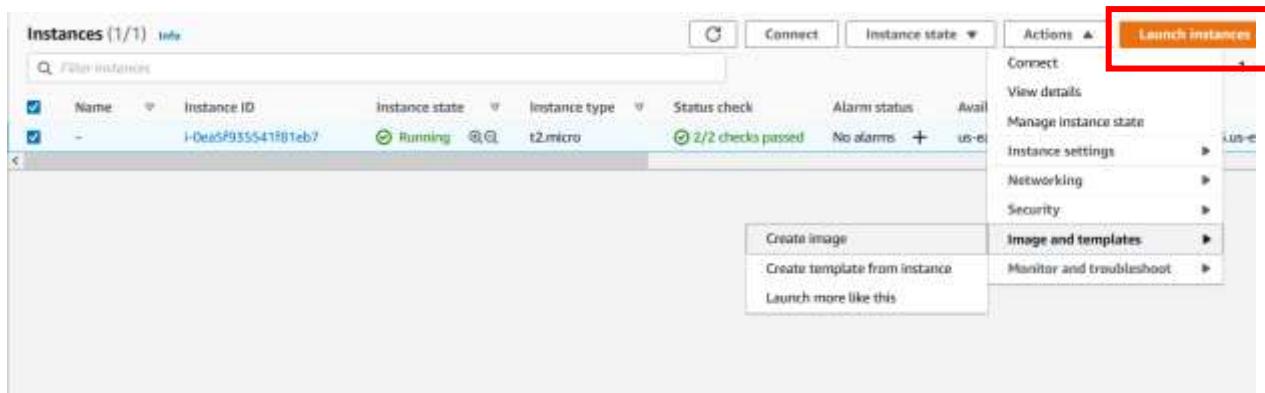


## Lab: Create an Amazon Machine Image (AMI) of Our Linux Web Server

Log in to the AWS Console, Go to Services > Compute > EC2.



Select the server and click on Actions > Image and templates > Create Image



Give the AMI a name and description. Then click **Create Image**.

EC2 > Instances > i-0ea5f935541fb1eb7 > Create image

**Create image** Info

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Instance ID  
i-0ea5f935541fb1eb7

**Image name:** MyWebserver  
Maximum 127 characters. Can be modified after creation.

**Image description - optional:** Apache webserver  
Maximum 255 characters

**Enabled:**

**Instance volumes:**

Volume type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/xvda	Create new snapshot from this volume	8	EBS General Purpose SSD	100	100 MiB/s	<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Encrypted

Add volume

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

**Tags - optional:** A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Tag image and snapshots together  
Tag the image and the snapshots with the same tag.

Tag image and snapshots separately  
Tag the image and the snapshots with different tags.

No tags associated with the resource.

Add tag

You can add 50 more tags.

Cancel **Create image**

Go to the AMIs page. It may take a few minutes for the AMI to go from ‘pending’ to ‘available.’ Select the AMI and click **Launch**.

Instances

Instances New

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances New

Dedicated Hosts

Capacity Reservations

Images

AMIs

Elastic Block Store

Launch **EC2 Image Builder** Actions

Owned by me

Name	AMI Name	AMI ID	Source	Owner	Visibility	Status
MyWebserver	ami-04fbff91c1794684d	471531149836/	471531149836	Private	available	

## Choose a Free Tier Eligible Instance type. Click Review and Launch.

### Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter By:	All instance families	Current generation	Show/Hide Columns					
Currently selected: t2.micro (1 EC2.Ul, 1 vCPU, 2.9 GHz, ~ 1 GB memory, EBS only)								
Family	Type	vCPUs	Memory (GB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support	
t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes	
t2	<b>t2.micro</b> <small>(Selected)</small>	1	1	EBS only	-	Low to Moderate	Yes	
t2	t2.small	2	2	EBS only	-	Low to Moderate	Yes	
t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes	
t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes	
t2	t2.xlarge	4	16	EBS only	-	Medium	Yes	
t2	t2.2xlarge	8	32	EBS only	-	Medium	Yes	
t3	t3.nano	2	0.5	EBS only	Yes	Up to 6 Gigabit	Yes	
t3	t3.micro	2	1	EBS only	Yes	Up to 6 Gigabit	Yes	
t3	t3.small	2	2	EBS only	Yes	Up to 6 Gigabit	Yes	
t3	t3.medium	2	4	EBS only	Yes	Up to 6 Gigabit	Yes	
t3	t3.large	2	8	EBS only	Yes	Up to 6 Gigabit	Yes	
t3	t3.xlarge	4	16	EBS only	Yes	Up to 6 Gigabit	Yes	
t3	t3.2xlarge	8	32	EBS only	Yes	Up to 6 Gigabit	Yes	
Da	Da.nano	2	0.5	EBS only	Yes	Up to 6 Gigabit	Yes	

Cancel Previous **Review and Launch** Next: Configure Instance Details

Click on **6. Configure Security Group** and select the existing security group for the previously created web server. *Note: Use default values for “3. Configure Instance, 4. Add Storage, and 5. Add Tags”. Click Review and Launch.*

1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
- 6. Configure Security Group**
7. Review

1. Choose AMI   2. Choose Instance Type   3. Configure Instance   4. Add Storage   5. Add Tags   6. Configure Security Group   7. Review

### Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon VPC security groups.

Assign a security group:  Create a new security group  Select an existing security group

Security Group ID	Name	Description	Actions
sg-0c3613bed2d760	default	default VPC security group	<a href="#">Copy to new</a>
sg-0482ff8674fb13c	launch-wizard-1	launch-wizard-1 created 2021-11-20T15:48:32.167-06:00	<a href="#">Copy to new</a>

Inbound rules for sg-0482ff8674fb13c (Selected security groups: sg-0482ff8674fb13c)

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0	
HTTP	TCP	80	::0	
SSH	TCP	22	0.0.0.0	
SSH	TCP	22	::0	
HTTPS	TCP	443	0.0.0.0	
HTTPS	TCP	443	::0	

[Cancel](#) [Previous](#) [Review and Launch](#)

Review the AMI and Security Group details. Click **Launch** the instance.

1. Choose AMI   2. Choose Instance Type   3. Configure Instance   4. Add Storage   5. Add Tags   6. Configure Security Group   7. Review

### Step 7: Review Instance Launch

AMI Details

**MyWebserver - ami-04f9f191c1794684d**

Apache webserver  
Root Device Type: sda1   Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

Security Groups

Security Group ID	Name	Description
sg-0482ff8674fb13c	launch-wizard-1	launch-wizard-1 created 2021-11-20T15:48:32.167-06:00

All selected security groups inbound rules:

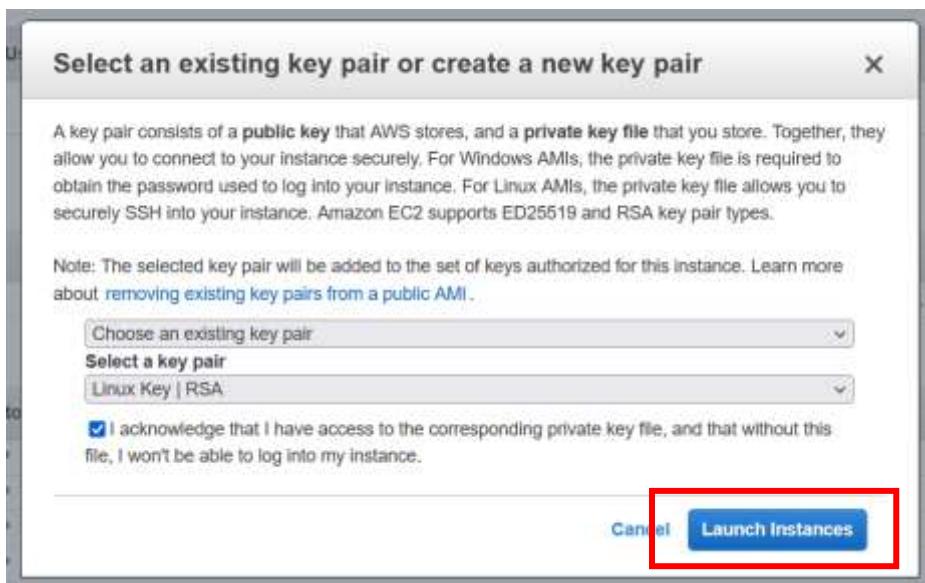
Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0	
HTTP	TCP	80	::0	
SSH	TCP	22	0.0.0.0	
SSH	TCP	22	::0	
HTTPS	TCP	443	0.0.0.0	
HTTPS	TCP	443	::0	

Instance Details

Storage

[Cancel](#) [Previous](#) [Launch](#)

Make sure you have the key pair downloaded from the previous lab, or create a new key pair. Click Launch Instances.



Congratulations! You've created and launched an AMI. You can check the status and view the instances.

## Launch Status

### Your instances are now launching

The following Instance launches have been initiated: i-031dceb8f272d1b1a · [View launch log](#)

### Get notified of estimated charges

[Create billing alerts](#) to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

## How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the running state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click [View Instances](#) to monitor your instances' status. Once your instances are in the running state, you can connect to them from the Instances screen. [Find out how to connect to your instances.](#)

### Here are some helpful resources to get you started

- [How to connect to your Linux instance](#)
- [Amazon EC2: User Guide](#)
- [Learn about AWS Free Usage Tier](#)
- [Amazon EC2: Discussion Forum](#)

While your instances are launching you can also

- [Create status check alarms to be notified when these instances fail status checks. \(Additional charges may apply\)](#)
- [Create and attach additional EBS volumes \(Additional charges may apply\)](#)
- [Manage security groups](#)

[View Instances](#)

## Lab: Create a Security Group for an EC2 Instance

From the management console, navigate to the VPC console.

Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.

Choose **Security Groups** in the navigation pane.



Choose **Create security group**.

Specify a name and description for the security group.

**Note:** Once created, security group names and descriptions cannot be changed. Choose the target VPC from **VPC**.

Security group inbound and outbound **Rules** can be added now or later.

The screenshot shows the 'Create security group' wizard. In the 'Basic details' section, the security group name is 'MySecurity Group' and the description is 'My first security group'. The VPC selected is 'vpc-0ff14cbf24a08823'. The 'Inbound rules' section contains two rules: one for HTTP (port 80) and one for HTTPS (port 443), both from 'Anywhere - IPv4' (0.0.0.0/0). The 'Outbound rules' section shows a single rule for 'All traffic' to 'Custom' destination (0.0.0.0/0).

Tags can be added now or later. Choose **Add new tag** and enter the tag key and value. Choose **Create security group**.

The screenshot shows the 'Create subnet' wizard. In the 'VPC' section, the 'Select a VPC' dropdown is highlighted with a red box. The 'Subnet settings' section has a note: 'Specify the CIDR blocks and Availability Zone for the subnet.' It also says 'Select a VPC first to create new subnets.' The 'Add new subnet' button is highlighted with a red box. At the bottom right, there are 'Cancel' and 'Create subnet' buttons, with 'Create subnet' highlighted with a red box.

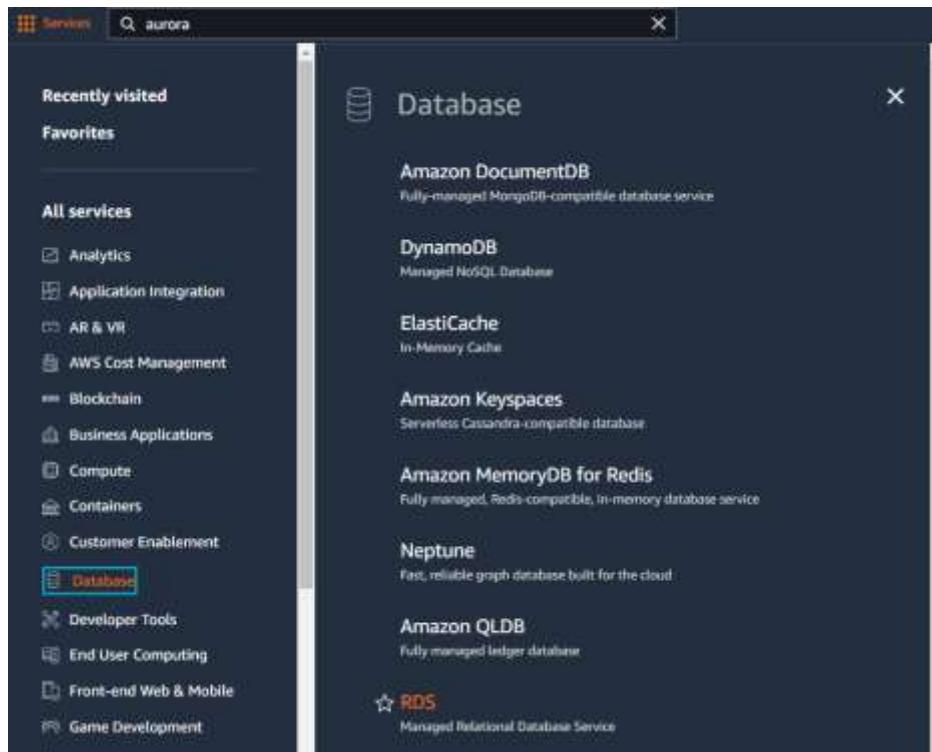
Select the newly created VPC (MYVPC).

## Chapter 5 Labs

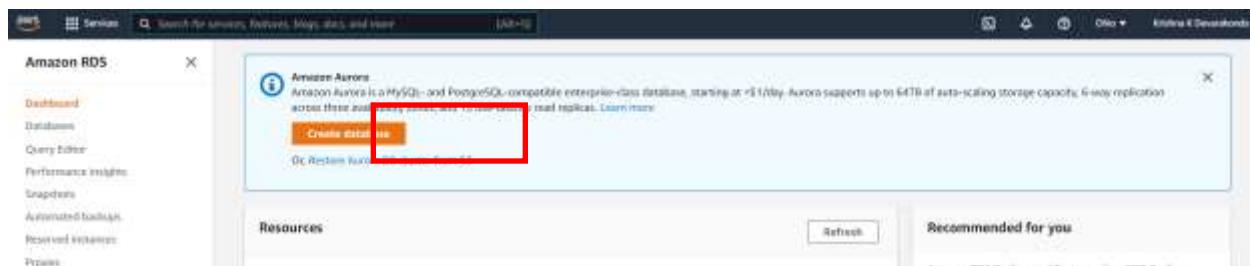
### Lab: Create an Amazon Aurora (Relational Database)

Log in to AWS Management Console.

Navigate to Services □ Database □ RDS databases as shown below:



Click on Create Database under Amazon Aurora.



Pick defaults for everything except for Availability & Durability.

**Availability & durability**

Multi-AZ deployment [Info](#)

- Create an Aurora Replica or Reader node in a different AZ (recommended for scaled availability)  
Creates an Aurora Replica for fast failover and high availability.
- Don't create an Aurora Replica

Under Advanced configuration, enter “Initial database name”,

▼ Additional configuration

Database options, encryption enabled, failover, backup enabled, backtrack disabled, Performance Insights enabled, Enhanced Monitoring enabled, maintenance, CloudWatch Logs, delete protection disabled.

**Database options**

Initial database name [Info](#)

GoCloud

If you do not specify a database name, Amazon RDS does not create a database.

Check “Log exports” as needed, check “Deletion protection”.

## Log exports

Select the log types to publish to Amazon CloudWatch Logs

- Audit log
- Error log
- General log
- Slow query log

## IAM role

The following service-linked role is used for publishing logs to CloudWatch Logs.

RDS service-linked role

i Ensure that general, slow query, and audit logs are turned on. Error logs are enabled by default. [Learn more](#)

## Maintenance

Auto minor version upgrade [Info](#)

- Enable auto minor version upgrade

Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the database.

## Maintenance window [Info](#)

Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.

- Select window
- No preference

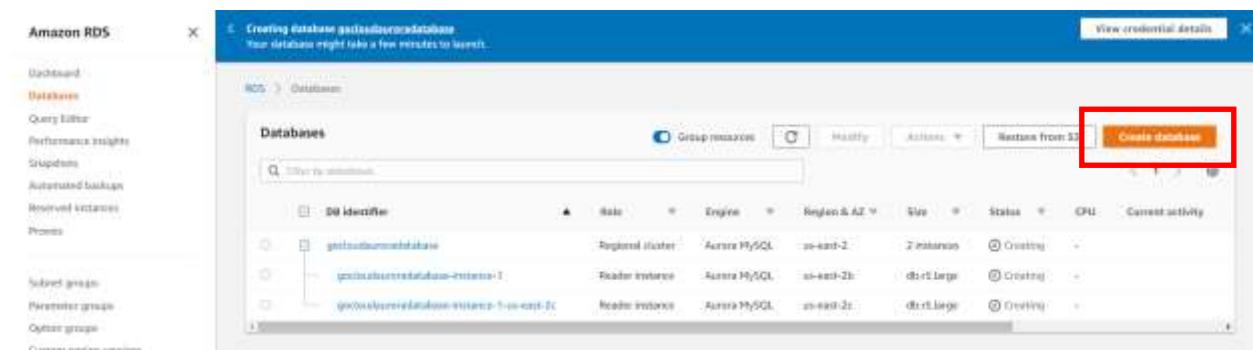
## Deletion protection

- Enable deletion protection

Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.

---

Click on “Create Database” and you will land on RDS Databases dashboard with “Creating” status



The screenshot shows the Amazon RDS Databases dashboard. A modal window titled "Creating database gestationurldatabase" is open, stating "Your database might take a few minutes to launch." Below the modal, the main dashboard lists three databases:

ID	Engine	Region & AZ	Status
gestationurldatabase	Aurora MySQL	us-east-2	2 instances
gestationurldatabase-instance-1	Aurora MySQL	us-east-2	db.t2.large
gestationurldatabase-instance-2-us-east-2	Aurora MySQL	us-east-2	db.t2.large

A red box highlights the "Create database" button at the bottom right of the main dashboard area.

Click on the database, and navigate to “Configuration” tab to view the configuration of your database.

Database			
Configuration	Capacity type	Availability	Encryption
DB cluster role	Provisioned: single-master	IAM DB authentication	Encryption
Regional cluster	DB cluster ID	Not enabled	Enabled
Engine version	gocloudauroradatabase	Master username:	AWS KMS key
5.7.mysql_aurora.2.07.2	DB cluster parameter group	admin	aws/rds
Resource ID	default.aurora-mysql5.7	Master password:	
cluster-5240UMKNNMIP6FVIP4D3YU4URQ	Deletion protection	*****	Published logs
Amazon Resource Name (ARN)	Enabled	Multi-AZ	CloudWatch Logs
arn:aws:rds:us-east-1:110680399916:cluster:gocloudauroradatabase		2 Zones	Audit
			Error
			General
			Slow query

On the “Connectivity & security” tab, under “Manage IAM roles”, check “Select IAM roles to add to this cluster” option and choose “AWSServiceRoleForRDS”. Click on “Add Role”.

Current IAM roles for this cluster (1)	
Role	Status
AWSServiceRoleForRDS	Pending

To Create database across multiple regions, we can choose the below option.

If you run into DB Subnet group error, follow steps 11 to 14. Otherwise, skip to step 15.

## Network & Security

### Destination region

The region in which the replica will be launched

US West (N. California)

### Destination DB subnet group



**The Amazon VPC and DB subnet group required for the Read Replica do not exist in the destination region.**

[Create an Amazon VPC and DB subnet group in the destination region.](#)

### Publicly accessible

Yes

EC2 instances and devices outside of the VPC hosting the DB instance will connect to the DB instances. You must also select one or more VPC security groups that specify which EC2 instances and devices can connect to the DB instance.

No

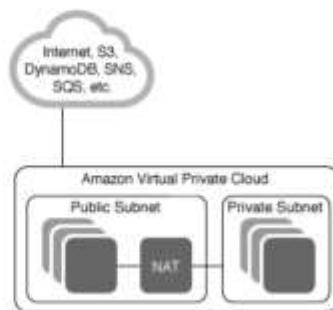
DB instance will not have a public IP address assigned. No EC2 instance or devices outside of the VPC will be able to connect.

Create new VPC under cross regions (here we picked us-west).

### Step 1: Select a VPC Configuration

VPC with a Single Public Subnet

In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet. Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation (NAT).



VPC with Public and Private Subnets

Creates:  
A /16 network with two /24 subnets. Public subnet instances use Elastic IPs to access the Internet. Private subnet instances access the Internet via Network Address Translation (NAT). (Hourly charges for NAT devices apply.)

Select

VPC with Public-and-Private Subnets and Hardware VPN Access

VPC with a Private Subnet Only and Hardware VPN Access

Step 2: VPC with a Single Public Subnet

IPv4 CIDR Block:	(10.0.0.0/16) (95371 IP addresses available)
IPv6 CIDR block:	<input checked="" type="radio"/> No IPv6 CIDR block <input type="radio"/> Amazon provided IPv6 CIDR block <input type="radio"/> IPv6 CIDR block owned by me
VPC name:	GoCloudVPC
Public subnet's IPv4 CIDR:	(10.0.0.0/24) (251 IP addresses available)
Availability Zone:	No Preference
Subnet name:	Public subnet
You can add more subnets after Amazon Web Services creates the VPC.	
Service endpoints:	<a href="#">Add Endpoint</a>
Enable DNS hostnames:	<input checked="" type="radio"/> Yes <input type="radio"/> No
Hardware tenancy:	Default
<a href="#">Cancel and Exit</a> <a href="#">Back</a> <a href="#">Create VPC</a>	

Create a new DB subnet group.

Amazon RDS

RDS > Subnet groups

Subnet groups (1)

[Create new subnet group](#)

Name	Description	Status	VPC
default	default	Complete	vpc-11686478

**Amazon RDS**

- Dashboard
- Databases
- Query Editor
- Performance insights
- Snapshots
- Automated backups
- Reserved instances
- Proxies

**Subnet groups**

- Parameter groups
- Option groups

Events

Event subscriptions

Recommendations 0

Certificate update

## Create DB subnet group

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

### Subnet group details

**Name**  
You won't be able to modify the name after your subnet group has been created.

Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

**Description**

**VPC**  
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.

### Add subnets

**Availability Zones**  
Choose the Availability Zones that include the subnets you want to add.

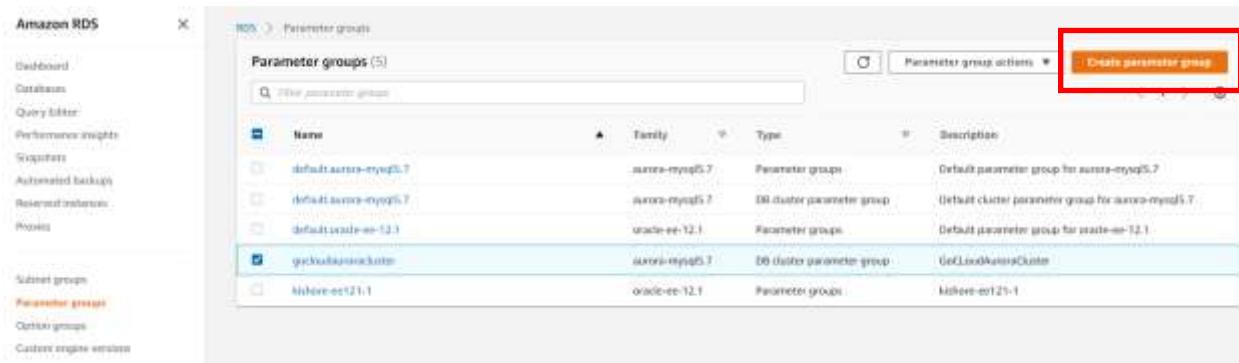
**Subnets**  
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.

### Subnets selected (2)

Availability zone	Subnet ID	CIDR block
us-west-1a	subnet-05c351e63a1b6680e	10.0.0.0/24
us-west-1c	subnet-01a501b1526a6648e	10.0.1.0/24

**Cancel** **Create**

## Create DB cluster group.

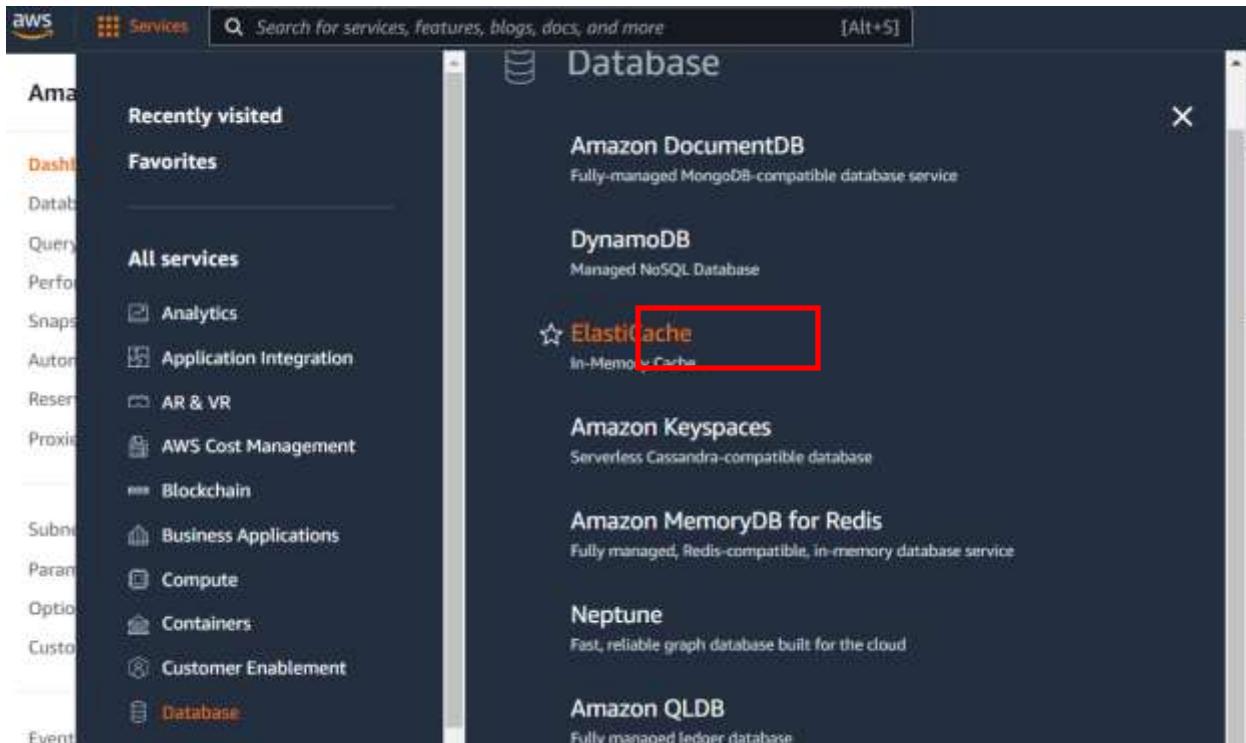


The screenshot shows the 'Parameter groups' section of the Amazon RDS console. On the left, there's a sidebar with various navigation links like Dashboard, Databases, Query editor, etc. The main area has a table titled 'Parameter groups (5)'. The table includes columns for Name, Family, Type, and Description. One row is selected, showing 'getCloudWatchLogs' as the name, 'aurora-mysql5.7' as the family, 'DB cluster parameter group' as the type, and 'GetCloudWatchLogs' as the description. A red box highlights the 'Create parameter group' button at the top right of the table area.

Name	Family	Type	Description
default.aurora-mysql5.7	aurora-mysql5.7	Parameter group	Default parameter group for aurora-mysql5.7
default.aurora-mysql5.7	aurora-mysql5.7	DB cluster parameter group	(Default cluster parameter group for aurora-mysql5.7)
default.oracle-ee-12.1	oracle-ee-12.1	Parameter group	Default parameter group for oracle-ee-12.1
getCloudWatchLogs	aurora-mysql5.7	DB cluster parameter group	GetCloudWatchLogs
kielene-ee12.1	oracle-ee-12.1	Parameter group	kielene-ee12.1

Modify the Aurora Database configuration to use the new DB cluster group. Stop & Start the RDS main cluster instance after modifying the configuration for synchronizing before creating the “Cross-region Replica”.

## Lab: Set Up Elastic Cache For Database Caching

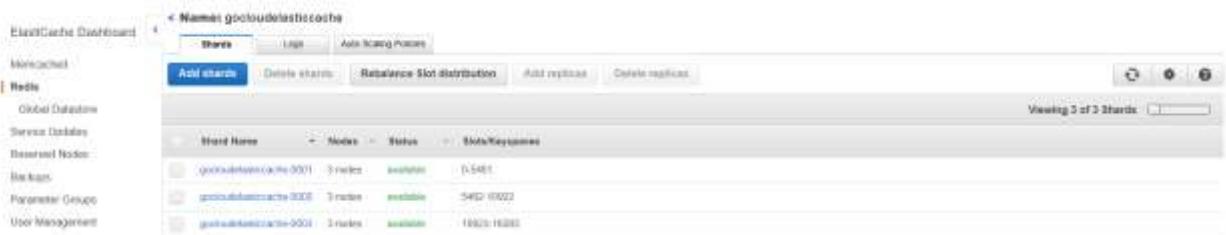


Create a new subnet group with the existing subnets in that AZ.

The screenshot shows the 'Create new' dialog for a Subnet group. The 'Name' field is set to 'GoCloudElasticCacheSubnet'. The 'Description' field is set to 'GoCloudElasticCacheSubnet'. The 'VPC ID' is set to 'vpc-11686478'. The 'Subnets' section lists three subnets: 'subnet-eb160b82' (Availability zone: us-east-2a, CIDR Block: 172.31.0.0/20), 'subnet-18227063' (Availability zone: us-east-2b, CIDR Block: 172.31.16.0/20), and 'subnet-fabc23b7' (Availability zone: us-east-2c, CIDR Block: 172.31.32.0/20).

Subnet ID	Availability zone	CIDR Block
subnet-eb160b82	us-east-2a	172.31.0.0/20
subnet-18227063	us-east-2b	172.31.16.0/20
subnet-fabc23b7	us-east-2c	172.31.32.0/20

Once created, it should look like this:

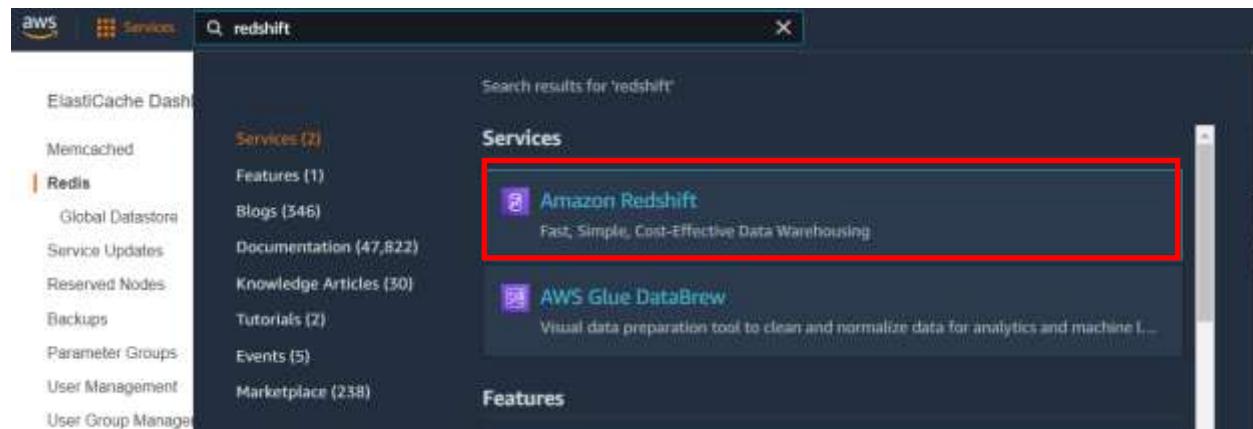


The screenshot shows the Redis dashboard for a cluster named 'gocloudelasticsearch'. The left sidebar includes options like 'ElastCache Dashboard', 'Monitored', 'Redis' (which is selected), 'Global Database', 'Service Catalog', 'Banned Nodes', 'Metrics', 'Parameter Groups', and 'User Management'. The main area displays three shards under the 'Shards' tab. Each shard is represented by a row in a table with columns: Shard Name, Nodes, Status, and Slave Keyspaces. The first shard has 3 nodes, is available, and has 1,5481 slave keyspaces. The second shard has 3 nodes, is available, and has 542 10022 slave keyspaces. The third shard has 3 nodes, is available, and has 19923 16283 slave keyspaces. There are tabs for 'Logs' and 'Auto Scaling Policies' at the top, and buttons for 'Add shards', 'Delete shards', 'Reschedule Slot distribution', 'Add replicas', and 'Delete replicas'.

Shard Name	Nodes	Status	Slave Keyspaces
gocloudelasticsearch-0001	3 nodes	available	1,5481
gocloudelasticsearch-0002	3 nodes	available	542 10022
gocloudelasticsearch-0003	3 nodes	available	19923 16283

## Lab: Set up an Amazon Redshift Data Warehouse

Select Redshift service.



Create Cluster by specifying a username and password.

A screenshot of the "Create cluster" configuration page in the Amazon Redshift console. The left sidebar has a "CLUSTERS" tab selected, which is highlighted in orange. The main area shows the "Cluster configuration" section. Under "Cluster identifier", the value "gocloud-redshift-cluster-1" is entered into a text input field. A note below says, "The identifier must be from 1-63 characters. Valid characters are a-z (lowercase only) and - (hyphen)." In the "What are you planning to use this cluster for?" section, there are two options: "Production" and "Free trial". The "Free trial" option is selected and highlighted with a blue border. A note next to it says, "Configure for learning about Amazon Redshift. This configuration is free for a limited time if your organization has never created an Amazon Redshift cluster." At the bottom, there is a callout box with the text: "When the free trial ends, delete your cluster to avoid incurring charges at on-demand rate for compute and storage. If you want to take a final snapshot of your cluster and store the snapshot on an S3, our on-demand rate applies." The "DASHBOARD", "QUERIES", "EDITOR", "DATASHARES", "CONFIG", "MIGRATE", and "ANSWER" tabs are also visible in the sidebar.

Once RedShift is created, it should look like this:

The screenshot shows the 'Clusters' page in the AWS Management Console. A single cluster, 'gocloud-redshift-cluster-1', is listed. It has an 'Available' status, d2 large node type, and 35733850.2186-4180 storage capacity. The cluster was created on November 23, 2021.

To Connect to the database and run queries, choose “Query in query editor v2” option.

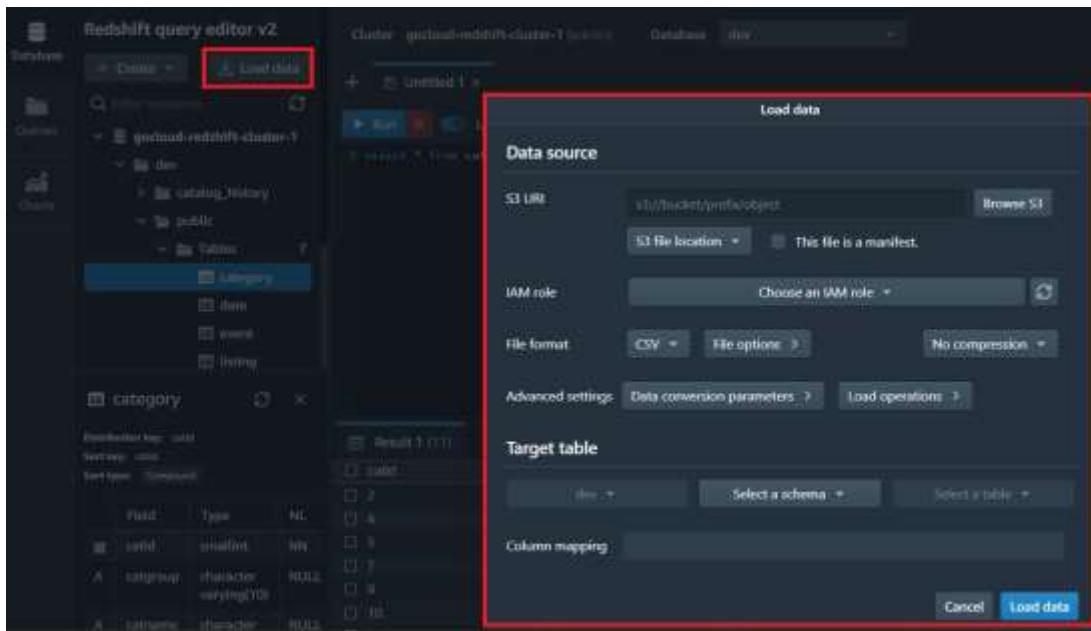
The screenshot shows the 'Cluster' details page for 'gocloud-redshift-cluster-1'. The 'Query in query editor v2' button is highlighted with a red box. Other options shown include 'Query in query editor' and 'Edit'.

Querying the database using V2 editor.

The screenshot shows the Redshift query editor v2 interface. A query is being run against the 'category' table in the 'dev' database. The results show 11 rows of data, including categories like Sports, NBA, NFL, etc., with their descriptions and codes.

code	categori	catname	catdesc
2	Sports	NFL	National Hockey League
4	Sports	NBA	National Basketball Assoc
3	Sports	MLS	Major League Soccer
7	Shows	Play	All rock and pop music co
9	Concerts	Jazz	All jazz singers and bands
10	Concerts	NFL	National League Football
1	Sports	NFL	National Football League
3	Sports	Mosicals	Musical theater
6	Shows		

To load data into Redshift, select a data source



Create S3 bucket – Uncheck “Block all public access” for the DEMO purpose only. Enable Bucket versioning as needed.



**Amazon S3**

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, please let us know.

**Buckets**

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3

Block Public Access settings for this account

**Storage Lens**

- Dashboards
- AWS Organizations settings

Feature spotlight (3)

AWS Marketplace for S3

**AWS Region**

US East (Ohio) us-east-2

Copy settings from existing bucket - optional  
Only the bucket settings in the following configuration are copied.

**Choose bucket**

**Block Public Access settings for this bucket**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

**Block all public access**  
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

**⚠ Turning off block all public access might result in this bucket and the objects within becoming public**  
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

**Bucket Versioning**

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

**Bucket Versioning**

Disable

Enable

For this lab, download files from this AWS [link](#) and upload to S3 bucket you created above.

The screenshot shows the AWS S3 'Upload' interface. The left sidebar includes links for Buckets, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, Access analyzer for S3, Block Public Access settings, Storage Lens, Dashboards, AWS Organizations settings, Feature spotlight (with 3 notifications), and AWS Marketplace for S3. The main area has a header with a feedback link and navigation to 'Amazon S3 > goclouds3bucket1 > Upload'. The 'Upload' section contains a large text input field with placeholder text 'Drag and drop files and folders you want to upload here, or choose Add files, or Add folders.' Below it is a table titled 'Files and folders (27 Total, 17.4 MB)' showing 27 entries, all of which are 'LoadingDataSampleFiles/' followed by various file names like 'customer-fw.manifest', 'customer-fw.tbl-000', etc., with sizes ranging from 1.5 MB to 482.0 B. There are 'Remove', 'Add files', and 'Add folder' buttons at the top of the table. The 'Destination' section shows 's3://goclouds3bucket1' as the target. The 'Permissions' and 'Properties' sections are collapsed. At the bottom right are 'Cancel' and 'Upload' buttons.

	Name	Folder	Type	Size
<input type="checkbox"/>	customer-fw.manifest	LoadingDataSampleFiles/	-	482.0 B
<input type="checkbox"/>	customer-fw.tbl-000	LoadingDataSampleFiles/	-	1.5 MB
<input type="checkbox"/>	customer-fw.tbl-000.bak	LoadingDataSampleFiles/	-	1.5 MB
<input type="checkbox"/>	customer-fw.tbl-001	LoadingDataSampleFiles/	-	1.5 MB
<input type="checkbox"/>	customer-fw.tbl-002	LoadingDataSampleFiles/	-	1.5 MB
<input type="checkbox"/>	customer-fw.tbl-003	LoadingDataSampleFiles/	-	1.5 MB
<input type="checkbox"/>	customer-fw.tbl-004	LoadingDataSampleFiles/	-	1.5 MB
<input type="checkbox"/>	customer-fw.tbl-005	LoadingDataSampleFiles/	-	1.5 MB
<input type="checkbox"/>	customer-fw.tbl-006	LoadingDataSampleFiles/	-	1.5 MB
<input type="checkbox"/>	customer-fw.tbl-007	LoadingDataSampleFiles/	-	1.5 MB

After files are uploaded, you should see upload complete status.

[Upload succeeded](#)  
[View details below](#)

## Upload: status

[Close](#)

This information below will no longer be available after you navigate away from this page.

Summary		
Destination s3://yourcloudbucket1	Status: <span style="color: green;">Succeeded</span> <span style="color: green;">22 files, 17.4 MB (100.00%)</span>	Failed: <span style="color: red;">0 files, 0 B (0%)</span>

[Play and Listen](#) | [Configuration](#)

**Files and folders** [22 Total, 17.4 MB]

Back up Database by creating a snapshot.

The screenshot shows the AWS Redshift console for a cluster named 'gocloud-redshift-cluster-1'. The 'Actions' menu is open, and the 'Create snapshot' option is highlighted with a red box. Other options visible in the menu include 'Delete', 'Modify maintenance', 'Modify publicly-accessible setting', 'Backup and disaster recovery', and 'Restore table'.

**Create snapshot**

Create a manual snapshot of **gocloud-redshift-cluster-1** cluster.

**Snapshot identifier**  
Identifier for the snapshot to be created.

**Manual snapshot retention period**  
Specify how long do you want to retain your snapshot.  
▼  
The retention period must be 1-3653 days.

**Cancel** **Create snapshot**

⌚ **goclouddredshift-backup was created successfully.**

Restore Snapshot from backup.

- a) Restore.

Amazon Redshift > Clusters > gocloud-redshift-cluster-1 > Restore table

## Restore table: gocloud-redshift-cluster-1

### Snapshots

#### From snapshot

The identifier of the cluster snapshot that contains the table to restore from.

b) Choose the backup from “View Snapshots” to retrieve existing snapshots.

c)

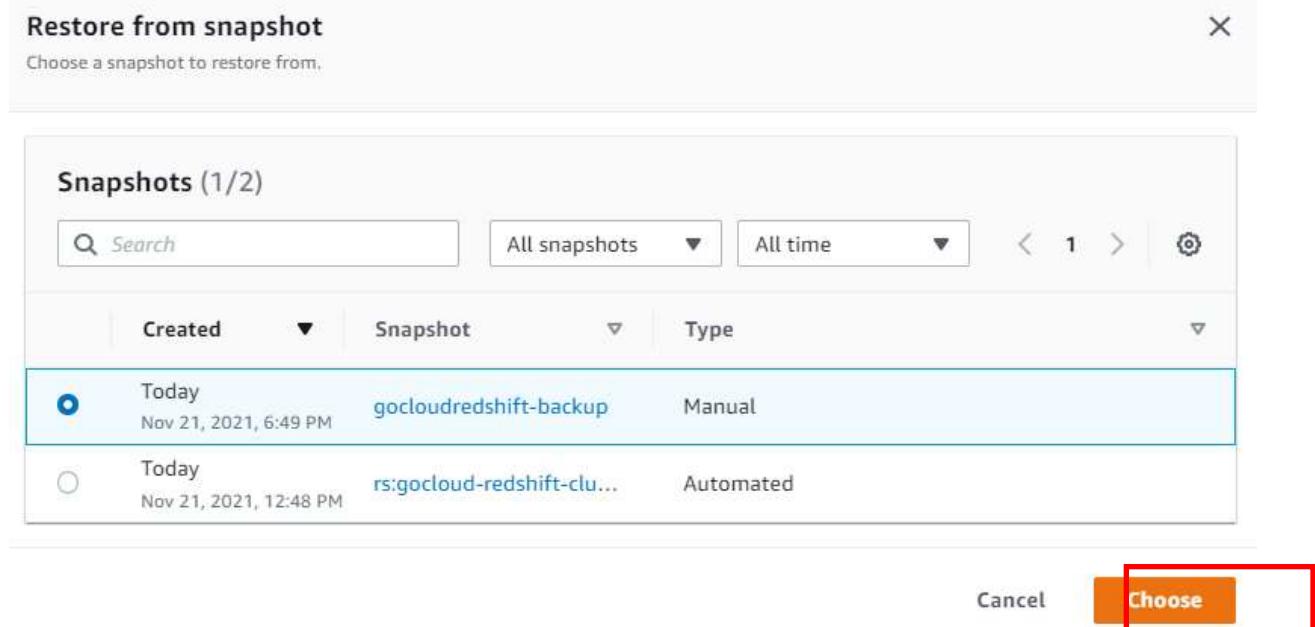
**Restore from snapshot**

Choose a snapshot to restore from.

**Snapshots (1/2)**

Created	Snapshot	Type
Today Nov 21, 2021, 6:49 PM	gocloudredshift-backup	Manual
Today Nov 21, 2021, 12:48 PM	rs:gocloud-redshift-clu...	Automated

Cancel **Choose**



d) Enter restore information.

**Restore**  
Choose the source table from the snapshot to restore to a target table in your cluster.

**Case sensitive names**  
Turn on to enter database, schema and table names as case-sensitive identifiers.

**Source table to restore from**

**Database**  
The name of the database from the cluster snapshot that contains the table to restore from.

**Schema**  
The name of the database schema from the cluster snapshot that contains the table to restore from.

**Table**  
The name of the table from the cluster snapshot to restore from.

**Target table to restore to**

**Database**  
The name of the database in the target cluster to restore the table to.

**Schema**  
The name of the database schema in the target cluster to restore the table to.

**New table name**  
The new name of the restored table. This name can't be the name of an existing table in the target database.

**Cancel** **Restore table**

**The table restore request was successfully created.**

**Amazon Redshift query editor v3 is now available**  
Query editor v3 provides new features such as multi-statement query execution, query parameterization, query versioning, visualizations, and query sharing. Learn more 

[Go to query editor v3](#) 

## Lab: Set Up Amazon ElastiCache with Amazon Aurora Serverless

Create Cloud9 IDE environment by searching “Cloud9” and selecting the service.

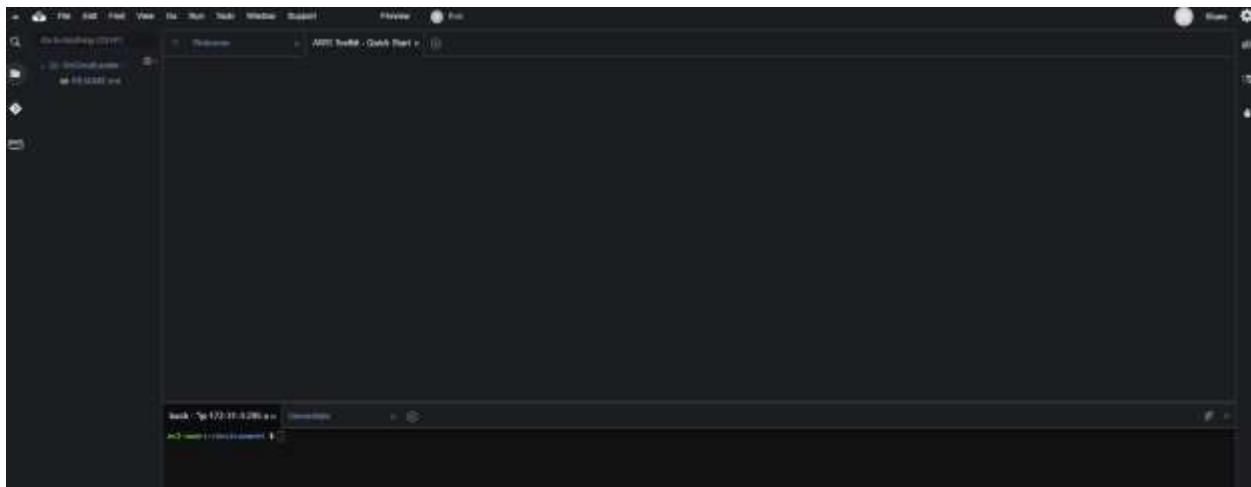
The screenshot shows the 'Configure settings' step of the AWS Cloud9 environment setup. The left sidebar lists 'Your environments', 'Shared with you', 'Account environments', and 'How-to guide'. The main area has three tabs: 'Step 1 Name environment' (disabled), 'Step 2 Configure settings' (selected), and 'Step 3 Review'. The 'Configure settings' tab contains the following sections:

- Environment settings**:
  - Environment type**: Create a new EC2 instance for environment (direct access)
  - Instance type**: t2.micro (1 GiB RAM + 1 vCPU)
  - Platform**: Amazon Linux 2 (recommended)
  - Cost-saving setting**: After 30 minutes (default)
  - IAM role**: AWS Cloud9 service-linked role

The following screen is displayed during creation.



Once connected, you should see a bash prompt connected to the EC2 instance.



Run the following commands:

- cd ~/environment
- curl -sL http://d118jxrmrxsq90.cloudfront.net/leaderboard.tar | tar -xv
- ls
- npm install --prefix scripts/ && npm install --prefix application
- echo "export AWS\_REGION=us-east-2" >> env.sh && source env.sh
- source env.sh

Type X.

**Choose a database creation method** Info

- Standard create**  
You set all of the configuration options, including ones for availability, security, backups, and maintenance.
- Easy create**  
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

**Engine options**

Engine type Info

- Amazon Aurora** 
- MySQL 
- MariaDB 
- PostgreSQL 
- Oracle 
- Microsoft SQL Server 

Edition

- Amazon Aurora MySQL-Compatible Edition
- Amazon Aurora PostgreSQL-Compatible Edition

Capacity type Info

- Provisioned**  
You provision and manage the server instance sizes.
- Serverless**  
You specify the minimum and maximum amount of resources needed, and Aurora scales the capacity based on database load. This is a good option for intermittent or unpredictable workloads.

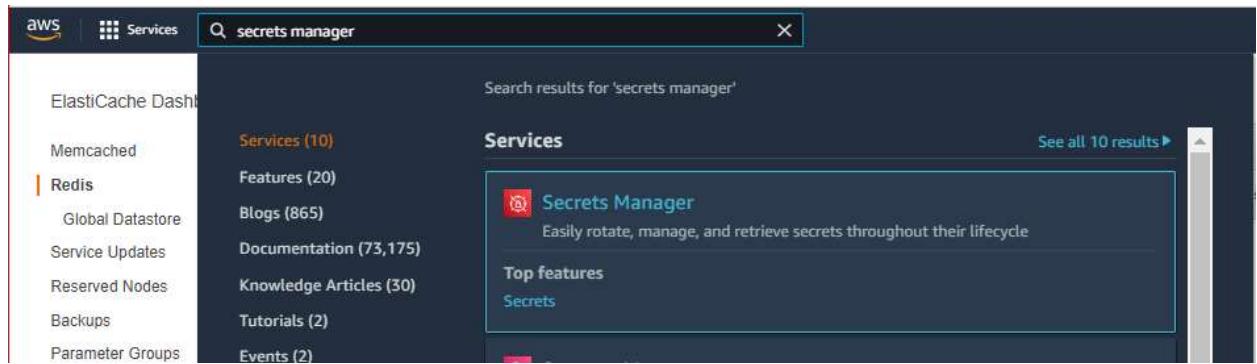
Available versions (2/55) Info

Aurora (MySQL)-5.6.10a ▼

(i) Aurora MySQL engine versions earlier than 2.09.2 don't support the newest r6g generation instance classes.

