Release Pipeline for Framework .Net API and Frontend – Windows

# Release Pipeline for .Net API:-

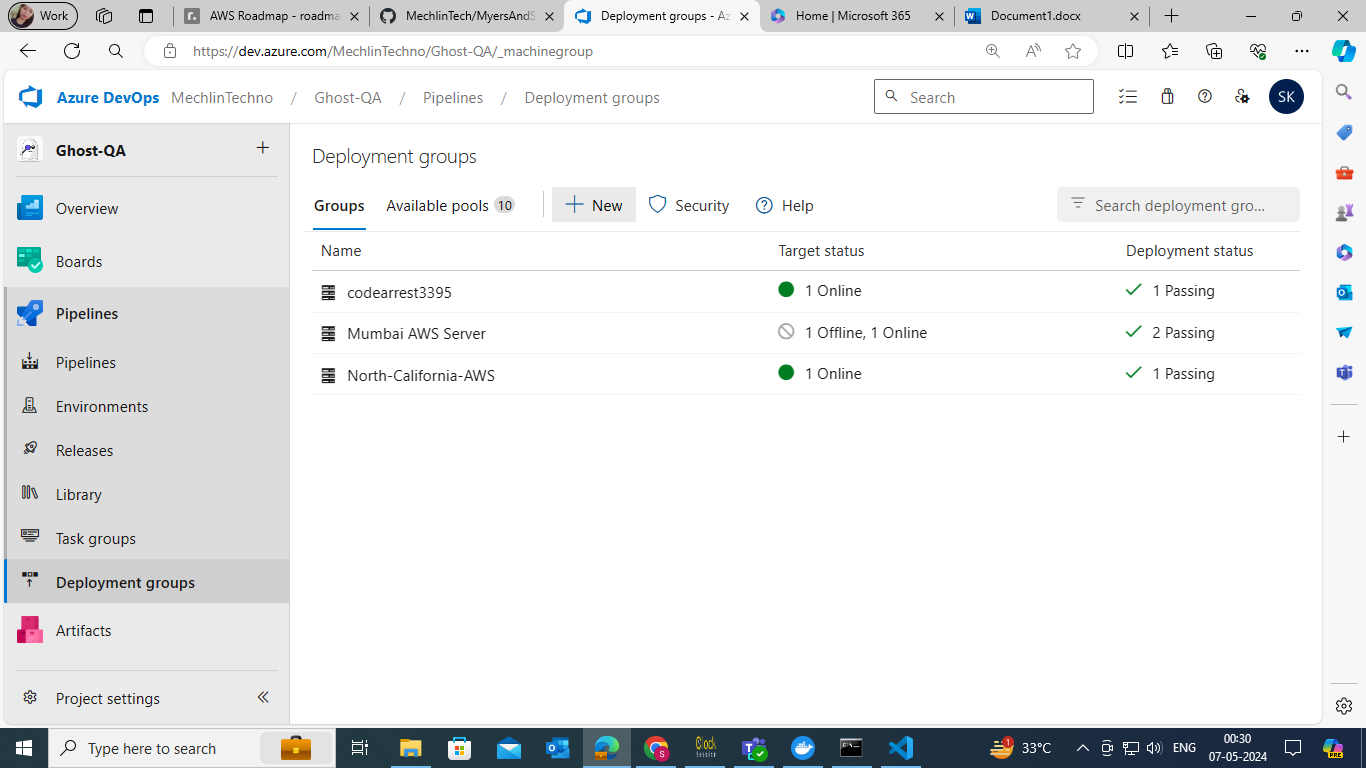
## Create Deployment Group on Azure DevOps:

⦁ Log in to Azure DevOps.

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

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

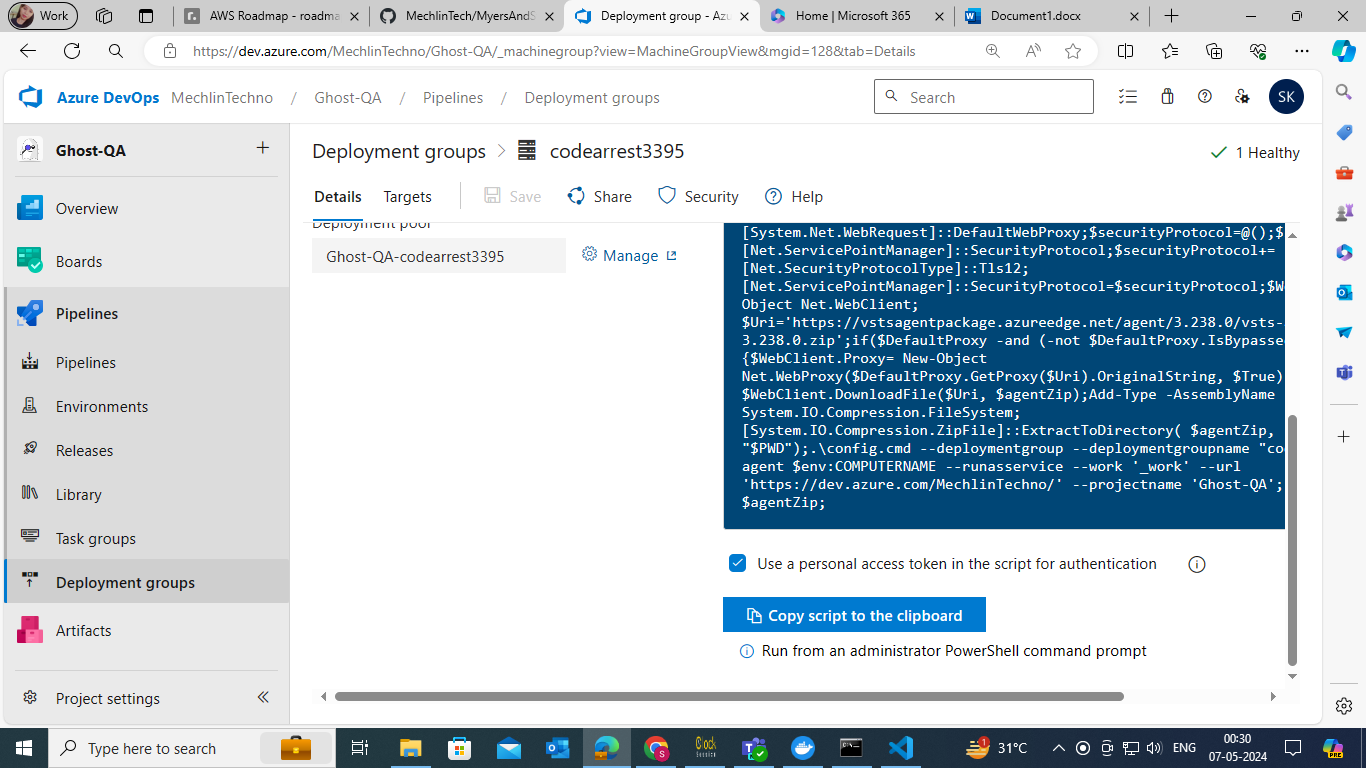
⦁ Click on "New" to create a new deployment group.



