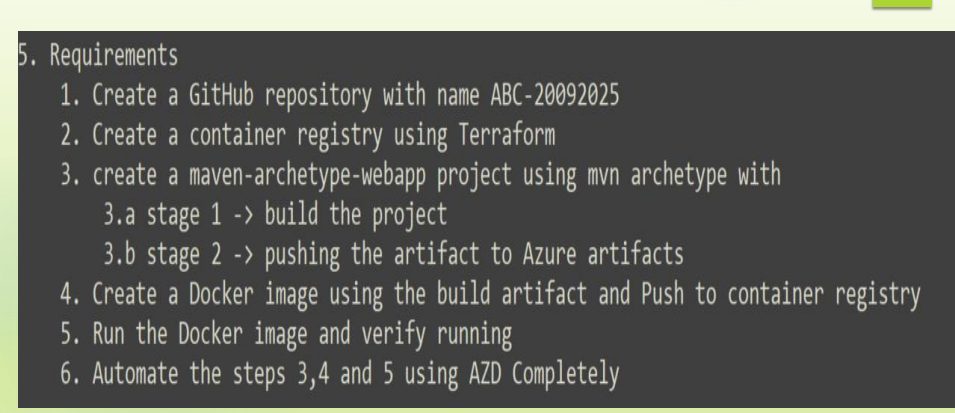
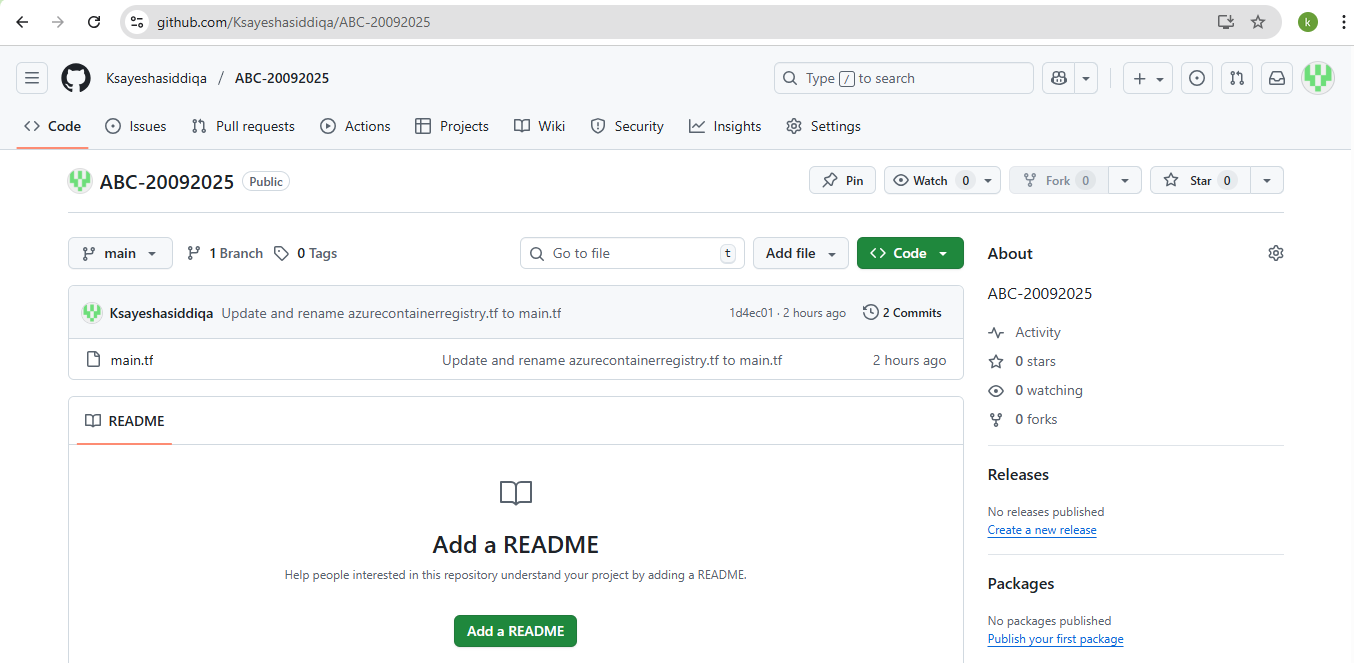
**TASKS ON AZURE DEVEOPS**

