BLACKBUCKS INTERNSHIP REPORT

THREE TIER ARCHITECTURE

SUBMITTED BY

20B91A0544 - BURADA DINESH KUMAR
20B91A0545 - CHAKKA DEEPAK SAI HEMANTH
20B91A0550 - CHEEMALADINNE SUDHARSHAN

UNDER THE GUIDANCE OF MR. AASHU DEV

Blackbuck Engineers Pvt.Ltd
Road No 36, Jubilee Hills, Hyderabad

BLACKBUCKS INTERNSHIP WORK

Title:

AWS architecture for THREE TIER ARCHITECTURE

Abstract:

The 3-tier AWS architecture is a widely used approach for building scalable and reliable applications in the cloud. This architecture consists of three distinct layers: the presentation layer, the application layer, and the data layer. Each layer serves a specific purpose and can be independently scaled and managed, resulting in improved performance, availability, and fault tolerance.

In the presentation layer, the user interface or client-side application resides. This layer is responsible for rendering the application's interface and handling user interactions. It typically utilizes web technologies such as HTML, CSS, and JavaScript. AWS services like Amazon CloudFront and Amazon S3 can be used to host and distribute static assets, ensuring fast and reliable content delivery to users across the globe.

The application layer, also known as the business logic layer, contains the core functionalities and processing logic of the application. It handles user requests, executes business operations, and communicates with the data layer. AWS provides various services to support this layer, such as Amazon EC2 for virtual servers, AWS Lambda for serverless computing, and Amazon API Gateway for creating APIs and managing requests.

The data layer is responsible for storing and managing the application's data. It can consist of various AWS services, including Amazon RDS for relational databases, Amazon DynamoDB for NoSQL databases, Amazon Elastic Cache for in-memory caching, and Amazon S3 for object storage. This layer ensures data persistence, scalability, and high availability.

By adopting the 3-tier AWS architecture, organizations can benefit from the flexibility and scalability of cloud computing. The modular design allows each layer to be independently scaled and updated without affecting the other layers, enabling seamless expansion and efficient resource utilization. Moreover, AWS offers a wide range of services and tools that simplify the deployment, management, and monitoring of applications, further enhancing the overall performance and reliability of the system.

AWS PROJECT

DOCUMENTATION

TEAM

20B91A0544 – B DINESH KUMAR

20B91A0545 – CH DEEPAK

20B91A0550 – CH SUDHARSHAN

INTRODUCTION

CLOUD COMPUTING

Cloud computing is the delivery of computing services—including servers, storage, databases, networking, software, analytics, and intelligence—over the Internet ("the cloud") to offer faster innovation, flexible resources, and economies of scale. You typically pay only for cloud services you use, helping you lower your operating costs, run your infrastructure more efficiently, and scale as your business needs change.

Benefits of Cloud Computing

Cloud computing is a big shift from the traditional way businesses think about IT resources. Here are seven common reasons organizations are turning to cloud computing services:

Cost

- Moving to the cloud helps companies optimize IT costs. This is because cloud computing eliminates the capital expense of buying hardware and software and setting up and running onsite data centres—the racks of servers, the round-the-clock electricity for power and cooling, and the IT experts for managing the infrastructure. It adds up fast.

Speed

- Most cloud computing services are provided self service and on demand, so even vast amounts of computing resources can be provisioned in minutes, typically with just a few mouse clicks, giving businesses a lot of flexibility and taking the pressure off capacity planning.

Performance

- The biggest cloud computing services run on a worldwide network of secure datacentres, which are regularly upgraded to the latest generation of fast and efficient computing hardware. This offers several benefits over a single corporate datacentre, including reduced network latency for applications and greater economies of scale.

Reliability

- Cloud computing makes data backup, disaster recovery, and business continuity easier and less expensive because data can be mirrored at multiple redundant sites on the cloud provider's network.

Security

- Many cloud providers offer a broad set of policies, technologies, and controls that strengthen your security posture overall, helping protect your data, apps, and infrastructure from potential threats.

CLOUD COMPUTING SERVICES

- IaaS (Infrastructure-as-a-Service)
- PaaS (Platform-as-a-Service)
- SaaS (Software-as-a-service)

IaaS (Infrastructure-as-a-Service)

IaaS provides on-demand access to fundamental computing resources—physical and virtual servers, networking, and storage—over the internet on a pay-as-you-go basis. IaaS enables end users to scale and shrink resources on an as-needed basis, reducing the need for high, up-front capital expenditures or unnecessary on-premises or 'owned' infrastructure and for overbuying resources to accommodate periodic spikes in usage.

In contrast to SaaS and PaaS (and even newer PaaS computing models such as containers and serverless), IaaS provides the users with the lowest-level control of computing resources in the cloud.

IaaS was the most popular cloud computing model when it emerged in the early 2010s. While it remains the cloud model for many types of workloads, use of SaaS and PaaS is growing at a much faster rate.

PaaS (Platform-as-a-service)

PaaS provides software developers with on-demand platform—hardware, complete software stack, infrastructure, and even development tools—for running, developing, and managing applications without the cost, complexity, and inflexibility of maintaining that platform on-premises.

With PaaS, the cloud provider hosts everything—servers, networks, storage, operating system software, middleware, databases—at their datacentre. Developers simply pick from a menu to 'spin up' servers and environments they need to run, build, test, deploy, maintain, update, and scale applications.

Today, PaaS is often built around *containers*, a virtualized compute model one step removed from virtual servers. containers virtualize the operating system, enabling developers to package the application with only the operating system services it needs to run on any platform, without modification and without need for middleware.

SaaS (Software-as-a-Service)

SaaS—also known as cloud-based software or cloud applications—is application software that's hosted in the cloud, and that user's access via a web browser, a dedicated desktop client, or an API that integrates with a desktop or mobile operating system.

In most cases, SaaS users pay a monthly or annual subscription fee; some may offer 'pay-as-you-go' pricing based on your actual usage.

In addition to the cost savings, time-to-value, and scalability benefits of cloud, SaaS offers the following:

- **Automatic upgrades:** With SaaS, users take advantage of new features as soon as the provider adds them, without having to orchestrate an on-premises upgrade.
- **Protection from data loss:** Because SaaS stores application data in the cloud with the application, users don't lose data if their device crashes or breaks.

SaaS is the primary delivery model for most commercial software today—there are hundreds of thousands of SaaS solutions available, from the most focused industry and departmental applications to powerful enterprise software database and AI (artificial intelligence) software.

AWS (AMAZON WEB SERVICES)

The AWS service is provided by the Amazon that uses distributed IT infrastructure to provide different IT resources available on demand. It provides different services such as infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS).

Amazon launched AWS, a cloud computing platform to allow the different organizations to take advantage of reliable IT infrastructure.

Advantages of AWS

Flexibility

- We can get more time for core business tasks due to the instant availability of new features and services in AWS.
- It provides effortless hosting of legacy applications. AWS does not require learning new technologies and migration of applications to the AWS provides the advanced computing and efficient storage.
- AWS also offers a choice that whether we want to run the applications and services together or not. We can also choose to run a part of the IT infrastructure in AWS and the remaining part in data centres.

Cost-effectiveness

AWS requires no upfront investment, long-term commitment, and minimum expense when compared to traditional IT infrastructure that requires a huge investment.

Scalability

Through AWS, autoscaling and elastic load balancing techniques are automatically scaled up or down, when demand increases or decreases respectively. AWS techniques are ideal for handling unpredictable or very high loads. Due to this reason, organizations enjoy the benefits of reduced cost and increased user satisfaction.

Security

- AWS has a virtual infrastructure that offers optimum availability while managing full privacy and isolation of their operations.
- AWS provides end-to-end security and privacy to customers.
- Customers can expect high-level of physical security because of Amazon's several years of experience in designing, developing and maintaining large-scale IT operation centers.
- AWS ensures the three aspects of security, i.e., Confidentiality, integrity, and availability of user's data.

AWS Services

- Amazon EC2 (Elastic Compute Cloud)
- Amazon RDS (Relational Database Services)
- Amazon S3 (Simple Storage Service)
- Amazon Lambda
- Amazon Cognito
- Amazon SNS (Simple Notification Service)
- Amazon VPC (Virtual Private Cloud)
- Amazon CloudWatch
- Amazon Cloud9
- Amazon Elastic Beanstalk
- Amazon CodeCommit
- Amazon IAM (Identity and Access Management)
- Amazon Dynamo DB
- AWS Amplify and more…

Amazon EC2

Amazon Elastic Compute Cloud (EC2) is a part of Amazon.com's cloud-computing platform, Amazon Web Services (AWS), that allows users to rent virtual computers on which to run their own computer applications. EC2 encourages scalable deployment of applications by providing a web service through which a user can boot an Amazon Machine Image (AMI) to configure a virtual machine, which Amazon calls an "instance", containing any software desired. A user can create, launch, and terminate server-instances as needed, paying by the second for active servers – hence the term "elastic". EC2 provides users with control over the geographical location of instances that allows for latency optimization and high levels of redundancy. In November 2010, Amazon switched its own retail website platform to EC2 and AWS.

Amazon announced a limited public beta test of EC2 on August 25, 2006, offering access on a first-come, first-served basis. Amazon added two new instance types (Large and Extra-Large) on October 16, 2007. On May 29, 2008, two more types were added, High-CPU Medium and High-CPU Extra Large. There were twelve types of instances available.

Amazon added three new features on March 27, 2008, static IP addresses, availability zones, and user selectable kernels. On August 20, 2008, Amazon added Elastic Block Store (EBS) This provides persistent storage, a feature that had been lacking since the service was introduced.

Amazon RDS

Amazon Relational Database Service (or **Amazon RDS**) is a distributed relational database service by Amazon Web Services (AWS). It is a web service running "in the cloud" designed to simplify the setup, operation, and scaling of a relational database for use in applications. Administration processes like patching the database software, backing up databases and enabling point-in-time recovery are managed automatically. Scaling storage and compute resources can be performed by a single API call to the AWS control plane on-demand. AWS does not offer an SSH connection to the underlying virtual machine as part of the managed service.

Multiple Availability Zone (AZ) Deployment

In May 2010 Amazon announced Multi-Availability Zone deployment support. Amazon RDS Multi-Availability Zone (AZ) allows users to automatically provision and maintain a synchronous physical or logical "standby" replica, depending on database engine, in a different Availability Zone (independent infrastructure in a physically separate location). Multi-AZ database instance can be developed at creation time or modified to run as a multi-AZ deployment later. Multi-AZ deployments aim to provide enhanced availability and data durability for MySQL, MariaDB,

Oracle, PostgreSQL and SQL Server instances and are targeted for production environments. In the event of planned database maintenance or unplanned service disruption, Amazon RDS automatically fails over to the up-to-date standby, allowing database operations to resume without administrative intervention.

Multi-AZ RDS instances are optional and have a cost associated with them. When creating a RDS instance, the user is asked if they would like to use a multi-AZ RDS instance. In Multi-AZ RDS deployments backups are done in the standby instance so I/O activity is not suspended any time, but users may experience elevated latencies for a few minutes during backups.

Read replicas

Read replicas allow different use cases such as scale in for read-heavy database workloads. There are up to five replicas available for MySQL, MariaDB, and PostgreSQL. Instances use the native, asynchronous replication functionality of their respective database engines. They have no backups configured by default and are accessible and can be used for read scaling. MySQL and MariaDB read replicas and can be made writeable again since October 2012; PostgreSQL read replicas do not support it. Replicas are done at database instance level and do not support replication at database or table level.

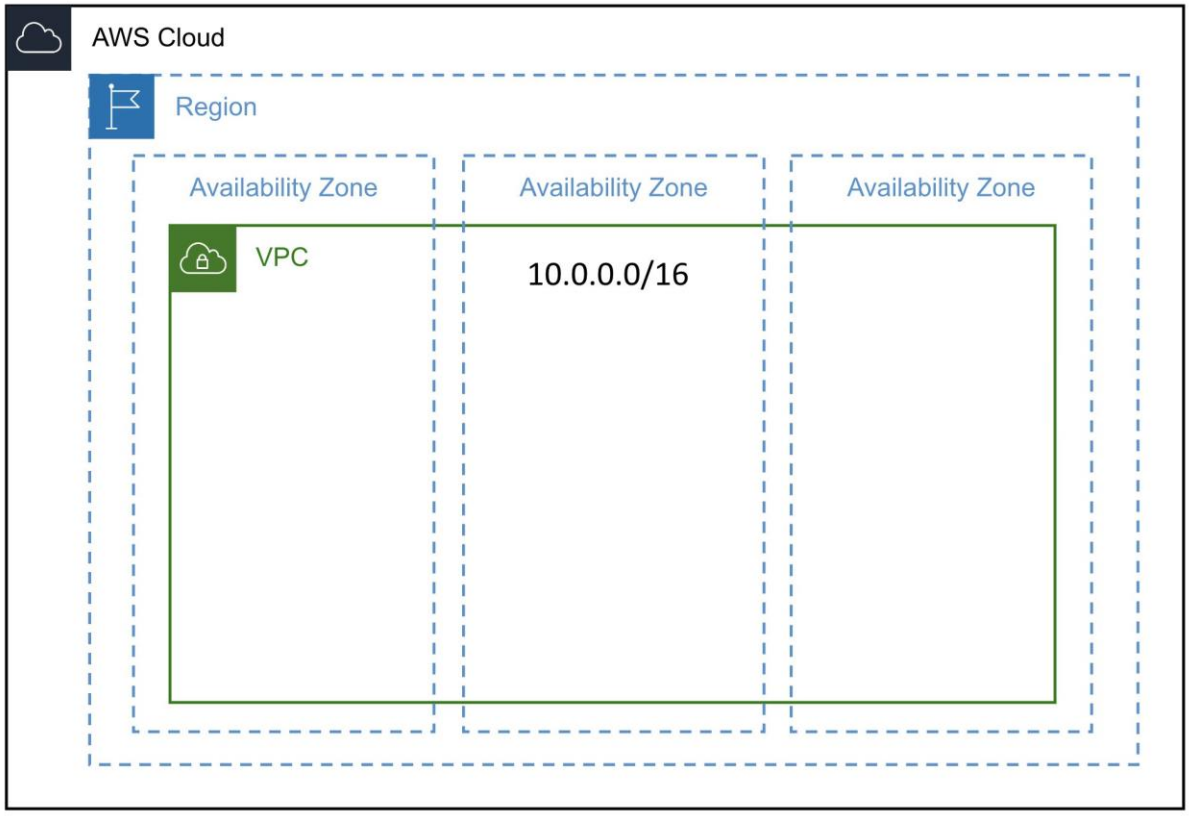
Performance metrics and monitoring

Performance metrics for Amazon RDS are available from the AWS Management Console or the Amazon CloudWatch API. In December 2015, Amazon announced an optional enhanced monitoring feature that provides an expanded set of metrics for the MySQL, MariaDB, and Aurora database engines.

Amazon VPC

Amazon Virtual Private Cloud (VPC) is a commercial cloud computing service that provides a virtual private cloud, by provisioning a logically isolated section of Amazon Web Services (AWS) Cloud. Enterprise customers are able to access the Amazon Elastic Compute Cloud (EC2) over an IPsec based virtual private network. Unlike traditional EC2 instances which are allocated internal and external IP numbers by Amazon, the customer can assign IP numbers of their choosing from one or more subnets.

Amazon Web Services launched Amazon Virtual Private Cloud on 26 August 2009, which allows the Amazon Elastic Compute Cloud service to be connected to legacy infrastructure over an IPsec VPN.In AWS, the basic VPC is free to use, with users being charged by usage for additional features.EC2 and RDS instances running in a VPC can also be purchased using Reserved Instances, however will have a limitation on resources being guaranteed.



IBM Cloud launched IBM Cloud VPC on 4 June 2019, provides an ability to manage virtual machine-based compute, storage, and networking resources. Pricing for IBM Cloud Virtual Private Cloud is applied separately for internet data transfer, virtual server instances, and block storage used within IBM Cloud VPC.

Google Cloud Platform resources can be provisioned, connected, and isolated in a virtual private cloud (VPC) across all GCP regions. With GCP, VPCs are global resources and subnets within that VPC are regional resources. This allows users to connect zones and regions without the use of additional networking complexity as all data travels, encrypted in transit and at rest, on Google's own global, private network. Identity management policies and security rules allow for private access to Google's storage, big data, and analytics managed services. VPCs on Google Cloud Platform leverage the security of Google's data centres.

Amazon S3

Amazon S3 manages data with an object storage architecture which aims to provide scalability, high availability, and low latency with high durability. The basic storage units of Amazon S3 are objects which are organized into buckets. Each object is identified by a unique, user-assigned key. Buckets can be managed using the console provided by Amazon S3, programmatically with the AWS SDK, or the REST application programming interface.

