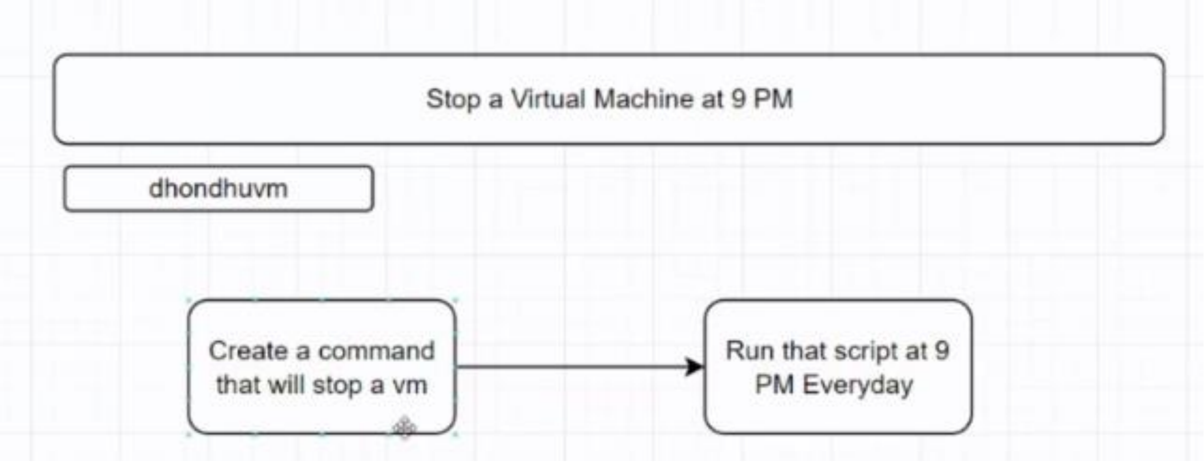
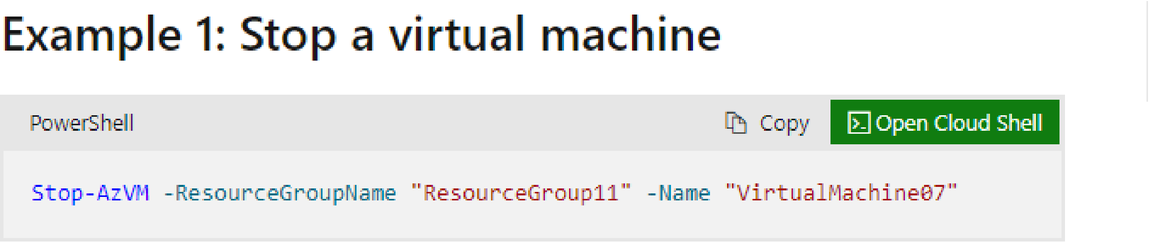
28 July 2024


Stop a Virtual Machine at 9 PM
dhondhuvm
Create a command that will stop a vm
Run that script at 9 PM Everyday

1) SEARCH – stop an azure vm using a command

https://learn.microsoft.com/en-us/powershell/module/az.compute/stop-azvm?view=azps-12.3.0



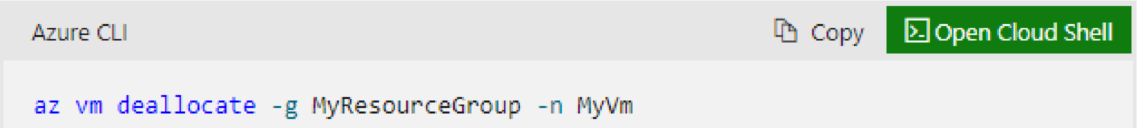
Example 1: Stop a virtual machine

```
Stop-AzVM -ResourceGroupName "ResourceGroup11" -Name "VirtualMachine07"
```

Stop-AzVM -ResourceGroupName "ResourceGroup11" -Name "VirtualMachine07"

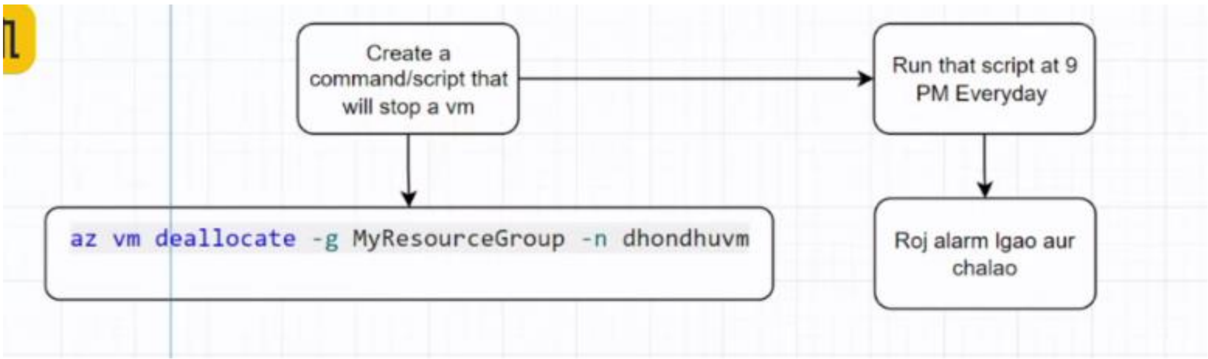
stop azure vm using az cli

https://learn.microsoft.com/en-us/cli/azure/vm?view=azure-cli-latest

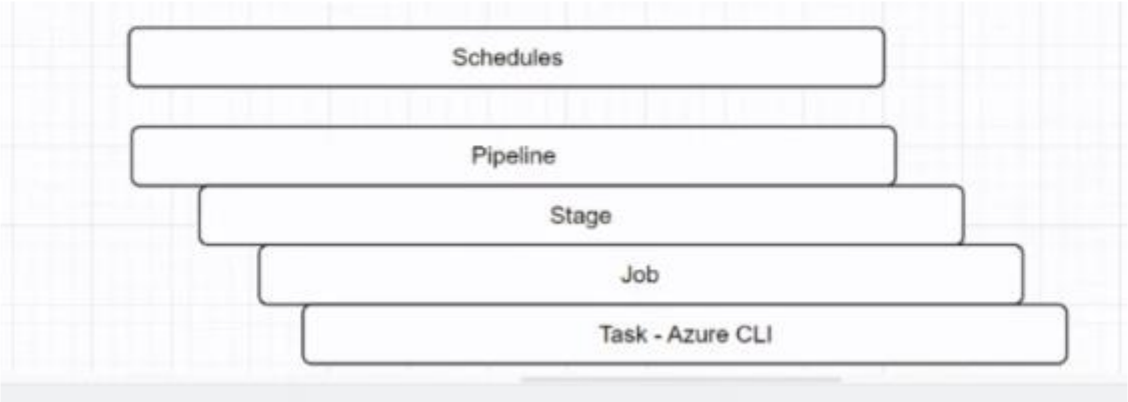


```
az vm deallocate -g MyResourceGroup -n MyVm
```

az vm deallocate -g MyResourceGroup -n MyVm


Create a command/script that will stop a vm
Run that script at 9 PM Everyday
Roj alarm lgao aur chalao

```
az vm deallocate -g MyResourceGroup -n dhondhuvm
```

