

AWS-BANGLA



Topic to be covered:

1. Introduction
2. Opening AWS Account and Free Tier
3. Getting Started
4. Cost And Budgets
5. User, Role and IAM User
6. AWS Architecture, Regions and AZ
7. Play with EC2
8. AWS Storage services
9. Database in AWS
10. ELB, ASG
11. AWS Managed Services
12. AWS Serverless
13. AWS Queue and Streams
14. Route53 and CDN
15. AWS Networking
16. Security and Monitoring
17. AWS Machine Learning
18. AWS Well-Architected
19. Project run on AWS
20. AWS Certification

1) Introduction

Cloud computing: Cloud computing provides on-demand computing resources access via the internet, like applications, servers, storage, databases, software, etc.

No need to own and maintain physical server/hardware and infrastructure.



Service Models:

- **Infrastructure as a Service (IaaS):** Provides virtualized computing resources.
- **Platform as a Service (PaaS):** Offers a platform that includes infrastructure.
- **Software as a Service (SaaS):** Delivers software applications over the internet.



Different Cloud Provider

Top Cloud Provider AND MORE



Here are key characteristics of Cloud:

On-Demand Self-Service:

Users can provision and manage computing resources as needed, without requiring human intervention from the service provider.

Broad Network Access:

Cloud services are accessible over the internet from anywhere.

Rapid Elasticity:

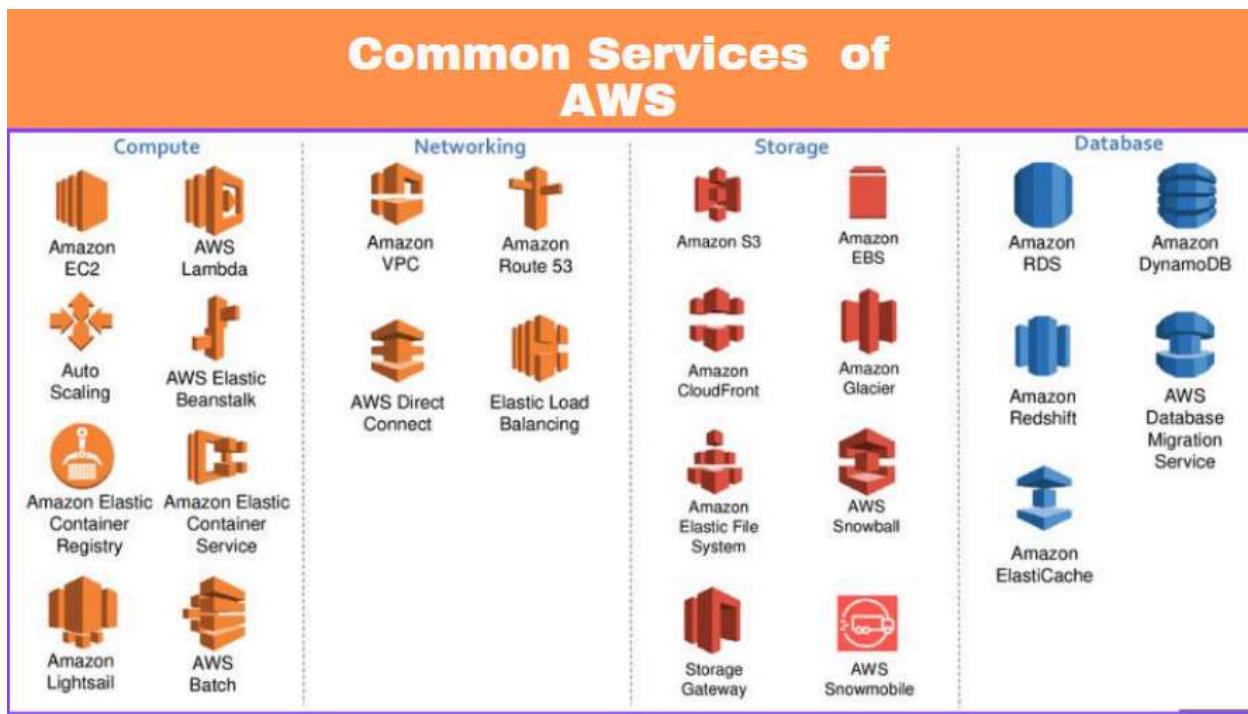
Resources can be scaled up or down quickly to accommodate changes in demand.

Measured Service:

Cloud resources are metered, and users are billed based on their usage.



AWS provides **over 200 different services** across various categories, including computing, storage, databases, machine learning, analytics, security, networking, Internet of Things (IoT), developer tools, and more.



1. Compute Services:

EC2 (Elastic Compute Cloud): Provides scalable virtual servers in the cloud, allowing users to run applications and workloads.

2. Networking:

VPC (Virtual Private Cloud): Allows users to provision a logically isolated section of the AWS Cloud.

Route53: Scalable and highly available domain name system (DNS) web service.

3. Storage and Content Delivery:

S3 (Simple Storage Service): Offers scalable object storage with high durability.

EBS (Elastic Block Store): Provides block-level storage volumes for use with EC2 instances.

Glacier: Offers low-cost, long-term archival storage.

CloudFront: Content delivery network (CDN) service for securely delivering data, videos, applications, and APIs to customers globally.

4. Databases:

RDS (Relational Database Service): Managed relational database service supporting multiple database engines like MySQL, PostgreSQL, Oracle, and Microsoft SQL Server.

DynamoDB: Fully managed NoSQL database service that scales automatically.

And More:



5. Machine Learning and Artificial Intelligence:

SageMaker: Fully managed service for building, training, and deploying machine learning models.

Polly: Text-to-speech service.

Rekognition: Image and video analysis service.

6. Security and Identity:

IAM (Identity and Access Management): Manages access to AWS services and resources.

GuardDuty: Managed threat detection service.

7. Management and Monitoring:

CloudWatch: Monitoring and observability service for AWS resources and applications.

CloudTrail: Records API calls for your account and delivers log files for audit.

8. Developer Tools:

CodePipeline: Continuous integration and continuous delivery (CI/CD) service.

CodeDeploy: Automates code deployments to various compute services.

9. Containers and Kubernetes:

ECS (Elastic Container Service): Container orchestration service.

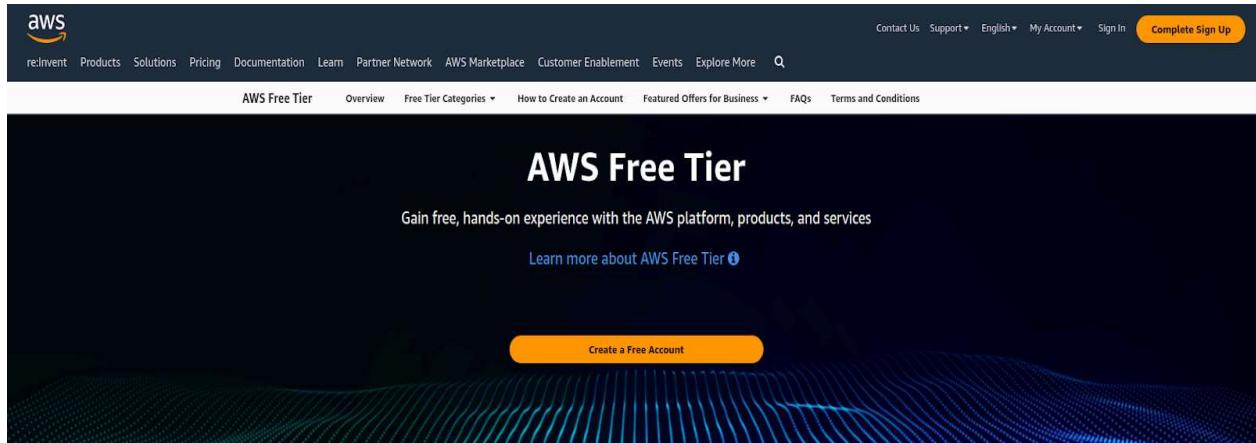
EKS (Elastic Kubernetes Service): Managed Kubernetes service.

10. Serverless Computing:

Step Functions: Serverless function orchestration service.

Fargate: Serverless compute engine for containers.

Opening AWS Account and Free Tier:



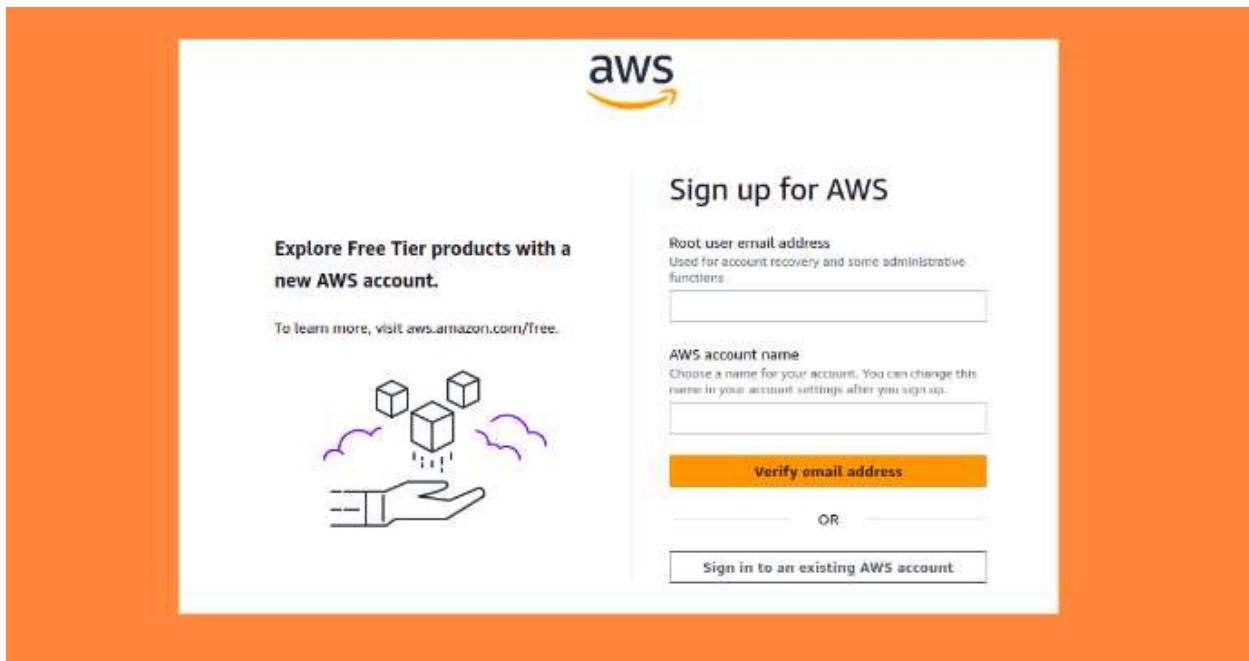
To open an AWS (Amazon Web Services) account, you'll need the following:

- **Email Address:** You must have a valid email address to create an AWS account.
- **Valid Card:** AWS requires a valid credit card to verify your identity and to set up billing.
- **Phone Number:** AWS may also require a valid phone number for additional account verification purposes also for two-factor authentication.
- **Billing Address:** You'll need to provide a billing address associated with your credit card for invoicing purposes.
- **Agree to AWS Terms and Conditions:** You'll need to review and agree to AWS's terms of service, which outline the terms and conditions of using their services.

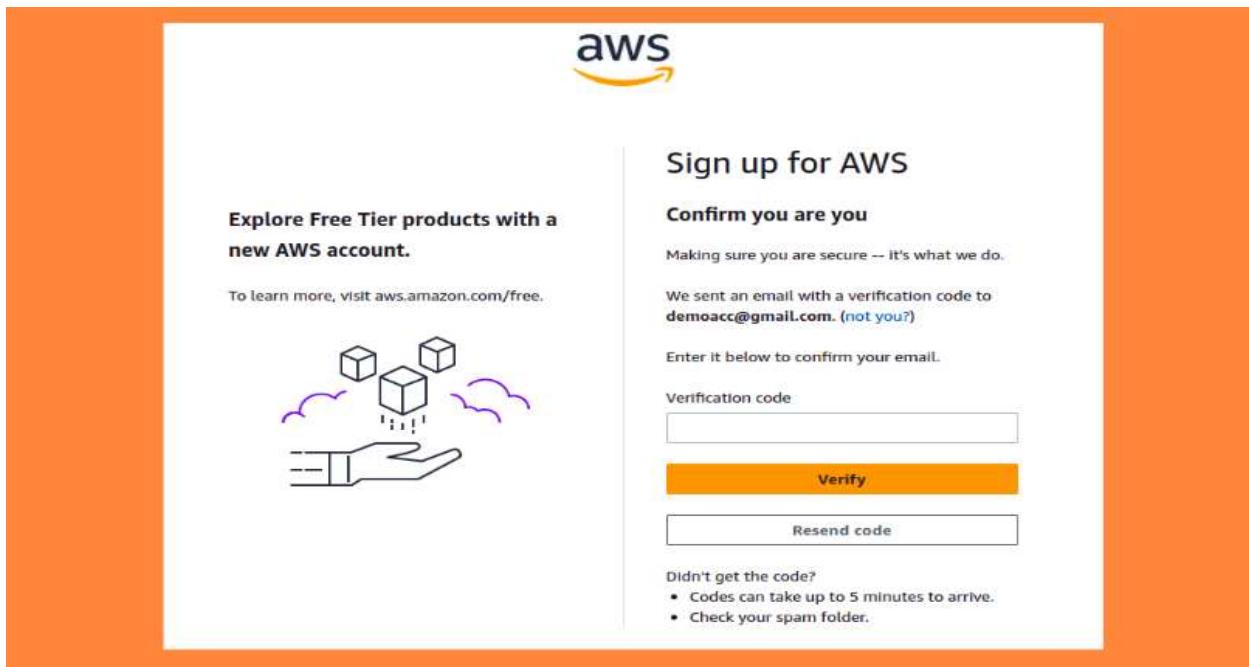
SignUp Link: https://console.aws.amazon.com/console/home?nc2=h_ct&src=header-signin

Email verification and account name:

- AWS Account name
- Email address
- Password



Verify Email:



Contact Information and Billing Information:

The screenshot shows two side-by-side pages of the AWS sign-up process. On the left, the 'Contact Information' step is shown, where users can choose between 'Business' or 'Personal' usage, enter their name, phone number, and address. On the right, the 'Secure verification' step is shown, which includes a note about temporary hold on funds and a shield icon indicating verification status. A large orange arrow points from the contact information section to the secure verification section.

Select for “Basic Support - Free”

The screenshot shows the 'Select a support plan' step of the AWS sign-up process. It lists three options: 'Basic support - Free', 'Developer support - From \$20/month', and 'Business support - From \$100/month'. The 'Basic support - Free' option is highlighted with a red box. To the right, there is a graphic of a rocket launching from a cloud. Below the support plans, there is a section for 'Enterprise level support'.

AWS Free Tier:

The screenshot shows the 'Explore Top Product Categories' section of the AWS Free Tier details page. It features a grid of seven categories: Compute, Database, Storage, Containers, Web & Mobile Apps, Serverless, and Machine Learning. Below this, the 'Free Tier details' section is displayed, with a search bar labeled 'Search free tier products'. The results are categorized into three columns: COMPUTE, STORAGE, and DATABASE. Each column lists a service, its tier, duration, and a large blue callout for usage limits.

| Category | Service | Tier | Duration | Usage Limit |
|----------|------------|-----------|----------------|--|
| COMPUTE | Amazon EC2 | Free Tier | 12 MONTHS FREE | 750 Hours per month |
| | | | | Resizable compute capacity in the Cloud. |
| STORAGE | Amazon S3 | Free Tier | 12 MONTHS FREE | 5 GB of standard storage |
| | | | | Secure, durable, and scalable object storage infrastructure. |
| DATABASE | Amazon RDS | Free Tier | 12 MONTHS FREE | 750 Hours per month of database usage (applicable DB engines) |
| | | | | Managed Relational Database Service for MySQL, PostgreSQL, MariaDB, or SQL Server. |

Link:

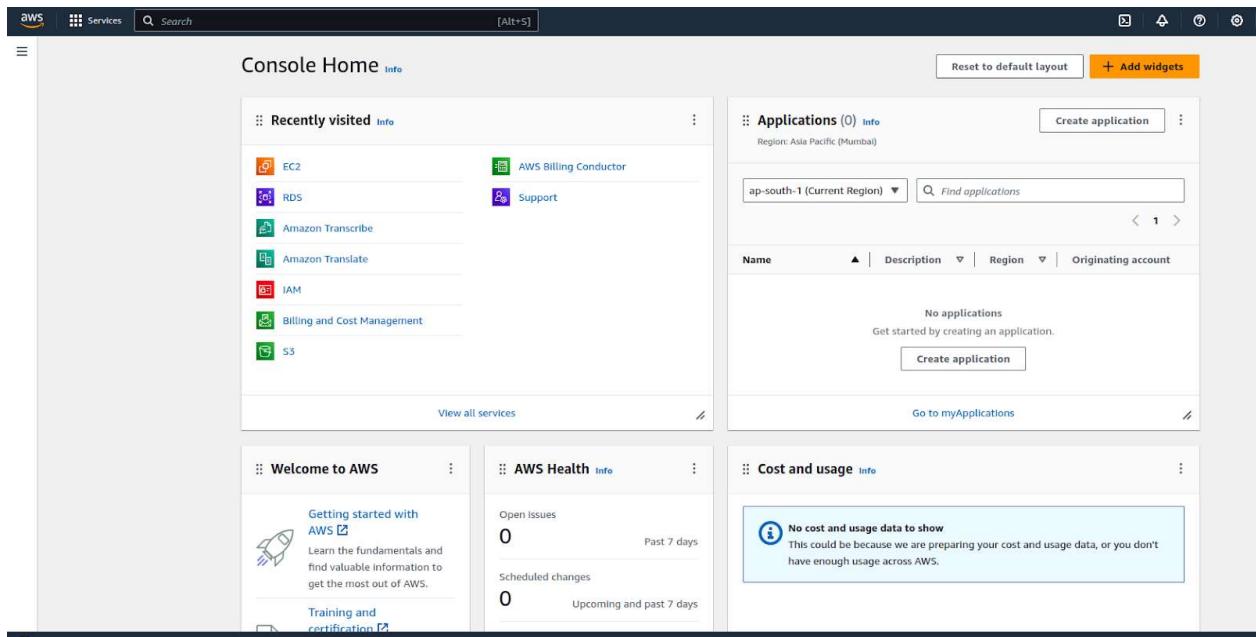
https://aws.amazon.com/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=tier%2312monthsfree&awsf.Free%20Tier%20Categories=*all

Duration: The Free Tier is available for 12 months from the date you sign up for an AWS account.

Services: The Free Tier includes a selection of AWS services, such as EC2, S3, RDS etc. Each service has its own usage limits and conditions.

Usage Limits: If you exceed these Free Tier limits, standard AWS service rates will apply.

3) Getting Started:



For working with aws we can use:

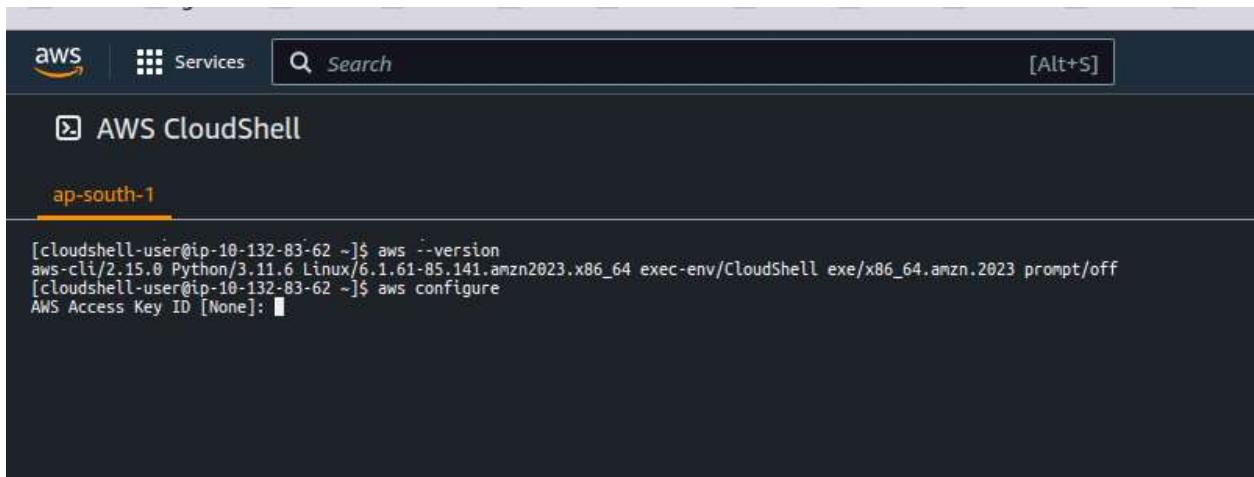
- AWS Management Console with **password**
- Command Line Interface (CLI) with protected by **access keys**
- Software Developer Kit (SDK) with protected by **access keys**

AWS Management Console:

This screenshot shows the AWS Management Console Instances page. On the left, a sidebar lists various EC2-related options like EC2 Dashboard, Global View, Events, Instances (selected), Images, AMIs, and AMI Catalog. The main area displays a table titled 'Instances (1/1) Info' with one row for an instance named 'app-server'. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IPv4 ... The instance details show it's 'Running' on an 't2.micro' type in 'ap-south-1b' with a public IP of '15.206.164.245.ap...'. Below the table, a detailed view for 'Instance: i-0ac2b9f652a175f76 (app-server)' is shown with tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. Under 'Details', it shows the Instance ID, Public IPv4 address ('15.206.164.245'), and Instance state ('Running'). It also lists Private IPv4 addresses ('172.31.5.231') and Public IPv4 DNS ('ec2-15-206-164-245.ap-south-1.compute.amazonaws.com').

