

# INTERNSHIP REPORT

CST318

Abstract

This report is being presented as part of CST318 course to describe the details of the internship done as part of the winter internship process of the DKTE’s TEI Ichalkaranji, curriculum.

Atharv Mannur

PRN: 19UCS084

## Internship at

AICTE-Eduskills **AWS Cloud Virtual Internship**.

About the company:

Amazon Web Services, Inc. (AWS) is a subsidiary of [Amazon](https://en.wikipedia.org/wiki/Amazon.com) providing [on-demand](https://en.wikipedia.org/wiki/Software_as_a_service) [cloud computing](https://en.wikipedia.org/wiki/Cloud_computing) [platforms](https://en.wikipedia.org/wiki/Computing_platform) and [APIs](https://en.wikipedia.org/wiki/Application_programming_interface) to individuals, companies, and governments, on a metered pay-as-you-go basis. These cloud computing [web services](https://en.wikipedia.org/wiki/Web_services) provide a variety of basic abstract technical infrastructure and [distributed computing](https://en.wikipedia.org/wiki/Distributed_computing) building blocks and tools. One of these services is [Amazon Elastic Compute Cloud](https://en.wikipedia.org/wiki/Amazon_Elastic_Compute_Cloud) (EC2), which allows users to have at their disposal a [virtual](https://en.wikipedia.org/wiki/Virtualization) [cluster of computers](https://en.wikipedia.org/wiki/Computer_cluster), available all the time, through the Internet. AWS's virtual computers emulate most of the attributes of a real computer, including hardware [central processing units](https://en.wikipedia.org/wiki/Central_processing_unit) (CPUs) and [graphics processing units](https://en.wikipedia.org/wiki/Graphics_processing_unit) (GPUs) for processing; local/[RAM](https://en.wikipedia.org/wiki/Random-access_memory) memory; hard-disk/[SSD storage](https://en.wikipedia.org/wiki/Solid-state_drive); a choice of operating systems; networking; and pre-loaded application software such as [web servers](https://en.wikipedia.org/wiki/Web_server), [databases](https://en.wikipedia.org/wiki/Database), and [customer relationship management](https://en.wikipedia.org/wiki/Customer_relationship_management) (CRM).

The AWS technology is implemented at [server farms](https://en.wikipedia.org/wiki/Server_farm) throughout the world, and maintained by the Amazon subsidiary. 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](https://en.wikipedia.org/wiki/Availability_(system)), [redundancy](https://en.wikipedia.org/wiki/Redundancy_(engineering)), [security](https://en.wikipedia.org/wiki/Computer_security), and service options. Subscribers can pay for a single virtual AWS computer, a dedicated physical computer, or clusters of either. [[8]](https://en.wikipedia.org/wiki/Amazon_Web_Services#cite_note-AWSagreement-10) Amazon provides select portions of security for subscribers (eg. 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 from many global geographical regions including 6 in North America.

Amazon Elastic Compute Cloud (EC2)

By using AWS EC2 **helps users to avoid the investment in hardware up front, so the user can deploy and develop applications easier**. It is used to launch many virtual servers, configure networking and security, and managing storage.

**Task: Launch Your Amazon EC2 Instance**

In this task, you will launch an Amazon EC2 instance with termination protection. Termination protection prevents you from accidentally terminating an EC2 instance. You will deploy your instance with a User Data script that will allow you to deploy a simple web server.

### Step 1: Choose an Amazon Machine Image (AMI)

An **Amazon Machine Image (AMI)** provides the information required to launch an instance, which is a virtual server in the cloud. An AMI includes:

* A template for the root volume for the instance (for example, an operating system or an application server with applications)
* Launch permissions that control which AWS accounts can use the AMI to launch instances
* A block device mapping that specifies the volumes to attach to the instance when it is launched

The **Quick Start** list contains the most commonly-used AMIs. You can also create your own AMI or select an AMI from the AWS Marketplace, an online store where you can sell or buy software that runs on AWS.

Choose Select next to **Amazon Linux 2 AMI** (at the top of the list).

### Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more instance sizes, allowing you to scale your resources to the requirements of your target workload.

You will use a **t2.micro** instance which should be selected by default. This instance type has 1 virtual CPU and 1 GiB of memory. **NOTE**: You may be restricted from using other instance types in this lab.

Choose Next: Configure Instance Details

### Step 3: Configure Instance Details

This page is used to configure the instance to suit your requirements. This includes networking and monitoring settings.

The **Network** indicates which Virtual Private Cloud (VPC) you wish to launch the instance into. You can have multiple networks, such as different ones for development, testing and production.

For **Network**, select **Lab VPC**.

The Lab VPC was created using an AWS CloudFormation template during the setup process of your lab. This VPC includes two public subnets in two different Availability Zones.

For **Enable termination protection**, select **Protect against accidental termination**.

When an Amazon EC2 instance is no longer required, it can be terminated, which means that the instance is stopped and its resources are released. A terminated instance cannot be started again. If you want to prevent the instance from being accidentally terminated, you can enable termination protection for the instance, which prevents it from being terminated.

