Technical Report:

Multi-Platform Cloud Deployment & Comparative Analysis

Optimizing Static Web Hosting for Scalability and Cost-Efficiency

Prepared by: Fahad Alshehri

Contact:

github.com/FahadAlshehrii

linkedin.com/in/f-alshehrii

Table of Contents

Summary

This report evaluates the deployment of a static Hugo-based web application across three distinct Microsoft Azure service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Object Storage. The objective is to determine the optimal hosting environment based on cost-optimization, service availability (SLAs), and operational overhead.

Implementation Overview

The deployment was executed across three environments to test the versatility of the Hugo static site engine:

A. Infrastructure as a Service (VM Deployment)

- **Environment: Linux Virtual Machine (Ubuntu).**
- **Web Server: Configured Nginx for high-performance static file serving.**
- **Workflow: Leveraged Git-based deployment for seamless file transfer from local environment to the production VM.**

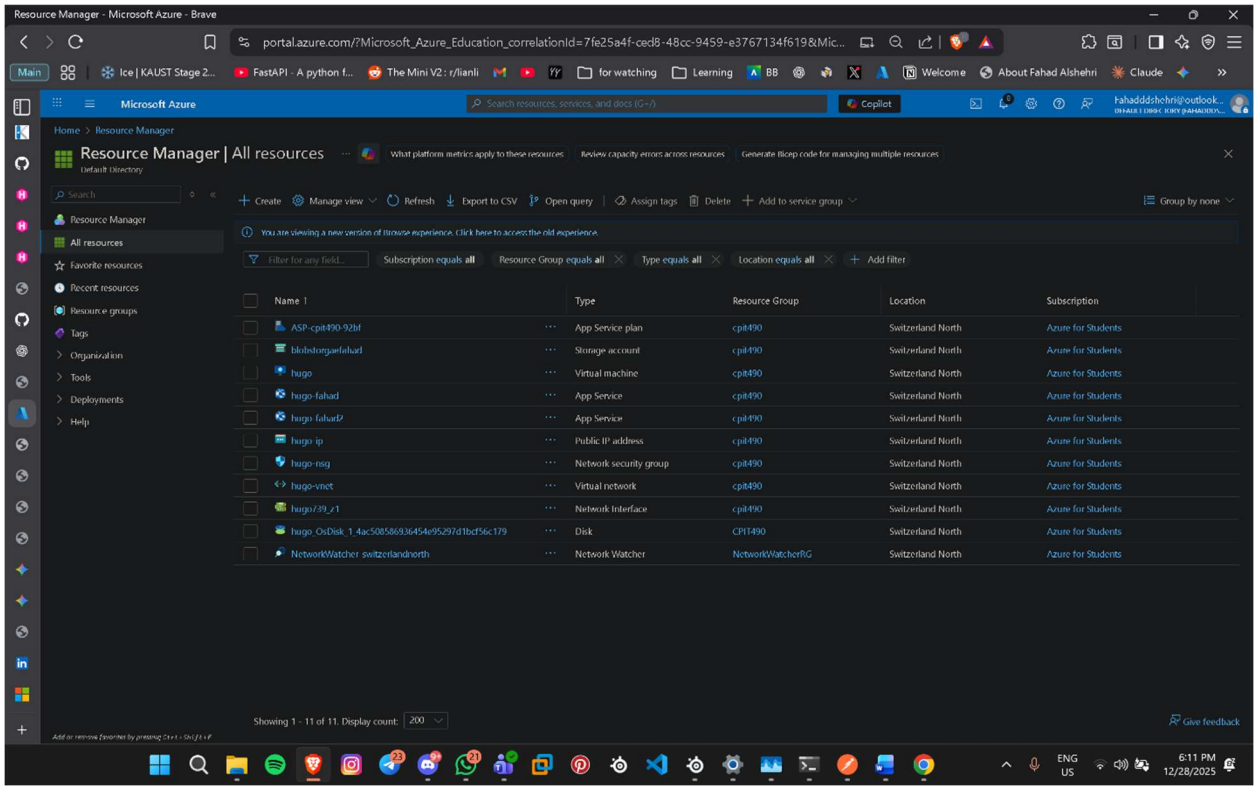
B. Platform as a Service (Azure App Service)

- **Environment: Managed PaaS (App Service).**
- **Deployment Method: Utilized SCM & FTP integration for optimized artifact deployment.**
- **Key Insight: Optimized the deployment speed by implementing a Zip-based artifact transfer via Kudu File Manager, significantly reducing deployment latency compared to individual file uploads.**

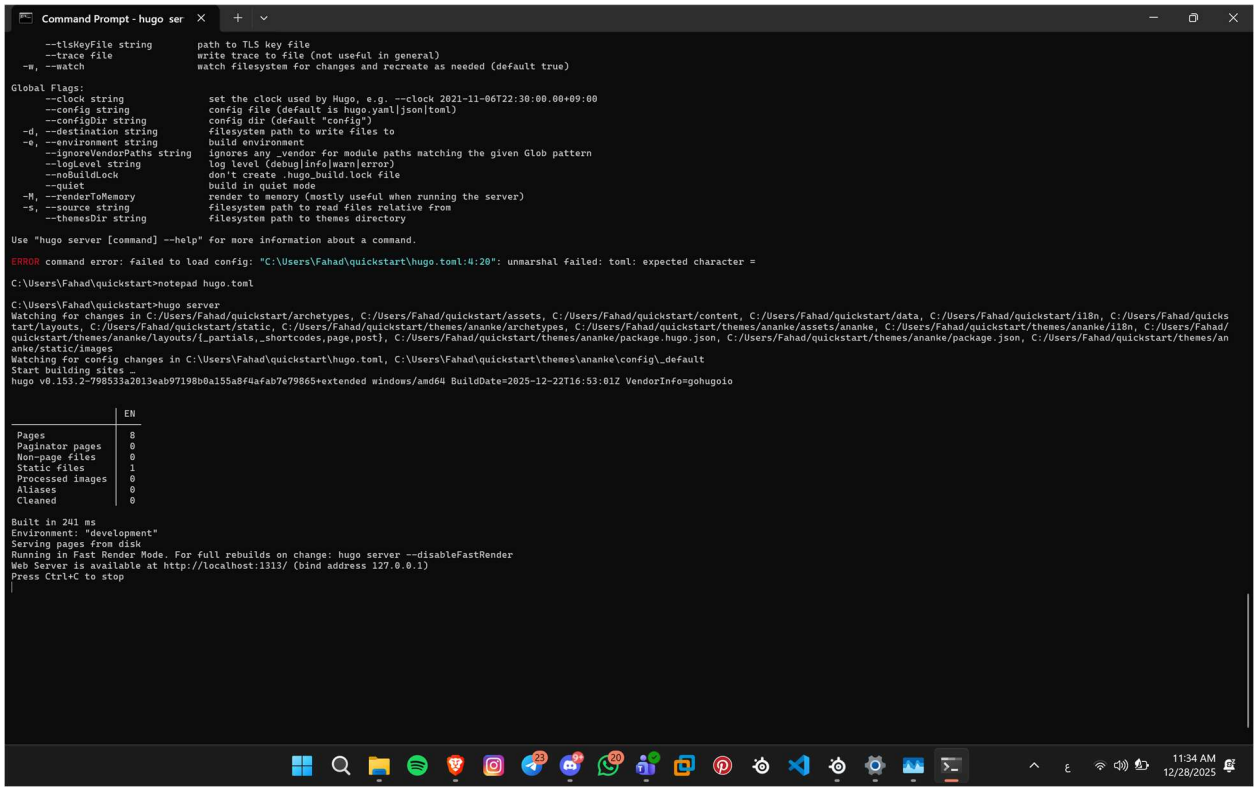
C. Cloud Object Storage (Azure Blob Storage)

- **Environment: Serverless Static Website Hosting via Azure Blob Storage.**
- **Configuration: Enabled anonymous public access and configured specific index/error document endpoints.**
- **Storage: Direct deployment to the $web container, bypassing server management entirely.**