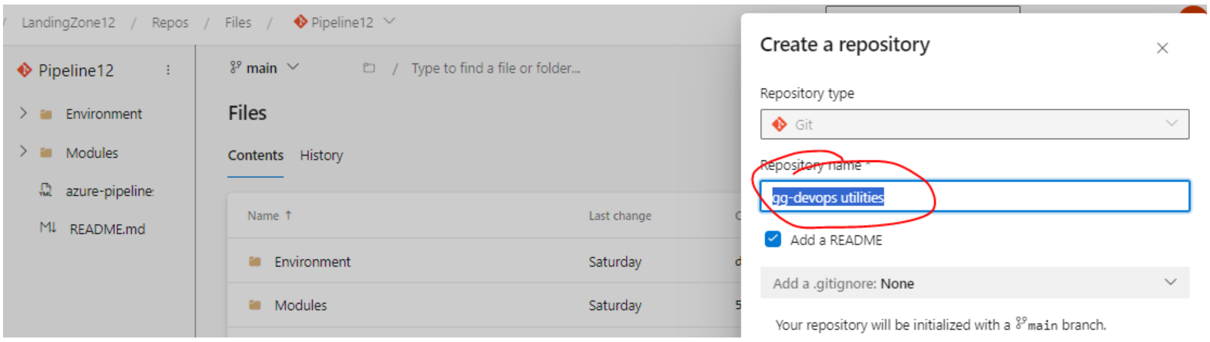


++

AGENDA – Custom pipeline in Azure which has only command but no code is there

1) We will select new pipeline and select new repo where yaml file of custom pipeline will be stored.

2) In company we will make new custom repo



Create a repository

Repository type: Git

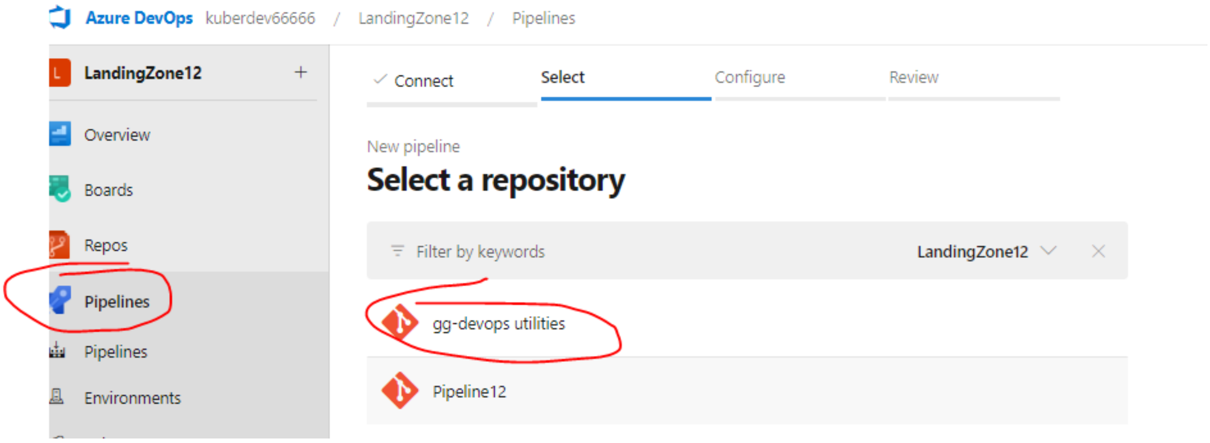
Repository name *: gg-devops utilities

☑ Add a README

Add a .gitignore: None

Your repository will be initialized with a main branch.

3)



