Intro Of Cloud with Basic AWS Services

Sadhu Veera Mohan

Basic Introduction To AWS Cloud

Cloud computing has revolutionized the way businesses operate, enabling them to leverage scalable and flexible computing resources without the need to maintain physical infrastructure.

Before starting AWS we need to know about the cloud.

Introduction to Cloud Computing

Cloud computing is a technology that allows businesses and individuals to access computing resources (like servers, storage, databases, networking, software, and more) over the internet. Instead of owning and maintaining physical servers and infrastructure, users can rent these resources from cloud service providers. This model provides flexibility, scalability, and cost-efficiency, making it an attractive option for many organizations.

Before cloud users can buying and maintaining physical server its not cost effective and some times crashing servers the whole data will lost.

After introducing cloud users no need to maintain any server maintaing instead of using some cloud providers.The cloud providers will provide all the kind of services like like servers, storage, databases, networking, software, and more.

The Main cloud Providers are:

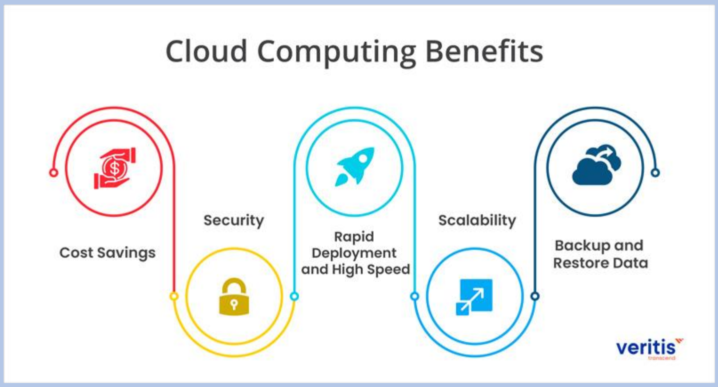
Amazon Web Services(AWS)

Azure

Google Cloud Provider(GCP)

Sadhu Veera Mohan

Key Benefits of Cloud Computing:



Cost-Effective:
Cloud services are typically offered on a pay-as-you-go model, meaning businesses only pay for what they use, helping them save money on hardware and maintenance costs.

Scalability:
Cloud computing provides easy scalability, meaning resources can be adjusted based on demand. Whether a company needs more storage or computing power, it can scale up or down quickly.

Flexibility:
With cloud computing, organizations can work from anywhere as long as they have an internet connection, providing mobility and flexibility.

Security:
Many cloud service providers offer robust security features such as encryption, firewalls, and multi-factor authentication, ensuring the safety of data stored on the cloud.

Reliability:
Cloud providers offer high uptime guarantees and data backup solutions to ensure that services remain available and data is protected in case of failure.

Types of Cloud Computing

Cloud computing services come in various types, depending on how resources are delivered and managed. These include **Cloud Deployment Models** and **Cloud Service Models**:

1. Cloud Deployment Models

These models define where and how cloud resources are deployed. There are three primary types:

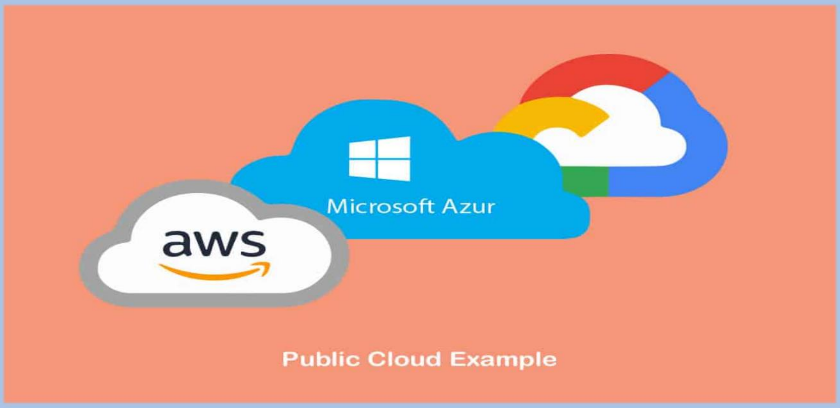
Public Cloud:

What it is: The public cloud is a cloud environment where the infrastructure and services are owned and operated by third-party providers (like Amazon Web Services, Microsoft Azure, Google Cloud). These resources are shared across multiple organizations.

How it works: Anyone can use public cloud services, and resources are made available over the internet. Businesses and individuals pay for what they use, with no need to manage or maintain the hardware.

Examples: AWS, Google Cloud, Microsoft Azure.

Best for: Small businesses or startups that need scalable resources without heavy investment in infrastructure.



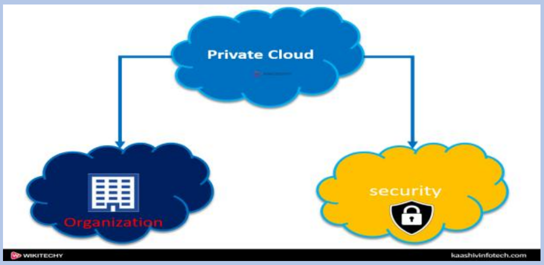
Private Cloud:

What it is: A private cloud is a cloud environment used exclusively by one organization. It can either be hosted on the company’s own premises or by a third-party provider.

How it works: The infrastructure is dedicated to one organization, giving it full control over the cloud environment, security, and data management.

Examples: An organization hosting its cloud infrastructure in its own data center or with a private cloud service like VMware or IBM.

Best for: Large enterprises or organizations with strict security and regulatory needs.



Sadhu Veera Mohan

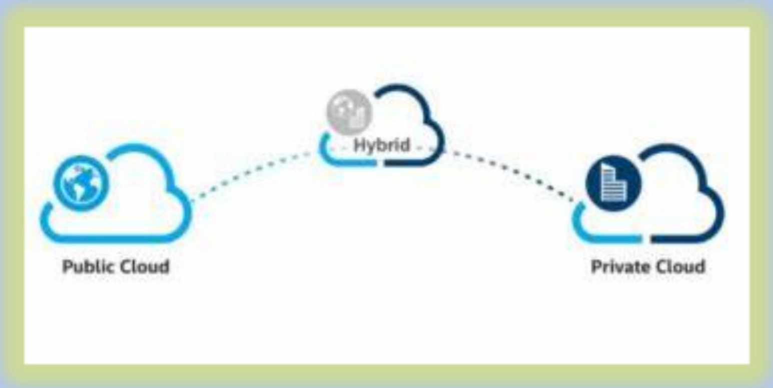
3. Hybrid Cloud:

What it is: A hybrid cloud combines both **public** and **private** clouds, allowing data and applications to be shared between them.

How it works: It offers the flexibility to keep sensitive workloads in the private cloud and other tasks keep in public cloud because they don't need any high security and compliance.

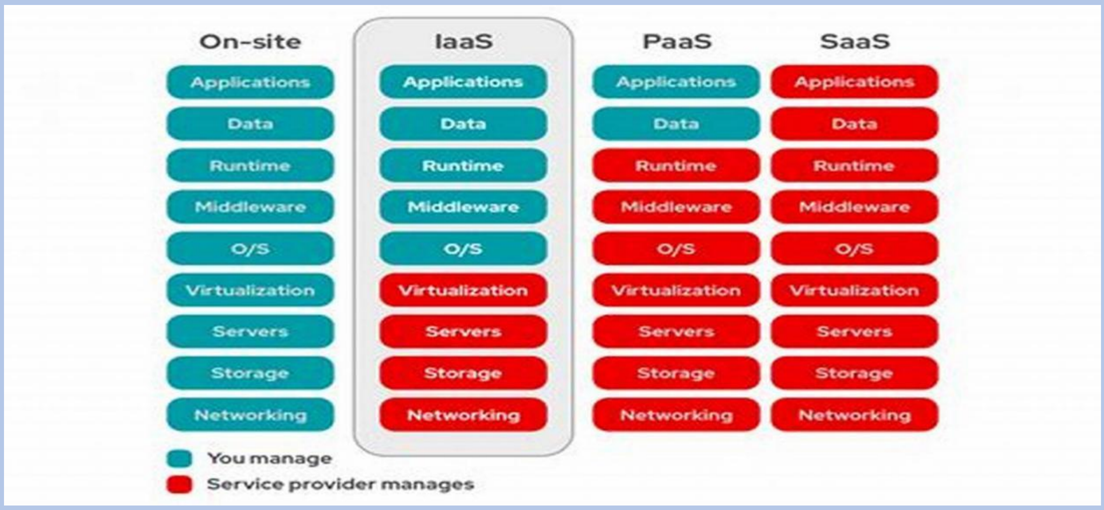
Examples: A business might keep its sensitive customer data in a private cloud but use a public cloud to run its website or process big data.