Scroll down, then expand **Advanced Details**.

A field for **User data** will appear.

When you launch an instance, you can pass user data to the instance that can be used to perform common automated configuration tasks and even run scripts after the instance starts.

Your instance is running Amazon Linux, so you will provide a shell script that will run when the instance starts.

Copy the following commands and paste them into the **User data** field:

*yum -y install httpd*

*systemctl enable httpd*

*systemctl start httpd*

*echo '<html><h1>Hello From Your Web Server!</h1></html>' > /var/www/html/index.html*

The script will:

* + Install an Apache web server (httpd)
  + Configure the web server to automatically start on boot
  + Activate the Web server
  + Create a simple web page

Choose Next: **Add Storage**

### Step 4: Add Storage

Amazon EC2 stores data on a network-attached virtual disk called Elastic Block Store.

You will launch the Amazon EC2 instance using a default 8 GiB disk volume. This will be your root volume (also known as a 'boot' volume).

Choose Next: **Add Tags**

### Step 5: Add Tags

Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type — you can quickly identify a specific resource based on the tags you have assigned to it. Each tag consists of a Key and a Value, both of which you define.

Choose Add Tag then configure:

* + **Key:** Name
  + **Value:** Web Server

Choose Next: Configure Security Group

### Step 6: Configure Security Group

A security group acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group.

On **Step 6: Configure Security Group**, configure:

* + **Security group name:** Web Server security group
  + **Description:** Security group for my web server

In this lab, you will not log into your instance using SSH. Removing SSH access will improve the security of the instance.

Delete the existing SSH rule.

Choose Review and Launch

### Step 7: Review Instance Launch

The Review page displays the configuration for the instance you are about to launch.

Choose Launch

A **Select an existing key pair or create a new key pair** window will appear.

Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. To log in to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance.

In this lab you will not log into your instance, so you do not require a key pair.

Choose the **Choose an existing key pair** drop-down and select Proceed without a key pair.

Select **I acknowledge that ...**.

Choose Launch Instances

Your instance will now be launched.

Choose View Instances

The instance will appear in a pending state, which means it is being launched. It will then change to running, which indicates that the instance has started booting. There will be a short time before you can access the instance.

The instance receives a public DNS name that you can use to contact the instance from the Internet.

Your **Web Server** should be selected. The **Description** tab displays detailed information about your instance.

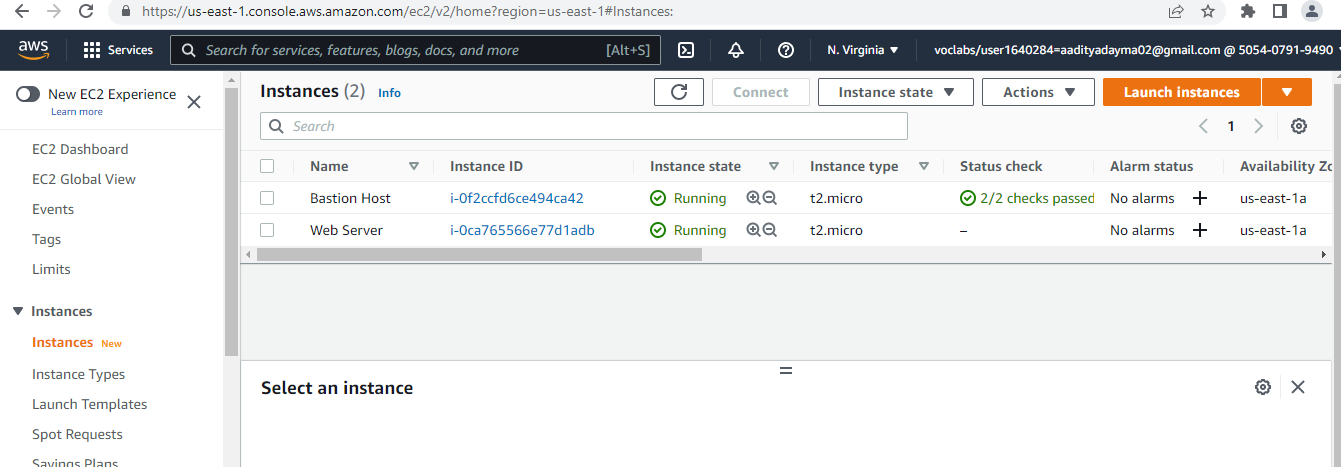
To view more information in the Description tab, drag the window divider upwards.

Review the information displayed in the **Details** tab. It includes information about the instance type, security settings and network settings.

Wait for your instance to display the following:

* **Instance State:** running
* **Status Checks:** 2/2 checks passed

We have successfully launched your first Amazon EC2 instance.



