AWS Cloud Virtual Internship

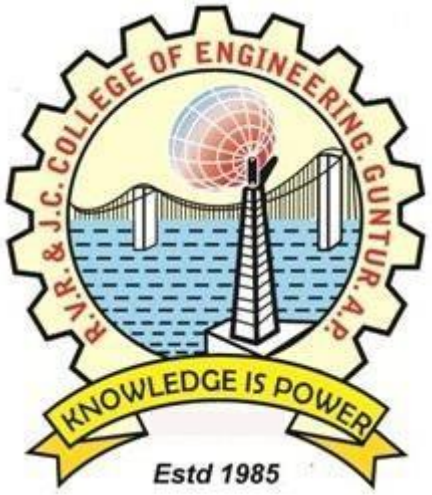
Submitted in partial fulfillment of requirements to CSE (Data Science)

Summer Internship (CD - 353)

III/IV B. Tech CSE(DS) (V Semester)

Submitted by

Bhanu Saketh Pirrala (Y21CD040)



DECEMBER 2023

R.V.R. & J.C. College of Engineering (Autonomous)

(NAAC A+ Grade) (Approved by A.I.C.T.E.)

(Affiliated to Acharya Nagarjuna University)

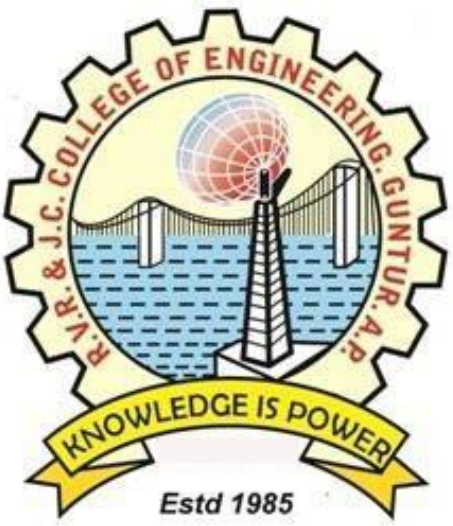
Chandramoulipuram : : Chowdavaram

Guntur – 522 019

R. V. R. & J. C. COLLEGE OF ENGINEERING

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)



CERTIFICATE

This is to certify that this internship report **"AWS Cloud Virtual Internship"** is the Bonafide work of **"Bhanu Saketh Pirrala (Y21CD040)"** who has carried out the work under my supervision and submitted in partial fulfillment for the award of **Summer Internship (CD-353)** during the year 2023 - 2024.

Mrs. B. Dhivya
Internship Incharge

Dr. M. V. P. Chandra Sekhara Rao
Prof. & HoD, CSE(DS)

ACKNOWLEDGEMENT

I would like to express our sincere gratitude to these dignitaries, who are with us in the journey of my Summer Internship "**AWS Cloud Virtual Internship**."

First and foremost, we extend our heartfelt thanks to **Dr. Kolla Srinivas**, principal of **R. V. R. & J. C. College of Engineering**, Guntur, for providing me with such overwhelming environment to undertake this internship.

I am utmost grateful to **Dr. M. V. P. Chandra Shekara Rao**, Head of the Department of Computer Science and Engineering (Data Science) for paving a path for me and assisting me to meet the requirements that are needed to complete this internship.

I extend my gratitude to the Incharge **Mrs. B. Dhivya** for her ecstatic guidance and feedback throughout the internship. Her constant support and constructive criticism have helped me in completing the internship.

I would also like to express our sincere thanks to my friends and family for their moral support though out our journey.

Bhanu Saketh Pirrala
(Y21CD040)

SUMMER INTERNSHIP CERTIFICATE

ABSTRACT

Amazon Web Services (AWS) stands as a comprehensive and dynamic cloud computing platform, providing a diverse array of services tailored to aid organizations in constructing, deploying, and managing applications and infrastructure within the cloud. Tailored to accommodate businesses of varied scales, from nascent startups to expansive enterprises, AWS delivers scalable and adaptable solutions.At its core, AWS furnishes fundamental services such as Amazon EC2 for computing power, Amazon S3 for storage, and Amazon RDS for databases. Augmenting these foundational offerings, AWS boasts an extensive suite of services spanning networking, security, machine learning, analytics, and IoT, addressing multifaceted facets of cloud computing. These services collectively empower businesses to harness cloud capabilities, fostering innovation, cost efficiency, and operational scalability.Central to AWS's infrastructure is its global footprint, comprising multiple regions and availability zones. This architecture ensures robust high availability and fault tolerance, crucial for application and data integrity. This expansive global reach enables organizations to strategically deploy applications proximate to their user base, resulting in diminished latency and enhanced performance.One of the pivotal advantages of AWS lies in its pay-as-you-go pricing model, affording businesses the flexibility to remunerate solely for the resources they consume. This approach mitigates upfront investments in hardware or infrastructure, rendering experimentation, development, and deployment of applications cost-effective.Additionally, AWS provides an arsenal of management tools automating processes, simplifying administration, and bolstering operational efficiency. These tools facilitate meticulous monitoring, robust security enforcement, and optimal resource utilization, ensuring the dependable and secure operation of applications within the AWS ecosystem.

Table of Contents

List of Figures

Chapter 1 INTRODUCTION

Cloud computing is defined as the use of hosted services, such as data storage, servers, databases, networking, and software over the internet. Since cloud computing began, the world has witnessed an explosion of cloud-based applications and services in IT, which continue to expand.

1.1 History of Cloud Computing :-

- Before Computing was come into existence, client Server Architecture was used where all the data and control of client resides in Server side.
- If a single user want to access some data, firstly user need to connect to the server and after that user will get appropriate access. But it has many disadvantages.
- It also has certain limitations. So in order to remove limitations faced in distributed system, cloud computing was emerged.

1.2 Characteristics of Cloud Computing

- **On-demand self-services**: The Cloud computing services does not require any human administrators, user themselves are able to provision, monitor and manage computing resources as needed.
- **Broad network access:** The Computing services are generally provided over standard networks and heterogeneous devices.
- **Rapid elasticity:** The Computing services should have IT resources that are able to scale out and in quickly and on as needed basis. Whenever the user require services it is provided to him and it is scale out as soon as its requirement gets over.
- **Virtualization:** Cloud computing providers use virtualization technology to abstract underlying hardware resources and present them as logical resources to users.

1.3 Types of Services in cloud computing

Cloud computing offers various types of services, commonly categorized into three main models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Each model provides different levels of control, flexibility, and management for users.

Infrastructure as a Service (IaaS):

IaaS delivers virtualized computing resources over the internet. It provides essential building blocks such as virtual machines, storage, networking, and other fundamental

computing resources. Users have more control over their infrastructure, allowing them to install operating systems, applications, and manage various components.

Platform as a Service (PaaS):

PaaS offers a platform allowing customers to develop, run, and manage applications without dealing with the underlying infrastructure. It provides a complete environment for developing, testing, and deploying applications, including tools, middleware, databases, and development frameworks. Users can focus on coding and deploying applications without managing the underlying infrastructure.

Software as a Service (SaaS):

SaaS delivers software applications over the internet on a subscription basis.Users can access these applications via a web browser without needing to install or maintain software on their devices. The software is hosted and managed by the service provider. Users typically pay for SaaS on a subscription or pay-as-you-go basis.

Chapter-2 AWS Academy Cloud Foundations

2.1 Introduction to AWS:-

Amazon Web Services (AWS) is a comprehensive and widely used cloud computing platform offered by Amazon.com. Launched in 2006, AWS provides a range of cloud services that include computing power, storage solutions, networking, databases, machine learning, analytics, Internet of Things (IoT), security, and more. These services are delivered on-demand, allowing businesses and individuals to access technology resources without the need to invest in and maintain physical infrastructure.

2.2 Introduction to AWS IAM :-

AWS Identity and Access Management (IAM) is a web service that enables Amazon Web Services (AWS) customers to manage users and user permissions in AWS. With IAM, you can centrally manage **users**, **security credentials** such as access keys, and **permissions** that control which AWS resources users can access.

I have done the following tasks in the IAM interface of AWS:

- Exploring pre-created **IAM Users and Groups**
- Inspecting **IAM policies** as applied to the pre-created groups
- Following a **real-world scenario**, adding users to groups with specific capabilities enabled
- Locating and using the **IAM sign-in URL**
- **Experimenting** with the effects of policies on service access.

