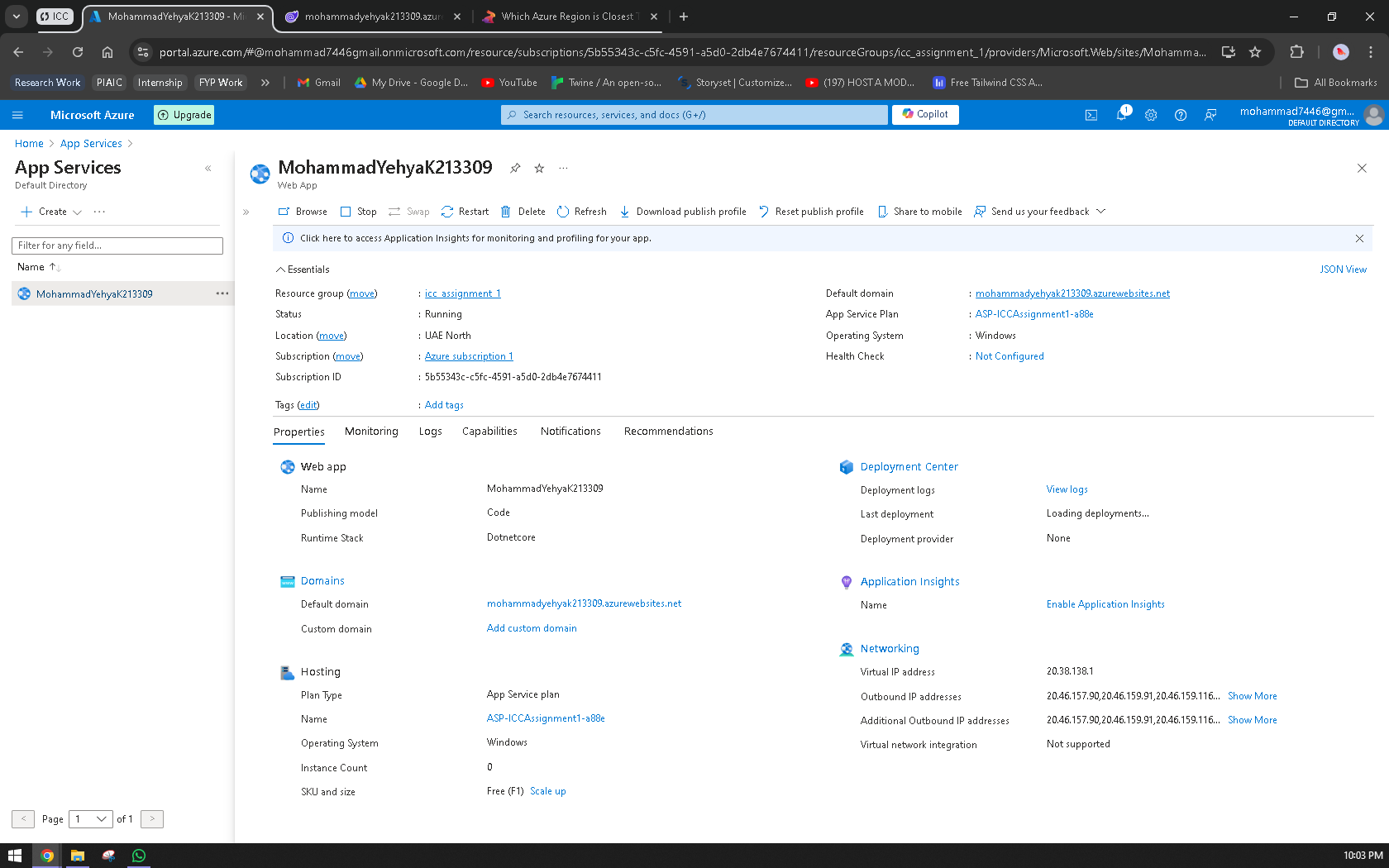
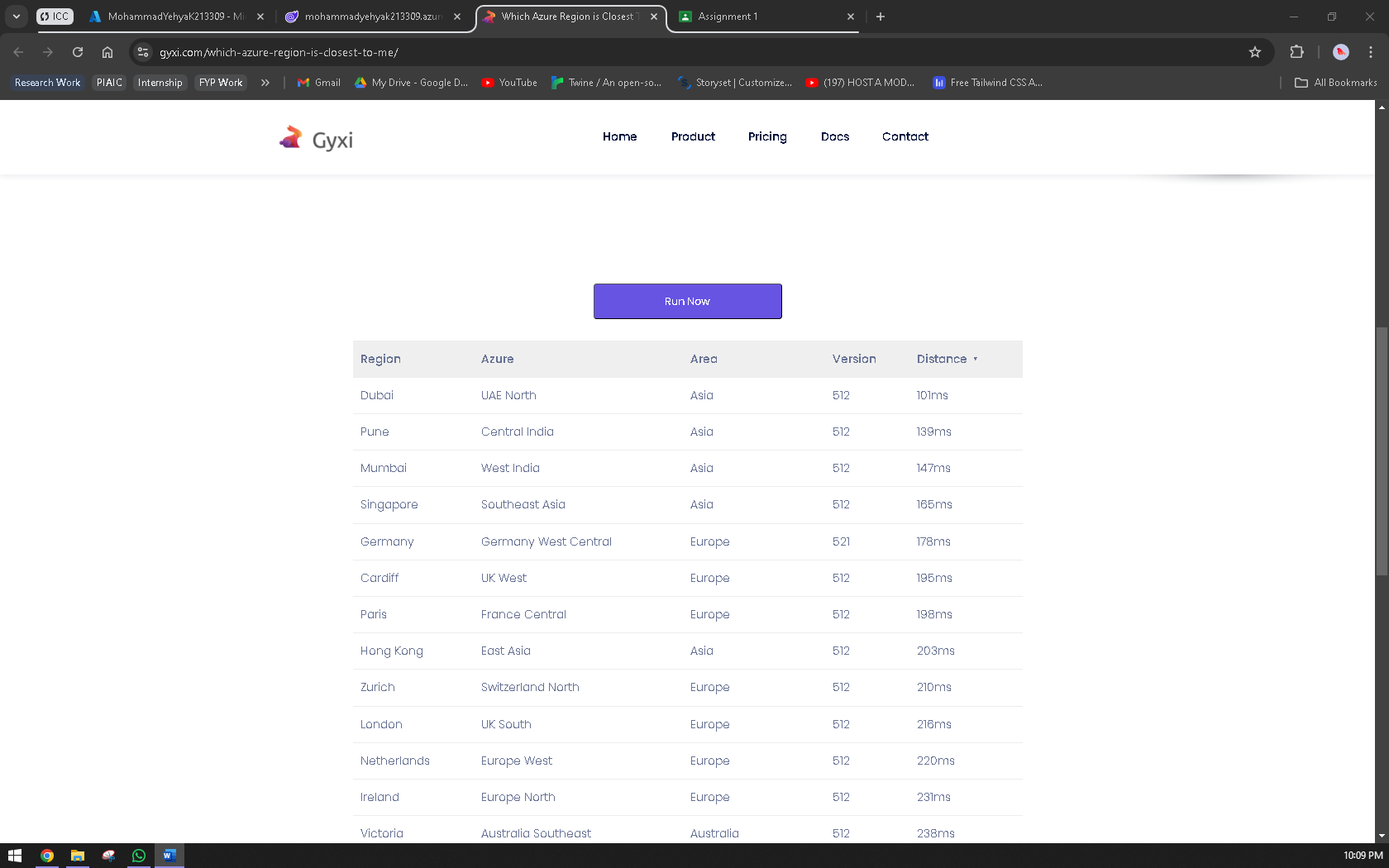
# Step 1