Tools used:

* Amazon EC2
* Amazon S3
* Amazon DynamoDB
* Amazon RDS
* AWS Lambda
* Amazon Sage-maker
* Amazon VPC
* Amazon Aurora
* AWS Elastic Beanstalk
* Amazon ECS
* Amazon DevOps
* Amazon EKS
* Amazon Redshift
* AWS CloudTrail

Working Methodology

After shortlisting for the internship, AWS Academy provided Two fundamental courses: AWS Cloud Foundation Course and AWS Cloud Architecting Course. For the Cloud Foundation course, the faculty member conducts the session. In this foundation course, we have 12 Modules that we have completed in a given time. Initially, we started watching videos on the AWS Academy learning platform. After completion of the course, AWS Academy provides us with a course completion certificate, which we have submitted to the Eduskills Foundation. After completion, AWS Cloud Architecting was assigned to us. For this course, some experts from the Eduskill Foundation and AICTE conducted sessions that were helpful for us. This AWS Cloud Architecting course consists of 13 Modules and a Capstone Project. This course has two labs: a guided lab where the AWS service is described in detail, and another lab where we need to achieve the objective by reviewing the guided lab.

AWS Academy Cloud Foundation

AWS Academy Cloud Foundations is intended for students who seek an overall understanding of cloud computing concepts, independent of specific technical roles. It provides a detailed overview of cloud concepts, AWS core services, security, architecture, pricing, and support.

AWS Academy Cloud Architecting

AWS Academy Cloud Architecting covers the fundamentals of building IT infrastructure on Amazon Web Services, or AWS. The course is designed to teach solution architects how to optimize the use of the AWS Cloud by understanding AWS services and how these services fit into cloud-based solutions. Because architectural solutions can differ depending on the industry, type of application, and size of business, this course emphasizes best practises for the AWS Cloud and recommends various design patterns to help you think through the process of architecting optimal IT solutions on AWS. It also presents case studies throughout the course, which showcase how some AWS customers have designed their infrastructures and the strategies and services that they have implemented. Finally, this course also provides opportunities to build a variety of infrastructures via a guided, hands-on approach.

About the Projects

After Successful completion of both Course i.e., foundation course and Solution Architect course AWS academy assign us one final Capstone project which was last in this internship program. Capstone Project was totally based AWS cloud facilities- AWS EC2, AWS Lambda, AWS S3, AWS Cloudform. This project provides us with an opportunity to demonstrate the solution design skills that you develop throughout this course. our task is to design and deploy a solution for the following case. By the end of this project, you should be able to apply the architectural design principles that you learned in this course to:

• Deploy a PHP application that runs on an Amazon Elastic Compute Cloud (Amazon EC2) instance

• Create a database instance that the PHP application can query

• Create a MySQL database from a structured query language (SQL) dump file

• Update application parameters in an AWS Systems Manager Parameter Store

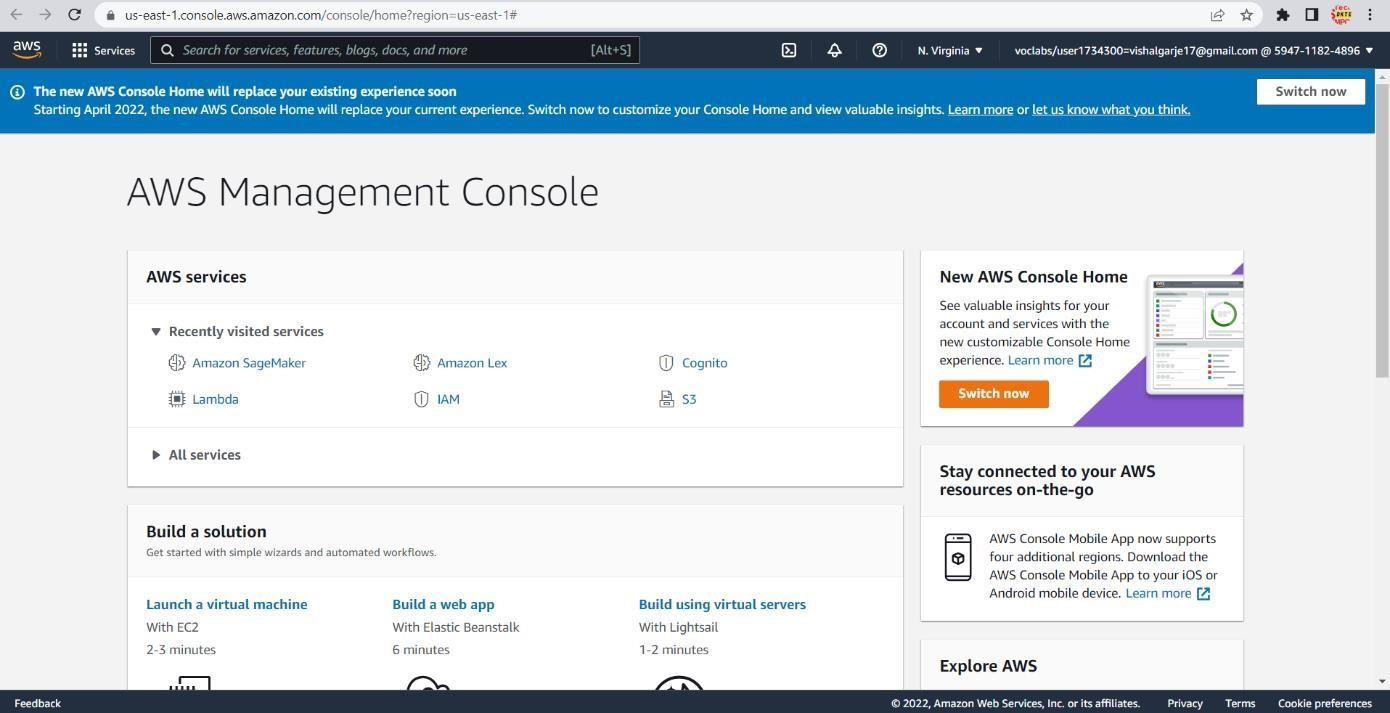
• Secure the application to prevent public access to backend systems

