INTERNSHIP-GIFTOS HORIZON WEBSITE USING ELB, CLOUD FRONT

A report submitted in partial fulfillment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND BUSINESS SYSTEMS

By

BOKKA VENKATA SRI KARTHIK

Regd. No: 21B91A5707

Under Supervision of Mr. GURU SANTHOSH

Blackbuck Engineers Pvt.Ltd

(Duration: 5th July 2023 to 5th September 2023)

DEPARTMENT OF COMPUTER SCIENCE AND BUSINESS SYSTEMS

S.R.K.R. ENGINEERING COLLEGE

(Autonomous)

SRKR MARG, CHINNA AMIRAM, BHIMAVARAM-534204, A.P

(Recognized by A.I.C.T.E New Delhi) (Accredited by NBA & NAAC)

(Affiliated to JNTU, KAKINADA)

SAGI RAMA KRISHNAM RAJU ENGINEERING COLLEGE

(Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the Summer Internship Report titled "**GIFTOS HORIZON WEBSITE USING ELB, CLOUD FRONT"** is the bonafide work done by **Mr. Bokka Venkata Sri Karthik (Reg no. 21B91A5707**) at the end of second year second semester at **Blackbuck Engineers Pvt.Ltd** from 5th July 2023 to 5th September 2023 in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in **Computer Science and Business Systems.**

Department Internship Coordinator **Dean -T & P Cell** **Head of the Department**

BLACKBUCK ENGINEERS

TAP TAP

Internship Certificate

Certificate ID: BB231NT00667 **Issued Date: 9th September 2023**

To Whom It May Concern

This is to certify that **BOKKA VENKATA SRI KARTHIK** bearing **Reg No: 21B91A5707** has successfully completed an internship at **Blackbuck Engineers Pvt Ltd** Hyderabad from **5th July 2023 to 5th September 2023.**

He/She has worked on a project titled **GIFTOS HORIZON WEBSITE USING ELB, CLOUD FRONT** by learning and incorporating Amazon Web Services concepts under the supervision of our project mentor.

We found that he/she is sincere, hardworking, technically sound and result oriented. He/She worked well as part of a team during his tenure.

We wish him/her all the best for his/her future endeavors.

Best regards,

Kathyayani

Kathyayani. R

Project Head

Mounika

Mounika Bezawada

HR Manager

ABSTARCT

This project, Giftos Web Hosting using ELB and CloudFront Access, aims to design, develop, and deploy a resilient and scalable web hosting service on the AWS cloud platform. By leveraging the robust features of AWS Elastic Load Balancing (ELB) and CloudFront, we propose to create an architecture that allows for high availability, fault tolerance, and rapid content delivery.

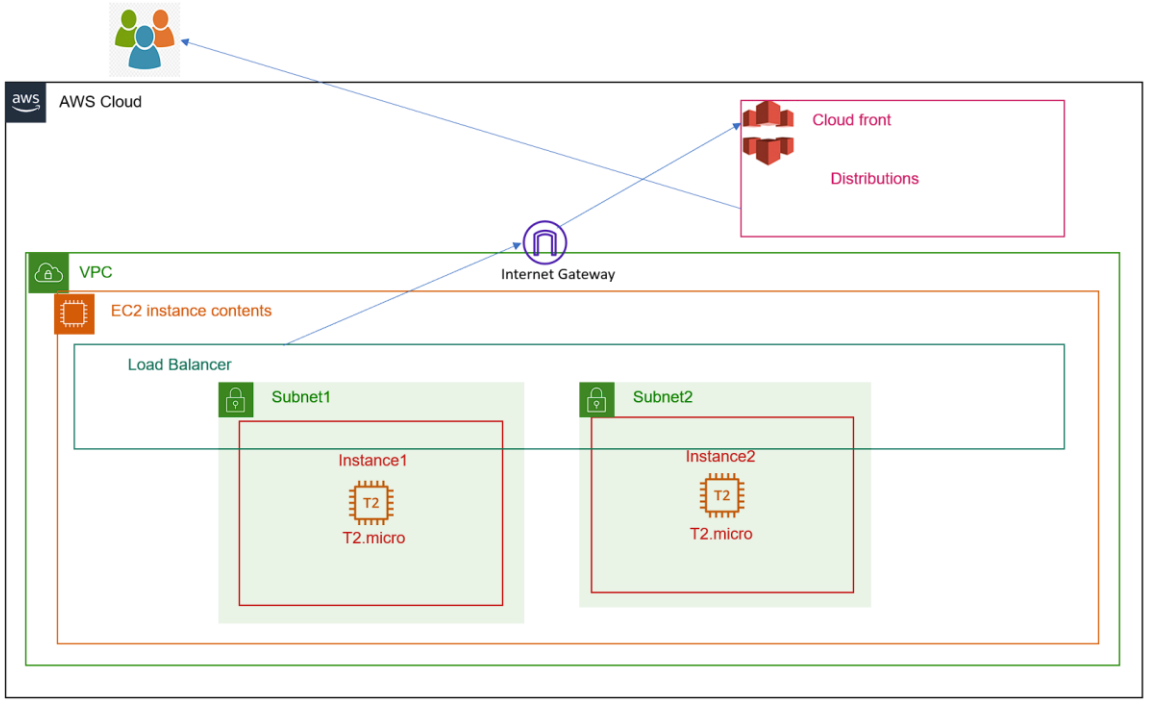
ELB will be used to distribute incoming application traffic across multiple Amazon EC2 instances, ensuring that our application performs optimally under different load conditions. This traffic distribution strategy will improve both the application's availability and its capability to handle varying workloads, reducing latency and the possibility of overloading a single server.

Table of Contents

Services used:

- Identity Access Management (IAM)
- Elastic compute cloud (EC2)
- Cloud Front

Architecture:



Description:

The architecture of a web service load balancer is designed to efficiently distribute incoming network traffic across multiple backend servers or instances hosting the web service. This architecture aims to enhance performance, scalability, and availability, ensuring that the web service can handle a large number of concurrent users and traffic spikes without overloading any single server. At the forefront of the architecture is the frontend load balancer. It acts as the entry point for incoming client requests. Clients send their requests to the load balancer, which then determines the appropriate backend server to handle the request. Behind the frontend load balancer, there are multiple backend servers or instances. These are the actual servers that host the web service and process client requests.

Cloud computing:

Cloud computing is on-demand access, via the internet, to computing resources—applications, servers (physical servers and virtual servers), data storage, development tools, networking capabilities, and more—hosted at a remote data center managed by a cloud services provider (or CSP). The CSP makes these resources available for a monthly subscription fee or bills them according to usage

Compared to traditional on-premises IT, and depending on the cloud services you select, cloud computing helps do the following:

- **Lower IT costs:** Cloud lets you offload some or most of the costs and effort of purchasing, installing, configuring, and managing your own on-premises infrastructure.

- **Improve agility and time-to-value:** With cloud, your organization can start using enterprise applications in minutes, instead of waiting weeks or months for IT to respond to a request, purchase and configure supporting hardware, and install software. Cloud also lets you empower certain users—specifically developers and data scientists.

- **Scale more easily and cost-effectively:** Cloud provides elasticity—instead of purchasing excess capacity that sits unused during slow periods, you can scale capacity up and down in response to spikes and dips in traffic. You can also take advantage of your cloud provider's global network to spread your applications closer to users around the world.

The term 'cloud computing' also refers to the technology that makes cloud work. This includes some form of virtualized IT infrastructure—servers, operating system software, networking, and other infrastructure that's abstracted, using special software, so that it can be pooled and divided irrespective of physical hardware boundaries. For example, a single hardware server can be divided into multiple virtual servers.

Cloud Computing Services:

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

are the three most common models of cloud services, and it's not uncommon for an organization to use all three.

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 amuch 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 data center. 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.

Cloud Service Providers:

- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform
- Oracle
- IBM cloud
- Salesforce

Amazon Web Services:

Amazon Web Services, Inc. (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered, pay- as-you-go basis. Oftentimes, clients will use this in combination with autoscaling (a process that allows a client to use more computing in times of high application usage, and then scale down to reduce costs when there is less traffic). These cloud computing web services provide various services related to networking, computing, storage, middleware, IoT and other processing capacity, as well as software tools via AWS server farms. This frees clients from managing, scaling, and patching hardware, and operating systems

One of the foundational services is Amazon Elastic Compute Cloud (EC2), which allows users to have at their disposal a virtual cluster of computers, with extremely high availability, which can be interacted with over the internet via REST APIs, a CLI or the AWS console. AWS's virtual computers emulate most of the attributes of a real computer, including hardware central processing units (CPUs) and graphics processing units (GPUs) for processing; local/RAM memory; hard disk /SSD storage; a choice of operating systems; networking; and pre-loaded application software such as web servers, databases, and customer relationship management (CRM).

AWS services are delivered to customers via a network of AWS server farms located throughout the world. Fees are based on a combination of usage (known as a "Pay-as-you-go" model), hardware, operating system, software, or networking features chosen by the subscriber required availability, redundancy, security, and service options. Subscribers can pay for a single virtual AWS computer, a dedicated physical computer, or clusters of either.

Amazon provides select portions of security for subscribers (e.g., physical security of the data centers) while other aspects of security are the responsibility of the subscriber (e.g., account management, vulnerability scanning, patching). AWS operates for many global geographical regions including seven in North America.

Amazon markets AWS to subscribers as a way of obtaining large-scale computing capacity more quickly and cheaply than building an actual physical server farm. All services are billed based on usage, but each service measures usage in varying ways.

Why AWS?

- **Easy to use:**

  AWS is designed to allow application providers, ISVs, and vendors to host your applications quickly and securely – whether an existing application or a new SaaS-based application. You can use the AWS Management Console or well-documented web services APIs to access AWS's application hosting platform**.**

- **Flexible:**

  AWS enables you to select the operating system, programming language, web application platform, database, and other services you need. With AWS, you receive a virtual environment that lets you load the software and services your application requires. This eases the migration process for existing applications while preserving options for building new solutions.

- **Cost-effective:**

  You pay only for the compute power, storage, and other resources you use, with no long-term contracts or up-front commitments. For more information on comparing the costs of other hosting alternatives with AWS, see the AWS Economics Center.

- **Reliable:**

  With AWS, you take advantage of a scalable, reliable, and secure global computing infrastructure, the virtual backbone of Amazon.com's multi-billion-dollar online business that has been honed for over a decade**.**

- **Scalable and High performance:**

  Using AWS tools, Auto Scaling, and Elastic Load Balancing, your application can scale up or down based on demand. Backed by Amazon's massive infrastructure, you have access to compute and storage resources when you need them.

- **Secure:**

  Using AWS tools, Auto Scaling, and Elastic Load Balancing, your application can scale up or down based on demand. Backed by Amazon's massive infrastructure, you have access to compute and storage resources when you need them.

List of AWS Services:

Amazon, the preeminent cloud vendor, broke new ground by establishing the first cloud computing service, Amazon EC2, in 2008. AWS offers more solutions and features than any other provider and has free tiers with access to the AWS Console, where users can centrally control their ministrations.
Designed around ease-of-use for various skill sets, AWS is tailored for those unaccustomed to software development utilities. Web applications can be deployed in minutes with AWS facilities, without provisioning servers or writing additional code.

- Amazon EC2 (Elastic Compute Cloud)
- Amazon RDS (Relational Database Services)
- Amazon S3 (Simple Storage Service)
- Amazon Lambda
- Amazon Cognito
- Amazon Glacier
- Amazon SNS (Simple Notification Service)
- Amazon VPC (Virtual Private Cloud)
- Amazon Lightsail
- Amazon CloudWatch
- Amazon Cloud9
- Amazon Elastic Beanstalk
- Amazon CodeCommit
- Amazon IAM (Identity and Access Management)
- Amazon Inspector
- Amazon Kinesis
- Amazon Dynamo DB
- Amazon Codecatalyst
- Amazon Kinesis
- AWS Athena
- AWS Amplify
- AWS Quicksight
- AWS Cloudformation

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.

Instance types:

Initially, EC2 used Xen virtualization exclusively. However, on November 6, 2017, Amazon announced the new C5 family of instances that were based on a custom architecture around the KVM hypervisor, called Nitro. Each virtual machine, called an "instance", functions as a virtual private server. Amazon sizes instances based on "Elastic Compute Units". The performance of otherwise identical virtual machines may vary. On November 28, 2017, AWS announced a bare-metal instance type offering marking a remarkable departure from exclusively offering virtualized instance types.

As of January 2019, the following instance types were offered:

- General Purpose: A1, T3, T2, M5, M5a, M4, T3a
- Compute Optimized: C5, C5n, C4
- Memory Optimized: R5, R5a, R4, X1e, X1, High Memory, z1d
- Accelerated Computing: P3, P2, G3, F1
- Storage Optimized: H1, I3, D2

As of April 2018, the following payment methods by instance were offered:

- On-demand: pay by the hour without commitment.
- Reserved: rent instances with one-time payment receiving discounts on the hourly charge.

- Spot: bid-based service runs the jobs only if the spot price is below the bid specified by bidder. The spot price is claimed to be supply-demand based, however a 2011 study concluded that the price was generally not set to clear the market but was dominated by an undisclosed reserve price.

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[citation needed].

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 blockstorage 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 subnetswithin that VPC are regional resources. This allows users to connect zones and regions withoutthe 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 centers.

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.

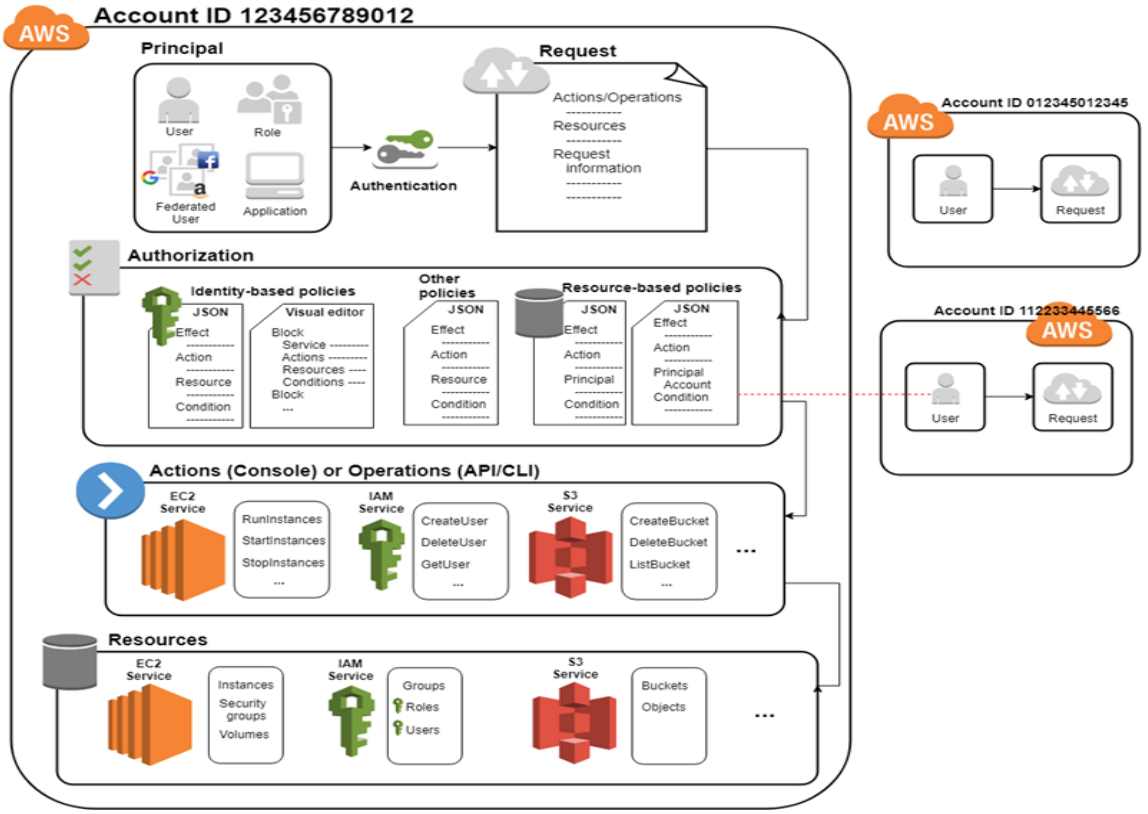
Objects can be up to five terabytes in size. Requests are authorized using an access control listassociated with each object bucket and support versioning which is disabled by default. Since buckets are typically the size of an entire file system mount in other systems, this access controlscheme is very 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, validfor 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 retrievethe 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 indata 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 systemmay not behave entirely as expected.

Amazon IAM:

IAM provides the infrastructure necessary to control authentication and authorizationfor your AWS account. The IAM infrastructure is illustrated by the following diagram.



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 responseto 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, andany 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 AWSaccount. 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 toresources. 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 apolicy 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 issupply 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 customizethe operating system on provided runtimes.

Lambda performs operational and administrative activities on your behalf, including managingcapacity, 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.

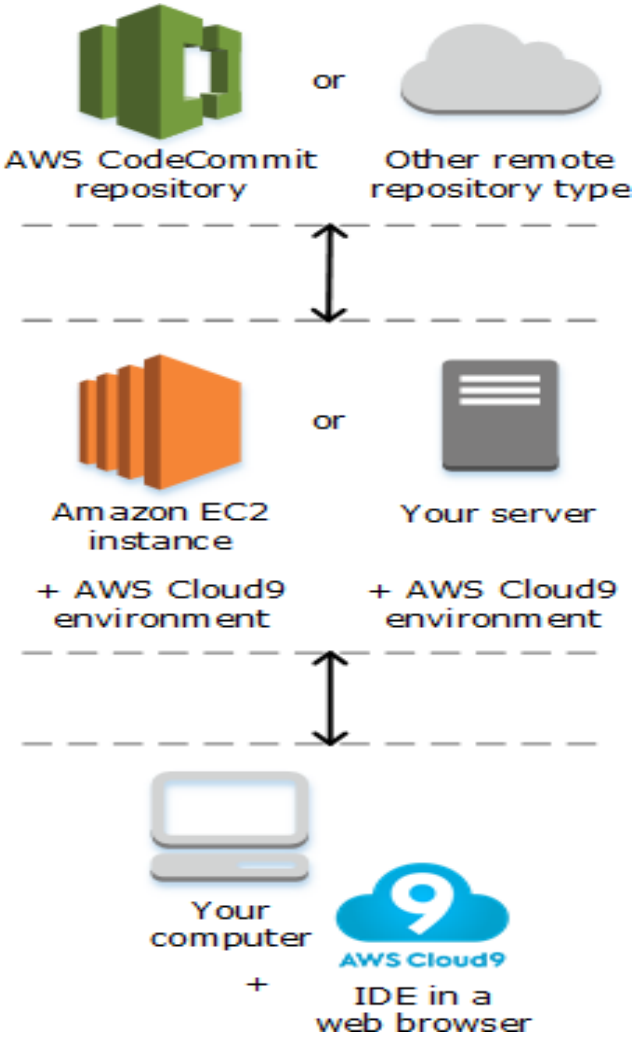
You can use environment variables to adjust your function's behavior without updating code. An environment variable is a pair of strings that is stored in a function's version-specific configuration. The Lambda runtime makes environment variables available to your code and sets additional environment 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 collectionof tools that you use to code, build, run, test, and debug software, and helps you release softwareto the cloud.

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



Environments and computing resources

Behind the scenes, there are a couple of ways you can connect your environments to computingresources:

- You can instruct AWS Cloud9 to create an Amazon EC2 instance, and then connect the environment to that newly created EC2 instance. This type of setup is called an EC2 environment.
- You can instruct AWS Cloud9 to connect an environment to an existing cloud compute instance or to your own server. This type of setup is called an SSH environment.

EC2 environments and SSH environments have some similarities and some differences. If you're new to AWS Cloud9, we recommend that you use an EC2 environment because AWS Cloud9 takes care of much of the configuration for you. As you learn more about AWS Cloud9,and want to understand these similarities and differences better, see EC2 environments compared with SSH environments in AWS Cloud9.

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 andhow 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 yourapplication, and Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring.

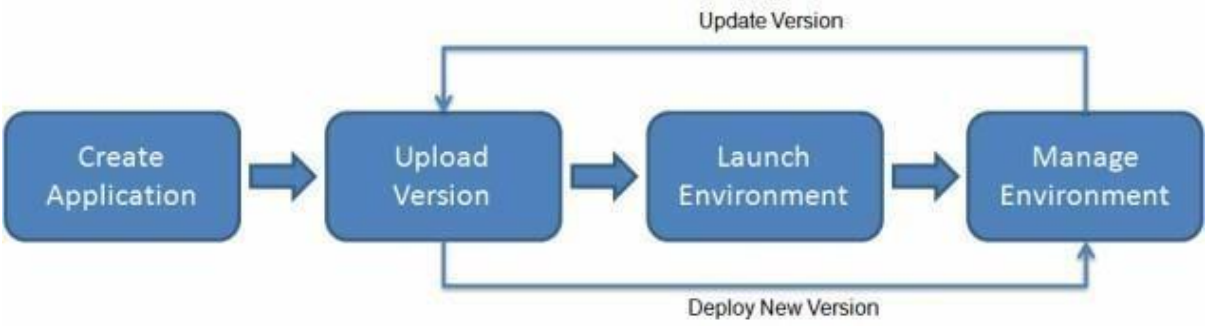
Elastic Beanstalk supports applications developed in Go, Java, .NET, Node.js, PHP, Python, and Ruby. When you deploy your application, Elastic Beanstalk builds the selected supportedplatform version and provisions one or more AWS resources, such as Amazon EC2 instances,to run your application.

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 ElasticBeanstalk.

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 webinterface (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.



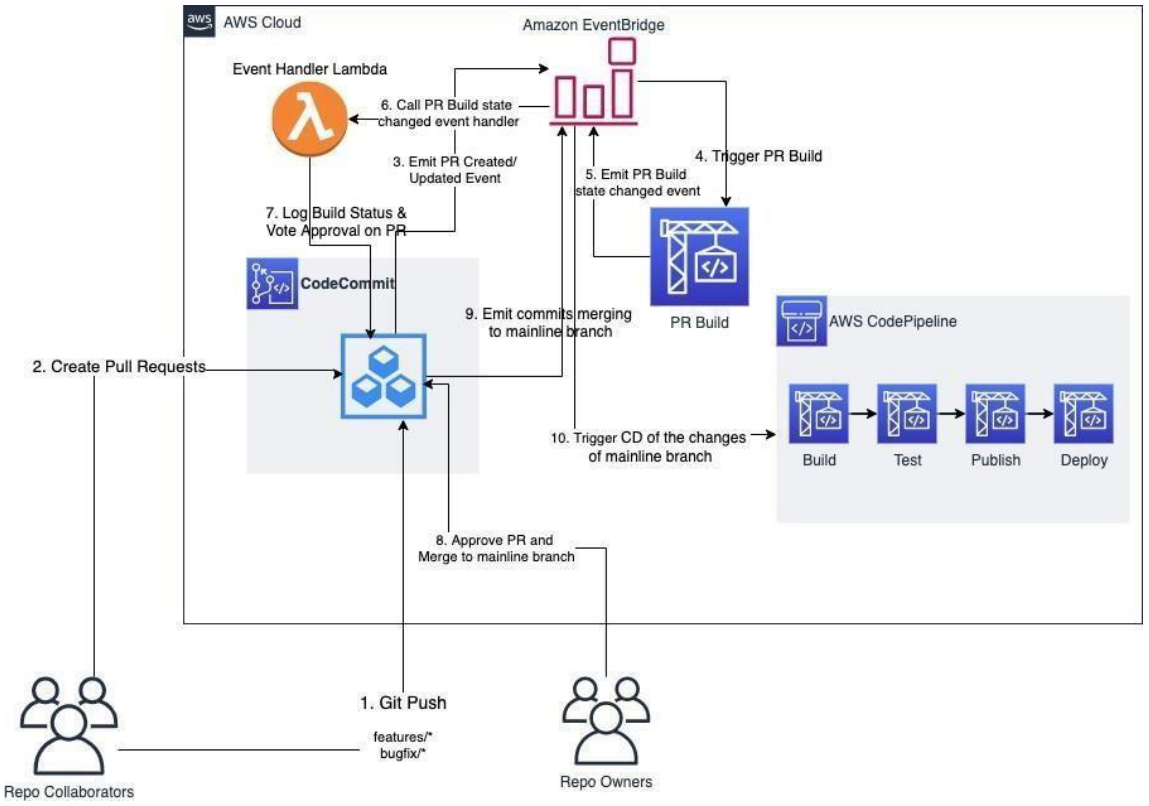
AWS CodeCommit:

CodeCommit is a secure, highly scalable, managed source control service that hosts private Gitrepositories. 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 withyour 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.
- **Use the Git tools you already know**. CodeCommit supports Git commands as well as its own AWS CLI commands and APIs.



