Future Ready Talent - Project Documentation

Name – Haressh Deshmukh

Project Title – Developed & Deployed Tetris Game App Using Azure Services & Created Docker Image.

**Project Statement**

The project aims to develop and deploy a Tetris game app using Azure services and Docker technology.

It addresses the common issue of managing dependencies by packaging all necessary components within a Docker image.

This project falls under the Lifestyle industry, providing entertainment and leisure for users.

The primary Azure technologies employed are Virtual Machines, Azure Container Registry, and Azure Container Instances.

The Tetris game app is built using HTML and JavaScript, ensuring cross-platform compatibility.

The local development environment is set up for testing and fine-tuning the app.

A Linux Virtual Machine is created, and Docker engine is installed and configured on it.

The project utilizes Docker to pull the Nginx image and creates a Dockerfile for customization.

Azure Container Registry is established to store the Docker image securely.

The final step involves deploying a container instance using the Azure Container Registry image.

**Description -**

The project begins by developing a Tetris game app using HTML and JavaScript, providing an engaging and interactive experience for users. The game is designed to be responsive and user-friendly across different devices. To ensure seamless deployment and management, Docker technology is leveraged.

Local development starts by running the Tetris game app on the developer's machine for testing and development. To make the deployment process more robust, a Linux Virtual Machine is created within Azure, where Docker is installed and configured. This VM acts as a secure and controlled environment for running the app.

A Docker image for Nginx is pulled, and a Dockerfile is created to customize the environment. This allows for the Tetris game to be served through a web server. Azure Container Registry is set up to store the Docker image securely, providing version control and access management.

The Azure CLI is installed for easy interaction with Azure services, and a Docker image is built using the Dockerfile. The image is then pushed to Azure Container Registry, ensuring that it is available for deployment in Azure Container Instances.