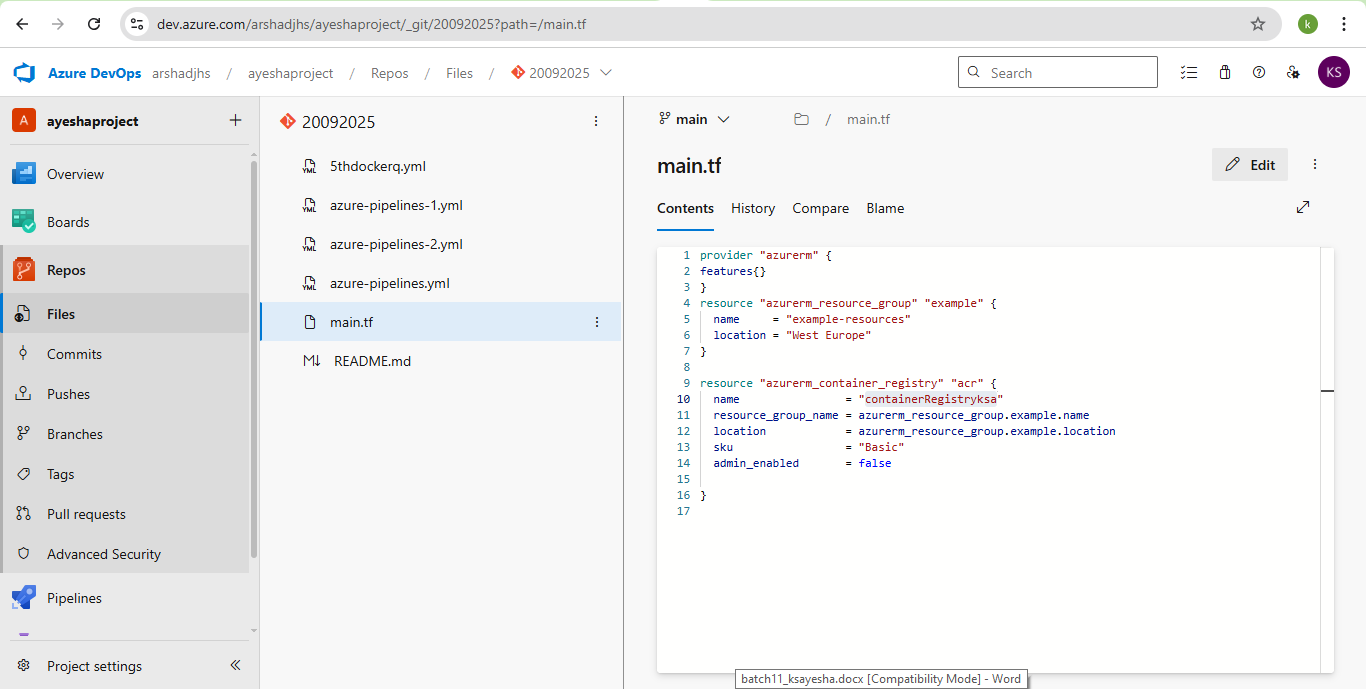
NAME: K S AYESHA SIDDIQA

