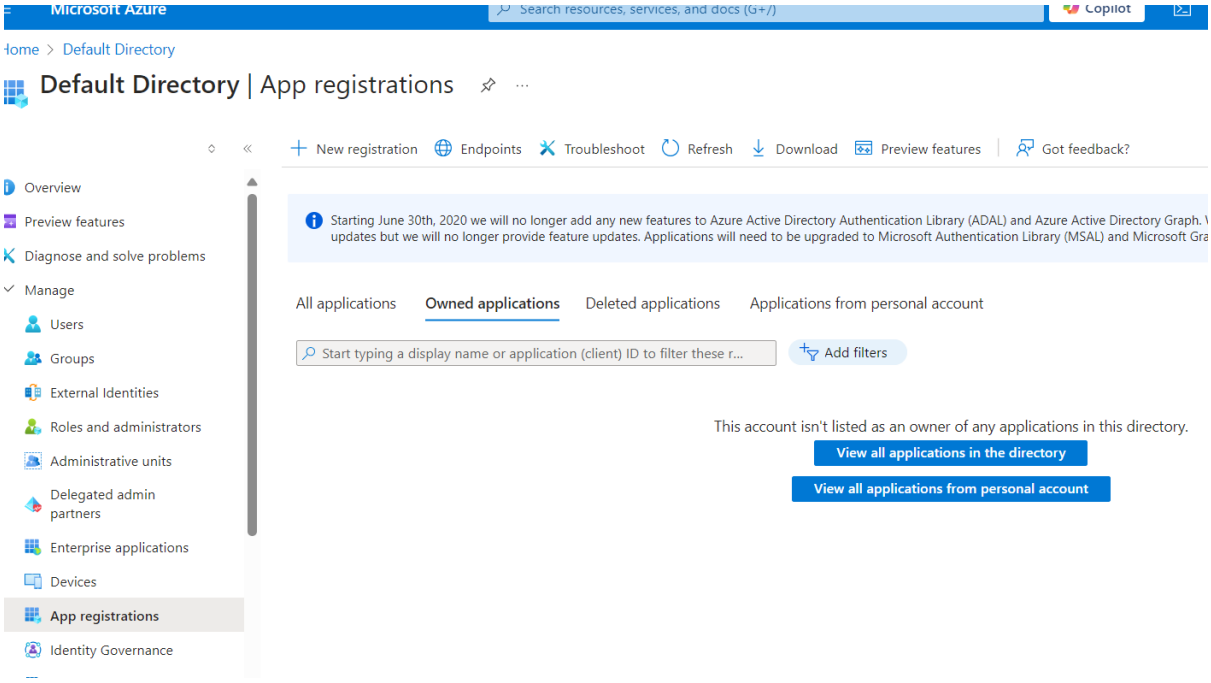
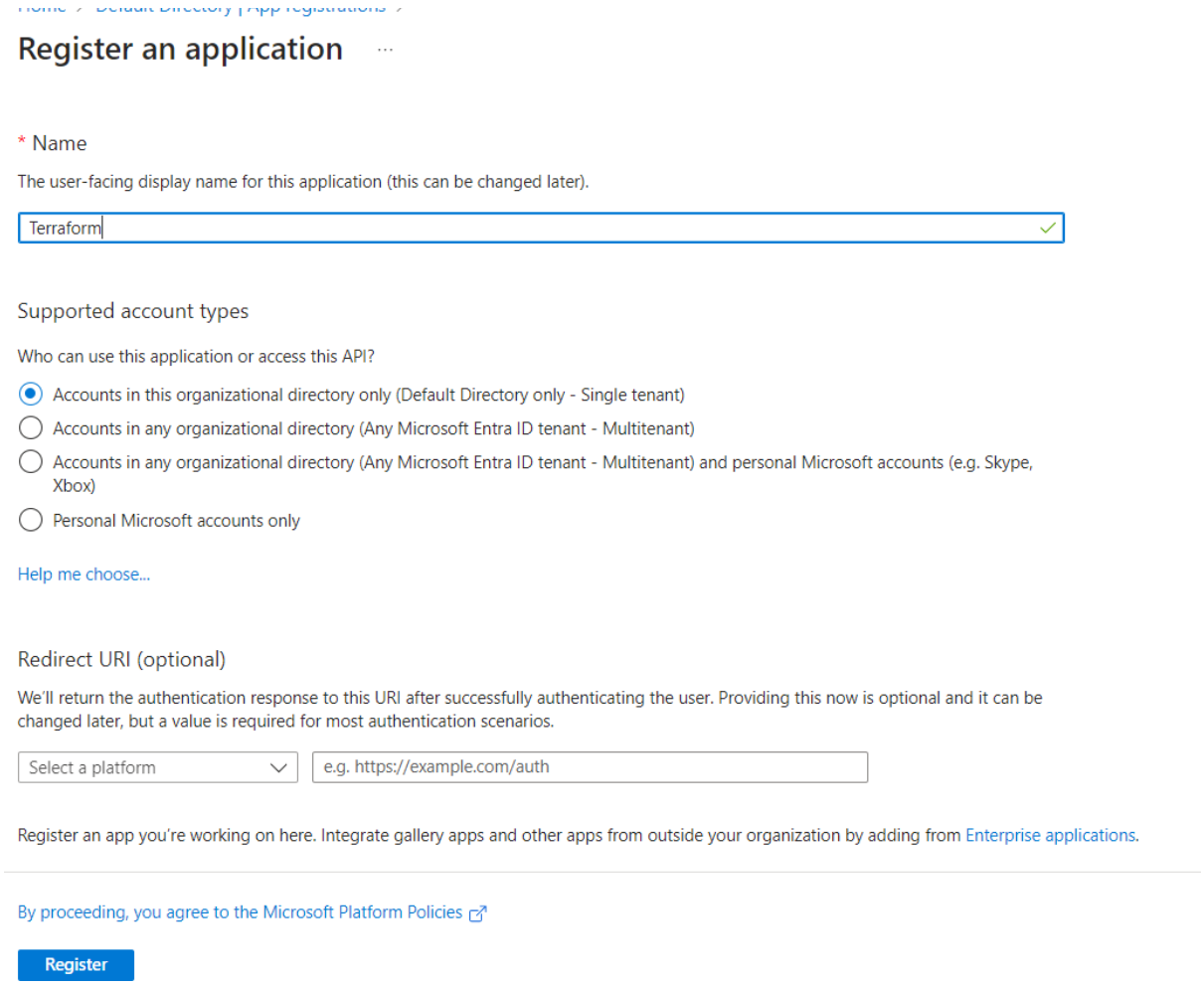
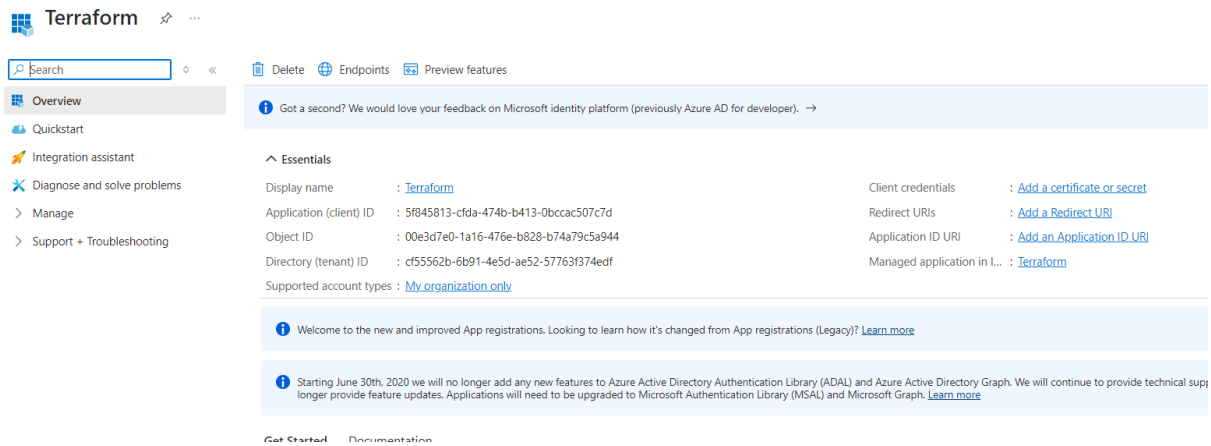
Terraform Integration with Azure CI/CD Pipelines

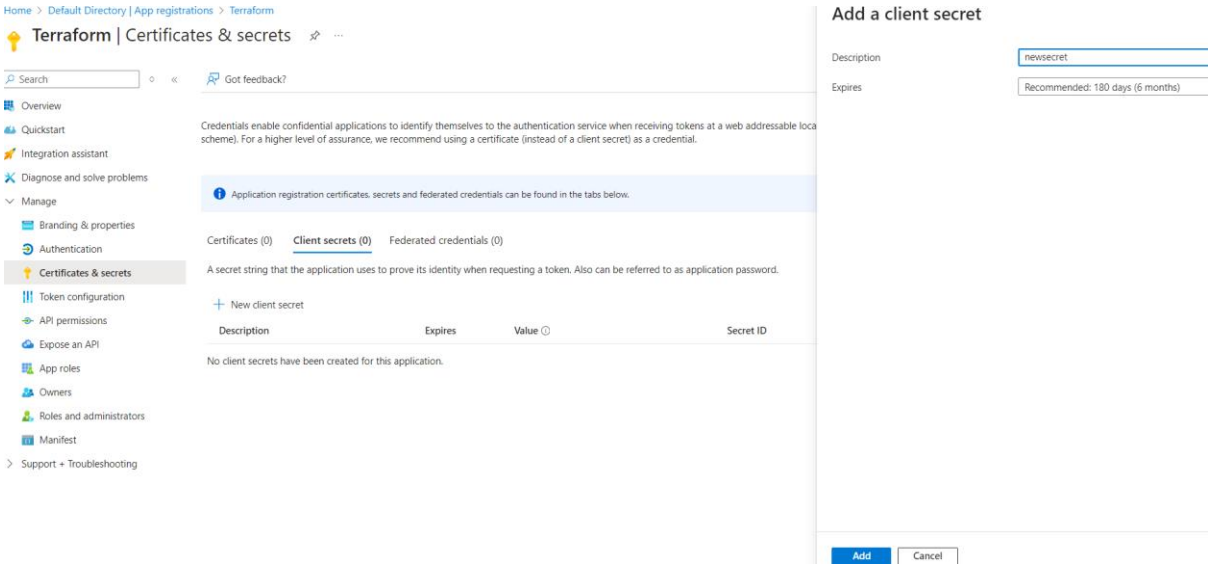
Create a service principal for Terraform in Microsoft Entra Id under app registration.







Add new clientsecret . Make a note of that. Once you close that page you will be no longer access to clientsecret.



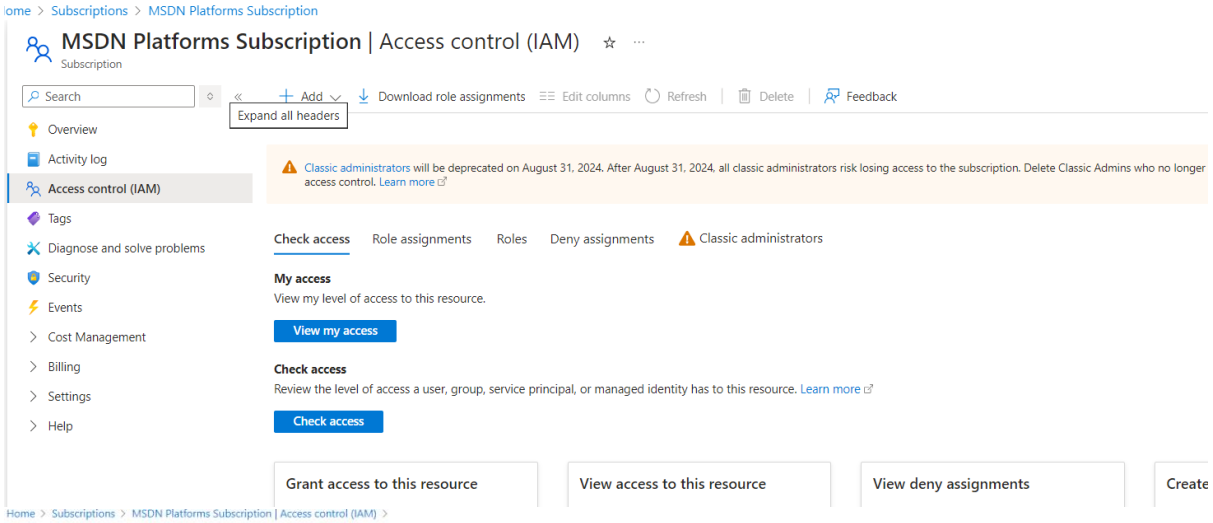
Replace Subscription ID, Tenant ID, Client ID and Client-Secret in your Terraform script, main.tf.

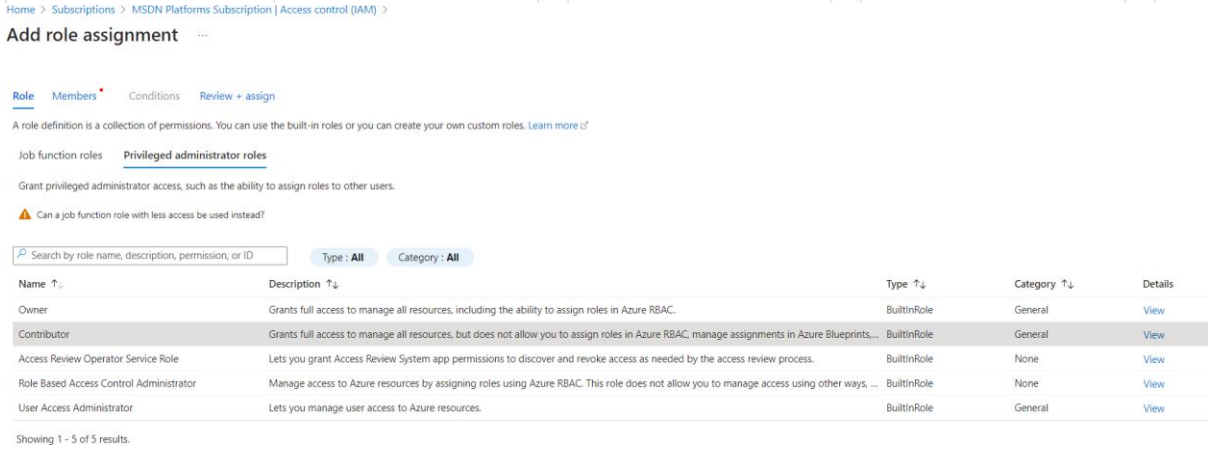
Here are the contents of the file main.tf

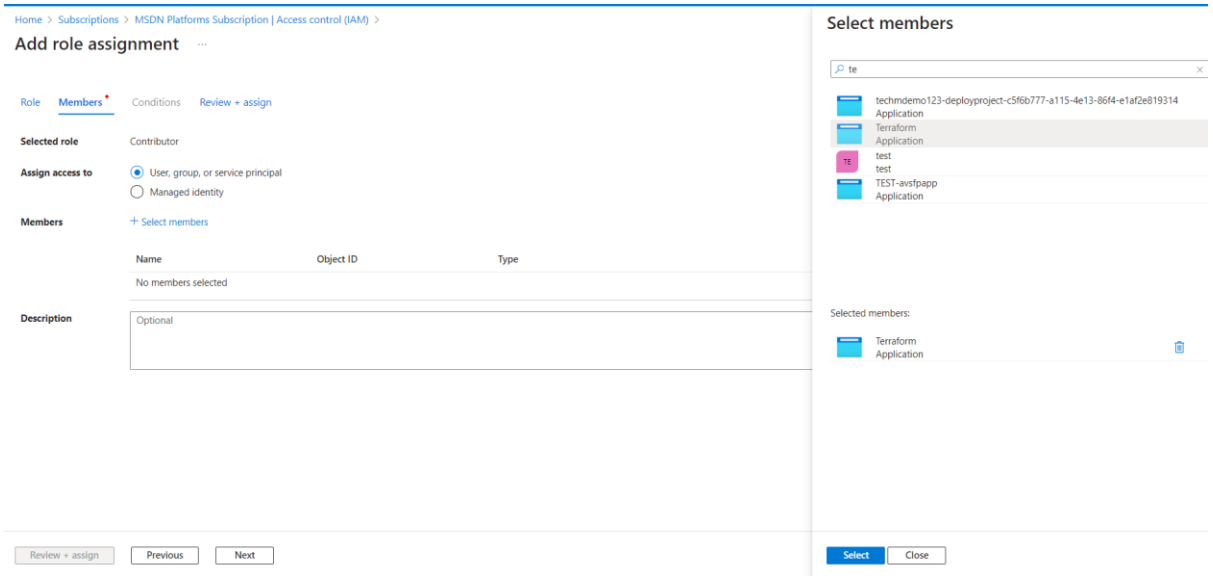
```
terraform {
 required_providers {
  azurerm={
     source="hashicorp/azurerm"
     version="3.17.0"
  }
 }
}

provider "azurerm" {
 subscription_id = "c5f6b777-a115-4e13-86f4-e1af2e819314"
 tenant_id = "cf55562b-6b91-4e5d-ae52-57763f374edf"
```