**Azure Services Used –**

Virtual Machine

Azure CLI

Container Registry

Container Instance

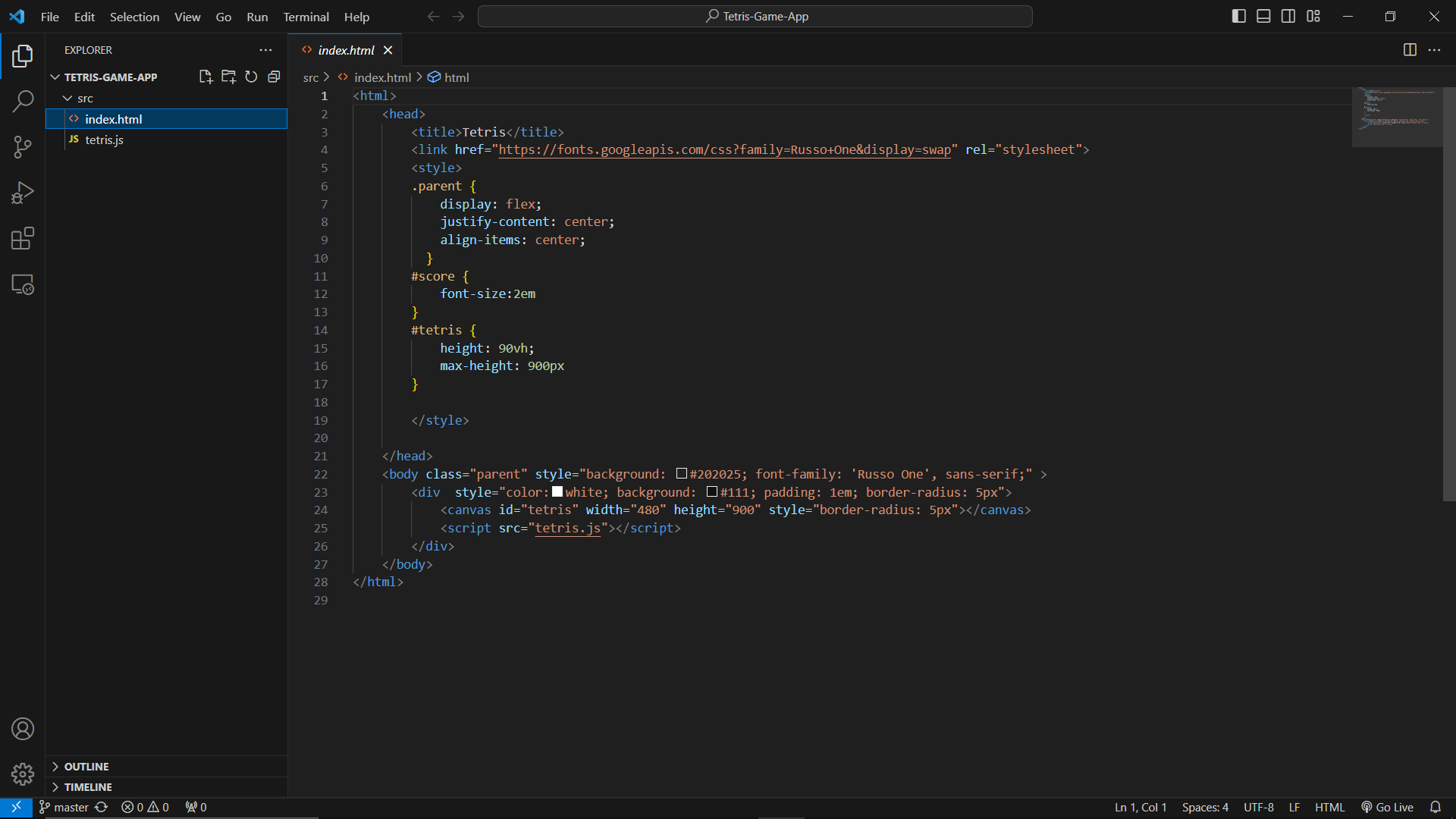
**Environment Used-**

VM – linux

System – Windows 11

**Project Screenshots**

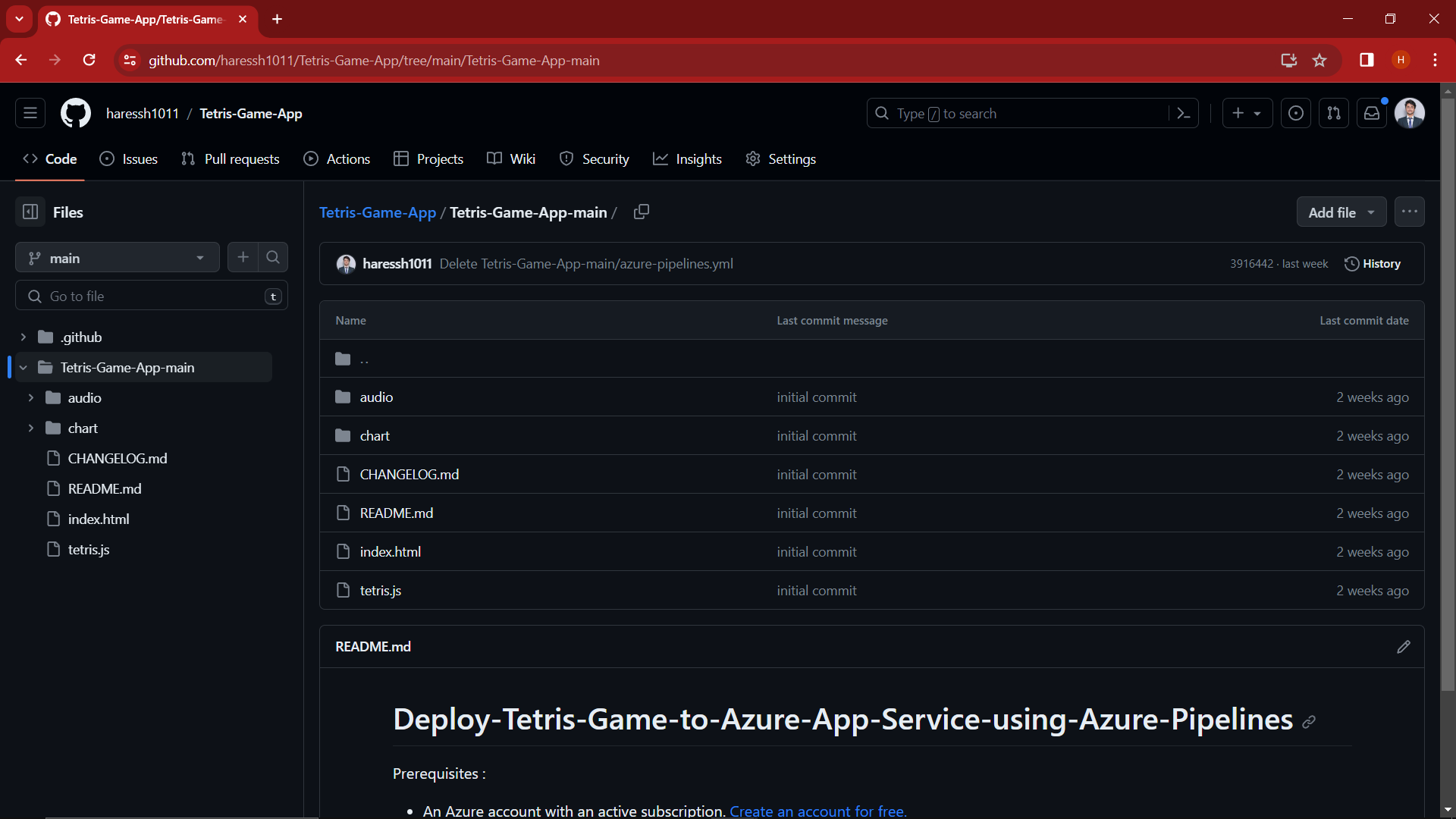
