Cloud Computing Project Report

Title: Creating a VM on Azure and Hosting a Static Website

Submitted by: Nikhil

Assigned by: 1stop.ai

Date: 06 June 2025

Table of Contents

ABSTRACT

This project explores the end-to-end process of deploying a static website using Microsoft Azure's Infrastructure as a Service (IaaS) platform. Specifically, we provision and configure a Virtual Machine (VM) running Ubuntu 24.04 LTS and install the Nginx web server to serve static content. The aim is to demonstrate how cloud-based virtual infrastructure can be leveraged for simple web hosting tasks, providing flexibility, scalability, and hands-on exposure to real-world deployment environments.

The report outlines the complete workflow—from creating and securing the VM, installing necessary software, uploading website files, and configuring firewalls and network settings, to finally verifying deployment through browser access via a public IP address. This practical implementation reinforces foundational concepts in cloud computing, such as resource provisioning, security group configuration, and virtual networking.

Through this project, students gain hands-on experience with Azure's VM ecosystem and learn how to deploy and manage static websites in a secure and controlled manner. It not only strengthens understanding of hosting fundamentals but also introduces industry-relevant skills necessary for future roles in DevOps, web development, and cloud infrastructure management.

OBJECTIVE

The primary objectives of this project are to:

- **Understand and apply the fundamentals of cloud computing** by working directly with Microsoft Azure's Infrastructure as a Service (IaaS) offerings.
- **Create, configure, and manage Virtual Machines (VMs)** within the Azure platform to host web services.
- **Gain practical experience working with Linux server environments**, particularly Ubuntu 24.04 LTS.
- **Install and configure the Nginx web server** to host and serve static content over HTTP.
- **Deploy and test a static website** built using HTML and CSS, simulating real-world frontend deployment.
- **Implement basic security configurations** using Azure Network Security Groups (NSG) to manage inbound and outbound traffic.
- **Verify deployment through public IP access** and ensure that the web server is functional and reachable.
- **Monitor VM performance** using Azure's built-in monitoring tools and metrics such as CPU usage and memory availability.
- **Apply secure access techniques** such as SSH key authentication and restricted port access to prevent unauthorized usage.
- **Bridge academic knowledge with real-world infrastructure**, developing familiarity with modern DevOps and cloud hosting techniques.

INTRODUCTION

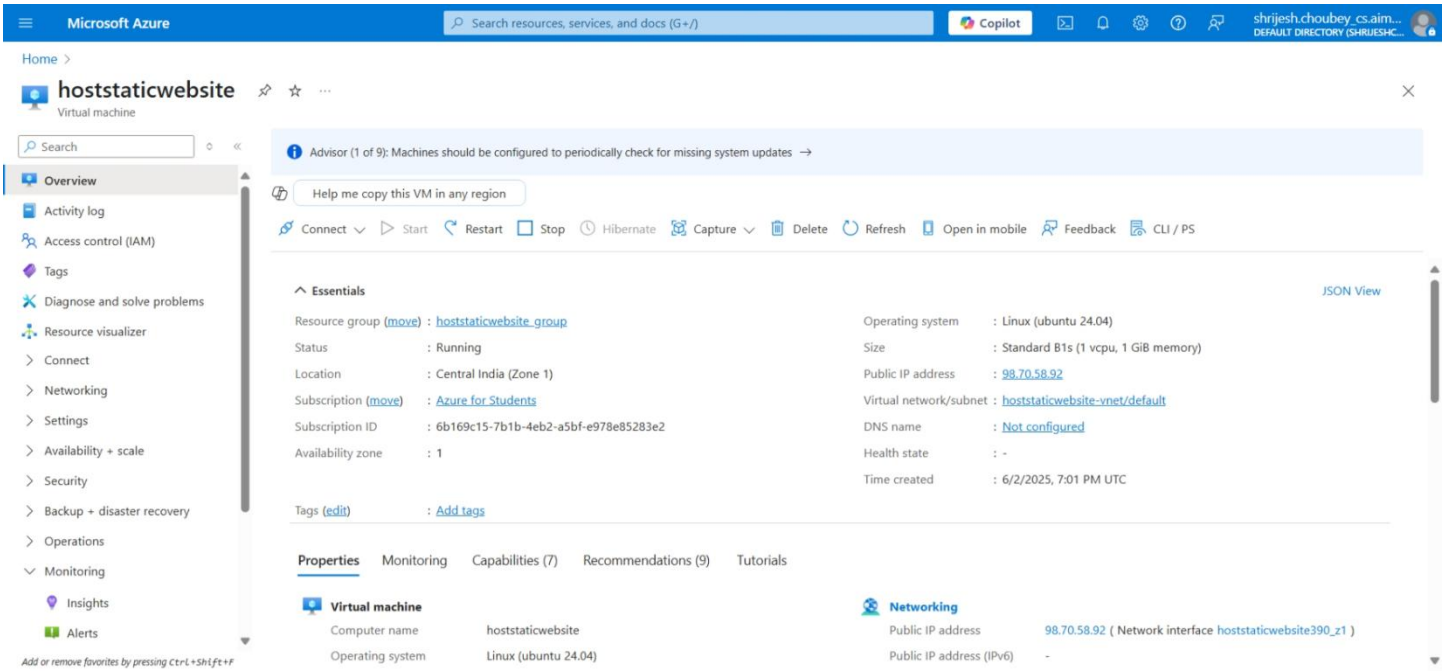
Cloud computing provides scalable, on-demand computing resources over the internet, enabling businesses, developers, and students to leverage virtual infrastructure without the need for physical hardware. Among the various service models, **Infrastructure as a Service (IaaS)** offers the greatest flexibility, allowing users to provision virtual machines (VMs), configure network settings, install software, and control the server environment completely.

Microsoft Azure, one of the leading cloud platforms, delivers robust IaaS capabilities through its Virtual Machine service. Azure VMs can run various operating systems, such as Linux or Windows, and are often used to host web servers, databases, applications, and static or dynamic websites.

In this project, we leverage Azure to create a **Linux-based (Ubuntu 24.04 LTS)** virtual machine and host a **static website themed around the Indian Army**. The website is developed using standard frontend technologies like **HTML and CSS**, along with multimedia components such as images and embedded videos. To serve the site, we configure **Nginx**, a lightweight and efficient open-source web server. The deployment process also involves **security configurations** via Azure's **Network Security Groups (NSG)** and **performance monitoring** using Azure's native diagnostic tools.

This hands-on exercise introduces students to real-world cloud deployment scenarios and provides practical insights into hosting, configuring, and securing a server in a cloud environment. It also strengthens foundational skills in Linux administration, web server setup, and cloud-based resource management—essential knowledge areas in today's DevOps and cloud-native development landscape.