AWS CLI:

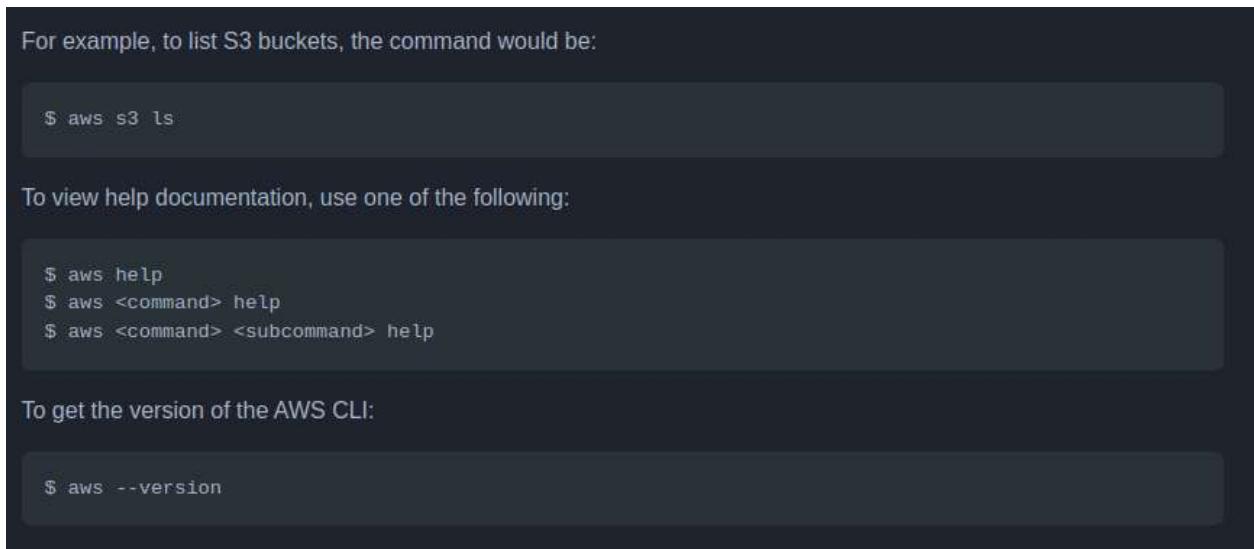
From Browser:



The screenshot shows the AWS CloudShell interface within a web browser. At the top, there's a navigation bar with the AWS logo, a 'Services' dropdown, a search bar containing 'Search', and a keyboard shortcut '[Alt+S]'. Below the navigation bar, the title 'AWS CloudShell' is displayed next to a square icon. Underneath the title, the region 'ap-south-1' is selected, indicated by an orange underline. The main area is a dark terminal window showing command-line output. The output includes the command 'aws --version' followed by its output: 'aws-cli/2.15.0 Python/3.11.6 Linux/6.1.61-85.141.amzn2023.x86_64 exec-env/CloudShell exe/x86_64.amzn.2023 prompt/off'. Then, the command 'aws configure' is run, and the prompt 'AWS Access Key ID [None]:' is shown.

Python SDK: AWS SDK for Python, commonly known as Boto3, is the official SDK provided by AWS to interact with AWS services using Python.

<https://github.com/aws/aws-cli>



The screenshot shows a section of the AWS CLI documentation. It starts with a note: 'For example, to list S3 buckets, the command would be:' followed by a code block containing '\$ aws s3 ls'. Below this, another note says 'To view help documentation, use one of the following:' followed by three code blocks: '\$ aws help', '\$ aws <command> help', and '\$ aws <command> <subcommand> help'. Finally, a note says 'To get the version of the AWS CLI:' followed by a code block containing '\$ aws --version'.

Software Developer Kit (SDK):

The screenshot shows the AWS SDK for JavaScript homepage. At the top, there's a navigation bar with links like 'reInvent', 'Products', 'Solutions', 'Pricing', 'Documentation', 'Learn', 'Partner Network', 'AWS Marketplace', 'Customer Enablement', 'Events', 'Explore More', and a search icon. Below the navigation is a secondary navigation bar with 'Developer Center', 'Learning', 'Programming Languages' (which is currently selected), 'Events', 'Tools', 'Community', and 'More Resources'. A banner at the top says 'AWS SDK for JavaScript' and describes it as providing first-class TypeScript support and making it easy to call AWS services using idiomatic JavaScript APIs. There's a prominent orange 'Install from NPM' button. Below the banner, there are three main sections: 'Develop Server-side apps' (with a Node.js logo), 'Develop Web apps' (with a JS logo), and 'Develop Mobile apps' (with a React Native logo). Each section has a brief description and a 'Get started' link.

Create a S3 Bucket using NodeJS sdk: \$npm install aws-sdk

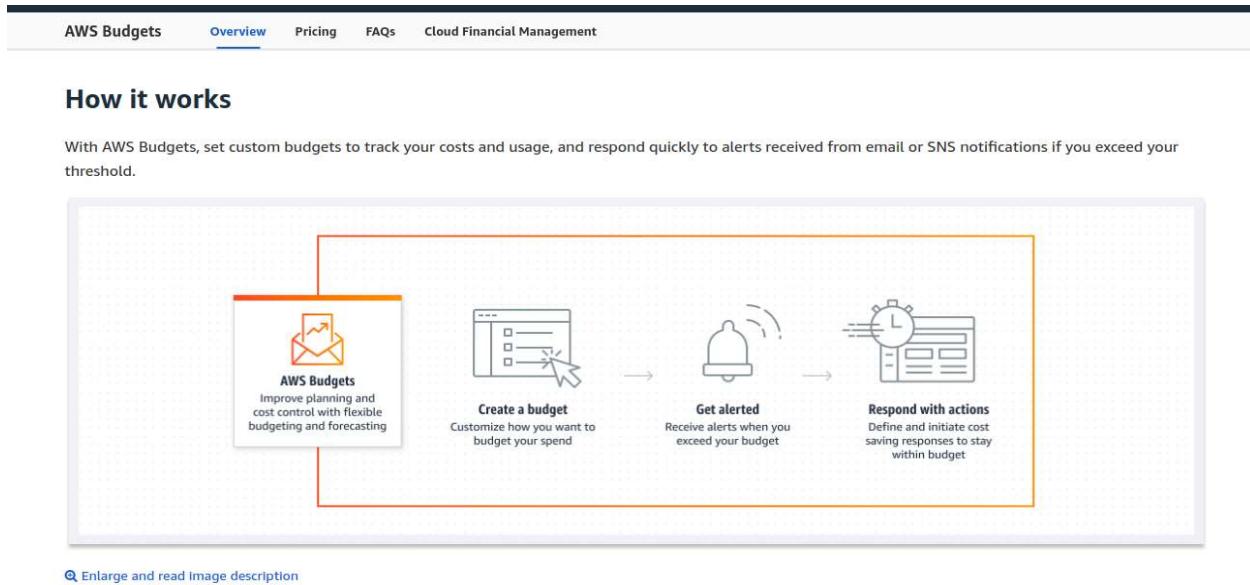
```
// Import the AWS SDK
const AWS = require('aws-sdk');

// Set your AWS credentials and region
AWS.config.update({
  accessKeyId: 'YOUR_ACCESS_KEY_ID',
  secretAccessKey: 'YOUR_SECRET_ACCESS_KEY',
  region: 'us-east-1' // Set your preferred AWS region
});

// Create an S3 object
const s3 = new AWS.S3();
```

\$node s3-example.js

4) Cost And Budgets



[Enlarge and read image description](#)

Use cases

[Monitor costs and usage](#)

[Create scheduled reports](#)

[Respond to thresholds](#)

AWS Pricing Calculator Link: <https://calculator.aws/#/>

Estimate the cost for your architecture solution.

Configure a cost estimate that fits your unique business or personal needs with AWS products and services.

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

[Create estimate](#)

How it works

AWS Pricing Calculator
Estimate the cost of AWS products and services

Step 1: Add services
Search and add AWS services that you need

Step 2: Configure service
Enter the details of your usage to see service costs

Step 3: View estimate totals
See estimated costs per service, service groups, and totals

More resources

[User guide](#)

[Pricing assumptions and variations](#)

Need help with estimation? Connect with AWS certified expert on AWS IQ

AWS Modernization Calculator for Microsoft Workloads
Estimate the cost of transforming Microsoft workloads to a modern architecture that uses open source and cloud-native services deployed on AWS.

Zero spend budget Config for Free :

Create a budget that notifies you once your spending exceeds \$0.01 which is above the AWS Free Tier limits.

The screenshot shows the AWS Billing and Cost Management console. In the top left, there's a sidebar with links like Home, Getting Started, and various sections under Billing and Payments and Cost Analysis. The main area is titled 'Choose budget type' and contains two options: 'Use a template (simplified)' (selected) and 'Customize (advanced)'. Under 'Templates - new', there are two options: 'Zero spend budget' (selected) and 'Monthly cost budget'. The 'Zero spend budget' description states: 'Create a budget that notifies you once your spending exceeds \$0.01 which is above the AWS Free Tier limits.'

My Budget:

The screenshot shows the 'Overview' page for budgets. It displays a table with one row for 'My Zero-Spend Budget'. The table columns include Name, Thresholds, Budget, Amount used, Forecasted amount, Current vs. budgeted, and Forecasted vs. budgeted. The 'Budget' column shows '\$1.00' and '\$0.00'. The 'Current vs. budgeted' column shows '0.00%'.

| Name | Thresholds | Budget | Amount used | Forecasted amount | Current vs. budgeted | Forecasted vs. budgeted |
|----------------------|------------|--------|-------------|-------------------|----------------------|-------------------------|
| My Zero-Spend Budget | OK | \$1.00 | \$0.00 | - | 0.00% | - |

5) User, Role and IAM User:

Root User: This user created during the registration process. Has full access to all AWS services and resources. It is recommended not to use the root user for day-to-day activities and to create IAM users instead.

The screenshot shows the AWS Documentation page for 'AWS Identity and Access Management' under 'User Guide'. The main content is titled 'AWS account root user'. It includes a sidebar with navigation links like 'What is IAM?', 'Getting set up', 'Security best practices and use cases', 'Tutorials', and 'Identities'. A red box highlights an 'Important' note: 'We strongly recommend that you don't use the root user for your everyday tasks and that you follow the root user best practices for your AWS account. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials.' Below this, a 'Tasks' section lists various management tasks. On the right side, there are social sharing icons for LinkedIn, Twitter, and GitHub.

IAM (Identity and Access Management) Users: IAM users have specific credentials (username and password or access keys) and can be assigned individual permissions. IAM users are commonly used for security best practices to avoid using the root user for regular tasks.

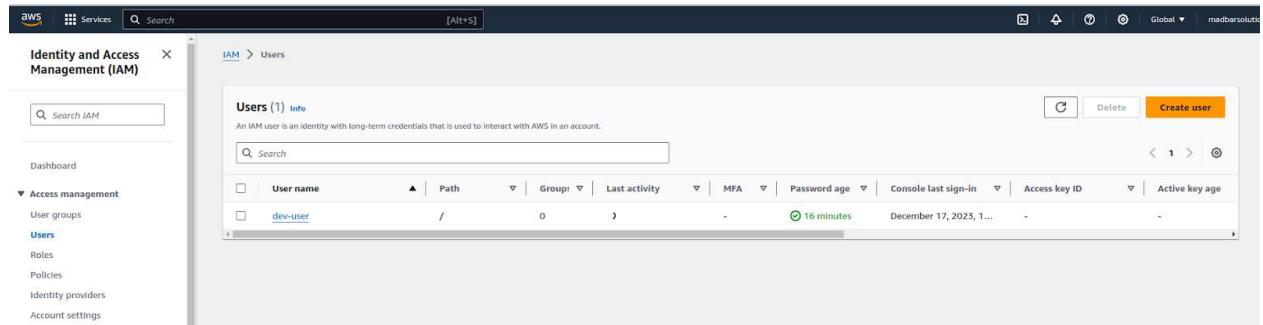
Federated Users: Users who are authenticated by an external identity provider (e.g., Active Directory, Google, Facebook) and then given access to AWS resources using IAM roles.

AWS Account: An AWS account represents a unique customer or business entity within AWS. Each AWS account has its own resources, billing, and security settings.

AWS User doc Link : https://docs.aws.amazon.com/IAM/latest/UserGuide/id_root-user.html

IAM Users:

IAM users have specific credentials (username and password or access keys) and can be assigned individual permissions.

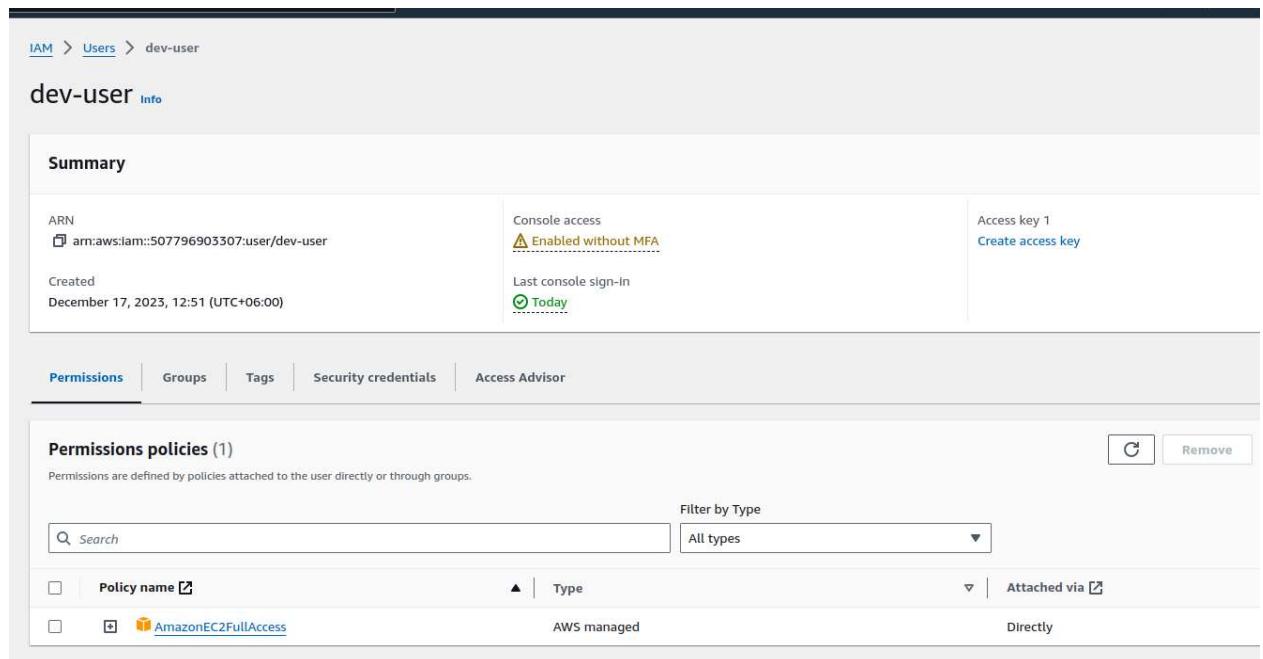


The screenshot shows the AWS IAM console with the 'Users' section selected. A single user, 'dev-user', is listed in the table. The table columns include User name, Path, Group, Last activity, MFA, Password age, Console last sign-in, Access key ID, and Active key age. The 'dev-user' row shows a path of '/', a group of '0', last activity at '16 minutes' ago, and a console sign-in on 'December 17, 2023, 1...'. There are buttons for 'Create user' and 'Delete' at the top right of the table.

User Permission/Policies:

IAM policies are attached to IAM users, groups, or roles to grant or restrict access to AWS resources.

An IAM user policy consists of statements that describe the allowed or denied actions and the resources on which those actions can be performed.



The screenshot shows the 'dev-user' details page. The 'Summary' tab is active, displaying the ARN (arn:aws:iam::507796903307:user/dev-user), console access status (Enabled without MFA), creation date (December 17, 2023, 12:51 (UTC+06:00)), and last console sign-in (Today). There is a link to 'Create access key'. Below the summary, tabs for 'Permissions', 'Groups', 'Tags', 'Security credentials', and 'Access Advisor' are visible. The 'Permissions' tab is selected, showing one attached policy: 'AmazonEC2FullAccess'. A 'Remove' button is available to detach this policy.

User Group:

Instead of assigning policies to individual users, you can assign policies to groups.

The screenshot shows the AWS IAM User Groups console. At the top, it displays the path: IAM > User groups > developer-group. Below this, the group name "developer-group" is shown with an "Info" link. A "Summary" section provides details: User group name (developer-group), Creation time (December 17, 2023, 13:41 (UTC+06:00)), and a "Actions" button. Below the summary, there are tabs for "Users", "Permissions" (which is selected), and "Access Advisor". The "Permissions policies" section shows two managed policies: "AmazonEC2FullAccess" and "AmazonRDSFullAccess", both listed as "AWS managed". There is also a search bar and a "Filter by Type" dropdown set to "All types".

Sign with IAM user:

The sign-in page features the AWS logo at the top. Below it, the text "Sign in" is displayed. Two options are presented: "Root user" (unselected) and "IAM user" (selected). The "IAM user" option includes a sub-note: "User within an account that performs daily tasks." Below these options is a field labeled "Account ID (12 digits) or account alias" with a placeholder box. At the bottom is a blue "Next" button.



6) AWS Architecture, Regions and AZ



Key concepts related to AWS architecture include:

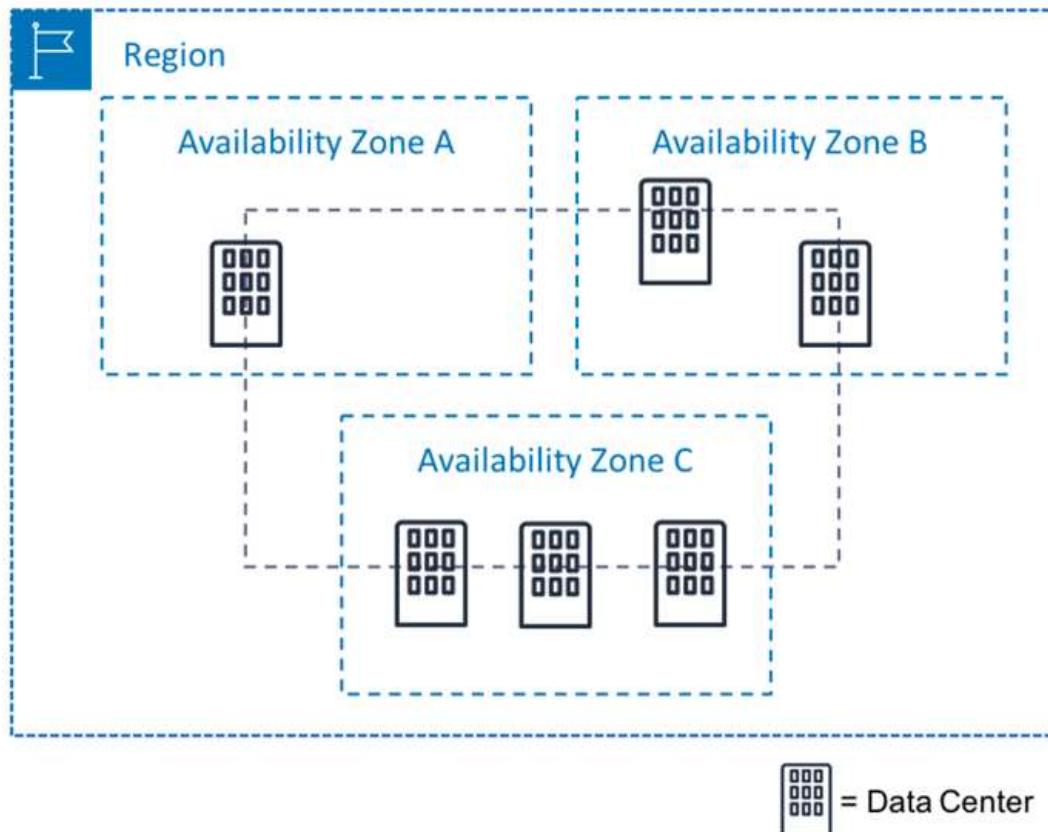
- **Regions**
- **Availability zones (AZs)**

Regions: A region is a geographical area. Each AWS region is independent and isolated from other regions, providing high availability and fault tolerance.

AZ: An Availability Zone is a data center or a collection of data centers within an AWS region.

Regions and AZ for:

Regions, and availability zones is crucial for designing and deploying scalable, reliable, fault tolerance and high-performance applications in the cloud.

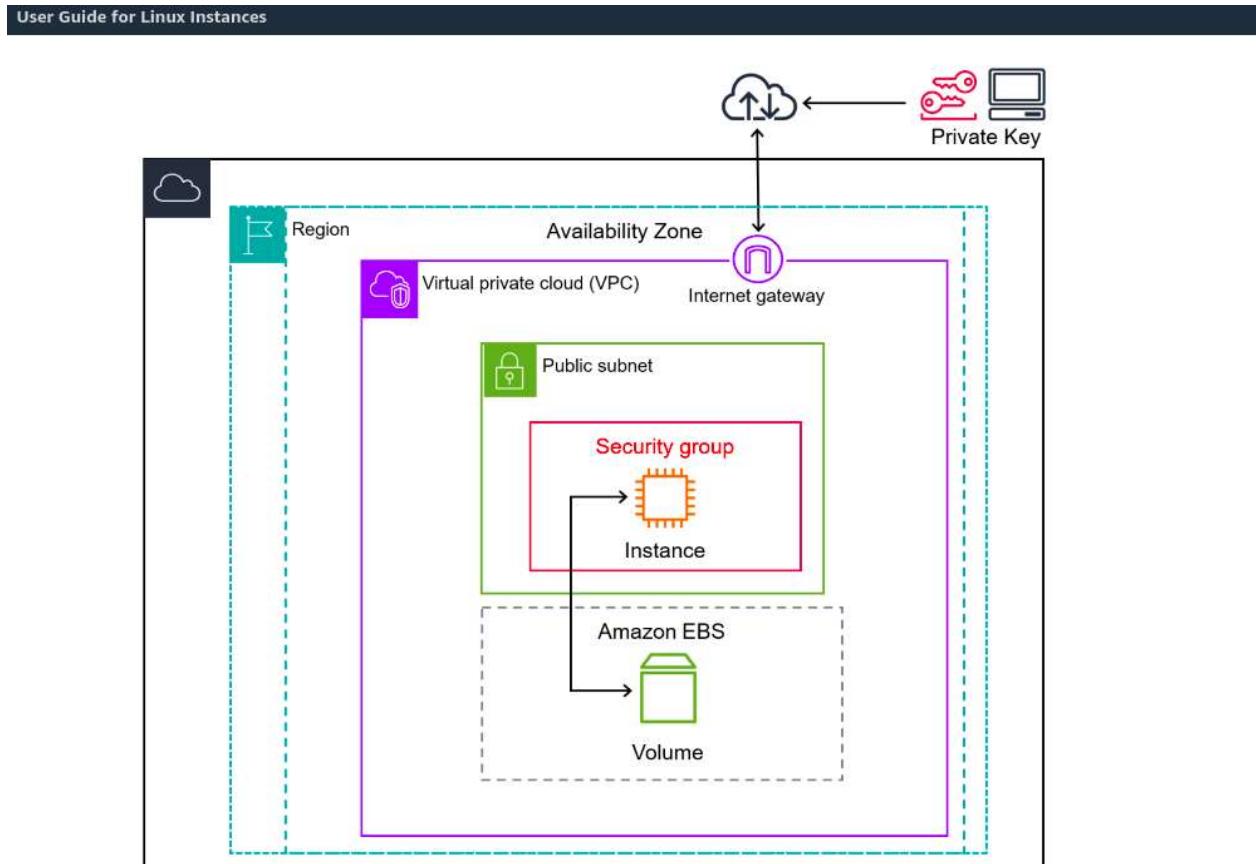


AWS Regions and Availability Zones dock Link:

<https://docs.aws.amazon.com/whitepapers/latest/get-started-documentdb/aws-regions-and-availability-zones.html>

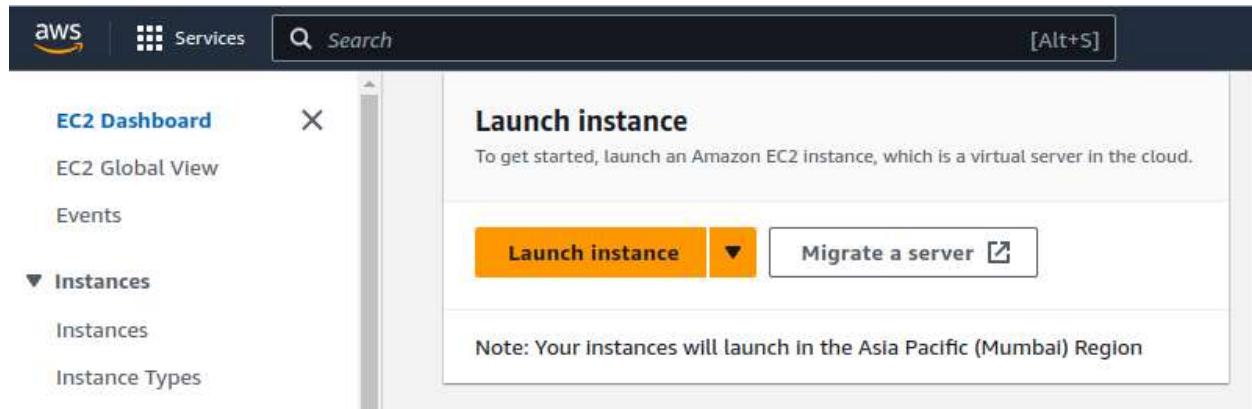
7) Play with EC2 (Infrastructure as a Service)

EC2 instances are essentially virtual machines (VMs) that you can use to host and run applications, process data, and perform various computing tasks. EC2 provides on-demand, scalable computing capacity in the AWS Cloud.