* **This is the HTML file**



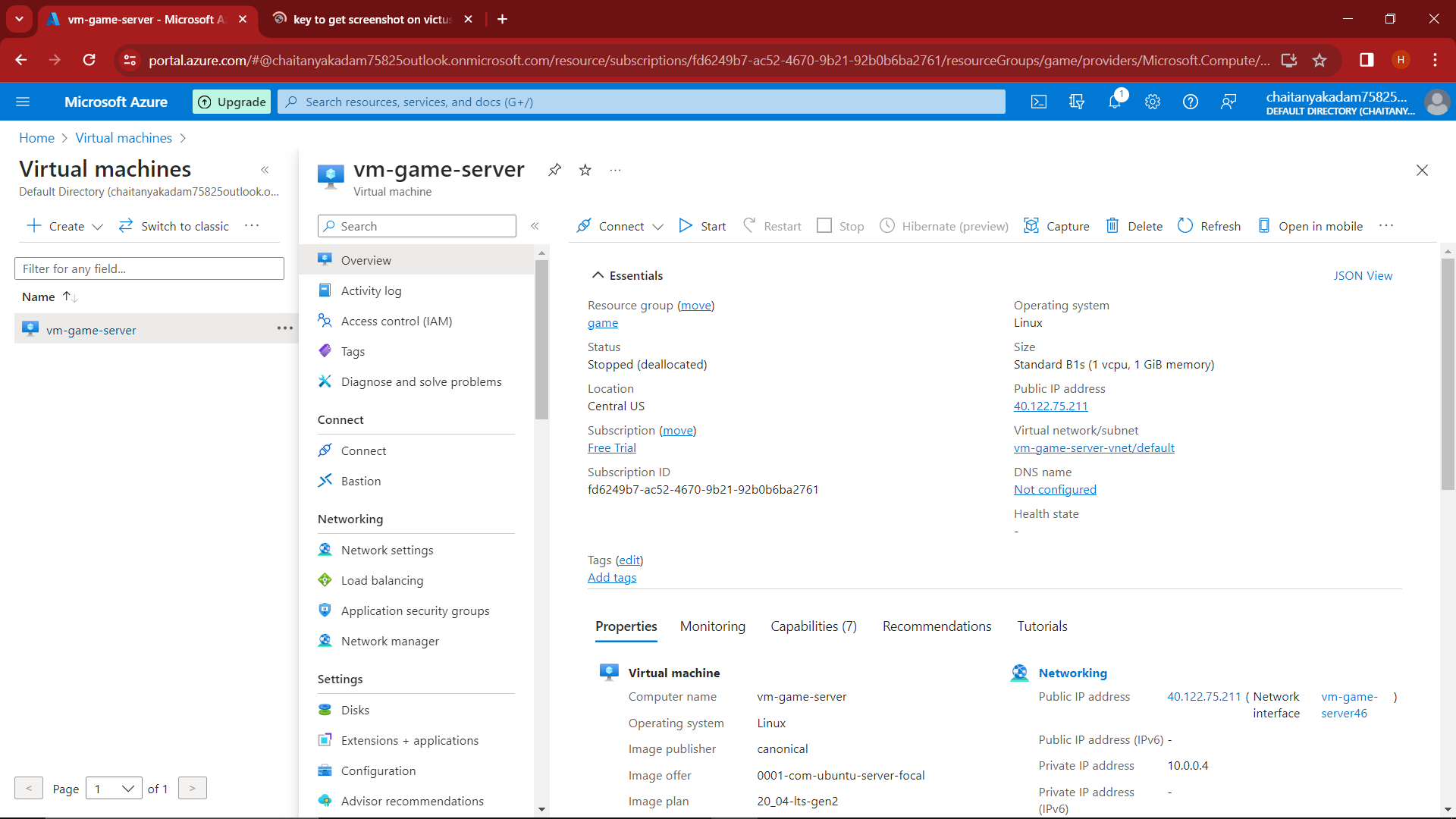
* **This is the JavaScript file**



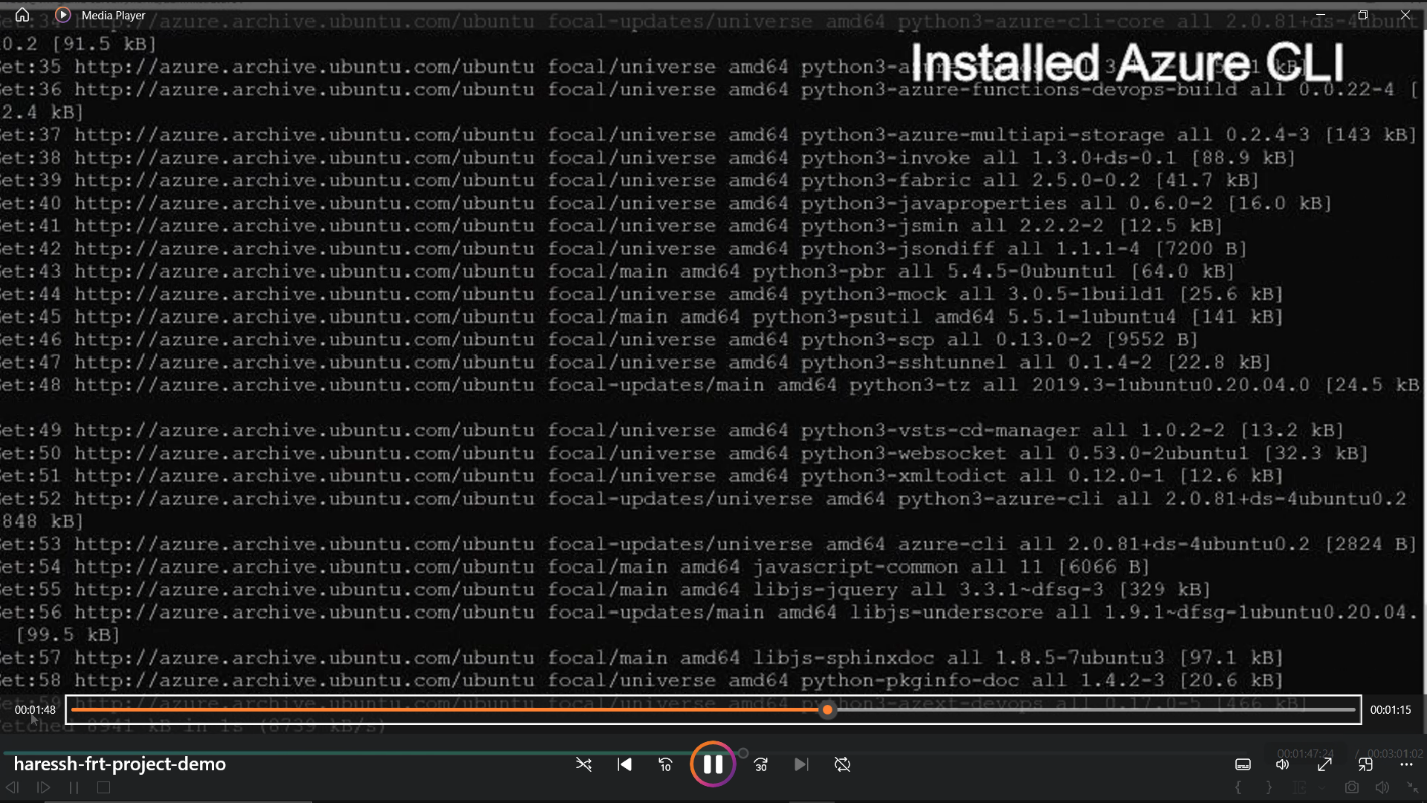
* **Github account and the project repository**