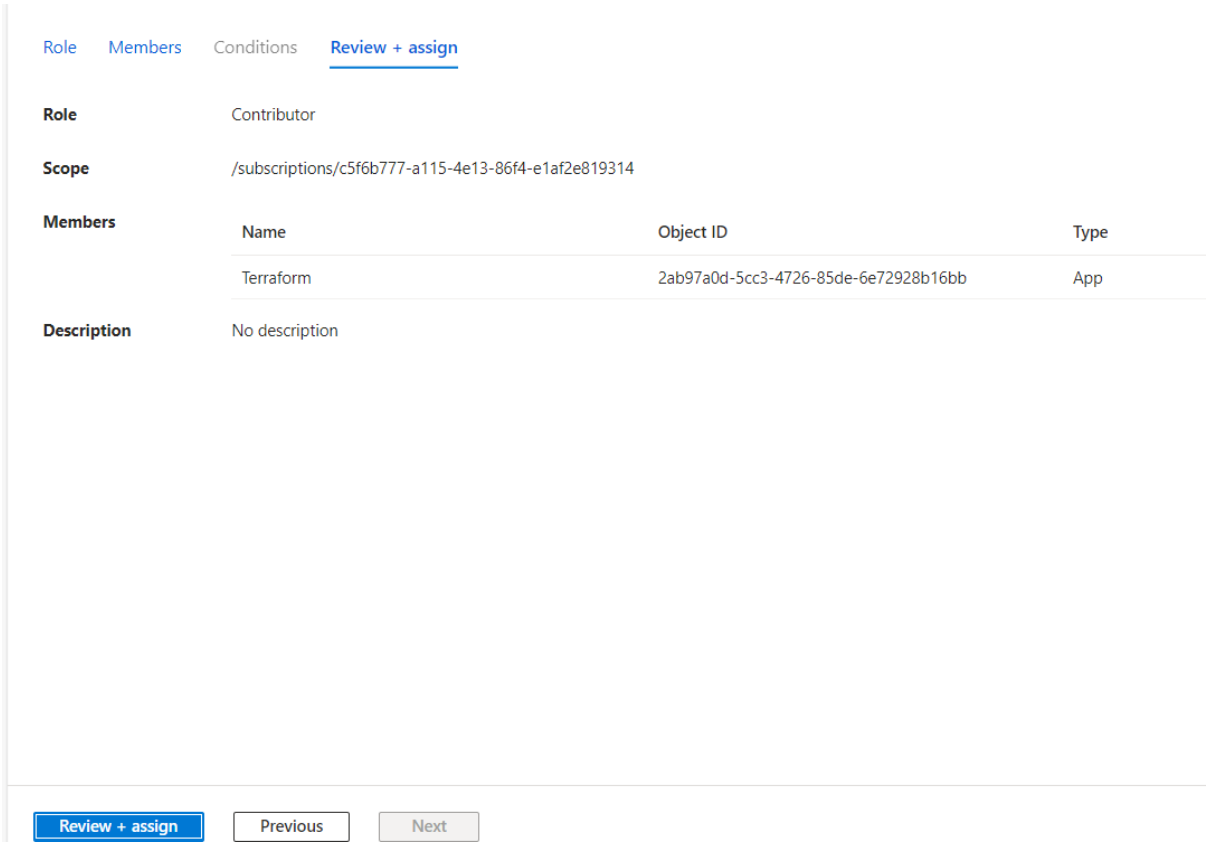
```
 client_id = "5f845813-cfda-474b-b413-0bccac507c7d"
 client_secret = "t.w8Q~w4--yGImnfXj8ExUQmzjuwjqtzhraDdbAZ"
 features {
 }
}

resource "azurerm_service_plan" "demowebnew456" {
 name                = "demowebnew456"
 resource_group_name = "template-grp"
 location            = "North Europe"
 os_type             = "Windows"
 sku_name            = "S1"
}

resource "azurerm_windows_web_app" "latestwebapp765" {
 name                = "latestwebapp765"
 resource_group_name = "template-grp"
 location            = "North Europe"
 service_plan_id     = azurerm_service_plan.demowebnew456.id

 site_config {
  always_on = false
  application_stack{
     current_stack="dotnet"
     dotnet_version="v6.0"
  }
 }

 depends_on = [
  azurerm_service_plan.demowebnew456
 ]
}
```

Provide contributor access to Terraform Service Principal under IAM in your subscription.

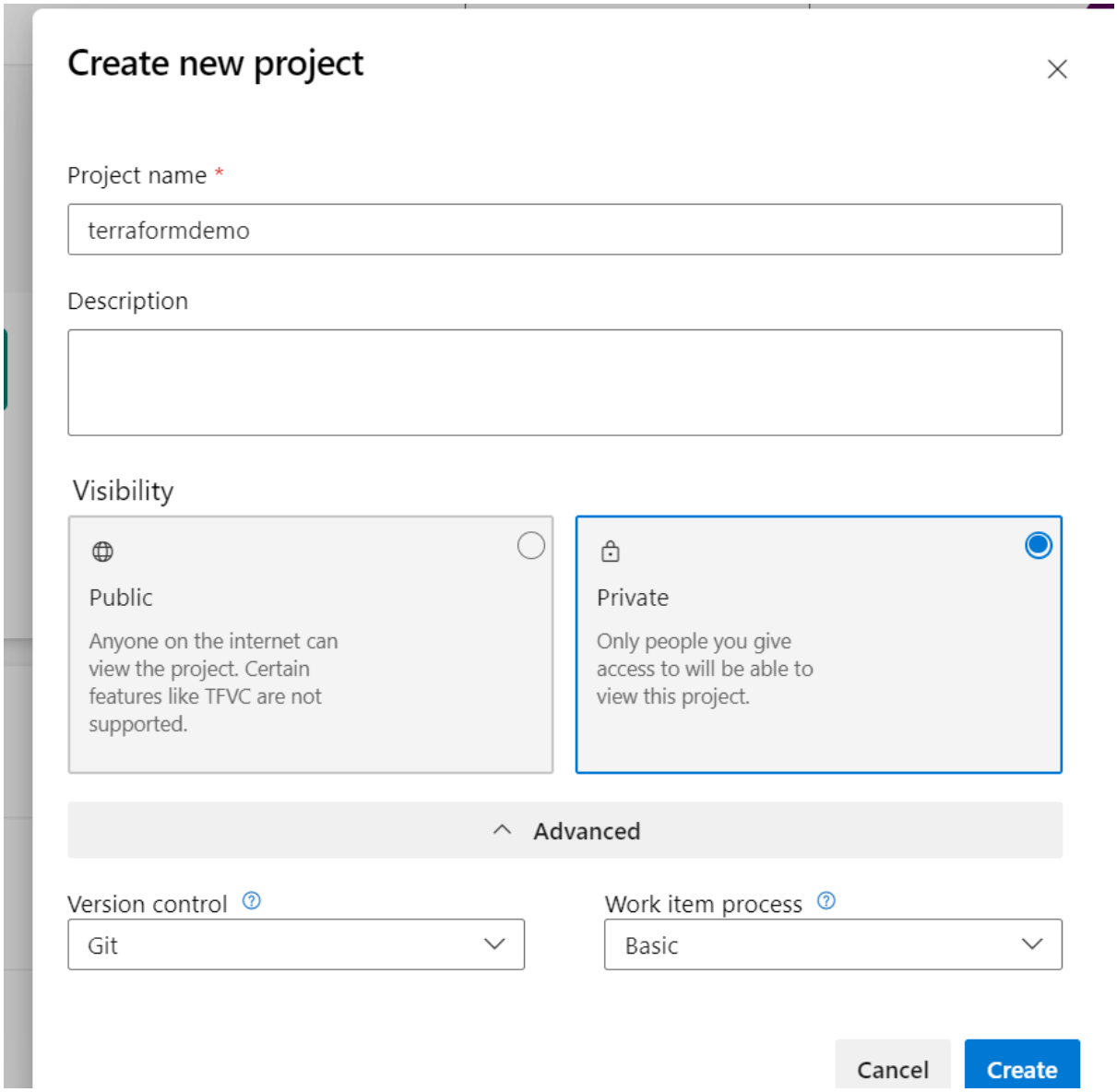

ome > Subscriptions > MSDN Platforms Subscription
MSDN Platforms Subscription | Access control (IAM)
Subscription
Search
Add
Download role assignments
Edit columns
Refresh
Delete
Feedback
Expand all headers
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Security
Events
Cost Management
Billing
Settings
Help
Classic administrators will be deprecated on August 31, 2024. After August 31, 2024, all classic administrators risk losing access to the subscription. Delete Classic Admins who no longer access control. Learn more
Check access
Role assignments
Roles
Deny assignments
Classic administrators
My access
View my level of access to this resource.
View my access
Check access
Review the level of access a user, group, service principal, or managed identity has to this resource. Learn more
Check access
Grant access to this resource
View access to this resource
View deny assignments
Create


Home > Subscriptions > MSDN Platforms Subscription | Access control (IAM) >
Add role assignment
Role
Members
Conditions
Review + assign
A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. Learn more
Job function roles
Privileged administrator roles
Grant privileged administrator access, such as the ability to assign roles to other users.
Can a job function role with less access be used instead?
Search by role name, description, permission, or ID
Type : All
Category : All
Name
Description
Type
Category
Details
Owner
Grants full access to manage all resources, including the ability to assign roles in Azure RBAC.
BuiltInRole
General
View
Contributor
Grants full access to manage all resources, but does not allow you to assign roles in Azure RBAC, manage assignments in Azure Blueprints,...
BuiltInRole
General
View
Access Review Operator Service Role
Lets you grant Access Review System app permissions to discover and revoke access as needed by the access review process.
BuiltInRole
None
View
Role Based Access Control Administrator
Manage access to Azure resources by assigning roles using Azure RBAC. This role does not allow you to manage access using other ways, ...
BuiltInRole
None
View
User Access Administrator
Lets you manage user access to Azure resources.
BuiltInRole
General
View
Showing 1 - 5 of 5 results.

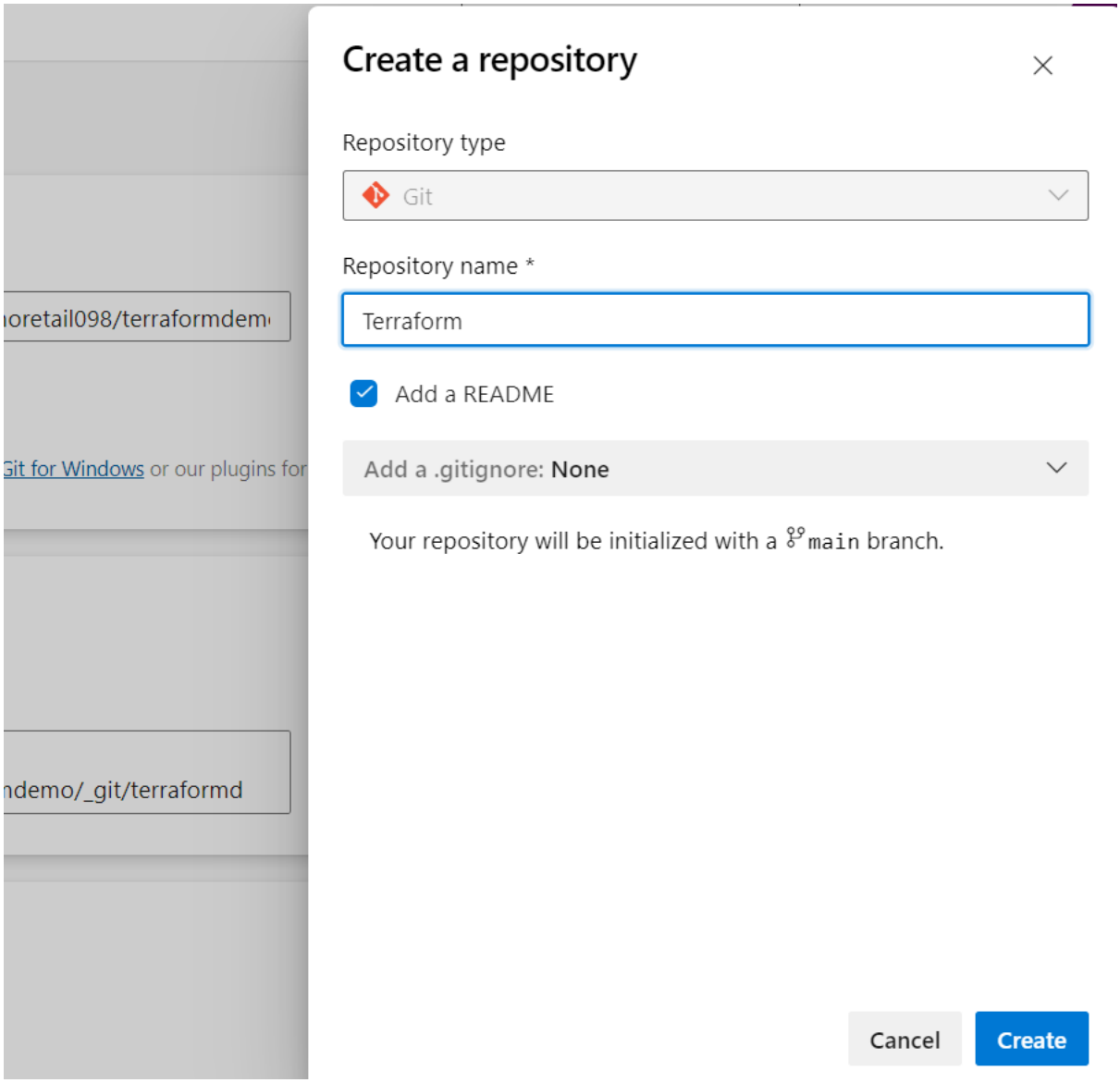

Home > Subscriptions > MSDN Platforms Subscription | Access control (IAM) >
Add role assignment
Role
Members
Conditions
Review + assign
Selected role
Contributor
Assign access to
User, group, or service principal
Managed identity
Members
+ Select members
Name
Object ID
Type
No members selected
Description
Optional
Review + assign
Previous
Next
Select members
te
techmdemo123-deployproject-c5f6b777-a115-4e13-86f4-e1af2e819314
Application
Terraform
Application
test
test
TEST-avsfpapp
Application
Selected members:
Terraform
Application
Select
Close



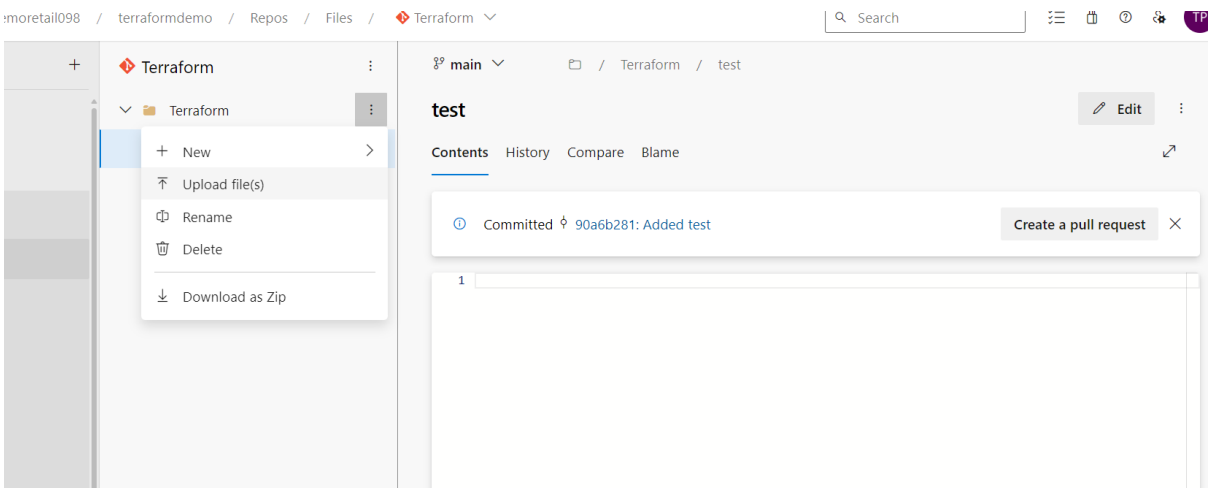
On Azure DevOps Portal creating a new Project and pushing the sample ASP.Net web application code from local repo to Azure repos.

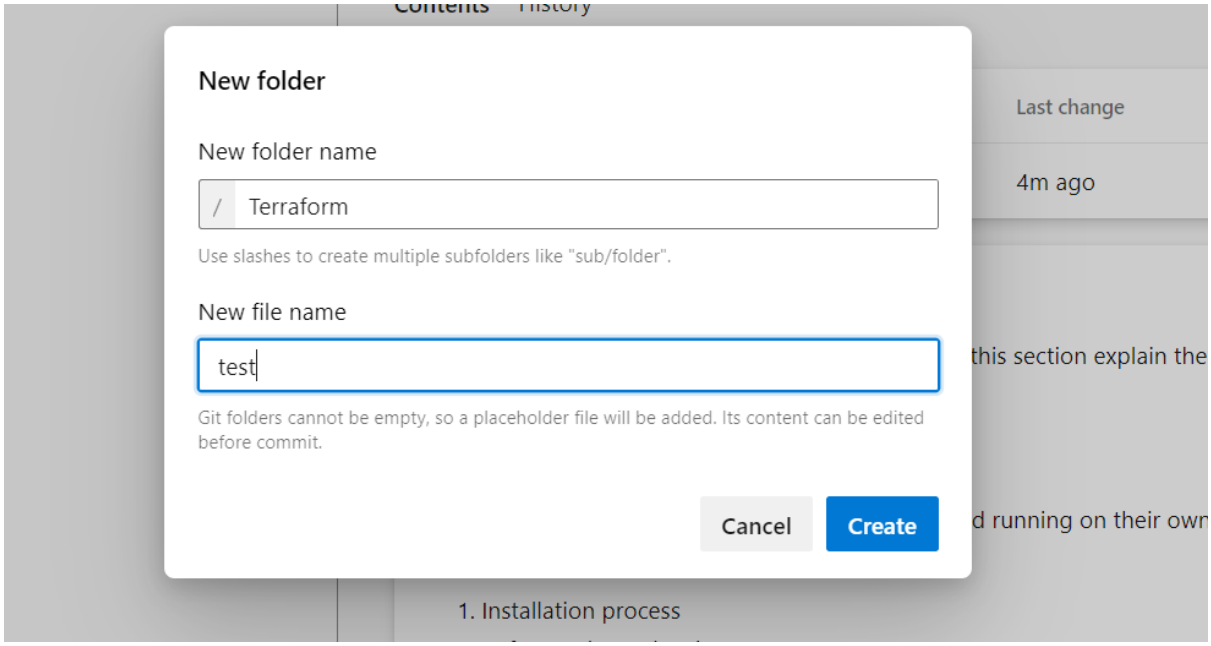


Create a Repository with name Terraform in Azure Repos.