Best for: Businesses that need to balance security with flexibility and want to optimize their IT costs.



Cloud Service Types:

Cloud services are offered in different **types**, depending on what level of infrastructure or software you need. These are known as the **cloud computing models**:



On-site	IaaS	PaaS	SaaS
🟦 Applications	🟦 Applications	🟦 Applications	🟥 Applications
🟦 Data	🟦 Data	🟦 Data	🟥 Data
🟦 Runtime	🟦 Runtime	🟥 Runtime	🟥 Runtime
🟦 Middleware	🟦 Middleware	🟥 Middleware	🟥 Middleware
🟦 O/S	🟦 O/S	🟥 O/S	🟥 O/S
🟦 Virtualization	🟥 Virtualization	🟥 Virtualization	🟥 Virtualization
🟦 Servers	🟥 Servers	🟥 Servers	🟥 Servers
🟦 Storage	🟥 Storage	🟥 Storage	🟥 Storage
🟦 Networking	🟥 Networking	🟥 Networking	🟥 Networking

🟦 You manage
🟥 Service provider manages

Infrastructure as a Service (IaaS):

What it is: IaaS provides virtualized computing resources over the internet. It’s like renting IT infrastructure (like servers, storage, and networking) rather than owning it.

How it works: Users can rent infrastructure and set up their own virtual machines, networks, and storage as needed. The provider manages the physical hardware, but the user has control over their software and applications.

Examples: Amazon Web Services (AWS), Microsoft Azure, Google Compute Engine.

Best for: Businesses that want to control and manage their IT resources without investing in physical hardware.

Platform as a Service (PaaS):

What it is: PaaS provides a platform that allows developers to build, test, and deploy applications without worrying about managing the underlying hardware or operating system.

How it works: PaaS offers a development framework, databases, and tools needed to create apps. so developers can focus on coding and building apps.

Examples: Google App Engine, Microsoft Azure App Services, Heroku.

Best for: Developers who want to focus on application development without managing servers or infrastructure.

Software as a Service (SaaS):

What it is: SaaS provides software applications over the internet, which users can access directly through their browsers. There's no need to install or maintain the software on your devices.

How it works: SaaS providers host the applications and manage the infrastructure. Users just log in and use the software, paying based on usage.

Examples: Google Workspace (Docs, Sheets), Microsoft 365, Salesforce, Dropbox.

Best for: Businesses or individuals who want to use ready-made software without worrying about installation, updates, or maintenance.

Sadhu Veera Mohan

After get understand these all topic now we can start AWS

Understanding AWS:

Amazon Web Services (AWS) is the world's most comprehensive and widely adopted cloud platform, offering over 200 fully featured services from data centers globally. It is like a resource pool where all kind of services will available like networking,databases etc.AWS helps businesses scale their applications, manage their data, and innovate with powerful, flexible cloud solutions. Let's dive into some of the main AWS services and explain the difference between **serverless computing** and traditional **server-based** solutions.

Before knowing about services we should understand about server less and traditional servers.because in aws both type of service will available.

What is Serverless?

Serverless computing means that you don't have to worry about managing or provisioning servers. With AWS services like Lambda, the cloud provider (AWS) takes care of all the server management, such as scaling, patching, and infrastructure maintenance. Your code runs in response to specific triggers (like a new file upload or an API request).

Key benefits of serverless:

No server management: You don’t need to provision or manage servers.

Scalable: Automatically scales based on demand.

Cost-efficient: You pay only for the compute time you use, with no idle server costs.

Example: Imagine you want to process images are uploaded to your website. Instead of setting up a server to constantly run and process those images, you can use AWS Lambda to automatically run the image processing code only when an image is uploaded. This is both cost-effective and efficient.

What are Traditional Servers?

Traditional servers, like those provisioned through Amazon EC2, are instances where you need to manage the operating system, runtime environment, and applications. You’re responsible for scaling the server, ensuring security, and handling updates. This method gives you full control over your server environment but also requires more management.

Key benefits of traditional servers:

Full control: You manage and configure the server’s resources.

Consistency: You have a stable environment, as you control the entire stack (OS, software, etc.).

Suitable for complex apps: Ideal for apps that require persistent, consistent performance or specialized setups.

Example: If you're running a complex web application that needs consistent performance and a customized environment (e.g., specific software or dependencies), you might choose an EC2 instance to manage those needs.

Real-world applications or apps that use AWS in real-time:

Storage: Dropbox,Google Drive,S3

Software Development:AWS,Azure

Big Data Analytics: Redshift

Content Delivery and St

reaming: Netflix,Amazon Prime and YouTube

Ecommerce: Amazon and eBay

Collabration Tools: Teams,Slack,Zoom

Before diving into the specific AWS services and applications, it's important to understand a few core concepts related to **real-time applications** and **cloud computing** that AWS relies on.

Region:

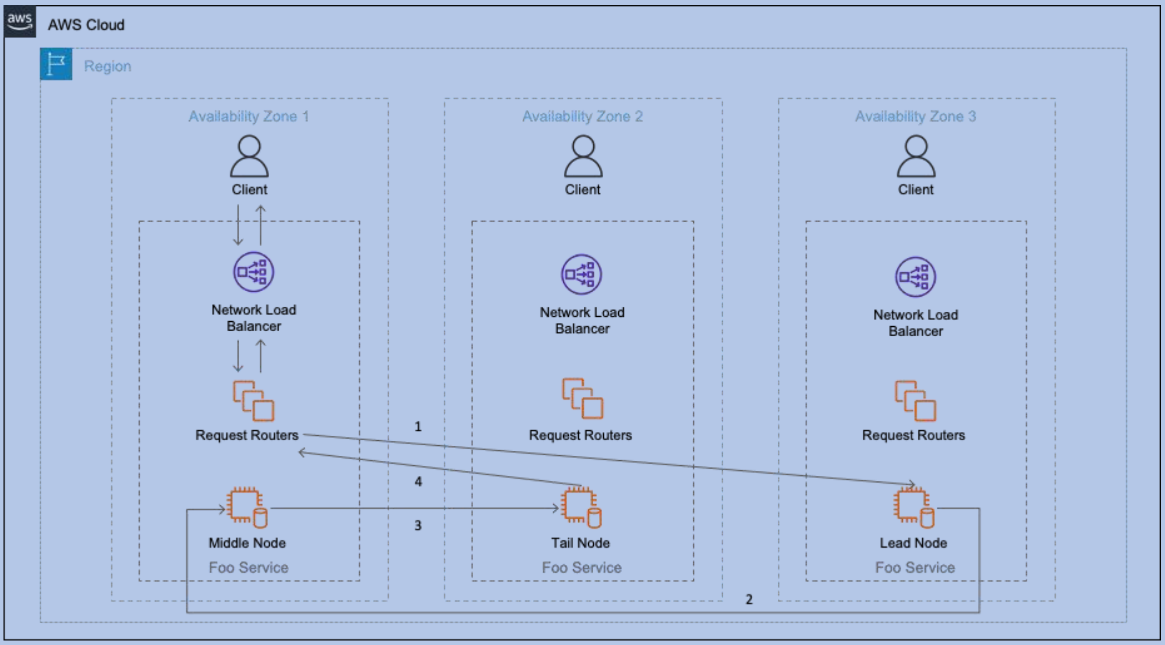
- A **Region** is a geographic area where AWS data centers are located. Each region is designed to be isolated from other regions.
- **Example**: US-East-1 (Northern Virginia), EU-West-1 (Ireland), etc.
- In india there is a mumbai region.