Microsoft Azure

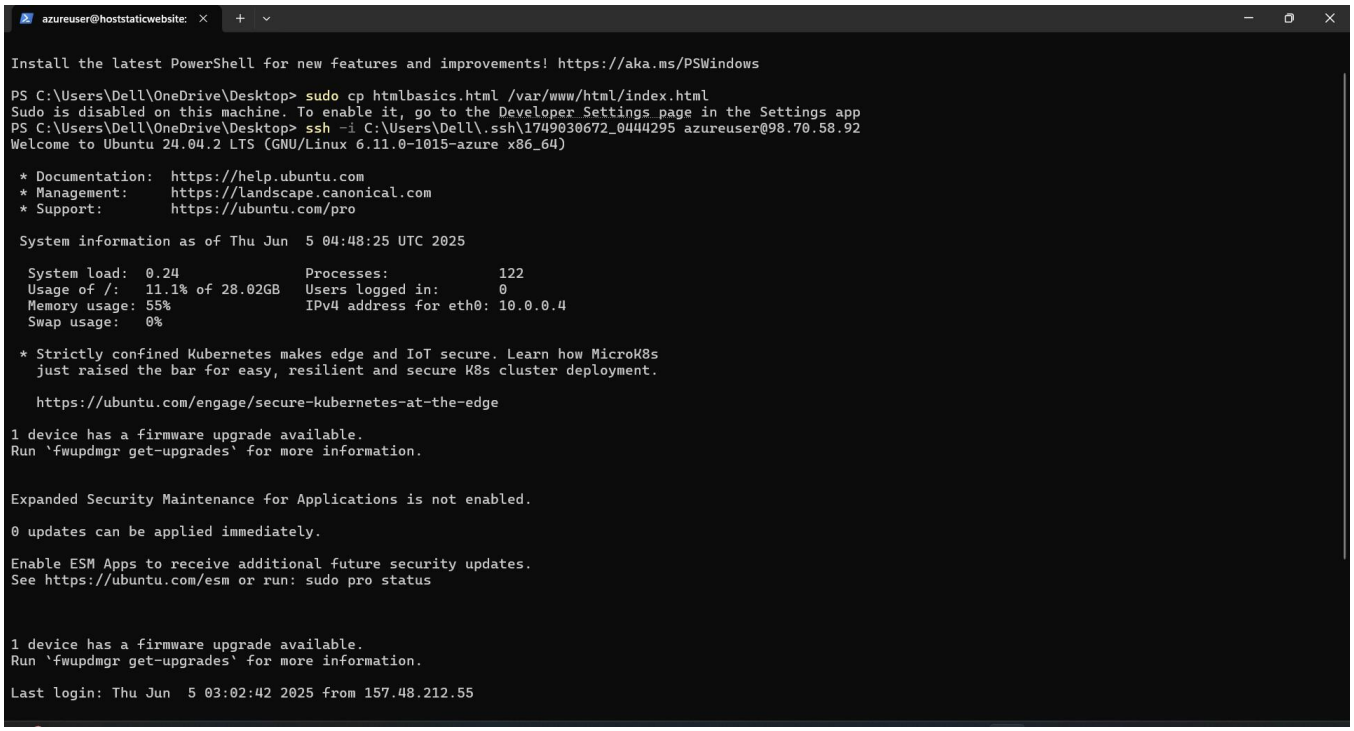


METHODOLOGY

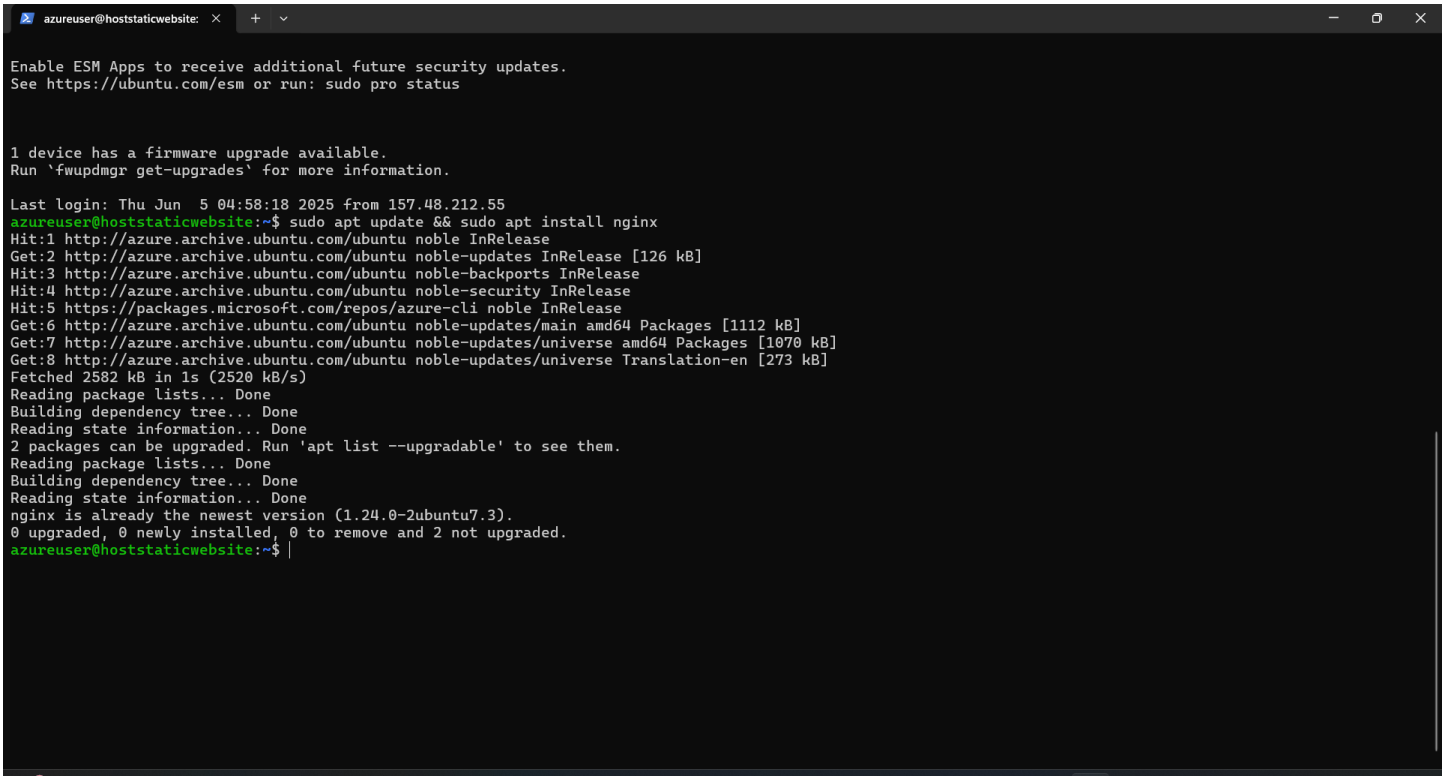
The project began with logging into Azure and creating a new Virtual Machine under the resource group named 'hoststaticwebsite_group'. The configuration included selecting 'Ubuntu 24.04 LTS' as the OS image and choosing a 'Standard_B1s' instance type for its cost-effectiveness.

SSH key authentication was set up to ensure secure login. Inbound port rules were configured to allow traffic on ports 22 (SSH), 80 (HTTP), and 443 (HTTPS). Nginx was installed using the command `sudo apt update && sudo apt install nginx`.

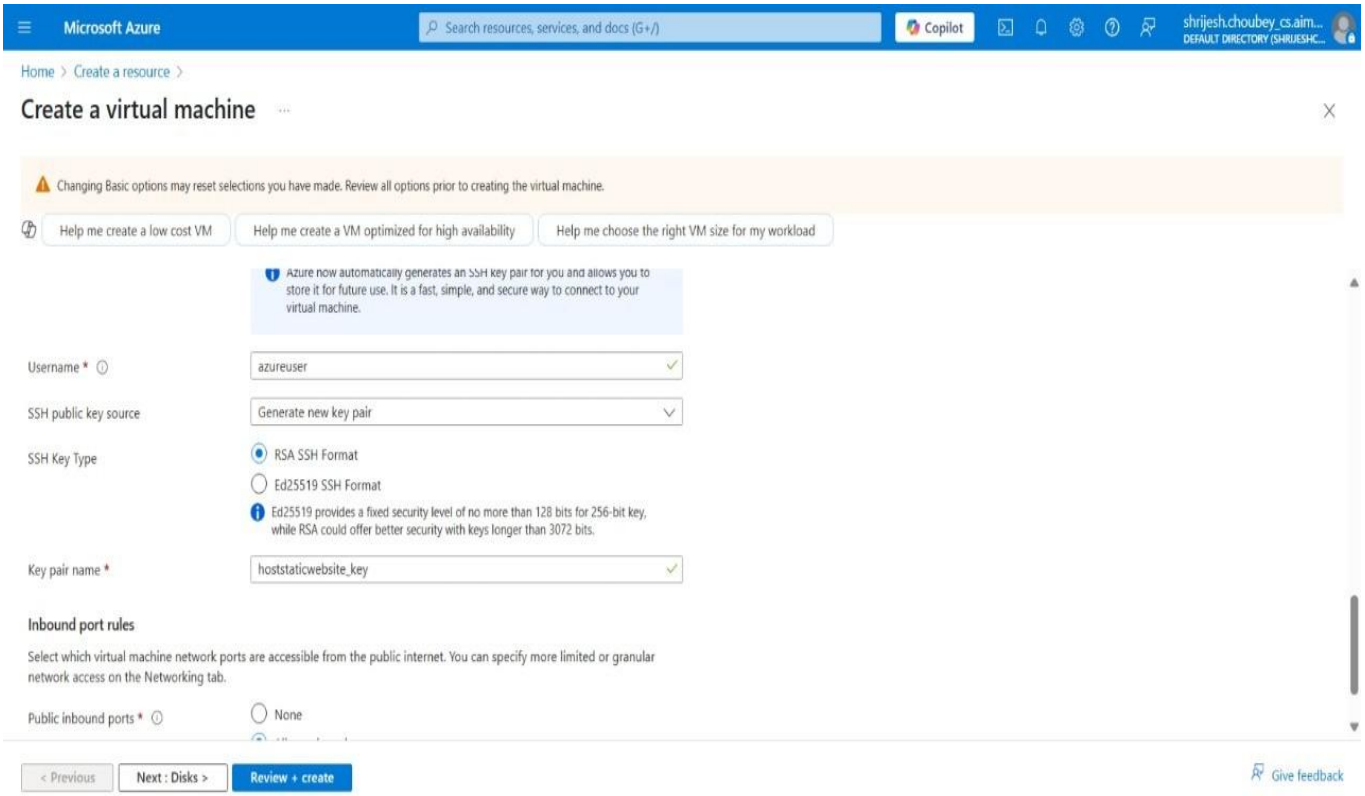
The static website files were then securely uploaded to the VM's Nginx web root directory at `/var/www/html` using SCP from the local machine. After uploading, the Nginx server was restarted and tested to verify it served the uploaded HTML page.