**Problem Statement:**

Build Web App with Database and Deploy this App on AWS cloud. After Successful Deploy check whether all functions are working properly or not.

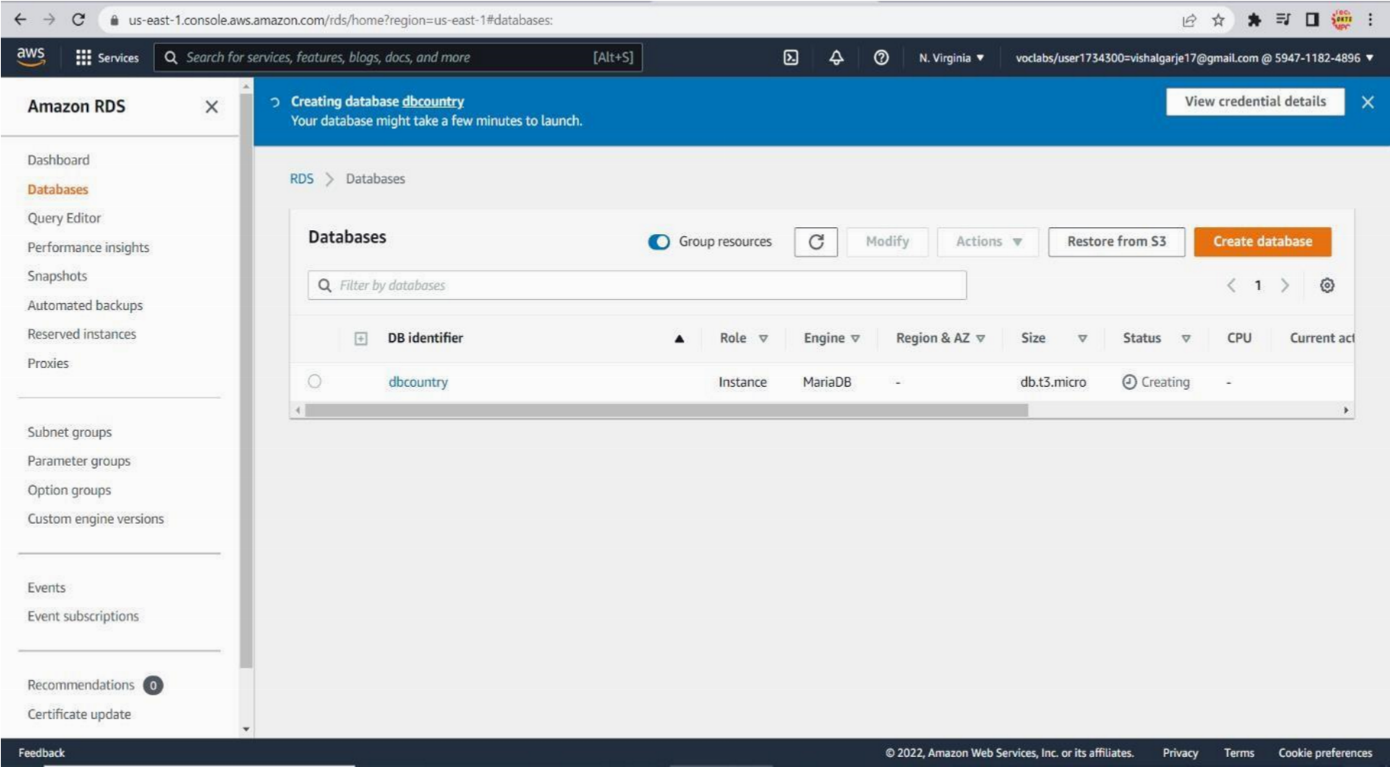
**Solution:**

**Open Amazon console:**



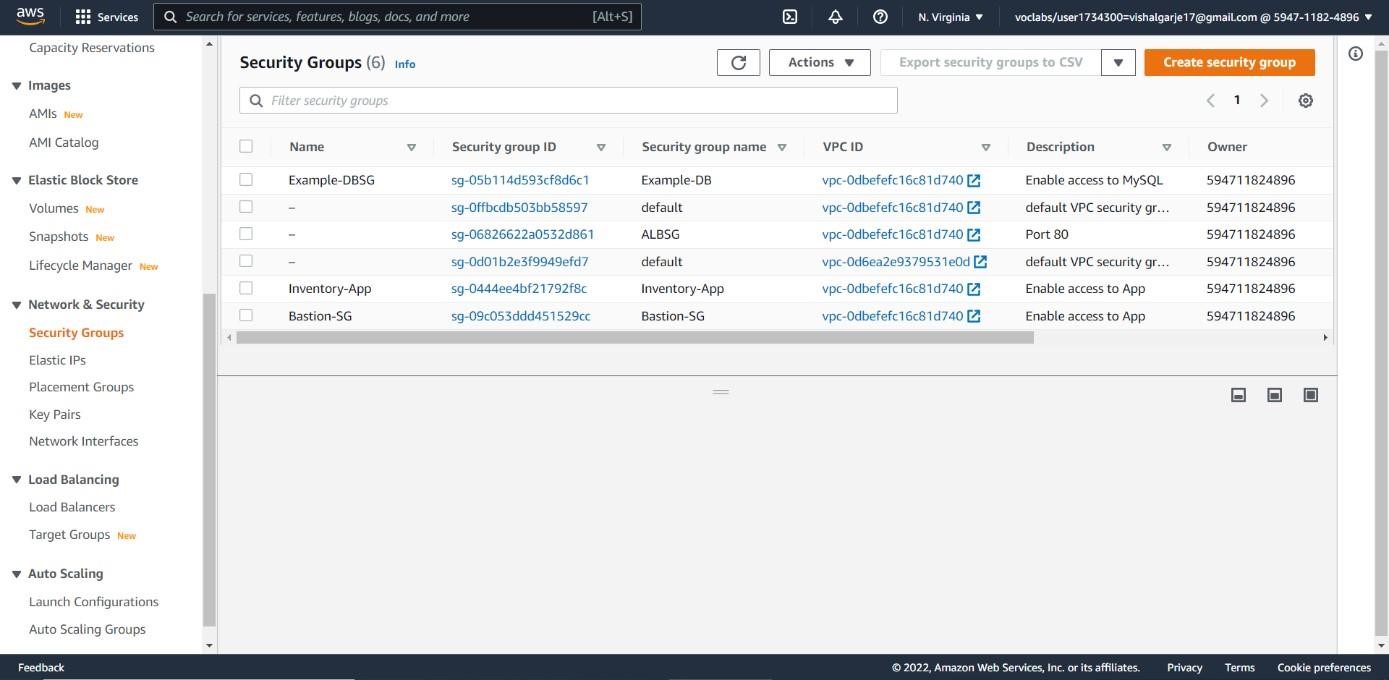
**Open Amazon RDS and Create Database:**

