



Aniket Kumar Sinha for This is Learning

Posted on Jun 11 • Updated on Jun 13



Deploy Azure Infrastructure using Terraform Cloud

[#azure](#) [#terraform](#) [#devops](#) [#github](#)

Terraform Cloud (2 Part Series)

- 1 **Deploy Azure Infrastructure using Terraform Cloud**
- 2 Implementing Policy-as-Code to Terraform workflow usin...

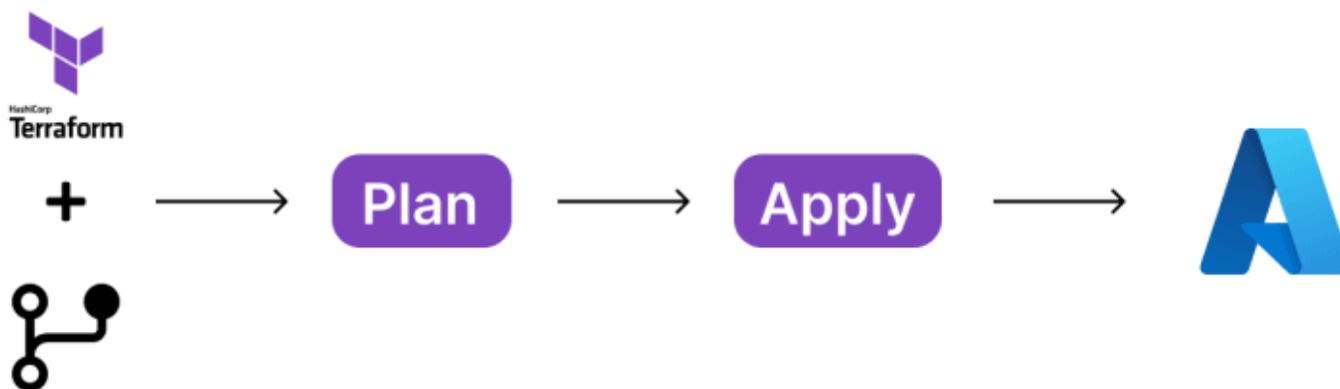
Note: You can get the whole code from this repository: [aniketkumarsinha/azure-terraform-infrastructure](https://github.com/aniketkumarsinha/azure-terraform-infrastructure)

What is Terraform?

[Terraform](#) is an infrastructure as code tool that lets you define both cloud and on-prem resources in human-readable configuration files that you can version, reuse, and share.

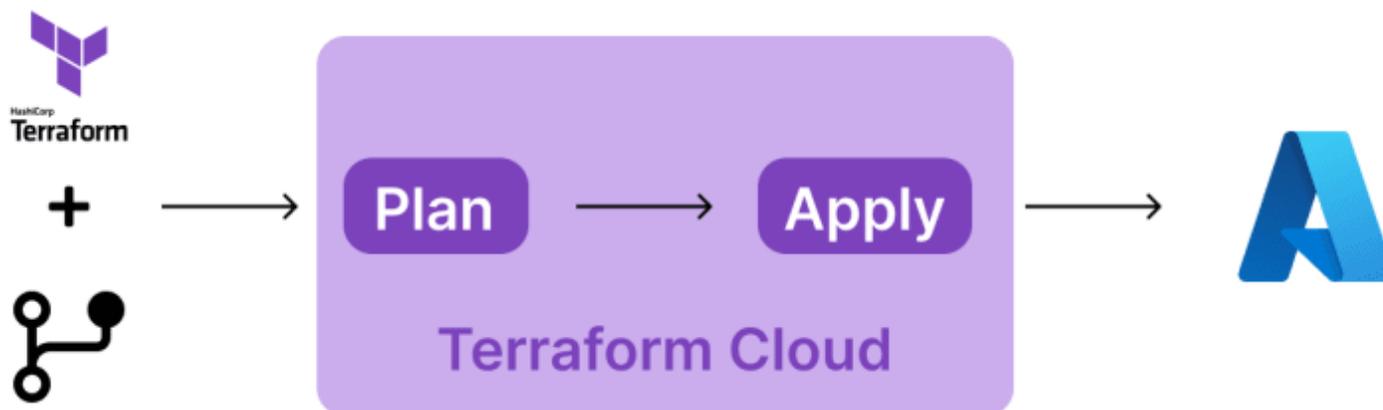
Terraform Flow

- First you have the terraform code.
- Then we have Terraform Plan phase. The terraform plan command creates an execution plan, which lets you preview the changes that Terraform plans to make to your infrastructure.
- Terraform Apply phase executes the actions proposed in a Terraform plan.
- And everything gets deployed over the CSP, here Azure.



What is Terraform Cloud?

[Terraform Cloud](#) is a managed service offering by HashiCorp that eliminates the need for unnecessary tooling and documentation for practitioners, teams, and organizations to use Terraform in production. It allows you to provision infrastructure in a remote environment that is optimized for the Terraform workflow.



In this blog, we will be creating Azure Infrastructure using Terraform and will be deploying it over to Azure using Terraform Cloud.

Infrastructure Code

We are creating a VM and resources related to it.

```
1  # Configure Azure Provider
2  terraform {
3    required_providers {
4      azurearm = {
5        source = "hashicorp/azurearm"
6        version = ">= 3.59.0"
7      }
8    }
9    required_version = ">= 0.14.9"
10 }
11
12 provider "azurearm" {
13   features {}
14
15   skip_provider_registration = "true"
16 }
17
18 variable "prefix" {
19   default = "terraform"
20 }
21
22 resource "azurearm_resource_group" "rg" {
23   name      = "${var.prefix}-ResourceGroup"
24   location = "Central India"
25 }
26
27 resource "azurearm_virtual_network" "vnet" {
28   name            = "${var.prefix}-VNet"
29   address_space  = ["10.0.0.0/16"]
30   location        = azurearm_resource_group.rg.location
```

```
31   resource_group_name = azurerm_resource_group.rg.name
32 }
33
34 resource "azurerm_subnet" "internal" {
35   name                = "${var.prefix}-internal"
36   resource_group_name = azurerm_resource_group.rg.name
37   virtual_network_name = azurerm_virtual_network.vnet.name
38   address_prefixes     = ["10.0.2.0/24"]
39 }
40
41 resource "azurerm_network_interface" "nic" {
42   name                = "${var.prefix}-NIC"
43   location            = azurerm_resource_group.rg.location
44   resource_group_name = azurerm_resource_group.rg.name
45
46   ip_configuration {
47     name                = "tfconfiguration1"
48     subnet_id          = azurerm_subnet.internal.id
49     private_ip_address_allocation = "Dynamic"
50   }
51 }
52
53 resource "azurerm_virtual_machine" "vm" {
54   name                = "${var.prefix}-vm"
55   location            = azurerm_resource_group.rg.location
56   resource_group_name = azurerm_resource_group.rg.name
57   network_interface_ids = [azurerm_network_interface.nic.id]
58   vm_size             = "Standard_DS1_v2"
59
60   storage_image_reference {
61     publisher = "Canonical"
62     offer     = "UbuntuServer"
63     sku       = "16.04-LTS"
```

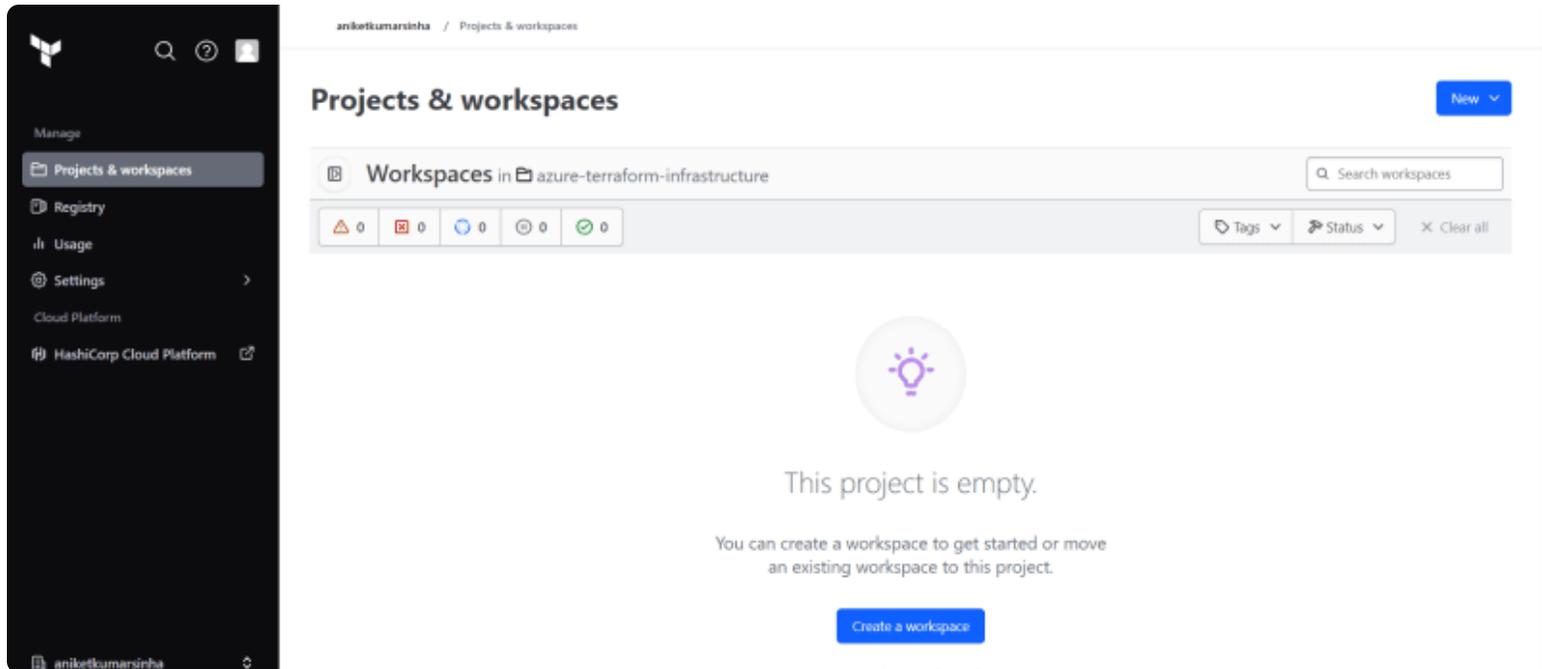
```
64     version    = "latest"
65   }
66   storage_os_disk {
67     name          = "myosdisk1"
68     caching       = "ReadWrite"
69     create_option = "FromImage"
70     managed_disk_type = "Standard_LRS"
71   }
72   os_profile {
73     computer_name = "hostname"
74     admin_username = "tfadmin"
75     admin_password = "Password1234!"
76   }
77   os_profile_linux_config {
78     disable_password_authentication = false
79   }
80   tags = {
81     environment = "staging"
82   }
83 }
```