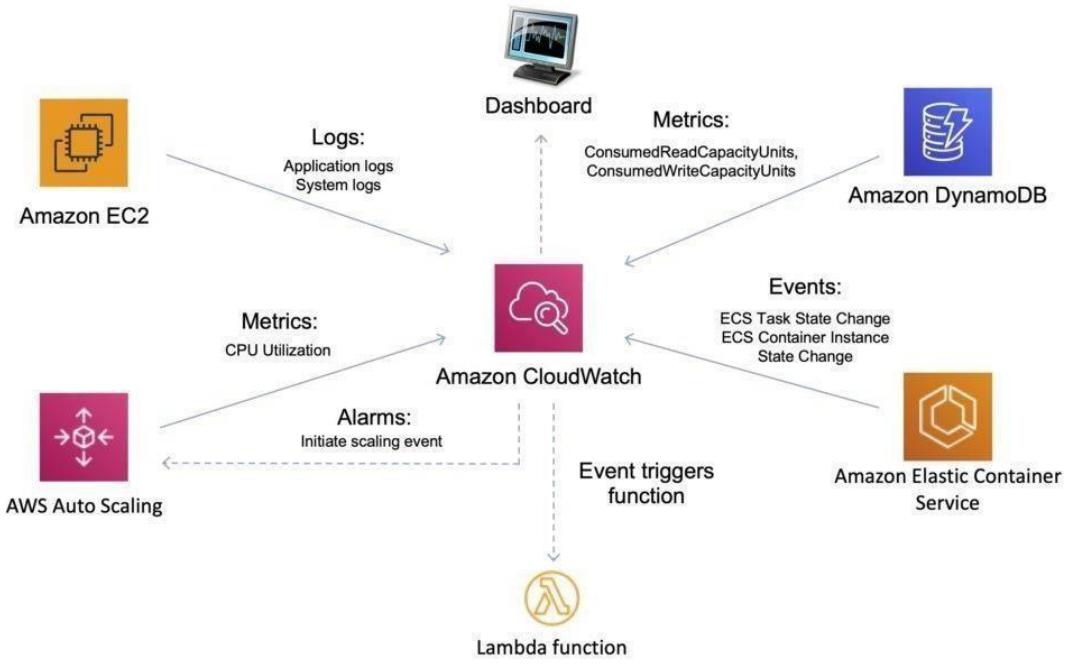
Amazon CloudWatch:

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 changesto 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 usethat 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.

With CloudWatch, you gain system-wide visibility into resource utilization, application performance, and operational health.



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 thesevolumes as devices on your instances. EBS volumes that are attached to an instance are exposedas 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 thatperform long, continuous reads and writes.

With Amazon EBS, you pay only for what you use. For more information about Amazon EBSpricing, 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 center 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 ofIOPS 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.

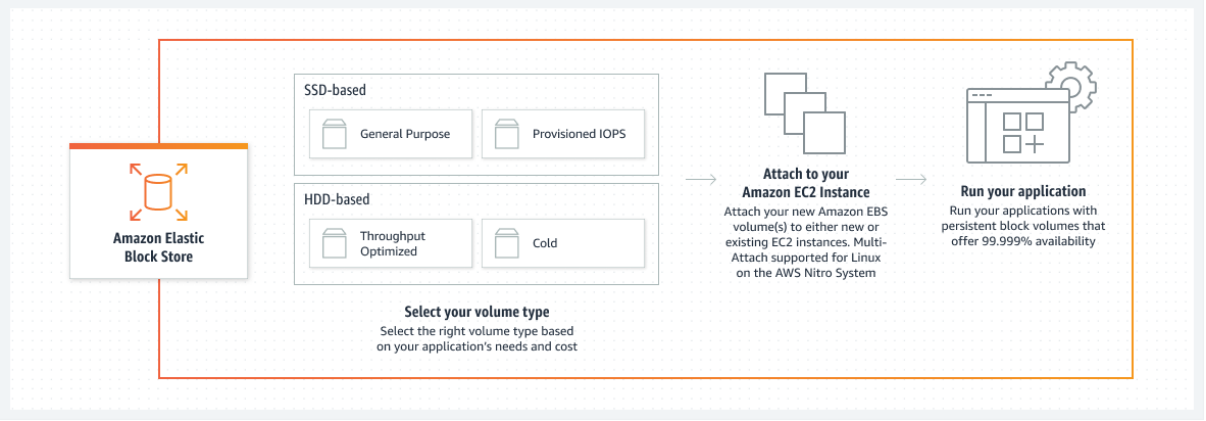


Fig. High-Performance Block Storage

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 speedand 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 toa maximum size of 128 tebibytes (TiB). Aurora also automates and standardizes database clustering and replication, which are typically among the most challenging aspects of databaseconfiguration and administration.

Aurora is part of the managed database service Amazon Relational Database Service (AmazonRDS). 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 PostgreSQLinto 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 andRDS for PostgreSQL applications to Aurora.

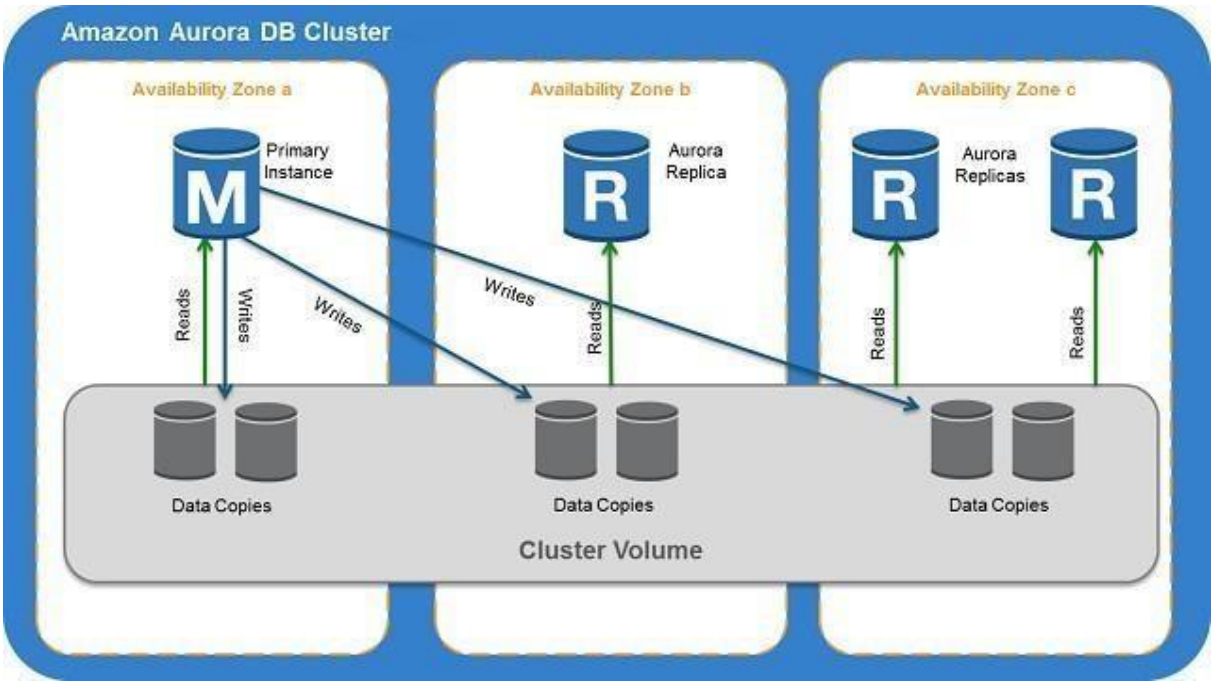
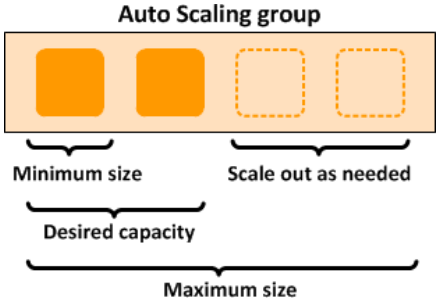


Fig. Amazon Auora DB Clusters

AWS Autoscaling:

Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2instances 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, AmazonEC2 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 yourapplication increases or decreases.

For example, the following Auto Scaling group has a minimum size of one instance, a desiredcapacity of two instances, and a maximum size of four instances. The scaling policies that youdefine adjust the number of instances, within your minimum and maximum number ofinstances, 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 anddecrease

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 w

Create an IAM user with EC2 full access and Cloud front full access

IAM Service:

AWS Identity and Access Management (IAM) is a web service provided by Amazon Web Services (AWS) that enables you to securely manage access to AWS resources. IAM allows you to control who can use which resources and what actions they can perform on those resources. It helps you set up and manage user identities, groups, and permissions within your AWS account.

IAM User:

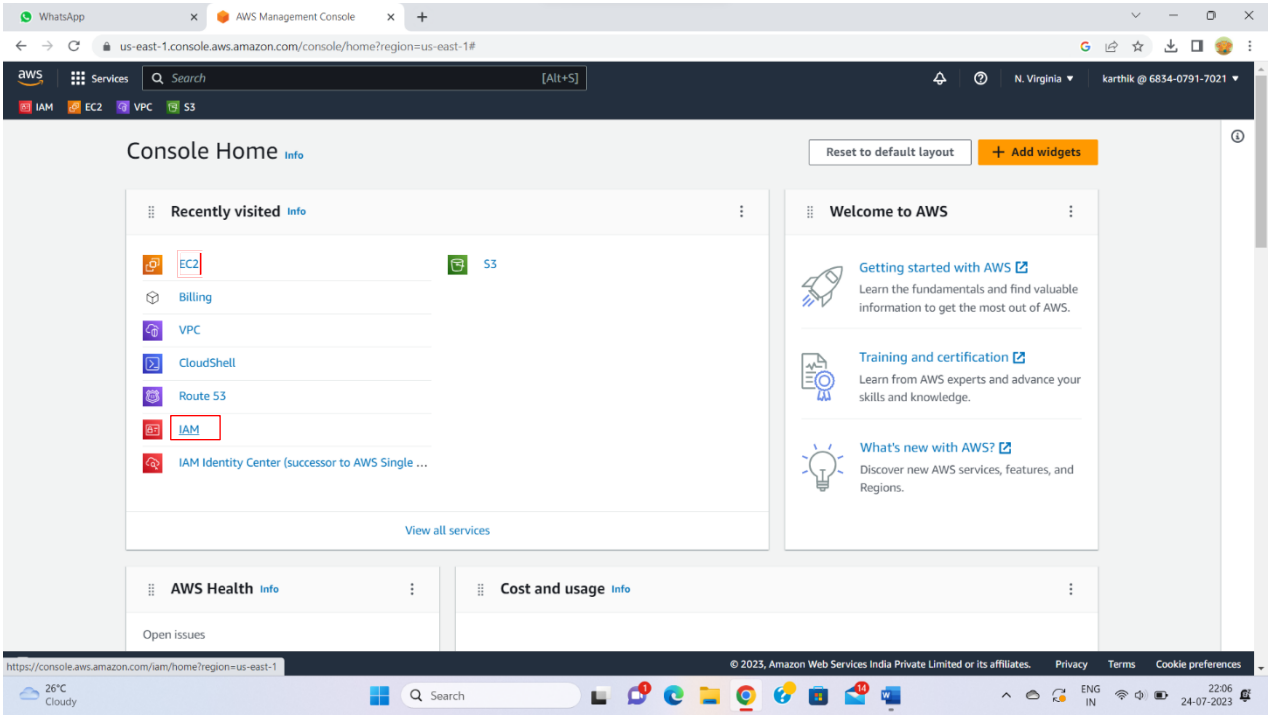
IAM user is an identity that represents a person or an application that interacts with AWS services. IAM (Identity and Access Management) is the service within AWS that enables you to manage users, groups, and permissions to access and control AWS resources securely.

IAM users are used to control access to AWS services and resources without the need to share AWS account credentials (i.e., the root user credentials). Instead, each user is provided with unique access keys (access key ID and secret access key) or can authenticate using IAM user name and password, multi-factor authentication (MFA), or temporary security tokens. By creating IAM users and assigning appropriate permissions (using IAM policies), AWS administrators can manage and limit the actions users can perform on AWS resources.

Creating IAM User:

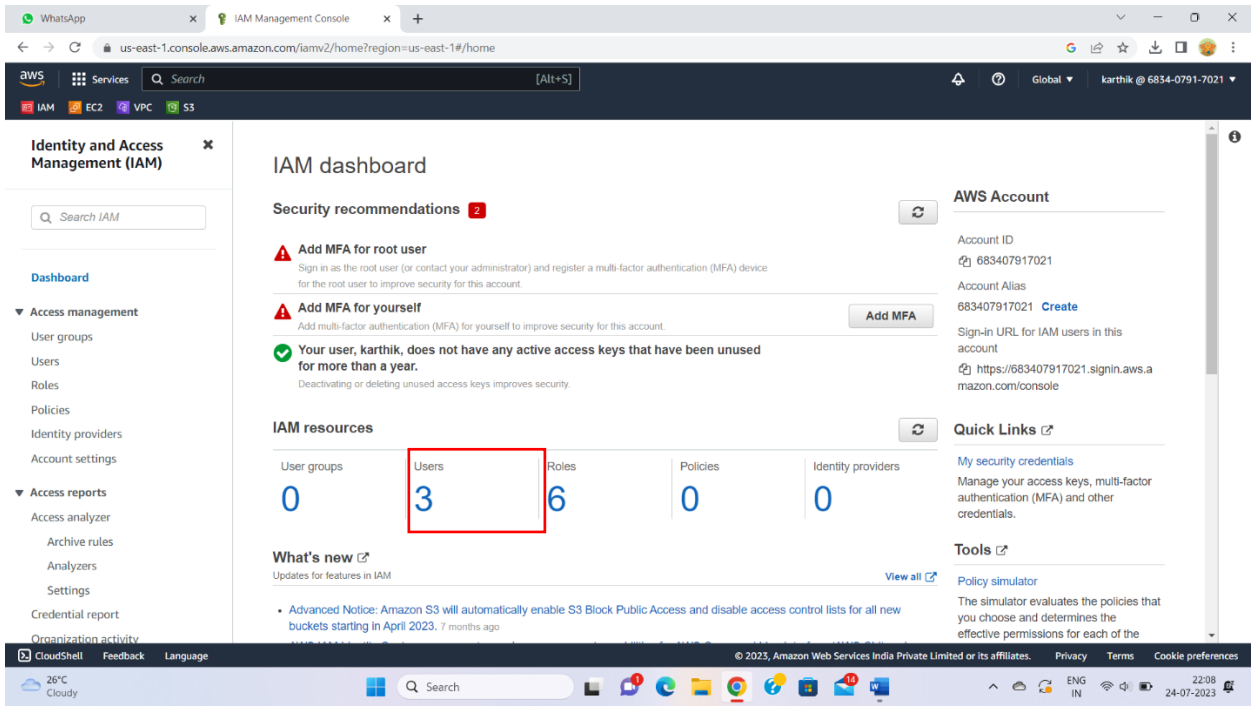
By following the below steps we can create an IAM user with ec2fullaccess and cloudfrontfullaccess.

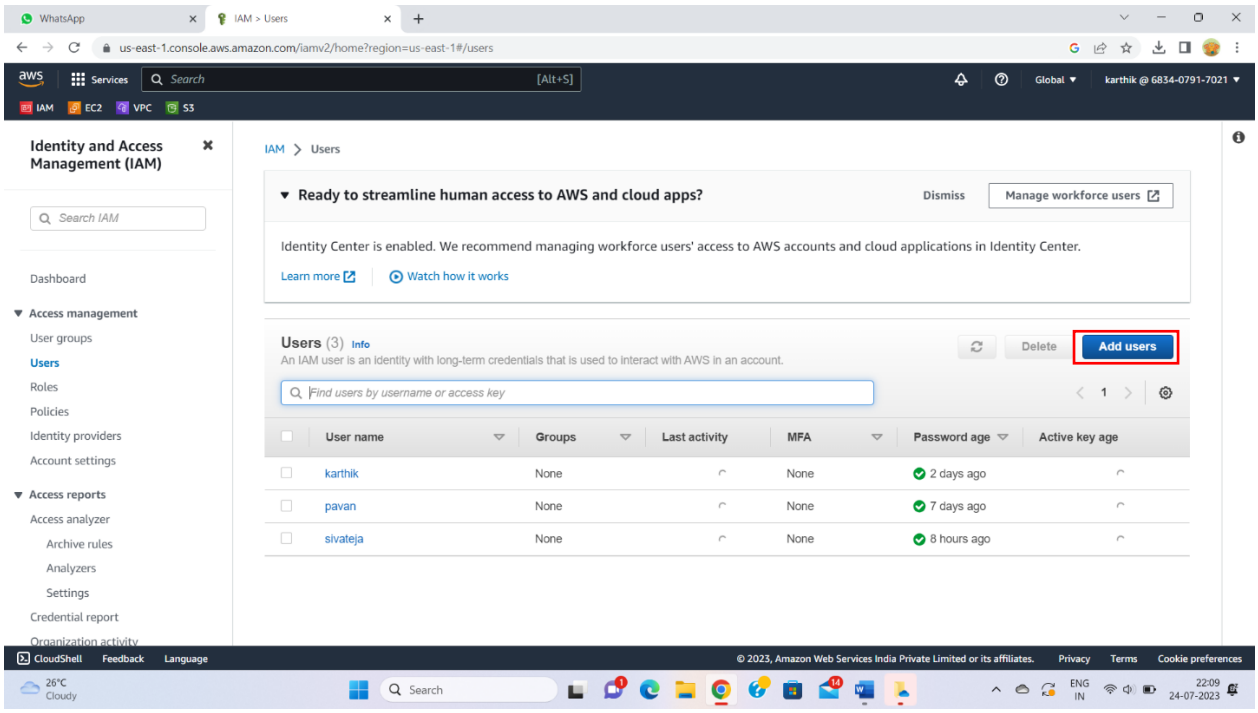
Step 1: Log in to the AWS Management Console using your administrator account and open the IAM service by searching for "IAM" in the AWS Management Console search bar and selecting the IAM result.



Step 2: In the IAM dashboard, click on "Users" in the left navigation pane. A list of already existing users will be displayed.

click on the "Add user" button to create a new IAM user.

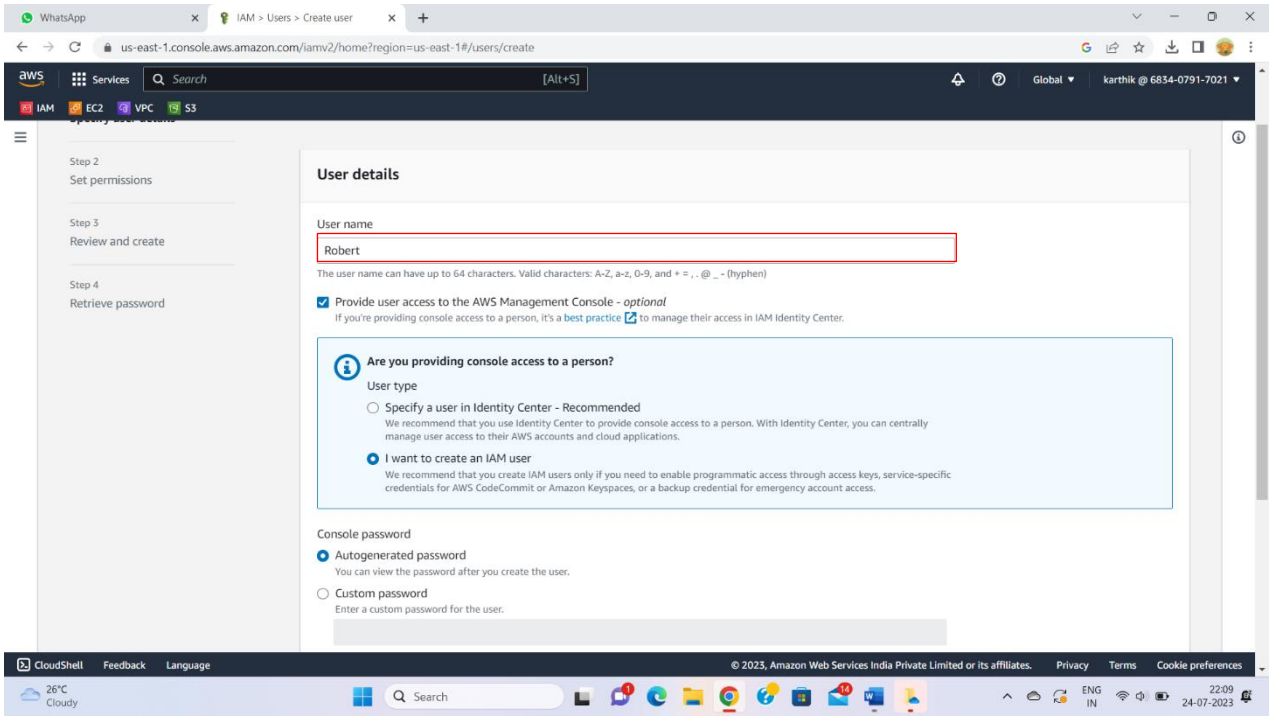




Step 3: On the Specify user details page, under User details, in User name, enter the name for the new user. This is their sign-in name for AWS

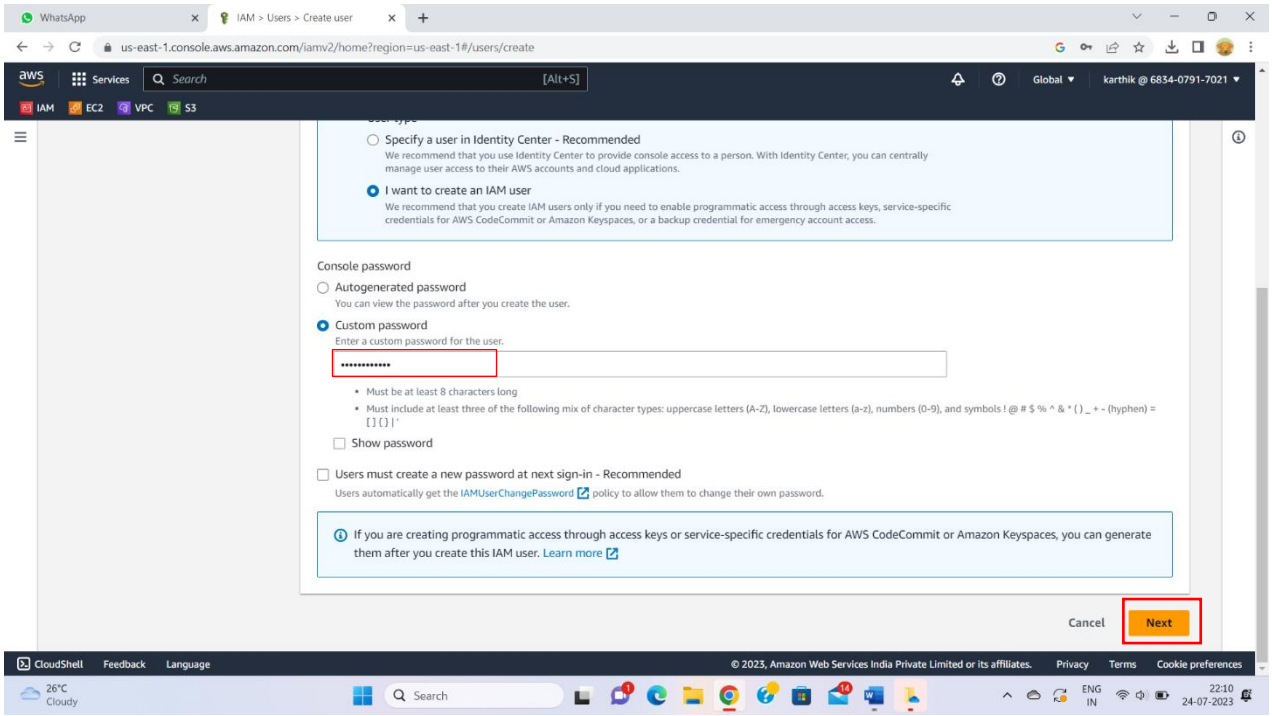
Select “Provide user access to the – AWS Management Console”. This produces AWS Management Console sign-in credentials for the new user.