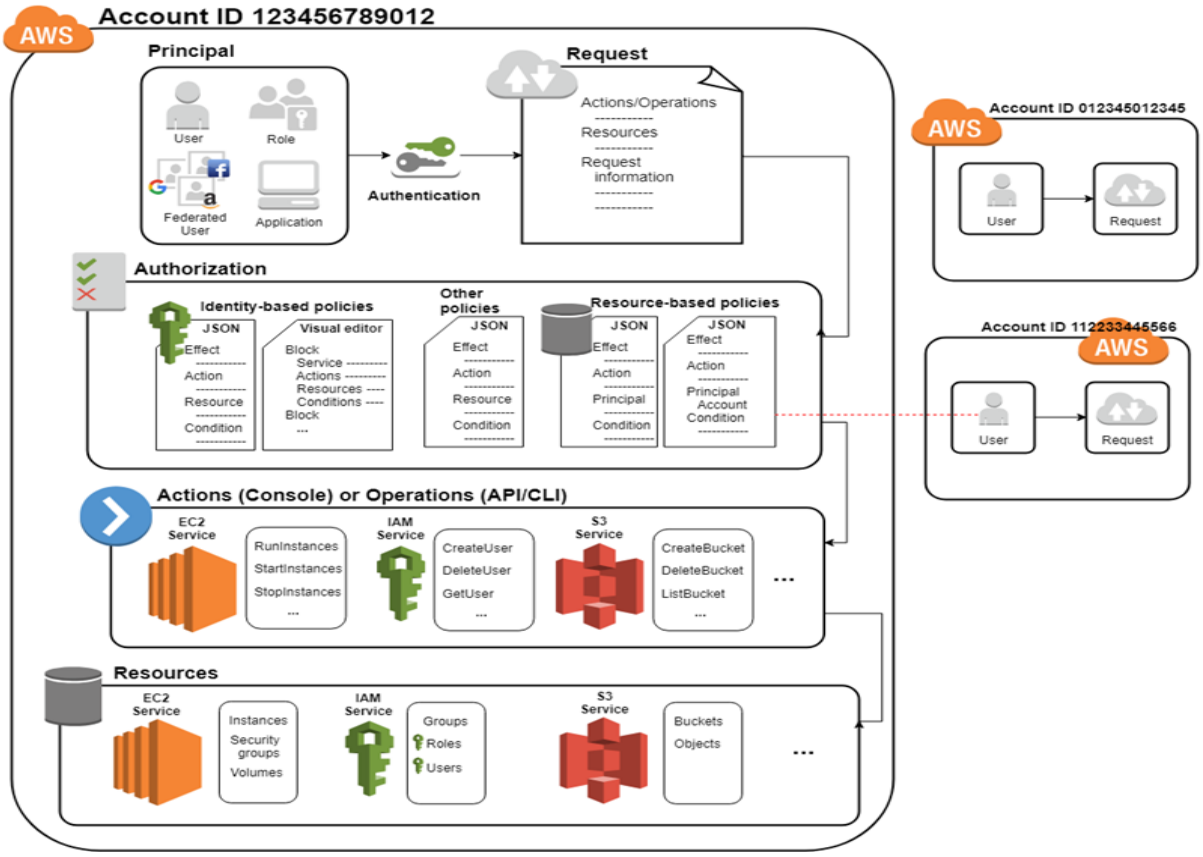
Objects can be up to five terabytes in size. Requests are authorized using an access control list associated with each object bucket and support versioning which is disabled by default. Since buckets are typically the size of an entire file system mount in other systems, this access control scheme is very coarse-grained. In other words, unique access controls cannot be associated with individual files. [citation needed] Amazon S3 can be used to replace static web-hosting infrastructure with HTTP client-accessible objects, index document support and error document support.

The Amazon AWS authentication mechanism allows the creation of authenticated URLs, valid for a specified amount of time. Every item in a bucket can also be served as a BitTorrent feed. The Amazon S3 store can act as a seed host for a torrent and any BitTorrent client can retrieve the file. This can drastically reduce the bandwidth cost for the download of popular objects. A bucket can be configured to save HTTP log information to a sibling bucket; this can be used in data mining operations.

There are various User Mode File System (FUSE)–based file systems for Unix-like operating systems (for example, Linux) that can be used to mount an S3 bucket as a file system. The semantics of the Amazon S3 file system is not that of a POSIX file system, so the file system may not behave entirely as expected.

Amazon IAM

IAM provides the infrastructure necessary to control authentication and authorization for your AWS account.



First, a human user or an application uses their sign-in credentials to authenticate with AWS. Authentication is provided by matching the sign-in credentials to a principal (an IAM user, federated user, IAM role, or application) trusted by the AWS account.

Next, a request is made to grant the principal access to resources. Access is granted in response to an authorization request. For example, when you first sign into the console and are on the console home page, you are not accessing a specific service. When you select a service, the request for authorization is sent to that service and it looks to see if your identity is on the list of authorized users, what policies are being enforced to control the level of access granted, and any other policies that might be in effect. Authorization requests can be made by principals within your AWS account or from another AWS account that you trust.

Once authorized, the principal can take action or perform operations on resources in your AWS account. For example, the principal could launch a new Amazon Elastic Compute Cloud instance, modify IAM group membership, or delete Amazon Simple Storage Service buckets.

The previous illustration we used specific terminology to describe how to obtain access to resources. These IAM terms are commonly used when working with AWS:

IAM Resources

The user, group, role, policy, and identity provider objects that are stored in IAM. As with other AWS services, you can add, edit, and remove resources from IAM.

IAM Identities

The IAM resource objects that are used to identify and group. You can attach a policy to an IAM identity. These include users, groups, and roles.

IAM Entities

The IAM resource objects that AWS uses for authentication. These include IAM users and roles.

Principals

A person or application that uses the AWS account root user, an IAM user, or an IAM role to sign in and make requests to AWS. Principals include federated users and assumed roles.

Human users

Also known as human identities; the people, administrators, developers, operators, and consumers of your applications.

Workload

A collection of resources and code that delivers business value, such as an application or backend process. Can include applications, operational tools, and components.

AWS Lambda

AWS Lambda is a compute service that lets you run code without provisioning or managing servers.

Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging. With Lambda, all you need to do is supply your code in one of the language runtimes that Lambda supports.

You organize your code into Lambda functions. The Lambda service runs your function only when needed and scales automatically. You only pay for the compute time that you consume—there is no charge when your code is not running.

When using Lambda, you are responsible only for your code. Lambda manages the compute fleet that offers a balance of memory, CPU, network, and other resources to run your code. Because Lambda manages these resources, you cannot log in to compute instances or customize the operating system on provided runtimes.

Lambda performs operational and administrative activities on your behalf, including managing capacity, monitoring, and logging your Lambda functions.

If you do need to manage your compute resources, AWS has other compute services to consider, such as:

- AWS App Runner builds and deploys containerized web applications automatically, load balances traffic with encryption, scales to meet your traffic needs, and allows for the configuration of how services are accessed and communicate with other AWS applications in a private Amazon VPC.
- AWS Fargate with Amazon ECS runs containers without having to provision, configure, or scale clusters of virtual machines.
- Amazon EC2 lets you customize operating system, network and security settings, and the entire software stack. You are responsible for provisioning capacity, monitoring fleet health and performance, and using Availability Zones for fault tolerance.

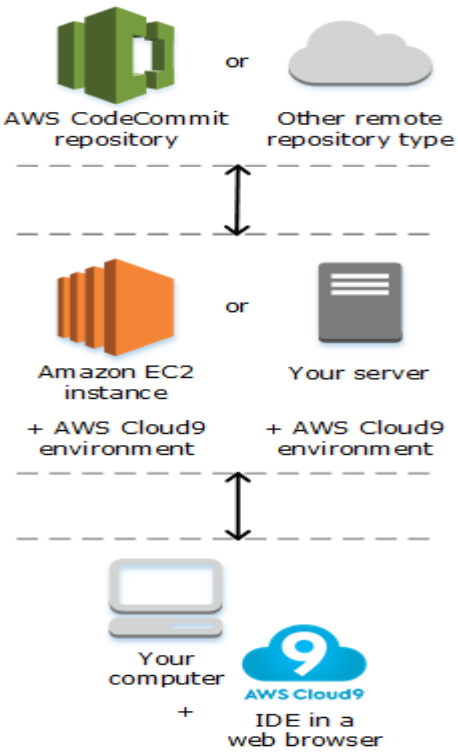
You can use environment variables to adjust your function's behaviour without updating code. An environment variable is a pair of strings that is stored in a function's version-specific configuration. The Lambda runtime makes environment variables available to your code and sets additional environment variables that contain information about the function and invocation request.

AWS Cloud9

AWS Cloud9 is an integrated development environment, or *IDE*.

The AWS Cloud9 IDE offers a rich code-editing experience with support for several programming languages and runtime debuggers, and a built-in terminal. It contains a collection of tools that you use to code, build, run, test, and debug software, and helps you release software to the cloud.

You access the AWS Cloud9 IDE through a web browser. You can configure the IDE to your preferences. You can switch colour themes, bind shortcut keys, enable programming language-specific syntax colouring and code formatting, and more.



AWS Elastic BeanStalk

Amazon Web Services (AWS) comprises over one hundred services, each of which exposes an area of functionality. While the variety of services offers flexibility for how you want to manage your AWS infrastructure, it can be challenging to figure out which services to use and how to provision them.

With Elastic Beanstalk, you can quickly deploy and manage applications in the AWS Cloud without having to learn about the infrastructure that runs those applications. Elastic Beanstalk reduces management complexity without restricting choice or control. You simply upload your application, and Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring.

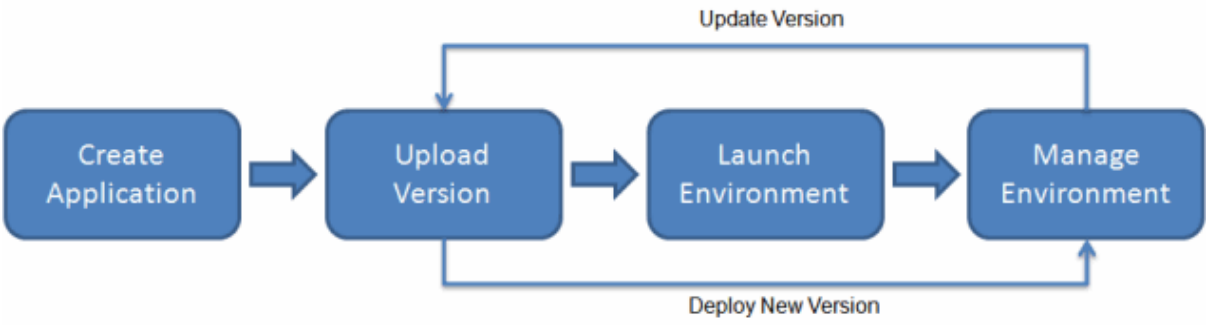
Elastic Beanstalk supports applications developed in Go, Java, .NET, Node.js, PHP, Python, and Ruby. When you deploy your application, Elastic Beanstalk builds the selected supported platform version and provisions one or more AWS resources, such as Amazon EC2 instances, to run your application.

You can interact with Elastic Beanstalk by using the Elastic Beanstalk console, the AWS Command Line Interface (AWS CLI), or **eb**, a high-level CLI designed specifically for Elastic Beanstalk.

To learn more about how to deploy a sample web application using Elastic Beanstalk, see Getting Started with AWS: Deploying a Web App.

You can also perform most deployment tasks, such as changing the size of your fleet of Amazon EC2 instances or monitoring your application, directly from the Elastic Beanstalk web interface (console).

To use Elastic Beanstalk, you create an application, upload an application version in the form of an application source bundle (for example, a Java .war file) to Elastic Beanstalk, and then provide some information about the application. Elastic Beanstalk automatically launches an environment and creates and configures the AWS resources needed to run your code. After your environment is launched, you can then manage your environment and deploy new application versions. The following diagram illustrates the workflow of Elastic Beanstalk.



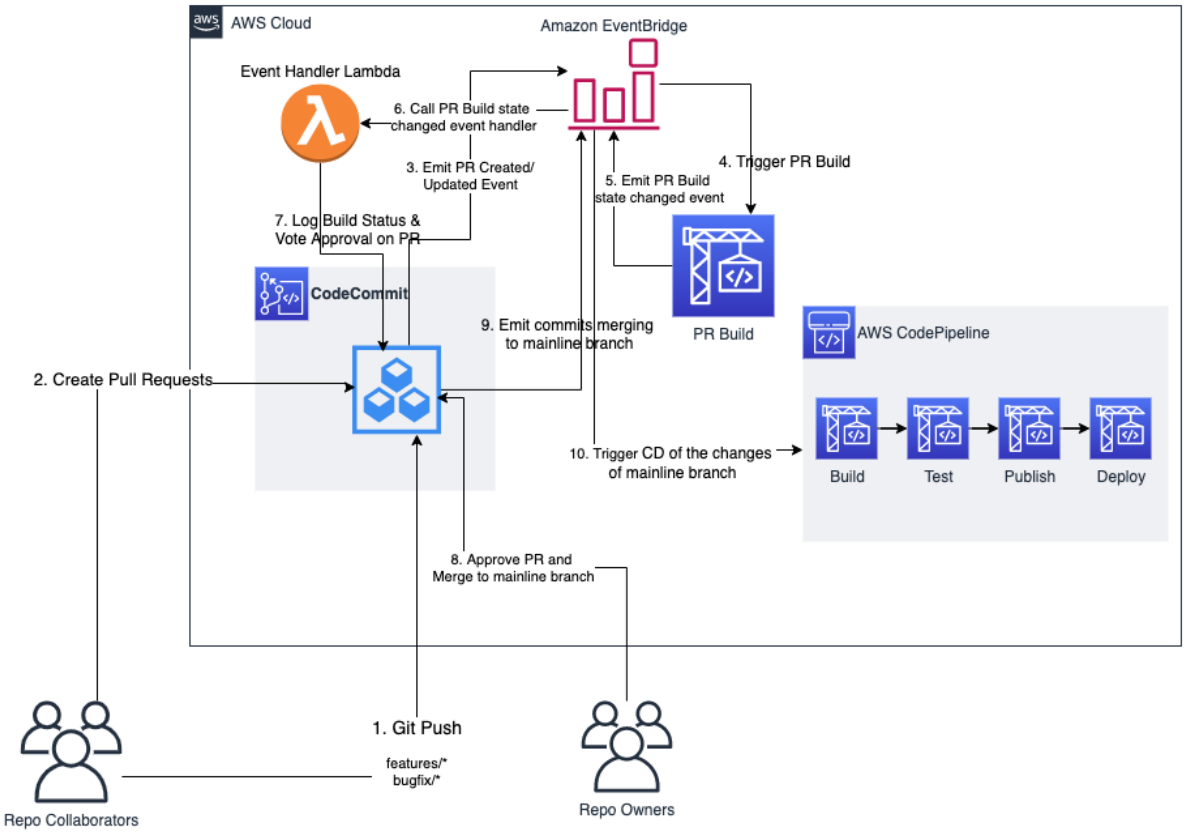
AWS CodeCommit

CodeCommit is a secure, highly scalable, managed source control service that hosts private Git repositories. CodeCommit eliminates the need for you to manage your own source control system or worry about scaling its infrastructure. You can use CodeCommit to store anything from code to binaries. It supports the standard functionality of Git, so it works seamlessly with your existing Git-based tools.

With CodeCommit, you can:

- Benefit from a fully managed service hosted by AWS. CodeCommit provides high service availability and durability and eliminates the administrative overhead of managing your own hardware and software. There is no hardware to provision and scale and no server software to install, configure, and update.
- Store your code securely. CodeCommit repositories are encrypted at rest as well as in transit.
- Work collaboratively on code. CodeCommit repositories support pull requests, where users can review and comment on each other's code changes before merging them to branches; notifications that automatically send emails to users about pull requests and comments; and more.
- Easily scale your version control projects. CodeCommit repositories can scale up to meet your development needs. The service can handle repositories with large numbers of files or branches, large file sizes, and lengthy revision histories.
- Store anything, anytime. CodeCommit has no limit on the size of your repositories or on the file types you can store.
- Integrate with other AWS and third-party services. CodeCommit keeps your repositories close to your other production resources in the AWS Cloud, which helps increase the speed and frequency of your development lifecycle. It is integrated with IAM and can be used with other AWS services and in parallel with other repositories. For more information, see Product and service integrations with AWS CodeCommit.

- Easily migrate files from other remote repositories. You can migrate to CodeCommit from any Git-based repository.

