COMP\_SCI-5525-CLOUD\_COMPUTING

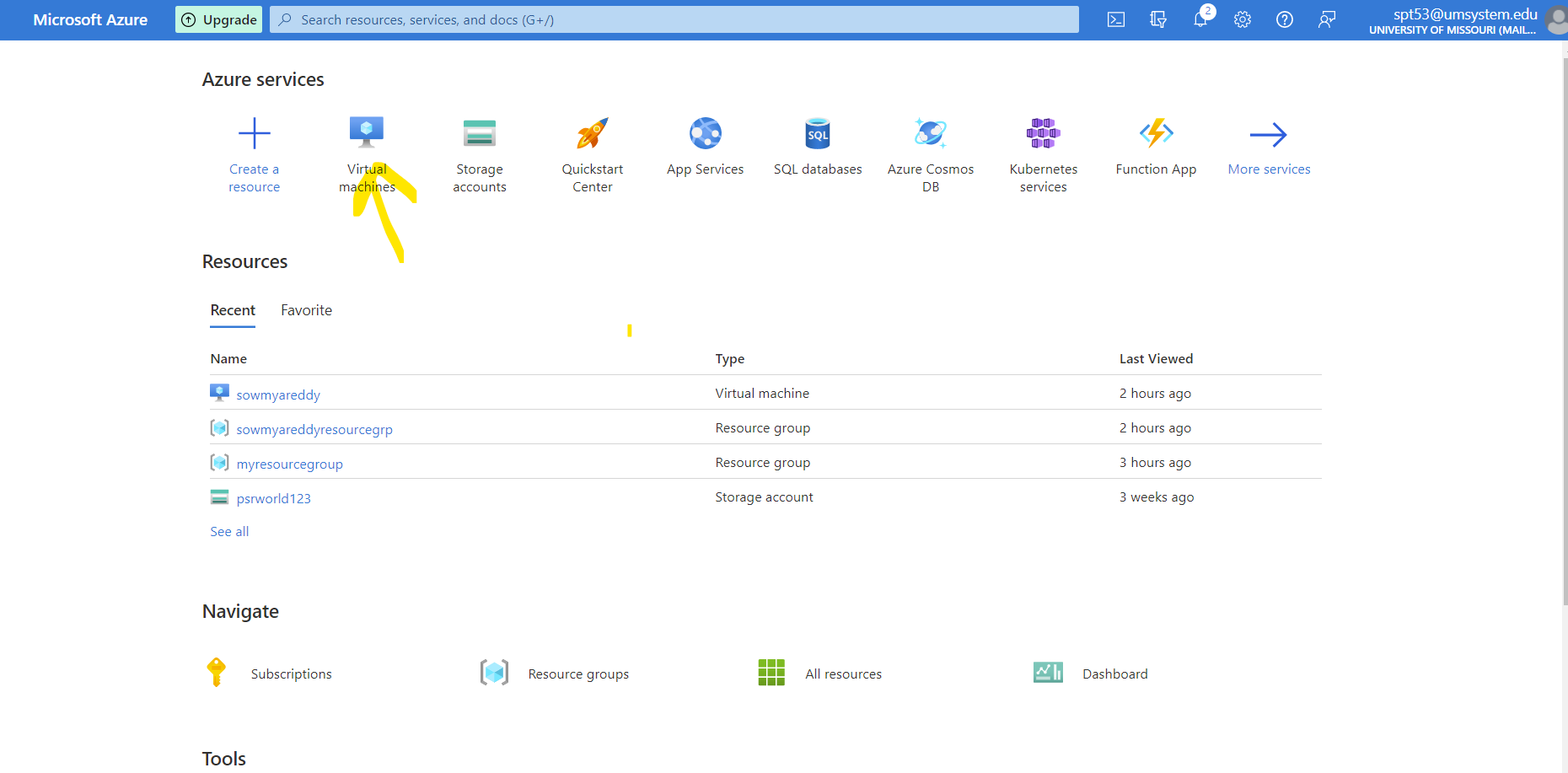
Assignment-3

**Sowmya Patlolla**

1. **Deploying a static website on Microsoft Azure using a Virtual Machine**

**Website URL:** [**http://20.213.93.247/**](http://20.213.93.247/)

Step 1: Open the Azure cloud portal and then click on Virtual machines.