* Provide secure hosting of the MySQL database

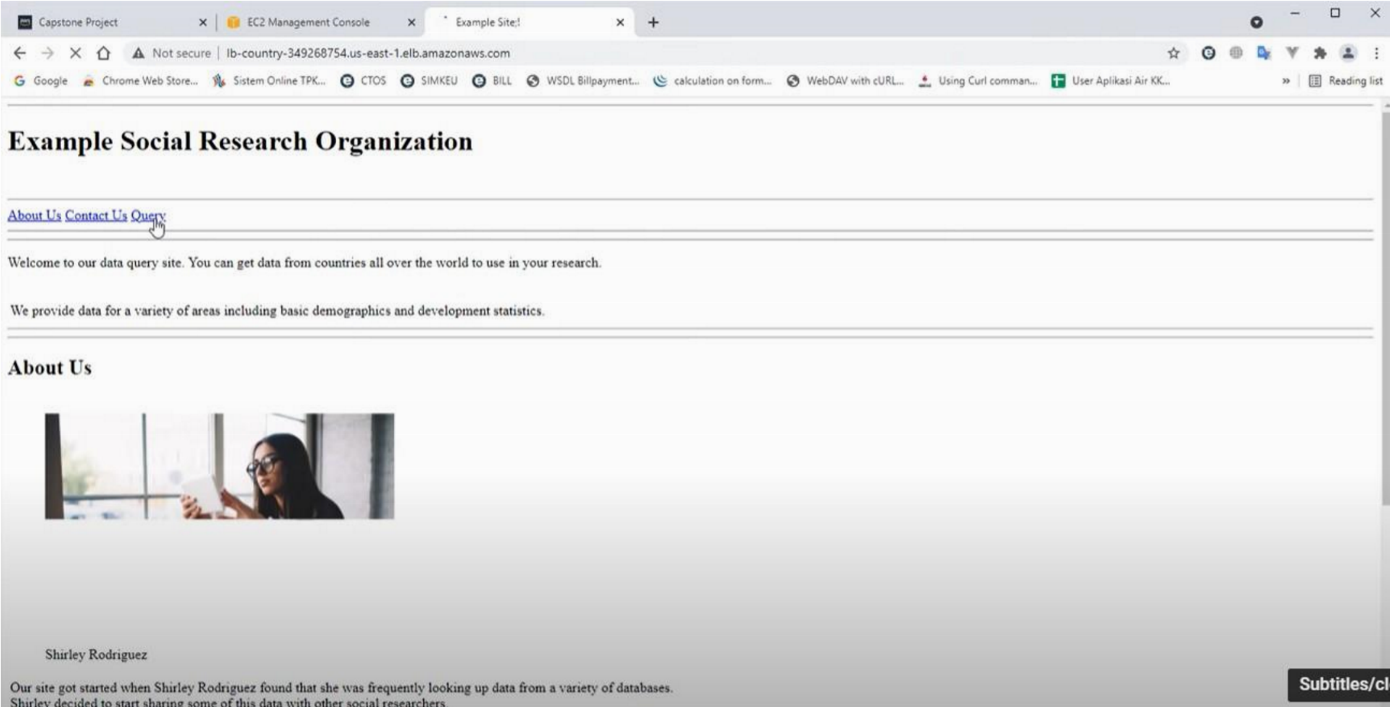


**Open Amazon EC2 Instance and Provide access:**

* Provide secure access for an administrative user.
* Provide anonymous access to web users

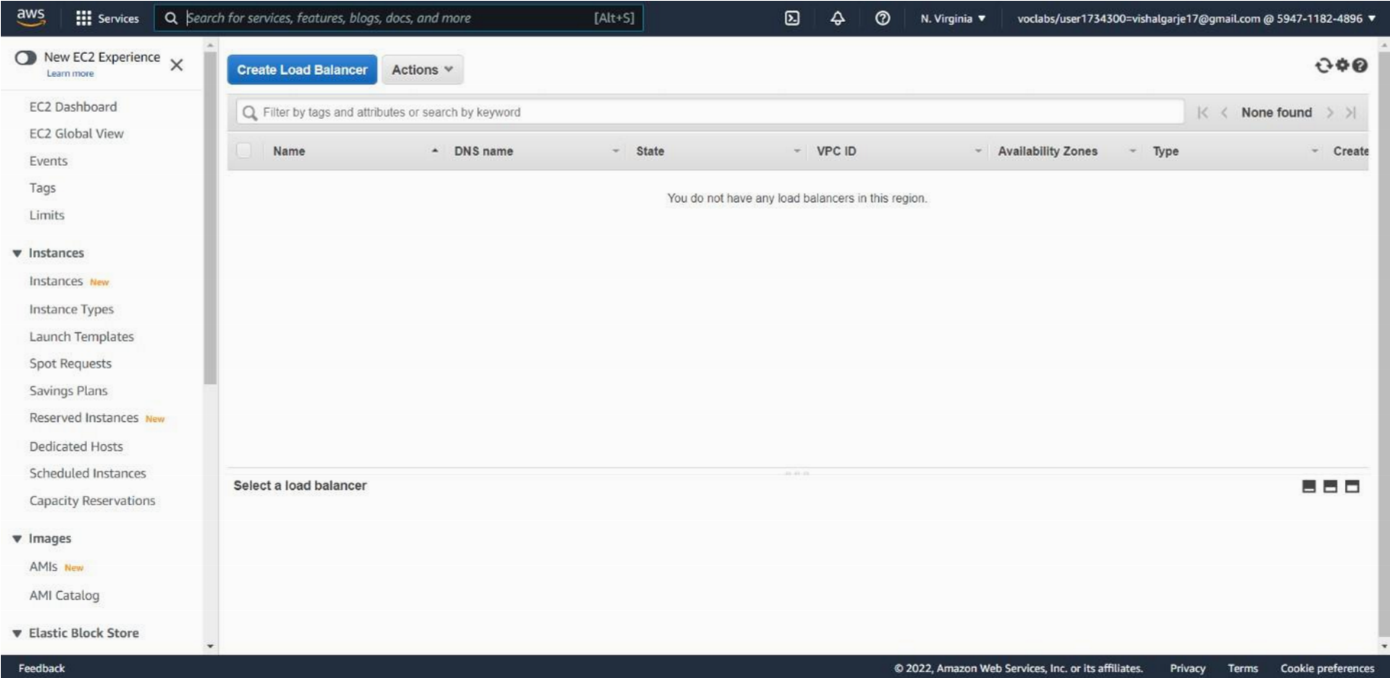


* Run the website on a t2.small EC2 instance, and provide Secure Shell (SSH) access to administrators