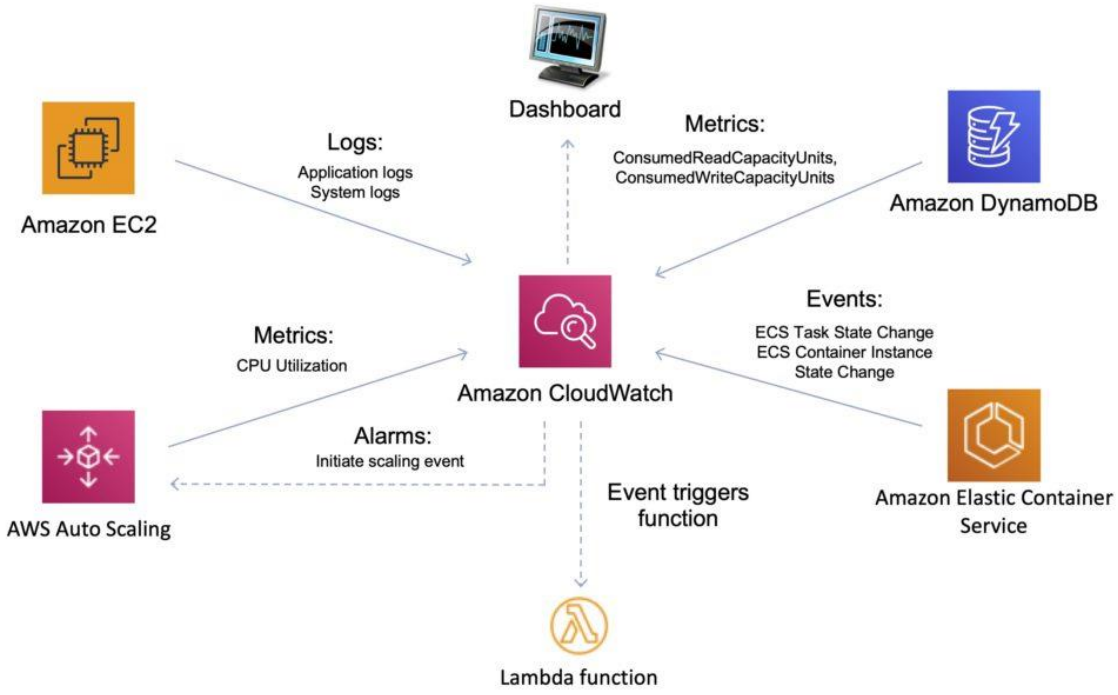


Amazon CloudWatch

Amazon CloudWatch monitors your Amazon Web Services (AWS) resources and the applications you run on AWS in real time. You can use CloudWatch to collect and track metrics, which are variables you can measure for your resources and applications.

The CloudWatch home page automatically displays metrics about every AWS service you use. You can additionally create custom dashboards to display metrics about your custom applications and display custom collections of metrics that you choose.

You can create alarms that watch metrics and send notifications or automatically make changes to the resources you are monitoring when a threshold is breached. For example, you can monitor the CPU usage and disk reads and writes of your Amazon EC2 instances and then use that data to determine whether you should launch additional instances to handle increased load. You can also use this data to stop underused instances to save money.



Amazon EBS (Elastic Block Store)

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

We recommend Amazon EBS for data that must be quickly accessible and requires long-term persistence. EBS volumes are particularly well-suited for use as the primary storage for file systems, databases, or for any applications that require fine granular updates and access to raw, unformatted, block-level storage. Amazon EBS is well suited to both database-style applications that rely on random reads and writes, and to throughput-intensive applications that perform long, continuous reads and writes.

With Amazon EBS, you pay only for what you use. For more information about Amazon EBS pricing, see the Projecting Costs Section of the Amazon Elastic Block Store page.

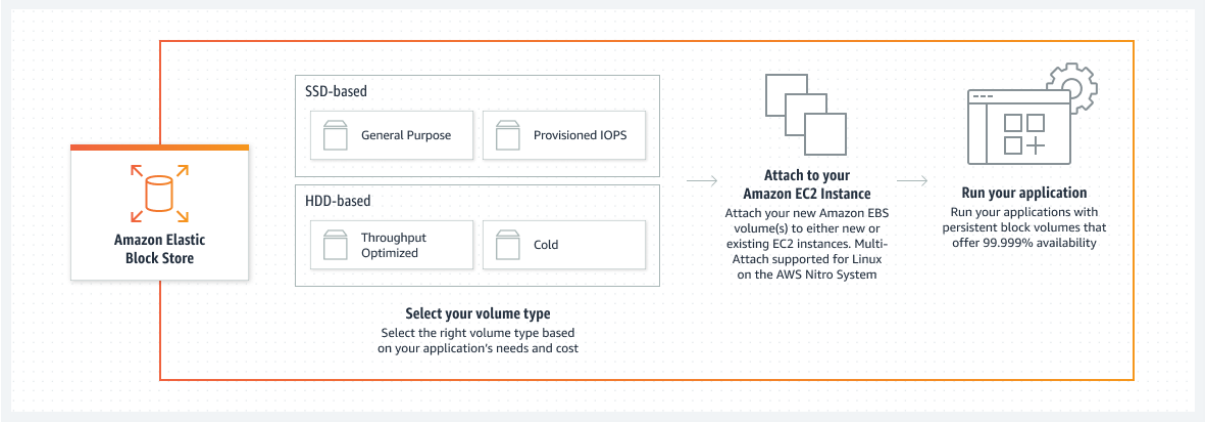
Features of Amazon EBS

- You create an EBS volume in a specific Availability Zone, and then attach it to an instance in that same Availability Zone. To make a volume available outside of the Availability Zone, you can create a snapshot and restore that snapshot to a new volume anywhere in that Region. You can copy snapshots to other Regions and then restore them to new volumes there, making it easier to leverage multiple AWS Regions for geographical expansion, data centre migration, and disaster recovery.

- Amazon EBS provides the following volume types: General Purpose SSD, Provisioned IOPS SSD, Throughput Optimized HDD, and Cold HDD. For more information, see EBS volume types.
  The following is a summary of performance and use cases for each volume type.
  - General Purpose SSD volumes (gp2 and gp3) balance price and performance for a wide variety of transactional workloads. These volumes are ideal for use cases such as boot volumes, medium-size single instance databases, and development and test environments.
  - Provisioned IOPS SSD volumes (io1 and io2) are designed to meet the needs of I/O-intensive workloads that are sensitive to storage performance and consistency. They provide a consistent IOPS rate that you specify when you create the volume. This enables you to predictably scale to tens of thousands of IOPS per instance. Additionally, io2 volumes provide the highest levels of volume durability.
  - Throughput Optimized HDD volumes (st1) provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. These volumes are ideal for large, sequential workloads such as Amazon EMR, ETL, data warehouses, and log processing.
  - Cold HDD volumes (sc1) provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. These volumes are ideal for large, sequential, cold-data workloads. If you require infrequent access to your data and are looking to save costs, these volumes provide inexpensive block storage.

- You can create your EBS volumes as encrypted volumes, in order to meet a wide range of data-at-rest encryption requirements for regulated/audited data and applications. When you create an encrypted EBS volume and attach it to a supported instance type, data stored at rest on the volume, disk I/O, and snapshots created from the volume are all encrypted. Encryption occurs on the servers that host EC2 instances, providing encryption of data-in-transit from EC2 instances to EBS storage. For more information, see Amazon EBS encryption.

- Performance metrics, such as bandwidth, throughput, latency, and average queue length, are available through the AWS Management Console. These metrics, provided by Amazon CloudWatch, allow you to monitor the performance of your volumes to make sure that you are providing enough performance for your applications without paying for resources you don't need.



Amazon Aurora

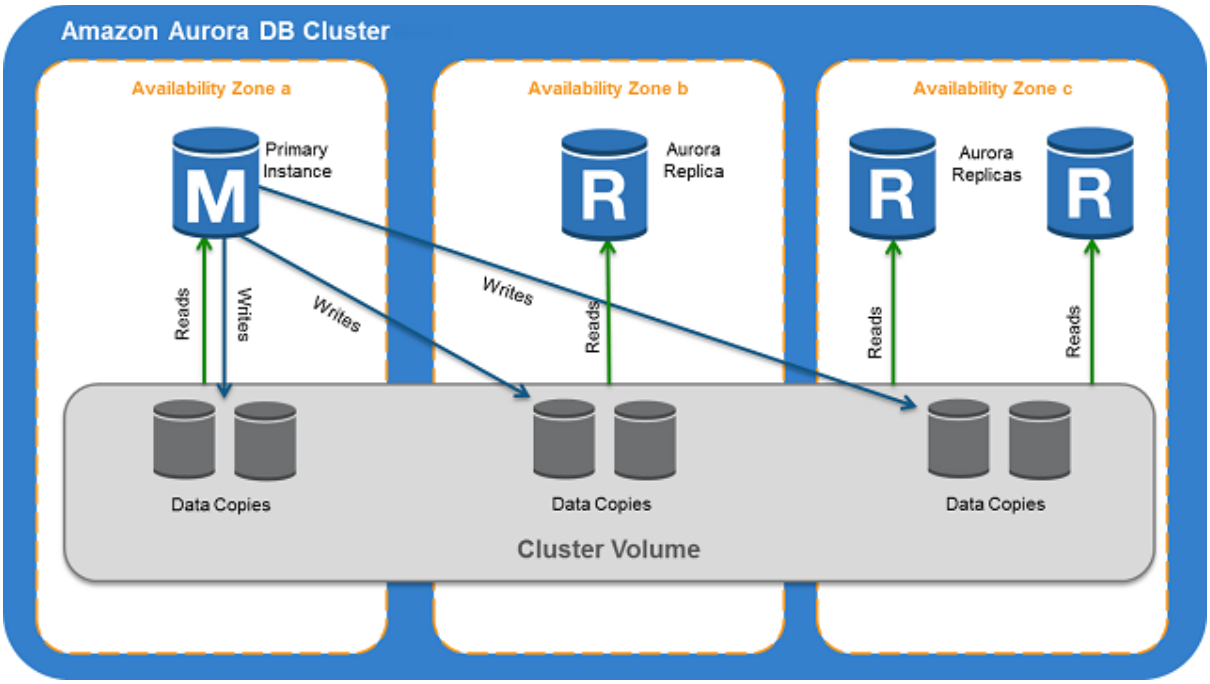
Amazon Aurora (Aurora) is a fully managed relational database engine that's compatible with MySQL and PostgreSQL. You already know how MySQL and PostgreSQL combine the speed and reliability of high-end commercial databases with the simplicity and cost-effectiveness of open-source databases. The code, tools, and applications you use today with your existing MySQL and PostgreSQL databases can be used with Aurora. With some workloads, Aurora can deliver up to five times the throughput of MySQL and up to three times the throughput of PostgreSQL without requiring changes to most of your existing applications.

Aurora includes a high-performance storage subsystem. Its MySQL- and PostgreSQL-compatible database engines are customized to take advantage of that fast distributed storage. The underlying storage grows automatically as needed. An Aurora cluster volume can grow to a maximum size of 128 tebibytes (TiB). Aurora also automates and standardizes database clustering and replication, which are typically among the most challenging aspects of database configuration and administration.

Aurora is part of the managed database service Amazon Relational Database Service (Amazon RDS). Amazon RDS is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. If you are not already familiar with Amazon RDS, see the *Amazon Relational Database Service User Guide*.

The following points illustrate how Amazon Aurora relates to the standard MySQL and PostgreSQL engines available in Amazon RDS:

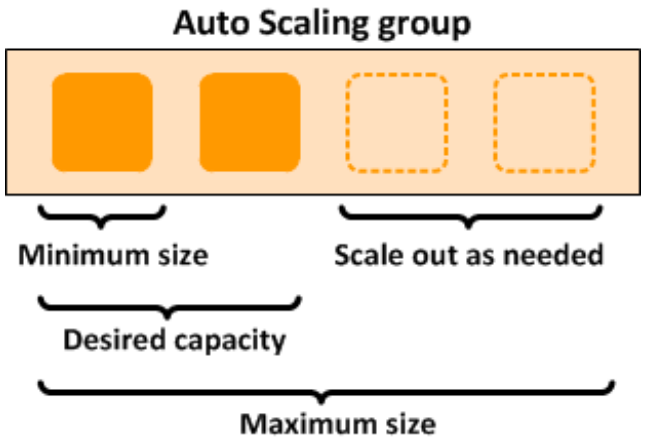
- You choose Aurora MySQL or Aurora PostgreSQL as the DB engine option when setting up new database servers through Amazon RDS.
- Aurora takes advantage of the familiar Amazon Relational Database Service (Amazon RDS) features for management and administration. Aurora uses the Amazon RDS AWS Management Console interface, AWS CLI commands, and API operations to handle routine database tasks such as provisioning, patching, backup, recovery, failure detection, and repair.
- Aurora management operations typically involve entire clusters of database servers that are synchronized through replication, instead of individual database instances. The automatic clustering, replication, and storage allocation make it simple and cost-effective to set up, operate, and scale your largest MySQL and PostgreSQL deployments.
- You can bring data from Amazon RDS for MySQL and Amazon RDS for PostgreSQL into Aurora by creating and restoring snapshots, or by setting up one-way replication. You can use push-button migration tools to convert your existing RDS for MySQL and RDS for PostgreSQL applications to Aurora.



AWS Autoscaling

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 minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes below this size. You can specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size. If you specify the desired capacity, either when you create the group or at any time thereafter, Amazon EC2 Auto Scaling ensures that your group has this many instances. If you specify scaling policies, then Amazon EC2 Auto Scaling can launch or terminate instances as demand on your application increases or decreases.

For example, 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. The scaling policies that you define adjust the number of instances, within your minimum and maximum number of instances, based on the criteria that you specify.



Auto scaling benefits

Adding Amazon EC2 Auto Scaling to your application architecture is one way to maximize the benefits of the AWS Cloud. When you use Amazon EC2 Auto Scaling, your applications gain the following benefits:

- Better fault tolerance. Amazon EC2 Auto Scaling can detect when an instance is unhealthy, terminate it, and launch an instance to replace it. You can also configure Amazon EC2 Auto Scaling to use multiple Availability Zones. If one Availability Zone becomes unavailable, Amazon EC2 Auto Scaling can launch instances in another one to compensate.
- Better availability. Amazon EC2 Auto Scaling helps ensure that your application always has the right amount of capacity to handle the current traffic demand.
- Better cost management. Amazon EC2 Auto Scaling can dynamically increase and decrease capacity as needed. Because you pay for the EC2 instances you use, you save money by launching instances when they are needed and terminating them when they aren't.

