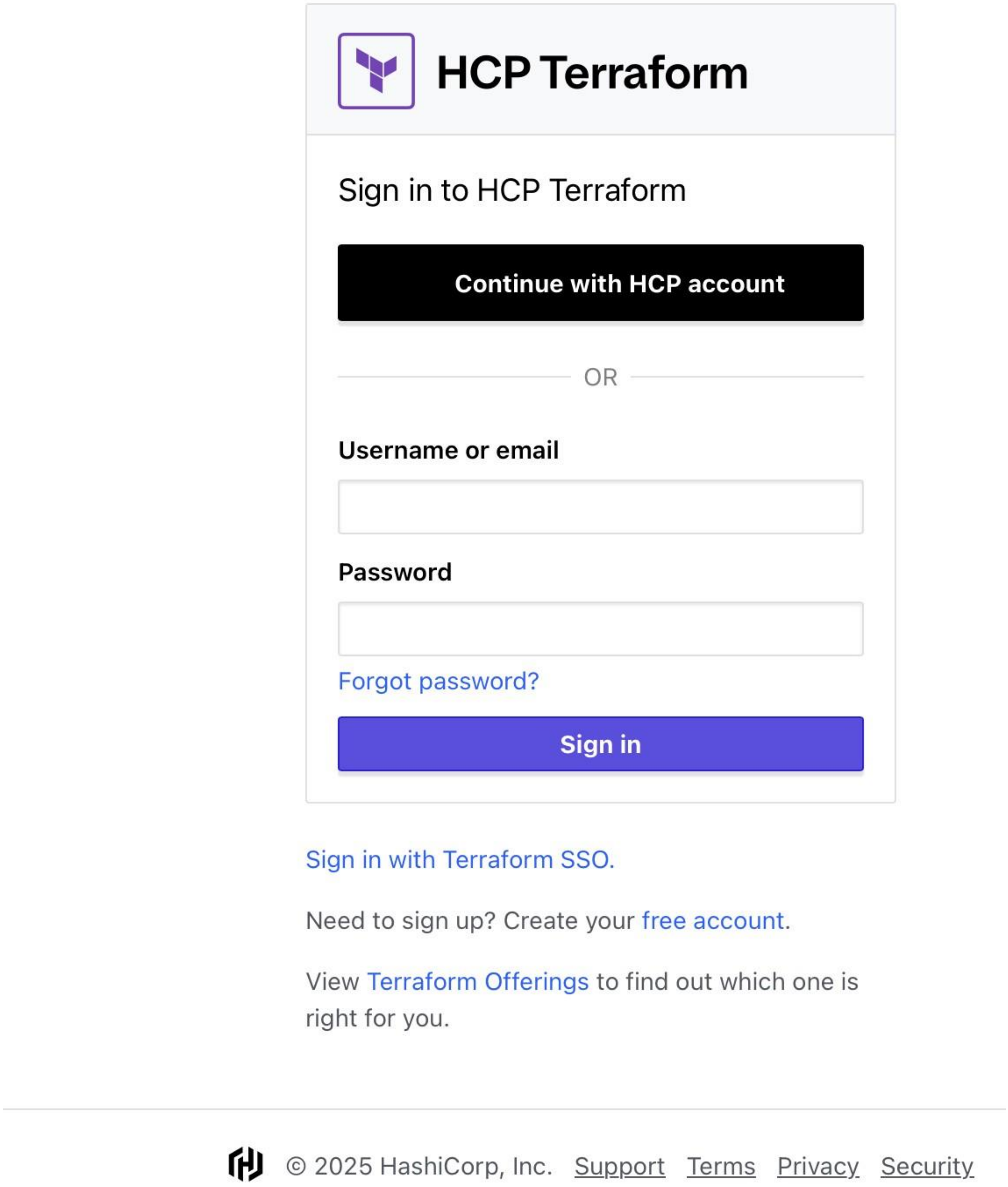
Create an account in TFE CLOUD

https://app.terraform.io/public/signup/account



Sign in to HCP Terraform

Username or email

Password

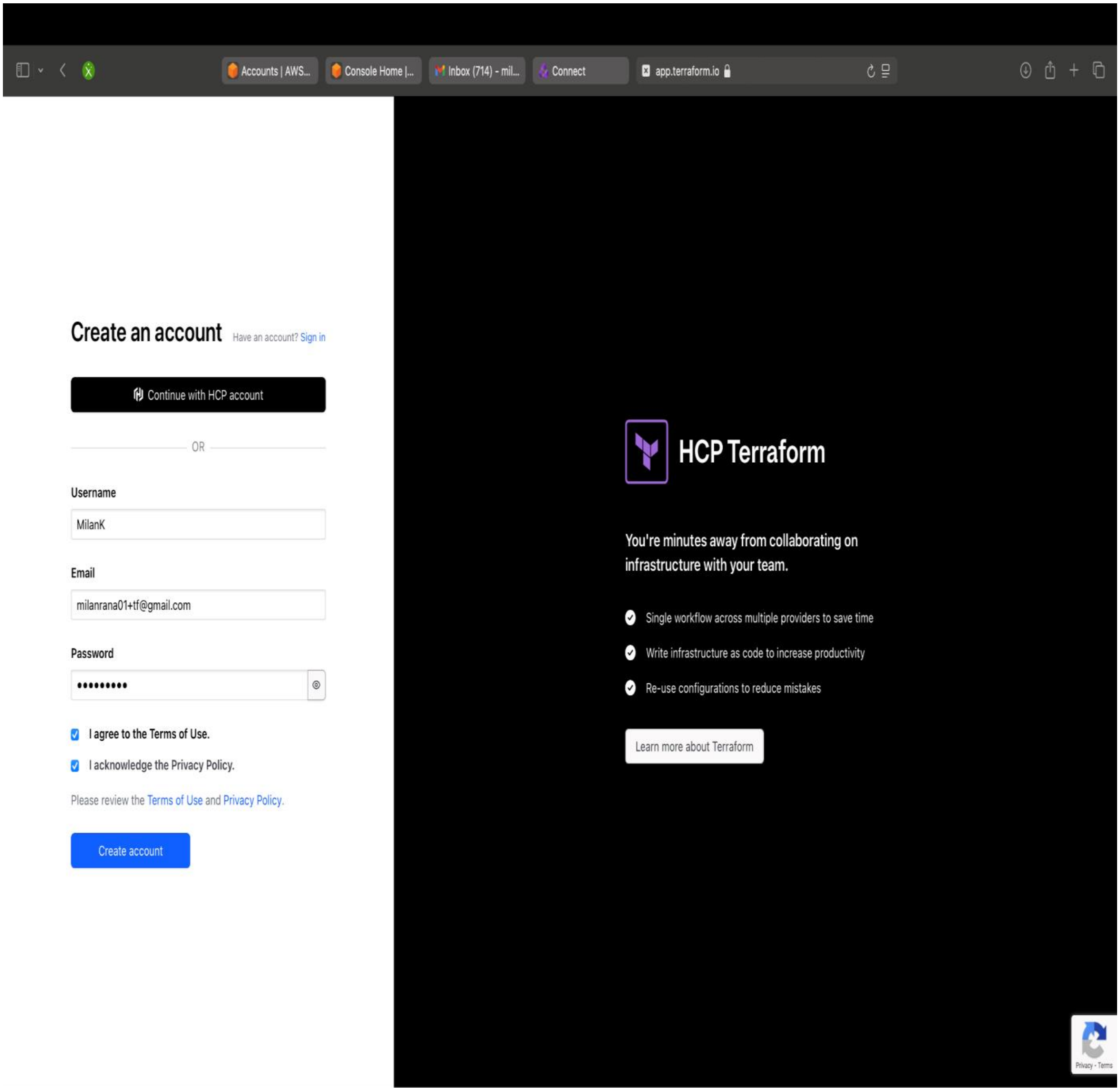
Sign in with Terraform SSO.

Need to sign up? Create your free account.

View Terraform Offerings to find out which one is right for you.

© 2025 HashiCorp, Inc. Support Terms Privacy Security

Click on free account. Enter your details



Create an account Have an account? Sign in

Username

MilanK

Email

milanrana01+tf@gmail.com

Password

•••••••••

☑ I agree to the Terms of Use.

☑ I acknowledge the Privacy Policy.