Choose default configurations but check “Data API” under Connectivity and uncheck “Deletion Protection” under Additional Configuration.

Secrets manager □ Store a new Secret.



Create and enter secret name as “gocloudsecret” □ Disable automatic rotation.

Edit env file.

```
ec2-user:~/environment $ echo "export DATABASE_ARN=arn:aws:rds:us-east-2:110680399916:cluster:gocloudleader" >> env.sh
ec2-user:~/environment $ echo "export SECRET_ARN=arn:aws:secretsmanager:us-east-2:110680399916:secret:gocloudsecret-LdeIky" >> env.sh
ec2-user:~/environment $ source env.sh
```

Edit testDatabase.js and update the database name.

```
ec2-user:~/environment $ cd scripts
ec2-user:~/environment/scripts $ cat testDatabase.js
// Copyright 2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0
const AWS = require('aws-sdk')

const rdsdataservice = new AWS.RDSDataService();

const params = {
  resourceArn: process.env.DATABASE_ARN,
  secretArn: process.env.SECRET_ARN,
  database: 'gocloudleader',
  sql: 'SELECT 1'
}

rdsdataservice.executeStatement(params, function(err, data) {
  if (err) {
    console.log(err, err.stack)
  } else {
    console.log(JSON.stringify(data, null, 2))
  }
})
ec2-user:~/environment/scripts $ node testDatabase.js
{
  "numberOfRecordsUpdated": 0,
  "records": [
    [
      {
        "longValue": 1
      }
    ]
  ]
}
```

```
ec2-user:~/environment/scripts $ node createTable.js
Table created successfully!
ec2-user:~/environment/scripts $ sed -i -- 's/leaderboard/
> ^C
ec2-user:~/environment/scripts $
ec2-user:~/environment/scripts $ sed -i -- 's/leaderboard/GoCloudLeader/g' *
sed: couldn't edit node_modules: not a regular file
ec2-user:~/environment/scripts $ ls -lrt
total 116
-rw-r--r-- 1 ec2-user ec2-user 321 Nov 11 2019 package.json
-rw-r--r-- 1 ec2-user ec2-user 471 Jun  2 2020 testRedis.js
-rw-r--r-- 1 ec2-user ec2-user 657 Jun  2 2020 remove-networking.sh
-rw-r--r-- 1 ec2-user ec2-user 11256 Nov 21 20:58 package-lock.json
drwxrwxr-x 22 ec2-user ec2-user 4096 Nov 21 20:58 node_modules
-rw-r--r-- 1 ec2-user ec2-user 508 Nov 21 21:24 testDatabase.js
-rw-r--r-- 1 ec2-user ec2-user 688 Nov 21 21:30 createTable.js
-rw-r--r-- 1 ec2-user ec2-user 1919 Nov 21 21:30 create-rest-api.sh
-rw-r--r-- 1 ec2-user ec2-user 1981 Nov 21 21:30 create-networking.sh
-rw-r--r-- 1 ec2-user ec2-user 3173 Nov 21 21:30 create-lambda.sh
-rw-r--r-- 1 ec2-user ec2-user 1652 Nov 21 21:30 fetchHighScoresForUser2.js
-rw-r--r-- 1 ec2-user ec2-user 489 Nov 21 21:30 dropTables.js
-rw-r--r-- 1 ec2-user ec2-user 787 Nov 21 21:30 delete-resources.sh
-rw-r--r-- 1 ec2-user ec2-user 579 Nov 21 21:30 create-user-pool.sh
-rw-r--r-- 1 ec2-user ec2-user 483 Nov 21 21:30 create-user-pool-client.sh
-rw-r--r-- 1 ec2-user ec2-user 976 Nov 21 21:30 getTopOverallScores.js
-rw-r--r-- 1 ec2-user ec2-user 27760 Nov 21 21:30 games.json
-rw-r--r-- 1 ec2-user ec2-user 478 Nov 21 21:30 flushRedis.js
-rw-r--r-- 1 ec2-user ec2-user 930 Nov 21 21:30 fetchHighScoresForUser.js
-rw-r--r-- 1 ec2-user ec2-user 1035 Nov 21 21:30 loadRedis.js
-rw-r--r-- 1 ec2-user ec2-user 868 Nov 21 21:30 insertGames.js
```

Fetch High scores.

```
ec2-user:~/environment/scripts $ node insertGames.js
Games inserted successfully!
ec2-user:~/environment/scripts $
ec2-user:~/environment/scripts $ node fetchHighScoresForUser.js
{
  "columnMetadata": [
    {
      "arrayBaseColumnType": 0,
      "isAutoIncrement": true,
      "isCaseSensitive": false,
      "isCurrency": false,
      "isSigned": true,
      "label": "game_id",
      "name": "game_id",
      "nullable": 0,
      "precision": 10,
      "scale": 0,
      "schemaName": "",
      "tableName": "games",
      "type": 4,
      "typeName": "INT"
    },
    {
      "arrayBaseColumnType": 0,
      "isAutoIncrement": true,
      "isCaseSensitive": false,
      "isCurrency": false,
      "isSigned": true,
      "label": "score",
      "name": "score",
      "nullable": 0,
      "precision": 10,
      "scale": 0,
      "schemaName": "public",
      "tableName": "high_scores",
      "type": 4,
      "typeName": "INT"
    }
  ]
}
```

Create ElastiCache – Redis from services with t2.micro and new subnet group.

## Redis settings

Ensure you have reviewed the five workload characteristics to consider when right sizing Amazon ElastiCache Redis clusters. [Learn more](#)

Name	GoCloudLeader	<a href="#">i</a>
Engine version compatibility	6.x	<a href="#">i</a>
Port	6379	<a href="#">i</a>
Parameter group	default.redis6.x	<a href="#">i</a>
Node type	cache.t2.micro (0.5 GiB)	<a href="#">i</a>
Number of replicas	0	<a href="#">i</a>
Multi-AZ	<input type="checkbox"/>	<a href="#">i</a>

### [i](#) Multi-AZ

Multi-AZ can not be enabled when the number of replicas is set to 0. Select one or more replicas to enable Multi-AZ. [Learn more](#)

## ▼ Advanced Redis settings

Advanced settings have common defaults set to give you the fastest way to get started. You can modify these now or after your cluster has been created.

Subnet group	Create new	<a href="#">i</a>																												
<table><tr><td>Name</td><td>GoCloudLeader-redis</td><td><a href="#">i</a></td></tr><tr><td>Description</td><td>GoCloudLeader-redis</td><td><a href="#">i</a></td></tr><tr><td>VPC ID</td><td>vpc-11686478</td><td><a href="#">i</a></td></tr><tr><td>Subnets</td><td colspan="2"><table><thead><tr><th></th><th>Subnet ID</th><th>Availability zone</th><th>CIDR Block</th></tr></thead><tbody><tr><td><input checked="" type="checkbox"/></td><td>subnet-eb160b82</td><td>us-east-2a</td><td>172.31.0.0/20</td></tr><tr><td><input checked="" type="checkbox"/></td><td>subnet-18227063</td><td>us-east-2b</td><td>172.31.16.0/20</td></tr><tr><td><input type="checkbox"/></td><td>subnet-fabc23b7</td><td>us-east-2c</td><td>172.31.32.0/20</td></tr></tbody></table></td></tr></table>			Name	GoCloudLeader-redis	<a href="#">i</a>	Description	GoCloudLeader-redis	<a href="#">i</a>	VPC ID	vpc-11686478	<a href="#">i</a>	Subnets	<table><thead><tr><th></th><th>Subnet ID</th><th>Availability zone</th><th>CIDR Block</th></tr></thead><tbody><tr><td><input checked="" type="checkbox"/></td><td>subnet-eb160b82</td><td>us-east-2a</td><td>172.31.0.0/20</td></tr><tr><td><input checked="" type="checkbox"/></td><td>subnet-18227063</td><td>us-east-2b</td><td>172.31.16.0/20</td></tr><tr><td><input type="checkbox"/></td><td>subnet-fabc23b7</td><td>us-east-2c</td><td>172.31.32.0/20</td></tr></tbody></table>			Subnet ID	Availability zone	CIDR Block	<input checked="" type="checkbox"/>	subnet-eb160b82	us-east-2a	172.31.0.0/20	<input checked="" type="checkbox"/>	subnet-18227063	us-east-2b	172.31.16.0/20	<input type="checkbox"/>	subnet-fabc23b7	us-east-2c	172.31.32.0/20
Name	GoCloudLeader-redis	<a href="#">i</a>																												
Description	GoCloudLeader-redis	<a href="#">i</a>																												
VPC ID	vpc-11686478	<a href="#">i</a>																												
Subnets	<table><thead><tr><th></th><th>Subnet ID</th><th>Availability zone</th><th>CIDR Block</th></tr></thead><tbody><tr><td><input checked="" type="checkbox"/></td><td>subnet-eb160b82</td><td>us-east-2a</td><td>172.31.0.0/20</td></tr><tr><td><input checked="" type="checkbox"/></td><td>subnet-18227063</td><td>us-east-2b</td><td>172.31.16.0/20</td></tr><tr><td><input type="checkbox"/></td><td>subnet-fabc23b7</td><td>us-east-2c</td><td>172.31.32.0/20</td></tr></tbody></table>			Subnet ID	Availability zone	CIDR Block	<input checked="" type="checkbox"/>	subnet-eb160b82	us-east-2a	172.31.0.0/20	<input checked="" type="checkbox"/>	subnet-18227063	us-east-2b	172.31.16.0/20	<input type="checkbox"/>	subnet-fabc23b7	us-east-2c	172.31.32.0/20												
	Subnet ID	Availability zone	CIDR Block																											
<input checked="" type="checkbox"/>	subnet-eb160b82	us-east-2a	172.31.0.0/20																											
<input checked="" type="checkbox"/>	subnet-18227063	us-east-2b	172.31.16.0/20																											
<input type="checkbox"/>	subnet-fabc23b7	us-east-2c	172.31.32.0/20																											
Availability zones placement	<input checked="" type="radio"/> No preference <input type="radio"/> Select zones	<a href="#">i</a>																												

Create Lambda security group.

## Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

### Basic details

Security group name Info

Name cannot be edited after creation.

Description Info

VPC Info

X

### sg-032f0bfd64b06e0a2 - GoCloudLeader-lambda

Actions ▼

#### Details

Security group name

 GoCloudLeader-lambda

Security group ID

 sg-032f0bfd64b06e0a2

Description

 GoCloudLeader-lambda

VPC ID

 vpc-11686478

Owner

 110680329916

Inbound rules source

 Permit specific

Outbound rules target

 Permit specific

```
ec2-user:~/environment/scripts $ echo "export SECURITY_GROUP_ID=sg-032f0bfd64b06e0a2" >> env.sh && source env.sh
```

Amazon ElastiCache instances must be provisioned in a VPC in a private subnet.

- Edit Inbound rules in Security Group. Use the security group from ElastiCache dashboard by expanding the Redis cache we created.

Security Groups (1/1) [View](#)

Actions [Export security groups to CSV](#) [Create security group](#)

Filter security groups  [Clear filters](#)

Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count	Outbound rules count
sg-0356ce3d	default	sg-0356ce3d	vpc-1168047E	default VPC security group	AWS	1 Permission entry	1 Permission entry

Details [Inbound rules](#) [Outbound rules](#) [Tags](#)

You can now check network connectivity with Reachability analyzer [Run Reachability Analyzer](#)

Inbound rules (1/1)

Filter security group name

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
sg-034272be03d30793	-	-	All traffic	All	All	sg-0356ce3d / default	-

Add Inbound rules for the traffic from Cloud9 instance.

Edit inbound rules [View](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules [View](#)

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sg-034272be03d30793	Custom TCP	TCP	6379	Custom	Cloud9 Puzzle Lambda
-	Custom TCP	TCP	6379	Custom	LambdaFunction4000Lambda

[Add rule](#)

Cancel [Preview changes](#) [Save rules](#)

Ping Redis instance.

```
e2-user:~/environment/scripts $ echo "export REDIS_ENDPOINT=gocloudleader.2gmpoyw.0001.use2.cache.amazonaws.com:6379" >> env.sh && source env.sh
e2-user:~/environment/scripts $ node testRedis.js
Successful ping! PONG
```

Load sample data into Redis and read top scores.

```
ec2-user:~/environment/scripts $ node loadRedis.js
Loaded data!
ec2-user:~/environment/scripts $ node getTopOverallScores.js
Top overall scores:
[
  {
    username: 'debbieschneider',
    gamedate: '2019-11-09T18:41:27',
    level: '28',
    score: '9895'
  },
  {
    username: 'alicia39',
    gamedate: '2019-11-09T10:39:59',
    level: '47',
    score: '9824'
  },
  {
    username: 'rosecolleen',
    gamedate: '2019-11-10T07:09:51',
    level: '58',
    score: '9765'
  },
  {
    username: 'allisonsandra',
    gamedate: '2019-11-07T22:43:32',
    level: '62',
    score: '9760'
  },
  {
    username: 'kathrynmorris',
    gamedate: '2019-11-05T04:31:37',
    level: '85',
    score: '9722'
  }
]
```

Amazon Cognito user pool.

```
ec2-user:~/environment/scripts $ cat create-user-pool.sh
# Copyright 2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
source env.sh
USER_POOL_ID=$(aws cognito-idp create-user-pool \
    --pool-name GoCloudLeader-users \
    --policies ' \
        {
            "PasswordPolicy": {
                "MinimumLength": 8,
                "RequireUppercase": true,
                "RequireLowercase": true,
                "RequireNumbers": true,
                "RequireSymbols": false
            }
        }' \
    --query 'UserPool.Id' \
    --output text)

echo "User Pool created with id ${USER_POOL_ID}"
echo "export USER_POOL_ID=${USER_POOL_ID}" >> env.sh
ec2-user:~/environment/scripts $
ec2-user:~/environment/scripts $ bash ./create-user-pool.sh
User Pool created with id us-east-2_x1GtfDqX0
ec2-user:~/environment/scripts $ bash ./create-user-pool-client.sh
User Pool Client created with id 2l6inhvbre7pj78q8jd9qr1qoa
```

Deploy your application.

Create network resources.

```
ec2-user:~/environment/scripts $ bash ./create-networking.sh
Fetching VPC Id
Fetching subnet Id
Creating Elastic IP address
Creating NAT Gateway
Waiting for NAT Gateway to be ready...
Creating private subnet
Creating route table
Creating route
Associating route table with subnet
Networking resources created!
```

Create Lambda function.

```
bash scripts/create-lambda.sh

Building zip file
Creating IAM role
Adding policy to IAM role
Sleeping for IAM role propagation
Creating Lambda function
Lambda function created with ARN arn:aws:elasticache:us-east-2:110680399916:lambda:gocloudleader
```

Create REST API.

```
bash scripts/create-rest-api.sh

Creating REST API
Fetching root resource
Creating proxy resource
Creating method
Adding integration
Creating deployment
Fetching account ID
Adding lambda permission
REST API created

Your API is available at: https://110680399916.execute-api.us-east-2.amazonaws.com/prod

source env.sh
```

Testing the Application

Register new user.

```
ec2-user:~/environment $ curl -X POST ${BASE_URL}/users \
>   -H 'Content-Type: application/json' \
>   -d '{
> "username": "puzzlemaster",
> "password": "Mypassword1",
> "email": "test@hello.com"
> }'
{"username":"puzzlemaster"}
```

Log in to fetch user credentials.

```
ec2-user:~/environment $ curl -X POST ${BASE_URL}/login \
>   -H 'Content-Type: application/json' \
>   -d '{
> "username": "puzzlemaster",
> "password": "Mypassword1"
> }'
{"idToken":"eyJraWQiO..."}
export ID_TOKEN=eyJraWQiO...
```

Create REST API and add data.

```
bash scripts/create-rest-api.sh

Creating REST API
Fetching root resource
Creating proxy resource
Creating method
Adding integration
Creating deployment
Fetching account ID
Adding lambda permission
REST API created

Your API is available at: https://110680399916.execute-api.us-east-2.amazonaws.com/prod

source env.sh

curl -X POST ${BASE_URL}/users/puzzlemaster \
-H 'Content-Type: application/json' \
-H "Authorization: ${ID_TOKEN}" \
-d '{
    "level": 37,
    "score": 6541
}'

{"username":"puzzlemaster","gametime":"2019-11-12T03:18:51.637Z","level":37,"score":6541}

curl -X POST ${BASE_URL}/users/puzzlemaster \
-H 'Content-Type: application/json' \
-d '{
    "level": "37",
    "score": "6541"
}'

{"message":"jwt must be provided"}
```

Add more data and get the scores.

```
curl -X POST ${BASE_URL}/users/puzzlemaster \
-H 'Content-Type: application/json' \
-H "Authorization: ${ID_TOKEN}" \
-d '{
    "level": "42",
    "score": "7142"
}'

curl -X POST ${BASE_URL}/users/puzzlemaster \
-H 'Content-Type: application/json' \
-H "Authorization: ${ID_TOKEN}" \
-d '{
    "level": "48",
    "score": "9901"
}'

curl -X GET ${BASE_URL}/users/puzzlemaster
```

## Labs: Amazon Simple Queue Service (SQS)

Choose SQS service.

The screenshot shows the AWS Services search interface. The search bar at the top contains 'sq'. Below it, a sidebar for 'Amazon RDS' lists 'Dashboard', 'Databases' (which is highlighted in orange), 'Query Editor', 'Performance insights', and 'Snapshots'. The main search results area is titled 'Search results for "sq"' and shows 'Services (1)'. It lists 'Blogs (98)', 'Documentation (58,872)', 'Knowledge Articles (30)', and 'Tutorials (3)'. A single result, 'Simple Queue Service' with the sub-section 'SQS Managed Message Queues', is displayed in a box with a red border.

Create queue. Once created, the following screen should appear:

The screenshot shows the 'Scores' queue details page. The top navigation bar includes 'Amazon SQS > Queues > Scores'. The main section is titled 'Scores' and contains a 'Details' tab. Key details shown include: Name (Scores), Type (Standard), URL (<https://amazon-sqs-1106800916.us-east-2.amazonaws.com>), Encryption (None), Visibility timeout (100 seconds), Create time (11/21/2021, 16:35:07 EST), Message retention period (4 Days), and Delivery delay (0 seconds). Buttons for 'Edit', 'Delete', 'Purge', and 'Send and receive messages' are visible at the top right.

Send messages to the queue using “Send and receive messages” button.

The screenshot shows the 'Send and receive messages' page for the 'Scores' queue. The top navigation bar includes 'Amazon SQS > Queues > Scores > Send and receive messages'. The main section is titled 'Send and receive messages' and has a sub-section 'Send message'. It includes fields for 'Message body' (containing '1 x Widget @ \$28.00 USD'), 'Delivery delay' (set to '0'), and 'Priority' (set to 'Normal'). A red box highlights the 'Send message' button at the bottom right.

## Receive messages by polling.

Receive messages 1%

Messages available: 1 Polling duration: 30 Maximum message count: 10 Polling progress: 1 message received 0%

Messages (1)

ID	Sent	Size	Received count
8839e81d-fd5a-4b50-b85b-ed6b46304a98	11/21/2021, 18:38:38 EST	49 bytes	1

**Message: 8839e81d-fd5a-4b50-b85b-ed6b46304a98** X

**Details** **Body** **Attributes**

ID 8839e81d-fd5a-4b50-b85b-ed6b46304a98	Size 49 bytes	MD5 of message body 84dd5d08cee13d57dcc10511d271f68f	Sender account ID 110680399916
Sent 11/21/2021, 18:38:38 EST	First received 11/21/2021, 18:40:32 EST	Receive count 1	Message attributes count -
Message attributes size -	MD5 of message attributes -		

**Done**

**Message: 8839e81d-fd5a-4b50-b85b-ed6b46304a98** X

**Details** **Body** **Attributes**

1 x Widget @ \$29.99 USD  
2 x Widget Cables @ \$4.99

**Done**

Delete messages from queue.

A screenshot of a web-based message list interface. At the top, there's a search bar labeled "Search messages". Below it is a table with columns: "ID", "Type", "Subject", "Send", "Size", "Receive count", and "Last seen". A single row is selected, highlighted with a blue border. The selected message has the ID "8839e81d-fd5a-4b50-b85b-ed6b46304a98", was sent at "11/27/2011, 10:38:08 EST", is 49 bytes in size, and has a receive count of 1. At the top right of the interface are "View details" and "Delete" buttons, along with some icons.

## Delete Messages



Are you sure you want to delete the following message? You can't undo this action.

- 8839e81d-fd5a-4b50-b85b-ed6b46304a98 (49 bytes)

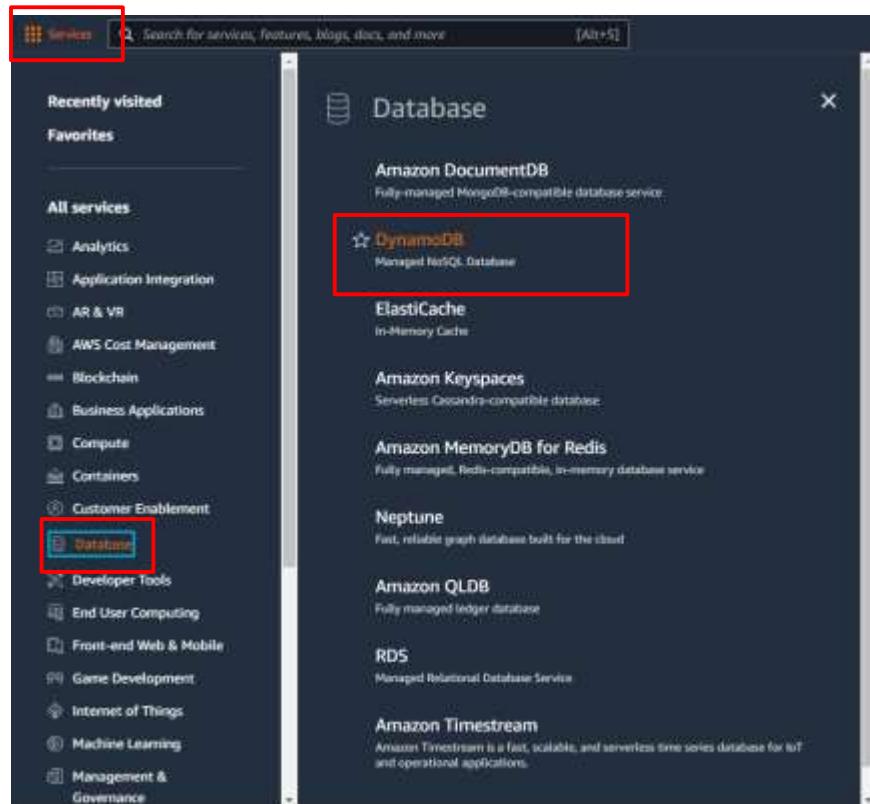
[Cancel](#)

[Delete](#)

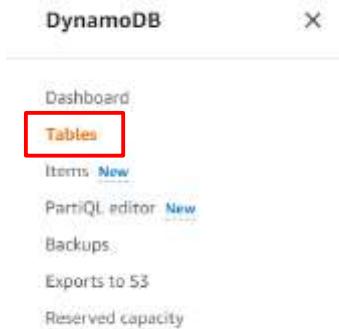
## Lab: Create a Table in DynamoDB Using the AWS Management Console

The objective of this lab is to create a table in DynamoDB using the AWS Management Console.

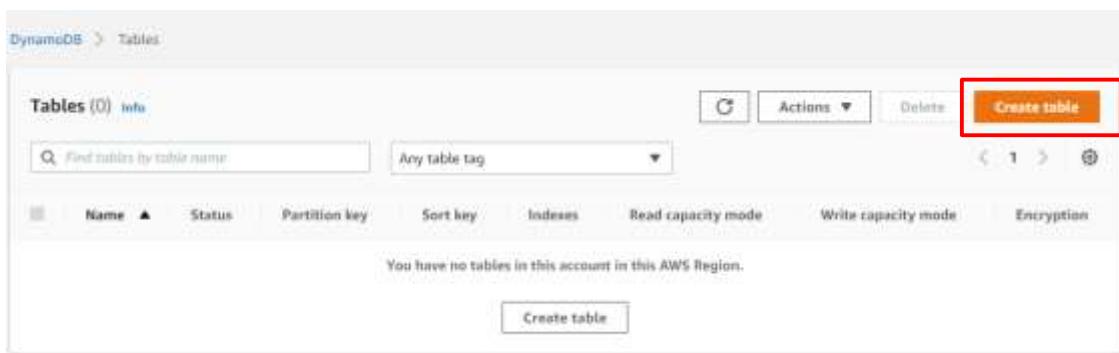
Start by logging in to the AWS Management Console at <<https://aws.amazon.com>>. Click on **Services** (top left), **Database** (left column), and **DynamoDB** (right column):



Click **Tables**.



Click the **Create Table** button.



Enter the table details as follows:

For the **Table name**, type “Cars”.

For the **Partition key**, type “Brand”.

Type “Model” as the **Sort key – optional**.

Leave **Default settings** selected.

## Create table

### Table details Info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

#### Table name

This will be used to identify your table.

Between 3 and 255 characters, containing only letters, numbers, underscores (\_), hyphens (-), and periods (.)

#### Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.



1 to 255 characters and case sensitive.

#### Sort key - optional

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.



1 to 255 characters and case sensitive.

### Settings

#### Default settings

The fastest way to create your table. You can modify these settings now or after your table has been created.

#### Customize settings

Use these advanced features to make DynamoDB work better for your needs.

Scroll to bottom, Click **Create table** button.

## Lab: Using the Database Migration Service and the Schema Conversion Tool

In this lab will use the DMS and SCT to migrate a database to AWS.

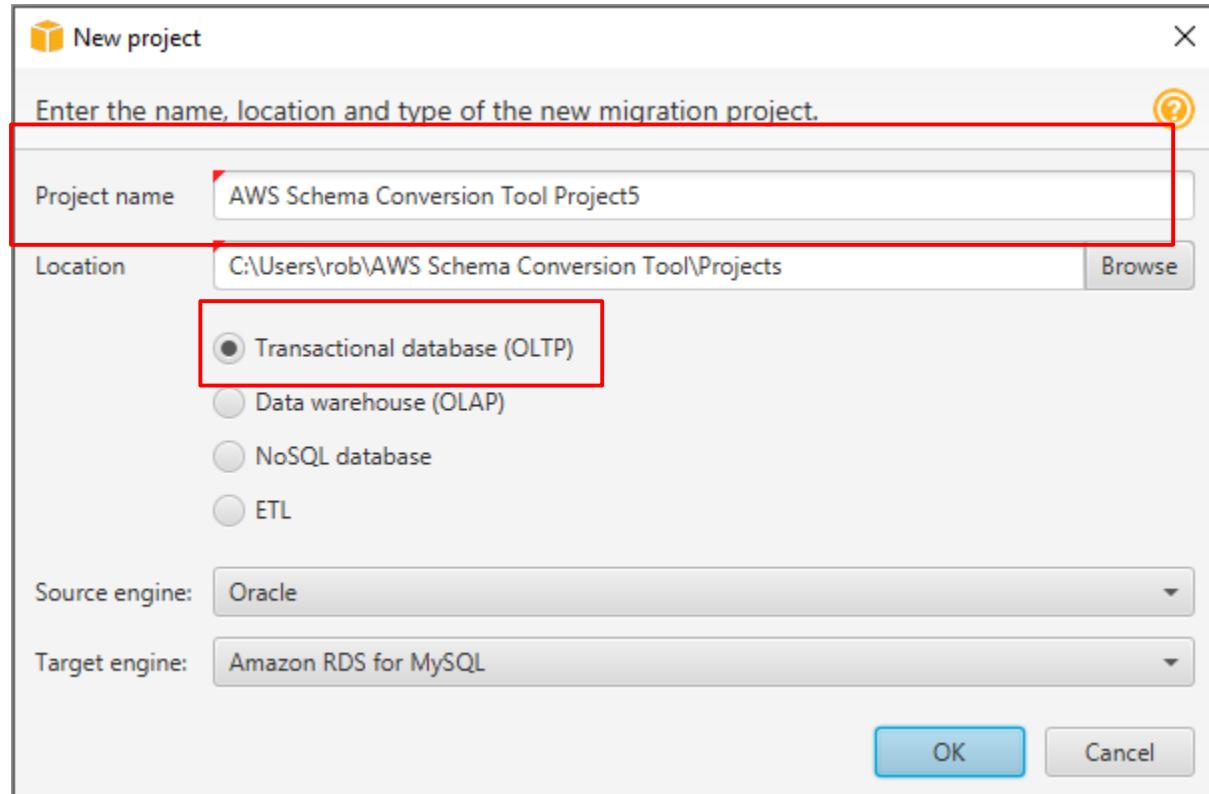
The following high-level steps can be used for a heterogeneous migration of an Oracle Database (OLTP) to an AWS RDS Postgres database using a combination of native tools like Oracle Data Pump and AWS SCT and DMS.

Record the SCN of the Oracle Database, take a backup. Next create a user on the source (ORACLE) database and grant them the necessary privileges per AWS Guidelines.

Now download and start the SCT client.

Enter a project Name.

Choose OLTP.



Enter source and target engines as Oracle and Postgres.

In the connect to dialog box, connect Oracle and fill in the necessary details. The user should be the same user created in step II and Test connection.

**Connect to Amazon RDS for PostgreSQL.** Repeat the above steps above.

**Connect to Oracle**

Specify parameters for new connections to the source

Type: Service Name

Server name: 10.0.0.30

Server port: 1521

Service name: pdb1.com

User name: scott

Password: \*\*\*\*\*

Use SSL

Store password

**Test connection** **OK** **Cancel**

**Connect to Amazon RDS for PostgreSQL**

Specify parameters for new connections to the target

Connection **SSL**

Server name: postgresql.cnbx0q035jku.us-east-1.rds.amazonaws.com

Server port: 5432

Database: postgresql

User name: postgresql

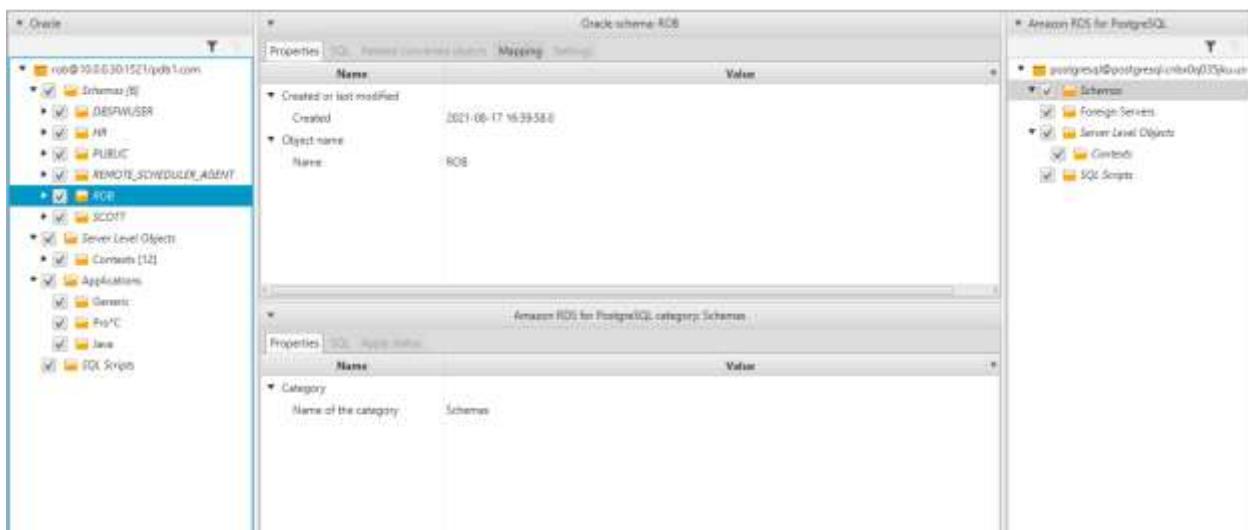
Password: \*\*\*\*\*

Use SSL

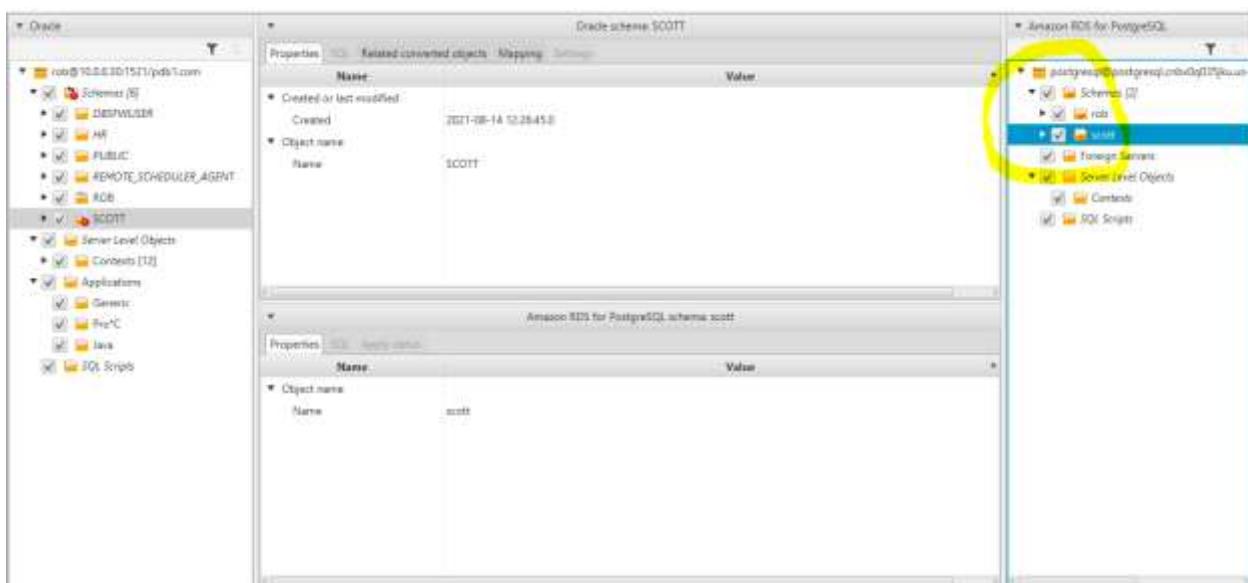
Store password

**Test connection** **OK** **Cancel**

Click to convert schema.



Right click on the target and apply.



Create a replication instance using the AWS DMS console.

## Create replication instance

### Replication instance configuration

#### Name

The name must be unique among all of your replication instances in the current AWS region.

Replication instance name must not start with a numeric value

#### Descriptive Amazon Resource Name (ARN) - *optional*

A friendly name to override the default DMS ARN. You cannot modify it after creation.

#### Description

The description must only have unicode letters, digits, whitespace, or one of these symbols: \_.:;/=-@. 1000 maximum character.

#### Instance class [Info](#)

Choose an appropriate instance class for your replication needs. Each instance class provides differing levels of

Specify source and target endpoints.

## Create endpoint

### Endpoint type [Info](#)

#### Source endpoint

A source endpoint allows AWS DMS to read data from a database (on-premises or in the cloud); or from other data source such as Amazon S3.

#### Target endpoint

A target endpoint allows AWS DMS to write data to a database, or to other data source.

Select RDS DB instance

### Endpoint configuration

---

Create a task and migrate data.

## Create database migration task

### Task configuration

Task identifier

Replication instance

Source database endpoint

Target database endpoint

To test replication:

Make sure that your database migration task shows a status of running but your initial database replication, started in the previous step, isn't complete.

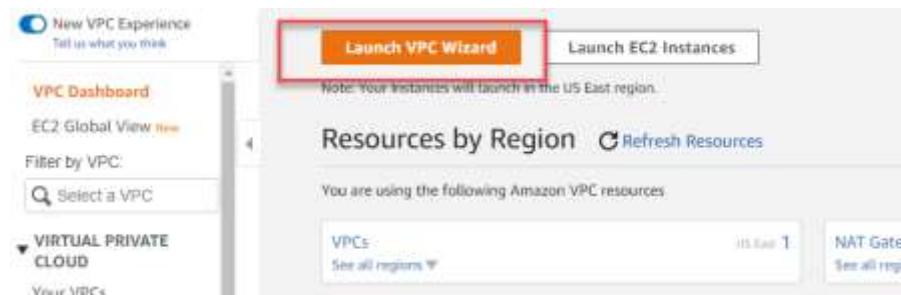
## Chapter 6 Labs

### Lab: Create a VPC

Open the VPC console at <https://console.aws.amazon.com/vpc/>.

The VPC will be created in a specific AWS Region. Take note of the region, found in the top right of the navigation bar. Ensure this region is selected for the entirety of this exercise.

**Launch the VPC Wizard**, from the **VPC Dashboard** located in the navigation pane.



Next, follow this path: **VPC with a Single Public Subnet > Select**



**Configure the VPC.** Type a name for the VPC in the **VPC Name** field; also enter a name for the subnet in the field titled **Subnet Name**.

This will aid in identifying each in the Amazon VPC console. Leave all other configurations as is for the purposes of this exercise. Then choose **Create VPC**.

A status window will show the creation progress. Once completed, close the status window by clicking **OK**.

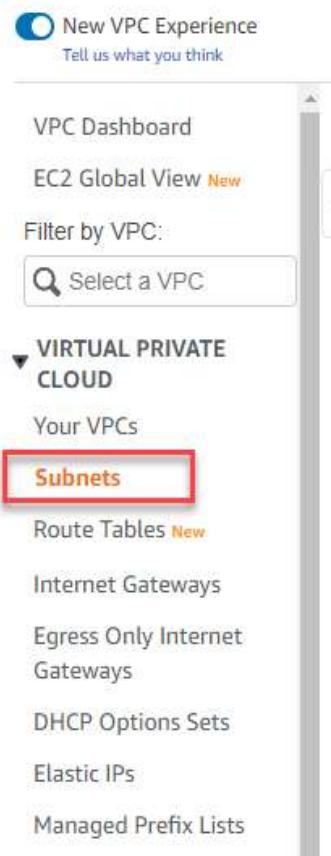
The newly created VPC will be found on the **Your VPCs** page along with a default VPC. The **Default VPC** column will display **No** for the new VPC as it is considered nondefault.

Your VPCs (2) <small>Info</small>						
<input type="text"/> Filter VPCs						
Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR		
-	vpc-04a2896e086d57a55	Available	172.31.0.0/16	-		
MYVPC	vpc-09ff4ebf24aa88253	Available	10.0.0.0/16	-		

## Create a Subnet for the VPC

An IPv4 CIDR block must first be selected from the newly created VPC's range and Availability Zone specified. There may be multiple subnets within the same zone. If an IPv6 CIDR block is associated with your VPC, an IPv6 block may be specified for the subnet.

Navigate to the Amazon VPC console: <https://console.aws.amazon.com/vpc/>. Choose **Subnets > Create subnet** from the navigation pane.



Enter subnet specifications as necessary. Choose **Create** and then select the VPC for which you created the subnet.

Create subnet (4%)

## VPC

## VPC ID

Create a subnet in this VPC:

## Associated VPC CIDR:

(IPv4 CIDR):

10.0.0.0/16

## Subnet settings

Specify the CIDR blocks and Availability Zones for the subnet.

## Subnet 1 of 1

## Subnet name

Create a unique name for your subnet and select the zone(s) it uses:

The name can be up to 256 characters long.

Availability Zone: **None**

Choose the zone or zones in your subnet will enable or les zones ottentes pour vous.

## IPv4 CIDR block (1):

## ▼ Tags - optional:

Key	Value - optional
Q. Name	X
Q. MySubnet01	X
<input type="button" value="Remove"/>	

You last add 40 more tags.

Cancel

## Lab: Create Network Access Control Lists (NACL)

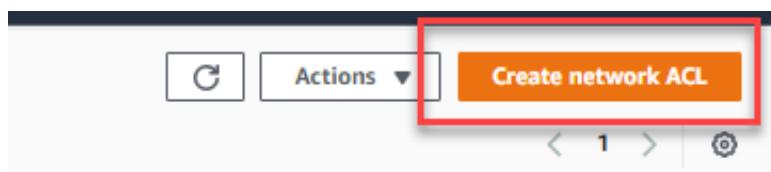
The objective of this lab is to create an access list to protect a subnet.

Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.

Choose **Network ACLs** in the navigation pane.



Choose **Create network ACL**.



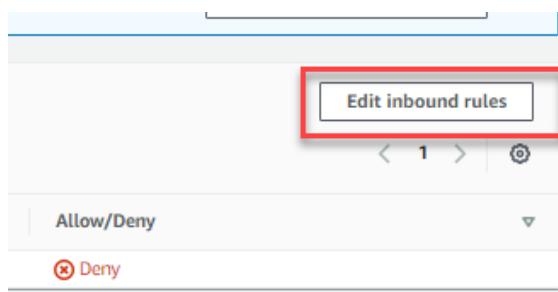
Name your network ACL, if desired, in the **Create Network ACL** dialog box. Then select the VPC ID from the **VPC** list and choose **Yes > Create**.

## Add and Delete Rules

Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.

Choose **Network ACLs**.

**Edit inbound rules** in the details pane.



Choose **Add new rule**.

In this example, we will block any incoming traffic from the CIDR range 192.168.0.0/16 from entering the subnet:

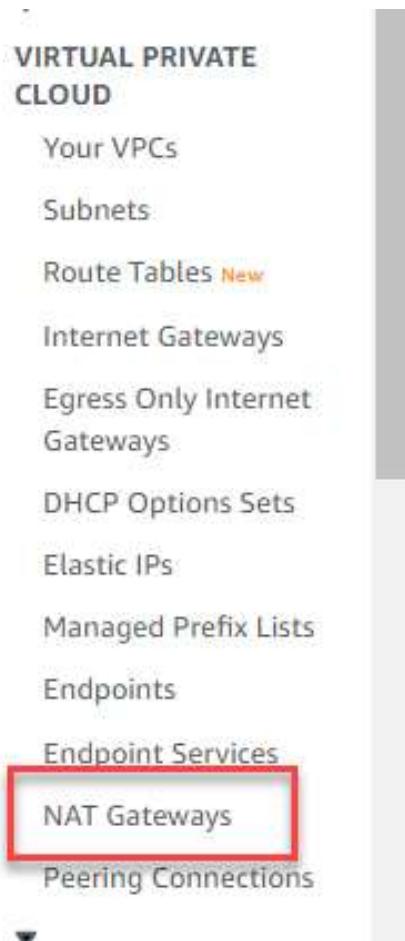


Now, we will block any outgoing traffic destined to CIDR range 192.168.0.0/16:



## Lab: Create A NAT Gateway

Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.



Choose **NAT Gateways** in the navigation pane, and then do the following:

**Create NAT gateway** Info

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

**NAT gateway settings**

Name - *optional*  
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Subnet  
Select a subnet in which to create the NAT gateway.

Connectivity type  
Select a connectivity type for the NAT gateway.

Public  
 Private

Elastic IP allocation ID Info  
Assign an Elastic IP address to the NAT gateway.

**Tags**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - <i>optional</i>
<input type="text" value="Name"/> <input type="button" value="X"/>	<input type="text" value="MyNATGateway"/> <input type="button" value="X"/> <input type="button" value="Remove"/>

Specify a name for the NAT gateway. This is optional. **Note:** Key is “Name” and Value is the name specified.

Select a target subnet in which to place the NAT gateway.

**Connectivity type:** AWS defaults to a **Public** gateway. Select **Private** to create a private NAT gateway if preferred.

\*Note For Public NAT gateway only: For **Elastic IP allocation ID**, select an Elastic IP address to associate with the NAT gateway.

(Optional) Choose **Add new tag** and enter the key name and value if additional tags are desired.

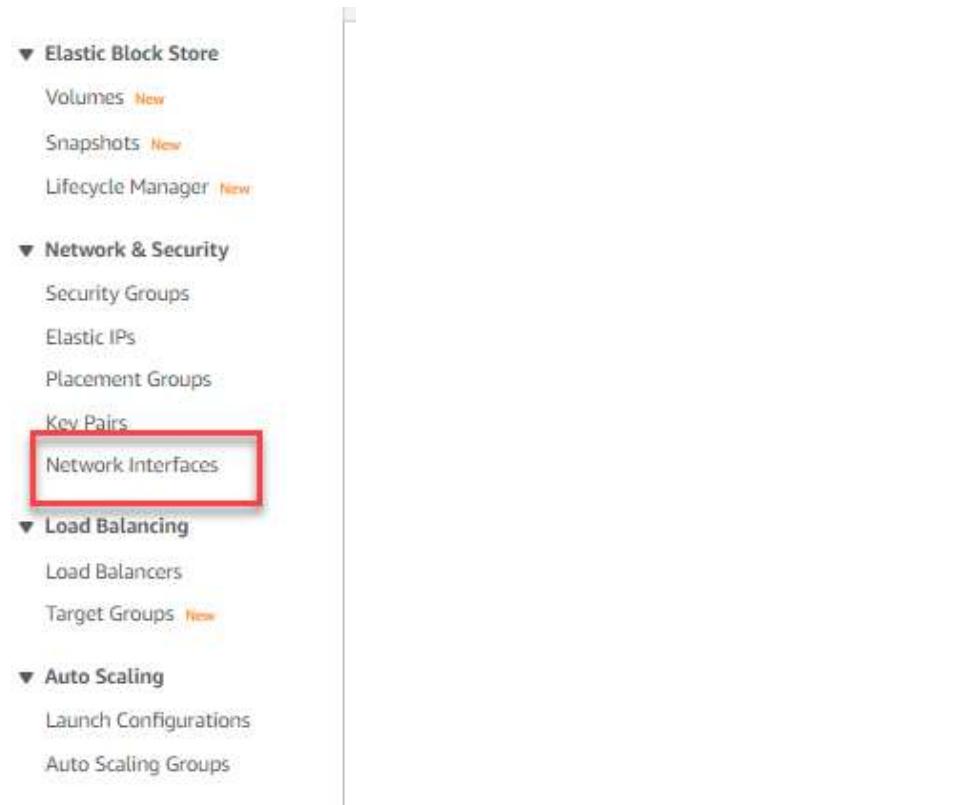
Choose **Create a NAT Gateway**.

## Lab: Create Elastic Network Interface

In this lab we will create an Elastic Network Interface.

Open the Amazon EC2 Console at <https://console.aws.amazon.com/ec2/>.

Choose **Network Interfaces** from the navigation pane.



Choose **Create network interface**.

Enter a **Description** name (optional).

Select the **Subnet** that was created.

For Private IPv4 address:

Choose **Auto-assign**, or

Choose **Custom** and enter an IPv4 address selected from the subnet.

Select one or more **Security Groups**, and if desired, **Add New Tag**.

Then **Create network interface**.

EC2 > Network interfaces > Create network interface

### Create network interface

An elastic network interface is a logical networking component in a VPC that represents a virtual network card.

**Details** Info

Description - *optional*  
A descriptive name for the network interface.  
  

Subnet  
This subnet must be in the same VPC as the instance.  
   X ↻

Private IPv4 address  
The private IPv4 address to assign to the network interface.  
 Auto-assign  
 Custom

Elastic Fabric Adapter  
 Enable

► Advanced settings

**Security groups (0/2)** Info

<input type="checkbox"/>	Group ID	▲	Group name	▼	Description
<input type="checkbox"/>	sg-00edb44be48ccfe8		default		default VPC security group
<input type="checkbox"/>	sg-0a322ac290ba43608		launch-wizard-2		launch-wizard-2 created 2021-11-2...

**Tags - optional**

## Attach Elastic Network Interface (ENI)

An interface can be attached to any instance in the same Availability Zone. To do so, do the following:

Open the Amazon EC2 console, <https://console.aws.amazon.com/ec2/>.

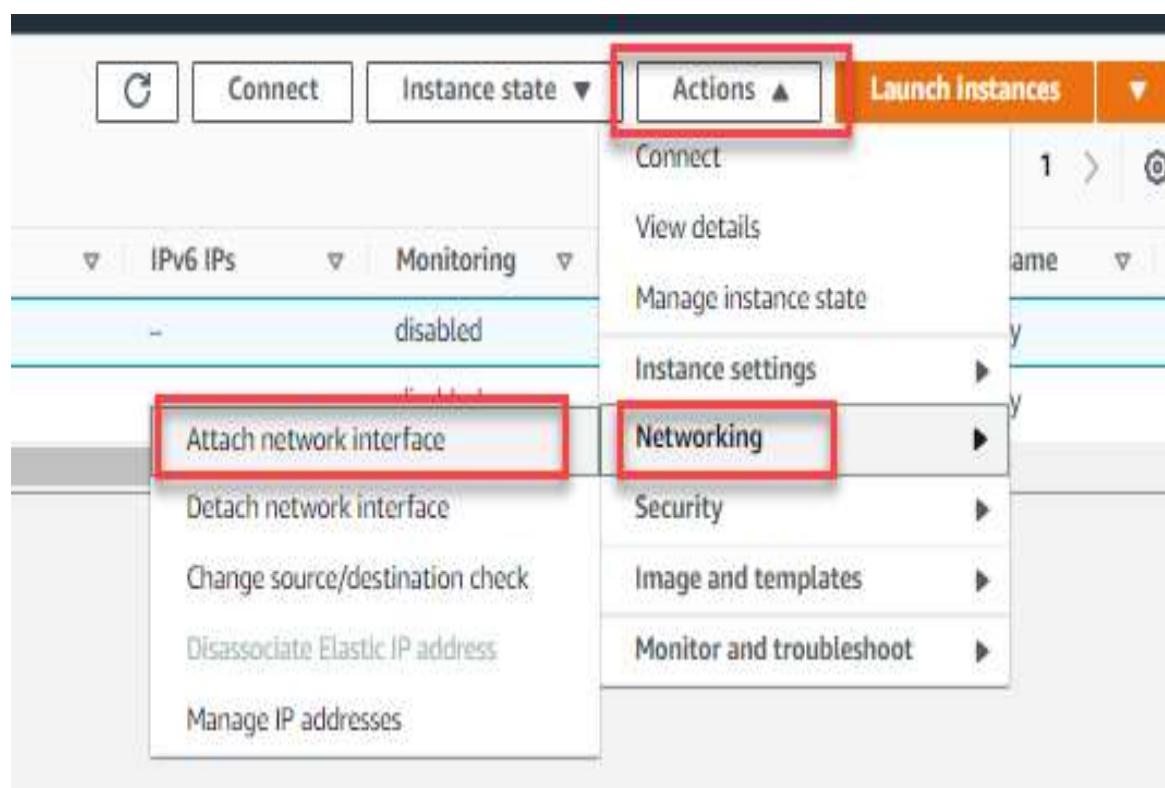
Choose **Instances**.