Azure VM Basic Configuration

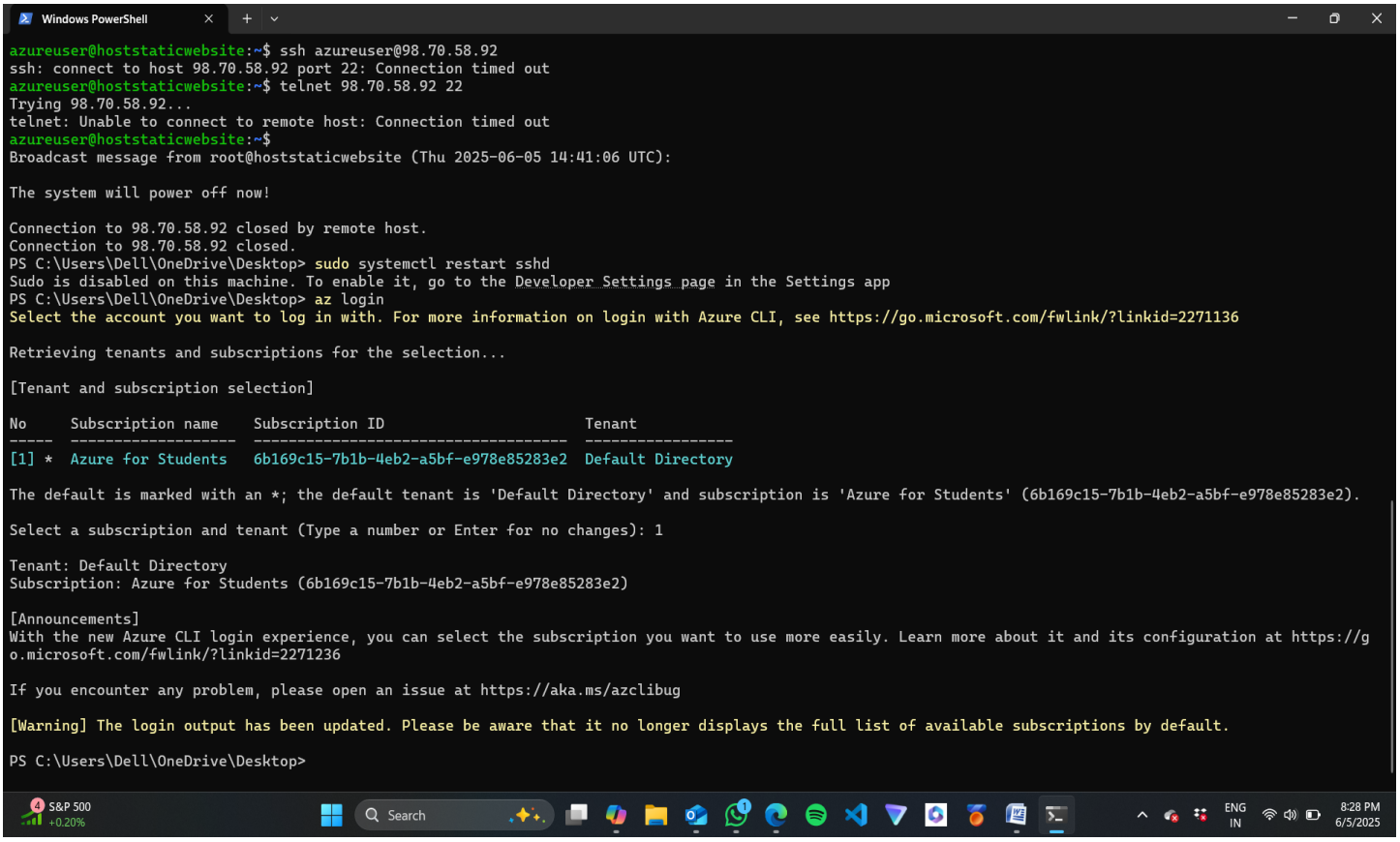


Nginx successfully installed

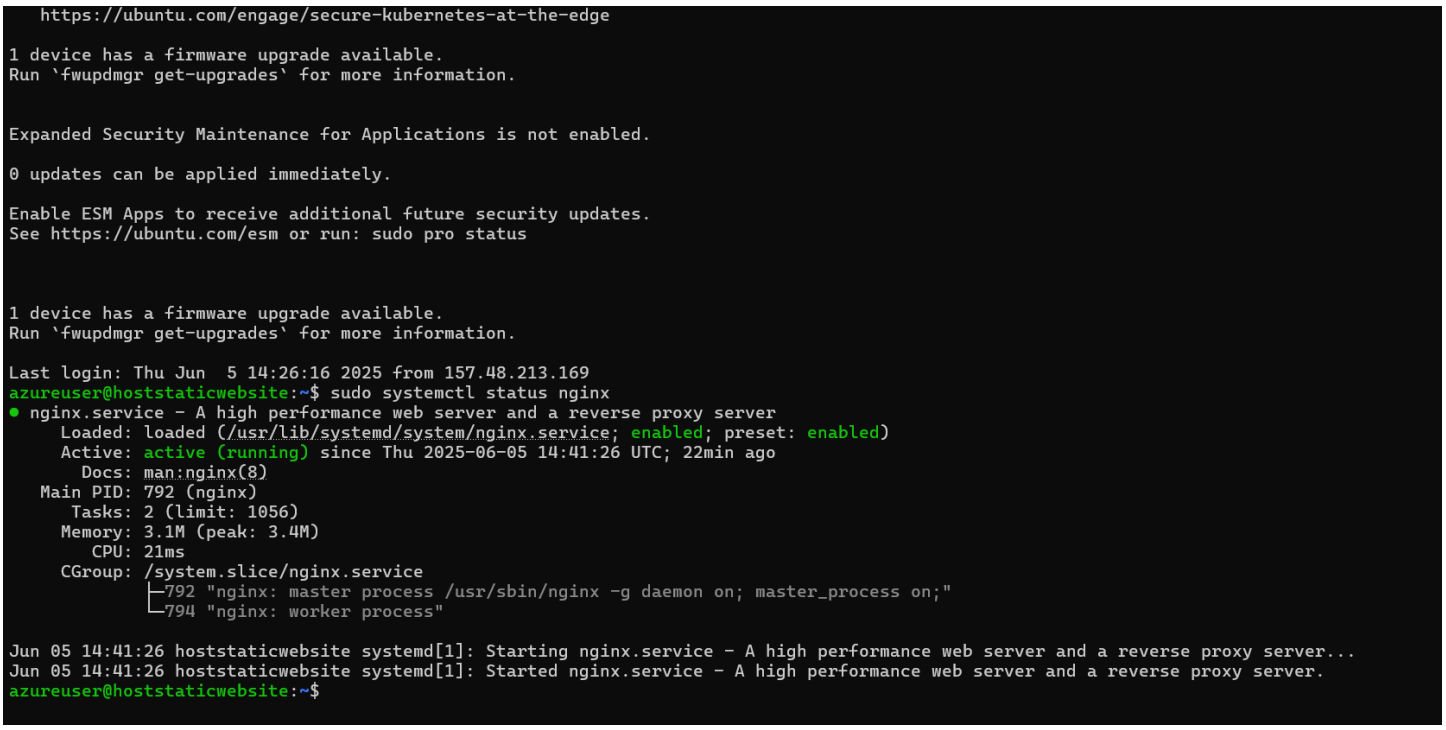


VM Deployment Confirmation

important codes that were used is –
ssh -i C:\Users\Dell\.ssh\1749030672_0444295 azureuser@98.70.58.92 – this is the code used to Connect to the remote server using ssh with using a specific private key.



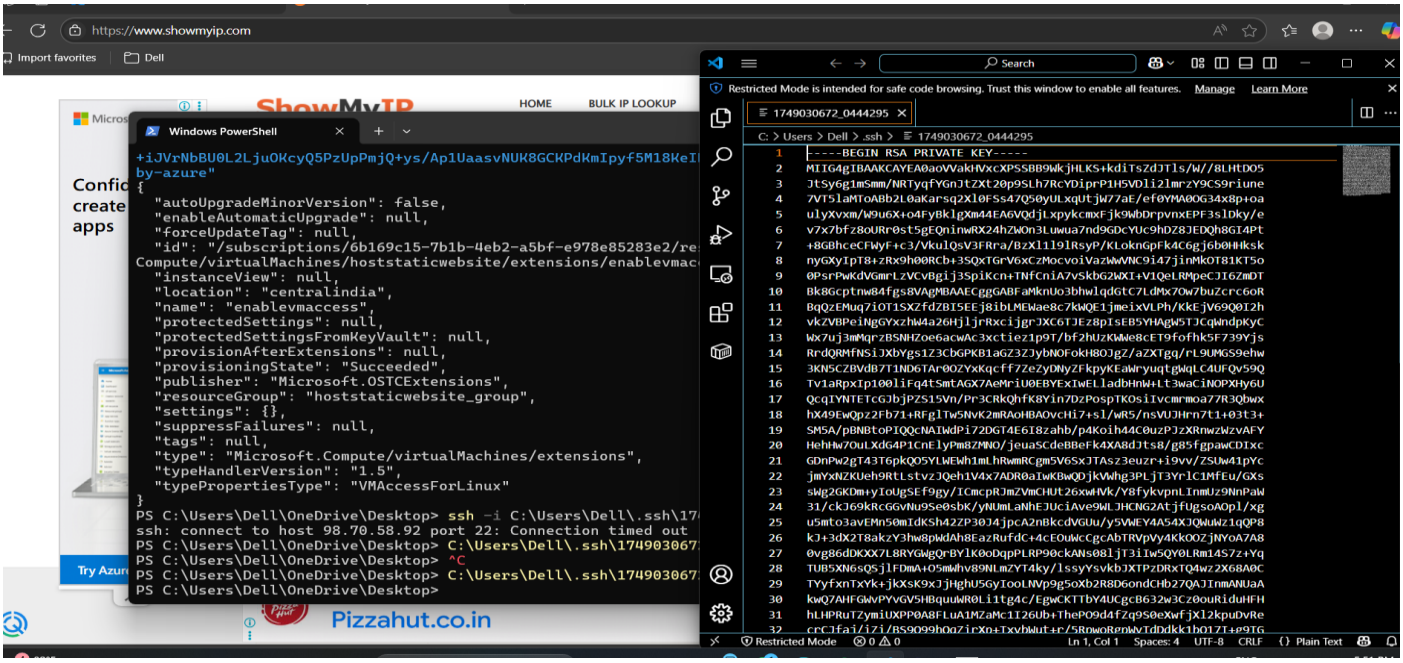
Code- az login- successful login



Nginx server status checked –successful

SCP command used - scp -i "C:/Users/Dell/.ssh/1749030672_0444295" "C:/Users/Dell/OneDrive/Desktop/coding folder/HTML-CSS-code/htmlbasics.html" azureuser@98.70.58.92:/var/www/html/

Cloud Computing Project Report



Got private key through command and connected to remote server with ssh command with private key

