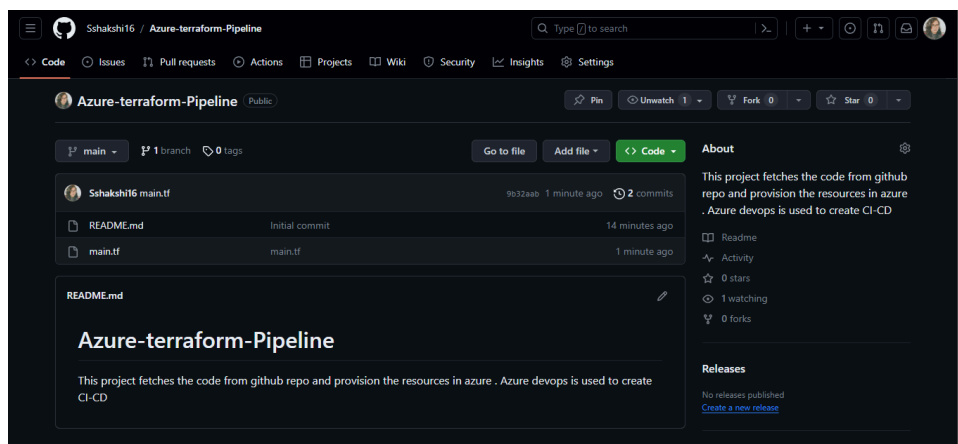


GitHub Repository

Create a new repository

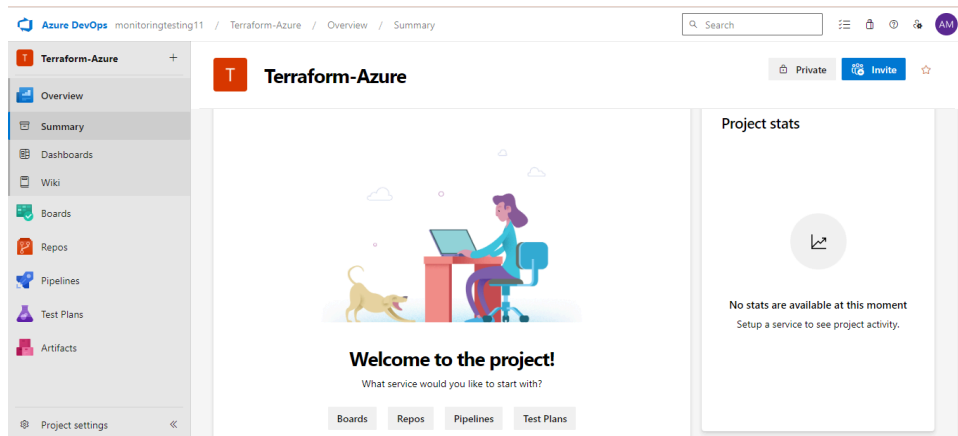
Add your main.tf file there.



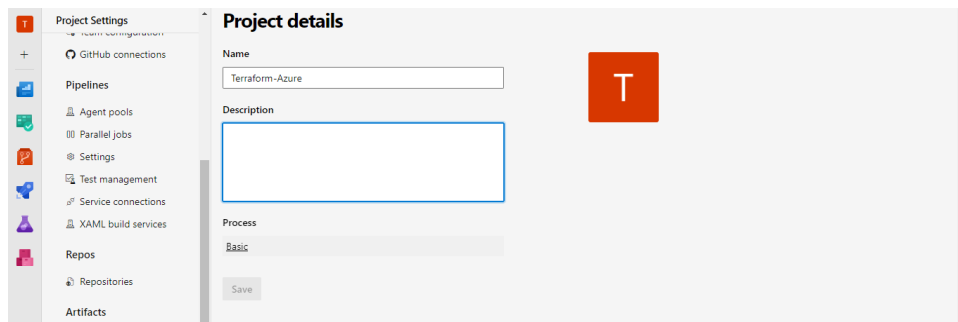
Azure DevOps Pipeline

Firstly, we need to create a Service Principal Name (SPN) to allow our Azure DevOps Organization project to deploy our environment.