main.tf hosted with ❤️ by GitHub

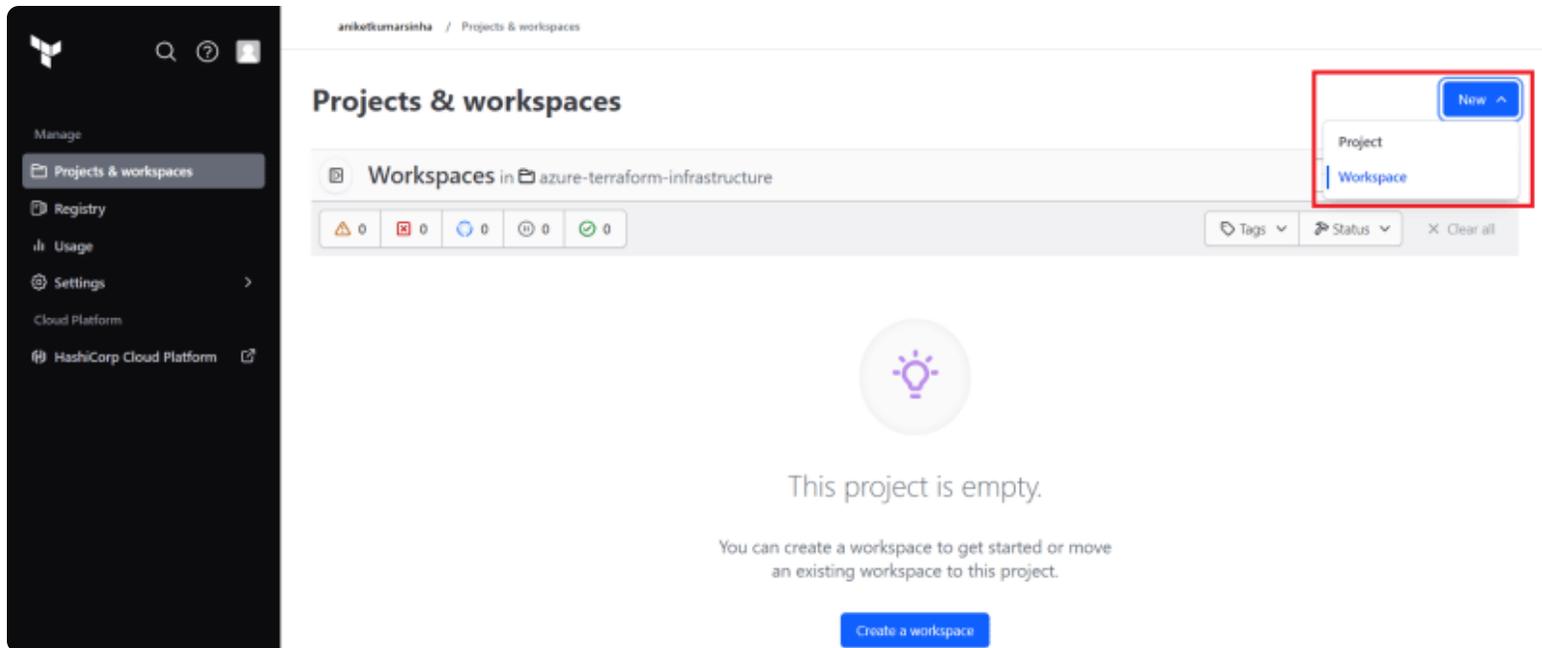
[view raw](#)

Setting up Terraform Cloud

1. Create Terraform Cloud Account - [Terraform Cloud](#)
2. Create a Project in Terraform Cloud:



3. Create a Workspace.



- Choose Version control workflow

aniketkumarsinha / Projects & workspaces / New Workspace

Create a new Workspace

Workspaces determine how Terraform Cloud organizes infrastructure. A workspace contains your Terraform configuration (infrastructure as code), shared variable values, your current and historical Terraform state, and run logs. [Learn more](#) about workspaces in Terraform Cloud.

- 1 Choose Type
- 2 Connect to VCS
- 3 Choose a repository
- 4 Configure settings

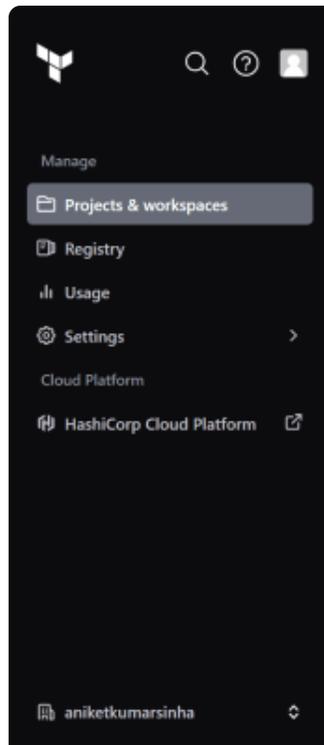
Choose your workflow

- Version control workflow** Most common
Store your Terraform configuration in a git repository, and trigger runs based on pull requests and merges. [Learn More](#)
- CLI-driven workflow**
Trigger remote Terraform runs from your local command line. [Learn More](#)
- API-driven workflow**
A more advanced option. Integrate Terraform into a larger pipeline using the Terraform API. [Learn More](#)

aniketkumarsinha

© 2023 HashiCorp, Inc. [Support](#) [Terms](#) [Privacy](#) [Security](#)

- Connect to a version control provider



Create a new Workspace

Workspaces determine how Terraform Cloud organizes infrastructure. A workspace contains your Terraform configuration (infrastructure as code), shared variable values, your current and historical Terraform state, and run logs. [Learn more](#) about workspaces in Terraform Cloud.

1 Choose Type 2 **Connect to VCS** 3 Choose a repository 4 Configure settings

Connect to a version control provider

Choose the version control provider that hosts the Terraform configuration for this workspace.



[Connect to a different VCS](#)

- Choose your Azure Infrastructure repository from your repository list. You can check the advance settings if you want to configure the workflow.

aniketkumarsinha / Projects & workspaces / New Workspace

Create a new Workspace

Workspaces determine how Terraform Cloud organizes infrastructure. A workspace contains your Terraform configuration (infrastructure as code), shared variable values, your current and historical Terraform state, and run logs. [Learn more](#) about workspaces in Terraform Cloud.

Choose Type ✓ Connect to VCS ✓ Choose a repository ✓ **Configure settings** 4

Configure settings

Workspace Name

The name of your workspace is unique and used in tools, routing, and UI. Dashes, underscores, and alphanumeric characters are permitted. [Learn more about naming workspaces](#)

Project

Every workspace must belong to a single project. Projects must be named uniquely within an organization. Workspaces may be moved between projects at any time from the workspace list or settings. [Learn more about projects](#)

Description

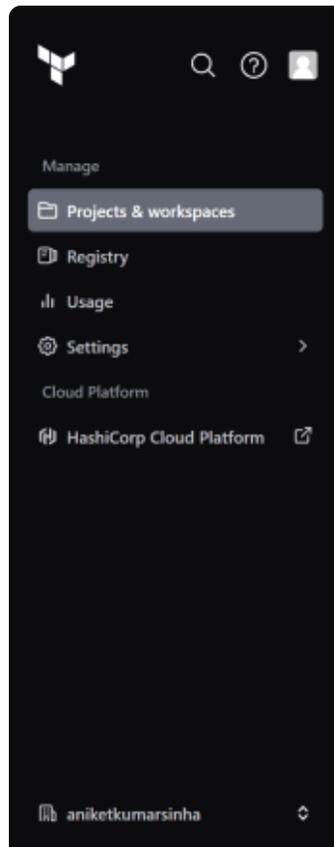
Optional

[Advanced options](#)

Create workspace

© 2023 HashiCorp, Inc. [Support](#) [Terms](#) [Privacy](#) [Security](#)

- Tap on **Create Workspace** button to create your workspace in terraform cloud Or Start new run from workspace overview page.
4. You can directly start your plan phase.



Workspace created!

[Go to workspace overview](#)

Workspace ✓ azure-terraform-infrastructure

Next step: Configure Terraform variables

No variables found

Your configuration does not contain input variable definitions that need values entered.

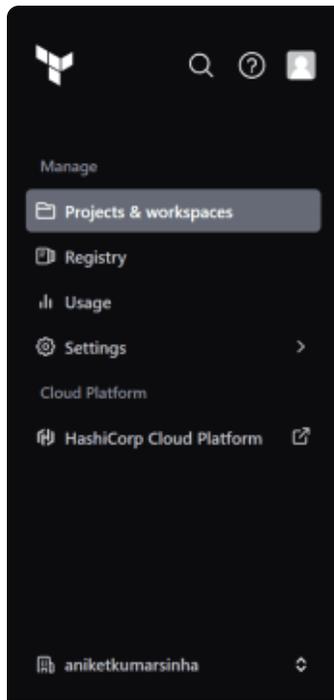
[Continue to workspace overview →](#)

Start your first plan

After you configure any required input variables, start your first plan.