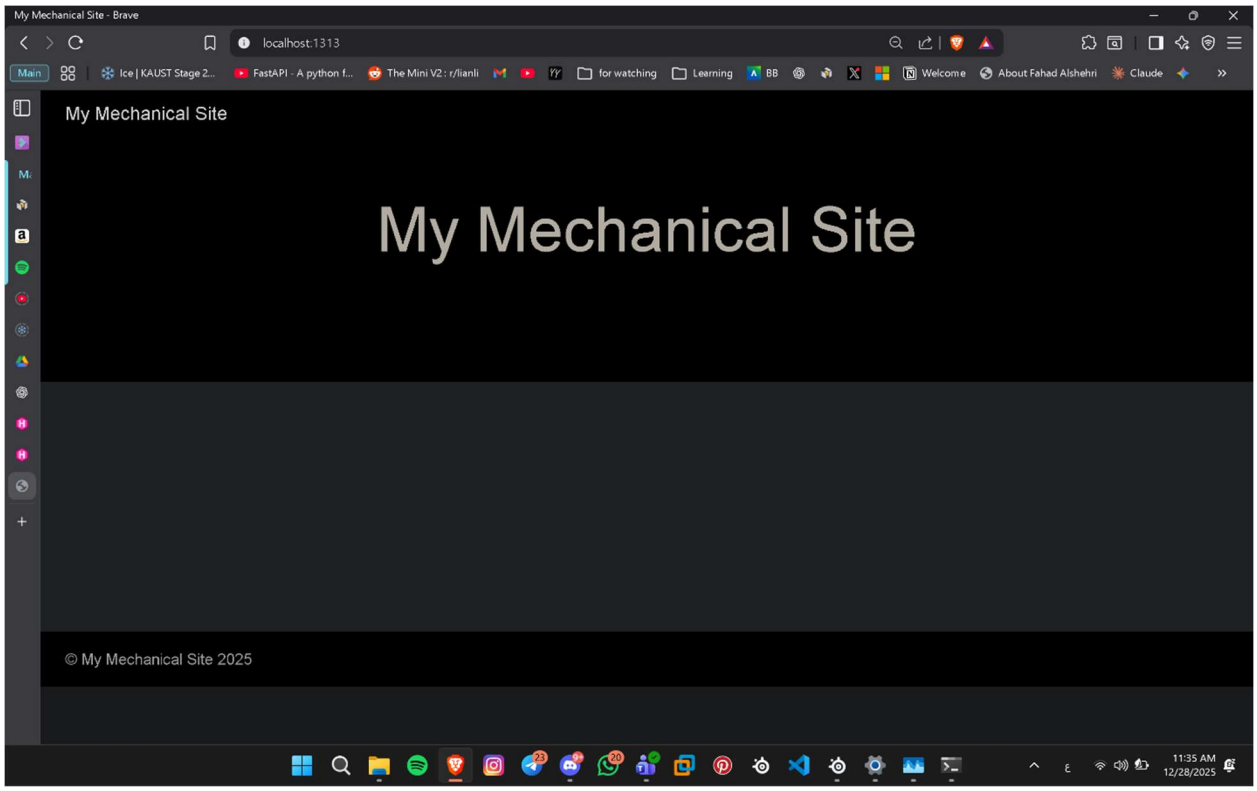
All resources overview



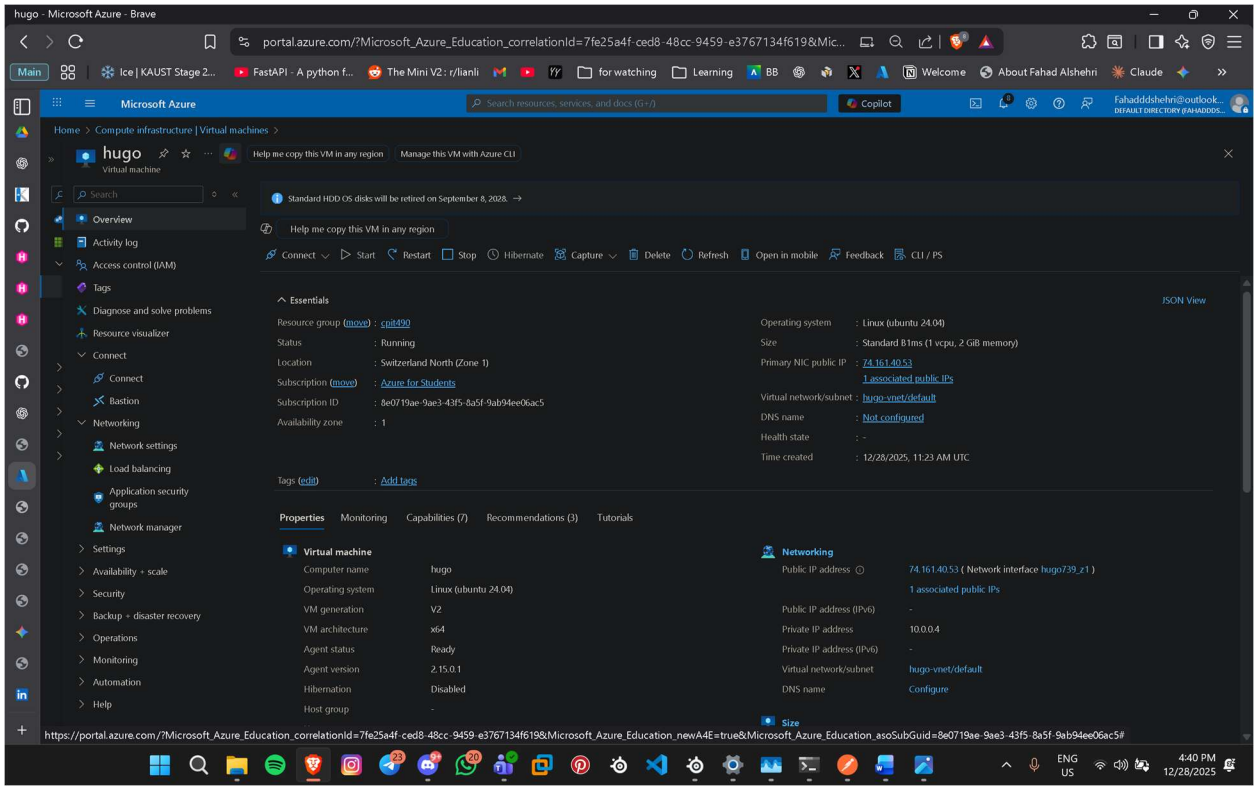
Prerequisites: downloading hugo and testing it



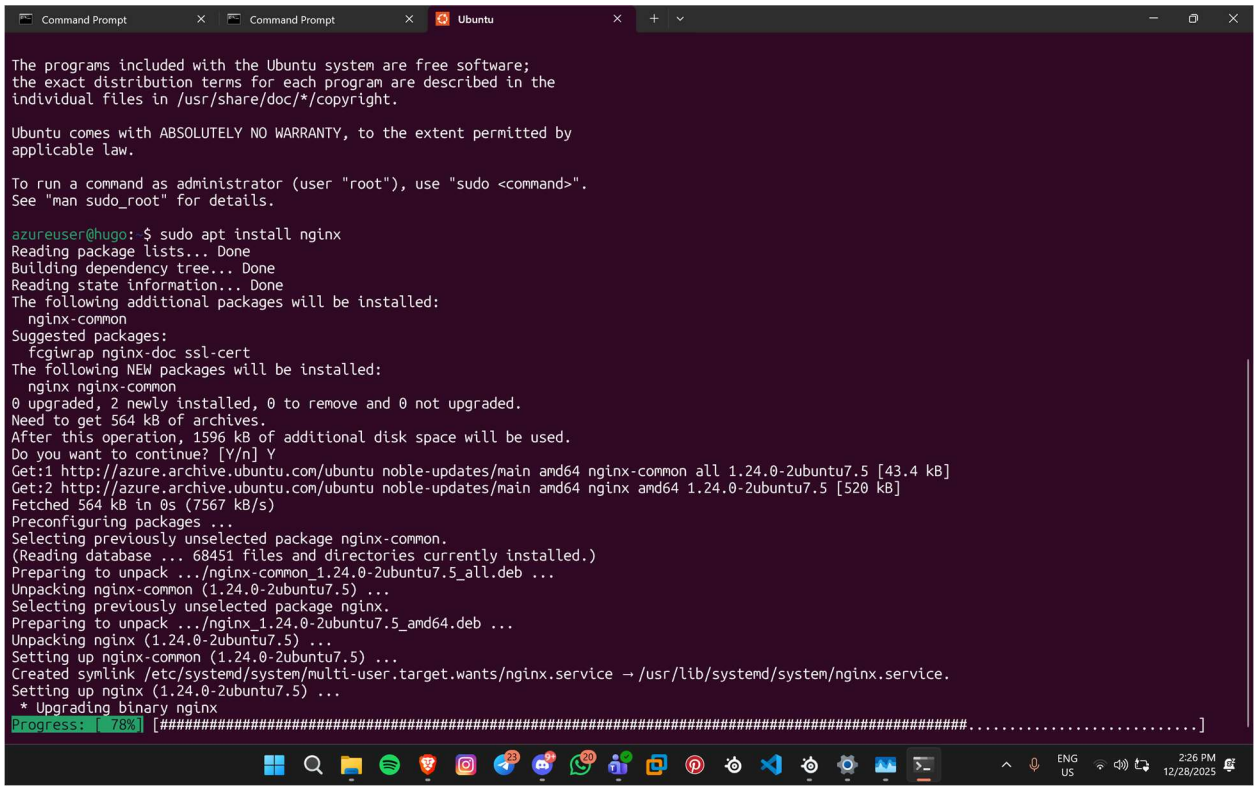
Accessing hugo site before doing deployments