You are asked whether you are providing console access to a person. Select “I want to create an IAM user”.

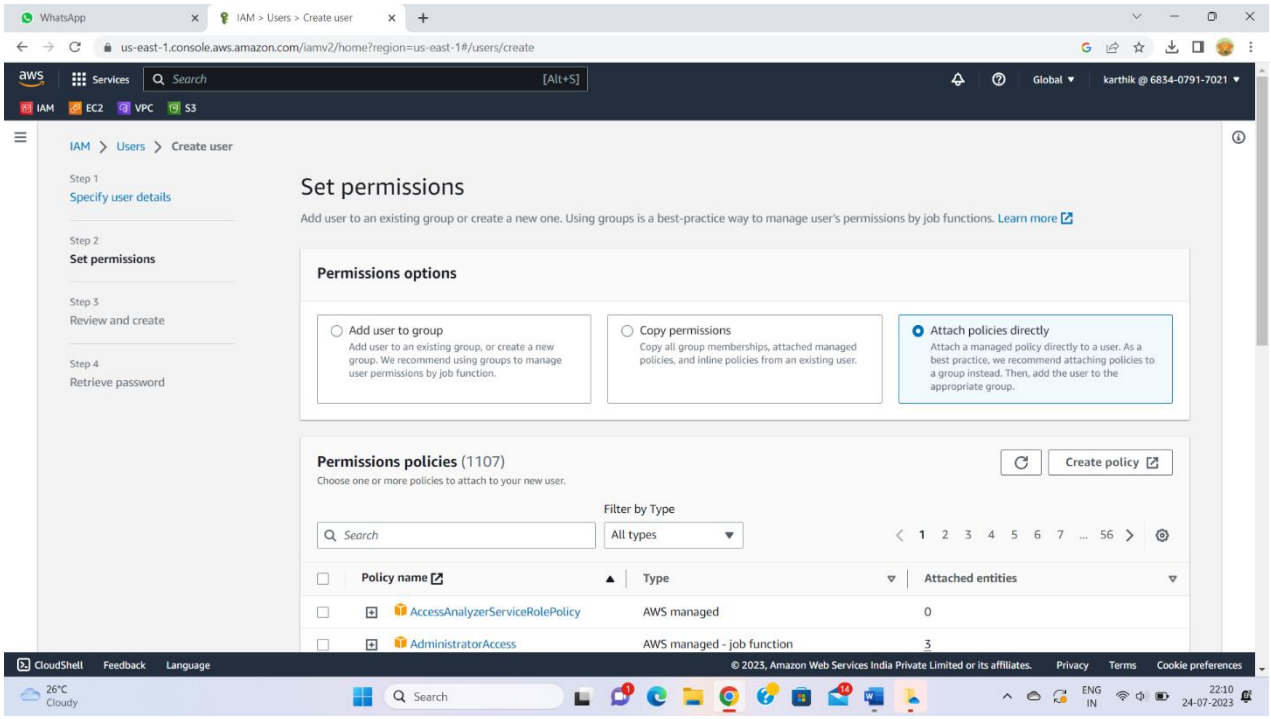


Step 4: For Console password, select "Autogenerated password" and "Users must create a new password at next sign-in". Click on next

- Autogenerated password – The user gets a randomly generated password that meets the account password policy. You can view or download the password when you get to the Retrieve password page.
- Custom password – The user is assigned the password that you enter in the box.



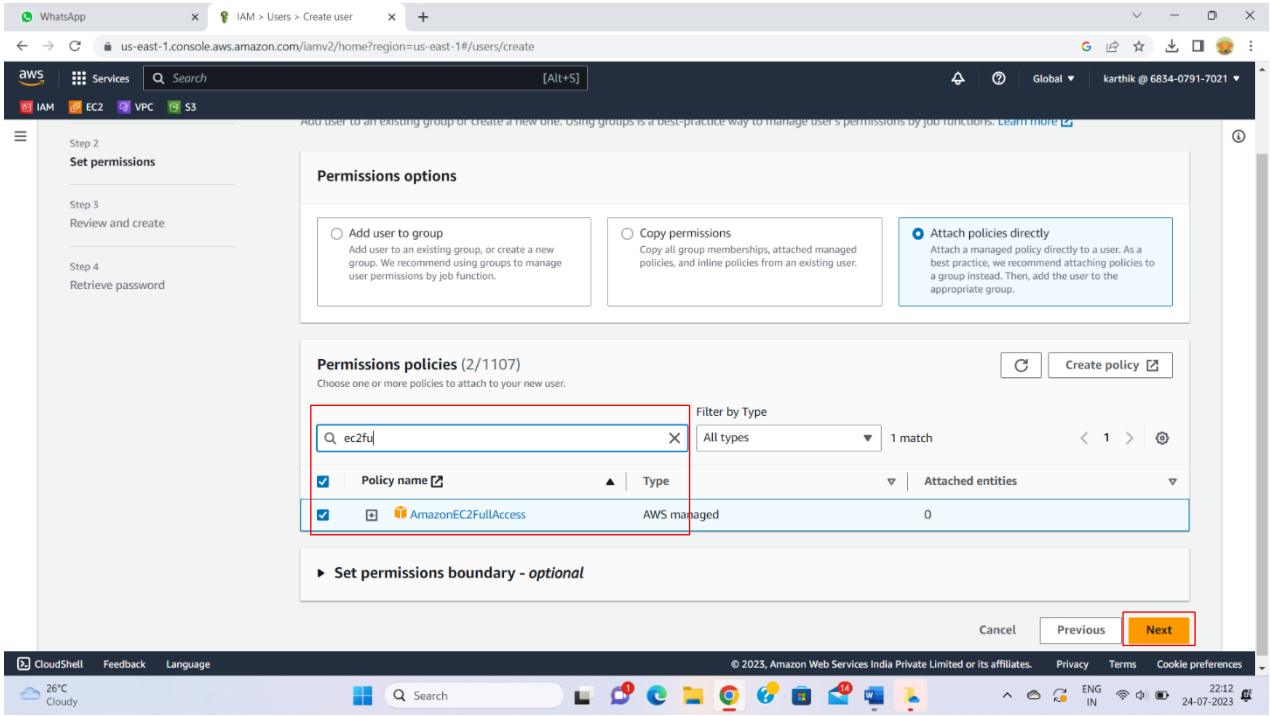
Step 5: On the Set permissions page, specify how you want to assign permissions for this user. Select "Attach policies directly" option to see a list of the AWS managed and customer managed policies in your account.



Add user to group – Select this option if you want to assign the user to one or more groups that already have permissions policies.

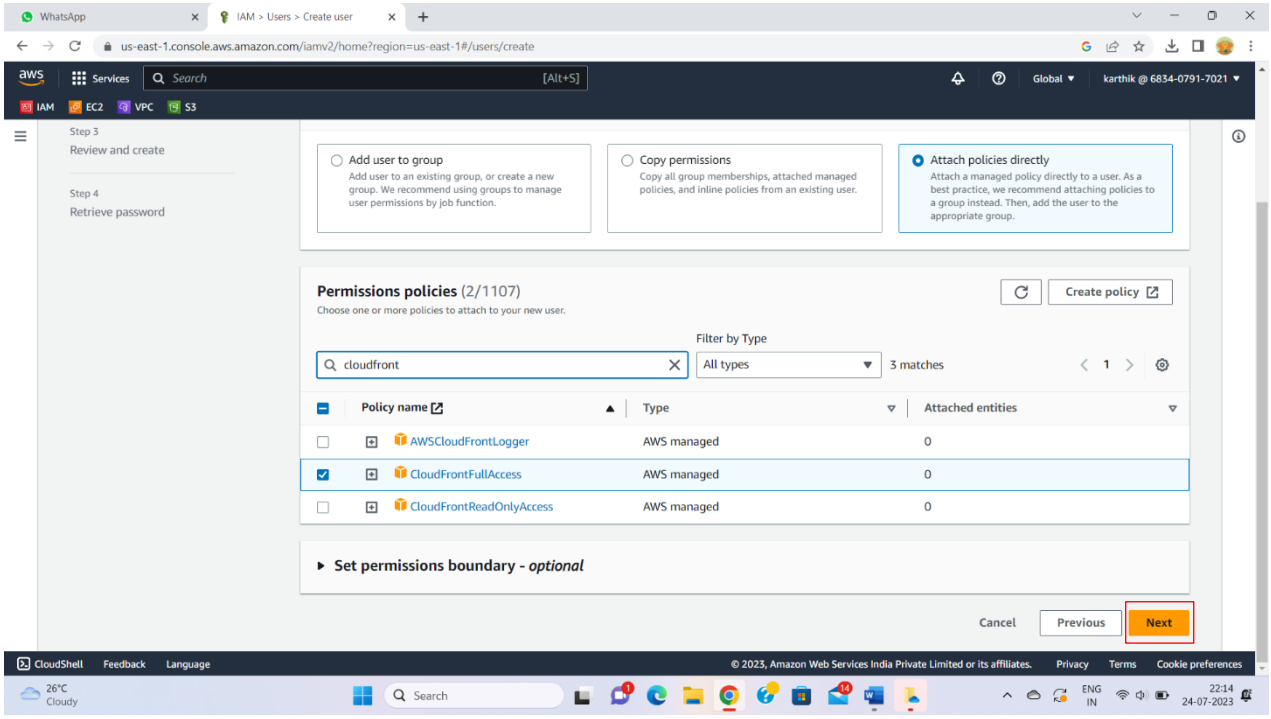
Copy permissions – Select this option to copy all the permissions from an existing user to the new user.

Step 6: In the search bar, search for "AmazonEC2FullAccess" policy and select the policy to attach it to the user.



Step 7: In the search bar, search for “CloudFrontFullAcess” policy and select the policy to attach it to the user.

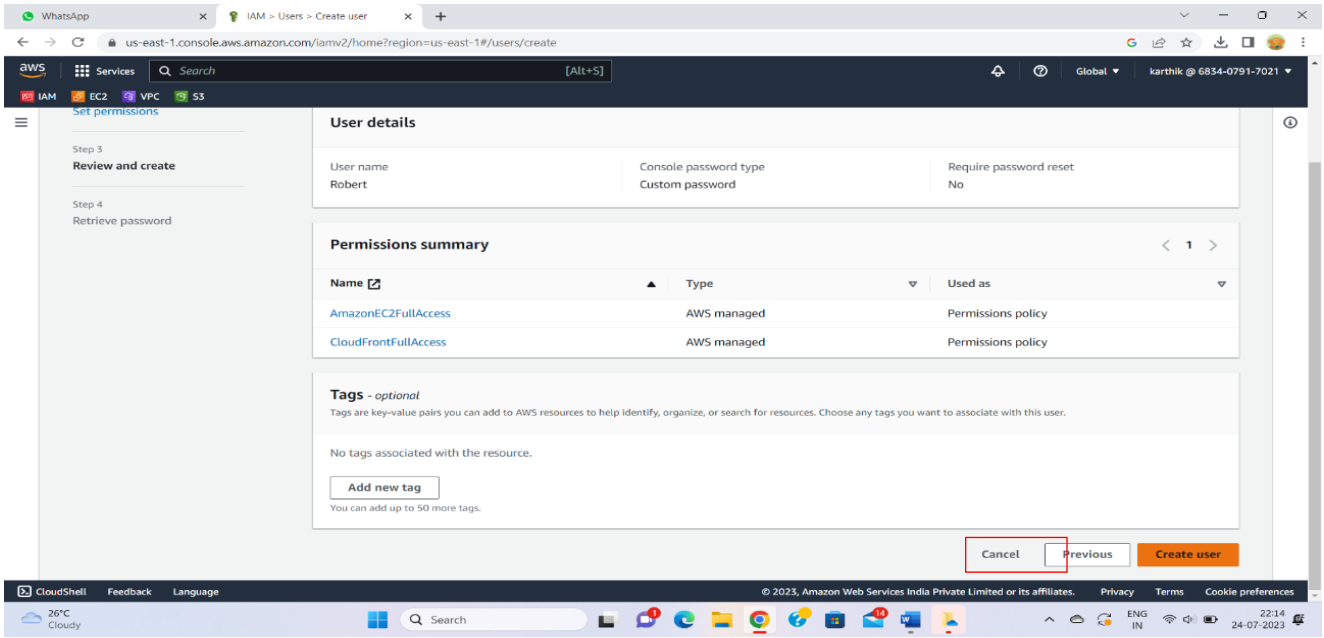
Click on next button



If you want to create a policy, select Create policy button to open a new browser tab and create a new policy.

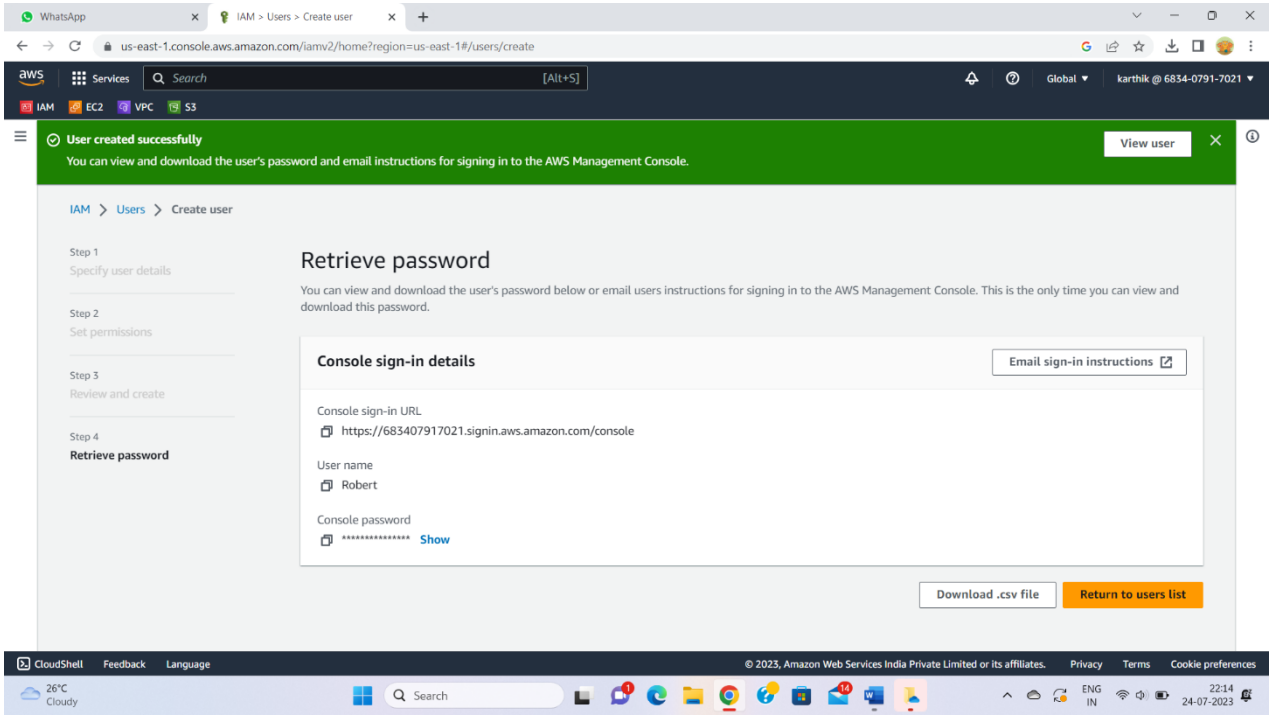
Step 8: review the permissions summary and ensure that the selected policies are included. Click on the "Create user" button to create the IAM user.

Review all of the choices you made up to this point. When you are ready to proceed, select Create user.



Step 9: On the Retrieve password page, get the password assigned to the user:

- Select Show next to the password to view the user's password so that you can record it manually.
- Select Download .csv to download the user's sign in credentials as a .csv file that you can save to a safe location.



You can select Email sign-in instructions. Your local mail client opens with a draft that you can customize and send to the user.

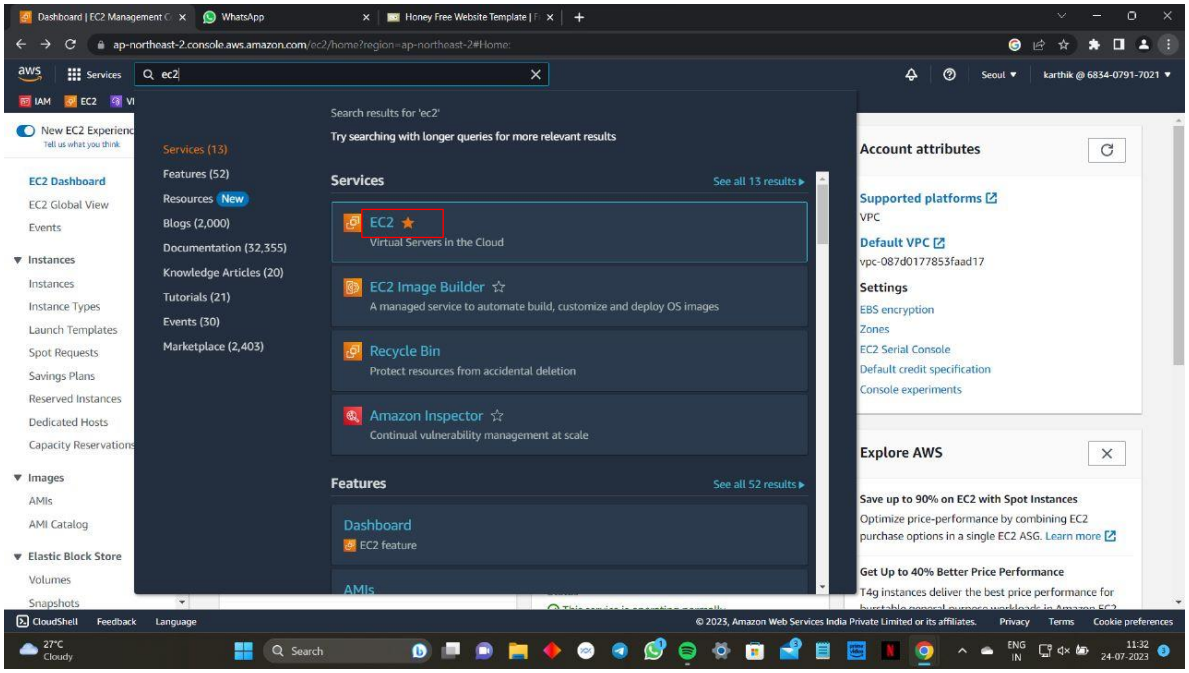
Launch an EC2 instance in two different availability zones and Configure web server in the instance

EC2 Instance:

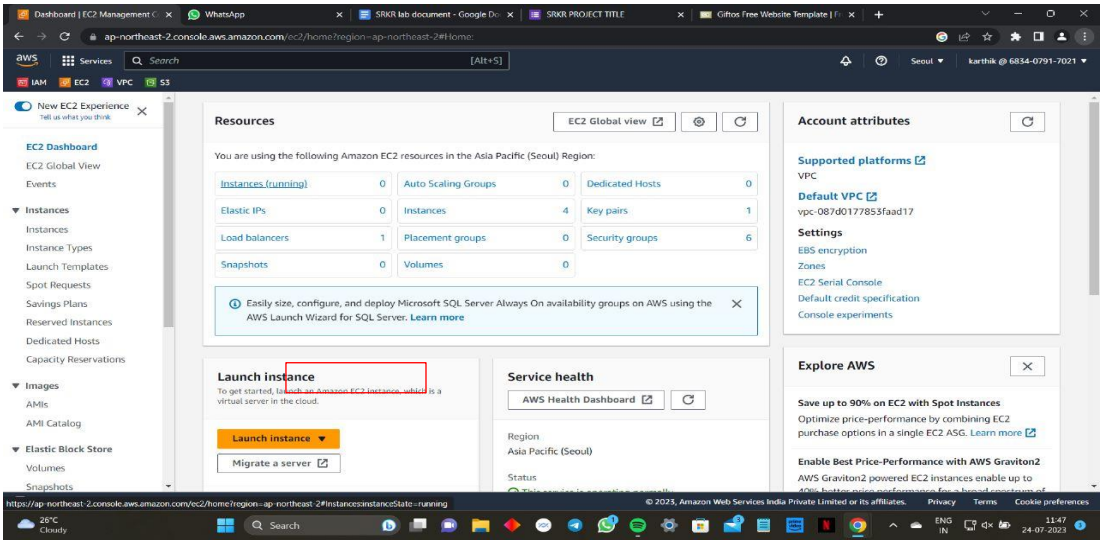
Amazon Elastic Compute Cloud (Amazon EC2) is a web service provided by Amazon Web Services (AWS) that allows users to rent virtual servers in the cloud, known as EC2 instances. EC2 instances are the fundamental building blocks of scalable, flexible, and customizable computing resources in AWS.

Launching EC2 Instances:

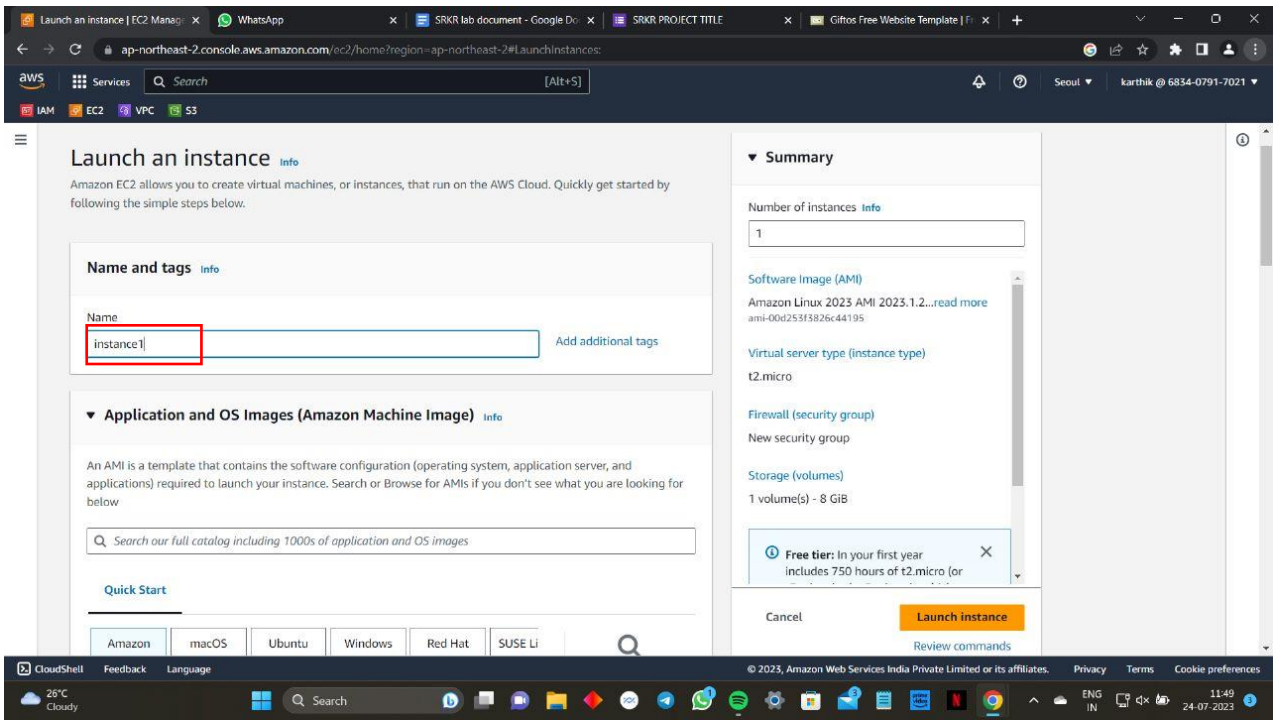
Step 1: Go to the AWS Management Console (https://aws.amazon.com/), and sign in using your credentials. Once you're logged in search for the EC2 service.