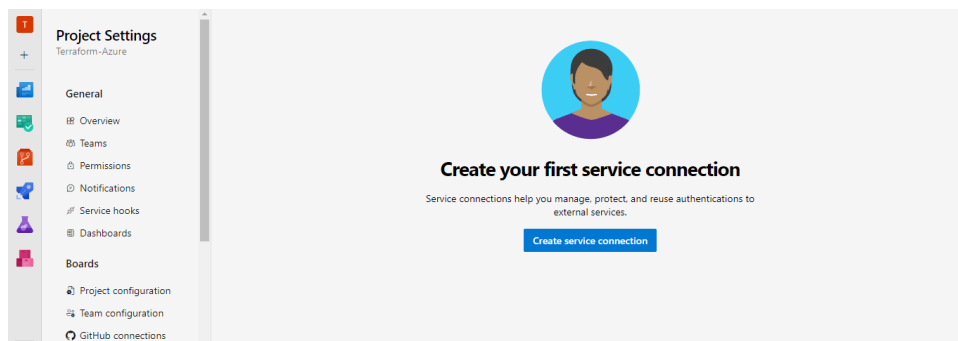
Within our Azure DevOps Project, navigate to Project Settings



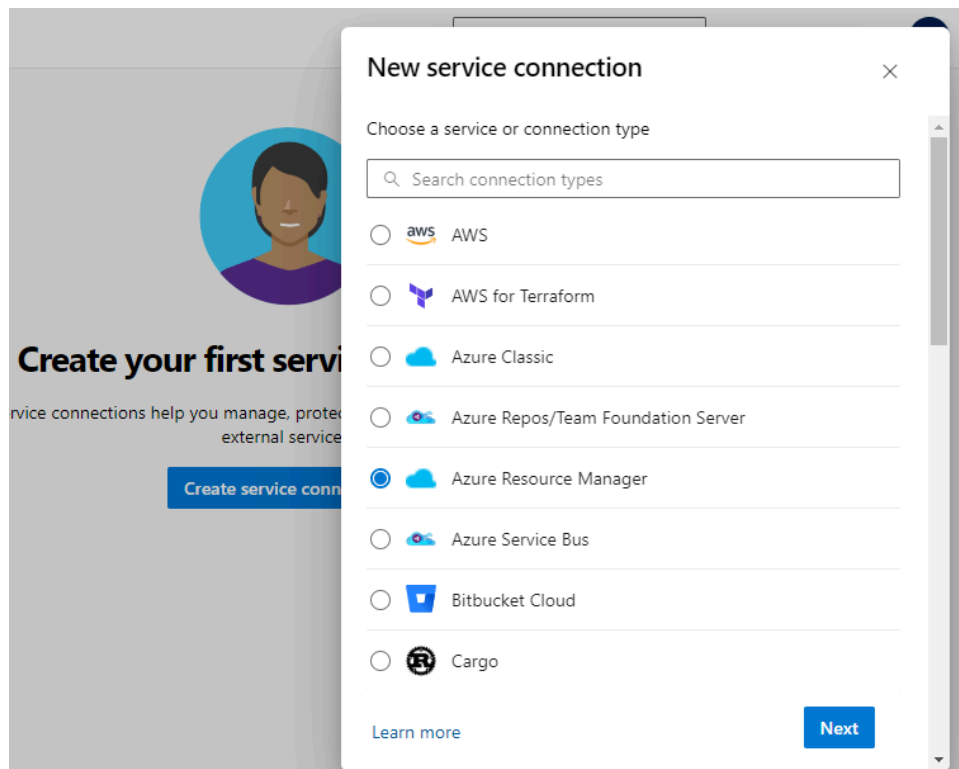
Click on Service Connections.