[Start new plan](#)

5. You can verify in your projects page that your workspace has been created.



aniketkumarsinha / Projects & workspaces

Projects & workspaces

New

Workspaces in azure-terraform-infrastructure

Search workspaces

0 0 0 0 0

Tags Status Clear all

Workspace Name	Run Status	Repo	Latest Change
azure-terraform-infrastructure		aniketkumarsinha/azure-terr...	4 minutes ago

1 - 1 of 1

Previous 1 Next

© 2023 HashiCorp, Inc. [Support](#) [Terms](#) [Privacy](#) [Security](#)

6. Open your workspace to start new run.

aniketkumarsinha / Projects & workspaces / azure-terraform-infrastructure / Overview

azure-terraform-infrastructure

ID: ws-TqD0IglTnAQL7sYBL

No workspace description available. [Add workspace description.](#)

Unlocked

Configuration uploaded successfully

Next step: configure variables

Configure any required variables (such as access keys or configuration values) before starting a run.

[Configure variables](#)

Not configuring variables?

If your configuration doesn't require setting variable values, you can start your first plan now.

[Start new plan](#)

Resources **0** Outputs **0**

This workspace does not have any resources. [Learn about resources.](#)

README.md

Provide additional context by adding a README.md file to the branch and working directory of this workspace.

Actions

- Start new run
- Lock workspace
- Execution mode: Remote
- Auto apply: Off

Project: azure-terraform-infrastructure

Metrics

Metrics will appear once your next run is applied.

Tags (0)

Add a tag

Tags have not been added to this workspace.

Run triggers

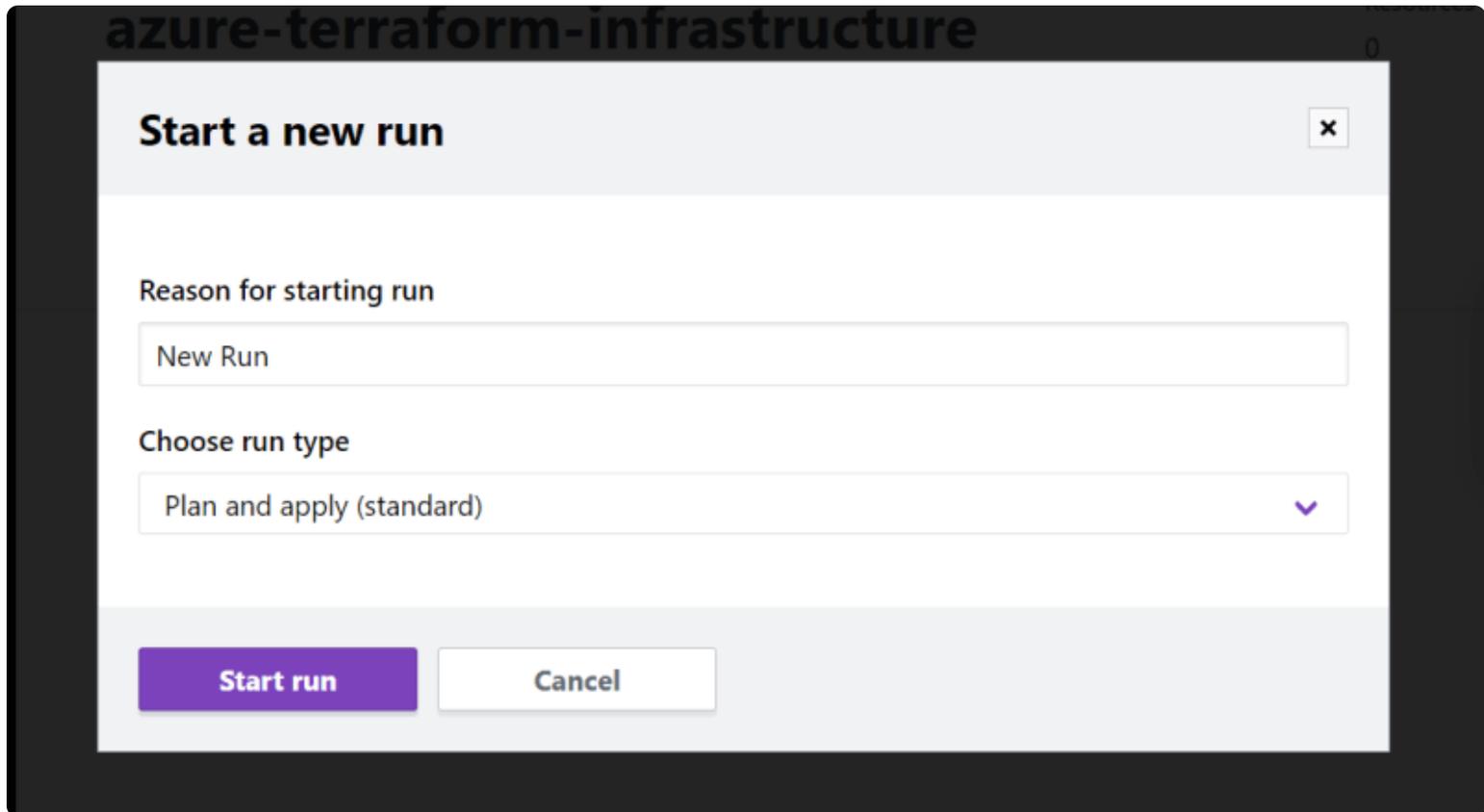
No source workspaces have been selected. Adding run triggers will allow runs to queue automatically in this workspace.

Contributors (1)

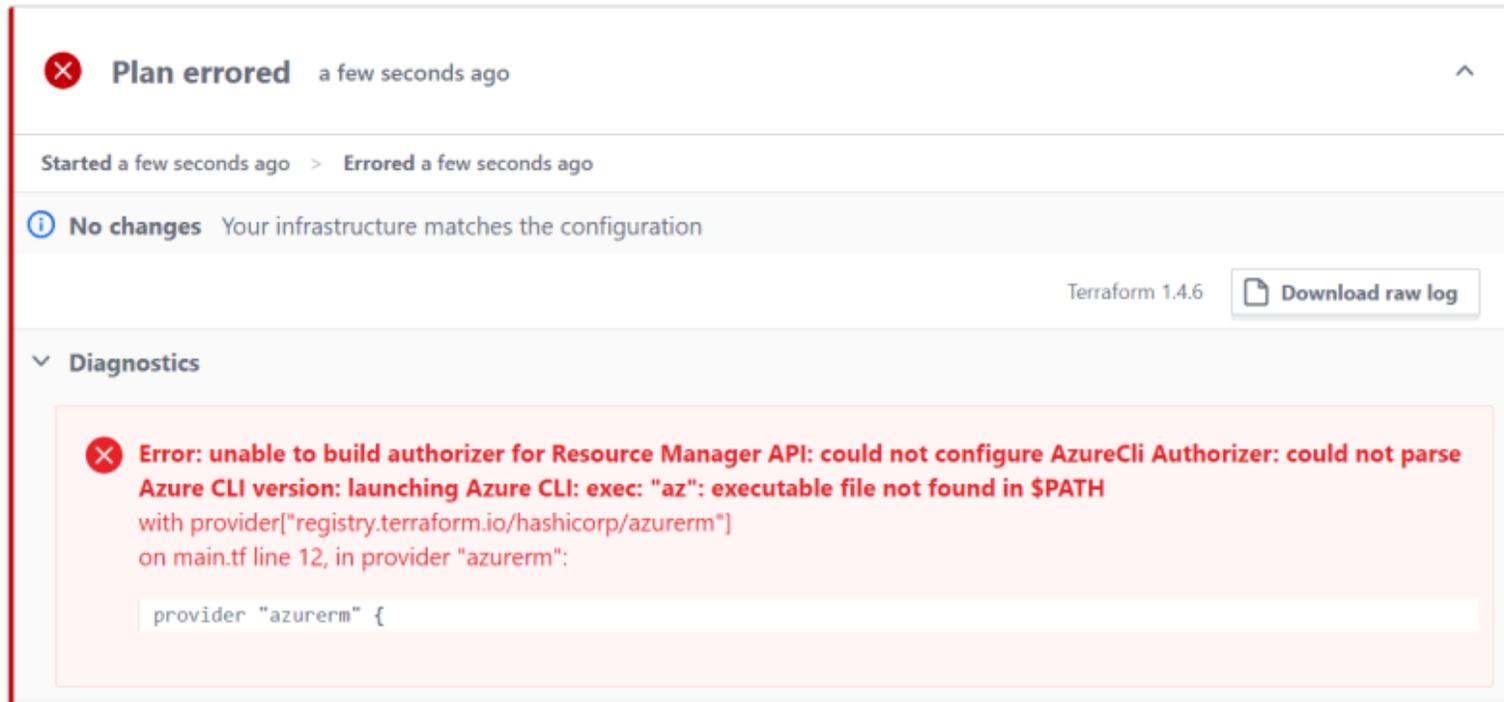
Team management is a paid feature. Free organizations only include an owners team that can have up to five members. Learn more about Terraform Cloud pricing.

© 2023 HashiCorp, Inc. [Support](#) [Terms](#) [Privacy](#) [Security](#)

7. Choose your run type and start run.



8. Ouch!! Errors!! We are getting error in provider phase. The reason behind this is that we have authorized our infrastructure to connect and write over our Azure. We will be creating an App provide all the necessary details to our infrastructure.



The screenshot shows the Terraform Cloud interface. At the top, a red 'X' icon indicates a 'Plan errored' status, with the text 'a few seconds ago'. Below this, a progress bar shows 'Started a few seconds ago' followed by 'Errored a few seconds ago'. A message with an information icon states 'No changes Your infrastructure matches the configuration'. On the right, it says 'Terraform 1.4.6' and has a 'Download raw log' button. The 'Diagnostics' section is expanded, showing a red error message: 'Error: unable to build authorizer for Resource Manager API: could not configure AzureCli Authorizer: could not parse Azure CLI version: launching Azure CLI: exec: "az": executable file not found in \$PATH with provider["registry.terraform.io/hashicorp/azurerem"] on main.tf line 12, in provider "azurerem":'. Below the error message, a code block shows the start of a provider configuration: 'provider "azurerem" {'.

Authorizing Terraform Infrastructure to write over Azure

- We are creating an App over Azure to authorize using Client Id, Client Secret and Tenant Id. So in your [Azure Portal](#) move into Azure Active Directory, and open App registrations from left pane, And add a new registration.

