Internship Project Progress Report on

INFRASTRUCTURE SETUP AMD APPLICATION DEPLOYMENT USING AWS AND DEVOPS TOOLS

Submitted in partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING (Specialization in Cloud Computing)

Submitted by:

SOMAN KUMAR **2015024**

Under the Guidance of

Project Team ID: ID No.: MP22CCIS05

Project Progress Report No: 1

Department of Computer Science and Engineering
Graphic Era (Deemed to be University)
Dehradun, Uttarakhand
April, 2022-23

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project progress report entitled **"Infrastructure Setup and Application Deployment using AWS and DevOps Tools"** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering **(Specialization in Cloud Computing)** in the Department of Computer Science and Engineering of the Graphic Era (Deemed to be University), Dehradun shall be carried out by the undersigned under the supervision of **mentor/supervisor**

Soman Kumar 2015024

The above-mentioned student shall be working under the supervision of the undersigned on the **"Infrastructure Setup and Application Deployment using AWS and DevOps Tools"**

(Signature)

Supervisor/mentor

Examination

Name of the Examiners: **Signature with Date**

1.

2.

Work Satisfactory: Yes / No

Table of Contents

Chapter 1

Introduction and Problem Statement

1.1 Introduction

Hays plc is a British multinational company providing recruitment and human resource services across 33 countries globally. Hays business solution is a captive unit which is located in Gurugram and Noida. It supports its parent company in the IT delivery services and in ITEs and RPO in UK and America. In this internship I got the opportunity to explore about the DevOps technology, learned how to setup infrastructure and deploy app in the cloud server using various tools like AWS and other DevOps service. With the increasing demand for faster and more reliable software delivery, there is a need for an automated and streamlined approach to software development and deployment. This is where DevOps comes in.

1.1.1 DevOps

DevOps is a set of practices and tools that aims to automate the software development lifecycle and ensure that the applications are delivered faster and with higher quality. One of the key aspects of DevOps is infrastructure automation, which involves setting up and managing the infrastructure required to run the application.

1.1.2 AWS

AWS is one of the leading cloud service providers that offers a wide range of services for infrastructure setup and application deployment. AWS is a cloud computing platform that offers a wide range of infrastructure and application services to support the entire software development lifecycle. AWS provides an extensive suite of tools and services that can be used to build, test, deploy and manage applications at scale, making it a popular choice for many organizations that have adopted DevOps practices.

Some of AWS services include Amazon EC2, AWS IAM, AWS Autoscaling Groups, AWS load balancer, AWS S3 and AWS Elastic Beanstalk.

These services enable organization to build and manage highly scalable, and reliable applications with ease, making AWS an essential tool for any DevOps team.

1.1.3 Jenkins

Jenkins is a widely used automation server that helps in building, testing and deploying software applications. Jenkins automates the entire software development process, from code building to deployment, making it an essential tool in the DevOps workflow. It enables developers to automate repetitive tasks, improve code quality and reduce time to market. Jenkins supports various programming languages, build tools, and testing frameworks, making it a versatile tool that can be used in any development environment. Jenkins plays a critical role in the DevOps workflow by automating the entire software development process, improving collaboration between teams, and reducing the time and effort required for building, testing and deploying the applications.

1.1.4 Git

Git is a widely used version control system that helps in managing source code and collaborating on software development projects. It allows multiple developers to work on the same codebase simultaneously and keep track of changes made to the code. Git provides a centralized repository where all changes to the code are stored, and developers can easily track the progress of the project, resolve conflicts, and collaborate with other team members.

1.2 Problem Statement

As a part of DevOps training program, main focus was on leveraging DevOps technology and AWS to set up necessary infrastructure and deploy a web application. The project work includes creating virtual machines using ec2, conFiguring load balancers, setting up auto-scaling groups, and deploying a sample web application We also aim to use various DevOps tools like Jenkins, git and docker to automate the development process, improve collaboration and increase efficiency. The end goal of this project is to provide hands on experience in DevOps technology and demonstrate the benefits of leveraging AWS for infrastructure setup and application deployment.

Chapter 2

Objectives

The proposed work objectives are as follows: -

- The main objective of this internship training is to get the benefit from the experience which we will get by learning how development and operations happens in an organization, how a project is deployed in the webserver using various DevOps tools.

- To develop a web application and to design and implement a scalable and reliable infrastructure setup using AWS services, such as EC2, S3, Elastic load balancer, Autoscaling groups, Elastic block Storage, that supports continuous integrations, continuous delivery and continuous deployment.

- To develop a comprehensive DevOps process that integrated with AWS services, including git for version control and Jenkins for continuous integration and delivery.

- To implement security best practices, such as using AWS identity access management (IAM) to control access to AWS services and encryption data at rest and in transit, to ensure security of the infrastructure and application.

- To establish monitoring and alerting system using AWS cloud watch and AWS cloud trail, which provide real-time visibility into the health and performance of the DevOps infrastructure and application

Chapter 3

Project Work Carried Out

3.1 Planning and Designing

Planning the requirement analysis is the most important and fundamental stage to carry a project in an organization. This step is mainly performed by the senior members of the team with the inputs from the customer/clients, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas. This is the very first step of the project where we used to plan and design the infrastructure setup and DevOps process after the requirement analysis is done. The very first step in this is developing a web application using the desired framework then using git and GitHub by creating a repository and storing our application there. Later creating a scalable infrastructure to deploy our application. This involves identifying the AWS services required, defining the architecture of the infrastructure, and outlining the steps needed to automate the deployment process.