ARCHITECTURE

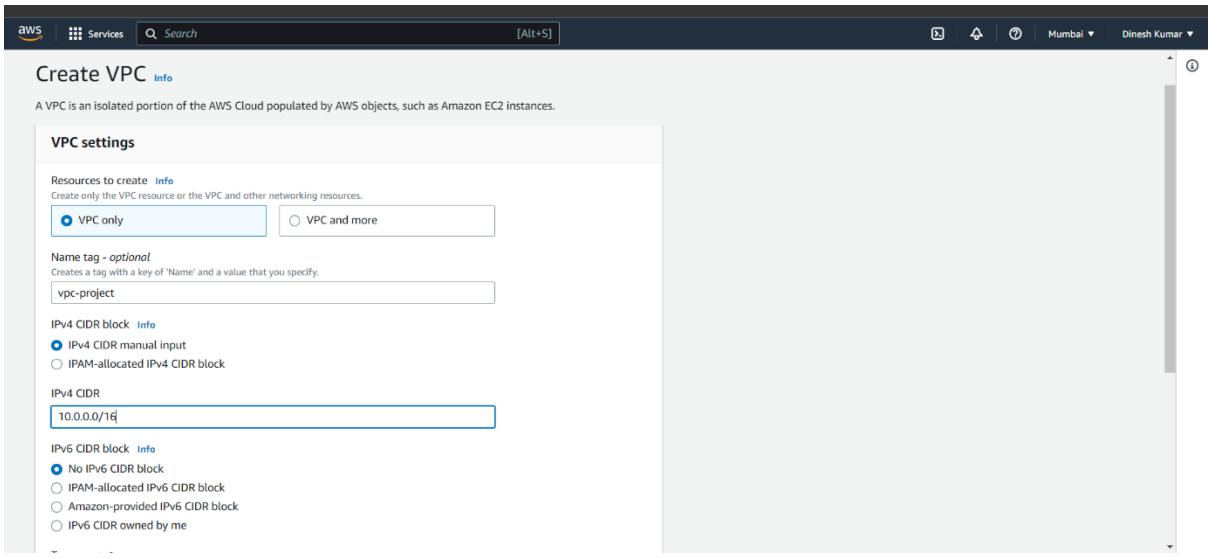
THREE-TIER ARCHITECTURE for AWS Cloud Infrastructure

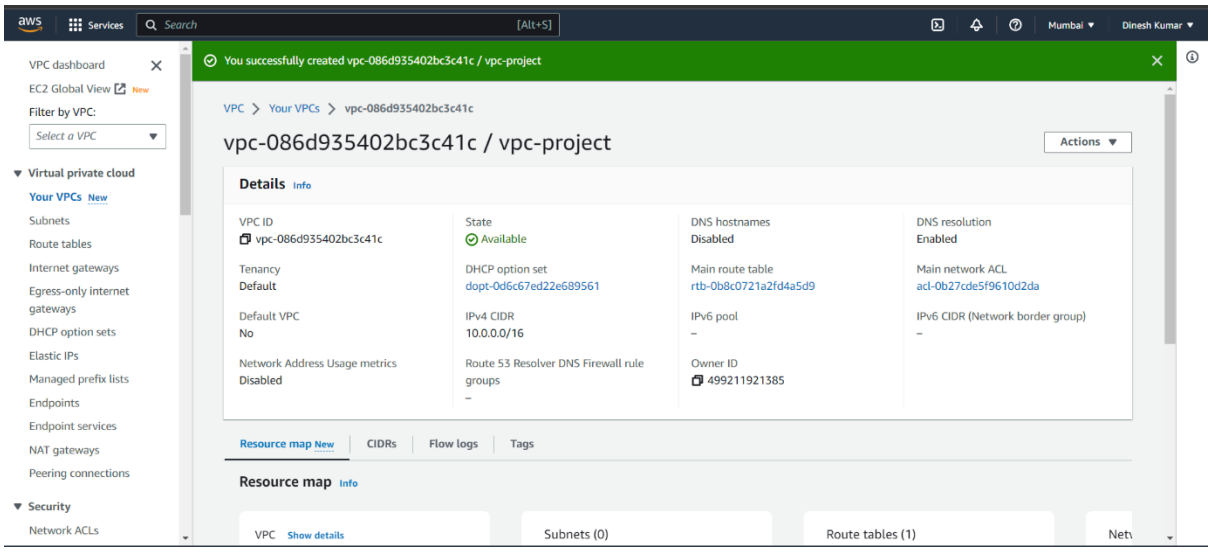
IMPLEMENTATION

Let's Begin by Creating a VPC

A virtual private cloud (VPC) is a virtual network allocated to your Three-Tier AWS Architecture account. It is logically removed from other virtual networks in the AWS Cloud. You can launch your **AWS connect resources**, like Amazon EC2 instances, into your VPC.

- Open the VPC section of the AWS services, and click on the **Create VPC** button.
- Name your VPC and give it a CIDR block of 10.0.0.0/16.
- Click on **Create VPC**.

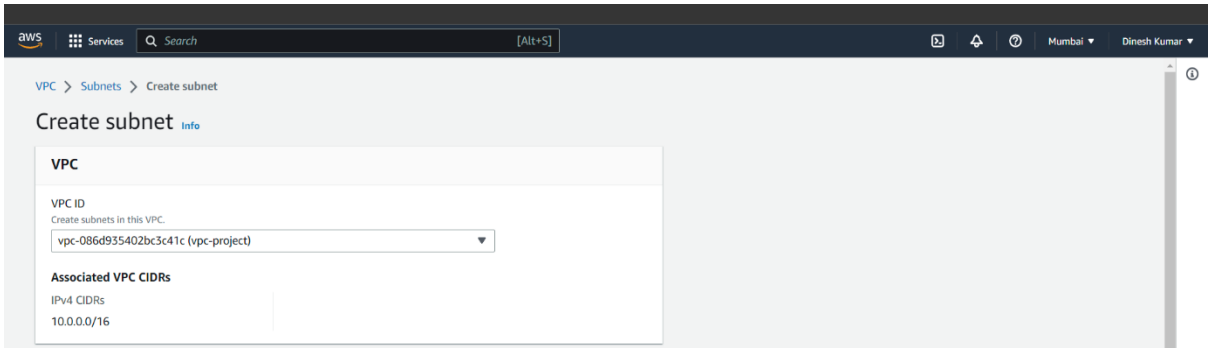




Create the Required Subnets

AWS subnets is the splitting up of an IP network by IP address. You much link each subnet to a routing table that lays out the eligible routes for outbound traffic leaving the subnet. Furthermore, every subnet you create is automatically linked with the primary route table and default network ACL of the VPC.

- Go to the **Subnets** page and click on **Create Subnet**.
- Locate the correct VPC from the VPC ID drop-down.



Here, we will be making 6 subnets. Every subnet needs a CIDR block and also each of these will fall into one of the two availability zones.

The division of the subnets is as follows:

- **Presentation Tier Subnets:**

web-pub-subnet1: 10.0.1.0/24 (ap-south-1a)

web-pub-subnet2: 10.0.2.0/24 (ap-south-1b)

- **Application Tier Subnets:**

app-pvt-subnet1: 10.0.3.0/24 (ap-south-1a)

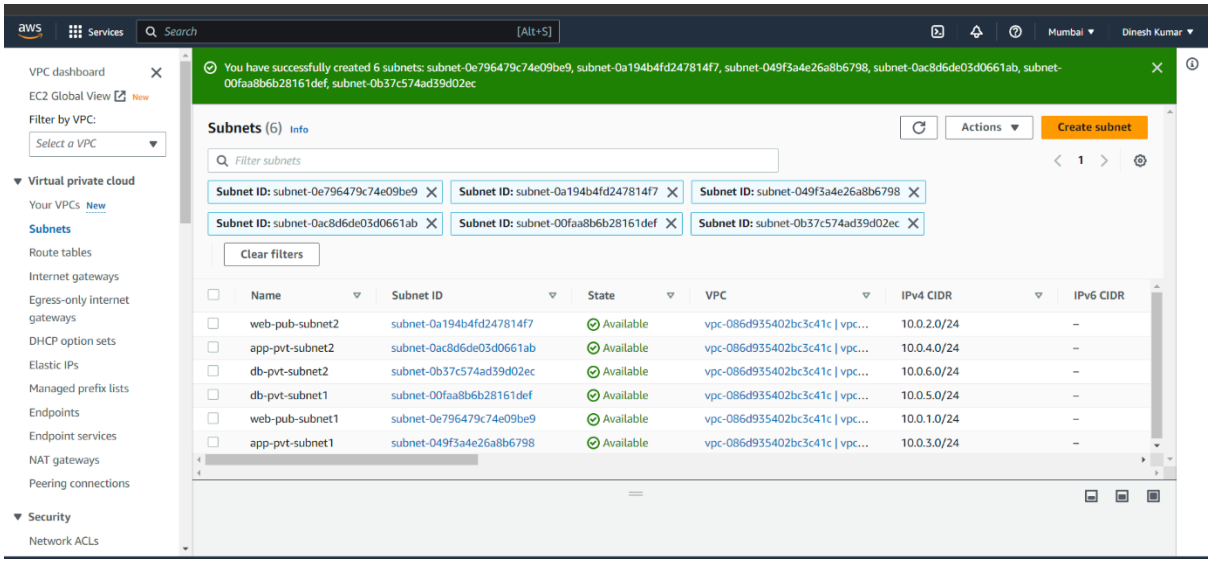
app-pvt-subnet2: 10.0.4.0/24 (ap-south-1b)

- **Database Tier Subnets:**

db-pvt-subnet1: 10.0.5.0/24 (ap-south-1a)

db-pvt-subnet2: 10.0.6.0/24 (ap-south-1b)

Select **Create** once you have configured all the subnets.

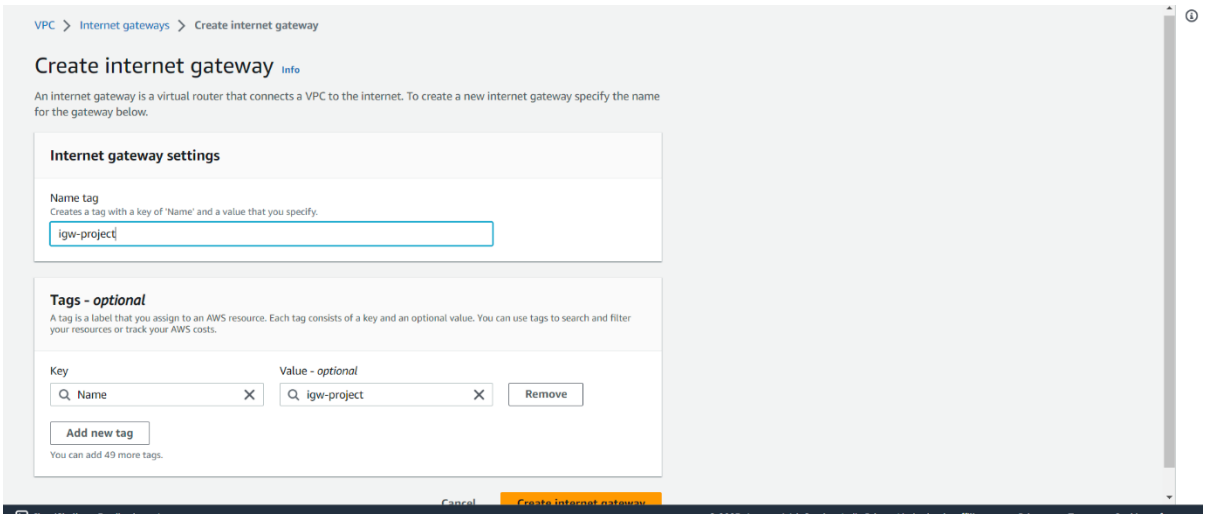


Setup an Internet Gateway

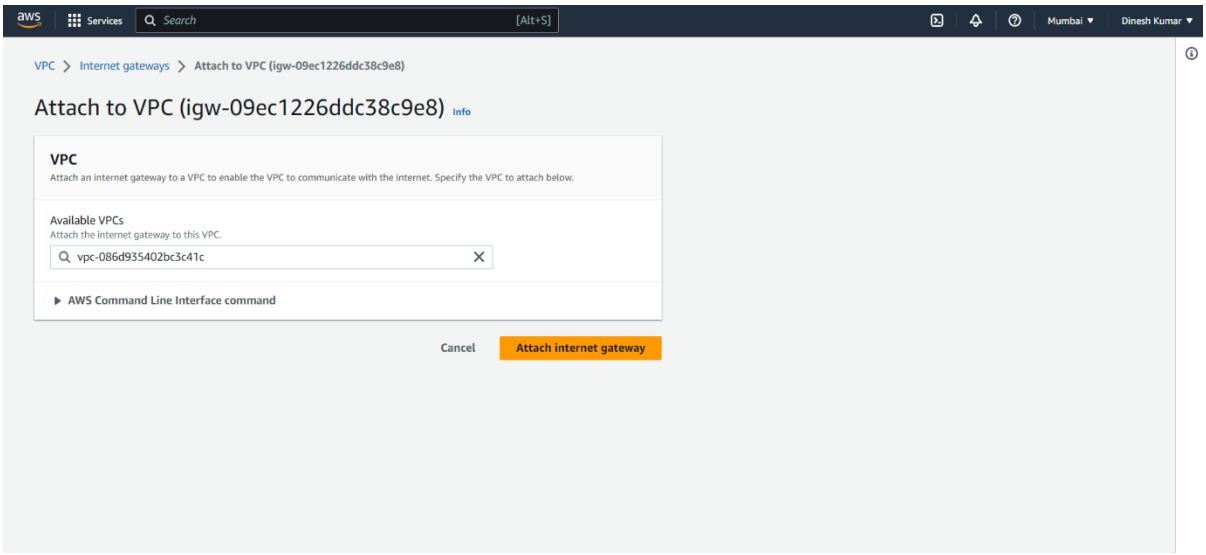
An Internet Gateway is a highly available, horizontally scaled, yet redundant VPC component. It enables communication between the instances in your VPC and the internet using VPC route tables for internet-routable traffic. It supports IPv4 and IPv6 traffic and does not cause bandwidth limitations or availability risks to your network traffic.

You can attach only one internet gateway per VPC. You are not charged additionally for the internet gateway in your account.

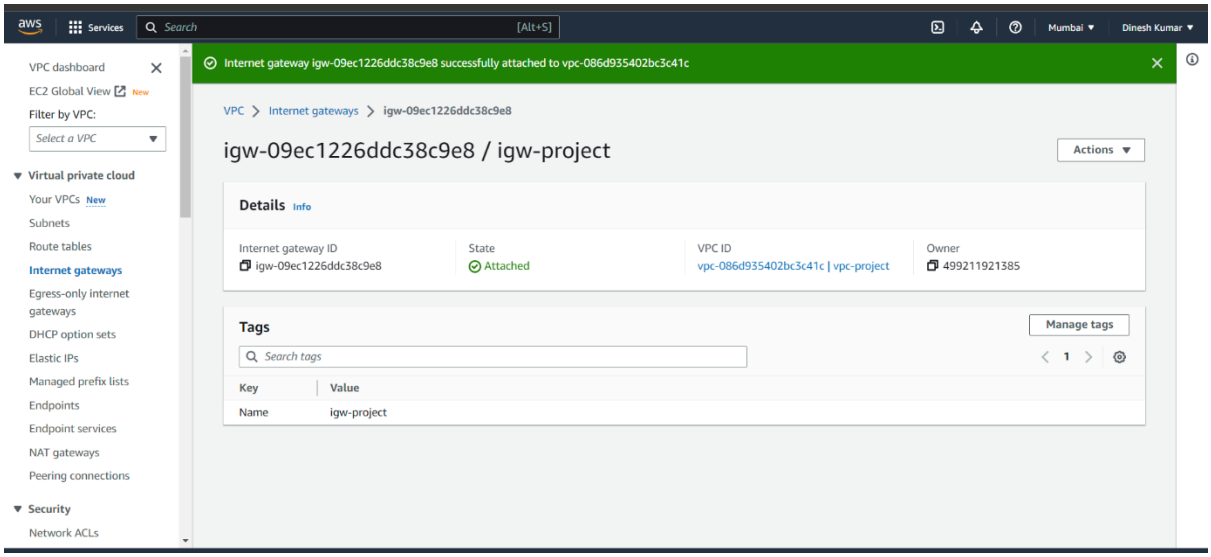
Go to the left-hand drop-down menu and select **Internet Gateway**. Choose to **Create internet gateway**, give it a name, and finish by clicking **Create internet gateway**.



Select **Actions** inside your internet gateway and **Attach to VPC**.