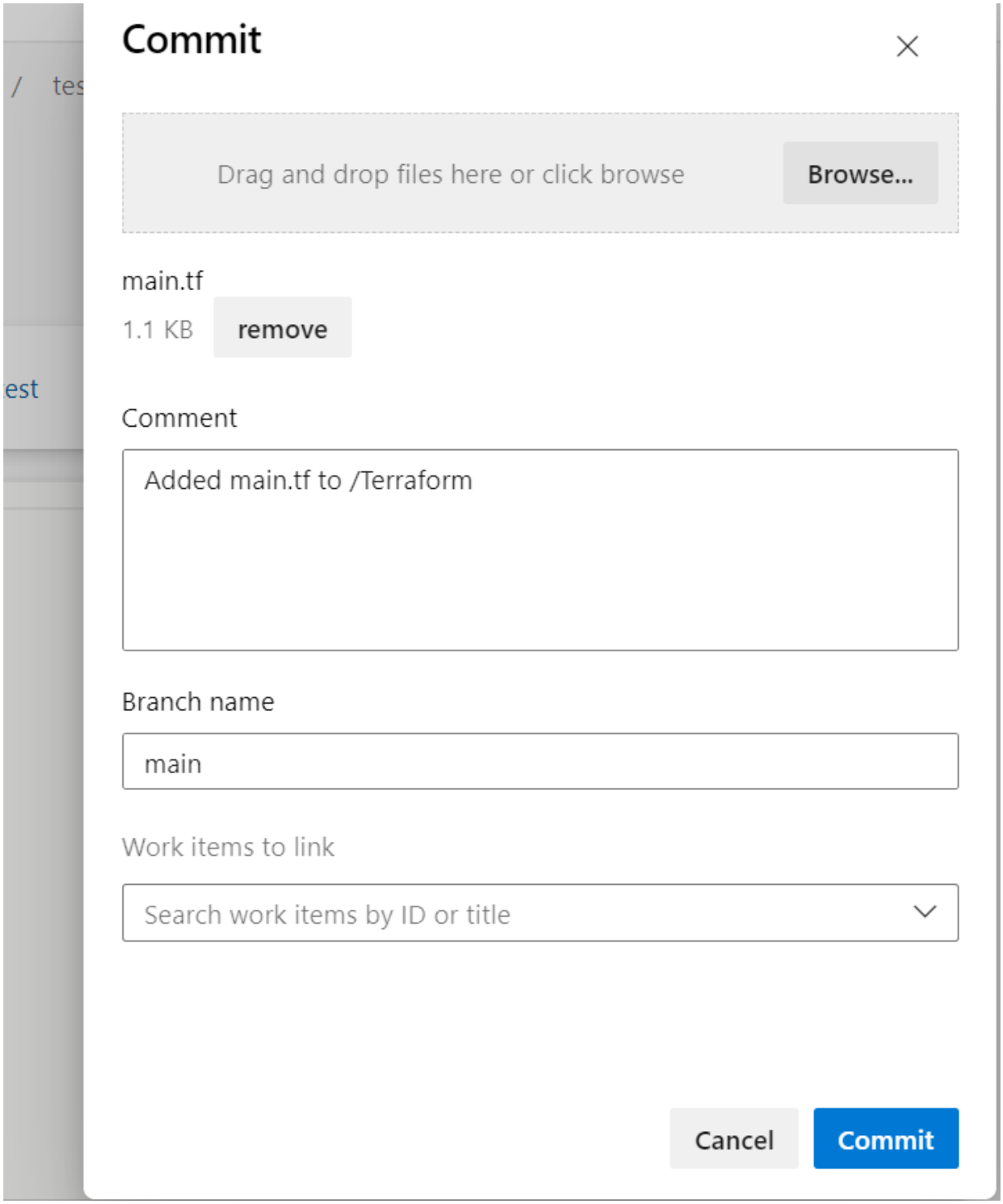


Create a new folder with name Terraform and new file “test”.

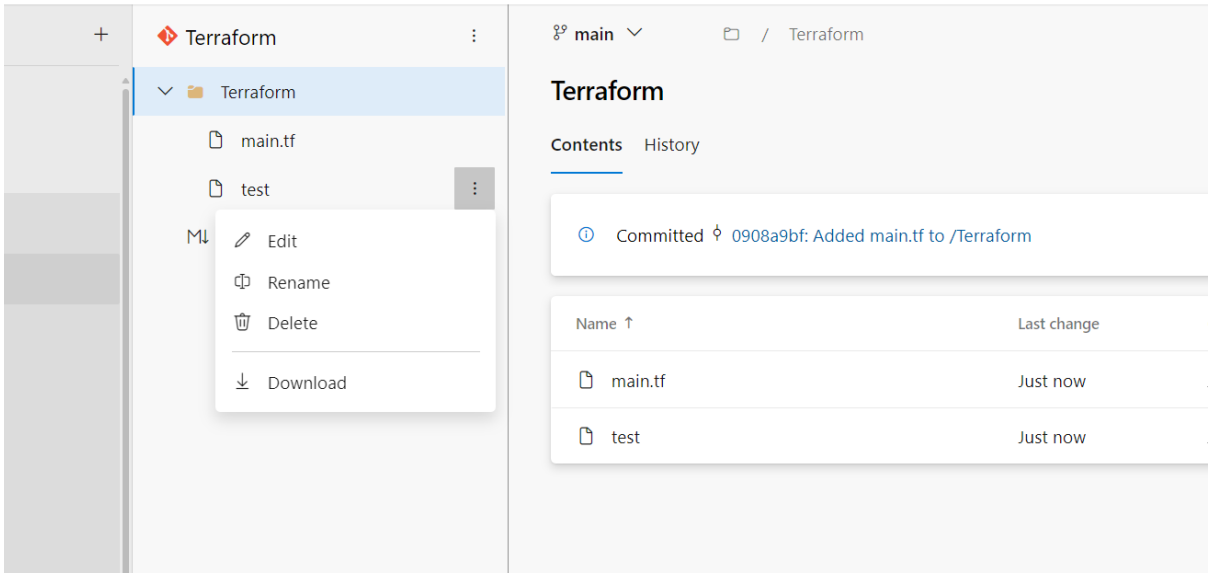


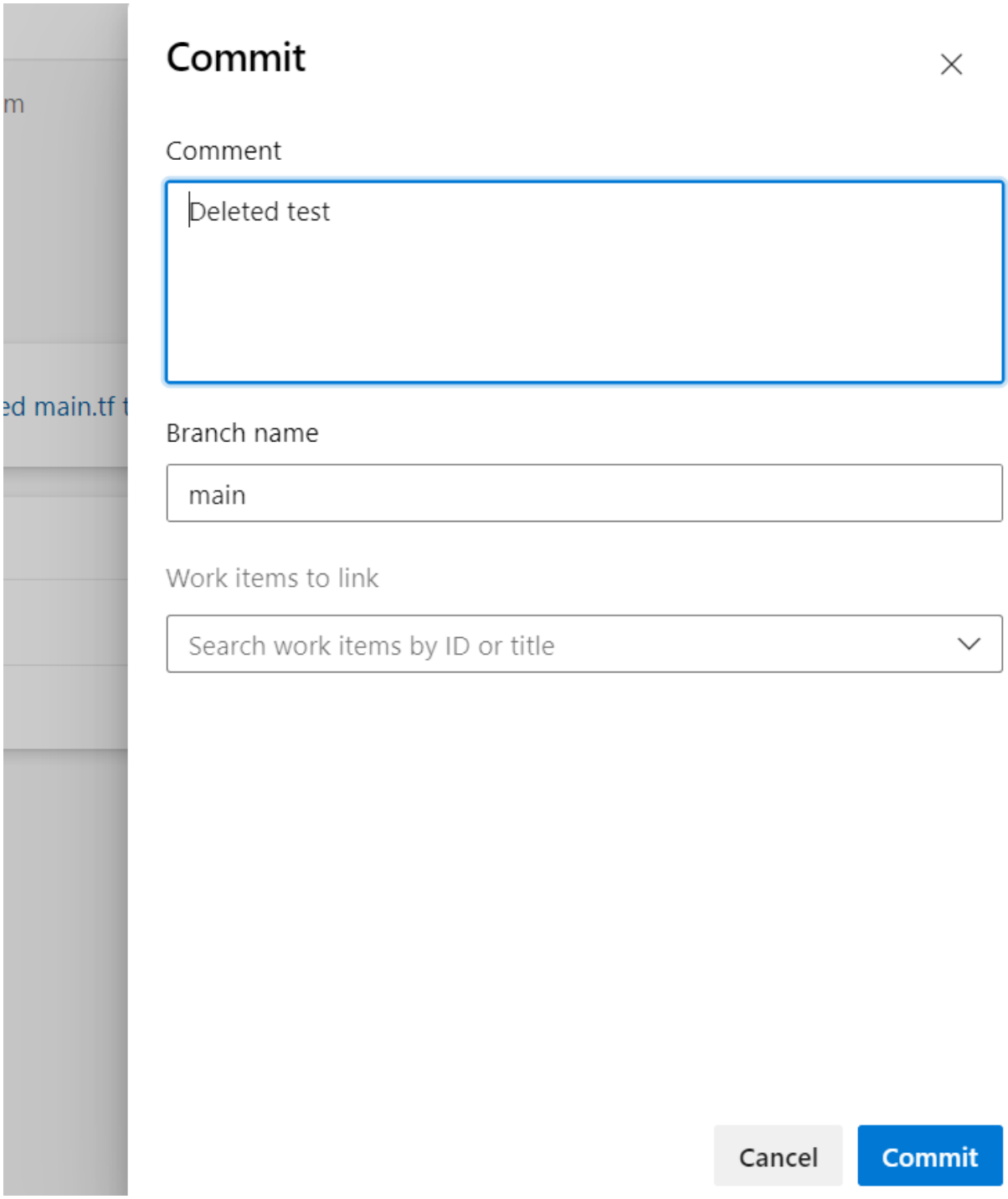


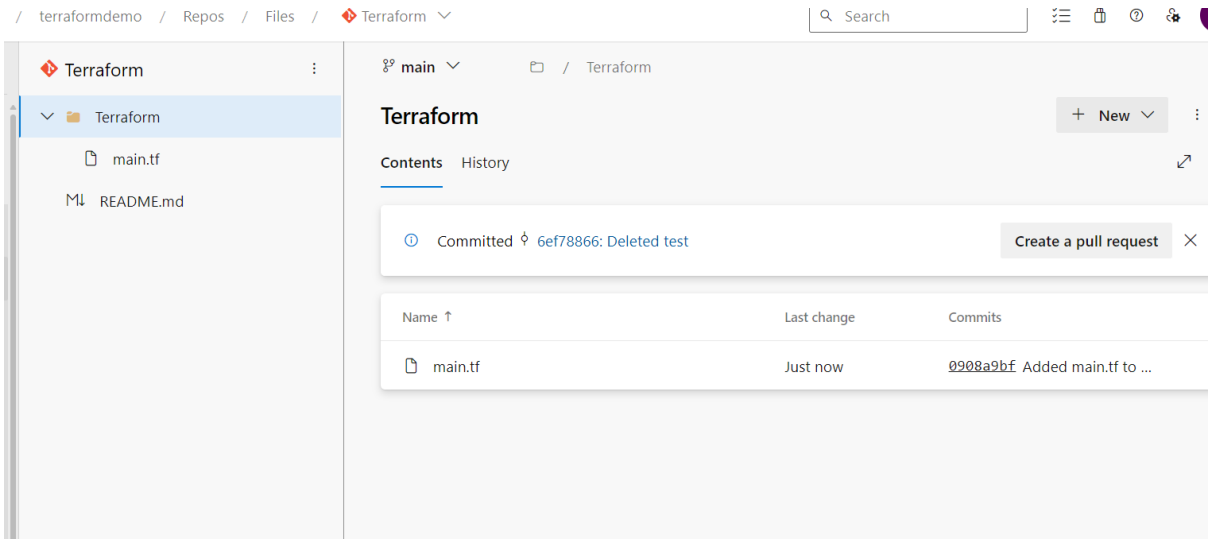
Upload main.tf in the same Terraform folder.



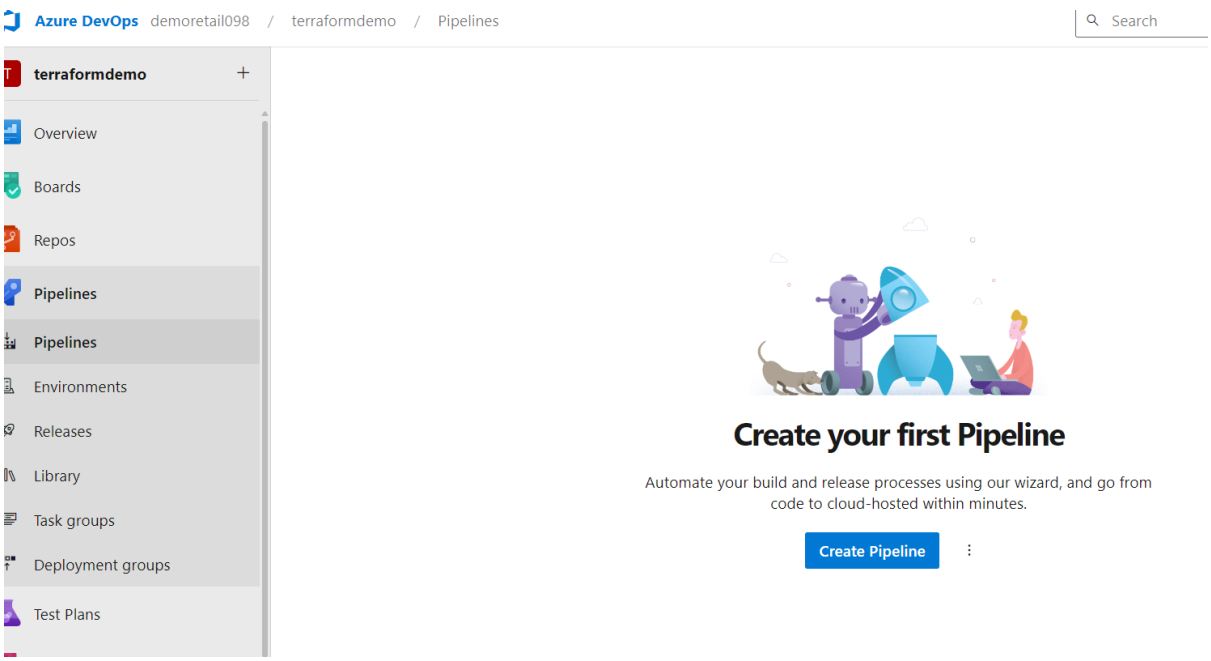
Delete test file and commit.

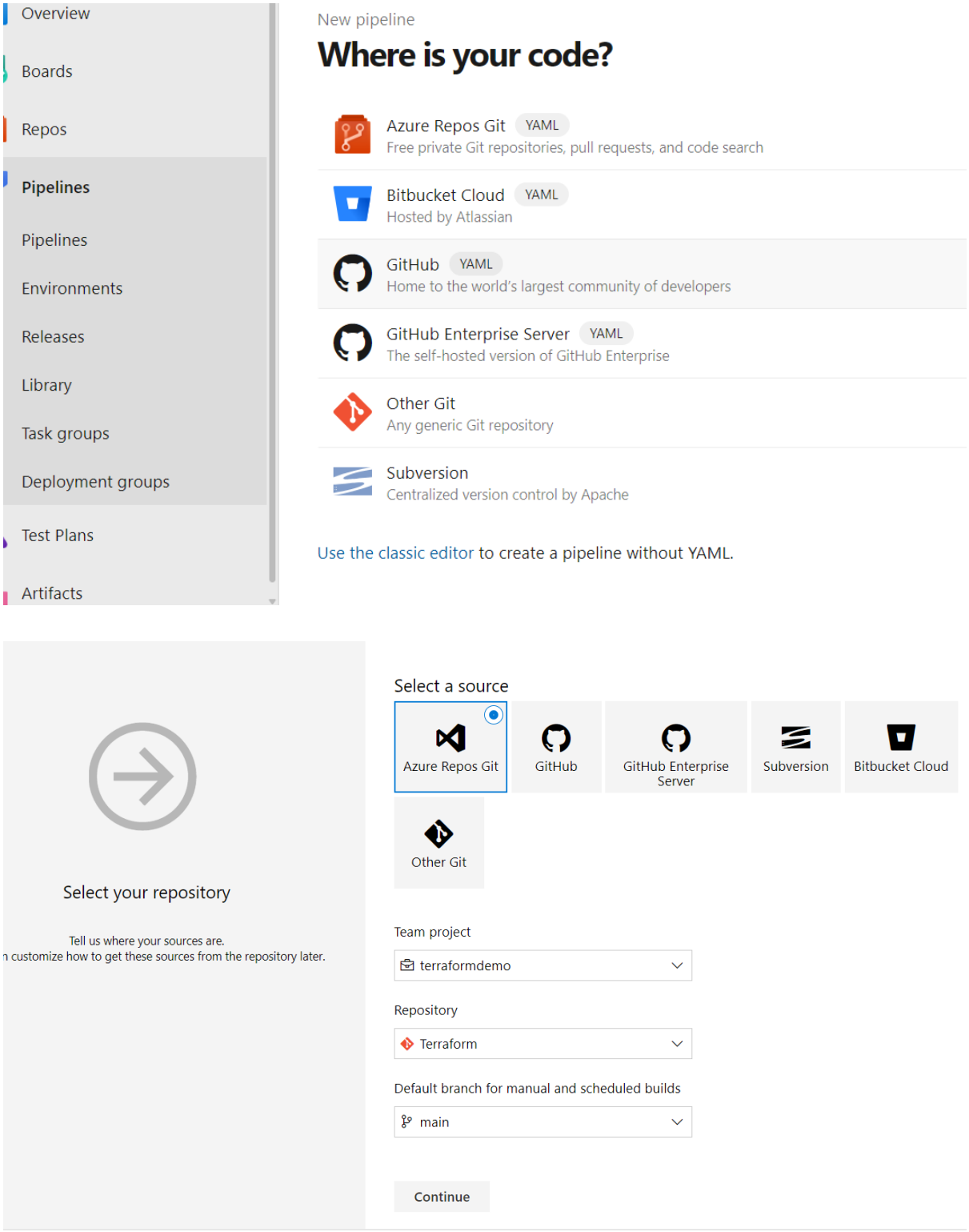

Terraform
Terraform
main.tf
test
Edit
Rename
Delete
Download
main
/ Terraform
Terraform
Contents
History
Committed 0908a9bf: Added main.tf to /Terraform
Name
Last change
main.tf
Just now
test
Just now


Commit
Comment
Deleted test
Branch name
main
Work items to link
Search work items by ID or title
Cancel
Commit

