Release for Code Engine- AWS

## 1. Create EC2 Instance on AWS:

⦁ Log in to the AWS Management Console.

⦁ Navigate to the EC2 dashboard.

⦁ Click on "Launch Instance."

⦁ Choose the Ubuntu AMI (Amazon Machine Image).

⦁ Select the instance type as "t2.large."

⦁ Set the instance location to Mumbai (ap-south-1).

⦁ Follow the prompts to configure security groups, storage, tags, and other settings as needed.

⦁ Launch the instance and download the .pem key file.

## 2. Create Deployment Group on Azure DevOps:

Log in to Azure DevOps.

⦁ Navigate to the project where you want to create the deployment group.

⦁ Go to "Pipelines" > "Deployment groups."

⦁ Click on "New" to create a new deployment group.

⦁ Provide a name for the deployment group and optionally, a description.

⦁ Select the appropriate target OS (Linux in this case).

⦁ Follow the instructions to install the deployment agent on the EC2 instance.

## 3. Connect EC2 Instance to Deployment Group:

⦁ Once the deployment agent is installed on the EC2 instance, it will automatically connect to the deployment group.

⦁ Verify the connection status in the Azure DevOps portal to ensure that the EC2 instance is successfully connected to the deployment group.

## 4. Download .pem Key File and Open Command Line:

⦁ Download the .pem key file that you obtained while launching the EC2 instance.

⦁ Open your command line interface (CLI) and navigate to the directory where the .pem file is located.

## 5. Connect to EC2 Instance:

⦁ Use the following command to connect to the EC2 instance:

⦁ Copy code

⦁ ssh -i your-key.pem ubuntu@ec2-instance-public-ip

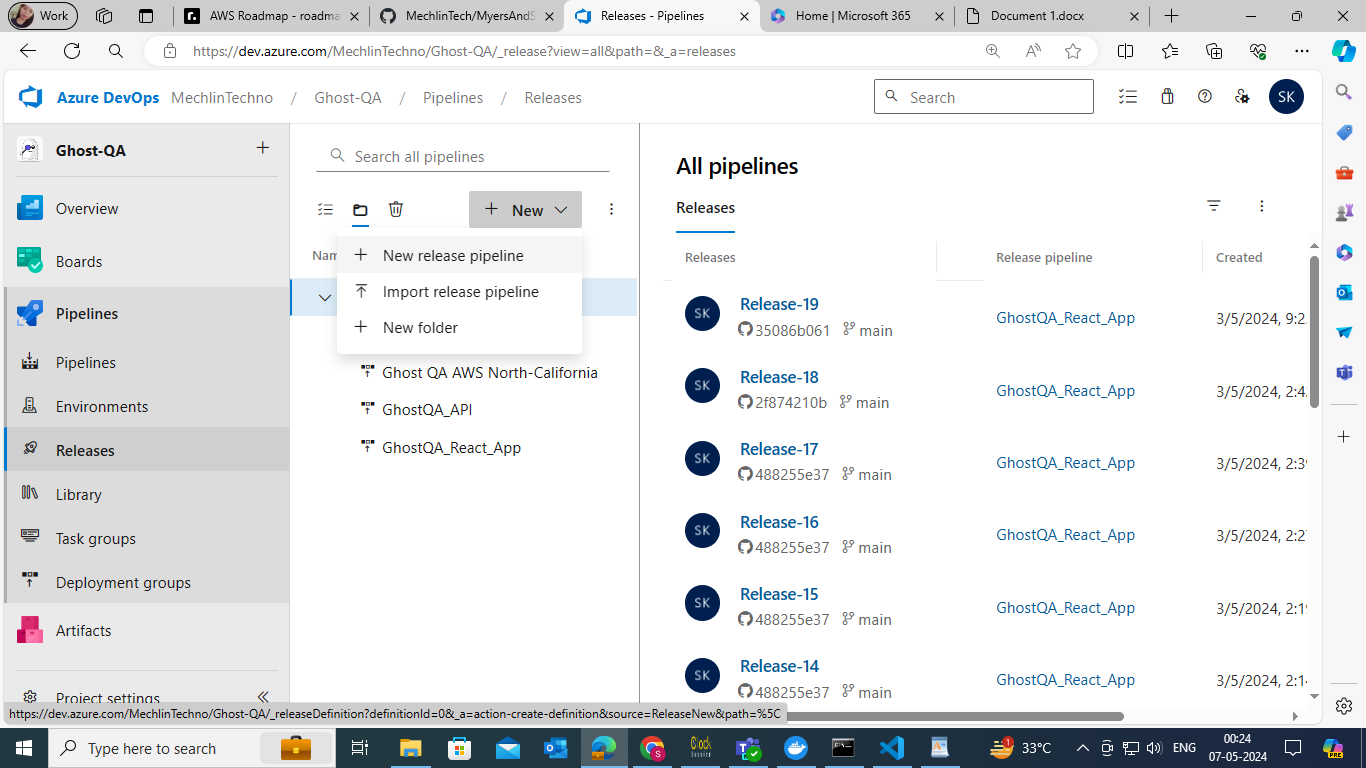
⦁ Replace "your-key.pem" with the name of your .pem key file and "ec2-instance-public-ip" with the public IP address of your EC2 instance.