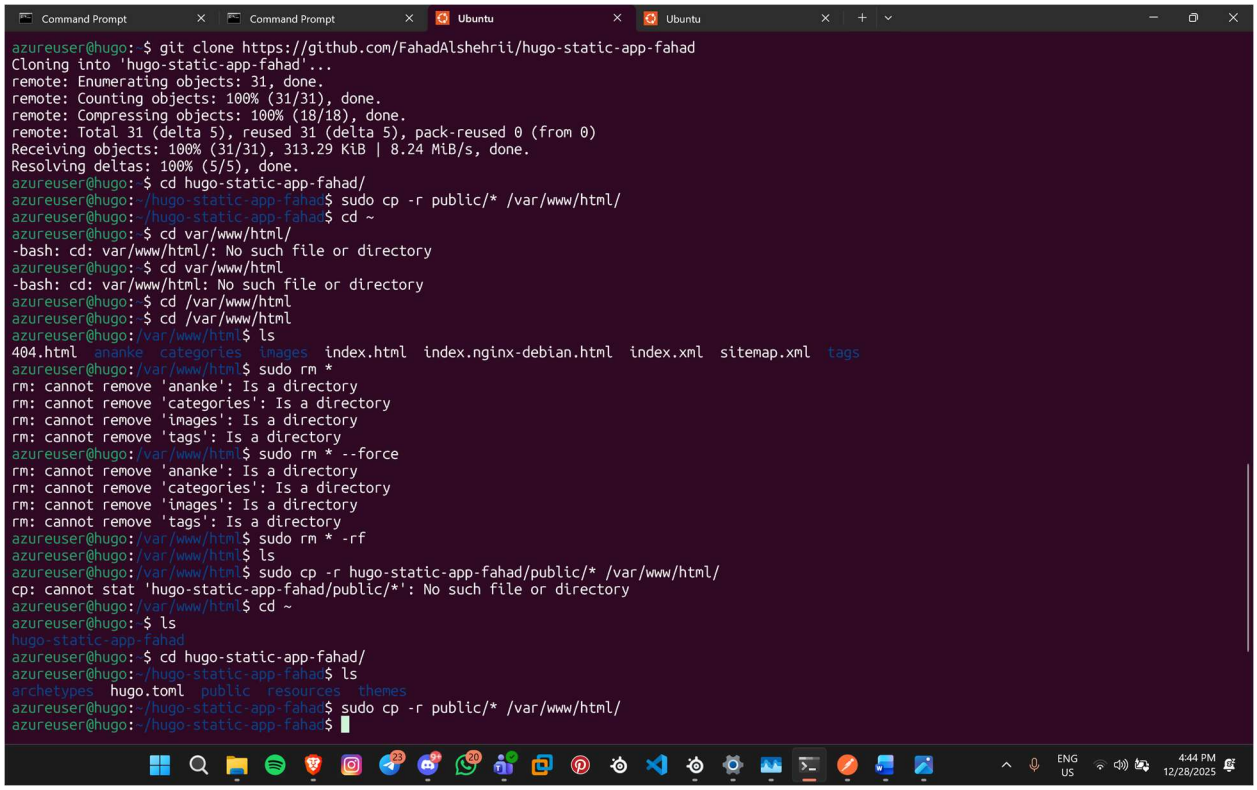
VM Deployment: VM creation



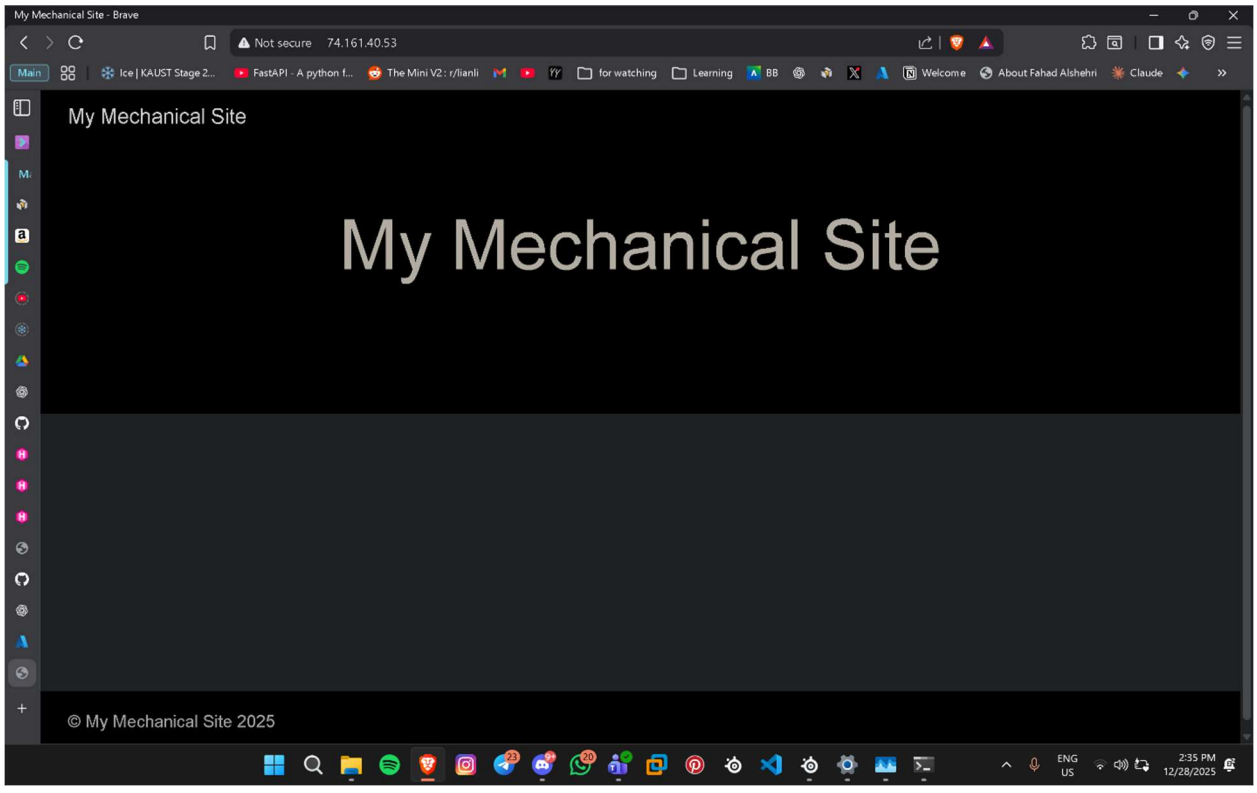
Downloading Nginx on VM



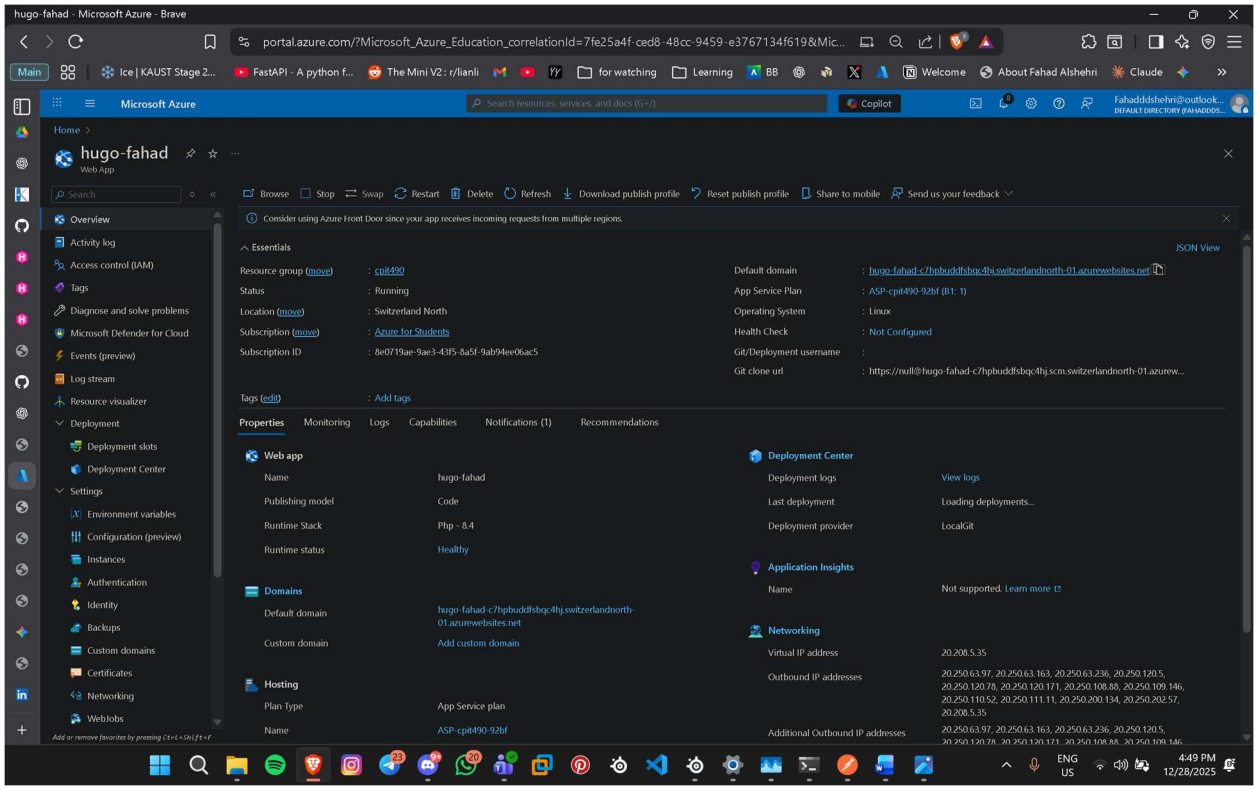
Moving files to VM using git clone to deploy