⦁ Provide a name for the deployment group (e.g., codearrest\_3395).

⦁ Select the OS as Windows.

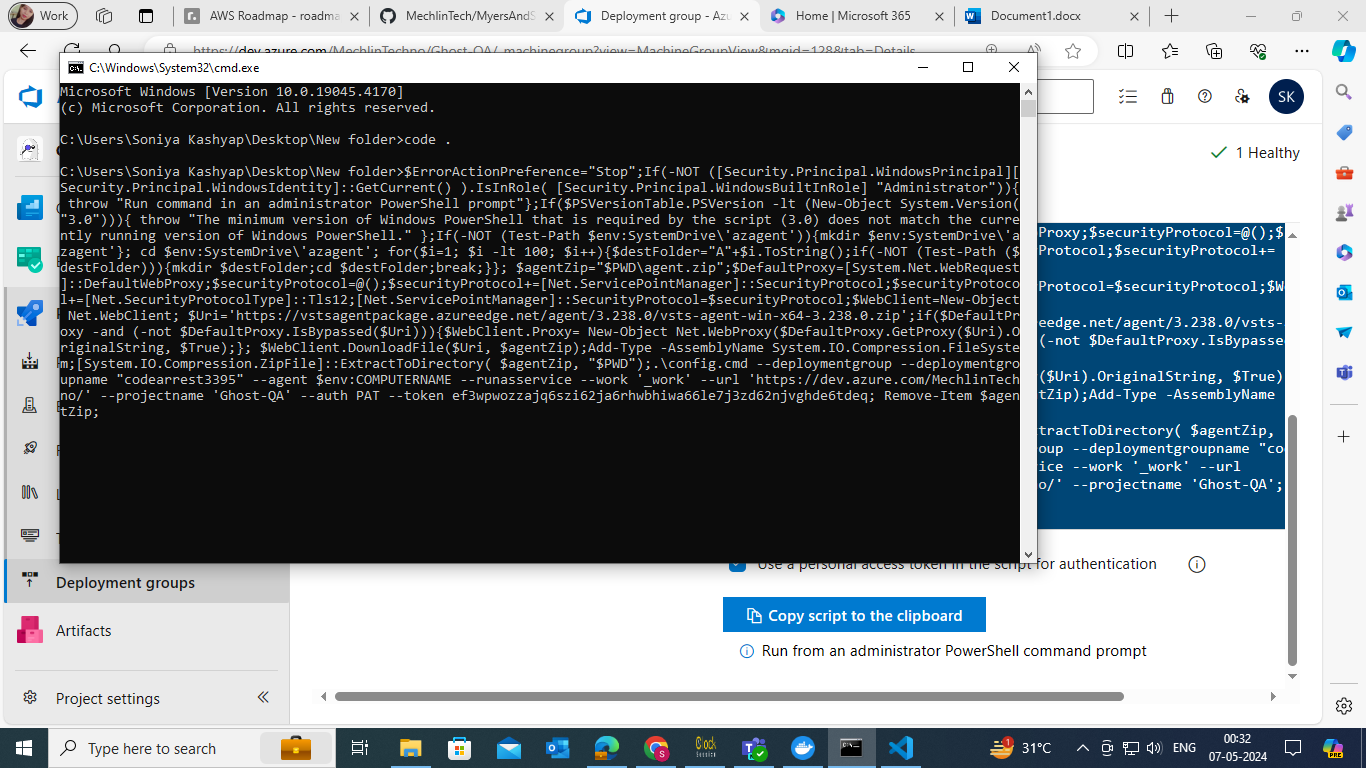
⦁ Copy the command provided along with the key.



## Set Up Deployment Group Agent:

⦁ Access your server (e.g., 3395).

⦁ Open the command prompt and paste the copied command.