Step 2: On the EC2 Dashboard, click the "Launch Instance" button to begin the instance creation process.5



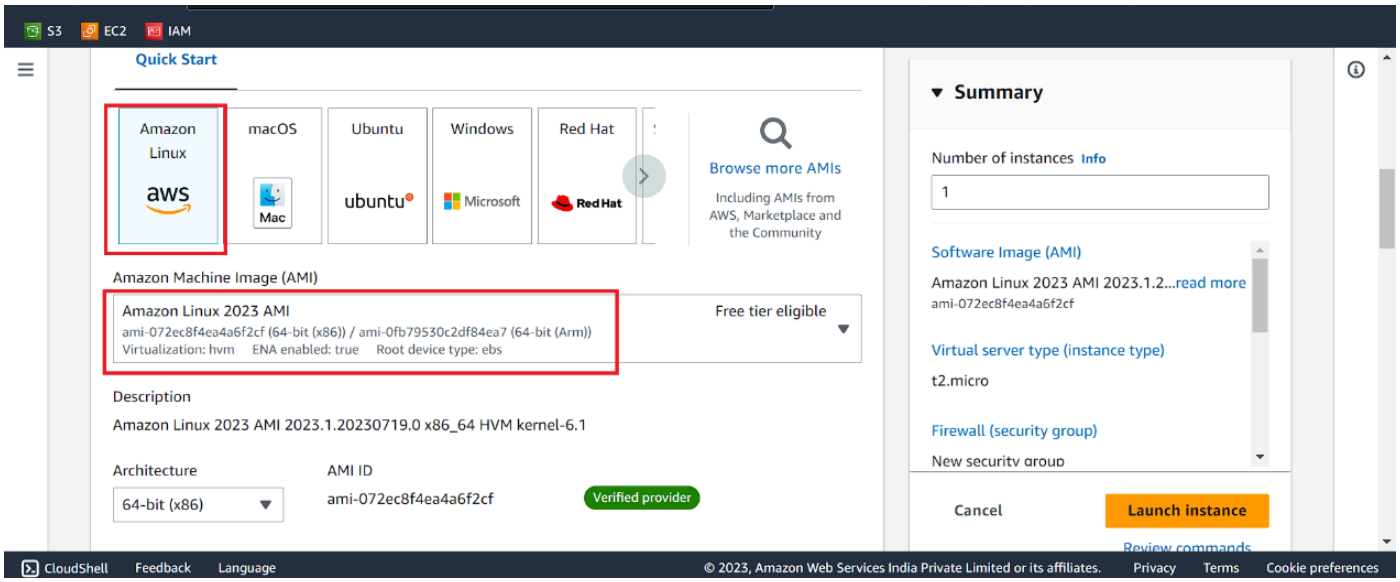
Step 3: Give name for the EC2 Instance.



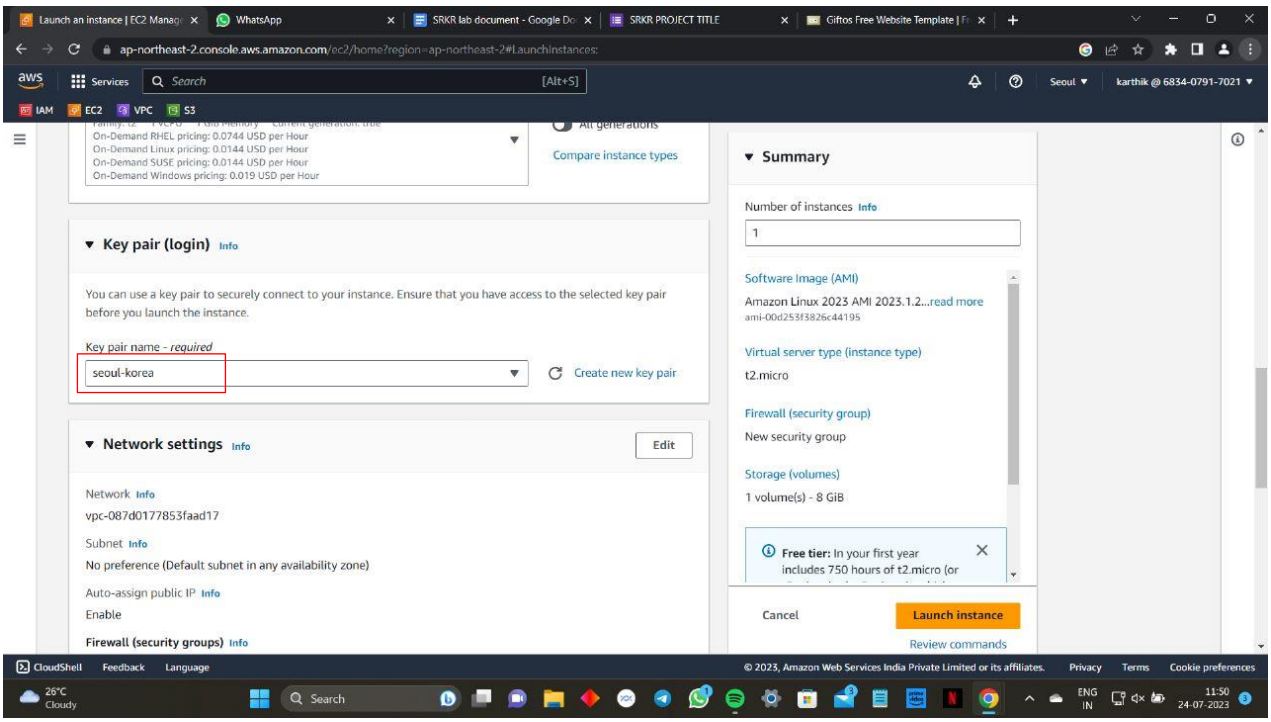
Step 4: Select the desired AMI for your EC2 instance. AMIs are pre-configured templates containing an operating system, software, and other configurations.

Choose Quick Start, and then choose Amazon Linux. This is the operating system (OS) for your instance.

From Amazon Machine Image (AMI), select Amazon Linux 2023 AMI. Notice that these AMIs are marked Free tier eligible.



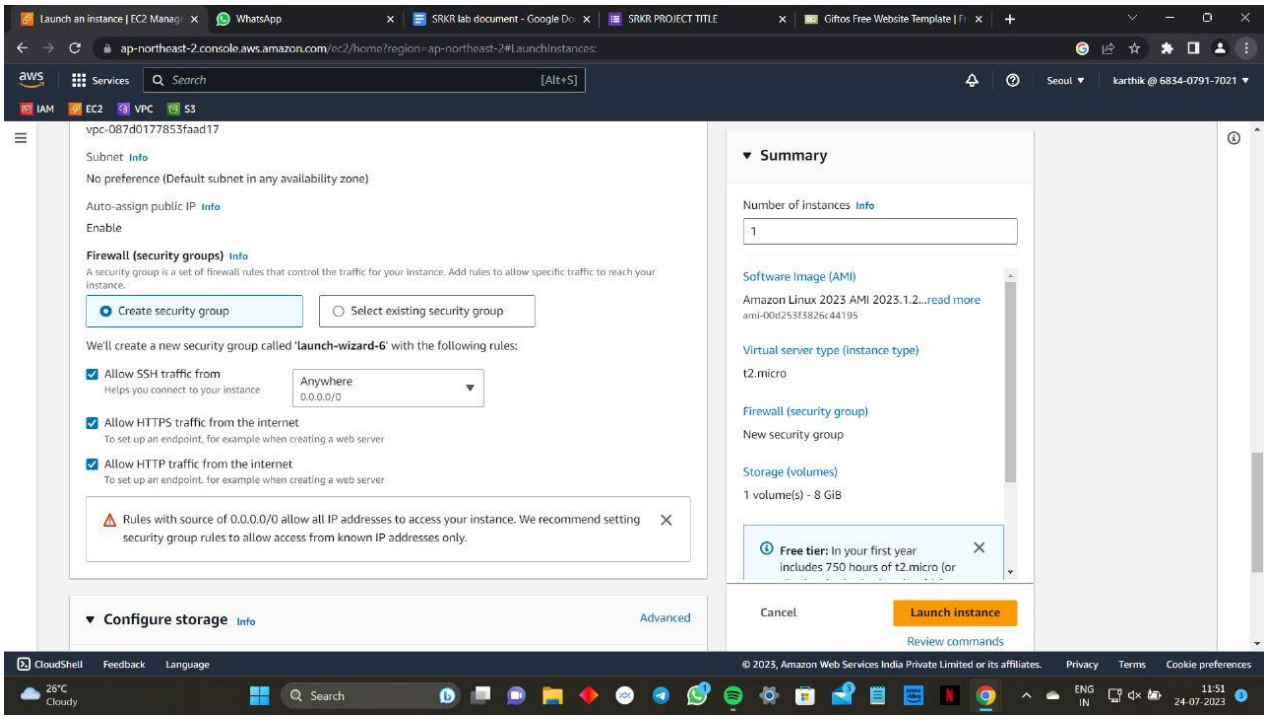
Step 5: Select the instance type that meets your requirements in terms of CPU, memory, storage, and other specifications. You'll be prompted to choose an existing key pair or create a new one. This key pair allows you to connect securely to your instance using SSH.



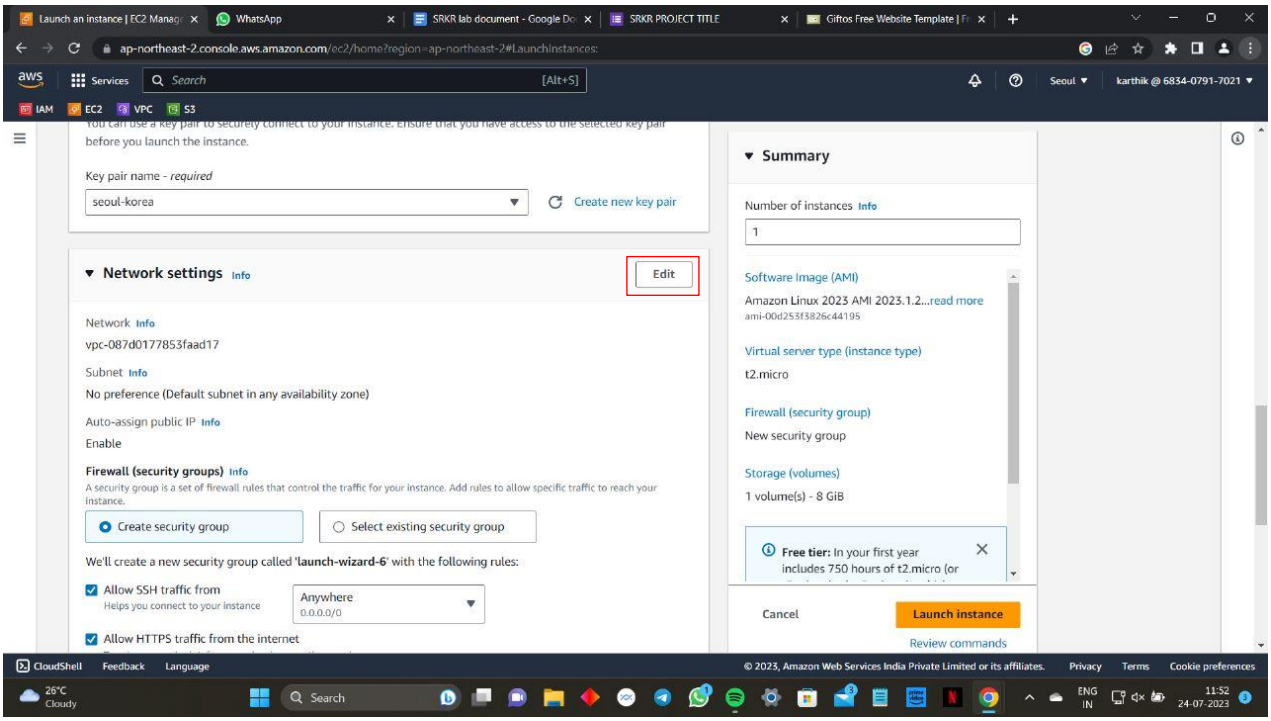
Step 6: Create a security group. Security groups act as virtual firewalls, controlling inbound and outbound traffic to your instance. Configure the necessary rules to allow access to your instance.

Select “Allow HTTPS traffic from the internet” and “Allow HTTP traffic from the internet”.

Enabling HTTP traffic (HTTP port 80) to access the website hosted on an EC2 instance is necessary to allow users to visit the website using their web browsers. When you enable HTTP traffic, you are essentially opening port 80 on the EC2 instance, which allows incoming HTTP requests from the internet.

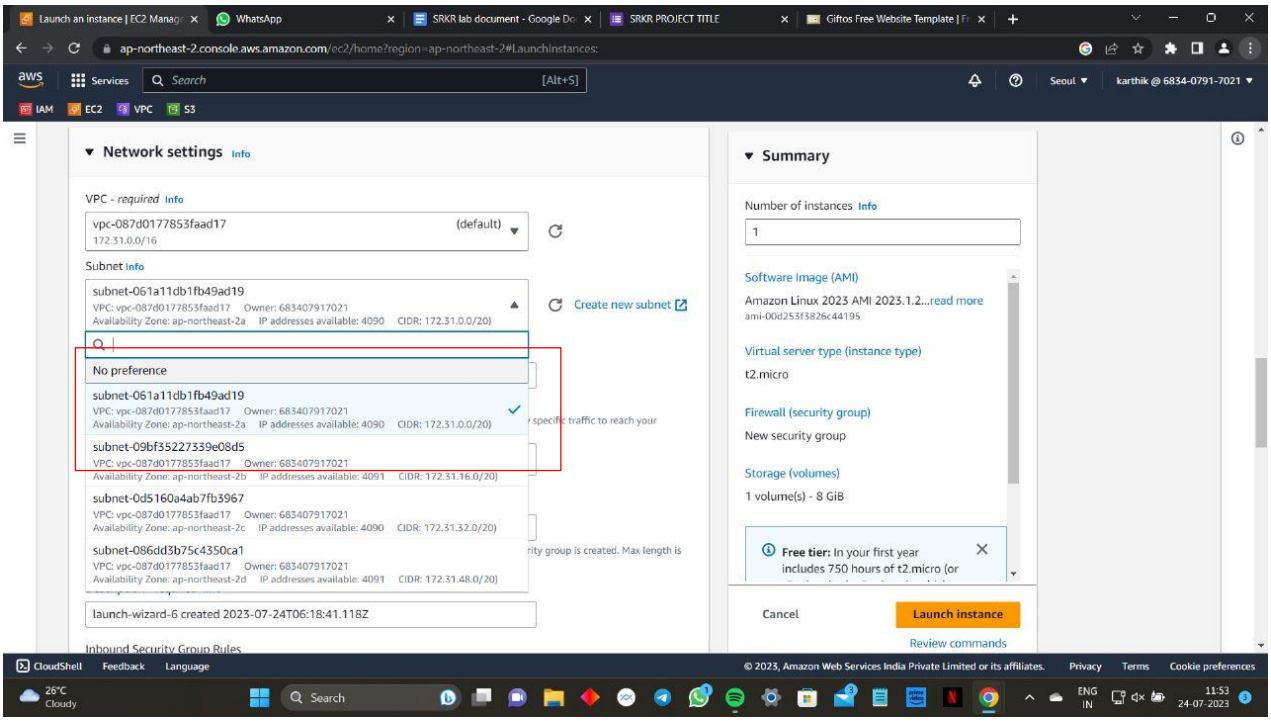


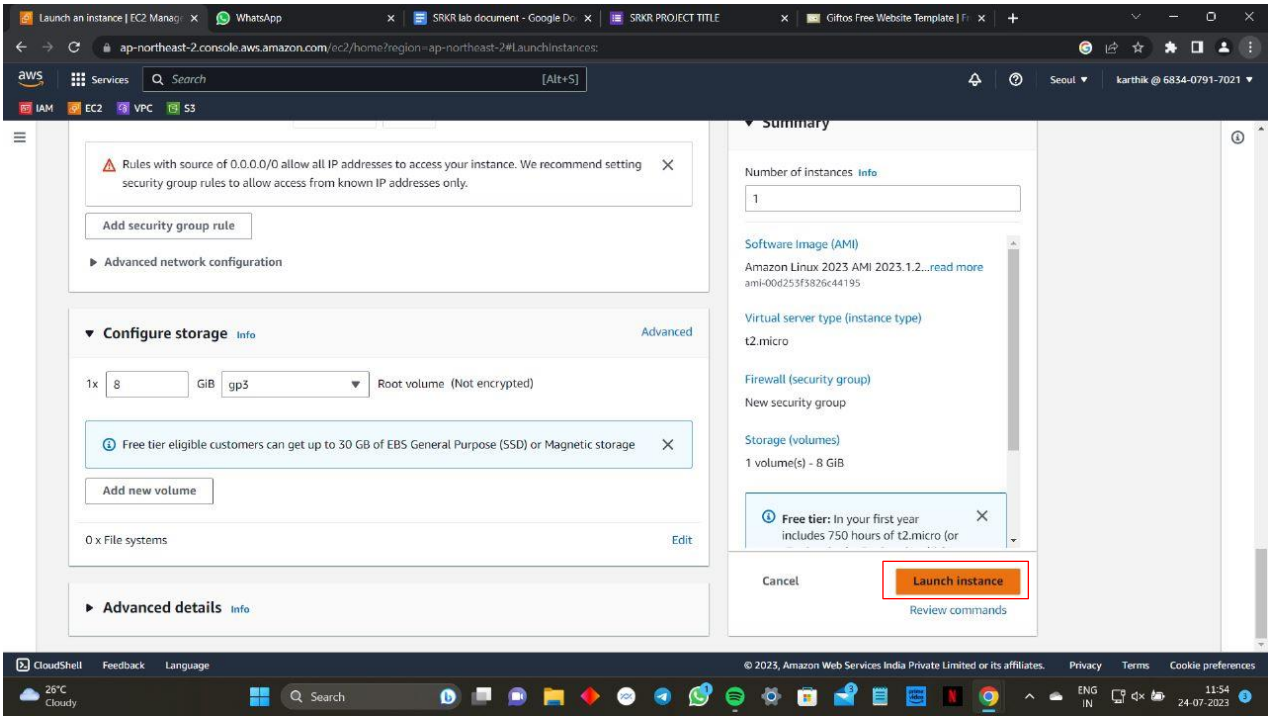
Step 7: Click on edit in network settings



Step 8: you need to select the subnet in which the EC2 instance will reside. Click on Launch Instance

Make sure to select subnet present in the different availability zone while creating the second instance.





Similarly by following the above steps launch the second Ec2 Instance in another availability zone. To launch an EC2 instance in a particular availability zone we have to select the subnet present in the respective availability zone in network settings.

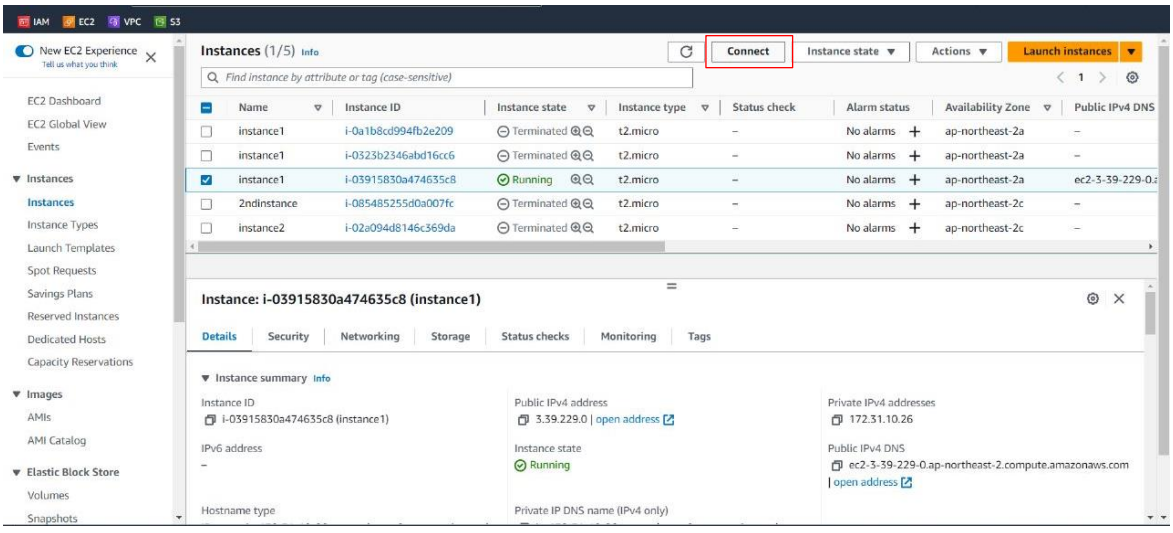
Web Server:

A web server is a software application or a computer system that delivers web content over the internet in response to client requests. It plays a central role in the World Wide Web by hosting websites and serving web pages to users' web browsers.

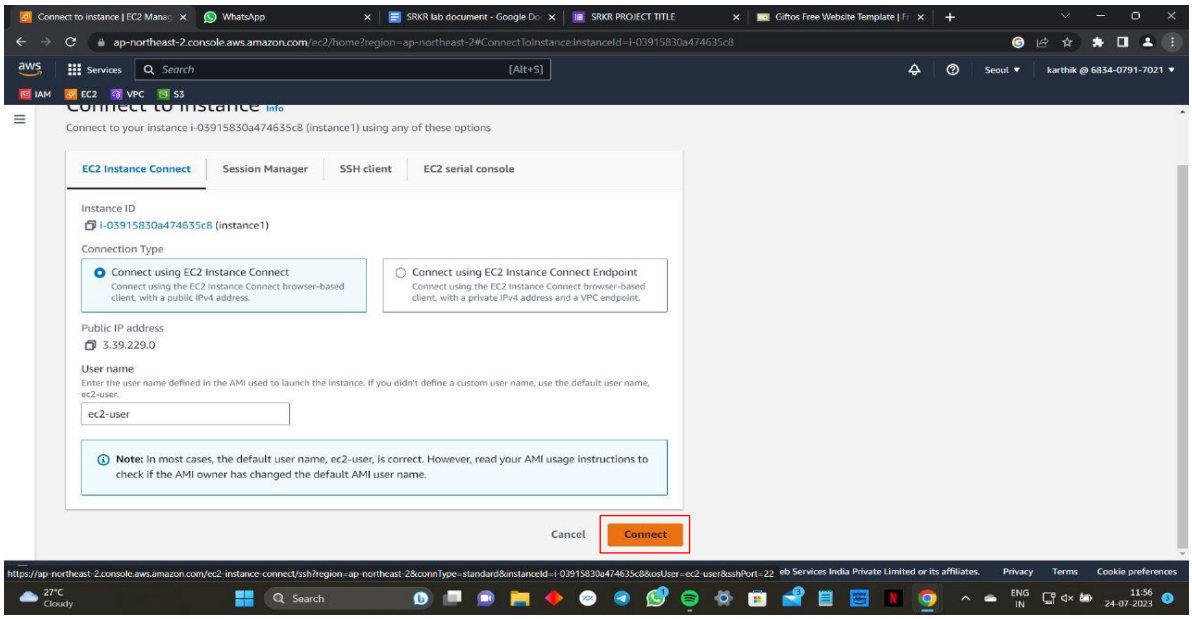
When a user enters a URL (Uniform Resource Locator) or clicks on a hyperlink in their web browser, the browser sends a request to the web server associated with that URL. The web server processes the request and sends back the requested web page or web content, which is displayed in the user's browser.