Availability Zones (AZs):

- A **Region** contains multiple **Availability Zones (AZs)**, which are isolated data centers within a region. Each AZ is independent but connected to the others via low-latency links to ensure high availability.
- **Example**: In **US-East-1**, there could be **us-east-1b**
- As of October 2024, AWS has over 100 Availability Zones (AZs) across many Regions worldwide.

Edge Locations:

- **Edge locations** are locations that are part of the **AWS Content Delivery Network (CDN)** (Amazon CloudFront) and are used to deliver content closer to end users.
- It means when ever we have to send any content to from one region to anothor region or one Az to anothor Az the data will stores on near edge locations then deliver the content closer to end users.



Sadhu Veera Mohan

In AWS there are 200+ services we look some main and important services.

To get all about this services practically we should create one AWS account it is free for one year you can practice using that account.

1. Amazon EC2 (Elastic Compute Cloud)

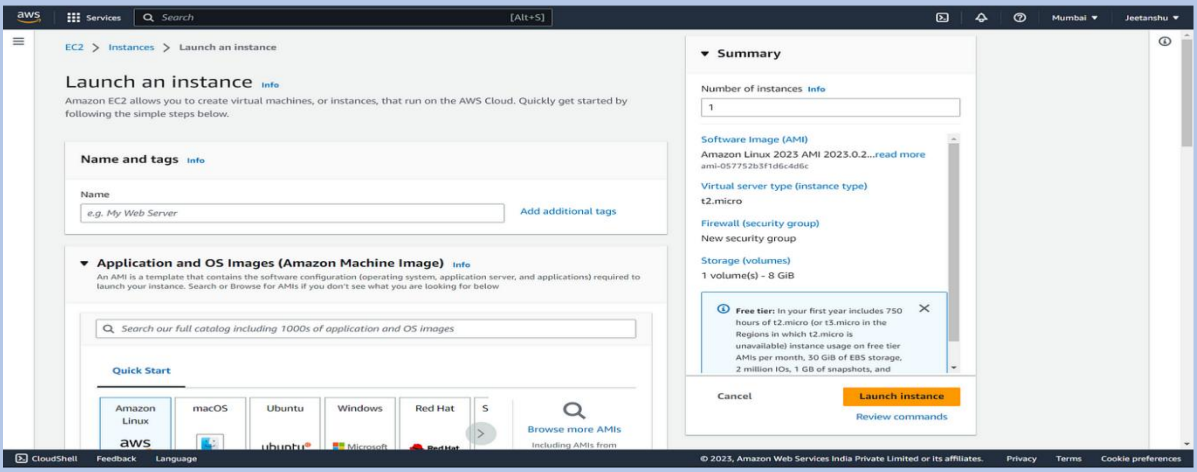
Amazon EC2 is one of the foundational services on AWS, providing scalable computing power in the cloud. You can think of EC2 as renting a server in the cloud to run your applications and services. EC2 allows you to choose the type of instance (virtual machine) based on your needs, such as CPU, memory, storage, and networking capacity.

How it works: With EC2, you manually provision and manage virtual servers (called instances). You can start, stop, or terminate these instances, and you're billed based on how long they're running.

Use case: Ideal for running applications that need constant compute power, such as websites, databases, and enterprise software.

Steps to create EC2:

1. Go to AWS console
2. Services -> compute->EC2
3. Click on launch instance
4. Fill all details like name,AMI,Architecture
5. Instance type:t2.micro(free)
6. For key pair:name,RSA.type and .pem file
7. Networking settings
8. Click on SSH
9. Launch instance



Launch an instance

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.

Name and tags

Name: e.g. My Web Server

Add additional tags

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.0.2...read more ami-057752b3f1d6c4d6c

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and

© 2023, Amazon Web Services India Private Limited or its affiliates.

2. Amazon S3 (Simple Storage Service):

Amazon S3 is a scalable storage service used to store and retrieve any amount of data, such as files, backups, and media. It's highly durable and designed to store data with minimal management. With S3, businesses can save large amounts of data in a secure and cost-effective way.

It stores 5gb of data

The data will stores in buckets.data will called as objects.

How it works: You can upload files into S3 buckets (containers for storage), organize them, and access them via the internet. Data is stored in a way that ensures high availability and redundancy.

Use case: Perfect for storing backups, media files, big data, and static content for websites or applications.

To create an Amazon S3 bucket, you can:

Log in to the AWS Management Console

Search for "S3" in the services search bar

Click S3

Click Create Bucket

Enter a name for your bucket

Choose a region for your bucket

Click Create

You can also use the AWS Command Line Tool to create an S3 bucket.

To upload an object to a bucket

Open the Amazon S3 console.

In the **Buckets** list, choose the name of the bucket that you want to upload your object to.

On the **Objects** tab for your bucket, choose **Upload**.

Under **Files and folders**, choose **Add files**.

Choose a file to upload, and then choose **Open.**

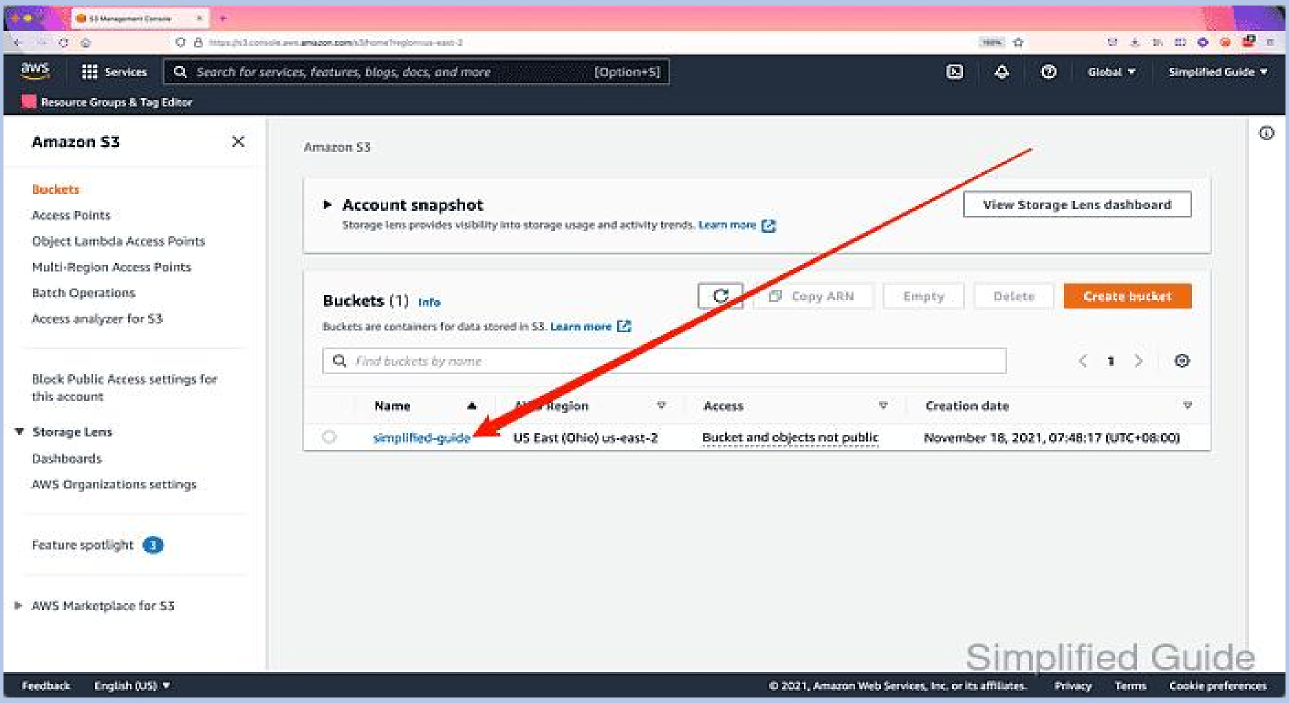
Choose **Upload**.