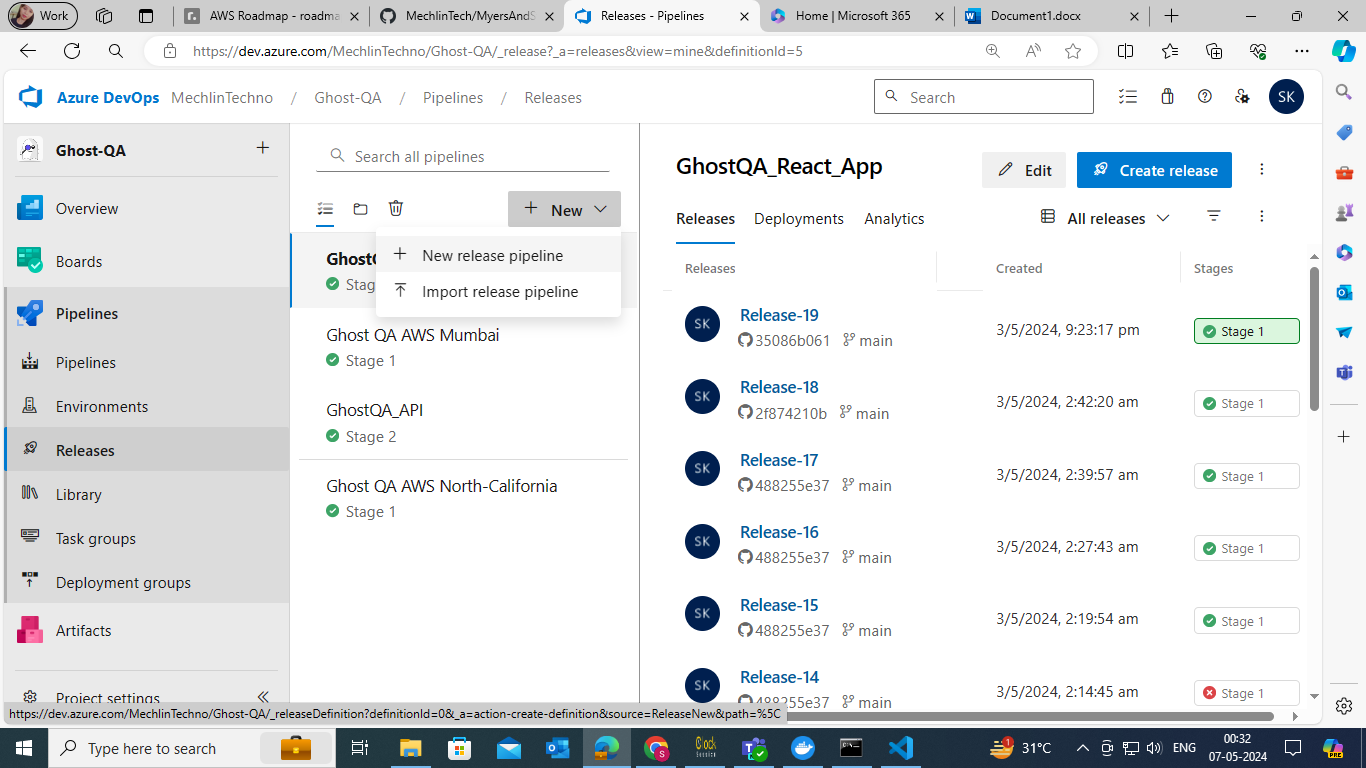


⦁ Press enter to create the agent group.

⦁ Create Release Pipeline:

⦁ Navigate to "Pipelines" > "Releases" in Azure DevOps.

⦁ Click on "New Pipeline."



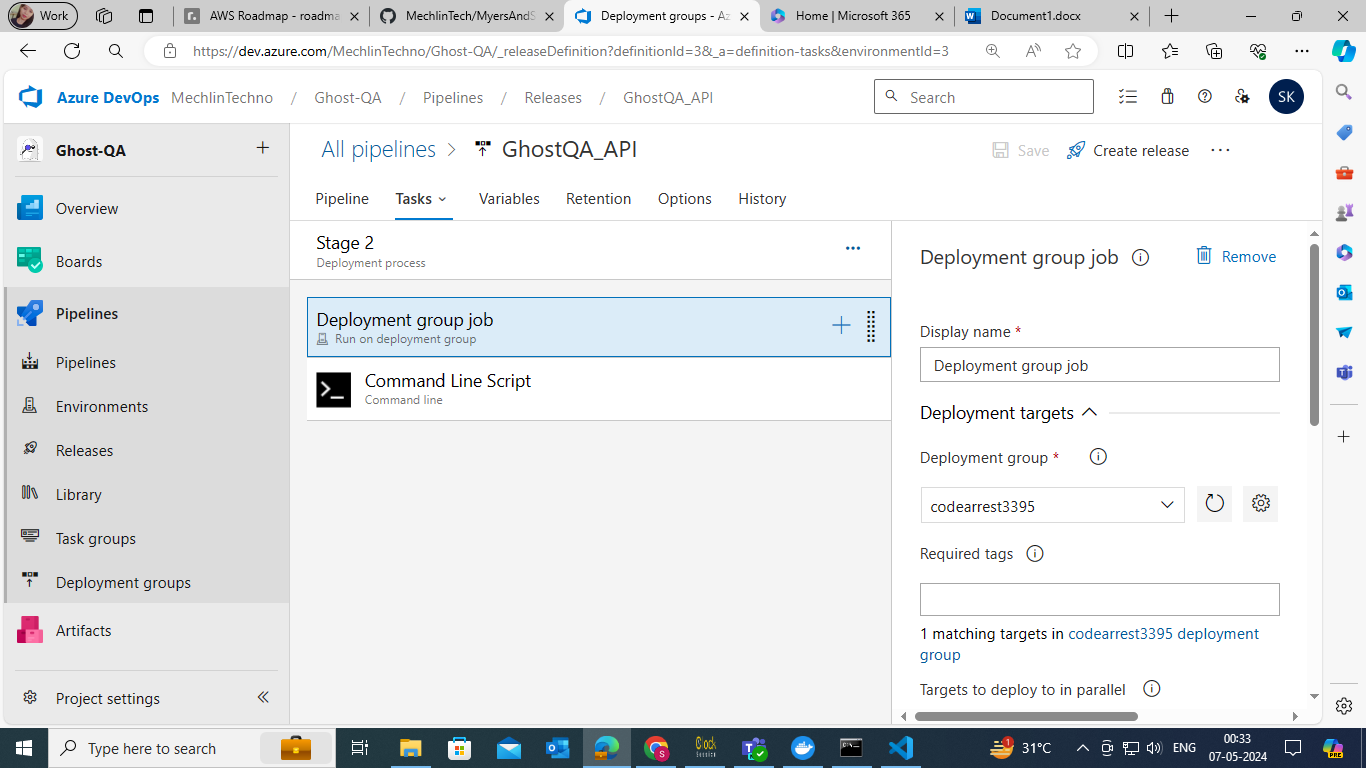
⦁ Add the appropriate artifact.