Home > Student Ambassadors | App registrations

Azure Active Directory

+ New registration | Endpoints | Troubleshooting | Refresh | Download | Preview features | Got feedback?

Manage

- Users
- Groups
- External Identities
- Roles and administrators
- Administrative units
- Delegated admin partners
- Enterprise applications
- Devices
- App registrations**
- Identity Governance
- Application proxy
- Custom security attributes (Preview)

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. [Learn more](#)

All applications | **Owned applications** | Deleted applications

Start typing a display name or application (client) ID to filter these r... | Add filters

2 applications found

Display name	Application (client) ID	Created on	Certificates & secrets
automationrclab3_pHIBXebcPwK7MtAF3+PoreLu20n4GnmSD...	9c8ac701-b5d6-4e67-a5ab-b7367...	3/17/2022	Expired
rc-testing	fdb3bf05-7771-4994-8a4b-1e027...	5/18/2023	Current

- Add your App details and register.

Home > Student Ambassadors | App registrations >

Register an application

Name

The user-facing display name for this application (this can be changed later).

azure-terraform ✓

Supported account types

Who can use this application or access this API?

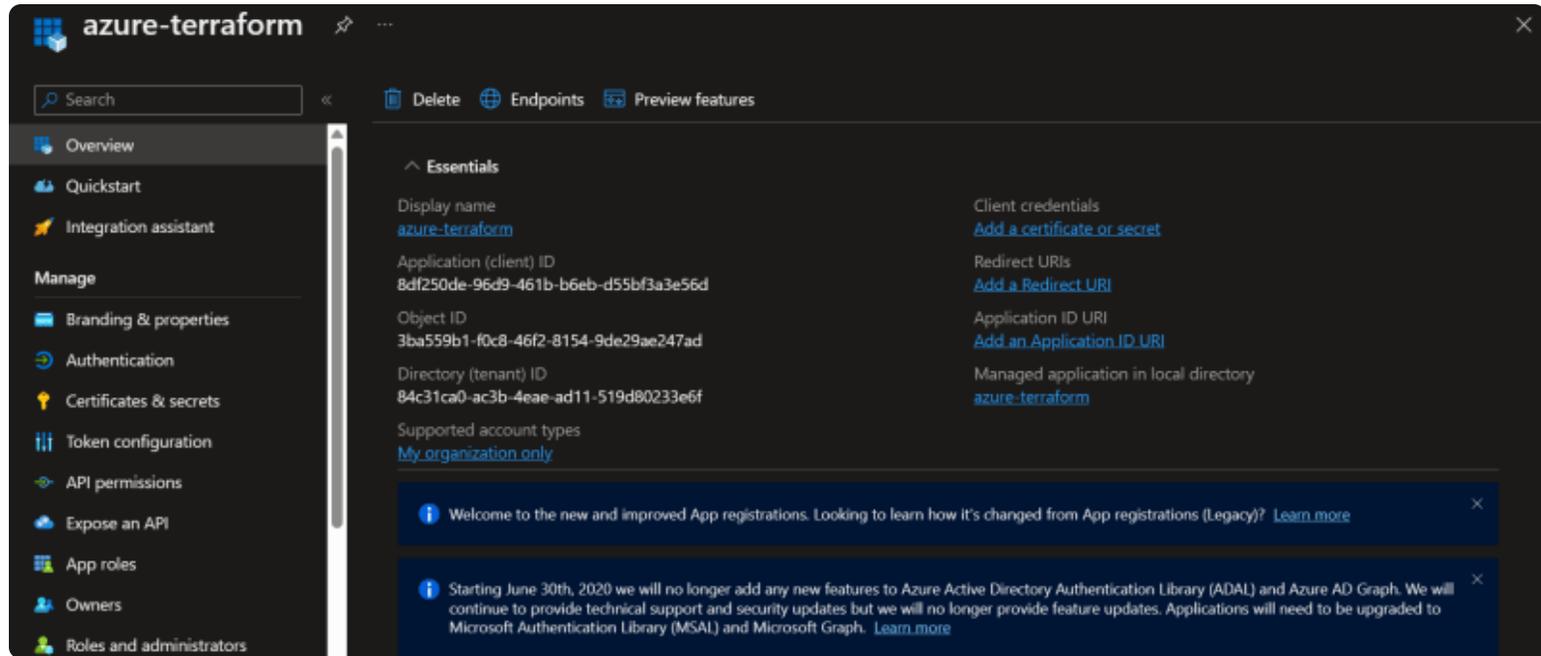
- Accounts in this organizational directory only (Student Ambassadors only - Single tenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- Personal Microsoft accounts only

[Help me choose...](#)

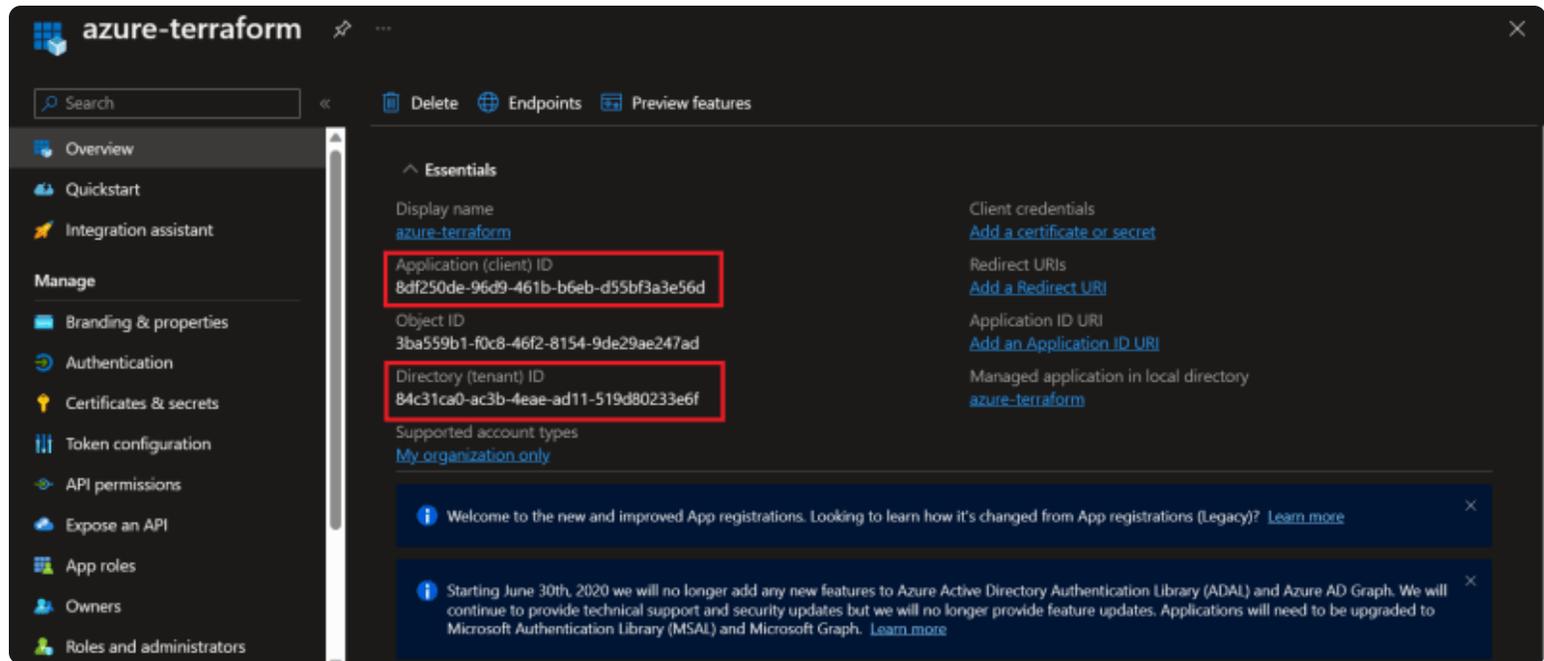
By proceeding, you agree to the [Microsoft Platform Policies](#)

Register

- So we have finally created our app.



- To connect our Infrastructure we need 4 details
 - Client Id
 - Client Secret
 - Tenant Id
 - Subscription Id We would be getting this one by one.
- Application (client) ID is Client Id. Directory (tenant) ID is Tenant Id.