Select the checkbox next to the Instance, then

**Actions > Networking > Attach network interface.**

Select desired network interface. Network cards may also be chosen if the instance supports them.

Choose **Attach**.



## Lab: Create an Internet Gateway

The objective of this lab is to create an internet gateway.

Open the Amazon VPC console <https://console.aws.amazon.com/vpc/>.

Choose **Internet Gateways > Create internet gateway** from the navigation pane.



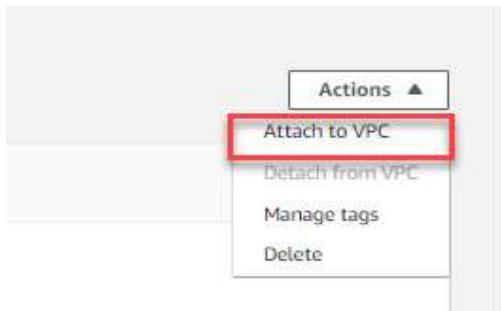
Name the gateway (optional).

Add or Remove Tag (optional).

Click **Create internet gateway**.

The screenshot shows the 'Create internet gateway' wizard. In the 'Internet gateway settings' step, there is a 'Name tag' input field containing 'MyIGW'. This field is highlighted with a red box. Below it is a 'Tags - optional' section with a single tag 'Name: MyIGW'. At the bottom right are 'Cancel' and 'Create Internet gateway' buttons, with 'Create Internet gateway' also highlighted with a red box.

Select the just-created internet gateway Then follow **Actions > Attach to VPC**.

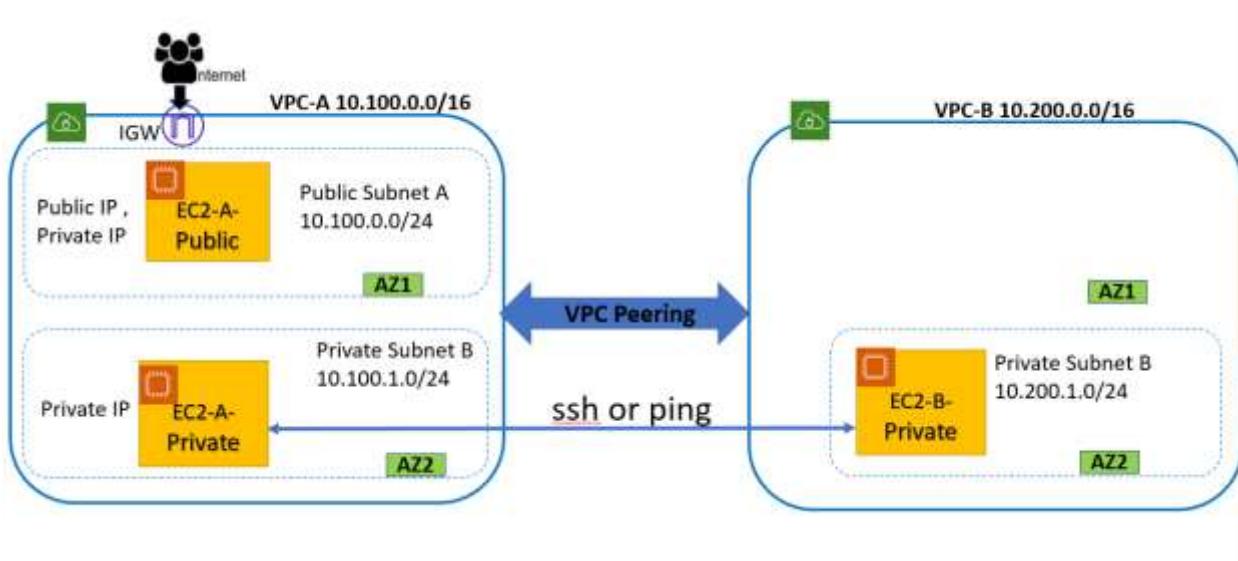


Select the target VPC from the list, and then choose **Attach internet gateway**.

The screenshot shows the 'Attach to VPC' configuration dialog. It lists a single VPC ('VPC: igw-0b03bbc4d023ff237') and allows selecting it. At the bottom are 'Cancel' and 'Attach Internet gateway' buttons, with 'Attach Internet gateway' highlighted with a red box.

## Lab: AWS VPC Peering

In this lab we will set up VPC peering. First, we will create the VPCs, and then we will establish VPC peering.



Create the first VPC:

VPC-A (IP address 10.100.0.0/16).

Screenshot of the AWS Create VPC settings page:

**VPC settings**

Name tag - optional: VPC-A

IPv4 CIDR block: 10.100.0.0/16

IPv6 CIDR block: No IPv6 CIDR block selected

Tenancy: Info

Most VPCs will have an internet gateway, so we will create one here:

Create Internet Gateways(VPC-A-IGW) and Attach it to VPC-A.

The image shows two screenshots of the AWS VPC interface. The top screenshot is titled 'Create internet gateway' and shows the configuration of a new internet gateway. It includes fields for a 'Name tag' (VPC-A-IGW) and a 'Tags - optional' section with a single tag ('Name' key, 'VPC-A-IGW' value). The bottom screenshot is titled 'Attach to VPC (igw-00a400d474dabbe16)' and shows the selection of a VPC to attach the gateway to. It lists 'Available VPCs' (vpc-0a64f967f24a5abbd - VPC-A) and provides a 'Select a VPC' dropdown. A red box highlights the 'Attach internet gateway' button at the bottom right of the attachment interface.

VPC > Internet gateways > Create internet gateway

## Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

### Internet gateway settings

Name tag  
Creates a tag with a key of 'Name' and a value that you specify.

VPC-A-IGW

**Tags - optional**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
Q Name	X Q VPC-A-IGW Remove

VPC > Internet gateways > Attach to VPC (igw-00a400d474dabbe16)

## Attach to VPC (igw-00a400d474dabbe16) Info

### VPC

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs  
Attach the internet gateway to this VPC.

Select a VPC

vpc-0a64f967f24a5abbd - VPC-A

▶ AWS Command Line Interface command

Cancel Attach internet gateway

Create public subnets (10.100.0.0/24) inside VPC-A. This is for demo purposes.

The screenshot shows the 'Create subnet' wizard in the AWS VPC console. The top navigation bar indicates the path: VPC > Subnets > Create subnet. The main title is 'Create subnet' with an 'Info' link. The first section, 'VPC', shows the selected VPC ID: 'vpc-0a64f967f24a5abbd (VPC-A)'. Below it, 'Associated VPC CIDRs' lists the IPv4 CIDR: '10.100.0.0/16'. The second section, 'Subnet settings', contains the configuration for 'Subnet 1 of 1'. It includes fields for 'Subnet name' (set to 'VPC-A-Subnet-Public'), 'Availability Zone' (set to 'No preference'), and 'IPv4 CIDR block' (set to '10.100.0.0/24'). A 'Tags - optional' button is present. At the bottom right of the wizard, there are 'Cancel' and 'Create subnet' buttons, with the 'Create subnet' button being highlighted by a red rectangle.

VPC > Subnets > Create subnet

## Create subnet Info

### VPC

**VPC ID**  
Create subnets in this VPC.

vpc-0a64f967f24a5abbd (VPC-A)

### Associated VPC CIDRs

IPv4 CIDRs:

10.100.0.0/16

### Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

#### Subnet 1 of 1

**Subnet 1 of 1**

**Subnet name**  
Create a tag with a key of 'Name' and a value that you specify.

VPC-A-Subnet-Public

The name can be up to 256 characters long.

**Availability Zone** Info  
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

No preference

**IPv4 CIDR block** Info

Q 10.100.0.0/24 X

▶ Tags - optional

Remove

Add new subnet

Cancel **Create subnet**

Create Public Route table then add internet route (destination :0.0.0.0/0 & target igw) and associate with Public Subnet of VPC-A.

The screenshot shows two sequential steps in the AWS VPC console:

**Create route table** (Step 1):

- Route table settings:**
  - Name - optional:** VPC-A-RT-Public
  - VPC:** The VPC to use for this route table. (vpc-0a64f967f24a5abbd (VPC-A))

**Edit routes** (Step 2):

- Routes:** A table showing two routes:

Destination	Target	Status	Propagated
10.100.0.0/16	local	Active	No
0.0.0.0/0	igw-00a400d474dabbe16	-	No
- Buttons at the bottom:** Add route, Cancel, Preview, Save changes (highlighted with a red box).
- Bottom navigation:** Routes (highlighted with an orange box), Subnet associations, Edge associations, Route propagation, Tags.
- Routes table (bottom):** Shows the same two routes listed again.

VPC > Route tables > rtb-026c1c39943f9e3d0 > Edit subnet associations

### Edit subnet associations

Change which subnets are associated with this route table.

**Available subnets (1/1)**

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
VPC-A-Subnet-Public	subnet-03f14efac0ff2e98a	10.100.0.0/24	-	rtb-026c1c39943f9e3d0

**Selected subnets:**

subnet-03f14efac0ff2e98a / VPC-A-Subnet-Public X

**Save associations**

Create Private Subnets (10.100.1.0/24) inside VPC-A.

### Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

#### Subnet 1 of 1

**Subnet name**  
Create a tag with a key of 'Name' and a value that you specify.

VPC-A-Subnet-Private

The name can be up to 256 characters long.

**Availability Zone** Info  
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

No preference ▾

**IPv4 CIDR block** Info  
10.100.1.0/24 X

**Tags - optional**

Key	Value - optional	Remove
Name	VPC-A-Subnet-Private	X

Create Private Route table (no need to connect to internet) and associate with private subnet of VPC-A.

The screenshot shows the 'Create route table' wizard in the AWS VPC console. The first step, 'Route table settings', is completed with a name 'VPC-A-RT-Private' and assigned to VPC 'VPC-A'. The second step, 'Edit subnet associations', lists available subnets and selected subnets. The 'Selected subnets' section contains 'subnet-0d4a8b25cd5a6290c / VPC-A-Subnet-Private'. The 'Save associations' button is highlighted with a red box.

VPC > Route tables > Create route table

## Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

### Route table settings

Name - optional  
Create a tag with a key of 'Name' and a value that you specify.

VPC-A-RT-Private

VPC  
The VPC to use for this route table.

vpc-0a64f967f24a5abbd (VPC-A)

### Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/2)						
<input type="text"/> Filter subnet associations						
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID	Actions	
<input type="checkbox"/> VPC-A-Subnet-Public	subnet-05ff14efac0ff82e98a	10.100.0.0/24	-	rtb-026c1c39943fb63d9 / VPC-A-RT-Public	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
<input checked="" type="checkbox"/> VPC-A-Subnet-Private	subnet-0d4a8b25cd5a6290c	10.100.1.0/24	-	Main (rtb-04f306230b3verf720l)	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>

Selected subnets

subnet-0d4a8b25cd5a6290c / VPC-A-Subnet-Private

## Launch Public EC2-A inside Public Subnet of VPC-A.

This screenshot shows the 'Step 3: Configure Instance Details' page of the AWS Launch Wizard. The page title is 'Step 3: Configure Instance Details'. Below it, a sub-instruction reads: 'Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.' The configuration fields include:

- Number of instances:** 1
- Purchasing option:** Request Spot Instances (unchecked)
- Network:** vpc-0a649907d4a6abb0 | VPC-A
- Subnet:** subnet-03f14ef0a0f82e96a | VPC-A-Subnet-Public
- Auto-assign Public IP:** Enable
- Placement group:** Add instance to placement group (unchecked)
- Capacity Reservation:** Open
- Domain join directory:** No directory
- IAM role:** None

At the bottom right are buttons for 'Cancel', 'Previous', 'Review and Launch' (which is highlighted in blue), and 'Next: Add Storage'.

## Launch Private EC2-A Inside Private Subnet of VPC-A.

This screenshot shows the 'Step 3: Configure Instance Details' page of the AWS Launch Wizard, identical to the previous one but for a private subnet. The configuration fields are the same, except for the subnet selection:

- Number of instances:** 1
- Purchasing option:** Request Spot Instances (unchecked)
- Network:** vpc-0a649907d4a6abb0 | VPC-A
- Subnet:** subnet-004a9b25cd5a0280c | VPC-A-Subnet-Private
- Auto-assign Public IP:** Disable
- Placement group:** Add instance to placement group (unchecked)
- Capacity Reservation:** Open
- Domain join directory:** No directory
- IAM role:** None

At the bottom right are buttons for 'Cancel', 'Previous', 'Review and Launch' (which is highlighted in blue), and 'Next: Add Storage'.

Create second VPC: VPC-B.

VPC > Your VPCs > Create VPC

## Create VPC Info

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances.

### VPC settings

Name tag - *optional*  
Creates a tag with a key of 'Name' and a value that you specify.

VPC-B

IPv4 CIDR block Info

10.200.0.0/16

IPv6 CIDR block Info

No IPv6 CIDR block  
 Amazon-provided IPv6 CIDR block  
 IPv6 CIDR owned by me

Tenancy Info

Create Private Subnets (10.200.1.0/24) inside VPC-B.

VPC > Subnets > Create subnet

## Create subnet Info

**VPC**

VPC ID  
Create subnets in this VPC.  
vpc-036ea69f34dfe817f (VPC-B)

Associated VPC CIDRs

IPv4 CIDRs  
10.200.0.0/16

**Subnet settings**  
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

**Subnet settings**  
Specify the CIDR blocks and Availability Zone for the subnet.

**Subnet 1 of 1**

**Subnet name**  
Create a tag with a key of 'Name' and a value that you specify.

**Availability Zone** [Info](#)  
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

**IPv4 CIDR block** [Info](#)

**Tags - optional**

Key	Value - optional	Remove
<input type="text" value="Name"/>	<input type="text" value="VPC-B-Subnet-Private"/>	<input type="button" value="Remove"/>
<input type="button" value="Add new tag"/>		

Create Private Route table (no need to connect to internet) and associate with private subnet.

VPC > Route tables > Create route table

### Create route table [Info](#)

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

**Route table settings**

**Name - optional**  
Create a tag with a key of 'Name' and a value that you specify.

**VPC**  
The VPC to use for this route table.

**Tags**

VPC > Route tables > rt-05616299cbe33ff > Edit subnet associations

### Edit subnet associations

Change which subnets are associated with this route table.

**Available subnets (1/1)**

<input type="checkbox"/>	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/>	VPC-B-Subnet-Private	subnet-0d553679078c15e	10.200.1.0/24	-	Main (rt-04635164337abfa)

**Selected subnets**

subnet-0d5536790796c35e / VPC-B-Subnet-Private <input type="button" value="X"/>
---------------------------------------------------------------------------------

## Launch Private EC2-A inside Private Subnet of VPC-B.

**Step 3: Configure Instance Details**

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of Instances	1	Launch Into Auto Scaling Group
Purchasing option	<input type="checkbox"/> Request Spot Instances	
Network	vpc-03bea63f04ab617f   VPC-B	Create new VPC
Subnet	subnet-0d5580780786c35e   VPC-B-Subnet-Private	Create new subnet 261 IP Addresses available
Auto-assign Public IP	Disable	
Placement group	<input type="checkbox"/> Add instance to placement group	
Capacity Reservation	Open	
Domain join directory	No directory	Create new directory
IAM role	None	Create new IAM role

**Buttons:** Cancel, Previous, Review and Launch, Next: Add Storage

**Step 6: Configure Security Group**

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group:  Create a new security group.  
 Select an existing security group.

Security group name:	VPC-B-SG-Private			
Description:	VPC-B-SG-Private			
Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 10.100.1.0/24	e.g. SSH for Admin Desktop
All ICMP - (IPv4)	ICMP	0 - 65535	Custom 10.100.1.0/24	e.g. SSH for Admin Desktop

**Buttons:** Add Rule, Cancel, Previous, Review and Launch

Connect Public subnet EC2 with Private subnet EC2 of VPC-A.

```

ec2-user@ip-10-100-1-210:~ 
login as: ec2-user
Authenticating with public key "imported-openssh-key"
[ec2-user@ip-10-100-1-210 ~] $ ssh ec2-user@10.100.1.210
The authenticity of host '10.100.1.210 (10.100.1.210)' can't be established.
ECDSA key fingerprint is SHA256:chDvUgidLCo5k94BS6ngFDu2EHB0aJ0M6WZ4xRNm6k.
ECDSA key fingerprint is MD5:43:6b:09:3d:43:9d:ed:6f:1c:3f:ef:c1:48:7b:f8:5d.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.100.1.210' (ECDSA) to the list of known hosts.
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-10-100-0-38 ~] $ vim demo.pem
[ec2-user@ip-10-100-0-38 ~] $ chmod 400 demo.pem
[ec2-user@ip-10-100-0-38 ~] $ ssh -i demo.pem ec2-user@10.100.1.210
[ec2-user@ip-10-100-1-210 ~] $ 

```

## VPCs Peering

Connect Private subnet EC2 of VPC-A to Private subnet EC2 of VPC-B.

The screenshot shows the AWS VPC Peering Connections interface. On the left, there's a sidebar with options like New VPC Experience, Egress Only Internet Gateways, Carrier Gateways, DHCP Options Sets, Elastic IPs, Managed Prefix Lists, Endpoints, Endpoint Services, NAT Gateways, Peering Connections (selected), and Security (Network ACLs, Security Groups). The main area is titled "Peering connections" and shows a table with columns: Name, Peering connection ID, Status, Requester VPC, Acceptor VPC, and Request. A message at the bottom says "No peering connection found". Below the table, it says "Select a peering connection above".

**Create peering connection**

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them privately. [Info](#)

**Peering connection settings**

Name - *optional*  
Create a tag with a key of 'Name' and a value that you specify.

VPC-A-VPC-B-Peering

Select a local VPC to peer with  
VPC ID (Requester)  
vpc-0a64f967f24a5abbd (VPC-A)

VPC CIDRs for vpc-0a64f967f24a5abbd (VPC-A)

CIDR	Status	Status reason
10.100.0.0/16	Associated	-

10.100.0.0/16 Associated

Select another VPC to peer with  
Account  
 My account  
 Another account  
Region  
 This Region (us-east-1)  
 Another Region  
VPC ID (Acceptor)  
vpc-036ea69f34dfe817f (VPC-B)

VPC CIDRs for vpc-036ea69f34dfe817f (VPC-B)

CIDR	Status	Status reason
10.200.0.0/16	Associated	-

**Tags**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter.

Set up routing table for VPC-A-RT-Private.

Destination Target Status Propagated

10.100.0.0/16	local	Active	No
10.200.1.0/24	pcr-0f8f488ac2fb6df9e3	Active	No

Add route Cancel Preview Save changes

Set up the routing table for VPC-B-RT-Private.

Destination Target Status Propagated

10.200.0.0/16	local	Active	No
10.100.1.0/24	pcr-0f8f488ac2fb6df9e3	Active	No

Add route Cancel Preview Save changes

Testing connectivity. Pinging from private EC2 of VPC-A -----private EC2 of VPC-B.

```
--> 10.200.1.0 ping statistics -->
3 packets transmitted, 3 received, 0% packet loss, time 7161ms
rtt min/avg/max/mdev = 0.723/0.725/0.729/0.049 ms
ping: user stopped 10-100-1-310.1.5 ping to 10.200.1.18
PING 10.200.1.0 (10.200.1.0) 56(84) bytes of data
44 bytes from 10.200.1.0: icmp_seq=1 ttl=64 time=0.679 ms
44 bytes from 10.200.1.0: icmp_seq=2 ttl=64 time=0.826 ms
44 bytes from 10.200.1.0: icmp_seq=3 ttl=64 time=0.834 ms
44 bytes from 10.200.1.0: icmp_seq=4 ttl=64 time=0.759 ms
44 bytes from 10.200.1.0: icmp_seq=5 ttl=64 time=0.778 ms
44 bytes from 10.200.1.0: icmp_seq=6 ttl=64 time=0.800 ms
44 bytes from 10.200.1.0: icmp_seq=7 ttl=64 time=0.757 ms
44 bytes from 10.200.1.0: icmp_seq=8 ttl=64 time=0.767 ms
44 bytes from 10.200.1.0: icmp_seq=9 ttl=64 time=0.834 ms
44 bytes from 10.200.1.0: icmp_seq=10 ttl=64 time=0.760 ms
44 bytes from 10.200.1.0: icmp_seq=11 ttl=64 time=0.764 ms
44 bytes from 10.200.1.0: icmp_seq=12 ttl=64 time=0.705 ms
44 bytes from 10.200.1.0: icmp_seq=13 ttl=64 time=0.751 ms
44 bytes from 10.200.1.0: icmp_seq=14 ttl=64 time=0.725 ms
```

Edit Security group private EC2 of VPC-A.

Testing connectivity. Pinging from private EC2 of VPC-B-----private EC2 of VPC-A.

The screenshot shows the 'Edit inbound rules' page for a security group named 'sg-049c6a1003b93ae18 - VPC-A-SG-Private'. The page displays two inbound rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-00b5ee4030cd2d385	SSH	TCP	22	Custom: 10.100.0.0/24	
	All ICMP - IPv4	ICMP	All	Custom: 10.200.1.0/24	

Buttons at the bottom include 'Cancel', 'Previous changes', and a highlighted 'Save rules' button.

```
[ec2-user@ip-10-200-1-14:~]# ECDSSA key fingerprint is MD5:95:5c:4b:3b:27:0f:58:01:75:1f:d7:9d:73:40:82:d8. Are you sure you want to continue connecting (yes/no)? yes Warning: Permanently added '10.200.1.8' (ECDSA) to the list of known hosts. Permission denied (publickey,gssapi-keyex,gssapi-with-mic). [ec2-user@ip-10-100-1-210 ~]$ vim demo.pem [ec2-user@ip-10-100-1-210 ~]$ chmod 400 demo.pem [ec2-user@ip-10-100-1-210 ~]$ ssh -i demo.pem ec2-user@10.200.1.8
```

Amazon Linux 2 AMI

```
[ec2-user@ip-10-200-1-8 ~]$ ping 10.100.1.210
```

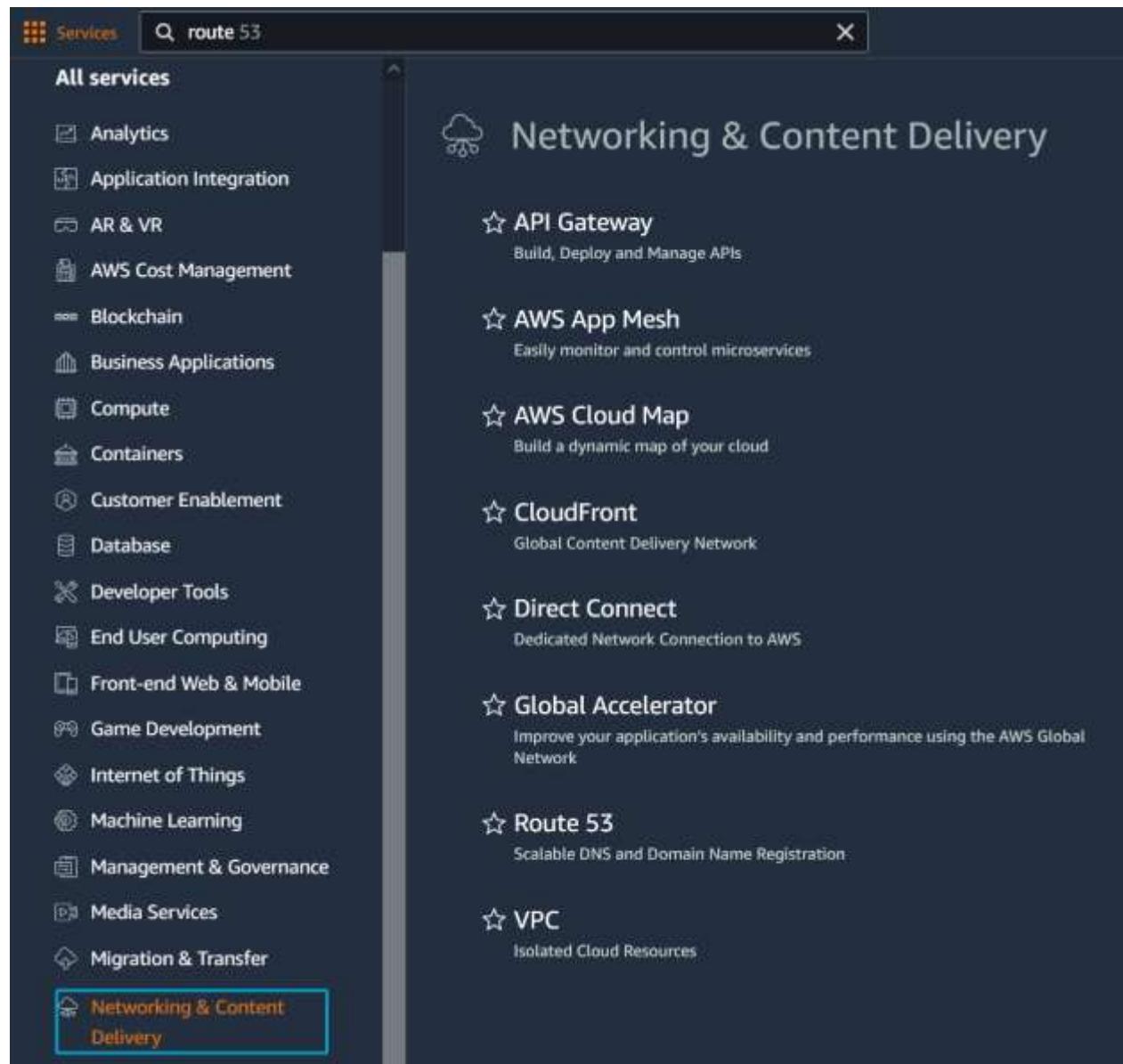
PING 10.100.1.210 (10.100.1.210) 56(84) bytes of data.  
64 bytes from 10.100.1.210: icmp\_seq=274 ttl=64 time=0.776 ms  
64 bytes from 10.100.1.210: icmp\_seq=275 ttl=64 time=0.737 ms  
64 bytes from 10.100.1.210: icmp\_seq=276 ttl=64 time=0.801 ms  
64 bytes from 10.100.1.210: icmp\_seq=277 ttl=64 time=3.17 ms  
64 bytes from 10.100.1.210: icmp\_seq=278 ttl=64 time=0.784 ms  
64 bytes from 10.100.1.210: icmp\_seq=279 ttl=64 time=16.9 ms  
64 bytes from 10.100.1.210: icmp\_seq=280 ttl=64 time=0.870 ms  
64 bytes from 10.100.1.210: icmp\_seq=281 ttl=64 time=0.799 ms  
64 bytes from 10.100.1.210: icmp\_seq=282 ttl=64 time=0.796 ms  
64 bytes from 10.100.1.210: icmp\_seq=283 ttl=64 time=0.820 ms  
64 bytes from 10.100.1.210: icmp\_seq=284 ttl=64 time=0.794 ms

## Chapter 7 Labs

### Lab: Create a Route 53 Hosted Zone

In this lab we will create a public hosted DNS zone using AWS Route 53. Note, you will need to purchase a domain name through AWS.

From the management console, Go to Services > Networking > Route 53.



Register a domain name by following the prompts. Make sure to leave Whois privacy enabled.

The screenshot shows the AWS Route 53 Registered domains interface. On the left, a sidebar lists navigation options: Dashboard, Hosted zones, Health checks, Traffic flow, Traffic policies, Policy records, Domains, Registered domains (which is selected and highlighted in orange), and Pending requests. The main content area is titled "Registered domains". It features three buttons at the top: "Register Domain" (blue), "Transfer Domain" (white), and "Domain Billing Report" (light gray). Below these are a search bar with placeholder text "Search domains by prefix" and a "Domain Name" input field. A dropdown menu next to the domain name field is set to "Privacy Protection". A large, semi-transparent watermark in the background says "No doi".

AWS has automatically created a Hosted Zone for the domain name. Go back to the main Route 53 dashboard and click “Hosted Zone” to view it.

The screenshot shows the AWS Route 53 Dashboard. The left sidebar includes "Route 53" (selected), "Dashboard" (highlighted in orange), "Hosted zones", "Health checks", and a collapsed "Traffic flow" section containing "Traffic policies" and "Policy records". Below the sidebar, there's a note about verifying and purchasing a domain. The main content area is titled "Route 53 Dashboard" with an "Info" link. It displays "1 Hosted zone" and a "DNS management" section. Below this, there's a table for "Availability for 'mytechblog.com'" showing one entry: "mytechblog.com" with a red "X" icon and the status "Unavailable". Under "Related domain suggestions", another table shows "findmytechblog.com" with a green checkmark icon and the status "Available", along with a price of "\$12.00" and an "Add to cart" button.

Click on your domain.

The screenshot shows the AWS Route 53 Hosted zones page. At the top, there is a breadcrumb navigation: Route 53 > Hosted zones. Below the navigation, a section titled "Hosted zones (1)" is displayed. A note says "Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings". There are four buttons: "View details", "Edit", "Delete", and a prominent orange "Create hosted zone" button. Below these buttons is a search bar with the placeholder text "Filter hosted zones by property or value". A table follows, with columns: Domain name, Type, and Created by. The table contains one row for the domain "nicklovesthecloud.com", which is Public and was created by Route 53.

On the Hosted Zones page, click **Create record**.

The screenshot shows the AWS Route 53 Hosted zones page for the domain "nicklovesthecloud.com". The breadcrumb navigation shows "Route 53 > Hosted zones > nicklovesthecloud.com". The main title is "nicklovesthecloud.com" with an "Info" link. Below the title are three buttons: "Delete zone", "Test record", and "Configure query logging". A section titled "Hosted zone details" has an "Edit" button. Below this, there are tabs for "Records (2)", "DNSSEC signing", and "Hosted zone tags (0)". The "Records (2)" tab is selected. A note at the top of this section says "Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings". At the bottom of the records section are four buttons: "C", "Delete record", "Import zone file", and a prominent orange "Create record" button, which is highlighted with a red box. Below these buttons is a search bar with the placeholder text "Filter records by property or value".

Find the Public IPv4 address of your web server and copy it to the value box. Make sure to select a DNS “A” record from the dropdown. Then click **Create record**.

The image consists of two screenshots. The top screenshot shows the EC2 Instances page with the instance ID i-0ea5f935541f81eb7 selected. The Public IPv4 address is listed as 18.188.233.207. The bottom screenshot shows the Route 53 Quick create record interface. A new A record is being created for the domain nicklovesthecloud.com with the name blog and the value 18.188.233.207. The TTL is set to 300 seconds. The 'Create records' button is highlighted with a red box.

EC2 > Instances > i-0ea5f935541f81eb7

Instance summary for i-0ea5f935541f81eb7

Updated less than a minute ago

Instance ID: i-0ea5f935541f81eb7

Public IPv4 address: 18.188.233.207 | open address

Route 53 > Hosted zones > nicklovesthecloud.com > Create record

Quick create record

Record name: blog

Record type: A – Routes traffic to an IPv4 address and ...

Value: 18.188.233.207

TTL (seconds): 300

Routing policy: Simple routing

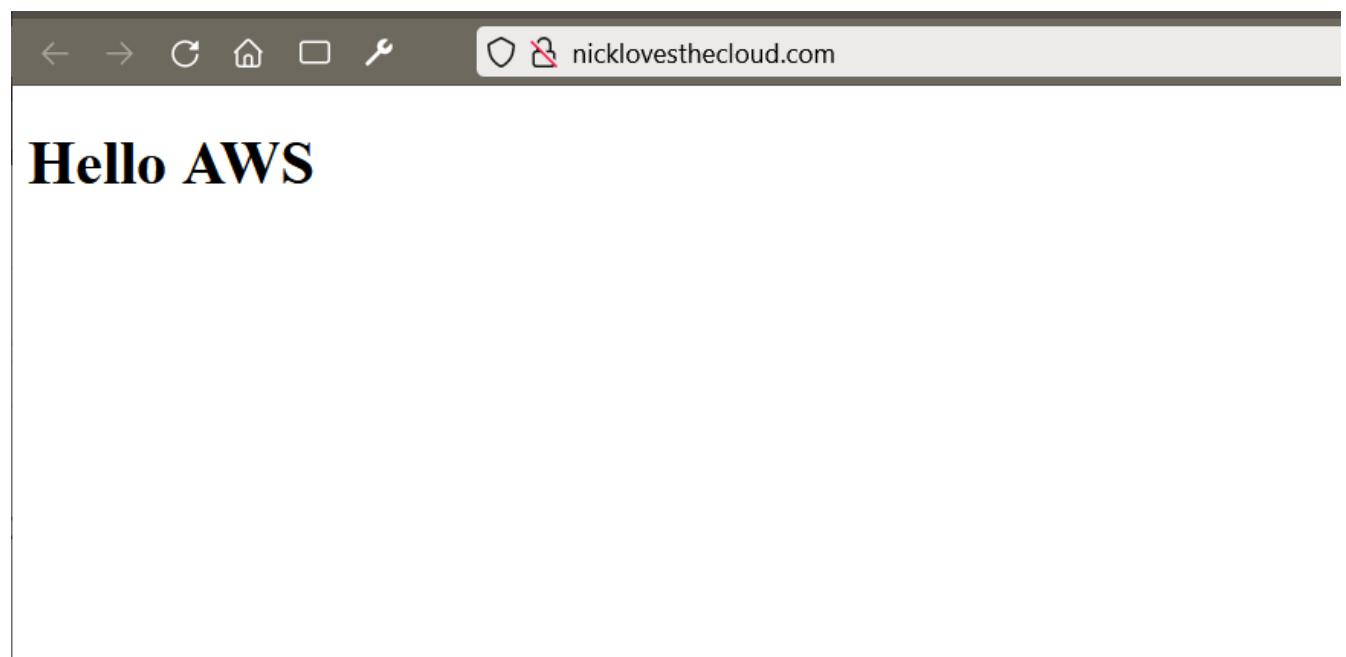
Add another record

Create records

Create a second record as follows.

The screenshot shows the 'Quick create record' interface in the AWS Route 53 console. The 'Record name' field contains 'www' and the 'Value' field contains '18.188.233.207'. The 'Record type' is set to 'A'. The 'TTL (seconds)' is set to '60'. The 'Routing policy' is set to 'Simple routing'. Below the TTL field, there are buttons for '+1m', '1h', and '1d'. A note below the TTL field says 'Recommended values: 60 to 172800 (two days)'. The URL in the browser bar is 'nicklovesthecloud.com'.

Your web server should now be reachable by going to your domain name.



## Lab: Create a DNS Health Check Using Route 53

From the management console, navigate to the Route 53 dashboard.

On the Route 53 dashboard click on “Health Checks”, then click “Create Health Check.”

The screenshot shows the AWS Route 53 dashboard. On the left, there is a navigation sidebar with links for 'Dashboard', 'Hosted zones', 'Health checks' (which is highlighted with a red box), 'Traffic flow', and 'Traffic policies'. The main content area is titled 'Route 53 Dashboard' and contains a sub-section titled 'DNS manager'. Below this, there is a large button labeled 'Create health check'.

Configure the health check with your domain name.

The screenshot shows the 'Configure health check' wizard. Step 1: Set up the health check. It asks 'What to monitor' (monitor or endpoint). The 'Endpoint' option is selected and highlighted with a red box. Below it, there is a section for specifying the endpoint, which is also highlighted with a red box. This section includes fields for Protocol (HTTP), IP address (18.188.233.207), Host name (www.nicklovesthecloud.com), Port (80), and Path (/index.html). There is also an 'Advanced configuration' link.

\* Required

[Cancel](#) [Next Step](#)

Set up SNS notifications for the health check.

### Get notified when health check fails

If you want CloudWatch to send you an Amazon SNS notification, such as an email, when the status of the health check changes to unhealthy, create an alarm and specify where to send notifications.

Create alarm    Yes    No [?](#)

CloudWatch sends you an Amazon SNS notification whenever the status of this health check is unhealthy for at least one minute. The alarm will be located in the **us-east-1** region.

Send notification to    Existing SNS topic    New SNS topic [?](#)

Topic name \*  [?](#)

Recipient email addresses \*  [?](#)

Separate multiple addresses with a comma, a semicolon, or a space.

\* Required

[Cancel](#)

[Previous](#)

[Create health check](#)

Congratulations, you've created a health check. Notice that the Status reads "Healthy".

Name	Status	Description	Alarms	ID
nicklovesthecoud.com	<span style="background-color: green; color: white; padding: 2px;">Healthy</span>	http://16.188.233.207:80/index.html	<span style="color: green;">1 of 1 in OK</span>	7bfac94d-

Test the health check by stopping the web server.

EC2 > Instances > i-0ea5f935541f81eb7

**Instance summary for i-0ea5f935541f81eb7** [Info](#)  
Updated less than a minute ago:

Instance ID <a href="#">i-0ea5f935541f81eb7</a>	Public IPv4 address -	Private IPv6 address -	Private IPv4 address <a href="#">172.31.24.11</a>	Start instance
IPv6 address -	Instance state <a href="#">Stopped</a>	Instance type t2.micro	Public IPv4 DNS -	Reboot instance
Private IPv4 DNS <a href="#">ip-172-31-24-11.us-east-2.compute.internal</a>	AWS Compute Optimizer finding <a href="#">Opt-in to AWS Compute Optimizer for recommendations.</a> <a href="#">Learn more</a>	IAM Role -	Elastic IP addresses -	Hibernate instance
VPC ID <a href="#">vpc-074561f609493c230</a>				Terminate instance
Subnet ID <a href="#">subnet-0244cd4b6adff7e416</a>				

[Details](#) [Security](#) [Networking](#) [Storage](#) [Status checks](#) [Monitoring](#) [Tags](#)

[Instance details](#) [Info](#)

Platform	AMI ID	Monitoring
----------	--------	------------

The health check should fail because the server is unreachable.

Name	Status	Description	Alarms
nicklovesthecloud.com	Unhealthy 15 minutes ago	http://18.188.233.207:80/index.html	1 of 1 in OK

Following the health check, restart the web server.

EC2 > Instances > i-0ea5f935541f81eb7

Instance summary for i-0ea5f935541f81eb7 [Info](#)

(Updated less than a minute ago)

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0ea5f935541f81eb7	-	172.31.24.11
IPv6 address	Instance state	Public IPv4 DNS
-	Stopped	-
Private IPv4 DNS	Instance type	Elastic IP addresses
ip-172-31-24-11.us-east-2.compute.internal	t2.micro	-
VPC ID	AWS Compute Optimizer finding	IAM Role
vpc-074561f609493c230	Opt-in to AWS Compute Optimizer for recommendations.   Learn more	-
Subnet ID		
subnet-0244cd4b6adef7e416		

[Details](#) [Security](#) [Networking](#) [Storage](#) [Status checks](#) [Monitoring](#) [Tags](#)

▼ Instance details [Info](#)

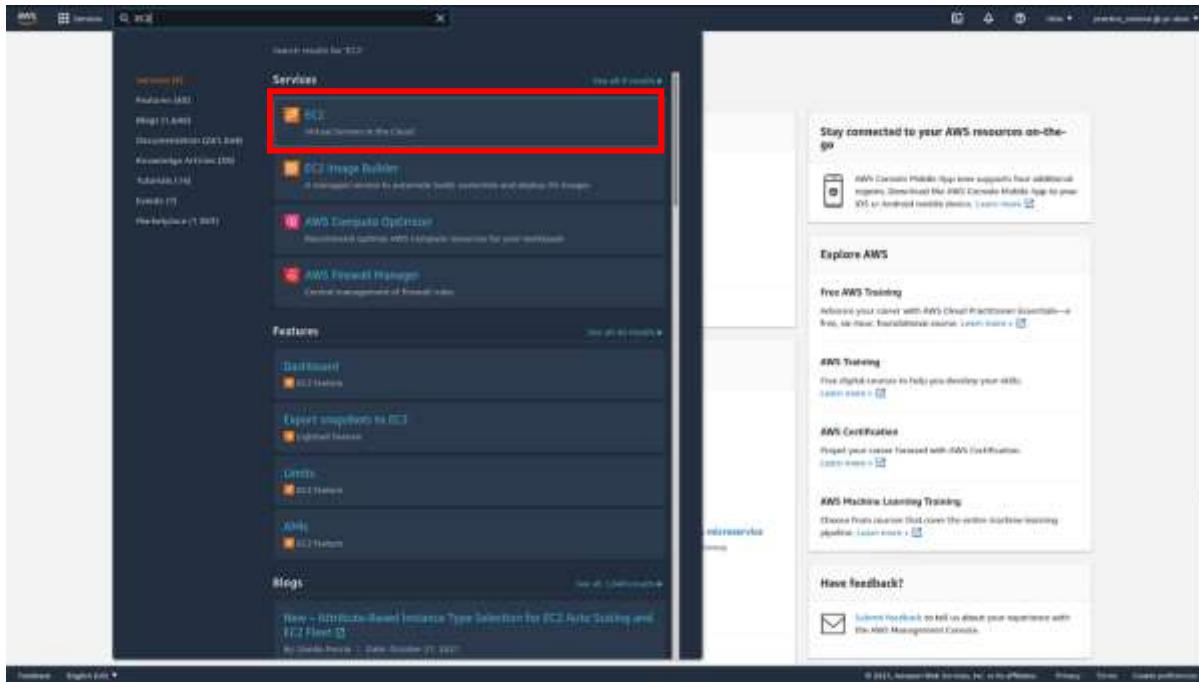
Platform	AMI ID	Monitoring
----------	--------	------------

## Lab: Set Up an Application Load Balancer & Target Groups

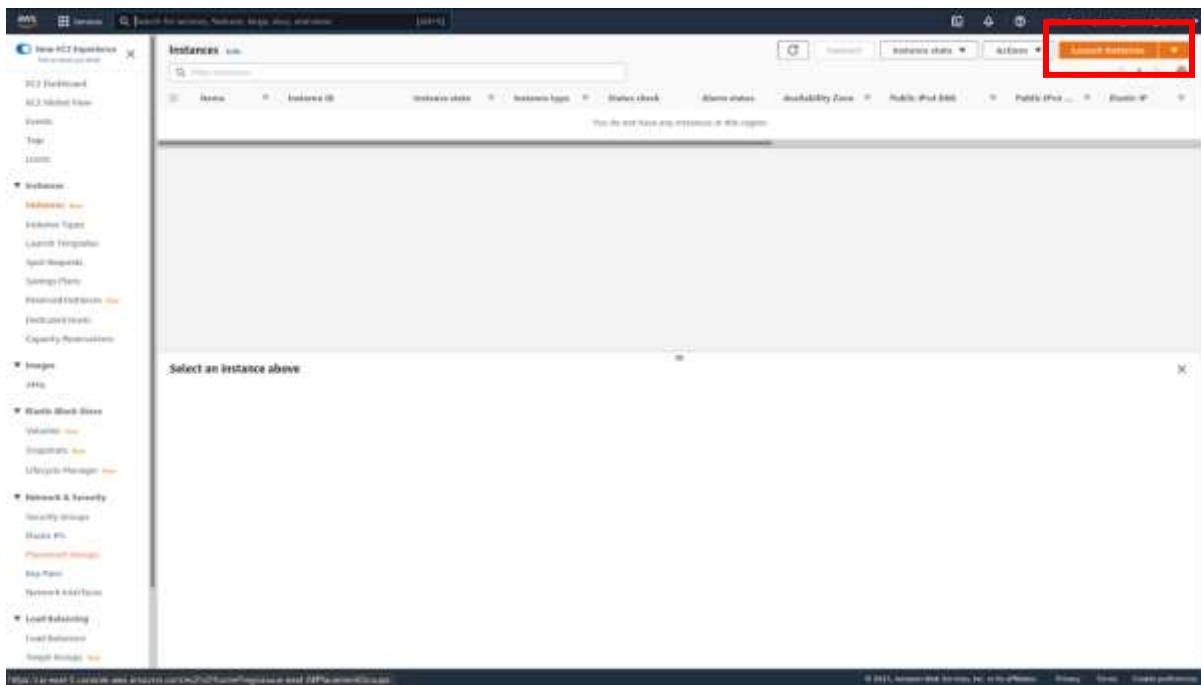
In this lab we will set up an application load balancer that will distribute traffic to a target group. Load balancing can play a major role in high-availability applications

### Configure Your Target Groups

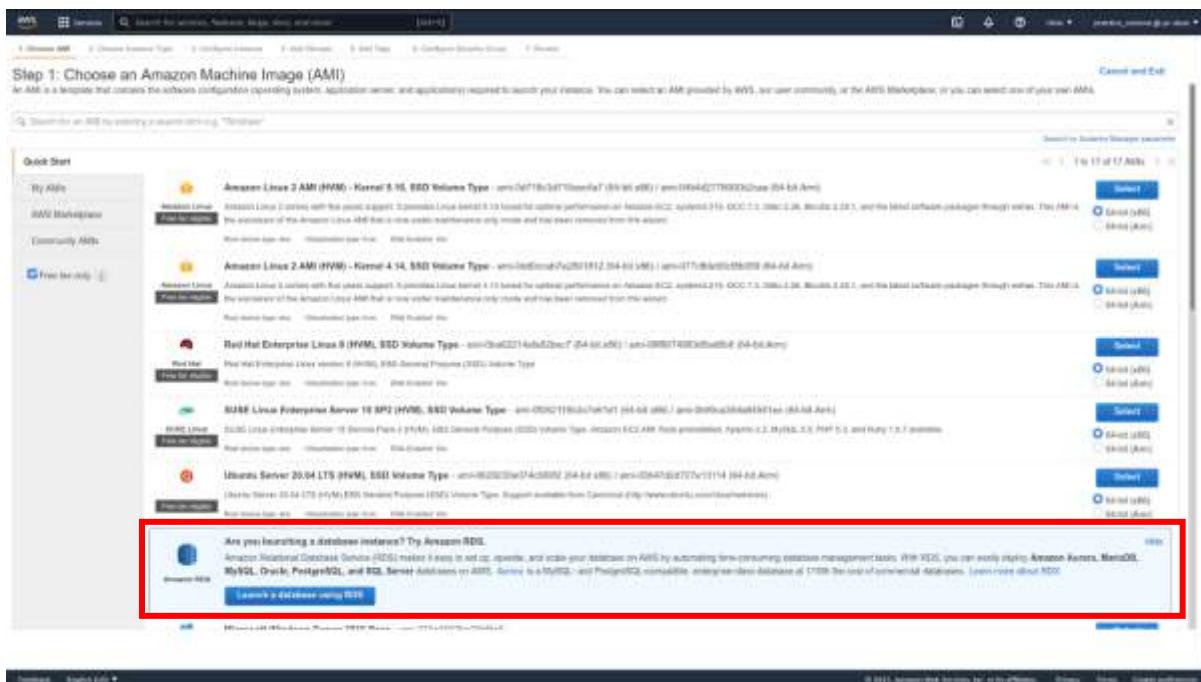
Before we create our load balancer, we need to include our instances in a Target Group. From the management console, navigate to the **EC2** console.



Navigate to the **Instances** section in the sidebar. Click on **Launch instances**.



Select the Free Tier image of your choice. We will choose Amazon Linux.



Select **t2.micro**. Click **Next: Configure Instance Details**.

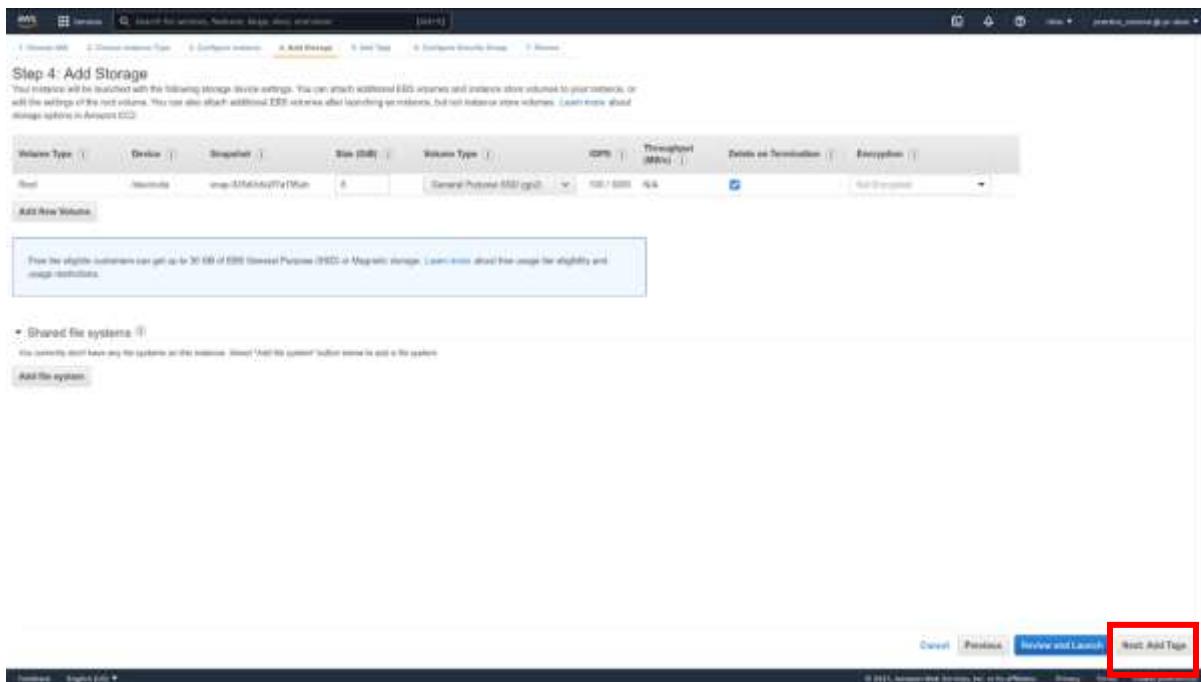
The screenshot shows the AWS Lambda Step 2: Choose an Instance Type page. A table lists various instance types with their details. The **t2.micro** row is selected and highlighted with a green background. At the bottom right of the page, the **Next: Configure Instance Details** button is highlighted with a red box.