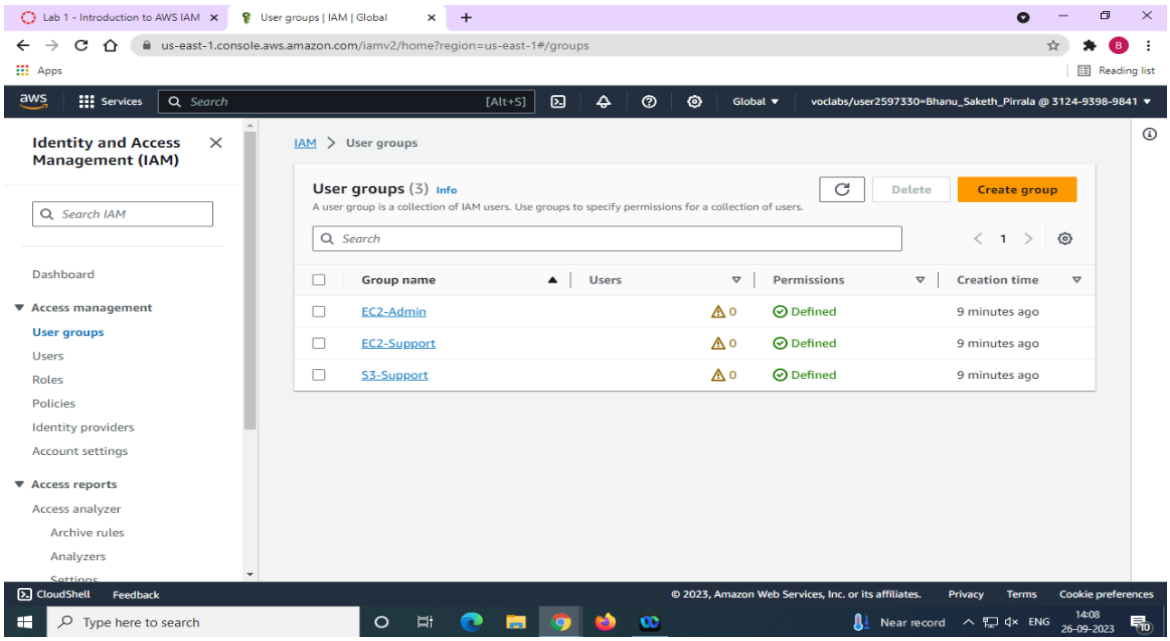


Fig 2.1 AWS IAM users and groups

2.3 AWS Virtual Private Cloud :-

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch Amazon Web Services (AWS) resources into a virtual network that you defined.This virtual network closely resembles a traditional network that you would operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

I have done the following tasks in the VPC interface of AWS:

- Create a VPC.
- Create subnets.
- Configure a security group.
- Launch an EC2 instance into a VPC.

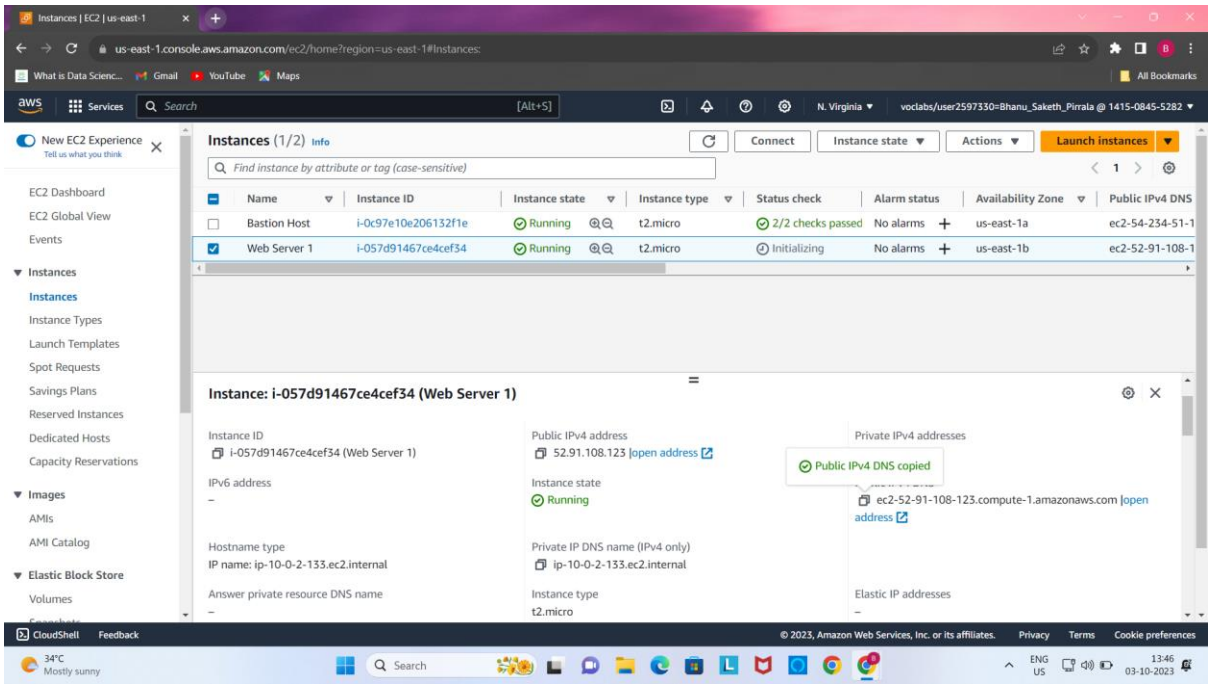


Fig 2.3.1 VPC Web Server

2.4 Amazon Elastic Compute Cloud and Amazon Lambda:

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment.

AWS Lambda is a serverless computing service provided by Amazon Web Services (AWS) that enables users to run code without provisioning or managing servers. With Lambda,

users can execute code in response to specific events, such as changes in data, HTTP requests, file uploads, database updates, or custom triggers, without worrying about server management.

I have done the following tasks in EC2 interface :-

- Launch a web server with termination protection enabled
- Monitor Your EC2 instance
- Modify the security group that your web server is using to allow HTTP access
- Resize your Amazon EC2 instance to scale
- Explore EC2 limits
- Test termination protection
- Terminate your EC2 instance

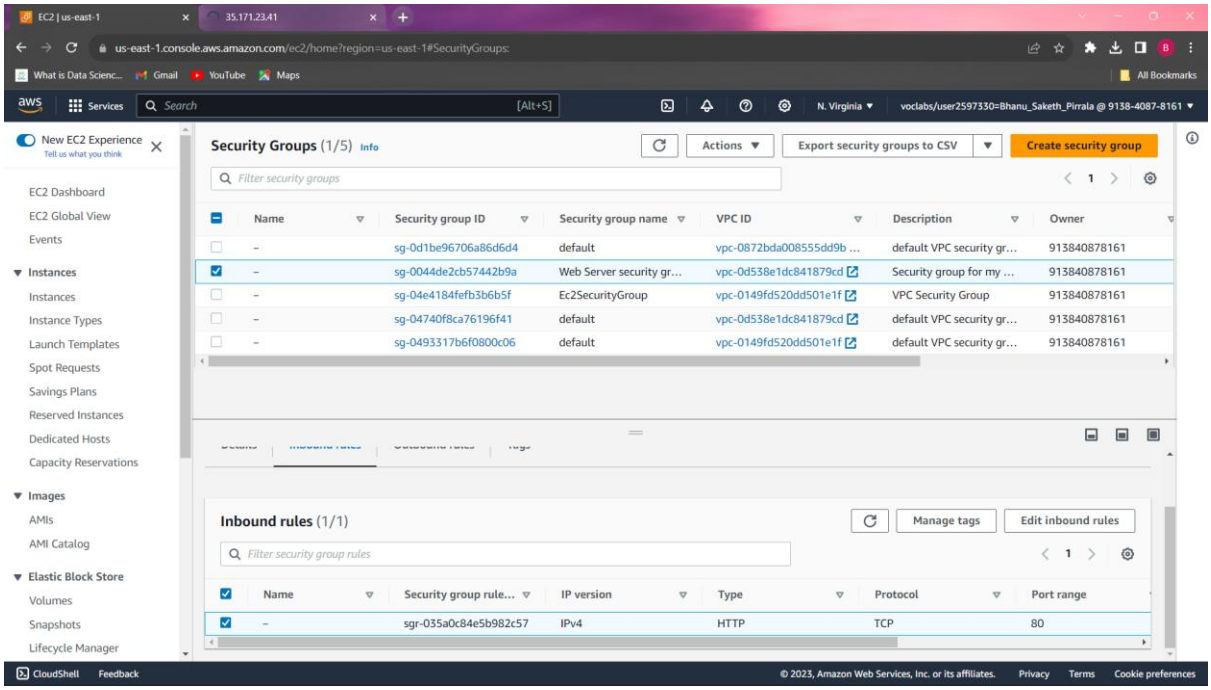


Fig 2.4.1 AWS EC2 Security Group

2.5 AWS EBS and AWS S3:

Amazon Elastic Block Store (EBS):

Description: EBS provides block level storage volumes that can be attached to EC2 instances. These volumes act like hard drives and are used for persistent storage.

Uses:

Primary storage for EC2 instances: EBS volumes are commonly used as boot volumes (where the operating system is installed) and for storing data that requires high-performance and low latency access.

Database storage: EBS volumes are suitable for database applications where consistent and low latency storage is essential.

Block Level storage: Useful for applications that require specific I/O operations and file systems.

Amazon Simple Storage Service (S3):

Description: S3 is an object storage service that allows you to store and retrieve data from anywhere on the web. It's highly scalable, durable, and designed for large scale storage of unstructured data.

Uses:

Data backup and archiving: S3 is commonly used for backup and long term storage of data.

Hosting static websites: S3 can host static websites by serving static HTML, CSS, JavaScript files, etc.

Media hosting and distribution: It's suitable for storing and serving media files like images, videos, and audio.

Application data and analytics: S3 can store application data and logs, making it a valuable resource for analytics and big data applications.