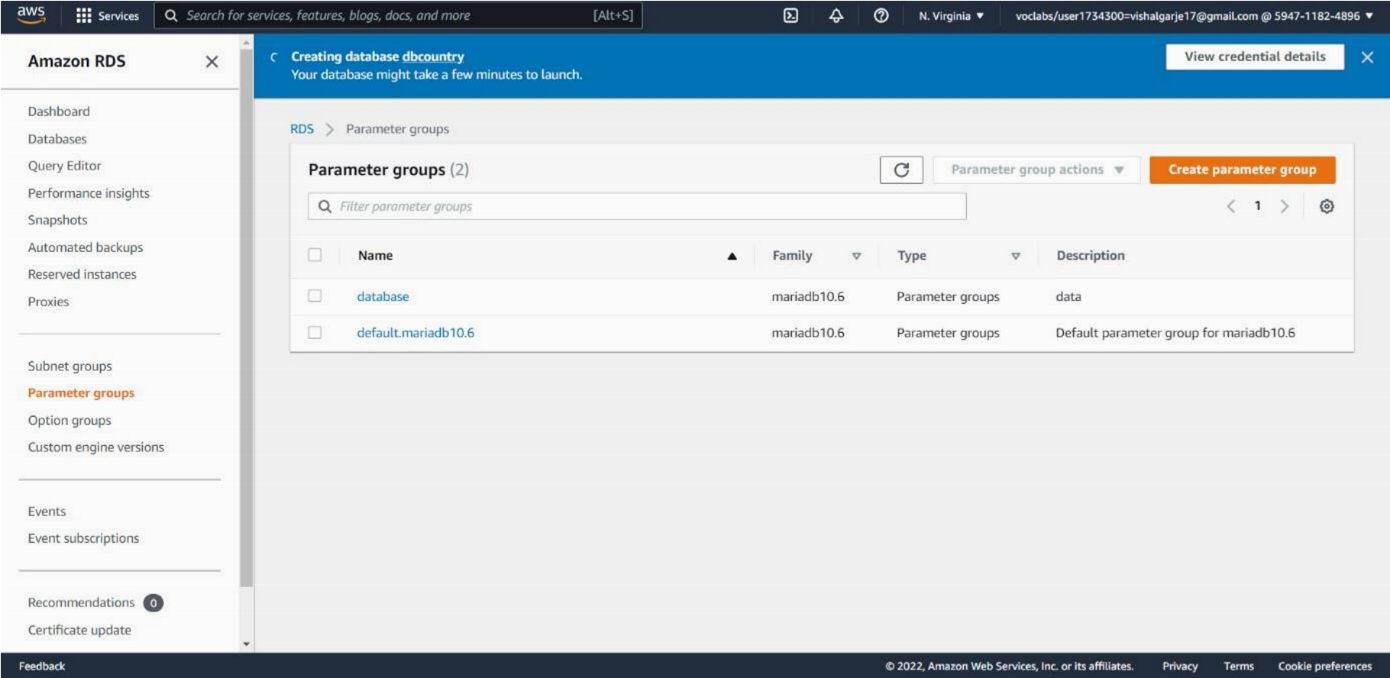
**Open Amazon EC2 Instance and Create Load Balancer:**