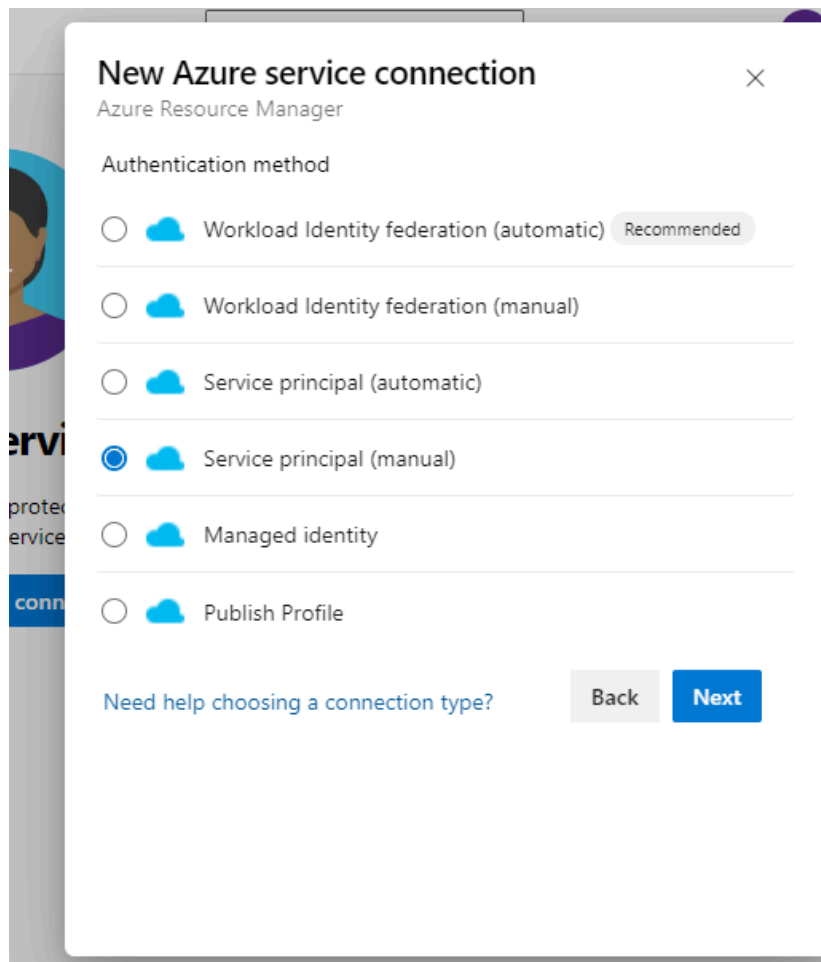
Click on create your first service connection



Click on 'Azure Resource Manager' -> 'Next'.



Then select 'Service principal (automatic)' -> 'Next'.



These are the scope settings used for my SPN:

Select scope level -> Subscription

Resource group -> Select resource group according to your choice

Service connection name -> Give any suitable name to the connection

New Azure service connection ✕

Azure Resource Manager using service principal (automatic)

Scope level

☒ Subscription

☐ Management Group

☐ Machine Learning Workspace

Subscription

Free Trial (ec88c61c-3230-434f-9bf2-a6eff5fe3e48) ▼

Resource group

example-resourcegroup ▼

Details

Service connection name

Azure-Connection

Description (optional)

Azure connection

Check the Grant access permission to all pipelines
Click Save

Security

☒ Grant access permission to all pipelines

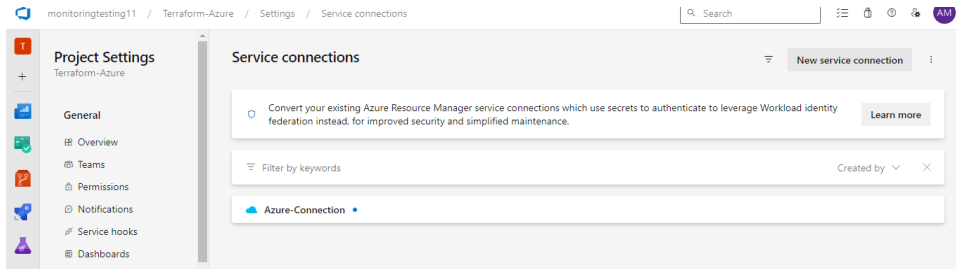
[Learn more](#)

[Troubleshoot](#)