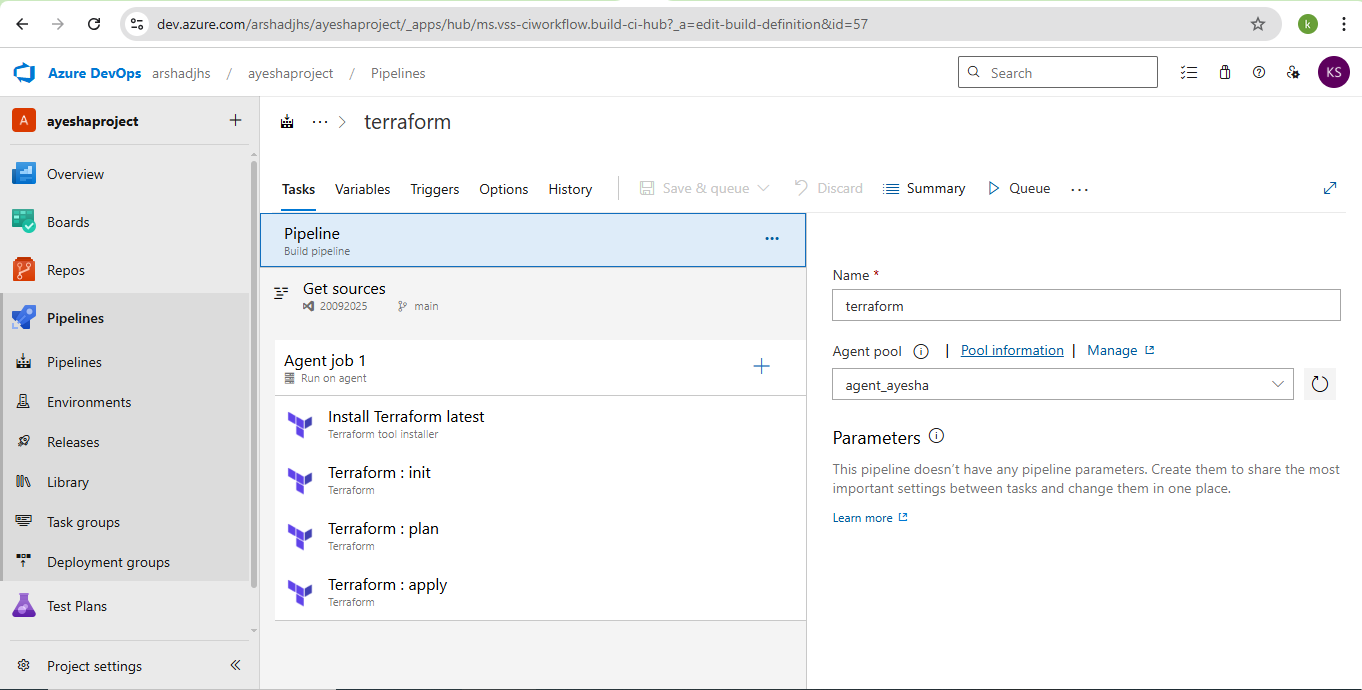


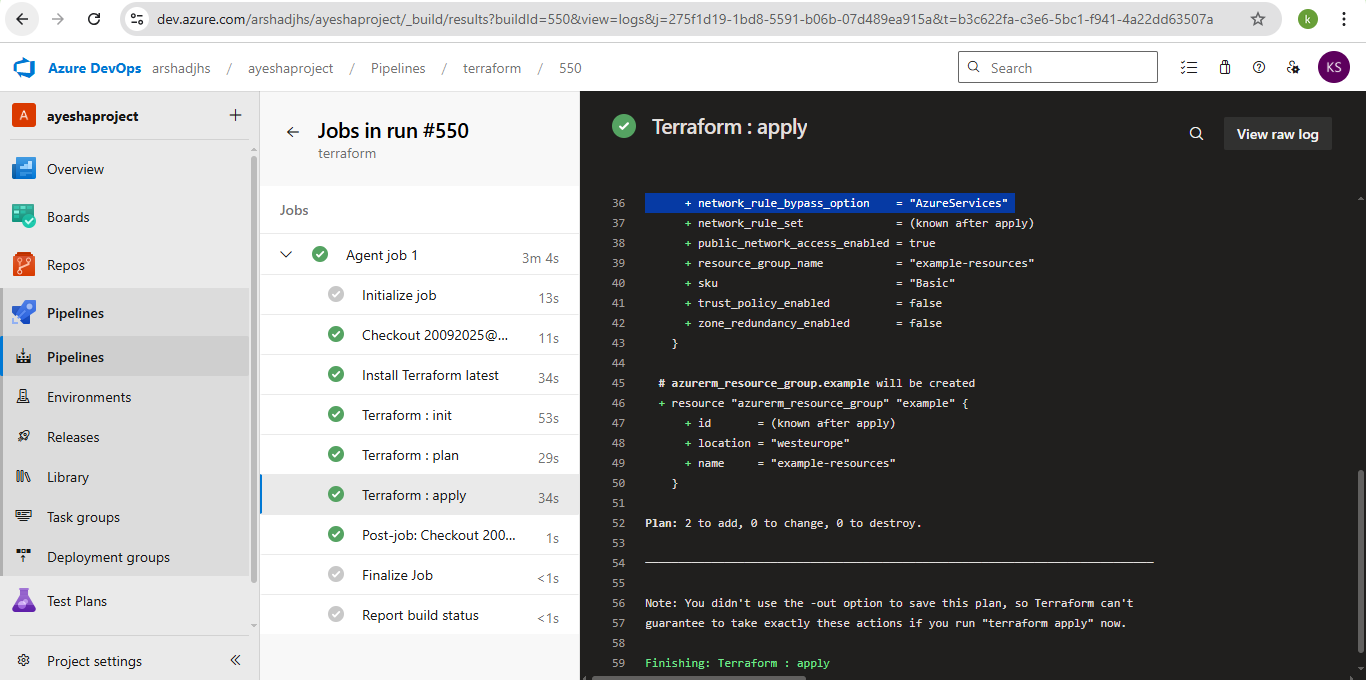
