**Question 1 : Scenario**

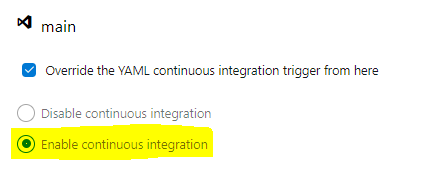
1. The build should trigger as soon as anyone in the dev team checks in code to master branch.
2. There will be test projects which will create and maintained in the solution along the Web and API. The trigger should build all the 3 projects - Web, API and test.
3. The build should not be successful if any test fails.
4. The deployment of code and artifacts should be automated to Dev environment.
5. Upon successful deployment to the Dev environment, deployment should be easily promoted to QA and Prod through automated process.
6. The deployments to QA and Prod should be enabled with Approvals from approvers only.

Explain how each of the above the requirements will be met using Azure DevOps configuration.

Explain the steps with configuration details.

**Solution :**

1. “Enable Continuous Integration” for MASTER branch.



1. Define 3 stages in the build, once the trigger happens, it will run all the build

**stages:**

**- stage: Web**

**displayName: "Run Web"**

**jobs:**

**- template: YourYamlFile.yml**

**- stage: API**

**displayName: "Run API"**

**dependsOn: []**

**jobs:**

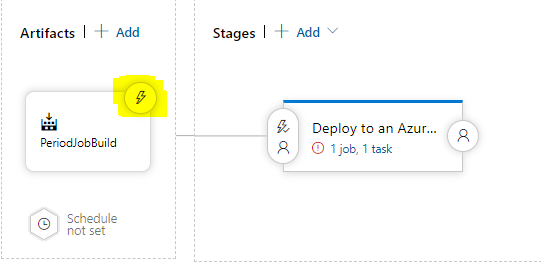
**- template: YourYamlFile.yml**

**- stage: Tests**

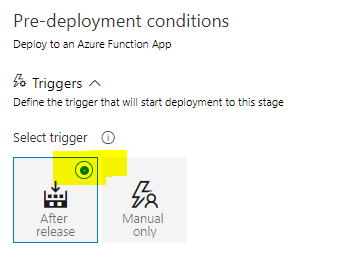
**displayName: "Run test cases"**

**dependsOn: []**

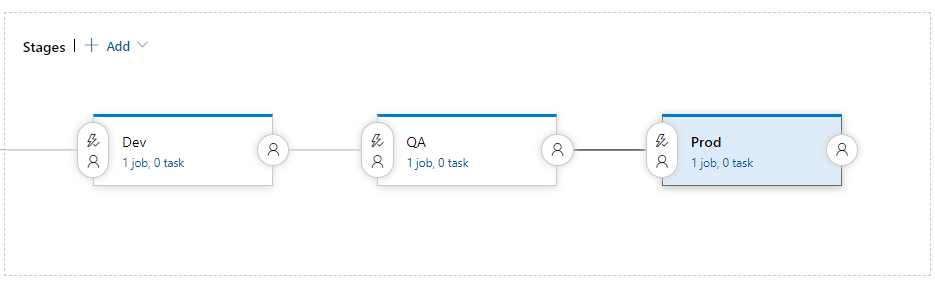
1. The failOnStderr: true can be used in the YAML, this will fail the pipeline if there is any failure in any of the test cases.
2. Release pipeline – Enabled the highlighted Continuous Deployment trigger.



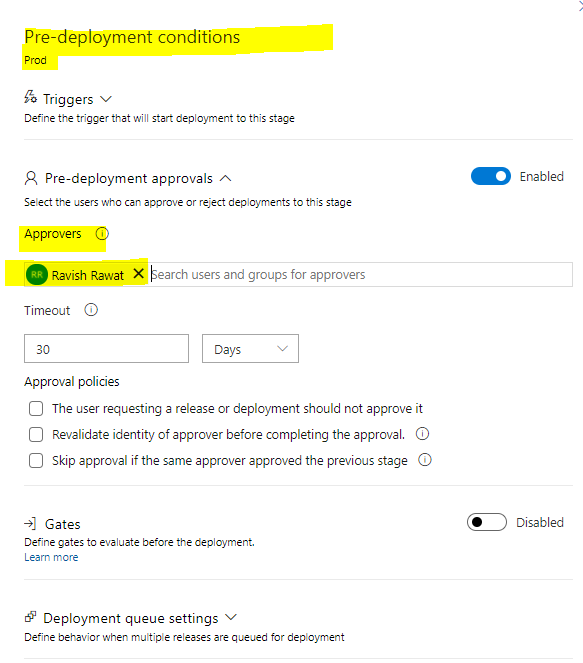
The Dev stage should be set to after release automated to Dev environment

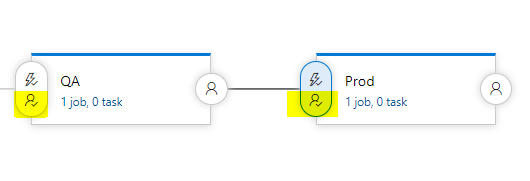


1. Once dev is successful QA and Prod will be deployed automatically, as it is set after Dev stage



1. Add preapproval to QA and Prod environments and after that the tick is visible on both the stages as highlighted in the screenshot below.





**Question 2 : Scenario**

Macro Life, a healthcare company has recently setup the entire Network and Infrastructure on Azure.

The infrastructure has different components such as Virtual N/W, Subnets, NIC, IPs, NSG etc.

The IT team currently has developed PowerShell scripts to deploy each component where all the properties of each resource is set using PowerShell commands.

The business has realized that the PowerShell scripts are growing over period of time and difficult to handover when new admin onboards in the IT.

The IT team has now decided to move to Terraform based deployment of all resources to Azure.

All the passwords are stored in a Azure Service known as key Vault. The deployments needs to be automated using Azure DevOps using IaC(Infrastructure as Code).

1) What are different artifacts you need to create - name of the artifacts and its purpose

2) List the tools you will to create and store the Terraform templates.

3) Explain the process and steps to create automated deployment pipeline.

4) Create a sample Terraform template you will use to deploy Below services:

Vnet

2 Subnet

NSG to open port 80 and 443

1 Window VM in each subnet

1 Storage account

5) Explain how you will access the password stored in Key Vault and use it as Admin Password in the VM Terraform template.

**Solution :**

1. The artefacts that we need to create are as follows:
2. An artefact for all the TerraForm files – It will act as an IAAC to create infrastructure.
3. The tools we need to create are as follows:
4. Terraform installed in your workstation/build pipeline.
5. Visual Studio Code or any other editor.
6. YAML file plugin to ease the issues in the script file, like indentations, etc.
7. A rough model or a diagram to refer.
8. Terragrunt installed in build pipeline.
9. The steps are as follows
10. Build Pipeline – Steps are
11. Checks out code from the repo (the TF files).
12. Creates an artefact for these files.
13. Makes it ready for release pipeline to be picked.
14. Release pipeline – Steps are
15. It will have the artefact from Build pipeline.
16. Artefact will have Continuous Deployment enabled.
17. Different stages will be set to “After Release” to automate the deployment to all the stages.
18. Sample TerraForm file – Attached with the solution.zip
19. You can replace the admin password by replace token task and using the library group where Key-Vault is getting called.