2.6 Amazon Relational Database Service

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, which allows you to focus on your applications and business..

I have done the following tasks :

- Launch an Amazon RDS DB instance with high availability.
- Configure the DB instance to permit connections from your web server.
- Open a web application and interact with your database.

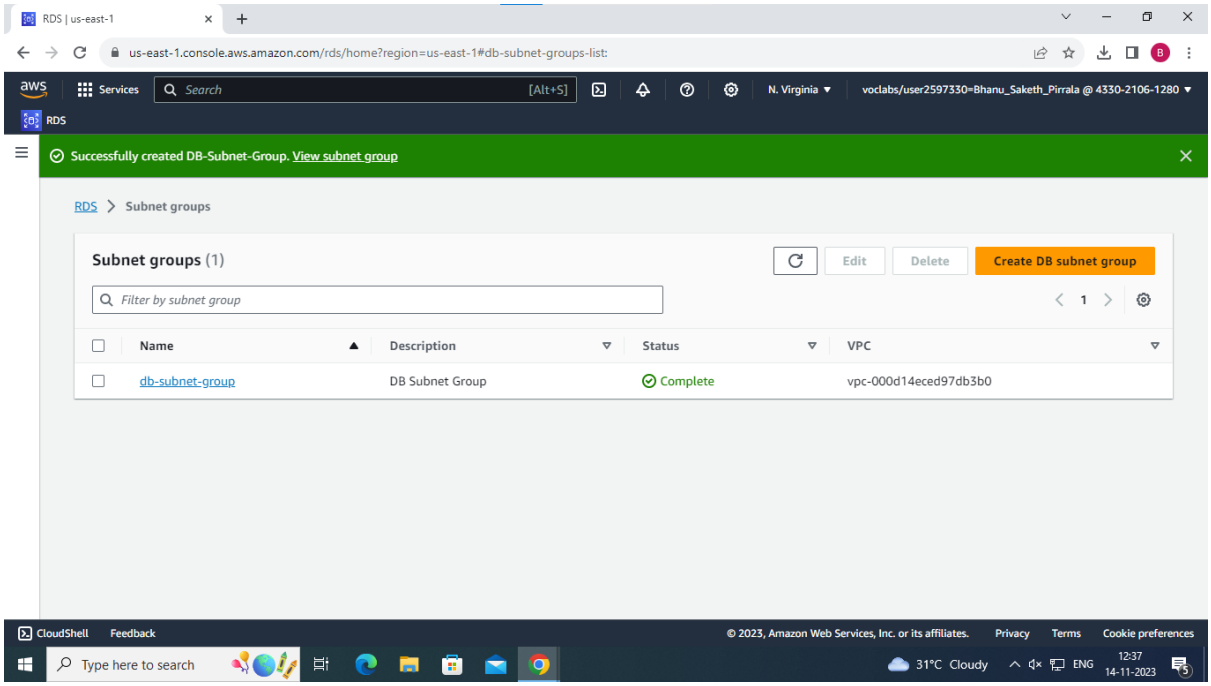


Fig 2.6.1 RDS Subnet Groups

2.7 Auto Scaling and Monitoring

Auto Scaling and Monitoring are crucial components in cloud computing for maintaining optimal performance, managing costs, and ensuring availability of resources.

Auto Scaling:

Auto Scaling is a cloud computing feature that automatically adjusts the number of compute resources (such as virtual machines, containers, or servers) based on varying workloads. It ensures that the application or system can handle fluctuations in traffic and demand without manual intervention.

How does Auto Scaling work?

Auto Scaling uses predefined policies or rules to dynamically add or remove resources based on metrics like CPU utilization, network traffic, or custom application metrics. When demand increases, it automatically scales out by adding more resources. Conversely, during periods of low demand, it scales in by removing excess resources.

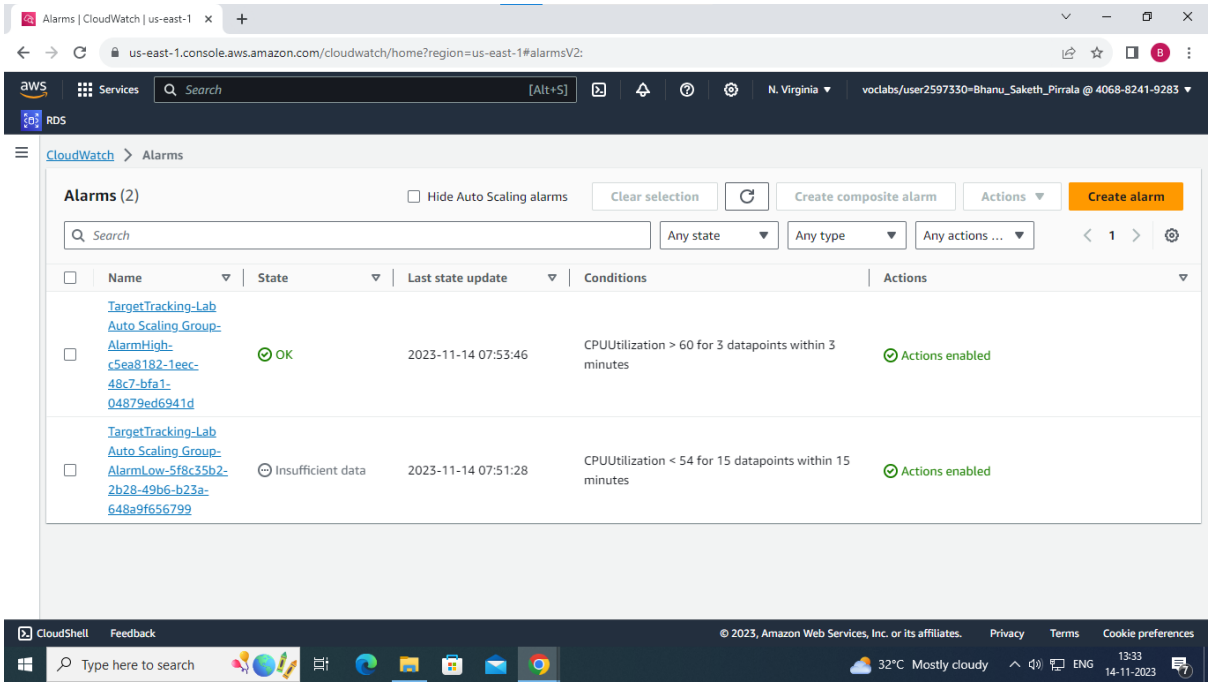


Fig 2.7.1 AWS CloudWatch

Monitoring

Monitoring involves the continuous tracking and observation of various metrics and performance indicators within a system or application. It helps in understanding system behavior, identifying issues, and ensuring efficient operation.

How does Monitoring work?

Monitoring tools collect and analyze data from various sources such as servers, applications, databases, networks, and user interactions. This data is used to generate insights, detect anomalies, and trigger alerts based on predefined thresholds or conditions.

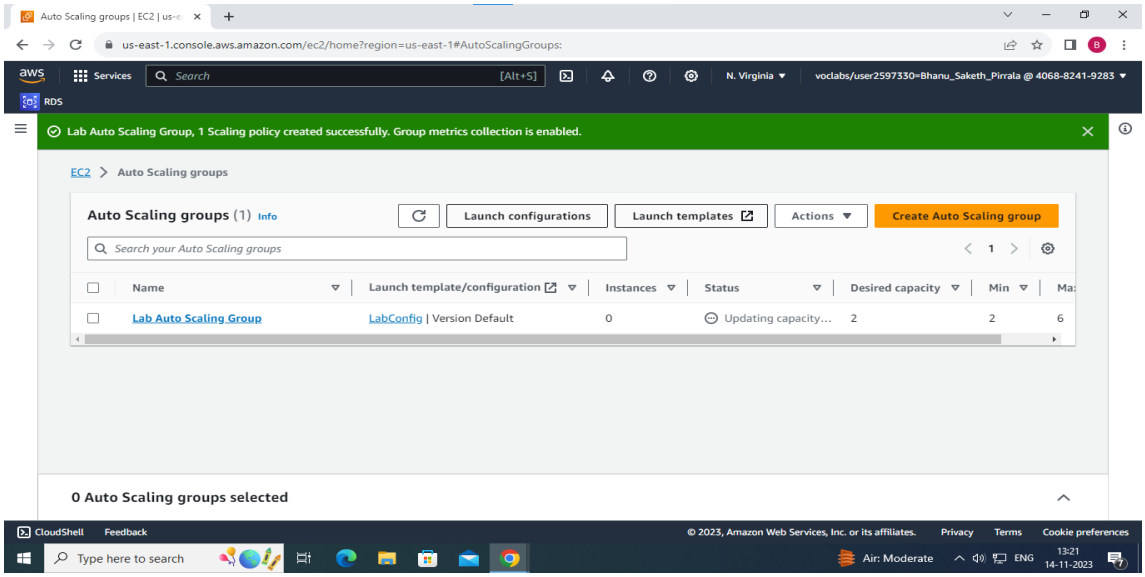


Fig 2.7.2 Auto Scaling groups

Chapter-3 AWS Academy Cloud Architecting

3.1 Introduction to Cloud Architecture:

Cloud architecture is the practice of applying cloud characteristics to a solution that uses cloud services and features to meet an organization's technical needs and business use cases.