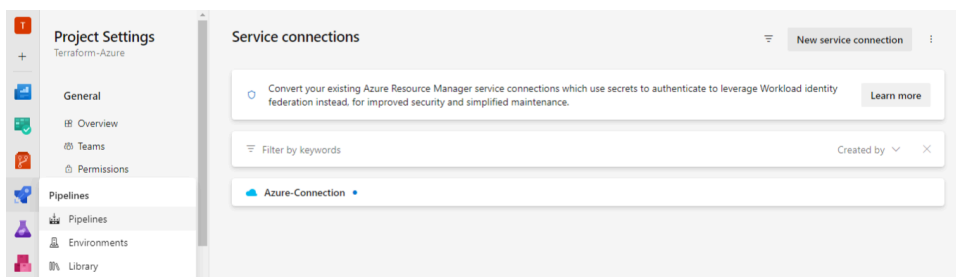
Back

Save

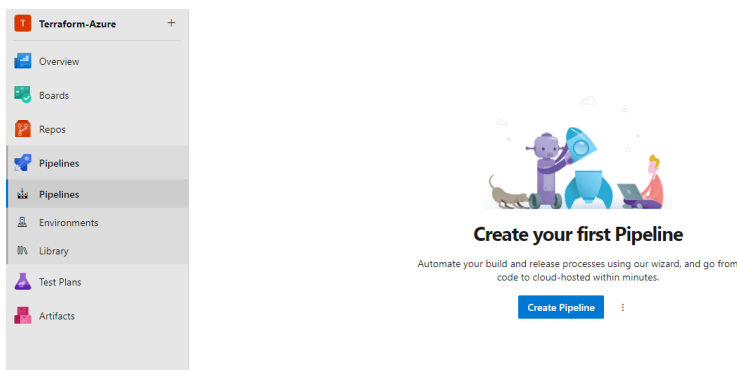
Connection created successfully. Our connection name is Azure-Connection



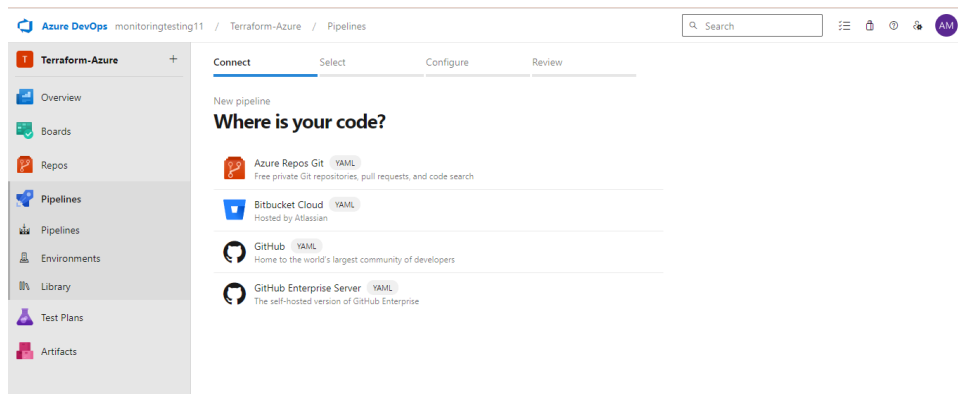
Selecting 'Pipelines' -> 'Pipelines'.