⦁ Create an empty stage.

Configure Release Pipeline Task:

⦁ In the stage, click on the job/task.

⦁ Remove the default agent job.



⦁ Add a new task: it will be cmd task

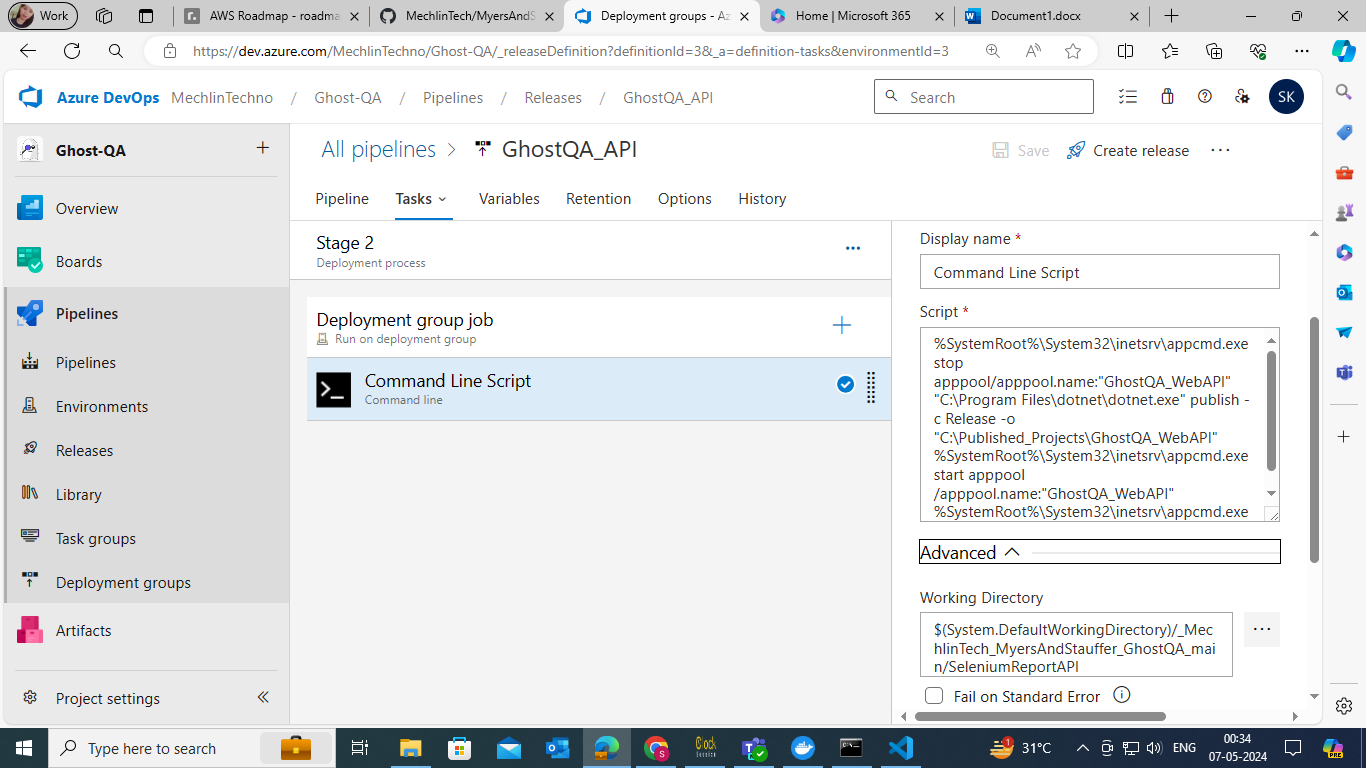
⦁ Type: Command Line

%SystemRoot%\System32\inetsrv\appcmd.exe stop apppool/apppool.name:"GhostQA\_WebAPI"

"C:\Program Files\dotnet\dotnet.exe" publish -c Release -o "C:\Published\_Projects\GhostQA\_WebAPI"

%SystemRoot%\System32\inetsrv\appcmd.exe start apppool /apppool.name:"GhostQA\_WebAPI"