Work Flow of our infrastructure and Workflow of our Sample Web application

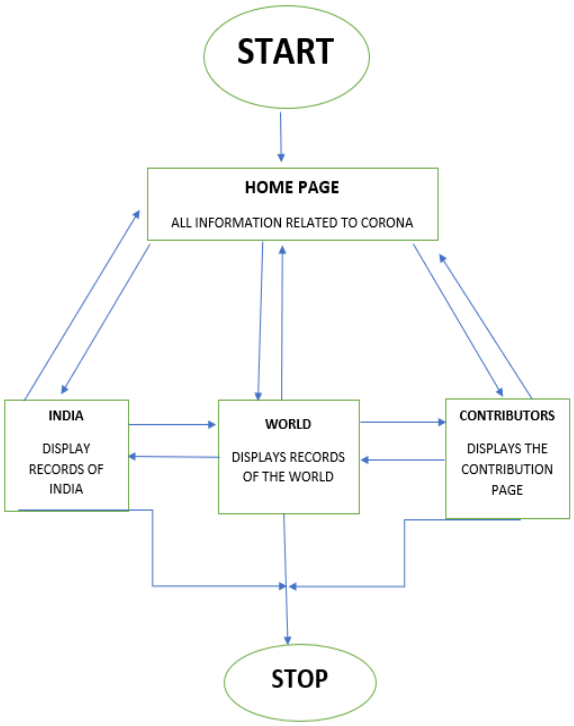

Fig 3.1.1 Workflow of Web-App

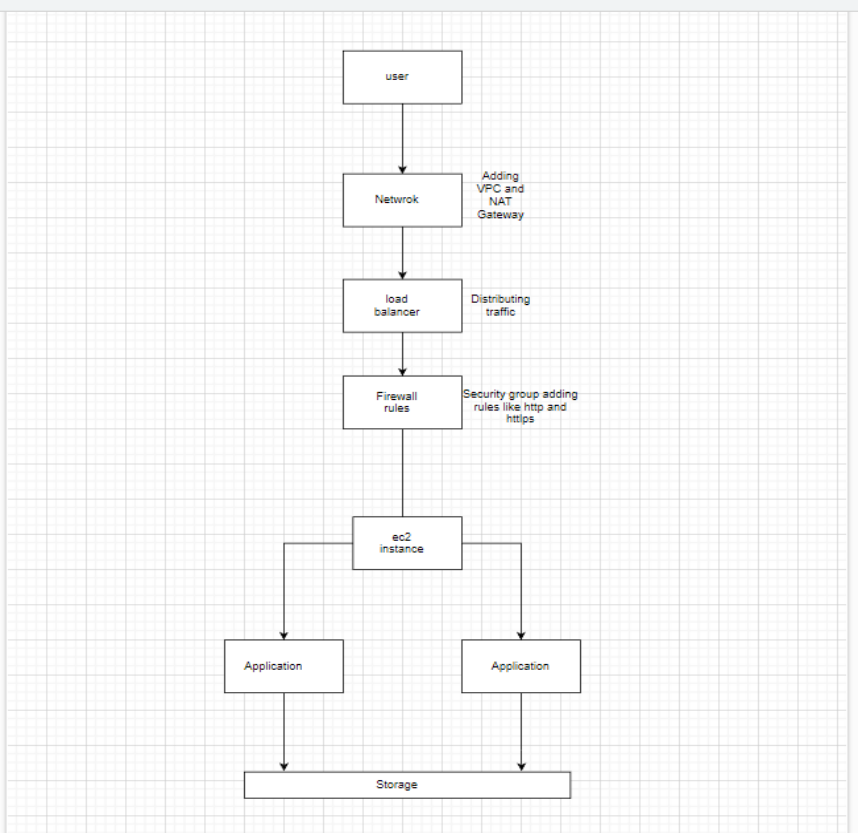

Fig3.1.2 Workflow of Infrastructure

3.1.1 Learned about Software Development Approach and Various SDLC model.

In the planning and designing step, further learned about various SDLC models i.e., software development life cycle model which include models such as waterfall model, incremental model, code and fix model. V-shape model, spiral model, prototyping model which included both throwaway prototyping and evolutionary prototyping model, the agile methodology and etc. To know more about DevOps and how it came, it was important to understand previous models in software development approach.

So, the very first stage was the water fall model, which is a software development approach where the life cycle of the software is divided into sequential phases. It follows one-way downward flow of information.

Then came the agile model, in the agile methodology each project is broken up into several iterations. All iterations should be of the same time duration (between 2 to 8 weeks). At the end of each iteration, a working product should be delivered.

Both models had various disadvantages and solution to that was the DevOps approach which was also a software development approach which involves both development and operations.

3.2 Learned About Version Control System

A version control system (VCS) is a software tool that helps in tracking and managing changes to source code and other files. It allows multiple people to work on the same codebase simultaneously, without the risk of overwriting each other's changes. It enables developers to collaborate effectively, keeping the track of all the changes made top the codebase and easily revert to earlier versions if needed.

Through this got to know about the source code management system that is git which is a free and open-source distributed version control system which can be used to manage major and minor project with speed and efficiency. Using git, we used to push the code or application which is been made to the GitHub, GitHub is an open-source code hosting platform for version control and collaboration. Used various git commands such as git init, git add . , git push , git pull , git clone and other merging , branching and rebasing commands to successfully push the code to the GitHub.

We used git because it provides certain features like it is distributed which is it allows distributed development of code and every developer has a local copy pf the entire development history and changes are copied from one repository to another, it is compatible, it supports branching i.e., it takes only a few seconds to create and merge branches, it is lightweight more reliable and secure.

3.3 Infrastructure Setup

In this step I learned about how to set up the infrastructure using AWS services such as EC2, S3, RDS elastic load balancing and autoscaling group. The very first thing was creating an AWS account and setting up the necessary access and permissions. Later I used AWS ec2 (elastic compute cloud) for creating an instance **(Fig 3.3.1)** that will be used for hosting the web application, the configuration may vary according to the need of client such as in my case I was using a configured AMI of windows server machine. Created a S3 bucket **(Fig 3.3.2)** to store files as it is an object storage service which is highly scalable and durable solution designed for storing and retrieving any type of data such as photos, videos, documents, and

application backup which later can be used in our ec2 instance. Then I have created a RDS (Relational database server) which is a fully managed relation database service that provides easy deployment, management and scaling of MySQL, PostgreSQL and etc. databases. Used VPC **(Fig 3.3.3)** and subnets for the infrastructure and set up few security groups for the VPC as it is used to create a secure and isolated network environment for deploying our application. VPC stands for virtual private cloud is a virtual network environment that allows users to launch resources such as EC2 instance, RDS database and other services in a virtual network. Lastly, we created few autoscaling groups **(Fig 3.3.4)** and load balancers **(Fig 3.3.5)** for our ec2 server for scalability and reliability of our application.

3.4 Setting up Web application

After setting up the necessary infrastructure and configuring the required services, the next step was to develop the web application and then deploy it. We have used various framework to develop the web application. We have created a covid 19 tracking system which is used to track the data from the API and display the record on the webpage. This system provides all the useful and necessary information related to coronavirus and displays all the records of no. of people affected, dead, active, recovered and etc. on daily bases. The source code of the web application is stored in a version control system like GitHub, **(Fig 3.4.1).** Later using various git command, it could be fetched to the instance using various git command.
Sample code for Covid-19 Tracking System is

Table 3.1. Code of Implementation

```
from flask import Flask , render_template,redirect,url_for
import requests

app=Flask(__name__)

@app.route('/')

@app.route('/Information')
def Information():
   return render_template("information.html")



@app.route("/home")
```

```
def home():
    return redirect(url_for('covid'))


@app.route("/covid")
def covid():


    data =
requests.get("https://disease.sh/v2/countries/India?yesterday=true&strict=true")
    data_dict = data.json()
    return render_template("home.html",data=data_dict)

@app.route('/world')
def world():

    world = requests.get("https://corona.lmao.ninja/v2/all")
    world_data = world.json()
    return render_template("world.html",world=world_data)


@app.route('/contribute')
def contribute():
    return render_template("contribute.html")


if __name__=="__main__":
    app.run(host="0.0.0.", port=80)
```

3.5 Deploying the Web Application

We can use elastic beanstalk to deploy the web application. So, for that I configured the necessary settings such as the application version, environment type and instance type. Also set up the database connection if required and configure the load balancer to distribute the traffic across the instances. It had certain disadvantages such as having limited control over the underlying infrastructure and has a complex configuration.