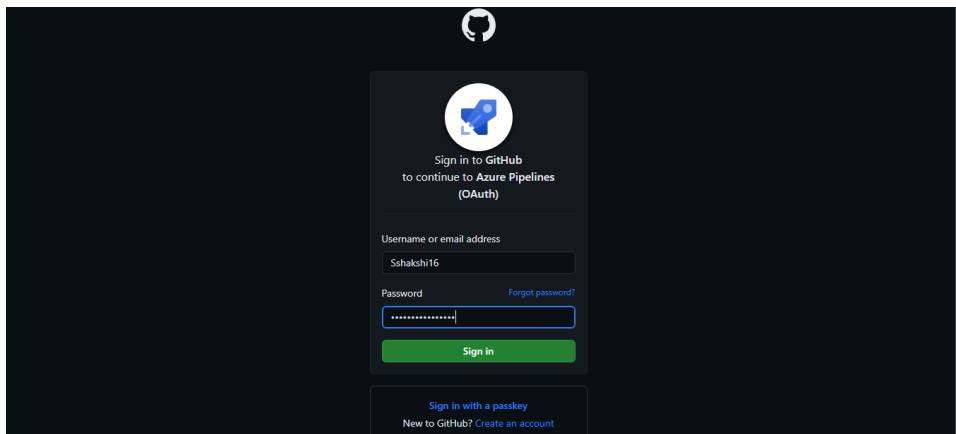
Click create pipeline



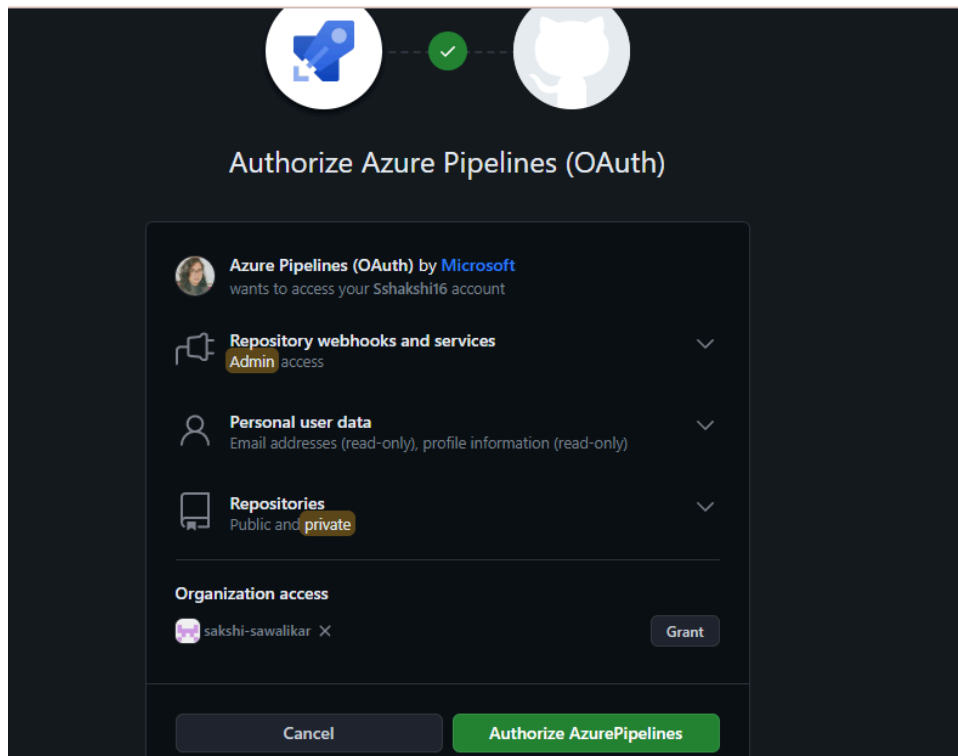
Selecting GitHub



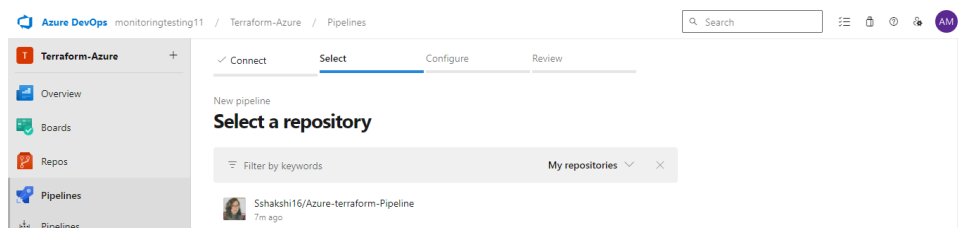
Provide credentials -> Press Sign in



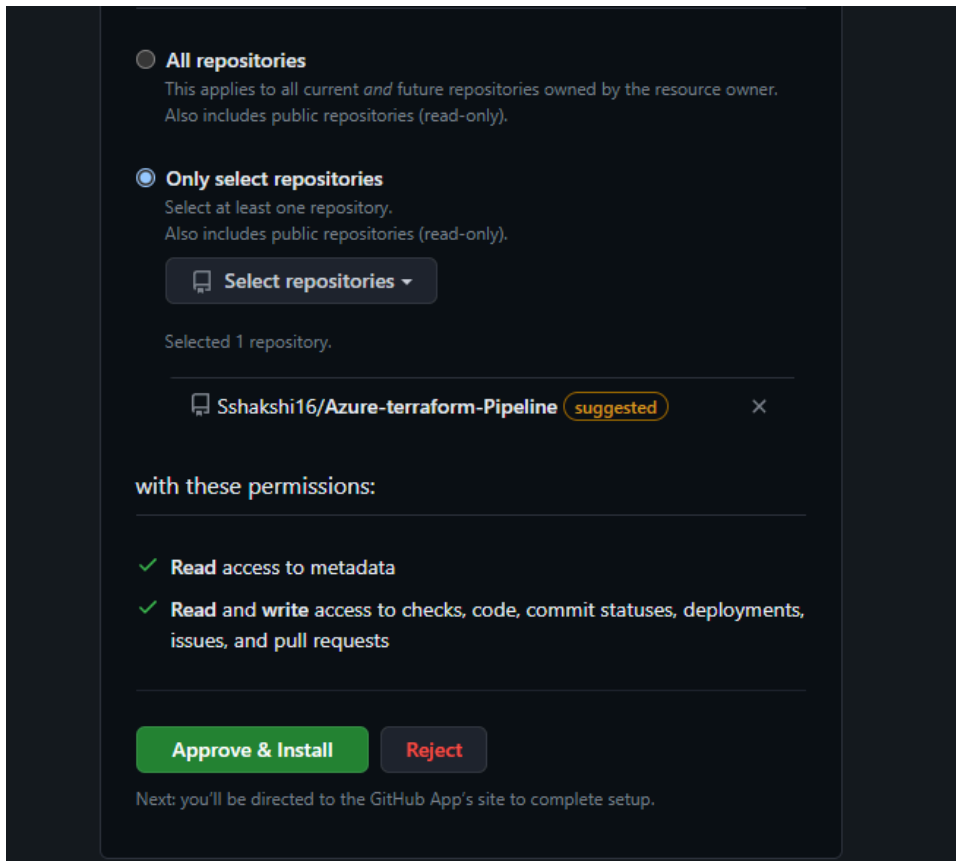
Click Authorise AzurePipelines