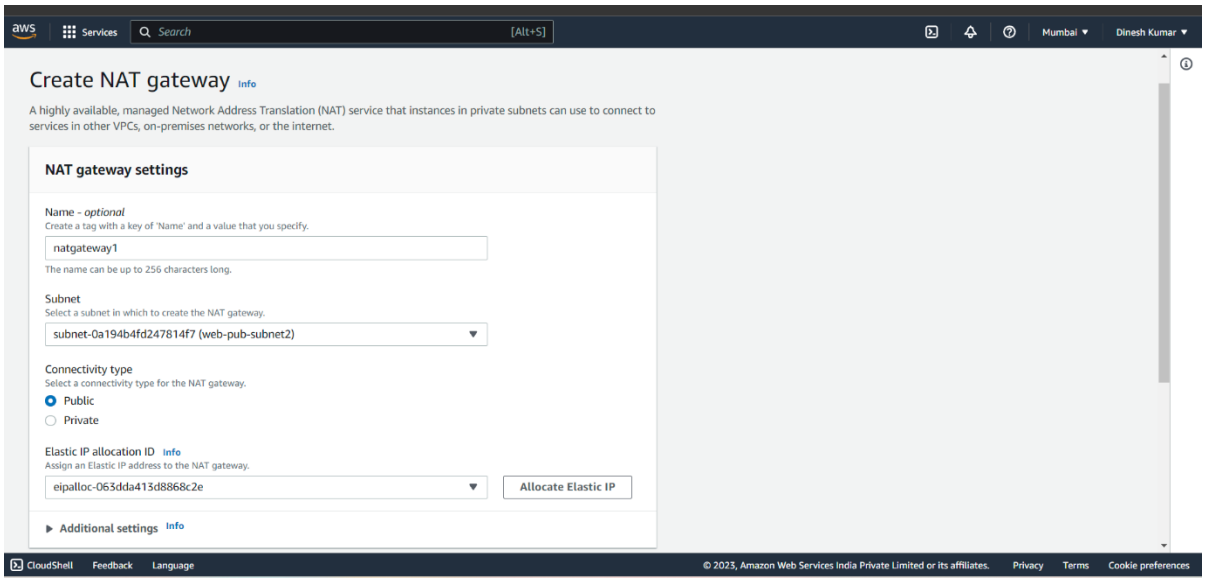
Next, choose your VPC and then click on **Attach internet gateway.**

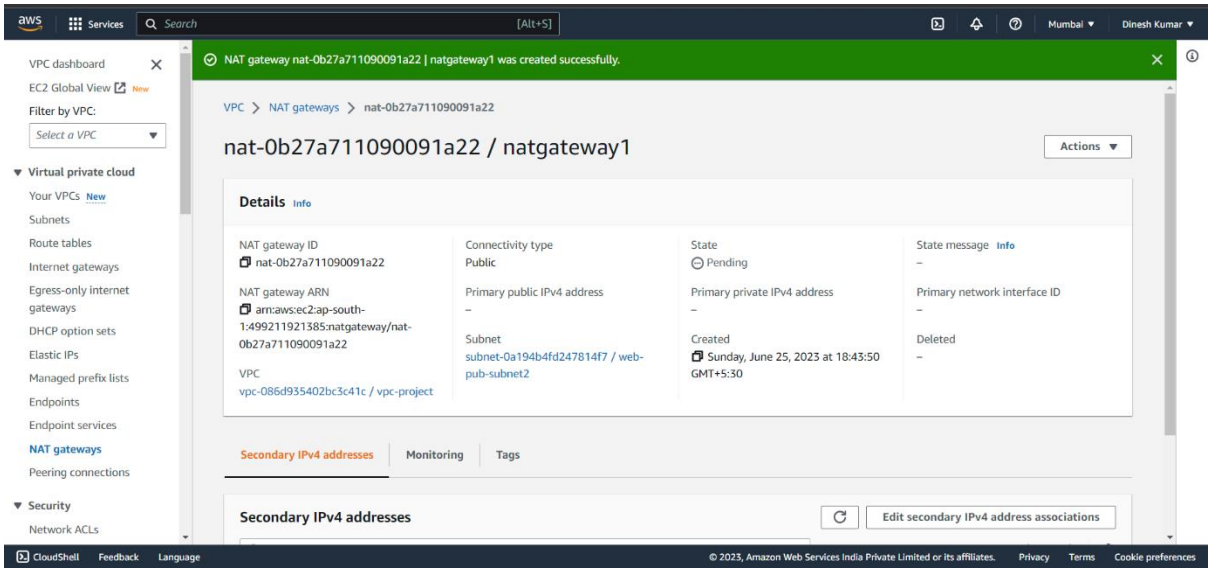


Setup a NAT Gateway

A NAT gateway stands for Network Address Translation (NAT) service. You can employ a NAT gateway to enable the instances in a private subnet to connect to services outside the VPC. Still, external services cannot initiate a connection with those instances.

1. In the navigation panel, choose **NAT Gateways**.
2. Choose **Create NAT gateway** and do the following:
    1. Specify a name for the NAT gateway.
    2. Choose the subnet in which you want to create the NAT gateway.
    3. For **Elastic IP allocation ID**, select an Elastic IP address to associate with the NAT gateway.
    4. Choose **Create a NAT gateway**.





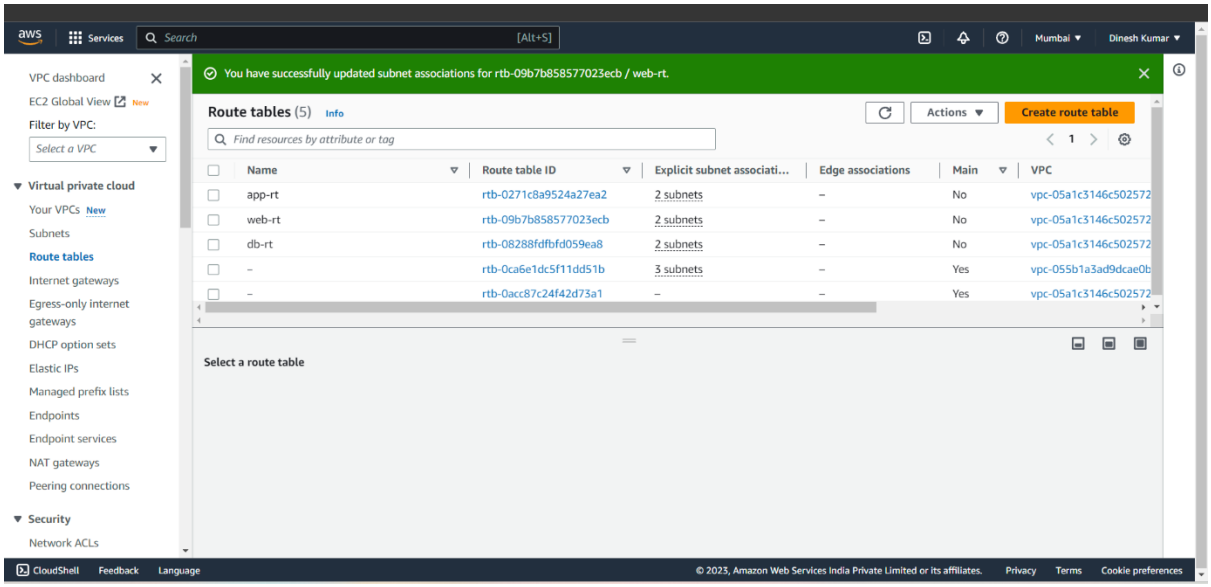
Create the Route Tables and Associate them with the Relevant Subnets

Next, we will go to the **Route Tables** page. A route table carries a set of rules, called routes, that you use to ascertain the network traffic's direction from your gateway or subnet.

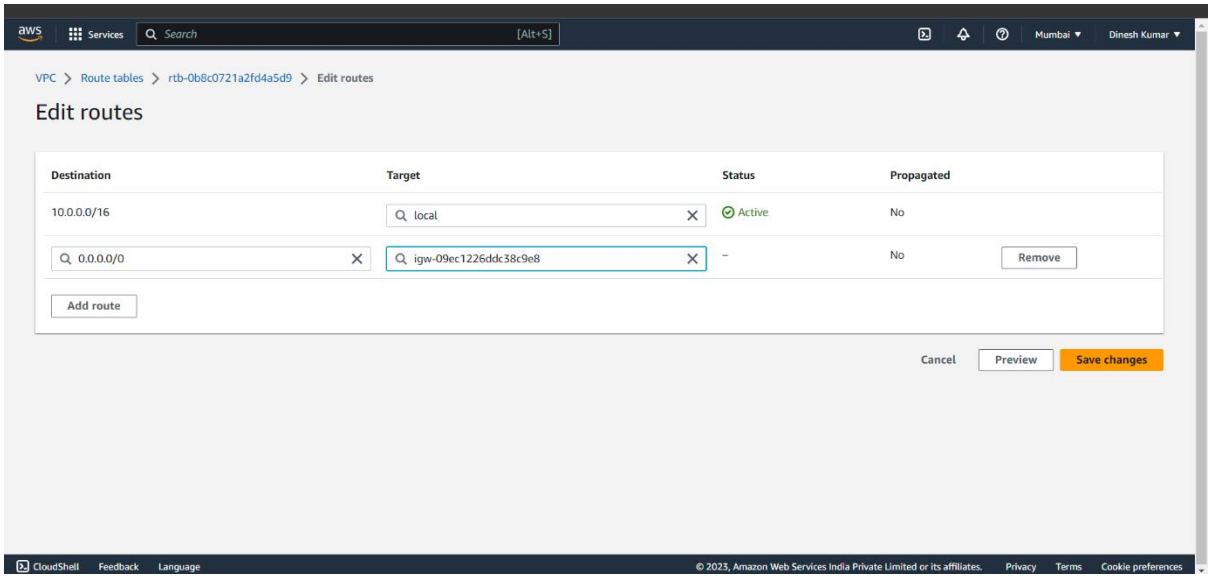
For the Subnets of Application and Database Tiers, you will create two Private Route Tables for each Subnet. Once you are done creating them, you will have to make a public route table for your two Presentation (Web) Subnets.

You can start this by selecting **Create route table**, naming it, then finding your VPC and choosing **Create route table**. Repeat the same process for DB and Web Route Tables.

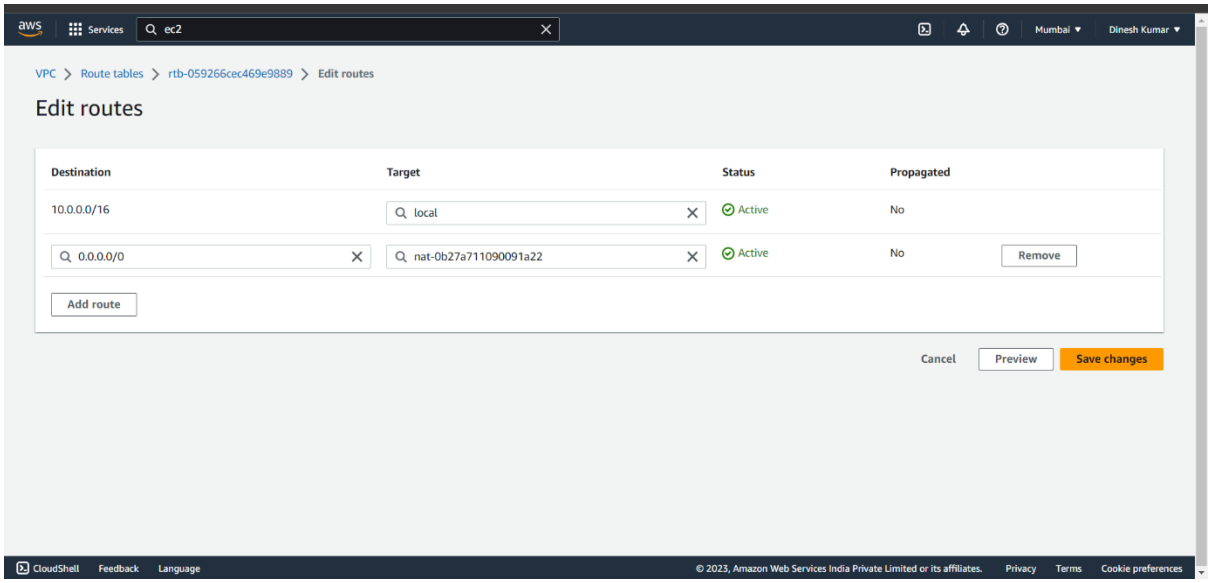
Select **Subnet associations** and then **Edit Subnet associations** inside your new route table. For each table, you can associate the different subnets. Click on **Save association** at the end of each configuration.



Now, we need to associate our public route table with the internet gateway that we created earlier under the **Edit routes** section. Then click on **Save changes**.



Next, go to your private Application Tier & Database Tier route table and select the NAT gateway as the target. Then click **Save changes**.

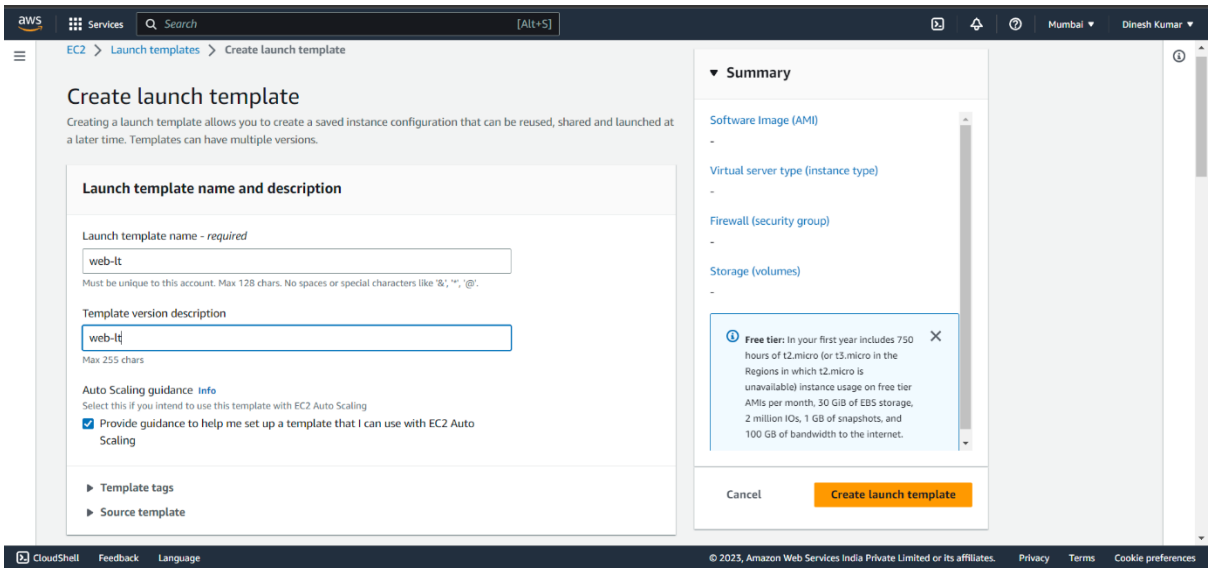


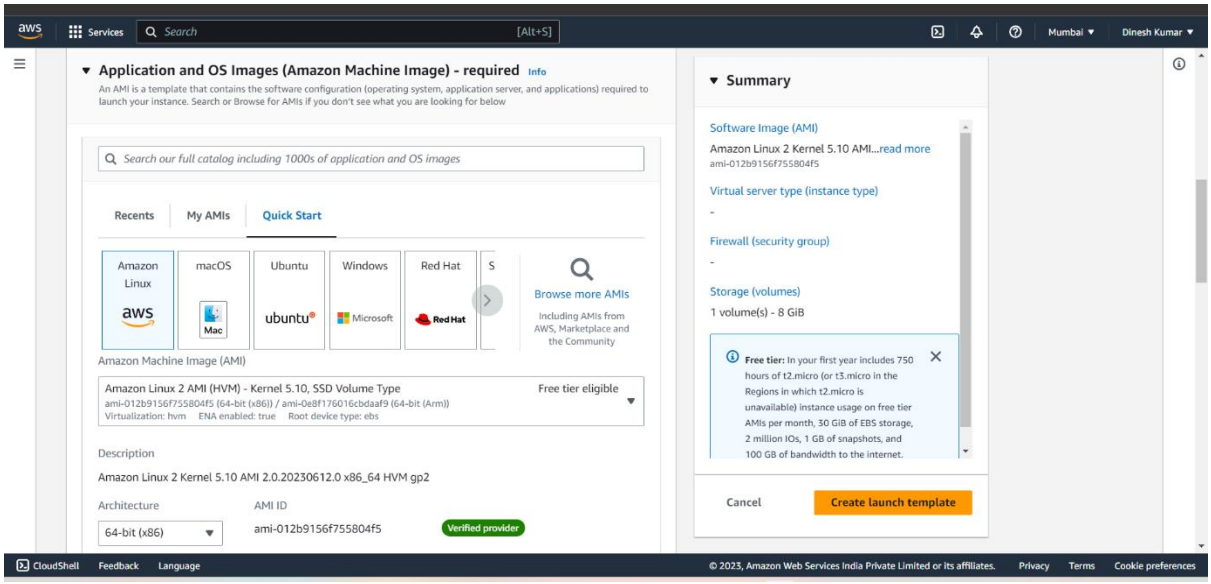
Create a Launch Template with a Bootstrap script

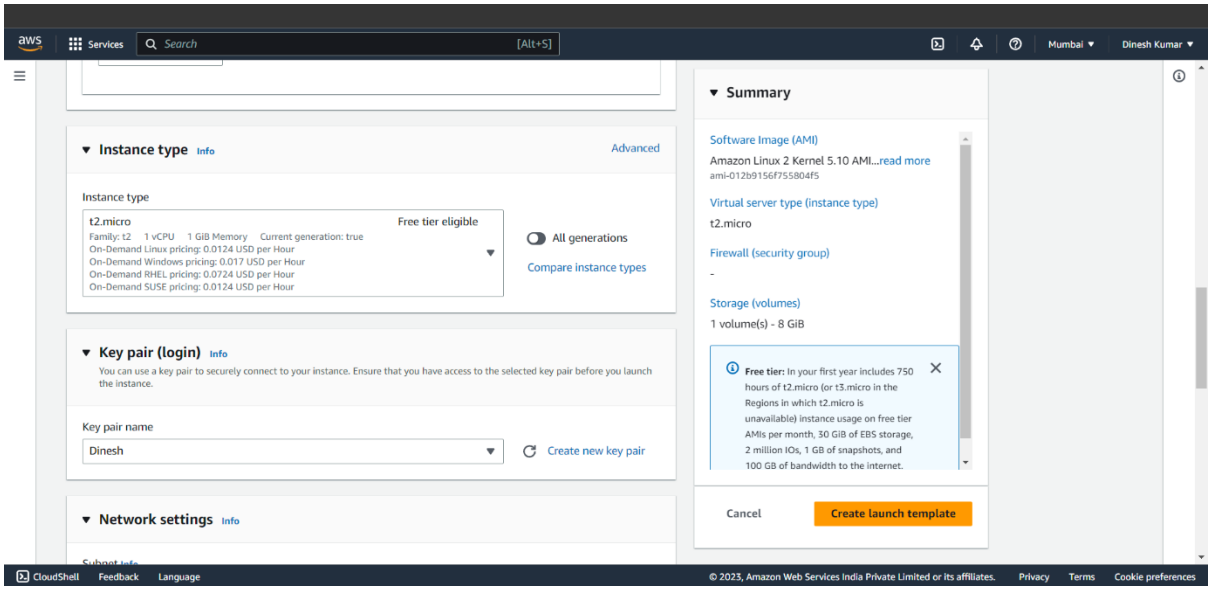
You can build a launch template that has the configuration details to launch an instance. You can use launch templates to store launch parameters so that you do not have to specify them every time you launch an instance. For example, a launch template can contain the instance type, AMI ID, and network settings that you usually utilize to launch instances.