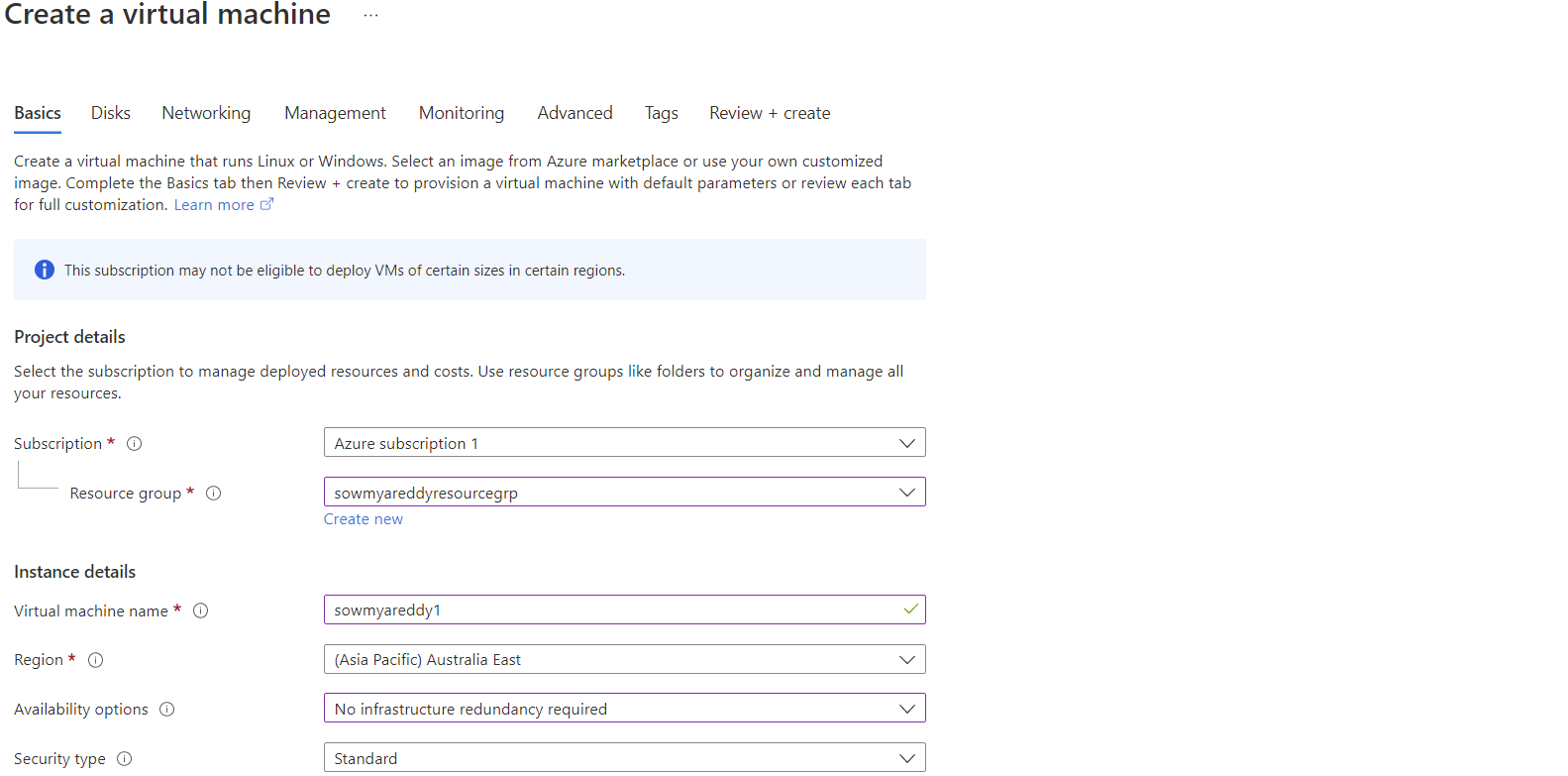


Step 2: Now create a VM by clicking on create Azure virtual machine.

