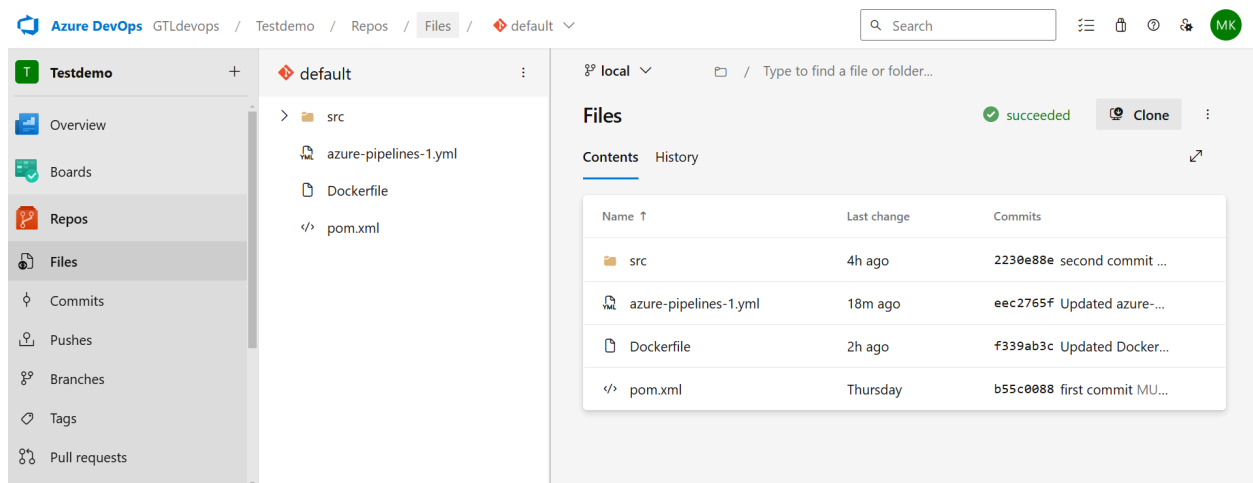


AZURE DEVOPS CI/CD BASICS

To deploy an application to a local machine using **environment resources** in Azure DevOps, need to configure the pipeline to target the machine with the Azure DevOps agent installed and use **environments** in pipeline for tracking deployments

Steps:

➤ Clone the repository and Push the source code into Azure repo



➤ Create pipeline Environment and add resource as VM and run the given agent Registration script in local machine

```
PS C:\Windows\system32> $ErrorActionPreference="Stop";If(-NOT ([Security.Principal.WindowsPrincipal][Security.Principal.WindowsIdentity]::GetCurrent()).IsInRole([Security.Principal.WindowsBuiltInRole] "Administrator")){ throw "Run command in an administrator PowerShell prompt";};If($PSVersionTable.PSVersion -lt (New-Object System.Version("3.0"))){ throw "The minimum version of Windows PowerShell that is required by the script (3.0) does not match the currently running version of Windows PowerShell." };If(-NOT (Test-Path $env:SystemDrive\azagent')){mkdir $env:SystemDrive\azagent'; cd $env:SystemDrive\azagent'; for($i=1; $i -lt 100; $i++){ $destFolder="A"+$i.ToString();if(-NOT (Test-Path ($destFolder))){mkdir $destFolder;cd $destFolder;break;}}; $agentZip="$PWD\agent.zip";$DefaultProxy=[System.Net.WebRequest]::DefaultWebProxy;$securityProtocol=@();$securityProtocol+=[Net.ServicePointManager]::SecurityProtocol;$securityProtocol+=[Net.SecurityProtocolType]::Tls12;[Net.ServicePointManager]::SecurityProtocol=$securityProtocol;$WebClient=New-Object Net.WebClient; $Uri='https://vstsagentpackage.azureedge.net/agent/3.246.0/vsts-agent-win-x64-3.246.0.zip';if($DefaultProxy -and (-not $DefaultProxy.IsBypassed($Uri))){$WebClient.Proxy= New-Object Net.WebProxy($DefaultProxy.GetProxy($Uri).OriginalString, $True);}; $WebClient.DownloadFile($Uri, $agentZip);Add-Type -AssemblyName System.IO.Compression.FileSystem;[System.IO.Compression.ZipFile]::ExtractToDirectory( $agentZip, "$PWD");.\config.cmd --environment --environmentname "localhost" --agentenv:COMPUTERNAME --runasservice --work '_work' --url 'https://dev.azure.com/GTLdevops/' --projectname 'Testdemo' --authPAT --token pdejumcu2ny7bnjk2lgniohgzx3f3xxwimhsiangqccposglazjkq; Remove-Item $agentZip;
```