New pipeline

Select a repository

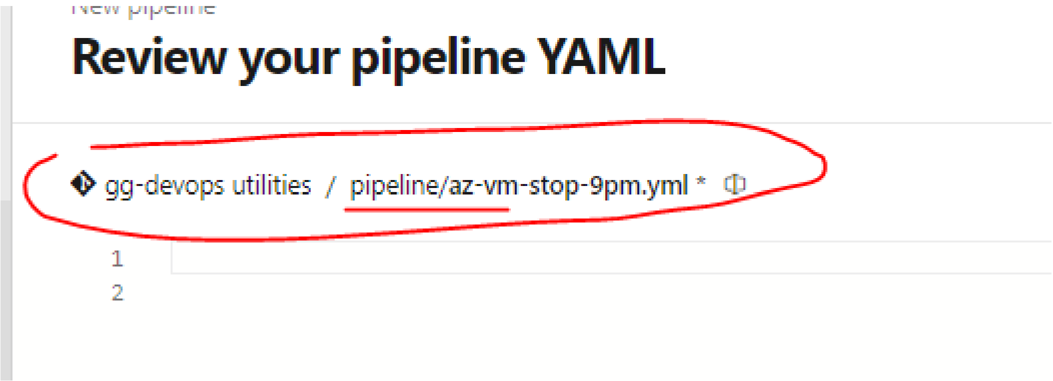
gg-devops utilities

Pipeline12

4) search – mono repo

https://monorepo.tools/

5) Now creating pipeline folder in our repo

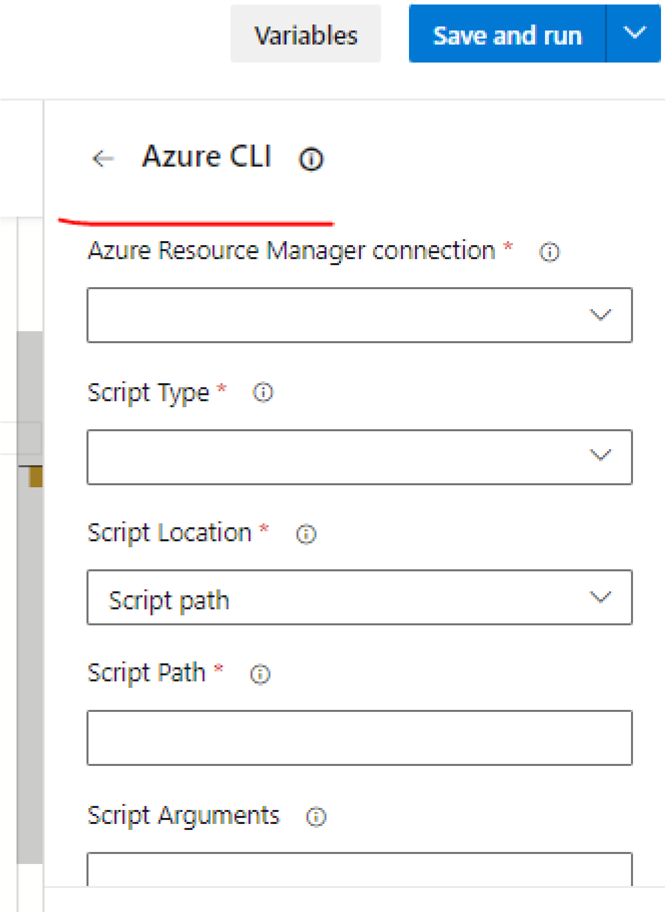


New pipeline

Review your pipeline YAML

gg-devops utilities / pipeline/az-vm-stop-9pm.yml *

6)



Azure CLI

Azure Resource Manager connection *

Script Type *

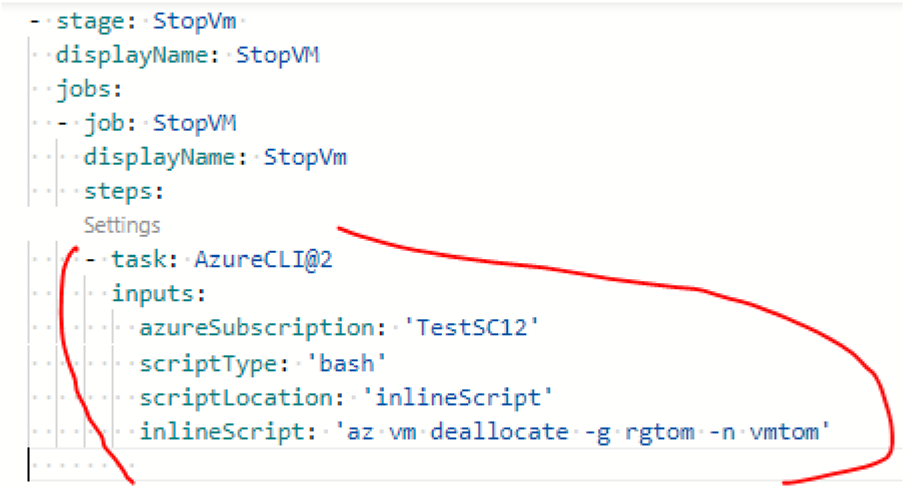
Script Location *

Script path

Script Path *

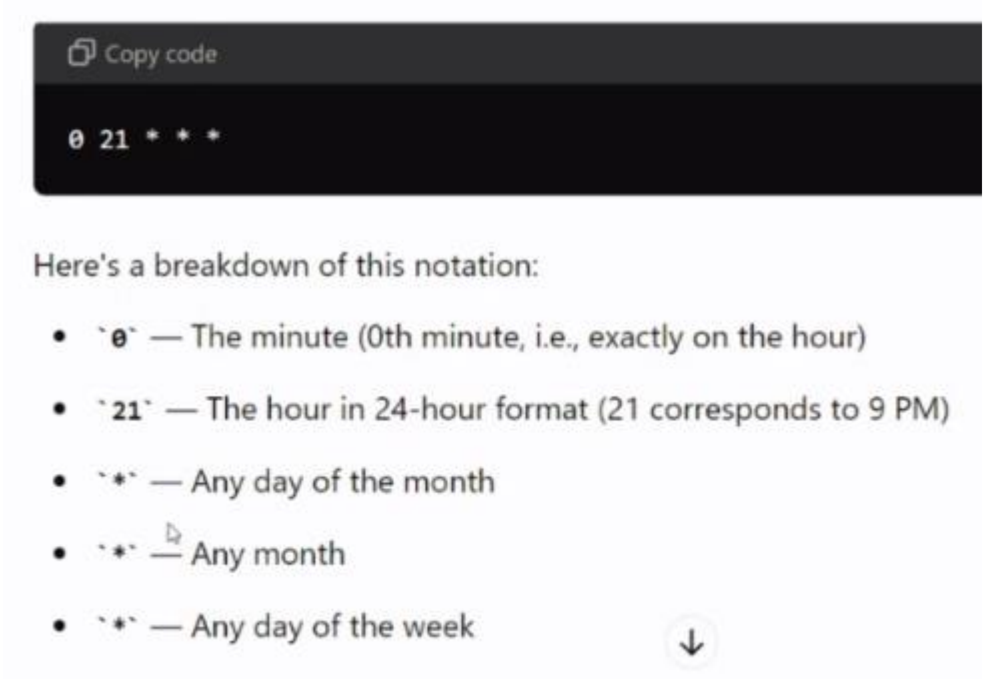
Script Arguments

7) **az vm deallocate -g rgtom -n vmtom**



```yaml
- stage: StopVm
  displayName: StopVM
  jobs:
  - job: StopVM
    displayName: StopVm
    steps:
    - task: AzureCLI@2
      inputs:
        azureSubscription: 'TestSC12'
        scriptType: 'bash'
        scriptLocation: 'inlineScript'
        inlineScript: 'az vm deallocate -g rgtom -n vmtom'
```

AGENDA – SET SCHEDULES



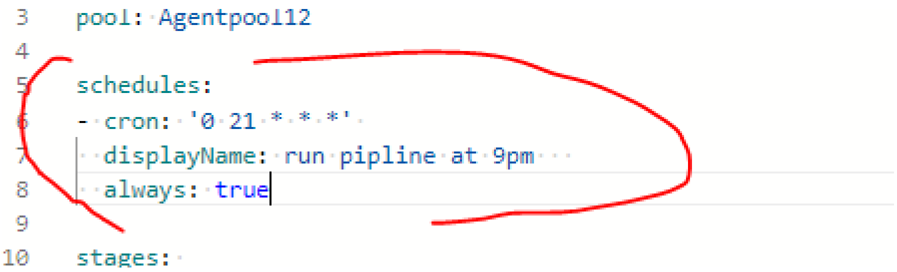
```
0 21 * * *
```

Here's a breakdown of this notation:

- `0` — The minute (0th minute, i.e., exactly on the hour)
- `21` — The hour in 24-hour format (21 corresponds to 9 PM)
- `*` — Any day of the month
- `*` — Any month
- `*` — Any day of the week

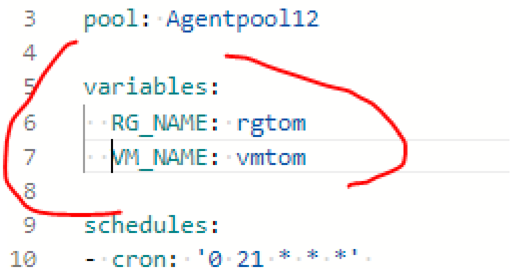
8) SEARCH – crontab.guru.com

9)

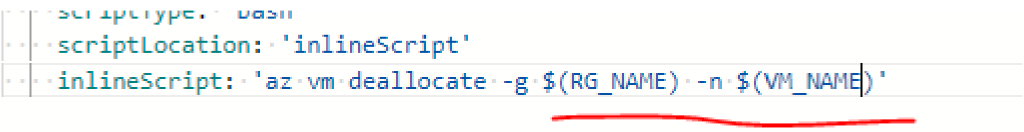


```yaml
pool: Agentpool12

schedules:
- cron: '0 21 * * *'
  displayName: run pipline at 9pm
  always: true

stages:
```

10) Setting variables for rg and vm

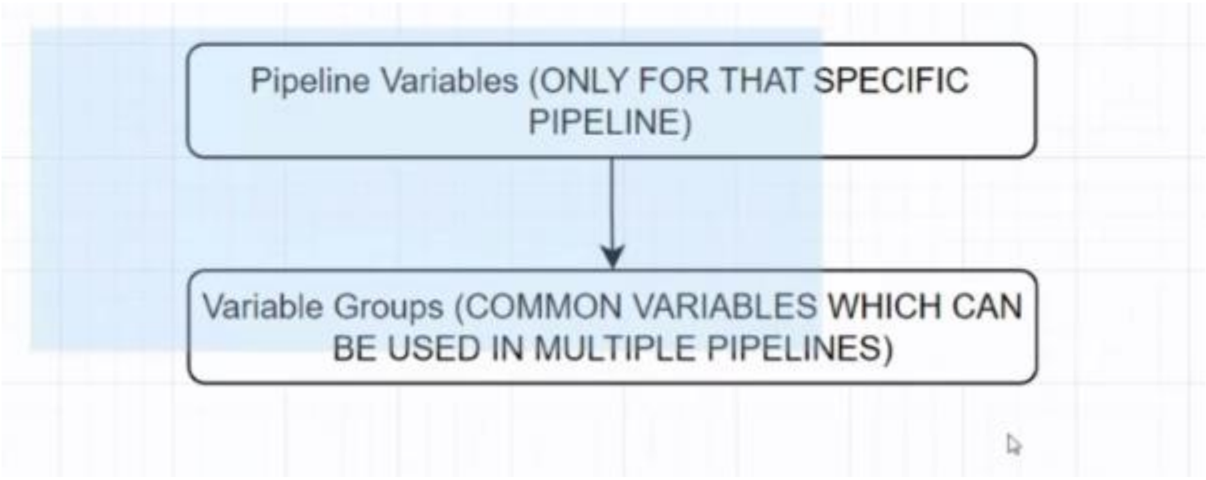


```
pool: Agentpool12

variables:
  RG_NAME: rgtom
  VM_NAME: vmtom

schedules:
- cron: '0 21 * * *'
```



```
    scriptType: 'bash'
    scriptLocation: 'inlineScript'
    inlineScript: 'az vm deallocate -g $(RG_NAME) -n $(VM_NAME)'
```

AGENDA – VARIABLE GROUPS



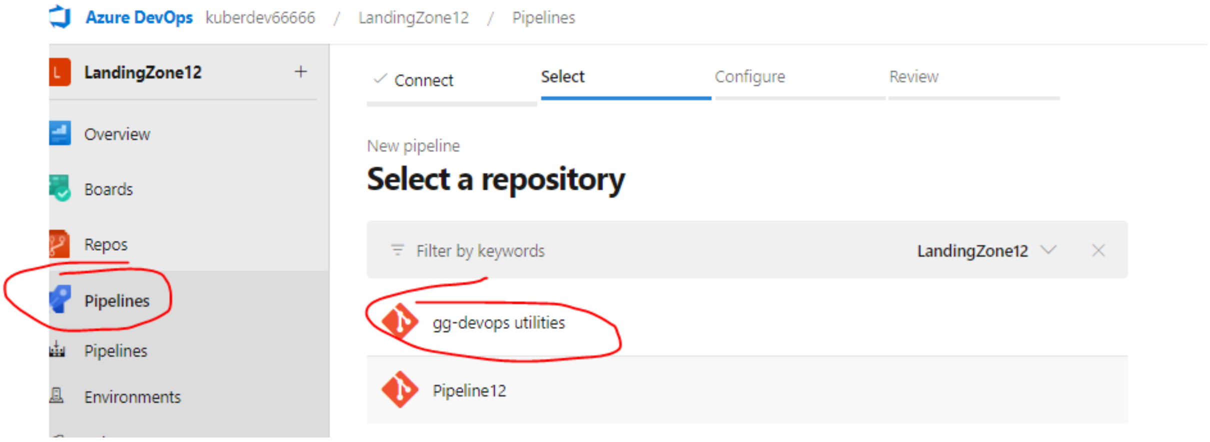
Pipeline Variables (ONLY FOR THAT SPECIFIC PIPELINE)

Variable Groups (COMMON VARIABLES WHICH CAN BE USED IN MULTIPLE PIPELINES)

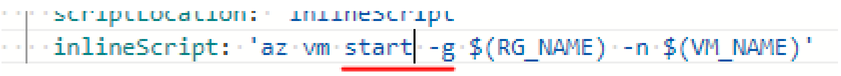
In a CI/CD pipeline, variables.group defines a collection of related variables that can be reused across multiple pipelines or stages. This approach promotes better organization, reduces duplication, and simplifies maintenance by allowing you to update variables in one central location. It can also enhance security by managing sensitive information effectively

AGENDA – CREATING ANOTHER PIPELINE TO START THE VM

1) So, make new pipeline using same repo “gg waali ”