## 6. Verify Connection:

⦁ After executing the SSH command, you should be connected to the EC2 instance via the command line interface. You can now execute commands on the EC2 instance as needed.

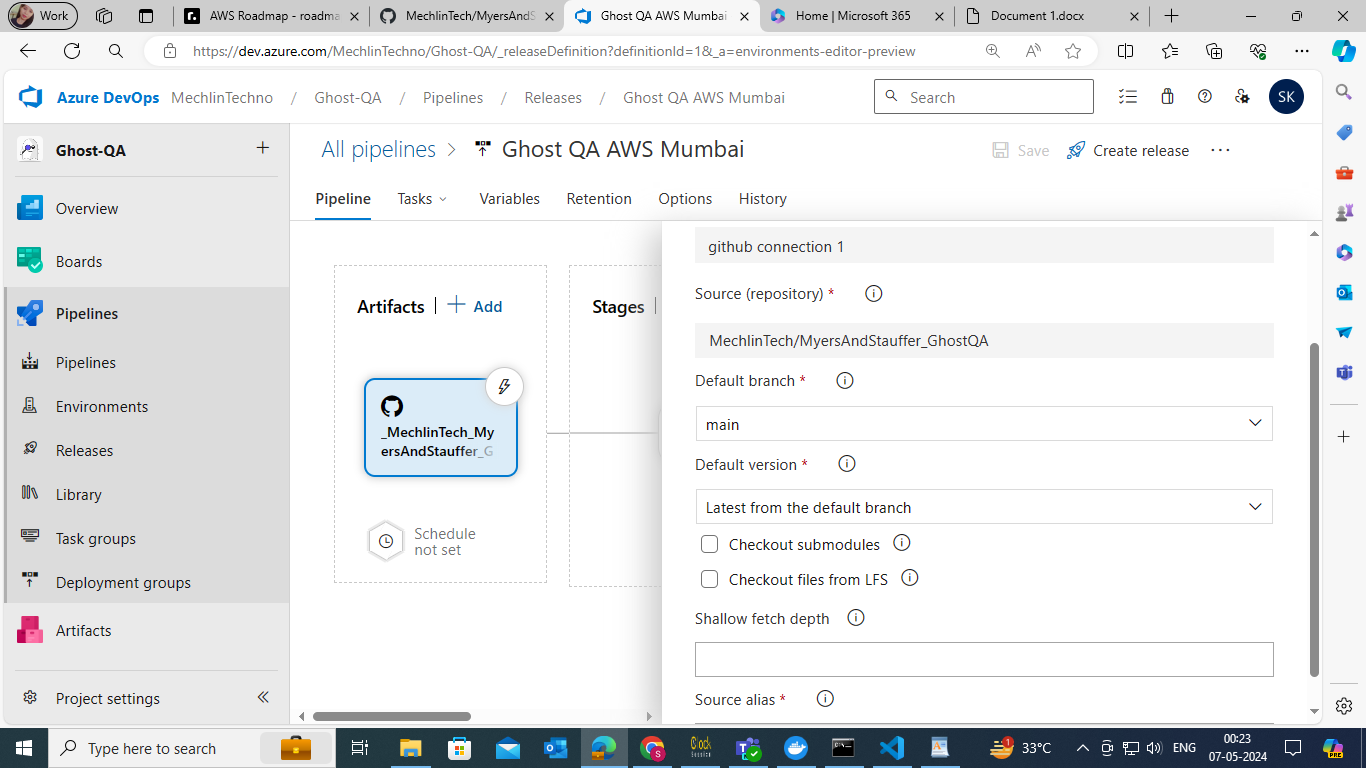
⦁ Download docker in new machine

## 7. Create Pipeline:



Go to the release section and click on "New Pipeline."