Family	Name	vCPU	Memory (GB)	Instance Storage (GB)	Amazon ElastiCache Compatible	Network Performance	MP Support
I1	I1-micro	1	0.5	EBS only	No	Low I/O Moderate	Yes
I2	I2-micro	1	1	EBS only	No	Low I/O Moderate	Yes
I3	I3-micro	1	2	EBS only	No	Low I/O Moderate	Yes
M1	M1-micro	2	4	EBS only	No	Low I/O Moderate	Yes
M2	M2-micro	2	8	EBS only	No	Low I/O Moderate	Yes
M3	M3-micro	4	16	EBS only	No	Medium	Yes
M4	M4-micro	8	32	EBS only	No	Medium	Yes
T1	T1-micro	2	0.5	EBS only	Yes	Up to 5 IOPS	Yes
T2	T2-micro	2	1	EBS only	Yes	Up to 5 IOPS	Yes
T3	T3-micro	3	2	EBS only	Yes	Up to 5 IOPS	Yes
T4	T4-micro	2	4	EBS only	Yes	Up to 5 IOPS	Yes
T5	T5-micro	2	8	EBS only	Yes	Up to 5 IOPS	Yes
T6	T6-micro	4	16	EBS only	Yes	Up to 5 IOPS	Yes
V1	V1-micro	8	32	EBS only	Yes	Up to 5 IOPS	Yes
V2	V2-micro	2	0.5	EBS only	Yes	Up to 5 IOPS	Yes
V3	V3-micro	2	1	EBS only	Yes	Up to 5 IOPS	Yes

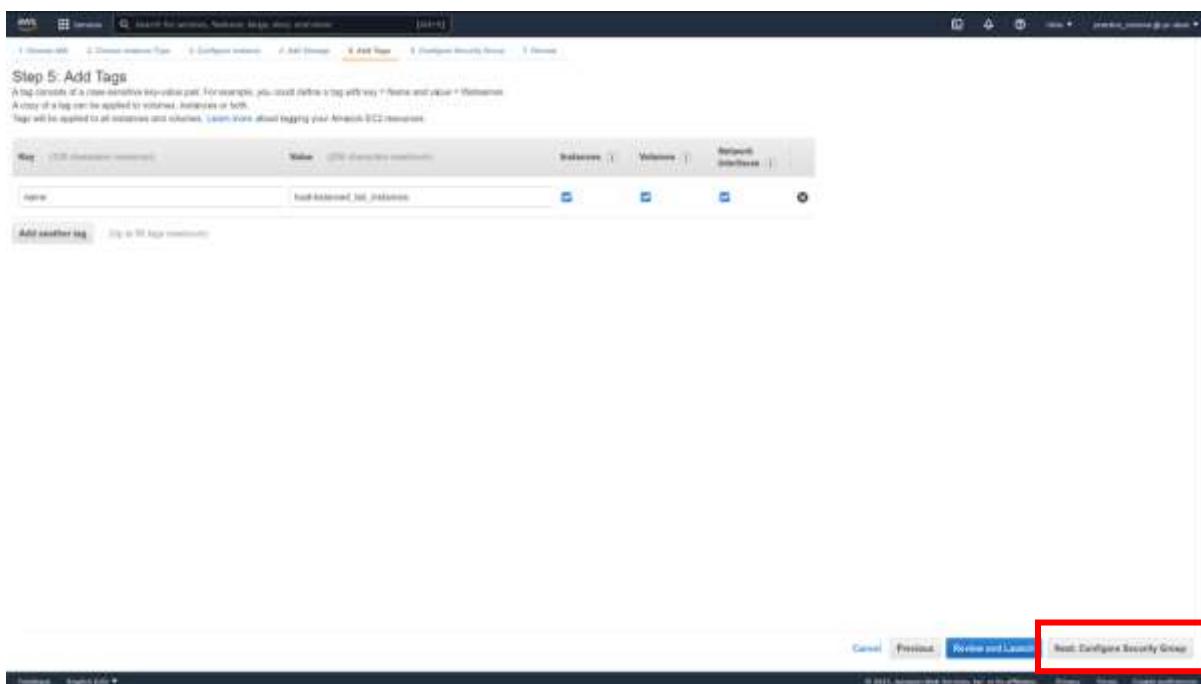
Set the Number of Instances to 3. Click **Next: Add Storage**.

The screenshot shows the AWS Lambda Step 3: Configure Instance Details page. The **Number of instances** field is set to **3** and highlighted with a red box. At the bottom right of the page, the **Next: Add Storage** button is highlighted with a red box.

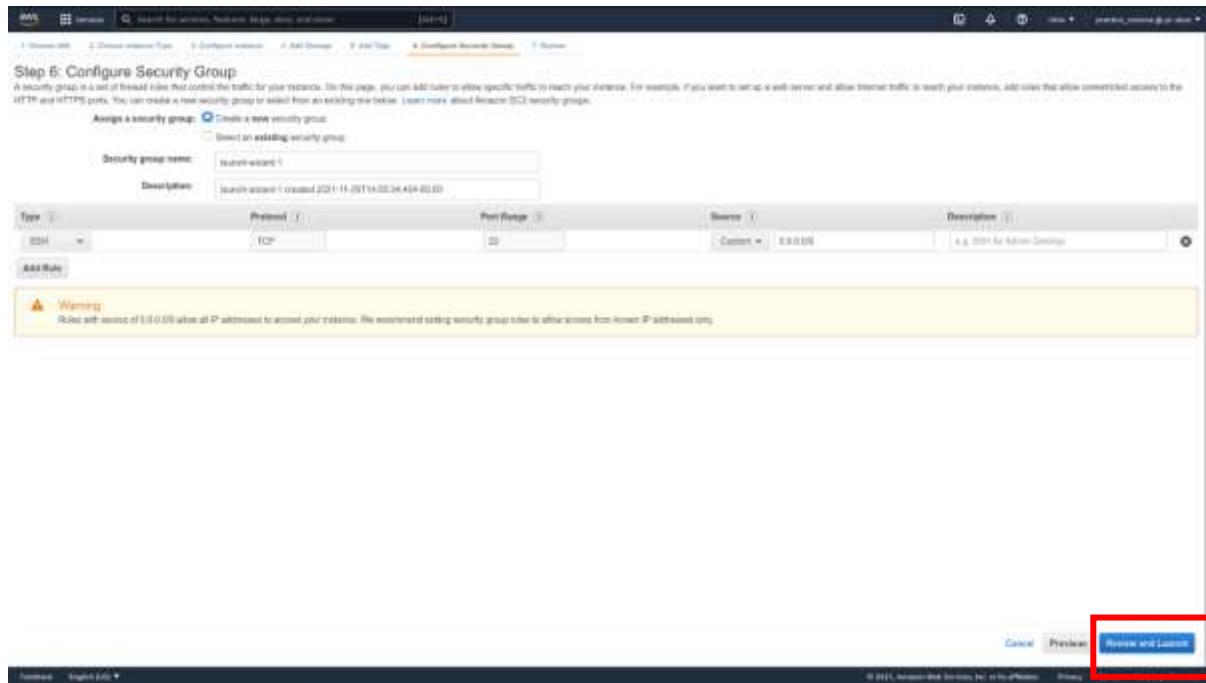
## Click Next: Add Tags.



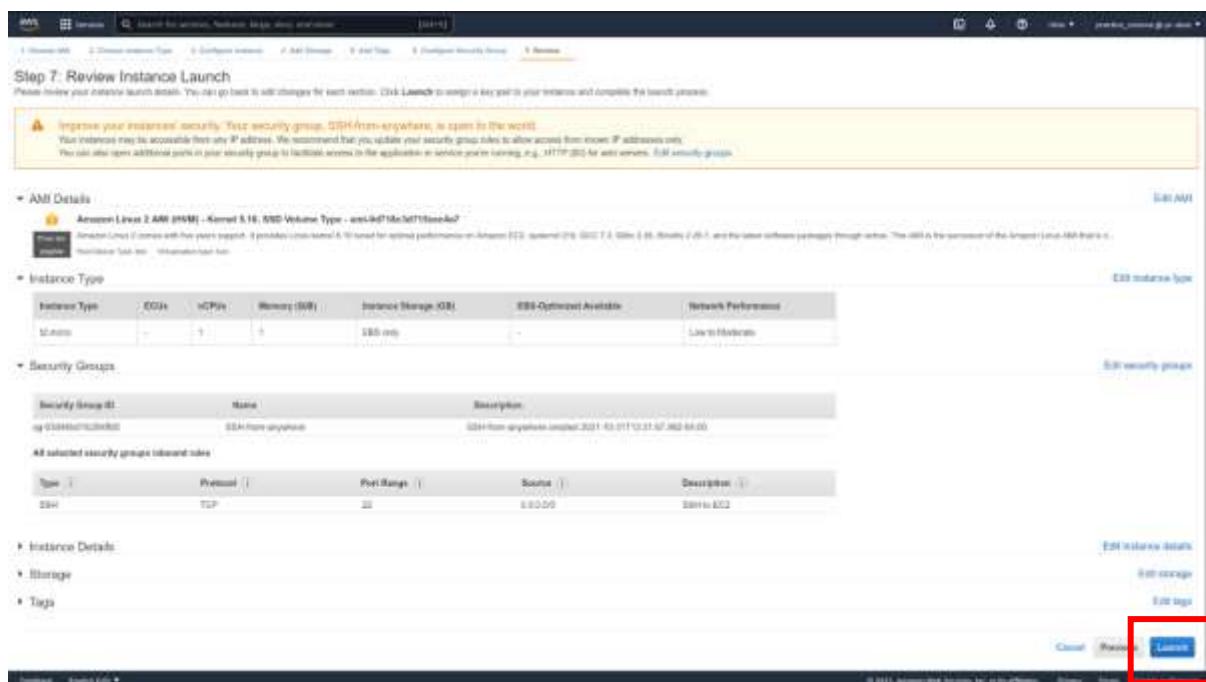
Add a name tag and give it a value. Click **Next: Configure Security Group**.



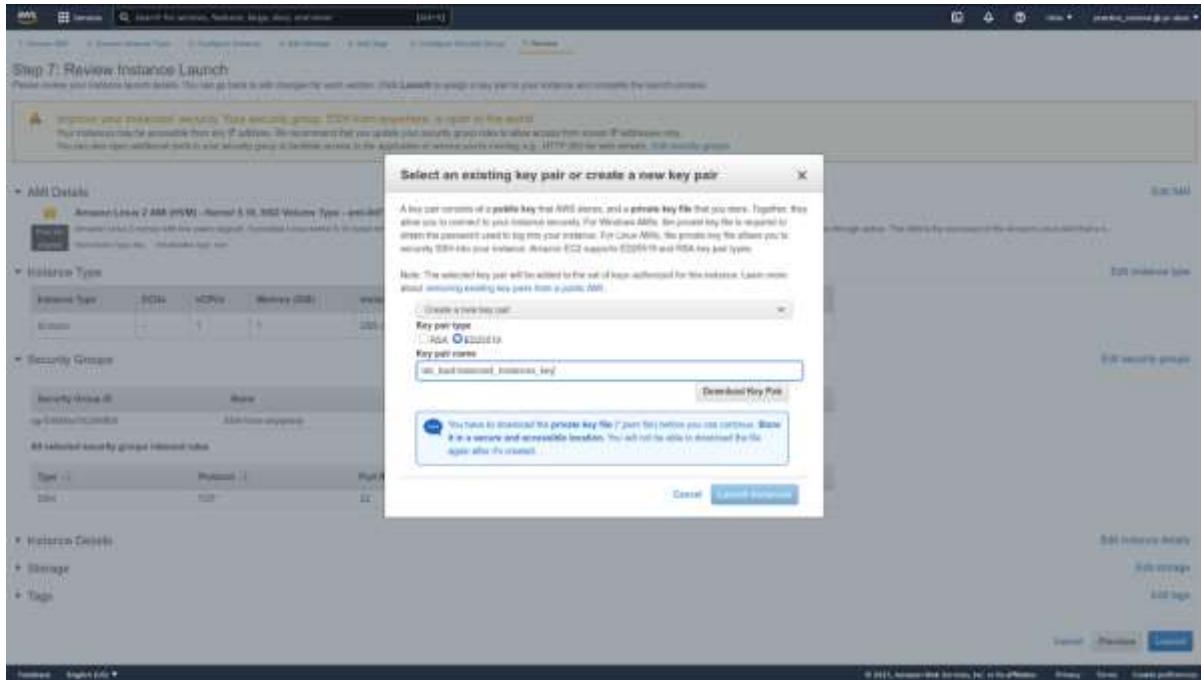
You can leave this at the defaults or use an existing Security Group. Click **Review and Launch**.



When you are ready, click **Launch**.



Give a name to the SSH key. Make sure to download the SSH key. When you have, click **Launch Instances**.



Now, let's go back to Target Groups.

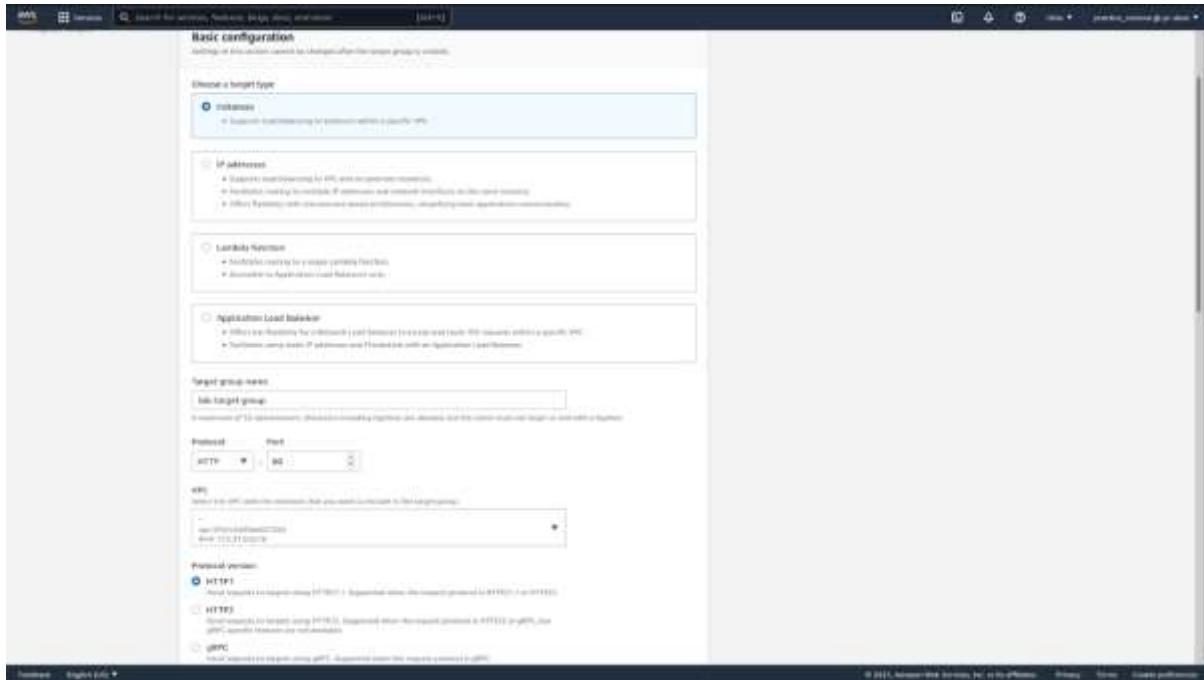
From the main EC2 console, in the sidebar, under **Load Balancing**, select **Target Groups**. If you can't see it, scroll down within the sidebar.

The screenshot shows the AWS EC2 Instances page. On the left, there is a sidebar with various services: EC2 Dashboard, EC2 Metrics View, Events, Tags, Lambda, and a large section for **Load Balancing** which is currently selected. Under Load Balancing, there are sub-options for Target Groups, Listener Integrations, and Spur-Request. The main content area displays a table titled "Instances [3]". The table has columns: Name, Instance ID, Instance State, Instance Type, Status Check, Alarm Status, Availability Zone, Public IP v4, and Private IP. Three instances are listed, all in the "running" state. The Public IP column shows values like 3.133.43.242, 3.133.191.152, and 19.27.29.193. The Private IP column shows values like 172.31.1.100, 172.31.1.101, and 172.31.1.102.

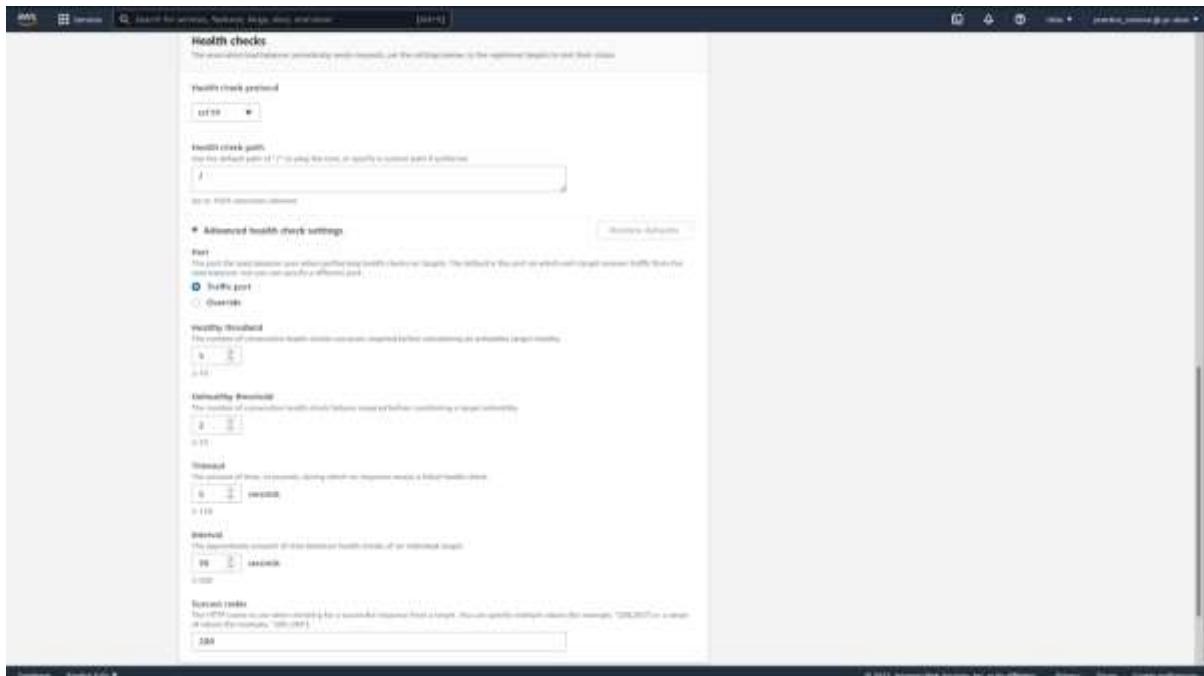
Click **Create target group**.

The screenshot shows the AWS Target Groups page. The sidebar on the left is identical to the previous screenshot, showing the Load Balancing section is selected. The main content area is titled "Target groups [0]" and contains a message: "No target groups to display." Below this, there is a form with fields for "Name", "ARN", "Protocol", "Port", "Description", "VPC ID", and "Tags". At the bottom of the form, there is a large orange button labeled "Create target group". This button is highlighted with a red rectangle.

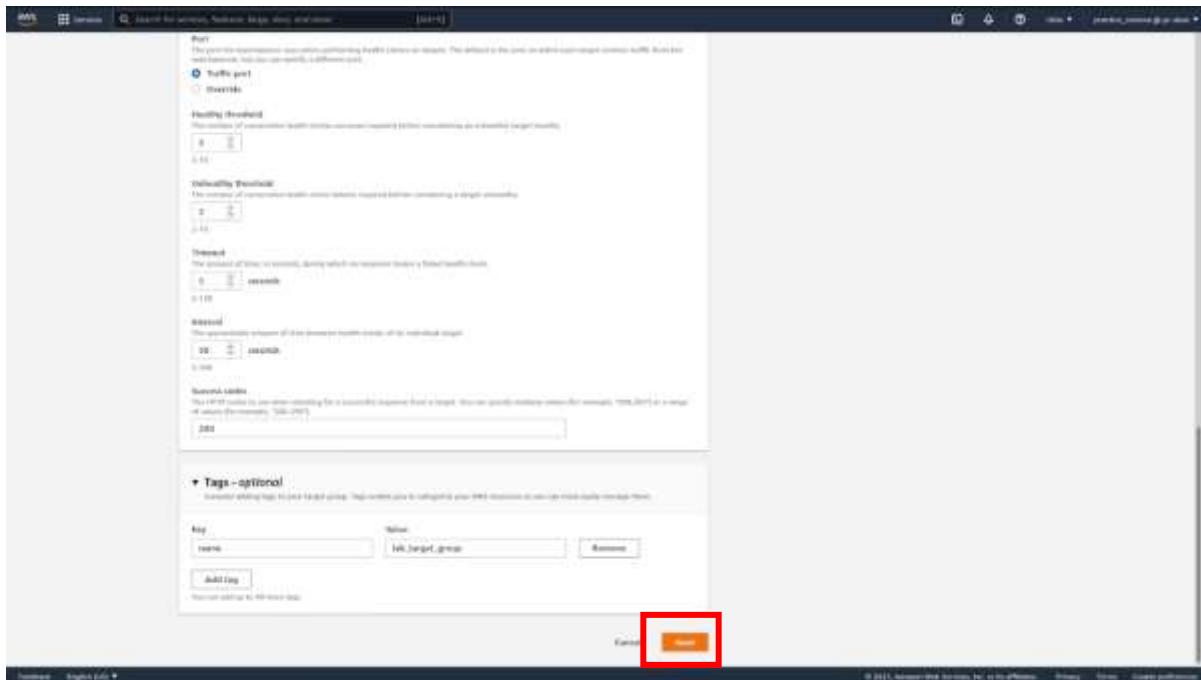
Leave everything at default. Make sure the VPC is the one in which you want to place the load balancer. Give a name to your Target Group.



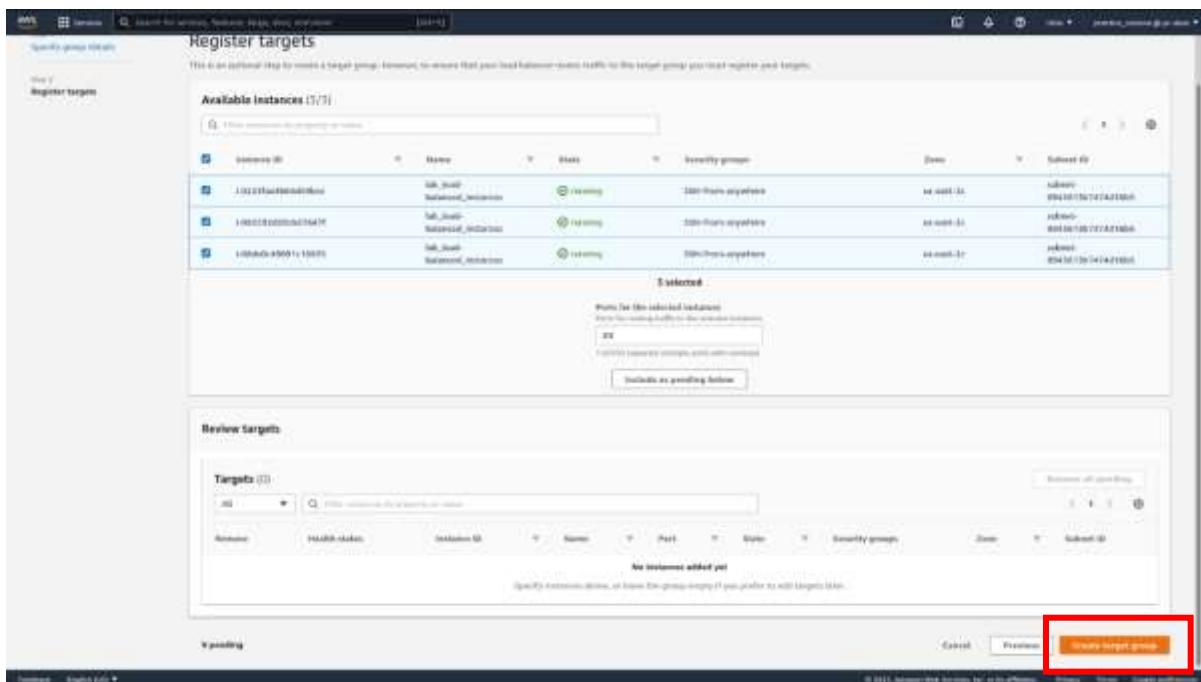
Click on **Advanced health check settings**. Leave everything at default but look at the sections and their values. This will help you understand how the health check system works to keep tabs on the status of your instances.



Add a name tag. Give it a value. Click **Next**.



Select the three instances created before and click **Include as pending** below. Click **Create target group**.



You have successfully created your Target Group to be used by a load balancer.

The screenshot shows the AWS CloudFormation console with a green header bar indicating 'Successfully created target group [MyTargetGroup]'. The main area displays a table titled 'Target groups (1)'. A single row is present in the table:

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
MyTargetGroup	arn:aws:lambda:us-east-1:123456789012:function:MyFunction	80	HTTP	Instance	MyLoadBalancer	vpc-d741c534loadbalancer

The sidebar on the left lists various AWS services, and the bottom navigation bar includes links for 'Create', 'Deploy', 'Edit', and 'CloudWatch Metrics'.

## Creating the Application Load Balancer

Application load balancers provide intelligence based on Application layer logic in http/https protocols.

Select **Load Balancers** from the sidebar, just above Target Groups. Click **Create Load Balancer**.

The screenshot shows the AWS CloudFormation console with a red box highlighting the 'Create Load Balancer' button in the top navigation bar. The main area displays a table titled 'Selected a load balancer' with one entry:

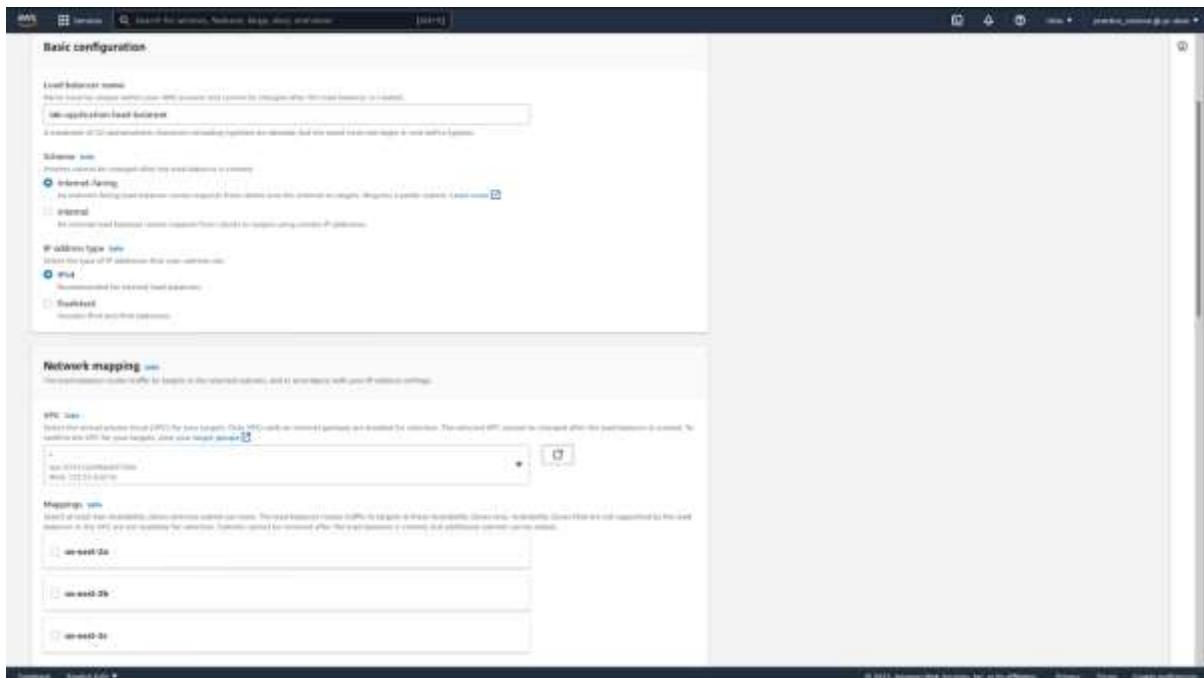
Name	ARN	Port	VPC ID	Availability zones	Type	Created at	Last modified at	Monitoring
MyLoadBalancer	arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/app/MyLoadBalancer/7449132345678901	80	vpc-d741c534loadbalancer	us-east-1a, us-east-1b	Application	2021-07-20T14:45:21Z	2021-07-20T14:45:21Z	Monitoring

The sidebar on the left lists various AWS services, and the bottom navigation bar includes links for 'Create', 'Deploy', 'Edit', and 'CloudWatch Metrics'.

Select **Create** in Application Load Balancer.



Give a name to your application load balancer. Leave the rest of Basic Configuration at their default.



In Network mapping, make sure you have selected the desired VPC. Select two Availability Zones (AZ) of your choice for this VPC. Then select a subnet for each AZ.

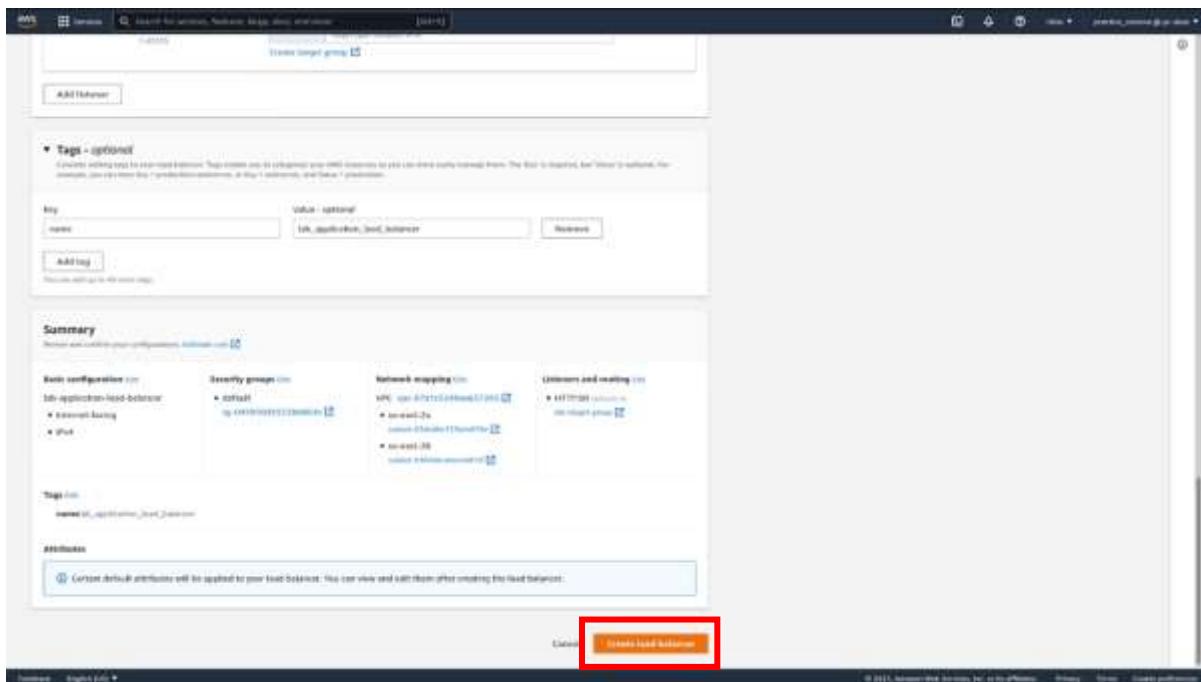
The screenshot shows the AWS Network interface configuration screen. It displays two subnets assigned to different availability zones:

- eni-0000000000000000**: Associated with **us-west-2a**, assigned by **AWS**.
- eni-0000000000000001**: Associated with **us-west-2b**, assigned by **AWS**.

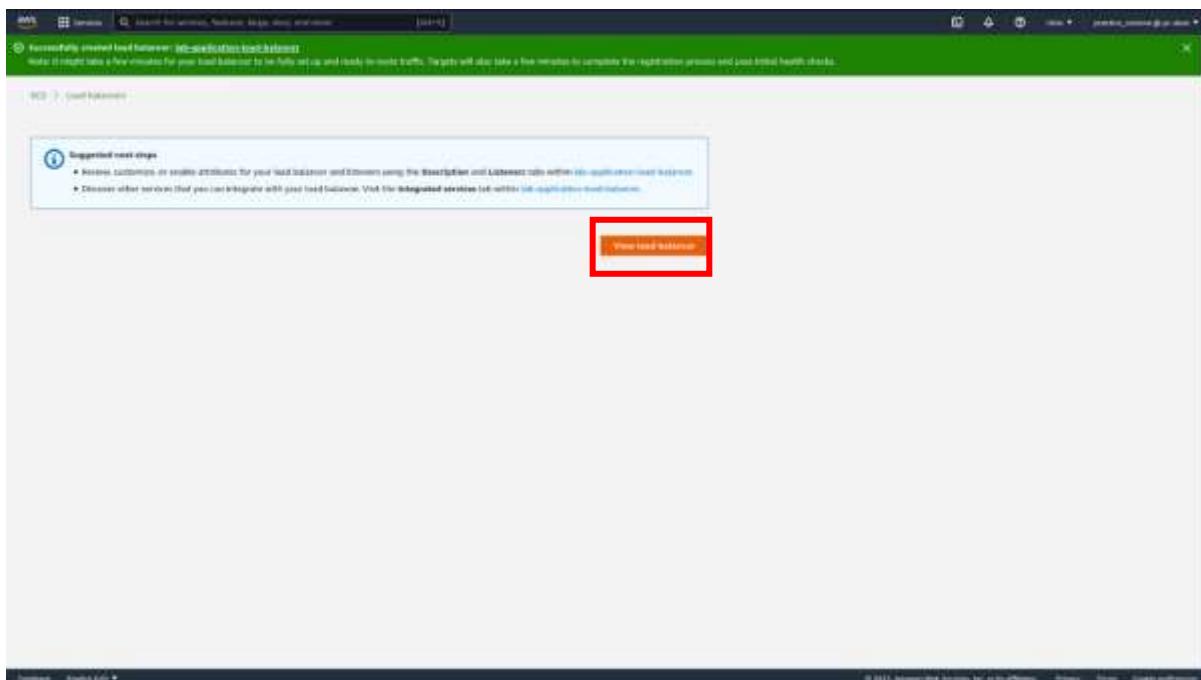
Leave Listener to http at port 80. Select your target group from just before.

The screenshot shows the AWS Listener configuration screen. The **Listener HTTP:80** section is displayed, showing the target group **lambda-target-group** selected for the listener rule.

Add a name tag. Give it a value. Click **Create load balancer**.



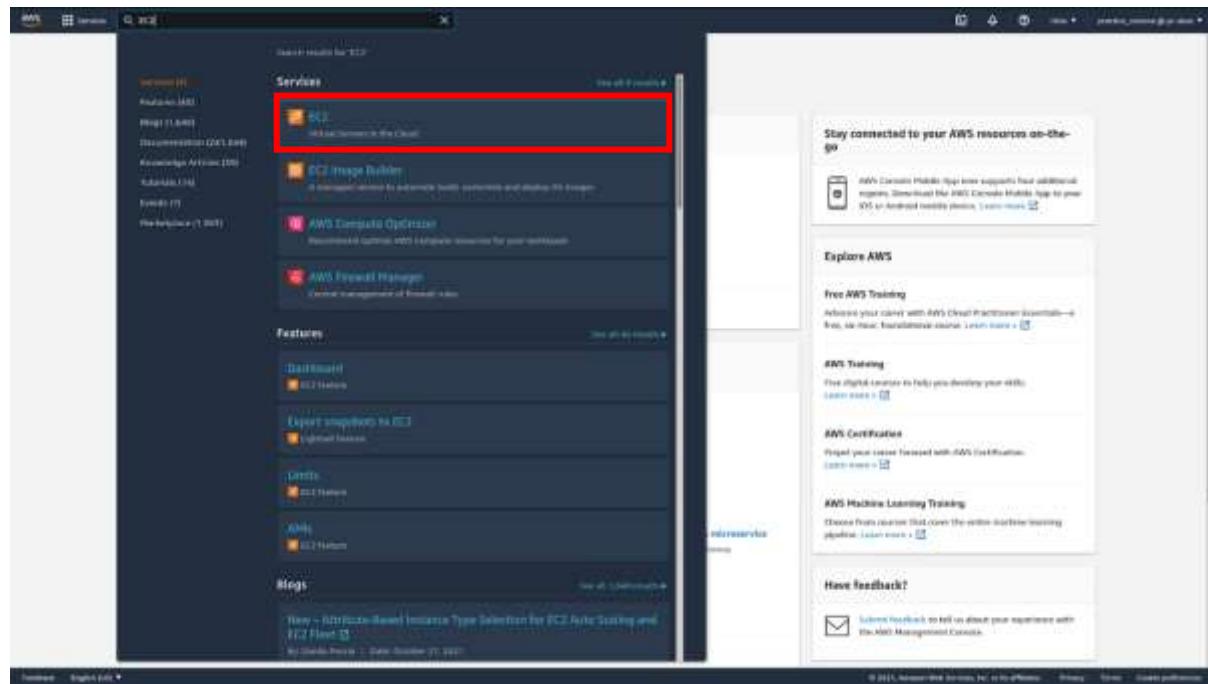
You have successfully created your first application load balancer. Click **View load balancer** to return to the dashboard.



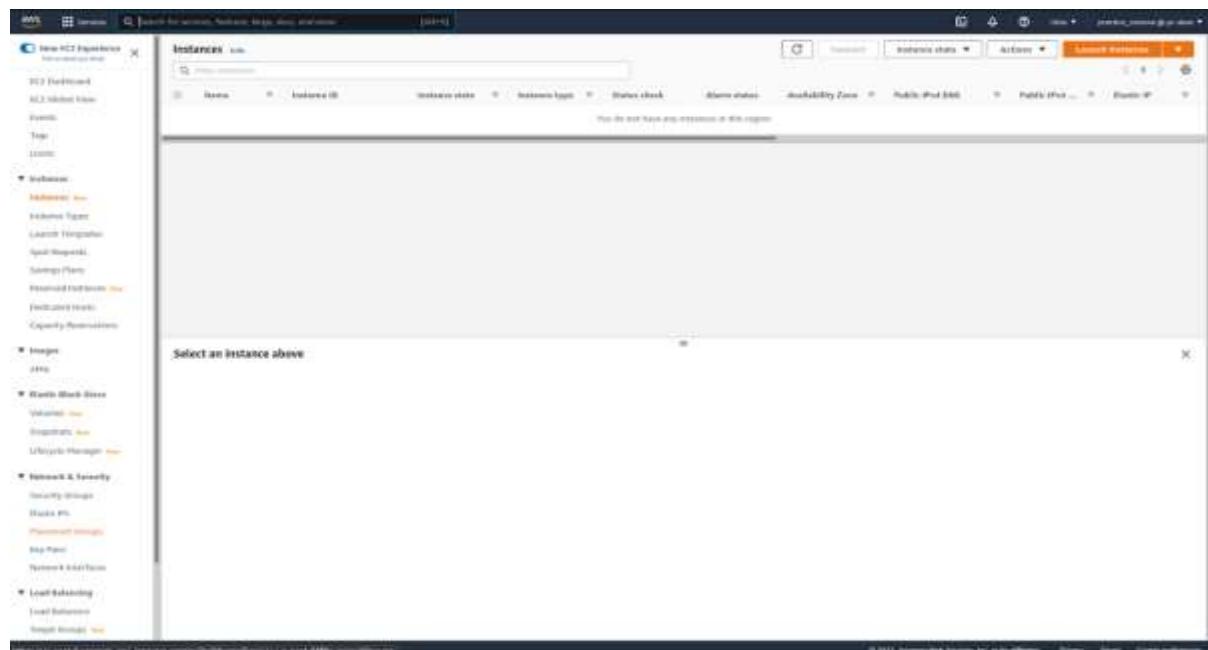
## Lab: Create a Cluster Placement Group

In this lab, we will set up a cluster placement group. Then we will place some virtual machines into the cluster placement group.

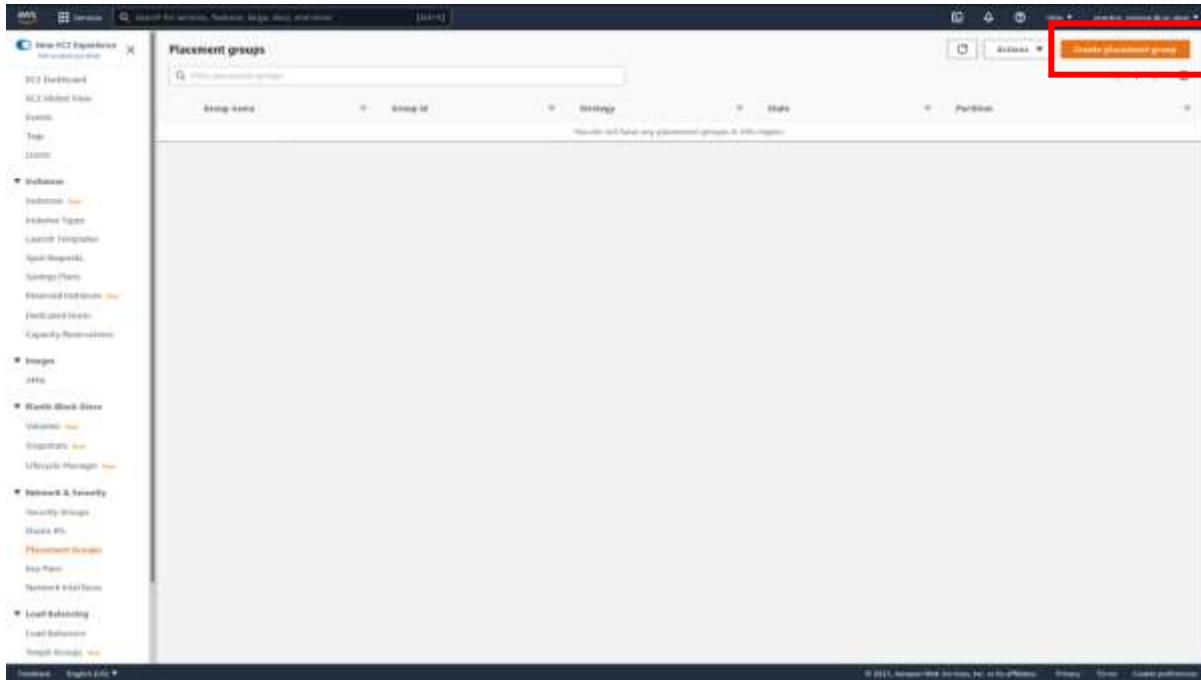
From the management console, start by navigating to the **EC2 console**.



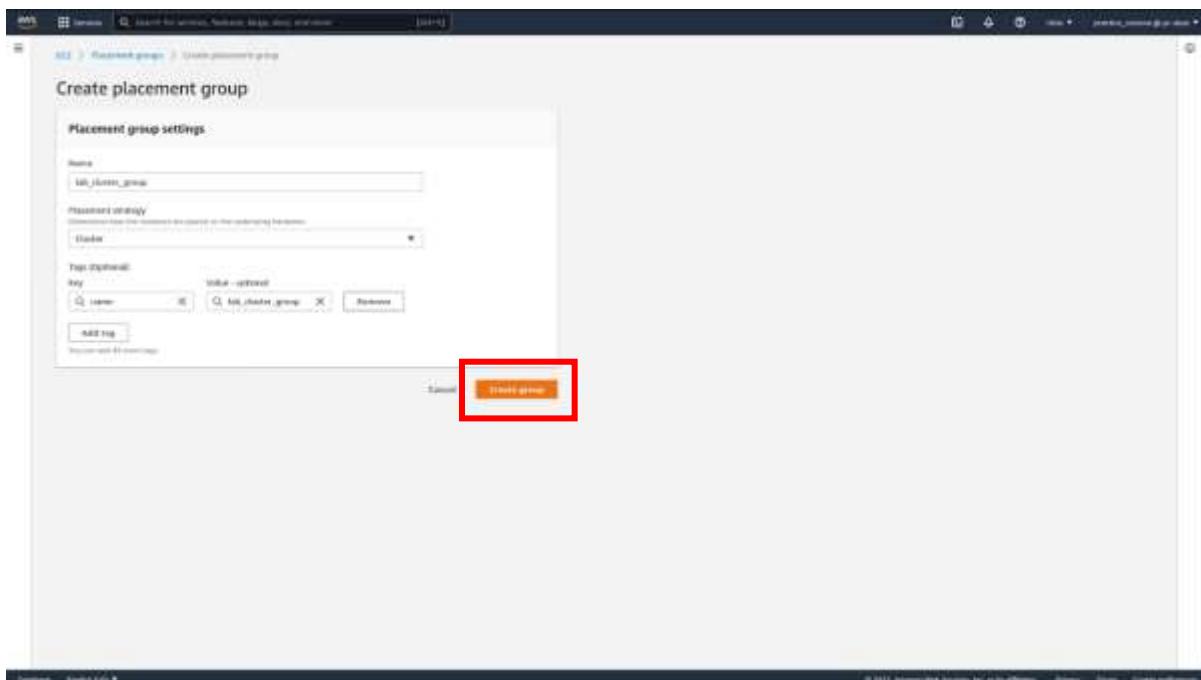
In the sidebar on the left, scroll down and locate **Placement Groups**.



**Click on **Create placement group**.**



Choose a name for your cluster placement group. In **Placement strategy**, select **Cluster**. Add a name tag for your new group and give it a value. Click **Create group**.



And there you go, you have successfully created your first cluster placement group.

The screenshot shows the AWS EC2 Placement Groups console. On the left, a navigation sidebar lists various services like VPC, Lambda, and CloudWatch Metrics. The main area is titled 'Placement groups (1/1)' and contains a table with one row. The table columns are 'Group name', 'Group ID', 'Owner', 'State', and 'Partition'. The single entry is 'lak\_cluster\_group' with Group ID 'pg-0ba40f62121d1734', Owner 'lak', State 'available', and Partition '1'. At the top right, there is a 'Create placement group' button. The bottom of the screen shows standard AWS navigation links for Home, Help, and Support.

## Next, we put the virtual machines into the cluster placement group

Navigate to the **Instances** subsection of the EC2 console.

The screenshot shows the AWS EC2 Instances console. The left sidebar includes 'Instances' under the 'Compute' section. The main area has a search bar and several filter buttons: Status, Instance ID, Instance state, Instance type, Status check, Alarm state, Availability Zone, Public IP (not used), Public IP (used), and Instance IP. A red box highlights the 'Launch instances' button at the top right. Below the filters, a modal dialog box is open with the text 'Select an instance above'. The bottom of the screen shows the usual AWS navigation links.

## Select the Amazon Linux image.

The screenshot shows the AWS Lambda Step 1: Choose an Amazon Machine Image (AMI) page. It lists several AMI options:

- Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-0f71bc3d7100ea7 (39.6 GiB) / ami-094a4277f00002bae (34.6 GiB)
- Amazon Linux 2 (arm64) with AWS Lambda Layer 1.13 bundled custom performance-tier Lambda@Edge, Apache 2.4.42, OC3 1.1, OpenSSL 3.0.2, MySQL 8.0.27, and the latest software packages through updates. This AMI is the successor to the Amazon Linux 2 AMI and is recommended for new users.
- Amazon Linux 2 AMI (HVM) - Kernel 4.14, SSD Volume Type - ami-0edc520201812354-41.vmls (ami-077d860558109) (84.4 GiB)
- Amazon Linux 2 (arm64) with AWS Lambda Layer 1.13 bundled the optimal performance-on-AWS EC2 asymmetric 2.15, OC3 1.1, OpenSSL 3.0.2, MySQL 8.0.27, and the latest software packages through updates. This AMI is the successor to the Amazon Linux 2 AMI and is recommended for new users.
- Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-03a0214ab5d23ec7 (54.6 GiB) / ami-039074030f00028 (54.6 GiB)
- RHEL 8 (ami-03a0214ab5d23ec7) (54.6 GiB) / ami-039074030f00028 (54.6 GiB)
- SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type - ami-03027110a0f0f0f (44.6 GiB) / ami-03027110a0f0f0f (44.6 GiB)
- SLES 15 SP2 (ami-03027110a0f0f0f) (44.6 GiB) / ami-03027110a0f0f0f (44.6 GiB)
- Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-0320230e174c50892 (34.6 GiB) / ami-0347d2877e11114 (34.6 GiB)
- Ubuntu Server 20.04 LTS (arm64) (34.6 GiB) / ami-0347d2877e11114 (34.6 GiB)

A red box highlights the "Launch in database engine RDS" button at the bottom of the page.

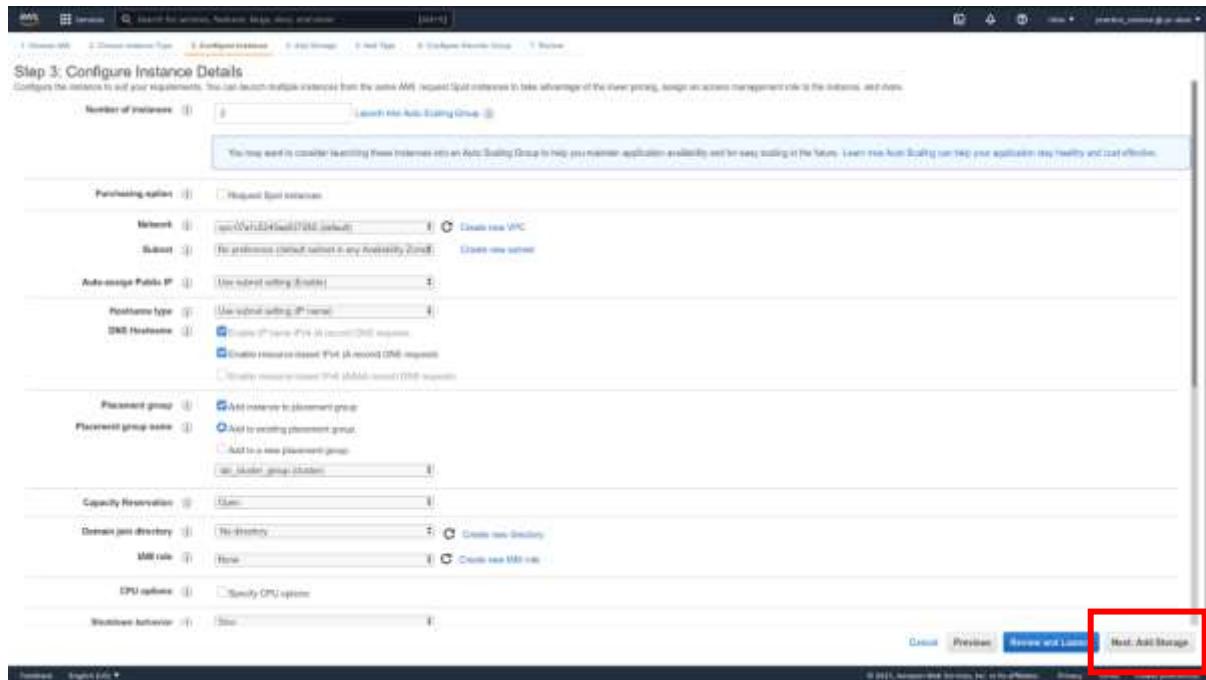
Select the **c5a.large** instance type. This is the least expensive instance type that supports cluster placement groups. You can delete as soon as the lab is over to save on costs. Click **Next: Configure Instance Details**.