Graphical user interface, text, application, Word, email

Description automatically generated

Step 3: Create a virtual machine with following details.



Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Step 4: After entering all the required configurations click on Review create button to create virtual machine.

Graphical user interface, text, application, email

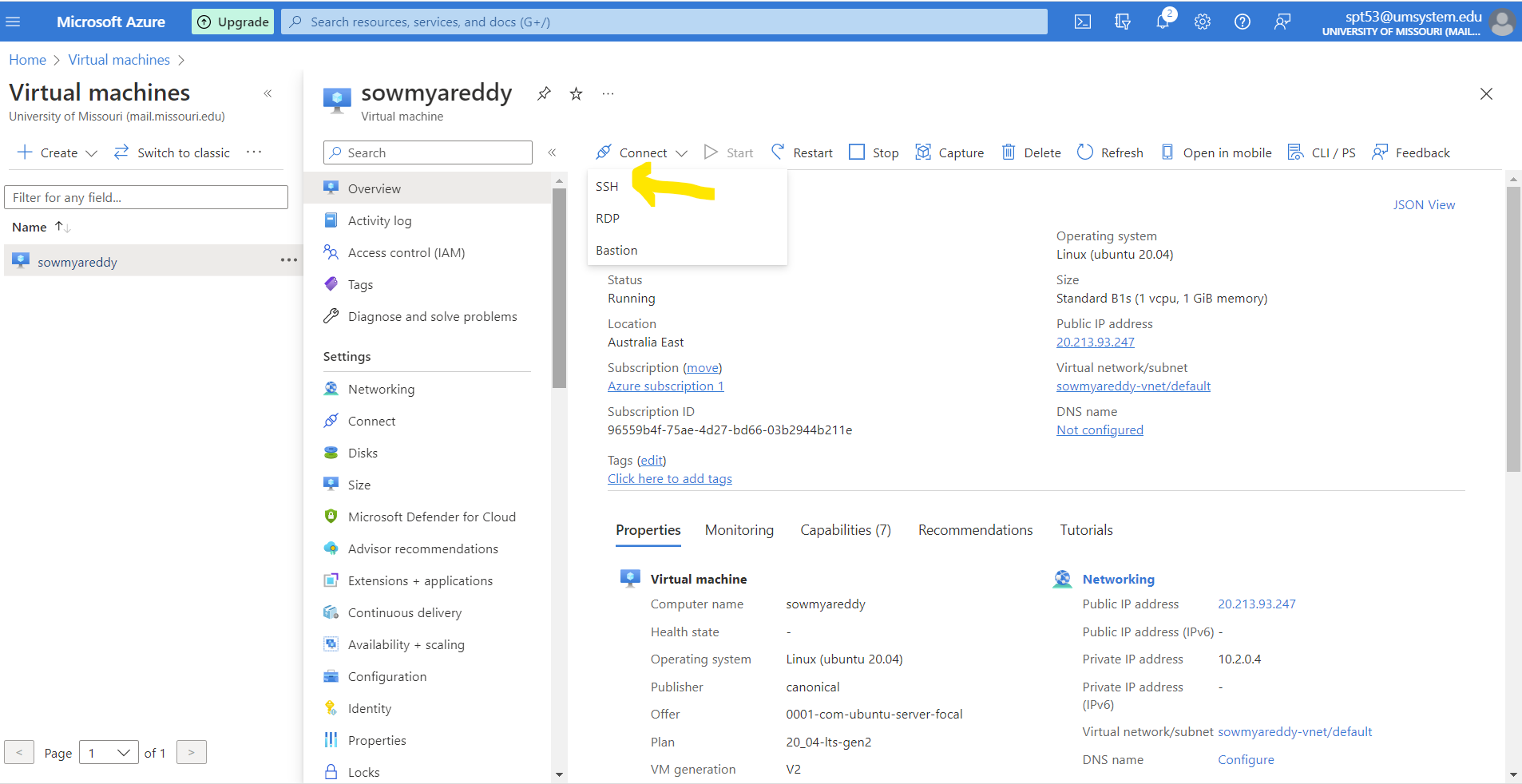
Description automatically generated

Step 5: View the created virtual machine by selecting the virtual machines in the home page.

Graphical user interface, text, application, email

Description automatically generated

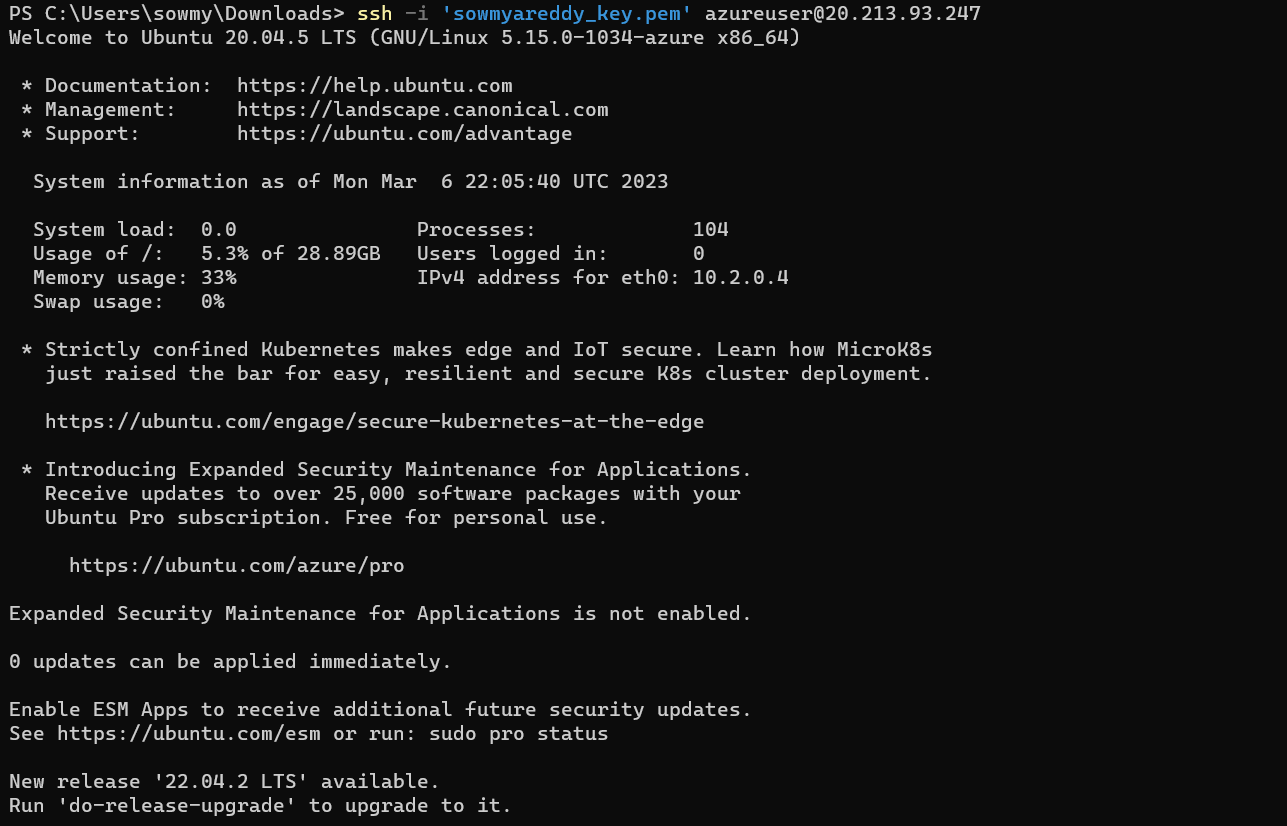
Step 6: Now click on the selected VM and click on connect then select SSH.