We used AWS EC2 instance and in that we uploaded our application code using the git from the GitHub repository **(Fig 3.4.1)** . Configured the security group by adding inbound rule that is adding http and https to it, also configuring the firewall rules to allow traffic to our application **(Fig 3.5.1)**. Setting up the necessary software components on the EC2 instance, including web server, application server, and database client software. Connecting our instance to the S3 using CLI if we want few files which were uploaded on S3. Configuring the web application to use the database. The VPC used for both EC2 and RDS is same as this help them to communicate easily with each other. Installing the database client libraries to our instance same as on the database engine. In such case we used a SQL database and hence established a connection between them. After the connection is established, we are able to use RDS database from the EC2 instance. Once all the application files are uploaded, we started our application using the necessary commands or scripts **(Fig 3.5.3).** We also attached Elastic block storage **(Fig 3.5.6)** , load balancers and auto scaling group **(Fig 3.5.4 and Fig 3.5.5)** to our instance because by using an autoscaling group and load balancer, we ensure that our application is available. Autoscaling group allow us to automatically add pr remove instances based on the traffic to our application. Load balancing distributes the incoming traffic across multiple instances, which helps to distribute the load and improves the overall performance of our application. The autoscaling group and load balancer ensure that our application is fault-tolerant and if any instance fails the load balancer will redirect the traffic to other healthy instances, thereby reducing the impact of the failure on our application. At the end we can see that our web application is successfully running on the following ec2 server. (Fig 3.5.7a to 3.5.7e)

3.6 Security Setup

To implement security best practices, such as using AWS identity and Access management (IAM) to control access to AWS services and encrypting data set at rest and in transit, to ensure the security of the infrastructure and application. Planned and design IAM for our infrastructure on AWS. IAM stands for Identity and Access Management, which is a service provided by AWS to manage access to various resources in AWS account. With IAM we were able to create and manage users, groups and permissions to access AWS resources including EC2 instances , S3 bucket and RDS instances. IAM provides various security aspects such as it allows us to control who can access the AWS resources and what actions they can perform.