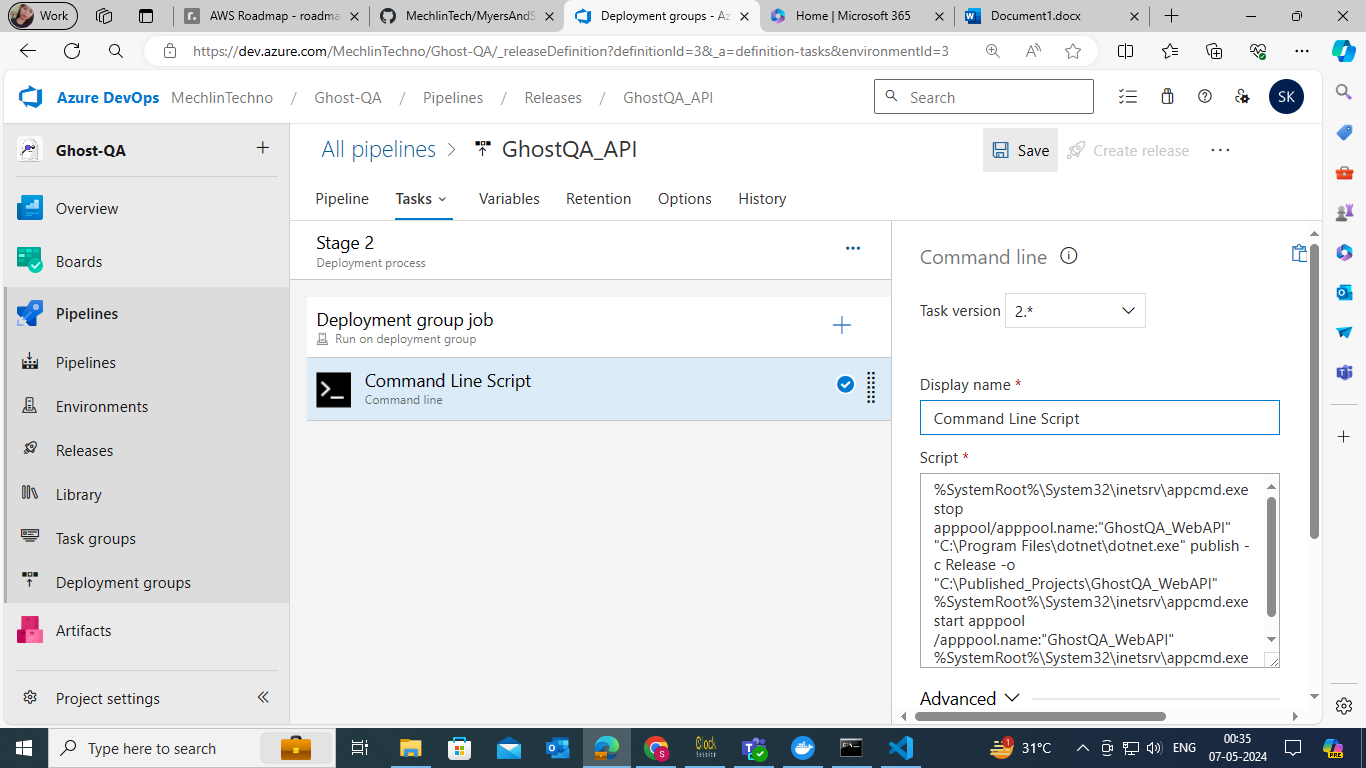
%SystemRoot%\System32\inetsrv\appcmd.exe start site /site.name:"GhostQA\_WebAPI"



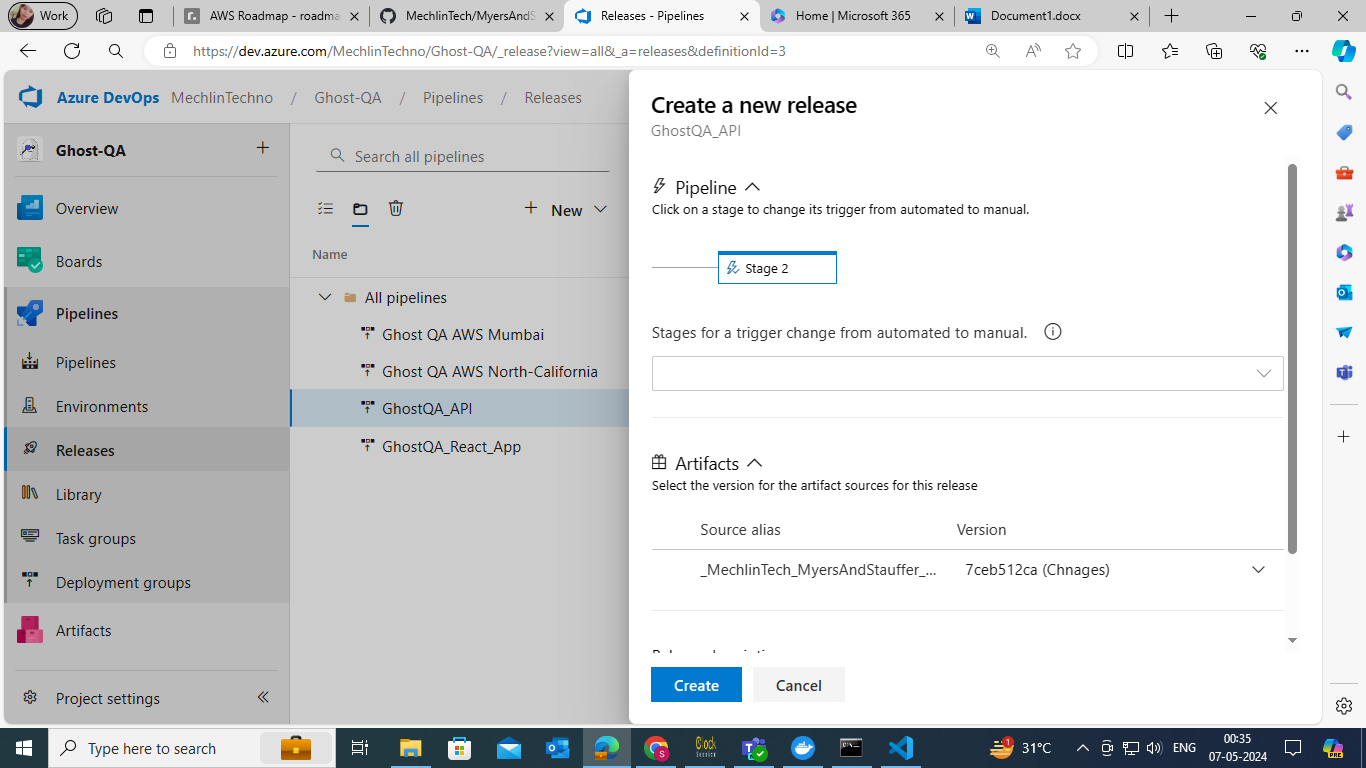
In this command you have to change to apppool name, site name, and the publishing path.