You can use AWS services to create highly available, scalable, and reliable Architectures. Amazon Web Services (AWS) is a comprehensive and widely used cloud computing platform that offers a multitude of services for computing power, storage solutions, networking, databases, machine learning, and more. Its architecture forms the foundation for hosting and delivering a variety of applications and services on the internet.

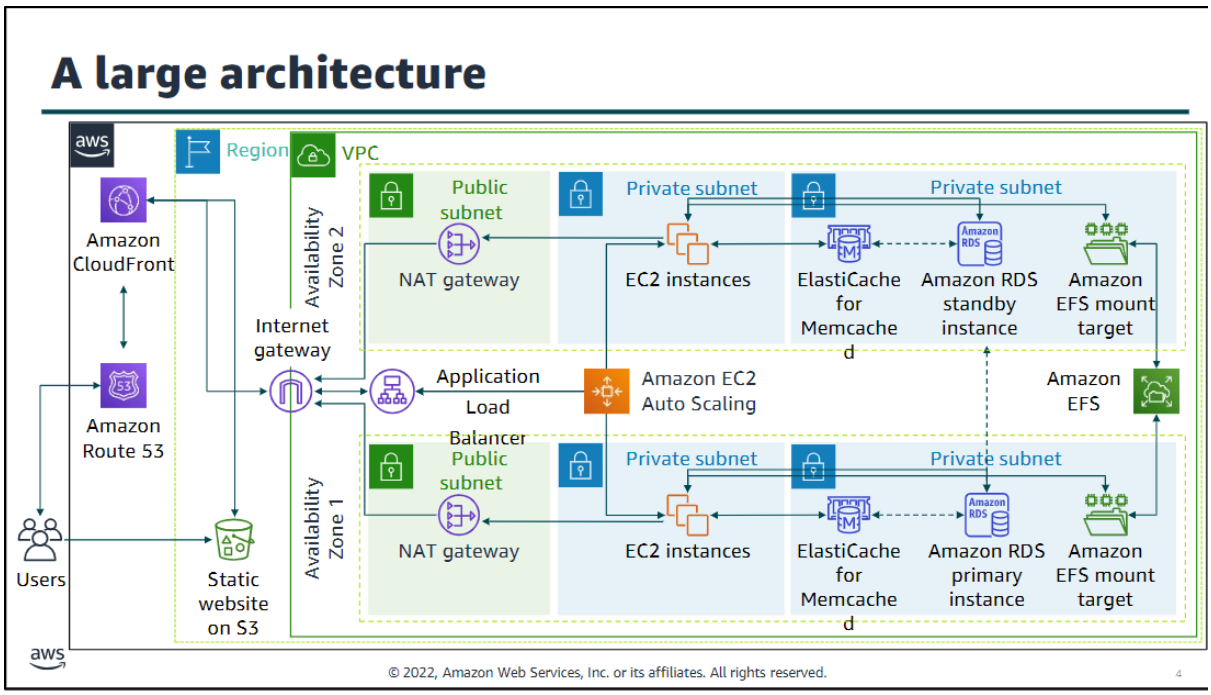


Fig 3.1.1 AWS Architecture

3.2 Architecture Principles and Best Practices:

- **Security:** AWS offers various security features and services like encryption, identity management, monitoring, and compliance tools to protect data and applications.
- **Scalability**: Utilize auto-scaling, load balancing, and AWS services that support scalability to meet changing demands and maintain performance.
- **Reliability:** Leverage multiple AZs, data redundancy, and fault-tolerant services to ensure high availability and minimize downtime.

- **Cost Optimization:** Employ AWS cost management tools, reserve instances, and choose the right pricing models to optimize costs based on usage.

3.3 Adding a Storage layer

Amazon S3 is an object storage service. It enables you to store virtually unlimited amounts of data. Data files are stored as objects. You place objects in a bucket, which you define. Every bucket must have name that is globally unique across Regions. This means that the bucket name must be unique across all AWS customer accounts.

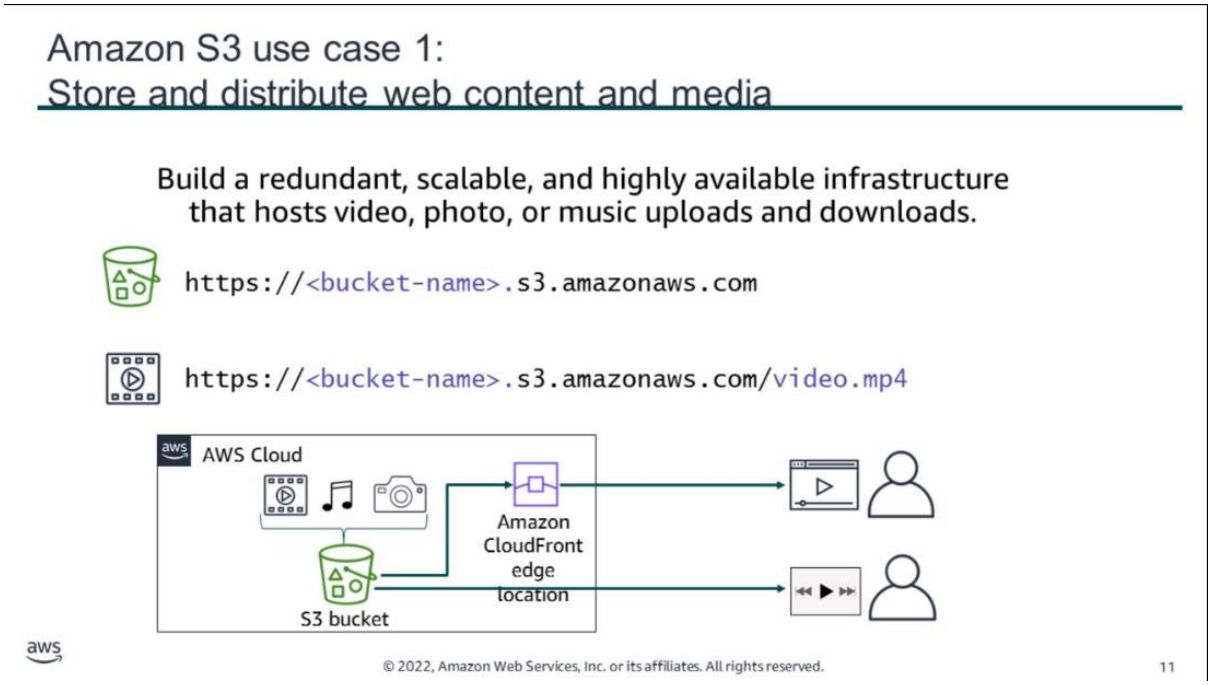


Fig 3.3.1 S3 Use Case

3.3.1 Hosting a Static Website

To host a static website, configure an S3 bucket for website hosting. Then, upload your website content to the bucket. The example shows that the static site might consist of HTML files, images, videos, and client-side scripts in formats such as JavaScript. With this approach, you do not need to run a virtual machine that hosts a web server.

In fact, you do not need to run a server. However, you can still host a website. Amazon S3 provides a low-cost solution for web hosting that includes high performance, scalability, and availability.

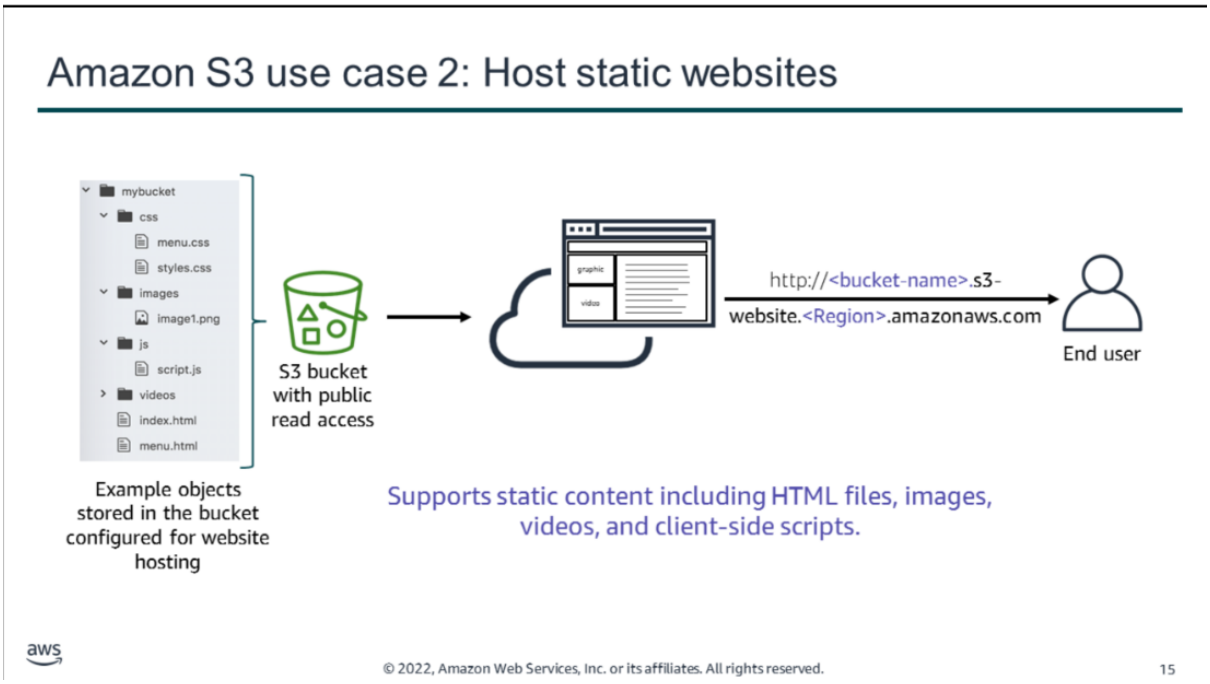


Fig 3.3.2 Static Website

3.4 Adding a Compute Layer

AWS offers several compute options to meet different needs. As you design the architecture to support a given type of workload, it is important that you understand the available compute options. As the diagram shows, the key runtime compute choices can be grouped into four cloud compute model categories: virtual machines (VMs); containers; platform as a service, which is also known as PaaS; and serverless.