You've successfully uploaded an object to your bucket.

Sadhu Veera Mohan

Additional configuration:

- You can add additional functionality to your bucket by clicking into its properties.
- You can configure bucket permissions for public access.
- You can add a bucket policy.
- You can choose to keep variants of objects in your bucket.
- You can add tags to your bucket.
- You can configure default encryption.
- You can enable S3 Object Lock.



Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date
simplified-guide	US East (Ohio) us-east-2	Bucket and objects not public	November 18, 2021, 07:48:17 (UTC+08:00)

© 2021, Amazon Web Services, Inc. or its affiliates.

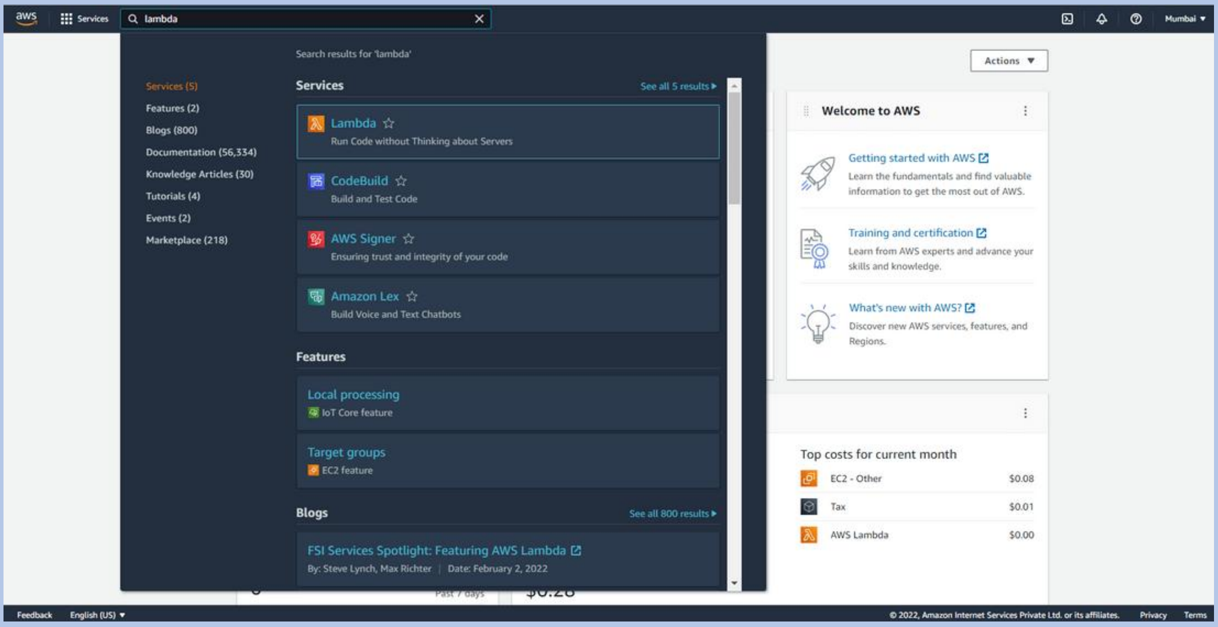
3. AWS Lambda (Serverless):

AWS Lambda is Amazon’s serverless computing service. This means you can run code (functions) without needing to manage any servers. Lambda automatically scales your applications by running code in response to events (such as file uploads or HTTP requests).

How it works: You simply upload your code to Lambda and define when it should run. AWS handles the infrastructure, scaling, and execution. You’re charged only for the compute time your code uses.

Use case: Lambda is great for handling short bursts of computing needs like image resizing, data processing, or running API backends without maintaining servers.

Example: Automatically processing images when they're uploaded to S3, or running backend functions for APIs.



Search results for 'lambda'

Services — See all 5 results ▸

- Lambda ☆ — Run Code without Thinking about Servers
- CodeBuild ☆ — Build and Test Code
- AWS Signer ☆ — Ensuring trust and integrity of your code
- Amazon Lex ☆ — Build Voice and Text Chatbots

Features

- Local processing — IoT Core feature
- Target groups — EC2 feature

Blogs — See all 800 results ▸

- FSI Services Spotlight: Featuring AWS Lambda — By: Steve Lynch, Max Richter | Date: February 2, 2022

Welcome to AWS

- Getting started with AWS — Learn the fundamentals and find valuable information to get the most out of AWS.
- Training and certification — Learn from AWS experts and advance your skills and knowledge.
- What's new with AWS? — Discover new AWS services, features, and Regions.

Top costs for current month

EC2 - Other	$0.08
Tax	$0.01
AWS Lambda	$0.00

© 2022, Amazon Internet Services Private Ltd. or its affiliates.

Sadhu Veera Mohan

AWS IAM (Identity and Access Management)

What it is: IAM is a service that helps you manage users and permissions for your AWS resources.

- **Use Case:** If you're managing a team, IAM helps you assign specific permissions to each user or group to ensure they can access only the resources they need.

How IAM Works

Users: These are individual people or services who need access to your AWS resources. Each user gets a unique set of credentials (username and password, access keys) to authenticate.

Groups: You can organize users into groups. Instead of assigning permissions to each user one by one, you assign permissions to a group (e.g., admins, developers), and everyone in that group inherits those permissions.

Roles: Roles are like temporary access permissions that can be assumed by a user or an AWS service. For example, a Lambda function can assume a role to access an S3 bucket without needing a username or password.