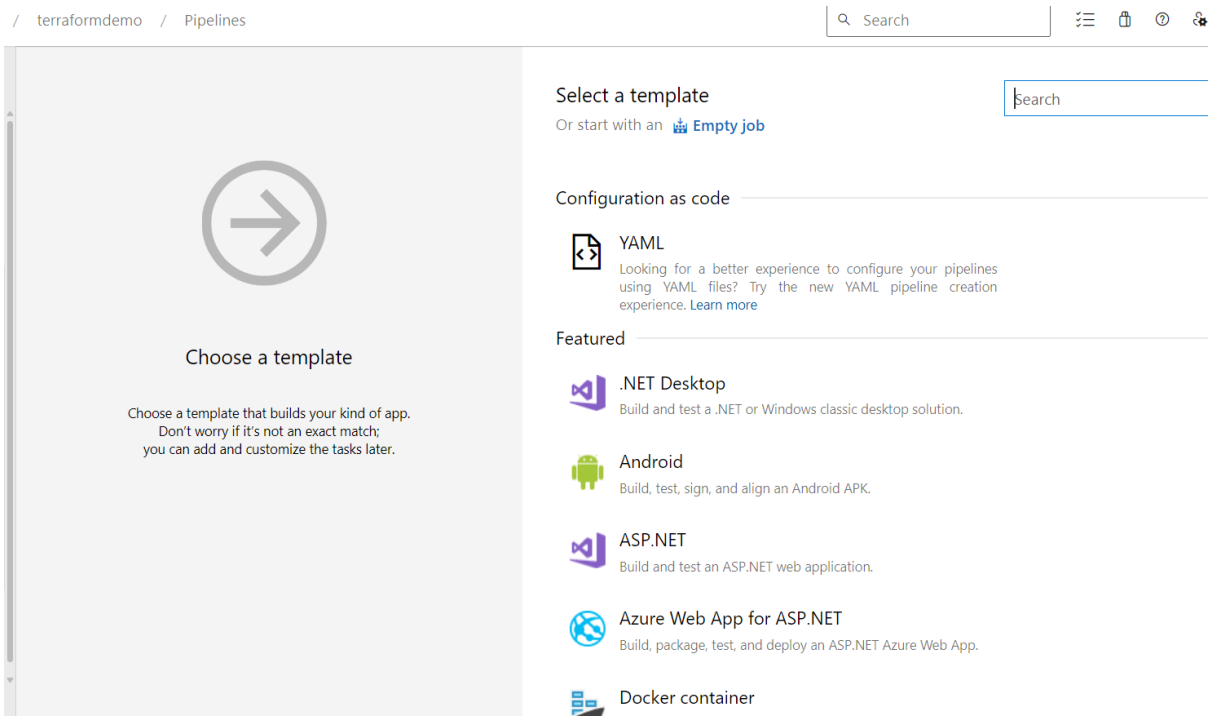


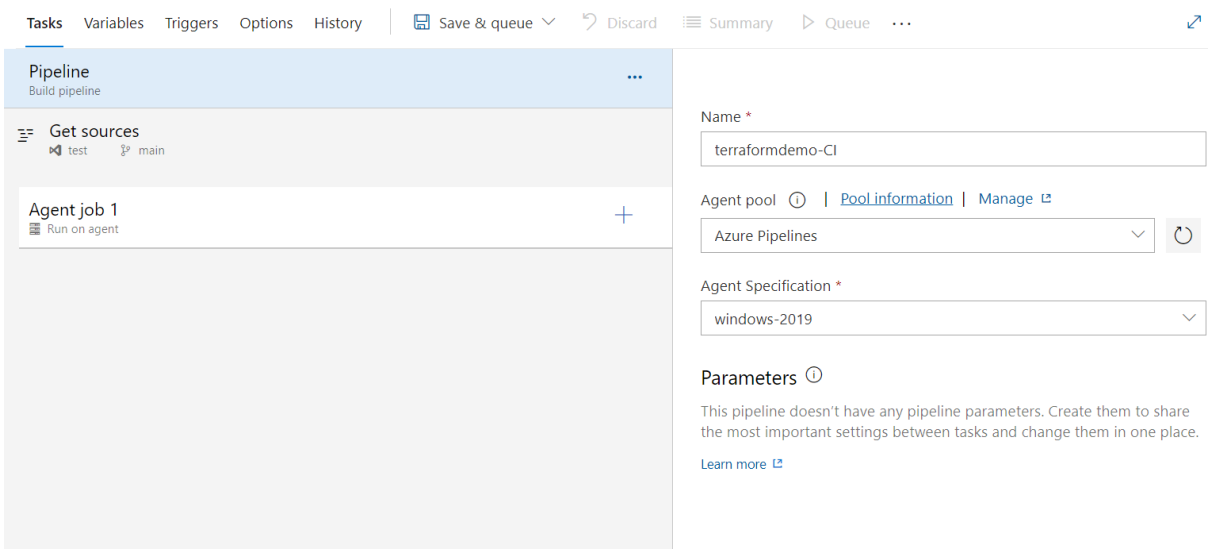
Create Classic Editor Build Pipeline



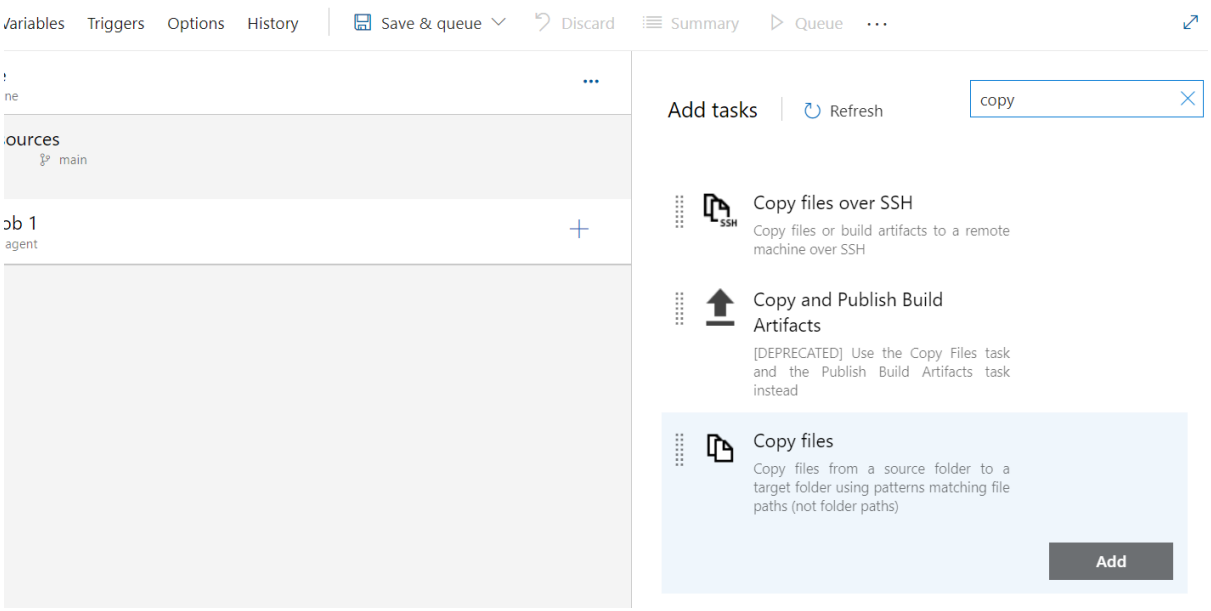


Click on Empty Job

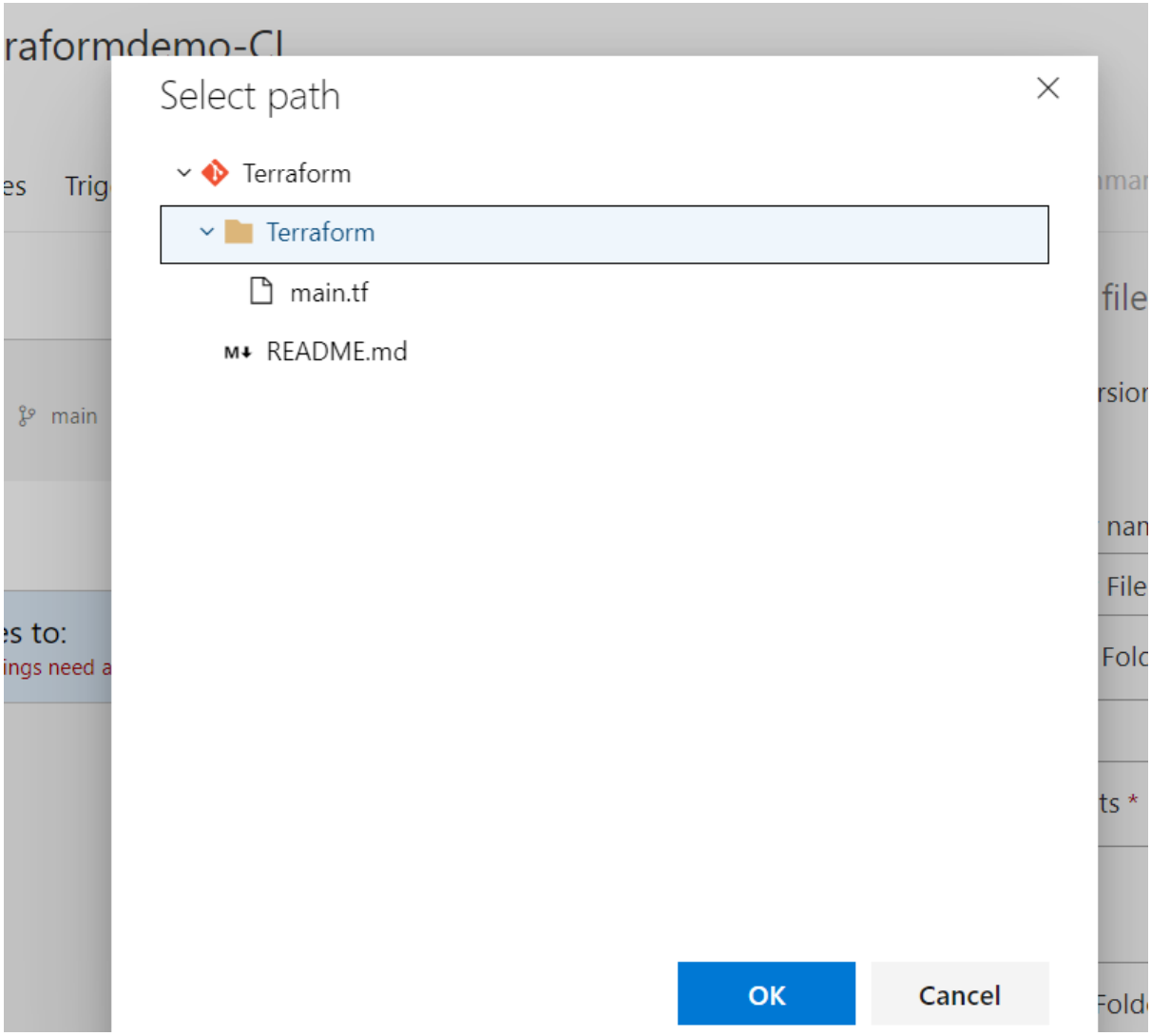




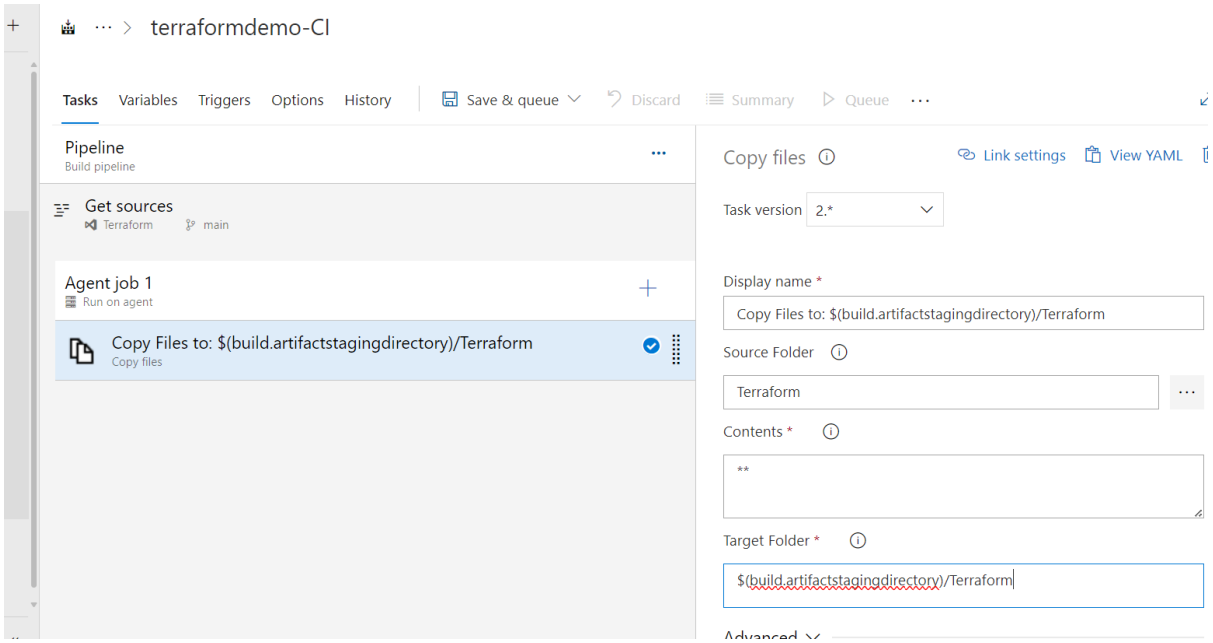
Add Task Copy files



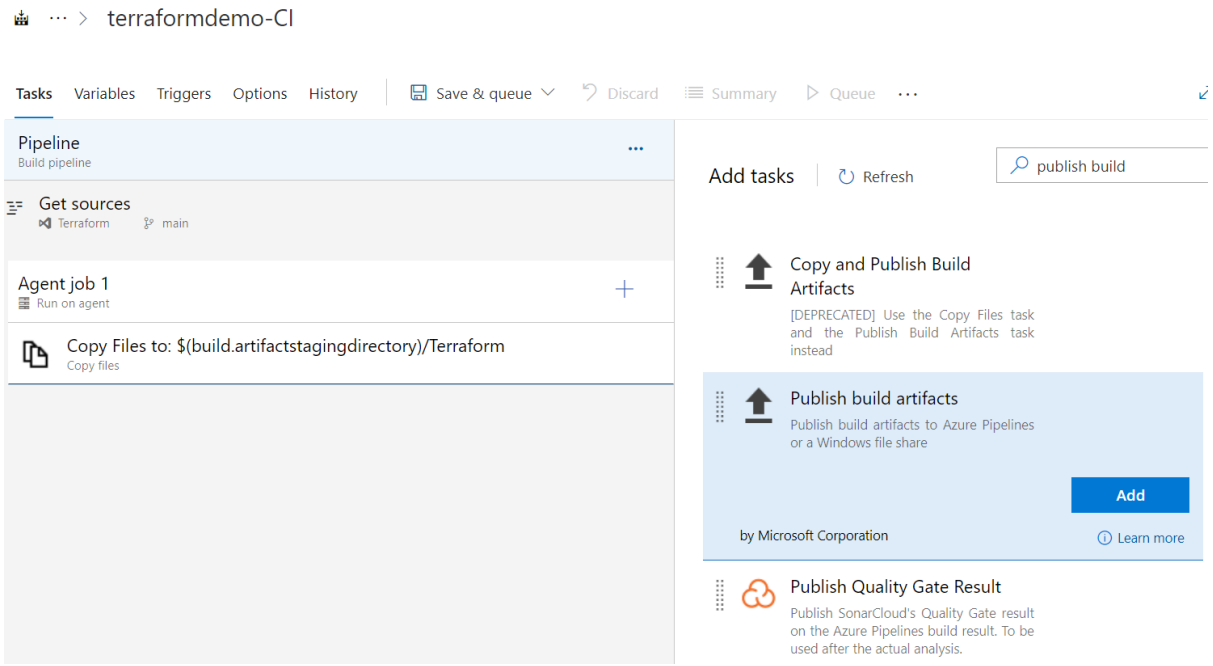
Select Source Folder- click on Terraform and ok.