Configuring web Server:

Step 1: In the EC2 dashboard, locate your running instance and select it. Click on the "Connect" button at the top of the page.

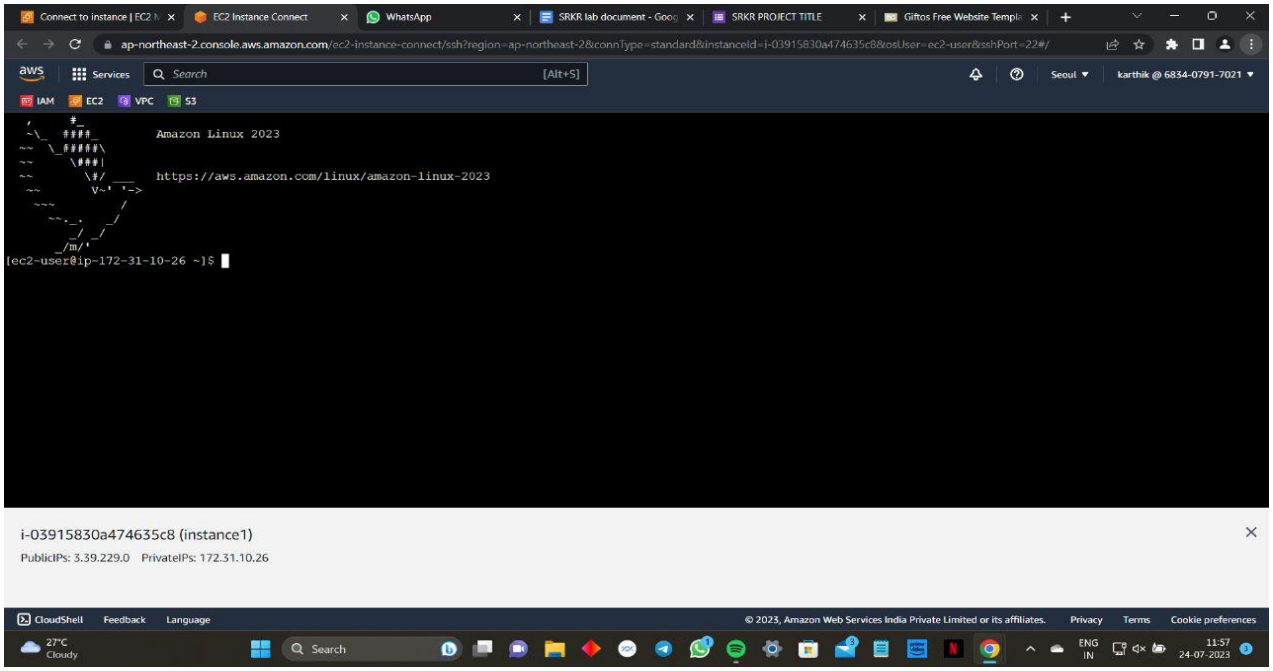


Step 2: Select the "EC2 Instance Connect" option from the "Connect" menu and click "Connect."



Step 3: A new browser-based SSH session will open, and you will be automatically connected to your EC2 instance using Instance Connect.

You should see the command prompt of your EC2 instance in the browser, indicating that you have successfully connected.

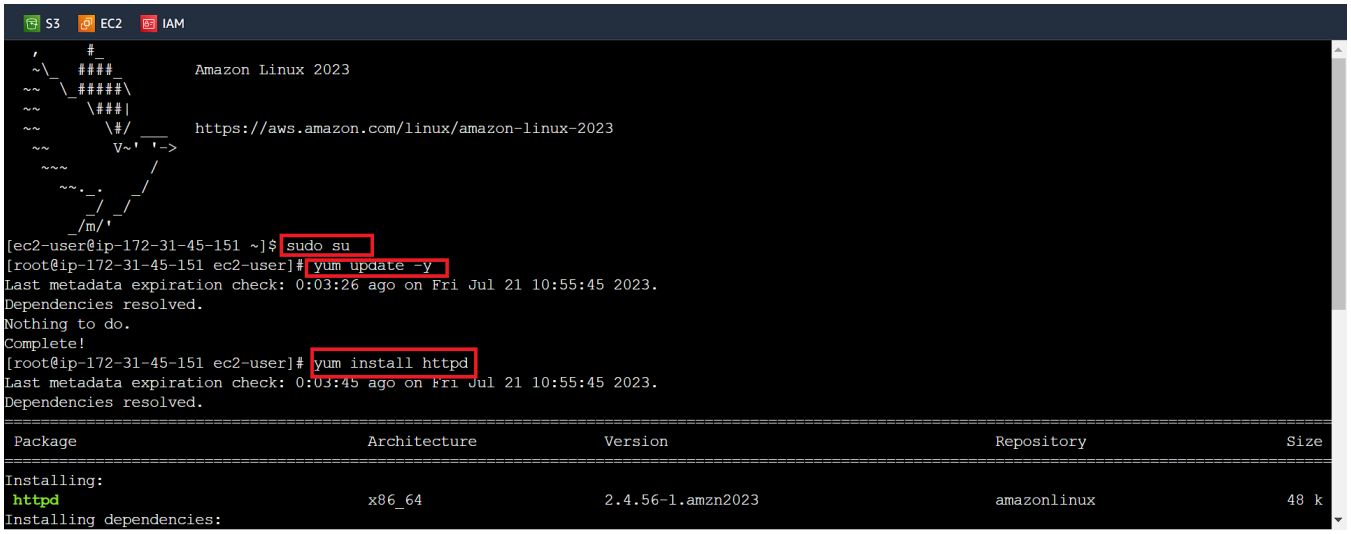


Step 4: Enter the following commands to install the Apache web server.

1.Sudo su: To switch to the superuser (root) account.

2.Yum update -y: It is used to update all installed packages on the system to their latest available versions without user interaction.

3.Yum install httpd: It is used to install the Apache HTTP Server (often simply referred to as "Apache") on Linux distributions.

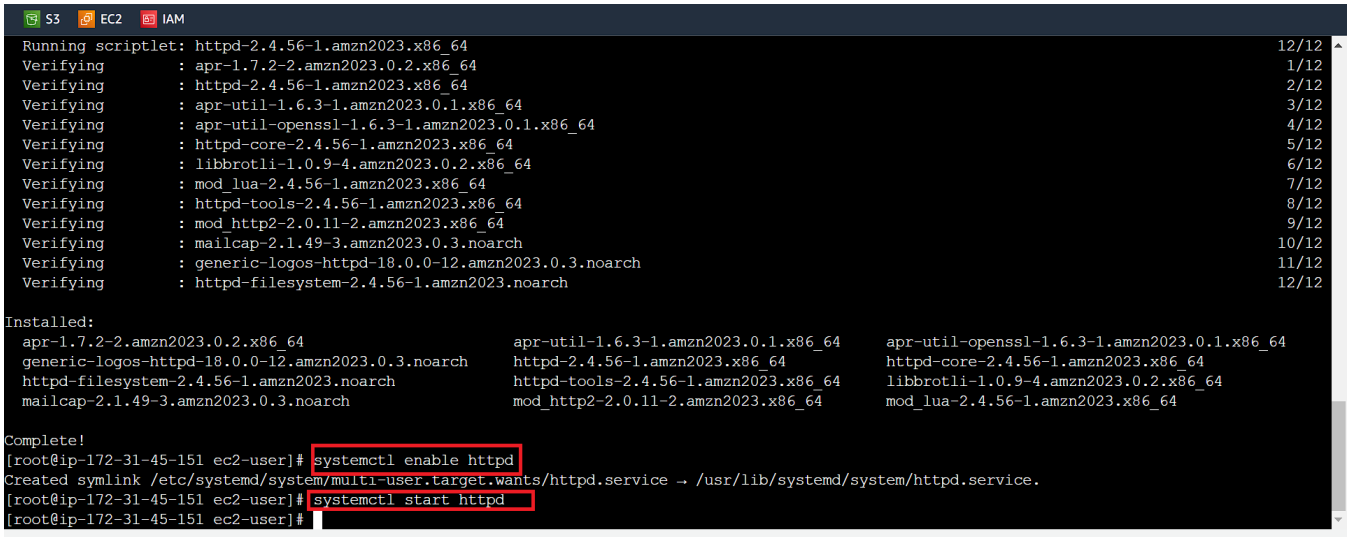


The Apache HTTP Server is one of the most widely used web servers globally. It is an open-source and powerful web server software that can host websites, serve web content, and process HTTP requests from clients (such as web browsers).

Step 5: Start the Apache service and enable it to start automatically on boot. The commands are

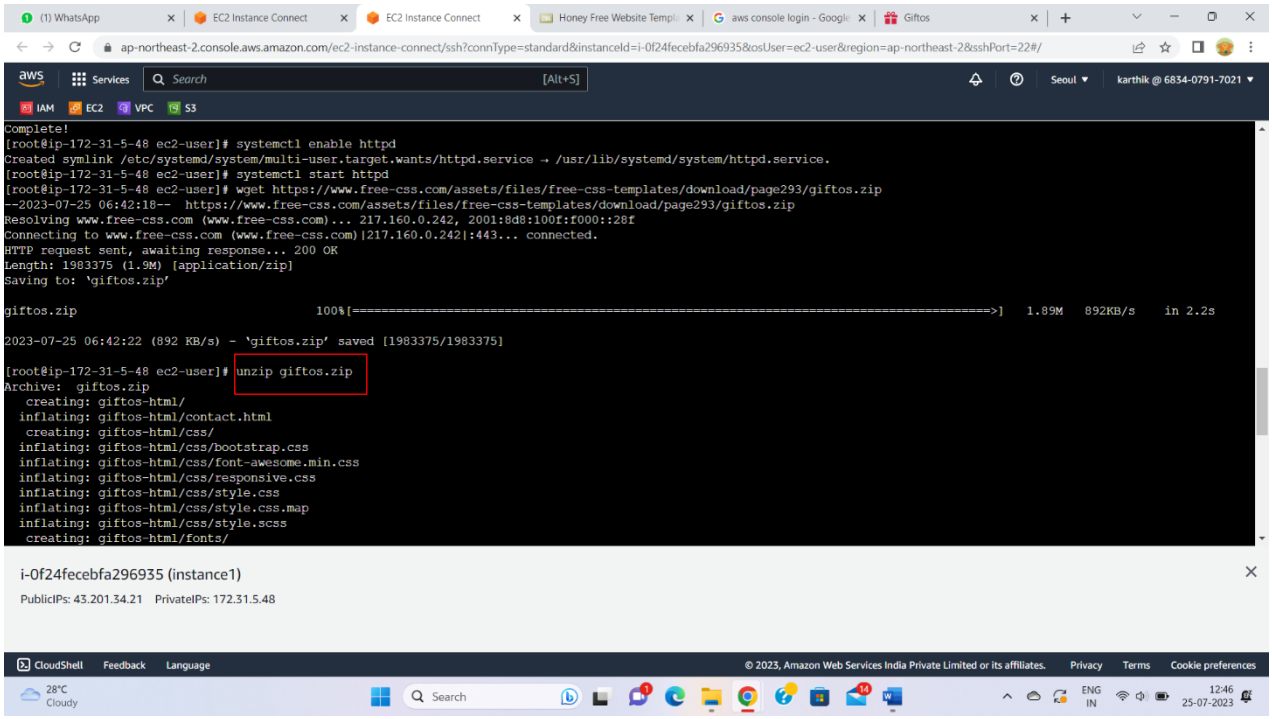
1.Systemctl enable httpd: It is used to enable the Apache HTTP Server (httpd) service to start automatically at system boot on Linux distributions.

2.Systemctl start httpd: It is used to manually start the Apache HTTP Server (httpd) service.



Hosting Website:

Step 1: Upload the link of your html file and unzip it. To unzip we use ‘unzip’ command.



After unzipping process is completed a new folder will be created and all the template files are placed in it. To know the name of that folder we use ls command

Ls: It is used to list the contents of a directory.

Step 2: Move the extracted template files to the Apache document root directory. The web server needs to know where to find the files that make up your website. By default, the web server's "Document Root" directory is the location where it looks for website files. The document root is typically located at /var/www/html/. Use the following commands to move the files.

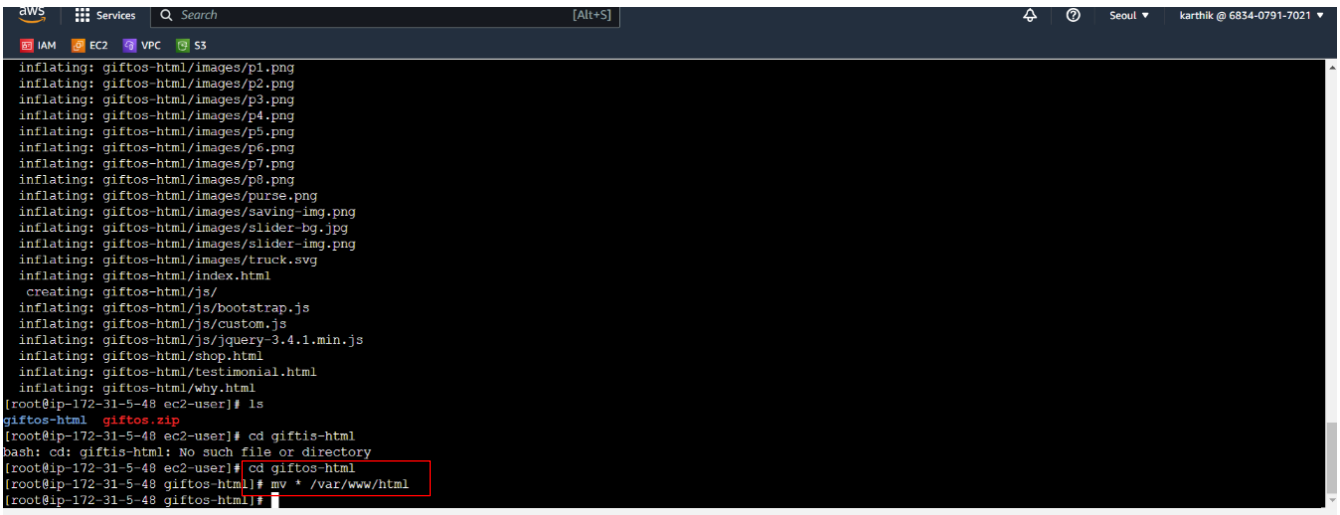
1.ls: To know the name of folder created after unzipping

2.cd: To change the directory

3.mv * /var/www/html/: To move all the template files to root directory.

4.cd /var/www/html/: To move to root directory.

5.ls: check for the files.



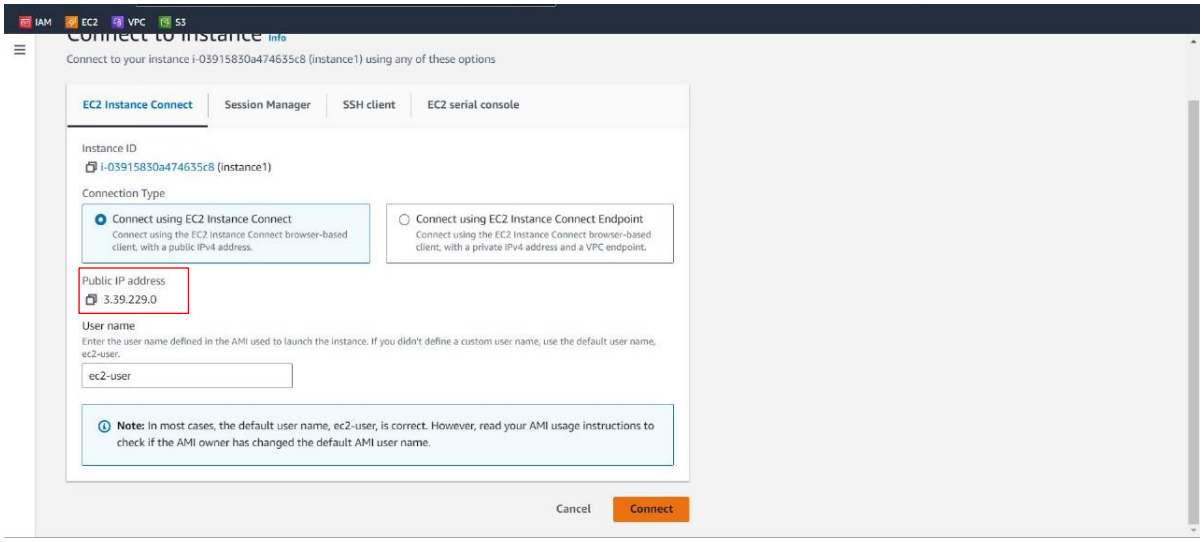
the website files are uploaded and the web server is properly configured, visitors can access your website by entering the public IP address or domain name of your EC2 instance in their web browsers.

Access Your Website:

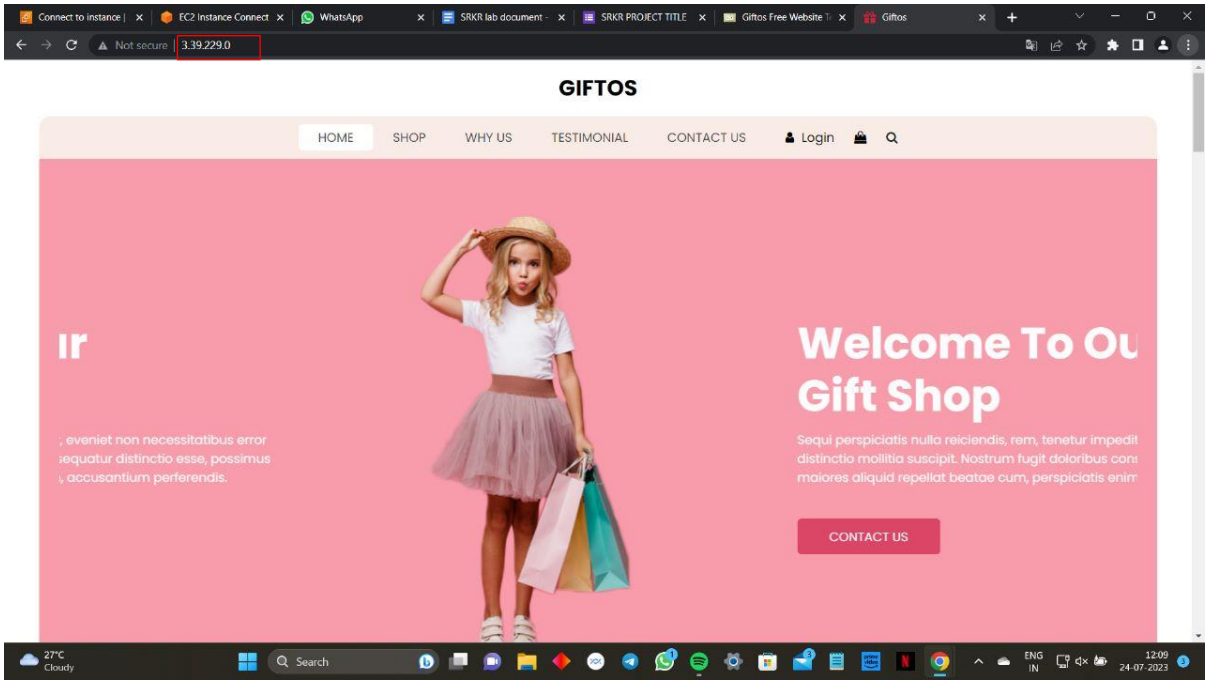
Once the website files are uploaded and the web server is properly configured, visitors can access your website by entering the public IP address or domain name of your EC2 instance in their web browsers.

Ensure that your EC2 instance has a security group configured to allow inbound traffic on the necessary ports (e.g., port 80 for HTTP or port 443 for HTTPS). If you are using the default security group, it should already allow inbound traffic on port 22 for SSH and port 80 for HTTP.

Step 1: Find the EC2 instance hosting your website and copy its public IP address.



Step 2: Enter the public IP address of your EC2 instance directly into the browser's address bar and press Enter.



Similarly configure the webserver in the second instance and host a different website in it.

Create a Load balancer in the specified region and balance traffic between two instances.

Load balancer:

In AWS (Amazon Web Services), a load balancer is a service that distributes incoming application or network traffic across multiple servers (instances) to ensure that no single server becomes overwhelmed, and that the workload is evenly distributed. The load balancer helps to improve the availability and fault tolerance of your application, as well as enhances its performance and scalability.

Load balancers continuously monitor the health and status of servers in the backend pool. They perform health checks by sending periodic requests to the servers and verifying their responses. If a server fails the health check, the load balancer automatically removes it from the pool, directing traffic only to healthy servers.

Target Group:

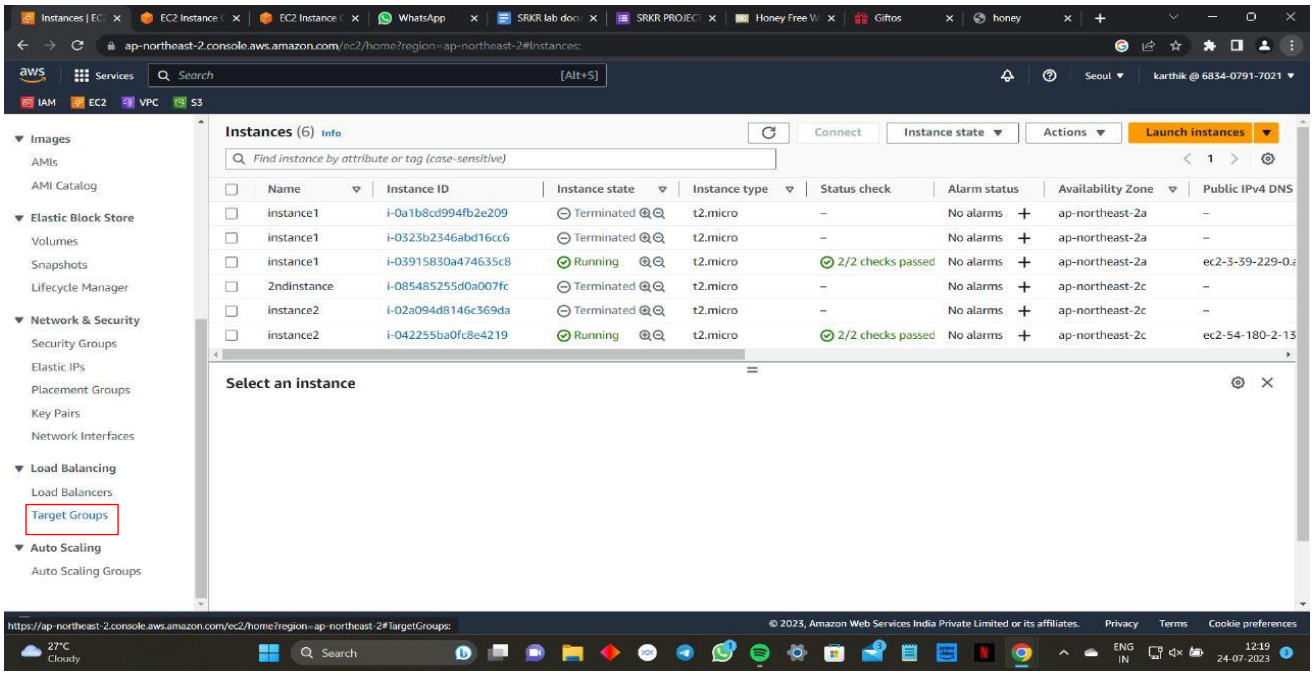
A target group is a logical group of targets (instances or services) that the load balancer routes requests to. When you set up an application load balancer or network load balancer, you create one or more target groups, and each target group is associated with a specific set of ports on the instances.