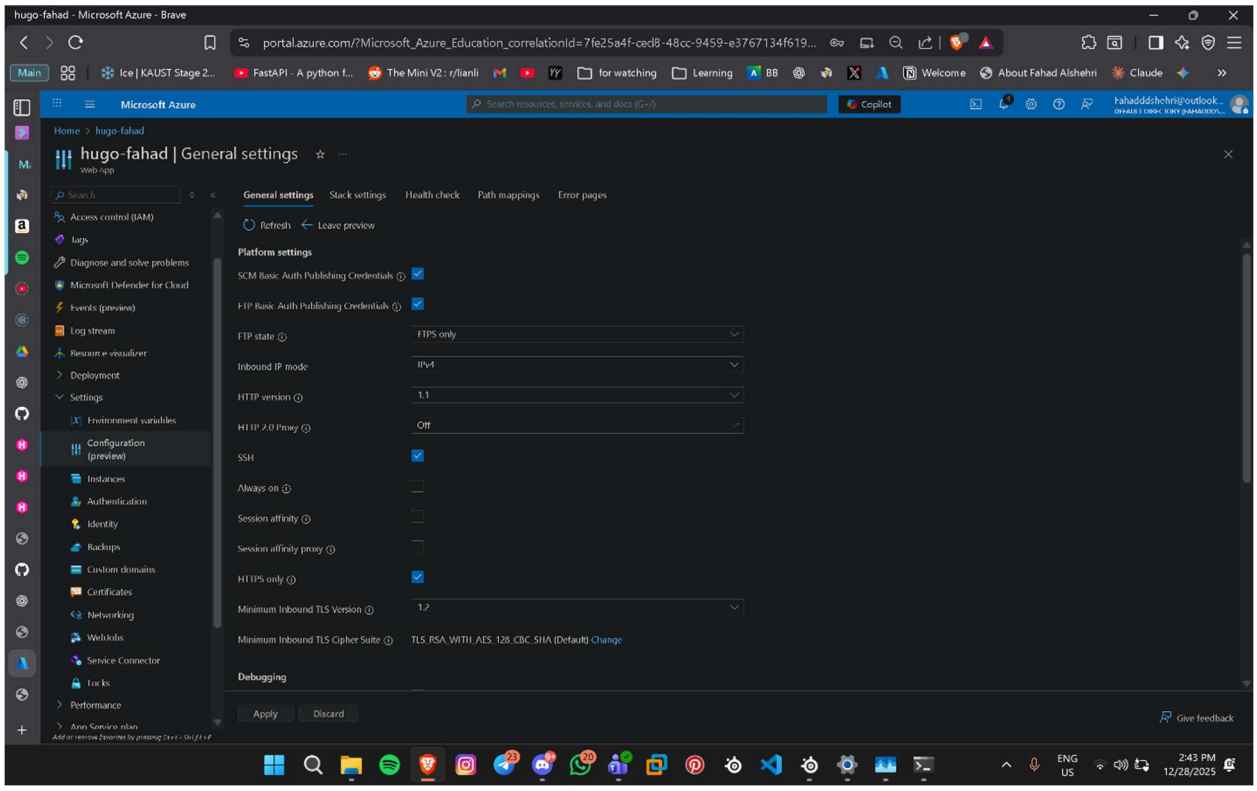
Site on VM using VM IP



PaaS Deployment: Creation

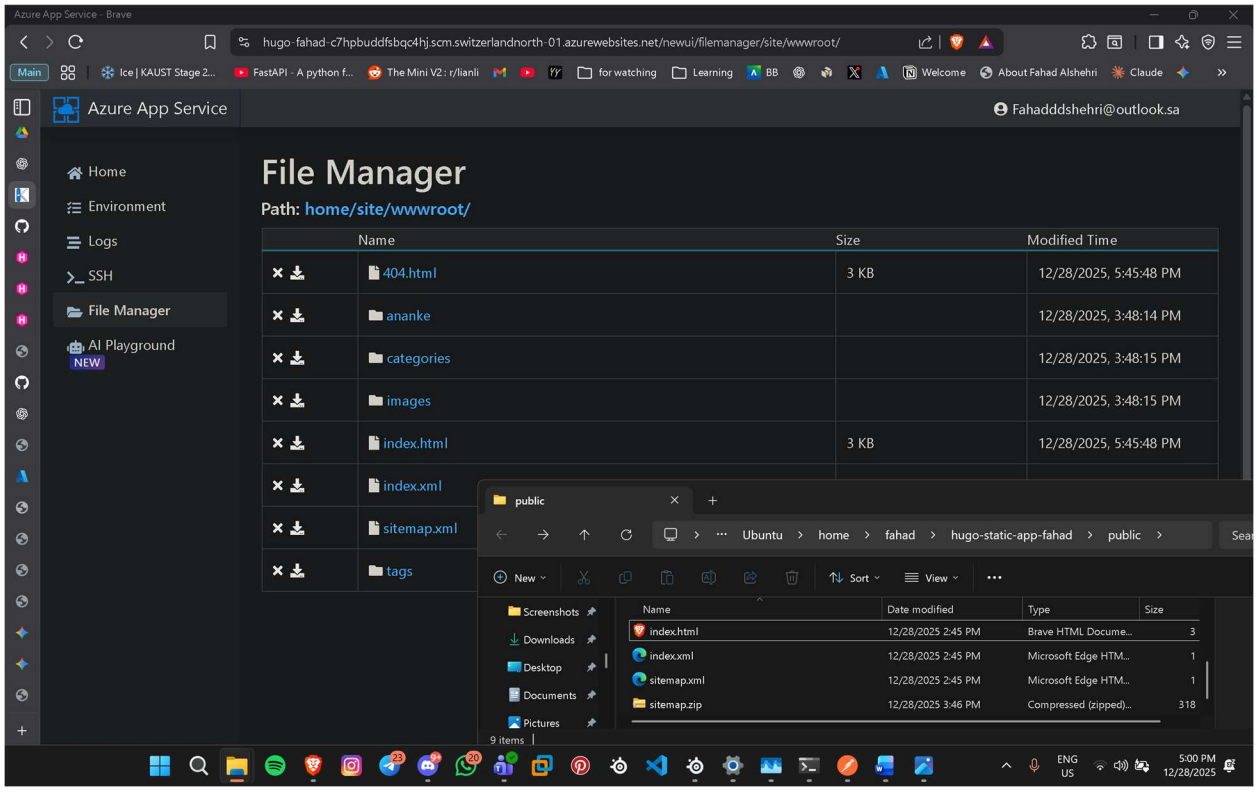


Enabling SCM & FTP

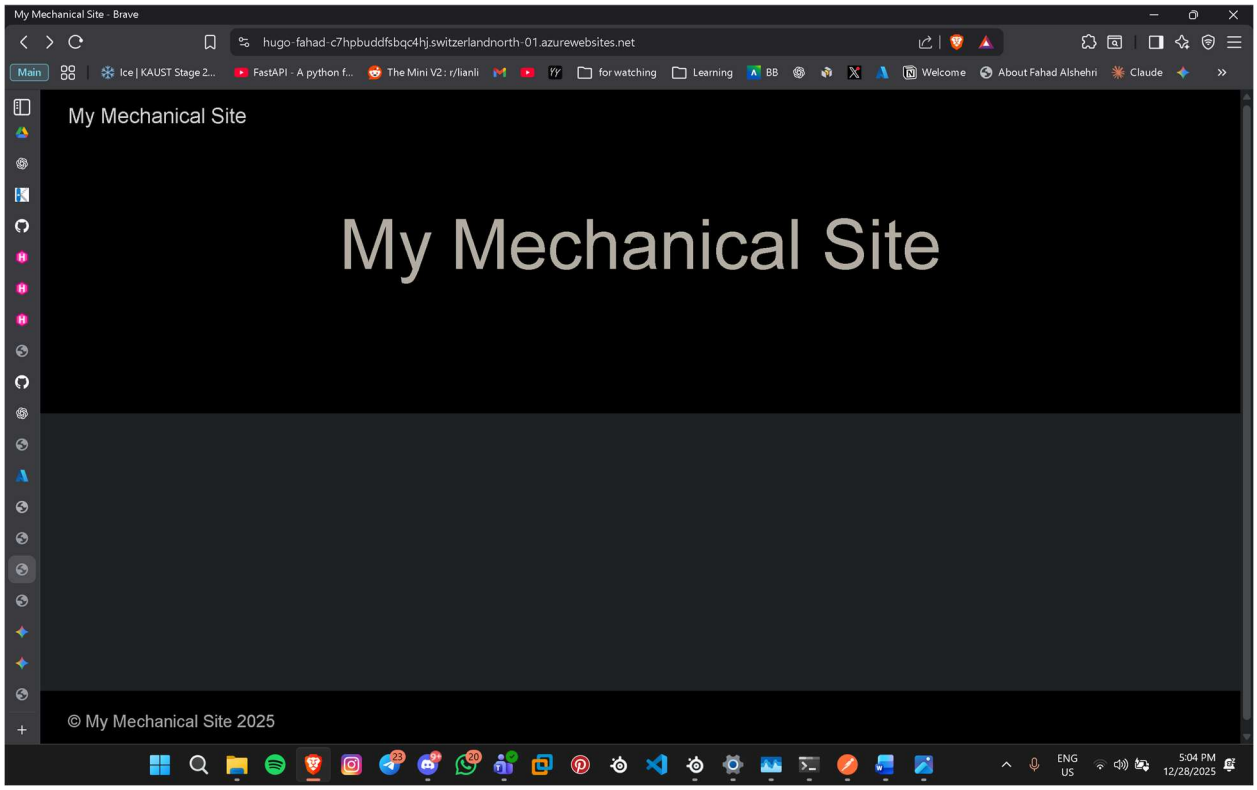