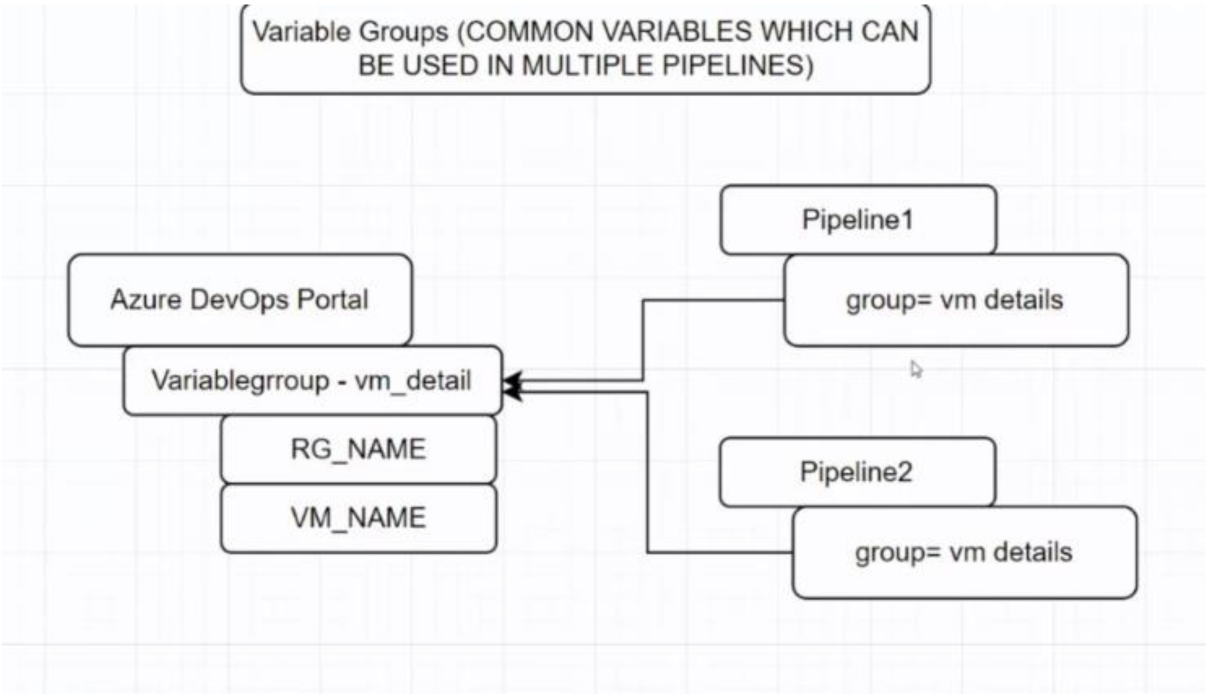


2) Copy the same code and instead of deallocate we will put start in code



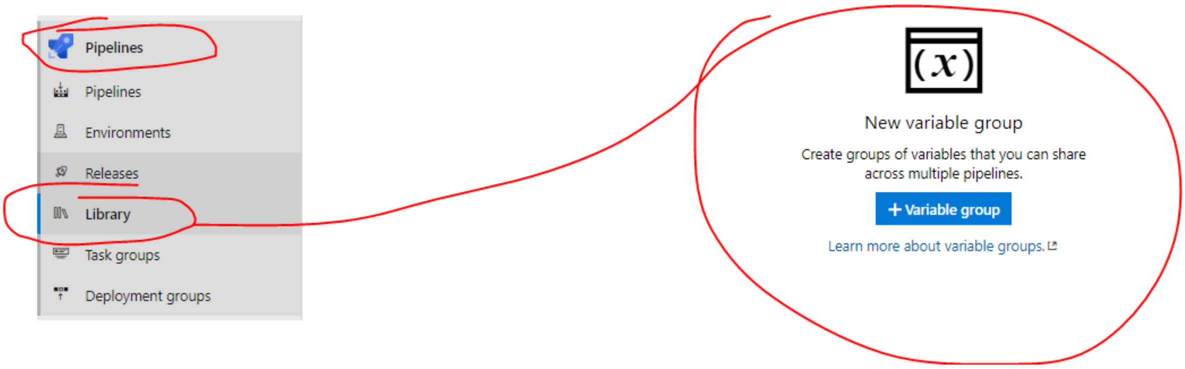
```
    scriptLocation: 'inlineScript'
    inlineScript: 'az vm start -g $(RG_NAME) -n $(VM_NAME)'
```

3)


Variable Groups (COMMON VARIABLES WHICH CAN BE USED IN MULTIPLE PIPELINES)
Pipeline1
group= vm details
Azure DevOps Portal
Variablegrroup - vm_detail
RG_NAME
VM_NAME
Pipeline2
group= vm details

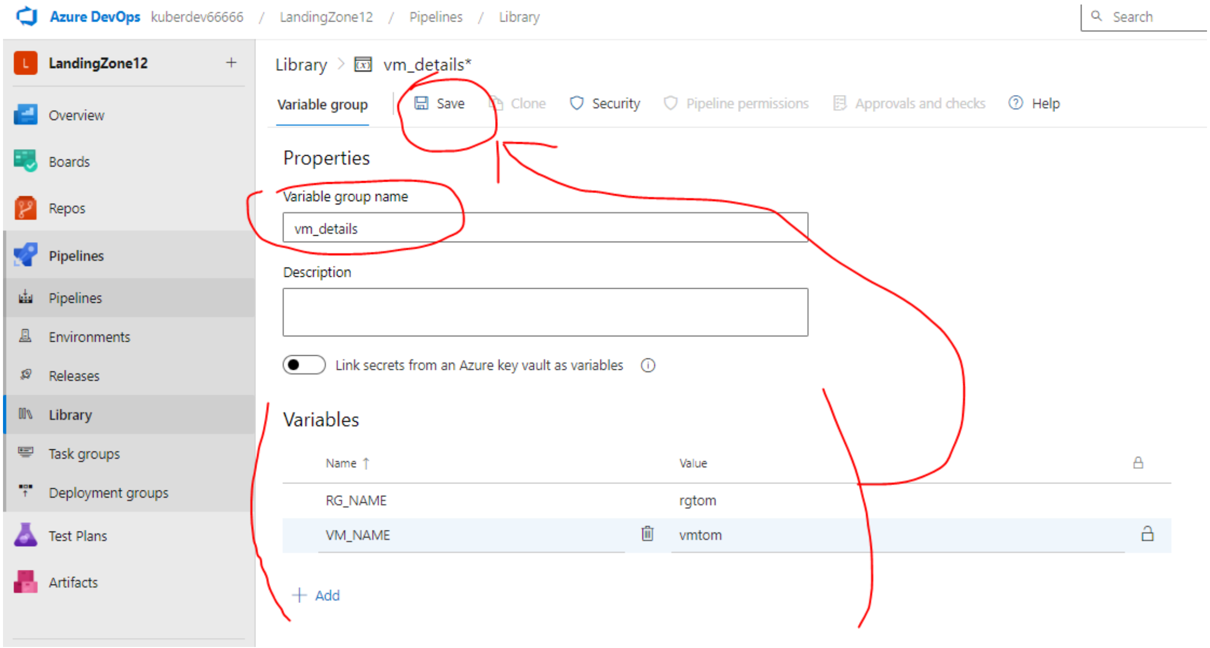
4) SEARCH - https://learn.microsoft.com/en-us/azure/devops/pipelines/library/variable-groups?view=azure-devops&tabs=azure-pipelines-ui%2Cyaml