Step 7: Connect to the SSH through power shell in windows by copy the following command from created virtual instance page.

Graphical user interface, text, application, email

Description automatically generated



Step 8: After connecting to the SSH, install the apache2 server to host the static website by the following command.

Text

Description automatically generated

Step 9: Go to the html file path and fetch the static website html file by the providing the following command.

Text

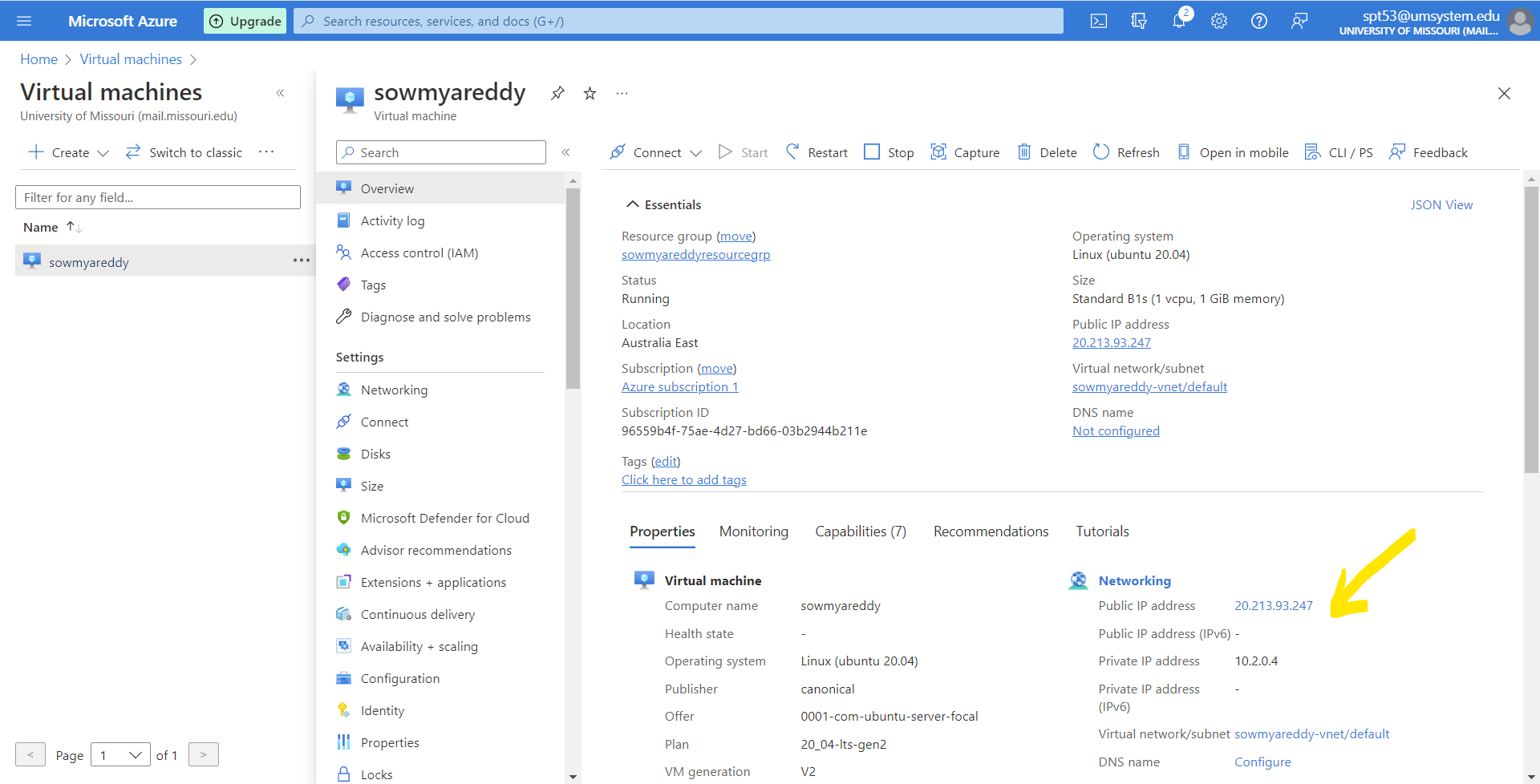
Description automatically generated

Step 10: Write the required code for the static website. My index file contains the following code.

Text

Description automatically generated

Step 11: Access the static website by clicking on the public address provided in the created virtual machine page.



Step 12: Test the static website by the following URL.

**Website URL:** <http://20.213.93.247/>

**Graphical user interface, application

Description automatically generated**

**2.Deploying a static website on GCP using a Virtual Machine**

**Website URL:** [**http://34.125.16.158/**](http://34.125.16.158/)

Step1: Create and setup Google Cloud Platform account and login to it.

Graphical user interface, application

Description automatically generated

Step 2: Go to the compute engine and select VM instances to create a virtual machine.

Graphical user interface

Description automatically generated

Step 3: Click on create instance to create a virtual machine instance.

Graphical user interface, text, email

Description automatically generated

Step 4: Give a name to VM and select the required region, zone and configurations based on our requirement.

Graphical user interface, application

Description automatically generated

Step 5: Select both firewall options to allow http traffic and then click on create button.

Graphical user interface, text, application, email

Description automatically generated

Step 6: Below image shows where created instances will be visible.

Graphical user interface, text, application, email

Description automatically generated

Step 7: Connecting to the virtual machine by clicking on the SSH. It opens the SSH terminal in new tab.

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

Step 8: Install the following command to install the Apache web server.