3.7 Code and Screenshot of AWS dashboard and Application

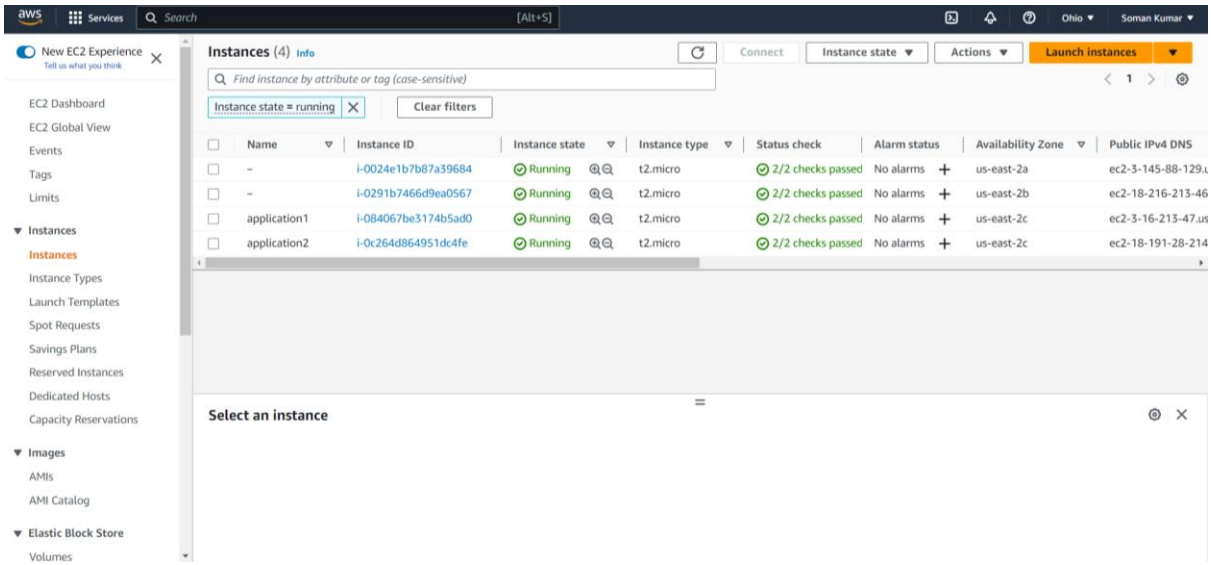

Fig 3.3.1 EC2 Dashboard

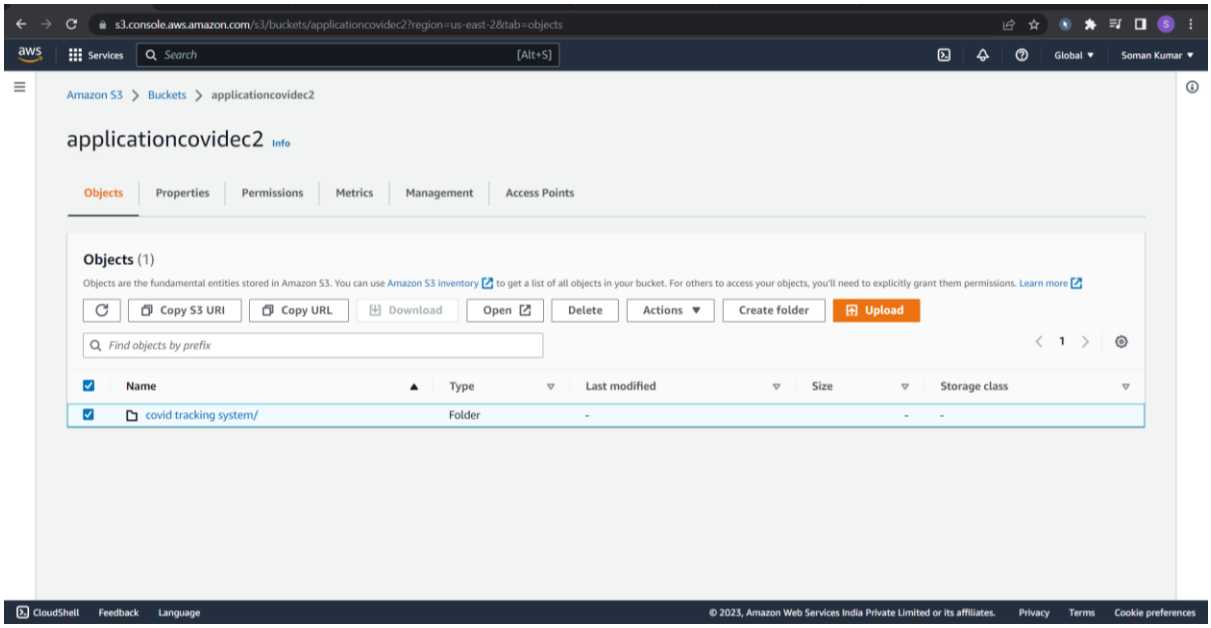

Fig 3.3.2 Created S3 Bucket and Stored File

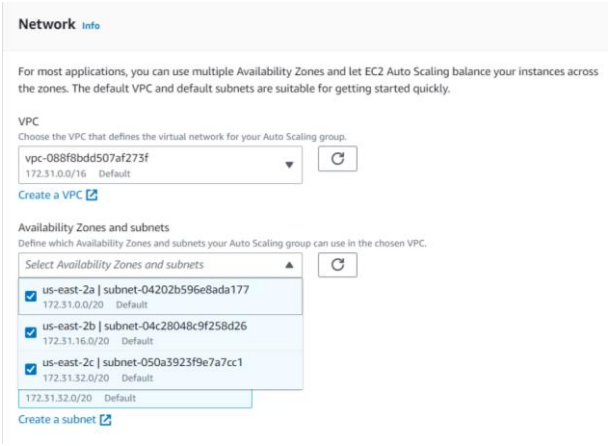

Fig 3.3.3 ConFiguring VPC

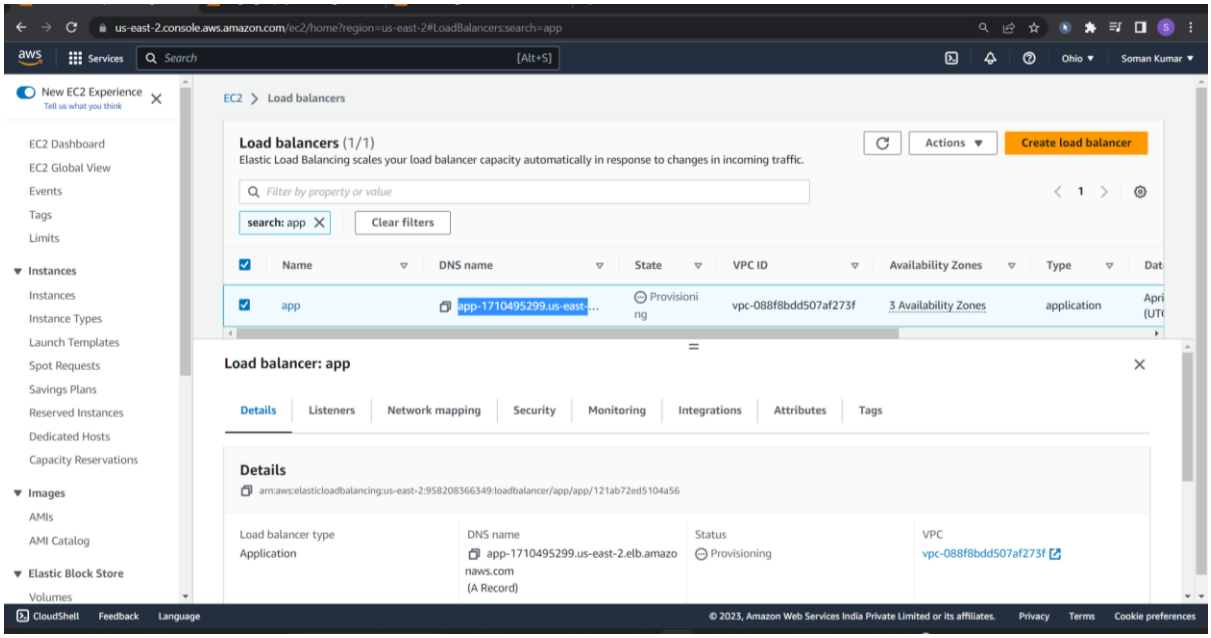

Fig 3.3.4 Setting up load balancers

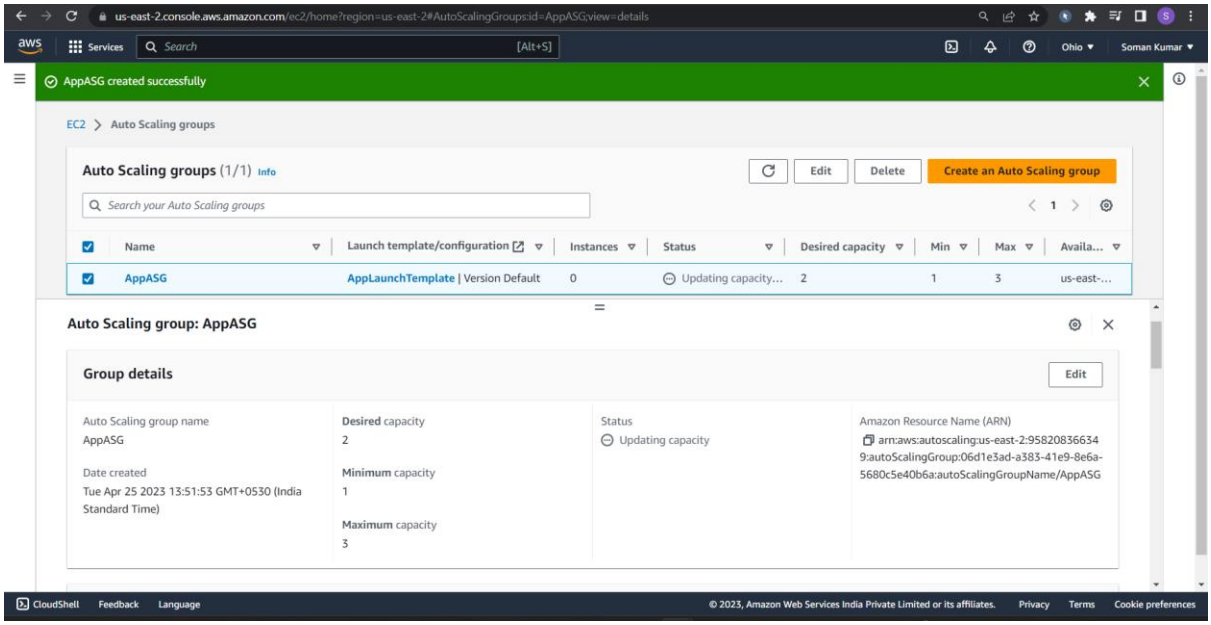

Fig 3.3.5 Setting up autoscaling group

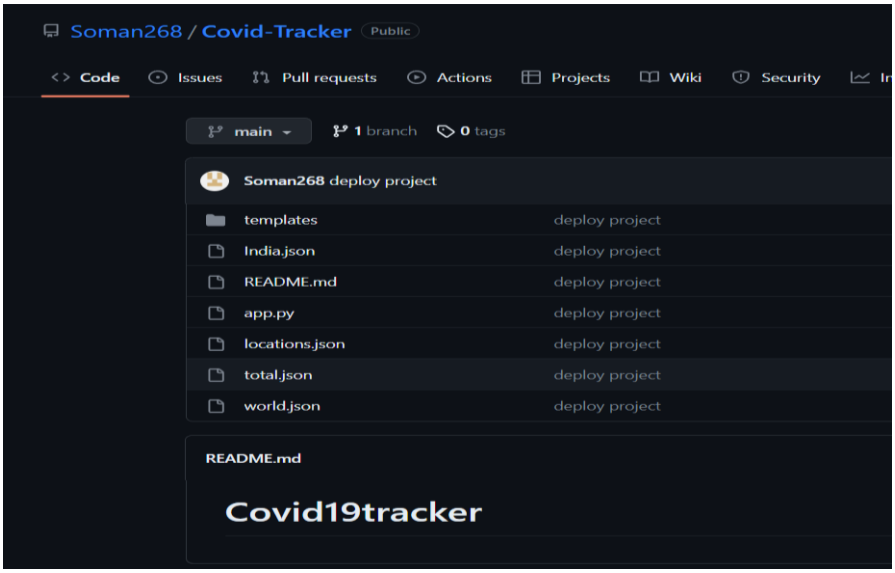

Fig 3.4.1 GitHub Repository of the Application

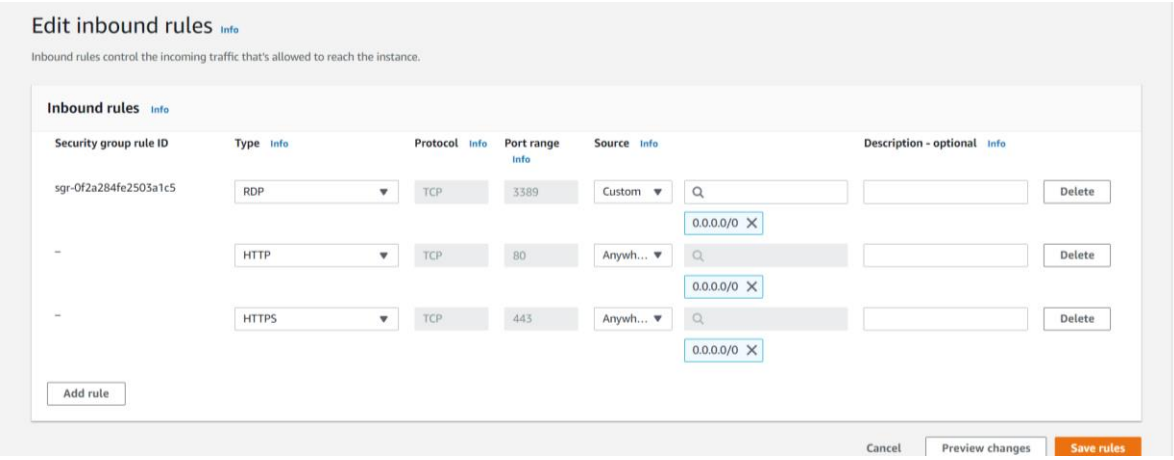

Fig 3.5.2 Adding inbound rules for http and https

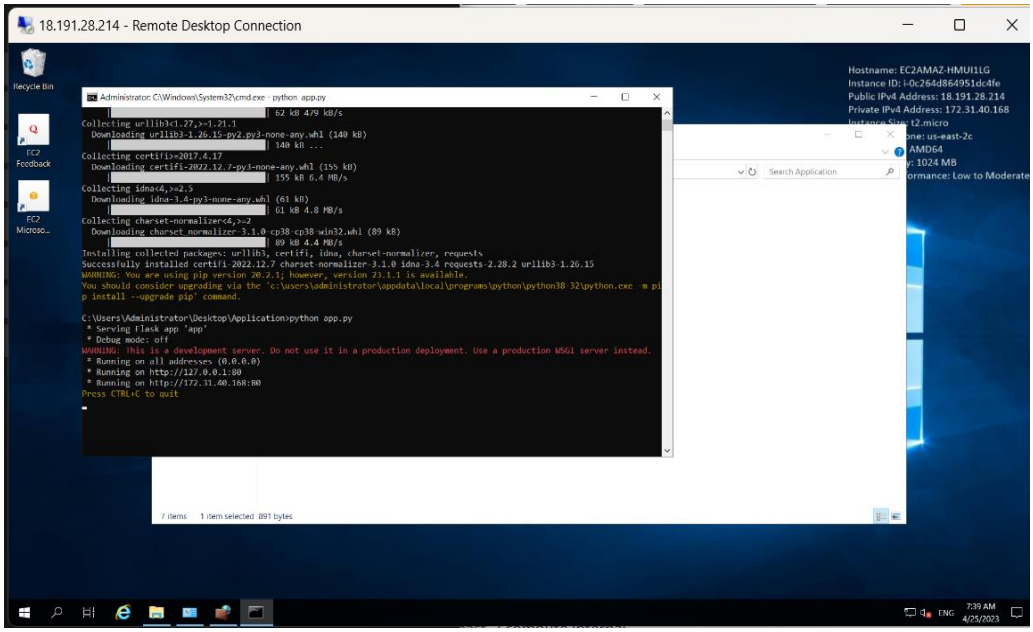

Fig 3.5.3 Running the recommended Scripts