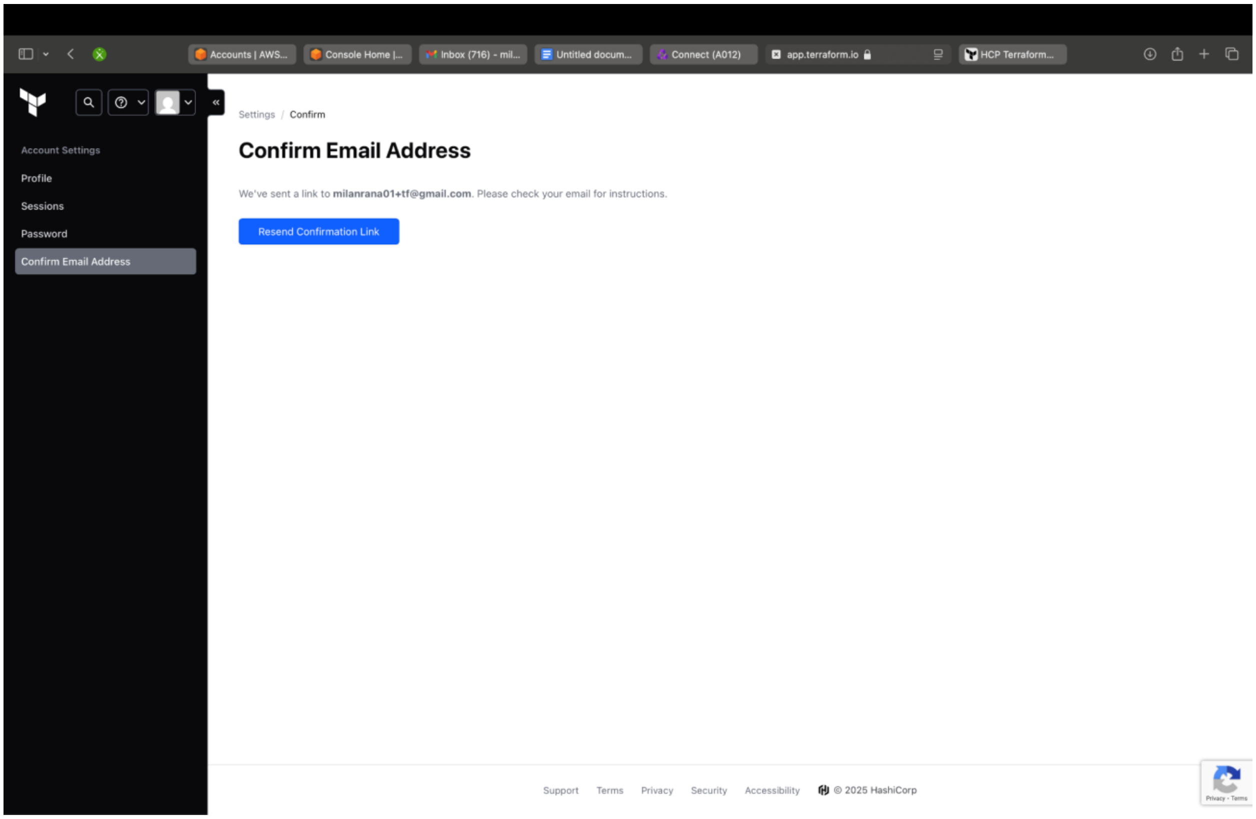
Please review the Terms of Use and Privacy Policy.

HCP Terraform

You're minutes away from collaborating on infrastructure with your team.

- Single workflow across multiple providers to save time
- Write infrastructure as code to increase productivity
- Re-use configurations to reduce mistakes