CODE EXPLANATION

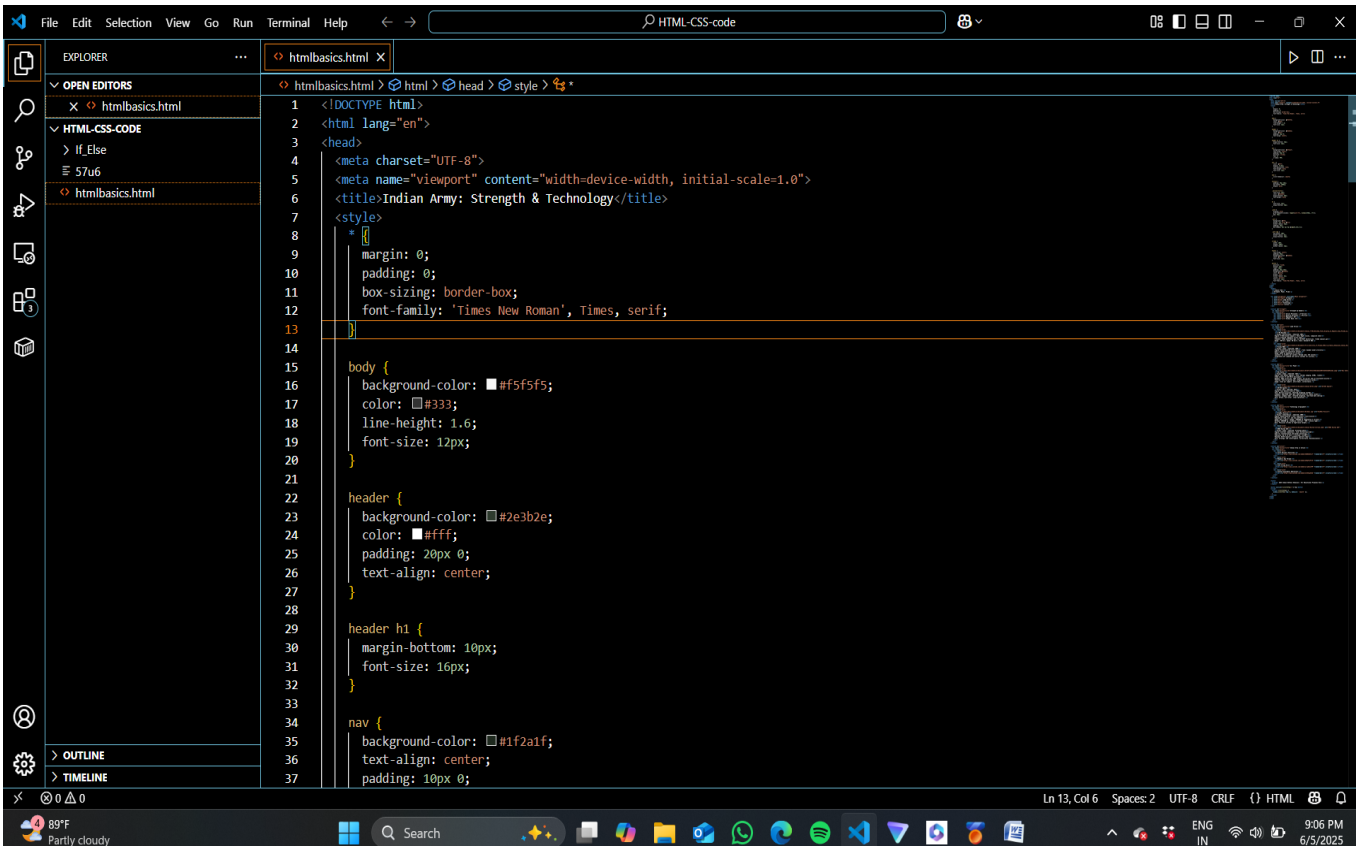
The HTML structure of the static website is designed with multiple semantic tags like `<header>`, `<nav>`, `<section>`, and `<footer>`. These provide clear divisions for user interface layout.

Styling was handled using embedded CSS, with a military green color scheme to reflect the Indian Army theme. Fonts were set explicitly to 'Times New Roman' to align with academic formatting requirements.

The website featured a responsive grid using:

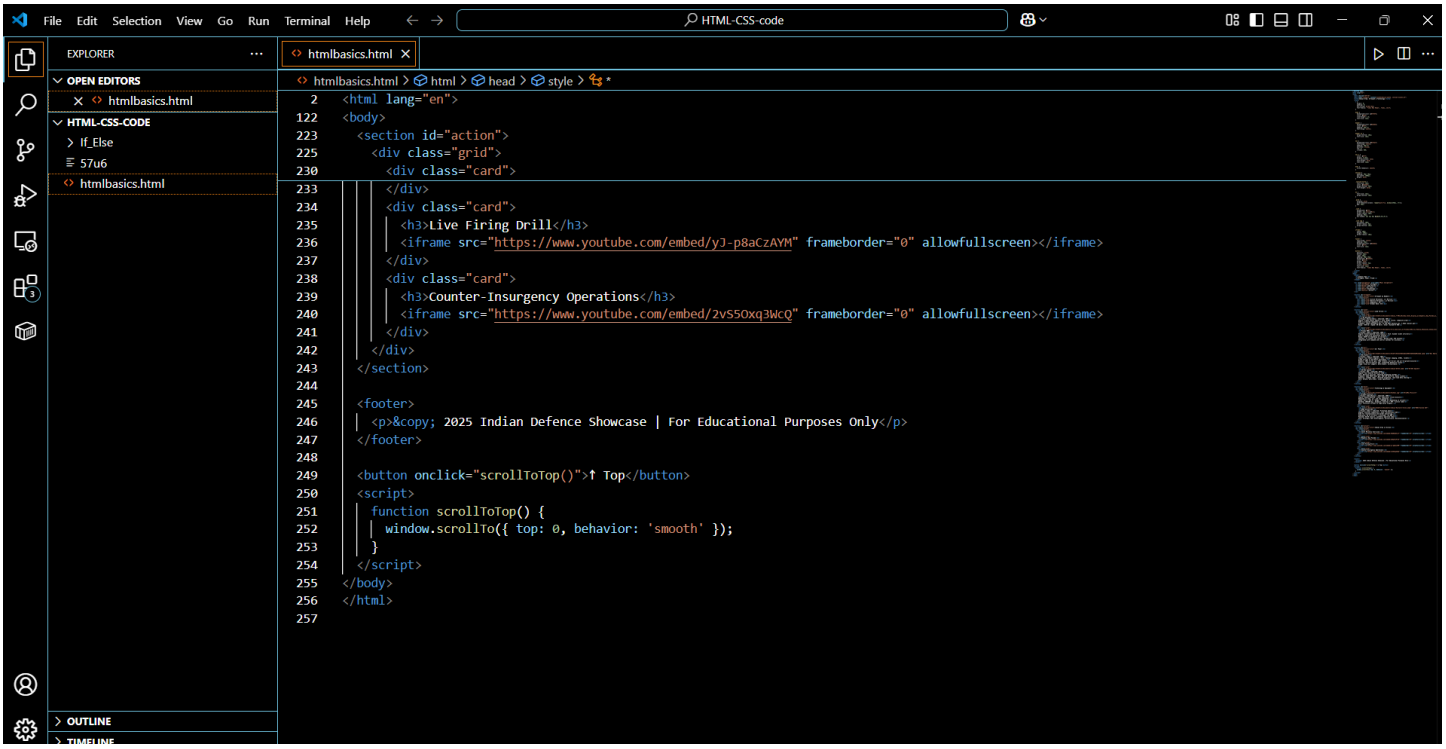
```
display: grid;
grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
```

This layout adapts to screen size, making the site mobile-friendly. Additional features included image cards, YouTube video embeds for military showcases, and a 'Back to Top' button using JavaScript scroll behavior.