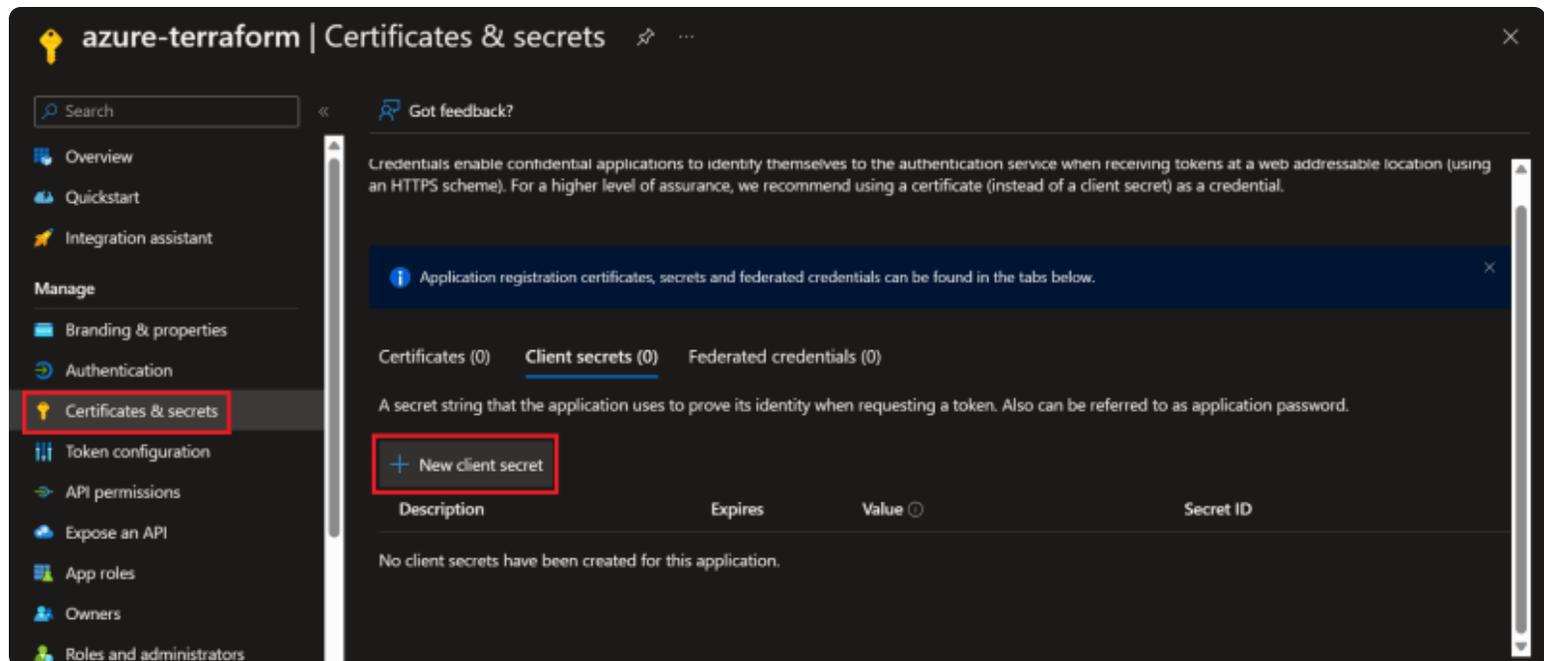


The screenshot shows the Azure portal interface for an application registration named 'azure-terraform'. The left sidebar contains navigation options like Overview, Quickstart, Integration assistant, and Manage. The main content area is titled 'Essentials' and displays key information for the application:

- Display name: [azure-terraform](#)
- Application (client) ID: **8df250de-96d9-461b-b6eb-d55bf3a3e56d** (highlighted with a red box)
- Object ID: 3ba559b1-f0c8-46f2-8154-9de29ae247ad
- Directory (tenant) ID: **84c31ca0-ac3b-4eae-ad11-519d80233e6f** (highlighted with a red box)
- Supported account types: [My organization only](#)

Client credentials are also listed with links to add certificates or secrets, redirect URIs, and application ID URIs. Two informational messages are visible at the bottom of the page.

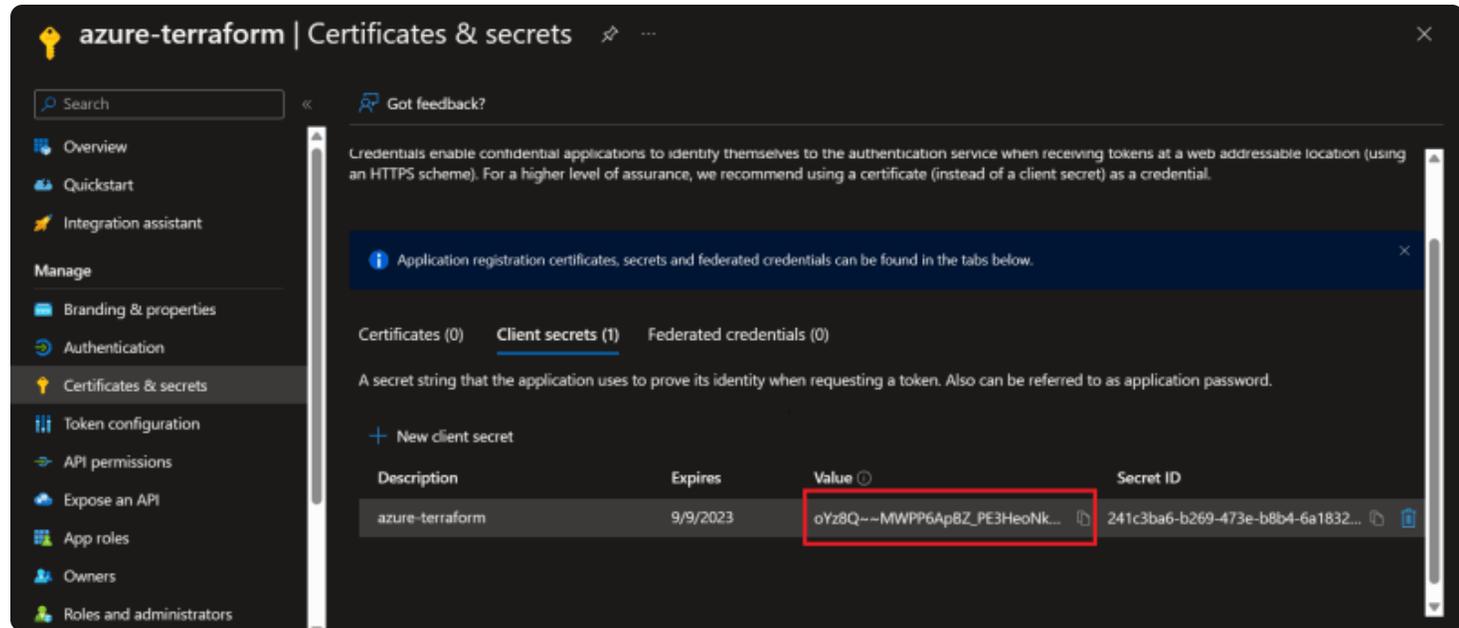
- To get the Client Secret, we first need to create the secret.



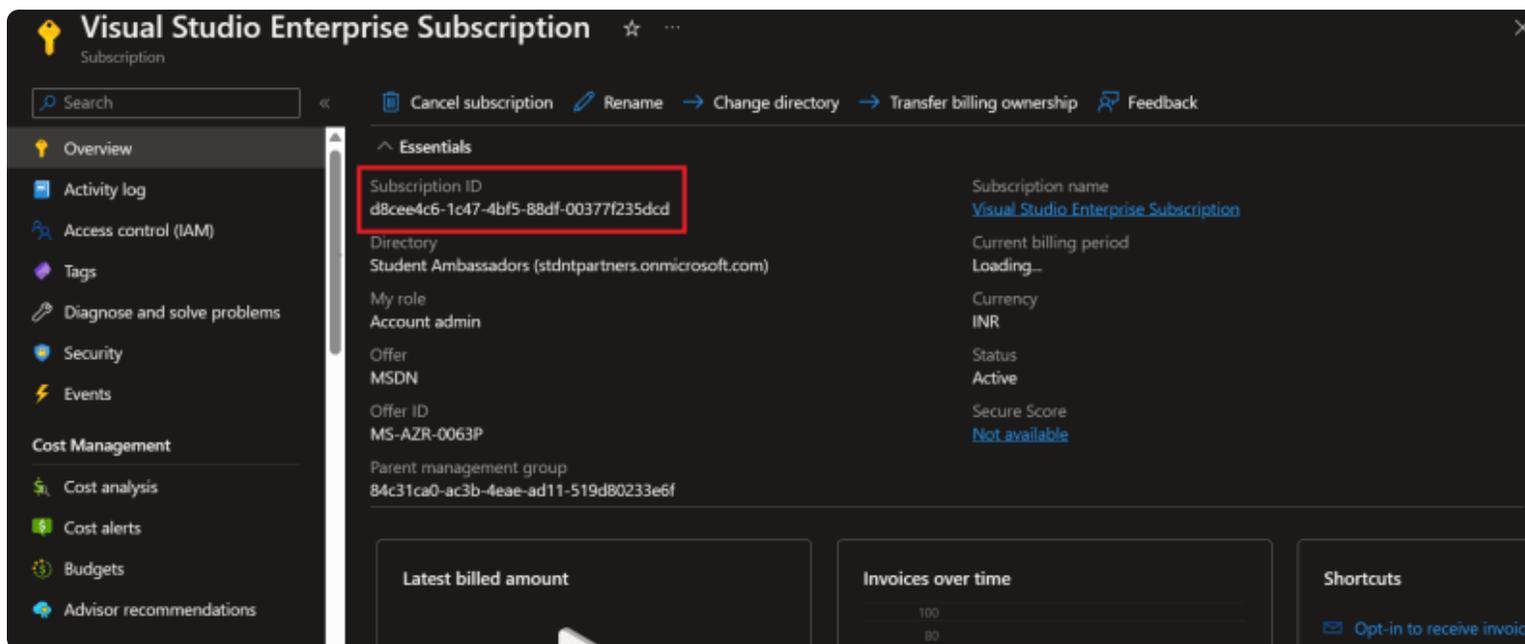
The screenshot shows the 'Certificates & secrets' page for the 'azure-terraform' application registration. The left sidebar is the same as in the previous screenshot. The main content area is titled 'Certificates & secrets' and provides information about credentials:

- Introduction: Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.
- Navigation: Application registration certificates, secrets and federated credentials can be found in the tabs below.
- Current state: Certificates (0), **Client secrets (0)**, Federated credentials (0)
- Definition: A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.
- Action: **+ New client secret** (highlighted with a red box)
- Table: A table with columns for Description, Expires, Value, and Secret ID. The message 'No client secrets have been created for this application.' is displayed below the table.

- Add description and expiry of this secret which you're creating.
- Copy this value under Value column, and save it somewhere as we wouldn't be able to access this later. This is our Client secret.



- Search Subscription from search box and open your subscription. Copy your subscription id.



So now we have all the required values.

- Let's add these values in our Infrastructure.
 - In the provider block add all four details. We would be saving the values in Terraform Cloud variables for security purpose.

```
provider "azurerm" {  
  features {}  
  
  skip_provider_registration = "true"  
  
  # Connection to Azure  
  subscription_id = var.subscription_id  
  client_id       = var.client_id  
  client_secret   = var.client_secret
```

```
tenant_id = var.tenant_id  
}
```

- Add variables in Variables page.