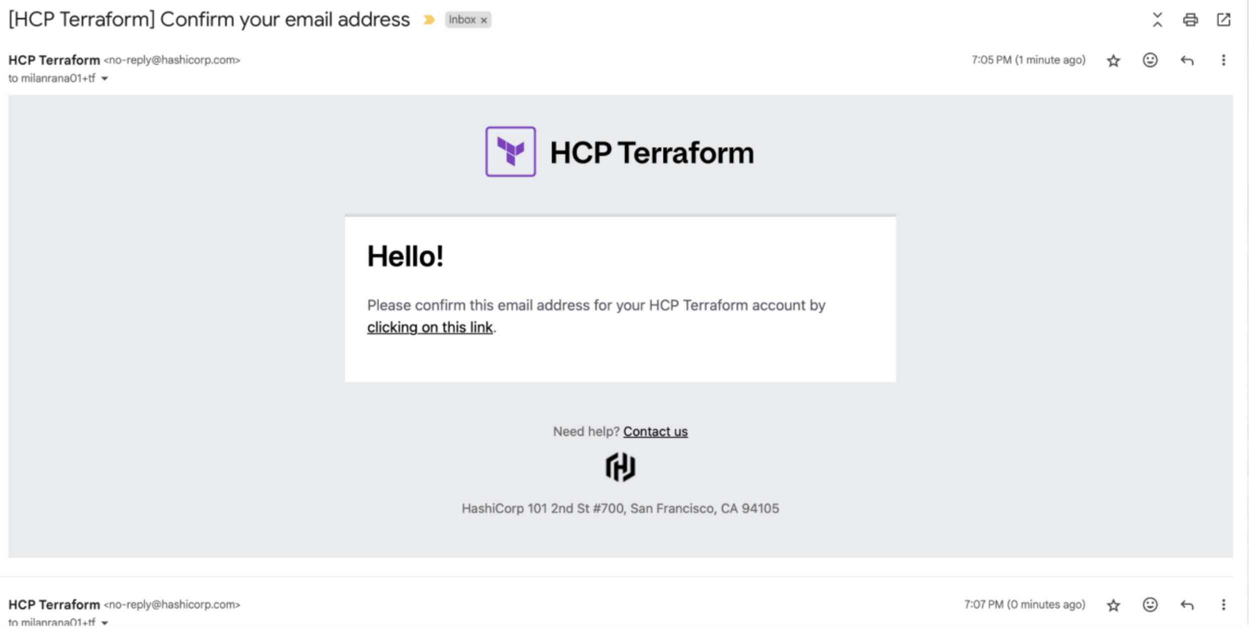
Once entered, you will be in the TFE Cloud, click on confirmation link that should send an email to you to verify



Confirm Email Address

We've sent a link to **milanrana01+tf@gmail.com**. Please check your email for instructions.

Resend Confirmation Link



[HCP Terraform] Confirm your email address

HCP Terraform <no-reply@hashicorp.com>
to milanrana01+tf

HCP Terraform

Hello!

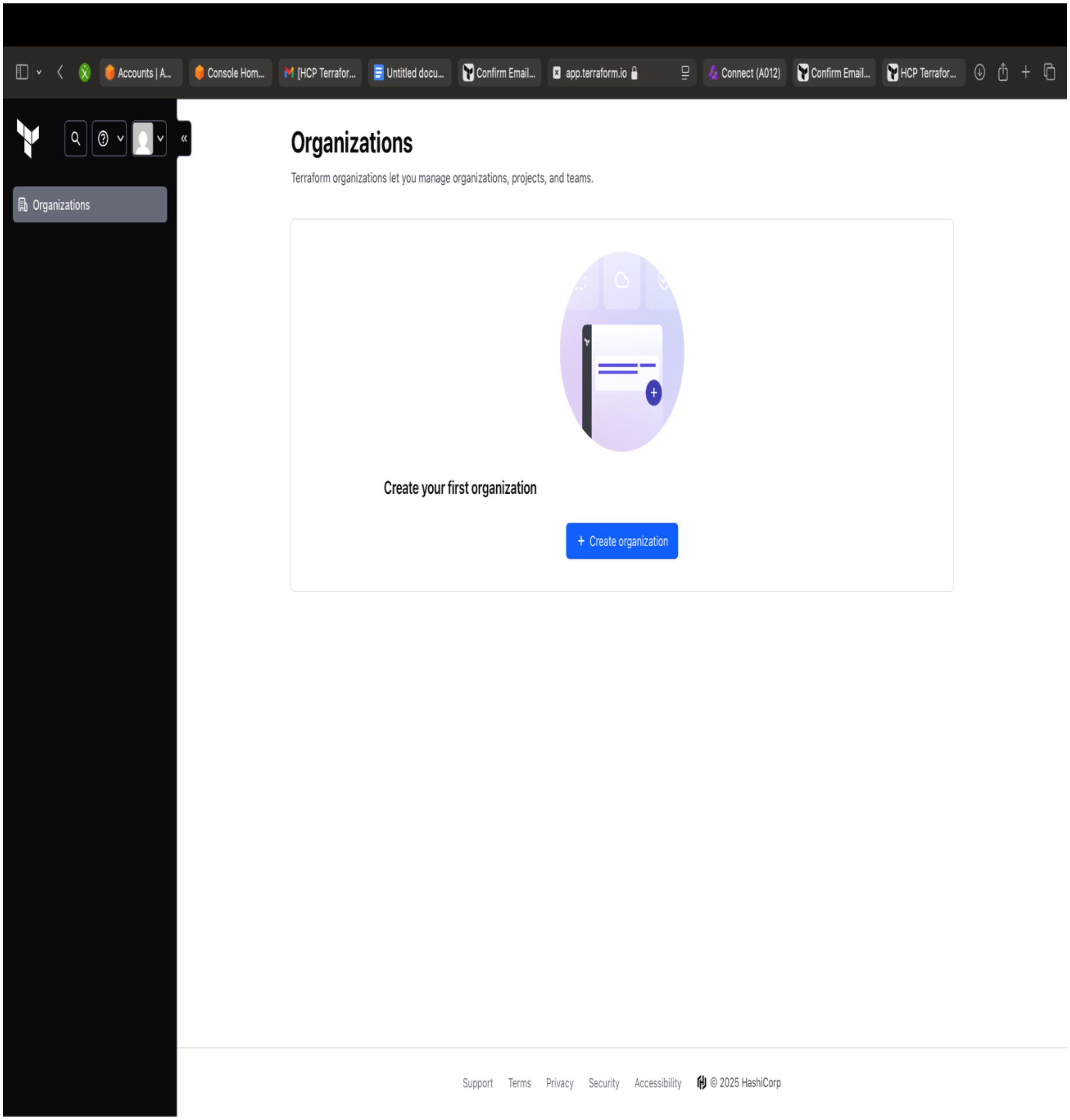
Please confirm this email address for your HCP Terraform account by clicking on this link.