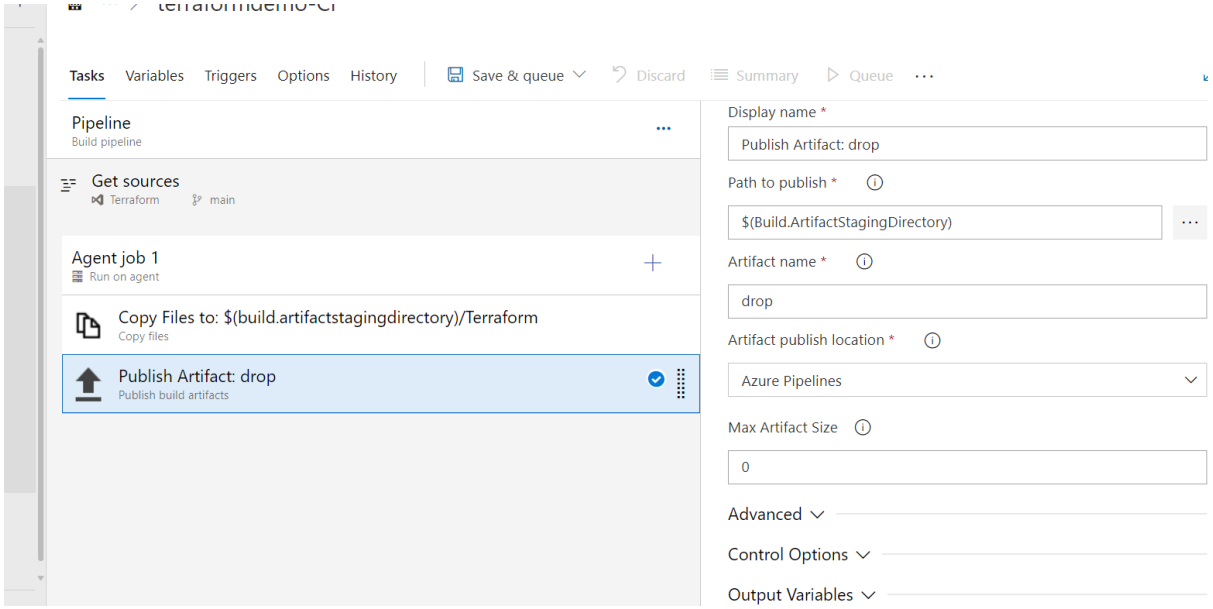
Select Target Folder as $(build.artifactstagingdirectory)/Terraform



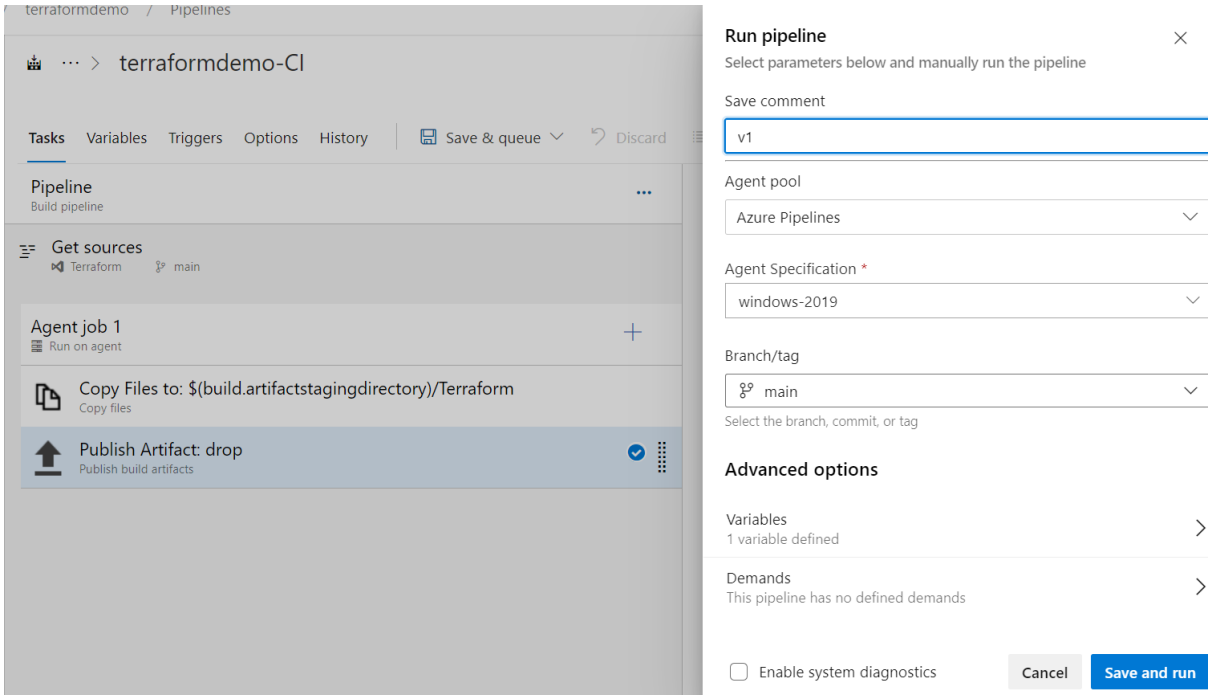
Add Task Publish Build Artifacts

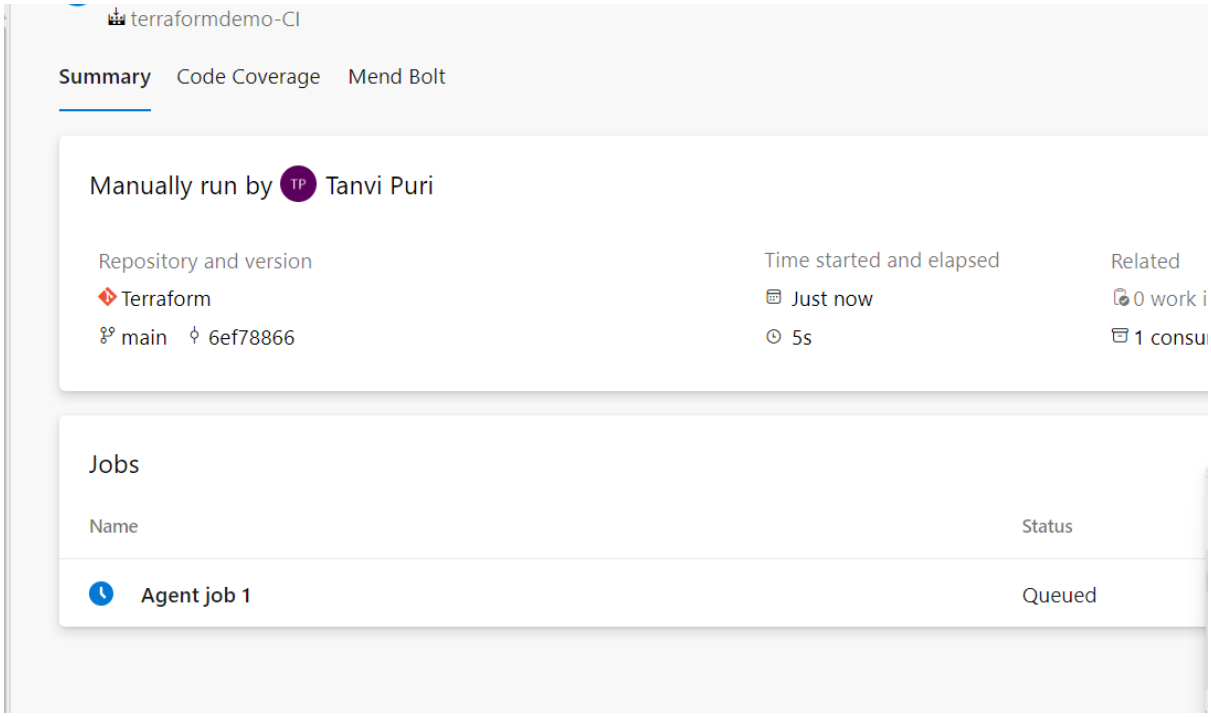


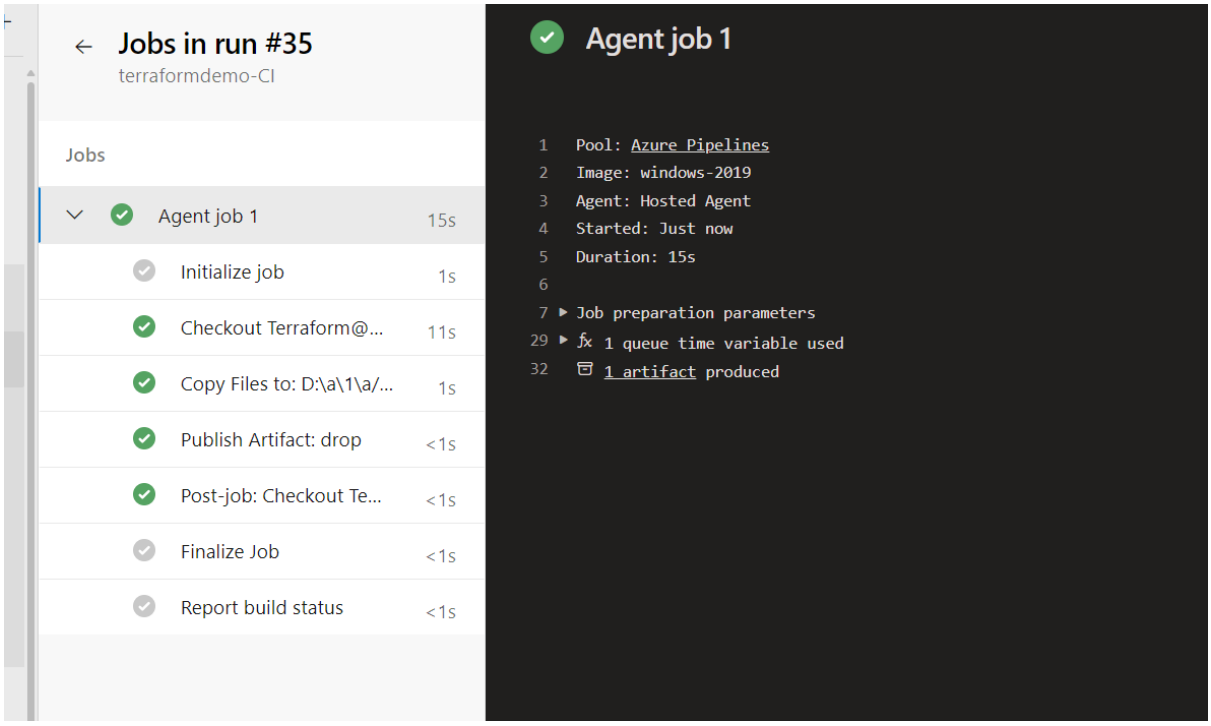
Go with default settings.