1.

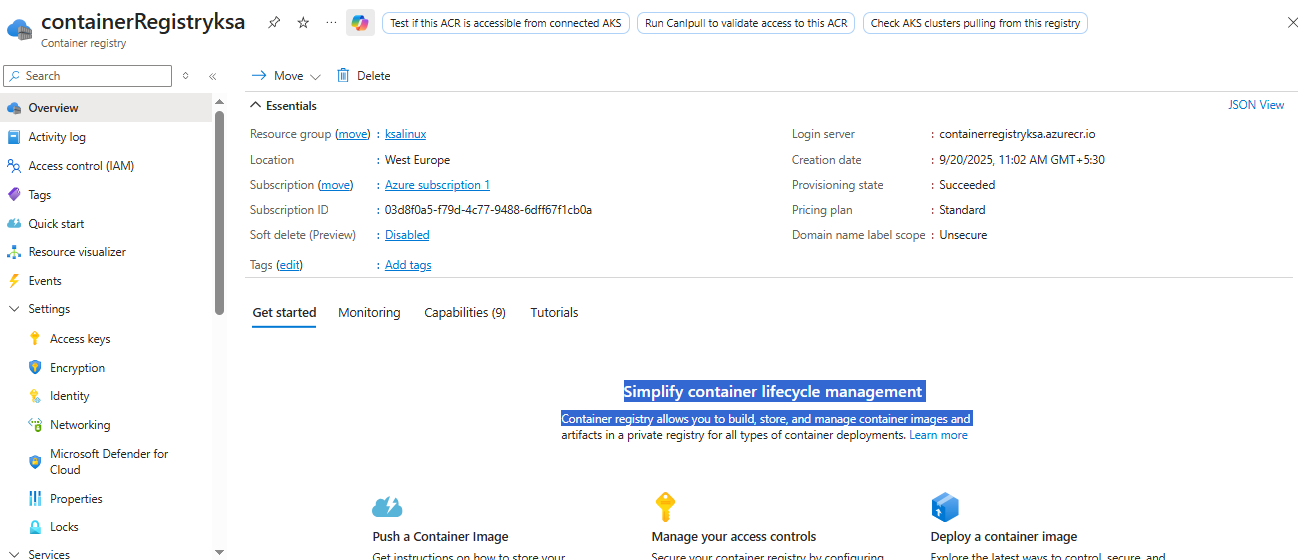


2.