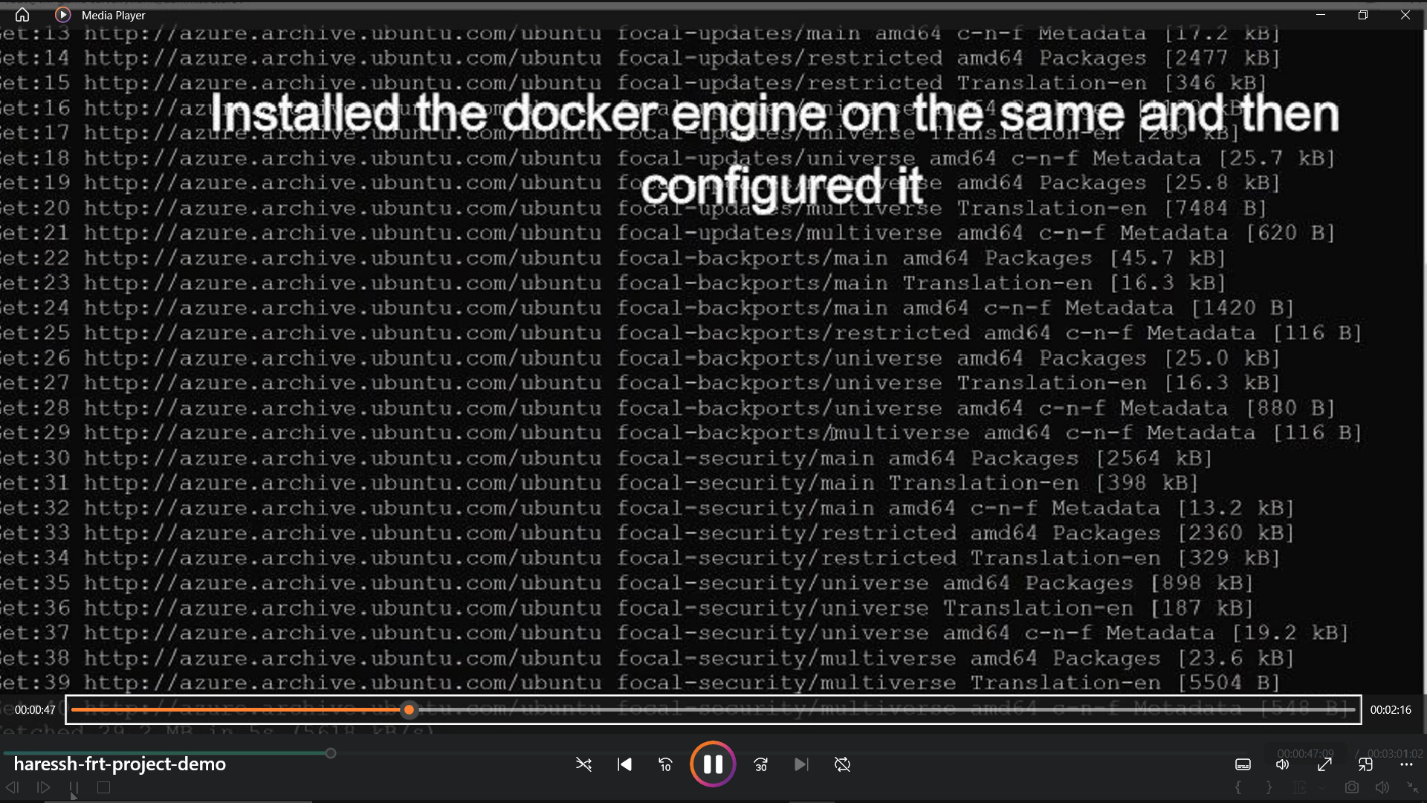


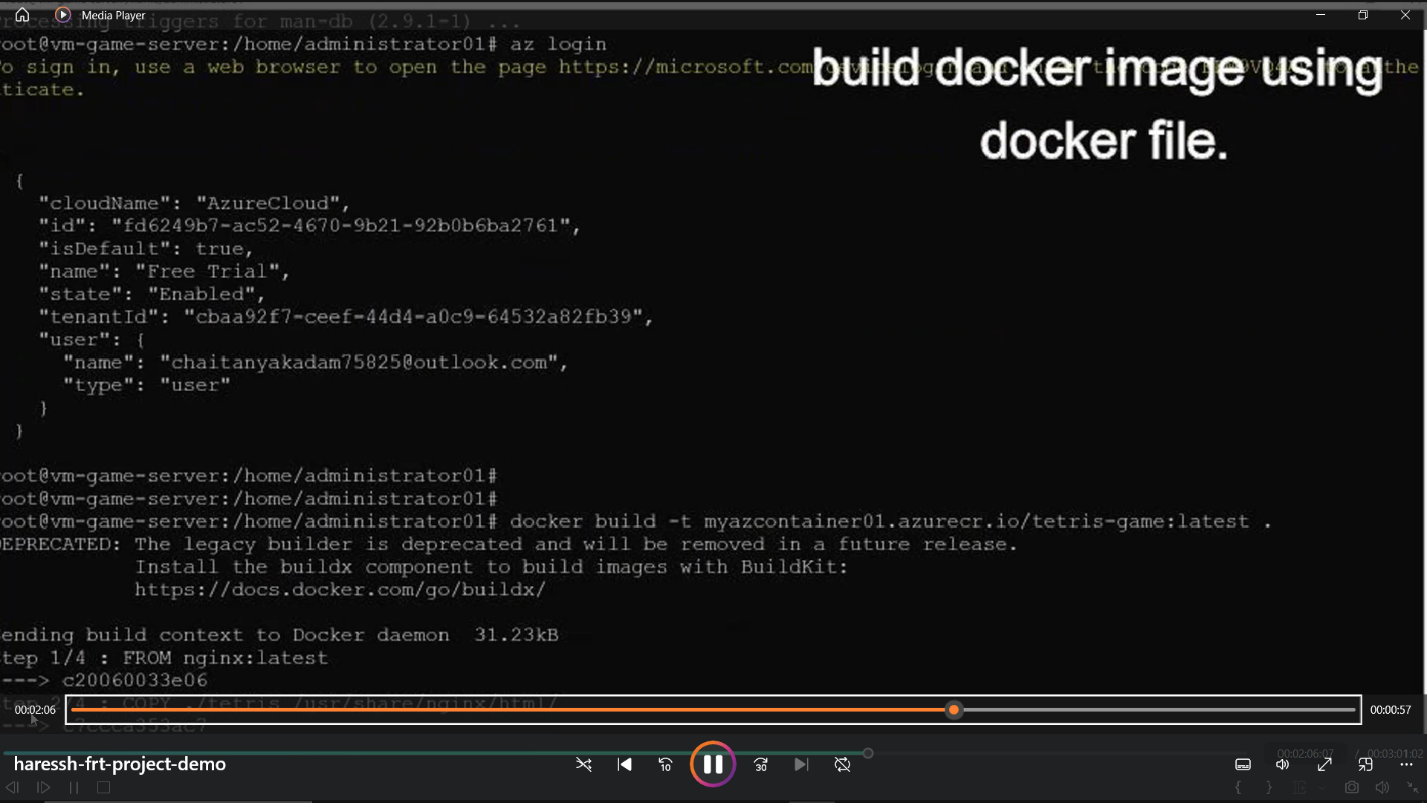
* **Azure virtual machine Service**

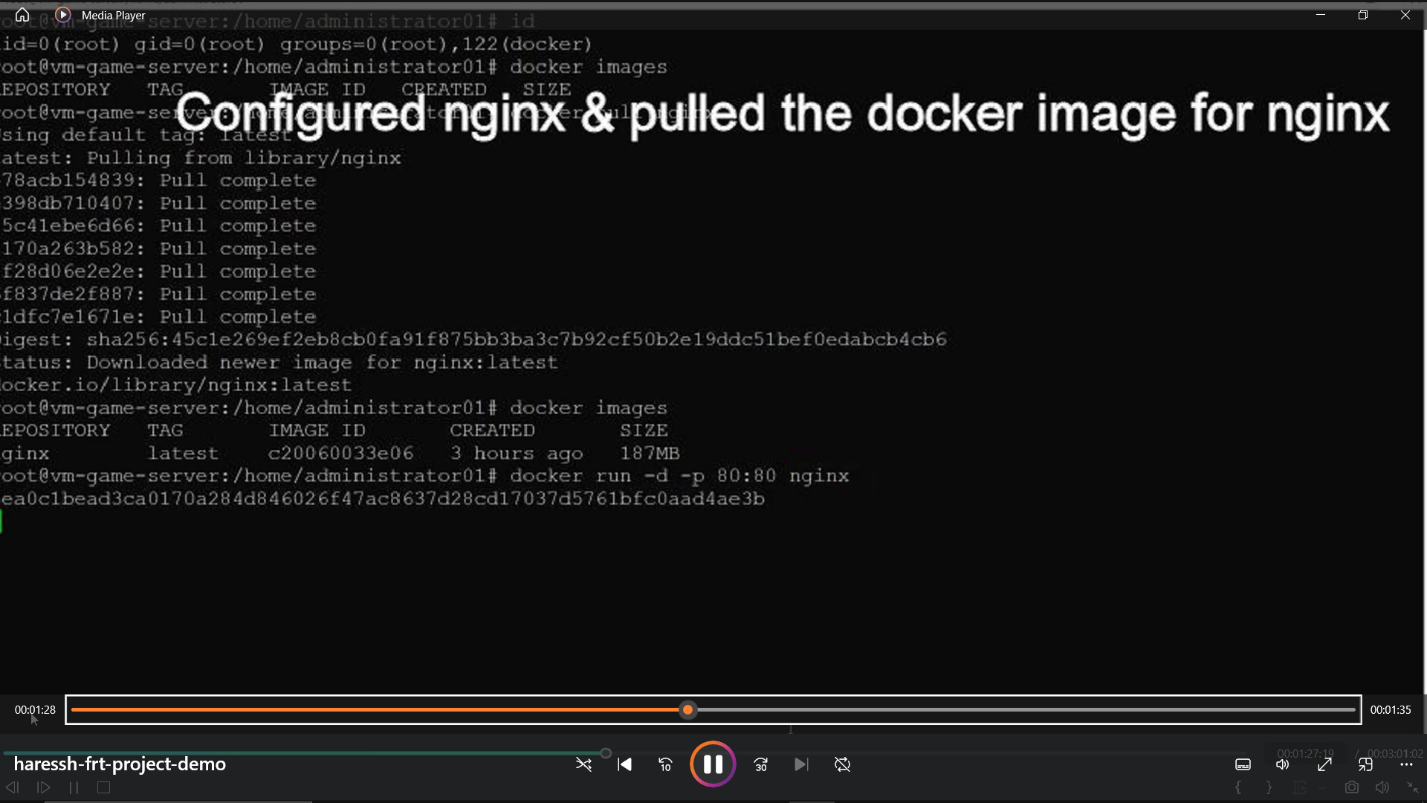
****

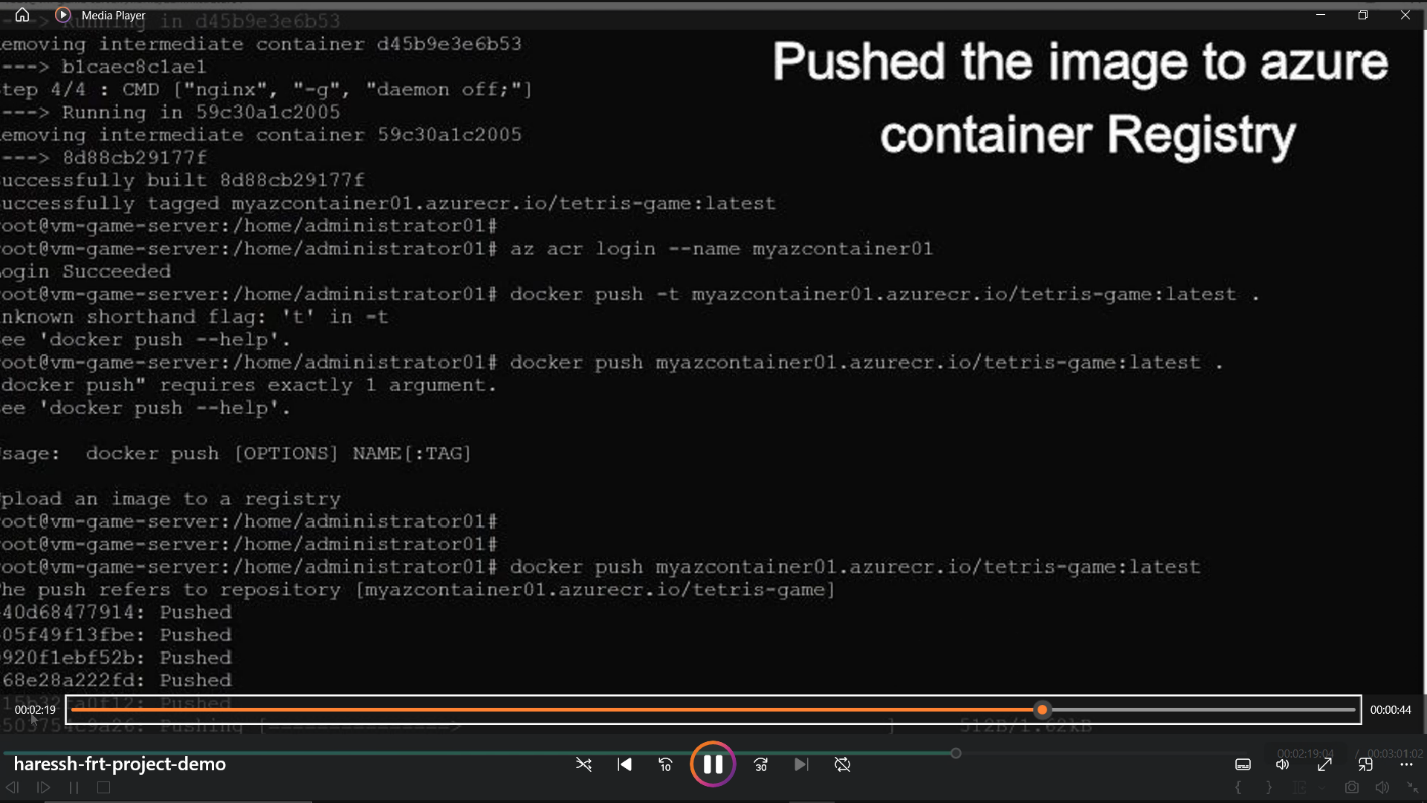
* **Installing azure CLI, Installing Docker & Configuring it, pulling the image for Nginx, creating a docker file, building docker image using docker file & push image to azure container registry.**