The screenshot shows the Terraform Cloud interface. On the left is a dark sidebar with a navigation menu: Workspaces, azure-terraform-infrastructure, Overview, Runs, States, Variables (highlighted with a red box), and Settings. The main content area shows the workspace details for 'azure-terraform-infrastructure' (ID: ws-TqDbfgTrAQL7sYBL). It indicates 0 resources, Terraform version 1.4.6, and was updated a few seconds ago. Below this, the 'Variables' section is active, showing an 'Unlocked' status and an 'Actions' dropdown. The 'Variables' section explains that Terraform uses Terraform and Environment variables. It lists 'Sensitive variables' and 'Workspace variables (0)'. A table with columns 'Key', 'Value', and 'Category' is shown, with the message 'There are no variables added.' Below the table is a '+ Add variable' button (highlighted with a red box). The 'Variable sets (0)' section explains that variable sets allow reusing variables across workspaces. At the bottom, a message states 'No variable sets have been applied to this workspace.' with an 'Apply variable set' button and a link to 'Learn about variable sets'.

- In key add the words with var, i.e., var. key , and in value add required Ids' and Secret.

Key	Value	Category	
client_id	8df250de-96d9-461b-b6eb-d55bf3a3e56d	terraform	...
client_secret	<i>Sensitive - write only</i>	terraform	...
	SENSITIVE		
subscription_id	d8cee4c6-1c47-4bf5-88df-00377f235dcd	terraform	...
tenant_id	84c31ca0-ac3b-4eae-ad11-519d80233e6f	terraform	...

Let's rerun the workflow!

Ohhhhh!!!! ERROR AGAIN!!!!

The screenshot shows the Terraform Cloud interface for a workspace named 'azure-terraform-infrastructure'. The run status is 'Failed'. The error message is 'Plan errored'. The diagnostics section lists several errors and warnings:

- Warning: Value for undeclared variable**: The root module does not declare a variable named "tenant_id" but a value was found in file "home\%agent%\%agent%\component\terraform\not-run-396yvtLk15AMMZ\terraform.tfvars". If you meant to use this value, add a "variable" block to the configuration. To silence these warnings, use TF_VAR_ environment variables to provide certain "global" settings to all configurations in your organization. To reduce the verbosity of these warnings, use the -compact-warnings option.
- Warning: Value for undeclared variable**: The root module does not declare a variable named "client_secret" but a value was found in file "home\%agent%\%agent%\component\terraform\not-run-396yvtLk15AMMZ\terraform.tfvars". If you meant to use this value, add a "variable" block to the configuration. To silence these warnings, use TF_VAR_ environment variables to provide certain "global" settings to all configurations in your organization. To reduce the verbosity of these warnings, use the -compact-warnings option.
- Warning: Values for undeclared variables**: In addition to the other similar warnings shown, 2 other variable(s) defined without being declared.
- Error: Reference to undeclared input variable**: on main.tf line 20, in provider "azurerm":

```
client_secret = var.client_secret
```

An input variable with the name "client_secret" has not been declared. This variable can be declared with a variable "client_secret" block.
- Error: Reference to undeclared input variable**: on main.tf line 21, in provider "azurerm":

```
tenant_id = var.tenant_id
```

An input variable with the name "tenant_id" has not been declared. This variable can be declared with a variable "tenant_id" block.
- Error: Reference to undeclared input variable**: on main.tf line 18, in provider "azurerm":

```
subscription_id = var.subscription_id
```

An input variable with the name "subscription_id" has not been declared. This variable can be declared with a variable "subscription_id" block.
- Error: Reference to undeclared input variable**: on main.tf line 19, in provider "azurerm":

```
client_id = var.client_id
```

An input variable with the name "client_id" has not been declared. This variable can be declared with a variable "client_id" block.

Additional messages at the bottom of the run details include 'Cost estimation will not run' and 'Apply will not run'.

But we are still left with some more code :P

We need to add variables.tf file mentioning about these variables.

```

1  variable "client_id" {
2    type = string
3  }
4

```

```
5  variable "client_secret" {
6    type = string
7  }
8
9  variable "subscription_id" {
10   type = string
11 }
12
13 variable "tenant_id" {
14   type = string
15 }
```

variables.tf hosted with ❤️ by GitHub

[view raw](#)

And now finally we can get the results from Plan and Apply phase. Let's rerun the pipeline.

- Wohooo!! Our Plan phase ran successfully:

aniketkumarsinha / Projects & workspaces / azure-terraform-infrastructure / Runs / run-sBvaToK54agPdBBQq

azure-terraform-infrastructure

ID: ws-TqDbfgTrAQL7sYBL

Resources: 0 | Terraform version: 1.4.6 | Updated: 3 minutes ago

No workspace description available. [Add workspace description.](#)

Running Actions

Run Workflow

CURRENT Cost estimated

Estimated cost increase: ↑ \$60.48 | Plan duration: Less than a minute

Resources to be changed: +5 -0 -0

- aniketkumarsinha triggered a run from UI 3 minutes ago Run Details
- ✓ **Plan finished** 3 minutes ago Resources: 5 to add, 0 to change, 0 to destroy
- ✓ **Cost estimation finished** 3 minutes ago Resources: 1 of 2 estimated - \$60.48/mo - +\$60.48
- 🕒 **Apply pending**

⚠️ Please review the following changes before continuing:

To create: + 5

Choosing "Confirm & Apply" below will execute the above changes. Please [review the plan output](#) before proceeding.

Confirm & Apply Discard Run Add Comment

© 2023 HashiCorp, Inc. [Support](#) [Terms](#) [Privacy](#) [Security](#)

- Expand Plan phase to check what resources are getting created. And if every configuration is fine then tap on `Confirm & Apply` button at the end of phase. So by default the Apply phase does not run automatically, we need manual approval, this is to make sure that someone checks the Plan output and verifies everything and then accordingly approve for Apply or reject.

✓ **Plan finished** 5 minutes ago Resources: 5 to add, 0 to change, 0 to destroy ^

Started 5 minutes ago > Finished 5 minutes ago

+ 5 to create

Filter resources by address... Filter by action ▾ Terraform 1.4.6 Download raw log

- > +  azurem_network_interface.nic
- > +  azurem_resource_group.rg
- > +  azurem_subnet.internal
- > +  azurem_virtual_machine.vm
- > +  azurem_virtual_network.vnet

Download Sentinel mocks Sentinel mocks can be used for [testing your Sentinel policies](#)

- Error again ☐

The screenshot displays the Terraform Cloud interface for a workspace named 'azure-terraform-infrastructure'. The 'Run Workflow' section shows a sequence of steps: 'Plan finished', 'Cost estimation finished', and 'Apply errored'. The 'Apply errored' step is expanded, showing the following error message:

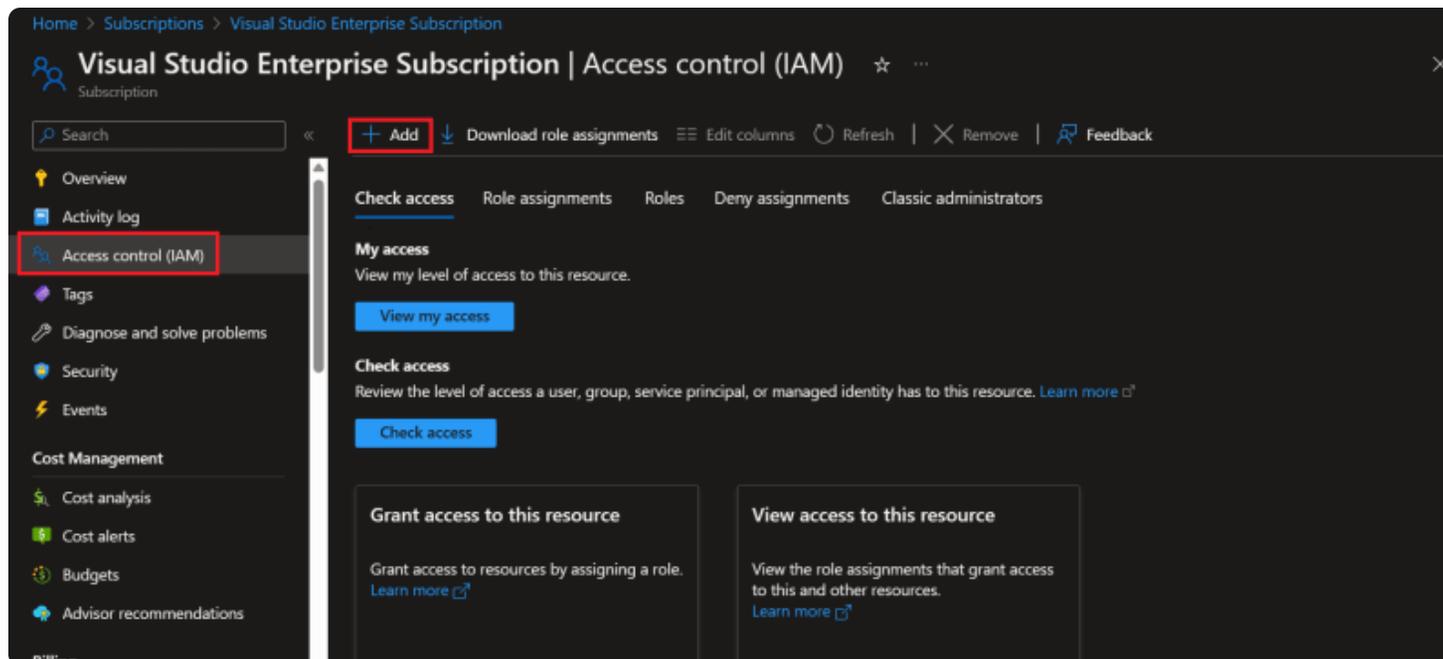
```

Error: checking for presence of existing resource group: resources.GroupsClient#Get: Failure responding to request:
StatusCode=403 -- Original Error: azure/azure: Service returned an error. Status=403 Code="AuthorizationFailed"
Message="The client 'acfedab6-c231-4d24-bede-49b810a97126' with object id 'acfedab6-c231-4d24-bede-49b810a97126' does not have authorization to perform action 'Microsoft.Resources/subscriptions/resourcegroups/read' over scope '/subscriptions/d8ee046-1c47-4bf5-88df-00377235dcd/resourcegroups/terraform-ResourceGroup' or the scope is invalid. If access was recently granted, please refresh your credentials."
on main.tf line 28, in resource "azure_resource_group": "rg":
  resource "azure_resource_group" "rg" {

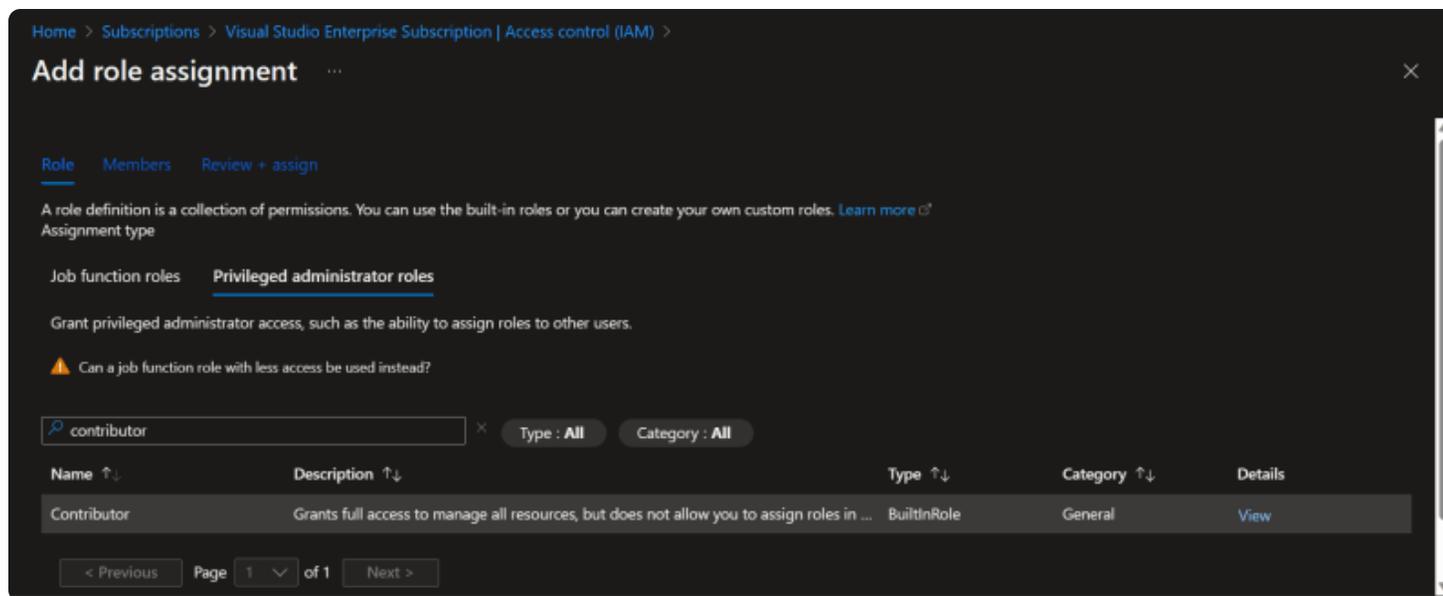
```