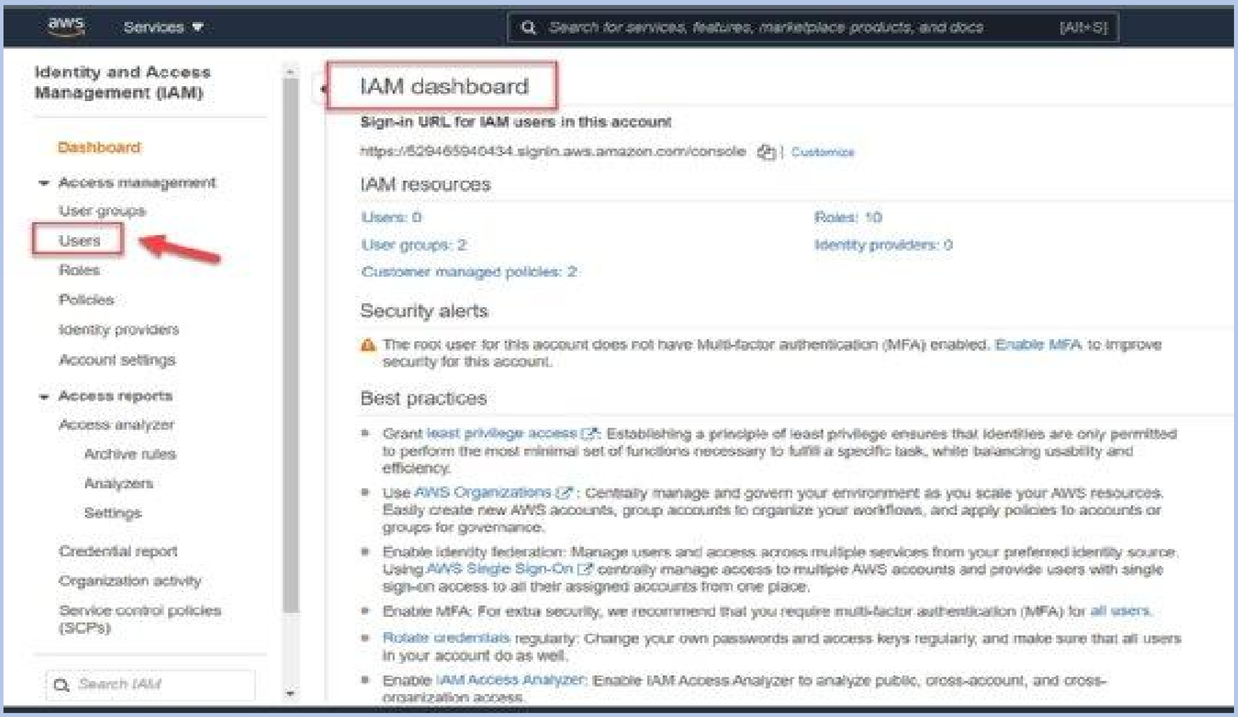
Policies: Policies define what actions a user, group, or role can perform on resources. They are written in JSON and specify permissions like "allow access to S3" or "deny access to EC2."

Example of Using IAM

We have an AWS account for your company. You can create IAM users for different employees, such as developers and administrators.

- **Admins** might have full access to all resources.
- **Developers** could have permission to use EC2 and S3 but not delete resources or access sensitive data.
- **Read-only Users** might only be allowed to view reports in CloudWatch but not make any changes.

With IAM, you can ensure that only the right people can access the right resources, helping to prevent accidental or malicious changes.



Sign-in URL for IAM users in this account

https://529465940434.signin.aws.amazon.com/console | Customize

IAM resources

Users: 0	Roles: 10
User groups: 2	Identity providers: 0
Customer managed policies: 2	

Security alerts

⚠ The root user for this account does not have Multi-factor authentication (MFA) enabled. Enable MFA to improve security for this account.

Best practices

- Grant least privilege access: Establishing a principle of least privilege ensures that identities are only permitted to perform the most minimal set of functions necessary to fulfill a specific task, while balancing usability and efficiency.
- Use AWS Organizations : Centrally manage and govern your environment as you scale your AWS resources. Easily create new AWS accounts, group accounts to organize your workflows, and apply policies to accounts or groups for governance.
- Enable identity federation: Manage users and access across multiple services from your preferred identity source. Using AWS Single Sign-On centrally manage access to multiple AWS accounts and provide users with single sign-on access to all their assigned accounts from one place.
- Enable MFA: For extra security, we recommend that you require multi-factor authentication (MFA) for all users.
- Rotate credentials regularly: Change your own passwords and access keys regularly, and make sure that all users in your account do as well.
- Enable IAM Access Analyzer: Enable IAM Access Analyzer to analyze public, cross-account, and cross-organization access.

Amazon VPC (Virtual Private Cloud):

What it is: VPC allows you to create a private network in the AWS cloud, where you can control IP addresses, subnets, and routing.

- **Use Case:** Think of it as creating a private data center in the cloud, with full control over how resources communicate with each other and the internet.

Why VPC?

When you launch resources on AWS (like EC2 instances, databases, or storage), they need a network to operate in. VPC lets you define your own network boundaries, ensuring that your resources are secure, private, and can communicate with each other in an organized way.

Key Features of Amazon VPC

Customizable Network: You can define your VPC with specific IP address ranges, subnets, and routing rules to fit your needs. It's like setting up the layout of your private network on AWS.

Subnets: A subnet is a smaller segment of a network. In VPC, you can divide your network into subnets (like sections of a building). You can create:

1. **Public Subnets**: Where resources like web servers are accessible from the internet.
2. **Private Subnets**: Where more sensitive resources (like databases) are kept away from direct internet access.

Security: VPC provides multiple layers of security, including:

1. **Security Groups**: Acts like a firewall for EC2 instances to control what traffic is allowed in or out.
2. **Network ACLs**: Additional layers of security for your subnets.
3. **VPN Connections**: Securely connect your on-premise network to your VPC.

Internet Gateway: If you want resources in your VPC to communicate with the internet (like a public web server), you can attach an Internet Gateway, which enables bi-directional traffic between your VPC and the internet.

Private Connectivity: For secure internal communication between resources, you can use VPC Peering or AWS Direct Connect, ensuring your data stays within the AWS network and is not exposed to the internet.

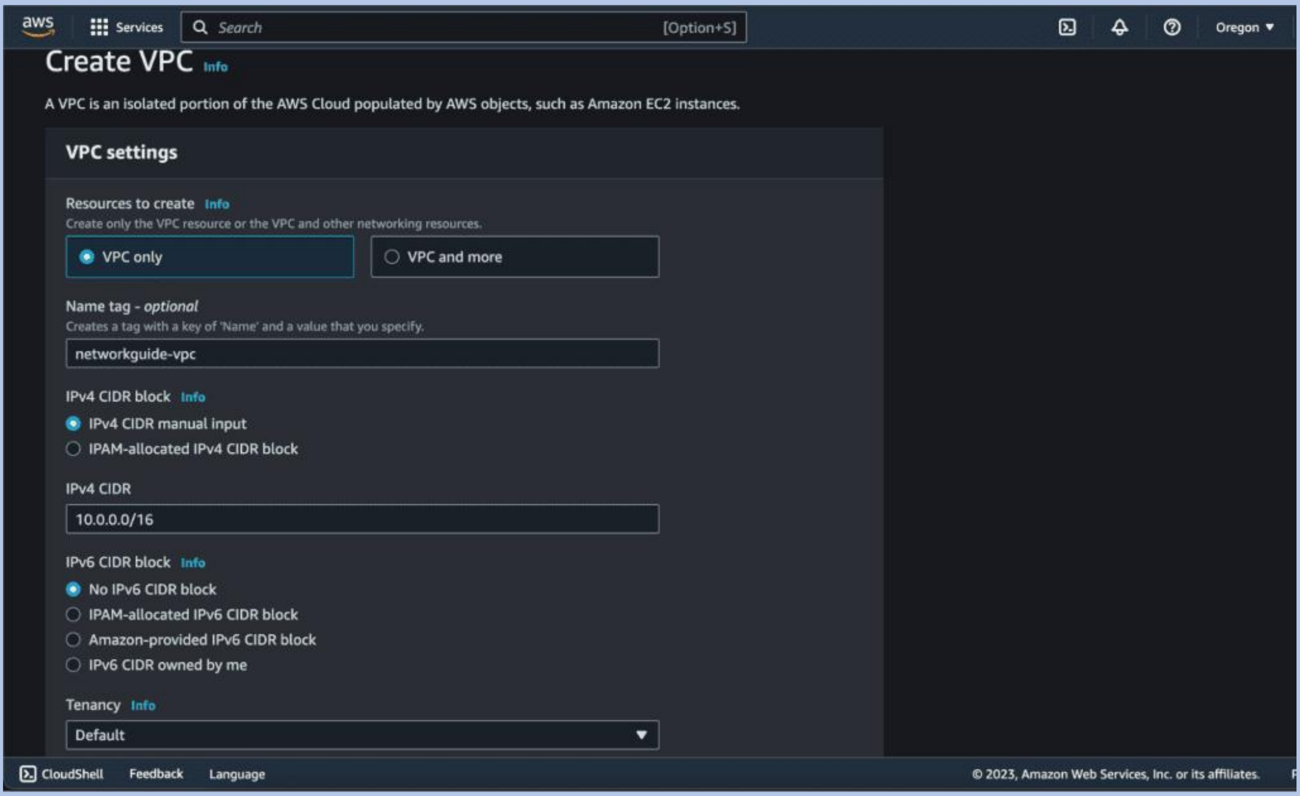
How VPC Works

Let's say you're setting up a web application. You would use a VPC to create a private network for your app's resources. Here's how it might work:

- **Public Subnet:** Your web server (EC2 instance) goes in the public subnet, accessible from the internet.
- **Private Subnet:** Your database (RDS instance) goes in the private subnet, isolated from the internet for security.
- **Security Groups:** You create rules that only allow traffic to your web server from the internet, and only allow your web server to access the database.