5) Creating variable group



New variable group

Create groups of variables that you can share across multiple pipelines.



Properties

Variable group name: vm_details

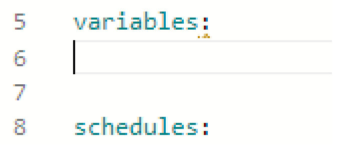
Description

○ Link secrets from an Azure key vault as variables

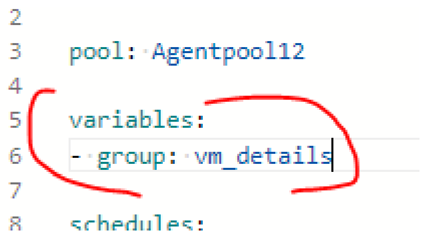
Variables

Name ↑	Value
RG_NAME	rgtom
VM_NAME	vmtom

6) Now open stop and start pipelines in 2 tabs and remove set variables from both pipelines and use variable groups

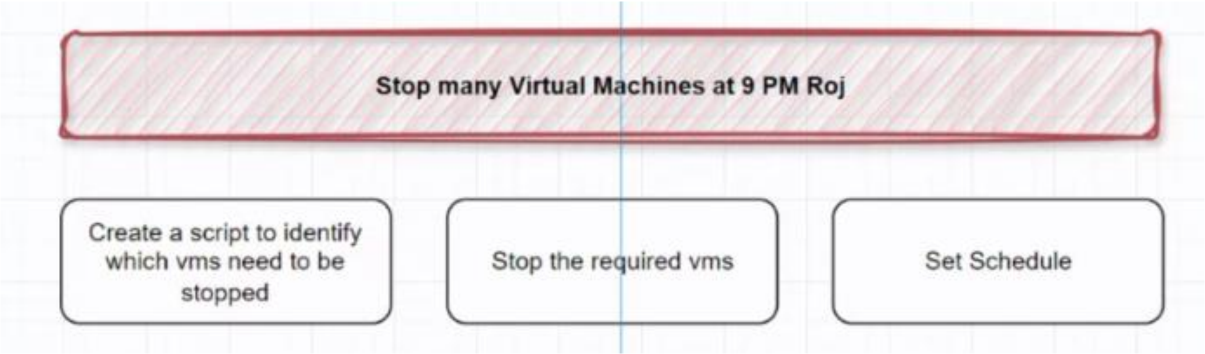


```
variables:


schedules:
```



```

pool: Agentpool12

variables:
- group: vm_details

schedules:
```

***) Suppose if we have to stop many Vms**



Stop many Virtual Machines at 9 PM Roj

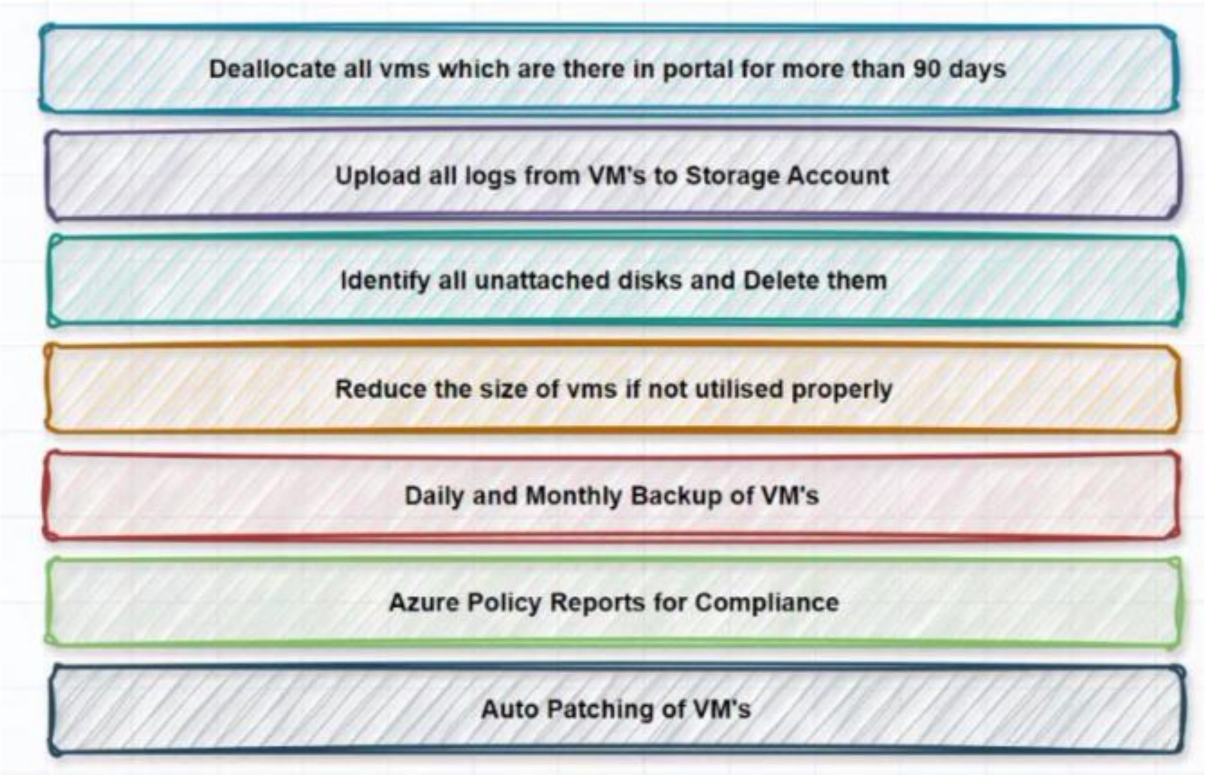
Create a script to identify which vms need to be stopped