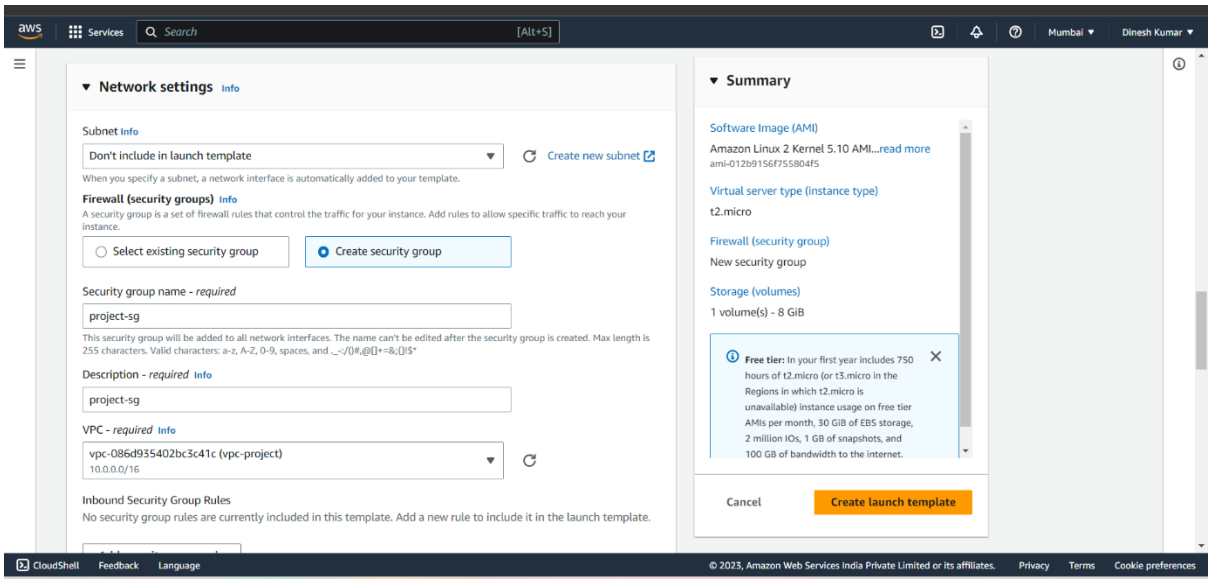
We need to create 2 launch templates – one for the Presentation (web tier) and the other for the Application Tier.

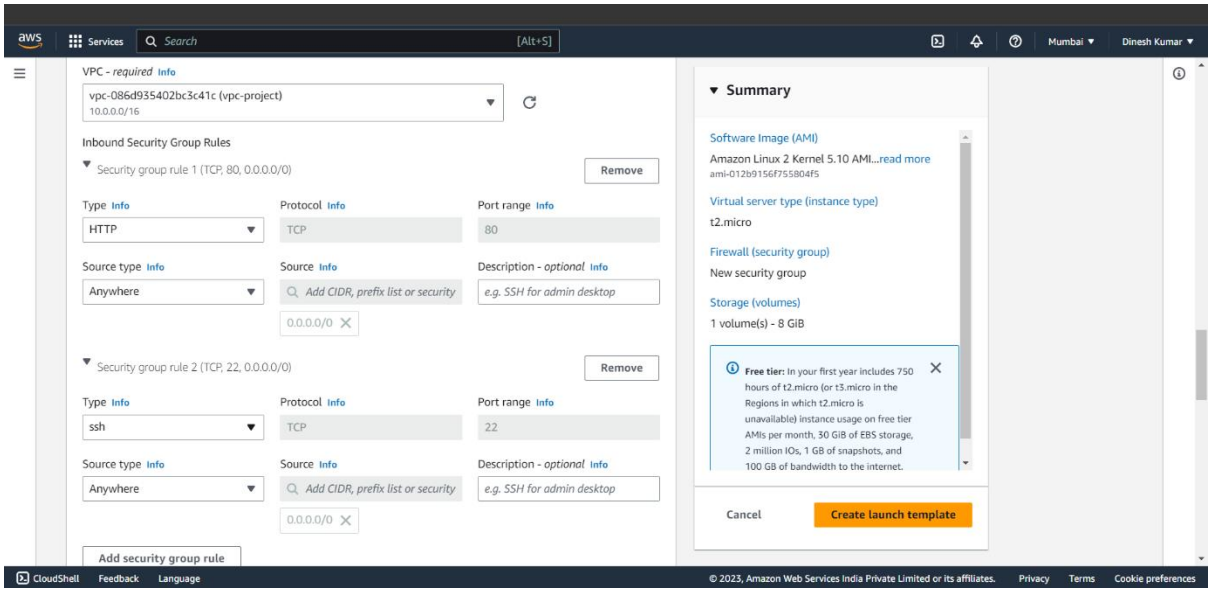
Select **Launch Templates** and **Create launch Templates** from the AWS EC2 Console. Click the Check the box for **Provide guidance to help me set up a template that I can use with EC2 Auto Scaling** under **Auto Scaling guidance**.






Services
Search
[Alt+S]
Mumbai
Dinesh Kumar
Instance type Info
Advanced
Instance type
t2.micro
Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux pricing: 0.0124 USD per Hour
On-Demand Windows pricing: 0.017 USD per Hour
On-Demand RHEL pricing: 0.0724 USD per Hour
On-Demand SUSE pricing: 0.0124 USD per Hour
All generations
Compare instance types
Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name
Dinesh
Create new key pair
Network settings Info
Summary
Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-012b9156f755804f5
Virtual server type (instance type)
t2.micro
Firewall (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 100 GB of bandwidth to the internet.
Cancel
Create launch template
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences


Services
Search
[Alt+S]
Mumbai
Dinesh Kumar
Network settings Info
Subnet Info
Don't include in launch template
Create new subnet
When you specify a subnet, a network interface is automatically added to your template.
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.
Select existing security group
Create security group
Security group name - required
project-sg
This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*
Description - required Info
project-sg
VPC - required Info
vpc-086d935402bc3c41c (vpc-project)
10.0.0.0/16
Inbound Security Group Rules
No security group rules are currently included in this template. Add a new rule to include it in the launch template.
Summary
Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-012b9156f755804f5
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 100 GB of bandwidth to the internet.
Cancel
Create launch template
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences



Lastly, scroll down under **Advanced details** until you see **user data**. Here you can bootstrap an Apache web server with your site details.

Please copy and paste the following code to launch a test web server.

#!/bin/bash

use this code for your user data (script without newlines)

install httpd (linux 2 version)

yum update -y

yum install -y httpd.x86_64

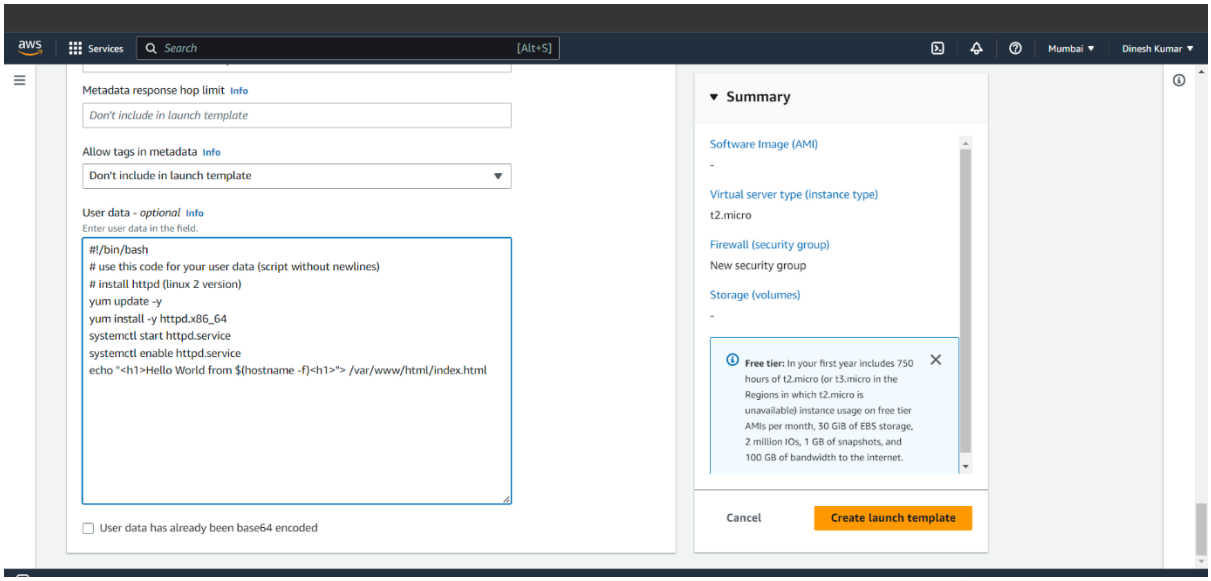
systemctl start httpd.service

systemctl start httpd.service

systemctl enable httpd.service

echo "<h1>Hello World from $(hostname -f)<h1>"> /var/www/html/index.html

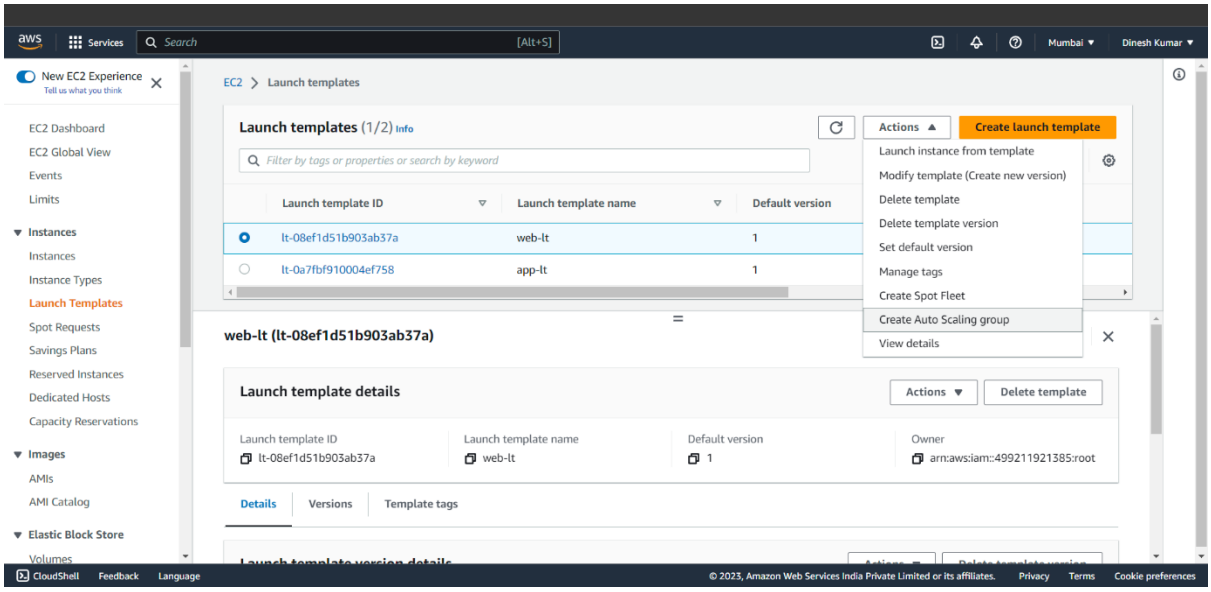
Finish up by selecting “Create launch template”.



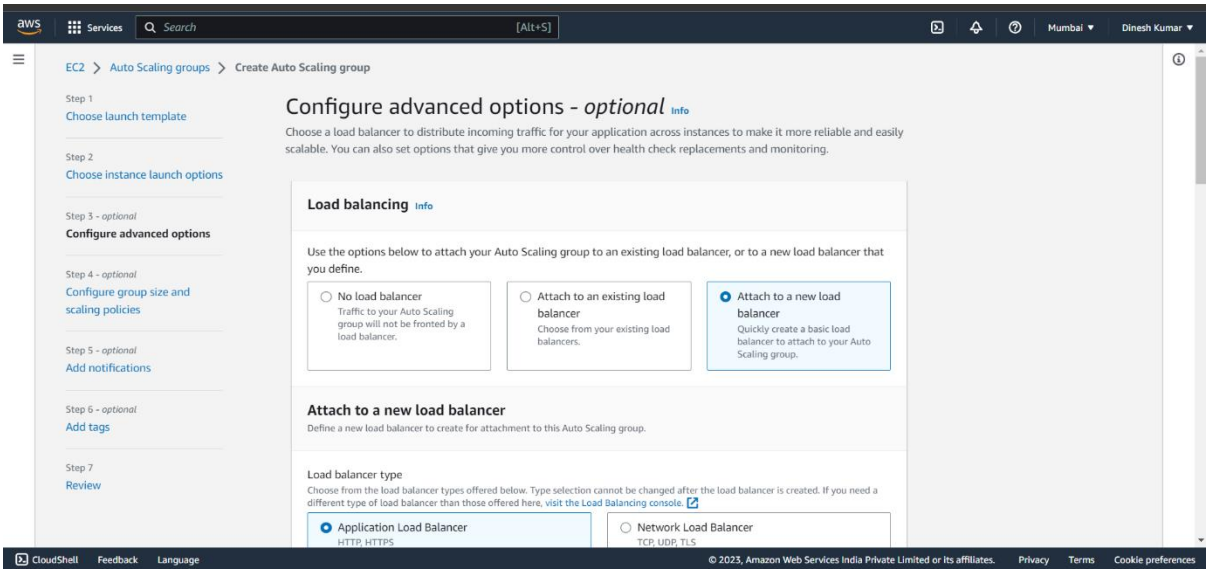
Create one more Launch Template for Application Tier.