Text

Description automatically generated

Step 9: Go to the html file path and fetch the static website html file by the providing the following command.

Text

Description automatically generated

Step 10: Write the required code for the static website. My index file contains the following code.

Text

Description automatically generated

Step 11: Access the static website by clicking on the public address provided in the created virtual machine page.

Graphical user interface, text, email

Description automatically generated

Step 12: Test the static website by the following URL.

Website URL: <http://34.125.16.158/>

Graphical user interface, application

Description automatically generated**3. Deploying a static website on AWS using a Virtual Machine and Docker Container**

**Website URL:**[**http://54.144.46.217/My\_Page.html**](http://54.144.46.217/My_Page.html)

Step 1: Open the AWS console and then click on the EC2 instance to create a virtual machine instance.

Graphical user interface, text, application

Description automatically generated

Step 2: Click on launch instance button to create an amazon EC2 instance.

Graphical user interface, application

Description automatically generated

Step 3: Now give the name to the instance and select application and OS image as Ubuntu.

Graphical user interface, application

Description automatically generated

Step 4: For the key pair, click on the create new key pair option then it generates a unique key to access the EC2 instance and check all the firewall options provided in the page to access the static website and then click on launch instance option to create instance.

Graphical user interface, text, application

Description automatically generated

Step 5: Following is the created EC2 instance.

Graphical user interface, text, application

Description automatically generated

Step 6: Click on the created EC2 instance and then go to connect to instance and then select the SSH client.

Graphical user interface, text, application, email

Description automatically generated

Step 7: Enter the following command in the power shell to connect to the SSH.

ssh -i "sowmya\_aws\_key.pem" ubuntu@ec2-54-144-46-217.compute-1.amazonaws.com

Text

Description automatically generated

Step 8: Install the following command to install the Apache web server.