Benefits of Using Amazon VPC

- **Isolation and Security:** You can create isolated environments for different parts of your application (public vs private), ensuring sensitive data and resources are protected.
- **Flexibility:** With VPC, you have full control over your network configuration, IP addressing, routing, and security settings.
- **Scalability:** VPCs are highly scalable. You can adjust network configurations as your needs grow, adding more resources or changing settings without disrupting your application.
- **Hybrid Cloud:** If you have an existing on-premise data center, VPC can be connected securely to extend your infrastructure into the AWS cloud.



Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

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 - *optional*

Creates a tag with a key of 'Name' and a value that you specify.

networkguide-vpc

IPv4 CIDR block Info

● IPv4 CIDR manual input

○ IPAM-allocated IPv4 CIDR block

IPv4 CIDR

10.0.0.0/16

IPv6 CIDR block Info

● No IPv6 CIDR block

○ IPAM-allocated IPv6 CIDR block

○ Amazon-provided IPv6 CIDR block

○ IPv6 CIDR owned by me

Tenancy Info

Default ▼

© 2023, Amazon Web Services, Inc. or its affiliates.

Sadhu Veera Mohan

3. Amazon RDS (Relational Database Service)

What it is: RDS is a managed database service that helps you set up, operate, and scale relational databases such as MySQL, PostgreSQL, and SQL Server.

- **Use Case:** You don't have to worry about database maintenance tasks (backups, patching, scaling). AWS manages it for you, allowing you to focus on building your application.

Why RDS?

Managing a database can be complex and time-consuming. With RDS, Amazon takes care of the heavy lifting for you—things like hardware provisioning, database setup, patching, backups, and scaling. You can focus on developing your application without worrying about database management.

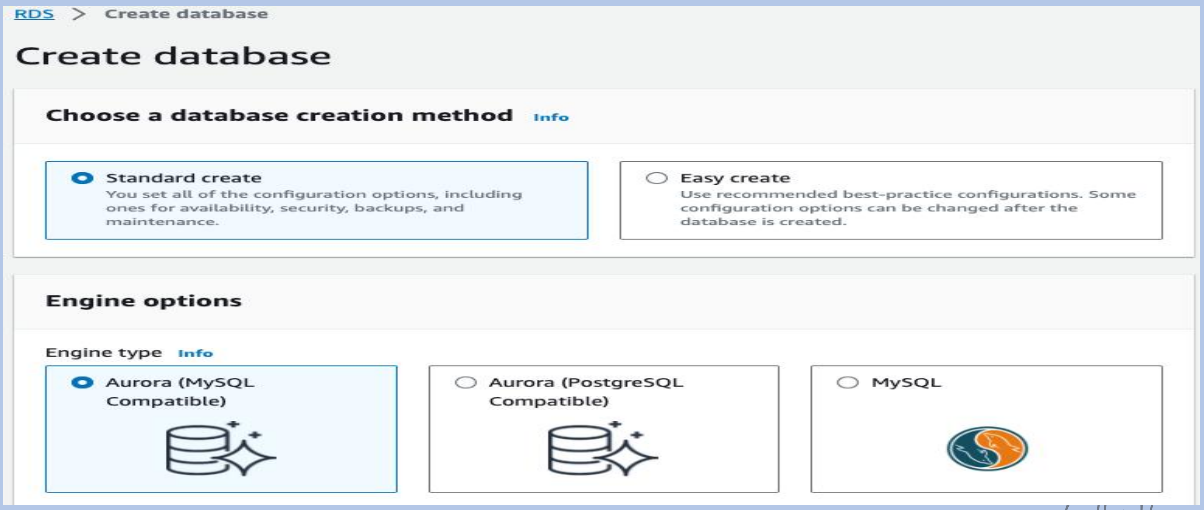
Key Features of Amazon RDS

Automated Backups: RDS automatically backs up your database and allows you to restore it to any point within the backup retention period. This ensures that your data is safe and recoverable.

Scalability: RDS can scale your database's compute and storage resources based on your needs. If you need more power to handle traffic spikes, you can scale up easily.

High Availability: RDS offers options like Multi-AZ deployments, which replicate your database across multiple availability zones (data centers). This ensures high availability and automatic failover if something goes wrong.

Security: You can use IAM (Identity and Access Management) to control who can access your database, and you can encrypt your data at rest and in transit for added security.



RDS > Create database

Create database

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

● Aurora (MySQL Compatible)

○ Aurora (PostgreSQL Compatible)

○ MySQL

Amazon DynamoDB:

What it is: DynamoDB is a fully managed NoSQL database that provides fast and predictable performance at any scale.

- **Use Case:** If you have a large-scale application, such as a mobile app with millions of users, DynamoDB can handle high traffic and automatically scale to accommodate the load.

What is NoSQL?

Unlike traditional relational databases (like MySQL or PostgreSQL), which store data in tables with rows and columns, NoSQL databases like DynamoDB are more flexible. They can handle large amounts of unstructured or semi-structured data and scale more easily to meet high demands.

Why Use DynamoDB?

High Performance: DynamoDB is optimized for fast read and write operations, even at large scale. It provides single-digit millisecond response times, making it ideal for applications that require low latency.

Scalable: As your application grows, DynamoDB automatically scales to handle more traffic without you needing to manage or provision more servers. Whether you're handling a few requests or millions per second, DynamoDB adjusts seamlessly.

Fully Managed: AWS takes care of all the heavy lifting for you—provisioning hardware, database setup, backups, patching, and scaling. This means you don't need to worry about managing infrastructure.