CSS Styling

Cloud Computing Project Report

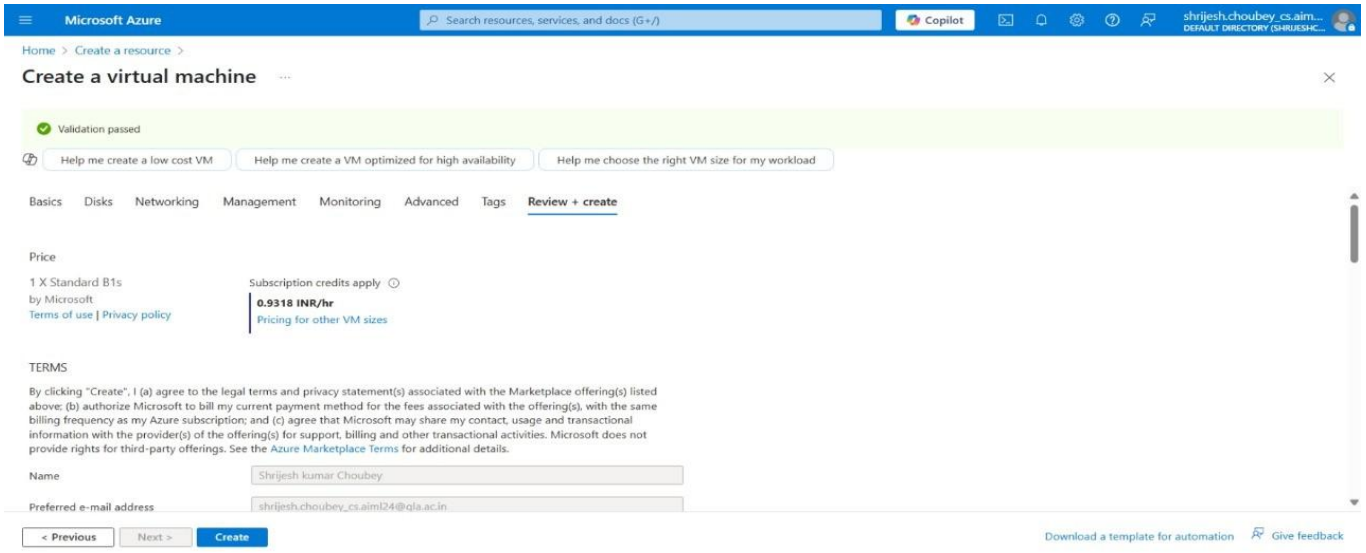


HTML code – structure

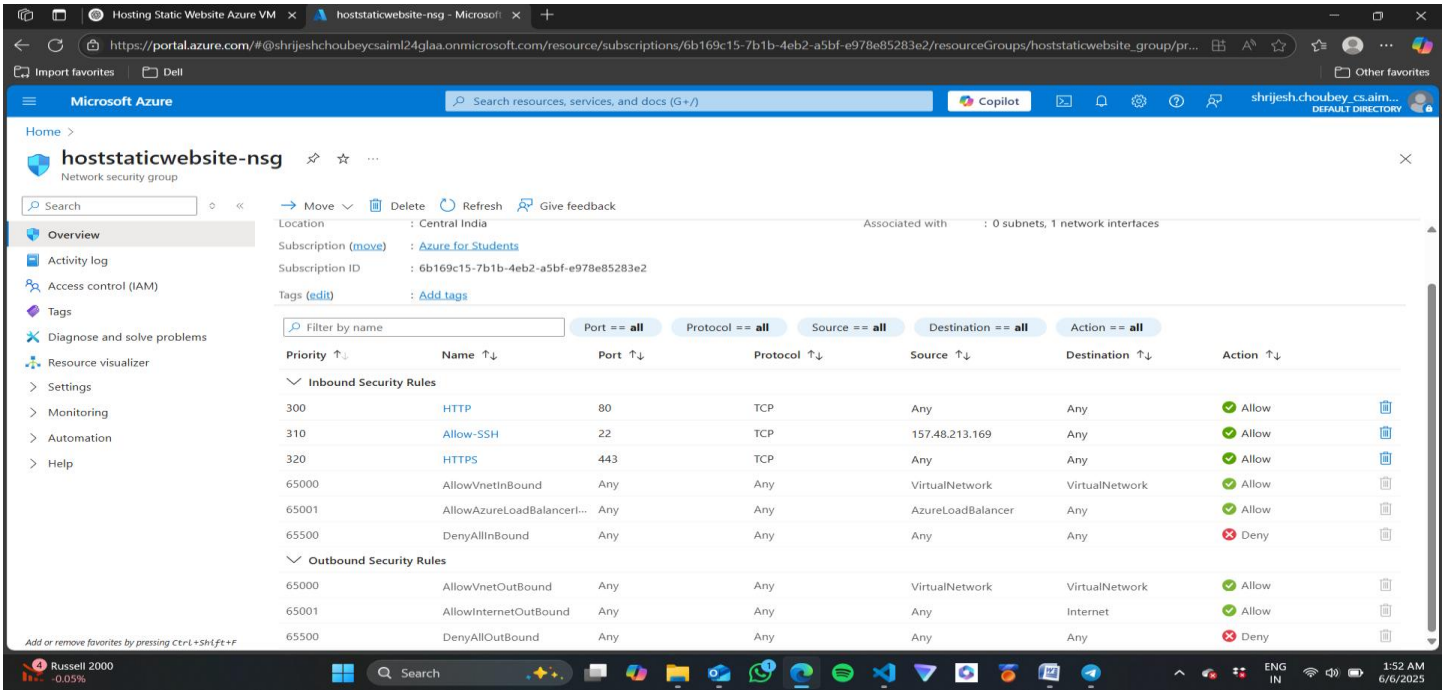
NETWORK SECURITY CONFIGURATION

The VM's security posture was managed using Azure's Network Security Groups (NSG). Inbound rules were added to allow only HTTP (80), HTTPS (443), and SSH (22) traffic. SSH was restricted to the creator's IP to enhance access control.

These rules ensure minimum surface exposure to external threats. No unnecessary ports were left open. Azure also allows tracking of security metrics through its built-in security center, which helps identify misconfigurations.



VM DEPLOYMENT SUCCESSFUL

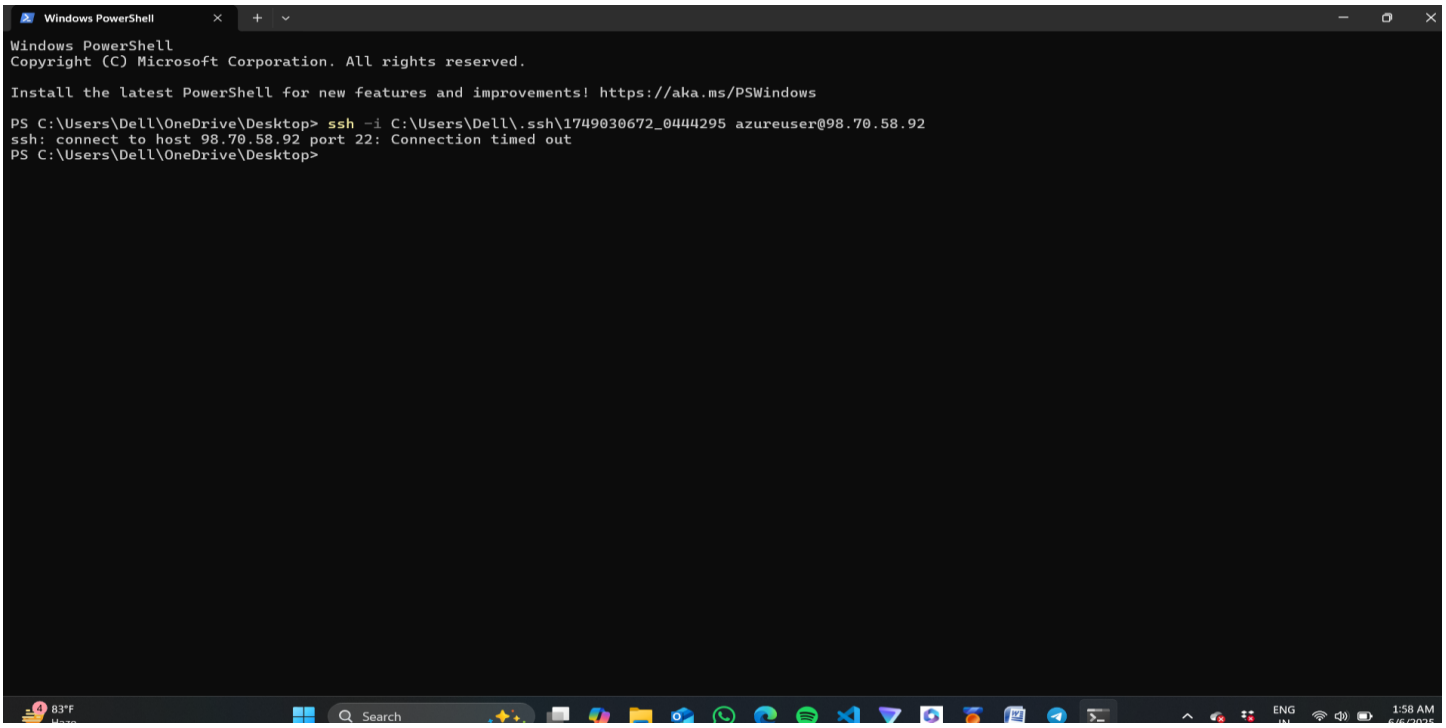


NSG Rules for HTTP, HTTPS, SSH

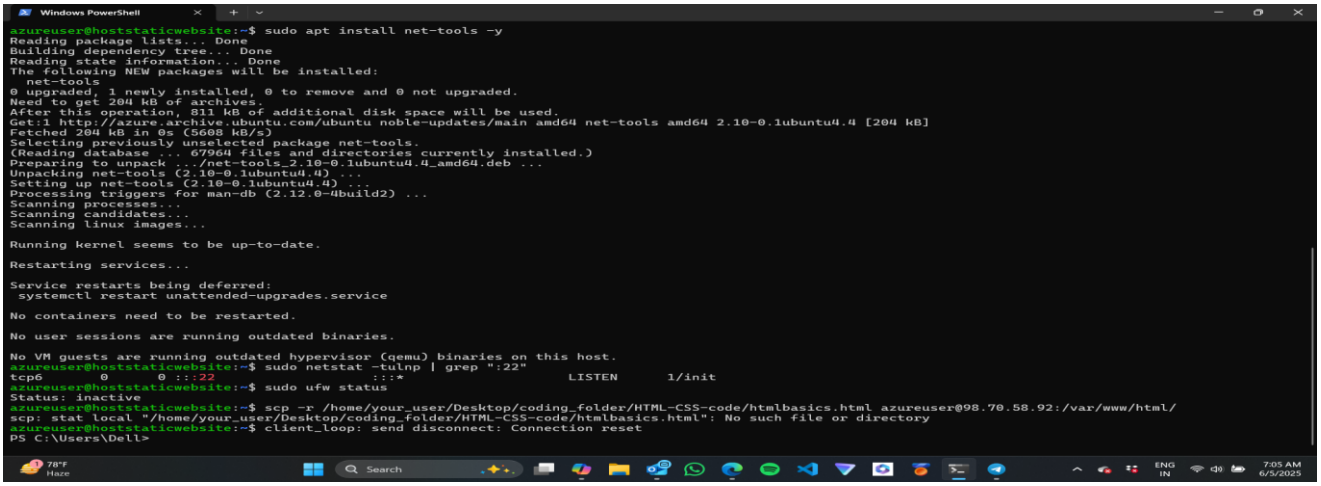
DEPLOYMENT CHALLENGES

During deployment, the initial issue faced was SSH and HTTP port inaccessibility due to NSG misconfiguration. A `telnet` test revealed port blocks.