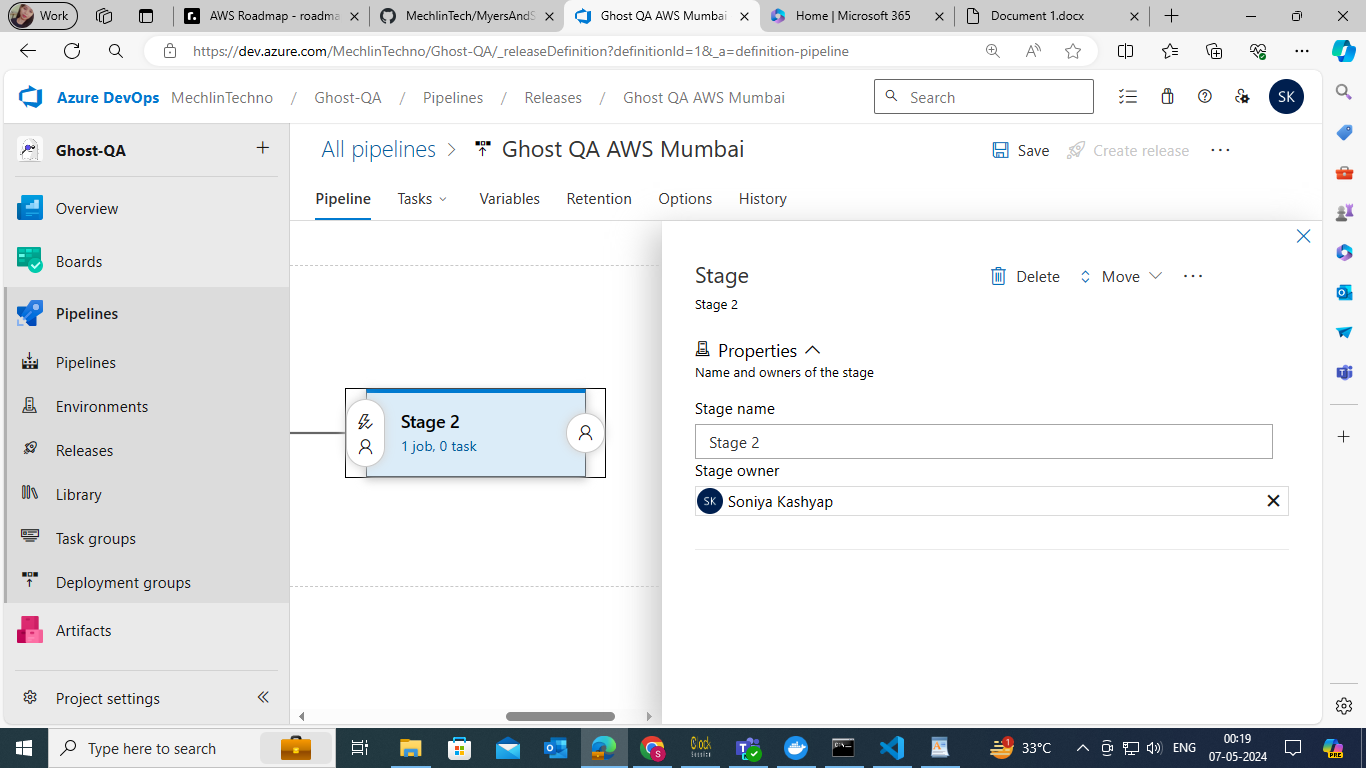
## 8. Add Artifact:



Add the artifact from GitHub: GitHub - \_MechlinTech\_MyersAndStauffer\_GhostQA, branch: main.