Below the error message, a list of resources is shown, with 'azure_resource_group' marked as 'Create failed'. The 'State versions created' section shows a single version for the workspace, created on Jun 11, 2023 at 00:57:11 am.

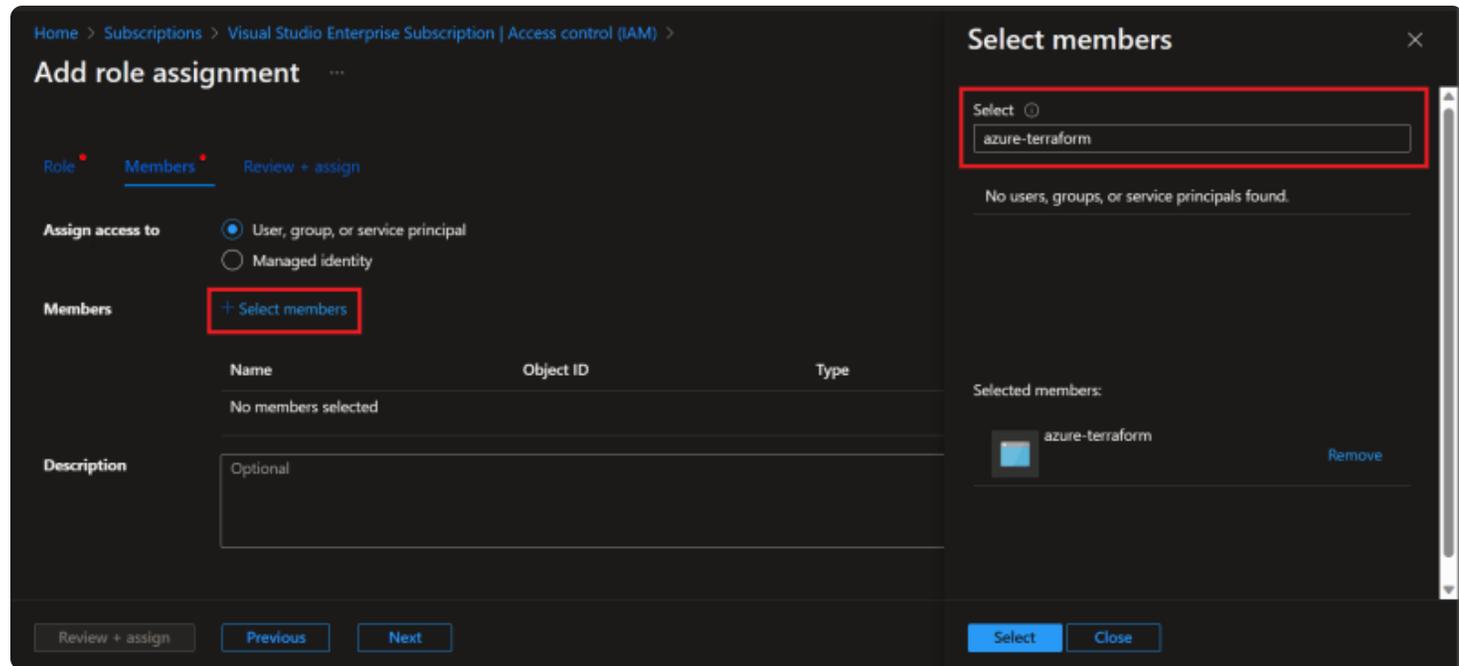
- It seems our Azure app don't have permission to add anything. Let's provide the contributor role to our app.
 - Under Subscription, got to Access Control (IAM) and Add a role there.



- Add a contributor role under Privileged administrator roles.



- Under Members tab, select our app as member.



Home > Subscriptions > Visual Studio Enterprise Subscription | Access control (IAM) >

Add role assignment

Role **Members** Review + assign

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

Select

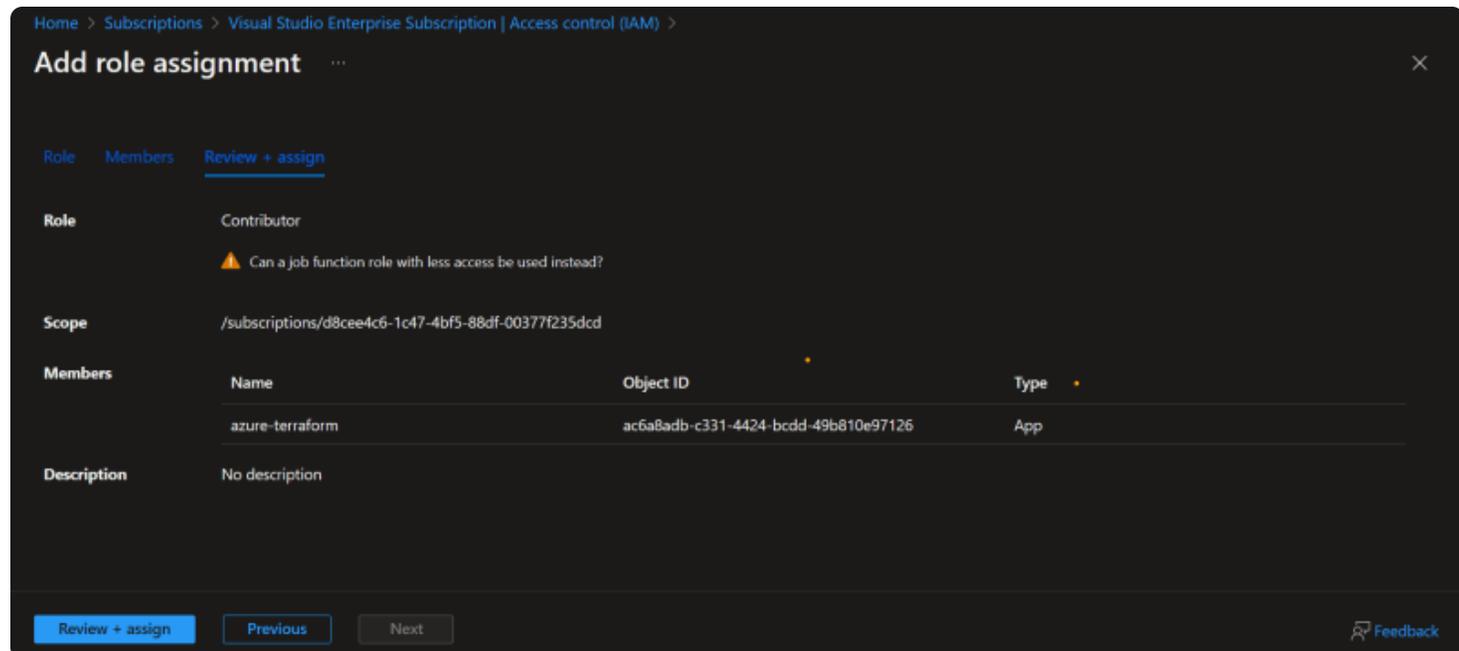
No users, groups, or service principals found.

Selected members:

- azure-terraform [Remove](#)

Select Close

- And then tap on Review+Assign. So now our App has contributor role and can make changes over Azure subscription.



Home > Subscriptions > Visual Studio Enterprise Subscription | Access control (IAM) >

Add role assignment

Role **Members** Review + assign

Role Contributor
⚠ Can a job function role with less access be used instead?