Using Kudo file manager to put zipped file “sitemap.zip” that compressed public directory files

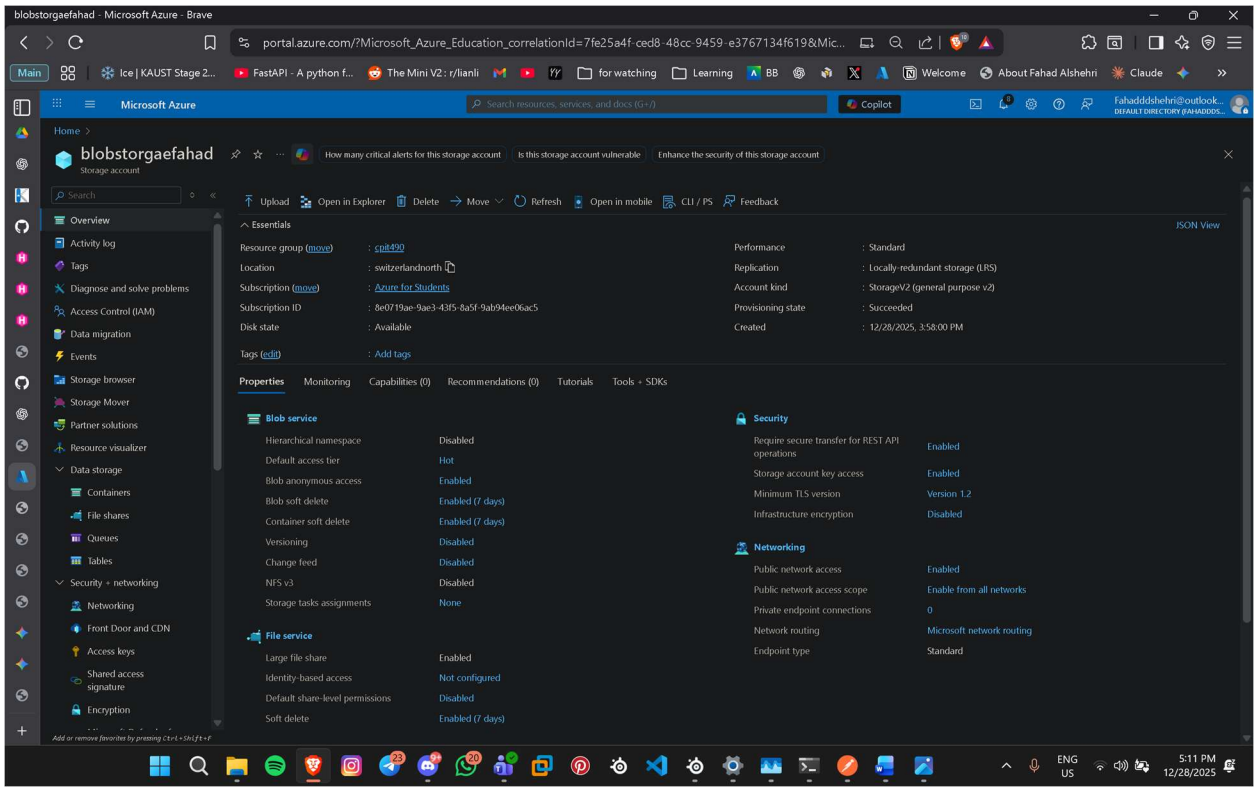
Note: tried multiple ways but zip file method is the best and fast