The screenshot shows the AWS Lambda Step 2: Choose an Instance Type page. It displays a table of instance types:

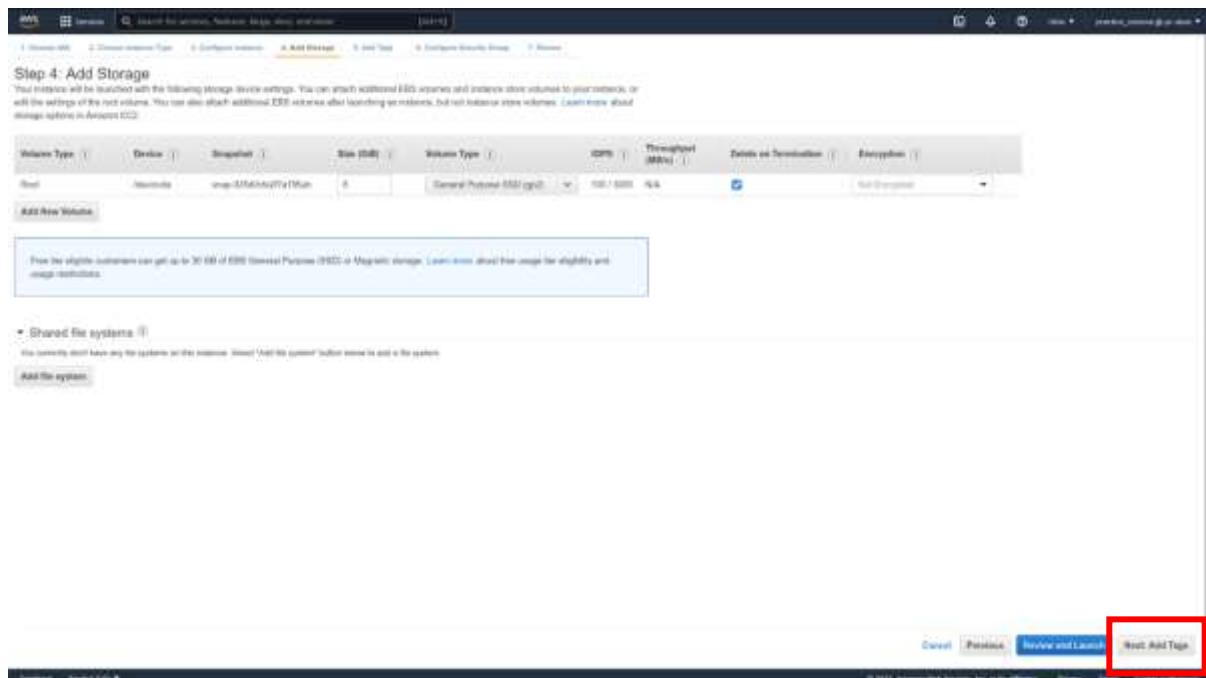
Instance Type	Cores	Threads	Memory (GiB)	Storage (GiB)	Processor	Supports Asymmetric	Max CPU
c5	c5.large	4	8	SSD only	No	Up to 10 Gigabit	Yes
c5	c5.xlarge	8	16	SSD only	No	Up to 10 Gigabit	Yes
c5	c5.2xlarge	16	32	SSD only	No	Up to 10 Gigabit	Yes
c5	c5.4xlarge	32	64	SSD only	No	10 Gigabit	Yes
c5	c5.12xlarge	48	96	SSD only	No	12 Gigabit	Yes
c5	c5.16xlarge	72	144	SSD only	Yes	12 Gigabit	Yes
c5	c5.32xlarge	96	192	SSD only	Yes	25 Gigabit	Yes
c5	c5.64xlarge	96	192	SSD only	Yes	37 Gigabit	Yes
c5a	c5a.large	2	4	SSD only	No	Up to 10 Gigabit	Yes
c5a	c5a.xlarge	4	8	SSD only	No	Up to 10 Gigabit	Yes
c5a	c5a.2xlarge	8	16	SSD only	Yes	Up to 10 Gigabit	Yes
c5a	c5a.4xlarge	16	32	SSD only	Yes	Up to 10 Gigabit	Yes
c5a	c5a.12xlarge	32	64	SSD only	Yes	10 Gigabit	Yes
c5a	c5a.16xlarge	48	96	SSD only	Yes	12 Gigabit	Yes
c5a	c5a.32xlarge	64	128	SSD only	Yes	20 Gigabit	Yes
c5a	c5a.64xlarge	96	192	SSD only	Yes	37 Gigabit	Yes
c5d	c5d.large	2	4	1 x 70 (SSD)	No	Up to 10 Gigabit	Yes
c5d	c5d.xlarge	4	8	1 x 150 (SSD)	No	Up to 10 Gigabit	Yes
c5d	c5d.2xlarge	8	16	1 x 300 (SSD)	No	Up to 10 Gigabit	Yes
c5d	c5d.4xlarge	16	32	1 x 600 (SSD)	Yes	Up to 10 Gigabit	Yes

A red box highlights the "Next: Configure Instance Details" button at the bottom right of the page.

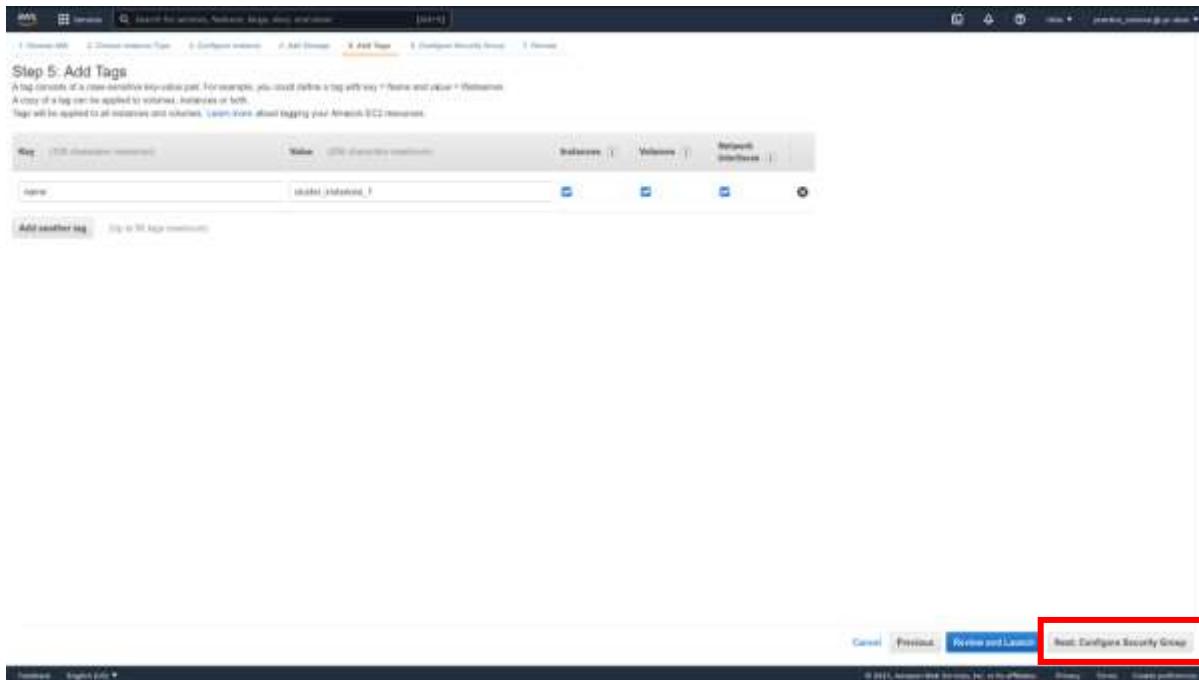
In **Number of instances**, enter 2. In **Placement group**, tick **Add instance to placement group**. In **Placement group name**, select the placement group you created just previously. Click **Next: Add Storage**.



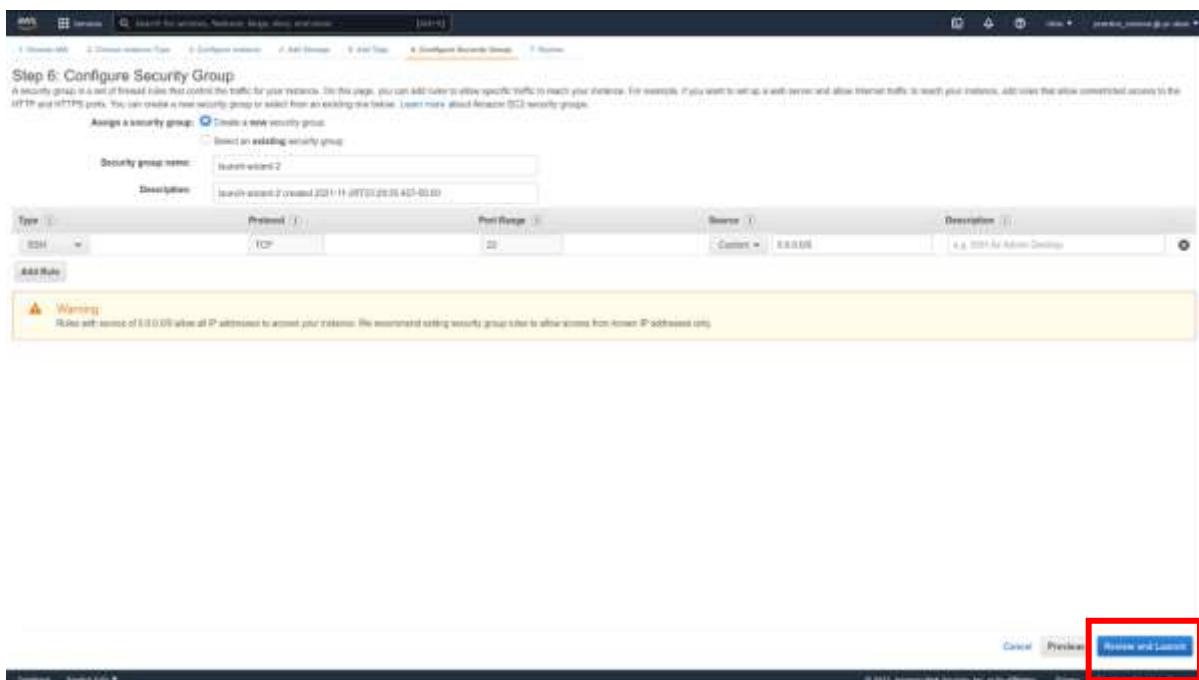
Click **Next: Add Tags**.



Add a name tag. Give it a value. Click **Next: Configure Security Group**.



You can leave the security group to its default or use an existing one if you already have one set up. Click **Review and Launch**.



When you are ready, click **Launch**.

The screenshot shows the 'Step 7: Review Instance Launch' page. At the top, there's a note: 'Please review your instance launch details. You can go back to edit changes by the next section. Click Launch to assign a key pair to your instance and complete the launch process.' Below this, there are two warning boxes:

- A yellow box: 'Your instance configuration is not eligible for the free usage tier. To launch an instance that's eligible for the free usage tier, check your AMI selection, instance type, configuration options, or storage choices. Learn more about free usage tier eligibility and usage limitations.'
- A yellow box: 'Important: Your instance security group, SSH from anywhere, is open to the world. Your instance may be accessible from any IP address. We recommend that you update your security group rules to allow access from specific IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service ports running, e.g., HTTP (80) or web servers. Edit security groups.'

Below these are sections for 'AMI Details', 'Instance Type', 'Security Groups', and 'Instance Details'. In the 'Instance Details' section, the 'Launch' button is highlighted with a red box.

Enter a name for your SSH key. Make sure to download the key. When you have, click **Launch Instances**.

The screenshot shows the 'Step 7: Review Instance Launch' page. A modal dialog titled 'Select an existing key pair or create a new key pair' is open. It contains instructions: 'A key pair consists of a public key that AWS stores, and a private key file that you store. Together, they allow you to connect to your instances securely. For Windows AMIs, this process isn't required to obtain the password needed to log into your instance. For Linux AMIs, this private key file allows you to securely SSH into your instance. Amazon EC2 supports RSA2048 and RSA4096 key pair types.' Below this, there are dropdowns for 'Key pair type' (set to 'RSA') and 'Key pair name' (set to 'key'), and a 'Download Key Pair' button. A note at the bottom says: 'You have to download the private key file (.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created.' The 'Create New Key Pair' button is highlighted with a red box.

## Chapter 8 Labs

### Lab: Create an Identity and Access Management (IAM) User

#### Create an IAM User

From the AWS console, follow these steps to create an IAM user:

Sign in and open the **IAM console** at <https://console.aws.amazon.com/iam/>.

The screenshot shows the IAM dashboard with the following details:

- Security recommendations:**
  - Root user has MFA (Green checkmark)
  - Root user has no active access keys (Green checkmark)
- IAM resources:**
  - User groups: 0
  - Users: 0
  - Policies: 4
  - Identity providers: 0
- What's new:**
  - IAM Access Analyzer helps you generate fine-grained policies that specify the required actions for more than 30 services.
  - IAM Access Analyzer helps you generate IAM policies based on access activity found in your organization trail.
  - IAM Access Analyzer adds new policy checks to help validate conditions during IAM policy authoring.
  - AWS Amplify announces support for IAM permissions boundaries on Amplify-generated IAM roles.
- AWS Account:**
  - Account ID: A16744747936
  - Region: us-east-1
  - Create: B16744747936
  - Sign in URL: https://16744747936.signin.aws.amazon.com/console
- Quick Links:**
  - My security credentials: Manage your access keys, multi-factor authentication (MFA) and other credentials.
- Tools:**
  - Policy simulator: The simulator evaluates the policies that you choose and determines the effective permissions for each of the actions that you specify.
  - Web identity federation playground: Authenticate yourself to any of the supported web identity providers, see the documentation.

Choose **Users** in the navigation pane and select **Add Users**.

The screenshot shows the IAM Users page with the following interface elements:

- Navigation pane:** Shows the IAM service and the "Users" option selected under "Access management".
- Header:** IAM > Users
- Info:** Users (0). Info: An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.
- Buttons:** Delete, Add users (highlighted with a red box).
- Search bar:** Find users by username or access key.
- Table:** Displays columns for User name, Groups, Last activity, MFA, Password age, and Actions.
- Message:** No resources to display.

Enter the user's name in **User name** field, and select the type of access the user should have (Programmatic or AWS Management Console Access). Click **Next: Permissions**.

Add user

Set user details

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

User name\*   

[Add another user](#)

Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Select AWS credential type\*  **Access key - Programmatic access**  
Enables an [access key ID](#) and [secret access key](#) for the AWS API, CLI, SDK, and other development tools.

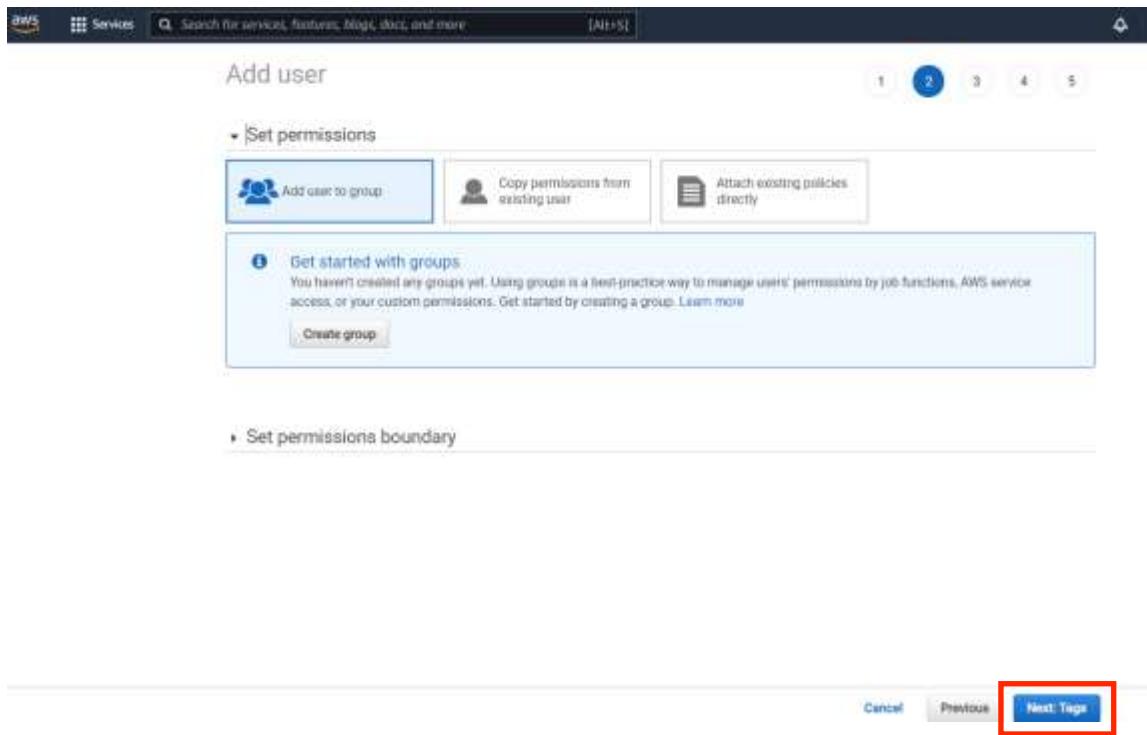
**Password - AWS Management Console access**  
Enables a [password](#) that allows users to sign-in to the AWS Management Console.

Console password\*  Autogenerated password  Custom password  
 Show password

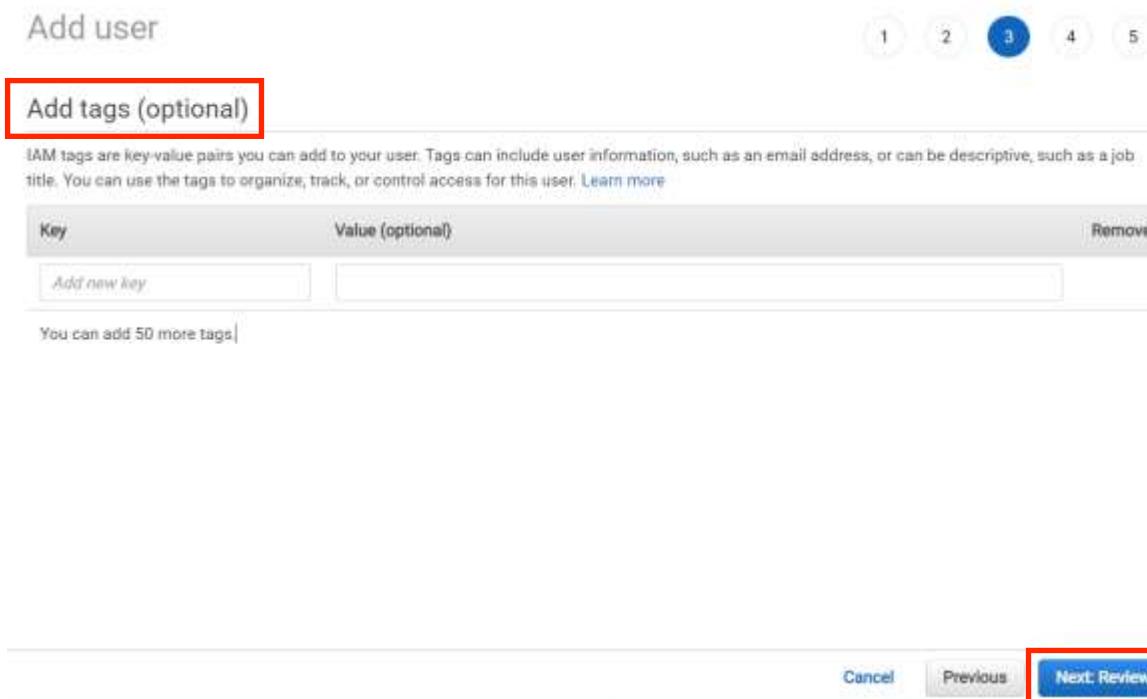
Require password reset:  User must create a new password at next sign-in  
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

\* Required.   [Cancel](#) [Next: Permissions](#)

On the **Set Permissions** page, add user to the target group. Permissions can be copied from an existing user, or the existing policies of the specified group may be attached directly. Click **Next: Tags**.



Add tags, if desired, entering a Key and Value. Click **Next: Review**.



Review and verify the **User details**, then click **Create user**.

Add user

1 2 3 4 5

**Review**

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

**User details**

User name	Gocloudcareers_test_user
AWS access type	Programmatic access and AWS Management Console access
Console password type	Custom
Require password reset	No
Permissions boundary	Permissions boundary is not set

**Permissions summary**

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AdministratorAccess

**Tags**

No tags were added.

Cancel Previous **Create user**

Choose **Show** to view users' **Secret access keys**. Access keys may be downloaded to a .csv file as well. Choose **Download.csv** then save the file to a safe location. **Note** - Access to the secret access keys will **not** be available after this step. Click **Close**.

The screenshot shows the AWS Management Console interface for adding a user. At the top, there's a navigation bar with the AWS logo, a search bar, and a 'Global' dropdown. Below the navigation is a success message box with a green checkmark icon. It states: "Success: You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time." It includes a link to "signin.aws.amazon.com/console". Below the message is a table with user information. The table has columns: User, Access key ID, Secret access key, and Email login instructions. A red box highlights the "Show" link next to the "Secret access key" for the user "Gocloudcareers\_test\_user". There's also a "Download .csv" button and a "Send email" link. At the bottom right of the table area is a "Close" button.

User	Access key ID	Secret access key	Email login instructions
Gocloudcareers_test_user	AKIA34IKOH2	***** Show	Send email

Use newly created User name and password for the new user account in the **Sign-in URL**.

The screenshot shows the AWS Identity and Access Management (IAM) dashboard. On the left, there's a sidebar with navigation links like 'Dashboard', 'Access management', 'Identity providers', and 'Access reports'. The main area displays 'IAM resources' with counts: 0 User groups, 1 Users, 4 Roles, 0 Policies, and 0 Identity providers. Below this, a 'What's new' section lists recent updates, including 'IAM Access Analyzer helps you generate fine-grained policies that specify the required actions for more than 50 services.' A red box highlights the 'Sign-in URL' link under the 'Quick Links' section, which is described as 'IAM users in this account can sign in to the IAM console using their IAM user name and password.' The URL listed is <https://signin.aws.amazon.com/console>.

## LAB: Setting Up Multi-Factor Authentication (MFA) for the AWS Account Root User

In this lab we will set up multifactor authentication for the root user. This is a security best practice.

Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.

Click **Add MFA**.

### IAM dashboard

The screenshot shows the IAM dashboard with the following sections:

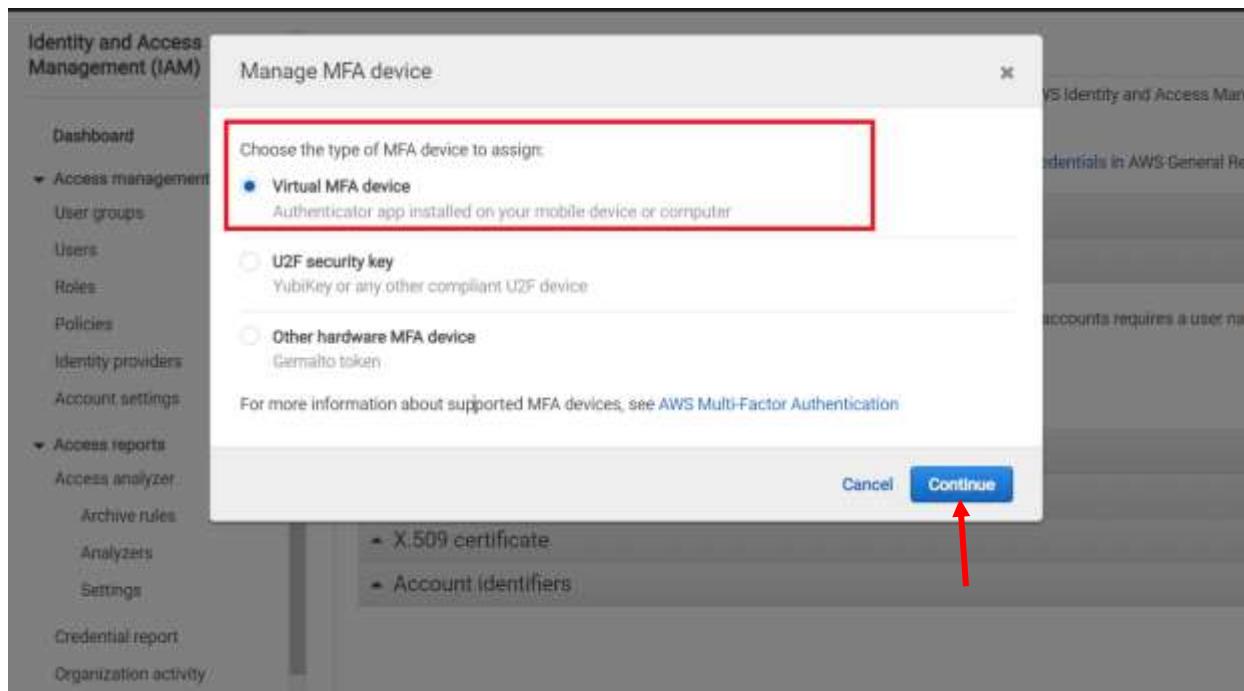
- Security recommendations:** A red box highlights the "Add MFA for root user" item, which is marked with a red exclamation icon and says "Enable multi-factor authentication (MFA) for the root user to improve security for this account." A blue "Add MFA" button is also highlighted with a red box.
- Root user has no active access keys:** This section is marked with a green checkmark and says "Using access keys attached to an IAM user instead of the root user improves security."
- IAM resources:** Shows counts for User groups (0), Users (0), Roles (4), Policies (0), and Identity providers (0).
- What's new:** Shows "Updates for features in IAM" and a "View all" link.

Click **Activate MFA**.

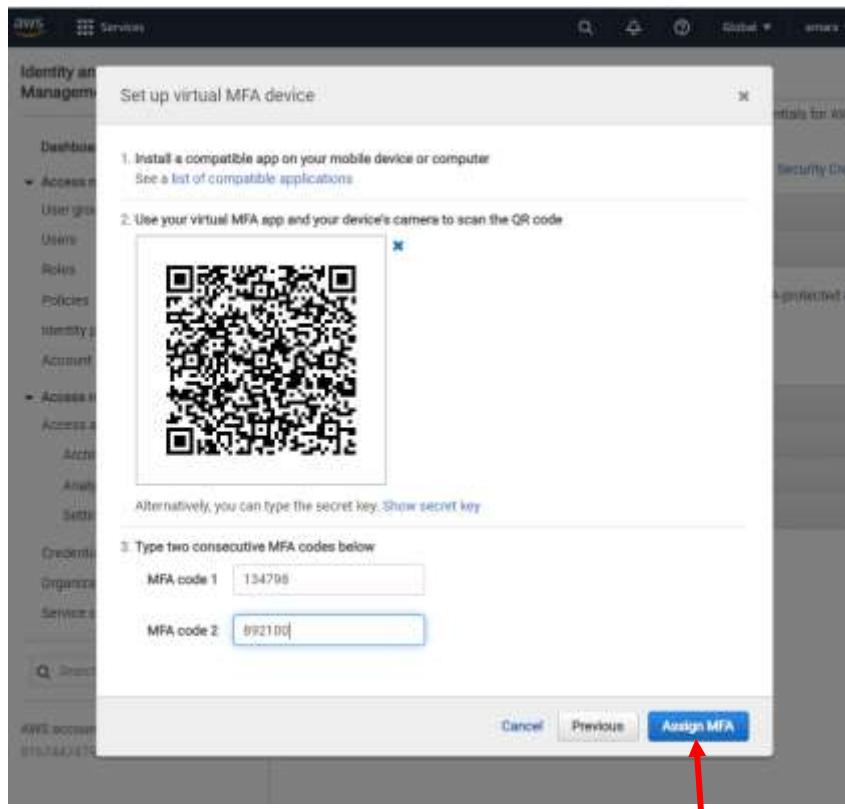
The screenshot shows the "Your Security Credentials" page in the IAM console, with the following structure:

- Identity and Access Management (IAM)** sidebar with links like Dashboard, Access management, Access reports, Credential report, Organization activity, and Service control policies (SCPs).
- Your Security Credentials** main content area:
  - A brief description of managing credentials for the AWS account.
  - A list of credential types:
    - >Password
    - Multi-factor authentication (MFA) **Activate MFA** (button highlighted with a red box)
  - A note about using MFA for account security.
  - A list of other account identifiers:
    - Access keys (access key ID and secret access key)
    - CloudFront key pairs
    - X.509 certificate
    - Account identifiers

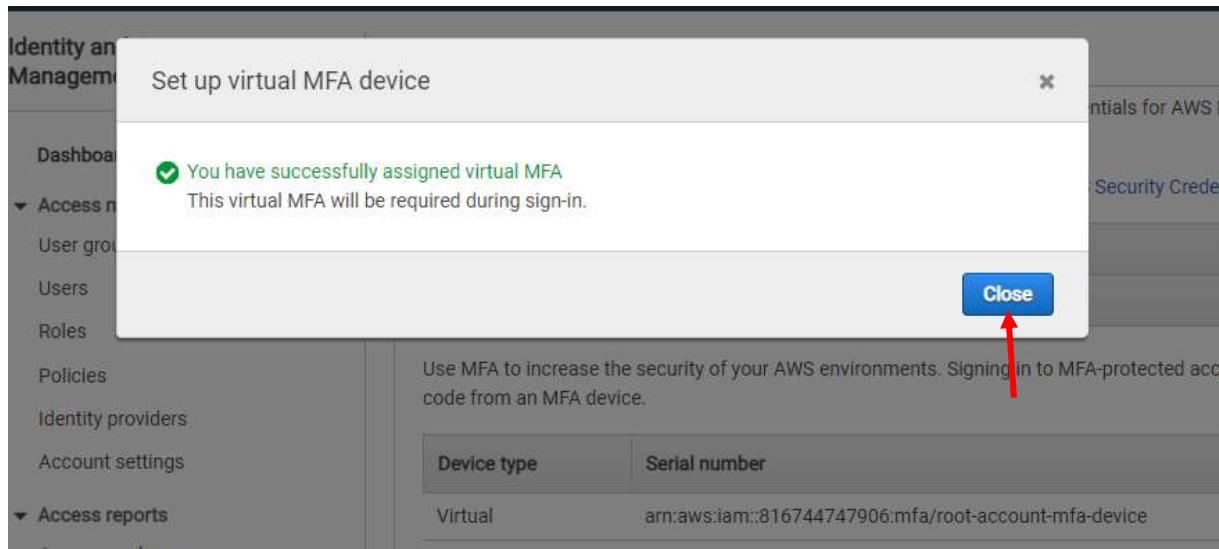
Choose **Virtual MFA devices** and click **Continue**.



Install one of the compatible application Authenticators on your device (E.g.: Google Authenticator), scan the QR code, enter two MFA codes, click **Assign MFA**.



After successfully assigned MFA, click **Close**.



## Lab: Set Up a Static Website and Connect It with CloudFront

In this lab we will set up a static website on S3 and serve that website from the CloudFront content delivery network.

Create an S3 bucket.

The screenshot shows the 'Create bucket' page in the Amazon S3 console. At the top, there is a breadcrumb navigation: 'Amazon S3 > Create bucket'. Below the title 'Create bucket' is a link 'Info' and a note: 'Buckets are containers for data stored in S3. Learn more' with a help icon. The main section is titled 'General configuration'. It includes fields for 'Bucket name' (containing 'movieapp1232'), 'AWS Region' (set to 'US East (N. Virginia) us-east-1'), and a 'Copy settings from existing bucket - optional' section which is currently empty. A note below the copy section states: 'Only the bucket settings in the following configuration are copied.' and a 'Choose bucket' button.

Make the bucket public.

### Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

#### **Block all public access**

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

##### **Block public access to buckets and objects granted through new access control lists (ACLs)**

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

##### **Block public access to buckets and objects granted through any access control lists (ACLs)**

S3 will ignore all ACLs that grant public access to buckets and objects.

##### **Block public access to buckets and objects granted through new public bucket or access point policies**

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

##### **Block public and cross-account access to buckets and objects through any public bucket or access point policies**

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.



**Turning off block all public access might result in this bucket and the objects within becoming public**

AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

## Upload content.

The screenshot shows the Amazon S3 'Upload' interface. At the top, the path 'Amazon S3 > movieapp1232 > Upload' is visible. Below it, the word 'Upload' is prominently displayed with a 'Info' link. A note below the title says: 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more' with a link icon. A large dashed box in the center contains the instruction: 'Drag and drop files and folders you want to upload here, or choose Add files, or Add folder.' Below this is a table titled 'Files and folders (2 Total, 2.5 KB)'. The table lists two files: 'thankyou.html' and 'index.html'. The table includes columns for Name, Folder, Type, and Size. At the bottom of the table are 'Remove', 'Add files', and 'Add folder' buttons. A search bar labeled 'Find by name' is also present. The 'Destination' section is partially visible at the bottom.

<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	thankyou.html	-	text/html	293.0 B
<input type="checkbox"/>	index.html	-	text/html	2.2 KB

**Destination**

Create bucket policy.

AWS Policy Generator

The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see key concepts in Using AWS Identity and Access Management. Here are sample policies.

**Step 1: Select Policy Type**

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.

Select Type of Policy  S3 Bucket Policy

**Step 2: Add Statement(s)**

A statement is the formal description of a single permission. See a description of elements that you can use in statements.

Effect  Allow  Deny

Principal \*

AWS Service  Amazon S3  All Services (\*)

Actions  Action(s) Selected  All Actions (\*)

Amazon Resource Name (ARN) am aws s3: movieapp1232

Amazon Resource Name (ARN)

Policy JSON Document

Click below to edit. To save the policy, copy the text below to a text editor.  
Changes made below will **not be reflected in the policy generator tool**.

```
{ "Id": "Policy1629652981458", "Version": "2012-10-17", "Statement": [ { "Sid": "Stmt1629652969074", "Action": [ "s3:GetObject" ], "Effect": "Allow", "Resource": "arn:aws:s3:::movieapp1232/*", "Principal": "*" } ] }
```

You added the following statement(s):

**Principal**: \*

**Step 3: View Generated Policy**

A policy is a container for permissions.

This AWS Policy Generator is provided for informational purposes only; you are still responsible for your use of Amazon Web Services technologies and ensuring that your use is in compliance with all applicable terms and conditions. This AWS Policy Generator is provided as-is without warranty of any kind, whether express, implied, or statutory. This AWS Policy Generator does not modify the applicable terms and conditions associated with use of Amazon Web Services.

**Close**

## Enable static website hosting.



This screenshot shows the same configuration panel as the previous one, but with different settings. The 'Static website hosting' status is now 'Enabled'. Under 'Hosting type', the option 'Host a static website' is selected, with a note: 'Use the bucket endpoint as the web address. [Learn more](#)'. The other option, 'Redirect requests for an object', is also listed with its own note: 'Redirect requests to another bucket or domain. [Learn more](#)'. A callout box contains an info icon and text: 'For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#)'.

**Index document**  
Specify the home or default page of the website.

**Error document - optional**  
This is returned when an error occurs.

## Verify website.

A screenshot of a web browser window showing a user registration form. The URL in the address bar is <http://movieapp1232.s3-website-us-east-1.amazonaws.com>. The page contains fields for Firstname (babita), Lastname (sinha), Gender (Male selected), Phone (121460), Email (babita.sinha@gmail.com), Password (empty), Re-type password (empty), Movie Genres (Action selected), and Plans (Monthly Plan selected). A note at the bottom states: "By creating an account you agree to our [Terms & Privacy](#)". At the bottom left are "Submit" and "Reset" buttons.

Firstname: babita

Lastname: sinha

Gender:

- Male
- Female
- Other

Phone : 121460

Email: babita.sinha@gmail.com

Password: [REDACTED]

Re-type password: [REDACTED]

Movie Genres : Action

Plans : Monthly Plan

By creating an account you agree to our [Terms & Privacy](#).

Submit Reset



Add the CloudFront distribution.

The screenshot shows the AWS CloudFront Distribution Details page for a distribution named E19WPUIW3AJBH2. At the top, a green banner indicates "Successfully created new distribution." The main interface includes tabs for General, Origins, Behaviors, Error pages, Geographic restrictions, Invalidations, and Tags. The General tab is selected. Under the Details section, the Distribution domain name is listed as d1rcpg0qkhalqu.cloudfront.net, with an ARN of arn:aws:cloudfront::distribution/E19WPUIW3AJBH2. The Last modified timestamp is shown as 2024-02-22T16:00:00Z. In the Settings section, there are fields for Description (empty), Alternate domain names (empty), Standard logging (Off), and Cookie logging (disabled). An "Edit" button is located in the top right corner of the Settings section.

The screenshot shows the "Create distribution" wizard, Step 1: Origin. The title is "Create distribution". The first section is "Origin", titled "Origin domain". It asks to choose an AWS origin or enter the origin's domain name. A search bar contains the value "movieapp1232.s3.us-east-1.amazonaws.com". Below it is an "Origin path - optional" field with the placeholder "Enter the origin path". The next section is "Name", asking for a name for this origin, with the value "movieapp1232.s3.us-east-1.amazonaws.com" entered into the field. The final section is "S3 bucket access", with a note about using a CloudFront origin access identity (OAI) to access the S3 bucket. Two radio button options are shown: "Don't use OAI (bucket must allow public access)" (selected) and "Yes use OAI (bucket can restrict access to only CloudFront)".

Verify website after CloudFront.

The screenshot shows a web browser window with a light blue background. The URL in the address bar is <https://d1cp9gkfxdoqu.cloudfront.net/index.html>. The page contains a user registration form with the following fields:

- Firstname: [Text Input Field]
- Lastname: [Text Input Field]
- Gender :
  - Male
  - Female
  - Other
- Phone : +91 [Text Input Field] [Text Input Field]
- Email: [Text Input Field]
- Password: [Text Input Field]
- Re-type password: [Text Input Field]
- Movie Genres : [Movie Genres ▾](#)
- Plans : [Plans ▾](#)
- By creating an account you agree to our [Terms & Privacy](#).
- Buttons: Submit [Text Input Field], Reset [Text Input Field]

## Lab: Setting Up WAF, Shield, and CloudFront

In this lab we will set up CloudFront, WAF, and Shield.

Starting from the management console, navigate to load balancers and set up a target group with an EC2-based web server.



Create a Target Group with your web server EC2 instance as a registered target. EC2 instances can only work with CloudFront when a load balancer is used.

A screenshot of the AWS Management Console showing the 'Target Groups' page. A target group named 'WebTestTarget' is selected. The 'Details' tab is active, showing settings like 'Protocol: HTTP' and 'Port: 80'. The 'Health' tab shows 3 healthy targets. The 'Targets' tab lists one target: 'ip-172-31-10-10.us-west-2.compute.internal'. The 'Monitoring' tab is also visible.

Set up CloudFront Distribution.

From the management console, navigate to CloudFront. Then create a CloudFront distribution.

A screenshot of the AWS Management Console showing the 'Amazon CloudFront' landing page under 'Networking &amp; Content Delivery'. The page features a large heading 'Amazon CloudFront' and subtext 'Securely deliver content with low latency and high transfer speeds'. A paragraph at the bottom explains what CloudFront is. On the right, there's a 'Get started with CloudFront' section and a large orange button with white text that says 'Create a CloudFront distribution'. This button is highlighted with a red rectangular box.

Select the S3 distribution you created. For this test, we will not enable Origin Shield.

CloudFront > Distributions > create

## Create distribution

### Origin

**Origin domain**  
Choose an Amazon Web Services origin, or enter your origin's domain name.

X

**Origin path - optional** [Info](#)  
Enter a URL path to append to the origin domain name for origin requests.

**Name**  
Enter a name for this origin.

**S3 bucket access** [Info](#)  
Use a CloudFront origin access identity (OAI) to access the S3 bucket.

Don't use OAI (bucket must allow public access)  
 Yes use OAI (bucket can restrict access to only CloudFront)

**Add custom header - optional**  
CloudFront includes this header in all requests that it sends to your origin.

**Enable Origin Shield** [Info](#)  
Origin Shield is an additional caching layer that can help reduce the load on your origin and help protect its availability.

No  
 Yes

► Additional settings

For this lab, we are only interested in the http and https protocols.

For this lab, we will not restrict viewer access.

**Default cache behavior**

Path pattern [Info](#)

Default (\*)

Compress objects automatically [Info](#)

No  
 Yes

**Viewer**

Viewer protocol policy

HTTP and HTTPS  
 Redirect HTTP to HTTPS  
 HTTPS only

Allowed HTTP methods

GET, HEAD  
 GET, HEAD, OPTIONS  
 GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

Restrict viewer access

If you restrict viewer access, viewers must use CloudFront signed URLs or signed cookies to access your content.

No  
 Yes

We have kept most of the default options on the next few screens, which are geared toward optimizing performance.

Click Create Distribution.

## Cache key and origin requests

We recommend using a cache policy and origin request policy to control the cache key and origin requests.

### Cache policy and origin request policy (recommended)

#### Legacy cache settings

##### Cache policy

Choose an existing cache policy or create a new one.

CachingOptimized

Default policy when CF compression is enabled

Recommended for S3 origins ▾



[Create policy](#) [View policy](#)

##### Origin request policy - optional

Choose an existing origin request policy or create a new one.

Select origin policy



[Create policy](#)

##### Response headers policy - optional

Choose an existing response headers policy or create a new one.

Select response headers



[Create policy](#)

### ▼ Additional settings

#### Smooth streaming

Choose No if your origin is configured to use Microsoft IIS for Smooth Streaming.

##### No

##### Yes

#### Field-level encryption [Info](#)

Choose a field-level encryption configuration.

Select a field-level encryption profile



#### Enable real-time logs [Info](#)

##### No

##### Yes

### Function associations - *optional* [Info](#)

Choose an edge function to associate with this cache behavior, and the CloudFront event that invokes the function.

	Function type	Function ARN / Name	Include body
Viewer request	No association ▾		
Viewer response	No association ▾		
Origin request	No association ▾		
Origin response	No association ▾		

## Settings

### Price class [Info](#)

Choose the price class associated with the maximum price that you want to pay.

- Use all edge locations (best performance)
- Use only North America and Europe
- Use North America, Europe, Asia, Middle East, and Africa

### Amazon WAF web ACL - *optional*

Choose the web ACL in Amazon WAF to associate with this distribution.

Choose web ACL ▾

### Alternate domain name (CNAME) - *optional*

Add the custom domain names that you use in URLs for the files served by this distribution.

Add item

[ⓘ To add a list of alternative domain names, use the bulk editor.](#)

### Custom SSL certificate - *optional*

Associate a certificate from Amazon Certificate Manager. The certificate must be in the US East (N. Virginia) Region (us-east-1).

Choose certificate ▾



Request certificate [↗](#)

### Supported HTTP versions

Add support for additional HTTP versions. HTTP/1.0 and HTTP/1.1 are supported by default.

- HTTP/2

### Default root object - *optional*

The object (file name) to return when a viewer requests the root URL (/) instead of a specific object.

### Standard logging

Get logs of viewer requests delivered to an Amazon S3 bucket.

- Off
- On

### IPv6

- Off
- On

### Description - *optional*

Cancel

Create distribution

## Lab: Create a Web Access Firewall (WAF)

In this lab, we have selected CloudFront distribution as the resource type..

The associated resource (CloudFront Distribution Name) can be added in this section, or you can add it later after the WAF has been created.

Click next:

AWS WAF > Web ACLs > Create web ACL

Step 1  
Describe web ACL and associate it to AWS resources [Info](#)

Step 2  
Add rules and rule groups

Step 3  
Set rule priority

Step 4  
Configure metrics

Step 5  
Review and create web ACL

### Describe web ACL and associate it to AWS resources

**Web ACL details**

Name: TestWAF  
The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and \_ (underscore).

Description - optional: Test WAF for Demo  
The description can have 1-256 characters.

CloudWatch metric name: TestWAF  
The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and \_ (underscore).

Resource type:  
 CloudFront distributions  
 Regional resources (Application Load Balancer, API Gateway, AWS AppSync)

Region:  
Choose the AWS region to create this web ACL in.  
Global (CloudFront)

**Associated AWS resources - optional**

Remove Add AWS resources

Find associated AWS resources

	Name	Resource type	Region
No results			
There are no results to display			

Cancel **Next**

In this lab, we have opted not to add any custom rules and will use the defaults.

Click next:

The screenshot shows the AWS WAF 'Create web ACL' wizard at Step 2: Add rules and rule groups. The left sidebar lists steps 1 through 5. Step 1 is 'Describe web ACL and associate it to AWS resources'. Step 2 is 'Add rules and rule groups', which is currently selected and highlighted in blue. Step 3 is 'Set rule priority', Step 4 is 'Configure metrics', and Step 5 is 'Review and create web ACL'. The main content area is titled 'Add rules and rule groups' with an 'Info' link. It contains a table for managing rules. The table has columns for 'Name' and 'Action'. There are three buttons at the top of the table: 'Edit', 'Delete', and 'Add rules ▲'. A tooltip for 'Add rules' says: 'A rule defines attack patterns to look for in web requests and the action to take when a request matches the patterns. Rule groups are reusable collections of rules. You can use managed rule groups offered by AWS and AWS Marketplace sellers. You can also write your own rules and use your own rule groups.' Below the table, a message says 'No rules.' and 'You don't have any rules added.' At the bottom of the page, there is a section titled 'Default web ACL action for requests that don't match any rules' with options for 'Allow' (selected) and 'Block'. A note says '▶ Custom request - optional'. At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons, with 'Next' being highlighted with a red box.

AWS WAF > Web ACLs > Create web ACL

Step 1  
Describe web ACL and associate it to AWS resources

Step 2  
Add rules and rule groups

Step 3  
**Set rule priority**

Step 4  
Configure metrics

Step 5

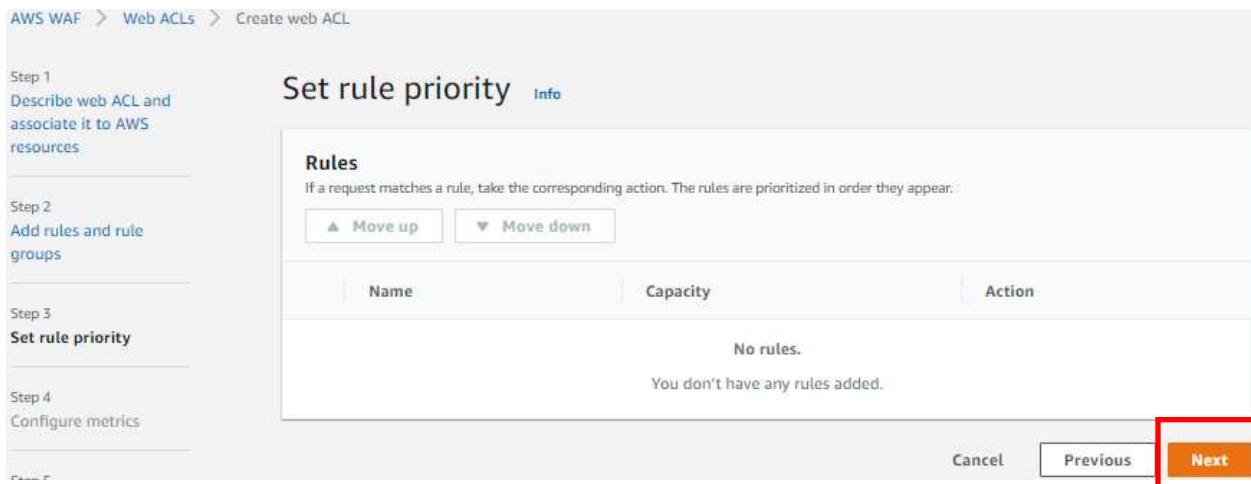
## Set rule priority Info

**Rules**  
If a request matches a rule, take the corresponding action. The rules are prioritized in order they appear.

▲ Move up ▼ Move down

Name	Capacity	Action
No rules. You don't have any rules added.		

Cancel Previous Next



AWS WAF > Web ACLs > Create web ACL

Step 1  
Describe web ACL and associate it to AWS resources

Step 2  
Add rules and rule groups

Step 3  
Set rule priority

Step 4  
**Configure metrics**

Step 5  
Review and create web ACL

## Configure metrics Info

**Amazon CloudWatch metrics**  
CloudWatch metrics allow you to monitor web requests, web ACLs, and rules.

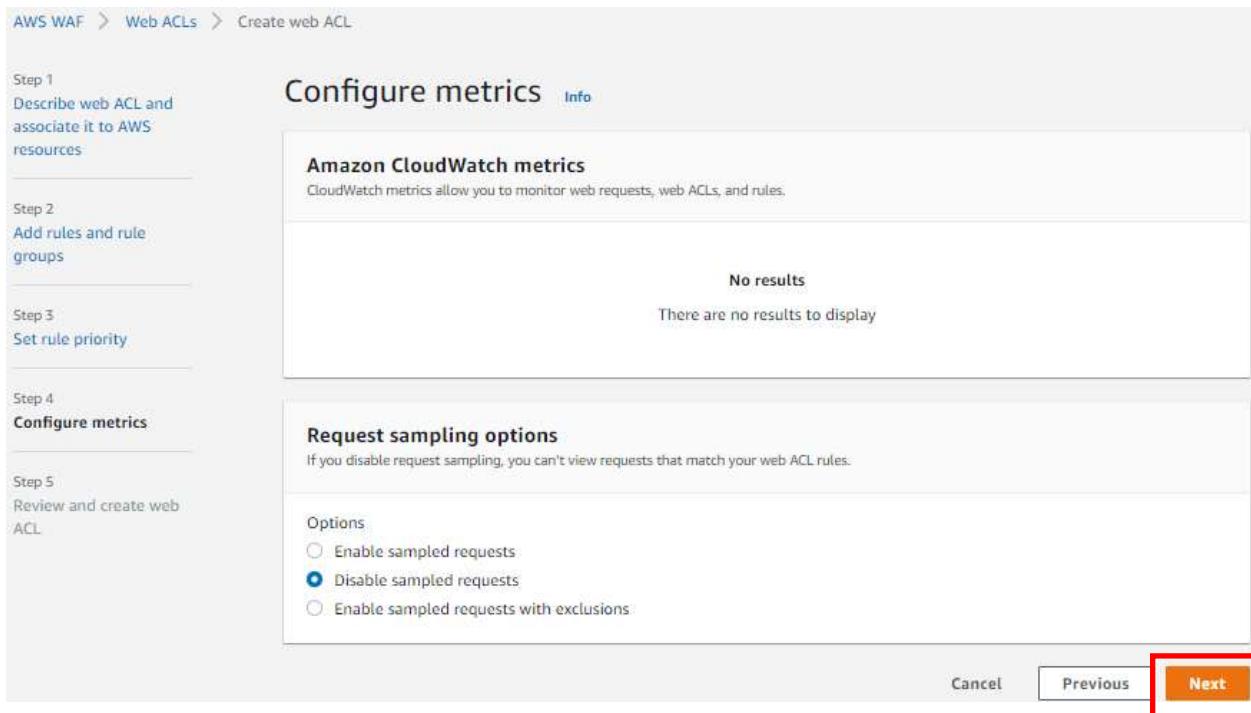
No results  
There are no results to display

**Request sampling options**  
If you disable request sampling, you can't view requests that match your web ACL rules.

**Options**

Enable sampled requests  
 Disable sampled requests  
 Enable sampled requests with exclusions

Cancel Previous Next



Review all the options selected and click on “Create Web ACL”.