In this command we are stoping IIS manager apppool first and then publishing the code after that we are starting the pool and site again.

Save Pipeline and Create Release:



⦁ Save the pipeline configuration.



⦁ Create a release.

# Release Pipeline for React App

## Create the release pipeline:

⦁ using same deployment group create the pipeline

⦁ Navigate to "Pipelines" > "Releases" in Azure DevOps.

⦁ Click on "New Pipeline."

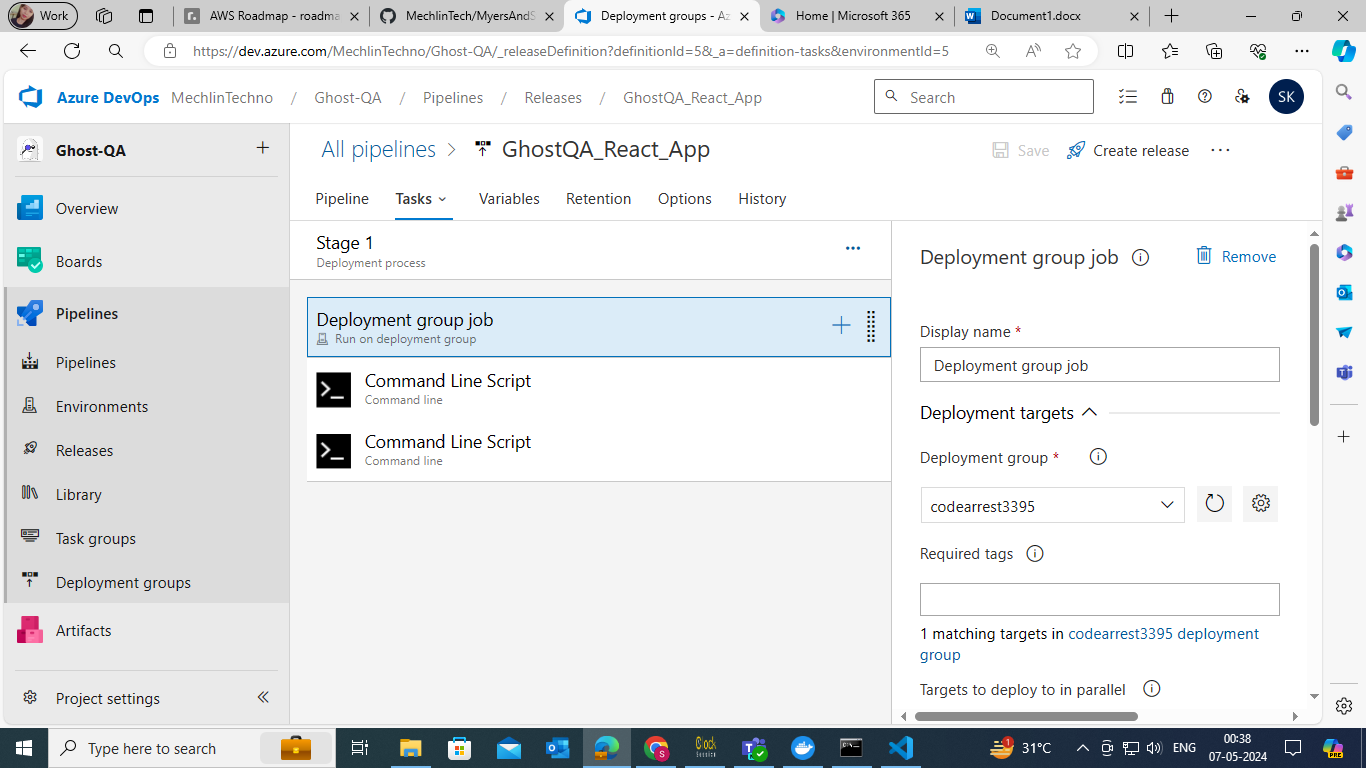
⦁ Add the appropriate artifact.

⦁ Create an empty stage.

Configure Release Pipeline Task:

⦁ In the stage, click on the job/task.