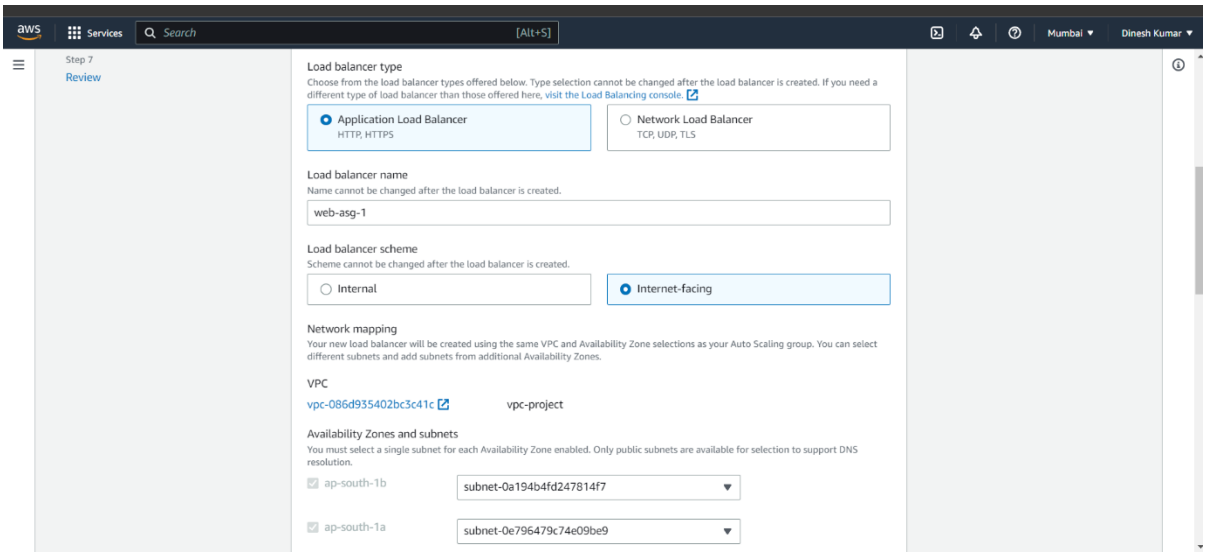
Create Auto Scaling Group

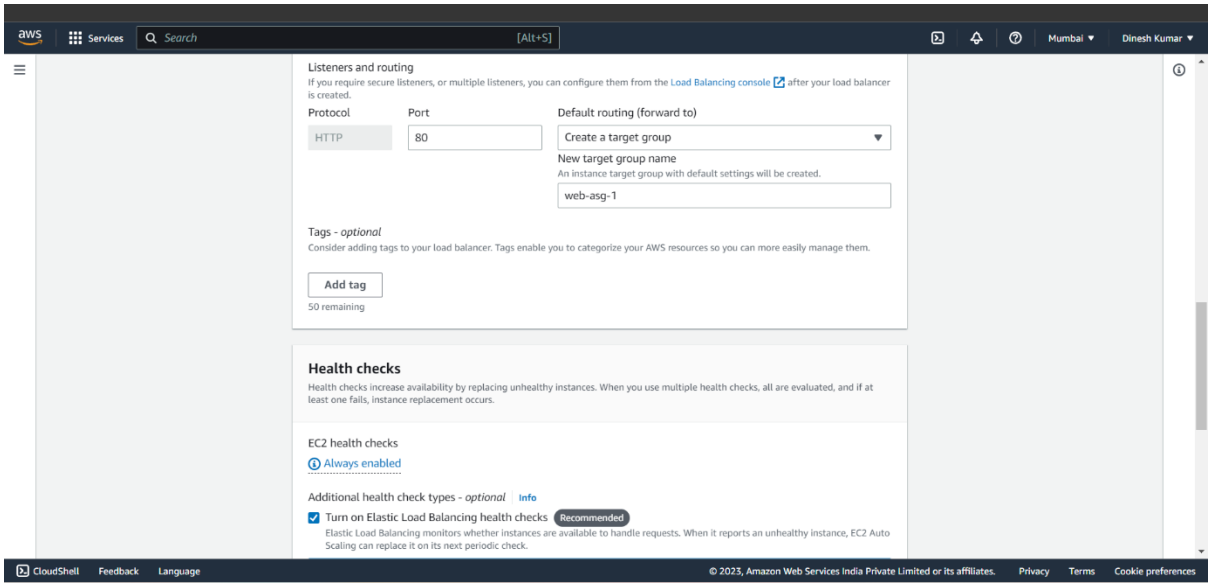
Let's start creating an **Auto Scaling Group** from the same Launch Template page by selecting any of the templates. Go to the **Actions** menu and select **Create Auto Scaling group**.



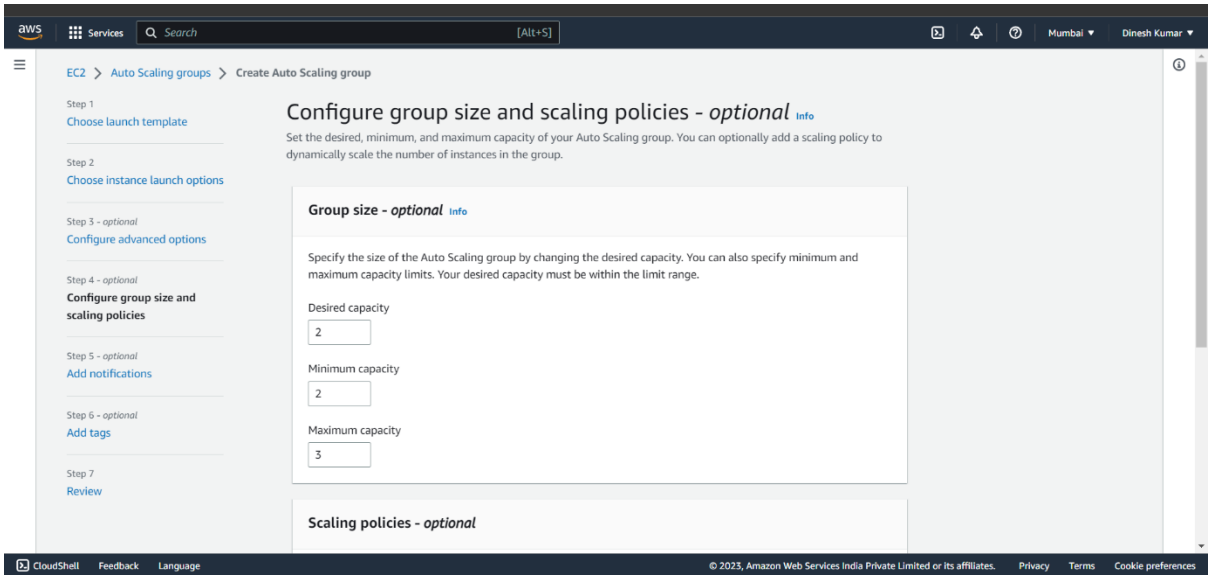
Next, we need to attach a new Internet-facing Application Load Balancer.

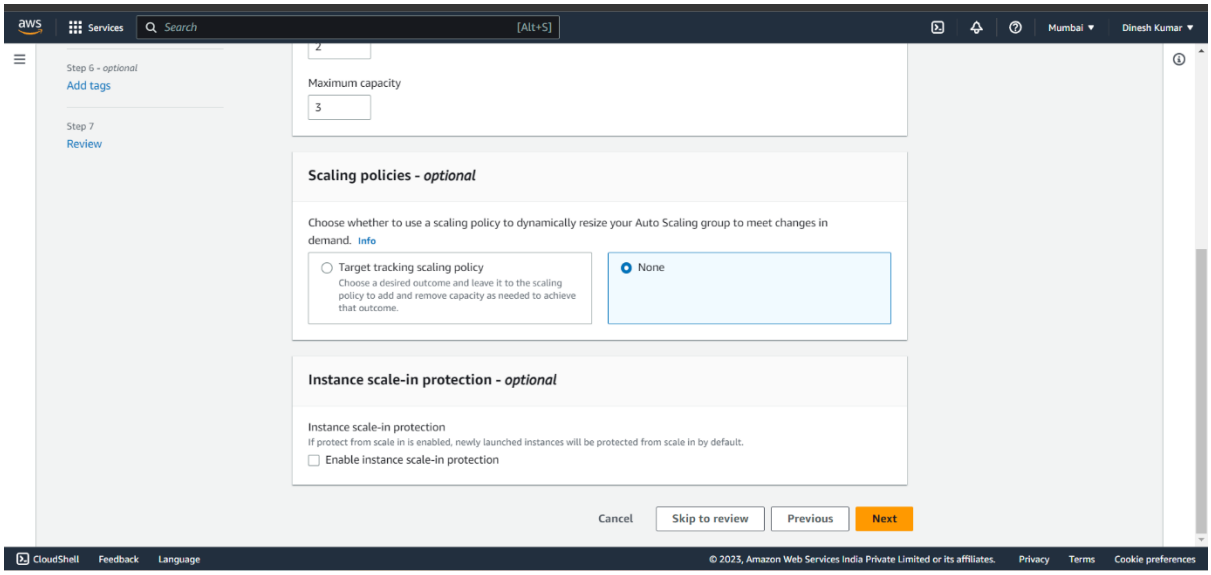




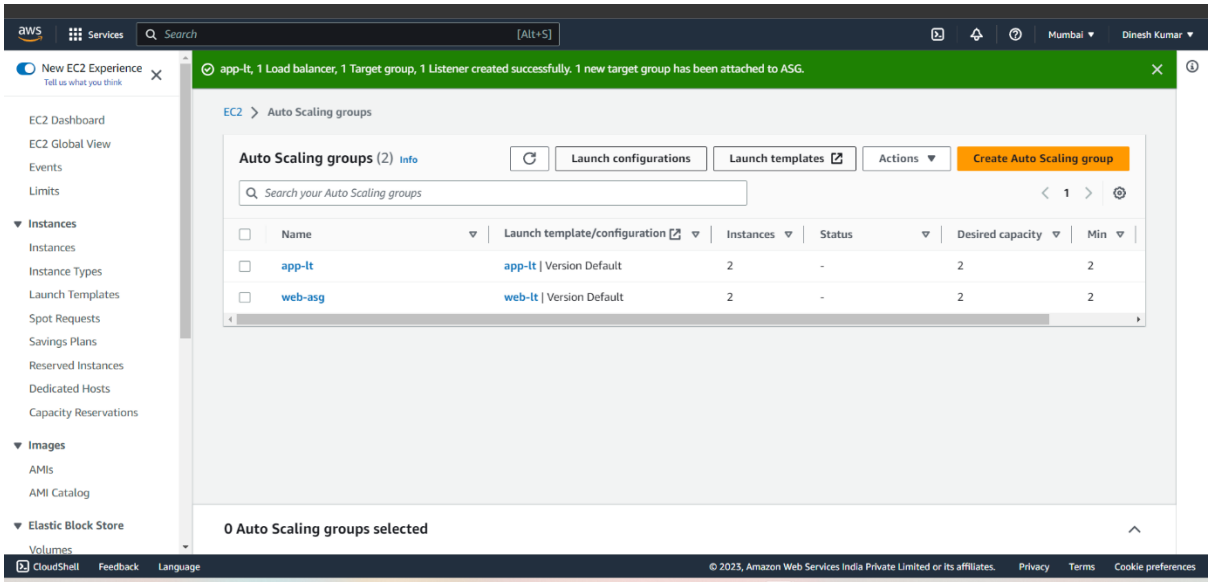


Now we have to configure the group size and scaling policies. For this demo, let's keep the desired capacity of 2, a minimum of 2, and a maximum of 3. You can adjust these numbers as per your needs. After this, you can select **Next** until you can choose to **Create Auto Scaling group**.



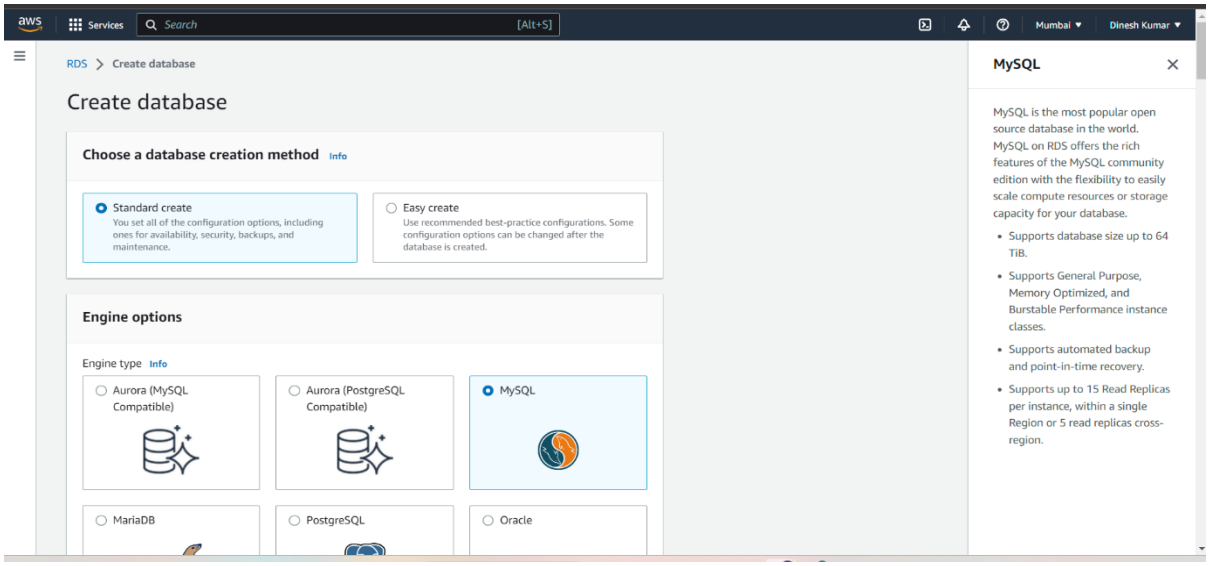


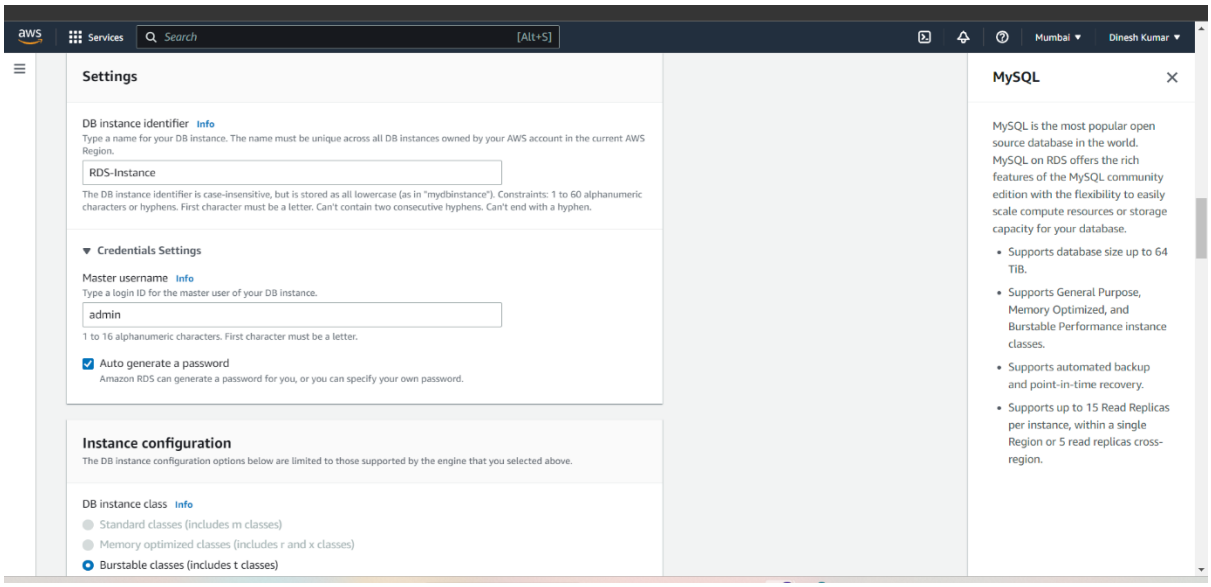
You need to repeat this for the application tier by selecting **Create an Auto Scaling group**. Choose a different name, launch template (app-lt template), and the subnets (application subnets). Please make sure that the ASG for the application tier should be internal facing, not internet facing like for the web tier.