It provides secure and resizable virtual servers in the cloud. The second service is Amazon Lightsail. It provides virtual private servers to run simple workloads in a cost-effective way.

3.4.1 EC2 Instance life Cycle

When an instance is first launched from an AMI, or when you start a stopped instance, it enters the pending state. This state indicates that the instance is being provisioned on a host computer and is booting. The instance type that is specified in the AMI, or for the original stopped instance, determines the hardware of the host computer for the new instance.

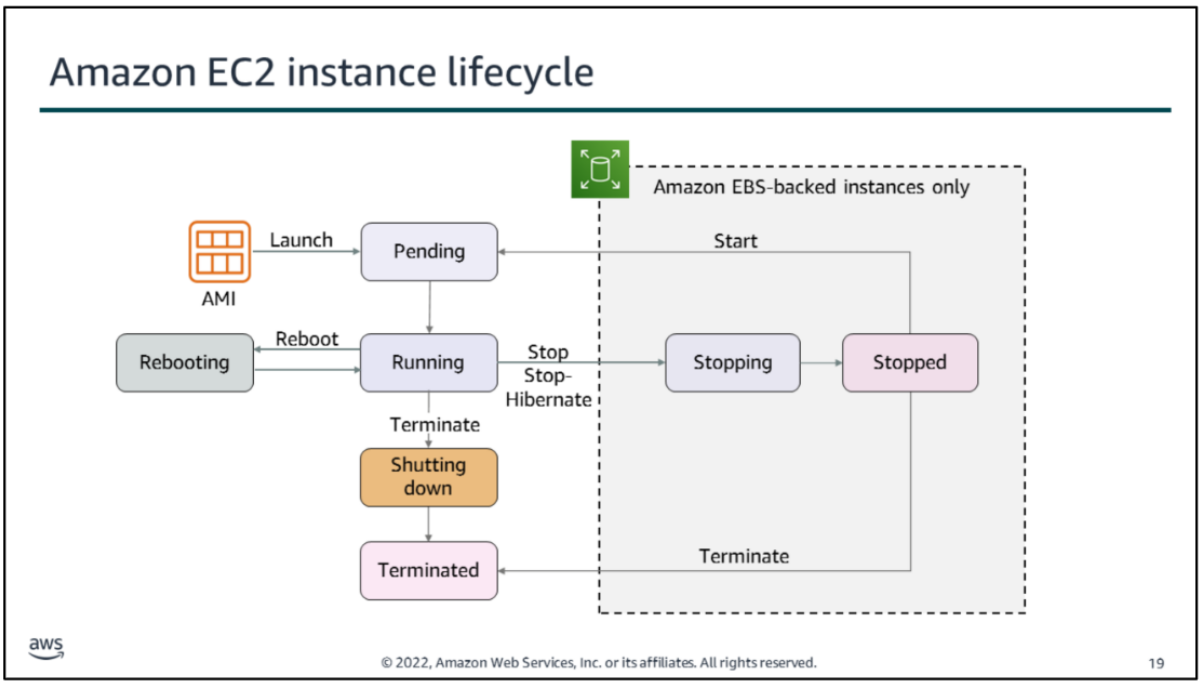


Fig 3.4.1 EC2 Lifecycle

3.4.2 Creating a Dynamic Website for the Cafe

Creating a dynamic website involves designing interactive and personalized user experiences. Choose a technology stack, set up a development environment, design a database schema, build the backend using a web framework, select a frontend framework, implement user authentication, and add dynamic content. Test thoroughly, deploy, and continuously update to ensure a scalable and secure web presence.

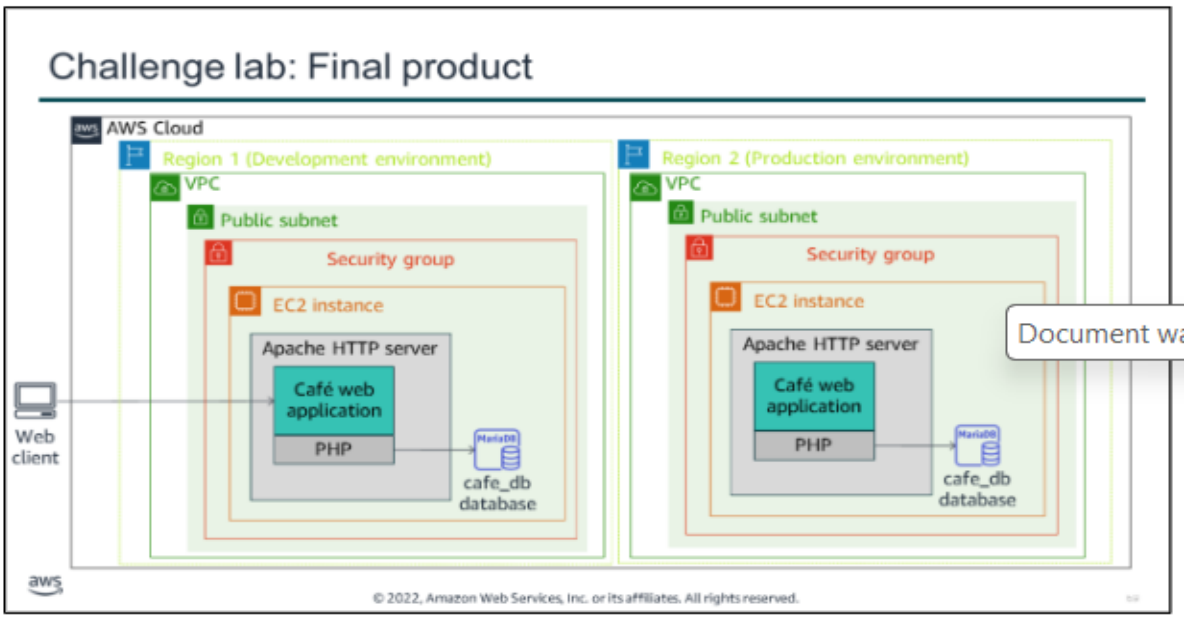


Fig 3.4.2 Dynamic Website

3.5 Adding a Database Layer

The data layer of your hybrid networking environment is important because it provides a data path for network traffic between applications hosted on AWS and an on-premises data center

3.5.1 Amazon RDS

Amazon RDS is a fully managed relational database service that creates and operates a relational database in the cloud. However, before you learn more details about Amazon RDS, you will first review the advantages of Amazon RDS as a managed database service.

3.5.3 Amazon DynamoDB

Amazon DynamoDB is a fully managed non-relational NoSQL database service. It provides fast and predictable performance with seamless scalability. DynamoDB works well for applications that handle a high volume of data and must scale quickly.
DynamoDB tables do not have fixed schemas, and each item can have a different number of Attributes.

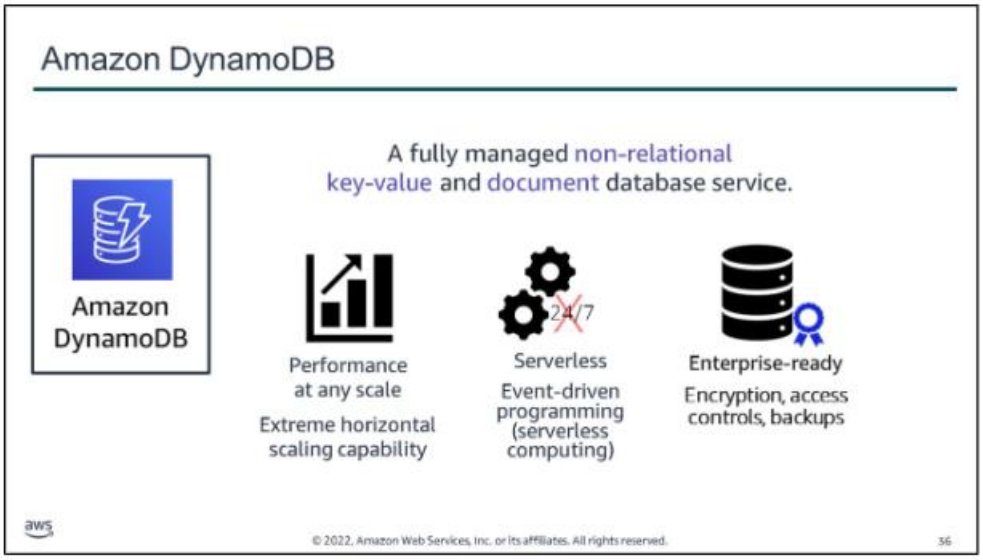


Fig 3.4.3 DynamoDB

3.6 Creating a Networking Environment

Creating a networking environment involves designing a robust infrastructure for seamless data flow. Establish an Amazon Virtual Private Cloud (VPC) with defined subnets, implementing Internet Gateways for external connectivity. .