EC2 Doc Link: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>

Run an EC2:



Amazon Machine Images (AMIs): Its like OS. AMIs can be customized, and users can create their own or choose from a wide range of public AMIs provided by AWS and the community.

Instance Types: (For CPU and Memory) EC2 offers a variety of instance types optimized for different use cases, such as **general-purpose** computing, **memory-intensive** tasks, **storage-optimized** applications.

Key pairs: Secure login information for your instances. AWS stores the public key and you store the private key in a secure place.

Network settings: (Security groups and VPC) Its allows you to specify the **protocols**, **ports**, and **source IP** ranges that can reach your instances, and the **destination IP** ranges to which your instances can connect.

Configure storage: Specify the storage options for the instance.

Other Option: Domain Name join, OS Profiling, Purchasing option, Shutdown behavior, **User data (Bootstrap-Script) etc.**

Lunch an EC2:

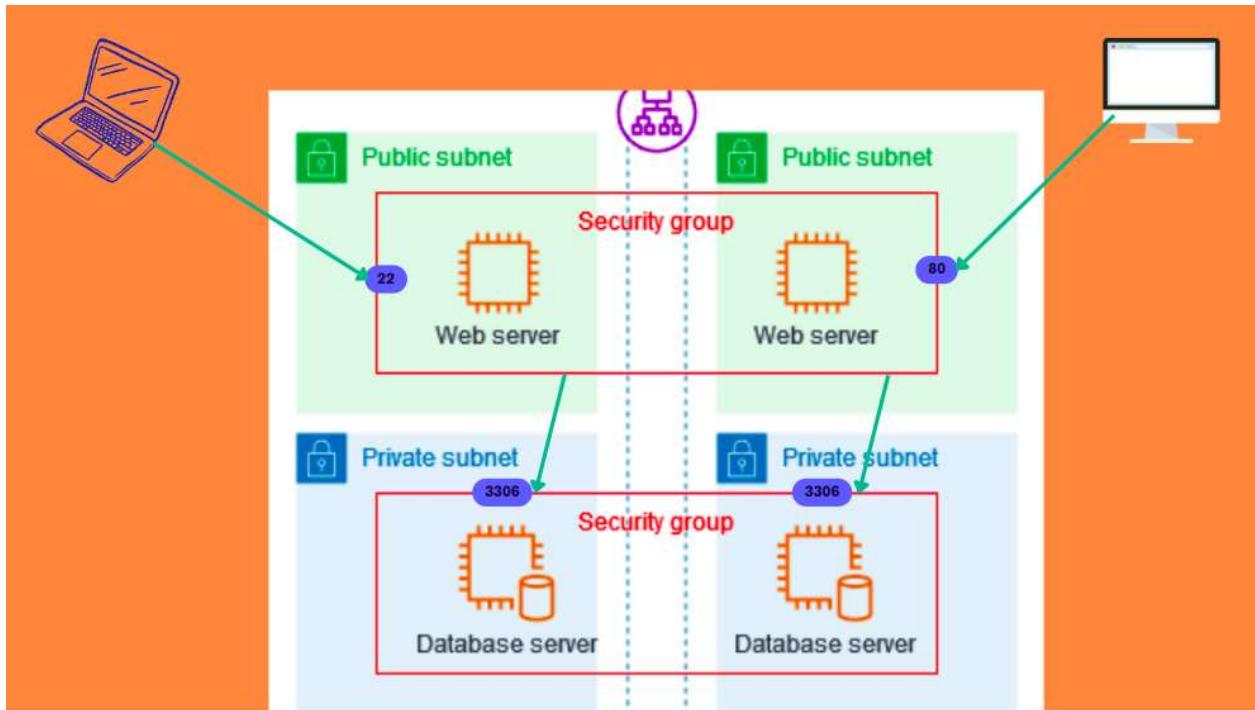
The screenshot shows the AWS Management Console interface for launching an EC2 instance. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, a search bar containing 'Search', and a keyboard shortcut '[Alt+S]'. Below the navigation is a breadcrumb trail: 'EC2 > Instances > Launch an instance'. The main title is 'Launch an instance' with an 'Info' link. A descriptive text below says: 'Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.' The first step is titled 'Name and tags' with an 'Info' link. It has a 'Name' field containing 'e.g. My Web Server' and a 'Add additional tags' button. The second step is titled 'Application and OS Images (Amazon Machine Image)' with an 'Info' link. It explains what an AMI is and includes a search bar: 'Search our full catalog Including 1000s of application and OS images'. Below this is a 'Quick Start' section with icons for various operating systems: Amazon Linux (aws logo), macOS (Mac logo), Ubuntu (Ubuntu logo), Windows (Microsoft logo), Red Hat (Red Hat logo), and SUSE Linux (SUSE logo). To the right of the quick start section is a 'Browse more AMIs' link with a magnifying glass icon, which includes a note: 'Including AMIs from AWS, Marketplace and the Community'.

Lunch Ubuntu EC2 With User Data (Bootstrap-Script):

```
#!/bin/bash
sudo apt update -y
sudo apt install nginx -y
sudo systemctl enable nginx
sudo chmod 666 /var/www/html/index.nginx-debian.html
sudo echo "<h1>Hello from AWS Bangla</h1>" > /var/www/html/index.nginx-debian.html
```

Network settings: (Security groups and VPC) :

A Security Group acts as a virtual firewall for your Amazon EC2. It controls inbound and outbound traffic at the instance level.



- It operate at the instance level, not the subnet level.
- They are stateful, if you allow inbound traffic, outbound traffic is automatically allowed.
- You can define rules based on IP addresses, protocols, and port ranges to control traffic.
- By default, all inbound traffic is denied, and all outbound traffic is allowed.

Create Security Group

| | |
|---------------------|-------------------------------------|
| Security group name | Web Server Security Group |
| Description | Security for production web server. |
| VPC | vpc-e68d9c81 DefaultVPC (default) |

Security group rules:

| Type | Protocol | Port Range | Source | Description |
|-------|----------|------------|----------|-------------------------------------|
| SSH | TCP | 22 | Anywhere | 0.0.0.0/0, ::/0 Admin access. |
| HTTP | TCP | 80 | Anywhere | 0.0.0.0/0, ::/0 Web traffic. |
| HTTPS | TCP | 443 | Custom | 0.0.0.0/0, ::/0 Secure web traffic. |

Add Rule

Cancel **Create**

Each EC2 instance must be associated with at least one Security Group, and each Security Group is associated with one VPC.

▼ Network settings [Info](#) [Edit](#)

Network [Info](#)
vpc-0853810a86fc473df

Subnet [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
Enable

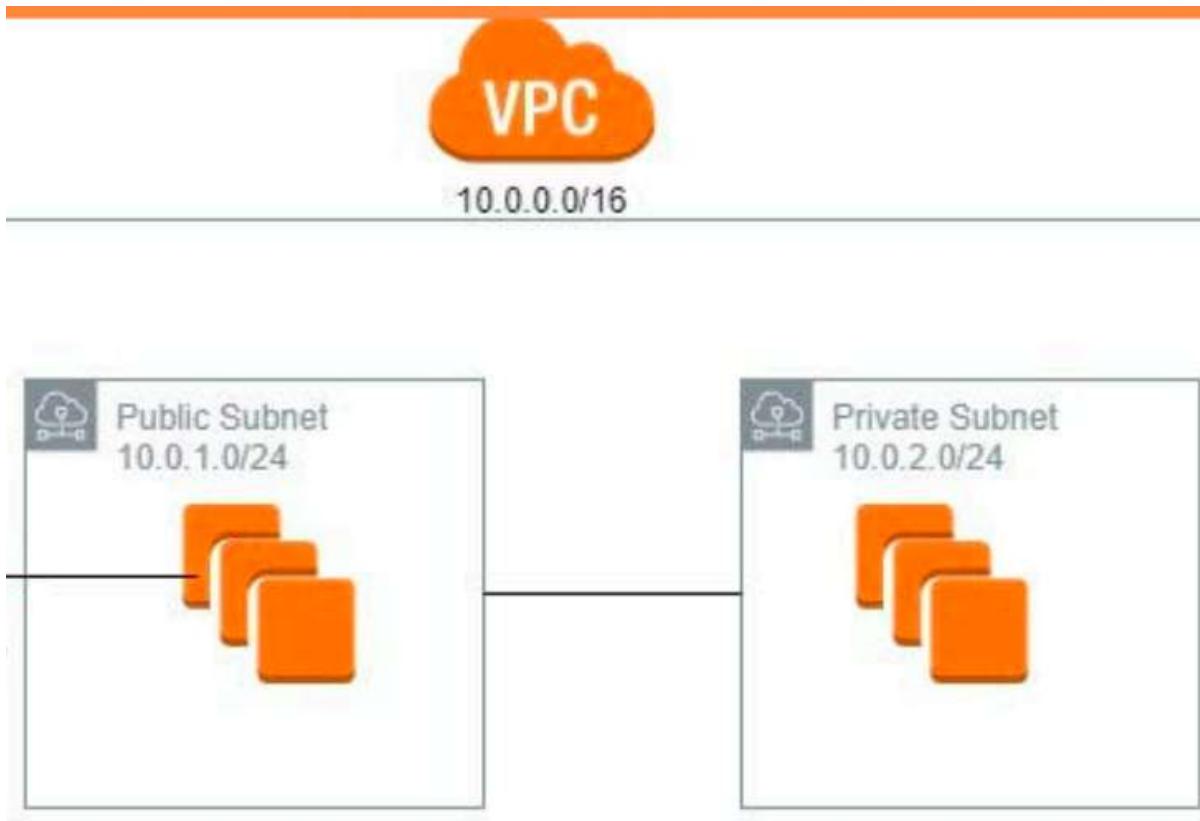
Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

Common security groups [Info](#)
 Select security groups ▾
 SecGroupForHttp sg-0fe5c16d8e7feab6a X
 VPC: vpc-0853810a86fc473df

 Compare security group rules

Virtual Private Cloud (VPC): It provides a virtual network in the cloud and allows you to define your own IP address range, create subnets, and configure route tables.



VPCs provide the overall networking structure, and Security Groups provide the firewall rules for the instances within that VPC.

SSH to EC2: Create a key pair or choose an existing one. This is crucial for SSH access.

```
LOUD > aws > aws.txt

$ssh -i your-key.pem ubuntu@ec2-instance-public-ip
#Access EC2 with ssh

$scp -i your-key.pem /path/to/local/file.txt ubuntu@13.233.103.153:/path/to/ec2/
#Copy file to EC2 from local-pc
```

From Local-PC:

```
$ssh -i your-key.pem ubuntu@ec2-public-ip
$scp -i your-key.pem /path/to/local/file.txt ubuntu@ec2-public-ip:/path/to/ec2/
```

```
bs960@BS-960:~/Downloads$ sudo ssh -i my-ec2-key.pem ubuntu@13.233.103.153
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 System information as of Mon Dec 18 10:58:14 UTC 2023

 System load:  0.0          Processes:           101
 Usage of /:   23.4% of 7.57GB   Users logged in:     1
 Memory usage: 20%           IPv4 address for eth0: 172.31.4.4
 Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

89 updates can be applied immediately.
56 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Mon Dec 18 10:58:15 2023 from 103.158.249.226
ubuntu@ip-172-31-4-4:~$
```

EC2 pricing model:

The screenshot shows the 'On-Demand Pricing' section of the Amazon EC2 Pricing page. A sidebar on the left lists various links related to EC2 pricing and usage. The main content area displays a table of On-Demand Instances. The table has columns for Instance name, On-Demand hourly rate, vCPU, Memory, Storage, and Network performance. The table includes rows for r7g.medium, m7gd.medium, c7gd.medium, r7gd.medium, m7a.medium, c6gd.medium, t2.nano, and x2gd.medium instances.

| Instance name | On-Demand hourly rate | vCPU | Memory | Storage | Network performance |
|---------------|-----------------------|------|---------|-----------------|---------------------|
| r7g.medium | \$0.0536 | 1 | 8 GiB | EBS Only | Up to 12500 Megabit |
| m7gd.medium | \$0.0534 | 1 | 4 GiB | 1 x 59 NVMe SSD | Up to 12500 Megabit |
| c7gd.medium | \$0.0454 | 1 | 2 GiB | 1 x 59 NVMe SSD | Up to 12500 Megabit |
| r7gd.medium | \$0.068 | 1 | 8 GiB | 1 x 59 NVMe SSD | Up to 12500 Megabit |
| m7a.medium | \$0.05796 | 1 | 4 GiB | EBS Only | Up to 12500 Megabit |
| c6gd.medium | \$0.0384 | 1 | 2 GiB | 1 x 59 NVMe SSD | Up to 10 Gigabit |
| t2.nano | \$0.0058 | 1 | 0.5 GiB | EBS Only | Low |
| x2gd.medium | \$0.0835 | 1 | 16 GiB | 1 x 59 SSD | Up to 10 Gigabit |

EC2 pricing model typically includes the following components:

- Instance Types
- On-Demand Instances
- Reserved Instances
- Spot Instances
- Dedicated Hosts
- Savings Plans
- Data Transfer

Aws pricing model doc Link: <https://aws.amazon.com/ec2/pricing/on-demand/>

The screenshot shows the AWS website with the navigation bar at the top. Under 'Products', 'Compute', 'Amazon EC2', and 'Amazon EC2 Pricing' are selected. The main heading is 'Amazon EC2 On-Demand Pricing'. Below it are two buttons: 'Get started for free' (orange) and 'Request a pricing quote' (blue). A large geometric graphic is on the right.

Instance Types:

AWS provides a variety of instance types (**compute-optimized, memory-optimized, storage-optimized etc**), Each instance type has its own pricing, reflecting the underlying hardware specifications and performance characteristics.

On-Demand Instances:

With on-demand instances, you pay for compute capacity on an hourly or per-second basis.

Reserved Instances:

AWS offers reserved instances. Reserved instances provide a significant discount compared to on-demand pricing but require a **one or three year** term commitment.

Spot Instances:

Spot instances allow you to bid for unused EC2 capacity, potentially providing substantial cost savings. However, **these instances can be terminated by AWS with short notice**.

Dedicated Hosts:

If you require a physical EC2 server for **compliance or licensing** reasons, you can use dedicated hosts. With dedicated hosts, you pay for the entire host, regardless of the number of instances running on it.

Savings Plans:

A flexible pricing model that provides significant savings in exchange for a commitment to a consistent amount of usage (measured in \$/hr) for a **1 or 3-year period**. Savings Plans offer lower prices compared to on-demand pricing.

Data Transfer:

Additional costs may be incurred for data transfer between **EC2 instances and the internet**, as well as **data transfer between regions**.

8) AWS Storage services

Amazon Web Services (AWS) offers a variety of storage solutions to meet different needs, providing scalable, durable, and cost-effective options.

The screenshot shows the AWS Storage Services page. At the top, there's a navigation bar with links like 'Contact Us', 'Support', 'English', and 'My Account'. Below the navigation, a blue banner says 'Cut your storage costs by more than half | Introducing Amazon EFS Archive, a cost-optimized storage class for long-lived file data »'. The main content area is titled 'AWS storage services' and 'Object, file, and block storage'. It features five service cards:

- Amazon Simple Storage Service (S3)**: Object storage with industry-leading scalability, availability, and security for you to store and retrieve any amount of data from anywhere.
- Amazon Elastic File System (EFS)**: A simple, serverless, elastic, set-and-forget file system for you to share file data without managing storage.
- Amazon FSx**: Fully managed, cost-effective file storage offering the capabilities and performance of popular commercial and open-source file systems.
- Amazon Elastic Block Store (EBS)**: Easy to use, high-performance block storage service for both throughput and transaction-intensive workloads at any scale.
- Amazon File Cache**: High-speed cache for datasets stored anywhere, accelerate cloud bursting workloads.

Here are some key AWS storage services:

Amazon Simple Storage Service (S3):

Object Storage. S3 is a highly scalable object storage service that allows you to store and retrieve any amount of data from anywhere on the web. S3 supports various storage classes, each offering different levels of durability, availability, and cost.

Amazon Elastic Block Store (EBS):

EBS provides block-level storage volumes for use with Amazon EC2 instances. It allows you to attach and detach volumes from EC2 instances, providing persistent block storage for your applications.

Amazon Elastic File System (EFS):

EFS provides scalable and fully managed file storage for EC2 instances. It supports the Network File System (NFS) protocol and can be used to share files across multiple EC2 instances.

The screenshot shows the AWS Storage Services page with several service cards:

- AWS Storage Gateway**: Hybrid cloud storage service that gives you on-premises access to virtually unlimited cloud storage.
- AWS Snow Family**: Edge compute, data collection, and data transfer services with security and end-to-end logistics for mobile and rugged deployments.
- Managed file transfer**: AWS Transfer Family
- Disaster recovery and backup**: AWS Elastic Disaster Recovery (DRS) and AWS Backup

AWS Storage Gateway:

Storage Gateway is a hybrid cloud storage service that enables on-premises applications to seamlessly use AWS cloud storage.

Amazon CloudFront:

While not a traditional storage service, CloudFront is a content delivery network (CDN) that accelerates the delivery of your content (including S3 content) to end-users globally.

AWS Snow Family:

The AWS Snow Family is a set of physical devices designed to help customers transfer large amounts of data into and out of the AWS Cloud when online data transfer is not practical.

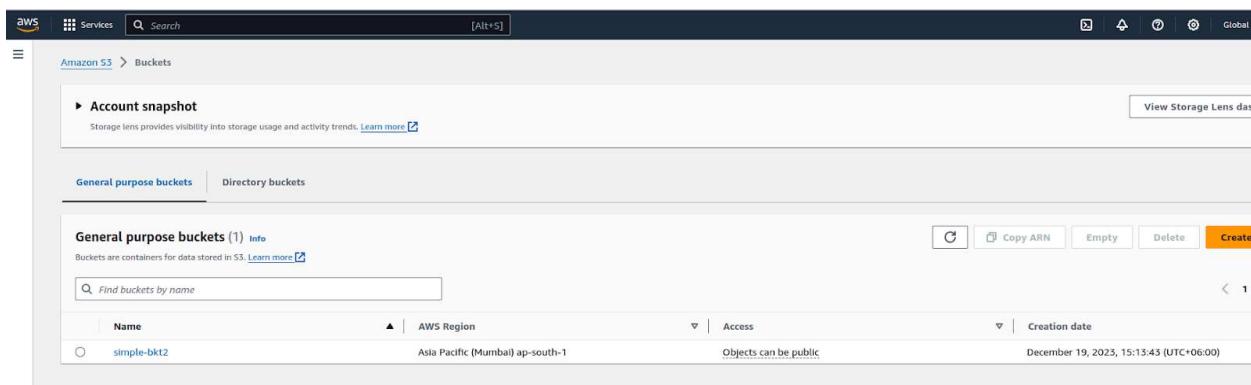
AWS Storage services doc Link: <https://aws.amazon.com/products/storage/>

Amazon Simple Storage Service (S3):

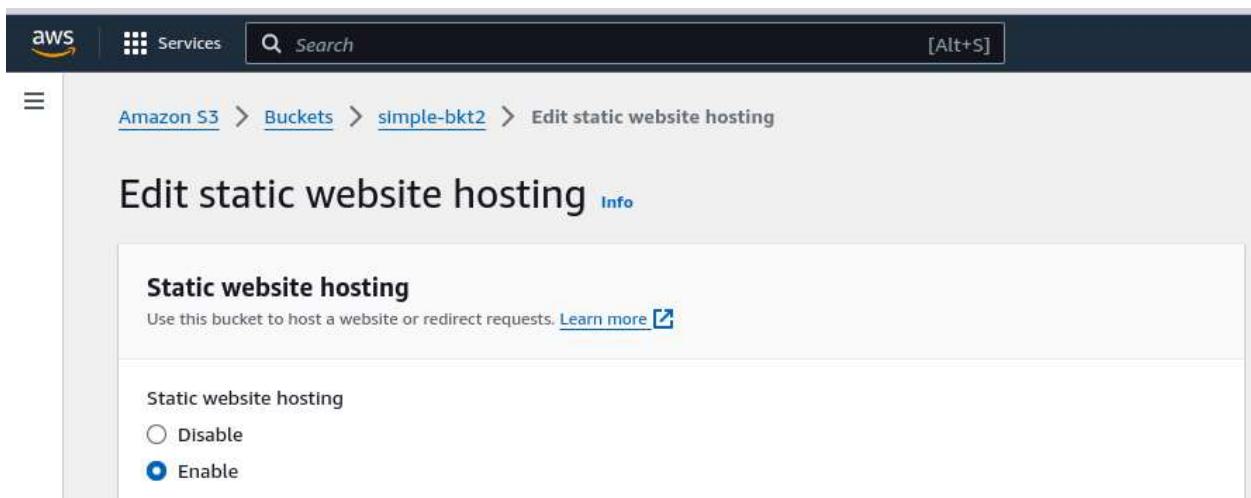
S3 Buckets are public cloud storage containers for objects stored in S3. S3 buckets can be likened to file folders and object storage.