⦁ Remove the default agent job.

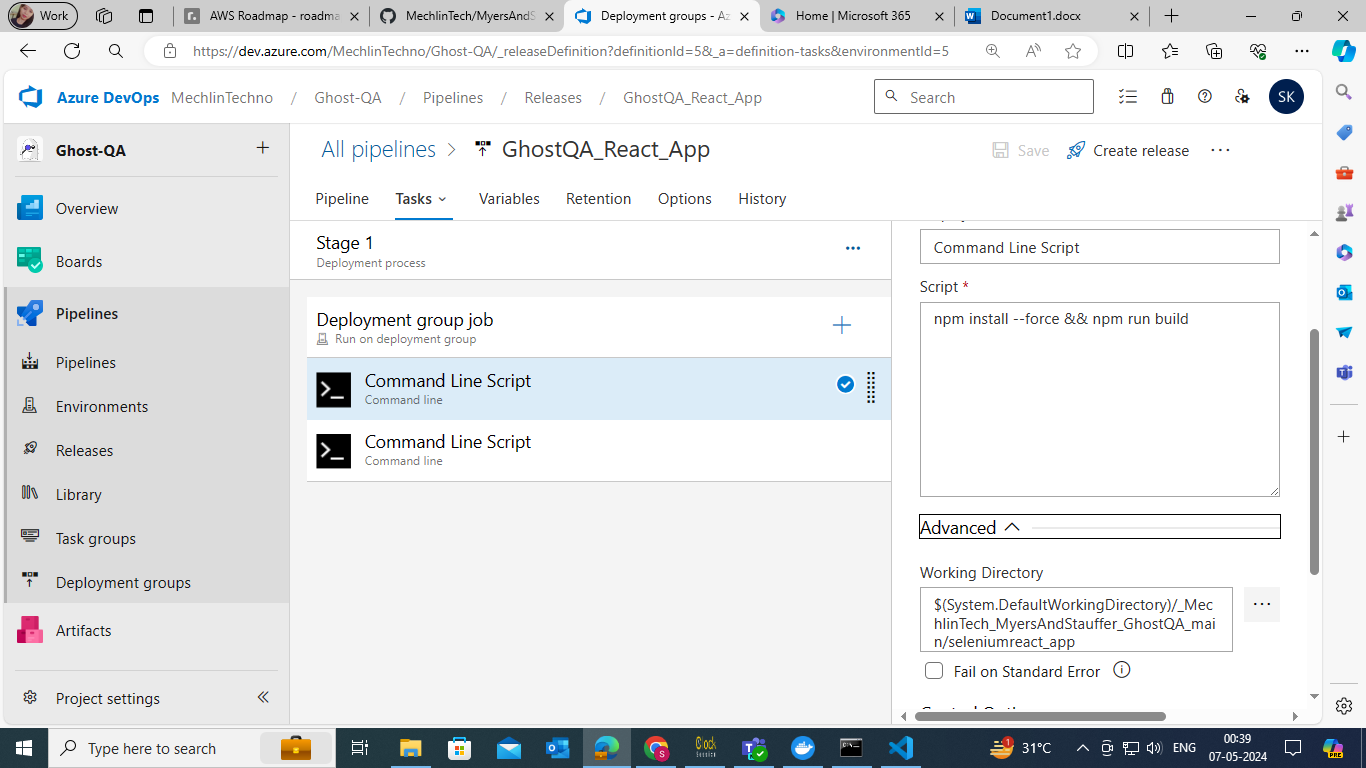


⦁ Add two new task: it will be cmd task

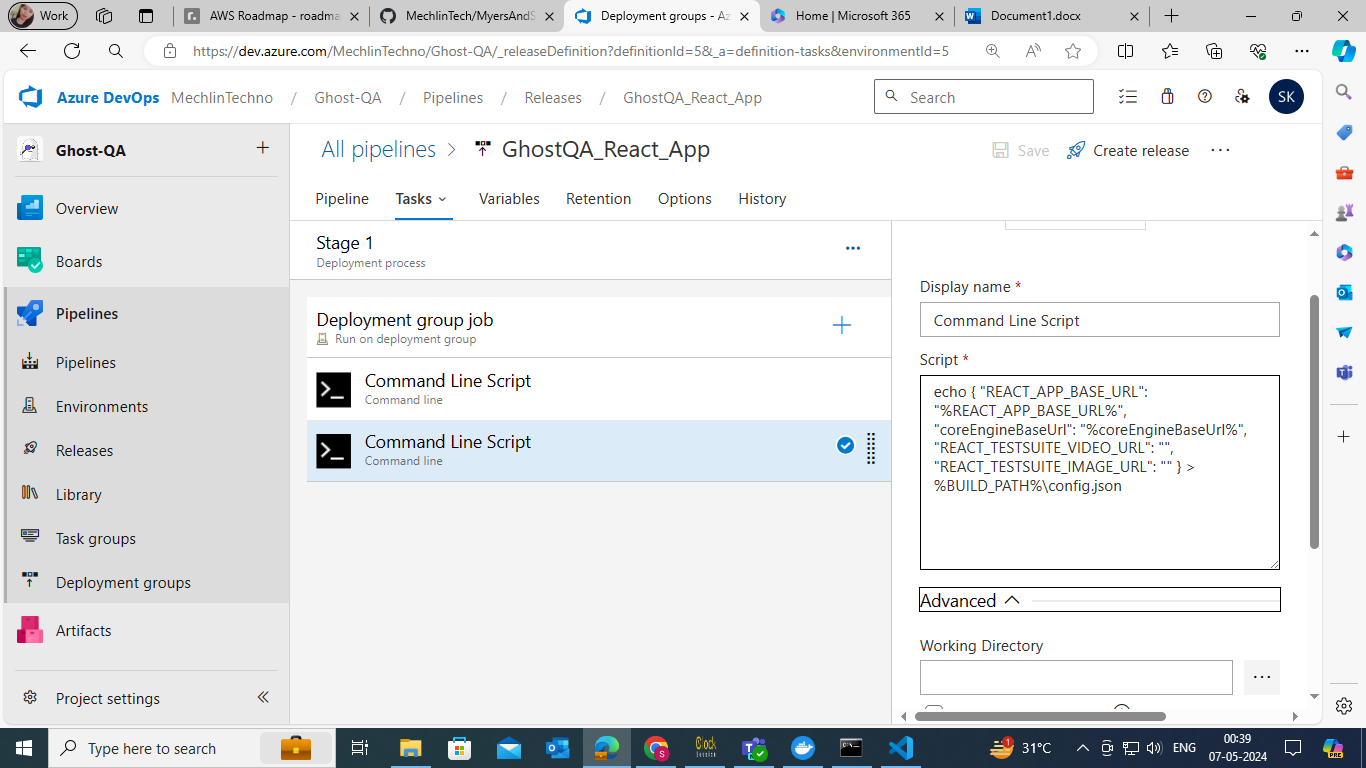
⦁ Both task will be cmd task

⦁ In first task we will run the npm and build using below command

npm install --force && npm run build

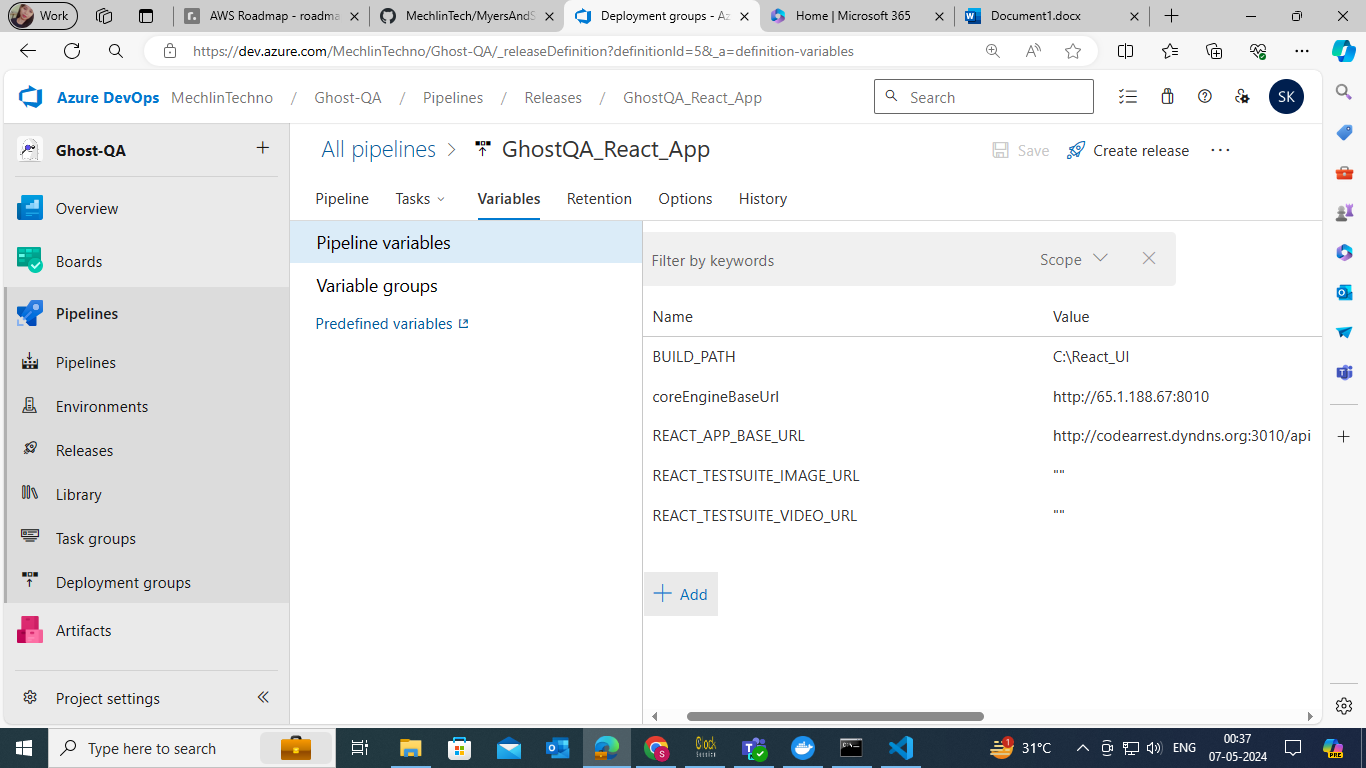


⦁ In the 2 one we will update the config.json using below command



echo { "REACT\_APP\_BASE\_URL": "%REACT\_APP\_BASE\_URL%", "coreEngineBaseUrl": "%coreEngineBaseUrl%", "REACT\_TESTSUITE\_VIDEO\_URL": "", "REACT\_TESTSUITE\_IMAGE\_URL": "" } > %BUILD\_PATH%\config.json

⦁ In this command we are using some enviroment variable which we have to define in pipeline using %% we can use that .



⦁ After this save the pipeline and create the release.