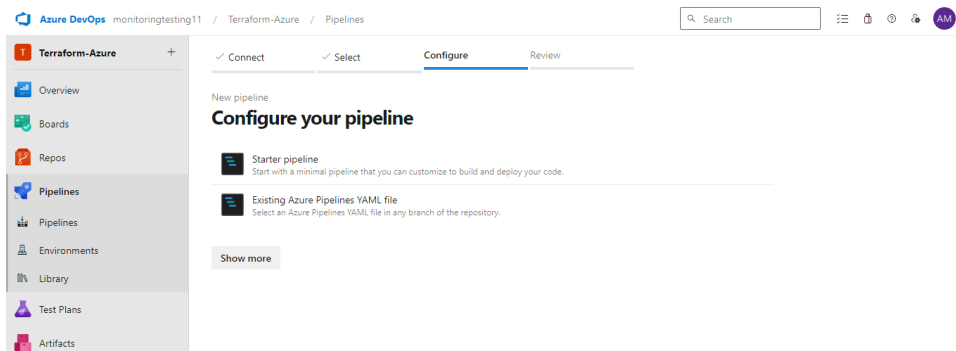
Now after authentication one will see all the GitHub repos.
Select Azure-terraform-pipeline as our main.tf file is present in that repository.



Scroll down to 'Repository Access' and selecting the repo, then clicking 'Approve and Install'.