Create bucket:



S3 Bucket for Static Website Hosting:



Permission required for S3:

- Bucket must have public access
- Add bucket policy for object access

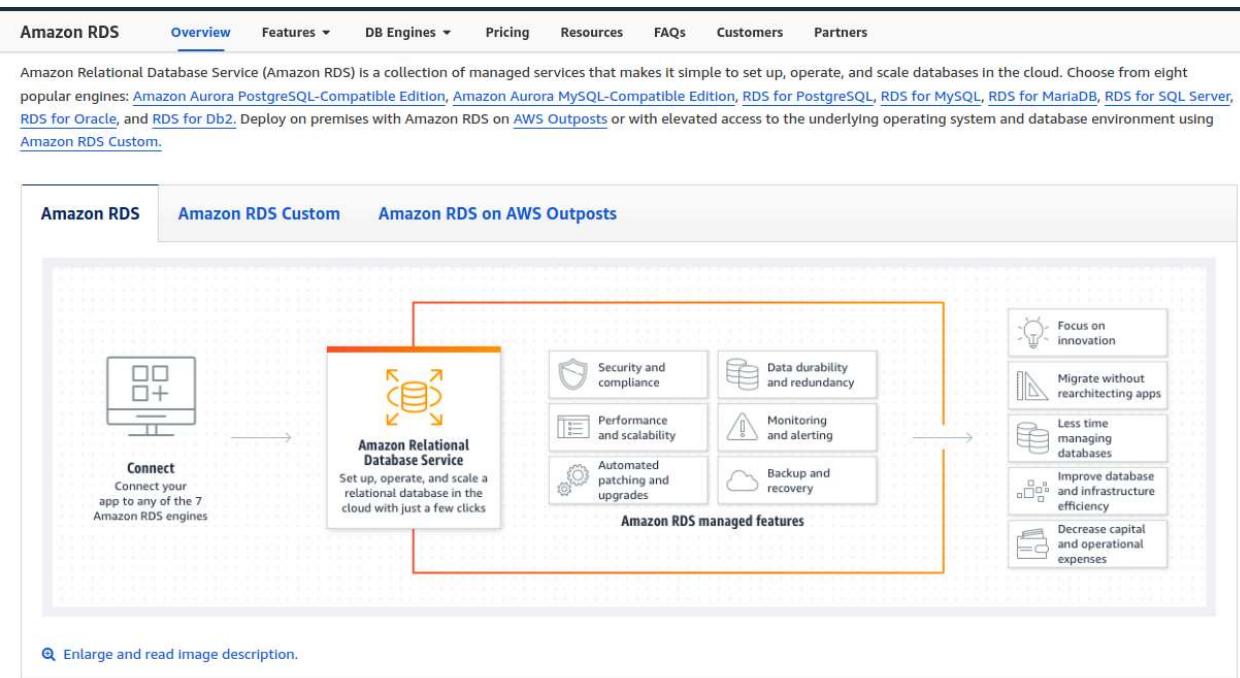
The screenshot shows the 'Edit bucket policy' page in the AWS S3 console. The URL in the address bar is `Amazon S3 > Buckets > simple-bkt2 > Edit bucket policy`. The main section is titled 'Bucket policy' with a sub-section 'Policy'. Below it is a code editor containing a JSON policy document. The JSON code is as follows:

```
1 Version: "2008-10-17",
2 Id: "PolicyForPublicWebsiteContent",
3 Statement: [
4     {
5         Sid: "PublicReadGetObject",
6         Effect: "Allow",
7         Principal: {
8             AWS: "*"
9         },
10        Action: "s3:GetObject",
11        Resource: "arn:aws:s3:::simple-bkt2/*"
12    }
13]
14}
15}
```

For S3 Public access, add this in Bucket Policy:

```
{
  "Version": "2008-10-17",
  "Id": "PolicyForPublicWebsiteContent",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::or-your-aws-bucket/*"
    }
  ]
}
```

9) Database in AWS



Amazon RDS engines:



[Amazon Aurora Pricing](#)



[Amazon RDS for MySQL Pricing](#)



[Amazon RDS for PostgreSQL Pricing](#)



[Amazon RDS for MariaDB Pricing](#)



[Amazon RDS for Oracle Pricing](#)



[Amazon RDS for SQL Server Pricing](#)

AWS RDS Doc Link: <https://aws.amazon.com/rds/>

Run a MySQL Server:

Screenshot of the AWS RDS 'Create database' wizard.

The top navigation bar shows the AWS logo, Services, Search, and [Alt+S].

The breadcrumb navigation shows RDS > Create database.

Create database

Choose a database creation method

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

Aurora (MySQL Compatible)

Aurora (PostgreSQL Compatible)

MySQL

MariaDB

Screenshot of the Amazon RDS 'Databases' page.

The top navigation bar shows the AWS logo, Services, Search, and [Alt+S].

The left sidebar shows the Amazon RDS menu with 'Databases' selected.

The breadcrumb navigation shows RDS > Databases.

Databases (1)

| DB identifier | Status | Role | Engine | Region & AZ | Size | CPU | Current activity | Maintenance | VPC | Multi-AZ |
|---------------|----------|----------|-----------------|-------------|-------------|-----|------------------|-----------------------|-----|----------|
| database-1 | Creating | Instance | MySQL Community | ap-south-1c | db.t3.micro | - | none | vpc-0853810a86fc473df | No | No |

Create MySql DB:

The screenshot shows the 'Connectivity & security' section of the AWS RDS console. On the left, there's a sidebar with links like Dashboard, Databases, Query Editor, etc. The main area has three columns: 'Endpoint & port' (Endpoint: database-1.cv2mk2cmopb3.ap-south-1.rds.amazonaws.com, Port: 3306), 'Networking' (Availability Zone: ap-south-1c, VPC: vpc-0853810a86fc473df, Subnet group: default-vpc-0853810a86fc473df, Subnets: multiple listed), and 'Security' (VPC security groups: allAllow (sg-01833d5aee90819c1) Active, SecurityGroupForRDS (sg-0d6836d55e2962f4c) Active, default (sg-0d38b0c43bd7ea438) Active, Publicly accessible: Yes).

Connect with workbench:

The screenshot shows the MySQL Workbench interface. The left sidebar lists 'MySQL Connections' with one entry named 'aws-db'. The main window shows the 'Manage Server Connections' dialog for 'aws-db'. It has tabs for Connection, Remote Management, and System Profile. Under Connection, it shows 'Connection Method: Standard (TCP/IP)', 'Parameters' tab selected, 'Hostname: database-1.cv2mk2cmopb3.ap-sout...', 'Port: 3306', 'Username: root', 'Password: Store in Keychain...', and 'Default Schema:'. There's also an 'SSL' tab and an 'Advanced' tab.

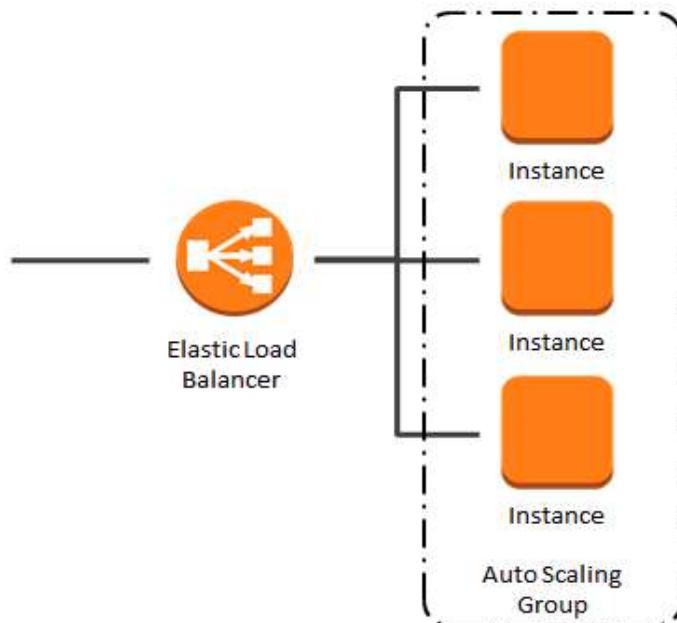
The screenshot shows the 'Databases' page in the AWS RDS console. The sidebar has links for Dashboard, Databases, Query Editor, etc. The main area shows a table for 'Databases (1)'. The table has columns: DB identifier, Status, Role, Engine, Region & AZ, Size, and Recommendations. One row is listed: 'database-1' (Status: Available, Instance: MySQL Community, Engine: MySQL Community, Region & AZ: ap-south-1a, Size: db.t3.micro).

10) ELB, ASG

Elastic Load Balancing automatically distributes your incoming traffic across multiple targets, such as EC2 instances, containers, and IP addresses, in one or more Availability Zones.

It monitors the health of its registered targets, and routes traffic only to the healthy targets.

ELB and ASG:



ASG:

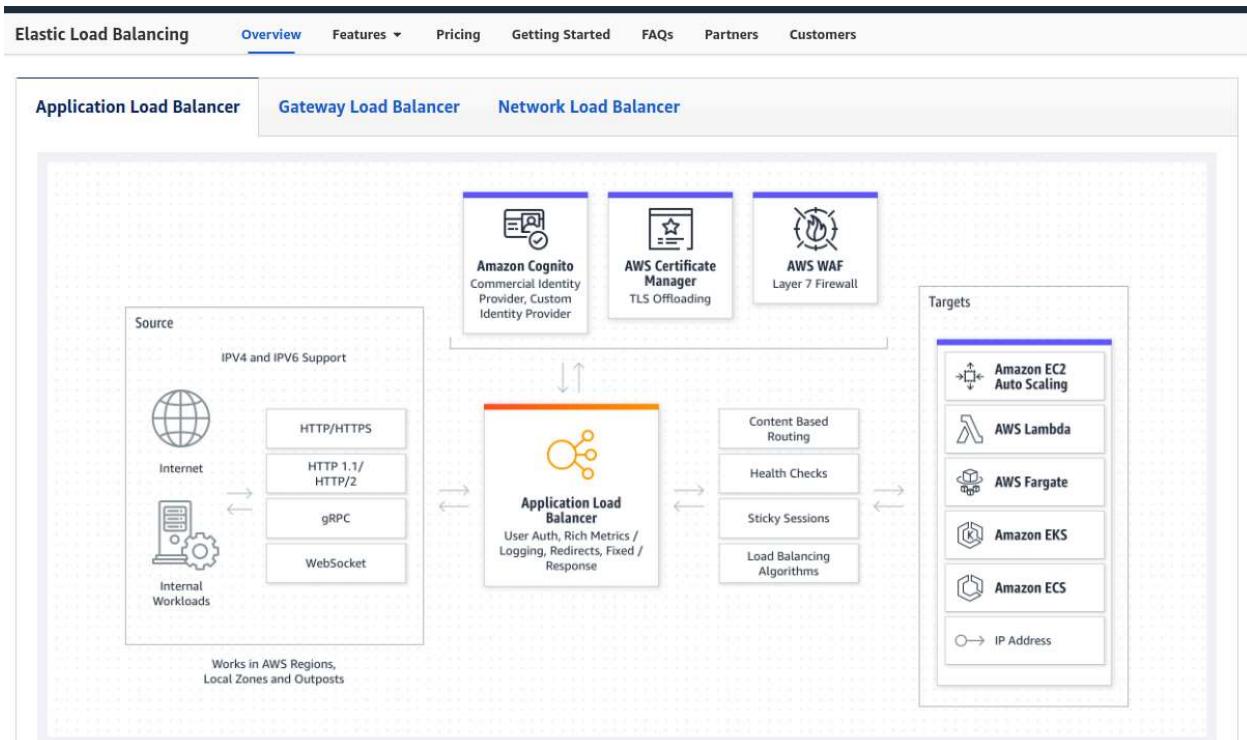
AWS Auto Scaling lets you build scaling plans that automate how groups of different resources respond to changes in demand.

Automatically launch or terminate EC2 instances based on user-defined policies, health status checks, and schedules resources.

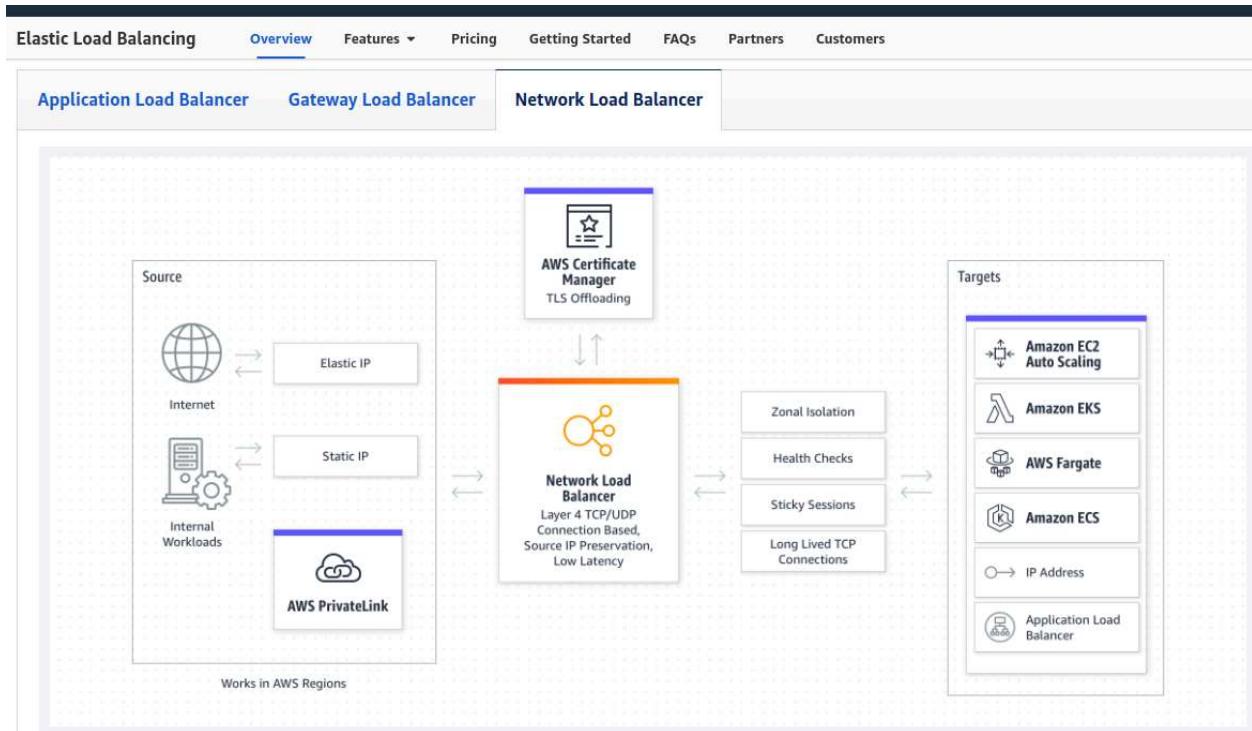
Type Of AWS ELB:

| Feature | Application Load Balancer | Network Load Balancer | Gateway Load Balancer | Classic Load Balancer |
|--------------------------------|---------------------------|---|--|---------------------------|
| Load Balancer type | Layer 7 | Layer 4 | Layer 3 Gateway + Layer 4 Load Balancing | Layer 4/7 |
| Target type | IP, Instance, Lambda | IP, Instance, Application Load Balancer | IP, Instance | |
| Terminates flow/proxy behavior | Yes | Yes | No | Yes |
| Protocol listeners | HTTP, HTTPS, gRPC | TCP, UDP, TLS | IP | TCP, SSL/TLS, HTTP, HTTPS |
| Reachable via | VIP | VIP | Route table entry | |

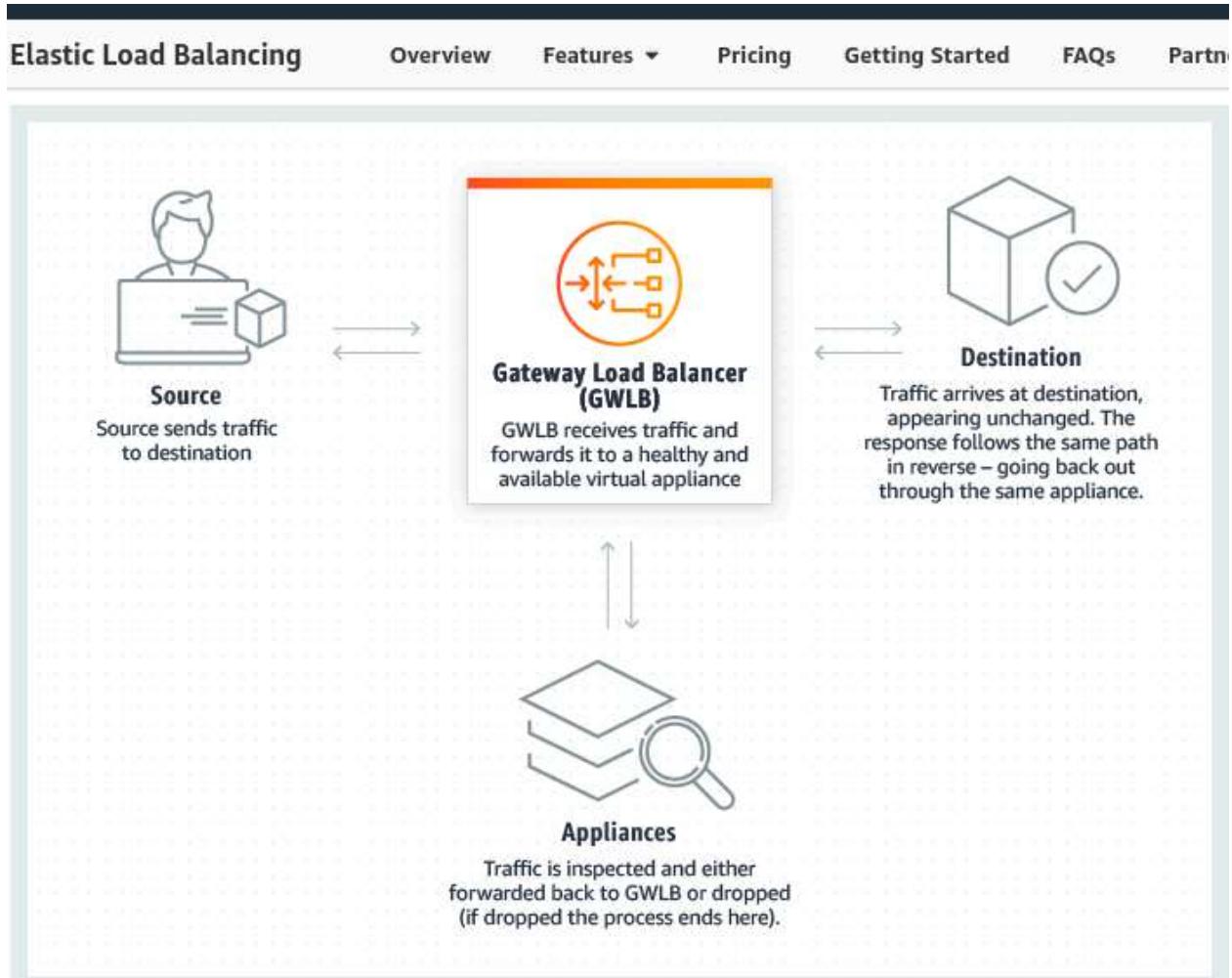
Application Load Balancers: Application Load Balancers are used to route HTTP/HTTPS (or Layer 7) traffic. Network Load Balancers and Classic Load Balancers are used to route TCP (or Layer 4) traffic.



Network Load Balancer: NLB is a layer 4 (TCP/UDP) load balancer that distributes traffic across multiple targets, such as EC2, containers, and IP addresses, within one or more Availability Zones. It can handle millions of requests per second.



Gateway Load Balancer: GLB helps you easily deploy, scale, and manage your third-party virtual appliances. It gives you one gateway for distributing traffic across multiple virtual appliances while scaling them up or down, based on demand.



Classic Load Balancer: You can load balance HTTP/HTTPS applications and use Layer 7-specific features, also You can also use Layer 4 load balancing for applications on the TCP protocol using CLB.

AWS recommend Application Load Balancer for Layer 7 traffic and Network Load Balancer for Layer 4 traffic.

Run an ALB:

Lunch Multiple EC2:

The screenshot shows the AWS CloudWatch Metrics interface. At the top, there's a table titled "Instances (2/2) Info" listing two EC2 instances: "server-1" and "server-2". Both instances are running t2.micro instances in the ap-south-1a availability zone, with Public IPv4 DNS and Public IPv4 addresses listed. Below the table, a message says "Instances: i-00ad9091212c65d70 (server-2), i-0d58dff2277bbdeb (server-1)". The main area is titled "Monitoring" and contains four line charts: "CPU utilization (%)" showing a sharp drop from ~0.59% to ~0.48% at 08:30; "Status check failed (any) (count)" showing one failure at 08:30; "Status check failed (instance) (count)" showing one failure at 08:30; and "Status check failed (system) (count)" showing one failure at 08:30.

Create a target group for request:

The screenshot shows the AWS Lambda Target groups page. It lists a single target group named "cluster-1" with ARN "arn:aws:elasticloadbalancing:ap-south-1:123456789012:targetgroup/cluster-1/12345678901234567890", port 80, protocol HTTP, target type Instance, and no associated load balancer or VPC. Below this, the "Targets" tab is selected for "cluster-1", showing two registered targets: "server-1" and "server-2", both with instance ID "i-0d58dff2277bbdeb" and port 80, located in the ap-south-1a zone. The health status for both is "Unused" and "Normal".

Create Load balancer:

Screenshot of the AWS CloudFront console showing the "Compare and select load balancer type" page.

The page displays three load balancer types:

- Application Load Balancer Info**: Handles traffic from a web browser (HTTP and HTTPS) and routes it to Lambda functions, API Gateways, and Amazon EC2 instances.
- Network Load Balancer Info**: Handles traffic from a VPC (TCP, UDP, TLS) and routes it to Amazon VPC endpoints, ALBs, and Amazon Lambda functions.
- Gateway Load Balancer Info**: Handles traffic from a VPC (HTTP, HTTPS) and routes it to AWS services like AWS Lambda, Amazon API Gateway, and Amazon VPC endpoints.

Screenshot of the AWS CloudFront console showing the "Load balancers" page.