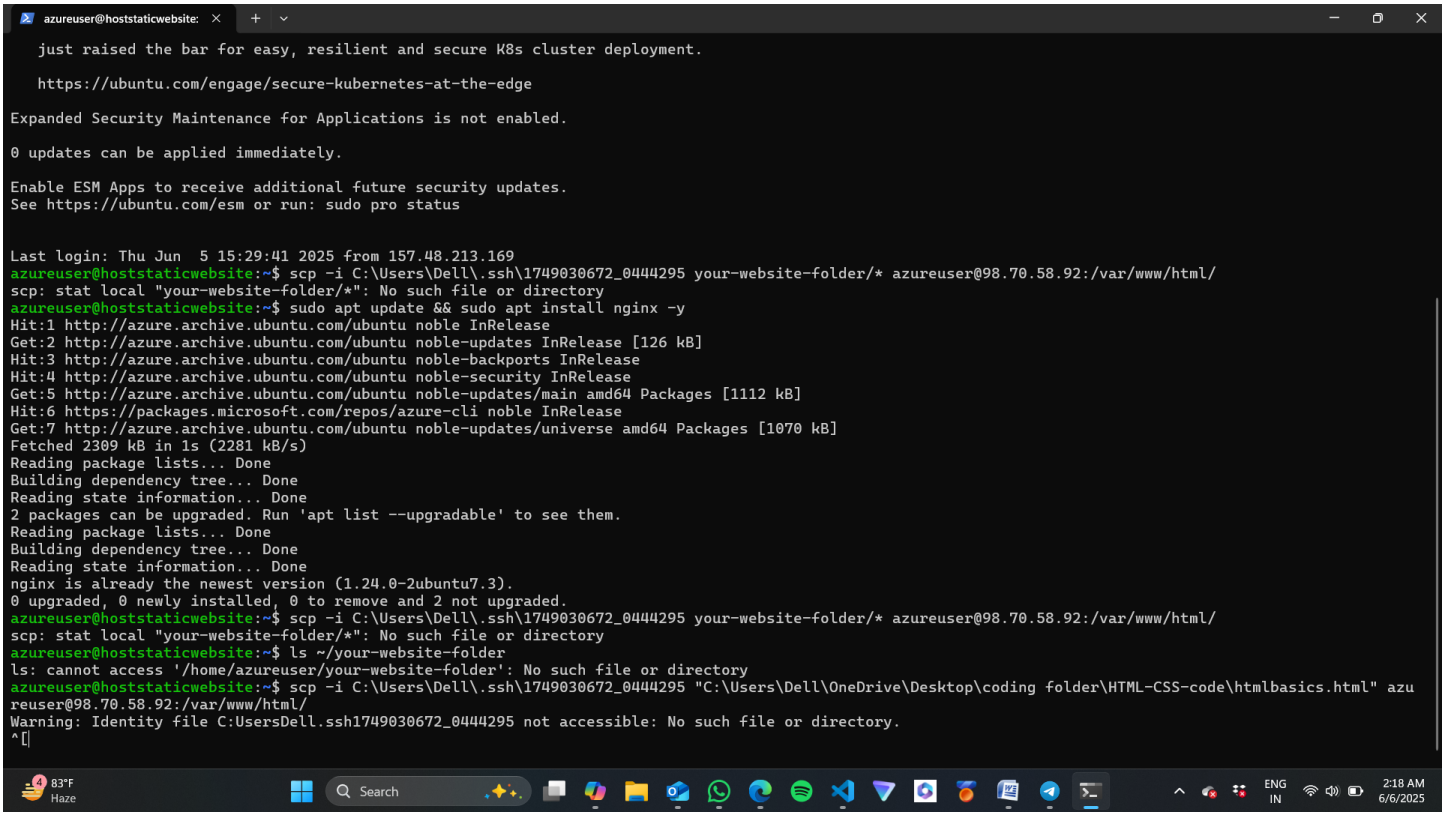
The solution was to manually edit NSG rules to ensure ports 22 and 80 were open to inbound connections. Image path issues due to local paths were fixed by correctly copying assets into the `/var/www/html/` folder and referencing them using relative links.



ssh connection timed out – due to port - 22



other major problems faced due to sudo command not operating properly



Errors occurred with secured copy protocol

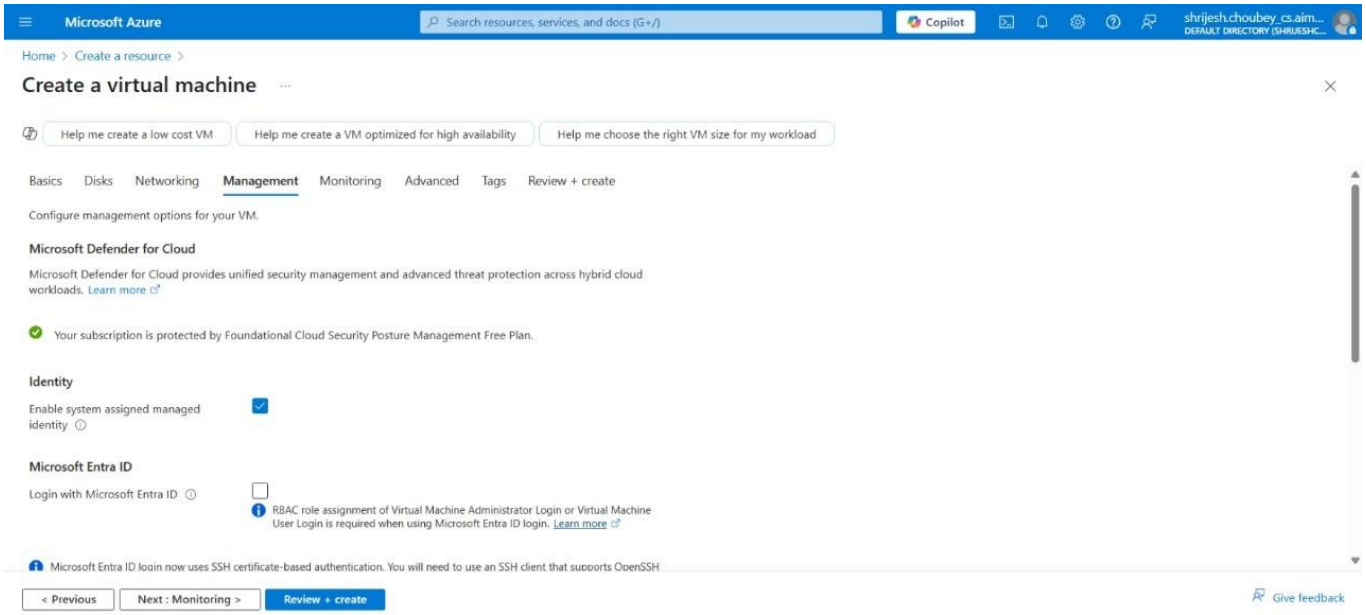
TESTING & VALIDATION

After deployment, the site was tested by visiting the public IP in a browser (`http://98.70.58.92/htmlbasics.html`). The site loaded successfully on both desktop and mobile browsers.

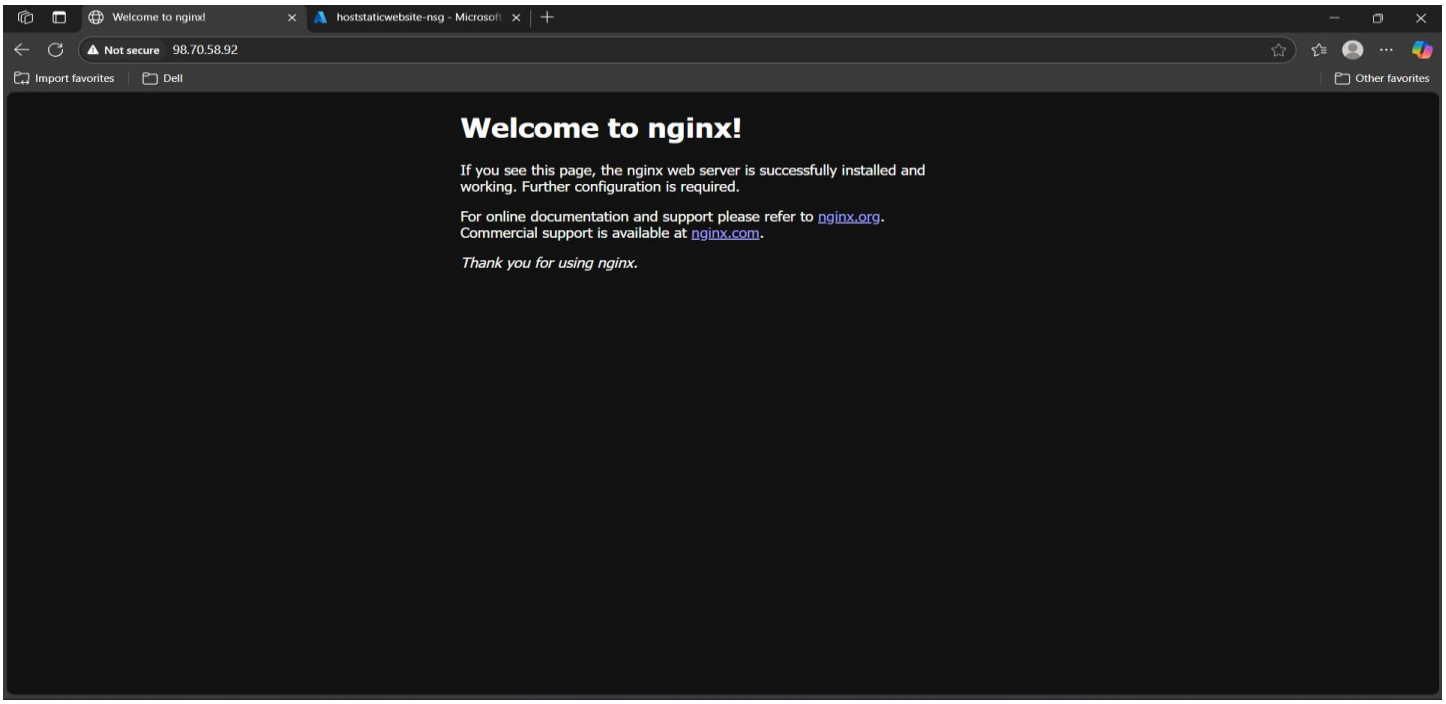
Additionally, Nginx service status was confirmed with:

```
systemctl status nginx
```

