

Question 1:

The build should trigger as soon as anyone in the dev team checks in code to master branch.1.

Answer:

- This can be done by **enabling the Continuous Integration Trigger** in the trigger section of the Build Pipeline
- As soon as the dev person commits any changes in the code in the "master" branch, the build gets triggered automatically by generating the artifacts

Question 2:

There will be test projects which will create and maintained in the solution along the Web and API. The trigger should build all the 3 projects - Web, API and test.The build should not be successful if any test fails.

Answer :

testResultsFormat: VSTest

searchFolder: '\$(Agent.TempDirectory)/TestResults'

testResultsFiles: '**/build_*.trx'

failTaskOnFailedTests: true

Question 3:

The deployment of code and artifacts should be automated to Dev environment.

Answer:

- The code can be automatically deployed by **enabling the continuous deployment trigger**.

```
stage: 'Development' #Stage name cannot have spaces
displayName: 'Development' #Name displayed when viewing in Azure DevOps
dependsOn: ['Build_Stage'] #List of previous stages that must complete before this stage runs
jobs:
- deployment: Deploy_Development #No spaces
  displayName: Development Deployment #Name displayed in UI
  pool:
    vmImage: $(AgentImage) #variable that was defined previously (not shown in this snippet)
  environment: Deployment
  strategy:
    runOnce:
      deploy:
        steps:
        - download: current
          artifact: 'app'
        - task: ExtractFiles@1
          displayName: 'Extract files'
          inputs:
            archiveFilePatterns: "$(Pipeline.Workspace)/app/application.zip"
            destinationFolder: "$(Pipeline.Workspace)/files"
        - task: AzureRmWebAppDeployment@4
          displayName: 'Azure App Service Deploy'
          inputs:
            ConnectionType: 'AzureRM' #This is the default Value
            WebAppKind: 'webApp' #This is the default value
            azureSubscription: 'PipelineApp Azure Connection' #Name of the Service Connection previously
            created
            WebAppName: 'Development-MWKS-PipelineApp' #Name of the App Service to be deployed to
            Package: '$(Pipeline.Workspace)/files/BasicCoreApp' #Application files to be deployed
            enableCustomDeployment: true #These two properties override the default of having
            DeploymentType: 'webDeploy' #the pipeline deciding for us what it thinks the best deployment option
            should be
```

Question 4:

Upon successful deployment to the Dev environment, deployment should be easily promoted to QA and Prod through automated process.

```
stage: 'UAT' #Stage name cannot have spaces
displayName: 'UAT' #Name displayed when viewing in Azure DevOps
dependsOn: ['Build_Stage', 'Development'] #List of previous stages that must complete before this stage runs
jobs:
- deployment: Deploy_UAT #No spaces
  displayName: UAT Deployment #Name displayed in UI
  pool:
    vmImage: $(AgentImage) #variable that was defined previously (not shown in this snippet)
  environment: UAT
  strategy:
    runOnce:
      deploy:
        steps:
        - download: current
          artifact: 'app'
        - task: ExtractFiles@1
          displayName: 'Extract files'
          inputs:
            archiveFilePatterns: "$(Pipeline.Workspace)/app/application.zip"
            destinationFolder: "$(Pipeline.Workspace)/files"
        - task: AzureRmWebAppDeployment@4
          displayName: 'Azure App Service Deploy'
          inputs:
            ConnectionType: 'AzureRM' #This is the default Value
            WebAppKind: 'webApp' #This is the default value
            azureSubscription: 'PipelineApp Azure Connection' #Name of the Service Connection previously
            created
            WebAppName: 'UAT-MWKS-PipelineApp' #Name of the App Service to be deployed to
            Package: "$(Pipeline.Workspace)/files/BasicCoreApp' #Application files to be deployed
            enableCustomDeployment: true #These two properties override the default of having
            DeploymentType: 'webDeploy' #the pipeline deciding for us what it thinks the best deployment option
            should be

- stage: 'Production' #Stage name cannot have spaces
  displayName: 'Production' #Name displayed when viewing in Azure DevOps
  dependsOn: ['Build_Stage', 'UAT'] #List of previous stages that must complete before this stage runs
  jobs:
  - deployment: Deploy_Production #No spaces
    displayName: Production Deployment #Name displayed in UI
    pool:
      vmImage: $(AgentImage) #variable that was defined previously (not shown in this snippet)
    environment: Production
    strategy:
      runOnce:
        deploy:
          steps:
          - download: current
            artifact: 'app'
          - task: ExtractFiles@1
            displayName: 'Extract files'
            inputs:
```