The page lists one load balancer:

| Name | DNS name | State | VPC ID | Availability Zones | Type | Date created |
|--------|-----------------------------|--------|-----------------------|----------------------|-------------|--------------------------------------|
| my-slb | my-slb-1753267658.ap-sou... | Active | vpc-0853810a86fc473df | 2 Availability Zones | application | December 25, 2023, 14:54 (UTC+06:00) |

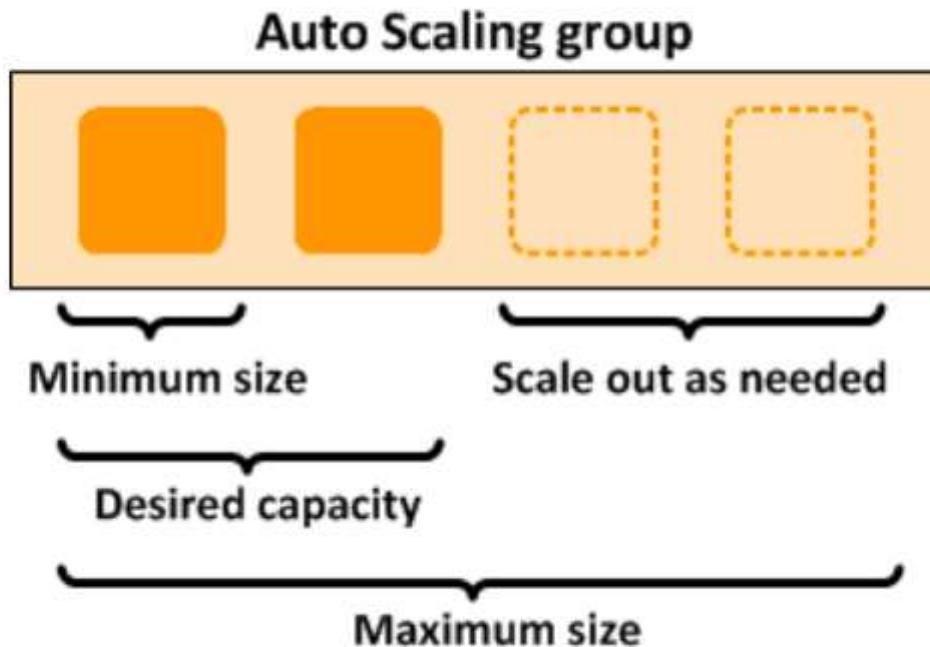
The "Load balancer: my-slb" dashboard shows the following metrics over the last hour:

- Target Response Time**: 1e-3 seconds.
- Requests**: Count 49, 24.5, 0.
- Rule Evaluations**: No unit, 1, 0.5, 0.
- HTTP 5XXs**: No unit, 1, 0.5, 0.

EC2 Auto Scaling:

Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. You create collections of EC2 instances, called Auto Scaling groups. You can specify the mini/Max number of instances in each Auto Scaling group.

For example, the following Auto Scaling group has a minimum size of one instance, a desired capacity of two instances, and a maximum size of four instances.



AWS ASG doc:

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html>

11) AWS Managed Services

AWS Managed Services aims to simplify the management of AWS infrastructure by providing a set of automated tools and processes.

It helps organizations achieve operational excellence by handling routine tasks such as patch management, monitoring, security, and backups.



Manage Database:

AWS takes care of routine database tasks such as patching, backups, and scaling for popular relational database engines like MySQL, PostgreSQL, DynamoDB, and others.

Amazon EKS (Elastic Kubernetes Service): A managed Kubernetes service that makes it easier to deploy, manage, and scale containerized applications using Kubernetes.

AWS Fargate(Managed Container Services:): A serverless compute engine for containers that allows you to run containers without managing the underlying infrastructure.

AWS Identity and Access Management (IAM): and AWS Key Management Service (KMS): A managed service for creating user/key and controlling the encryption keys used to encrypt data.

File Storage, CDN, Backup Services, Managed IoT Services, Managed Machine Learning Services and more.

Aurora (MySQL Compatible)



Aurora (PostgreSQL Compatible)



MySQL



MariaDB



PostgreSQL



Oracle

ORACLE

Microsoft SQL Server



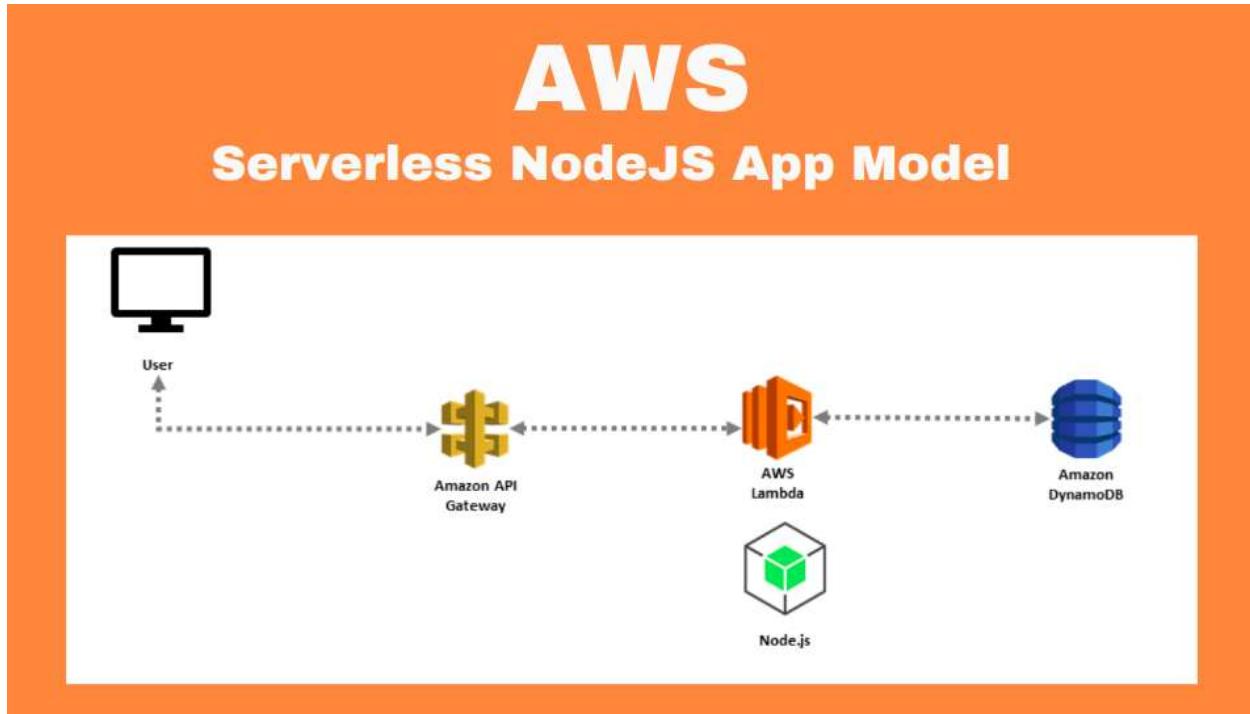
IBM Db2

IBM Db2

12) AWS Serverless

Instead of provisioning and managing servers AWS Serverless refers to a computing model where you can build and run applications without managing servers.

In a serverless architecture automatically handles the infrastructure for you.



AWS Lambda: AWS Lambda is a serverless compute service that lets you run your code without provisioning or managing servers. You upload your code, and Lambda automatically scales and executes.

Amazon API Gateway: This service allows you to create, publish, and manage APIs API at any scale.

With API Gateway, you can create RESTful APIs that integrate with Lambda functions, enabling you to build serverless backend services for your applications.

Amazon DynamoDB: A fully managed NoSQL database, DynamoDB can automatically scale based on demand, and you can use it to store and retrieve data for your serverless applications.

Amazon S3, AWS Step Function, AWS Cognito, AWS Secrets Manager, CloudFront, SNS, SQS and more.

AWS Lambda:

The screenshot shows the 'Create function' wizard in the AWS Lambda console. The 'Basic information' section is active, displaying fields for Function name (set to 'simple-node-api'), Runtime (Node.js 20.x), and Architecture (x86_64). The 'Code source' tab is selected, showing the code editor with the following Node.js code:

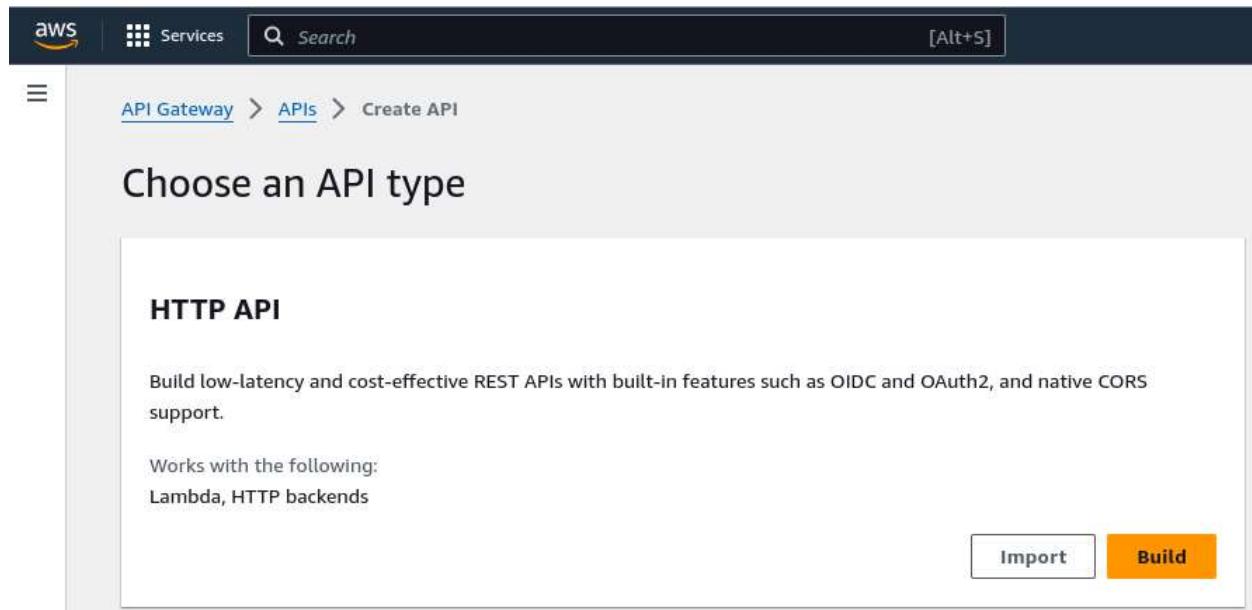
```
1 export const handler = async (event) => {
2   //
3   // Business logic will be here
4   //
5   //
6   //
7   //
8   //
9   //
10  return "This is my final response data !";
11};
12
```

Function for execute:

The screenshot shows the AWS Lambda function configuration interface. The 'Code source' tab is selected, displaying the Node.js code editor with the same code as shown in the previous screenshot. The 'Deploy' button is highlighted in blue, indicating that changes have not been deployed yet. The status bar at the bottom right shows 'Changes not deployed'.

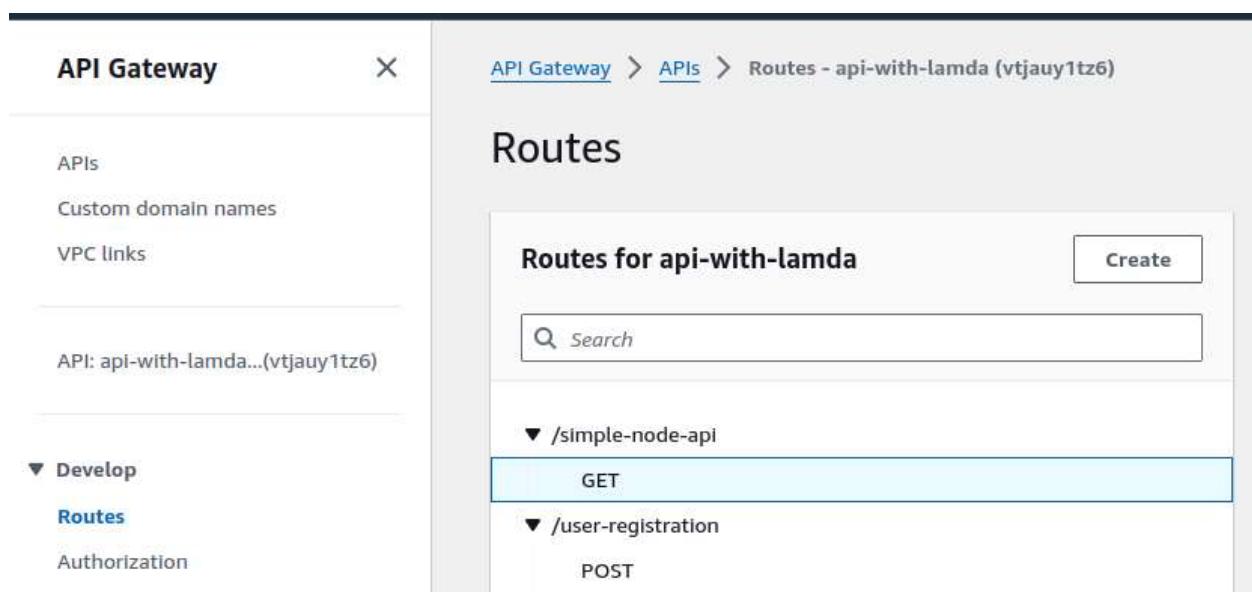
Amazon API Gateway:

Amazon API Gateway helps developers to create and manage APIs to back-end systems running on Amazon EC2, AWS Lambda, or any publicly addressable web service. With Amazon API Gateway, you can generate custom client SDKs for your APIs, to connect your back-end systems to mobile, web, and server applications or services.



The screenshot shows the AWS Management Console with the API Gateway service selected. The navigation bar includes the AWS logo, Services, a search bar, and a keyboard shortcut [Alt+5]. The current path is API Gateway > APIs > Create API. The main content area is titled "Choose an API type" and features a section for "HTTP API". It describes building low-latency and cost-effective REST APIs with built-in features like OIDC and OAuth2, and native CORS support. It also notes compatibility with Lambda and HTTP backends. At the bottom right of this section are "Import" and "Build" buttons. The overall interface is clean with a light gray background and white cards for different API types.

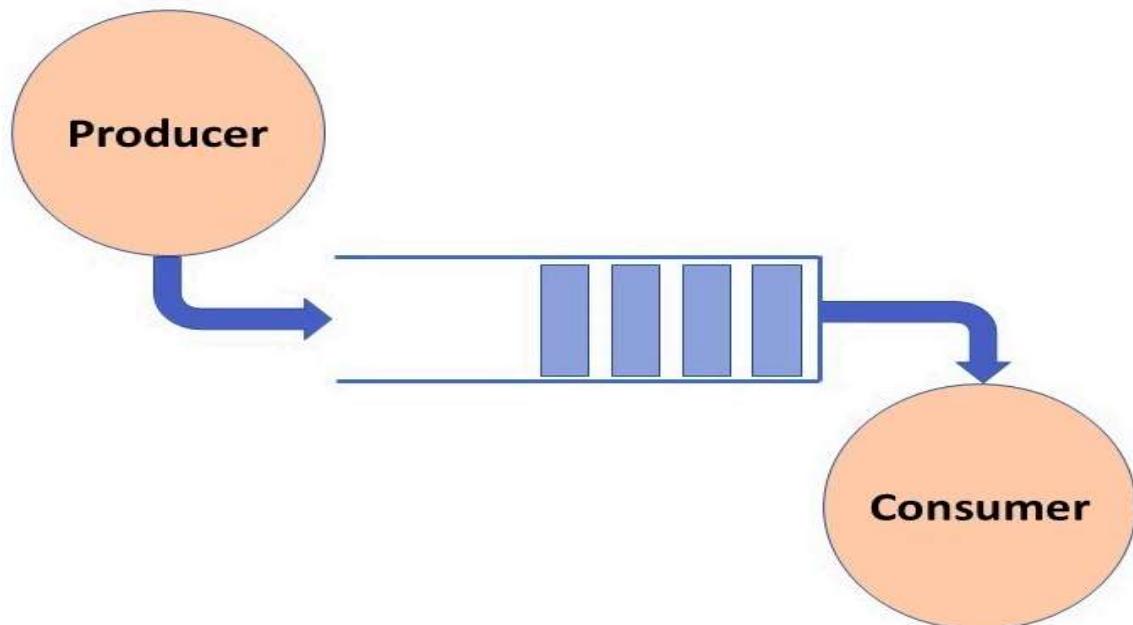
Create resource:



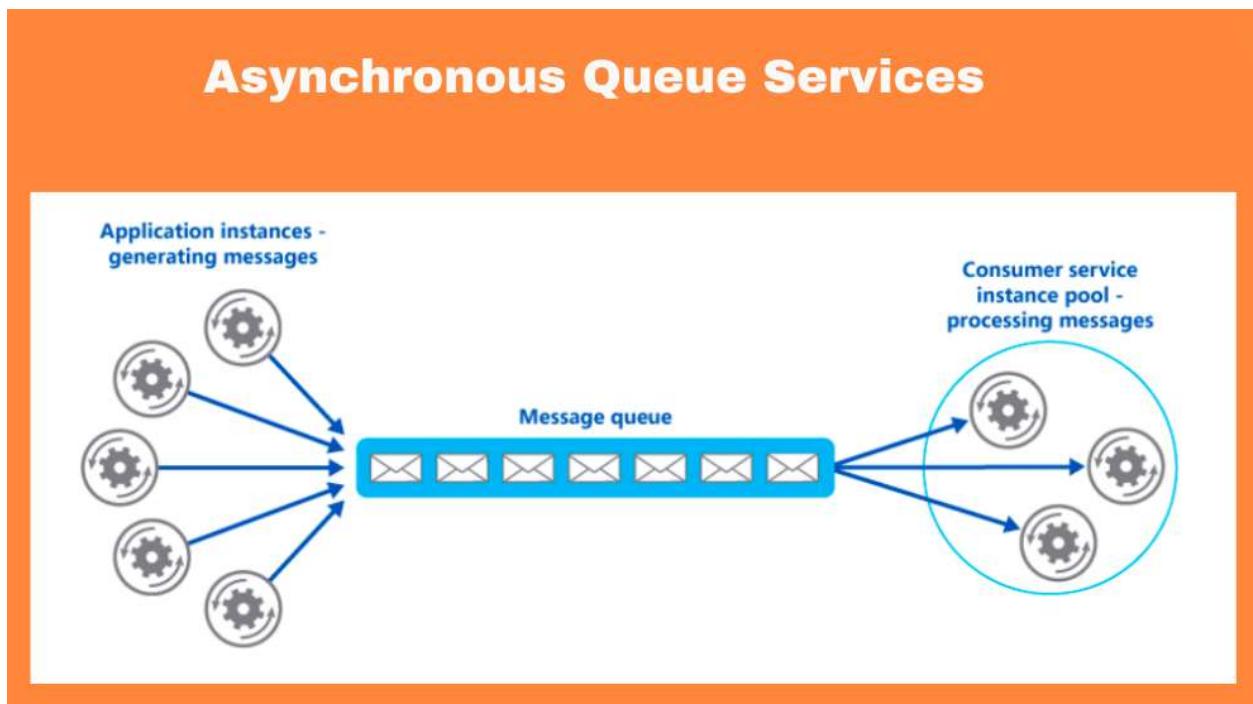
The screenshot shows the AWS Management Console with the API Gateway service selected. The navigation bar includes the AWS logo, Services, a search bar, and a keyboard shortcut [Alt+5]. The current path is API Gateway > APIs > Routes - api-with-lambda (vtjauy1tz6). The left sidebar has a tree view with "APIs", "Custom domain names", "VPC links", and "API: api-with-lambda...(vtjauy1tz6)" expanded. Under "Develop", "Routes" is selected. The main content area is titled "Routes" and shows a list of routes for the "api-with-lambda" API. One route, "/simple-node-api", is expanded to show methods: "GET" (selected) and "POST". Another route, "/user-registration", is shown below it. A "Create" button is located at the top right of the route list. The interface uses a dark header and light gray cards for the route details.

13) AWS Queue and Streams

Messaging and integration services that facilitate asynchronous communication between different components of a distributed system.

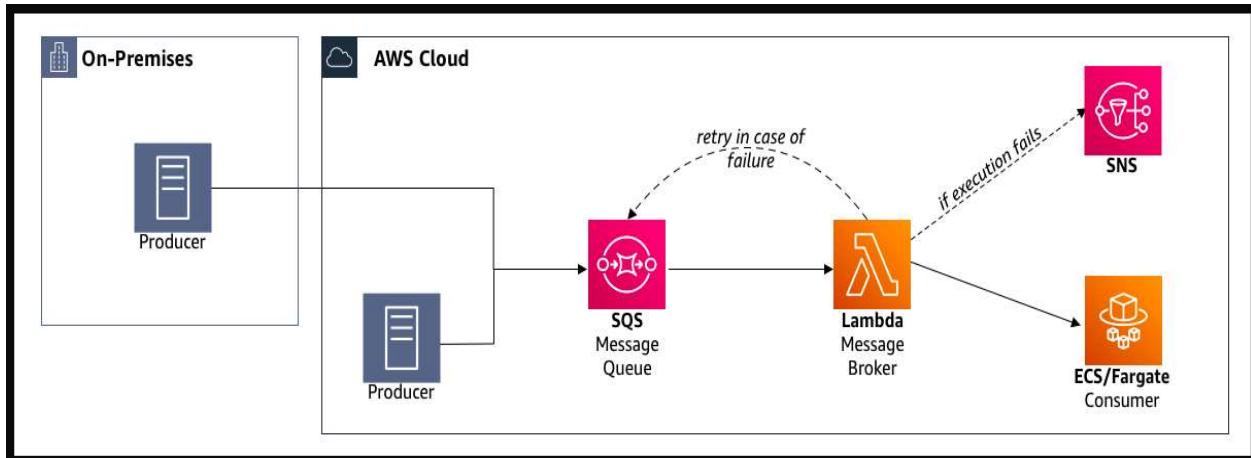


Amazon SQS, Amazon SNS and Kinesis are queue and streaming services that are highly scalable, simple to use, and don't require you to set up message brokers.



Amazon Simple Queue Service (SQS):

SQS is designed for reliable and scalable message queuing. Producers send messages to a queue, and consumers retrieve messages from the queue.