3.

trigger:

- main

# Build number format: YYYYMMDD.r

name: $(date:yyyyMMdd)$(rev:.r)

variables:

- name: BuildParameters.mavenPOMFile

value: pom.xml

resources:

repositories:

- repository: self

type: git

ref: refs/heads/main

jobs:

- job: MavenBuildAndPublish

displayName: 'Maven Build and Publish JAR'

pool:

name: 'agent\_ayesha' # Your agent pool

demands:

- agent.name -equals agent\_ayesha\_linux # Run only on this agent

steps:

# Step 1: Checkout repo

- checkout: self

clean: true

fetchTags: false

# Step 2: Run Maven build

- task: Maven@4

displayName: 'Build with Maven (pom.xml)'

inputs:

mavenPOMFile: $(BuildParameters.mavenPOMFile)

goals: 'clean package'

# Step 3: Copy built JAR files to staging directory

- task: CopyFiles@2

displayName: 'Copy JAR Files to: $(Build.ArtifactStagingDirectory)'

condition: succeededOrFailed()

inputs:

SourceFolder: 'Amazon-Core/target'

Contents: '\*\*/\*.jar'

TargetFolder: '$(Build.ArtifactStagingDirectory)'

# Step 4: Publish the JARs as artifacts

- task: PublishBuildArtifacts@1

displayName: 'Publish Artifact: drop'

condition: succeededOrFailed()

inputs:

PathtoPublish: '$(Build.ArtifactStagingDirectory)'

ArtifactName: 'drop'

publishLocation: 'Container'

I have error IN THE push to artifact

4.

Pipeline.yml

# Docker

# Build and push an image to Azure Container Registry

# https://docs.microsoft.com/azure/devops/pipelines/languages/docker

trigger:

- main

resources:

- repo: self

pool:

name: 'agent\_ayesha' # your agent pool

demands:

- agent.name -equals agent\_ayesha\_linux

variables:

# Container registry service connection established during pipeline creation

dockerRegistryServiceConnection: '8c952ee3-4196-455c-b905-a09224064f6f'

imageRepository: 'ksayeshasiddiqaproject'

containerRegistry: 'ksalinuxacr.azurecr.io'

dockerfilePath: '$(Build.SourcesDirectory)Dockerfile'

tag: '$(Build.BuildId)'

stages:

- stage: Build

displayName: Build and push stage

jobs:

- job: Build

displayName: Build

steps:

- task: Docker@2

displayName: Build and push an image to container registry

inputs:

command: buildAndPush

repository: $(imageRepository)

dockerfile: $(dockerfilePath)

containerRegistry: $(dockerRegistryServiceConnection)

tags: |

$(tag)

Dockerfile:

#DOCKERFILE FOR THE 4th question

# Use a base image

FROM python:3.12-slim

# Set working directory inside container

WORKDIR /app

# Copy requirements first (for caching)

COPY requirements.txt .

# Install dependencies

RUN pip install --no-cache-dir -r requirements.txt

# Copy the rest of the app

COPY . .