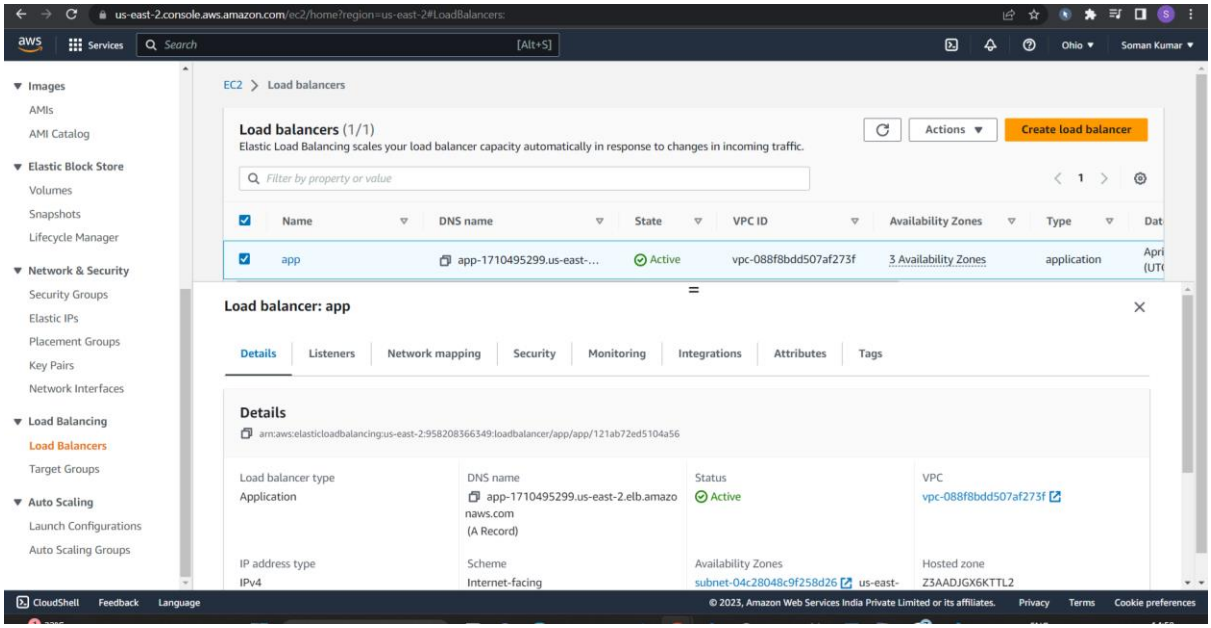

Fig 3.5.4 Adding load balancer to our application instance

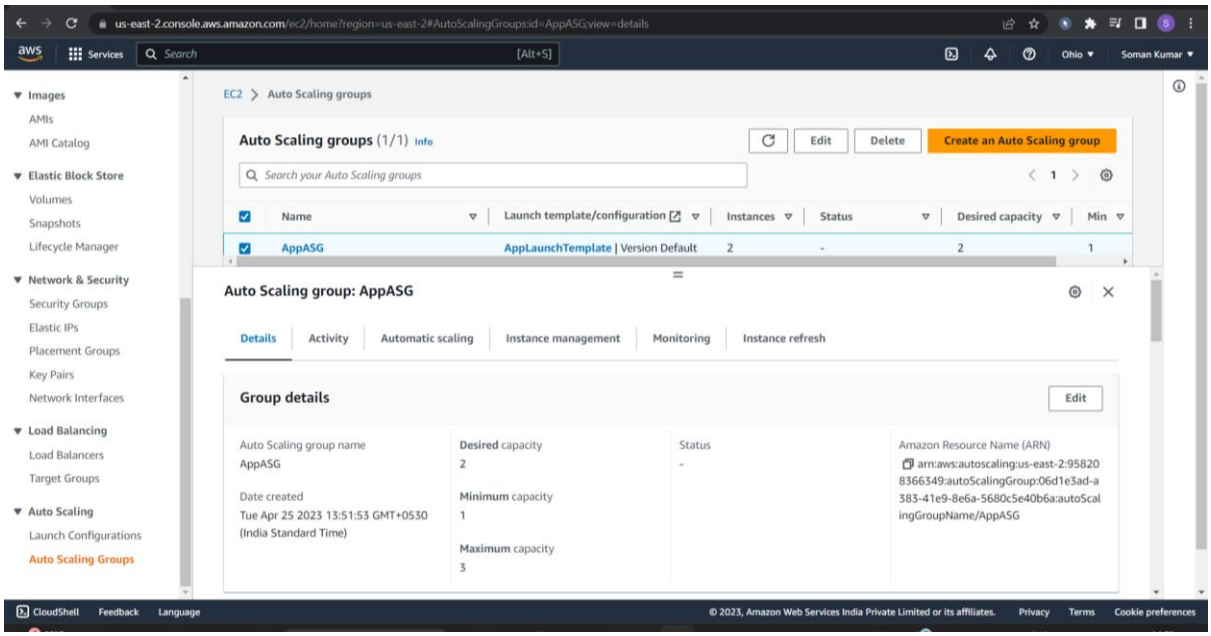

Fig 3.5.5 Adding Autoscaling group to our Application

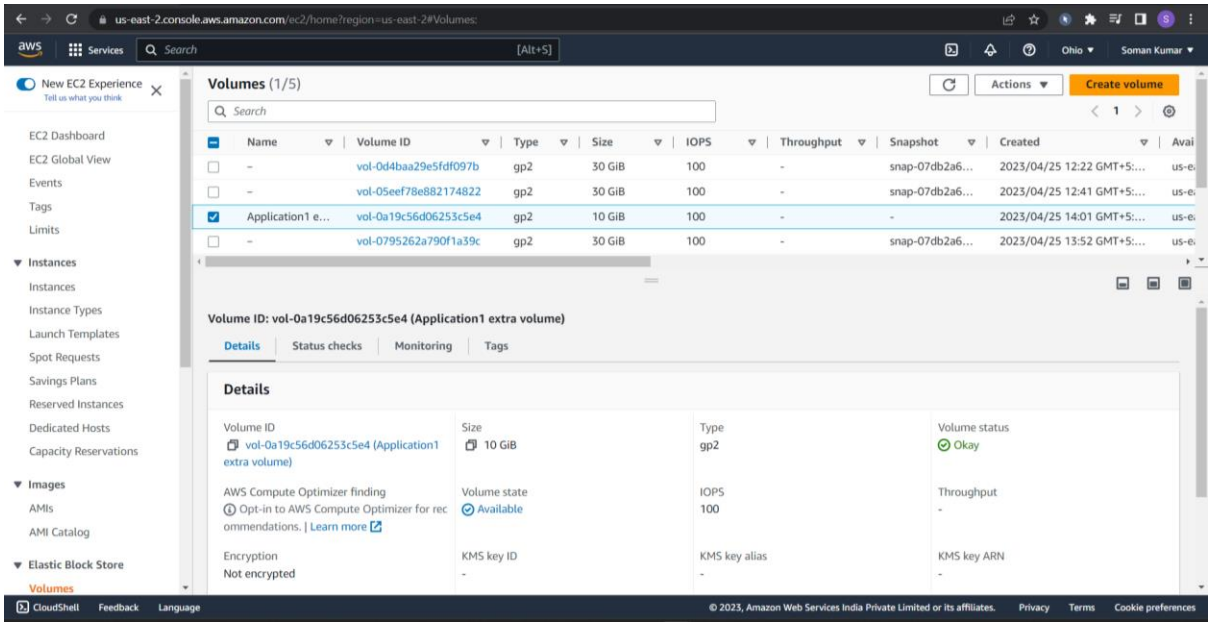

Fig 3.5.5 Adding elastic block Storage to our application

Result : Our web application which we have deployed running on our EC2 server

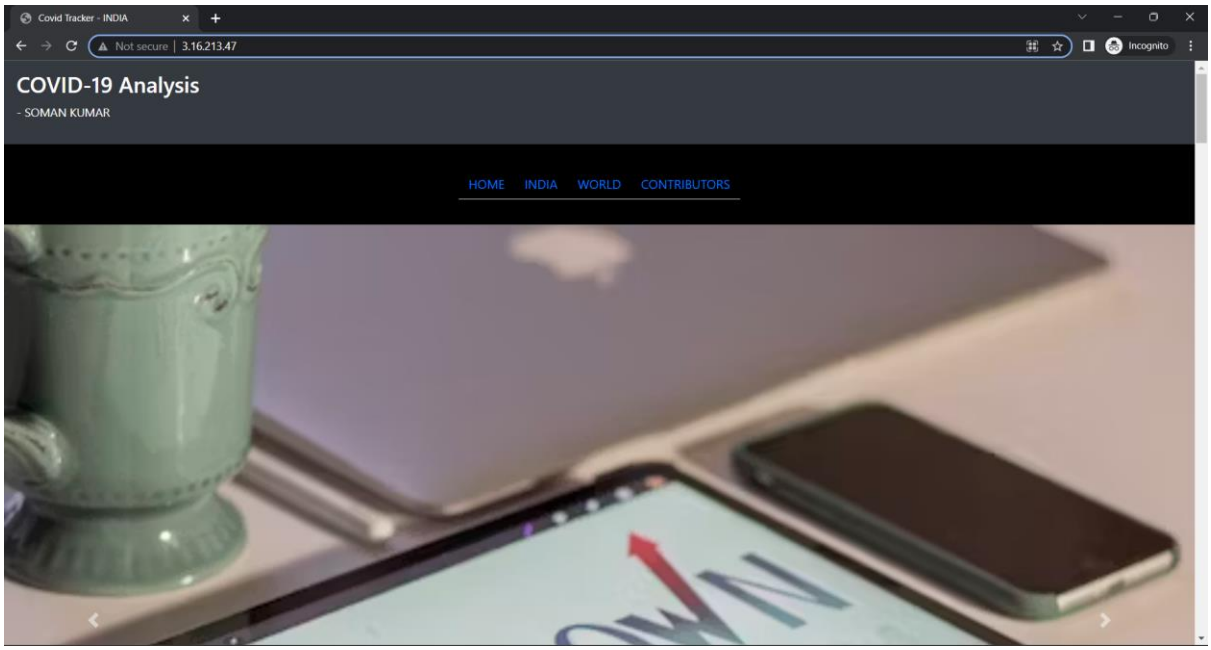

Fig 3.5.7a : Web App displaying the IP

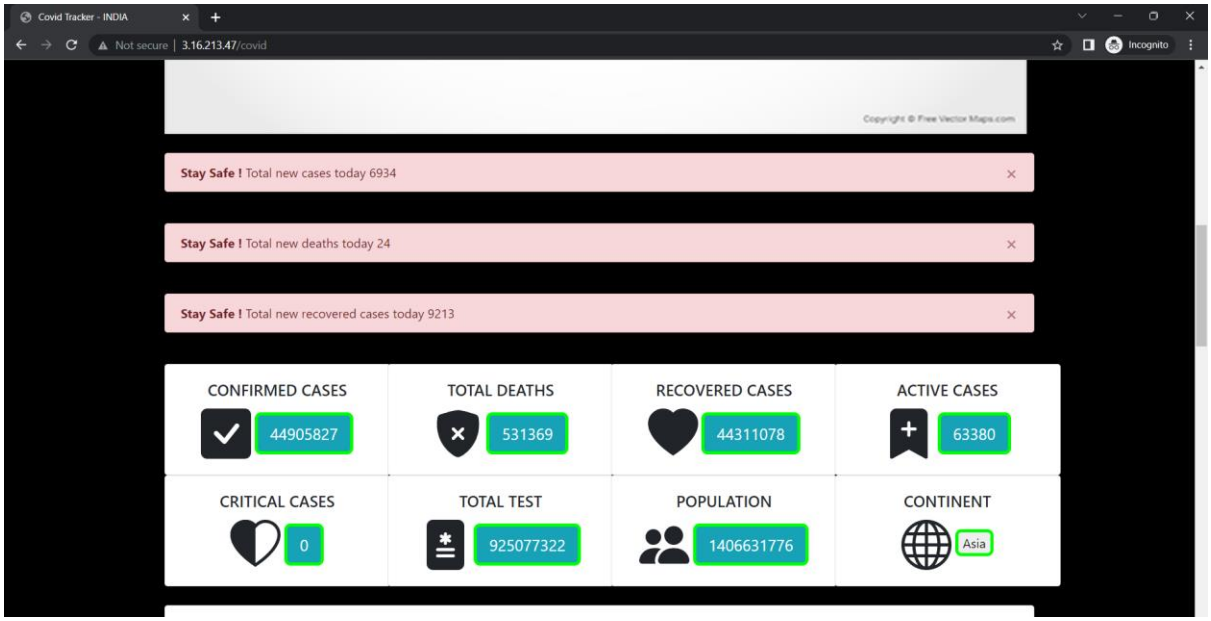

Fig 3.5.7b: Web App Page 2

Fig 3.5.7c :Web App Page 3

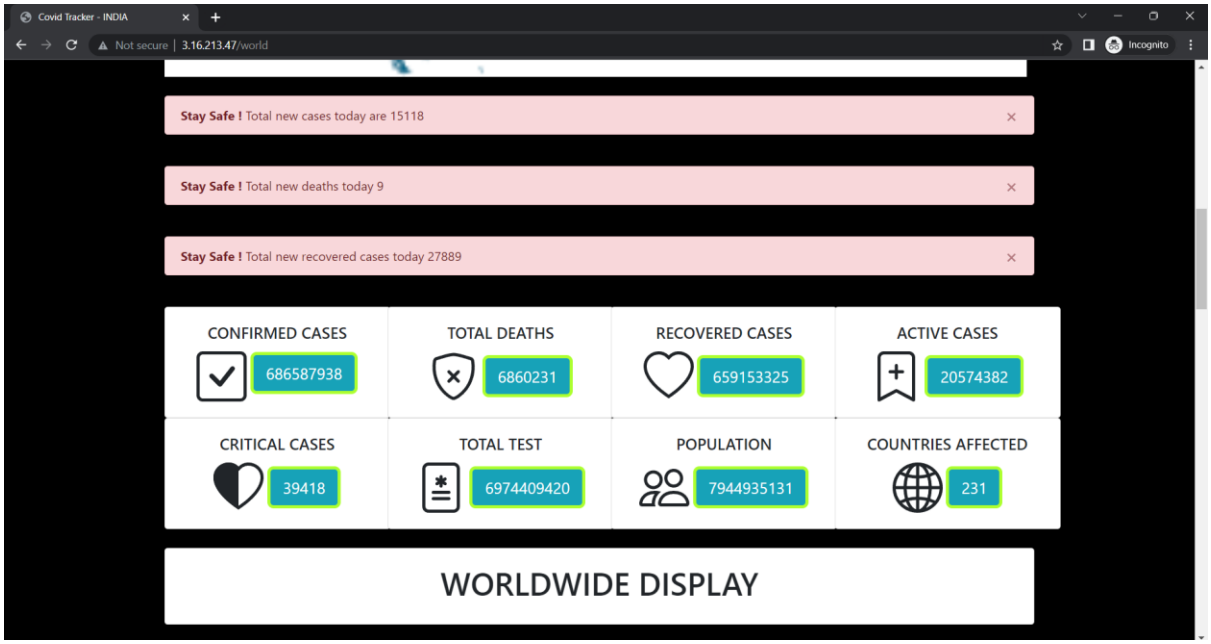

Fig 3.5.7d: Web App Page 4

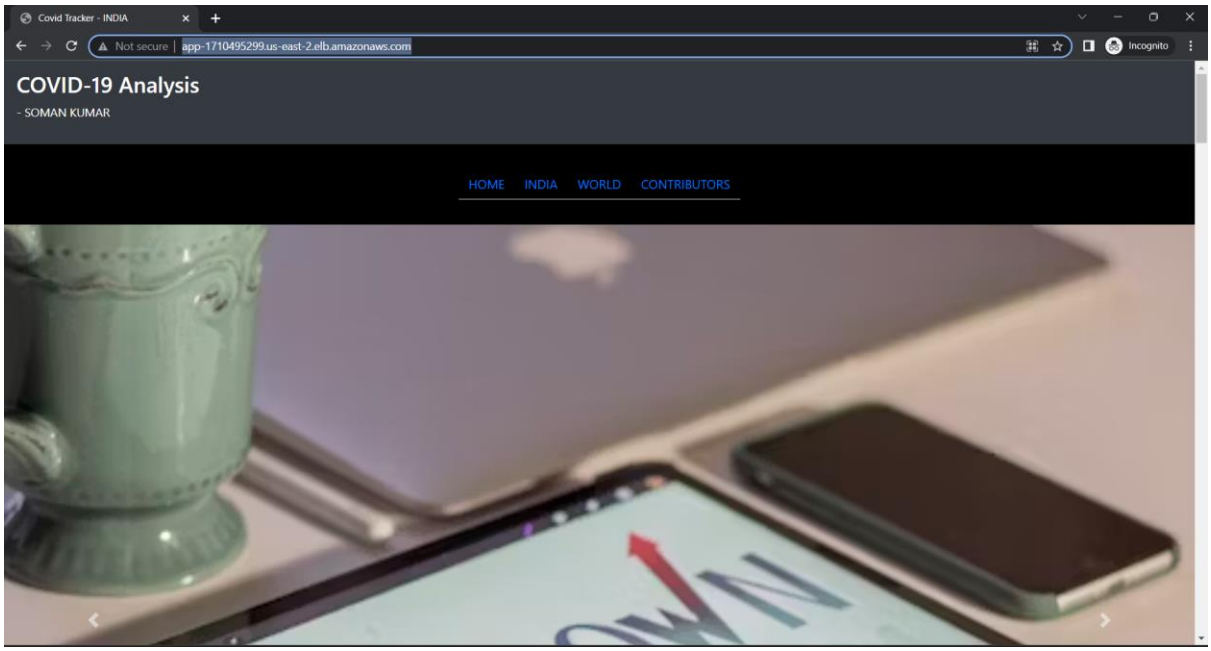