I chose the **C# (.NET Latest)** runtime stack because it is widely used for building robust, scalable web applications. The latest version of .NET offers performance improvements, better memory management, and advanced security features, making it suitable for modern cloud-based solutions. Using .NET on Azure also allows seamless integration with other Azure services like SQL databases, Azure Active Directory, and API management, making it a highly compatible choice for developing enterprise-grade applications.

For the **region**, I selected **UAE North**. This ensures reduced latency and quicker response times for users in that region (this can be seen in the image below which was taken from a website <https://gyxi.com/which-azure-region-is-closest-to-me/>). By deploying in a region close to the majority of users, the app benefits from faster data transfer and better overall performance, thanks to Azure's global network of data centers.

I opted for the **Free F1** pricing tier as this project is in its early stages and doesn't require extensive resources or high traffic support. The Free tier offers enough features for development, testing, and proof of concept, without incurring costs. While it lacks autoscaling and custom domain support, it is sufficient for small-scale applications with limited usage, making it ideal for initial development and experimentation.



# Step 2

Source code is zipped and uploaded onto GCR.

This web application, built using **C# .NET (Latest)**, provides a fundamental introduction to cloud computing. The application contains three pages aimed at educating users about the benefits, models, and evolution of cloud technology.

The first page, titled **"Introduction and Benefits of Cloud Computing"**, provides an overview of what cloud computing is and explains its major benefits, such as cost savings, scalability, and enhanced accessibility. It details how organizations can use cloud services to reduce infrastructure management burdens while gaining the ability to scale resources on demand.

The second page discusses the **"Cloud Deployment Models"**. It explains the four main deployment models: **Public Cloud**, **Private Cloud**, **Hybrid Cloud**, and **Community Cloud**. The **Public Cloud** model allows access to cloud services over the internet, managed by third-party providers. **Private Cloud** is designed for exclusive use by a single organization, offering more control and security. **Hybrid Cloud** combines public and private clouds, allowing data and applications to be shared across both environments. Finally, the **Community Cloud** is shared by multiple organizations with common interests or requirements.

The third page explores the **"Evolution of Cloud Computing"**, detailing key milestones from the inception of cloud technology to today’s advanced cloud infrastructure.

This app provides an easy-to-navigate, educational experience for users new to cloud computing.

# Step 3

Live URL shared on GCR.

Screenshot same as the first image in this doc.

During the deployment of my web application to Azure App Service, I encountered a few challenges. One of the main issues was related to the **runtime stack mismatch**. Initially, I selected the wrong .NET Core version on Azure, which led to compatibility issues between the application and the hosting environment. I resolved this by ensuring that the runtime stack on Azure matched the one I used in my development environment (C# .NET Core Latest).

Lastly, I faced issues with **publishing from Visual Studio**, which was resolved by manually resetting my Azure publishing credentials and re-establishing a secure connection to Azure via Visual Studio.