Directory: C:\

Mode	LastWriteTime	Length	Name
d----	10/27/2024 1:55 PM		azagent

```

Mode                LastWriteTime         Length Name
-----
d-----         10/27/2024    1:55 PM             A1

Azure Pipelines
agent v3.246.0 (commit 59c8fe5)

>> Connect:

Connecting to server ...

>> Register Agent:

Scanning for tool capabilities.
Connecting to the server.
Enter Environment Virtual Machine resource tags? (Y/N) (press enter for N) >
Successfully added the agent
Testing agent connection.
2024-10-27 08:27:28Z: Settings Saved.
Enter enable SERVICE_SID_TYPE_UNRESTRICTED for agent service (Y/N) (press enter for N) > y
Enter User account to use for the service (press enter for NT AUTHORITY\SYSTEM) >

```

➤ Configure the Dockerfile and Pipeline for CI/CD

Dockerfile:

```

FROM maven:3.8.6-openjdk-11 AS build
WORKDIR /app
COPY pom.xml .
COPY src ./src
RUN mvn clean package -DskipTests

FROM tomcat:9-jdk11-openjdk-slim
RUN rm -rf /usr/local/tomcat/webapps/*
COPY --from=build /app/target/vprofile-v2.war /usr/local/tomcat/webapps/ROOT.war
EXPOSE 8080
CMD ["catalina.sh", "run"]

```

Pipeline:

```

trigger:
- main

pool:
  vmImage: 'ubuntu-latest'

variables:
  dockerHubUser: $(Docker_username)
  dockerHubPassword: $(Docker_pass)
  IMAGE_NAME: 'geojit'

```

```
stages:
- stage: Build
  displayName: 'Build and Push Docker Image'
  jobs:
  - job: Build
    displayName: 'Build Docker Image'
    steps:
    - task: Docker@2
      displayName: 'Build Docker Image'
      inputs:
        command: build
        dockerfile: 'Dockerfile'
        repository: safuvanh/$(IMAGE_NAME)
        tags: |
          latest

    - task: Docker@2
      displayName: Docker Login
      inputs:
        command: login
        containerRegistry: dockerhub
        username: $(dockerHubUser)
        password: $(dockerHubPassword)

    - task: Docker@2
      displayName: Push image to Docker Hub
      inputs:
        command: push
        repository: safuvanh/$(IMAGE_NAME)
        tags: |
          latest

- stage: Deployment
  displayName: Deploy App
  jobs:
  - deployment: deploymentJob
    displayName: Deploying
    environment:
      name: localhost
      resourceType: VirtualMachine
    strategy:
      runOnce:
        deploy:
          steps:
          - script: |
```

```
echo "Running Docker container.."
docker run -d -p 80:8080 safuvanh/$(IMAGE_NAME):latest
displayName: 'Run Docker Container'
```

Prerequisites for Pipeline: create service connection of docker registry (dockerhub)
For pushing image to registry and add dockerhub username and password as secret variable

