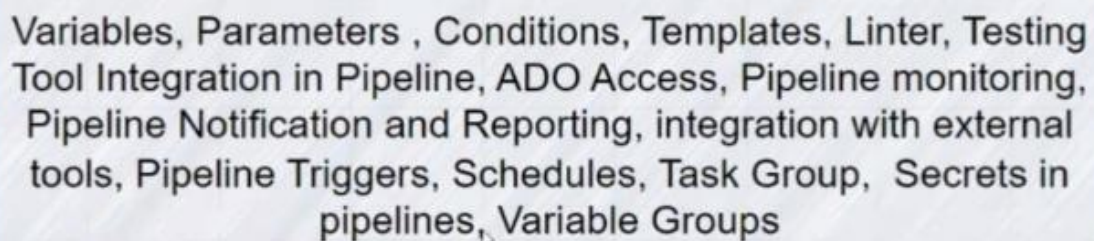


27 July 2024

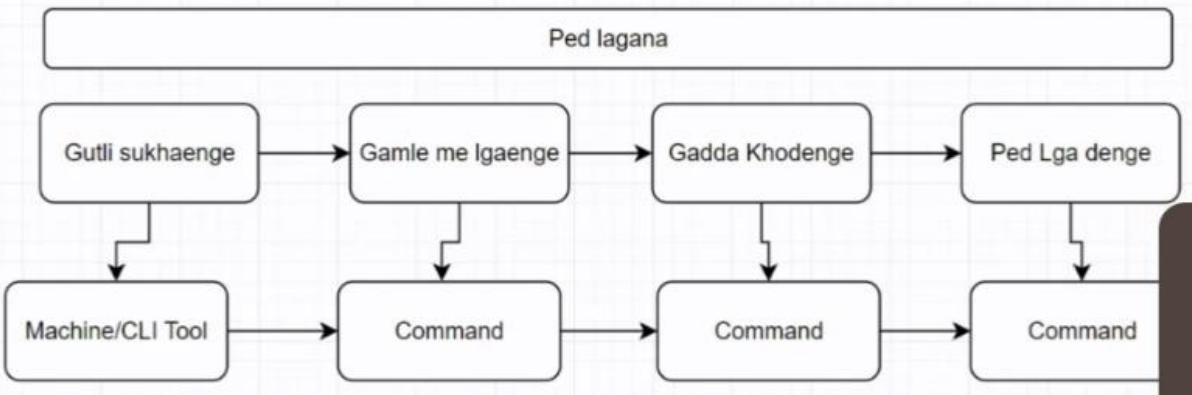
- 1) Suppose we got a ticket on jira to create a Vm or rg.
- 2) So we will create feature branch by doing git clone
- 3) After making feature branch, we will add the code into it.
- 4) In feature, we have done –
 - i) git add .
 - ii) git commit -m "code added"
 - iii) git push
- 5) In feature it will test and then we will raise PR
- 6) After raising PR (pull requests), it will firstly check in pipeline i.e. tfsec, does all scanning. After all scanning it will show plan and if this much is successful then only PR will go further else PR will fail
- 7) tfsec scans vulnerability, secrets and it will break if any problem will be found and again we have to fix it then only we will be able to move ahead
- 8) PR is staging pipeline
- 9) If PR will be fine then we will approve and it will merge
- 10) For dev we will run the pipeline then at manual validation we will give time then lead will approve and then apply will be done



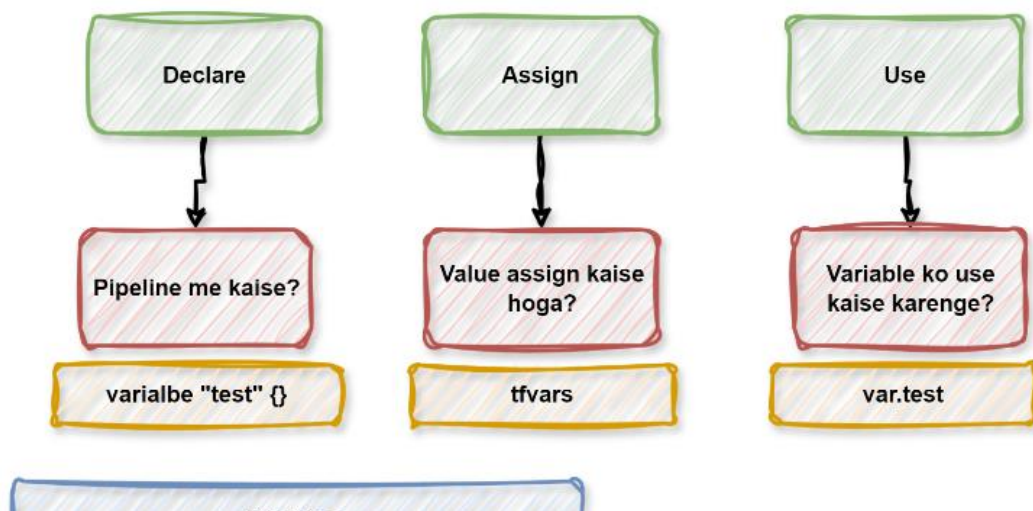
Variables, Parameters , Conditions, Templates, Linter, Testing
Tool Integration in Pipeline, ADO Access, Pipeline monitoring,
Pipeline Notification and Reporting, integration with external
tools, Pipeline Triggers, Schedules, Task Group, Secrets in
pipelines, Variable Groups