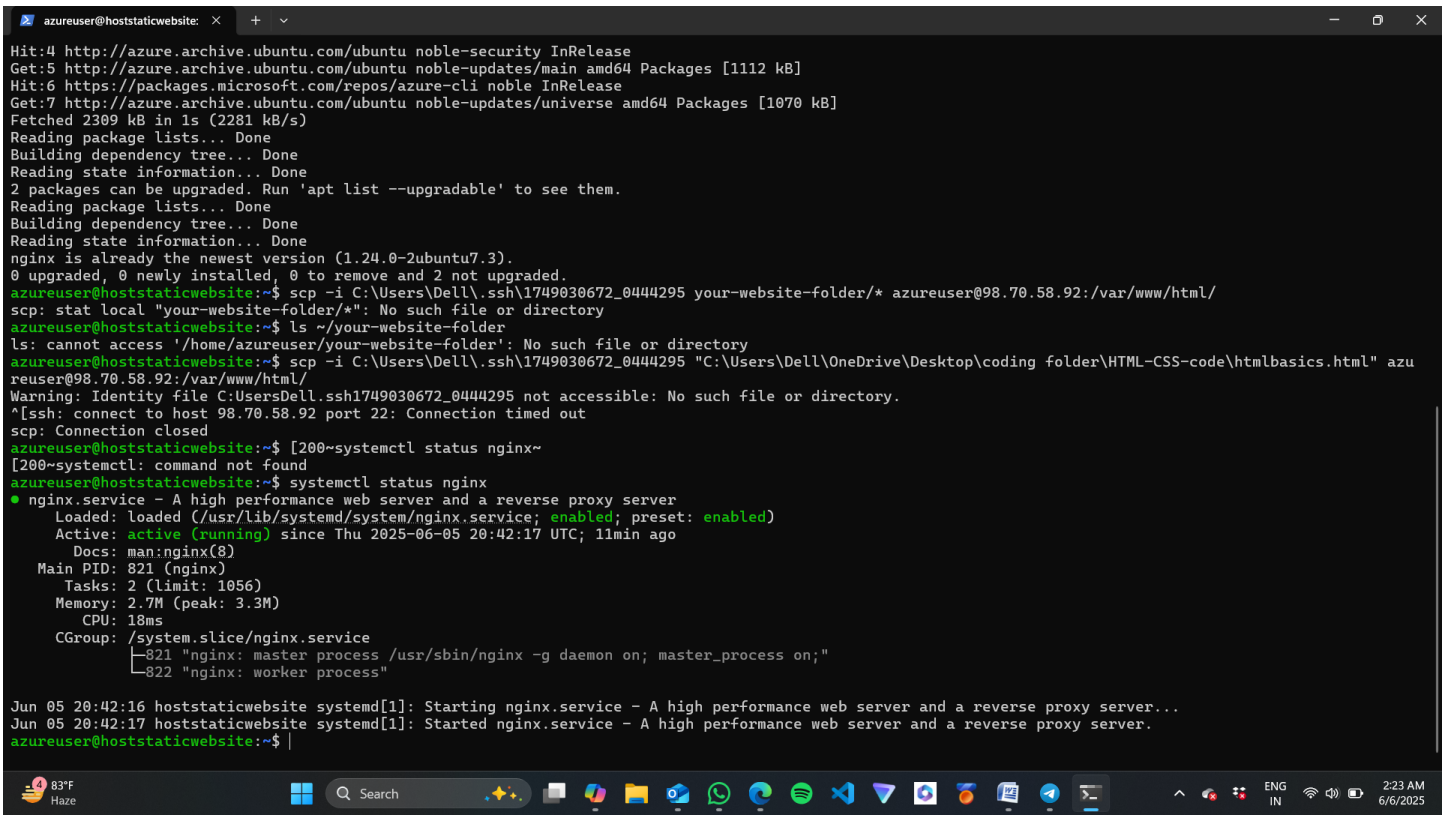
The Azure dashboard was checked to confirm that the VM status was 'Running' and inbound port traffic was being handled correctly.



Deployed Website Preview



Nginx service status confirmed



System status - Active

SECURITY BEST PRACTICES

Key practices followed include:

- SSH key authentication (disabling password login)
- IP whitelisting for SSH in NSG
- Only opening essential ports
- Installing only necessary software packages
- Keeping the system up to date with `apt upgrade`
- Avoiding root user login by using 'azureuser' with limited privileges.

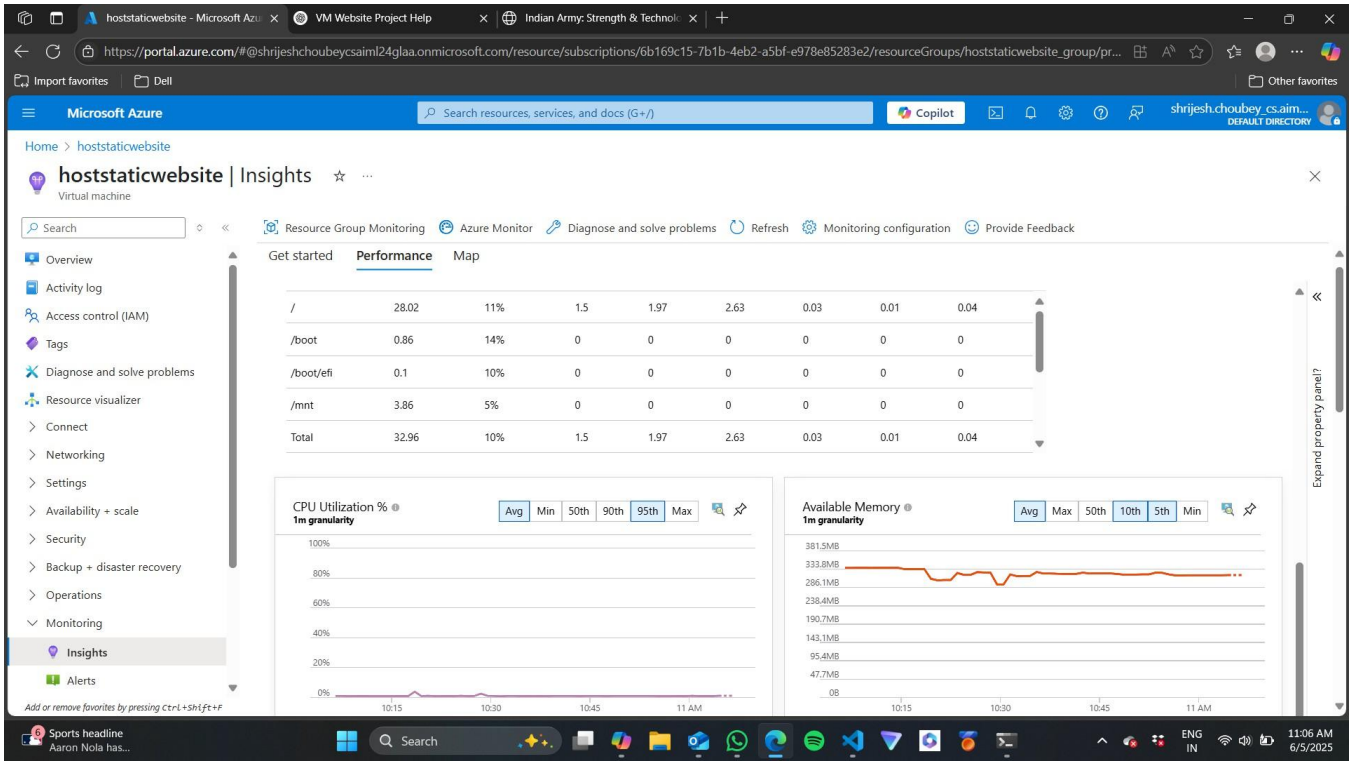
PERFORMANCE MONITORING

Azure Monitor and built-in diagnostics tools were used to track resource usage. CPU and RAM metrics showed the VM was operating well below capacity. This confirms that a Standard_B1s instance is sufficient for small static websites.

Installed Azure extensions include:

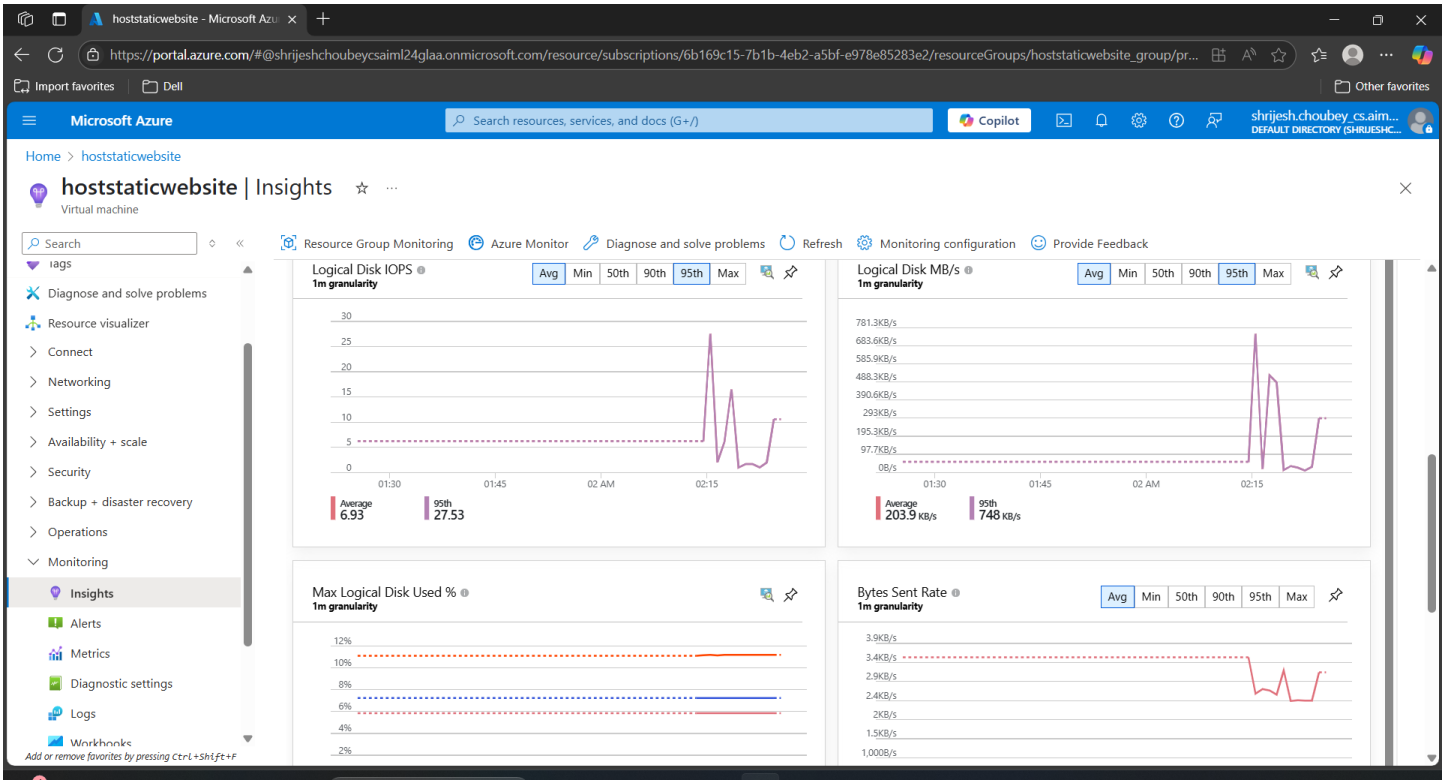
- Azure Monitor Linux Agent
- Azure Performance Diagnostics

These provide real-time metrics and enable performance tracing if needed.