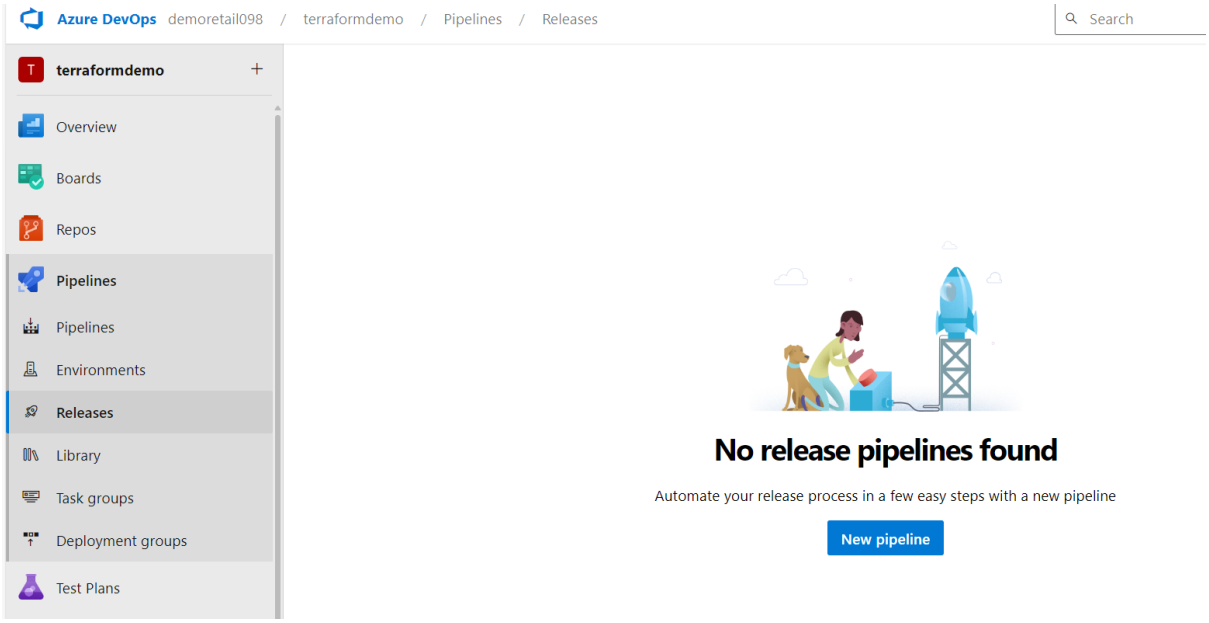
Run the Pipeline



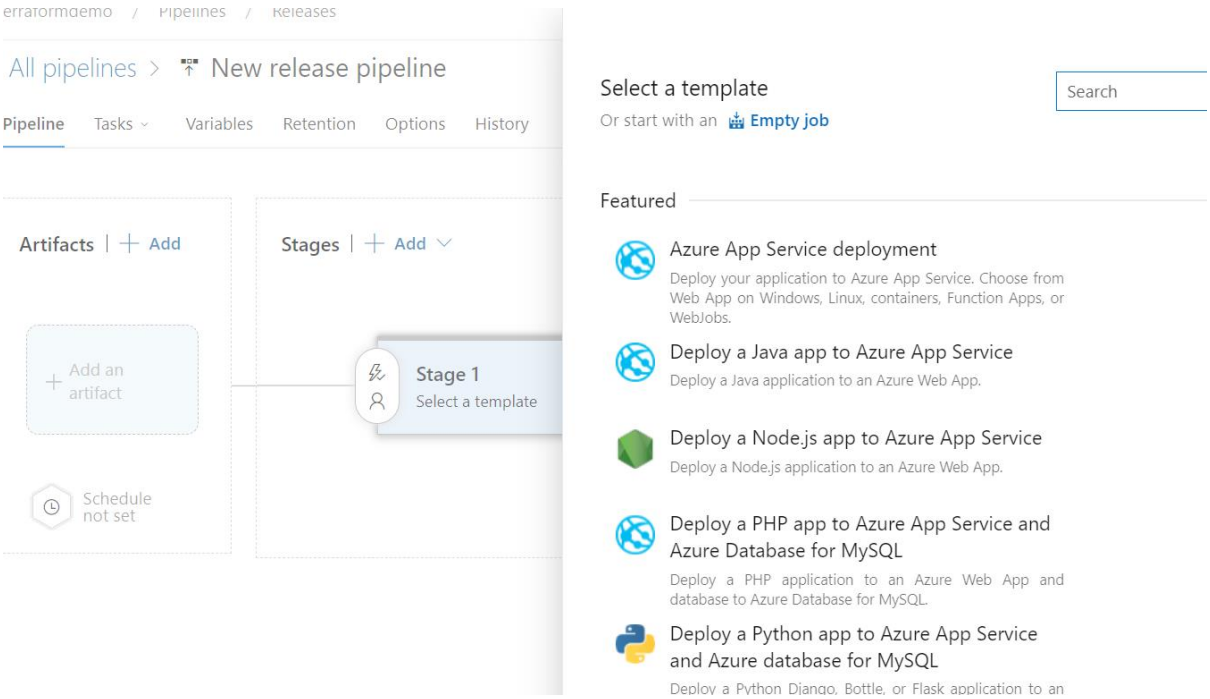




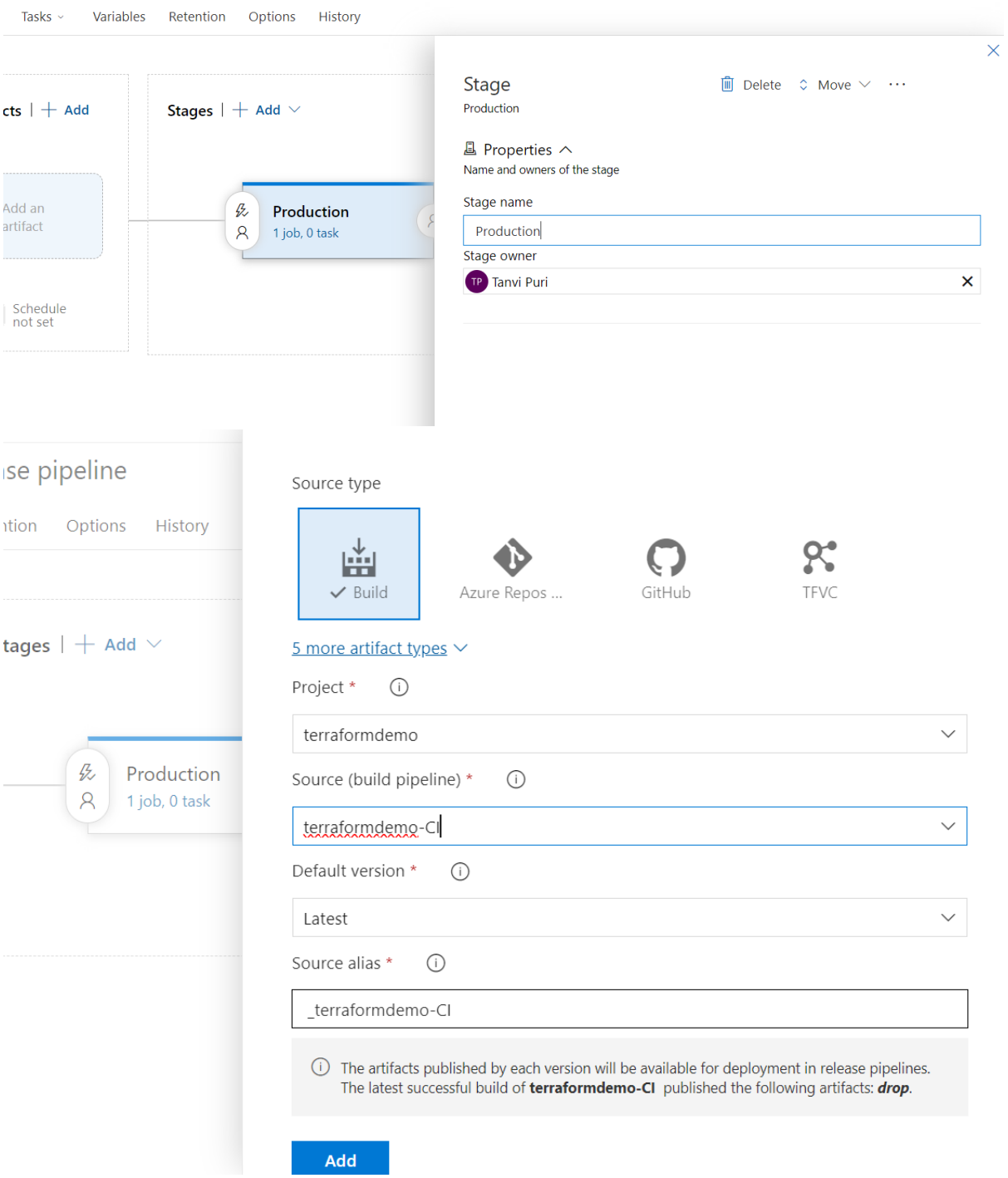
Create Release Pipeline



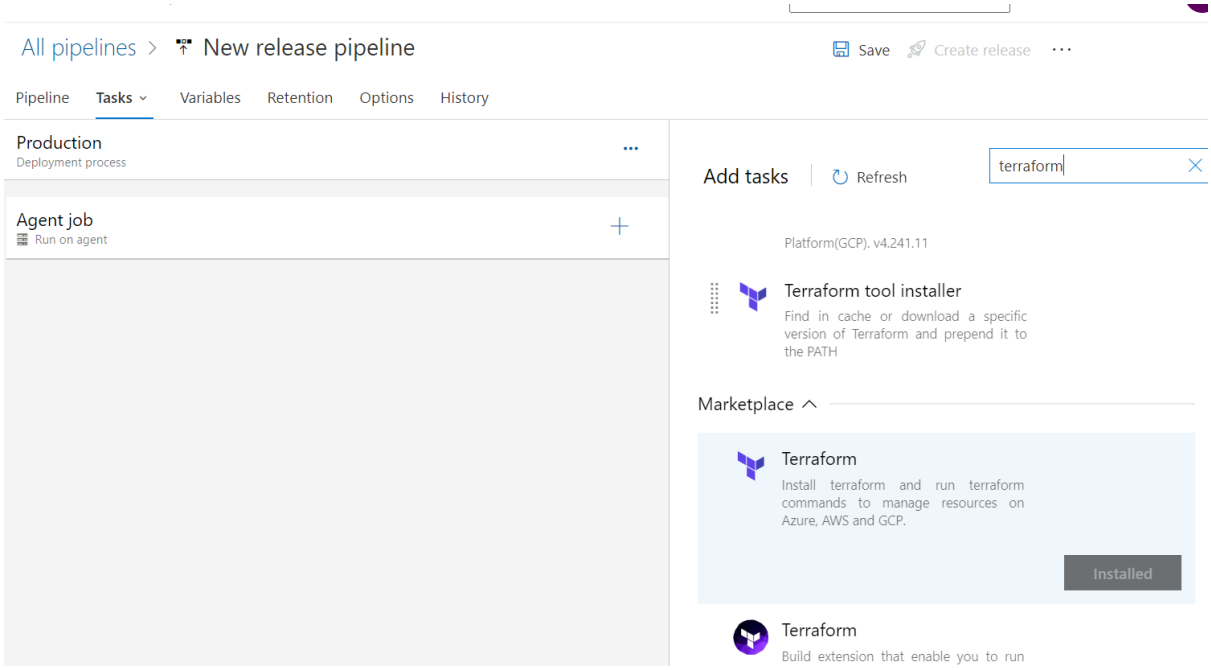
Click Empty Job



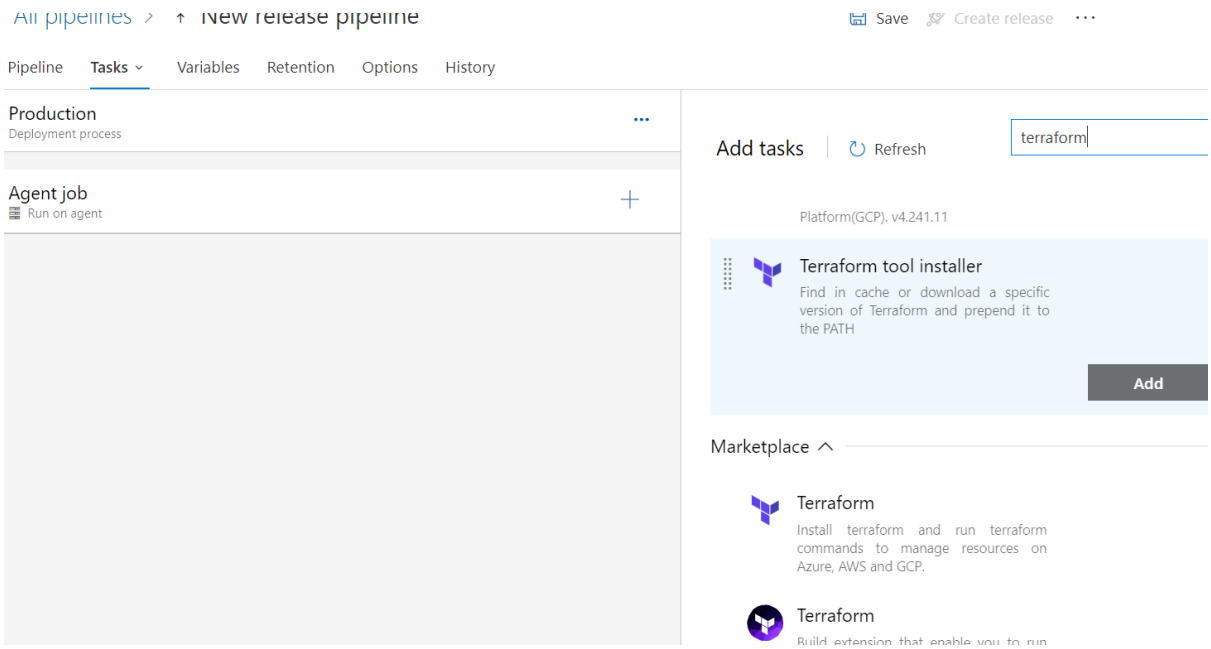
Change the name to Production. Save it. Add the Artifacts also.



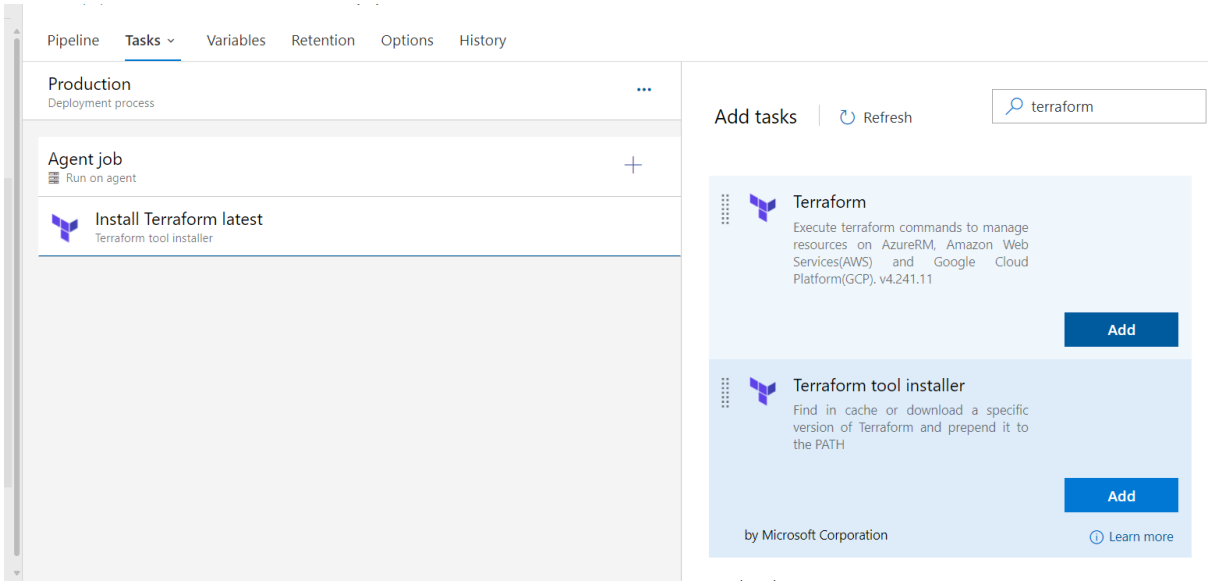
Add the Task Terraform , it will browse you to Marketplace, get it installed and then continue with your Azure Release Pipeline.



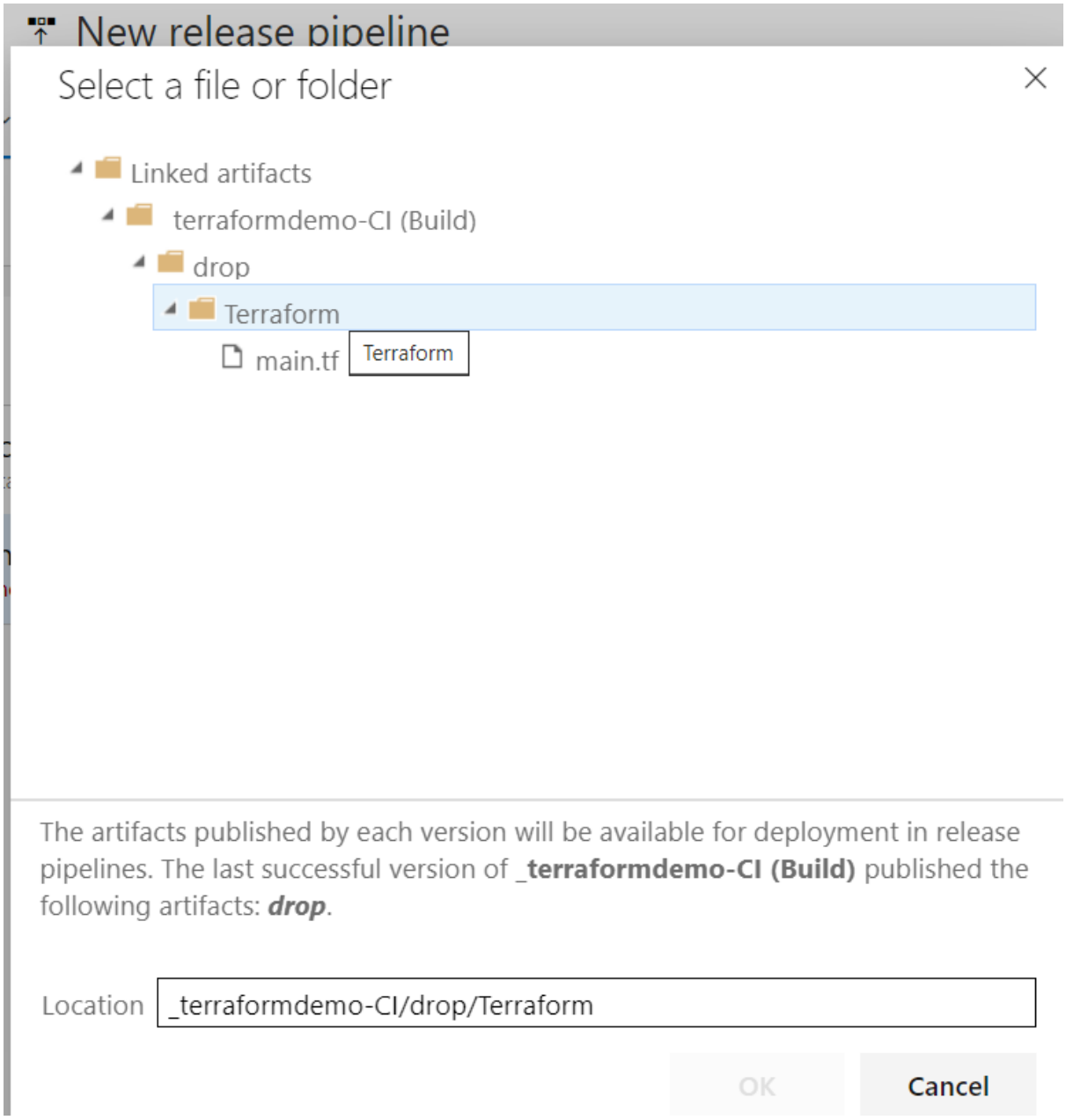
Add Task- Terraform Tool Installer and save it.

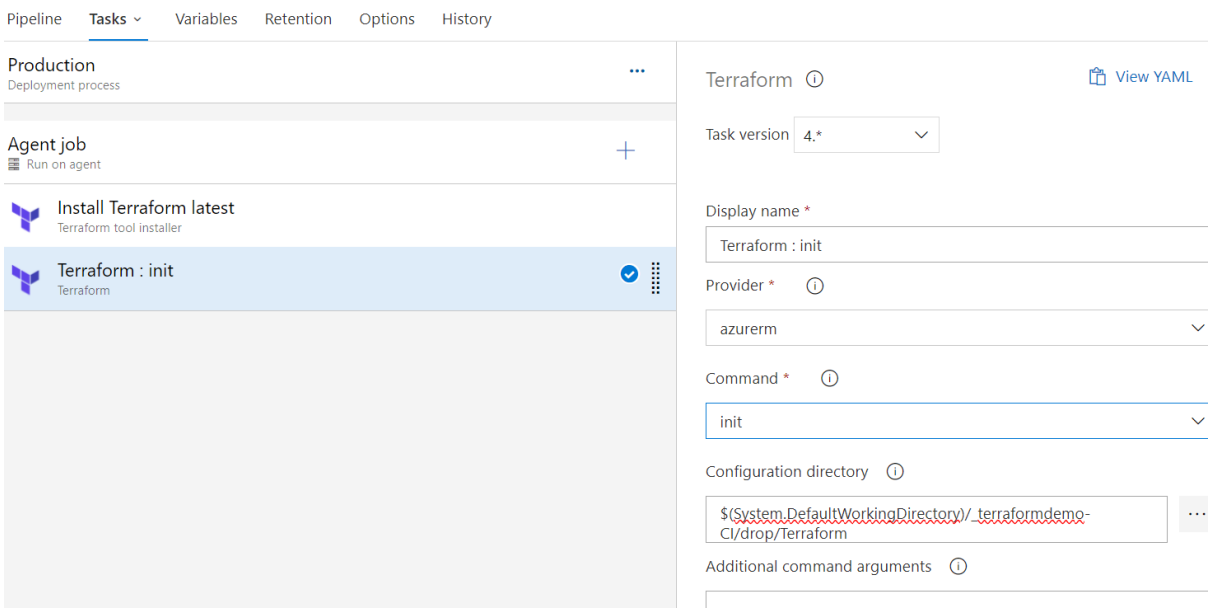


Add Task- Terraform command

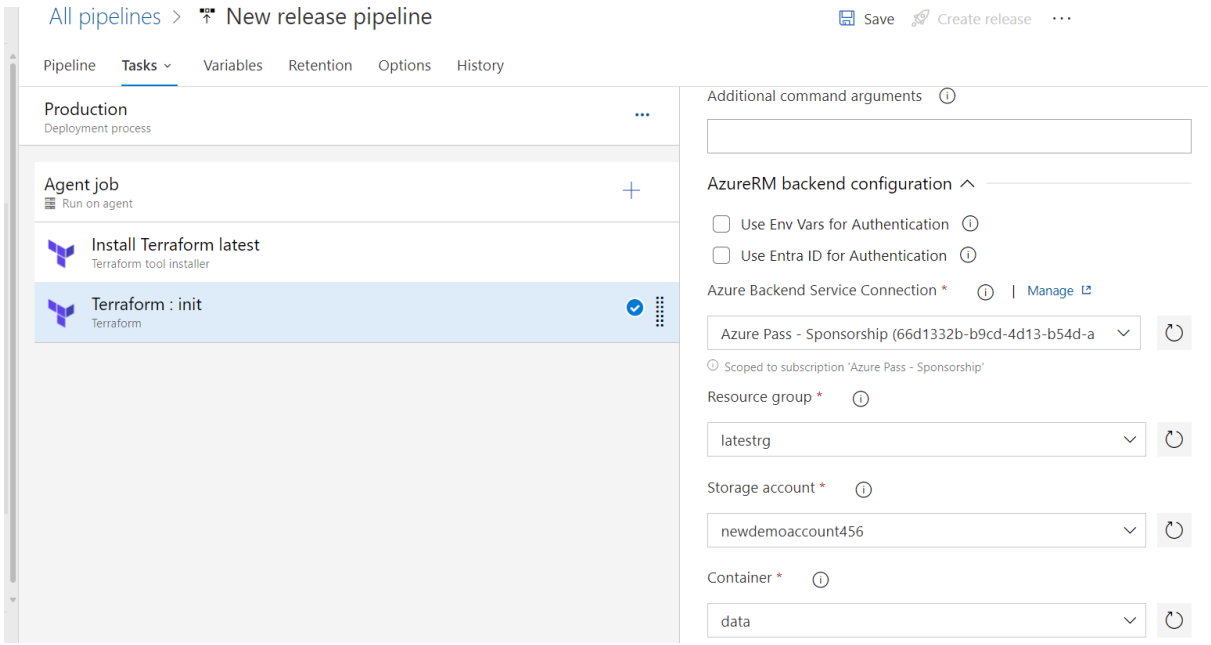


Select Terraform in Configuration Directory. Path should be like $(System.DefaultWorkingDirectory)/_terraformdemo-CI/drop/Terraform

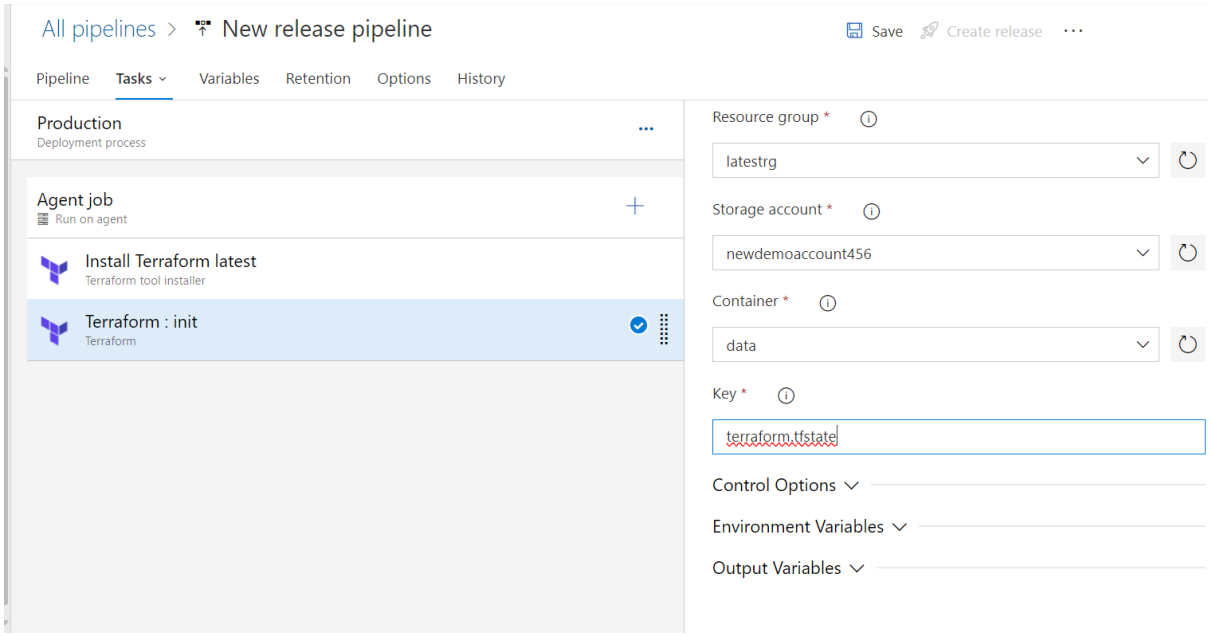

New release pipeline
Select a file or folder
Linked artifacts
terraformdemo-CI (Build)
drop
Terraform
main.tf
Terraform
The artifacts published by each version will be available for deployment in release pipelines. The last successful version of _terraformdemo-CI (Build) published the following artifacts: drop.
Location
_terraformdemo-CI/drop/Terraform
OK
Cancel



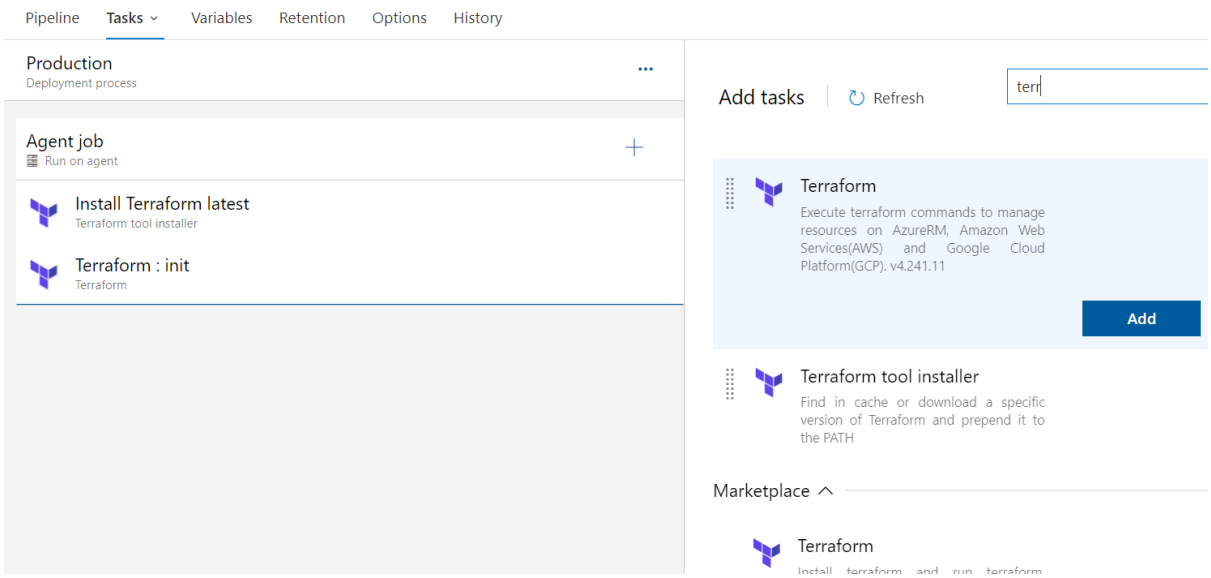
Make sure you have existing storage account in place on Azure Portal. Specify container name. After execution terraform state file will be stored as blob in that particular container. Also ensure you have existing Resource Group with name “template-grp” in North Europe Region as it was mentioned in Terraform script main.tf .