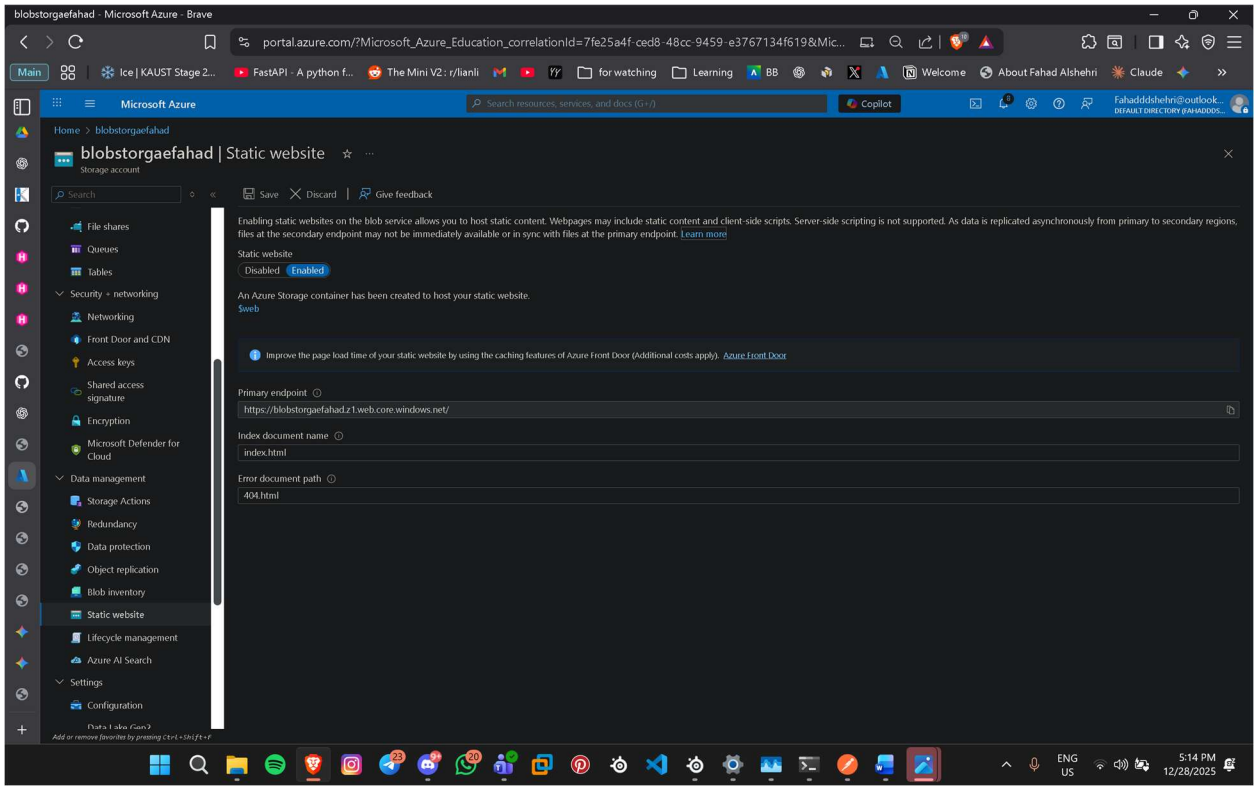
App deployed on PaaS as seen url “hugo-fahad...”



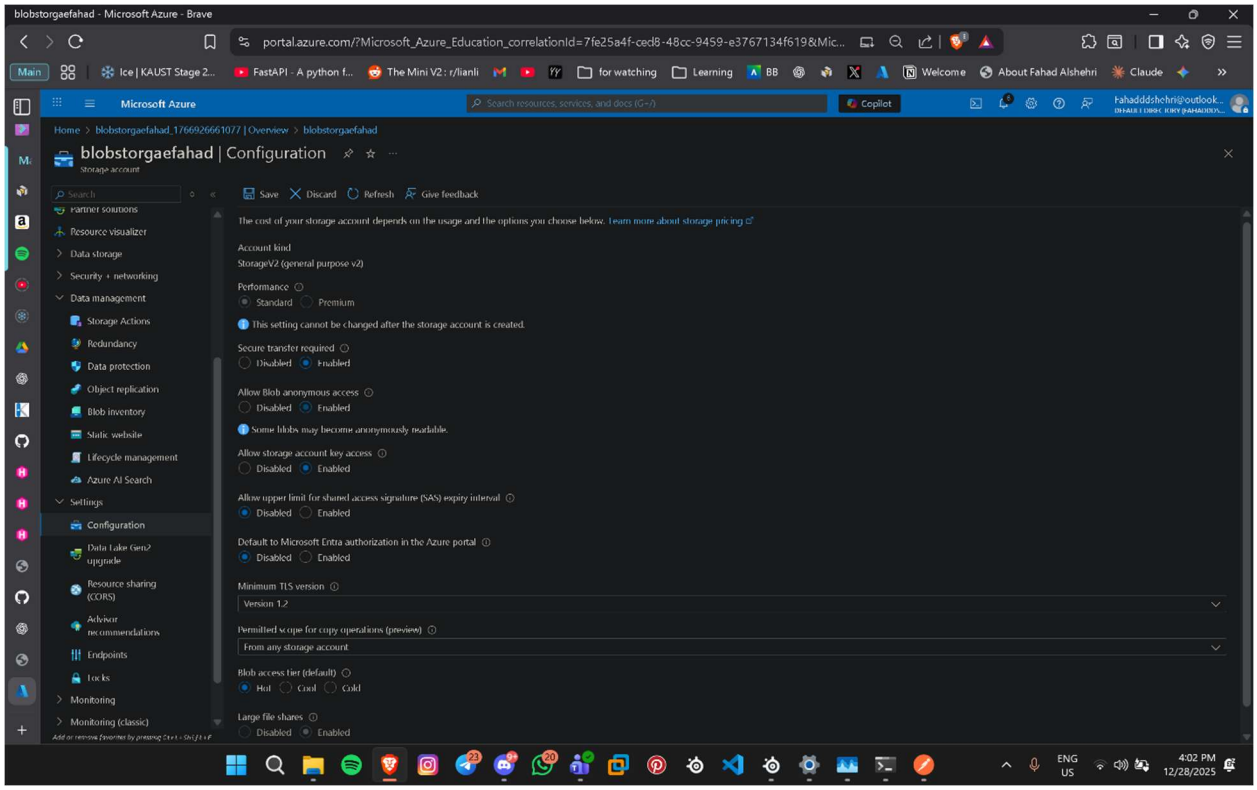
Object Storage Deployment:



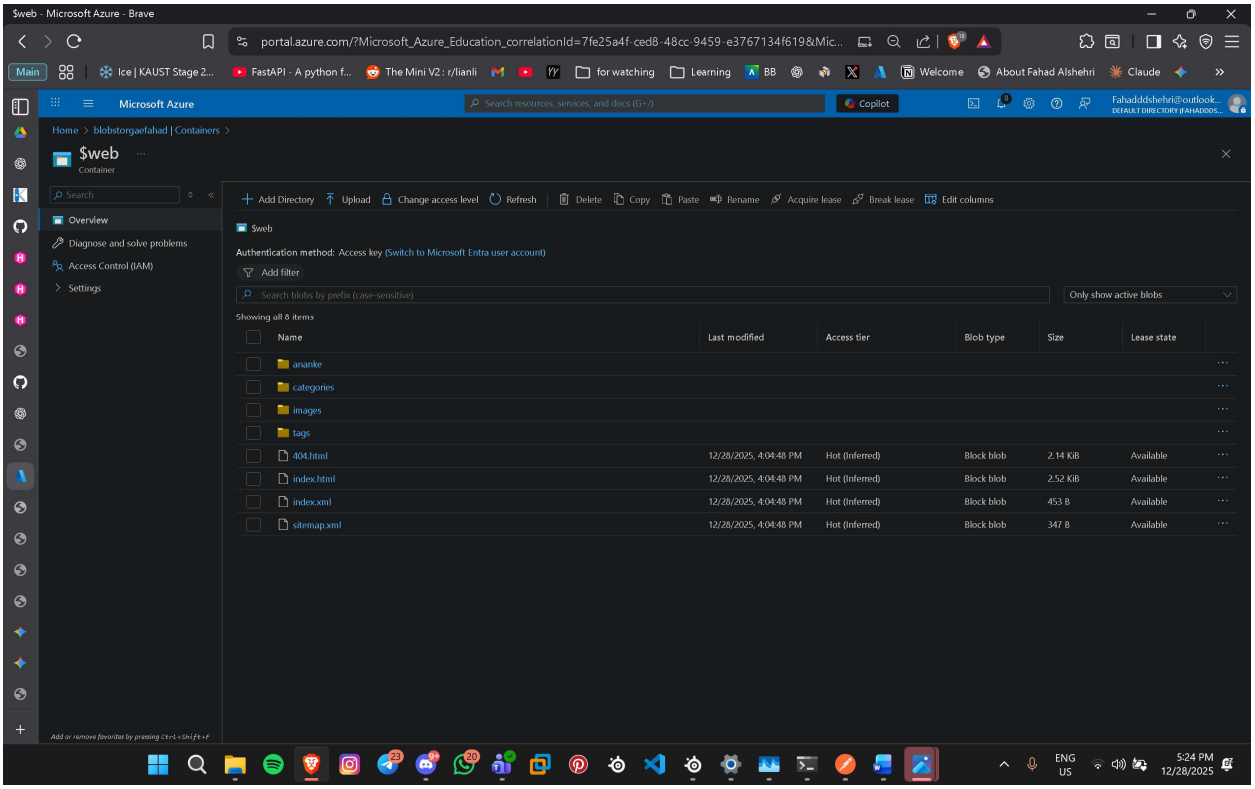
Endpoint url & index,error docs to use to show site