3.6.1 Creating an AWS networking environment

Amazon Virtual Private Cloud (Amazon VPC) is a service that enables you to provision a logically isolated section of the AWS Cloud (called a virtual private cloud, or VPC) where you can launch your AWS resources.

Amazon VPC gives you control over your virtual networking resources. For example, you can select your own IP address range, create subnets, and configure route tables and network gateways. You can use both IPv4 and IPv6 in your VPC for secure access to resources and applications.

3.6.2 Connecting your AWS networking environment to the Internet

An internet gateway is a VPC component that allows communication between resources in your VPC and the internet. It is horizontally scaled, redundant, and highly available. An internet gateway supports IPv4 and IPv6 traffic. An internet gateway serves two purposes. First, it provides a target in your VPC route tables for internet-routable traffic.

Second, the internet gateway performs network address translation (NAT) for instances that were assigned public IPv4 addresses. To make a subnet public, you must first create an internet gateway and attach it to your VPC.

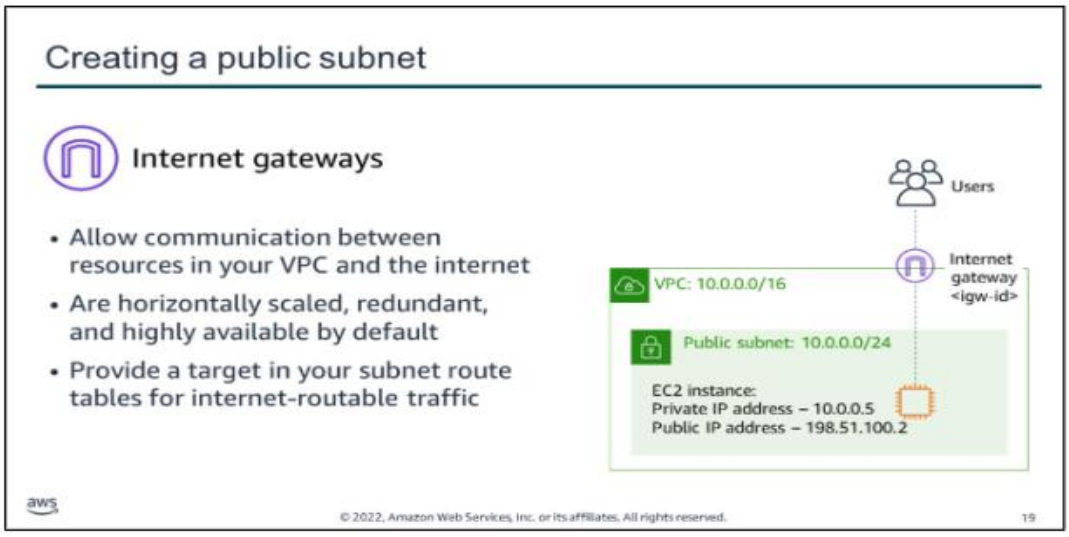


Fig 3.6.1 Creating public subnet

3.6.3 Securing your AWS networking environment

Security groups are stateful firewalls that act at the level of instance or network interface. Stateful means that return traffic is allowed is automatically allowed, regardless of any rules. For example, say that you initiate an Internet Control Message Protocol (ICMP) ping command to your instance from your home computer.

3.6.4 Creating a Virtual Private Cloud

Creating an AWS Virtual Private Cloud (VPC) involves establishing a private network within AWS. Define the VPC's IP address space, create subnets across multiple Availability Zones, and set up route tables for traffic control.

Attach an Internet Gateway for external connectivity and configure security groups to regulate inbound and outbound traffic. Implement Network ACLs for additional security at the subnet level. Designing a well-architected VPC ensures a secure and scalable foundation for hosting applications and resources in the AWS cloud.

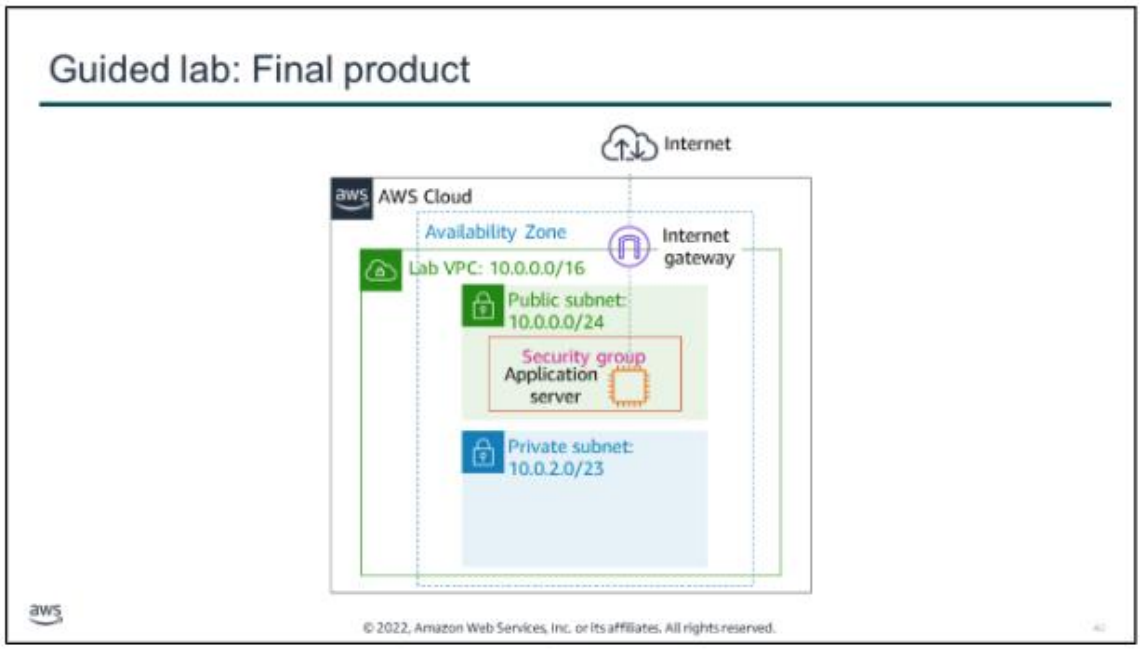


Fig 3.6.2 Final VPC layer

3.7 Securing User and Application Access

Securing user and application access in AWS involves employing robust identity and access management. Utilize AWS Identity and Access Management (IAM) to define roles and permissions, adhering to the principle of least privilege. Implement Multi-Factor Authentication (MFA) for an extra layer of user verification.

3.7.1 Account users and IAM

When you first create an AWS account, you begin with a root user. This user can log in to the AWS Management Console with the email address that was used to create the account. The AWS account root user has full access to all resources in the account, including billing information, personal data in the user profile, and all resources that were created in any AWS services in the account.

You cannot control the privileges of the AWS account root user credentials. AWS strongly recommends that you not use root user credentials for day-to-day interactions with AWS. Instead, create one or more IAM users. Keep the root user credentials in a secure location. For most ongoing account access and management tasks, you can use IAM user credentials.

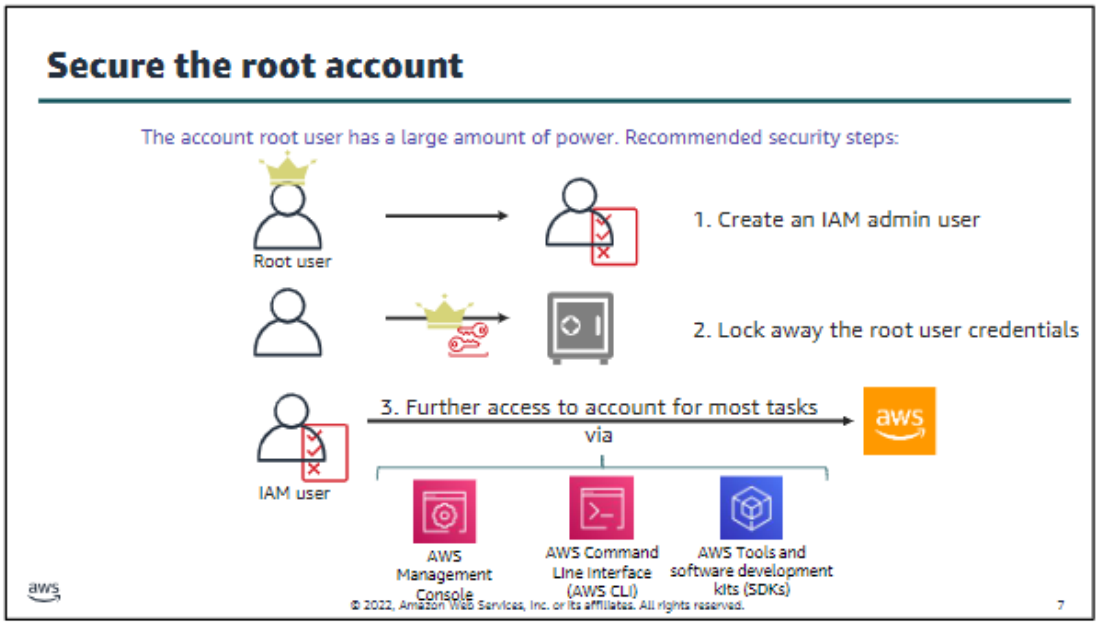


Fig 3.7.1 Secure the root account

3.7.2 Organizing Users

An IAM group is a collection of IAM users. Groups are a convenience that makes it easier to manage permissions for a collection of users, instead of managing permissions for each individual user.