# Expose port (change if needed)

EXPOSE 5000

# Set default command

CMD ["python", "app.py"]

requirements.txt:

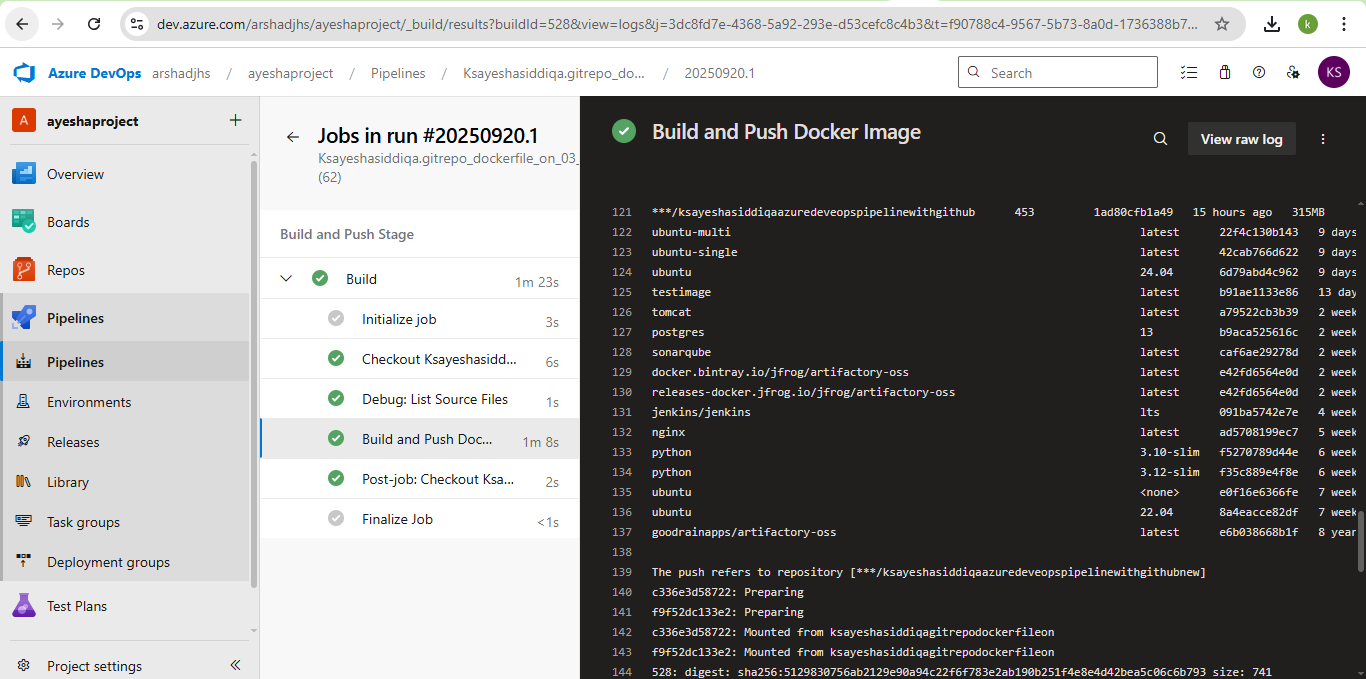
flask==3.0.2

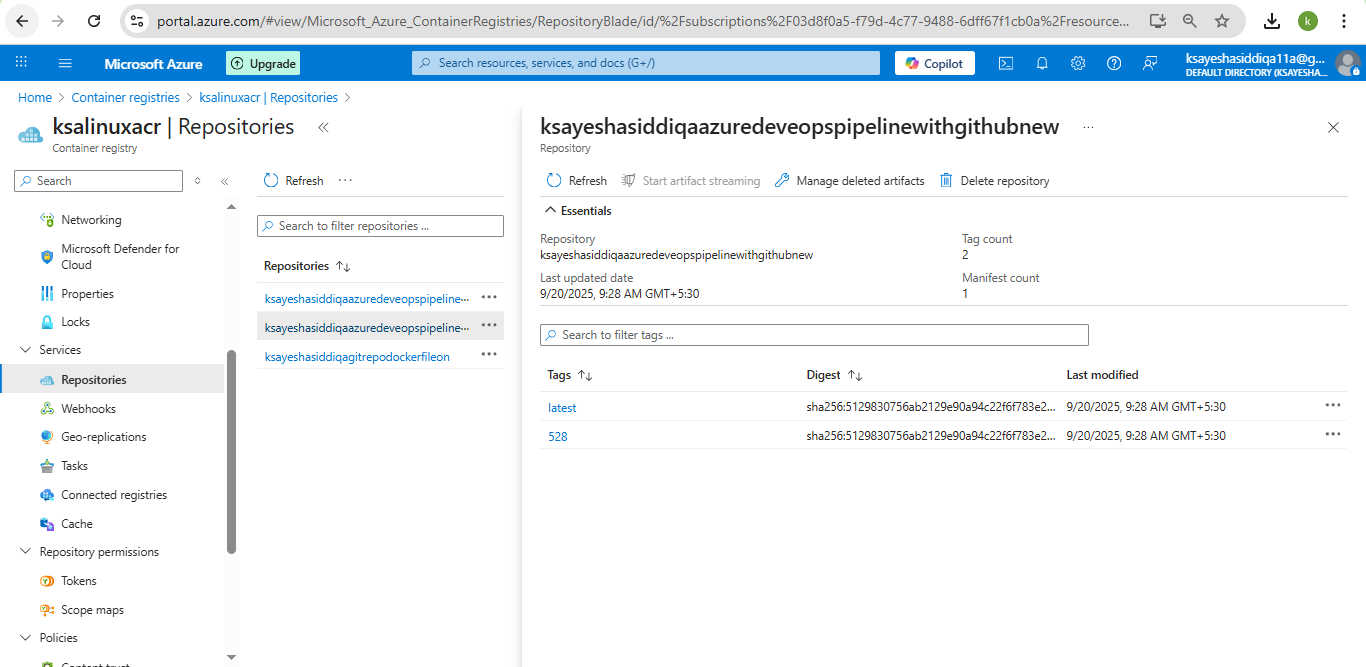
requests==2.31.0

pandas==2.2.1

sqlalchemy==2.0.29

psycopg2-binary==2.9.9 # if using PostgreSQL





5.

trigger:

- main

pool:

name: 'agent\_ayesha' # your agent pool

demands:

- agent.name -equals agent\_ayesha\_linux

variables:

dockerRegistryServiceConnection: 'azurecontainerregistrydocker' # Service connection name

imageName: 'containerregistryksa.azurecr.io/ksayeshasiddiqaazuredeveopspipelinewithgithubnew:latest' # Full ACR image path

stages:

- stage: RunAndVerify

jobs:

- job: RunImage

steps:

- task: Docker@2

displayName: 'Login to Azure Container Registry'

inputs:

command: login

containerRegistry: '$(dockerRegistryServiceConnection)'

- script: |

echo "Pulling image: $(imageName)"

docker pull $(imageName)

echo "Running image..."

docker run -d --name testcontainer $(imageName)

echo "Check container status"