Variables



Docker_pass



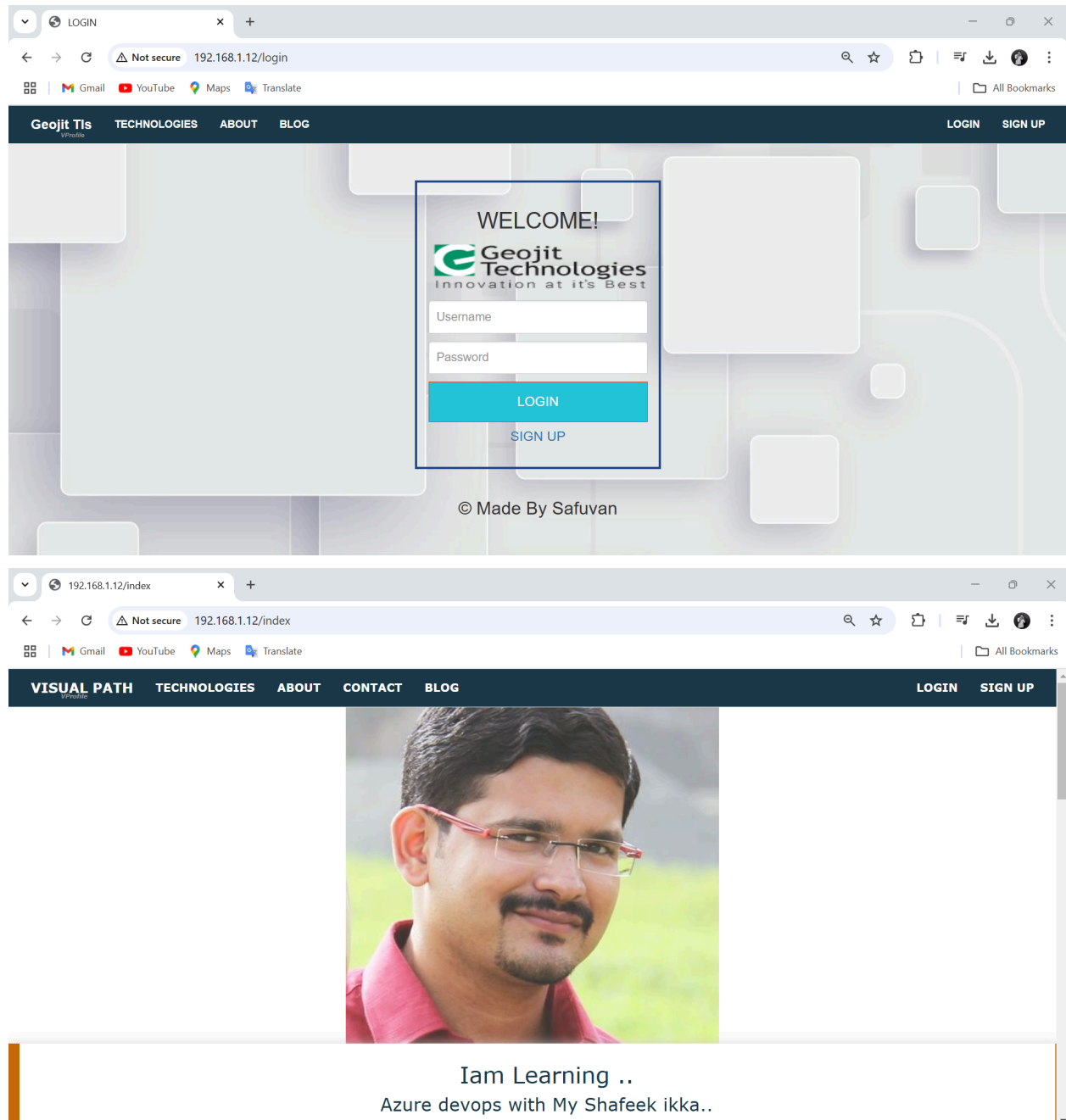
Docker_username

➤ Run the pipeline

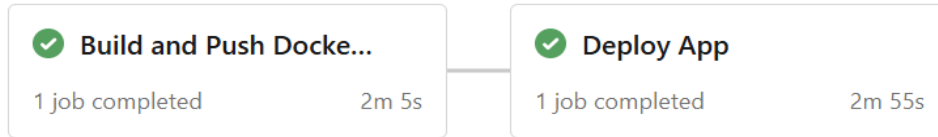
The screenshot shows the Azure DevOps interface for a pipeline named 'Build Docker Image'. The left sidebar shows the 'Testdemo' project with various views like Overview, Boards, Repos, Pipelines, Environments, Releases, Library, and Task groups. The main area displays the 'Jobs in run #202...' for the 'default (45)' pipeline. The job list includes: Initialize job (1s), Checkout defa... (1s), Build Doc... (1m 41s), Docker Login (<1s), Push image t... (18s), Post-job: Ch... (<1s), and Finalize Job (<1s). The 'Build Docker Image' job is currently running, and its console output is visible on the right, showing the Docker build process.

The screenshot shows the Azure DevOps interface for a pipeline named 'Deploying_Deploy_DESKTOP-26NTDJP'. The left sidebar shows the 'Testdemo' project with various views like Overview, Boards, Repos, Pipelines, Environments, Releases, Library, and Task groups. The main area displays the 'Jobs in run #20241...' for the 'default (45)' pipeline. The job list includes: Initialize job (<1s), Run Docker... (2m 53s), and Finalize Job (<1s). The 'Deploying_Deploy_DESKTOP-26NTDJP' job is currently running, and its console output is visible on the right, showing the deployment process.

➤ Access the localhost <ip address> through any browser



Stages Jobs





← localhost

Automatically created environment

Add resource

:

Deployments Resources Approvals and checks

Name	Latest job
 DESKTOP-26NTDJP	 #20241027.1 on default (45)

Source code :

Git repo : <https://github.com/safuvanh/Geojit.git>

Written By Muhammed Safuvan k

safuvaaan@gmail.com

Linkedin: <https://www.linkedin.com/in/safuvanh/>