Click starter pipeline



Now clicking show assistant

Variables

Save and run



Show assistant

Searching terraform installer

The screenshot shows the 'Review your pipeline YAML' step in the Terraform-Azure interface. The sidebar on the left contains navigation links: Overview, Boards, Repos, Pipelines, Environments, Library, Test Plans, and Artifacts. The main area displays a YAML configuration for a pipeline with steps for echo and script execution. The right sidebar shows a search bar with 'terra' entered and a list of tasks including Terraform, Terraform CLI, Terraform Installer, and Terraform tool installer.

```
8 pool:
9   vmImage: ubuntu-latest
10
11 steps:
12   - script: echo Hello, world!
13     displayName: 'Run a one-line script'
14
15   - script: |
16     echo Add other tasks to build, test, and deploy your project.
17     echo See https://aka.ms/yaml
18     displayName: 'Run a multi-line script'
```

Click add

← Terraform Installer ⓘ


Version * ⓘ

Download URL ⓘ

Add

Now selecting terraform CLI

Tasks ⓘ

 Terraform CLI
Execute terraform cli commands

Selecting init command -> Click add

← Terraform CLI ⓘ

Command * ⓘ

init ▼

Configuration Directory ⓘ

\$(System.DefaultWorkingDirectory)

Command Options ⓘ


Backend Type ⓘ

Add

Again selecting terraform CLI

Tasks ⓘ

🔍 terraform CLI

 Terraform CLI
Execute terraform cli commands

Selecting plan

← Terraform CLI ⓘ

Command * ⓘ

plan

Configuration Directory ⓘ

\$(System.DefaultWorkingDirectory)

☒ Run Azure CLI Login ⓘ

Command Options ⓘ

☒ Allow Telemetry Collection * ⓘ

Providers ^

AzureRM Provider Service Connection ⓘ

Azure-connection-recom

Again selecting terraform CLI

Tasks



🔍 terraform CLI



Terraform CLI

Execute terraform cli commands

Selecting apply

The screenshot shows the Terraform CLI configuration interface. At the top, there is a back arrow and the text 'Terraform CLI' followed by an information icon. Below this, the 'Command' field is set to 'apply'. The 'Configuration Directory' field contains the path '\$(System.DefaultWorkingDirectory)'. There are two checked checkboxes: 'Run Azure CLI Login' and 'Allow Telemetry Collection'. The 'Command Options' field is empty. Below this, there is a section titled 'Providers' with an expand/collapse arrow. Under 'Providers', the 'AzureRM Provider Service Connection' is set to 'Free Trial(ec88c61c-3230-434f-9bf2-a6e...)'. The 'Azure Subscription Id' field is empty.

← Terraform CLI ⓘ

Command * ⓘ

apply ▼

Configuration Directory ⓘ

\$(System.DefaultWorkingDirectory)

☒ Run Azure CLI Login ⓘ

Command Options ⓘ

☒ Allow Telemetry Collection * ⓘ

Providers ^

AzureRM Provider Service Connection ⓘ

Free Trial(ec88c61c-3230-434f-9bf2-a6e...) ▼

Azure Subscription Id ⓘ

azure-pipelines.yml

Starter pipeline

Start with a minimal pipeline that you can customise to build and deploy your code.

Add steps that build, run tests, deploy, and more:

<https://aka.ms/yaml>

trigger:

- main

pool:

vmImage: ubuntu-latest

steps:

- script: echo Hello, world!

displayName: 'Run a one-line script'

- task:

JasonBJohnson.azure-pipelines-tasks-terraform.azure-pipelines-tasks-terraform-installer.TerraformInstaller@0

displayName: 'Install Terraform'

inputs:

terraformVersion: 'latest'

- task: TerraformCLI@2

inputs:

command: 'version'

allowTelemetryCollection: true

- task: TerraformCLI@2

inputs:

command: 'init'

allowTelemetryCollection: true

- task: TerraformCLI@2

inputs:

command: 'plan'

environmentServiceName: 'Azure-connection-recom'

runAzLogin: true

allowTelemetryCollection: true

- task: TerraformCLI@2

inputs:

command: 'apply'

environmentServiceName: 'Free
Trial(ec88c61c-3230-434f-9bf2-a6eff5fe3e48)'

runAzLogin: true

allowTelemetryCollection: true

- script: |

echo Add other tasks to build, test, and deploy your project.