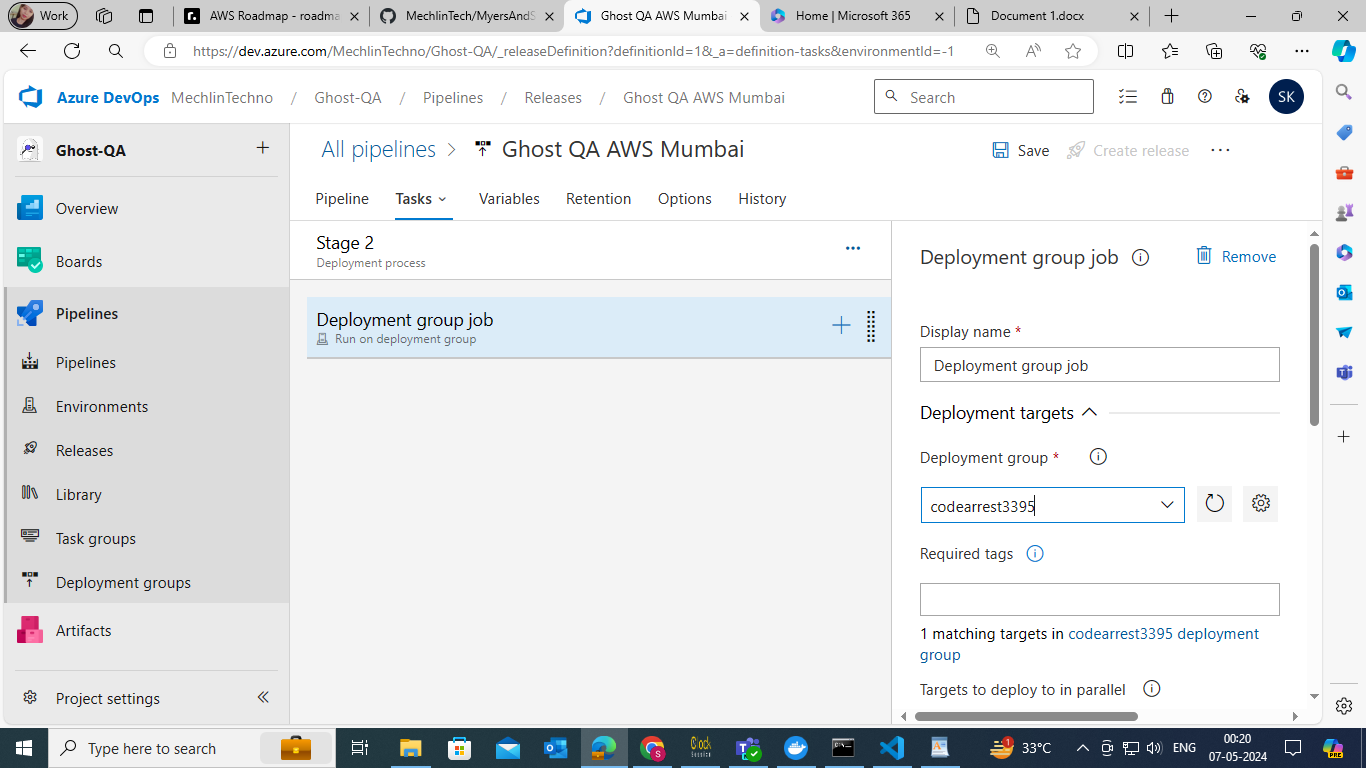
## 9. Create Stage:

⦁ Create a stage with an empty job.