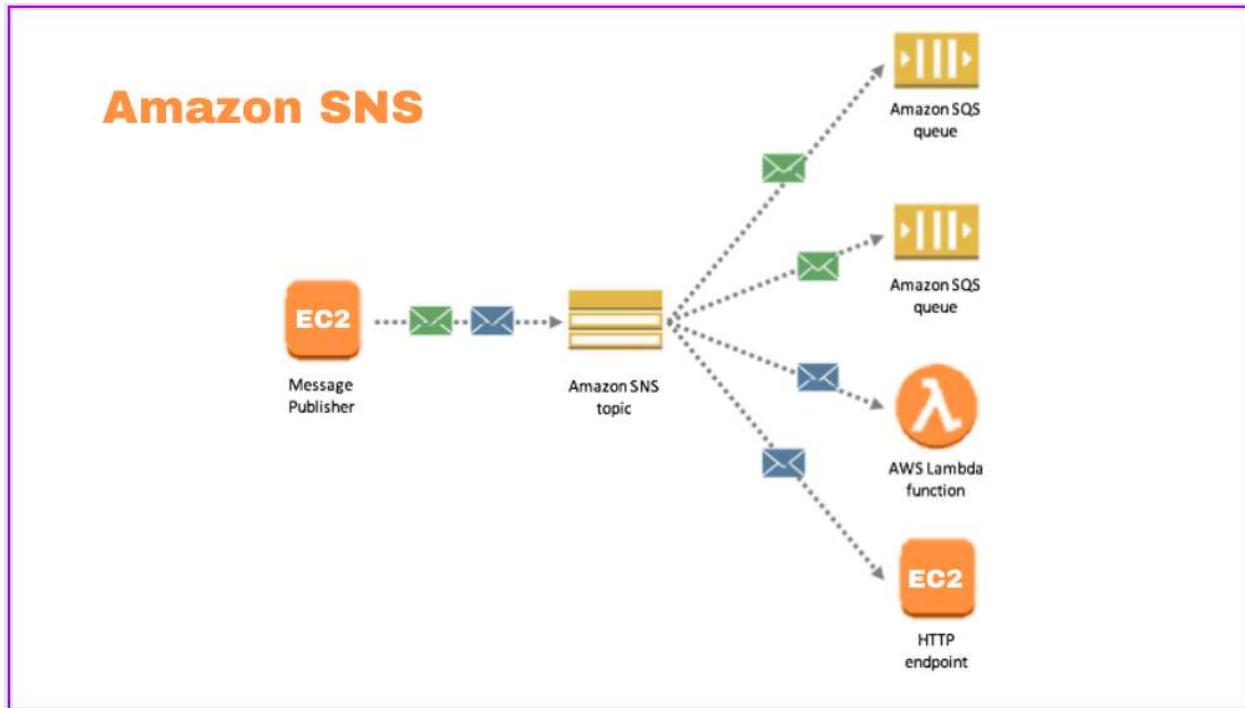


Key Features:

- Provides a reliable, highly scalable, and fully managed message queue service.
- Offers standard queues for high throughput and best-effort ordering.
- Suitable for decoupling and scaling microservices.

Amazon Simple Notification Service (SNS):

SNS is often used for the publish/subscribe model, where producers (publishers) send messages to a topic, and multiple consumers (subscribers) receive those messages.



Key Features:

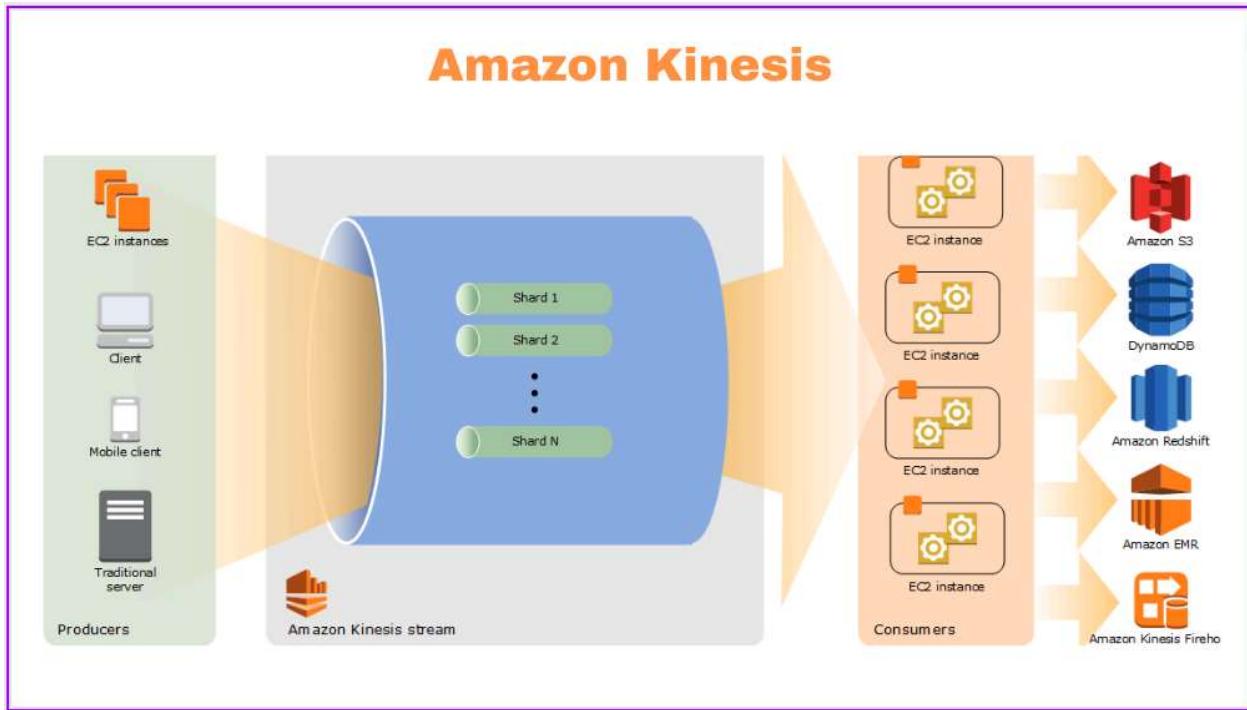
- Enables message and notification delivery to a large number of subscribers or endpoints.
- Supports publish/subscribe and fan-out scenarios, where a message published to a topic can be sent to multiple subscribers.
- Integrates with other AWS services, allowing you to automatically trigger actions based on events.

AWS SDK Doc Link:

https://docs.aws.amazon.com/code-library/latest/ug/python_3_sqs_code_examples.html

Amazon Kinesis Data Streams:

Kinesis Data Streams allows you to build custom applications that process or analyze streaming data in real-time.

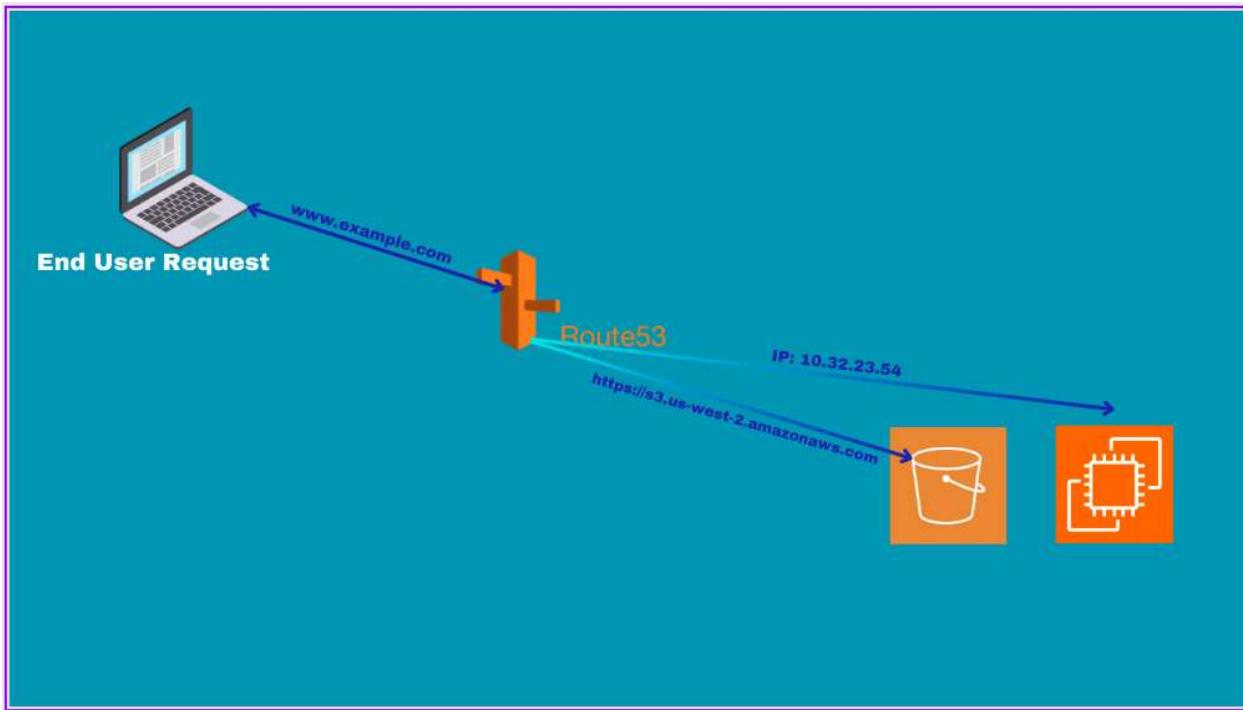


Key Features:

- Supports the creation of data streams to which producers can publish data, and consumers can subscribe to process the data.
- Enables real-time processing and analytics on streaming data.
- Integrates with various AWS services for downstream processing and storage.

14) Route53 and CDN

Route 53 allows you to register and manage domain names (e.g., example.com) directly through the service.



Amazon Route 53 is a scalable and highly available Domain Name System (DNS) web service provided by Amazon Web Services (AWS). It is designed to route end-user requests to globally distributed endpoints—whether those endpoints are in AWS or elsewhere.

key features and concepts:

Health Checks:

Route 53 can monitor the health of your resources and it can automatically adjust routing based on the health status.

Integration with AWS Services:

Route 53 integrates seamlessly with other AWS services. For example, you can use it to route traffic to resources like EC2, ELB.

Logs and Monitoring:

Route 53 provides detailed logs that include information about DNS queries, responses, and health checks. You can use these logs for monitoring and troubleshooting.

Private DNS:

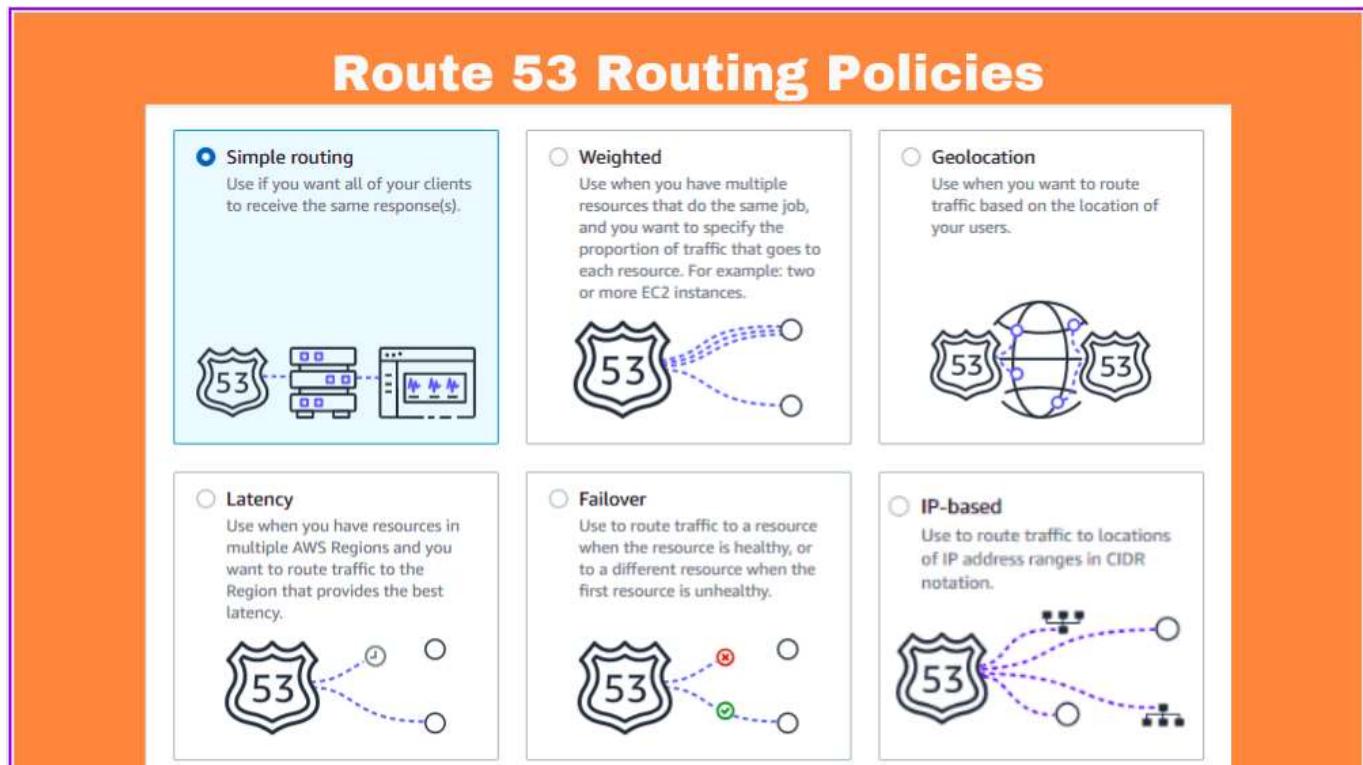
Route 53 supports private DNS, allowing you to create DNS records that are only accessible within your Amazon VPC (Virtual Private Cloud).

Traffic Flow Visualizations:

Route 53 provides visualizations of the traffic flow, which can be helpful in understanding how DNS queries are being routed.

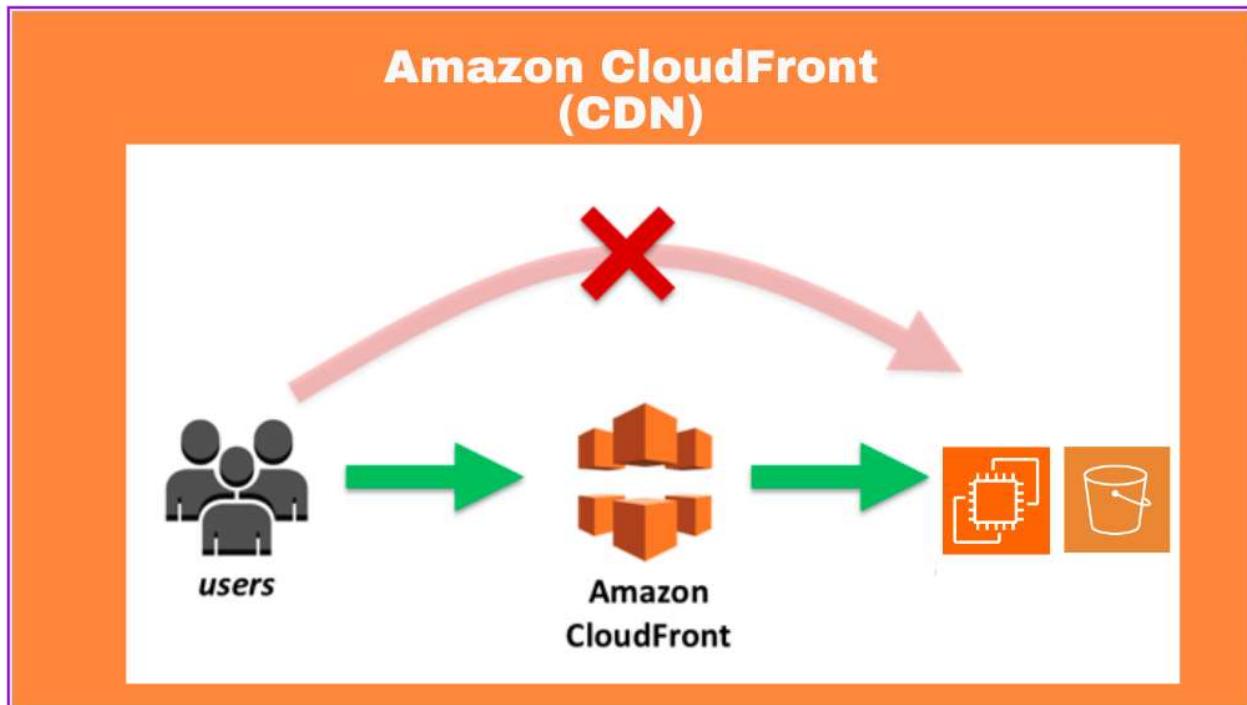
Routing Policies: Route 53 supports various routing policies, enabling you to define how traffic should be routed to your resources.

- **Simple Routing:** Maps a domain to a single resource.
- **Weighted Routing:** Distributes traffic based on specified weights.
- **Geolocation Routing:** Directs traffic based on the geographic location of the user.
- **Latency-based Routing:** Directs traffic based on the lowest latency for end-users.
- **Failover Routing:** Routes traffic to a standby resource in case the primary resource is unhealthy.
- **IP-based** routing policy is a network configuration strategy that directs incoming network traffic to specific destinations based on the Use and server location.



CDN:

Amazon Web Services (AWS) offers a content delivery network (CDN) service known as Amazon CloudFront. CloudFront is a globally distributed CDN that helps businesses deliver content, including web pages, videos, images, and other assets, with low latency and high transfer speeds.



key features and concepts:

Edge Locations:

CloudFront uses a network of edge locations strategically located around the world. These edge locations cache copies of your content, reducing the latency for end-users by serving the content from a location geographically closer to them.

Origins:

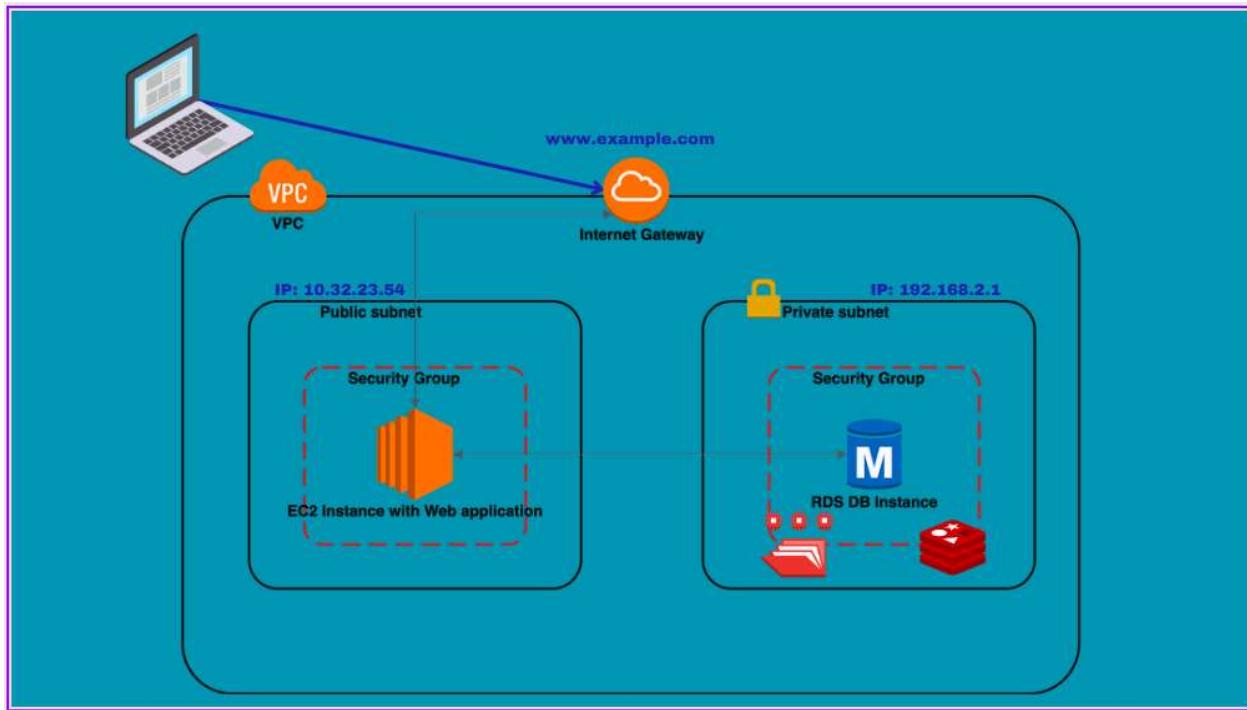
The origin is the location where your original files are stored. CloudFront can pull content from various types of origins, including Amazon S3 buckets, EC2 instances, or custom HTTP servers.

Security: CloudFront integrates with other AWS services to provide security features, including HTTPS support, custom SSL certificates, and integration with AWS Identity and Access Management (IAM) for access control.

Logging and Monitoring: CloudFront provides access logs that can be used for analysis and monitoring. Additionally, it integrates with AWS CloudWatch for real-time monitoring of CDN performance.

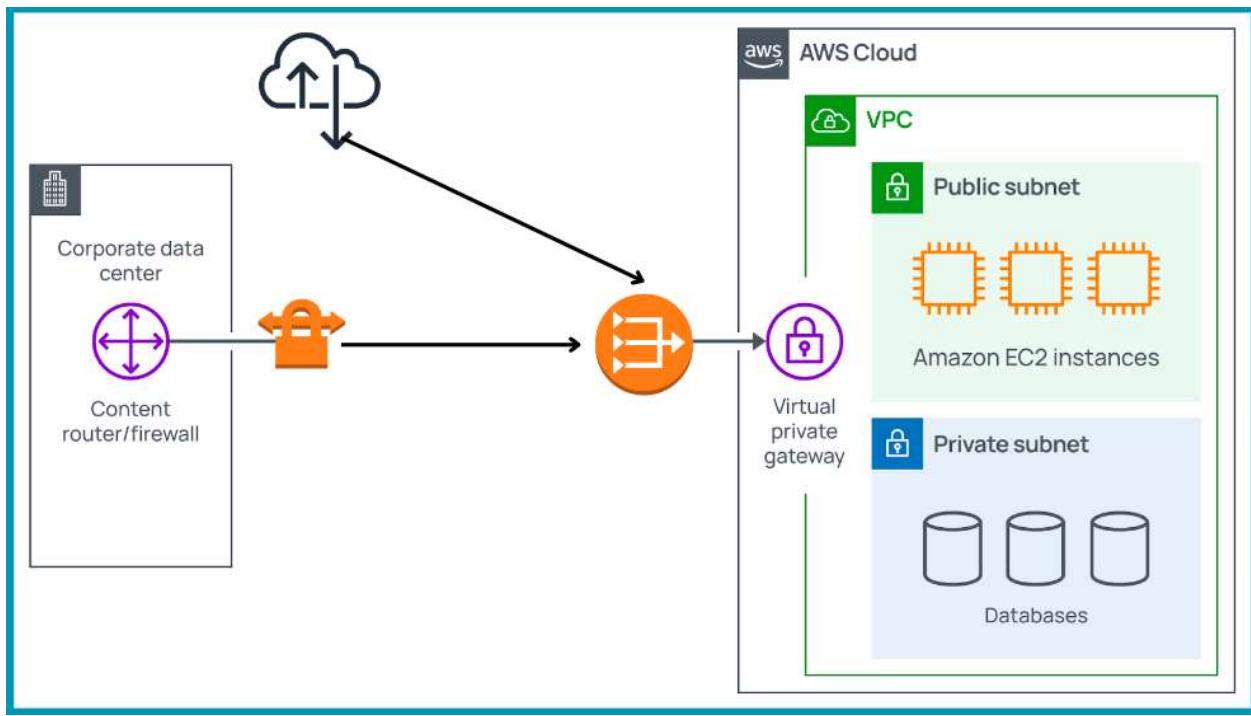
15) AWS Networking

AWS provides a variety of services designed to assist cloud consumers in establishing and **maintaining network** connectivity and **security** for their applications, both on-premises and in the cloud. These services include EC2, DB, and S3, among others.