* Provide high availability to the website through a load balancer

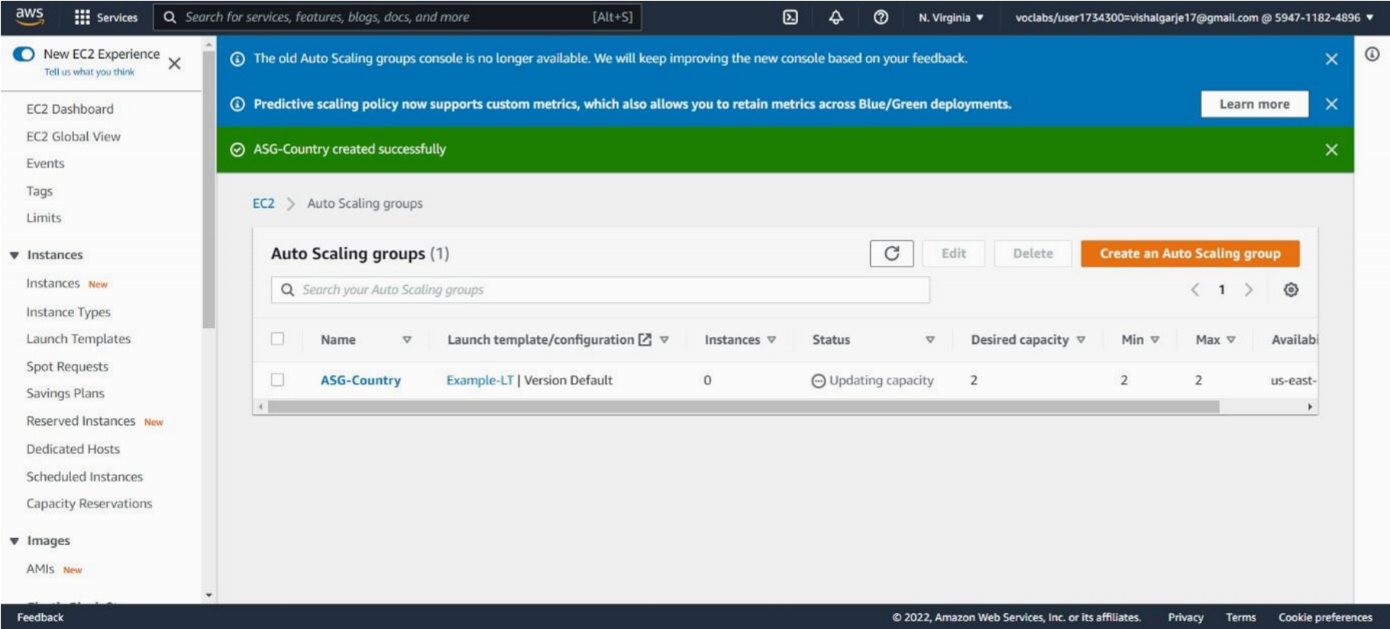


**Open Amazon RDS and Create Parameter Group:**

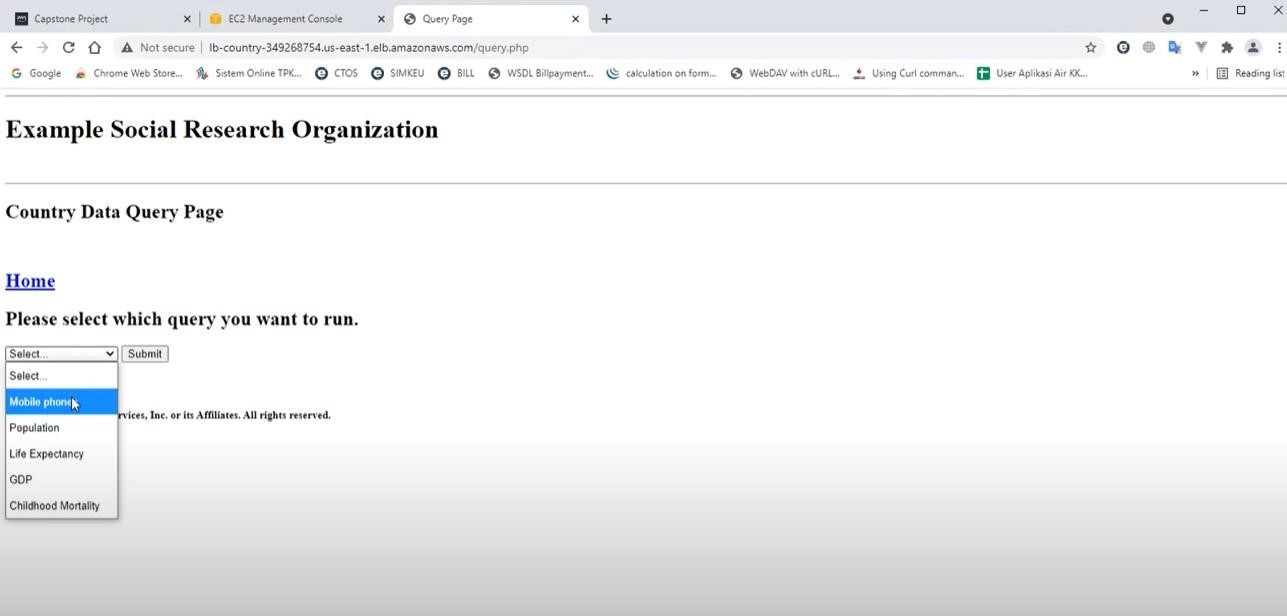
* + Store database connection information in the AWS Systems Manager Parameter Store



**Open Amazon EC2 Instance and Create Auto Scaling Group:**



**Test Web App via URL in Load Balancer:**



Conclusion

Through the AWS Academy, we have the chance to learn more about the cloud computing technologies and services that AWS has to offer. The guidance provided by faculty and experts from AICTE and the Edu-Skill Foundation was helpful for us. In the internship period, we learned a lot of things regarding Amazon Web Services, project management, and teamwork.

About my role

During my internship at AICTE-Eduskills AWS Cloud Virtual Internship, my role primarily revolved around gaining practical experience and proficiency in utilizing various Amazon Web Services (AWS) tools. As an intern, I had the opportunity to work with a wide range of AWS services, including Amazon EC2, Amazon S3, Amazon DynamoDB, Amazon RDS, AWS Lambda, Amazon VPC, Amazon Aurora, AWS Elastic Beanstalk, Amazon ECS, Amazon DevOps, Amazon EKS, Amazon Redshift, and AWS CloudTrail.

Throughout my role in this internship, I gained hands-on experience in working with a diverse range of AWS tools and services, honing my skills in cloud computing, infrastructure design, deployment, and security. The opportunity to apply the theoretical concepts learned in the courses to a real-world project has been invaluable in developing my understanding and proficiency in AWS cloud technologies.