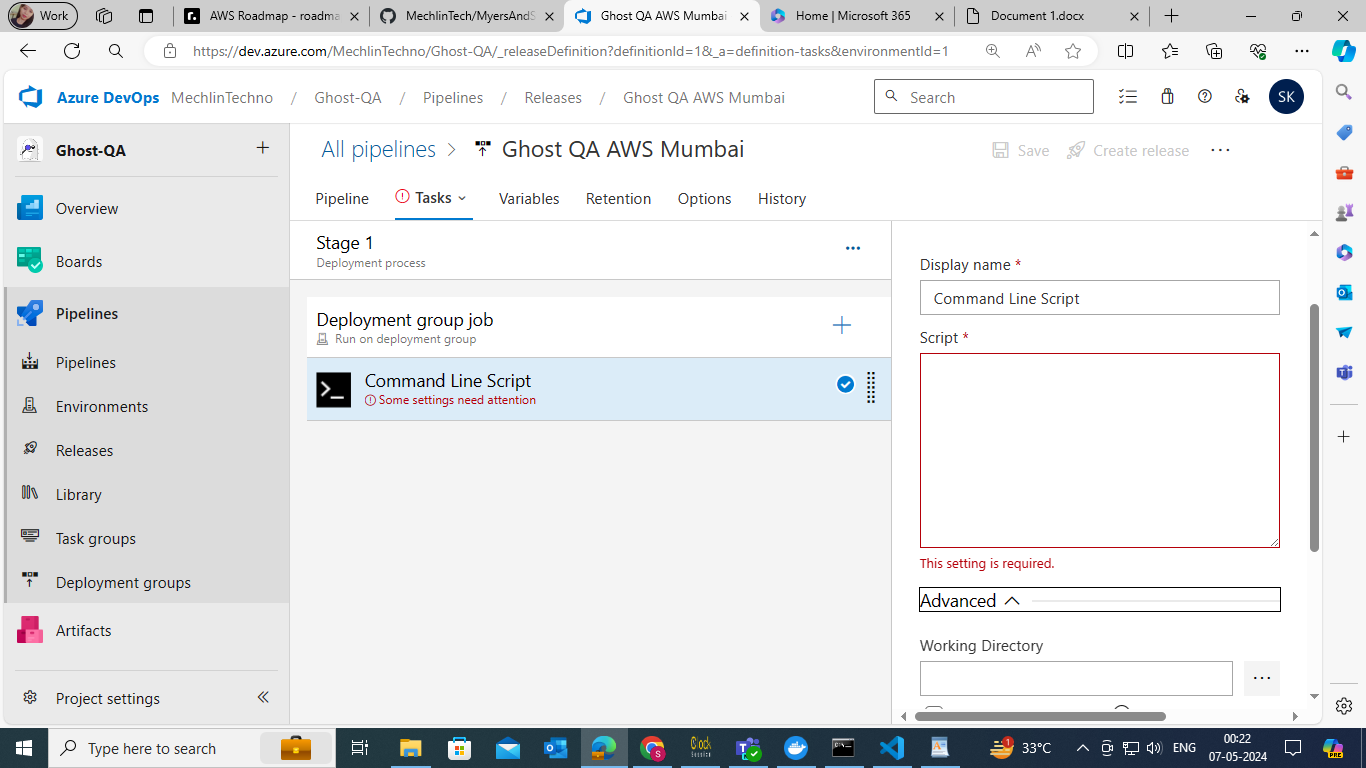


⦁ Configure Deployment Group and Task:

⦁ In the first job of stage 1, click on it, remove the agent job, then click on the ellipsis (...) of stage 1.



⦁ Add the deployment group and then click on the "+" icon to add a task.



⦁ Configure the task as follows:

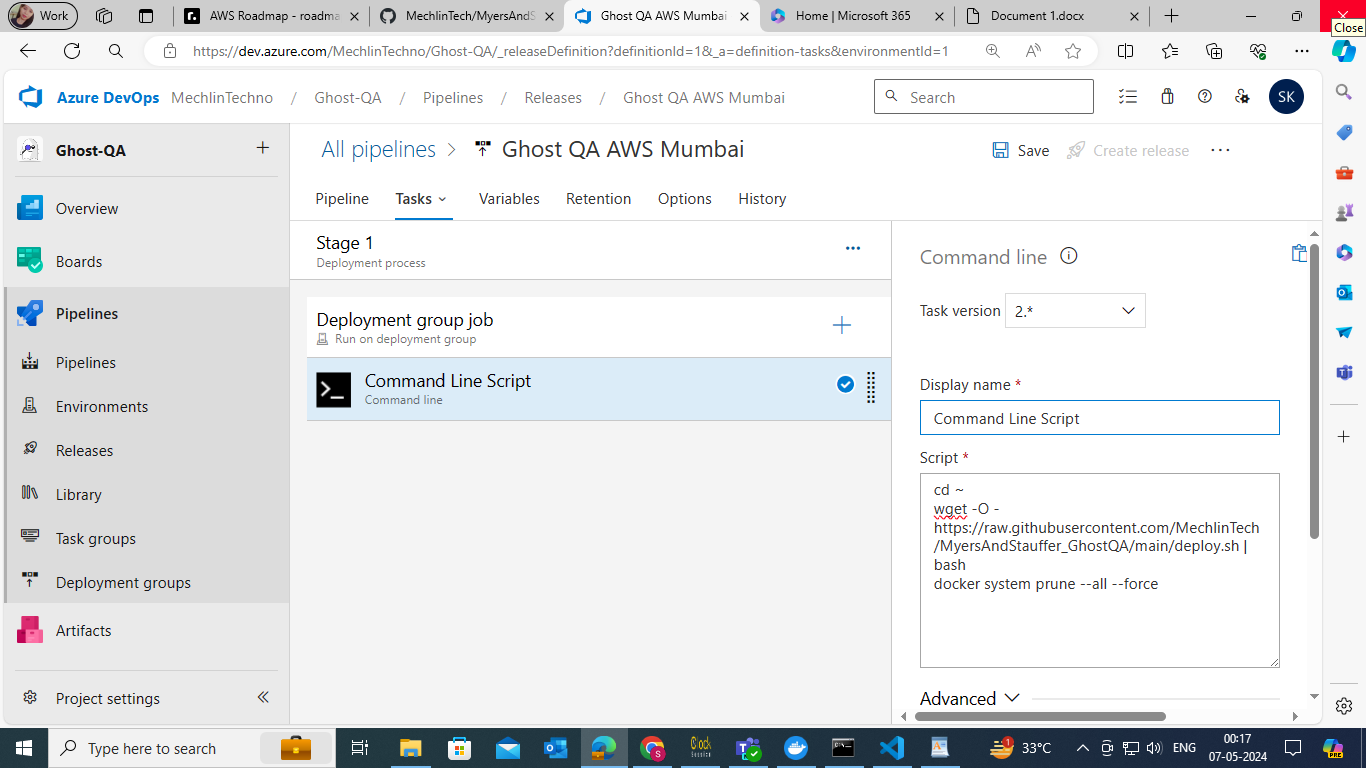
⦁ Type: Command Line

⦁ Command:

cd ~

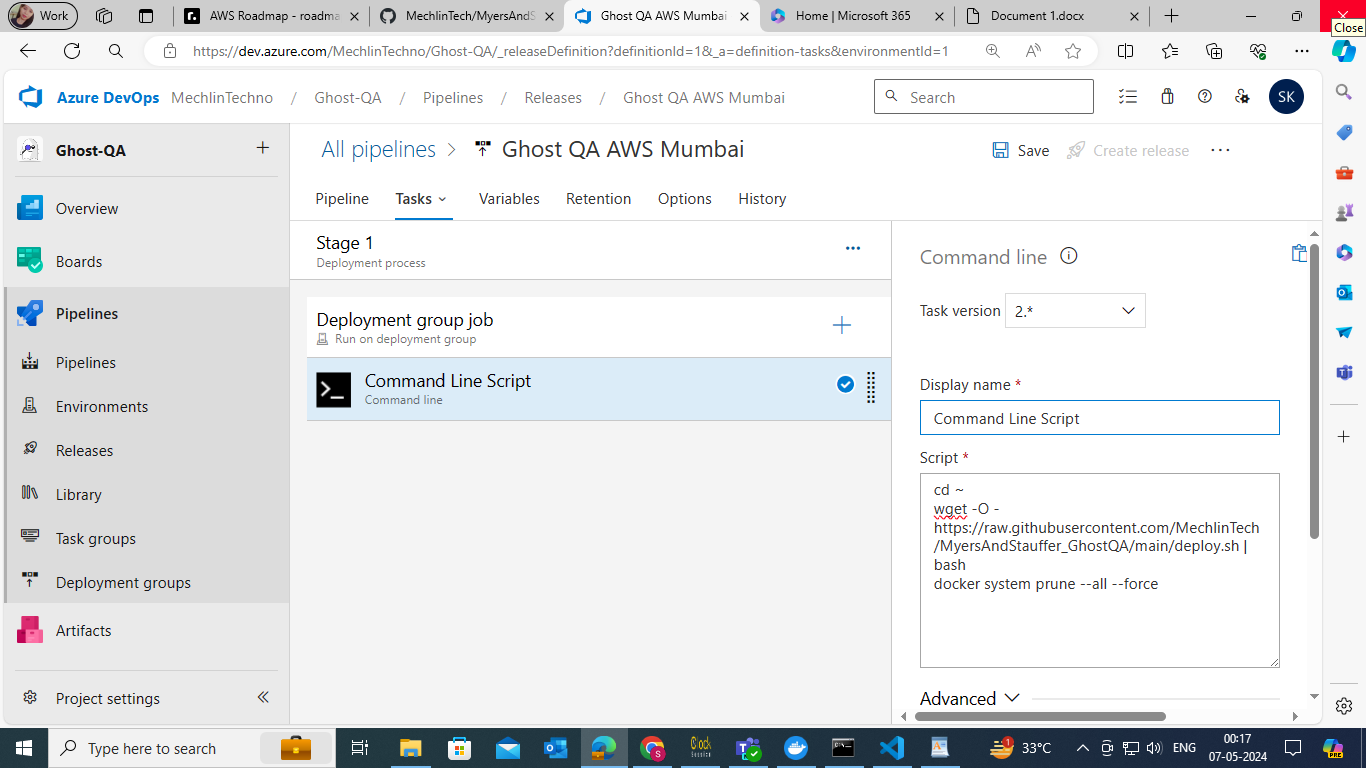
wget -O - https://raw.githubusercontent.com/MechlinTech/MyersAndStauffer\_GhostQA/main/deploy.sh | bash