AWS WAF > Web ACLs > Create web ACL

Step 1: Describe web ACL and associate it to AWS resources [Edit](#)

Step 2: Add rules and rule groups

Step 3: Set rule priority

Step 4: Configure metrics

Step 5: Review and create web ACL

### Review and create web ACL [Info](#)

#### Web ACL details

Name:	TestWAF	Scope:	CLOUDFRONT
Description:	Test WAF for Demo.	Region:	global
CloudWatch metric name:	TestWAF		

#### Steps 2 and 3: Add rules and set rule priority [Edit](#)

##### Rules

If a request matches a rule, take the corresponding action. The rules are prioritized in order they appear.

Name	Capacity	Action
No rules.		You don't have any rules added.

Web ACL rule capacity units used  
The total capacity units used by the web ACL can't exceed 1500.  
0/1500 WCU

#### Default web ACL action for requests that don't match any rules

Action	Custom request headers
Allow	-

#### Step 4: Configure metrics [Edit](#)

##### Amazon CloudWatch metrics

Rules	CloudWatch metric name
No results	There are no results to display

#### Sampled requests

Sampled requests:	Disabled	Sampled requests for web ACL default actions:	Enabled
-------------------	----------	-----------------------------------------------	---------

[Cancel](#) [Previous](#) [Create web ACL](#)

Once the ACL has been created, it can be associated with the CloudFront distribution.

The screenshot shows the AWS WAF Web ACLs list page. At the top, there's a search bar labeled 'Find web ACLs'. Below it, a table lists one item: 'TestWAF' with a description 'Test WAF for Demo'. To the right of the table, there are two buttons: 'Delete' and 'Edit'. At the bottom right of the page, there's a link 'Global (CloudFront)'.

For all CloudFront Distributions, AWS Shield will be automatically enabled.

The screenshot shows the AWS WAF Associated AWS resources page for a CloudFront distribution named 'CloudFront Distribution - https://cloudfront-test.s3.amazonaws.com/'. The 'Associated AWS resources' section has a search bar and a table with one row. A large red box highlights the 'Add AWS resources' button at the bottom right of the page.

## AWS Shield

As an AWS customer, you automatically have basic DDoS protection with the AWS Shield Standard plan, at no additional cost beyond what you already pay for AWS WAF and your other AWS services. For an additional cost, you can get advanced DDoS protection by activating the AWS Shield Advanced plan. The following table shows a comparison of the two plans.

Features	AWS Shield Standard	AWS Shield Advanced
<strong>Active monitoring</strong>		
Network flow monitoring	✓	✓
Automated application (layer 7) traffic monitoring	-	✓
<strong>DDoS mitigations</strong>		
Helps protect from common DDoS attacks, such as SYN floods and UDP reflection attacks	✓	✓
Access to additional DDoS mitigation capacity	-	✓
<strong>Visibility and reporting</strong>		
Layer 3/4 attack notification and attack forensic reports	-	✓
Layer 3/4/7 attack historical report	-	✓
<strong>DDoS response team support</strong>		
Incident management during high severity events	-	✓
Custom mitigations during attacks	-	✓
Post-attack analysis	-	✓
<strong>Cost protection</strong>		
Reimburse related Route 53, CloudFront, and ELB DDoS charges	-	✓
<strong>Status</strong>	Activated	Not activated
<strong>Price</strong>	No additional cost for all AWS customers	\$3,000/month plus additional data transfer fees AWS WAF included at no additional cost <a href="#">Learn more</a>
		<a href="#">Activate AWS Shield Advanced</a>

## Lab: Creating an AWS Organization with AWS Management Console

In this lab we will be setting up AWS organizations. From the AWS Management Console, click on your username (upper right) to access a drop-down menu.

The screenshot shows the AWS Management Console homepage. At the top, there's a search bar and a navigation bar with links for Services, Help, and Contact. On the left, there's a sidebar titled "AWS services" with sections for "Recently visited services" (AWS Organizations, S3, IAM) and "All services". Below this is a "Build a solution" section with three options: "Launch a virtual machine" (With EC2, 2-3 minutes), "Build a web app" (With Elastic Beanstalk, 6 minutes), and "Build using virtual servers" (With Lightsail, 1-2 minutes). To the right, there are two boxes: "Stay connected to your AWS resources on-the-go" (describing the AWS Console Mobile App) and "Explore AWS" (with sections for "Free AWS Training" and "AWS Cloud Training"). In the top right corner, there's a user profile icon with the text "Account ID: 1819-6144-6392" and a "Sign out" button.

Select Organization in the drop-down menu.

This screenshot is identical to the one above, showing the AWS Management Console homepage. However, the "Organization" option in the user profile dropdown menu has been highlighted with a red box. The rest of the interface is the same, including the sidebar, build solutions, and explore sections.

Click Create an organization:

The screenshot shows the AWS Organizations console. At the top, there's a search bar and a navigation bar with 'AWS Organizations' and other service links. The main title is 'AWS Organizations' with the subtitle 'Central management for multiple AWS accounts'. Below this, a call-to-action box says 'Create an organization to manage multiple AWS accounts. Learn more' with a 'Create an organization' button. To the right, there's a 'Pricing' section stating 'There is no cost to use AWS Organizations.' and a 'Getting started' section with links to 'Getting started with AWS Organizations' and 'Documentation'.

Creating an organization is a simple process:

1. Create the organization.
2. Create and add a new account.
3. Group accounts.
4. Apply policies.
5. Enable services.



An email will be sent to the user to verify that you have created an organization. Once you verify, you will be able to add accounts.

The screenshot shows the AWS Organizations console. On the left, there's a sidebar with 'AWS Organizations' selected, showing options like 'Invitations', 'Services', 'Policies', 'Settings', and 'Get started'. Below that is 'Organization ID: 111111111111'. Under 'AWS accounts', it says 'You successfully created an AWS organization.' and 'Check your email to finish verifying your management account.' A message states: 'We sent a verification email to r\_welch1@yahoo.com. After you verify your email address, you can invite existing AWS accounts into your organization.' At the bottom of the sidebar, it says 'AWS Organizations > AWS accounts'. The main content area is titled 'AWS accounts' with a 'Add an AWS account' button. It contains a table with columns 'Organization', 'Actions', 'Hierarchy', and 'List'. The table shows a single entry under 'Organization': 'Root' with 'r-welch' and 'Joined 2021/11/21'. The table also has a header 'Account created/joined date'.

The accounts listed below are members of your organization. The organization's management account is responsible for paying the bills for all accounts in the organization. You can use the tools provided by AWS Organizations to centrally manage these accounts.

This screenshot is similar to the one above, but it shows a new account has been added. The 'AWS accounts' table now includes a row for 'r-welch' with the status 'Invited' and 'Joined 2021/11/21'. The rest of the interface is identical to the first screenshot.

Create a new AWS group under your organization.

It may take some time to create a new account.

Click on Root and then select the action drop-down and create a new organizational group.

The screenshot shows the AWS Organizations AWS accounts page. On the left, there's a sidebar with 'AWS accounts' (Invitations, Services, Policies, Settings, Get started) and 'Organization ID' (Organization ID: 123456789012). The main area is titled 'AWS accounts' with a sub-section 'Organization'. It lists two accounts: 'awsletch' (Management account) and 'testlaborganization'. A red box highlights the 'Actions' dropdown menu above the table. Another red box highlights the 'testlaborganization' row in the table.

Enter a name:

The screenshot shows the 'Create organizational unit in Root' dialog box. The 'Details' section has a red box around the 'Organizational unit name' input field, which contains 'Architects'. Below it is a note: 'An OU name can be up to 128 characters.' The 'Tags' section is below, with a note: 'Tags are key-value pairs that you can add to AWS resources to help identify, organize, and monitor your AWS resources.' It says 'No tags are associated with this resource.' and has an 'Add tag' button. At the bottom right are 'Cancel' and 'Create organizational unit' buttons.

Please let us know what other features you want to be implemented via Feedback at the bottom of the console.

AWS Organizations > AWS accounts

### AWS accounts

The accounts listed below are members of your organization. The organization's management account is responsible for paying the bills for all accounts in the organization. You can use the tools provided by AWS Organizations to centrally manage these accounts. Learn more [\[?\]](#)

**Organization**

Organizational units (OU) enable you to group several accounts together and administer them as a single unit instead of one at a time.

Find AWS accounts by name, email, or account ID. Find an OU by the exact OU ID:

[Hierarchy](#) [List](#)

**Actions** [Actions](#)

**Organizational structure**

**Account created/joined date**

OU	Account Name	Created/Joined Date
Root	rwelch   rrwelch61@yahoo.com	Joined 2021/11/21
testlaborganization	242796156491   rrwelch61@gmail.com	Created 2021/11/21

Now let's move the new user under the Security Organization by selecting the name and the Actions drop-down. "Move" then select the organization you wish to place them in. Best Practice: prior to creating organizational hierarchies, think about how the company is structured and design to meet their needs.

The accounts listed below are members of your organization. The organization's management account is responsible for paying the bills for all accounts in the organization. You can use the tools provided by AWS Organizations to centrally manage these accounts. Learn more [\[?\]](#)

AWS Organizations > AWS accounts

### AWS accounts

The accounts listed below are members of your organization. The organization's management account is responsible for paying the bills for all accounts in the organization. You can use the tools provided by AWS Organizations to centrally manage these accounts. Learn more [\[?\]](#)

**Organization**

Organizational units (OU) enable you to group several accounts together and administer them as a single unit instead of one at a time.

Find AWS accounts by name, email, or account ID. Find an OU by the exact OU ID:

[Hierarchy](#) [List](#)

**Actions** [Actions](#)

**Organizational structure**

**Account created/joined date**

OU	Account Name	Created/Joined Date
Root	rwelch   rrwelch61@yahoo.com	Joined 2021/11/21
Architects	046-7514-yytfr(A2)	
Security	046-7504-0zy5l7v1	
Production	046-7514-gk5phx41	
Test Environment	046-7504-Sbargz	
testlaborganization	242796156491   rrwelch61@gmail.com	Created 2021/11/21
rrwelch	management account 181961446592   r.welch61@yahoo.com	Joined 2021/11/21

Select the services you wish to assign each user or organization.

Note: Prior to selecting a service, check the pricing before you do so to avoid any unexpected costs.

Note: The organization's "management account" is responsible for paying the bills for all accounts in the organization.

The screenshot shows the AWS Organizations Services page. The left sidebar has 'AWS accounts' expanded, with 'Services' highlighted and a red box around it. Under 'Services', there are links for 'Policies', 'Settings', and 'Get started'. The main content area is titled 'Integrated services' with a dropdown menu set to 'All services'. It lists four services: 'Amazon GuardDuty' (disabled), 'Amazon Macie' (disabled), 'Artifact' (disabled), and 'AWS Account Management' (disabled). Each service has a brief description and a status indicator.

Additionally, you can add policies such as "Service and Control" policies to only allow access to certain applications or other information that a user within this organization is allowed to see or use.

The screenshot shows the AWS Organizations Policies page. The left sidebar has 'AWS accounts' expanded, with 'Policies' highlighted and a red box around it. Under 'Policies', there are links for 'Settings' and 'Get started'. The main content area is titled 'Policies' with a note: 'Policies in AWS Organizations enable you to manage different features of the AWS accounts in your organization.' It lists four policy types: 'AI services opt-out policies' (disabled), 'Backup policies' (disabled), 'Service control policies' (disabled), and 'Tag policies' (disabled). Each policy type has a brief description and a status indicator.

AWS single sign-on: This step may be needed if it has not been enabled.

Make sure to enable Single Sign-on.

The screenshot shows the AWS Single Sign-On (SSO) dashboard. On the left, a sidebar menu includes 'Dashboard', 'AWS accounts', 'Applications', 'Users', 'Groups', and 'Settings'. The main area is titled 'Welcome to AWS Single Sign-On' and contains a message about enabling SSO. Below this, 'Recommended setup steps' are listed: 1. Choose your identity source (with a note about the identity source being where users log in), 2. Manage SSO access to your AWS accounts (with a note about giving users access to specific AWS resources), and 3. Manage SSO access to your cloud applications (with a note about giving users access to cloud applications). At the bottom, there's information about the User portal and its URL: <https://id-8a6711d8fe.awsapps.com/start>.

You may delete an organization by selecting Settings.

## Chapter 9 Labs

### Lab: Container Clusters in AWS Using ECS/EKS

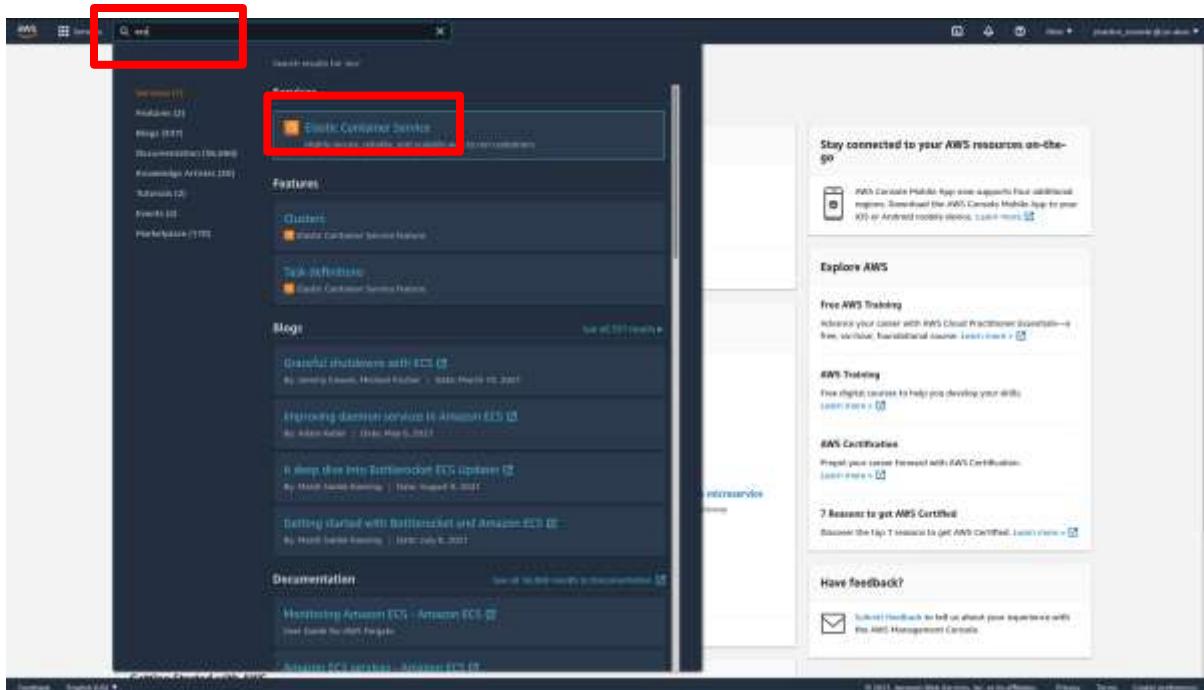
#### **Description:**

In this lab section we will focus on containers. We will be creating container clusters using the AWS Elastic Container Service and Elastic Kubernetes Service.

First, we will begin by setting up Elastic Container Service. Second, we will create Kubernetes clusters with Elastic Kubernetes Service.

#### **ECS Cluster (Elastic Container Service)**

From the Management Console, go to the **Elastic Container Service** dashboard.



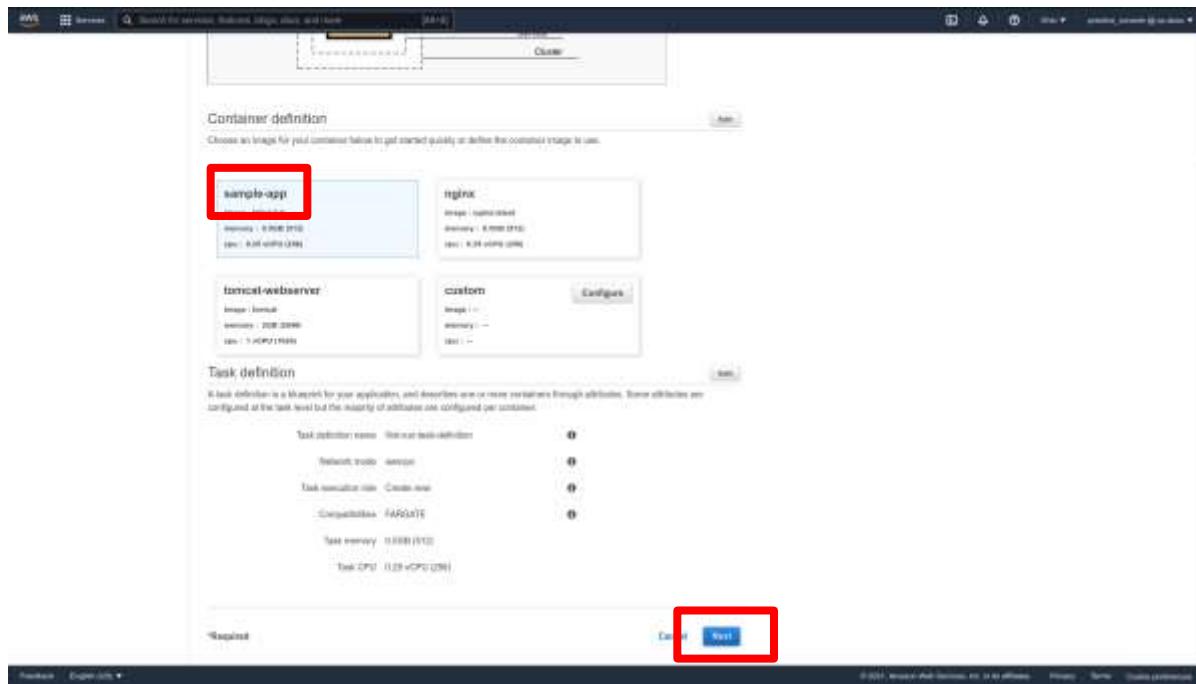
Click on the **Clusters** section, under **Amazon ECS**.

The screenshot shows the AWS Management Console for Amazon Elastic Container Service (ECS). The left sidebar has a red box around the 'Clusters' link under the 'Amazon ECS' section. The main content area is titled 'Amazon Elastic Container Service (ECS)' and features a video thumbnail about ECS. Below the video, there's a paragraph about ECS making it easy to deploy, manage, and scale Docker containers running applications, services, and batch processes. It mentions integration with ELB, EC2 security groups, EBS volumes, and IAM roles. There are three cards below the paragraph: 'Run containers at scale' (with a container icon), 'Flexible container placement' (with a cluster icon), and 'Integrated and extensible' (with a network icon). At the bottom, there are links for 'Get started' and 'Learn more about Amazon ECS'.

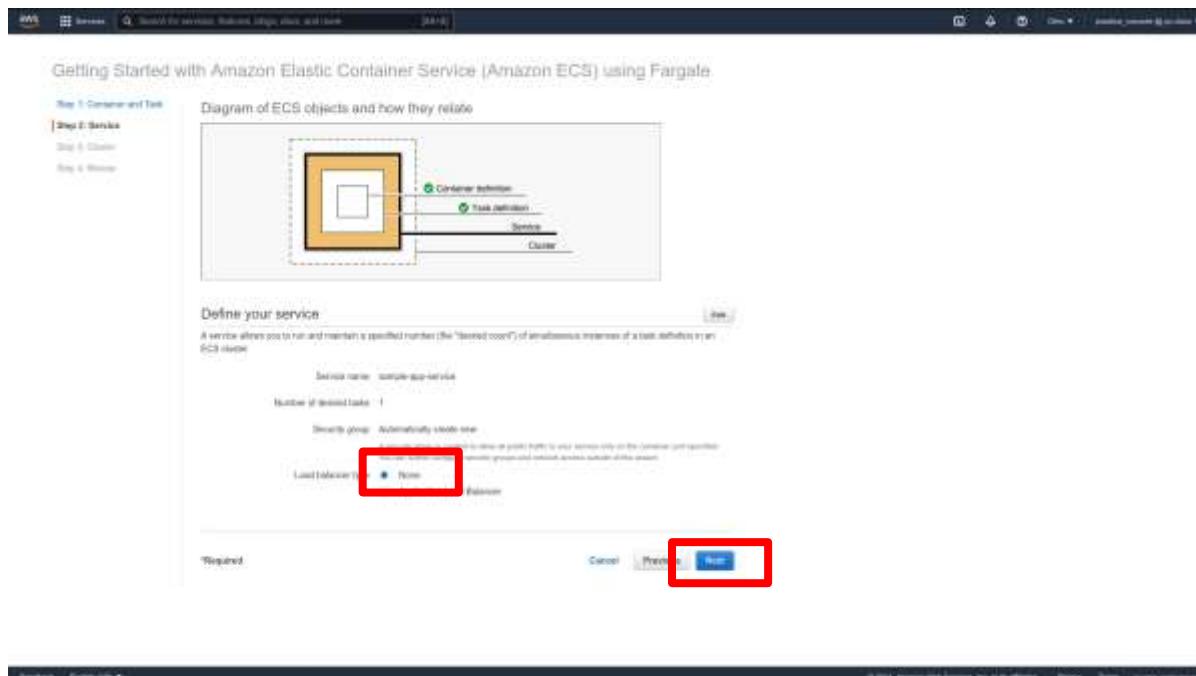
Click on **Create Cluster**.

The screenshot shows the 'Clusters' page in the AWS Management Console for ECS. The left sidebar has a red box around the 'Create Cluster' button. The main content area is titled 'Clusters' and contains a paragraph about what an ECS cluster is. It includes a 'Create Cluster' button and a 'Selected' button. Below this is a table with one row labeled 'No clusters found'. At the bottom, there is a 'Get Started' button.

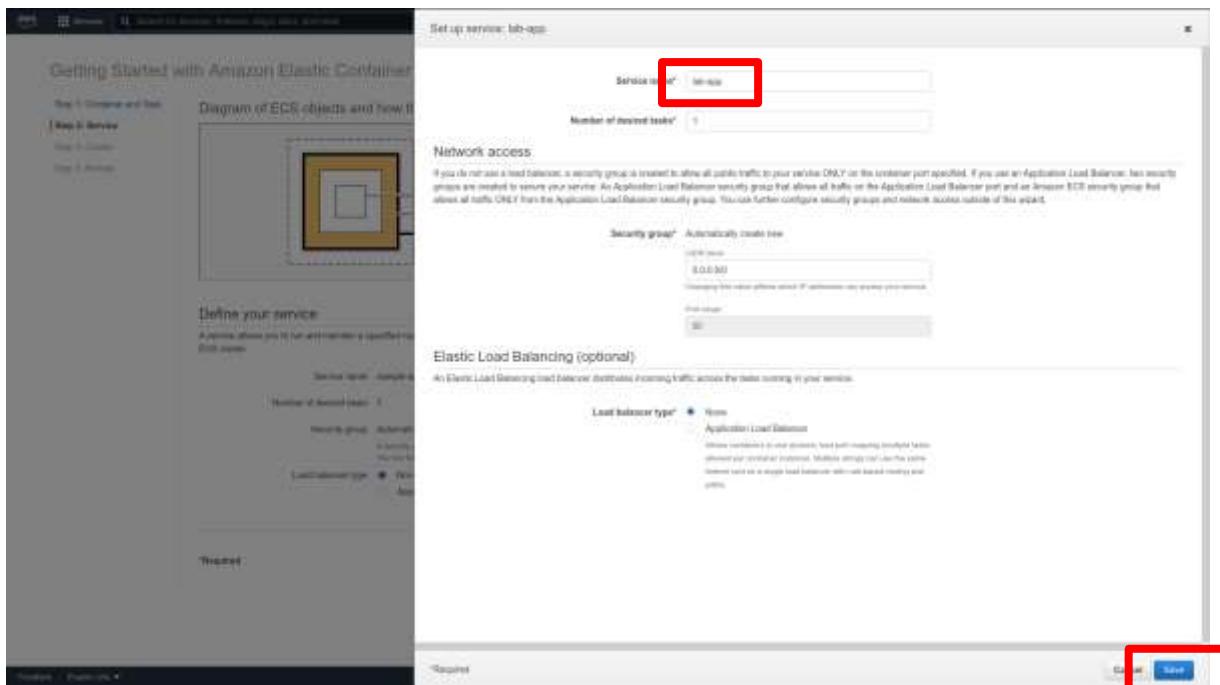
Leave the container image on **sample-app** and click **Next**.



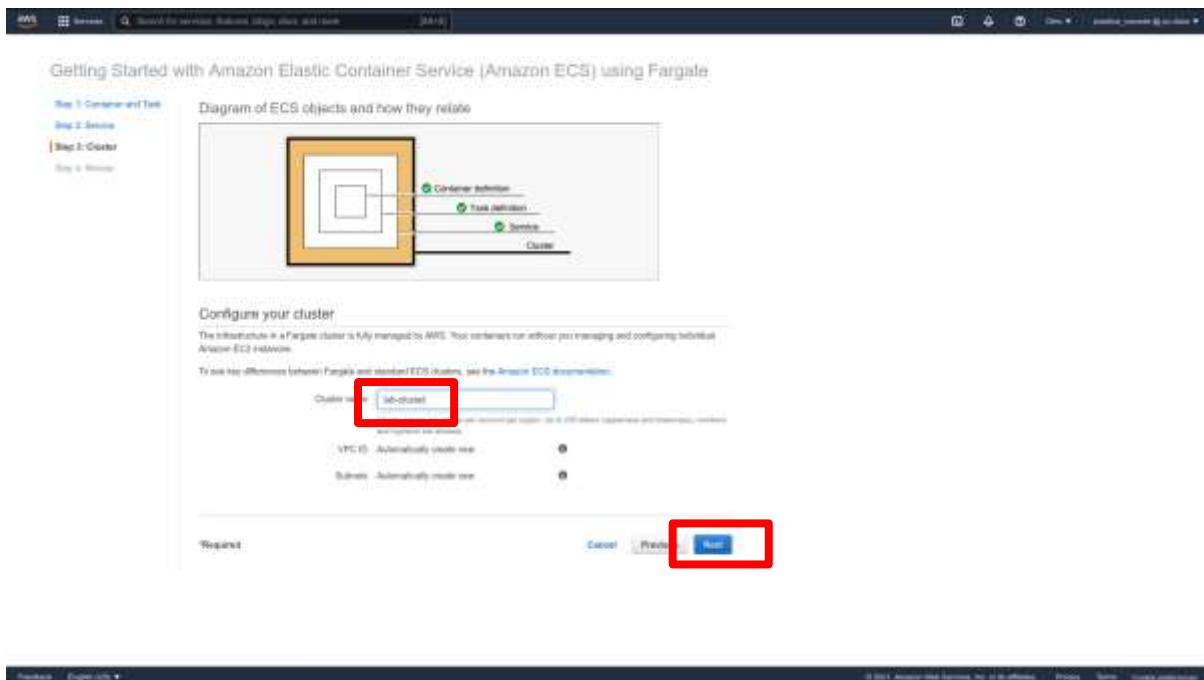
Leave the service as default. For this lab select **None** in the **Load balancer type**. Click **Next**.



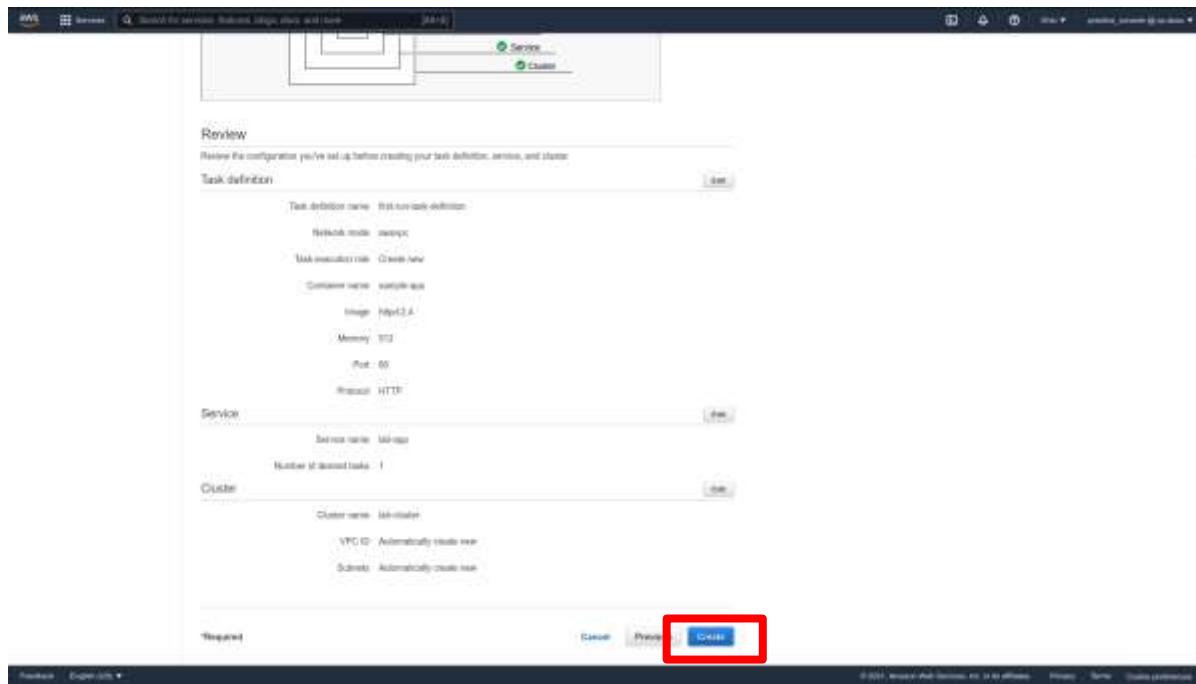
This sidebar will appear. Give a descriptive name to your service. Leave the other settings at their default. Click **Save**.



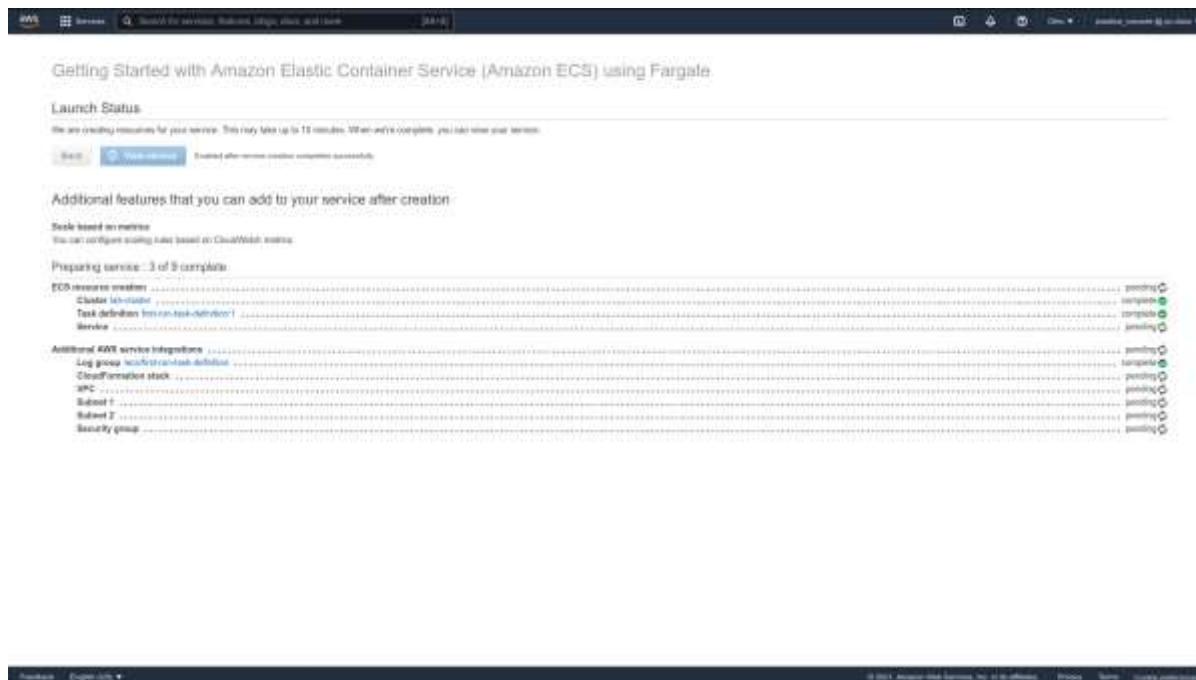
Give a descriptive name for your cluster. Click **Next**.



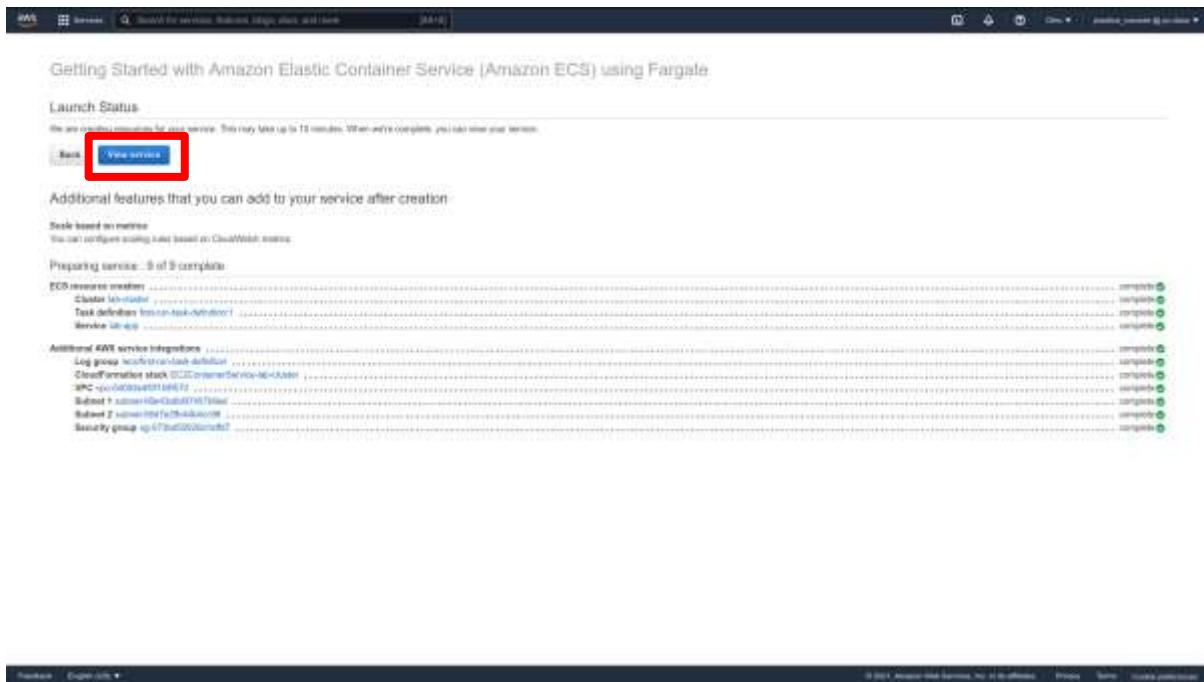
Review the settings. When you are ready, click **Create**.



This page will show the creation process for your cluster. Allow it to complete.

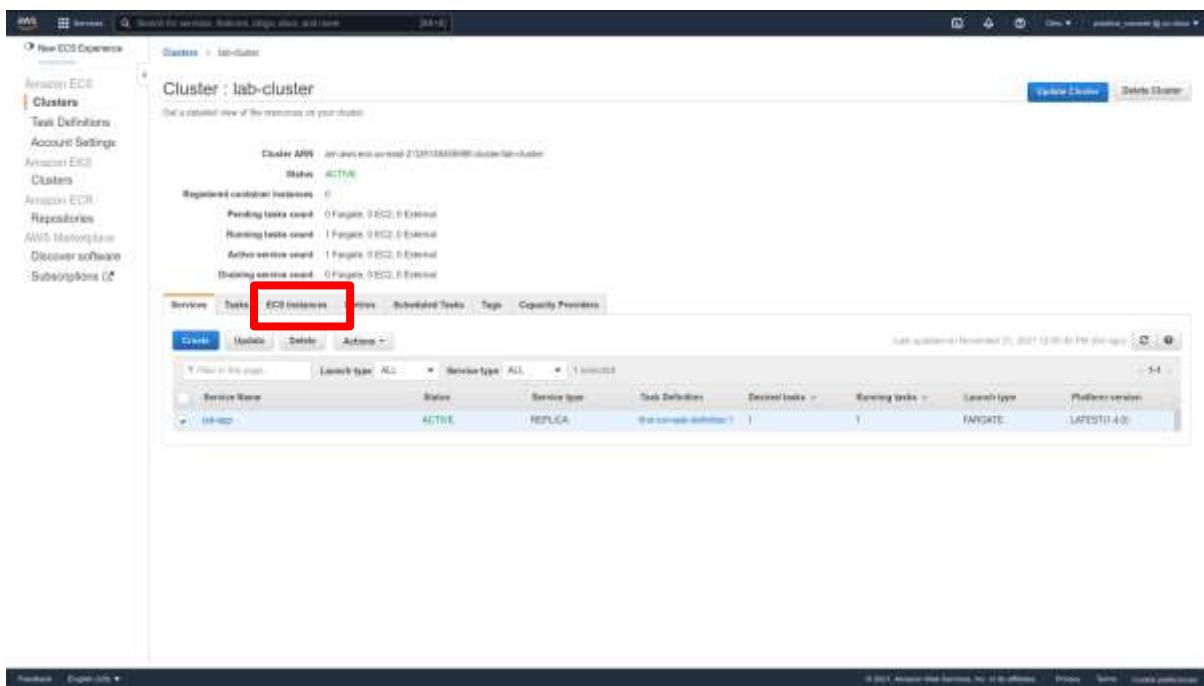


When all is complete, click on **View service** to view your new cluster.



The screenshot shows the 'Getting Started with Amazon Elastic Container Service (Amazon ECS) using Fargate' page. At the top, there's a 'Launch Status' section with a progress bar indicating 'Preparing service... 9 of 9 complete'. Below this, a 'View service' button is highlighted with a red box. Further down, there are sections for 'Additional features that you can add to your service after creation' (including 'Scale based on metrics' and 'AWS Lambda triggers') and 'Additional AWS service integrations' (listing Log Insights, CloudFormation stacks, S3, CloudWatch Metrics, and others). At the bottom right, there are links for 'Feedback', 'English (US)', and '© 2014 Amazon Web Services, Inc. or its affiliates.'

Here is the summary for your cluster. Container instances will be in the ECS Instances tab. You can also edit or delete your cluster from here.



The screenshot shows the 'Cluster : lab-cluster' summary page. It displays basic cluster statistics: Cluster ARN, Status (ACTIVE), Registered container instances (0), Pending tasks count (0 Fargate, 0 EC2, 0 External), Running tasks count (1 Fargate, 0 EC2, 0 External), Active service count (1 Fargate, 0 EC2, 0 External), and Pending service count (0 Fargate, 0 EC2, 0 External). Below this, there are tabs for 'Services', 'Tasks', 'ECS Instances' (which is highlighted with a red box), 'Services', 'Scheduled Tasks', 'Tags', and 'Capacity Provision'. A table below lists one service entry: Service Name (HTTP-001), Status (ACTIVE), Service type (REPLICA), Task Definition (http-service-definition), Desired tasks (1), Running tasks (1), Launch type (FARGATE), and Last registered version (LATEST(4)). At the bottom right, there are links for 'Feedback', 'English (US)', and '© 2014 Amazon Web Services, Inc. or its affiliates.'

## EKS Cluster (Kubernetes)

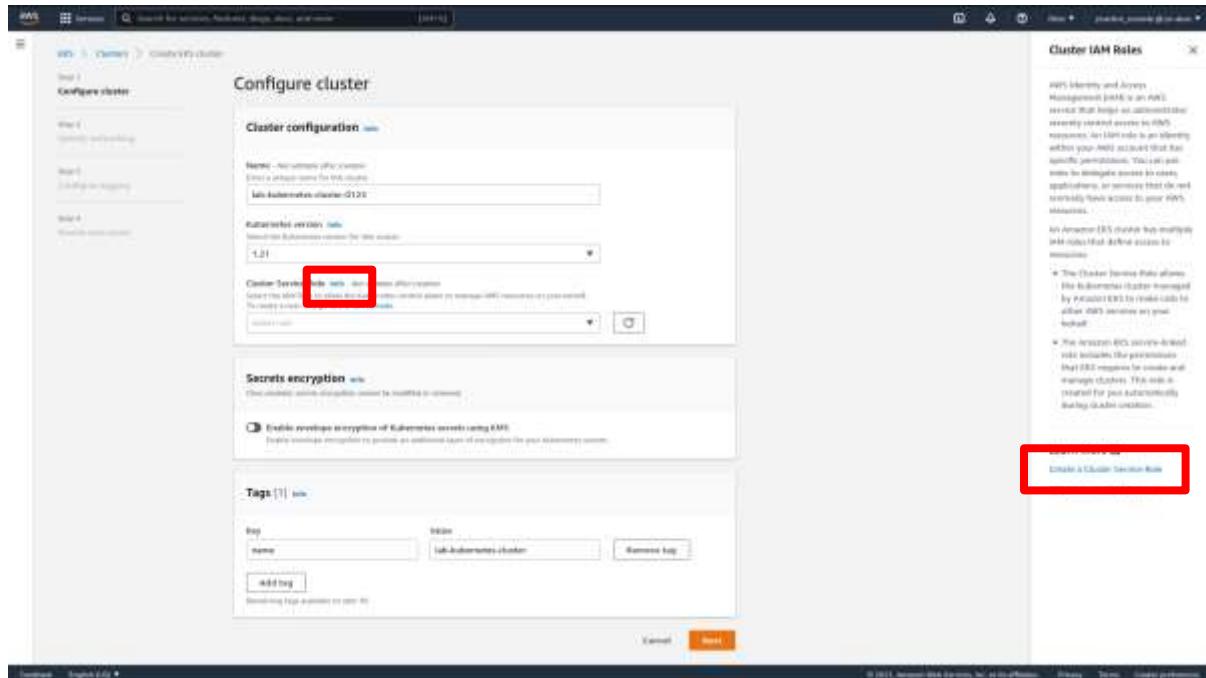
Now, let's create a Kubernetes cluster. Click on **Clusters** under the **Amazon EKS** section.

The screenshot shows the AWS EKS Cluster management interface. On the left, a sidebar menu is open with the 'Clusters' option highlighted and a red box drawn around it. The main content area displays a cluster named 'lab-cluster'. It shows the Cluster ARN, Status (ACTIVE), and various metrics like Registered container instances (0), Pending tasks (0), Running tasks (1), Active service record (1), and Pending service record (0). Below this is a table titled 'Services' with columns for Service Name, Status, Service Type, Task Definition, Desired Tasks, Running Tasks, Layout Type, and Platform Version. One entry, 'HF-HD', is listed with an ACTIVE status and REPLICAS layout type. At the bottom right of the main content area, there is a timestamp: 'Last updated on November 01, 2021 12:00 AM (UTC)'. The bottom of the page includes standard AWS navigation links and a copyright notice: '© 2021 Amazon Web Services, Inc. or its affiliates. All Rights Reserved.'

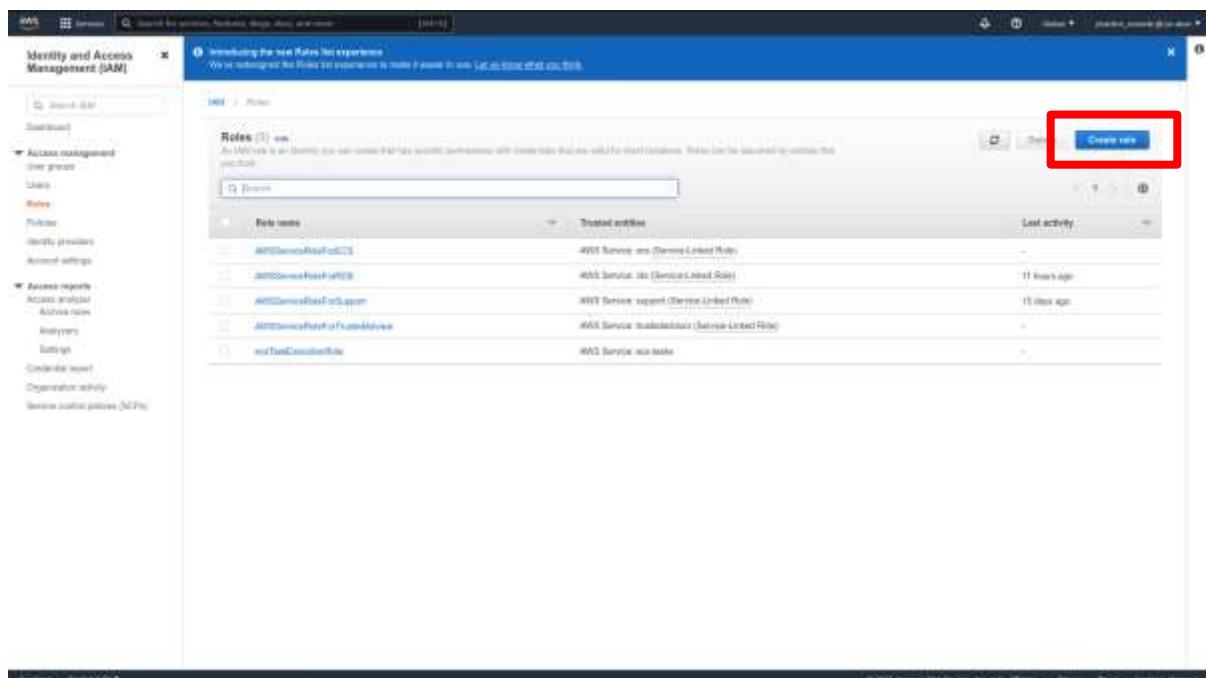
Click on **Add cluster**.

The screenshot shows the AWS EKS landing page. The main heading is 'Elastic Kubernetes Service (Amazon EKS)' followed by the subtext 'Fully managed Kubernetes control plane'. To the right, there is a call-to-action box with the 'Add cluster' button, which is also highlighted with a red box. Below this, there are sections for 'How it works', 'Why use Amazon EKS?', and 'Amazon EKS resources'. The 'How it works' section contains a diagram showing a flow from 'Compute & storage' through 'Networking & security' to 'Container orchestration'. The 'Why use Amazon EKS?' section lists 'Hybrid container deployments' and 'Microservices'. The 'Amazon EKS resources' section lists 'Amazon EKS clusters', 'Amazon EKS roles', and 'Amazon EKS dashboards'. The bottom of the page includes a footer with links to 'Home', 'English (US)', and 'Feedback'.

Don't start filling out the form yet. First, we need to create an IAM role for Kubernetes clusters, as there is none set up for you by default. Click on **Info** next to Cluster Service Role. Then, a sidebar should appear. Briefly read the provided description. When done, click **Create a Cluster Service Role**. This will send us to the **Roles** section of the IAM Console.



Click **Create role**.



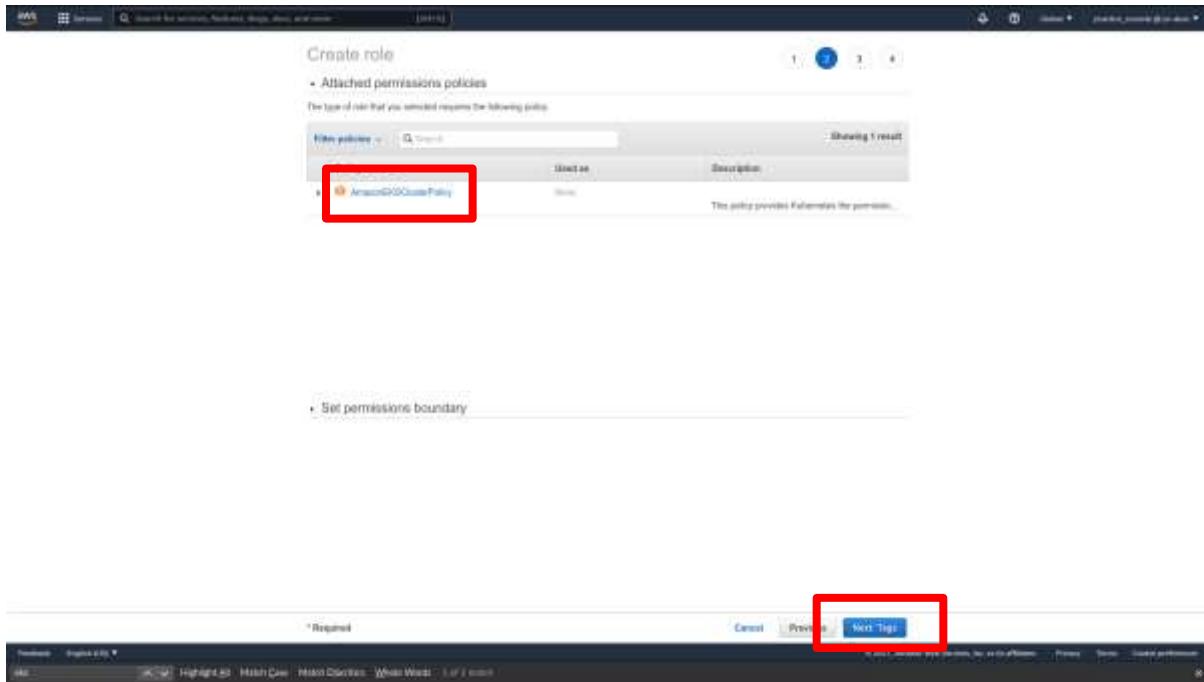
Search for the term **EKS** and click on it.

The screenshot shows a search results page for AWS Lambda functions. A red box highlights the 'EKS' entry in the list, which is described as 'Amazon EKS Cluster'. Below the list, there is a section titled 'Select your use case' with several options, one of which is highlighted with a red box. At the bottom right of the page, there is a blue button labeled 'Next: Permissions'.