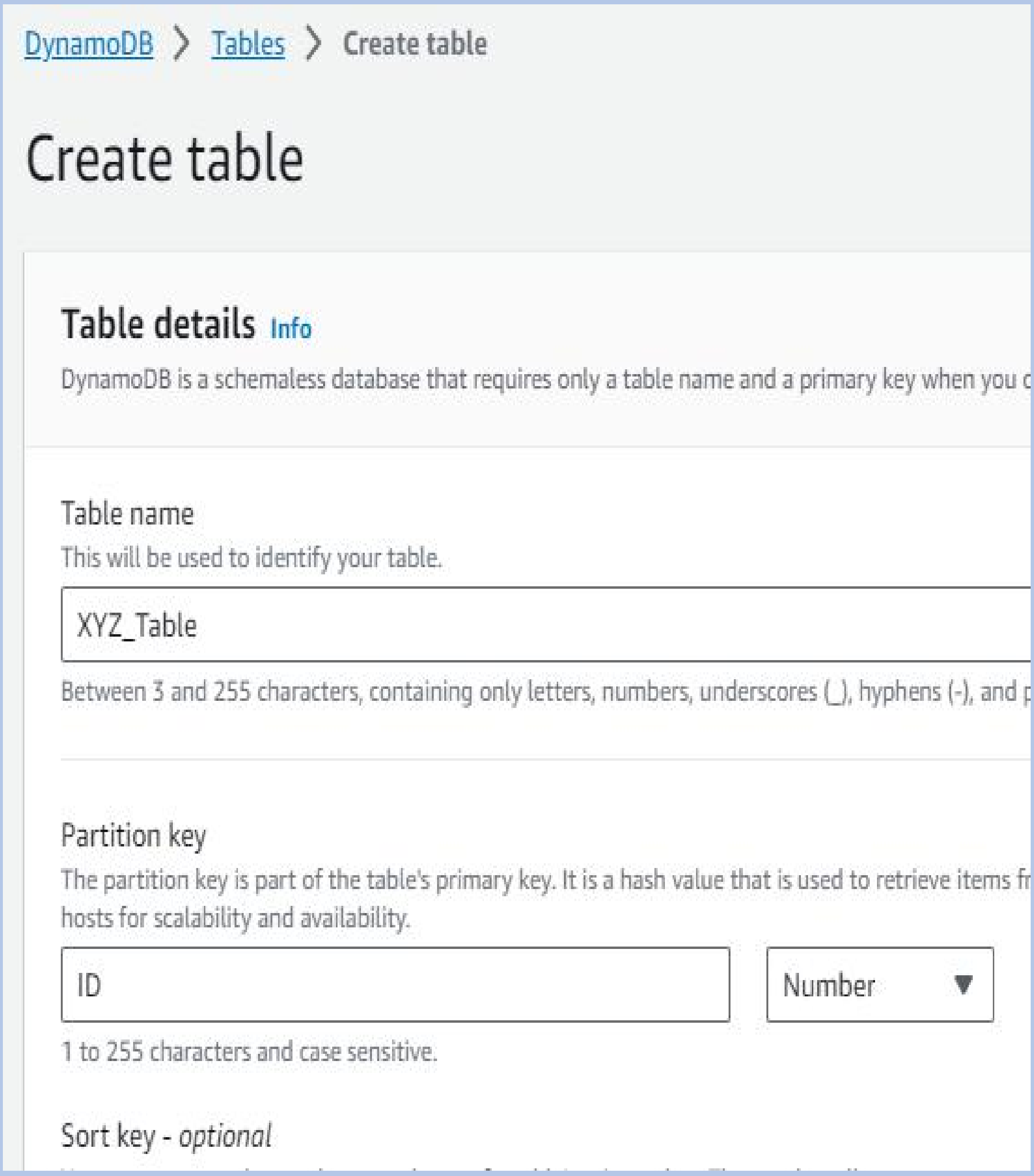
Serverless: DynamoDB is a serverless service, meaning you don't need to worry about managing servers. You simply define your table and its read/write capacity, and DynamoDB handles the rest.

How DynamoDB Works

Let's say you're building a simple mobile app that tracks user activity (e.g., what movies they watch). You can store user data (name, email, movies watched) in a DynamoDB table. Here's how it might look:

- **Primary Key**: The user's email address might be the partition key (unique identifier).
- **Attributes**: Each user's data (like watched movies, preferences, etc.) can be stored as attributes in the table.

- **Global Secondary Index**: If you want to search users by the movie they watched, you can set up a GSI to search by movie title efficiently.



Create table

Table details

DynamoDB is a schemaless database that requires only a table name and a primary key when you c

Table name

This will be used to identify your table.

XYZ_Table

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and p

Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items fr hosts for scalability and availability.

ID

Number ▼

1 to 255 characters and case sensitive.

Sort key - *optional*

AWS CloudWatch:

What it is: CloudWatch is a monitoring and observability service that gives you data and insights about your AWS resources and applications.

- **Use Case:** You can track metrics like CPU usage or disk space, set alarms for specific thresholds, and visualize performance trends to maintain the health of your systems.

Why CloudWatch?

CloudWatch is like a dashboard that shows you everything happening in your AWS environment. If you're running applications, services, or virtual machines (like EC2 instances) in the cloud, CloudWatch helps you monitor them and ensure they are performing as expected. If something goes wrong, you can spot issues early and take action.

Key Features of Amazon CloudWatch

Metrics: CloudWatch collects metrics from AWS resources such as EC2 instances, S3 buckets, RDS databases, and Lambda functions. These metrics include things like CPU usage, memory consumption, network traffic, and more. This helps you understand how your resources are performing.

Alarms: CloudWatch lets you set alarms that notify you when specific thresholds are breached. For example, you can set an alarm to alert you when your EC2 instance’s CPU usage goes above 80% for an extended period. This lets you react quickly to performance issues or resource bottlenecks.

Logs: CloudWatch can collect and store logs from your applications and AWS services. These logs provide detailed information about what’s happening inside your systems, helping you troubleshoot issues. For example, if your web app is crashing, CloudWatch logs might show you the error messages and what went wrong.

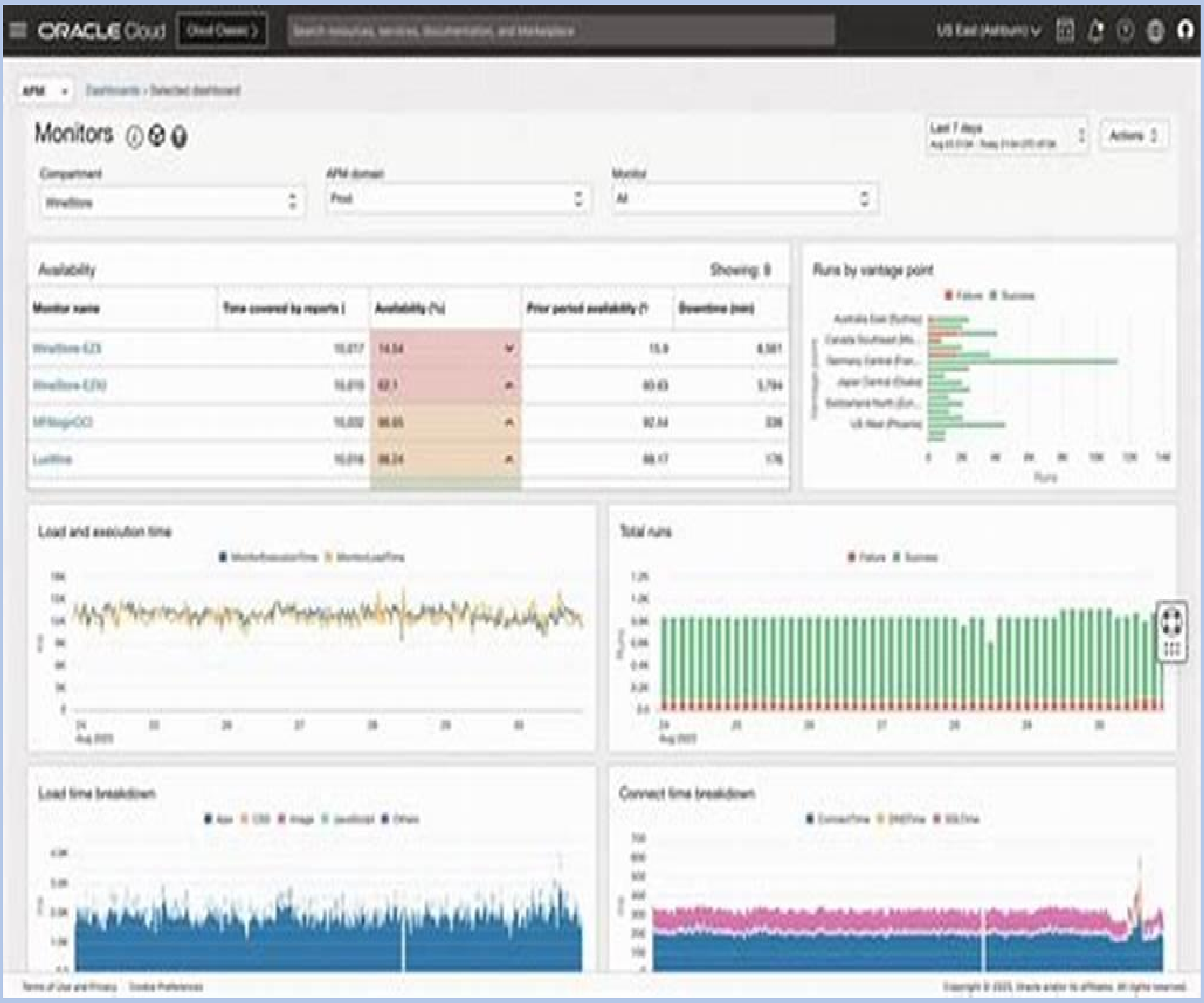
Events: CloudWatch Events automatically reacts to changes in your AWS environment. For example, if an EC2 instance is terminated or a new Lambda function is invoked, CloudWatch can trigger an event and perform actions such as sending notifications or running automation scripts.