Stop the required vms

Set Schedule

1) We can use tags also that has user keyword

2) Different scenarios for automating using power shell scripts



Deallocate all vms which are there in portal for more than 90 days

Upload all logs from VM's to Storage Account

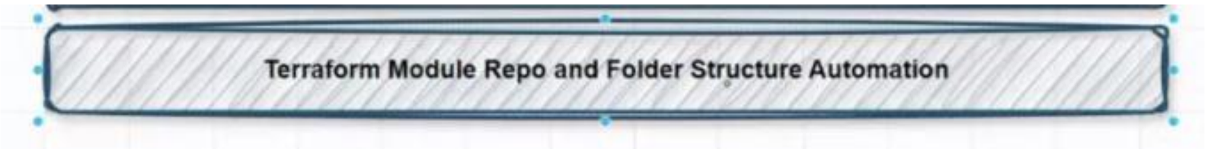
Identify all unattached disks and Delete them

Reduce the size of vms if not utilised properly

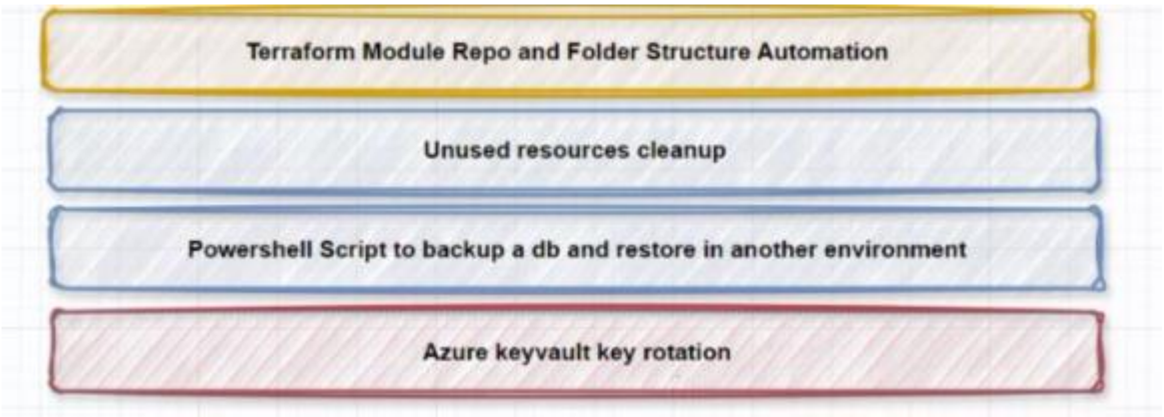
Daily and Monthly Backup of VM's

Azure Policy Reports for Compliance

Auto Patching of VM's



Terraform Module Repo and Folder Structure Automation



Terraform Module Repo and Folder Structure Automation

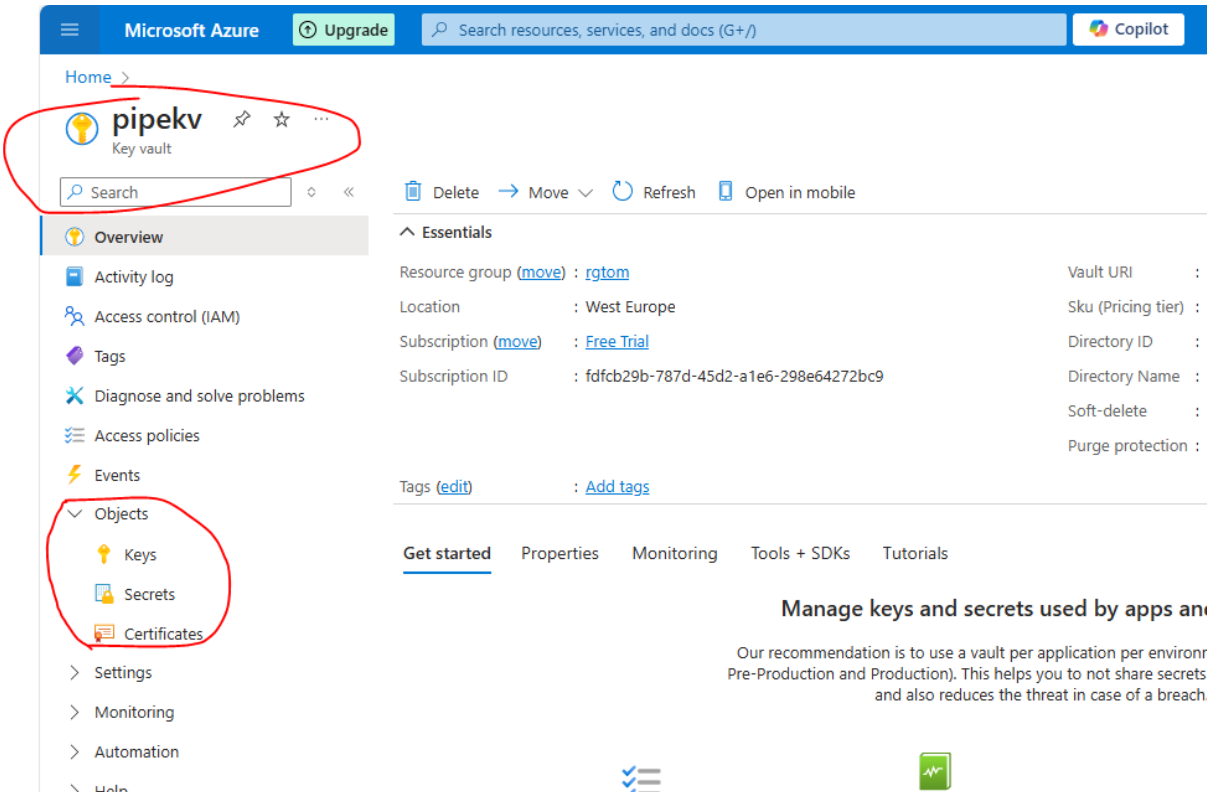
Unused resources cleanup

Powershell Script to backup a db and restore in another environment

Azure keyvault key rotation

++

7) Creating key vault – leave this one we will apply later.



pipekv

Key vault

Essentials

Resource group (move) : rgtom

Location : West Europe

Subscription (move) : Free Trial

Subscription ID : fdfcb29b-787d-45d2-a1e6-298e64272bc9

Tags (edit) : Add tags

Vault URI :