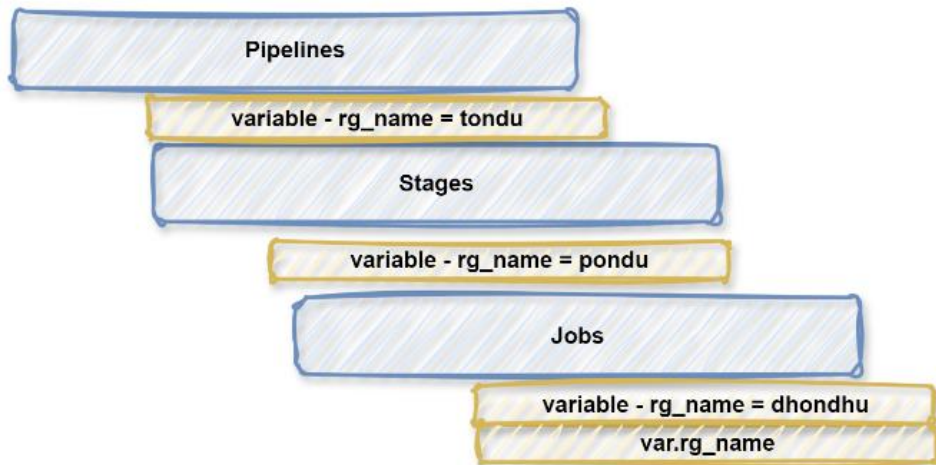
- 11) How will you access ADO?
- 12) How will you use variable group?
- 13) How will you set ability test?

Koi bhi chiz ko automate karne ke lie ek tool hai..



Variables





```

stages:
- stage: initandPlan
  variables: # stage-level
    STAGE_VAR: 'that happened'
  displayName: initandPlan
  jobs:

```

```

- stage: apply
  variables: # stage-level
    STAGE_VAR: 'that happened'
  displayName: apply
  jobs:

```

```

pool: Agentpool12

variables: # stage-level
  STAGE_VAR: 'that happened'

stages:

```

```

variables: # stage-level
  WORKDIR_VAR: '$(System.DefaultWorkingDirectory)/Environment/dev'

stages:

```

\$ means value of

```

workingDirectory: $(WORKDIR_VAR)
backendServiceArm: 'TestSC12'

```

Ctrl+f

Replace 1 by 1

+++++

AGENDA – Asking user on which environment pipeline should run

1) Setting parameter

```
pool: Agentpool12
parameters: [ parameter ]
stages:
```

2)

```
pool: Agentpool12
parameters:
  - name: string # Required as first property.
    displayName: string # Human-readable name for the parameter.
    type: string
    default: string | parameters | [ parameters ]
    values: [ string ]
  stages:
    - stage: initandPlan
```

```
pool: Agentpool12
parameters:
  - name: Environmentkaun # Required as first property.
    displayName: Environmentkaun # Human-readable name for the parameter.
    type: string
    default: dev
    values:
      - dev
      - qa
  stages:
    - stage: initandPlan
```

3) Now if we save and run, then we get the environments to select that in which env we want to run

/ LandingZone12 / Pipelines

← Pipeline12

main
Pipeline12 / azure-pipelines.yml *

```

4  --- main
5  - paths:
6  - include:
7  - /Environment/dev
8
9  pool: Agentpool12
10
11 parameters:
12 - name: Environmentkaun # Required as first property.
13 - displayName: Environmentkaun # Human-readable name for the parameter.
14 - type: string
15 - default: dev
16 - values:
17 - dev
18 - qa
19
20 stages:
21 - stage: initandPlan
22 - displayName: initandPlan
23 - jobs:
24 - job: initandPlan
25 - displayName: initandPlan
26 - steps:
27 - task: TerraformInstaller@1
28 - inputs:

```

Run pipeline
Select parameters below and manually run the pipeline

Branch/tag
main
Select the branch, commit, or tag

Environmentkaun
☒ dev
☐ qa

Advanced options

Variables
This pipeline has no defined variables

Stages to run
Run as configured

Resources
Use latest version of all resources

☐ Enable system diagnostics

Cancel Run

4)

```

trigger: none

jobs:
- job: stepList
  steps: ${ parameters.mySteplist }
- job: myStep
  steps:
    - ${ parameters.myStep }

```

```

variables: --- # stage-level
- WORKDIR_VAR: '$(System.DefaultWorkingDirectory)/Environment/${ parameters.myStep }'
- SERVICECONNECTION_VAR: 'TestSC12'

parameters:
- name: Environmentkaun # Required as first property.
- displayName: Environmentkaun # Human-readable name for the parameter.
- type: string
- default: dev
- values:
- dev
- qa

variables: --- # stage-level
- WORKDIR_VAR: '$(System.DefaultWorkingDirectory)/Environment/${ parameters.Environmentkaun }'
- SERVICECONNECTION_VAR: 'TestSC12'

stages:

```

5) So now if we run pipeline by selecting dev env so it is successful but by selecting qa it not successful because there is no code in qa env in terraform in qa folder (guess not sure will check later)

6)

