Day 4 Continuous Integration

Azure Boards:

Mainly used for planning. Similar to KanBan boards.

Azure Repos :

Used as central repository for code management. Similar to GitHub but is private to Azure Devops account.

Day4 and 5 Class

- AZURE PIPELINES
  - CI/CD Pipeline in Azure Devops
  - Hosted Agents vs Self Hosted Agents
  - YAML
  - Demo CI + CD [Basic .NET Application]
- CI/CD : Continuous Integration + Continuous Development/Deployment

Challenge - Faster integration of new feature to application
Manage consistent changes in their code

AZURE PIPELINES

- Create a balance between developer and IT operations.
- DEVOPS PIPELINE - Continuous Chain of building, testing and deploying code to an environment.
  - Azure Pipeline falls under Devops Pipeline.

Benefits of Azure Pipelines

- Automation
- Languages : Python, C#, Java, PHP,Ruby
- Functionality - CI + CD
- Deployment : Deploy it to multiple env.

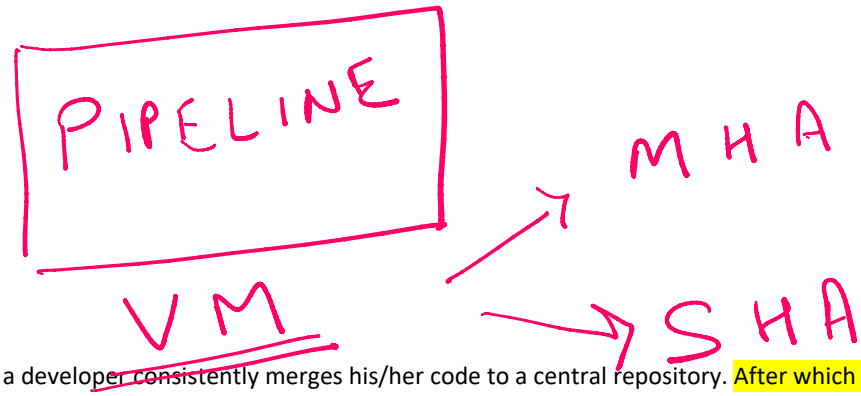
Hosted vs Private Agents

Microsoft Hosted Agents

- Pre defined images by MS.
- According to the YAML code, your image is automatically selected.
- You do not have tp do any kind of maintenance tasks.
- Upgrades are automatic of these images.
- You use these images to run your pipeline. AFTER a job is run the vm created out of these images is DISCARDED.

Self Hosted Agents [Private Agents]

- You create your own agents with your own choice of OS.
- You are responsible for installing the softwares in this image.
- You maintain the image from upgrades + security.



Continuous Integration

- This a devops practice where a developer consistently merges his/her code to a central repository. After which AUTOMATED builds and tests are RUN
- You can publich the code faster.

DEMO PART 1- Create a App in Local Env

- Use Visual Studio 2022.
- Create a Basic .NET Application.
- BUILD and then DEPLOY the app locally.
- VSCODE -> Used the existing solutions offered by VS Code to build a solution on .NET -> Gave the name for my app and selected the right framework [.Net 8.0] -> I had a solution file

DEMO PART 2 - Push the Code to Azure Repos

- Next we are going to integrate Azure repos as my remote repository and push our code to Azure Repos.

DEMO PART 3 - Start The CI in Azure Pipeline

- Selected the right repo + right project.
- Azure Pipeline created a YAML for us.
- YAML file is created and version controlled with your code.

```yaml
trigger:
- master


pool:
  vmImage: 'windows-latest'

variables:
  solution: '**/*.sln'
  buildPlatform: 'Any CPU'
  buildConfiguration: 'Release'


steps:
- task: NuGetToolInstaller@1

- task: NuGetCommand@2
  inputs:
    restoreSolution: '$(solution)'


- task: VSBuild@1
  inputs:
    solution: '$(solution)'
    msbuildArgs: '/p:DeployOnBuild=true /p:WebPublishMethod=Package /p:PackageAsSingleFile=true /p:SkipInvalidConfigurations=true /p:DesktopBuildPackageL
ocation="$(build.artifactStagingDirectory)\WebApp.zip" /p:DeployIisAppPath="Default Web Site"'
    platform: '$(buildPlatform)'
    configuration: '$(buildConfiguration)'
```