Sku (Pricing tier) :

Directory ID :

Directory Name :

Soft-delete :

Purge protection :

Manage keys and secrets used by apps an

Our recommendation is to use a vault per application per environr
Pre-Production and Production). This helps you to not share secrets
and also reduces the threat in case of a breach

++

AGENDA – PIPELINE MONITORING

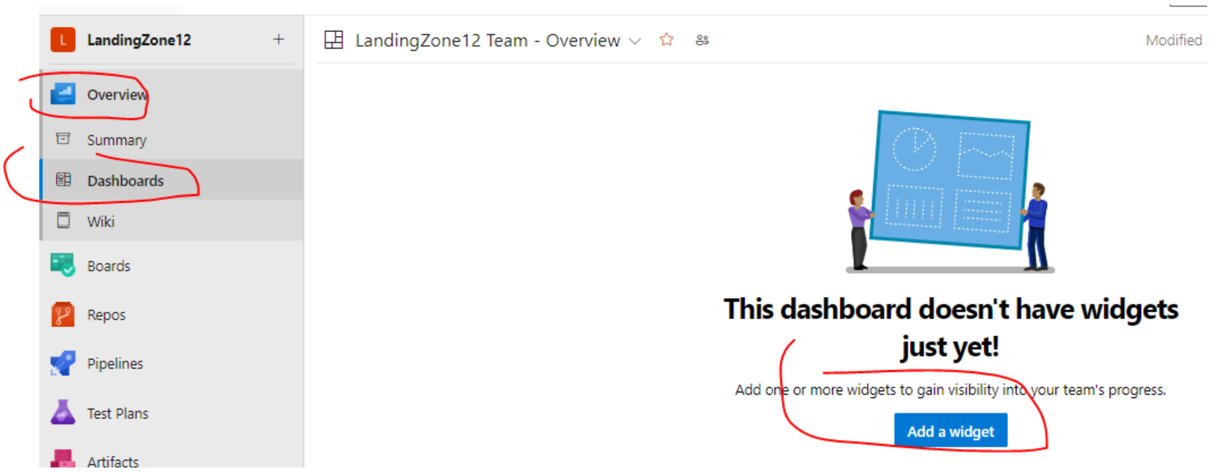
1) Monitoring methods are

i) Built in monitoring features

ii) Dashboards

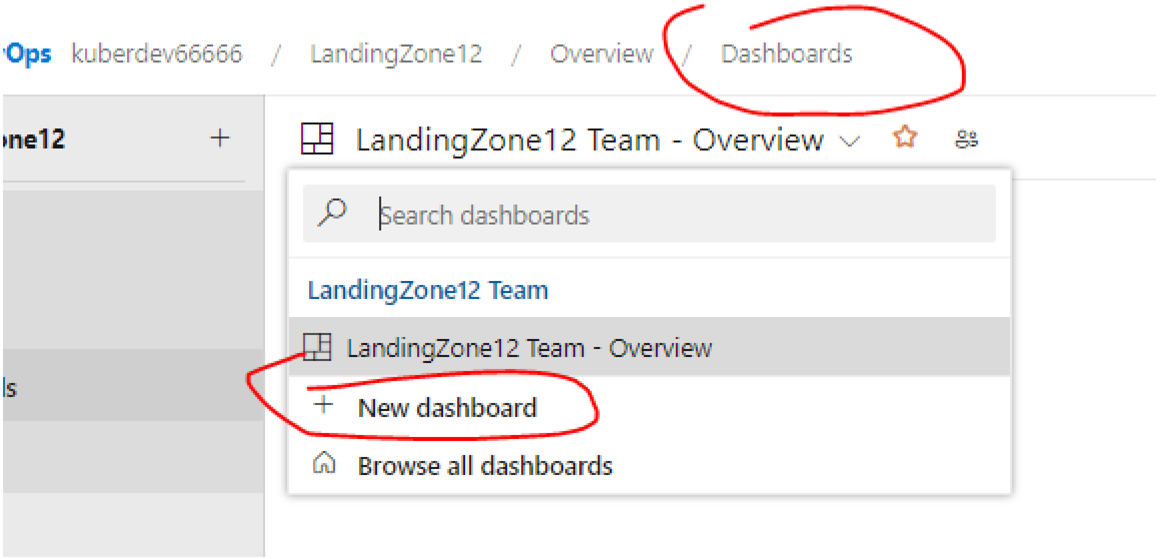
2) Go to overview -> Dashboards

3) Set script for pipeline alert mail goes on DL, on failing, using SMPT

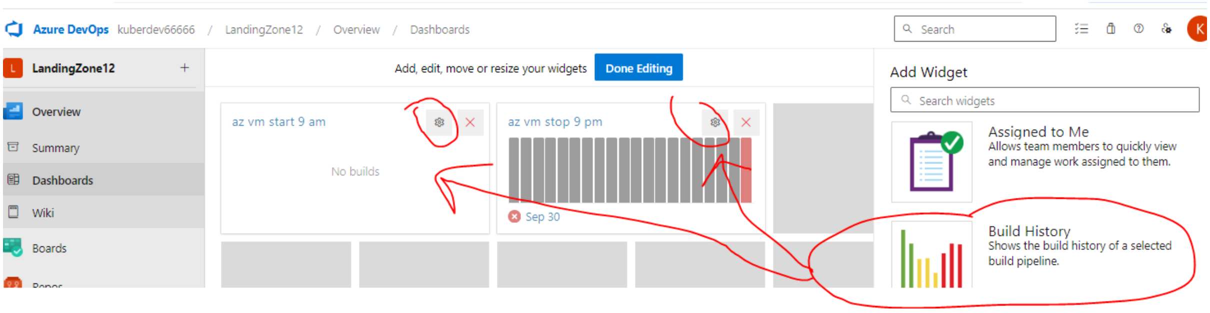


This dashboard doesn't have widgets just yet!

Add one or more widgets to gain visibility into your team's progress.


Ops kuberdev66666 / LandingZone12 / Overview / Dashboards
ne12
LandingZone12 Team - Overview
Search dashboards
LandingZone12 Team
LandingZone12 Team - Overview
New dashboard
Browse all dashboards

Rename it then – add widgets -> Build history -> add -> configure settings button -> Select pipeline -> save



Add, edit, move or resize your widgets

Assigned to Me
Allows team members to quickly view and manage work assigned to them.

Build History
Shows the build history of a selected build pipeline.