docker system prune --all –force

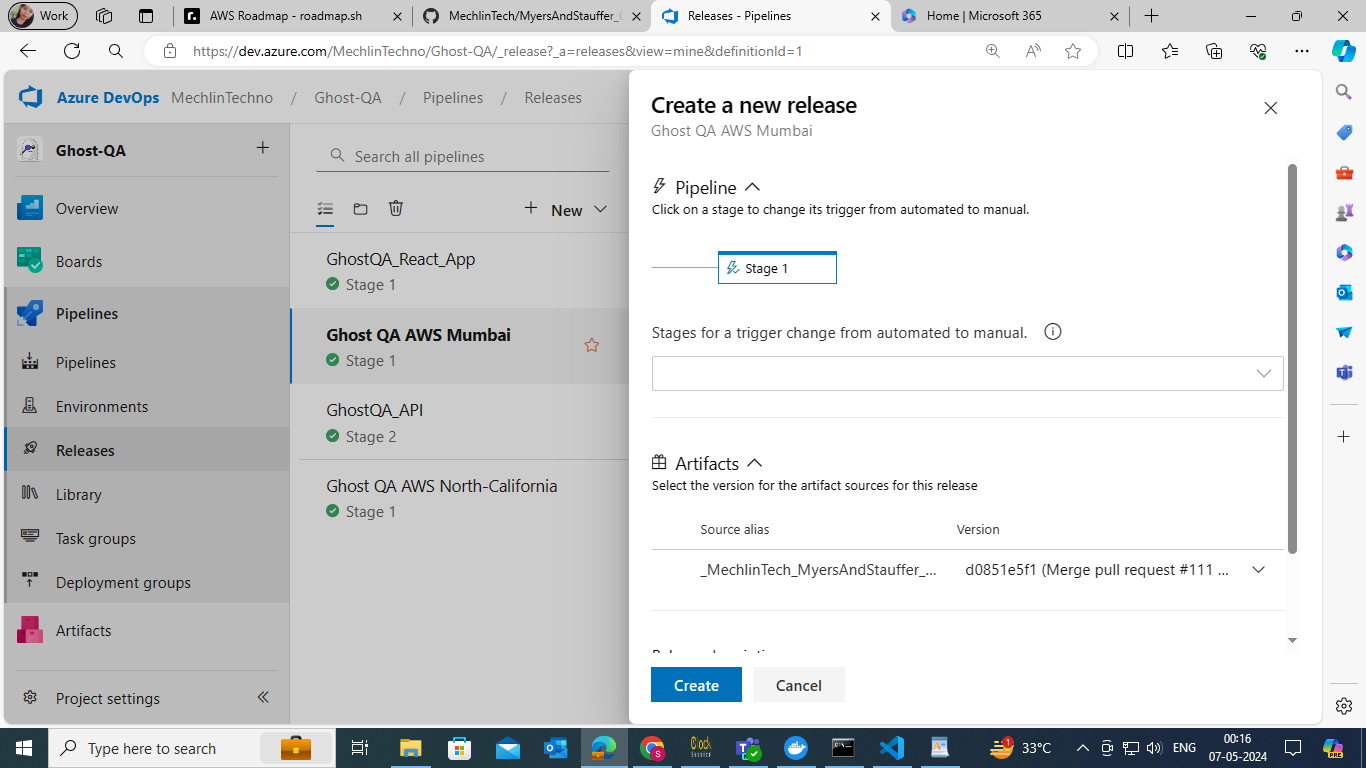


## 10. Save Pipeline:

Save the pipeline configuration.



## 11. Create Release:

Create the release and wait for it to run. Once completed, your pipeline is successfully created.