When creating a load balancer, you configure one or more target groups and specify the routing rules that determine how traffic is distributed among the targets. These rules can be based on conditions like path patterns, hostnames, or ports.

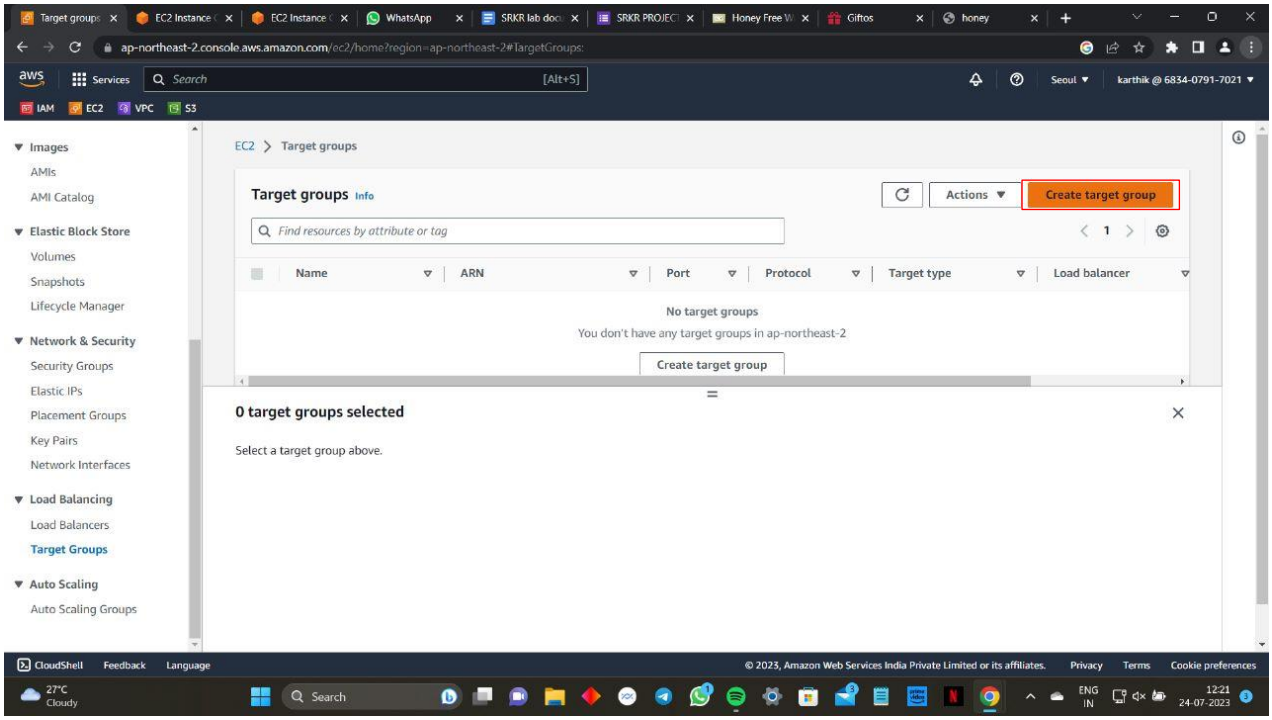
Creating a Target Group:

After creating two instances in two different availability zones follow the below stated steps to create a target group.

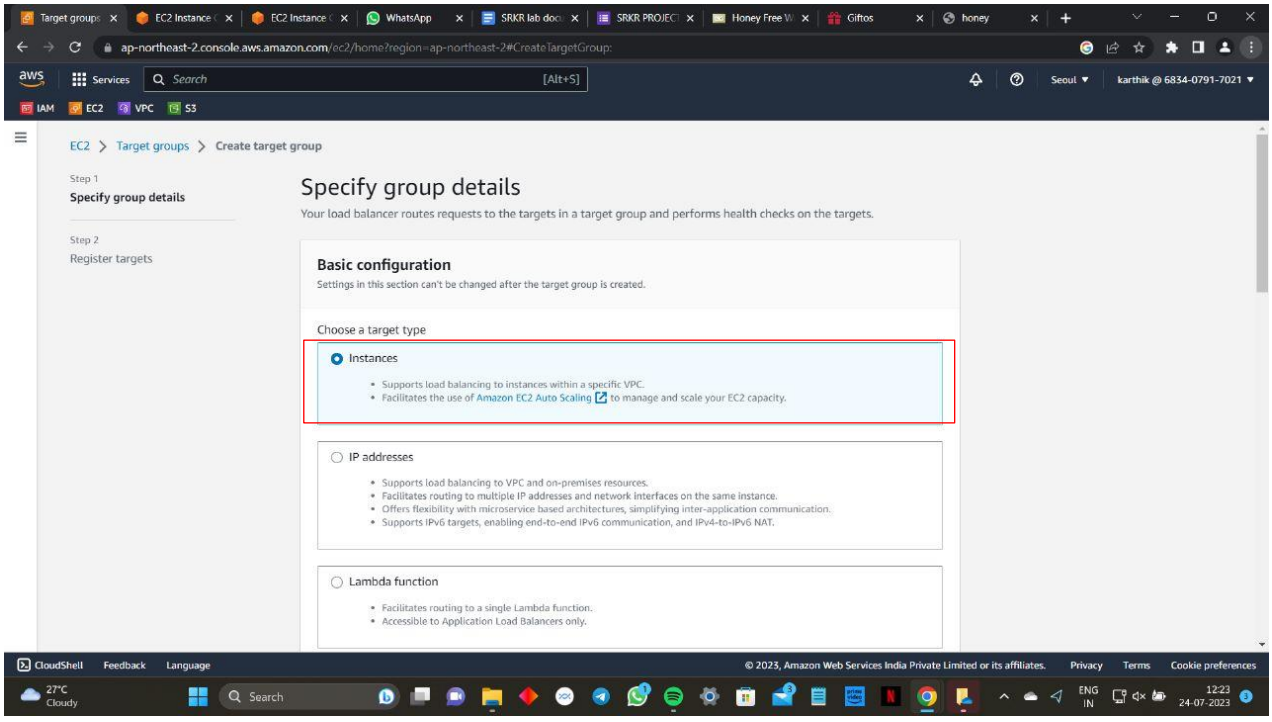
Step-1: In EC2 dashboard click on “Target groups” in the left navigation pane.



Step-2: click on “Create target group”



Step-3: In the Basic configuration choose the Target type as “Instance” since we are going to be working using instances.

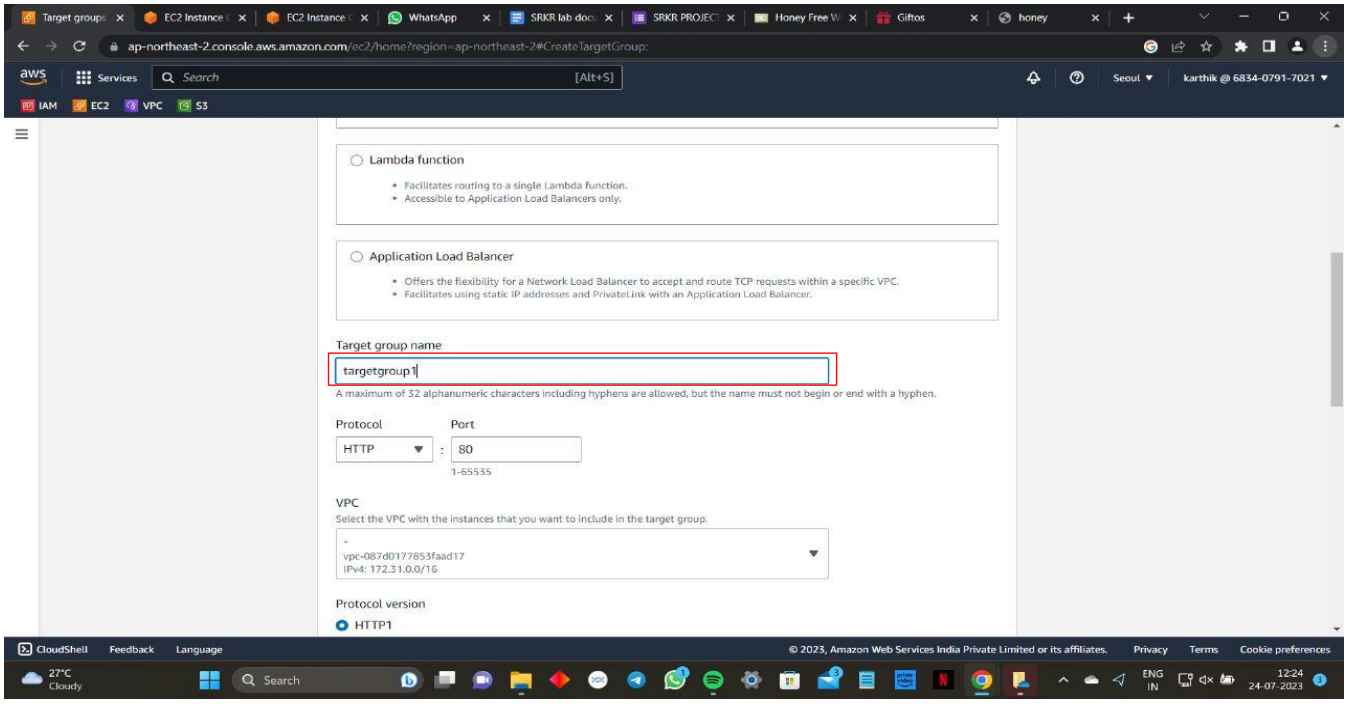


When you create a target group, you specify its target type, which determines the type of target you specify when registering targets with this target group. After you create a target group, you cannot change its target type.

When creating a new target group, you can select the IP address type of your target group. This controls the IP version used to communicate with targets and check their health status. Application Load Balancers support both IPv4 and IPv6 target groups. The default selection is IPv4.

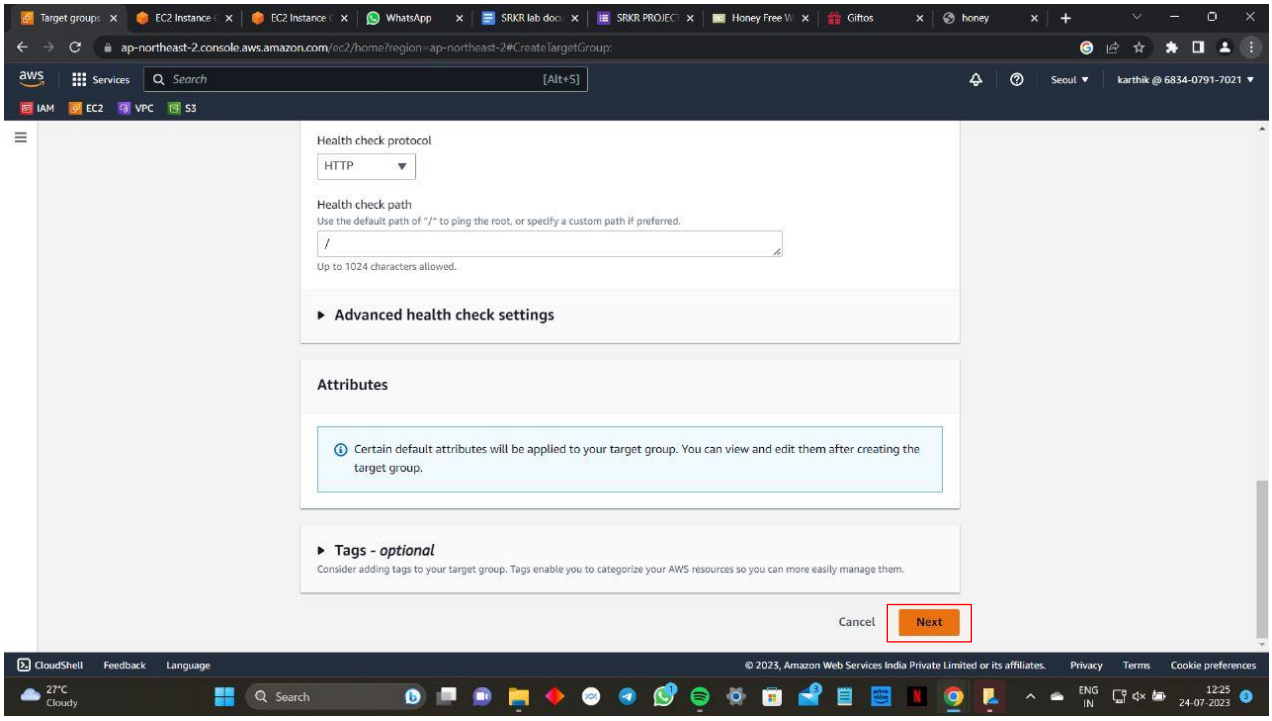
If the target type of your target group is `lambda`, you can register a single Lambda function. When the load balancer receives a request for the Lambda function, it invokes the Lambda function.

Step-4: Give the Target Group a name.

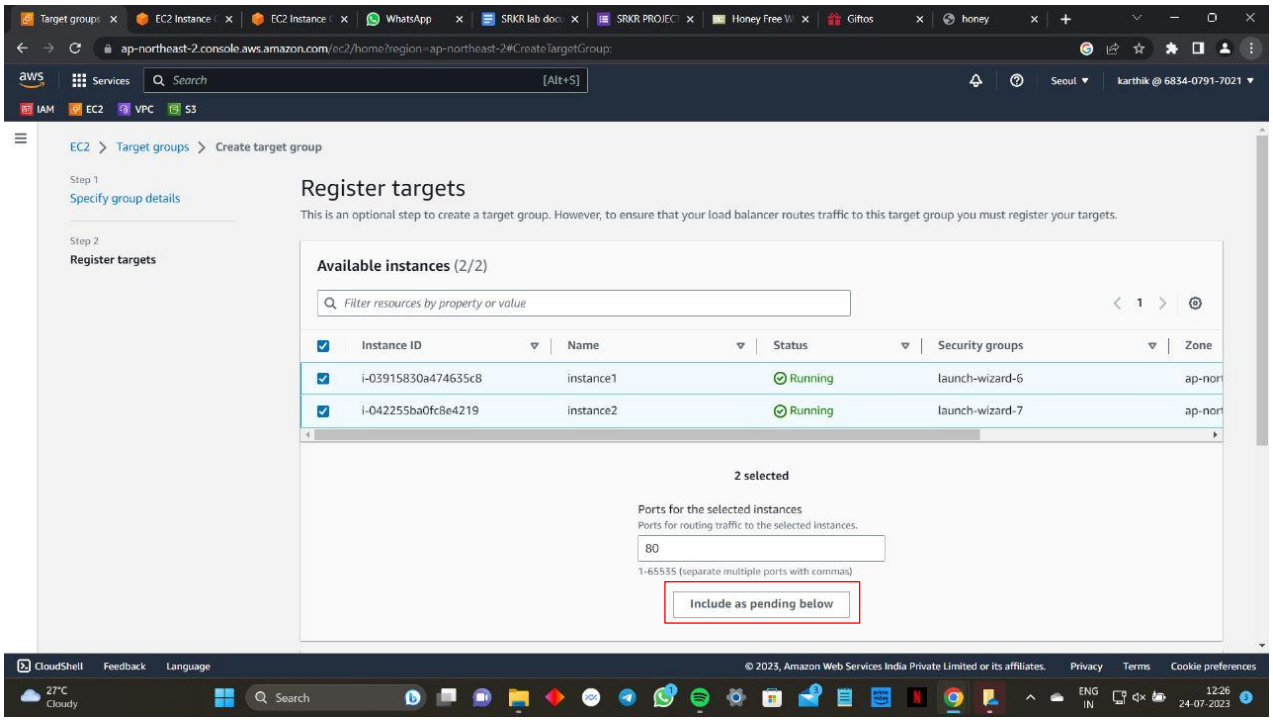


Step-5: Click on "Next".

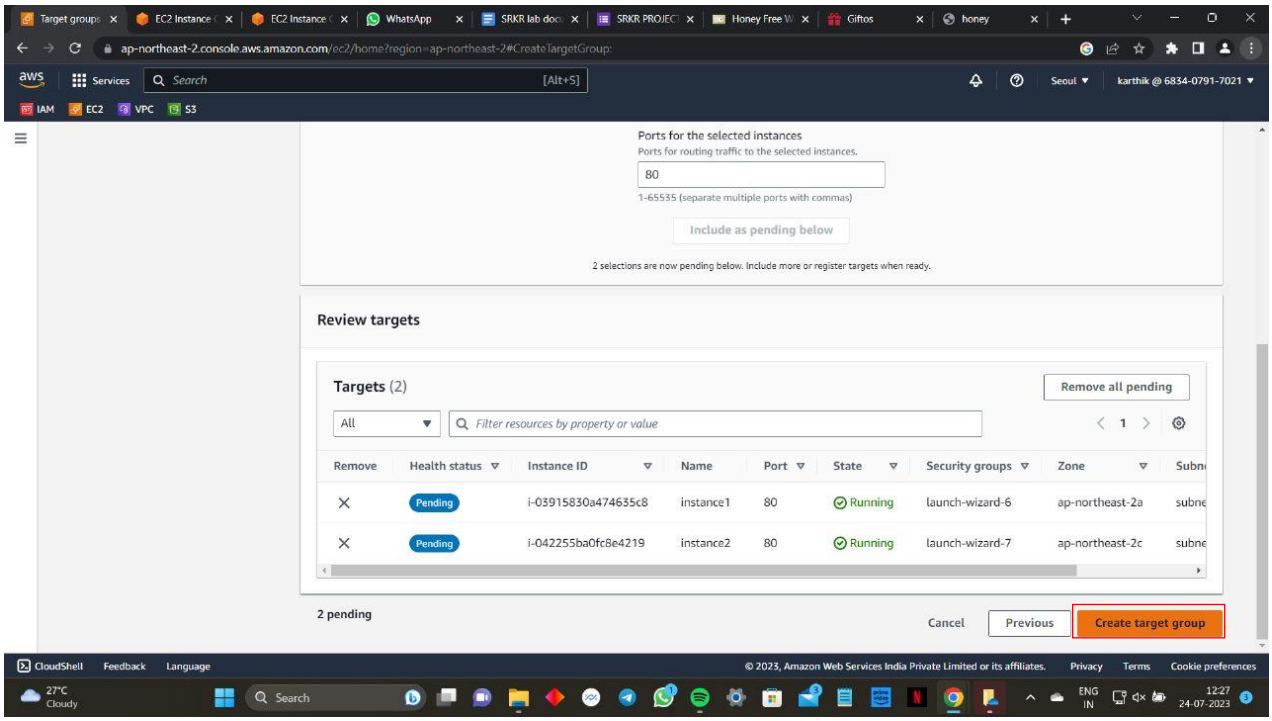
.

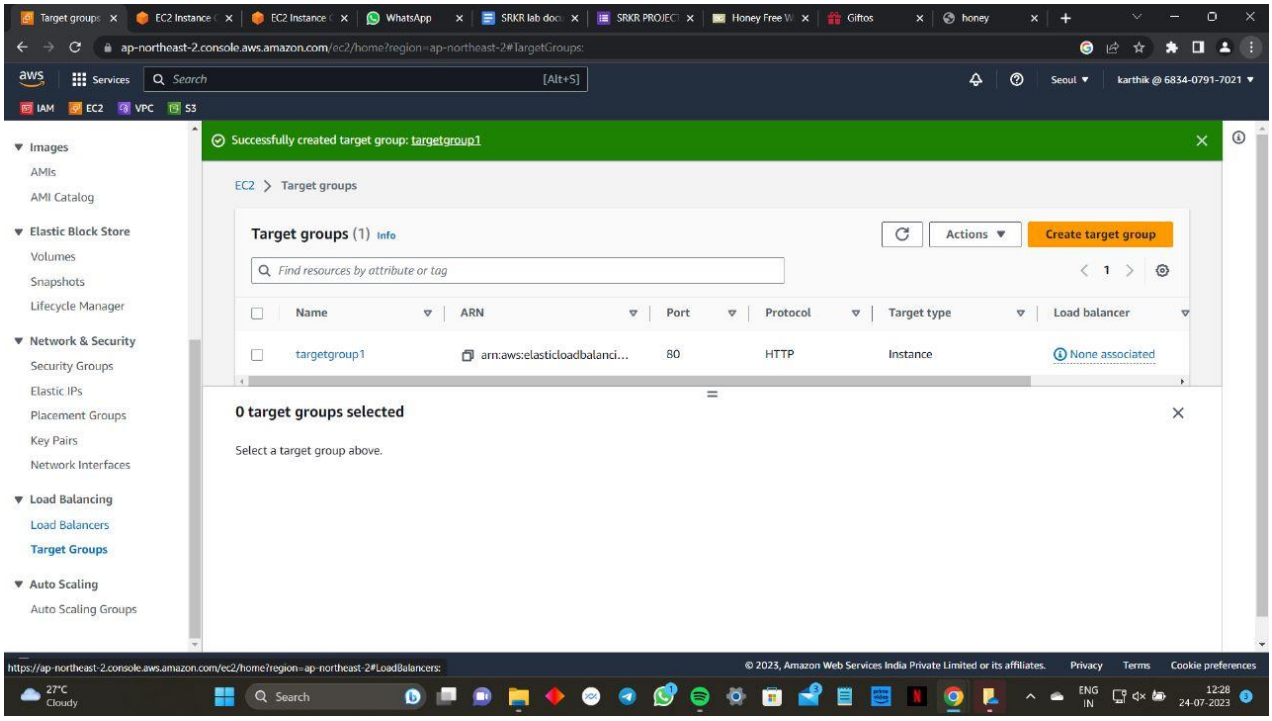


Step-6: Select the two available instances. Click on “Include as pending”.



Step-7: Click on “Create Target group”.





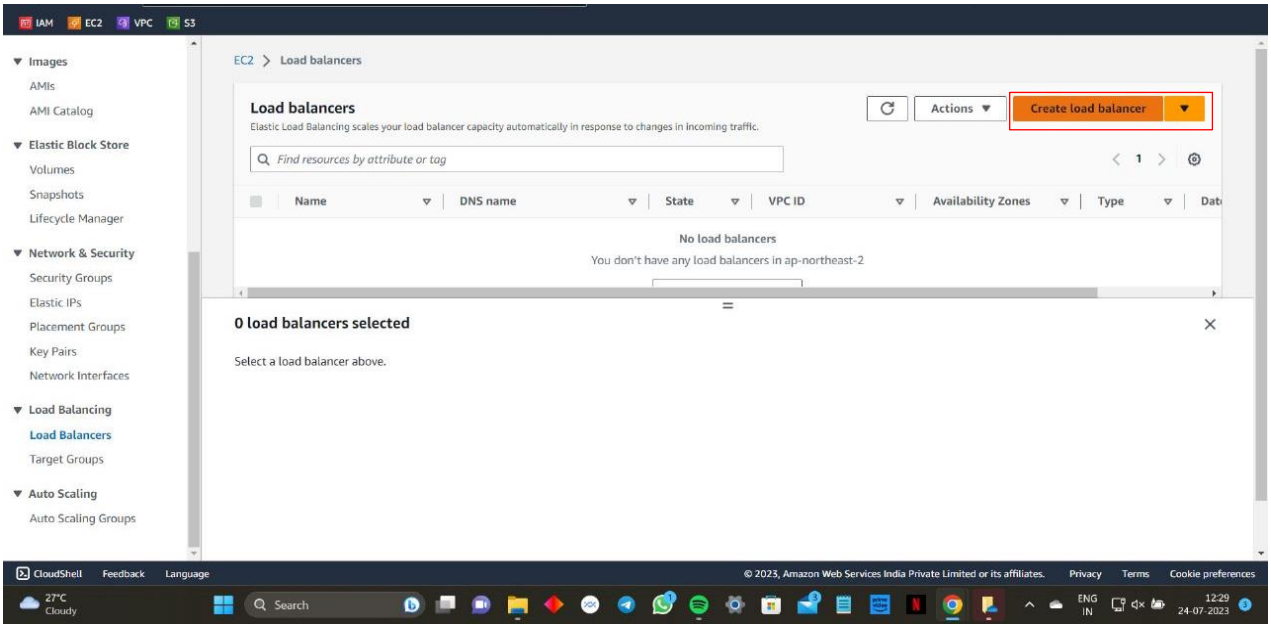
Target group is successfully created. Now we need to associate the target group with a load balancer and register the desired resources (instances, IP addresses, or Lambda functions) with the target group. This allows the load balancer to start directing traffic to the registered resources based on the defined routing rules and health checks.