****

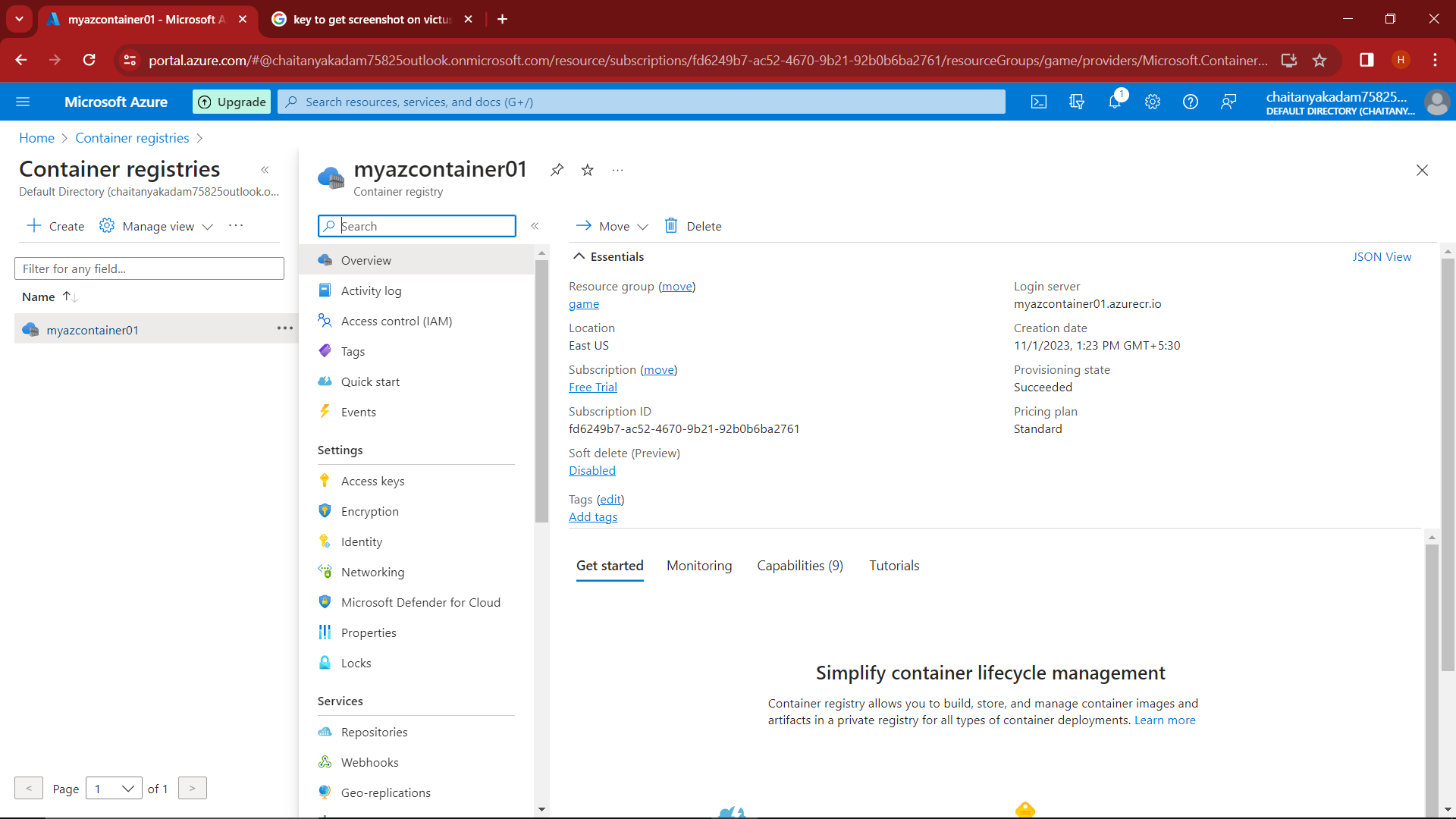
****



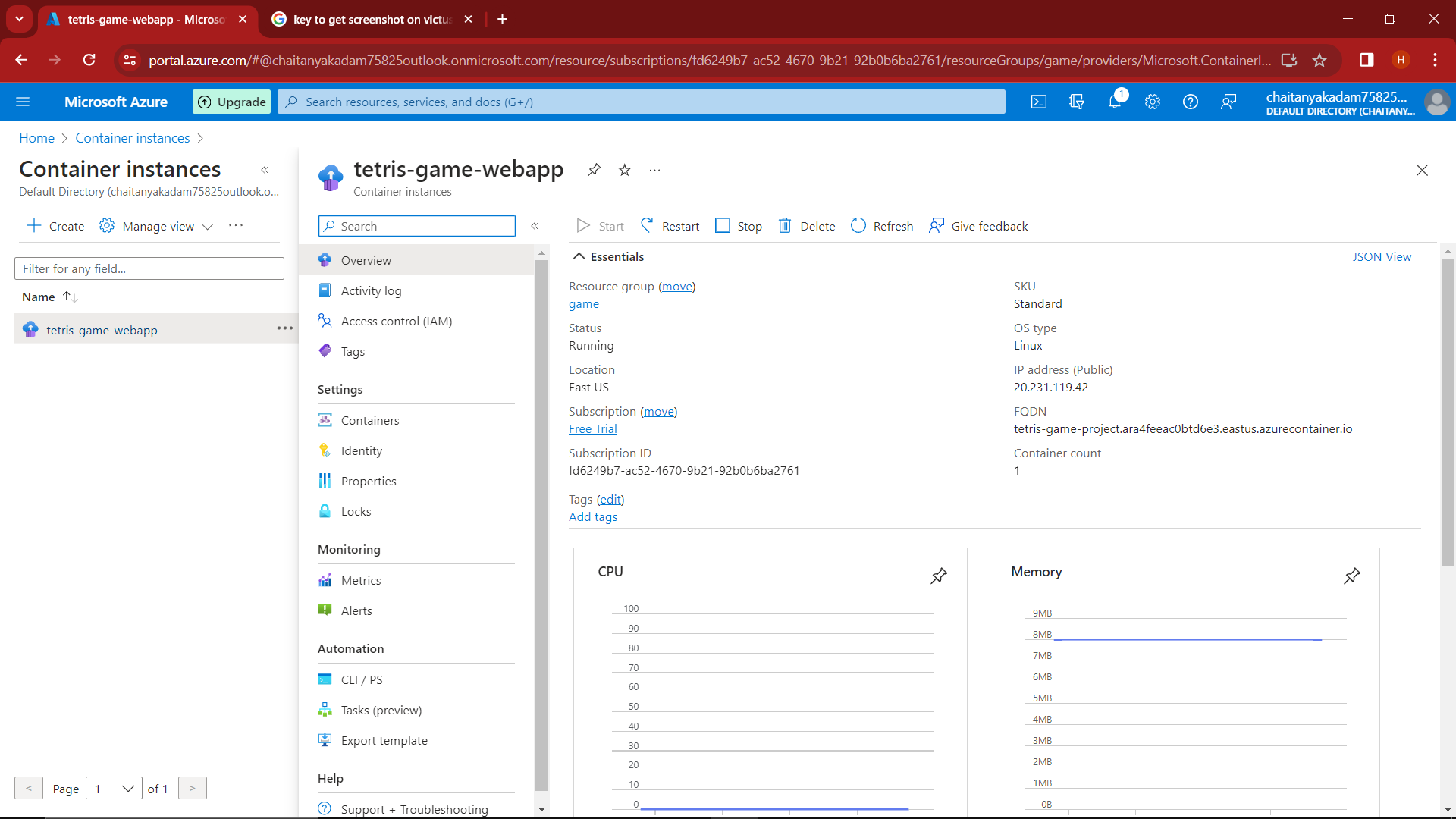




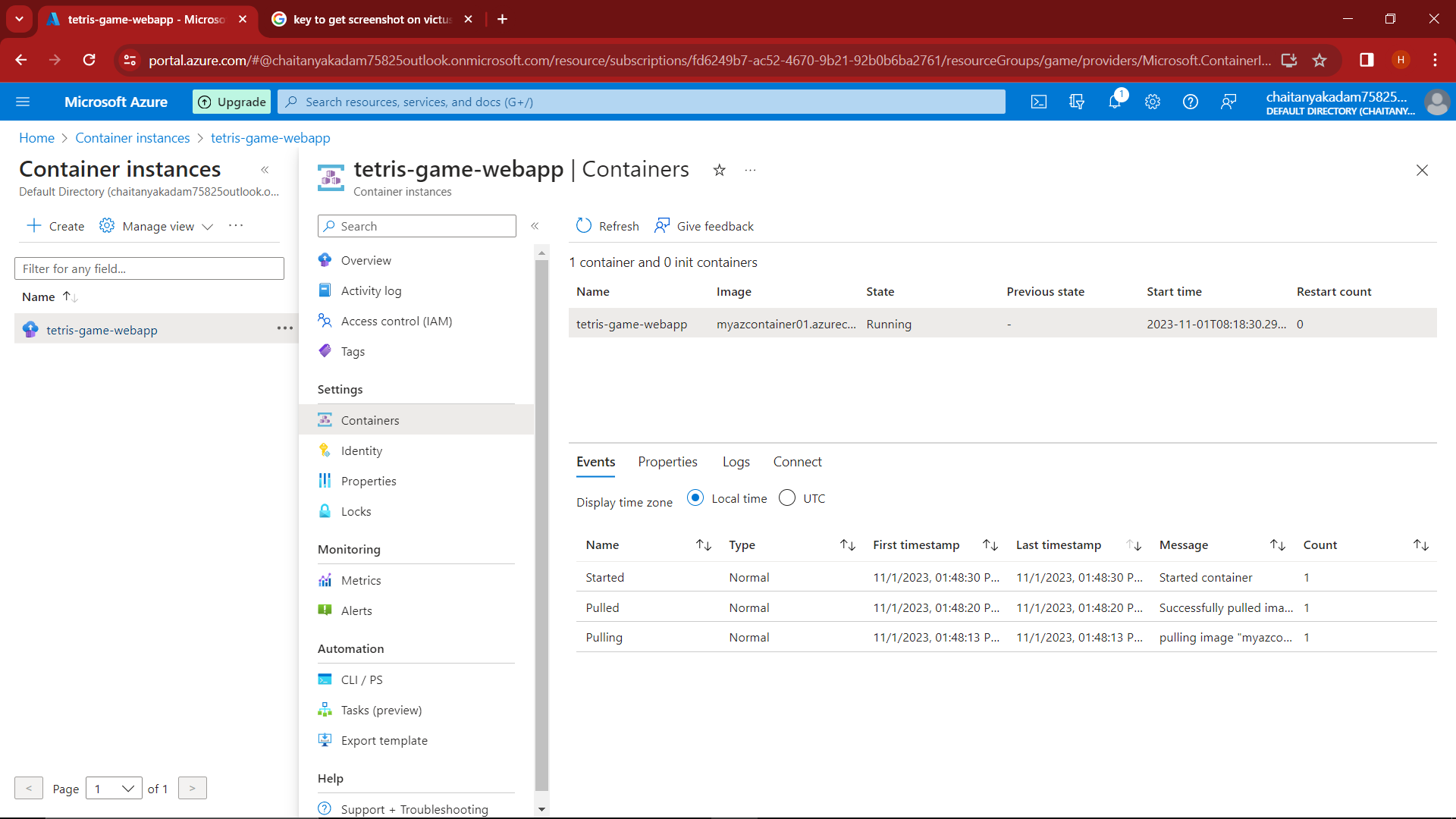
* **Azure Container Registry service**

****

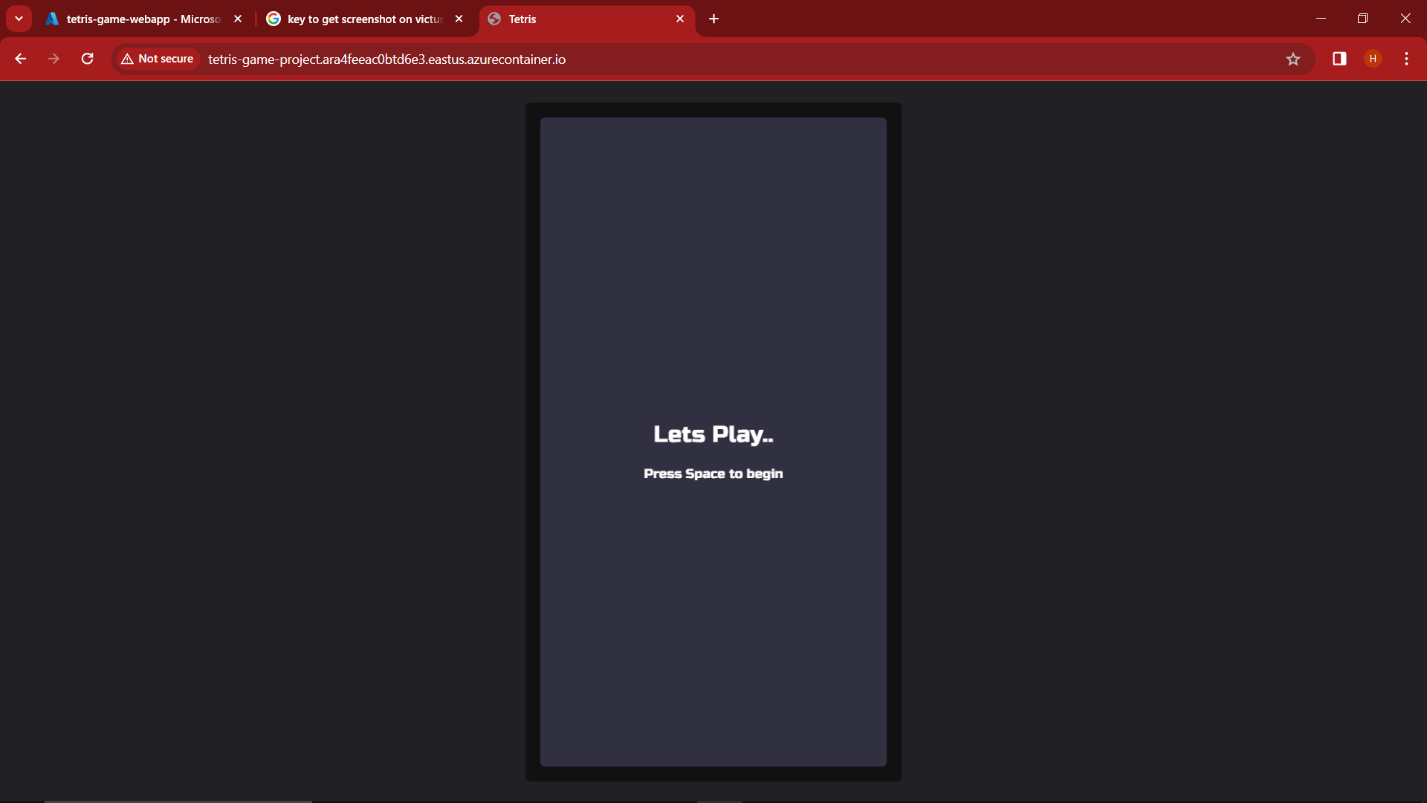
* **Azure Container Instance service**

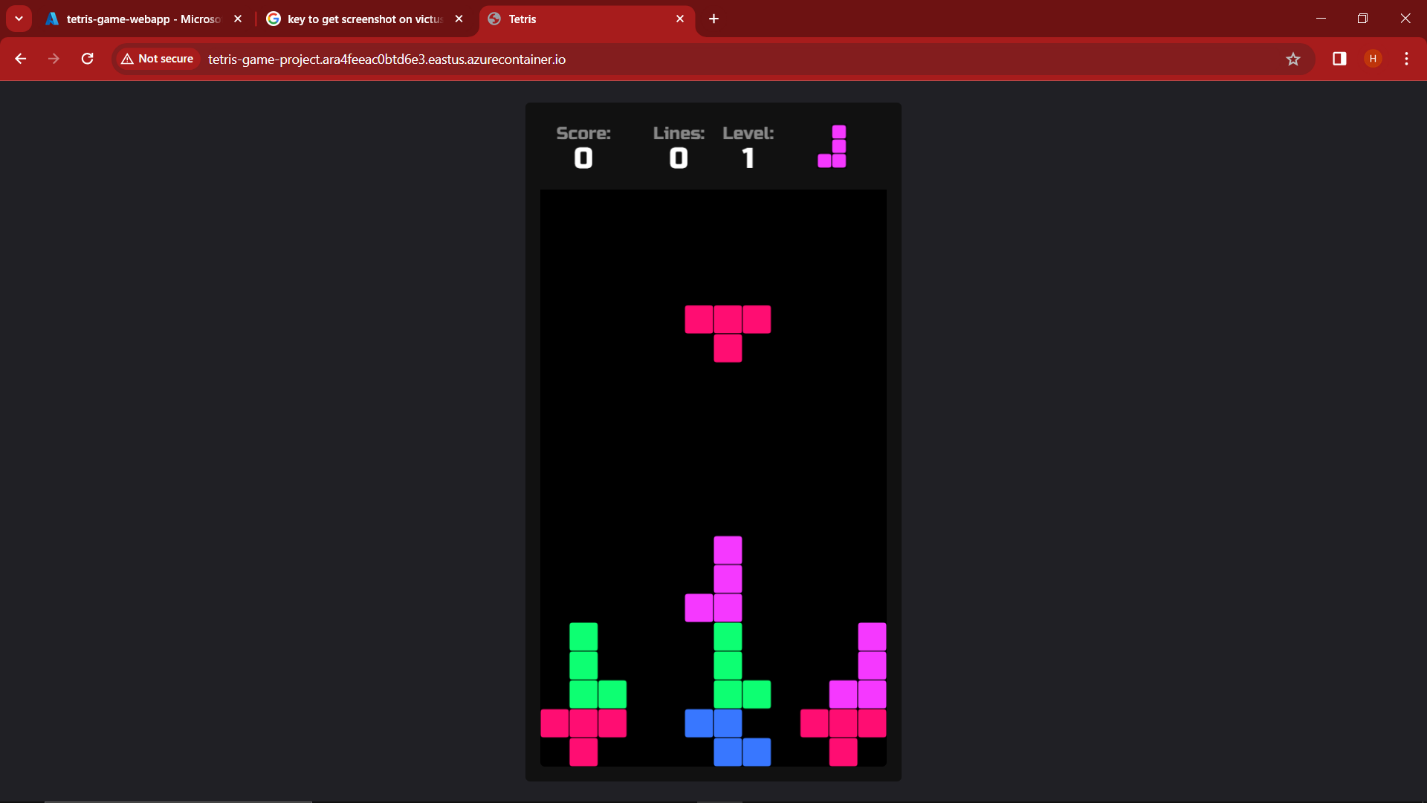
****

* **Docker Image in Azure Container Instance**

****

* **Final output of the project**





**Project Conclusion**

In conclusion, this project successfully addressed the challenge of managing dependencies by packaging the Tetris game app within a Docker image. The use of Azure services, including Virtual Machines, Azure Container Registry, and Azure Container Instances, provided a robust and scalable infrastructure for deploying the game.

Developing the Tetris game app in HTML and JavaScript allowed for flexibility and cross-platform compatibility, making it accessible to a wide range of users. The local development environment ensured that the app was thoroughly tested and fine-tuned before deployment.

Creating a Linux Virtual Machine with Docker on Azure provided a secure and controlled environment for hosting the game. The use of Docker technology simplified the deployment process and allowed for easy customization through Dockerfiles.

Azure Container Registry played a crucial role in securely storing the Docker image, ensuring version control and access management. The Azure CLI streamlined the interaction with Azure services, making deployment more efficient.

Overall, this project demonstrated the practical application of Azure services and Docker technology in a real-world scenario, highlighting their potential for future development and deployment of applications.