Dashboards: CloudWatch provides customizable dashboards to visualize your metrics and logs in one place. This allows you to track key performance indicators (KPIs) and gain insights into the health of your applications and infrastructure.

How CloudWatch Works

Let’s say you have a web application running on EC2 instances. Here’s how CloudWatch can help:

- **Metrics:** CloudWatch tracks metrics such as CPU usage, disk I/O, and network traffic for your EC2 instances. You can monitor these to ensure your servers are not overloaded.
- **Alarms:** You can set alarms to notify you if any EC2 instance’s CPU usage exceeds a certain limit for too long, signaling potential issues.
- **Logs:** CloudWatch can collect logs from your application, such as errors or user activity, making it easier to debug and optimize performance.



Compartment: WineStore

APM domain: Prod

Monitor: All

Availability — Showing: 8

Monitor name	Time covered by reports (	Availability (%)	Prior period availability (%	Downtime (min)
WineStore-EZ3	10,017	14.54	15.9	8,581
WineStore-EZ6	10,019	82.1	83.63	5,794
MFMagicOCI	10,022	98.05	97.84	339
LastFine	10,016	98.24	98.17	176

Sadhu Veera Mohan

These all are services in aws.

Compute Services

- **Amazon EC2 (Elastic Compute Cloud)**: Scalable virtual servers in the cloud.
- **AWS Lambda**: Serverless compute service that runs code in response to events.
- **Amazon Elastic Beanstalk**: Platform as a Service (PaaS) for deploying web applications.
- **Amazon Lightsail**: Simplified compute service for small-scale projects.
- **AWS Batch**: Managed service for running batch processing jobs.
- **AWS Elastic Load Balancing (ELB)**: Automatically distributes incoming application traffic across multiple targets.

Storage Services

- **Amazon S3 (Simple Storage Service)**: Scalable object storage for storing files and data.
- **Amazon EBS (Elastic Block Store)**: Block storage for EC2 instances.
- **Amazon Glacier**: Low-cost, long-term archival storage.
- **Amazon FSx**: Managed file storage for Windows and Lustre file systems.
- **AWS Storage Gateway**: Hybrid cloud storage for integrating on-premise systems with the cloud.

Database Services

- **Amazon RDS (Relational Database Service)**: Managed relational databases (e.g., MySQL, PostgreSQL).
- **Amazon DynamoDB**: Fully managed NoSQL database for fast and scalable performance.
- **Amazon Redshift**: Managed data warehouse for analytics.
- **Amazon Aurora**: High-performance relational database compatible with MySQL and PostgreSQL.
- **Amazon ElastiCache**: Managed in-memory cache for improving performance.
- **Amazon Neptune**: Managed graph database service.

Networking Services

- **Amazon VPC (Virtual Private Cloud)**: Isolated network environment to launch AWS resources.
- **Amazon Route 53**: Scalable DNS (Domain Name System) service.
- **AWS Direct Connect**: Dedicated network connection from your premises to AWS.
- **AWS Global Accelerator**: Improves application performance and availability with global routing.
- **Amazon API Gateway**: Managed service to create and publish APIs.

Security, Identity, and Compliance

- **AWS IAM (Identity and Access Management)**: Manages users, roles, and permissions.
- **Amazon Cognito**: User authentication and identity management.
- **AWS Shield**: Managed DDoS protection.
- **AWS WAF (Web Application Firewall)**: Protects applications from common web exploits.
- **AWS KMS (Key Management Service)**: Managed encryption keys for data protection.
- **AWS Secrets Manager**: Securely manages secrets and credentials.

Analytics Services

- **Amazon Athena**: Query data stored in S3 using SQL.
- **Amazon EMR (Elastic MapReduce)**: Managed big data processing using Hadoop and Spark.
- **Amazon Kinesis**: Real-time data streaming and analytics.
- **AWS Glue**: Managed ETL service for data integration and transformation.
- **Amazon QuickSight**: Business intelligence and data visualization service.
- **Amazon CloudSearch**: Managed search service for websites and applications.

Machine Learning Services

- **Amazon SageMaker**: End-to-end machine learning platform for building, training, and deploying models.
- **AWS Rekognition**: Image and video analysis with deep learning.
- **Amazon Polly**: Text-to-speech service.
- **Amazon Lex**: Build conversational interfaces like chatbots.
- **Amazon Comprehend**: Natural language processing for analyzing text.
- **AWS Deep Learning AMIs**: Preconfigured Amazon Machine Images for deep learning.

Developer Tools

- **AWS CodeCommit**: Managed Git repositories for source code.
- **AWS CodeBuild**: Managed build service for compiling source code.
- **AWS CodeDeploy**: Automates deployment of applications.
- **AWS CodePipeline**: Continuous integration and delivery (CI/CD) service.
- **AWS Cloud9**: Cloud-based integrated development environment (IDE).

Content Delivery and CDN

- **Amazon CloudFront**: Content delivery network (CDN) for fast content distribution.
- **AWS Snowball**: Physical data transfer device for large amounts of data.
- **AWS Global Accelerator**: Optimizes the path between your users and your AWS resources.

Migration & Transfer Services

- **AWS Migration Hub**: Centralized tracking for migrations to AWS.
- **AWS Database Migration Service (DMS)**: Migrate databases to AWS.
- **AWS Server Migration Service**: Automate server migrations to AWS.
- **AWS Snowcone**: Small, portable edge computing device for data collection and processing.
- **AWS Transfer Family**: Managed services for transferring files over SFTP, FTPS, and FTP.

IoT (Internet of Things)

- **AWS IoT Core**: Managed service to connect IoT devices to the cloud.
- **AWS IoT Greengrass**: Extends AWS services to local devices for edge computing.
- **AWS IoT Analytics**: Analytics service for IoT data.
- **AWS IoT Device Management**: Manages IoT devices at scale.

Management & Governance

- **AWS CloudWatch**: Monitoring and observability for AWS resources.
- **AWS CloudTrail**: Logs API calls to monitor activity in your AWS account.
- **AWS Config**: Tracks configuration changes and compliance of AWS resources.
- **AWS Systems Manager**: Centralized management for AWS resources.
- **AWS Organizations**: Centralized management for multiple AWS accounts.

Media Services

- **AWS MediaConvert**: File-based video transcoding.
- **AWS MediaLive**: Live video streaming processing.
- **AWS MediaPackage**: Video packaging and delivery service.

Game Development Services

- **Amazon GameLift**: Managed service for game server hosting.
- **AWS Lumberyard**: Game engine for building high-quality games.

Business Applications

- **Amazon WorkSpaces**: Managed desktop-as-a-service (DaaS) solution.
- **Amazon Chime**: Communication service for video meetings, chat, and voice calls.
- **Amazon WorkDocs**: Managed file storage and collaboration.
- **Amazon WorkMail**: Secure and managed business email service.

Thank You

Sadhu Veera Mohan

Sadhu Veera Mohan