Creating an Elastic Load Balancer:

A load balancer is essential for several reasons, especially when hosting web applications or services that receive significant traffic. Load balancers play a critical role in enhancing the performance, availability, and security of your application infrastructure.

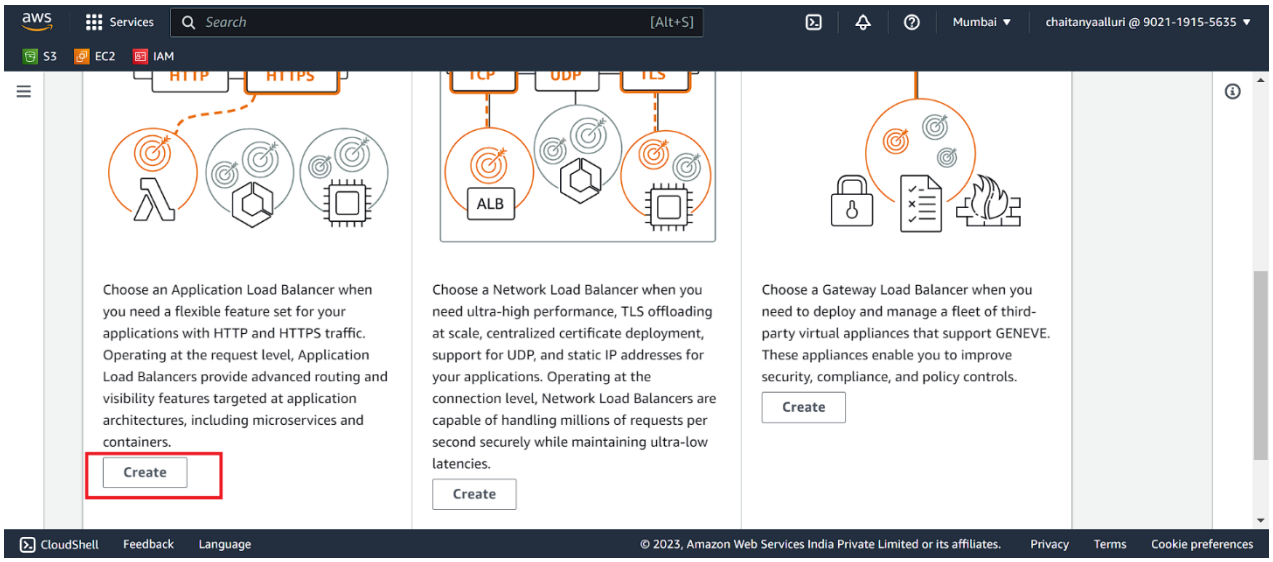
Follow the below stated steps to create an elastic load balancer to balance the traffic between two instances.

Step-1: In EC2 dashboard click on “Load balancers” in the left navigation pane and click on “Create Load balancer” button.

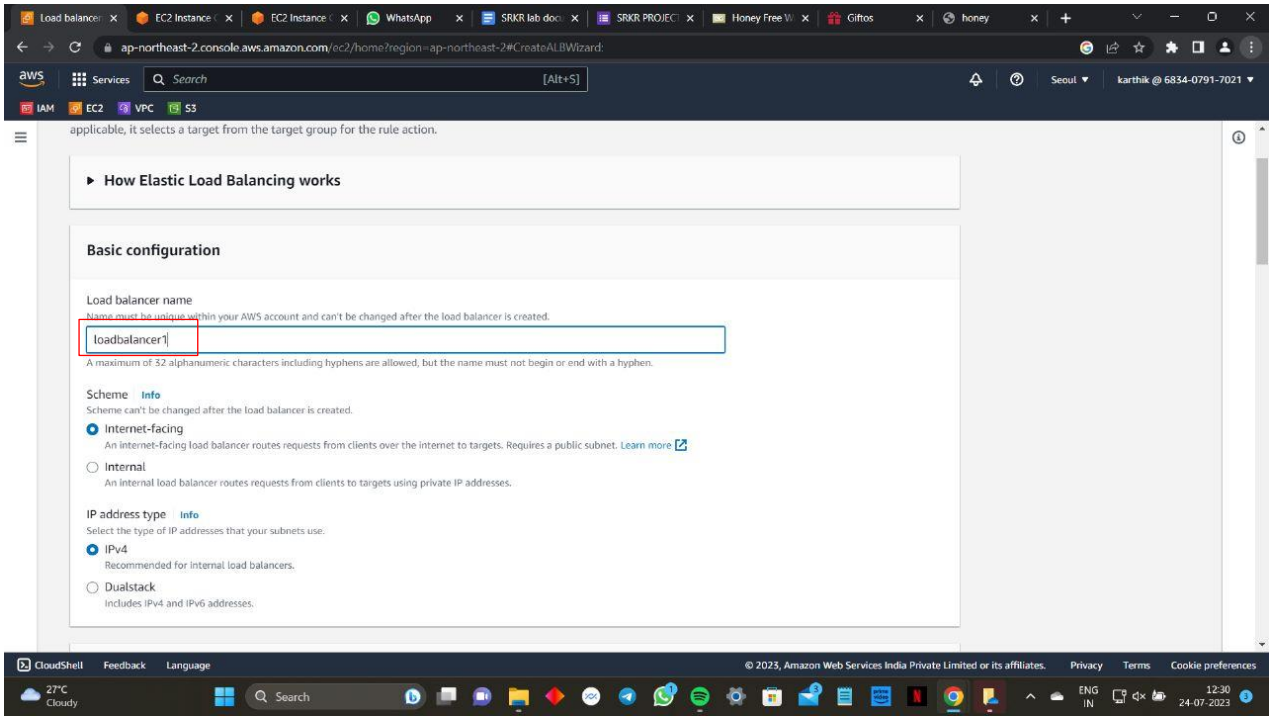


Step-2: Select the appropriate load balancer type based on your requirements. AWS offers Application Load Balancer (ALB) for HTTP/HTTPS traffic or Network Load Balancer (NLB) for TCP/UDP traffic.

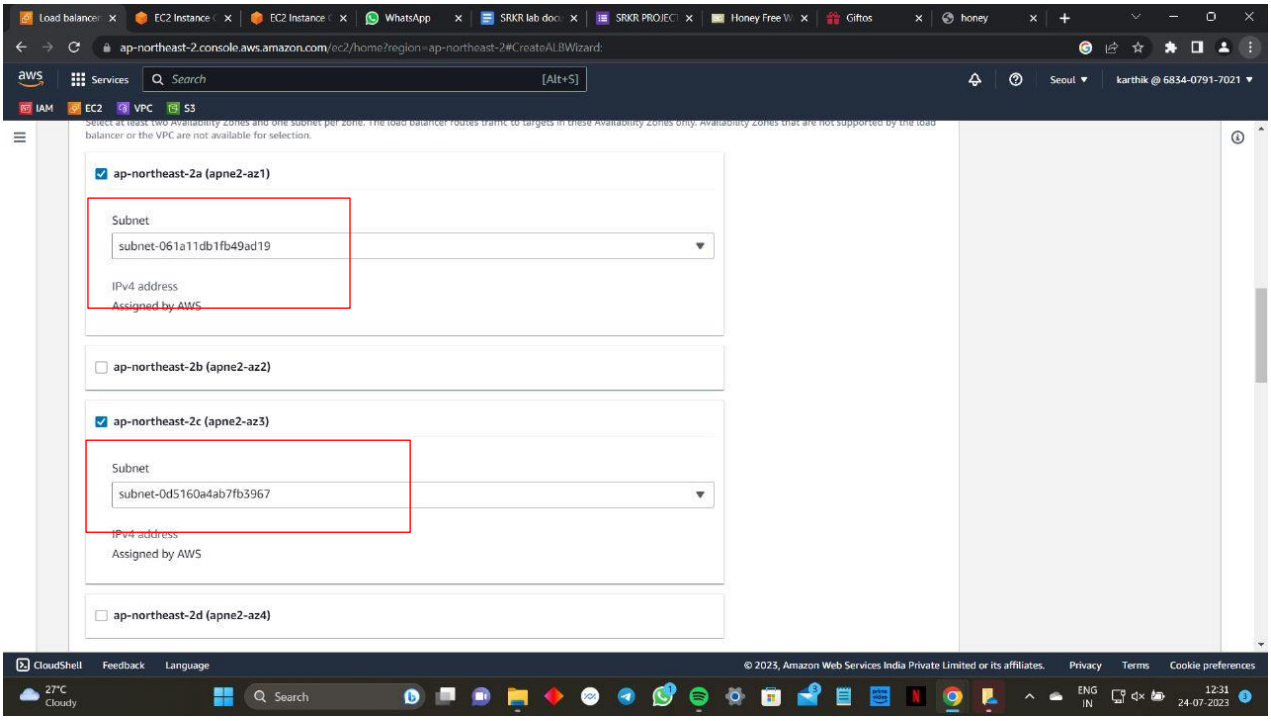
Click on “Create” in the Application Load balancer section.



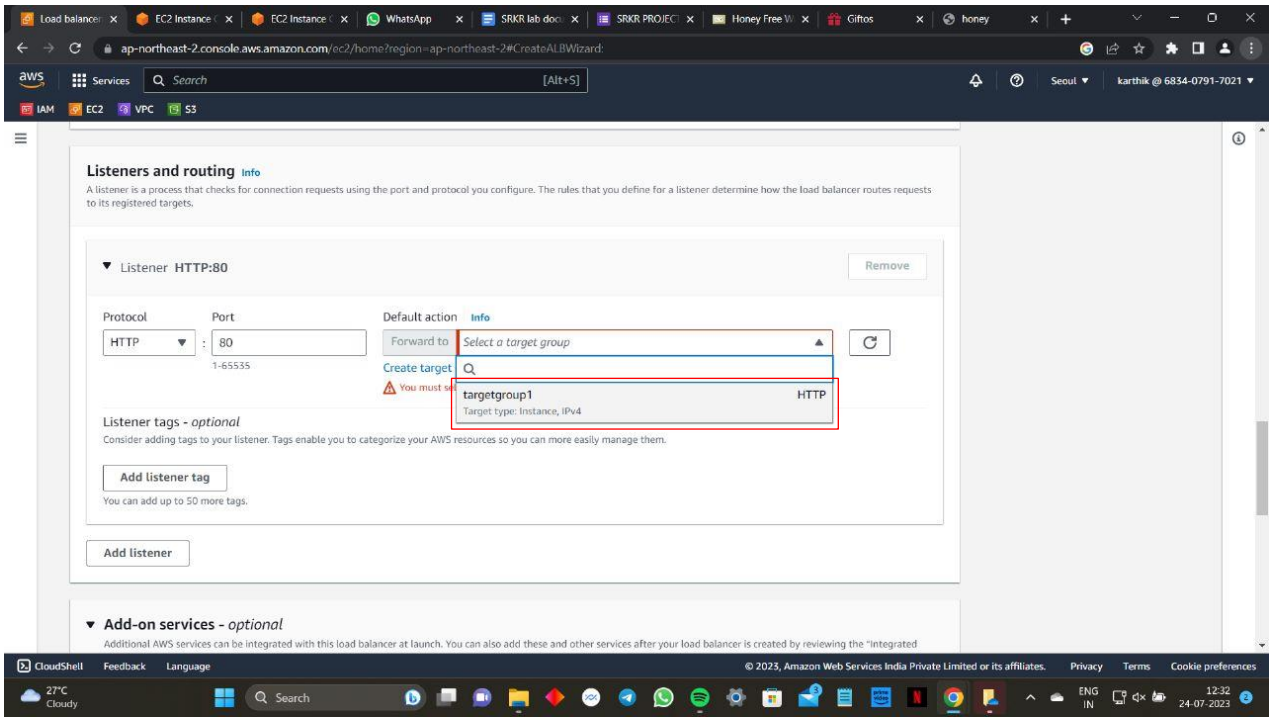
Step-3: In the Basic Configuration section, choose a name for the Load balancer.



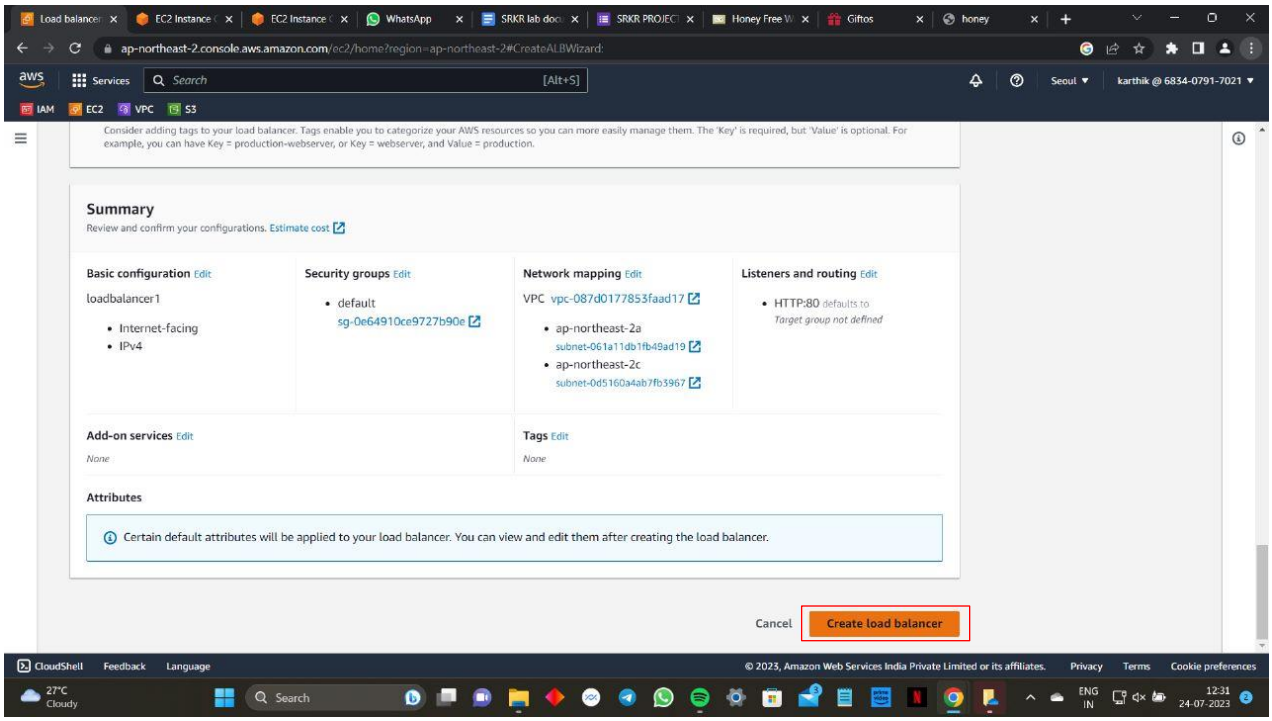
Step-4: In the Mapping section select the availability regions in which your Instances are present. This automatically selects a subnet for those selected availability regions. Or we can select all the availability regions in case of Autoscaling.



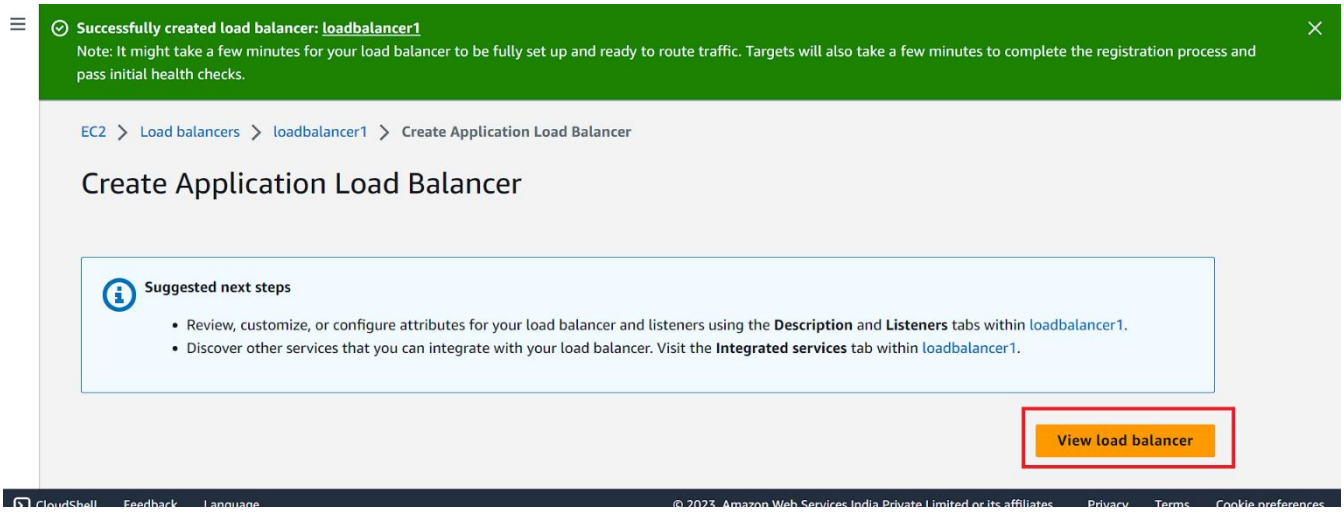
Step-5: In Listeners and routing section go to default action and select the target group which was previously created.



Step-6: Click on “Create Load balancer”.



Step-7: The page refreshes and displays that the Load balancer is created. To view the Load balancer, click on “View Load balancer” button.



Verifying the load balancer DNS on the browser

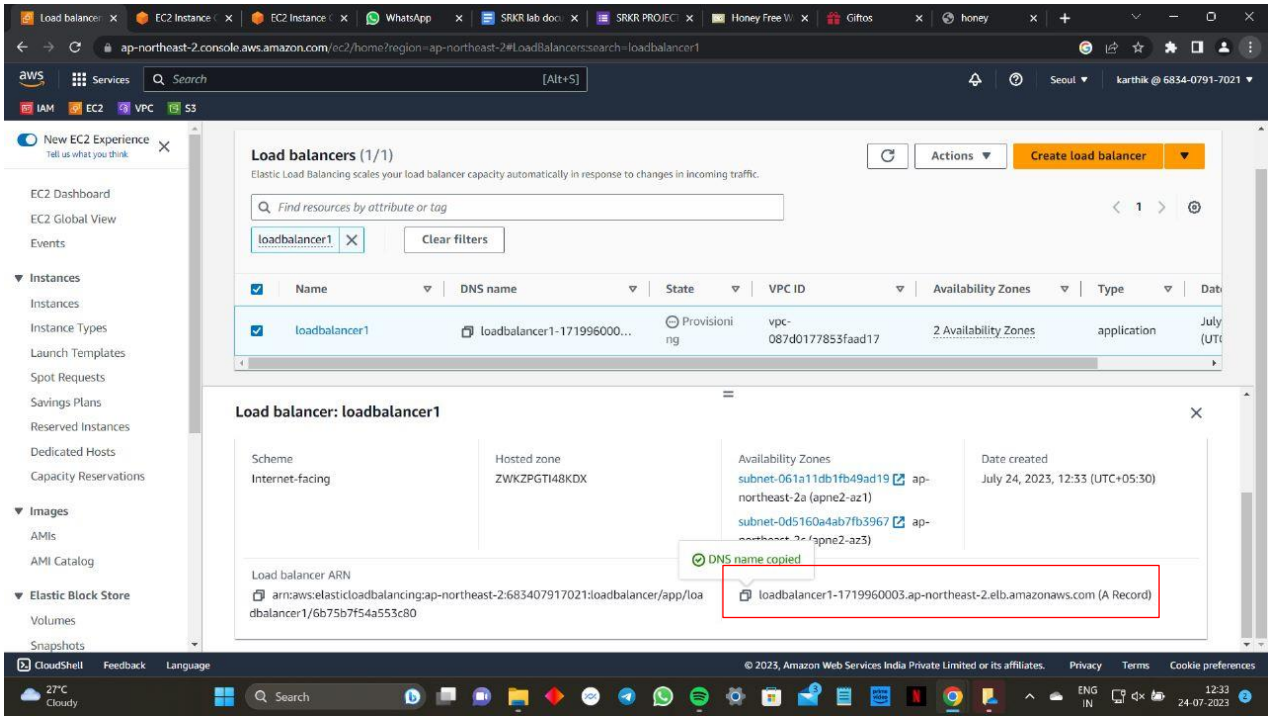
Load balancers are designed to distribute traffic across multiple backend servers. By checking the load balancer's DNS, you ensure that it is correctly distributing traffic among the healthy servers, providing high availability and redundancy.

Load balancers evenly distribute incoming requests among backend servers based on defined algorithms. Verifying the DNS helps you confirm that the load balancer is performing this function correctly, preventing any single server from being overwhelmed with traffic.

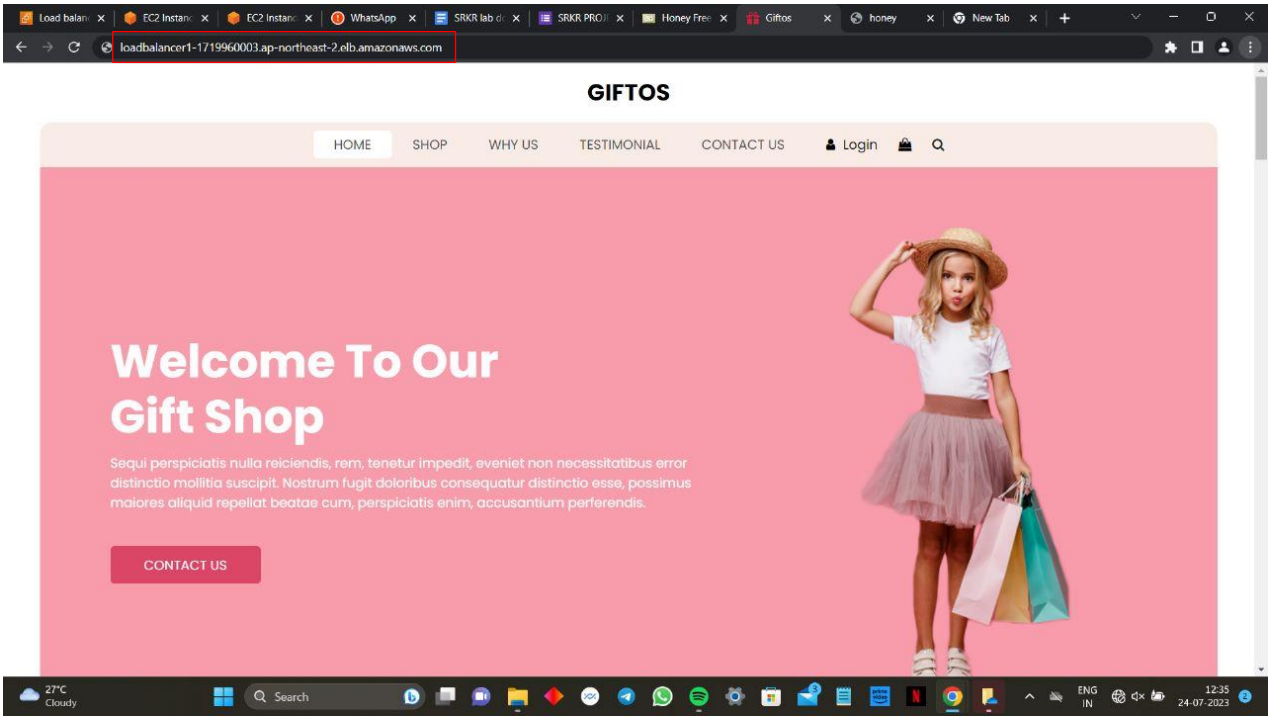
Verifying the Load balancer DNS:

By following the below steps, you can verify the load balancer's DNS configuration and ensure that it is functioning correctly to distribute traffic to your application or website.