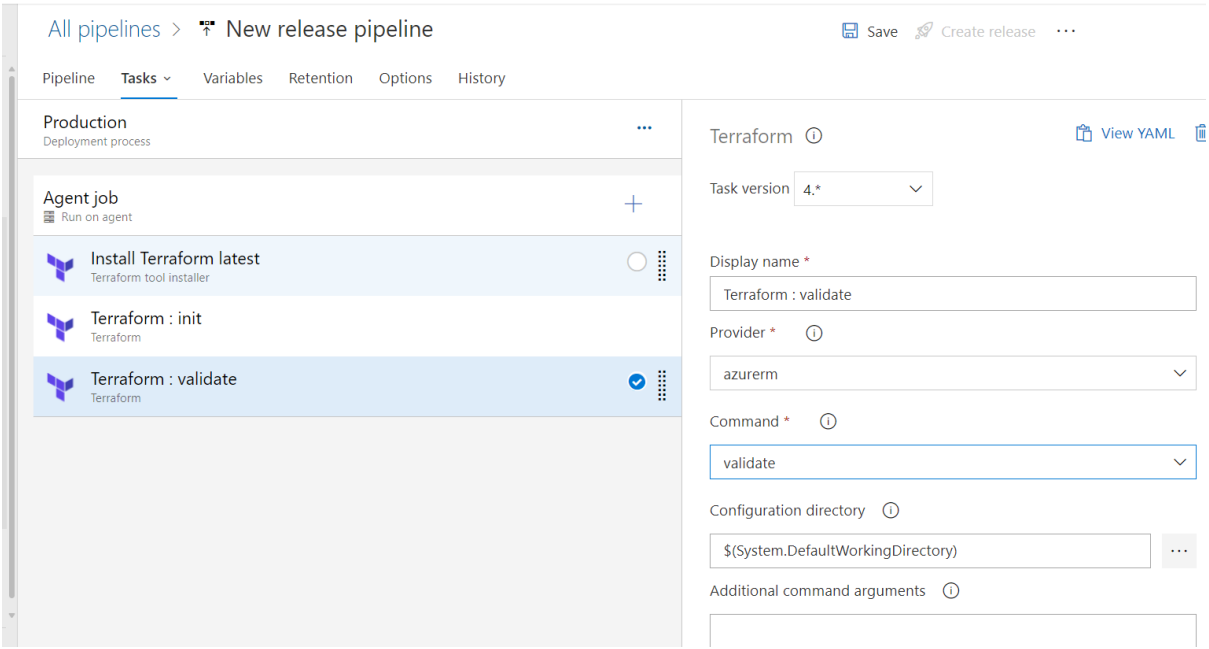


Specify the key name as terraform.tfstate [It is default name in Terraform]. Save it.

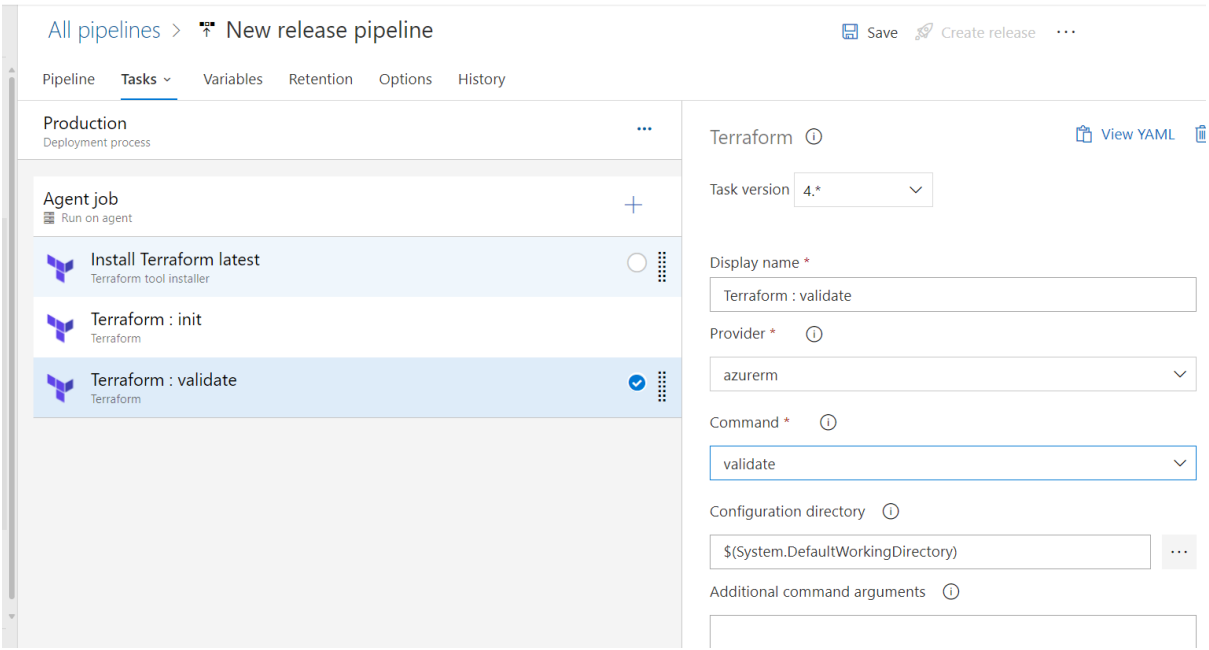


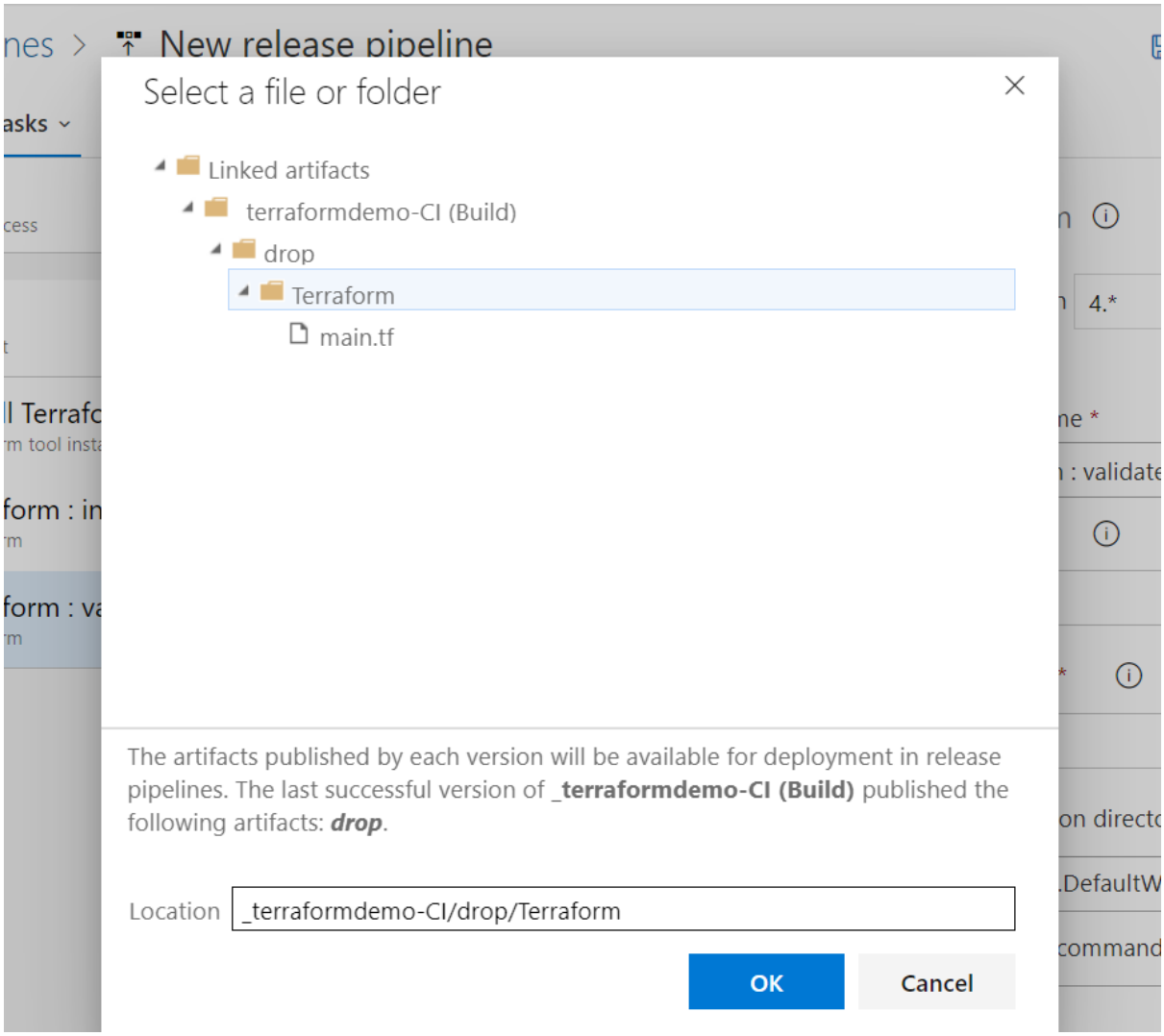
Add Task Terraform to execute Terraform Validate command.



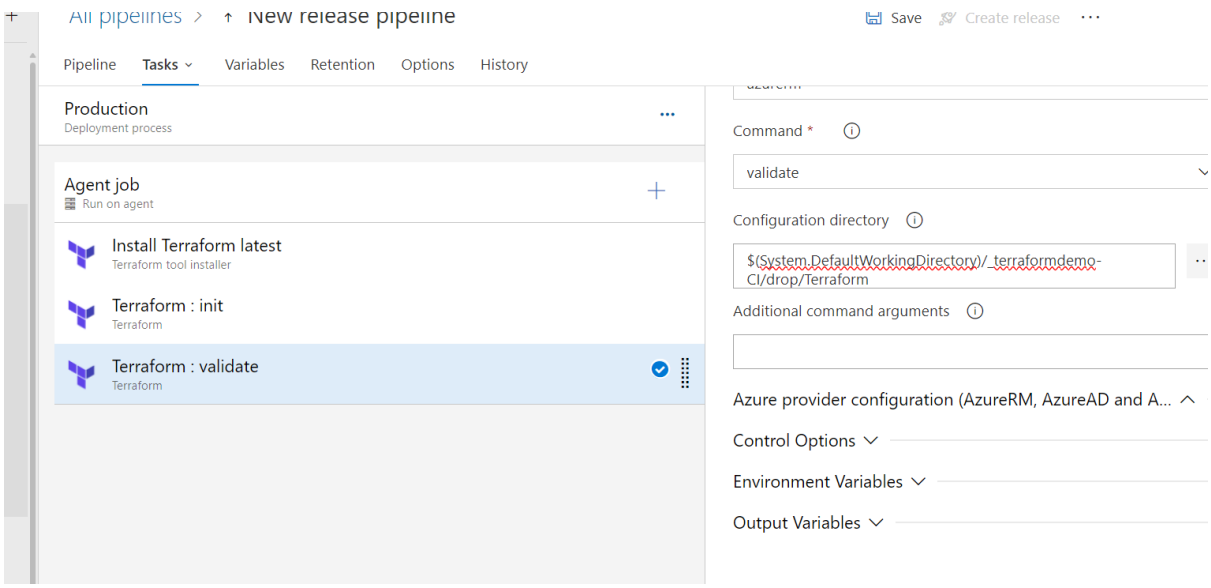


Select the correct path for configuration Directory. It should be $(System.DefaultWorkingDirectory)/_terraformdemo-CI/drop/Terraform

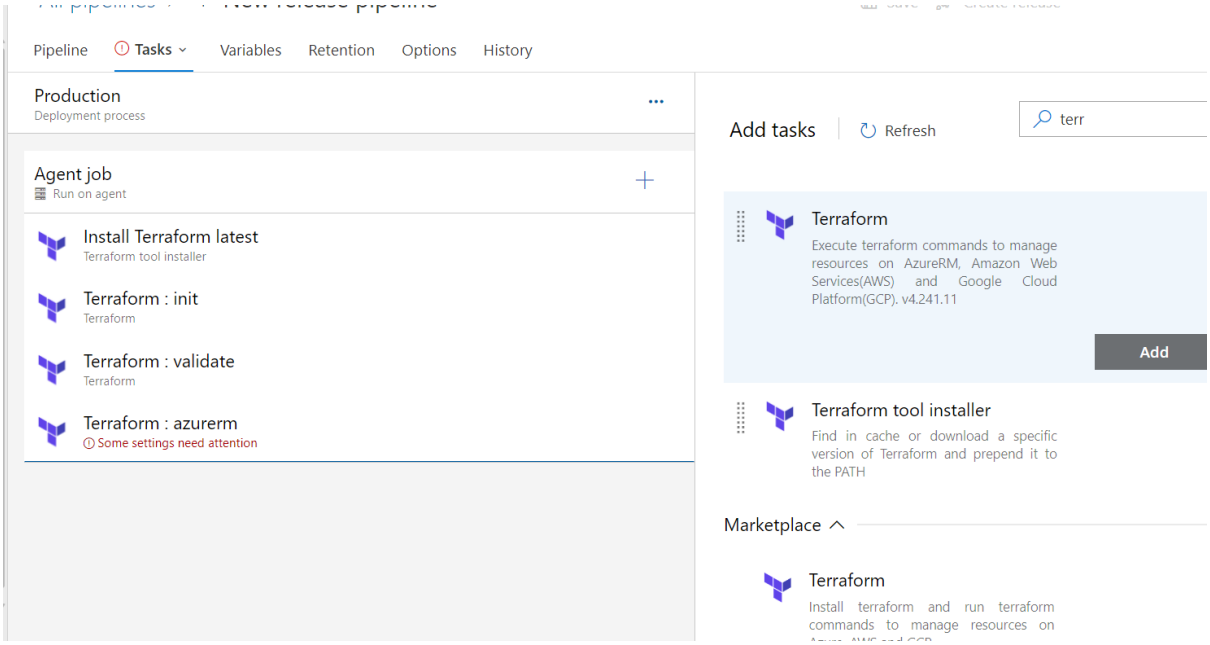




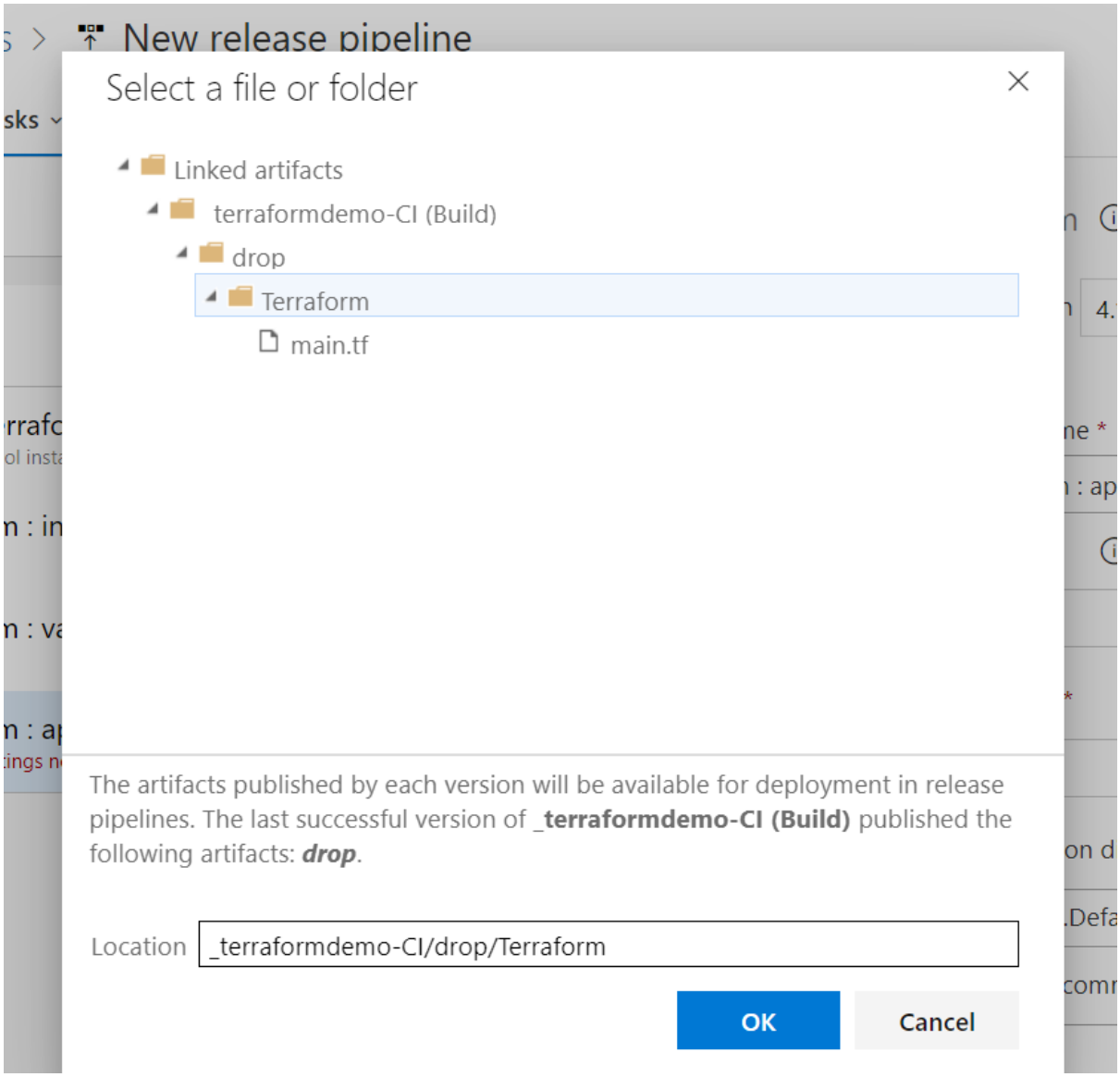
Keep other settings as default. Save it.