Hybrid Networking:

In a hybrid network Organizations can connect their on-premises data centers to cloud using private Link, VPN or Internet.



AWS networking key services and components:

AWS networking and content delivery services

Network foundations

 **Amazon VPC**

Customize and control your networking environment with Amazon Virtual Private Cloud (VPC)

 **AWS Transit Gateway**

Simplify your network with VPCs and on-premises networks connected to a single gateway

 **AWS PrivateLink**

Establish private connectivity between VPCs and AWS or on-premises services

Application networking

 **Amazon VPC Lattice**

Simplify service-to-service connectivity, security, and monitoring

 **AWS AppMesh**

Connect containers and microservices with application-level networking

 **AWS API Gateway**

Create, maintain, and secure APIs at any scale

<https://aws.amazon.com/products/networking/>

Amazon Route 53:

A scalable and highly available Domain Name System (DNS) web service designed to route end-user requests to globally distributed AWS resources. For **Domain Registration, Health Checks** and **Traffic Flow control**.

Amazon Elastic IP Addresses:

Static IPv4 addresses designed for dynamic cloud computing, allowing you to host websites, web applications, and other services.

Amazon Virtual Private Network (VPN):

A VPN service that enables you to securely connect your on-premises data centers to AWS.

Site-to-Site VPN: Connects on-premises data centers to AWS VPCs.

Client VPN: Allows remote users to securely connect to AWS resources.

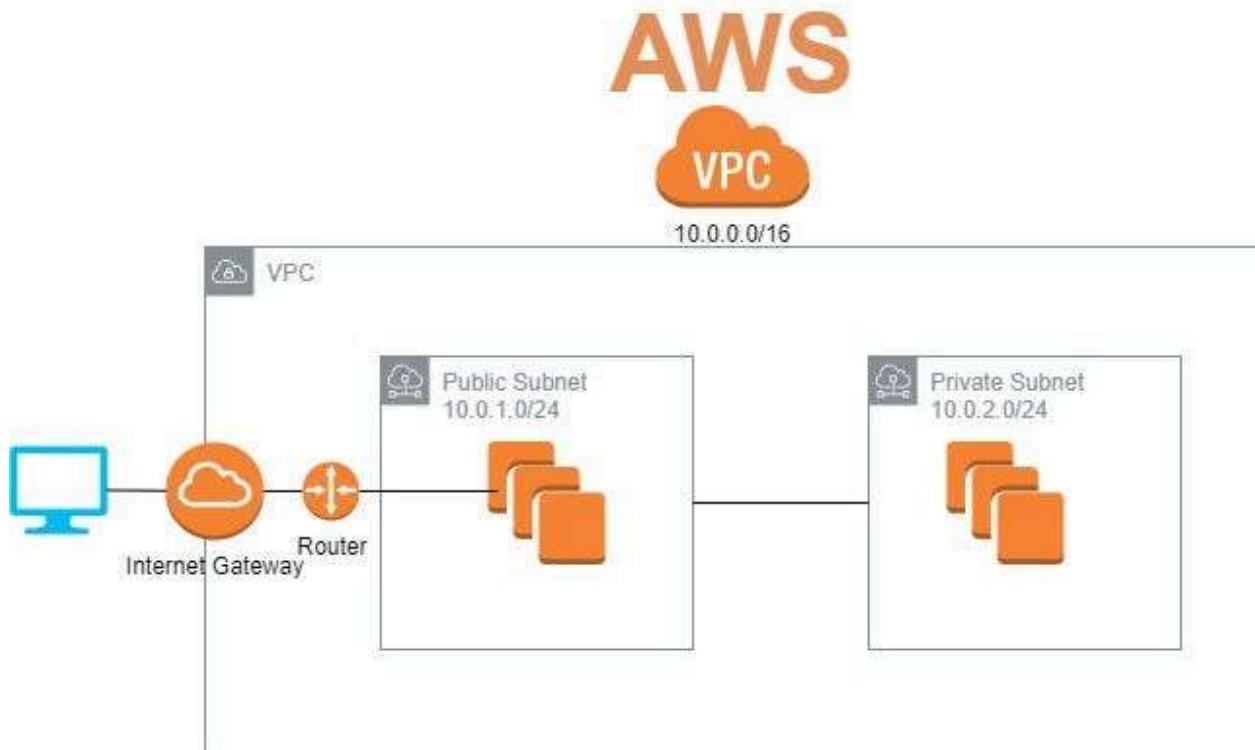
AWS Transit Gateway:

Simplifies network architecture and management by providing a hub-and-spoke model for connecting multiple VPCs and on-premises networks.

Amazon VPC (Virtual Private Cloud):

Amazon VPC enables you to launch AWS resources in a logically isolated section of the AWS Cloud, providing you with full control over the virtual network environment.

- **Subnets:** VPCs can be divided into subnets, allowing you to group resources based on security and operational requirements.
- **Route Tables:** Control the traffic between subnets using route tables.
- **Internet Gateway (IGW):** Allows communication between your VPC and the internet.
- **Elastic Load Balancer (ELB):** Distributes incoming application traffic across multiple targets, such as EC2 instances.



AWS reserved 5 (Five) IP address from each Subnet:

The first four IP addresses and the last IP address in each subnet CIDR block are not available for your use, its reserved.

1. 10.0.0.0: Network address.
2. 10.0.0.1: Reserved by AWS for the VPC router.
3. 10.0.0.2: Reserved by AWS for DNS.
4. 10.0.0.3: Reserved by AWS for future use.
5. 10.0.0.255: Network broadcast address (not support broadcast in a VPC).

Public Subnet:

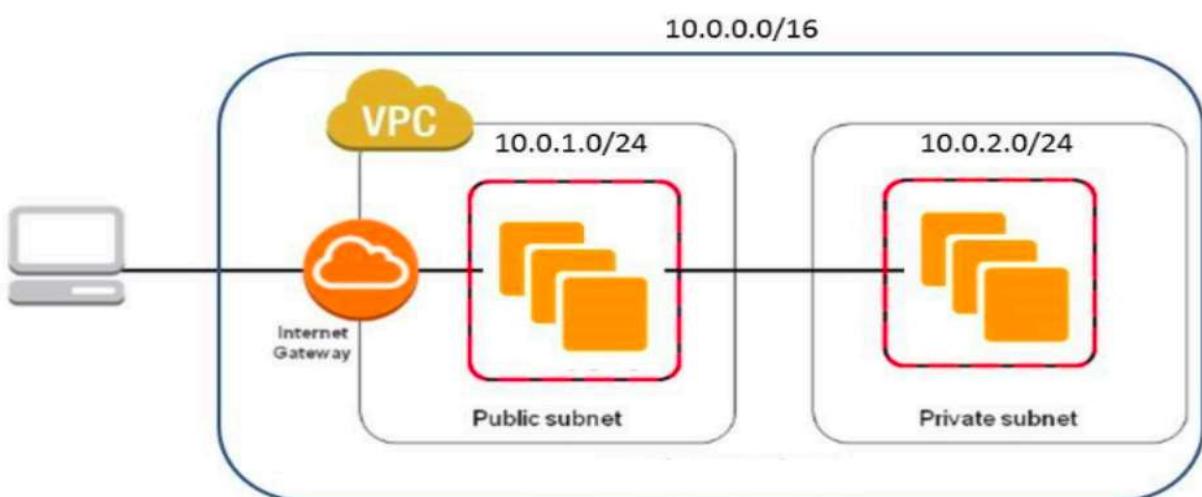
A public subnet is a subnet that has a route to the internet, typically through an Internet Gateway (IGW).

Common use cases for public subnets include hosting web servers, load balancers, and other resources that need to be publicly accessible.

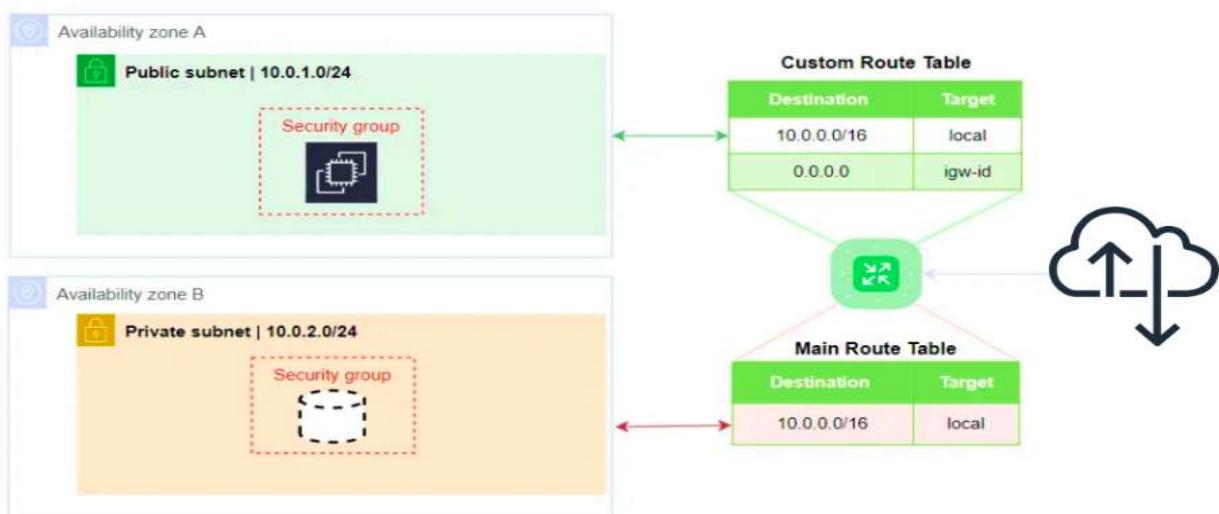
Private Subnet:

A private subnet is a subnet that does not have a direct route to the internet. It is isolated from the internet by default.

Common use cases for private subnets include databases, application servers, and other resources that should not be directly accessible from the internet.

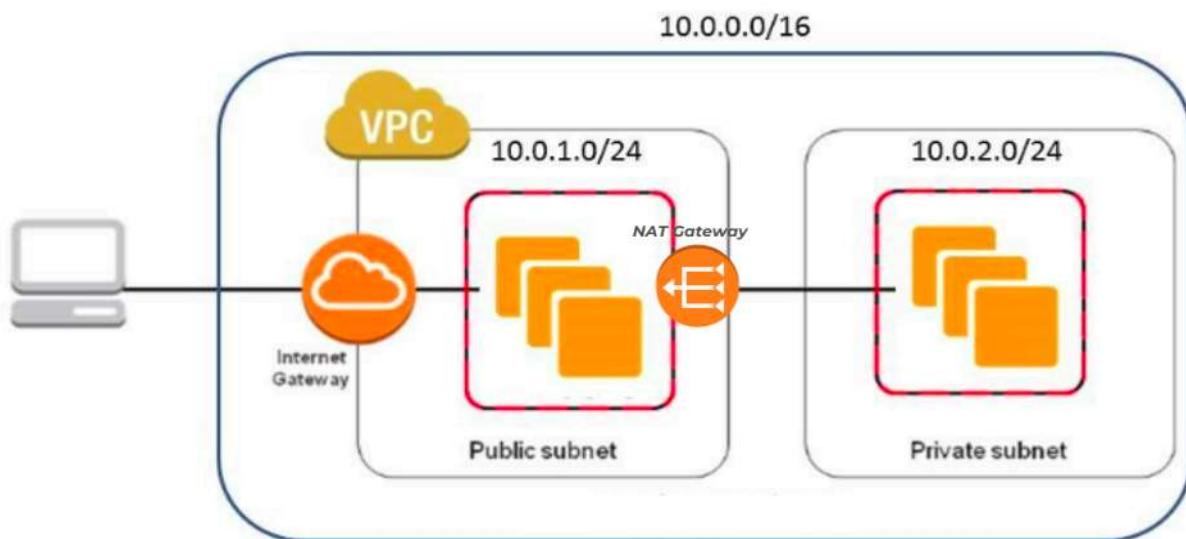


Route Table:



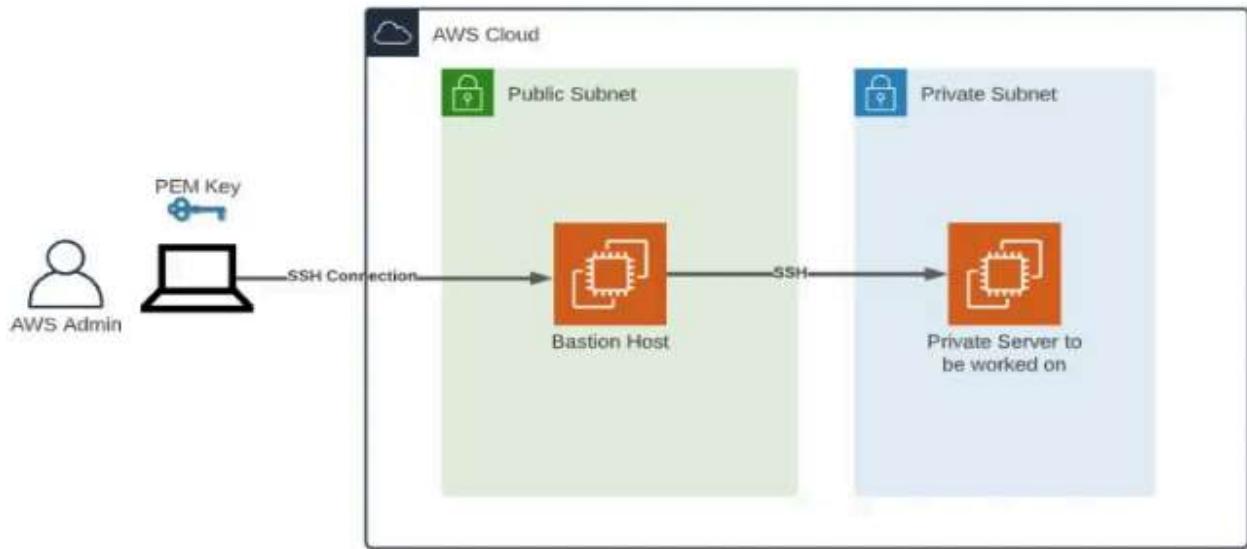
NAT Gateway/NAT Instance:

Instances in the private subnet do not have public IP addresses and rely on a NAT gateway or NAT instance for outbound internet access. Allows instances in a private subnet to initiate outbound traffic to the internet while preventing inbound traffic from the internet.



Bastion host:

A bastion host is a server whose purpose is to provide access to a private network from an external network, such as the Internet.



Create VPC:

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. Mouse over a resource to highlight the related resources.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

Name tag auto-generation Info
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

Auto-generate
my-custom

IPv4 CIDR block Info
Determine the starting IP and the size of your VPC using CIDR notation.

10.0.0.0/16 65,536 IPs
CIDR block size must be between /16 and /28.

IPv6 CIDR block Info
 No IPv6 CIDR block
 Amazon-provided IPv6 CIDR block

Tenancy Info
Default

Preview

VPC Show details
Your AWS virtual network
my-custom-vpc

Subnets (4)
Subnets within this VPC
ap-south-1a
my-custom-subnet-public1-ap-south-1a
my-custom-subnet-private1-ap-south-1a
ap-south-1b
my-custom-subnet-public2-ap-south-1b
my-custom-subnet-private2-ap-south-1b

Route tables (3)
Route network traffic to resources
my-custom-rtb-public
my-custom-rtb-private1-ap-south-1a
my-custom-rtb-private2-ap-south-1b

Network connections (2)
Connections to other networks
my-custom-igw
my-custom-vpc-s3

Create VPC workflow

Success

▼ Details

- ✓ Create VPC: vpc-0c2bbb1cbbbb67770 []
- ✓ Enable DNS hostnames
- ✓ Enable DNS resolution
- ✓ Verifying VPC creation: vpc-0c2bbb1cbbbb67770 []
- ✓ Create subnet: subnet-030e4a89c6ee48730 []
- ✓ Create subnet: subnet-0ad86087edf367a11 []
- ✓ Create Internet gateway: igw-0d64159ff77023529 []
- ✓ Attach Internet gateway to the VPC
- ✓ Create route table: rtb-027af5a65176add37 []
- ✓ Create route
- ✓ Associate route table
- ✓ Create route table: rtb-0a94fa586544a24a4 []
- ✓ Associate route table
- ✓ Verifying route table creation

16) Security and Monitoring

Amazon Web Services (AWS) provides a comprehensive set of services and features to help users secure their applications and data, as well as monitor their AWS resources.

Instances (1/1) [Info](#)

Find Instance by attribute or tag (case-sensitive)

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS |
|------------|---------------------|----------------|---------------|--------------|-------------------------------|-------------------|-----------------|
| my-web-app | i-0debda8a8f161ee72 | Running | t2.micro | Initializing | View alarms + | ap-south-1b | ec2-43-204-150 |

Instance: i-0debda8a8f161ee72 (my-web-app)

Details | Status and alarms [New](#) | **Monitoring** | Security | Networking | Storage | Tags

Manage detailed monitoring

Alarm recommendations

1h 3h 12h 1d 3d 1w Custom UTC timezone

| CPU utilization (%) | Network in (bytes) | Network out (bytes) | Network packets in (count) |
|---|---|---|---|
| No unit No data available. Try adjusting the dashboard time range. 0.5 | No unit No data available. Try adjusting the dashboard time range. 0.5 | No unit No data available. Try adjusting the dashboard time range. 0.5 | No unit No data available. Try adjusting the dashboard time range. 0.5 |

Identity and Access Management (IAM):

IAM allows you to manage access to AWS services securely. You can create and manage AWS users, groups, and roles, and define policies to grant or deny access to resources.

Web Application Firewall (WAF):

WAF helps protect web applications from common web exploits by allowing you to control which traffic can access your applications.

Security Groups and Network ACLs:

These are used to control inbound and outbound traffic at the **instance** and **subnet** level, respectively.

AWS CloudTrail:

CloudTrail records API calls made on your account, providing audit logs for compliance, security analysis, and resource tracking.

AWS Config:

AWS Config continuously monitors and records configurations of your AWS resources. It provides a history of resource changes and allows you to evaluate configurations against desired settings.

Amazon GuardDuty:

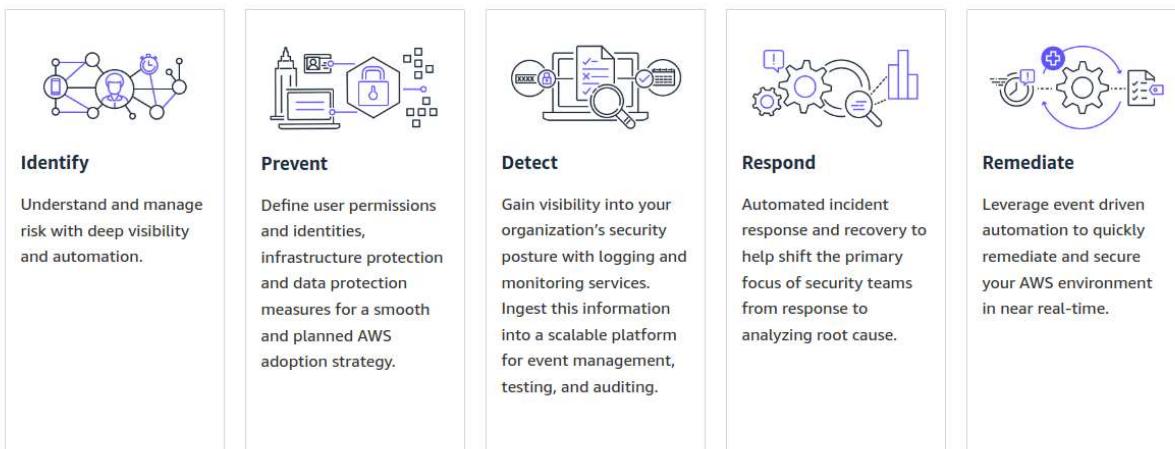
GuardDuty is a threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts and workloads.

AWS Secrets Manager:

Secrets Manager enables you to rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle.

Strategic Security

AWS is your guide in understanding and executing best practices to manage and reduce security risk, and protect your networks and data. Built by experts, AWS security, identity, and compliance services give you the confidence to keep building and innovating.



Encryption:

AWS offers various encryption services to protect data in transit and at rest. AWS Key Management Service (KMS) allows you to create and control encryption keys, while services like Amazon S3 and Amazon RDS provide encryption options.

Amazon CloudWatch:

CloudWatch is a monitoring and observability service that provides data and actionable insights for AWS resources. It can collect and track metrics, collect and monitor log files, and set alarms.

Amazon CloudWatch Events:

CloudWatch Events allow you to respond to changes in AWS resources. You can create rules that match events and route them to one or more target functions or streams.

AWS Personal Health Dashboard:

The Personal Health Dashboard provides alerts and remediation guidance when AWS is experiencing events that may impact your applications.