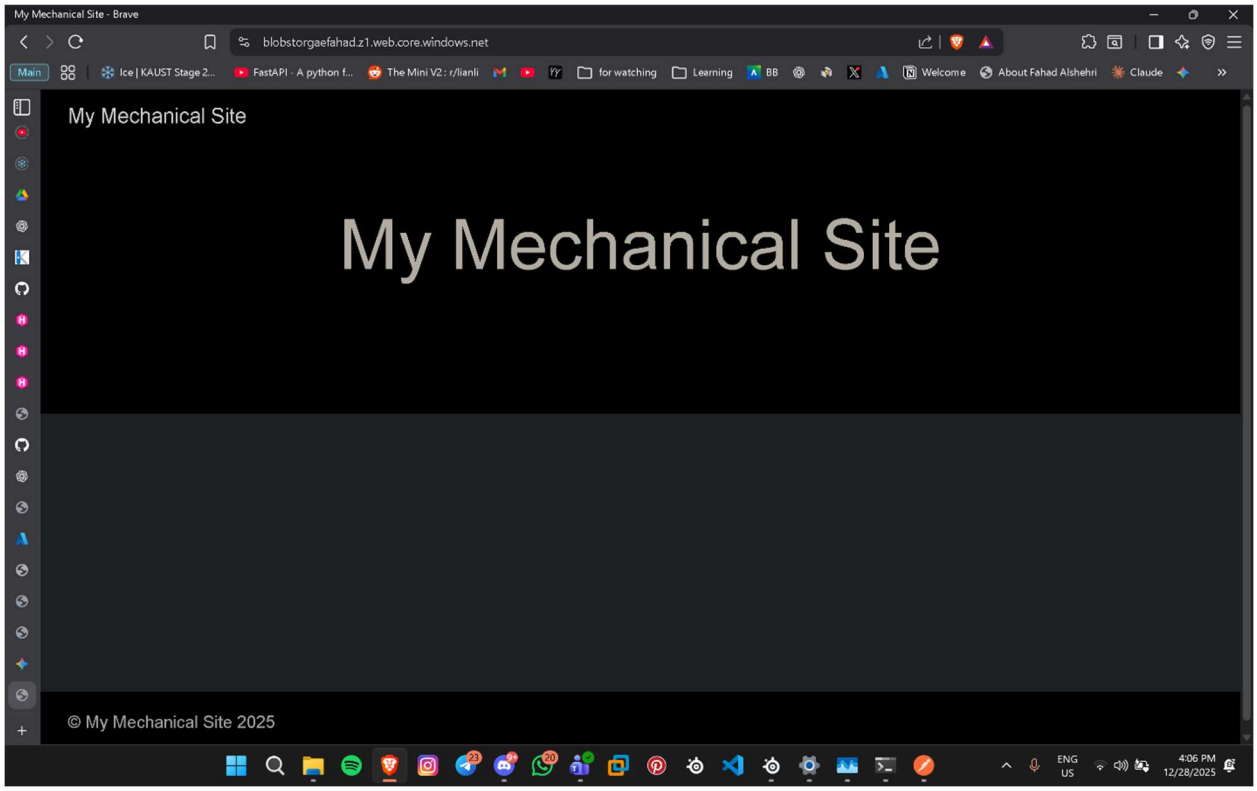
Enabling anonymous blob access to allow public access to site resources



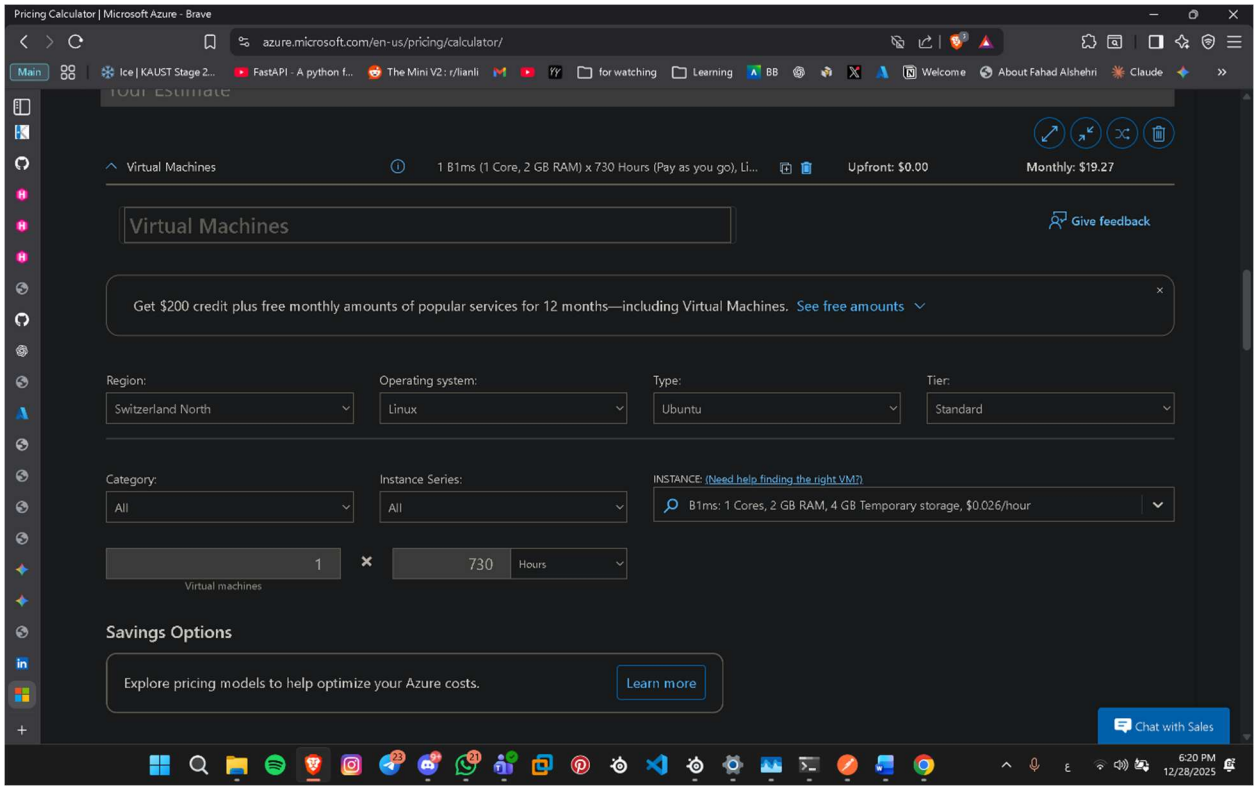
Upload of the Hugo site files HTML, XML etc.. in the $web container in Azure Blob Storage



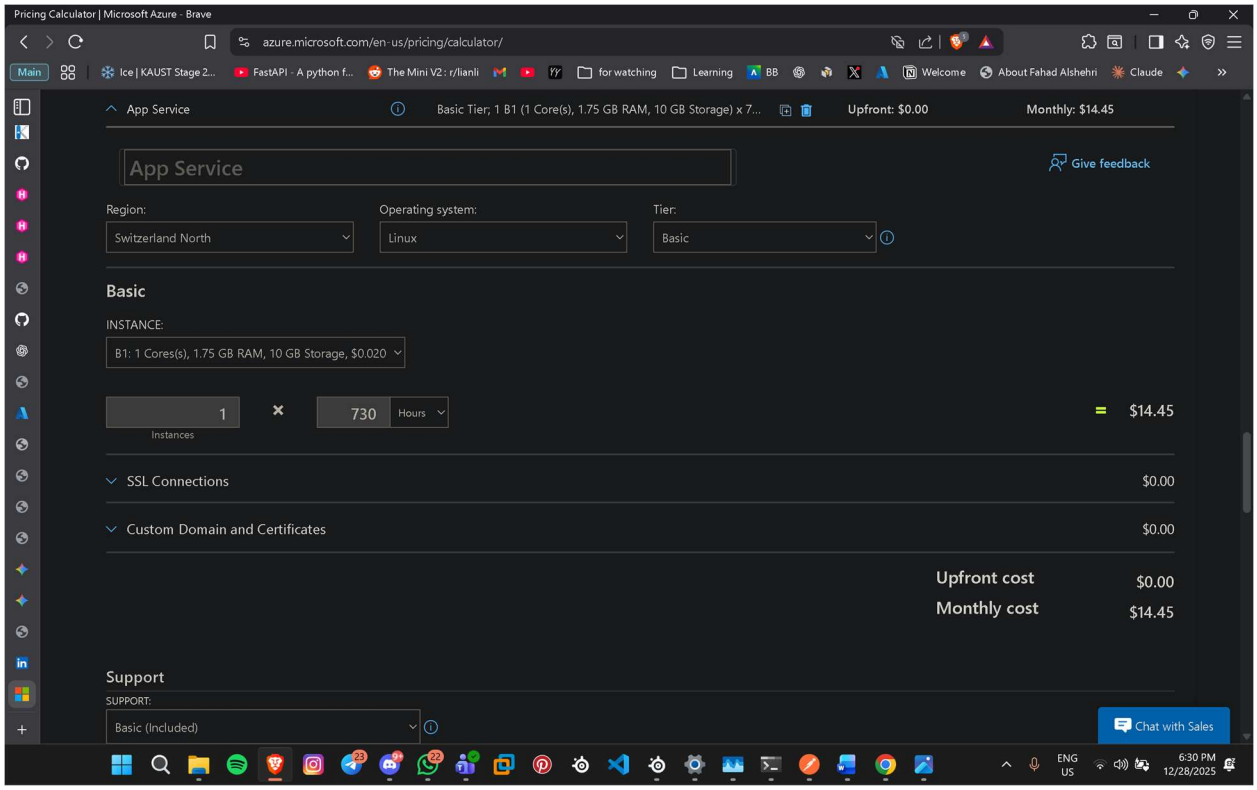
Accessing the live static website using the endpoint