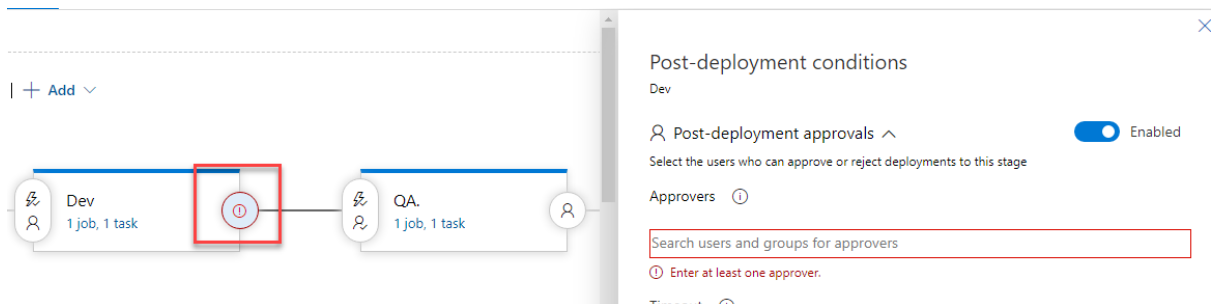
```
archiveFilePatterns: "$(Pipeline.Workspace)/app/application.zip"
destinationFolder: "$(Pipeline.Workspace)/files"
- task: AzureRmWebAppDeployment@4
  displayName: 'Azure App Service Deploy'
  inputs:
    ConnectionType: 'AzureRM' #This is the default Value
    WebAppKind: 'webApp' #This is the default value
    azureSubscription: 'PipelineApp Azure Connection' #Name of the Service Connection previously
created
    WebAppName: 'prod-MWKS-PipelineApp' #Name of the App Service to be deployed to
    Package: '$(Pipeline.Workspace)/files/BasicCoreApp' #Application files to be deployed
    enableCustomDeployment: true #These two properties override the default of having
    DeploymentType: 'webDeploy' #the pipeline deciding for us what it thinks the best deployment option
should be
```

Question 5

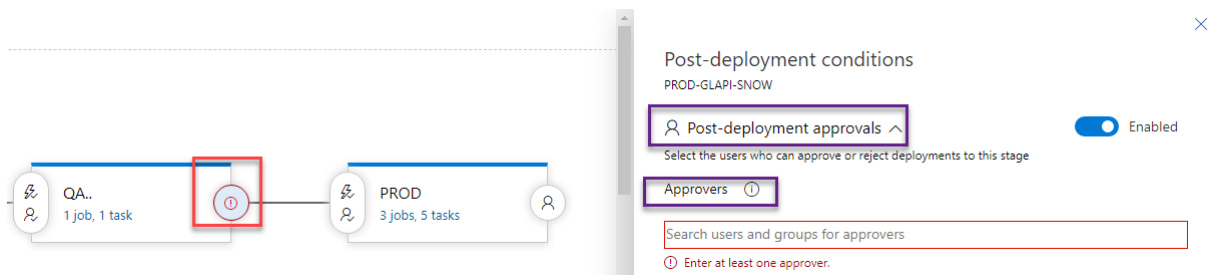
The deployments to QA and Prod should be enabled with Approvals from approvers only.

Answer:

- For deployments to be enabled by using Approver's Approval in QA stage, we need to add Approver's name to the **Post deployment conditions** by *enabling it in the DEV stage*.



- Similarly, we need to do the same with QA stage. In the post deployment conditions of the QA stage, we need to add the Approver's name for the PROD stage deployment. So that PROD deployment will be enabled once the assigned approver has approved the request for the deployment.



- Whenever we make any changes to our pipeline, we should always save the changes to reflect the saved changes.