3.7.3 Federating Users

An IAM role enables you to define a set of permissions to access the resources that a user or service needs. However, the permissions are not attached to an IAM user or group. Instead, the permissions are attached to a role, and the role is assumed by the user or the service.

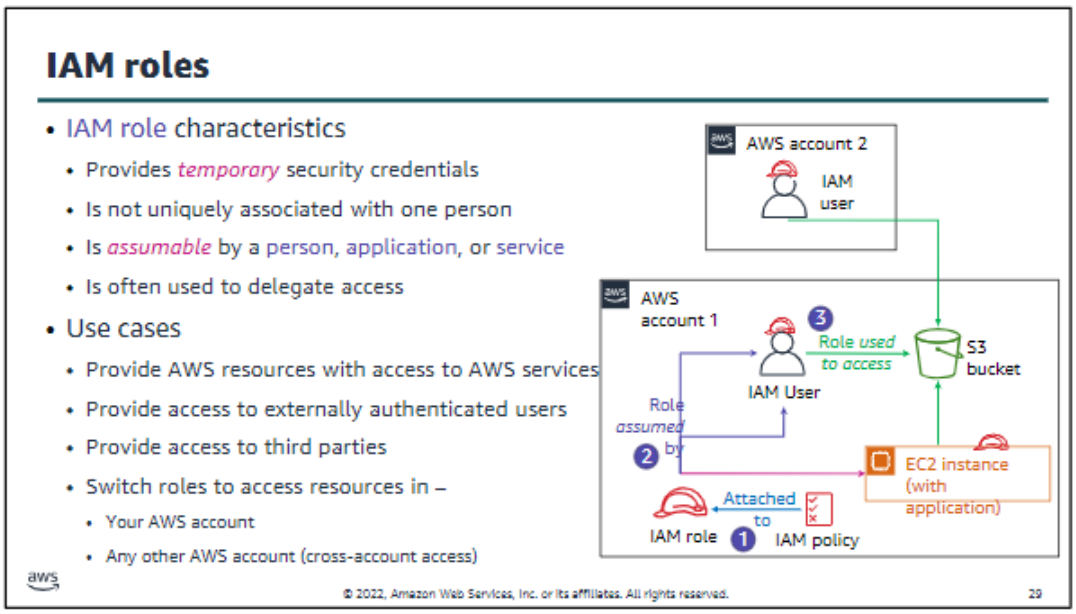


Fig 3.7.2 IAM roles

3.7.4 Controlling AWS Access by using IAM

Controlling AWS access via Identity and Access Management (IAM) is paramount for security. IAM allows precise management of user permissions, ensuring the principle of least privilege. Create roles and policies to grant access based on job responsibilities, and implement Multi-Factor Authentication (MFA) for an additional layer of user authentication.

IAM provides a central hub for monitoring and managing access, promoting a secure and well-governed AWS environment by limiting permissions and safeguarding against unauthorized actions.

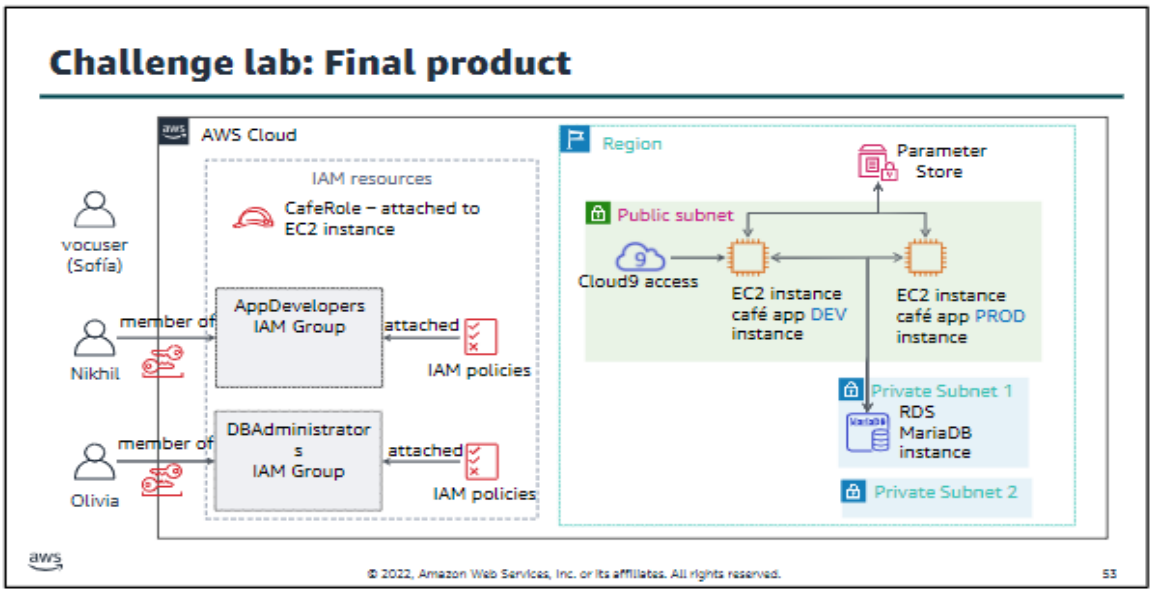


Fig 3.7.3 IAM final product

3.8 Automating Your Architecture

Automating your architecture in AWS streamlines operations and enhances efficiency. Leverage services like AWS CloudFormation or Terraform to define infrastructure as code, enabling consistent and repeatable deployments. Implement AWS Lambda for serverless automation of tasks, responding to events dynamically. Utilize AWS Step Functions for orchestrating workflows seamlessly.

3.8.1 Reasons to Automate

Manually creating resources and adding new features and functionality to your environment does not scale. If you are responsible for a large corporate application, there might not be enough people to manually sail the ship.

3.8.2 Automating your infrastructure

AWS CloudFormation provisions resources in a repeatable manner. It enables you to build and rebuild your infrastructure and applications without needing to perform manual actions or write custom scripts. You can think of this document as a model. You then use the model to create the reality, because AWS CloudFormation can actually create the resources in your account.

3.8.3 Automating deployments

AWS Systems Manager is a management service that is designed to be highly focused on automation. It enables the configuration and management of systems that run on-premises or in AWS. AWS Systems Manager enables you to identify the instances that you want to manage, and then define the management tasks that you want to perform on those instances.AWS Systems Manager is available at no cost, and it can manage both your Amazon EC2 and on-premises resources.

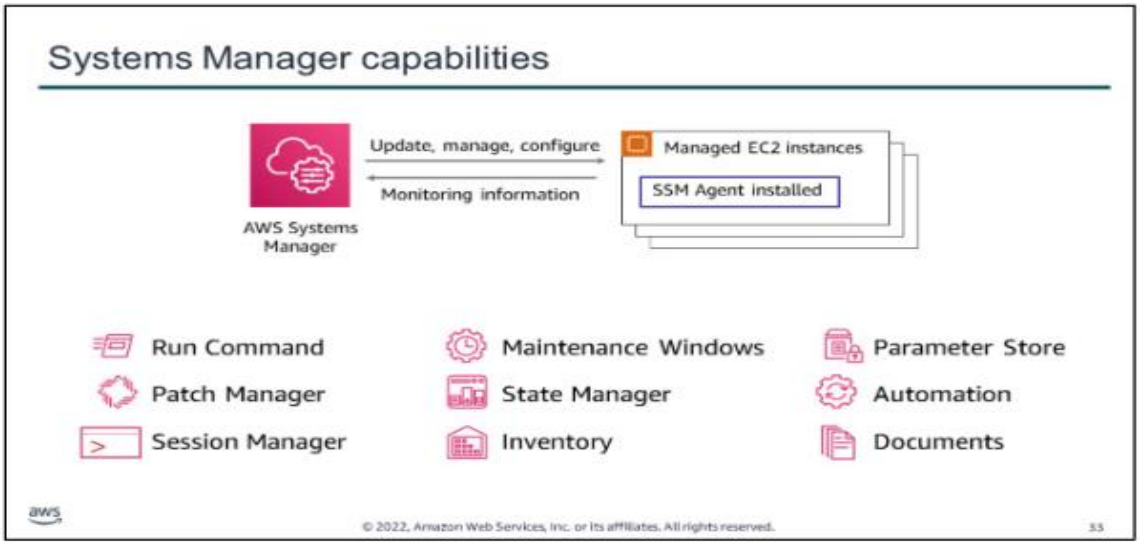


Fig 3.8.1 System Manager Capabilities

3.8.4 AWS Elastic Beanstalk

AWS Elastic Beanstalk is another AWS compute service option. It is a platform as a service (PaaS) offering that facilitates the quick deployment, scaling, and management of your web applications and services. It addresses many of the challenges that you just learned about.