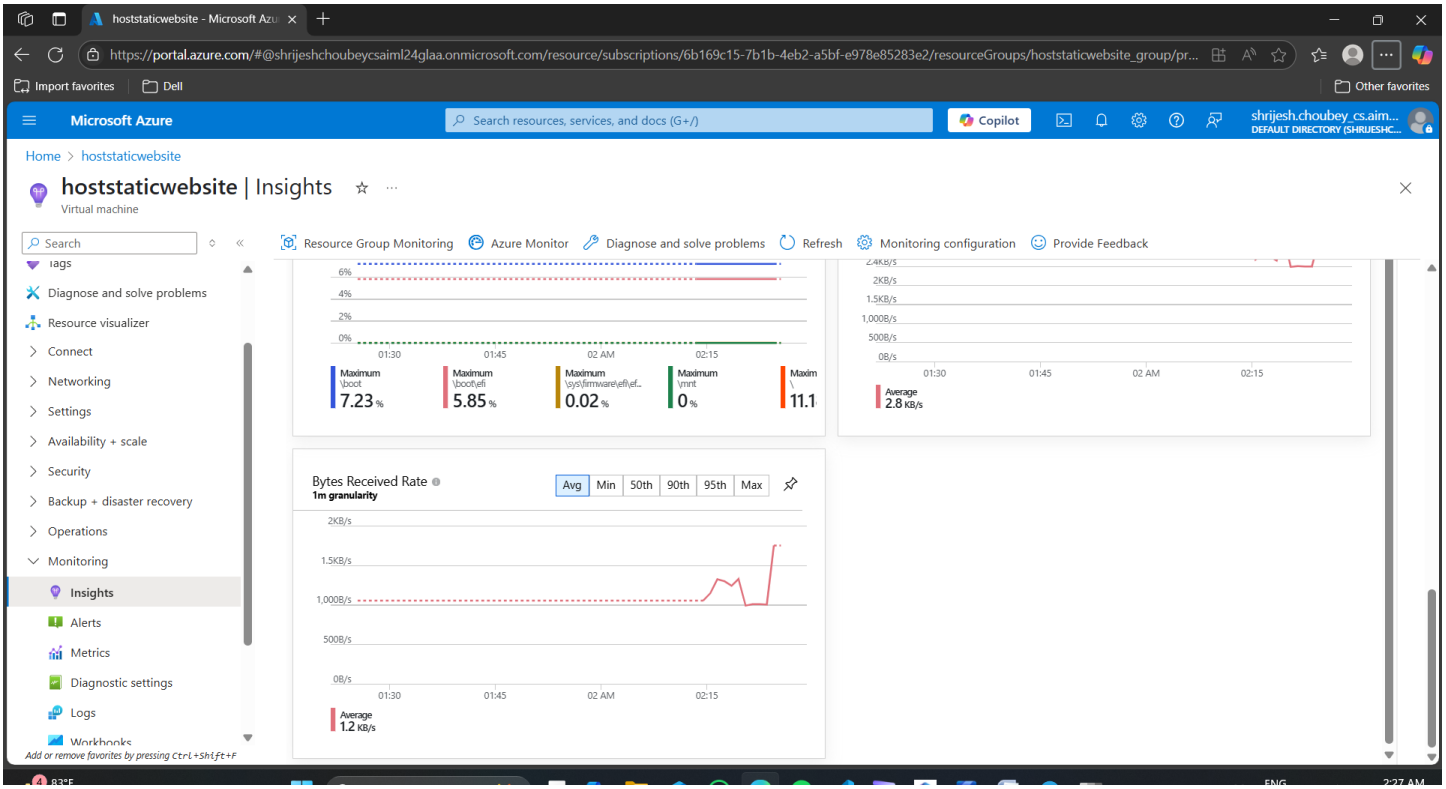


Azure Monitor - CPU and RAM Insights

Cloud Computing Project Report



Disk running performance

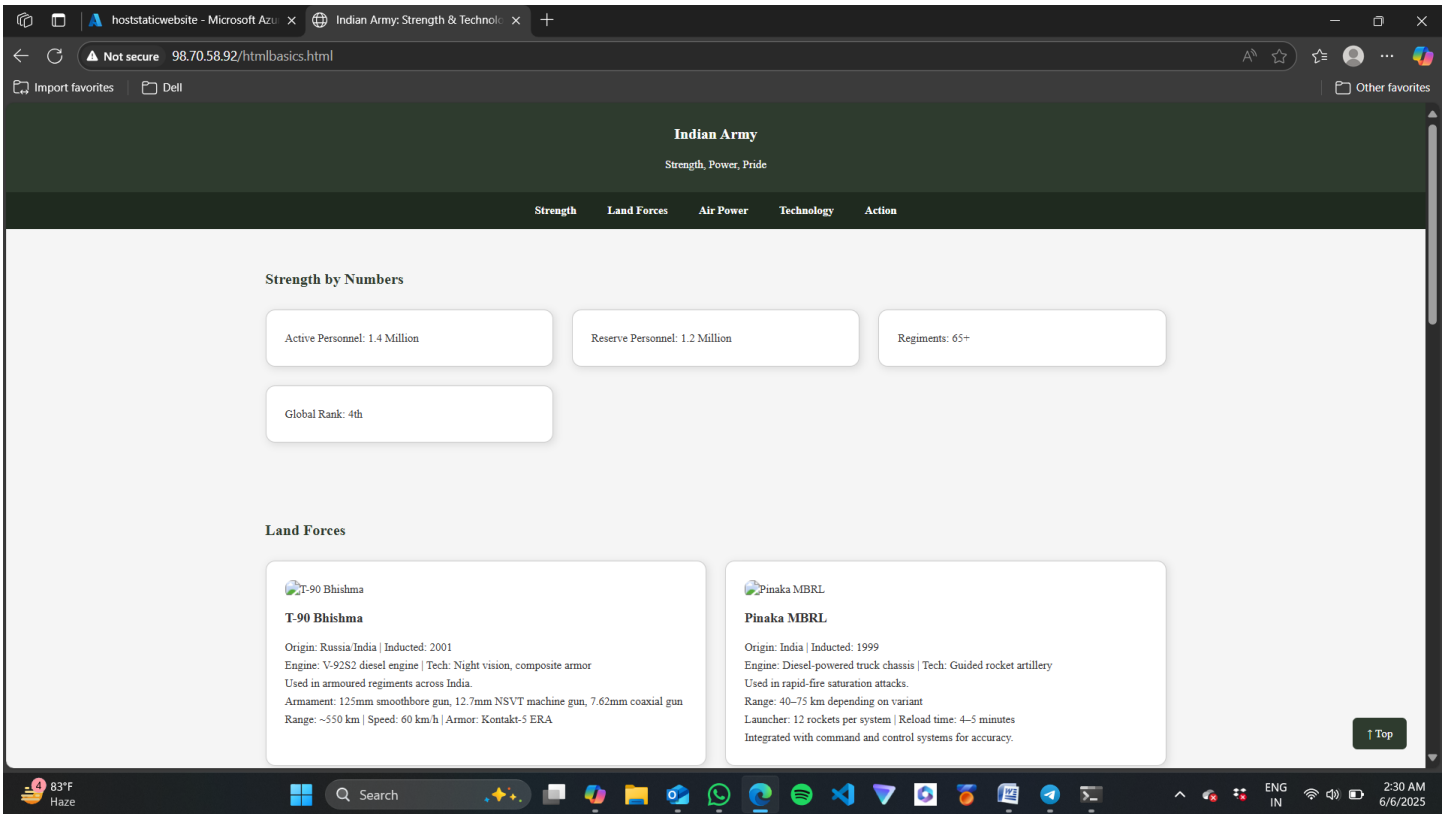


Other monitoring insights – bytes received rate

CONCLUSION

This project demonstrated a real-world example of provisioning a VM on Azure, configuring it securely, deploying a static website, and validating performance.

Students gain insights into cloud networking, web servers, file transfer security, and performance metrics. Future extensions could include deploying a React or Flask app, adding an SSL certificate via Let's Encrypt, or setting up a CI/CD deployment pipeline.



Website successfully working

REFERENCES

- Microsoft Azure Documentation: https://learn.microsoft.com/en-us/azure/
- Ubuntu Documentation: https://ubuntu.com/
- Nginx Documentation: https://nginx.org/en/docs/
- Mozilla Developer Network (HTML/CSS): https://developer.mozilla.org/
- ChatGPT by OpenAI (for technical explanations)