Text

Description automatically generated

Step 9: Go to the html file path and fetch the static website html file by the providing the following command.

Graphical user interface, text

Description automatically generated

Step 10: Write the required code for the static website. My index file contains the following code.

Text

Description automatically generated

Step 11: Access the static website by clicking on the public address provided in the created EC2 instance page.

Graphical user interface, application

Description automatically generated

Step 12: Test the static website by the following URL.

Website URL: <http://54.144.46.217/My_Page.html>

Graphical user interface, application

Description automatically generated

**Deploying the static website using Docker Container image in AWS cloud:**

Step 1: Installing the docker container image using the following command.

Text

Description automatically generated

Step 2: Building an image from Docker file using the following command.

Text

Description automatically generated

Step 3: Run the docker image using the following command.



Step 4: Use the following curl command to check the response of the port 40:80.

Text

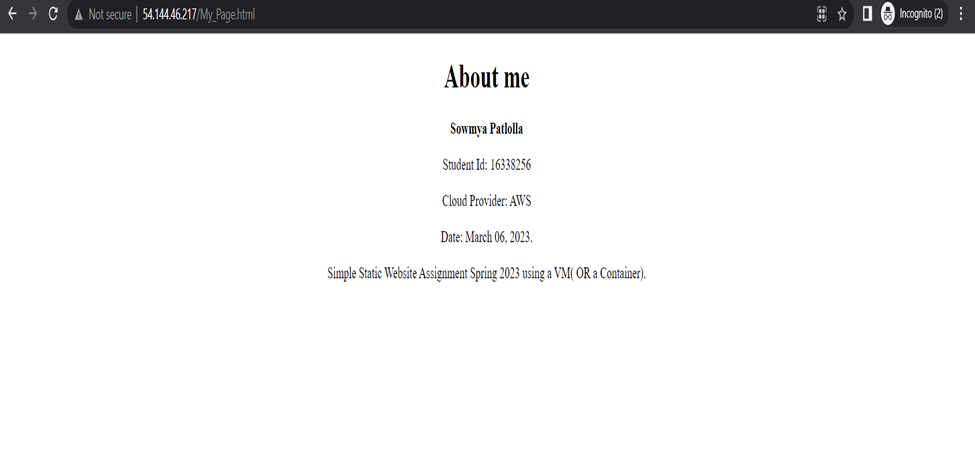
Description automatically generated

Step 5: Checking the status of the Apache web server by using the following command.

Text

Description automatically generated

Step 6: Click on the public Ip address provided in the EC2 instance page to test the static website.



**4.Epilog**

**Lessons learnt**:

Public clouds such as AWS, GCP, and Azure offers various services that can be used to deploy and manage static websites, such as virtual machines, container registries, load balancers, and content delivery networks.

Deploying a static website using virtual machines or containers in public clouds requires writing scripts or configuration files that describes the need of the infrastructure to run the website. This can teach us the importance of Infrastructure as Code and how it can help us to manage and automate the deployment and scaling of our applications. It also teaches us the importance of cost optimization. By choosing the right instance types, storage options, and network configurations, we can reduce the cost of running our application while maintaining performance and reliability. Using containers to deploy a static website can help us to package the website and its dependencies in a portable format. It can teach us the benefits of using containerization and it simplifies the application deployment and management.

**Fun Part:**

One of the enjoyable aspects of the task was exploring various cloud services and learning about their potential. I had a lot of fun exploring the many features and services that various cloud service providers, such as Amazon, Azure, and GCP, offer. As a result, I compared the functionality of cloud services provided by each of the cloud provider.

**Challenging Part:**

It can be difficult to set up and configure the virtual machines, especially if we are unfamiliar with the cloud platform or the operating system that the virtual machine is using. Web servers, databases, SSL certificates, and other software components would have to be manually installed and configured, which can be time-consuming and error-prone.