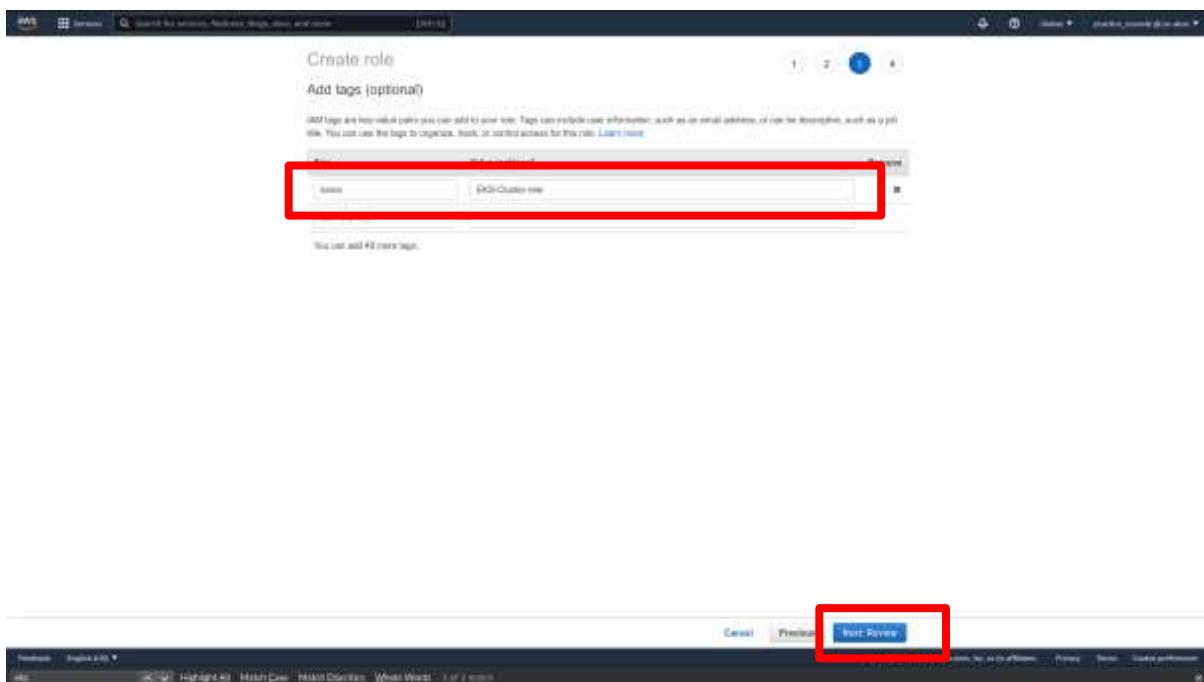
Select **EKS-Cluster**. Click **Next: Permissions**.

The screenshot shows the 'Select your use case' step in the AWS Lambda function creation wizard. A red box highlights the 'EKS - Cluster' option, which is described as 'Allows access to other AWS service resources that are required to operate clusters managed by EKS'. Below this, there are four other options: 'EKS - Fargate pod', 'EKS - Fargate profile', 'EKS - Fargate profile', and 'EKS - Headless'. At the bottom right of the page, there is a blue button labeled 'Next: Permissions'.

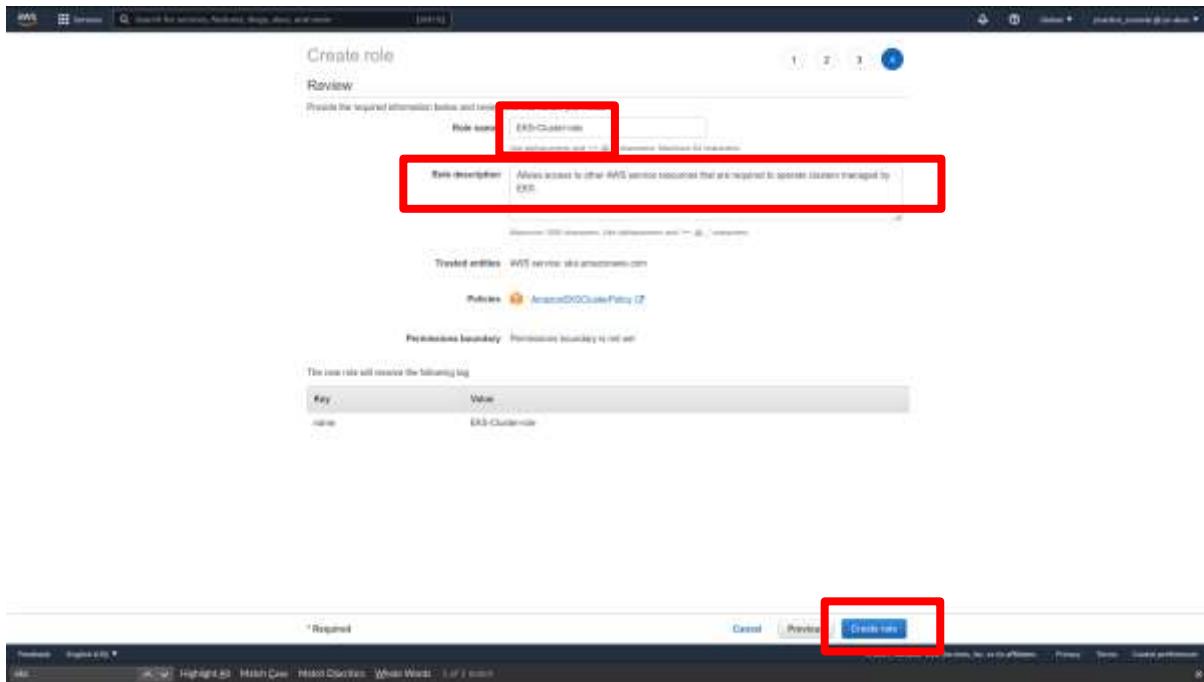
Select **AmazonEKSClusterPolicy** to attach this policy to your new IAM role. Click **Next: Tags**.



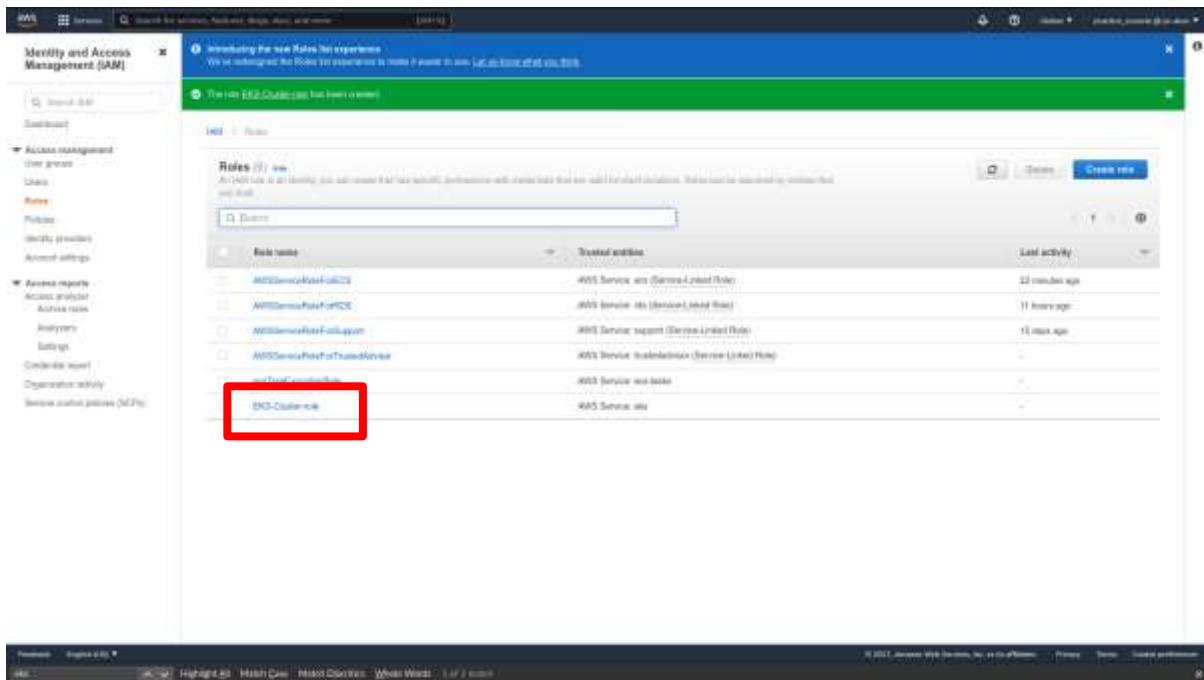
Add a “name” tag, and give it a descriptive name. Click **Next: Review**.



Give a descriptive name to your Kubernetes cluster role. Provide a description for others to understand its purpose. When you are ready, click **Create role**. This will be your new IAM role to handle Kubernetes clusters.



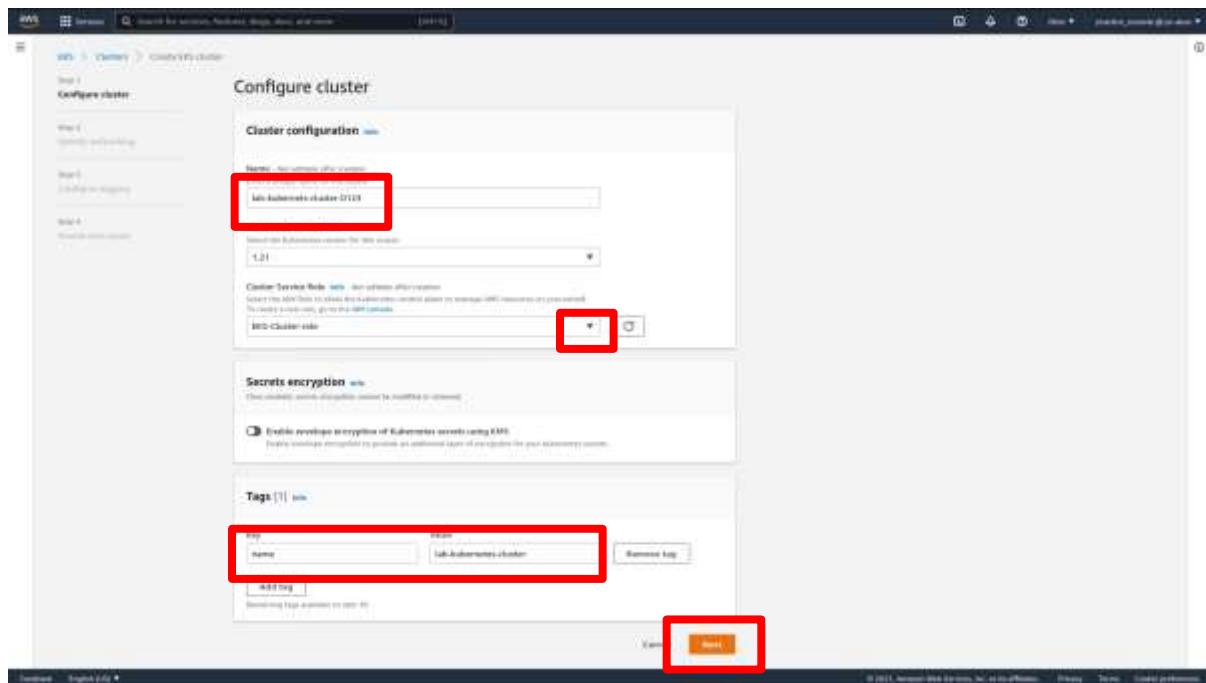
We can see the new role has been added. You can attribute this role to users you want to give access to EKS Cluster management. Our work here is done. Let's go back to the main page for EKS (scroll back up if you do not remember how). There, you can restart the process of creating a cluster (by clicking **Add cluster**).



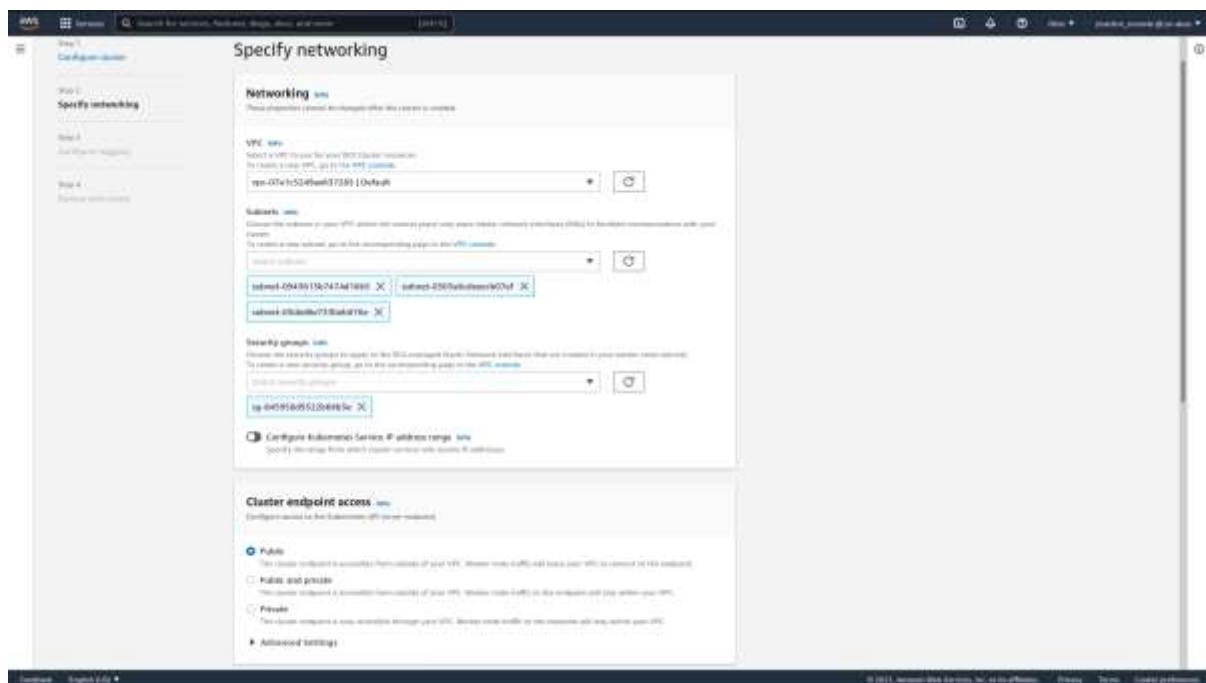
The screenshot shows the AWS Identity and Access Management (IAM) service. On the left, there is a navigation sidebar with options like User groups, Users, Roles, Policies, Metrics (CloudWatch), Account settings, and more. Under the Roles section, there is a sub-menu for Roles. The main content area displays a table of roles. One row, 'EKS-ClusterRole', is highlighted with a red box. The table columns include Role name, Trusted entities, and Last activity. Other roles listed include 'AmazonEC2RoleForAWSLambda', 'AmazonEC2RoleForComputeOptimizer', 'AmazonEC2RoleForCloudWatchLogs', 'AmazonEC2RoleForCloudWatchMetrics', 'AmazonEC2RoleForCloudWatchStreams', 'AmazonEC2RoleForCloudWatchLogsMetrics', and 'AmazonEC2RoleForCloudWatchLogsStreams'. The 'Last activity' column shows times ranging from 11 hours ago to 22 minutes ago.

Now that we are back, choose a **unique** name/identifier for this cluster. Leave the Kubernetes version to the default. In the Cluster Service Role box, select your newly created role. Now, you know what this box is for.

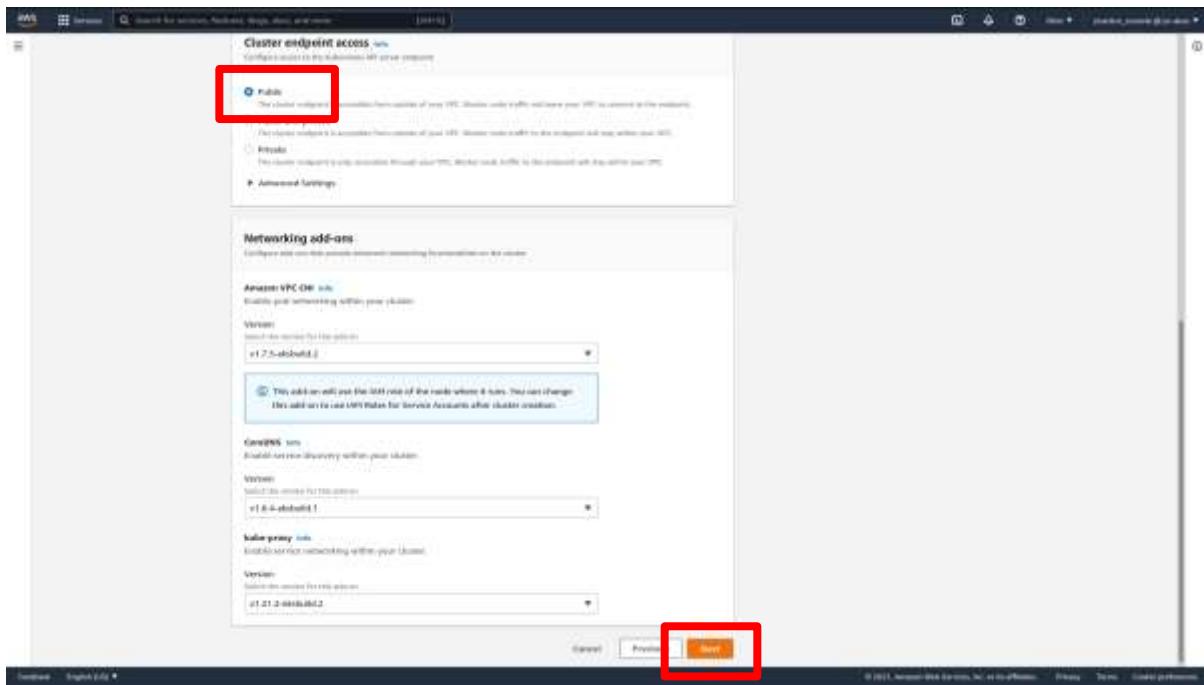
We won't be enabling encryption for this lab, but know that it is possible to do so. Add a "name" tag for your Kubernetes cluster, and give it a descriptive name. Click **Next**.



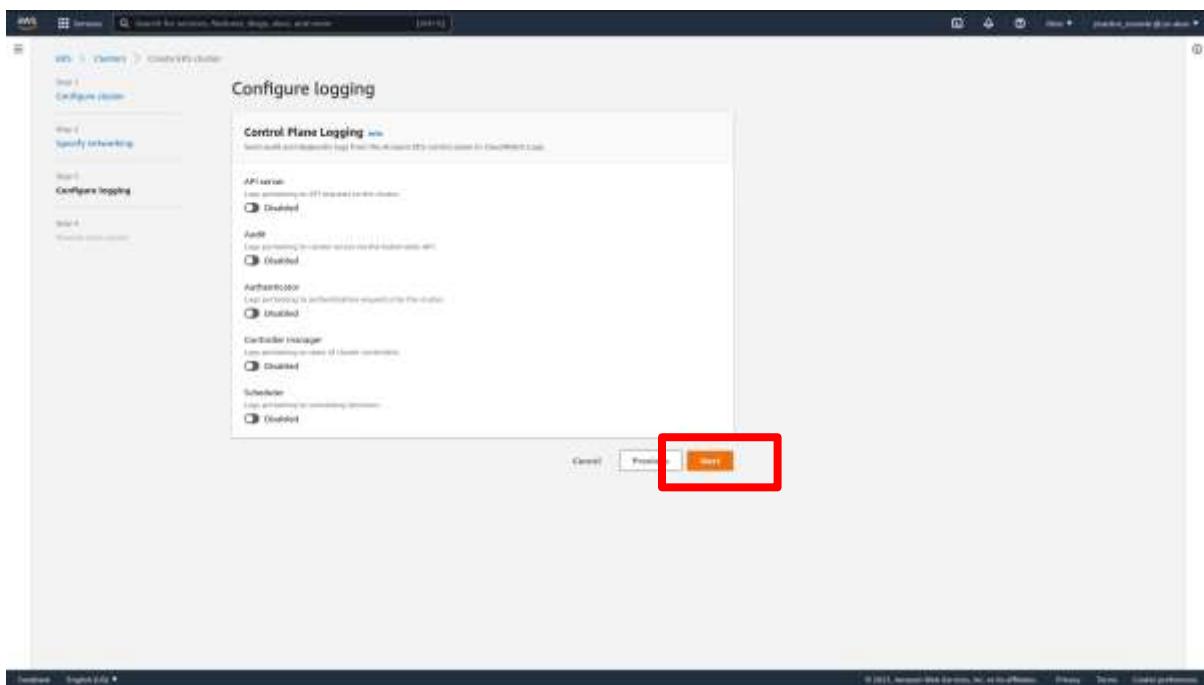
You can specify a specific **VPC** for your cluster. In this lab, we will leave it on the default VPC. We will also leave all the **subnets** of the default VPC enabled, but know that you can restrict the cluster to only certain subnets. We will also be leaving the default **security group**. Scroll down.



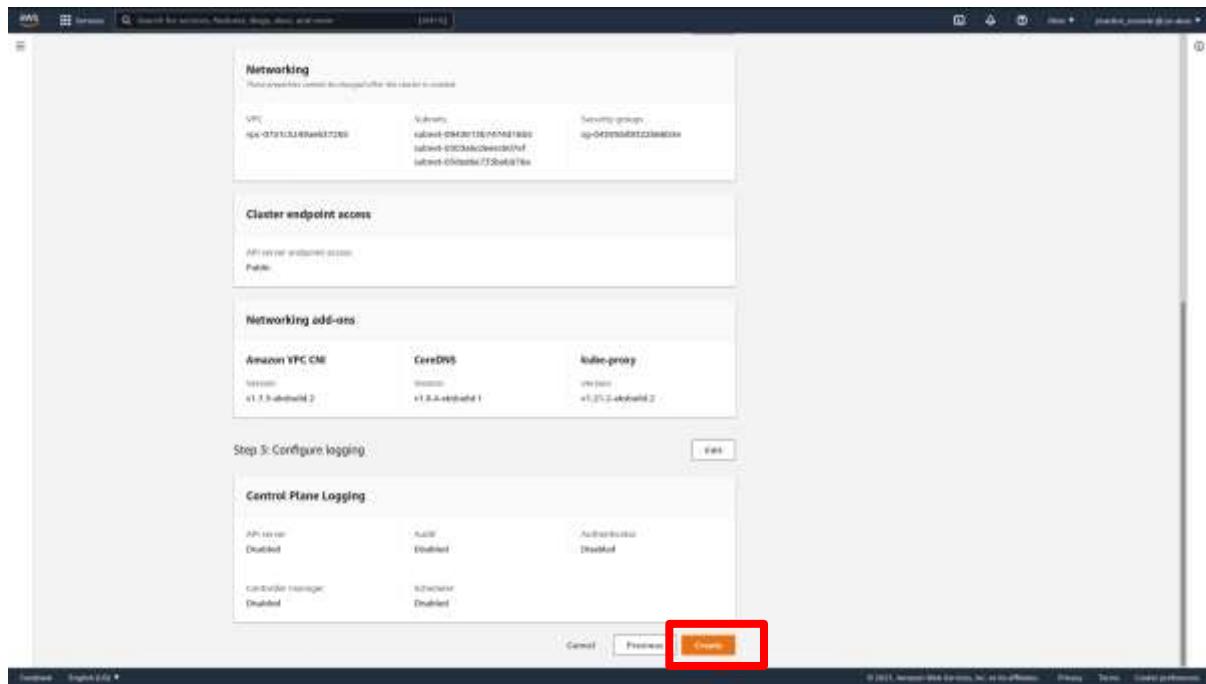
For this lab, we will allow **Public** access to the cluster. This means your Kubernetes workers will be accessible from the internet. Here, it says the worker traffic will leave the VPC to connect to an endpoint. Note, however, that it does not leave Amazon's internal network (it does not use the public internet). Leave the version numbers at their default. Click **Next**.



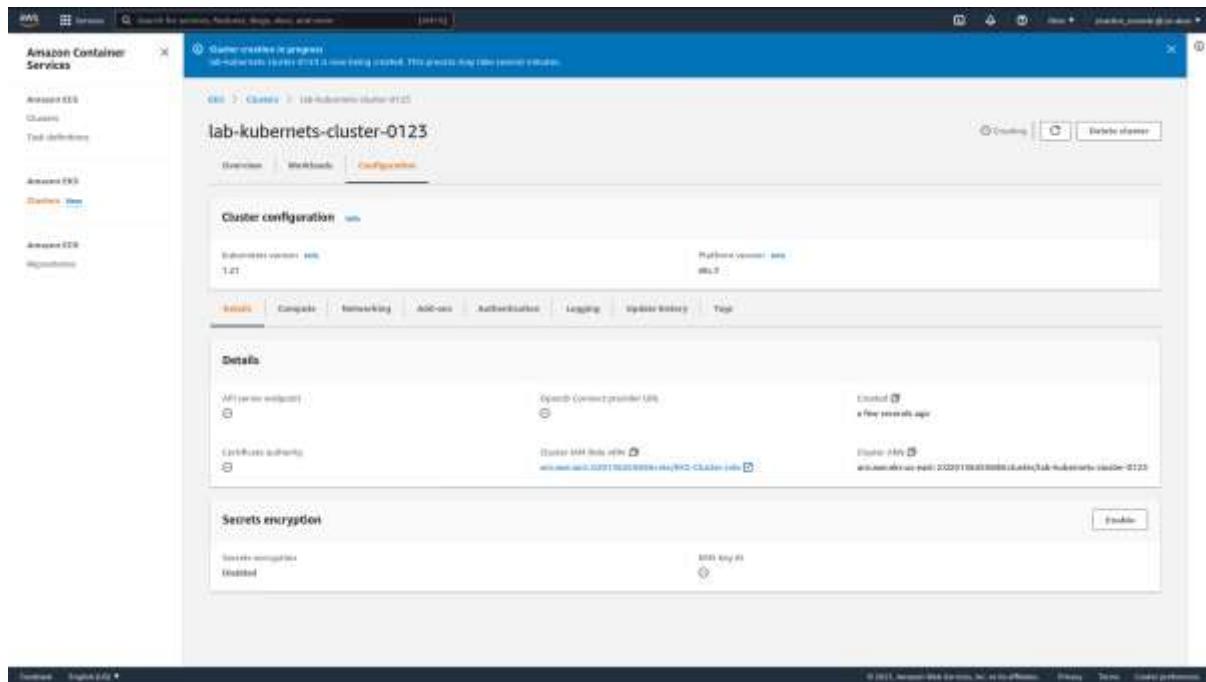
You can configure logs for your Kubernetes cluster. For now, you can simply click **Next**.



Review the settings for the Kubernetes cluster. When you are ready, click **Create**.



Your Kubernetes cluster will now begin the creation process. You can stay here or come back, and it will complete in the background. You can use this page to view information regarding your Kubernetes cluster, or to **delete** it.



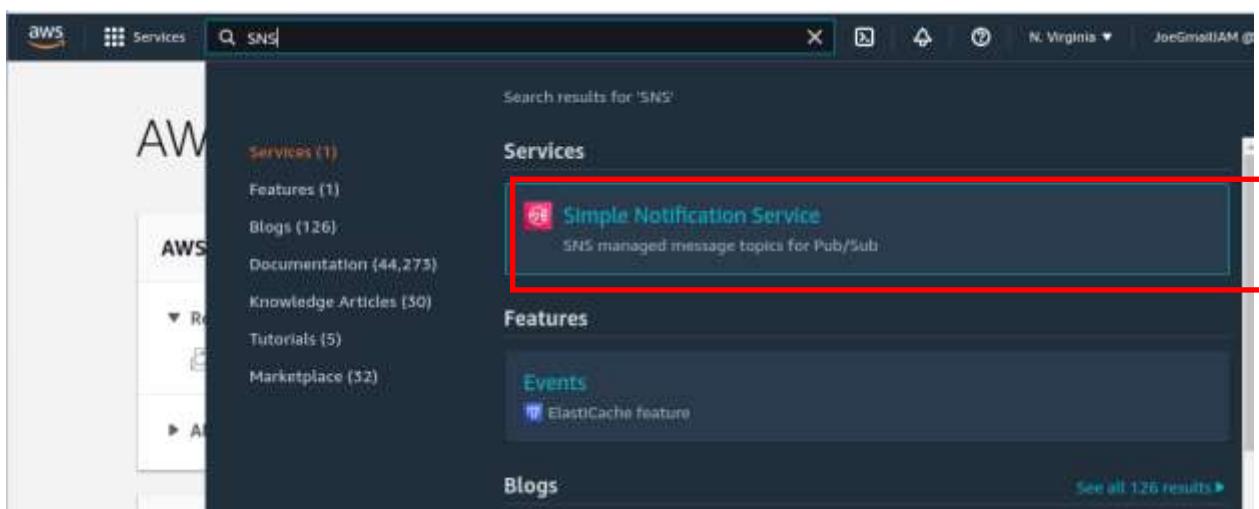
## Lab: Create AWS SNS Topic and Get Email Notifications

In this lab we are going to set up an AWS SNS Topic to alert us about the Budget we have set up on our account.

### **Creating an AWS SNS Topic**

At the AWS Management Console, in the search box at the top left-hand corner, type **SNS**.

From the results, click **Simple Notification Service**.



#### Services

##### **Simple Notification Service**

SNS managed message topics for Pub/Sub

#### Features

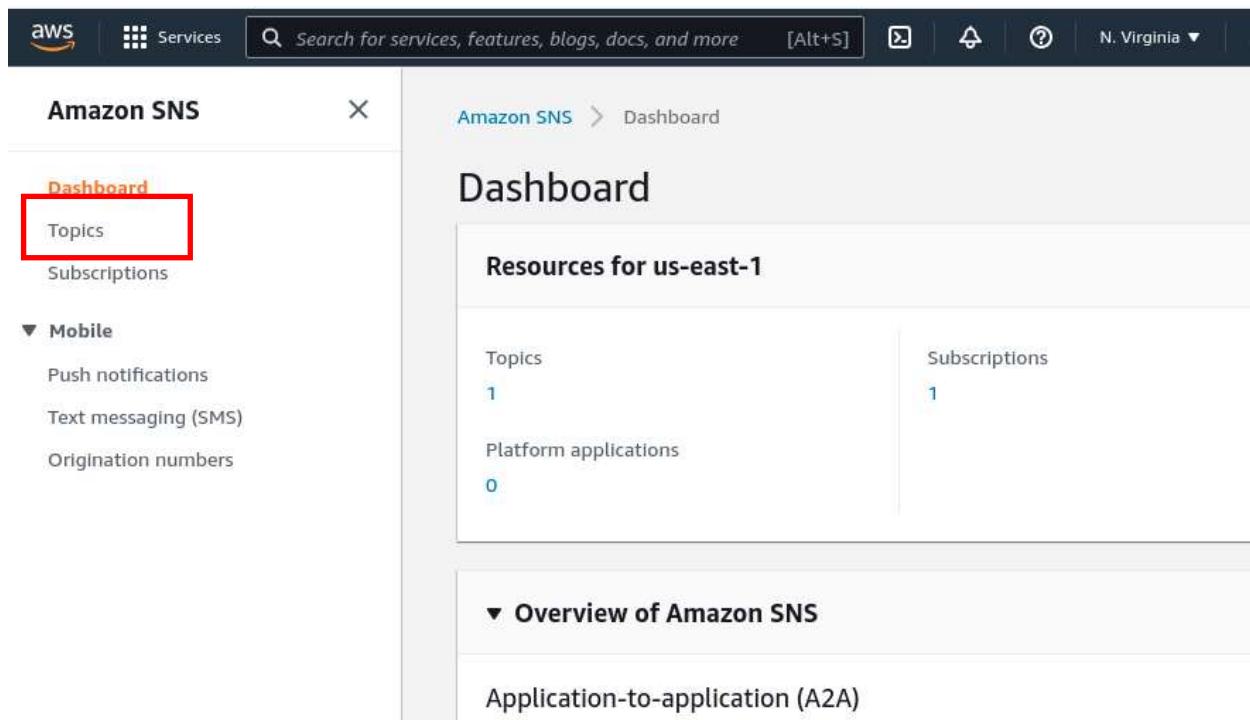
##### Events

ElastiCache feature

#### Blogs

[See all 126 results ▾](#)

From the resulting dashboard, click on **Topics**.



The screenshot shows the Amazon SNS Dashboard. On the left, there is a navigation sidebar with the following items:

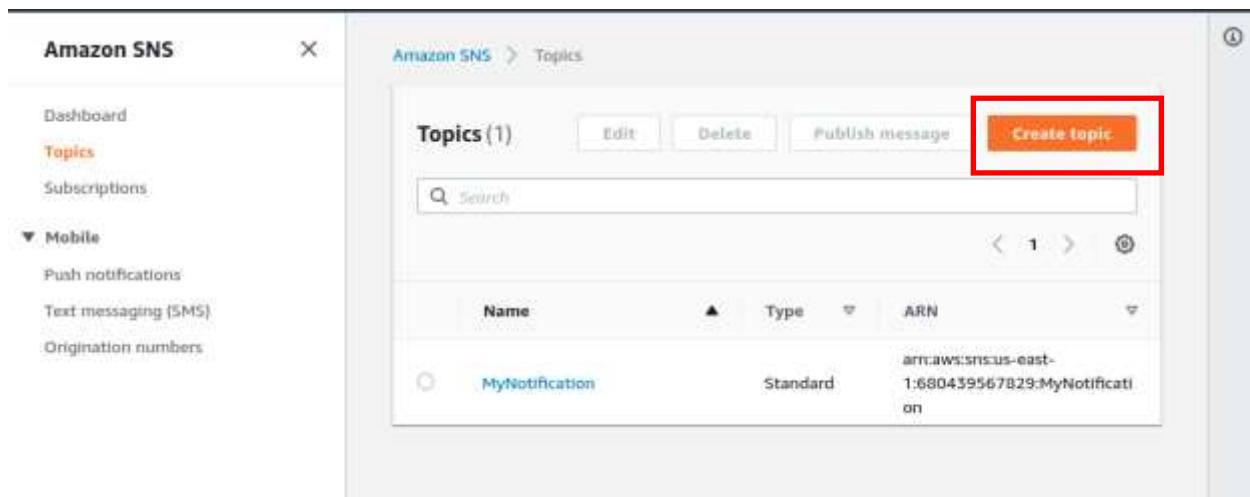
- Dashboard** (highlighted with a red box)
- Topics** (highlighted with a red box)
- Subscriptions

The main content area is titled "Dashboard" and contains a section titled "Resources for us-east-1". It displays the following statistics:

Topics	Subscriptions
1	1

Below this, there is a section titled "Overview of Amazon SNS" with the sub-section "Application-to-application (A2A)".

Click on the **Create Topic** button.



The screenshot shows the "Topics" page within the Amazon SNS service. The left sidebar has the following items:

- Dashboard
- Topics** (highlighted)
- Subscriptions

The main content area shows a table of topics:

Name	Type	ARN
MyNotification	Standard	arn:aws:sns:us-east-1:680439567829:MyNotification

A prominent orange "Create topic" button is located at the top right of the table area, with a red box highlighting it.

We will choose the type of SNS topic. For example, if we were setting up a topic to publish to an SQS FIFO queue, we would choose FIFO. As we are setting up email, we will choose the **Standard** type.

Type in a recognizable name. We have filled this field with **Budget\_Topic**.

Type in a Display name. This is optional, but we have input **Budget Topic**.

The screenshot shows the 'Create topic' wizard in the Amazon SNS console. On the left, there's a sidebar with 'Topics' selected. The main area has a header 'Create topic' and a 'Details' tab. Under 'Type', 'Standard' is selected, while 'FIFO (first-in, first-out)' is shown with its characteristics: 'First-in message ordering', 'At-least-once message delivery', 'High throughput, up to 100 messages/second', and 'Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoint'. The 'Name' field contains 'Budget\_Topic' and the 'Display name - optional' field contains 'Budget Topic'.

Accept the default configurations in optional sections and click **Create Topic**.

Amazon SNS

Dashboard

**Topics**

Subscriptions

▼ Mobile

Push notifications

Text messaging (SMS)

Origination numbers

Name

Budget\_Topic

Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (\_).

Display name - optional

To use this topic with SMS subscribers, enter a display name. Only the first 10 characters are displayed in an SMS message. [Info](#)

Budget Topic

Maximum 100 characters, including hyphens (-) and underscores (\_).

► Encryption - optional

Amazon SNS provides in-transit encryption by default. Enabling server-side encryption adds an extra layer of security to your topic.

► Access policy - optional

This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic. [Info](#)

► Delivery retry policy (HTTP/S) - optional

The policy defines how Amazon SNS retries failed deliveries to HTTP/S endpoints. To modify the default settings, expand this section. [Info](#)

► Delivery status logging - optional

These settings configure the logging of message delivery status to CloudWatch Logs. [Info](#)

► Tags - optional

A tag is a metadata label that you can assign to an Amazon SNS topic. Each tag consists of a key and an optional value. You can use tags to search and filter your topics and track your costs. [Learn more](#) [?](#)

Cancel

Create topic

As we see here, the topic has been created.

The screenshot shows the Amazon SNS console. On the left, a sidebar lists 'Dashboard', 'Topics' (which is selected and highlighted in orange), 'Subscriptions', and 'Mobile' (with sub-options: Push notifications, Text messaging (SMS), Origination numbers). The main content area is titled 'Budget\_Topic'. At the top, a green banner displays the message 'Topic Budget\_Topic created successfully. You can create subscriptions and send messages to them from this topic.' Below the banner are three buttons: 'Edit', 'Delete', and 'Publish message'. The 'Details' section contains the following information:

Name	Budget_Topic	Display name	Budget Topic
ARN	arn:aws:sns:us-east-1:680439567829:Budget_Topic	Topic owner	680439567829
Type	Standard		

Below the details are tabs for 'Subscriptions', 'Access policy', 'Delivery retry policy (HTTP/S)', 'Delivery status logging', 'Encryption', and 'Tags'. The 'Subscriptions' tab is selected, showing '(0)'.

Now that the topic is created, services can subscribe to the topic to receive notifications. We will subscribe to this topic with "Budget" and be notified by email.

## How to Subscribe to a Topic

Repeat Steps 1 and 2 to get to the Amazon SNS page, if you are not already there. Click on **Subscriptions**.

Click the **Create Subscriptions** button.

The screenshot shows the 'Subscriptions' page of the Amazon SNS console. The sidebar on the left is identical to the previous screenshot, showing 'Topics' as the selected item. The main content area is titled 'Subscriptions (1)'. It includes a toolbar with buttons for 'Edit', 'Delete', 'Request confirmation', and 'Confirm subscription'. A large orange button labeled 'Create subscription' is prominently displayed. Below the toolbar is a search bar with the placeholder 'Search'. At the bottom is a table with columns: ID, Endpoint, Status, Protocol, and Topic. The table shows one row of data corresponding to the previously created topic.

Click inside the textbox. From the drop-down menu, identify the Amazon Resource Name (ARN) of the topic you would like to subscribe to. You will recognize this number because the name you gave the topic will be at the end of this number.

The screenshot shows the 'Create subscription' page in the AWS SNS console. At the top, there is a banner with a warning about changes to sending SMS messages to US destinations. Below the banner, the navigation path is 'Amazon SMS > Subscriptions > Create subscription'. The main section is titled 'Create subscription' and contains a 'Details' tab. Under the 'Topic ARN' heading, there is a search bar with the placeholder 'my-topic' and a dropdown menu listing two ARNs: 'arn:aws:sns:us-east-1:600439567829:Budget\_Topic' and 'arn:aws:sns:us-east-1:600439567829:MyNotification'. A note at the bottom of the form states: 'After your subscription is created, you must confirm it.' with a 'Info' link.

When you click the desired ARN, the textbox will be populated with your choice.

This screenshot is identical to the one above, but the 'arn:aws:sns:us-east-1:600439567829:Budget\_Topic' option is now selected in the dropdown menu, indicating it has been chosen.

Click inside the textbox for Protocol.

As we would like to be notified by email, Select **Email** from the drop-down list.

The screenshot shows the 'Create subscription' page in the Amazon SNS console. At the top, there is a banner about important changes for sending SMS messages to US destinations. Below the banner, the navigation path is 'Amazon SNS > Subscriptions > Create subscription'. The main section is titled 'Create subscription' and has a 'Details' tab selected. Under the 'Topic ARN' field, the value 'arn:aws:sns:us-east-1:680439567829:Budget\_Topic' is entered. The 'Protocol' section is expanded, showing a list of options: 'Select protocol' (highlighted with a red box), 'Amazon Kinesis Data Firehose', 'Amazon SQS', 'AWS Lambda', 'Email' (highlighted with a red box), 'Email-JSON', 'HTTP', 'HTTPS', 'Platform application endpoint', and 'SMS'. Below the protocol list, there is a note about sending undeliverable messages to a dead-letter queue with a 'Info' link. Under the 'Redrive policy (dead-letter queue)' section, the 'Disabled' radio button is selected. At the bottom right, there are 'Cancel' and 'Create subscription' buttons.

The Text box will now be populated with your choice.

Your email address will be the endpoint for the topic. Fill this in, accept the defaults in the optional sections, and click the **Create subscription** button.

The screenshot shows the 'Create subscription' wizard in the Amazon SNS console. The current step is 'Details'. The 'Topic ARN' field contains 'arn:aws:sns:us-east-1:680439567829:Budget\_Topic'. The 'Protocol' dropdown is set to 'Email'. The 'Endpoint' field contains 'someone@provider.com'. A note below the endpoint says: 'After your subscription is created, you must confirm it.' In the 'Subscription filter policy - optional' section, there is a note: 'This policy filters the messages that a subscriber receives.' In the 'Redrive policy (dead-letter queue) - optional' section, there is a note: 'Send undeliverable messages to a dead-letter queue.' The 'Redrive policy (dead-letter queue)' dropdown has 'Disabled' selected. At the bottom right, there are 'Cancel' and 'Create subscription' buttons, with 'Create subscription' being highlighted by a red box.

Amazon SNS > Subscriptions > Create subscription

## Create subscription

### Details

Topic ARN  
arn:aws:sns:us-east-1:680439567829:Budget\_Topic

Protocol  
The type of endpoint to subscribe

Email

Endpoint  
An email address that can receive notifications from Amazon SNS.

someone@provider.com

ⓘ After your subscription is created, you must confirm it. [Info](#)

► Subscription filter policy - *optional*  
This policy filters the messages that a subscriber receives. [Info](#)

▼ Redrive policy (dead-letter queue) - *optional*  
Send undeliverable messages to a dead-letter queue. [Info](#)

Redrive policy (dead-letter queue)  
 Disabled  
 Enabled

Cancel **Create subscription**

Here, we see that the subscription has been created.

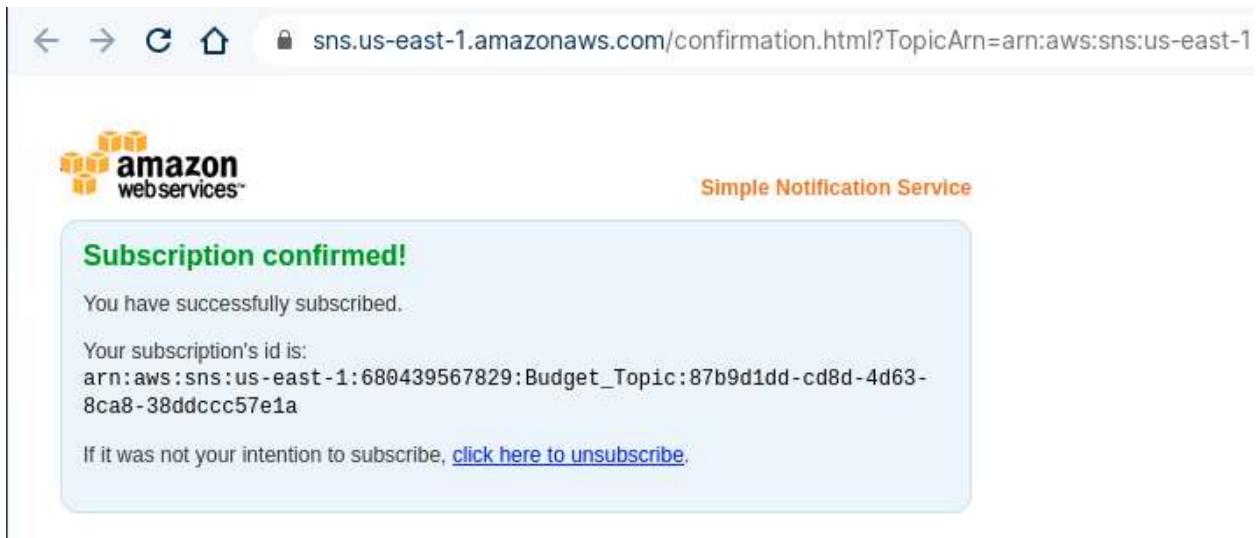
The screenshot shows the Amazon SNS Subscriptions page. On the left, there's a navigation sidebar with options like Dashboard, Topics, Subscriptions (which is highlighted), and Mobile (Push notifications, Text messaging (SMS), Origination numbers). The main content area shows a success message: "Subscription to Budget\_Topic created successfully. The ARN of the subscription is arn:aws:sns:us-east-1:680435567829:Budget\_Topic:f1773368-4401-48e7-80bf-ee87334045d2." Below this, the breadcrumb navigation shows: Amazon SNS > Topics > Budget\_Topic > Subscription: f1773368-4401-48e7-80bf-ee87334045d2. The main title is "Subscription: f1773368-4401-48e7-80bf-ee87334045d2". The "Details" section shows the ARN (arn:aws:sns:us-east-1:680435567829:Budget\_Topic:f1773368-4401-48e7-80bf-ee87334045d2), Status (Pending confirmation), Endpoint (someone@provider.com), and Topic (Budget\_Topic). Below the details, there are tabs for "Subscription filter policy" and "Redrive policy (dead-letter queue)". The "Subscription filter policy" tab is selected, showing a message: "No filter policy configured for this subscription. To apply a filter policy, edit this subscription." There's also an "Edit" button. At the bottom right of the main content area is a vertical scroll bar.

From the left-hand panel, click **Subscriptions**. You will notice your email under the heading **Endpoint**. The Status will be “pending confirmation”.

The screenshot shows the Amazon SNS Subscriptions page. The left sidebar is identical to the previous one, with the "Subscriptions" option highlighted by a red box. The main content area shows a table titled "Subscriptions (3)". The table has columns for ID, Endpoint, Status, Protocol, and Topic. One row is visible: ID is f1773368-4401-48e7-80bf-ee87334045d2, Endpoint is someone@provider.com, Status is Pending confirmation, Protocol is EMAIL, and Topic is Budget\_Topic. At the top of the table, there are buttons for Edit, Delete, Request confirmation, Confirm subscription, and Create subscription. A search bar is also present at the top of the table. The bottom right corner of the main content area has a vertical scroll bar.

You will receive an email, most likely with the subject line “AWS Notification – Subscription Confirmation”. Open the email and click on the “Confirm Submission” link.

You will be greeted by this screen in your browser:



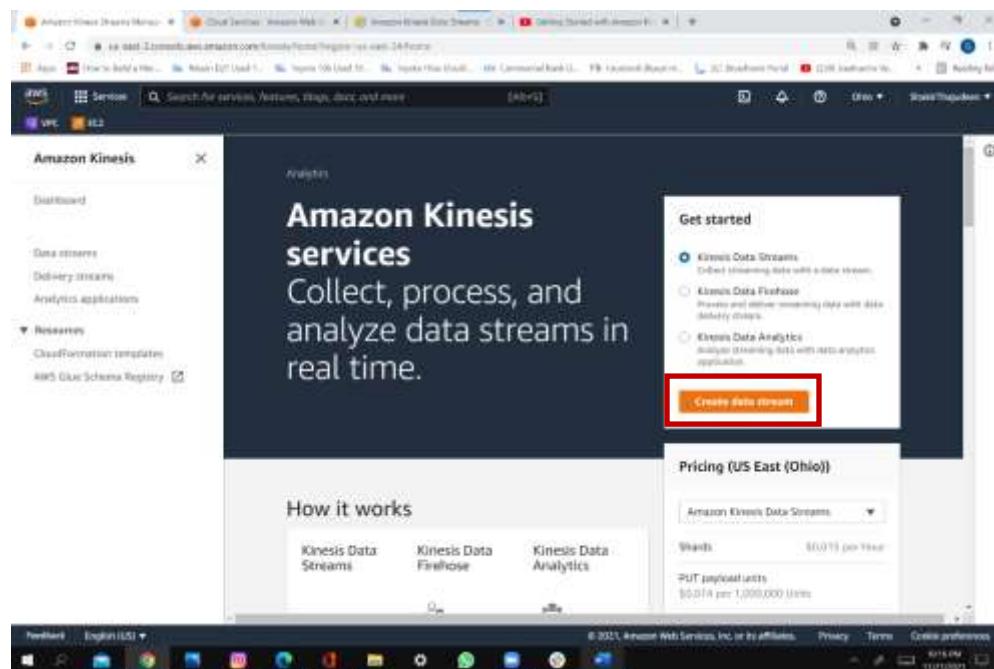
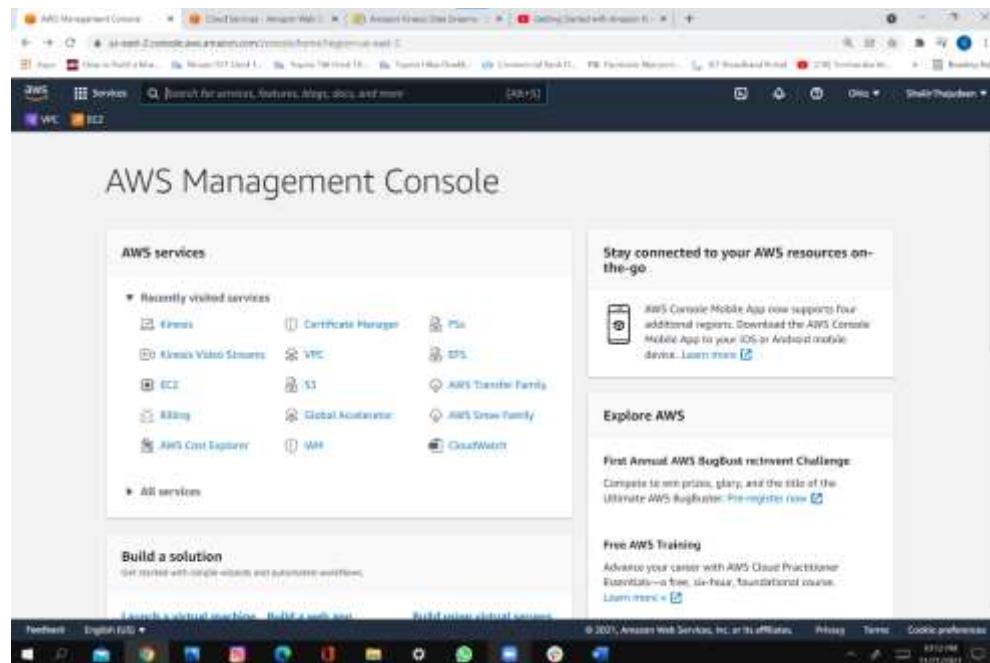
Back in the console, repeat steps 1 and 2 if you are not on the Amazon SNS dashboard. On the Amazon SNS dashboard, click **Subscriptions**.

You will notice that your Status has changed from “pending confirmation” to “confirmed”.