echo See <https://aka.ms/yaml>

displayName: 'Run a multi-line script'

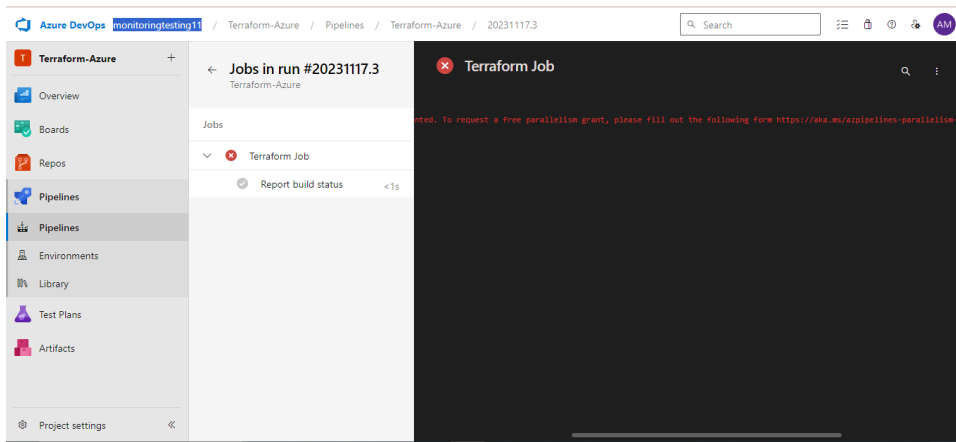
Click save and run

Variables

Save and run

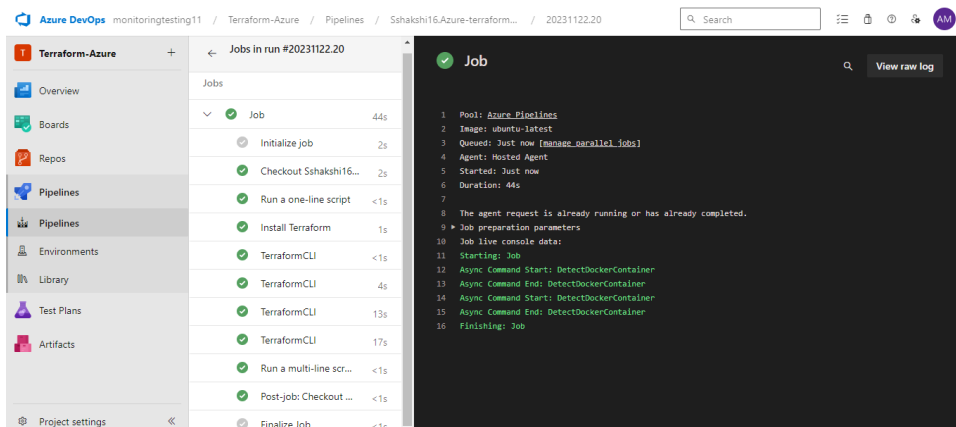


Error



Filling the form by clicking on the link , it took me 2-3 business days to get approval .

After that running the pipeline again



Here resource group created successfully

Resource groups

Default Directory (monitoringtesting11outlook.onmicrosoft.com)



Create Manage view Refresh Export to CSV Open query Assign tags

Filter for any field...

Subscription equals all

Location equals all



Add filter

Showing 1 to 4 of 4 records.

No grouping

List view

<input checked="" type="checkbox"/> Name ↑↓	Subscription ↑↓	Location ↑↓	
<input type="checkbox"/> cloud-shell-storage-centralindia	Free Trial	Central India	...
<input type="checkbox"/> example-resourcegroup	Free Trial	West Europe	...
<input checked="" type="checkbox"/> example-resources	Free Trial	West Europe	...
<input type="checkbox"/> NetworkWatcherRG	Free Trial	West Europe	...