docker ps

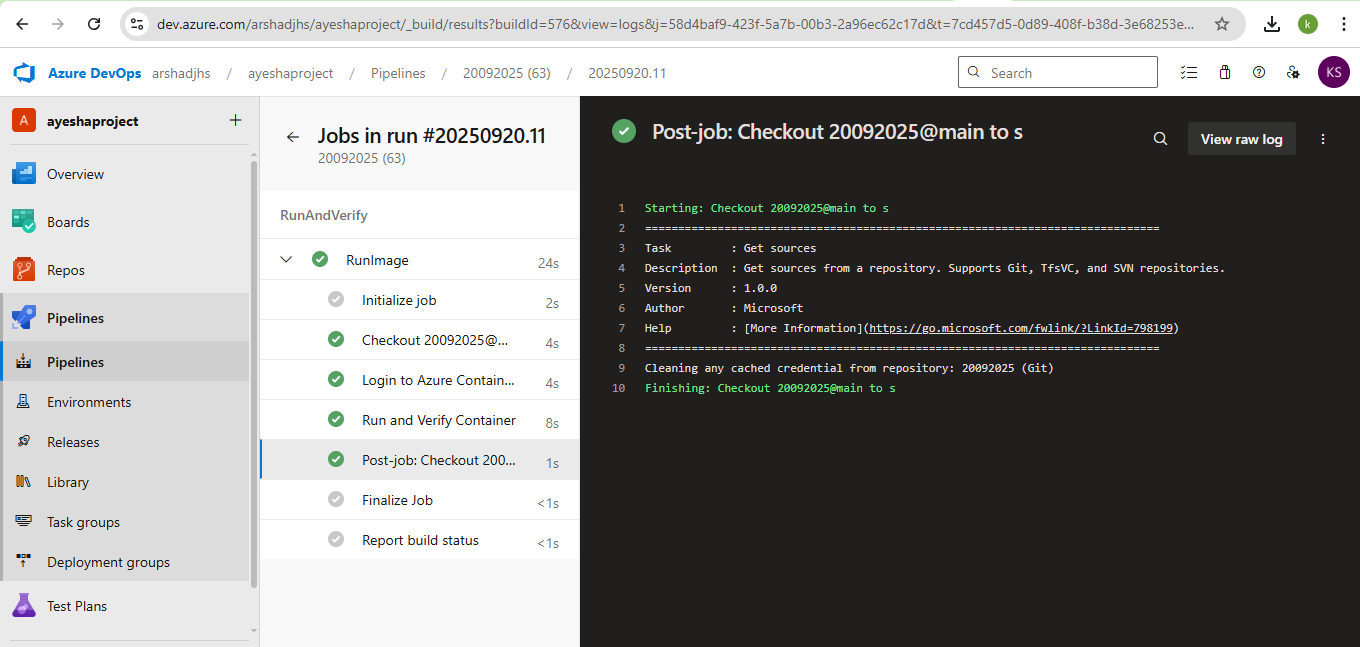
echo "Verify container by running a command inside"

docker exec testcontainer echo "Container is running successfully!"

echo "Cleaning up..."

docker rm -f testcontainer

displayName: 'Run and Verify Container'



6.

trigger:

- main

pool:

name: 'agent\_ayesha'

demands:

- agent.name -equals agent\_ayesha\_linux

variables:

mavenPomFile: 'pom.xml'

dockerRegistryServiceConnection: 'azurecontainerregistrydocker'

containerRegistry: 'containerregistryksa.azurecr.io'

imageRepository: 'ksayeshasiddiqaazuredeveopspipelinewithgithubnew'

imageTag: '$(Build.BuildId)'

fullImageName: '$(containerRegistry)/$(imageRepository):$(imageTag)'

steps:

# 1. Checkout code

- checkout: self

# 2. Build with Maven

- task: Maven@4

displayName: 'Build with Maven'

inputs:

mavenPomFile: $(mavenPomFile)

goals: 'clean package'

# 3. Build and push Docker image to ACR

- task: Docker@2

displayName: 'Build and Push to ACR'

inputs:

command: buildAndPush

repository: $(imageRepository)

dockerfile: '\*\*/Dockerfile'

containerRegistry: $(dockerRegistryServiceConnection)

tags: |

$(imageTag)

# 4. Login to ACR

- task: Docker@2

displayName: 'Login to ACR'

inputs:

command: login

containerRegistry: $(dockerRegistryServiceConnection)

# 5. Run and verify container

- script: |

echo "Pulling image: $(fullImageName)"

docker pull $(fullImageName)

echo "Running image..."

docker run -d --name testcontainer -p 5000:5000 $(fullImageName)

echo "Check container status"

docker ps

echo "Verify container response"

docker exec testcontainer echo "Container is running successfully!"

echo "Cleaning up..."

docker rm -f testcontainer

displayName: 'Run and Verify Container'

NOT GOT THE OUTPUT because the buld not working