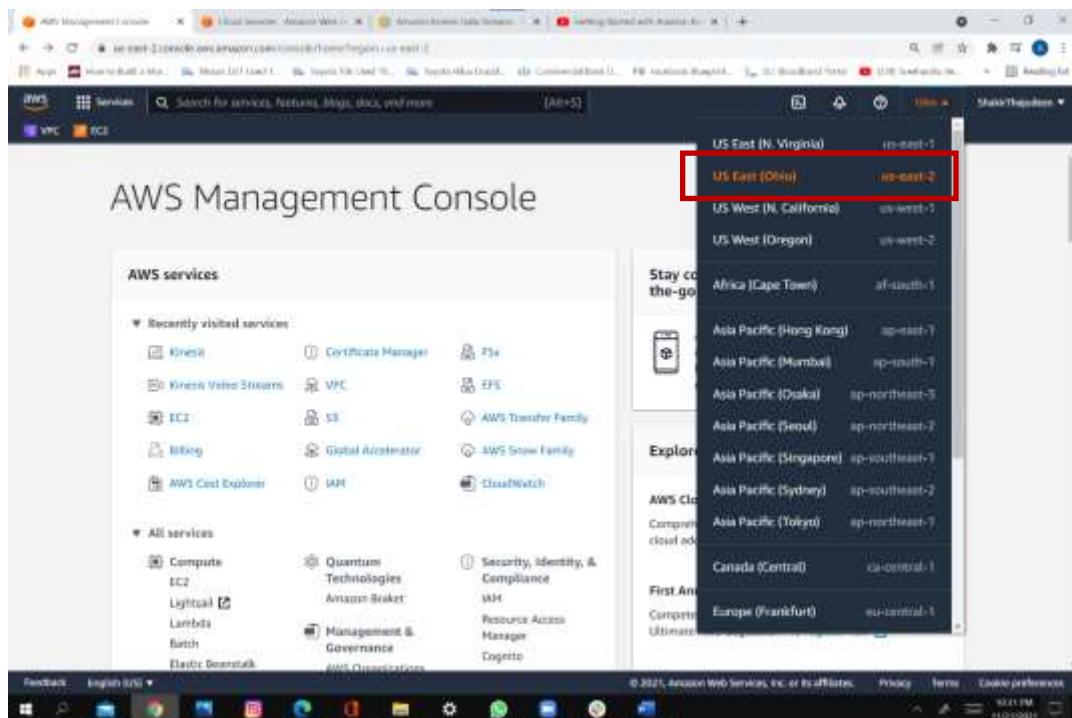
## Lab: Set Up Amazon Kinesis Streams

In this lab we will create a Kinesis Data Stream. **Here we are using the Kinesis Data Streams console.**

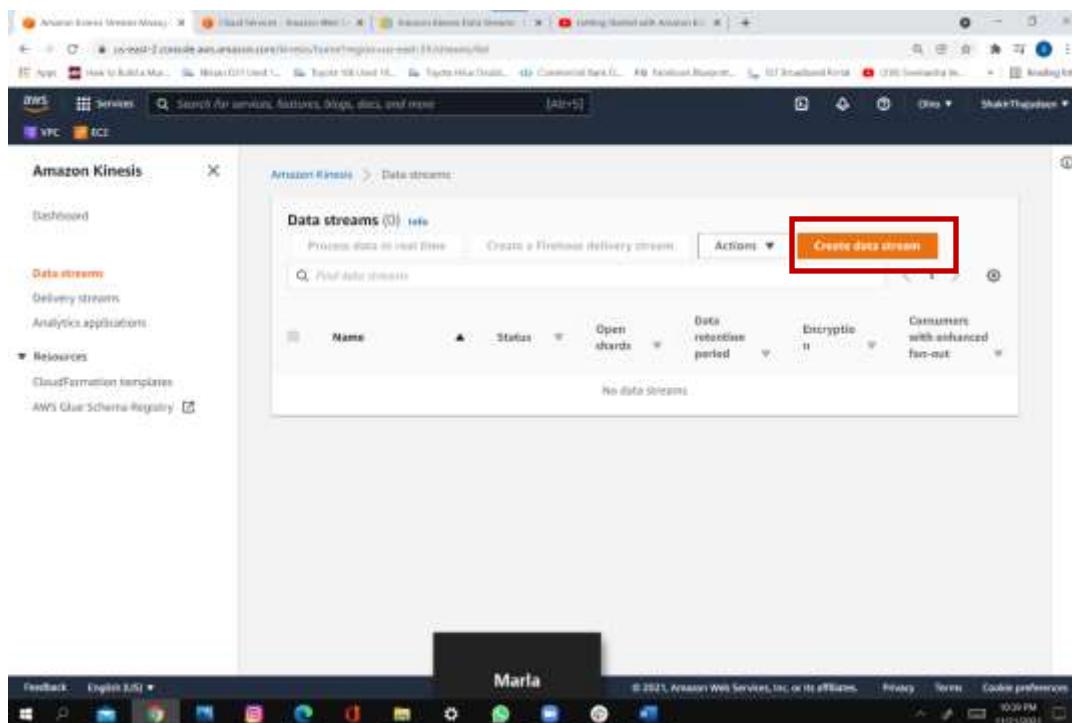
Select the AWS management console and then select the kinesis data stream console.



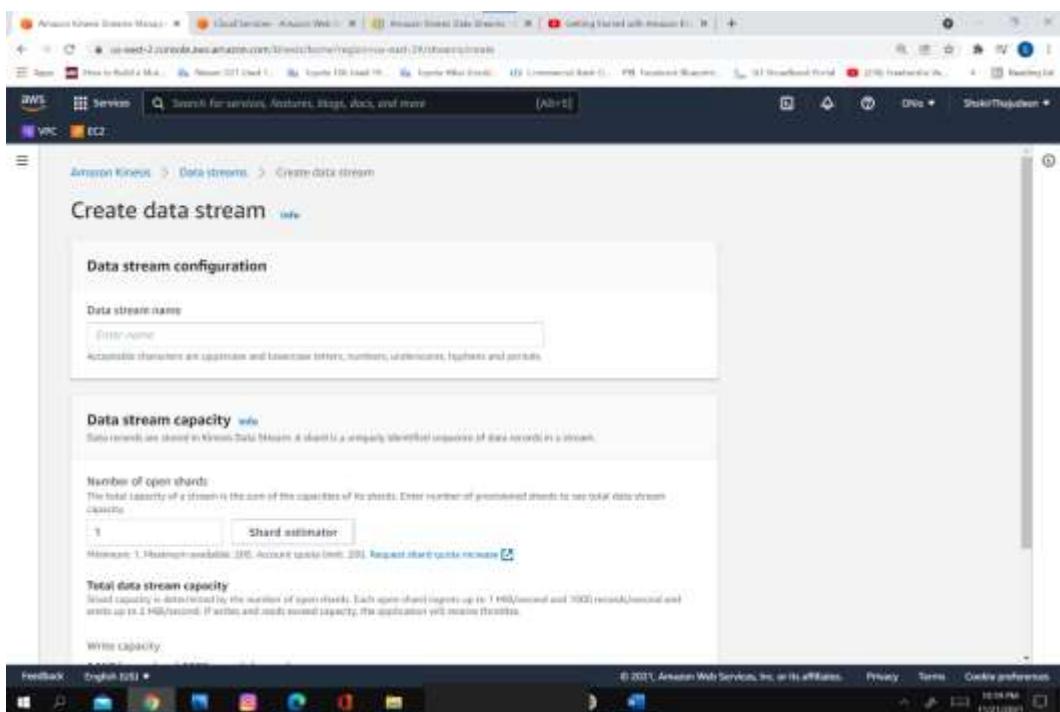
Select the region from the navigation bar.



Click create data stream.

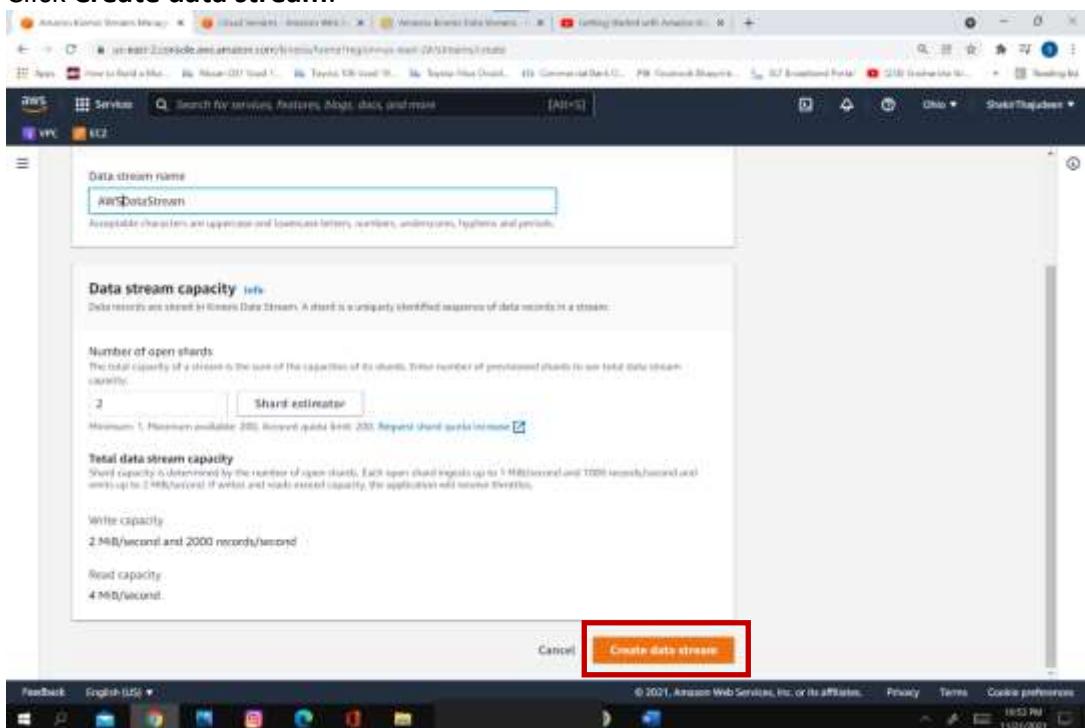


## Create data stream.



Enter stream name and the number of shards.

## Click Create data stream.



The screenshot shows the AWS Kinesis Data Streams console. A modal window titled "Creating data stream AWSDataStream" is open, indicating that creating a new data stream can take up to 10 minutes. The main page displays the "AWSDataStream" data stream summary, which includes the status (Creating), data retention period (1 day), ARN (arn:aws:kinesis:us-east-2:308984986296:stream/AWSDataStream), and creation time (November 21, 2021, 22:54 GMT+5:30). Below the summary, there are tabs for Applications, Monitoring, Configuration, and Enhanced fan-out (0). The "Producers" section shows options for Amazon Kinesis Agent, AWS SDK, and Amazon Kinesis Producer Library (KPL).

When created, status will change to active.

The screenshot shows the AWS Kinesis Data Streams console after the data stream has been successfully created. The modal window now displays a green success message: "Data stream AWSDataStream successfully created." The main page shows the "AWSDataStream" data stream summary with the status changed to Active. The other details remain the same: data retention period (1 day), ARN (arn:aws:kinesis:us-east-2:308984986296:stream/AWSDataStream), and creation time (November 21, 2021, 22:54 GMT+5:30). The "Producers" section still lists the three options: Amazon Kinesis Agent, AWS SDK, and Amazon Kinesis Producer Library (KPL).

The stream details page displays a summary of your stream configuration.

This screenshot shows the AWS Kinesis Data Stream details page. The main content area is titled "AWSDataStream". It features a "Data stream summary" card with the following information:

Status	Data retention period	ARN	Creation time
Active	1 day	arn:aws:kinesis:us-east-2:508584986296:stream/AWSDataStream	November 21, 2021, 22:54 GMT+5:30

Below this, there are tabs for "Applications", "Monitoring", "Configuration", and "Enhanced fan-out (0)". The "Applications" tab is selected. Under "Producers", there are three options: "Amazon Kinesis Agent", "AWS SDK", and "Amazon Kinesis Producer Library (KPL)". Each option has a "View in GitHub" button.

By selecting the tabs, you can see the Monitoring, Configuration, and Enhanced fan-out.

This screenshot shows the same AWS Kinesis Data Stream details page, but the "Monitoring" tab is selected. The main content area is titled "Stream metrics". It displays four monitoring charts:

- "Get records - sum (Bytes)": Shows the total number of bytes processed. The chart indicates a peak around 16:00 UTC on November 21, 2021.
- "Get records Iterator age - maximum (Milliseconds)": Shows the maximum iterator age in milliseconds. The chart indicates a peak around 16:00 UTC on November 21, 2021.
- "Get records latency - average (Milliseconds)": Shows the average latency in milliseconds. The chart indicates a peak around 16:00 UTC on November 21, 2021.
- "Get records - sum (Count)": Shows the total number of records processed. The chart indicates a peak around 16:00 UTC on November 21, 2021.

At the top of the monitoring section, there are buttons for "1h", "1d", "1w", "1m", "Custom", and "Add to dashboard".

The screenshot shows the AWS Kinesis Data Streams Configuration page for a stream named "aws-test-2". The "Configuration" tab is selected. Under "Data stream capacity", it shows 2 open shards. Under "Tags - optional", there are no tags associated with this stream. Under "Encryption", it says "Server-side encryption" is disabled. The "Consumers using enhanced fan-out" section is expanded, showing one consumer named "aws-test-2-consumer" with the ARN "arn:aws:kinesis:us-east-1:984061290000:aws-test-2:aws-test-2-consumer". A note states: "Consumers using enhanced fan-out must receive less than one 1 MByte/s of total throughput per shard. Using enhanced fan-out is optional and incurs additional cost." A "Configure enhanced fan-out" button is present.

## **Chapter 14 Passing the AWS Exam**

### **The AWS Certified Solutions Architect Exams**

AWS exams can be quite challenging, as there is a lot of material covered in the exams. It is our experience that these exams can cover an incredibly wide range of topics. Therefore, we advise strong preparation, so you are in the best position to answer challenging questions. There are some key elements of AWS questions that make them challenging to answer, including the following:

- Questions can be extremely wordy and challenging to understand. Don't be surprised if you need to read the questions two or three times each to understand the question.
- There may not be a correct answer among the options presented in the question. Choose what feels like the best answer.
- The questions frequently do not provide sufficient information to answer them without a lot of guessing and interpretation.
- Don't overthink the questions. If you have an extensive IT background, when you read the questions, you will see multiple options that could be right because there are many ways to accomplish the same goals. For the exam, forget past experiences and remember the AWS way. It is an AWS test, so think in terms of AWS.
- Sometimes the answers to the questions differ by only one word, so read very carefully.
- Sometimes AWS questions have *NOT* in them. For example, *Which one of these options is not required under these circumstances?* Read carefully so you don't miss the *NOT*.
- Don't spend too much time on each question, as you can go back to them and may find the answer elsewhere on the exam.

### **Recommendations**

We have several recommendations to help you pass the exam.

Read this book in its entirety. We spent a lot of time putting the materials in a short and easy-to-read format. Since we wrote for readability, even small sections of the book may contain a lot of information.

Look very carefully at the diagrams contained in this book. We designed those diagrams to help explain AWS concepts.

Read the AWS white papers. In our experience, a lot of AWS questions are taken from information contained in AWS white papers.

Use practice tests. We feel that knowledge is only part of passing the AWS exams. A lot of passing the AWS exam is learning to read and understand the complexity of AWS questions. Practice tests are a great way to assess your knowledge, reinforce tricky concepts, and get you

used to the way AWS asks questions. When you can score 95 percent or better on a practice test, we recommend scheduling the exam.

Avoid brain dumps. We strongly advise against using services that claim to have actual exam questions with answers. First, it is cheating and unethical. Second, when we find AWS questions on the internet, often the answers provided are incorrect. Additionally, if AWS suspects someone of cheating, AWS can pull any certifications they receive.

Don't cram the day of the exam. If you have prepared properly, you will have the tools necessary to pass the exam. We believe that for many, the biggest challenge is actually reading and understanding the questions. If you are tired from cramming, the questions may be challenging, if not impossible, to understand.

### **The Day Before the Exam**

The day before the exam, make every effort to get a good night's sleep so you will have the energy and concentration for a three-hour exam. Eat healthy foods the day before and the day of the exam so you will be at your best. Avoid alcohol or any substance that can affect thinking or judgment, unless prescribed by your physician for a health condition.

### **The Day of the Exam**

We recommend taking it easy on the day of the exam. As we have previously stated, the AWS questions are very challenging to understand, so it's best to arrive feeling refreshed and not fatigued. Arrive early for the exam, whether it's online or in person. Online exams can take time to set up, as there are photos to be taken and many other setup components. This can take thirty minutes or more in certain situations. When taking an in-person exam, there can be traffic problems, parking problems, or tech problems, so be early so you don't lose valuable exam time. Remember to have a valid photo ID when you take the exam.

### **Thank You**

Thank you for reading this book. We are excited for your journey into the world of cloud computing. We are always excited when our students pass a new certification exam. Please let us know about your success by sending an email to [elitetechcareers@gmail.com](mailto:elitetechcareers@gmail.com).

## **Practice Exam**

Below is a sample practice exam. Please note that not all questions are grammatically perfect. This is intentional to make the questions feel more like the actual exam. As stated in the book, we strongly recommend purchasing additional practice exams and scheduling your exam when you can consistently score above 95 percent.

1. An organization has an application in their on-premises data center that stores multiple 5GB files per day in S3. Recently many of these uploads have been failing. The customer's data center is geographically close to the S3 region where they store their data. What can the organization do to increase the reliability of data transfers to AWS without incurring substantial costs?
  - A) Upload data to S3 using transfer acceleration
  - B) Upload data as part of a multipart upload
  - C) Upload data to glacier and then copy to S3
  - D) Upgrade to a faster connection to the internet
2. An organization has users who upload a large number of files (each file is about 30MB) each day to S3. Recently, many of these uploads have been very slow. The organization's employees are spread throughout the world. What can the organization do to increase the performance of these transfers to S3?
  - A) Upload data to S3 using transfer acceleration
  - B) Upload data as part of a multipart upload
  - C) Upload data to Glacier and then copy to S3
  - D) Upgrade to a faster connection to the internet
3. You have deployed a three-tier architecture in a VPC with a CIDR block of 172.16.1.0/28. The initial deployment has two web servers, two application servers, two database servers, and a custom server deployed on an EC2 instance. All web, application servers, and database servers are spread across two availability zones. Additionally, there is an

ELB and DNS using Route 53. Demand for the application grows, and autoscaling is not able to keep up with demand, as autoscaling stops after adding two additional servers.

Why did autoscaling stop adding instances? Choose two:

- A) AWS reserves the first four and last IP addresses, so there are not enough addresses to launch additional instances.
  - B) There should be 15 usable addresses in a /28 subnet, so there must be a configuration error.
  - C) Autoscaling is configured improperly.
  - D) The customer needs a larger subnet, i.e. a /27 instead of a /28.
4. When using IAM, a group is regarded as a:
- A) Collection of AWS accounts
  - B) Collection of AWS users
  - C) Collection of computing instances
  - D) Link between a database and a compute instance
5. You have set up an autoscaling policy to scale in and out. You would like to control which instances are stopped first. How would you configure this?
- A) IAM Role
  - B) A termination policy
  - C) Route 53
  - D) DynamoDB
6. What are characteristics of VPC subnets? Choose 3:
- A) Each subnet maps to a single availability zone.
  - B) Subnets are spread across availability zones.

- C) Instances in a private subnet can access the internet if they have an elastic IP.
  - D) The smallest subnet on AWS is a /28.
  - E) With the default configuration all subnets can route between each other in a VPC.
7. In a CloudFormation template, each identified resource includes the following:
- A) An operating system and AMI
  - B) A dedicated host and hypervisor
  - C) Logical ID, resource type, and resource properties
  - D) Physical ID, resource type, and resource properties
8. Every time you attempt to delete an SSL certificate from the IAM certificate store, you keep getting the error “Certificate: <certificate-id> is being used by CloudFront”. What is the most likely reason for this error?
- A) SSL certificates cannot be deleted.
  - B) You do not have sufficient IAM permissions.
  - C) CloudFront is not set up properly.
  - D) Prior to deleting the SSL certificate, its necessary to rotate SSL certificates or revert to the default CloudFront certificate.
9. You plan on launching a new product. There is tremendous buzz and enthusiasm around the product launch, but you don't know exactly the demand. Orders will be sent to the database, so it's critical that writes to the database will not be lost. What is the best way to be sure orders are not lost when being written to the database?
- A) Use a Microsoft SQL server cluster.
  - B) Use DynamoDB with the max write capacity.

- C) Use an Amazon Simple Queue Service (SQS) to store orders until written to the database.
  - D) Add additional read replicas.
10. An organization wants autoscaling to scale out at 65 percent CPU utilization and scale in at 35 percent. How can the organization make sure this occurs?
- A) Use auto-scaling with the default policy.
  - B) Use autoscaling with a policy.
  - C) Use CloudWatch alarms to send an SNS message to autoscale.
  - D) It is not possible to scale at these CPU levels.
11. Which of the following EBS volume types is ideal for applications with light or burst I/O requirements?
- A) Provisioned IOPS
  - B) EBS General Purpose SSD (gp2)
  - C) EBS Throughput Optimized HDD (st1)
  - D) EBS Cold HDD (sc1)
12. Which of the following EBS volume types is ideal for applications requiring the lowest latency possible?
- A) Provisioned IOPS
  - B) EBS General Purpose SSD (gp2)
  - C) EBS Throughput Optimized HDD (st1)
  - D) EBS Cold HDD (sc1)
13. Your company is getting ready to make a major public announcement about a highly anticipated new product. The website is running on EC2 instances deployed across

multiple Availability Zones with a Multi-AZ RDS MySQL Extra Large DB Instance. There are a large number of read and writes on the database. After examination you discover that there is read contention on RDS MySQL. How can you best scale in this environment?

- A) Deploy ElastiCache in-memory cache running in each availability zone.
  - B) Add an SQS queue in front of the RDS MySQL database.
  - C) Increase the RDS MySQL instance size and implement provisioned IOPS.
  - D) Add an RDS MySQL read replica in each availability zone.
14. An organization has a requirement for the highest-throughput and lowest-latency storage option. The organization is willing to trade redundancy for performance. What is the best RAID option for this situation?
- A) Raid 0
  - B) Raid 1
  - C) Raid 5
  - D) Raid 10
15. An organization has a requirement for the highest-throughput and lowest-latency storage option with complete redundancy. What is the best RAID option for this situation?
- A) Raid 0
  - B) Raid 1
  - C) Raid 5
  - D) Raid 10
16. An organization requires a solution that provides complete redundancy. Speed is not a concern. What is the best RAID option?
- A) Raid 0

- B) Raid 1
  - C) Raid 5
  - D) Raid 10
17. Your company is developing a next-generation wearable device that collects health information to assist individuals with adopting healthy lifestyles. The sensor will push 25 KB of health data in JSON format every 2 seconds. The data should be processed and analyzed, and information should be sent to the individual's primary care provider.
- The application must provide the ability for real-time analytics of the inbound health data. The health data must be highly durable. The results of the analytic processing should persist for data mining.
- Which architecture outlined below will meet the initial requirements for the collection platform?
- A) Use S3 to collect the inbound sensor data analyze the data with Amazon Athena.
  - B) Use Amazon Kinesis to collect the inbound sensor data, analyze the data with Kinesis clients, and save the results to a Redshift cluster using EMR.
  - C) Send data to SNS to collect the inbound sensor data and save the results to AWS RDS Multi-AZ.
  - D) Send the data to SQS, which then sends to DynamoDB.
18. A new reality game show is being created. During the show users will vote for their favorite contestant. It is expected that millions of users will be voting. The votes must be collected into a durable, scalable, and highly available location. Which service should you use?
- A) Amazon DynamoDB
  - B) Amazon Redshift
  - C) Microsoft SQL Server
  - D) AWS S3

19. You are tasked with creating a solution to analyze a customer's clickstream data on a website to analyze user behavior. The analysis must provide the sequence of pages that are clicked by websites users. This data will be used in real time to optimize the website's performance in terms of page stickiness and advertising click-through rates. Which is the best option to capture and analyze user behavior in real time?

- A) Send web clicks data to Amazon S3, and then analyze and analyze behavior using Amazon Athena.
- B) Push web clicks data to Amazon Kinesis and analyze behavior using Kinesis workers.
- C) Write web clicks directly to DynamoDB.
- D) Write web clicks directly to Amazon RDS for Oracle.

20. An application provides data transformation services. Data to be transformed is uploaded to Amazon S3 and then transformed by a fleet of spot EC2 instances. VIP customers should have their files transformed before other customers. How should you implement a system that services VIP customers first?

- A) This cannot be performed, as the apposition process messages in the order they are received.
- B) Use an ELB to distribute VIP traffic first and then generic traffic to the spot fleet of transformation instances.
- C) Set up two SQS queues, priority queue for VIP customers and a second queue with default priority for everyone else. Have the transformation instances first poll the high-priority queue; if there is no message, then poll the default priority queue.
- D) Use SNS to send a message to administrators to manually send VIP customers data for immediate transformation.

21. An organization is planning on setting up a bastion host to help manage systems on their VPC. The bastion host must be reachable from all internet addresses. The bastion host must also be able to access the internal network and should only be open to SSH traffic from a small CIDR range of addresses. How can the bastion host be configured for this purpose?

- A) This cannot be performed, as the host is on a public subnet.
  - B) Create two network interfaces on two different subnets. Assign security groups to allow external traffic on the public interface and SSH traffic on the internal network interface.
  - C) Create two network interfaces with the same subnets. Assign security groups to allow external traffic on the public interface and SSH traffic on the internal network interface.
  - D) Separate the services. Put the web server on an EC2 instance and set up a second server for SSH traffic.
22. \_\_\_\_\_ pricing offers a significant discount over on-demand pricing. This pricing approach works well for mission-critical applications with known capacity utilization and known duration of use.
- A) Discount voucher
  - B) Reserved instance
  - C) AWS coupon code
  - D) Spot instance
23. An organization's security policy requires encryption of sensitive data at rest. The data is stored on an EBS volume which is attached to an EC2 instance. Which options would facilitate encrypting your data at rest? (Choose 3):
- A) Leverage third-party volume encryption tools.
  - B) Move data from EBS to S3.
  - C) Encrypt data prior to storing on EBS.
  - D) Encrypt data using native data encryption drivers at the file system level.
  - E) Unnecessary, as all data on AWS is encrypted.

24. What does the PollForTask action perform when it's called by a task runner in AWS Data Pipeline?

- A) It retrieves the pipeline definition.
- B) It sends an SNS message to AWS administrators.
- C) It sends the data to the next application in the task.
- D) It performs the next task to perform from AWS Data Pipeline.

25. Which of the following are customer responsibilities under the shared security model? Choose 3.

- A) Security groups
- B) ACLs
- C) Patch management of the serverless operating system
- D) IAM credentials
- E) Managing the underlying hardware of an EC2 instance

26. An organization has three separate divisions (VPCs), and they are main, autos, and auto parts. The main organization needs access to auto and auto parts. How can the main organization access the VPCs of autos and auto parts?

- A) Set up VPC peering between main and autos and auto parts.
- B) Open NACLs to allow for full communication.
- C) Make sure the security groups allow for the CIDR ranges of all VPCs.
- D) This is not possible, as VPCs cannot communicate with each other.

27. A company has 500 TB of business-critical data. The company had a fire at their facility and is in immediate need to move their data center to the AWS cloud. The company needs to perform this within 7 business days. The company has a 1-Gig direct

connection to AWS, which is running at near full capacity. How can you get the system fully operational within the short timeframe?

- A) Request multiple Snowball devices from AWS. Load data on the Snowball device. Have AWS download data to an S3 bucket.
- B) Order a 10GB direct connection and send over that link.
- C) Upload to S3 over the existing 1GB internet connection with multipart uploads.
- D) Use the AWS import/export service. Load data on the hard drives. Have AWS download data to an S3 bucket.

28. An organization is using ElastiCache in front of Amazon RDS database which has four read replicas deployed. The database CPU is at 65 percent with the ElastiCache and cannot meet current capacity if the ElastiCache Fails. The server has very limited write use and is mostly limited by read contention. What is a solution to mitigate the impact of an ElastiCache failure?

- A) Spread memory and capacity over a smaller number of larger cache nodes.
- B) Spread memory and capacity over a larger number of smaller cache nodes.
- C) Implement an SQS queue to assist with write capacity.
  - a. Use AWS SNS messenger to alert team of cache failures.

29. What indicates that an object is successfully stored when put in S3?

- A) An http 404 code is received.
- B) An http 300 code is received.
- C) Cloud watch logs show put was successful.
- D) An http 200 code is received, along with an MD5 hash.

30. What is the maximum number of VPCs per region?

- A) 10

- B) 50
- C) 100
- D) 5

31. S3 bucket policies are written in what language?

- A) JavaScript
- B) C++
- C) JSON
- D) Python

32. A global organization is hosting a website on S3. The company is experiencing large data charges for cross-region sharing from the S3 bucket. What changes can be made to reduce costs?

- A) VPC peering
- B) S3 cross-region replication
- C) Move the website off S3 and onto an EC2 instance
- D) CloudHub

33. An organization has been storing their data on instance storage. The server was patched for security vulnerability, and when it was rebooted, all data stored was gone. Why did this happen?

- A) Malware infection
- B) Not enough information is provided to troubleshoot
- C) Instance storage is deleted upon termination or reboot
- D) None of the above

34. An organization is using an AWS RDS database. The database is currently running on EBS general purpose storage. At times read and write latency is too high for the organization's needs. How can this be easily remedied?

- A) Upgrade the EBS volume to provisioned OPS
- B) Change storage type to EBS throughput optimized
- C) Change storage location to an EFS volume
- D) Change to high-speed instance storage

35. Relational databases follow the BASE model (Basically Available, Soft State, and Eventually Consistent).

- A) True
- B) False

36. An organization has noticed the CPU on their RDS database is consistently at 85%. When looking at the database, there is heavy read activity from frequent SQL queries from the finance department. What can the organization do to improve the performance and scalability of the database? Choose 2:

- A) Add a read replica and point the finance department's SQL queries to the read replica
- B) Add an ElastiCache to reduce read contention for frequently accessed information
- C) Add an SQS queue to reduce read contention
- D) Set up Multi-AZ for the RDS database

37. An organization has set up a high-availability database architecture using a Multi-AZ environment. If the primary database fails, which of the following will cause the database to failover to the backup database? Choose 4:

- A) The primary database instance fails.

- B) There is an outage in an availability zone.
- C) The database instance type is changed.
- D) The primary database is under maintenance (i.e., patching an operating system).
- E) The database is busy, with a CPU utilization of 90%.
38. An organization has a web server in a private subnet that is connected to the internet with an NAT gateway. External users cannot access the web server. What changes can the organization make to have this server reachable from the internet? Choose 3:
- A) Put the web server on a public subnet.
- B) Use a NAT instance with an internet gateway.
- C) Put an ELB in a public subnet and keep the web server in a private subnet.
- D) There must be a configuration error, as the server can ping addresses on the public internet.
39. An organization has noticed that when users connect to S3 their traffic is traversing the public internet. The organization is experiencing low performance and high internet costs. What can the organization do to increase the performance, privacy, and scalability of the solution?
- A) Create an endpoint for S3 and connect to the endpoint.
- B) S3 always uses the internet, so increase the internet connection speed and encrypt with a VPN.
- C) Set a routing policy to have the organization connect to S3 via the AWS network.
- D) Set up VPC peering to S3.
40. An organization has three VPCs, VPC A, VPC B, and VPC C. VPC A is peered with VPC B and VPC C. VPC A can reach VPC B and C. But VPC B and VPC C cannot communicate with each other. Why can't these VPCs communicate with each other?
- A) A firewall is blocking connectivity.

- B) There is a misconfigured ACL policy.
  - C) A configuration error occurred.
  - D) VPC peering is not transitive, and this is normal.
41. An organization has set up an NACL to increase the security of their VPC. The organization wants to allow web traffic, TCP Port 80, into the subnet where the web servers reside. They apply the NACL, but no web requests are making it to the web server. Why could this be happening? The ACL can be seen below:
- Rule 110 – Deny all traffic  
Rule 110 – Inbound Allow TCP Port 80 Source 192.168.1.1  
Rule 120 – Outbound Allow TCP Port 80 Source 192.168.1.1
- A) The ACL is properly configured. There must be another problem.
  - B) The order is incorrect, as all traffic is denied prior to being permitted by rules 110 and 120.
  - C) It is not possible to have a deny statement in an ACL.
  - D) The security group on the web server is improperly configured.
42. With NACL, both an inbound and outbound policy are necessary. Why are security groups only configured for an inbound policy?
- A) Security groups are stateful, so they allow outbound return traffic.
  - B) Security groups require both an inbound and outbound policy.
  - C) It's an inconsistency in the AWS Cloud.
  - D) Security groups don't allow outbound traffic.
43. If an organization is looking to maximize performance for specific applications but doesn't need high availability for this application, what would be the best option in terms of placement groups?

- A) Spread placement group.
  - B) Partition placement group.
  - C) Cluster placement group.
  - D) Placement groups don't affect performance.
44. Your organization requires encryption of all data at rest. What can you do to encrypt and protect data on the EBS volume that is mounted by an EC2 instance? Which of these options would allow you to encrypt your data at rest? Choose 2:
- A) Leverage third party volume encryption tools
  - B) Use SSH to encrypt data
  - C) Encrypt data prior to storing it on EBS
  - D) Use an EFS instance instead of EBS
45. You are designing internet connectivity for your organization's VPC. The organization has web servers with private addresses that must be reachable from the internet. The web servers must be highly available. What can you use to ensure the web servers are highly available and reachable from the internet? Choose 2:
- A) Configure an NAT instance in your VPC. Place web servers behind the NAT instance.
  - B) Configure CloudFront and place it in front of your web servers. Put CloudFront on a public subnet.
  - C) Assign EIPs to all web servers. Configure a Route 53 failover policy attached to the EIPs. Use Route 53 with health checks.
  - D) Put web servers in multiple availability zones. Create a DNS A record for all EIPs.
  - E) Put all web servers behind an ELB. Configure a Route 53 CNAME record that points to the ELB DNS name.

46. An organization is looking to implement an intrusion detection and prevention system into their VPC. This platform should have the ability to scale to meet the needs of a global enterprise organization. How should they design their VPC to achieve scalable IDS/IPS?

- A) Call AWS support and ask them to put a switch port in SPAN mode and attach a packet sniffer to the SPAN.
- B) Set up a proxy server and send all internet requests through the proxy server.
- C) Deploy IDS/IPS onto the organization's firewall.
- D) Put an agent on all servers that sends network traffic to the IDS/IPS for inspection and management.

47. An organization has been using a domain name for their business. The company hired a marketing firm that suggested they change their domain name to something more indicative of their brand. What can the organization do to use a new domain name while not losing its current customers using the old domain name?

- A) Migrate to the new domain. Set up a DNS CNAME record that redirects current users to the new URL.
- B) Set up a new website with the domain name and keep the old website operational.
- C) Migrate to the new domain. Set up a DNS A record that redirects current users to the new URL.
- D) Migrate to the new domain. Set up a DNS NS record that redirects current users to the new URL.

48. An organization is looking to use a load balancer to increase performance and availability of their website. The organization has tens of millions of customers and is looking for the highest speed load balancer available on the AWS platform. Which is the best option when speed is of utmost importance?

- A) Elastic Load Balancer – Application
- B) Elastic Load Balancer – Network
- C) Classic Load Balancer – Application

- D) Route 53 with policy routing instead of a load balancer
49. An organization has configured an IAM role for a user. The user cannot seem to access any information on the VPC. What could be wrong in this situation?
- A) The IAM role was configured in a manner that blocks access to all resources.
  - B) IAM roles are not for users. They are created for systems to access other systems in a VPC.
  - C) The IAM role was not properly applied to the user.
  - D) There is not enough information provided to answer this question.
50. Using the IAM concept of AAA (Authentication, Authorization, and Accounting), which component determines whether a user is allowed access to a resource?
- A) Authentication
  - B) Authorization
  - C) Accounting
51. An organization desires to connect its on-premise Microsoft Active Directory services with AWS for easier IAM management. How can the organization federate to the organization's Microsoft AD servers?
- A) This is not possible.
  - B) Use the AWS AD migration tool.
  - C) Build a connection using SAML 2.0.
  - D) Leverage AWS directory services and have them peer with on-premise AD servers.
52. An organization wants to give several developers access to all AWS resources except IAM. What access should the developers have to support their jobs?

A) Administrator

B) Developer

C) Power User

D) Sys Admin

53. An organization is looking for a means to store critical information such as passwords and software licenses. Which is the best and most secure option?

A) Store passwords on an encrypted database

B) Store passwords in a hidden folder in the root account of an EC2 instance

C) Store passwords and licenses in the Systems Manager Parameter Store

D) Store passwords in a spreadsheet, which is inside an encrypted folder on the CEO's computer

54. An organization is setting up a platform for managing and analyzing extremely large amounts of data. The organization is looking to use a serverless environment. What would be the easiest option to deploy and manage a big data framework for this customer?

A) Set up an EC2 instance and install Apache Hadoop

B) Use AWS EMR

C) Build a custom big data management platform and place on a Fargate container

D) Use DynamoDB

55. An organization wants to use CloudWatch to monitor memory utilization in an application with large memory demands. The organization would like updates every 1 minute. How should the organization set this up?

A) No setup is required. CloudWatch performs this service automatically.

B) Leverage CloudWatch detailed monitoring and set a CloudWatch custom metric.

- C) Leverage CloudTrail and not CloudWatch for this purpose.
- D) Set up a Lambda function that will use SNS to notify systems administrators when memory utilization goes over 80 percent.

56. Lambda functions can be set up in which of the following programming languages?

Choose 3:

- A) Node.js
- B) Visual basic
- C) Python
- D) C#
- E) C++
- F) Pascal

57. An organization wants to restrict access to information stored on S3. Which of the below options can be used for access control?

- A) ACLs
- B) IAM policies
- C) Bucket policies
- D) All of the above

58. Your organization stores a substantial amount of data on S3. The files stored on S3 are very large, most over 1 GB. You are traveling and in a part of the world with slow internet access. You only need some of the data in the file but not the entire file. What options, if any, can you use to work more efficiently?

- A) Find a location with better internet access.
- B) Use a range get.
- C) There is nothing you can do, so be patient.

- D) Use a Python script to download files overnight.
59. In order to maintain the integrity of autoscaling, autoscaling requests are signed with a hash based upon the user's private key. What hashing algorithm is used to calculate the hash?
- A) SHA-256
  - B) HMAC-SHA1
  - C) X11
  - D) X11Gost
60. AWS uses the term *elastic* for many of their services. What does the Amazon definition of elastic mean?
- A) Bursting capabilities
  - B) The ability to create instances easily
  - C) The ability to scale resources up and down with minimal challenges
  - D) The speed of deployment compared to a traditional data center
61. When restoring a database from a DB snapshot, which of the following occurs? Choose 2:
- A) A new instance is created.
  - B) It restores the original instance.
  - C) All information, including the operating system, database, and all data, is restored to the new instance.
  - D) All information, including the operating system, database, and all data is restored to the old instance.

62. An organization is creating a machine learning application on the AWS platform. Which is the best type of instance for this application?

- A) C5
- B) T3
- C) M5
- D) G3

63. When connecting to AWS via a direct connection, what routing protocol is used to exchange routing information for network layer reachability?

- A) OSPF
- B) EIGRP
- C) BGP
- D) RIP V2

64. An organization is using an application that requires physical access to the underlying hardware of the server. Which is the best type of tenancy on the AWS platform?

- A) Shared tenancy
- B) Dedicated instance
- C) Dedicated host
- D) Placement group

65. An organization that creates video games is launching a new game. They expect this application to have tens of millions of global users. What is the best database option to store game state?

- A) Microsoft SQL
- B) MySQL

- C) DynamoDB
  - D) Amazon Aurora
66. An organization is moving to the cloud. The organization is looking for a way to be more efficient and is willing to modify its current processes. The organization needs a relational database and a data warehouse. The organization has currently been using a custom-developed and problematic ETL tool to exchange information between databases and storage. What can the organization do to replace the current ETL tool?
- A) AWS Glue
  - B) DynamoDB
  - C) ElastiCache
  - D) VPC endpoint
67. An organization is looking to set up a mail server on an EC2 instance. What type of DNS record should be set up in Route 53?
- A) A record
  - B) CNAME record
  - C) MX record
  - D) NS record
68. An organization wants to set up a scalable IAM solution for mobile phones. The organization wants to authenticate via Facebook and other identity providers. What solution is recommended in this use case?
- A) Set up IPsec between the VPC and Facebook
  - B) Set up VPC peering with Facebook
  - C) Use Amazon Cognito
  - D) Use the AWS Directory service

69. When setting up an IAM policy, which statements are not required? Choose 2:

- A) Action
- B) Resource
- C) Condition
- D) Shield

70. It's often necessary to fan out messages to multiple systems for distributed workflows. Which AWS service is designed for this purpose?

- A) EMR
- B) ECS
- C) SNS
- D) EKS

71. An organization has been recently attacked by a hacker. The organization is looking for a means to find systems that do not comply with the organization's policies (operating system, patch level, security groups, etc.). What is the simplest method for the organization to find systems that do not meet organizational standards?

- A) AWS CloudWatch
- B) AWS CloudTrail
- C) AWS Config
- D) AWS CloudFront

72. An organization uses a substantial number of videos in its digital marketing campaigns. The organization would like to ensure that there is no content that could be potentially offensive to some customers. What would be the simplest means to achieve this on the AWS platform?

- A) Set up an EC2 G3 instance, with a Python script using a machine algorithm to identify suspect content
  - B) Use AWS Rekognition to identify suspect content
  - C) Use AWS Mechanical Turk to identify suspect content
  - D) Use the AWS Content Manager to identify suspect content
73. A migration from a traditional data center to the cloud can have a profound effect on the organization's technology expenses. A CFO is asking what type of effect a migration to the cloud would have on their technology costs. How would you describe the financial impact to the CFO? Choose 2 below:
- A) Cloud computing has lower capital expenses (CAPEX) and higher operational expenses (OPEX).
  - B) Cloud computing has lower capital expenses (OPEX) and higher operational expenses (CAPEX).
  - C) The total cost of ownership is likely more expensive, but there is increased business agility.
  - D) The total cost of ownership is likely less expensive, and there is increased business agility.
74. Creating a full security posture involves which of the following? Choose all that apply:
- A) NACLs
  - B) IAM
  - C) DDoS prevention
  - D) Firewalls
  - E) Linux only
75. An organization is looking for a means to sequence multiple Lambda functions. What is the simplest way to do this on the AWS platform?
- A) Lambda@Edge

B) Python script

C) Step Functions

D) Glue

## Answer Key

1. B
2. A
3. A, D
4. B
5. B
6. A, D, E
7. C
8. D
9. C
10. B
11. D
12. A
13. D
14. A
15. D
16. B
17. B
18. A
19. B
20. C
21. B
22. B
23. A, C, D
24. A
25. A, B, D
26. A
27. A
28. B
29. D
30. D
31. C
32. B
33. C
34. A
35. B
36. A, B
37. A, B, C, D
38. A, B, C
39. A
40. D
41. B
42. A

- 43. C
- 44. A, C
- 45. C, E
- 46. D
- 47. A
- 48. B
- 49. B
- 50. B
- 51. C
- 52. C
- 53. C
- 54. B
- 55. B
- 56. A, C, D,
- 57. D
- 58. B
- 59. A
- 60. C
- 61. A, C
- 62. D
- 63. C
- 64. C
- 65. C
- 66. A
- 67. C
- 68. C
- 69. C, D
- 70. C
- 71. C
- 72. B
- 73. A, D
- 74. A, B, C, D
- 75. C

References:

1. <https://aws.amazon.com/s3/>
2. <https://docs.aws.amazon.com/AmazonS3/latest/dev/using-iam-policies.html>
3. <https://learning.oreilly.com/library/view/aws-certified-solutions/9781119138556/c02.xhtml>
4. <https://docs.aws.amazon.com/kms/latest/developerguide/services-s3.html>
5. <https://docs.aws.amazon.com/AmazonS3/latest/dev/serv-side-encryption.html>
6. <https://aws.amazon.com/ebs/?ebs-whats-new.sort-by=item.additionalFields.postDateTime&ebs-whats-new.sort-order=desc>
7. <https://aws.amazon.com/ebs/volume-types/>
8. <https://www.enterprisestorageforum.com/storage-management/raid-levels.html>
9. <https://docs.aws.amazon.com/storagegateway/latest/userguide/WhatIsStorageGateway.html>
10. <https://aws.amazon.com/workdocs/?amazon-workdocs-whats-new.sort-by=item.additionalFields.postDateTime&amazon-workdocs-whats-new.sort-order=desc>
11. <https://aws.amazon.com/fsx/windows/?nc=sn&loc=2>
12. <https://learning.oreilly.com/library/view/aws-certified-solutions/9781119138556/c03.xhtml>
13. <https://www.oracle.com/database/what-is-a-relational-database/>
14. <https://www.ibm.com/cloud/learn/nosql-databases>
15. <https://www.ibm.com/cloud/learn/nosql-databases>
16. <https://aws.amazon.com/big-data/datalakes-and-analytics/what-is-a-data-lake/>

17. <https://aws.amazon.com/rds/>
18. <https://aws.amazon.com/dynamodb/>
19. <https://aws.amazon.com/nosql/>
20. <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-indexes-general.html>
21. <https://www.dummies.com/programming/big-data/hadoop/acid-versus-base-data-stores/>
22. <https://aws.amazon.com/redshift/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>
23. [https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER\\_WorkingWithAutomatedBackups.html](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_WorkingWithAutomatedBackups.html)
24. <https://aws.amazon.com/sqs/>
25. <https://aws.amazon.com/glue/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>
26. <https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html>
27. <https://www.networkworld.com/article/3239677/the-osi-model-explained-and-how-to-easily-remember-its-7-layers.html>
28. <https://tools.ietf.org/html/rfc1918>
29. <https://tools.ietf.org/html/rfc4291>
30. <https://blog.apnic.net/2020/01/14/bgp-in-2019-the-bgp-table/>
31. [https://docs.aws.amazon.com/vpc/latest/userguide/VPC\\_Internet\\_Gateway.html](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html)
32. <https://docs.aws.amazon.com/vpc/latest/userguide/egress-only-internet-gateway.html>

33. [https://docs.aws.amazon.com/vpc/latest/userguide/VPC\\_NAT\\_Instance.html](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_NAT_Instance.html)
34. <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html>
35. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html>
36. <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-endpoints.html>
37. <https://docs.aws.amazon.com/vpc/latest/userguide/vpce-interface.html>
38. <https://docs.aws.amazon.com/vpc/latest/userguide/vpce-gateway.html>
39. <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-peering.html>
40. <https://docs.aws.amazon.com/vpc/latest/peering/what-is-vpc-peering.html>
41. <https://docs.aws.amazon.com/whitepapers/latest/aws-vpc-connectivity-options/aws-vpn-cloudhub.html>
42. <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html>
43. [https://docs.aws.amazon.com/vpc/latest/userguide/VPC\\_SecurityGroups.html](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html)
44. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html>
45. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html#placement-groups-cluster>
46. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html#placement-groups-partition>
47. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html#placement-groups-spread>
48. <https://aws.amazon.com/route53/>
49. <https://www.f5.com/services/resources/glossary/load-balancer>
50. <https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.htm>

51. <https://docs.aws.amazon.com/elasticloadbalancing/latest/network/introduction.html>
52. <https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/introduction.html>
53. <https://aws.amazon.com/compliance/shared-responsibility-model/>
54. <https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#grant-least-privilege>
55. <https://aws.amazon.com/compliance/programs/>
56. <https://aws.amazon.com/iam/>
57. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html>
58. [https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_roles\\_common-scenarios\\_aws-accounts.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_common-scenarios_aws-accounts.html)
59. <https://aws.amazon.com/blogs/security/how-to-audit-cross-account-roles-using-aws-cloudtrail-and-amazon-cloudwatch-events/>
60. <https://aws.amazon.com/identity/federation/>
61. <https://aws.amazon.com/single-sign-on/>
62. <https://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html>
63. <https://aws.amazon.com/directoryservice/>
64. [https://docs.aws.amazon.com/IAM/latest/UserGuide/access\\_policies\\_managed-vs-inline.html#customer-managed-policies](https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_managed-vs-inline.html#customer-managed-policies)
65. <https://awspolicygen.s3.amazonaws.com/policygen.html>
66. <https://aws.amazon.com/organizations/getting-started/best-practices/>

67. <https://aws.amazon.com/waf/>
68. <https://aws.amazon.com/blogs/aws/aws-shield-protect-your-applications-from-ddos-attacks/>
69. <https://aws.amazon.com/servicecatalog/?aws-service-catalog.sort-by=item.additionalFields.createdDate&aws-service-catalog.sort-order=desc>
70. <https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-parameter-store.html>
71. <https://docs.aws.amazon.com/systems-manager/latest/userguide/what-is-systems-manager.html>
72. <https://aws.amazon.com/sqs/>
73. <https://aws.amazon.com/sns/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>
74. <https://aws.amazon.com/swf/>
75. <https://aws.amazon.com/emr/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>
76. <https://aws.amazon.com/kinesis/>
77. <https://www.docker.com/resources/what-container>
78. <https://aws.amazon.com/ecs/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc&ecs-blogs.sort-by=item.additionalFields.createdDate&ecs-blogs.sort-order=desc>

79. <https://aws.amazon.com/eks/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc&eks-blogs.sort-by=item.additionalFields.createdDate&eks-blogs.sort-order=desc>
80. <https://aws.amazon.com/elasticbeanstalk/>
81. [https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/viewing\\_metrics\\_with\\_cloudwatch.html](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/viewing_metrics_with_cloudwatch.html)
82. <https://aws.amazon.com/cloudwatch/>
83. <https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-user-guide.html>
84. <https://aws.amazon.com/blogs/networking-and-content-delivery/dynamic-whole-site-delivery-with-amazon-cloudfront/>
85. <https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Invalidation.html>
86. <https://learning.oreilly.com/library/view/aws-certified-solutions/9781119138556/c11.xhtml>
87. <https://aws.amazon.com/lambda/>
88. <https://aws.amazon.com/step-functions/>
89. <https://aws.amazon.com/rekognition/?blog-cards.sort-by=item.additionalFields.createdDate&blog-cards.sort-order=desc>
90. <https://docs.aws.amazon.com/cloudformation/>
91. <https://docs.aws.amazon.com/acm/latest/userguide/acm-overview.html>
92. <https://aws.amazon.com/certificate-manager/?nc=sn&loc=1>

93. <https://aws.amazon.com/blogs/aws-cost-management/getting-started-with-aws-budgets/>
94. <https://aws.amazon.com/premiumsupport/technology/trusted-advisor/>
95. <https://phoenixnap.com/blog/what-is-high-availability>