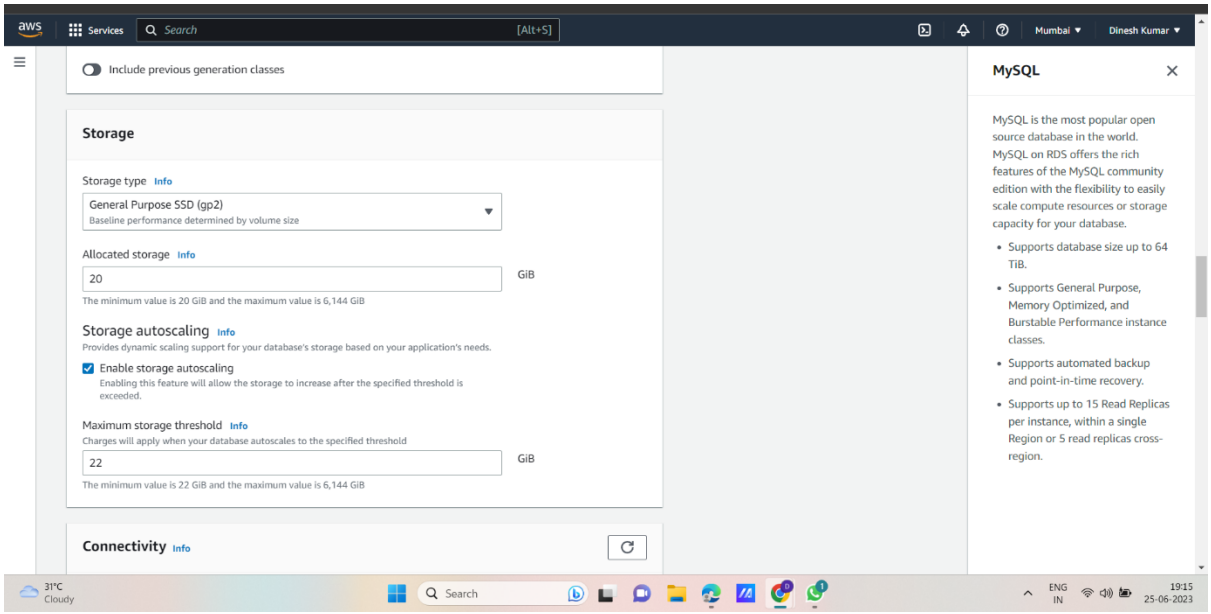


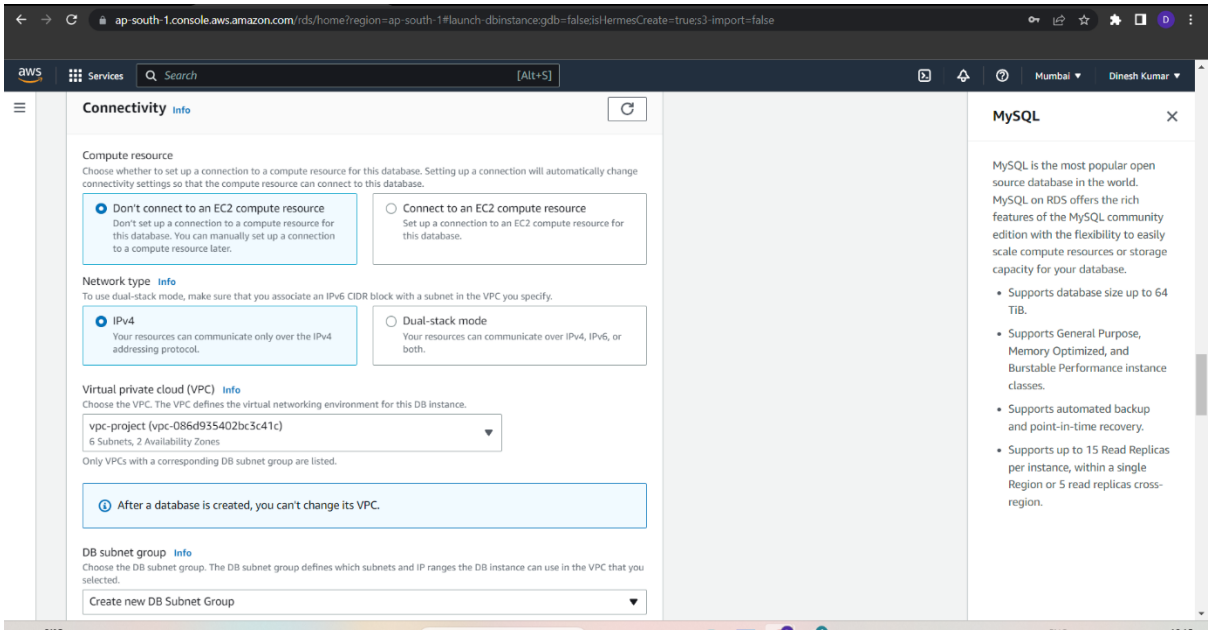
Navigate to the RDS Service

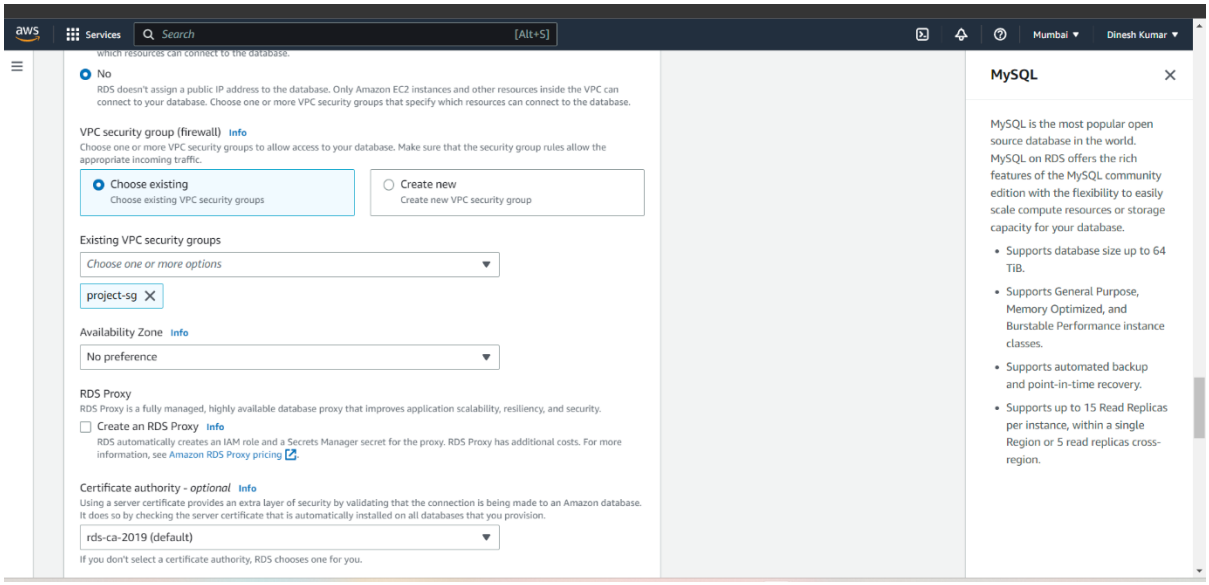
Go to the **RDS Service** on your AWS Management Console. Then click on **Create database.**

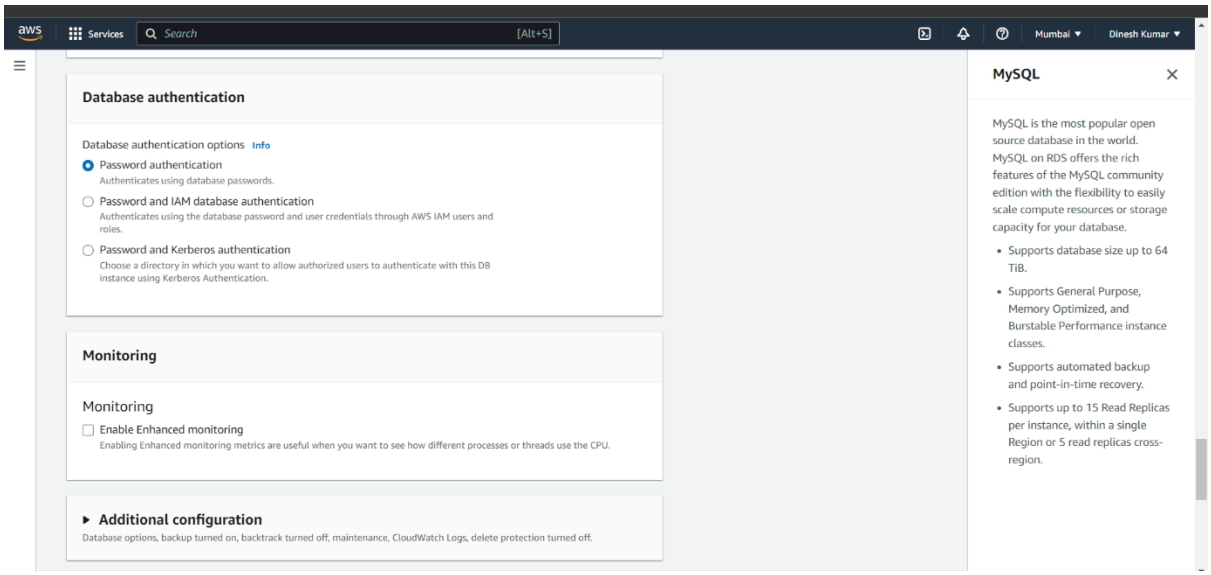


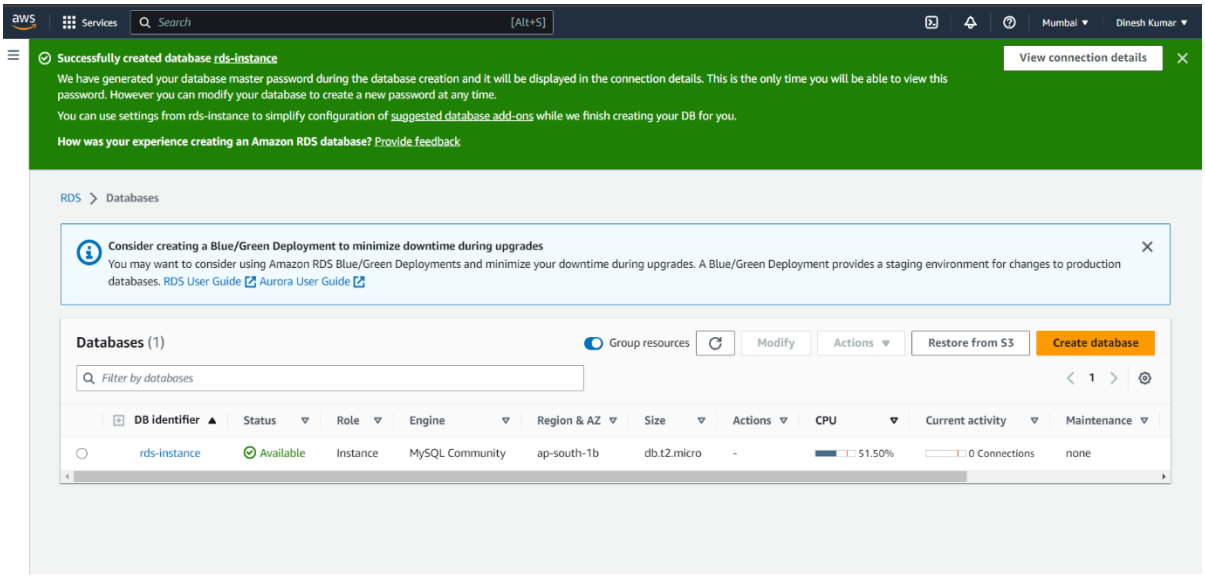



Services
Search
[Alt+S]
Mumbai
Dinesh Kumar
Include previous generation classes
Storage
Storage type Info
General Purpose SSD (gp2)
Baseline performance determined by volume size
Allocated storage Info
20
GiB
The minimum value is 20 GiB and the maximum value is 6,144 GiB
Storage autoscaling Info
Provides dynamic scaling support for your database's storage based on your application's needs.
Enable storage autoscaling
Enabling this feature will allow the storage to increase after the specified threshold is exceeded.
Maximum storage threshold Info
Charges will apply when your database autoscales to the specified threshold
22
GiB
The minimum value is 22 GiB and the maximum value is 6,144 GiB
Connectivity Info
MySQL
MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.
Supports database size up to 64 TiB.
Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
Supports automated backup and point-in-time recovery.
Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.
31°C
Cloudy
Search
ENG
IN
19:15
25-06-2023

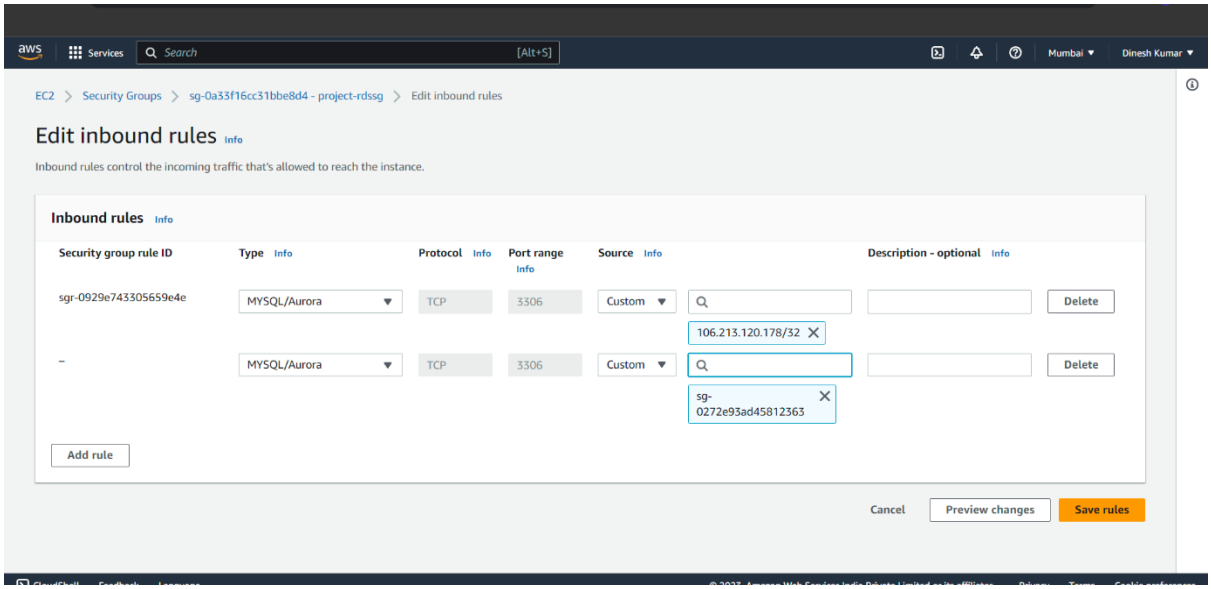

ap-south-1.console.aws.amazon.com/rds/home?region=ap-south-1#launch-dbinstance:gdb=false;isHermesCreate=true;s3-import=false
Services
Search
[Alt+S]
Mumbai
Dinesh Kumar
Connectivity Info
Compute resource
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.
Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.
Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.
Network type Info
To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.
IPv4
Your resources can communicate only over the IPv4 addressing protocol.
Dual-stack mode
Your resources can communicate over IPv4, IPv6, or both.
Virtual private cloud (VPC) Info
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.
vpc-project (vpc-086d935402bc3c41c)
6 Subnets, 2 Availability Zones
Only VPCs with a corresponding DB subnet group are listed.
After a database is created, you can't change its VPC.
DB subnet group Info
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.
Create new DB Subnet Group
MySQL
MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.
Supports database size up to 64 TiB.
Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
Supports automated backup and point-in-time recovery.
Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.


aws
Services
Search
[Alt+S]
Mumbai
Dinesh Kumar
which resources can connect to the database.
No
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.
VPC security group (firewall) Info
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.
Choose existing
Choose existing VPC security groups
Create new
Create new VPC security group
Existing VPC security groups
Choose one or more options
project-sg
Availability Zone Info
No preference
RDS Proxy
RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.
Create an RDS Proxy Info
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see Amazon RDS Proxy pricing.
Certificate authority - optional Info
Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.
rds-ca-2019 (default)
If you don't select a certificate authority, RDS chooses one for you.
MySQL
MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.
Supports database size up to 64 TiB.
Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
Supports automated backup and point-in-time recovery.
Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.


aws
Services
Search
[Alt+S]
Mumbai
Dinesh Kumar
Database authentication
Database authentication options Info
Password authentication
Authenticates using database passwords.
Password and IAM database authentication
Authenticates using the database password and user credentials through AWS IAM users and roles.
Password and Kerberos authentication
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.
Monitoring
Monitoring
Enable Enhanced monitoring
Enabling Enhanced monitoring metrics are useful when you want to see how different processes or threads use the CPU.
Additional configuration
Database options, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.
MySQL
MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.
Supports database size up to 64 TiB.
Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
Supports automated backup and point-in-time recovery.
Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.



You now need to ensure that you update the connectivity between the Application Tier and the Database Tier. To do this, go into your database, select the **RDS security groups** link, go to "inbound rules", and update them.



Now your Database Tier is ready to communicate with the Application Tier.

And that is how you make a highly available and fault-tolerant AWS three-tier Architecture.

Using the DNS name of the Load Balancer, we can test our web tier in the browser like this:

Our Web Tier is reachable to the App Tier and App Tier is reachable to Database Tier also.

```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-2-233 ~]$
[ec2-user@ip-10-0-2-233 ~]$
[ec2-user@ip-10-0-2-233 ~]$ ping 10.0.3.106
PING 10.0.3.106 (10.0.3.106) 56(84) bytes of data.
64 bytes from 10.0.3.106: icmp_seq=1 ttl=255 time=1.36 ms
64 bytes from 10.0.3.106: icmp_seq=2 ttl=255 time=0.933 ms
64 bytes from 10.0.3.106: icmp_seq=3 ttl=255 time=0.946 ms
^C
--- 10.0.3.106 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.933/1.082/1.367/0.201 ms
[ec2-user@ip-10-0-2-233 ~]$
```

CONCLUSION

Here we have done the required verifications for our Three-Tier AWS Architecture.

The three-tier architecture is a widely adopted design pattern for developing scalable and resilient applications on AWS. By separating the application into distinct tiers, including the Presentation Tier, Application Tier, and Data Tier, this architecture provides several benefits.

Firstly, the Presentation Tier handles the user interface, allowing for a seamless user experience through web-based interfaces. It utilizes load balancers to distribute incoming traffic and ensures high availability.

Secondly, the Application Tier hosts the application logic and integrates with backend services or databases. This tier allows for efficient processing of user requests and provides RESTful APIs for client-server communication.

Lastly, the Data Tier stores and manages the application's data, ensuring its durability and availability. Whether using relational or NoSQL databases, object storage, or other data storage solutions, the Data Tier plays a crucial role in data persistence and retrieval.