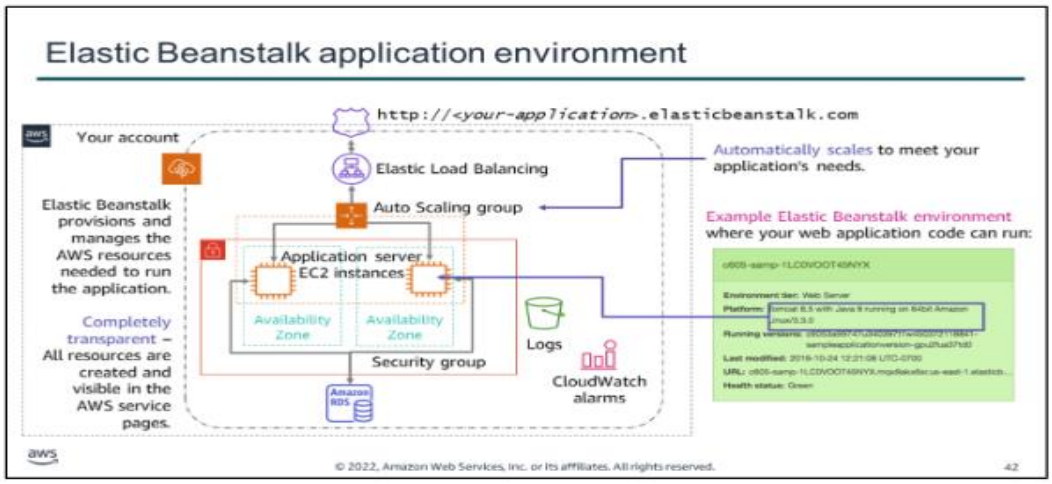


Fig 3.8.2 AWS EBS

You can choose from two types of environments when you work with Elastic Beanstalk. The single-instance environment enables you to launch a single EC2 instance and it does not include load balancing or automatic scaling. The other type of environment—which is in this example—can launch multiple EC2 instances, and it includes load balancing and an automatic scaling configuration. A managed database layer is optional.

3.9 Building Microservices and Serverless Architecture

Building microservices and serverless architecture in AWS empowers scalable, flexible, and efficient solutions. Microservices divide applications into independent, manageable components, enhancing development speed and maintenance. AWS Lambda enables serverless computing, eliminating the need for server management and scaling automatically based on demand.

3.9.1 Introducing Microservices

Microservices are an architectural and organizational approach to software development where applications are composed of independent services that

communicate over well-defined application programming interfaces (APIs). This approach is designed to speed up deployment cycles. The microservices approach fosters innovation and ownership, and improves the maintainability and scalability of software applications.

3.9.2 Building microservice applications with AWS container services

Containers are a method of operating system virtualization that enables you to run an application and its dependencies in resource-isolated processes. A container is a lightweight, standalone software package. It contains everything that a software application needs to run, such as the application code, runtime engine, system tools, system libraries, and configurations.

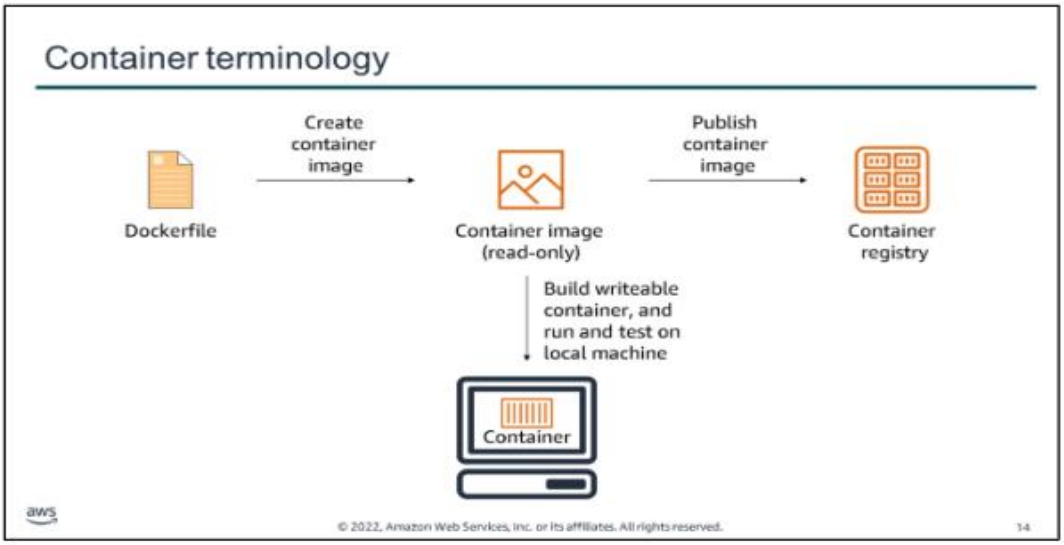


Fig 3.9.1 Container terminology

A container image is the snapshot of the file system that is available to the container. For example, you might have the Debian operating system as a container image.
When you run this container, a Debian operating system is available to it. You can also package all your code dependencies in the container image and use it as your code artifact.

3.9.3 Introducing Serverless architectures

Serverless is the native architecture of the cloud that enables you to shift more operational responsibilities to AWS, which can increase your agility and innovation.
Serverless enables you to build and run applications and services without thinking about servers.

3.9.4 Building serverless architectures with AWS Lambda

AWS Lambda integrates with other AWS services to invoke Lambda functions. A Lambda function is custom code that you write in one of the languages that Lambda

supports. You can configure triggers to invoke a function in response to resource lifecycle events, respond to incoming HTTP requests, consume events from a queue, or run on a schedule.

3.9.5 Implementing a serverless Architecture on AWS

Implementing a serverless architecture on AWS involves leveraging services like AWS Lambda to run code without managing servers. Break down applications into individual functions that execute on-demand, scaling automatically.

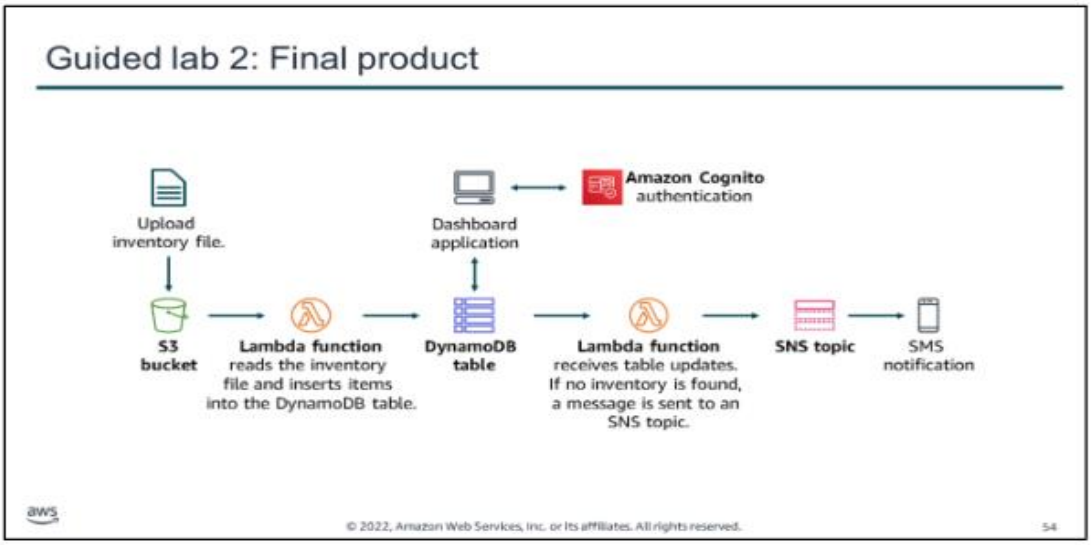


Fig 3.9.2 Serverless Architecture on AWS

References

[1] Mastering Cloud Computing Foundations and Applications Programming by RajkumarBuyya, Christian Vecchiola, S. ThamaraiSelvi, Morgan Kaufmann, 2013 .

[2] https://aws.amazon.com/training/awsacademy/

[3] Amazon Elastic Compute Cloud “Documentation” available at << http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html >>

[4] Amazon Simple Storage Service “Documentation” available at << http://docs.aws.amazon.com/AmazonS3/latest/gsg/GetStartedWithS3.html >>

[5] Amazon Virtual Private Cloud “Documentation” available at << http://docs.aws.amazon.com/AmazonVPC/latest/GettingStartedGuide/getting-started-ipv4.html >>

[6] Amazon Relational Database Service “Documentation” << http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html >>

[7] Amazon Simple Notification Service “Documentation” << http://docs.aws.amazon.com/sns/latest/dg/welcome.html >>

[8] Amazon CloudWatch“Documentation” available at << http://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html >>

[9] Elastic Load Balancing “Documentation” available at << http://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/what-is-load-balancing.html >>

[10] Auto Scaling “Documentation” available at << http://docs.aws.amazon.com/autoscaling/latest/userguide/WhatIsAutoScaling.html >>