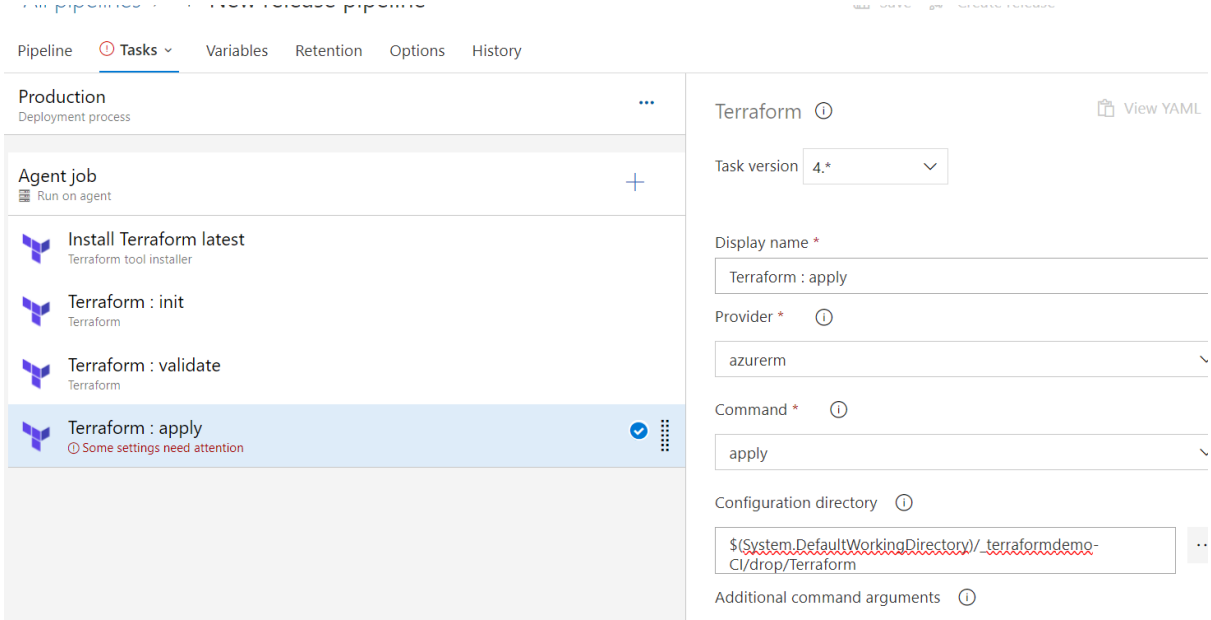


Add Task for Terraform command to execute Terraform Apply.

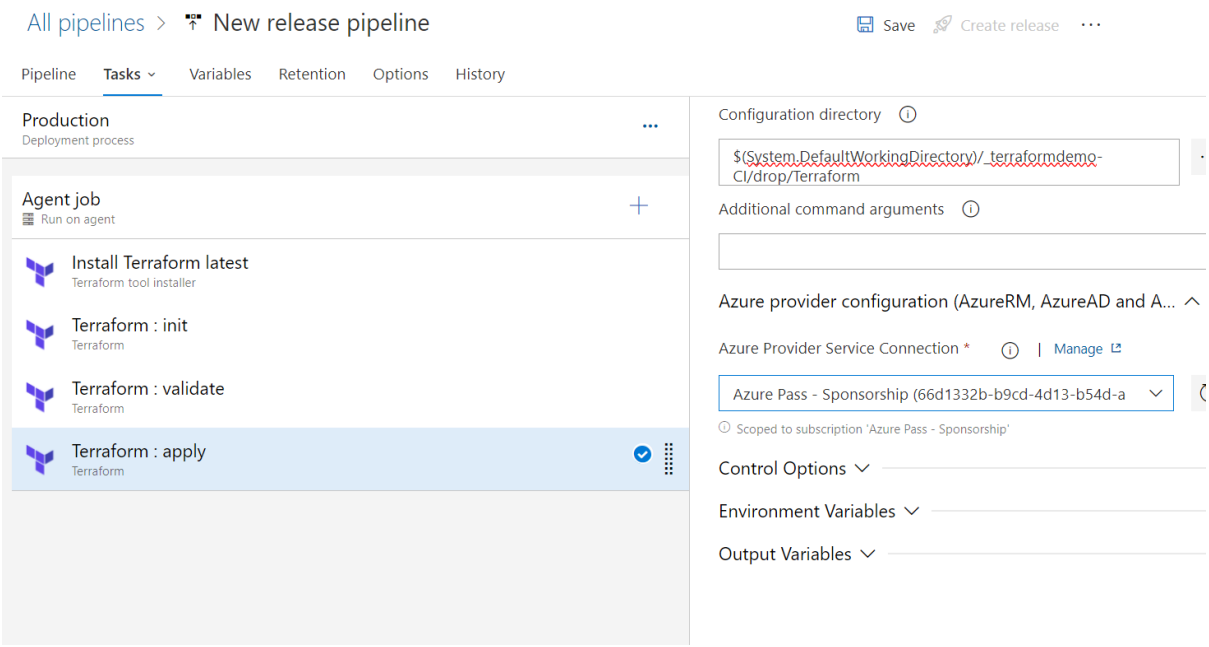


Select correct path – Terraform folder in configuration Directory. It should be: $(System.DefaultWorkingDirectory)/_terraformdemo-CI/drop/Terraform

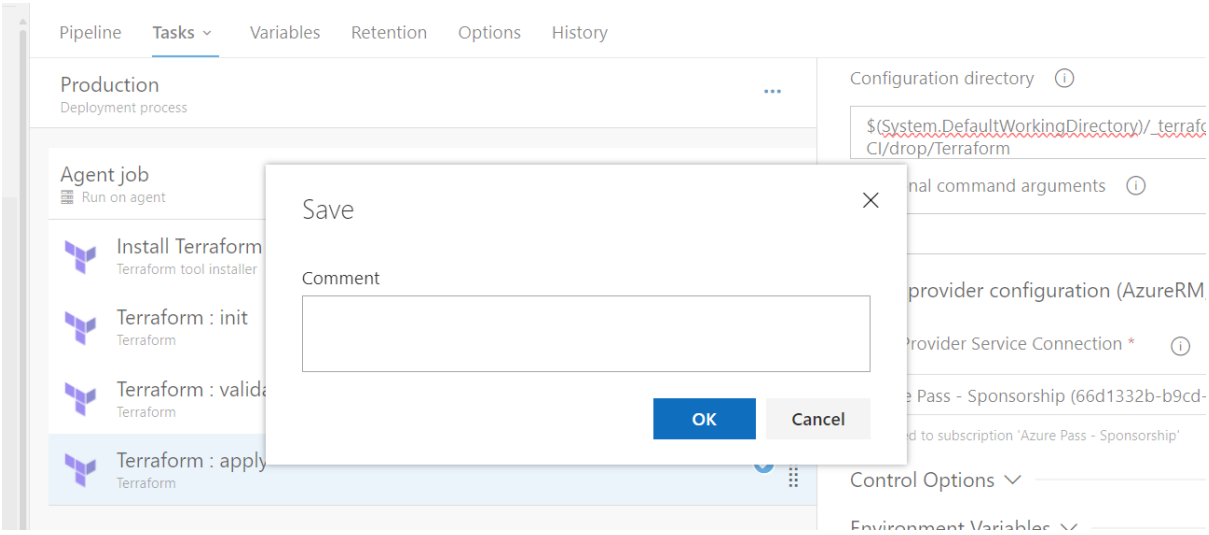

New release pipeline
Select a file or folder
Linked artifacts
terraformdemo-CI (Build)
drop
Terraform
main.tf
The artifacts published by each version will be available for deployment in release pipelines. The last successful version of _terraformdemo-CI (Build) published the following artifacts: drop.
Location
_terraformdemo-CI/drop/Terraform
OK
Cancel


Pipeline
Tasks
Variables
Retention
Options
History
Production
Deployment process
Agent job
Run on agent
Install Terraform latest
Terraform tool installer
Terraform : init
Terraform
Terraform : validate
Terraform
Terraform : apply
Some settings need attention
Terraform
View YAML
Task version
4.*
Display name *
Terraform : apply
Provider *
azurerm
Command *
apply
Configuration directory
$(System.DefaultWorkingDirectory)/_terraformdemo-CI/drop/Terraform
Additional command arguments

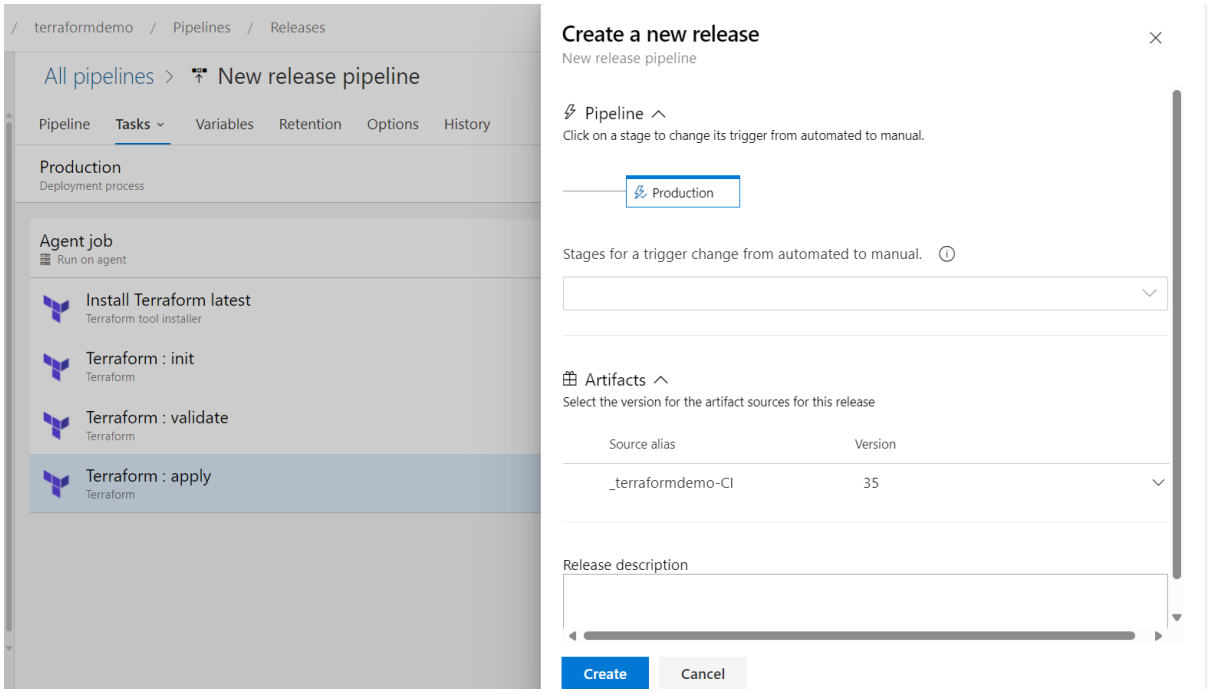
Select the Azure Provider Service Connection.

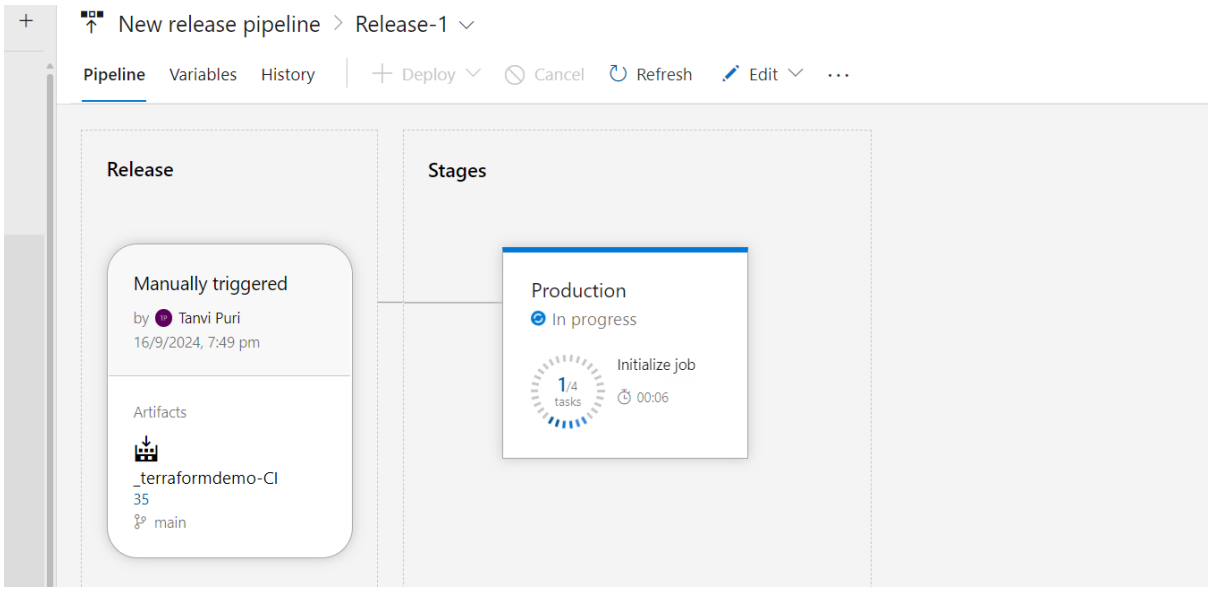


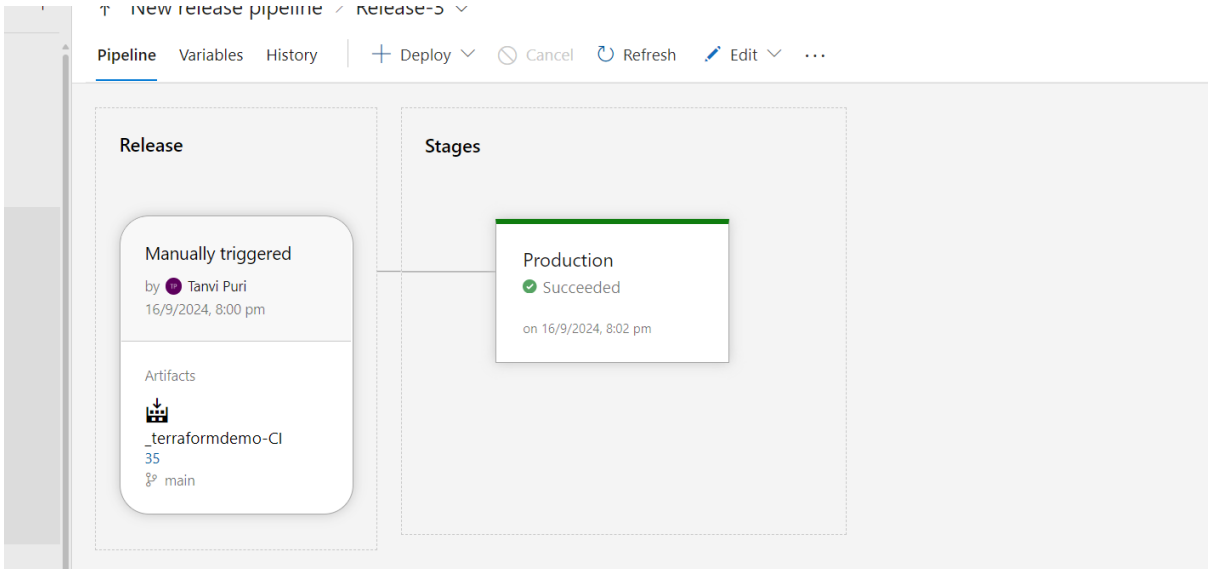
Click on Save



Create a Release.


terraformdemo / Pipelines / Releases
All pipelines > New release pipeline
Pipeline Tasks Variables Retention Options History
Production
Deployment process
Agent job
Run on agent
Install Terraform latest
Terraform tool installer
Terraform : init
Terraform
Terraform : validate
Terraform
Terraform : apply
Terraform
Create a new release
New release pipeline
Pipeline
Click on a stage to change its trigger from automated to manual.
Production
Stages for a trigger change from automated to manual.
Artifacts
Select the version for the artifact sources for this release
Source alias
Version
_terraformdemo-CI
35
Release description
Create
Cancel


New release pipeline > Release-1
Pipeline Variables History
Deploy Cancel Refresh Edit
Release
Manually triggered
by Tanvi Puri
16/9/2024, 7:49 pm
Artifacts
_terraformdemo-CI
35
main
Stages
Production
In progress
Initialize job
1/4
tasks
00:06



Verify provisioning of azure web app and app service plan on Azure Portal.