Scope /subscriptions/d8cee4c6-1c47-4bf5-88df-00377f235dcd

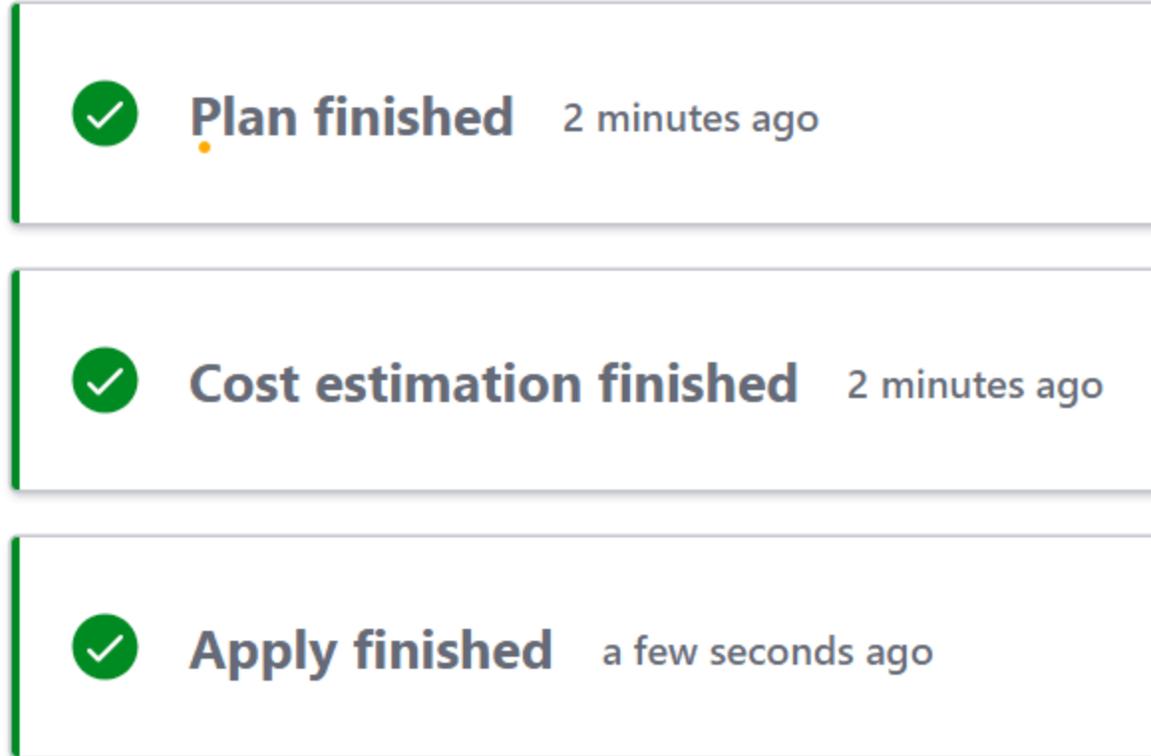
Name	Object ID	Type
azure-terraform	ac6a8adb-c331-4424-bcdd-49b810e97126	App

Description No description

Review + assign Previous Next

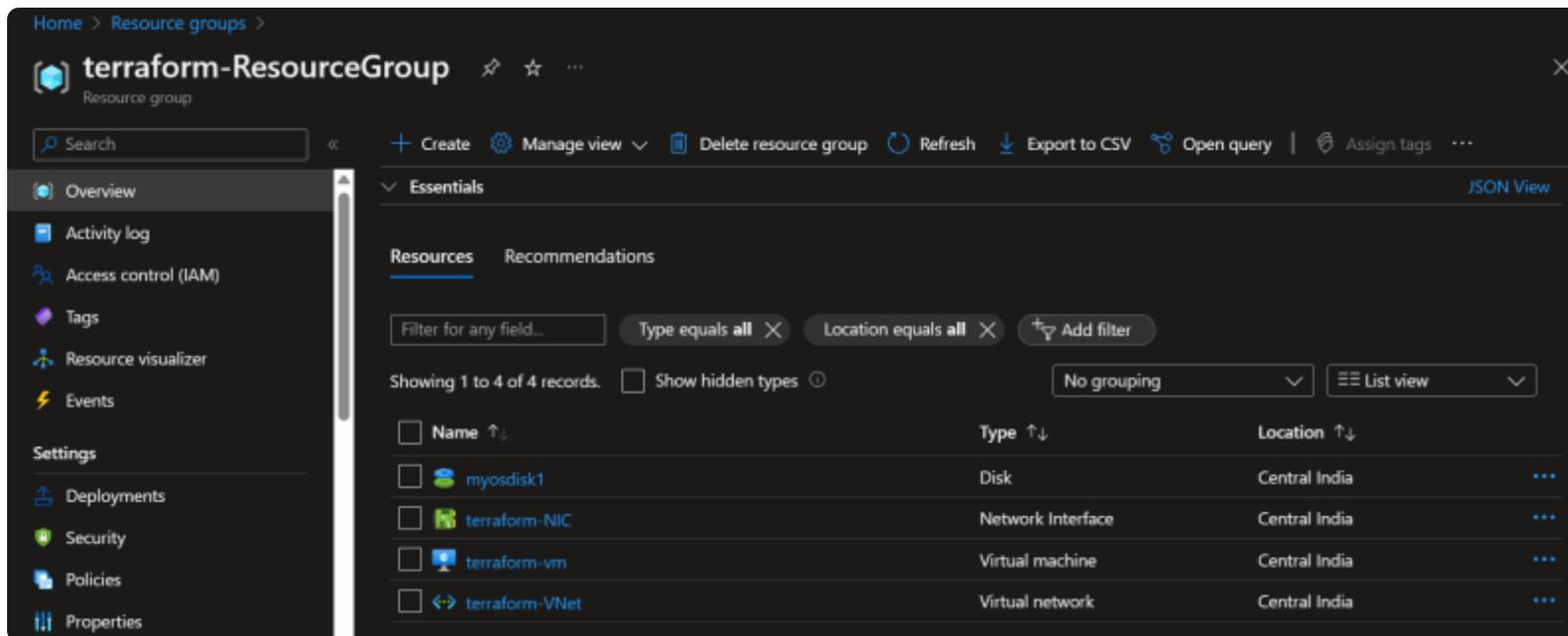
Feedback

- Rerun the pipeline. And wollaahh!! Everything ran successfully!



Let's confirm over our Azure Portal too.

Yes we can see all our resources present under our subscription.



The screenshot shows the Azure portal interface for a resource group named 'terraform-ResourceGroup'. The left sidebar contains navigation options like Overview, Activity log, Access control (IAM), Tags, Resource visualizer, Events, Settings, Deployments, Security, Policies, and Properties. The main content area shows the 'Essentials' section with a 'Resources' tab selected. A table lists the resources:

Name	Type	Location
myosdisk1	Disk	Central India
terraform-NIC	Network Interface	Central India
terraform-vm	Virtual machine	Central India
terraform-VNet	Virtual network	Central India



Damn!!!! We learnt so many things today. We first created the Terraform Infrastructure, setup our Terraform Cloud profile and then used it to deploy resources over Azure too. Now you can say you're a pro in Terraform and Terraform Cloud.

Let me provide you a bonus happiness. So the project which we configured over Terraform Cloud, automatically runs the workflow any branch is merged to the main branch or any new pushes are done to main branch.

You can get the whole Infrastructure as Code from this repository:

 [aniketkumarsinha / azure-terraform-infrastructure](https://github.com/aniketkumarsinha/azure-terraform-infrastructure)

Deploy Azure Infrastructure using Terraform Cloud

Deploy Azure Infrastructure using Terraform Cloud



[View on GitHub](#)

Share if your like the blog and follow for more!!



Aniket Kumar Sinha

[Follow](#)

Cloud Consultant at Rapid Circle | Gold Microsoft Learn Student Ambassador | GDSC Lead'21

[Terraform Cloud \(2 Part Series\)](#)

- 1 **Deploy Azure Infrastructure using Terraform Cloud**
- 2 Implementing Policy-as-Code to Terraform workflow usin...

Top comments (0)

[Code of Conduct](#) • [Report abuse](#)

DEV Community



Trending in DevOps

1. The DevOps community is exploring **Docker's new** `docker-compose watch` for a smoother dev experience and discussing the benefits of **SSH** for secure operations.
2. Recent talks in DevOps center on **canary deployments** and **feature flags** for safer production testing, alongside the utility of **Kubernetes tools**.
3. DevOps pros are interested in **Docker Extensions** to enhance development workflows and the simplification of **Kubernetes setups** with essential tools.
4. There's a focus on **Docker's watch feature** for hot-reloading and the integration of **SSH** in everyday development practices among DevOps enthusiasts.

5. Conversations in DevOps are delving into the ease of **testing in production** with canary deployments and the efficiency gains from new **Docker Extensions**.



Say Goodbye to Docker Volumes 🙋

Jonas Scholz · Nov 27

#docker #webdev #tutorial #devops



Explaining SSH to my Uber Driver

Jessica Wang · Nov 27

#ssh #softwareengineering #webdev #devops



🔥 5 Must have tools to install on your Kubernetes cluster 🚀

Eden Federman for Odigos · Nov 27

#webdev #programming #devops #opensource



Testing in Production with Canary Deployments: A How-To Guide

Pradumna Saraf · Nov 29

#opensource #devops #development #programming



5 Docker Extensions to make your development life easier

Pradumna Saraf · Nov 27

#docker #development #programming #devops



This is Learning

Free, open and honest software education.

Read our welcome letter which is an open invitation for you to join.

[Join us as a writer](#)

More from [This is Learning](#)

The Most Upvoted Visual Studio Code Feature

#vscode #microsoft #github #productivity

Set up GitHub Codespaces for a .NET 8 application

#dotnet #github #productivity #docker

GitHub Codespaces: A Faster Way to Develop in the Cloud

#github #docker #productivity #webdev

DEV Community



 **Life is too short to browse without [dark mode](#)**

You can customize your theme, font, and more [when you are signed in](#).