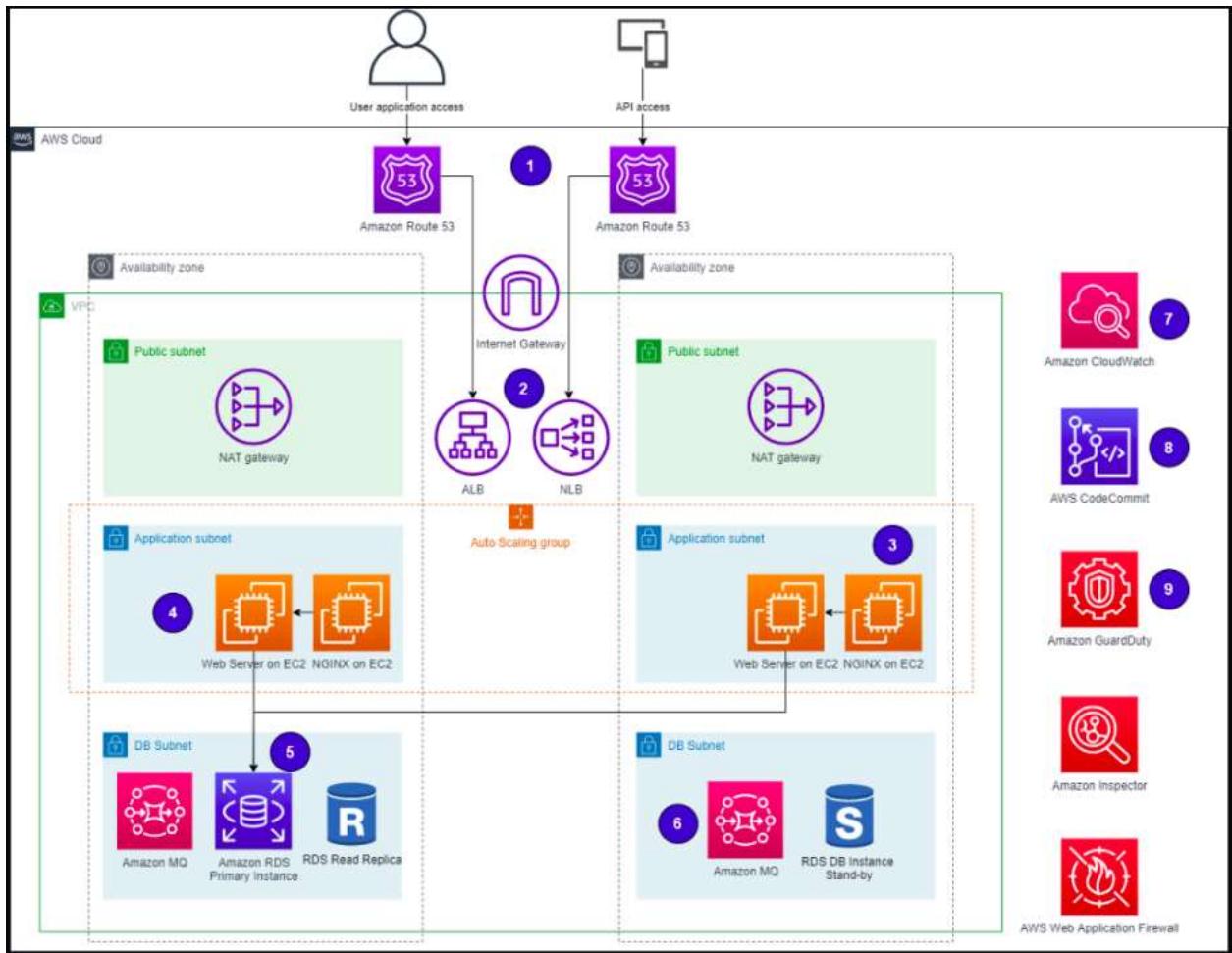
AWS Doc Link:

<https://aws.amazon.com/products/security/?nc=sn&loc=2&refid=14a4002d-4936-4343-8211-b5a150ca592>

AWS Security, Identity, & Compliance services

| Category | What Is It | AWS service |
|--|---|--|
| Identity and access management | Securely manage identities and access to AWS services and resources | AWS Identity and Access Management (IAM) |
| | Centrally manage workforce access to multiple AWS accounts and applications | AWS IAM Identity Center (successor to SSO) |
| | Implement secure, frictionless customer identity and access management that scales | Amazon Cognito |
| | Manage fine-grained permissions and authorization within custom applications | Amazon Verified Permissions |
| | Gain efficiency with a fully managed Microsoft Active Directory service | AWS Directory Service |
| | Simply and securely share your AWS resources across multiple accounts | AWS Resource Access Manager |
| Detection and response | Centrally manage your environment as you scale your AWS resources | AWS Organizations |
| | Protect AWS accounts with intelligent threat detection | Amazon GuardDuty |
| | Automated and continual vulnerability management at scale | Amazon Inspector |
| | Automate AWS security checks and centralize security alerts | AWS Security Hub |
| | Automatically centralize your security data in a few steps | Amazon Security Lake |
| | Analyze and visualize security data to investigate potential security issues | Amazon Detective |
| Compliance | Assess, audit, and evaluate configurations of your resources | AWS Config |
| | Observe and monitor resources and applications on AWS, on-premises, and on other clouds | Amazon CloudWatch |
| | Track user activity and API usage | AWS CloudTrail |

AWS Resource Monitoring:



17) AWS Machine Learning

Polly: Polly is a text-to-speech service that can convert written text into natural-sounding speech.

The screenshot shows the AWS Polly Text-to-Speech interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, a search bar, and a keyboard shortcut '[Alt+S]'. Below the navigation is a breadcrumb trail: 'Amazon Polly > Text-to-Speech'. The main title 'Text-to-Speech' has an 'Info' link. To the right are three buttons: 'Save to S3', 'Download', and 'Listen' (highlighted in orange). Under 'Engine' settings, 'Neural' is selected (radio button is checked), described as producing more natural and human-like speech than Standard Engine. There are also 'Long-Form' and 'Standard' options. In the 'Language' section, 'English, US' is chosen. In the 'Voice' section, 'Stephen, Male' is selected. The 'Input text' field contains the text 'Hi! My name is imran. I will read any text you type here.' On the far right, there's an 'SSML' section with an info link. The bottom of the interface shows a note about the Long-Form engine not being supported in every region, with a link to 'Feature and Region Compatibility'.

Translate: Translate is a machine translation service that supports the translation of text between languages.

The screenshot shows the AWS Translate Real-time translation interface. The top navigation bar includes the AWS logo, 'Services' dropdown, a search bar, and a keyboard shortcut '[Alt+S]'. The breadcrumb trail shows 'Amazon Translate > Real-time translation'. The main title 'Real-time translation' has an 'Info' link. A note below says 'Use real-time translations to deliver on-demand translation for text that you enter or a document that you upload.' The interface features a 'Translation' section with tabs for 'Text' (selected) and 'Document'. Under 'Text', 'Source language' is set to 'English (en)' and 'Target language' is set to 'Bengali (bn)'. The 'Enter text' field contains 'This is a AWS-Bangla video tutorial'. The 'Translated text' field shows the Bengali translation: 'এটি একটি AWS-বাংলা ভিডিও টিউটোরিয়াল'. At the bottom, a note states '36 characters, 36 of 10000 bytes used.' with an 'Info' link.

The screenshot shows the AWS Documentation website for the "Machine Learning (ML) and Artificial Intelligence (AI)" whitepaper. The top navigation bar includes the AWS logo, a search bar, and links for "Contact Us" and "English". The left sidebar contains a navigation tree with categories like Database, Developer tools, End user computing, etc., with "Machine Learning (ML) and Artificial Intelligence (AI)" being the selected category. The main content area features the title "Machine Learning (ML) and Artificial Intelligence (AI)", a green icon of a brain, and a "Topics" section listing various Amazon services. On the right side, there are social sharing icons (Facebook, Twitter, LinkedIn) and a purple hexagonal icon.

Machine Learning (ML) and Artificial Intelligence (AI)

PDF | RSS

Topics

- Amazon Augmented AI
- Amazon Bedrock
- Amazon CodeGuru
- Amazon CodeWhisperer
- Amazon Comprehend
- Amazon DevOps Guru
- Amazon Forecast
- Amazon Fraud Detector
- Amazon Comprehend Medical
- Amazon Kendra
- Amazon Lex
- Amazon Lookout for Equipment
- Amazon Lookout for Metrics
- Amazon Lookout for Vision
- Amazon Monitron

AWS Dock Link:

<https://docs.aws.amazon.com/whitepapers/latest/aws-overview/machine-learning.html>

SageMaker:

SageMaker is a fully managed service that allows you to build, train, and deploy machine learning models at scale.

Rekognition:

Rekognition is a computer vision service that can analyze images and videos for content. It can perform tasks such as facial recognition, object detection, and scene analysis.

Amazon Textract:

Textract is a service that automatically extracts text from scanned documents, images, and PDFs. and more.

18) AWS Well-Architected

The AWS Well-Architected Framework is a set of best practices and guidelines provided by Amazon Web Services (AWS) to help customers build secure, high-performing, resilient, and efficient infrastructure for their applications.



AWS Well-Architected helps cloud architects build secure, high-performing, resilient, and efficient infrastructure for a variety of applications and workloads. Built around six pillars—operational excellence, security, reliability, performance efficiency, cost optimization, and sustainability—AWS Well-Architected provides a consistent approach for customers and partners to evaluate architectures and implement scalable designs.

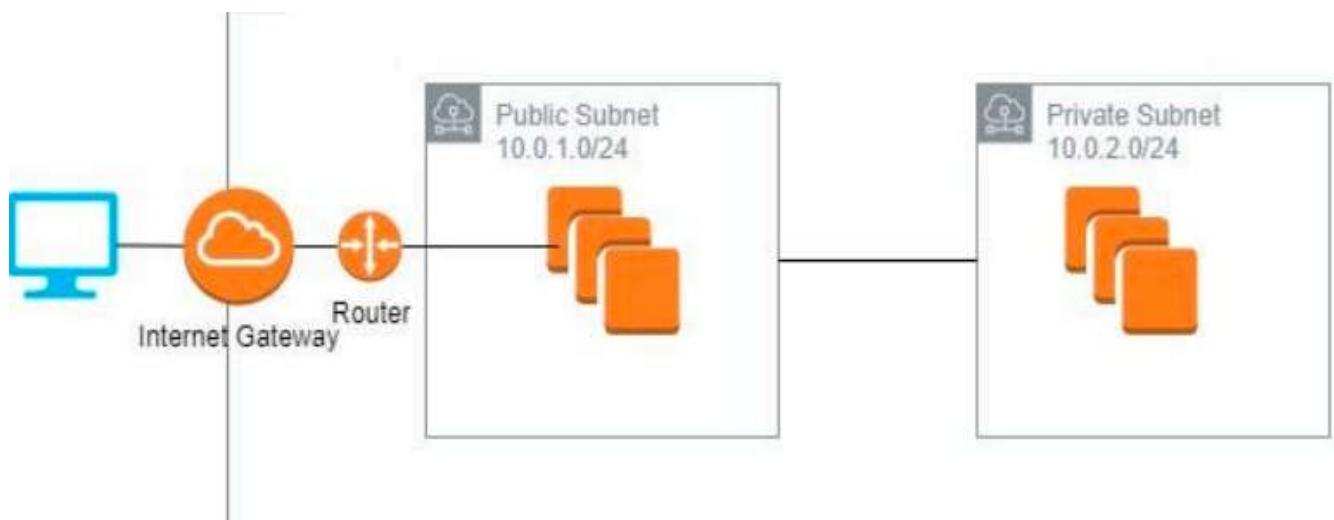
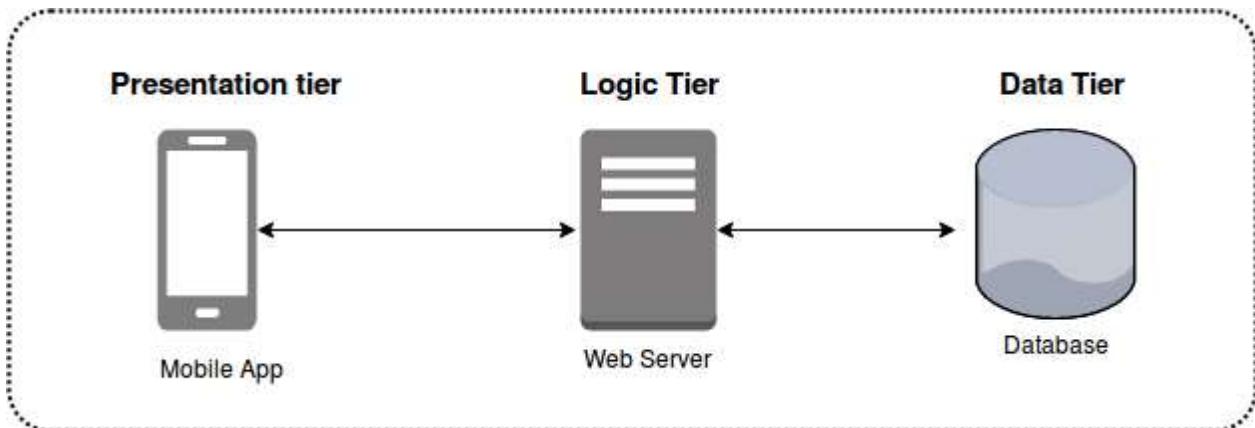
The AWS Well-Architected Framework includes domain-specific lenses, [hands-on labs](#), and the AWS Well-Architected Tool. The [AWS Well-Architected Tool](#), available at no cost in the [AWS Management Console](#), provides a mechanism for regularly evaluating workloads, identifying high-risk issues, and recording improvements.

AWS also provides access to an ecosystem of hundreds of members in the AWS Well-Architected Partner Program. [Engage a partner](#) in your area to help analyze and review your applications.

AWS Doc Link:

<https://aws.amazon.com/architecture/well-architected/?wa-lens-whitepapers.sort-by=item.additionalFields.sortDate&wa-lens-whitepapers.sort-order=desc&wa-guidance-whitepapers.sort-by=item.additionalFields.sortDate&wa-guidance-whitepapers.sort-order=desc>

19) Project run on AWS



Project Git-repository: <https://github.com/madbarsoft-github/aws-bangla-tut1>

20) AWS Certification

Training and Certification Get Trained ▾ Get Certified ▾ Develop Your Team AWS Partner Training ▾ Education Programs ▾ AWS Cloud Institute

Cloud Practitioner
FOUNDATIONAL

This certification validates cloud fluency and foundational AWS knowledge.

Exam price: 100 USD
Languages: English, Bahasa

Data Engineer
ASSOCIATE
Beta*

This certification validates skills and knowledge in implementing data pipelines and optimizing cost and performance on AWS.

Exam price: 75 USD*
Languages: English, French

Developer
ASSOCIATE

This certification validates the ability to write and deploy cloud-based applications.

Exam price: 150 USD
Languages: English, French

Solutions Architect
ASSOCIATE

This certification validates the ability to design and implement distributed systems on AWS.

Exam price: 150 USD
Languages: English, French

SysOps Administrator
ASSOCIATE

DevOps Engineer
PROFESSIONAL

Solutions Architect
PROFESSIONAL

Advanced Networking
SPECIALTY

Data Analytics
SPECIALTY
Retiring*

This certification validates expertise in using AWS data lakes and analytics services to get insights from data.

Exam price: 300 USD
Languages: English, French

Database
SPECIALTY

This certification validates expertise in recommending, designing, and maintaining optimal AWS database solutions.

Exam price: 300 USD
Languages: English, French

Machine Learning
SPECIALTY

This certification validates expertise in building, training, tuning, and deploying machine learning models on AWS.

Exam price: 300 USD
Languages: English, French

Security
SPECIALTY

This certification validates expertise in securing data and workloads in the AWS Cloud.

Exam price: 300 USD
Languages: English, French

SAP on AWS
SPECIALTY

This certification validates expertise in designing, implementing, migrating, and operating SAP workloads on AWS.

AWS Doc Link: <https://aws.amazon.com/certification/exams/>

FOUNDATIONAL

Knowledge-based certification for foundational understanding of AWS Cloud.

No prior experience needed.



PROFESSIONAL

Role-based certifications that validate advanced skills and knowledge required to design secure, optimized, and modernized applications and to automate processes on AWS.

2 years of prior AWS Cloud experience recommended.



ASSOCIATE

Role-based certifications that showcase your knowledge and skills on AWS and build your credibility as an AWS Cloud professional. **Prior cloud and/or strong on-premises IT experience recommended.**



SPECIALTY

Dive deeper and position yourself as a trusted advisor to your stakeholders and/or customers in these strategic areas. **Refer to the exam guides on the exam pages for recommended experience.**

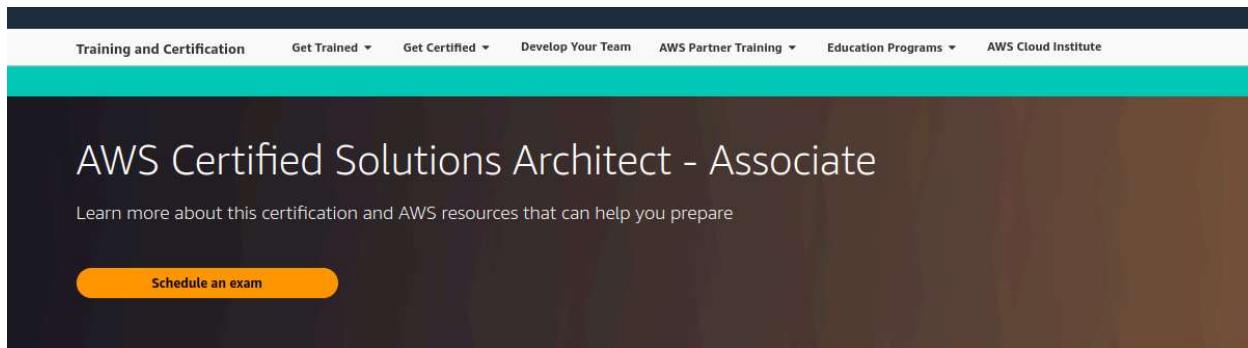


*AWS Certified Data Engineer - Associate is currently a beta exam.

*AWS Certified Data Analytics - Specialty is retiring in April 2024.

AWS Doc Link: https://aws.amazon.com/certification/?nc2=sb_ce_co

AWS Certified Solutions Architect - Associate:

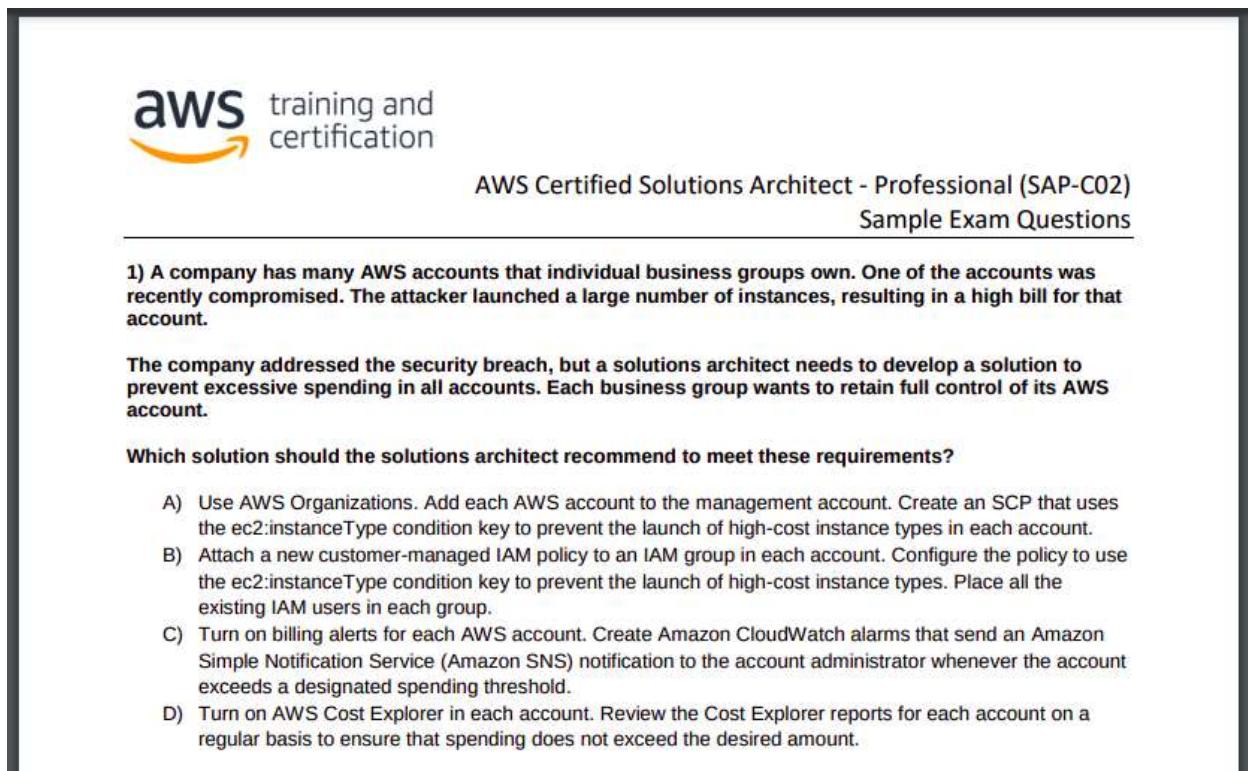


The screenshot shows the AWS Certified Solutions Architect - Associate landing page. At the top, there's a navigation bar with links: Training and Certification, Get Trained, Get Certified, Develop Your Team, AWS Partner Training, Education Programs, and AWS Cloud Institute. Below the navigation is a large teal header bar with the title "AWS Certified Solutions Architect - Associate". Underneath the title, a sub-header reads "Learn more about this certification and AWS resources that can help you prepare". A prominent yellow button labeled "Schedule an exam" is located below the sub-header. The main content area features a purple hexagonal badge with the AWS logo, the text "Solutions Architect", and "ASSOCIATE". To the right of the badge, a detailed description of the certification is provided, stating that it showcases knowledge and skills in AWS technology across a wide range of services, focusing on cost and performance optimization. It also mentions the AWS Well-Architected Framework and its benefits for career profile and earnings. A note below the badge specifies that the exam does not require deep hands-on coding experience but familiarity with basic programming concepts is an advantage.

AWS Certified Solutions Architect - Associate showcases knowledge and skills in AWS technology, across a wide range of AWS services. The focus of this certification is on the design of cost and performance optimized solutions, demonstrating a strong understanding of the AWS Well-Architected Framework. This certification can enhance the career profile and earnings of certified individuals and increase your credibility and confidence in stakeholder and customer interactions.

This exam does not require deep hands-on coding experience, although familiarity with basic programming concepts would be an advantage.

Question Example:



The screenshot displays a sample exam question from the AWS Certified Solutions Architect - Professional (SAP-C02) exam. The question is framed by a black border. At the top left is the AWS training and certification logo. To the right of the logo, the question title is "AWS Certified Solutions Architect - Professional (SAP-C02) Sample Exam Questions". The question itself is as follows:

1) A company has many AWS accounts that individual business groups own. One of the accounts was recently compromised. The attacker launched a large number of instances, resulting in a high bill for that account.

The company addressed the security breach, but a solutions architect needs to develop a solution to prevent excessive spending in all accounts. Each business group wants to retain full control of its AWS account.

Which solution should the solutions architect recommend to meet these requirements?

A) Use AWS Organizations. Add each AWS account to the management account. Create an SCP that uses the ec2:instanceType condition key to prevent the launch of high-cost instance types in each account.

B) Attach a new customer-managed IAM policy to an IAM group in each account. Configure the policy to use the ec2:instanceType condition key to prevent the launch of high-cost instance types. Place all the existing IAM users in each group.

C) Turn on billing alerts for each AWS account. Create Amazon CloudWatch alarms that send an Amazon Simple Notification Service (Amazon SNS) notification to the account administrator whenever the account exceeds a designated spending threshold.

D) Turn on AWS Cost Explorer in each account. Review the Cost Explorer reports for each account on a regular basis to ensure that spending does not exceed the desired amount.