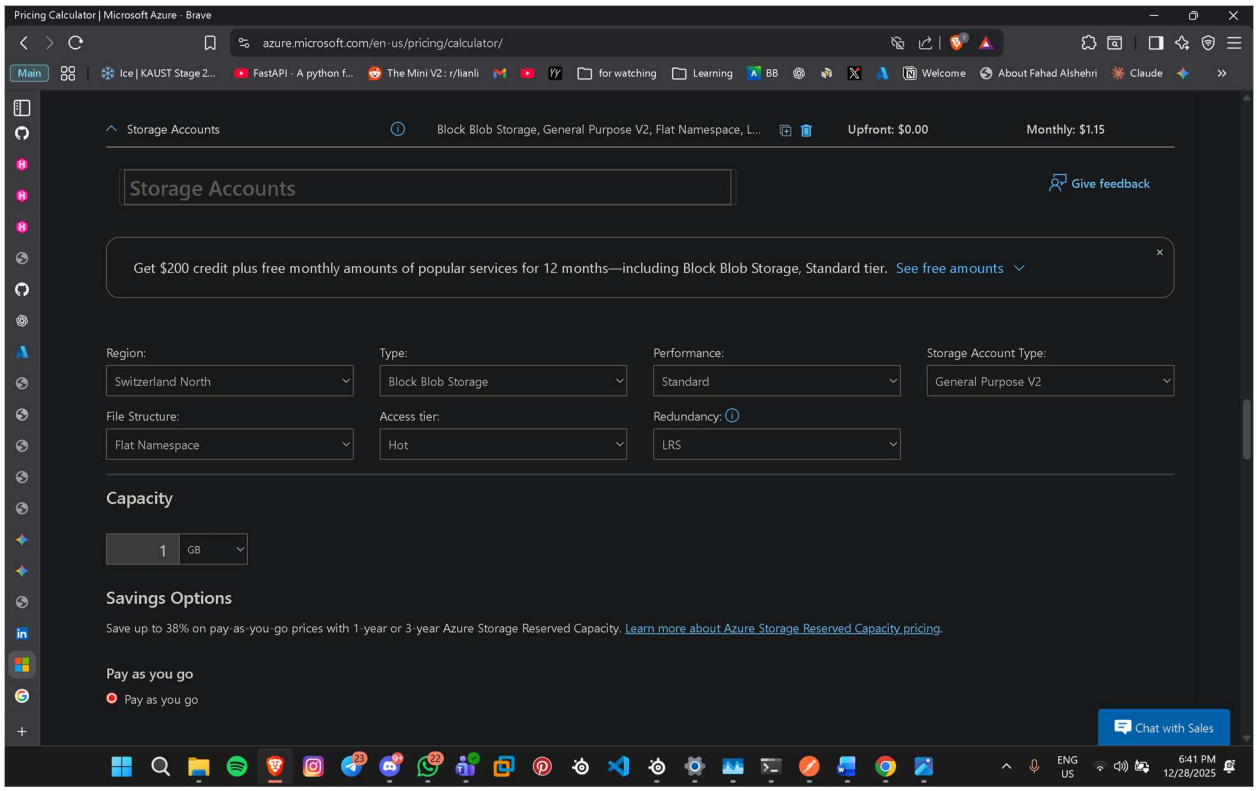
Prices Of Each VM, PaaS, Object Storage



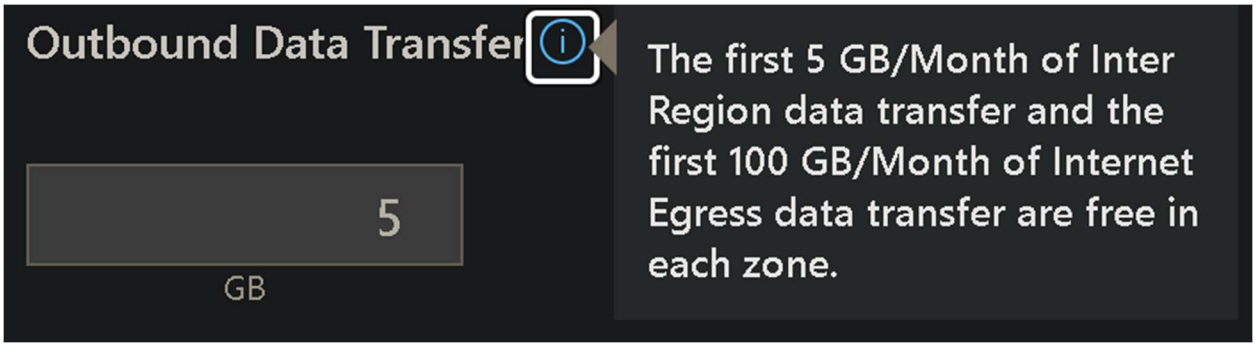
VM Cost: $19.27 monthly



PaaS Cost: $14.45 monthly



Blob price: $1.15 monthly



Since the first 100GB of data transfer is free on Azure, setting the limit to 2GB will not add any cost to the monthly estimate

Pricing comparison:

Estimated for 10,000 monthly visitors / 2GB Bandwidth.

Component	VM	PaaS	Object storage
Compute/Storage	$19.27 / month (Standard B1ms Instance)	$14.45 / month (Basic B1 Tier)	$1.15 / month (Standard, LRS)
Bandwidth (2GB)	(First 100GB/mo is free) $0	(First 100GB/mo is free) $0	(First 100GB/mo is free) $0
Total Monthly Cost	$19.27	$14.45	$1.15

Availability analysis:

Component	VM	PaaS	Object storage
SLAs	95% (Standard HDD)	99.95%	99.9%
Redundancy	Manually must configure backups	Automatic Azure manages hardware & runtime redundancy	Automatic Done by azure (LRS)
Failure Points	High, Risk of OS failure, single disk failure, and misconfiguration	Dependent on runtime stack and region uptime	Lower, no OS or server management risks

Recommendation:

I recommend option 3 (**Object storage**)

Cost: It is the cheapest option by far and It only costs about ($1.15) per month. The other options cost between ($14.45) and ($19.27) per month and Paying that much for a static website that doesn't need a server is just a waste of money**.**

Reliability: a static site doesn't need a complex server to run, This makes it safer because there is no OS to update and no risk of a server crash It also has a better uptime guarantee (99.9%) compared to the VM (only 95%).

Scalability: Azure handles the traffic automatically If the site suddenly gets a lot of visitors, Object storage will handle it without any issues and no need to manually add more servers or change settings.

Future Roadmap: DevOps Integration

As a next phase, the goal is to transition these manual deployments into a fully automated **CI/CD Pipeline** using:

- **Infrastructure as Code (IaC):** Leveraging **Terraform** to provision the Azure resources.
- **Continuous Deployment:** Using **GitHub Actions** to automate site builds and deployments upon every code push, ensuring a modern DevOps workflow.