Fig 3.5.7e : Web App running by connecting Load Balancer and Autoscaling Group

Chapter 4

Future Work Plan

The future work plan of our project are as follows:

S.No.	Work Description	Duration in Days
1.	Learn how to establish alert and monitoring system using AWS cloud watch and cloud trail	5 days
2.	To learn about AWS code commit, code deploy	5 days
3.	Using Jenkins to create , configure jobs for continuous integration and delivery	10 days
4.	Learn about docker	10-14 days

Chapter 5

Weekly Task

Week No.	Date: From-To	Work Allocated	Work Completed (Yes/No)	Remarks	Guide Signature
1		Learned about software engineering concepts and various SDLC models	YES		
2		Learned and implemented about various git commands and also about Linux commands	YES		
3		Created infrastructure setup like creating s3 bucket, RDS, EFS ,Elastic beanstalk, creating IAM users, groups, policies for security purpose.	YES		
4		Deployed application on ec2 and adding load balancer and autoscaling groups to scale up the application. Also Learned about connecting it to RDS database, S3.	YES		
5		Knowing more about servers ,networks, security groups learning about VPC ,subnets and NAT gateway	YES		

References

[1] What is DevOps and Its tools like git,Jenkins ect [online] accessed on 19th April : https://docs.oracle.com/en-us/iaas/Content/DevOps/using/home.htm

[2] Fournier, A., & Durand, N. (2018). A DevOps approach for developing a cloud-based application using AWS. Procedia Computer Science, 138, 88-95.

[4] Johnson, J., & Weisang, A. (2021). Development, Deployment and Scaling of Web Applications with AWS and DevOps Tools. In 2021 6th International Conference on Cloud Computing and Internet of Things (CCIOT) (pp. 67-72).

[4] AWS EC2 documentation: [online] accessed on 20th April : https://docs.AWS.amazon.com/ec2/index.html#amazon-ec2

[5] AWS S3 documentation:[online] accessed on 21st April : https://docs.AWS.amazon.com/s3/index.html

[6] AWS documentation IAM[online] accessed on 21st April: https://docs.AWS.amazon.com/iam/index.html

[7] AWS documentation VPC and Subnets [online] accessed on 22nd April: https://docs.AWS.amazon.com/iam/index.html