Step 1: Select the Load balancer. Copy the DNS Name of the selected load balancer.



Step 2: paste the DNS name of the load balancer on the browser. One of the hosted websites should be displayed on the browser.


loadbalancer1-1719960003.ap-northeast-2.elb.amazonaws.com
GIFTOS
HOME
SHOP
WHY US
TESTIMONIAL
CONTACT US
Login
Welcome To Our Gift Shop
Sequi perspiciatis nulla reiciendis, rem, tenetur impedit, eveniet non necessitatibus error distinctio mollitia suscipit. Nostrum fugit doloribus consequatur distinctio esse, possimus maiores aliquid repellat beatae cum, perspiciatis enim, accusantium perferendis.
CONTACT US
27°C
Cloudy
Search
ENG
IN
12:35
24-07-2023

Create a cloud front distribution and Push the load balancer DNS to the cloud front

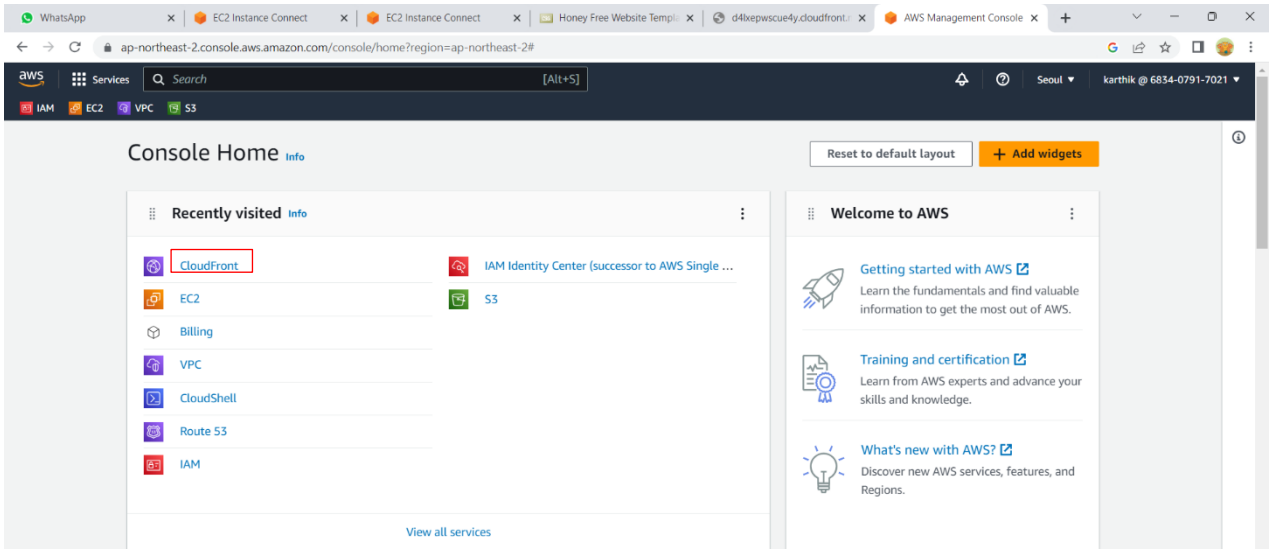
Cloud front:

In AWS (Amazon Web Services), Amazon CloudFront is a content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low-latency, high-speed transfers. It is designed to provide a scalable and efficient way to distribute content to users, reducing latency and improving the overall user experience.

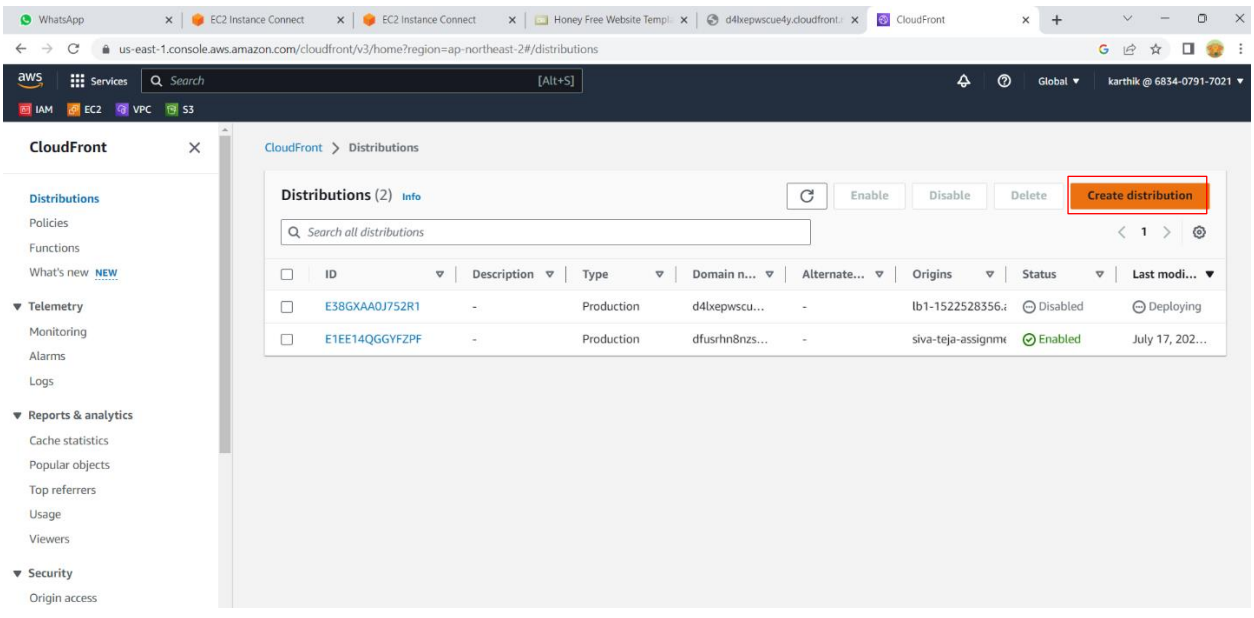
By using CloudFront, you can effectively offload traffic from your origin server, reduce latency, and deliver content to end-users with high availability and reliability across the globe. It is commonly used for serving static and dynamic content, live or on-demand streaming of media files, and improving website performance.

Steps to create cloud front distribution:

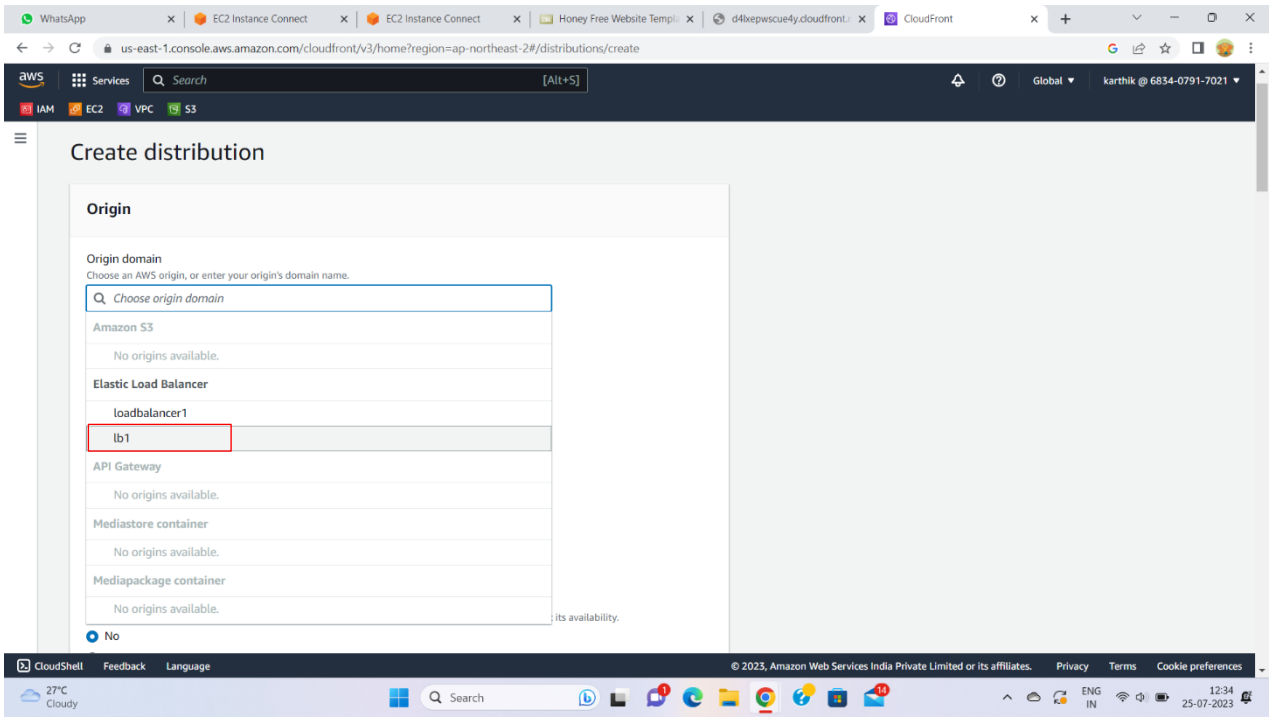
Step 1: go to the AWS Management Console and navigate to the "CloudFront" service.



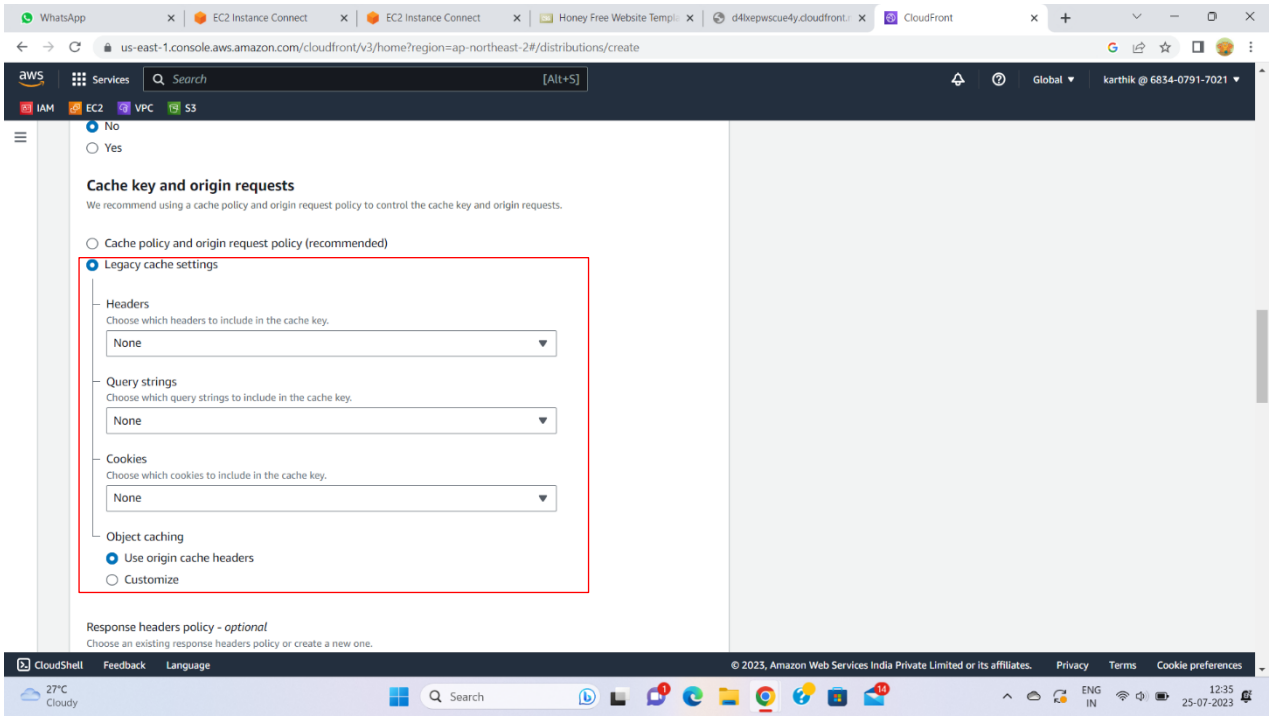
Step 2: Select create a distribution



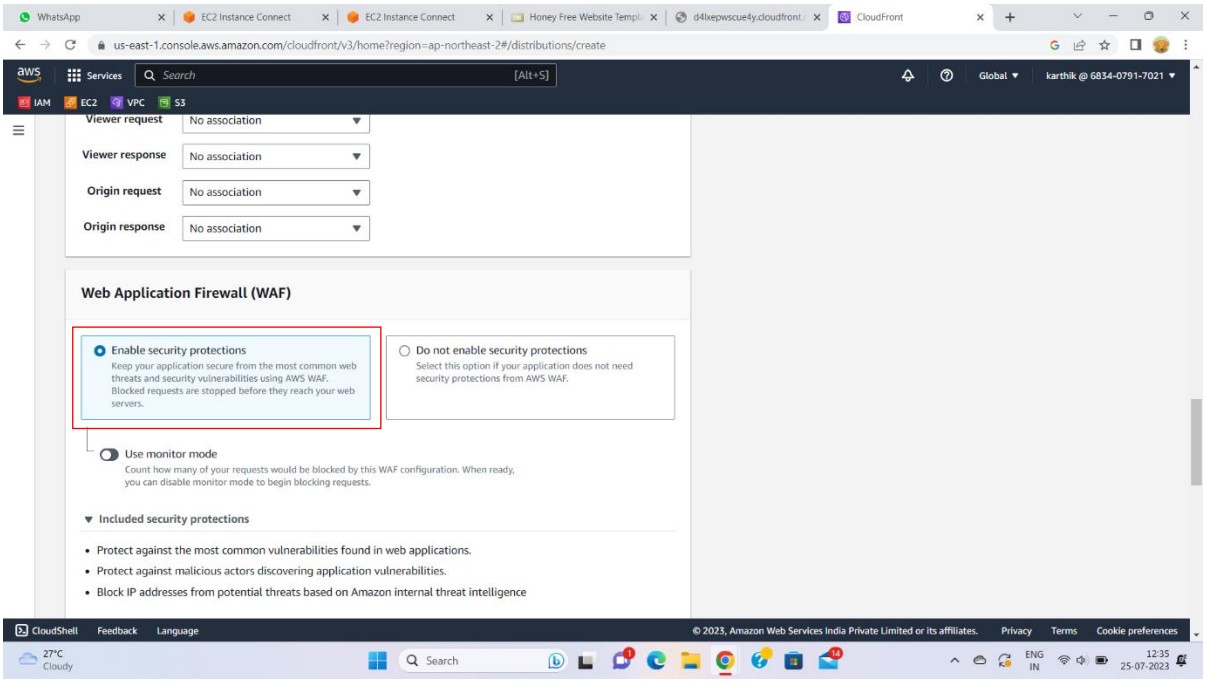
Step 3: In the distribution creation wizard, choose the "Web" delivery method, which is used for delivering content over HTTP and HTTPS.



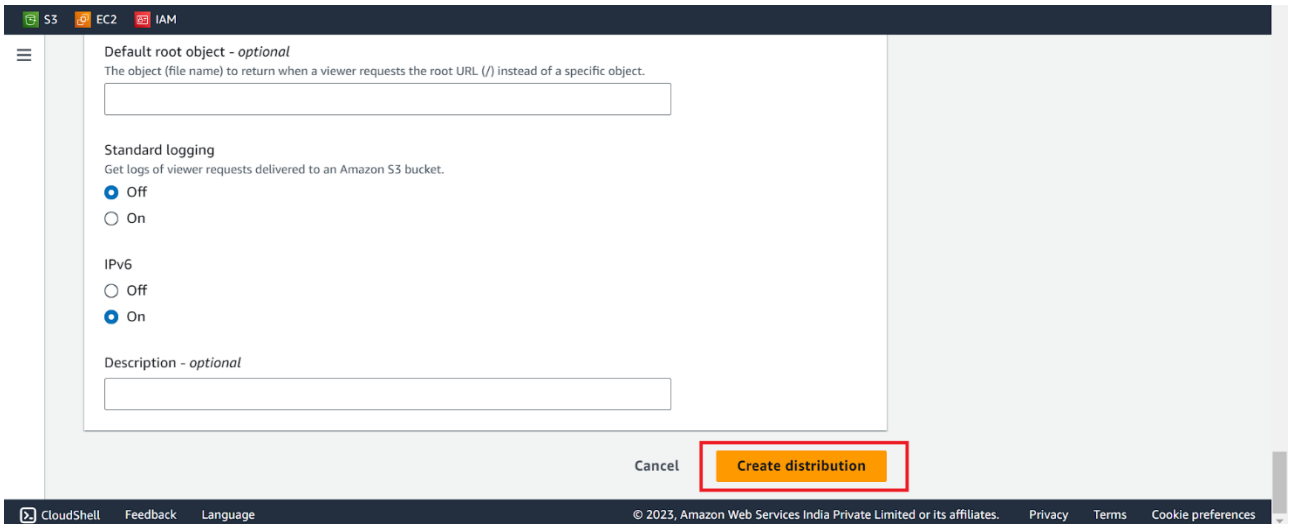
Step 4: under the cache key and origin requests section choose legacy cache settings and under Headers select all.



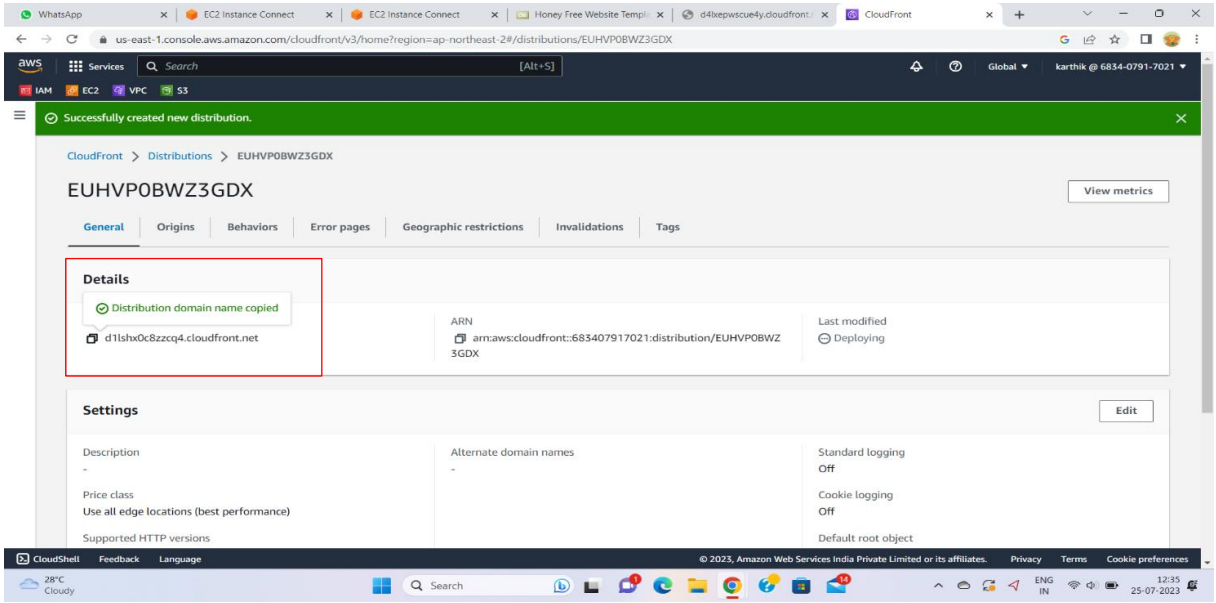
Step 5: Enable web application firewall



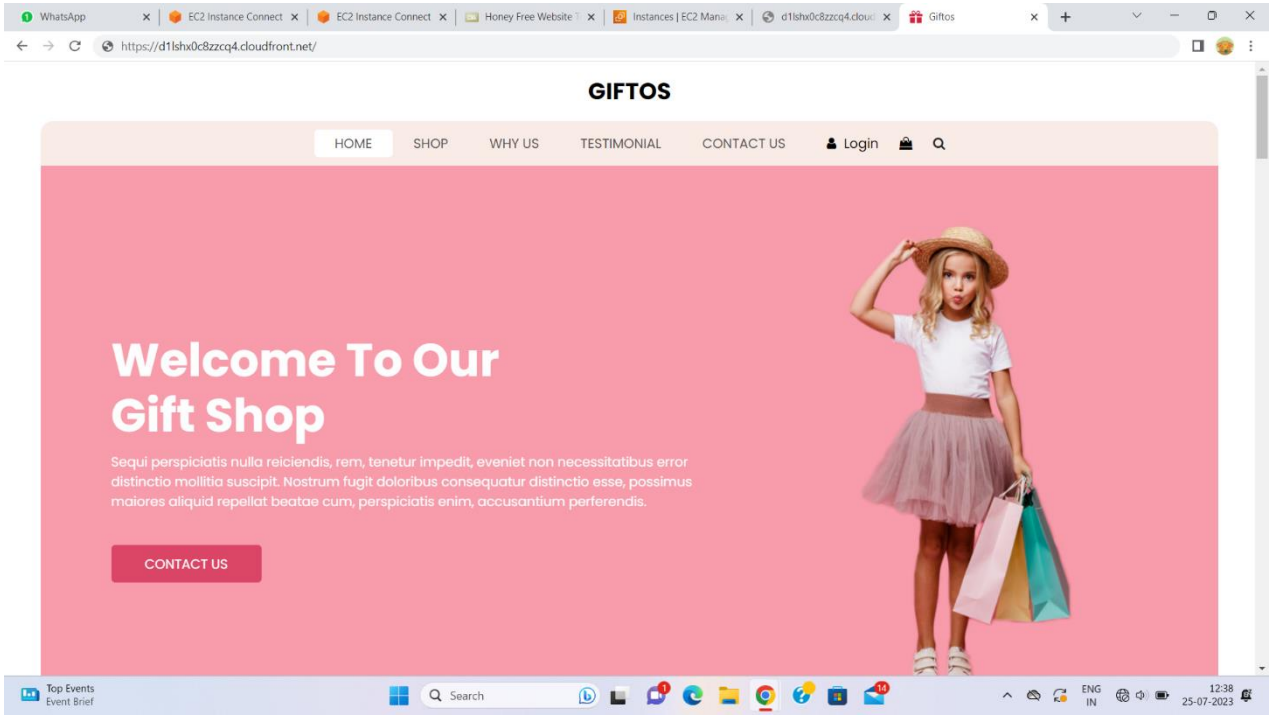
Step 6: click on create distribution.



Step 7: When your distribution is deployed, confirm that you can access your content using your new CloudFront domain name. copy the domain name



Step 7: paste it on the browser.



CONCLUSION :

The project "Giftos Horizon Website Using ELB Cloud Front" successfully demonstrates the implementation of a resilient and scalable web hosting solution on the AWS platform. By leveraging AWS Elastic Load Balancing (ELB) and CloudFront, the project achieves high availability, fault tolerance, and rapid content delivery. The architecture distributes incoming traffic across multiple EC2 instances, optimizing application performance under varying load conditions and reducing latency. The project underscores the robustness of AWS services in creating a reliable and efficient web hosting infrastructure.