Need help? Contact us

HashiCorp 101 2nd St #700, San Francisco, CA 94105

HCP Terraform <no-reply@hashicorp.com>
to milanrana01+tf

It takes you to the organization page that is basically a way Terraform let you manage organizations, projects, and teams.



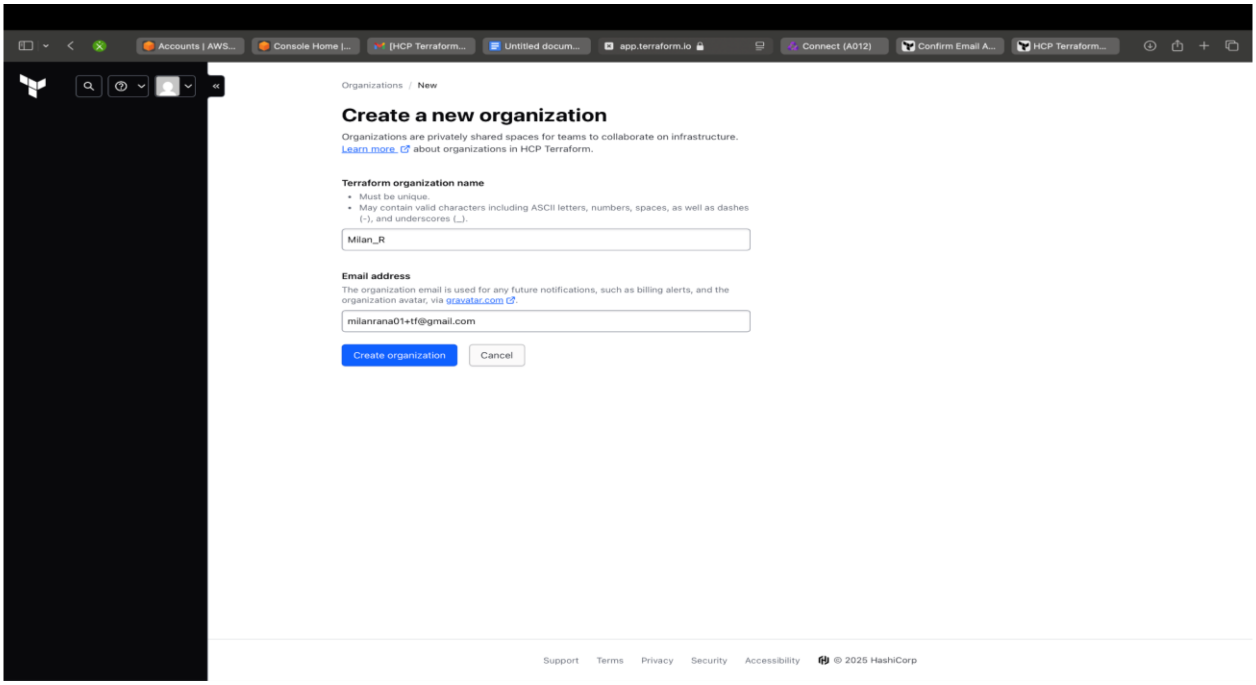
Organizations

Terraform organizations let you manage organizations, projects, and teams.

Create your first organization

+ Create organization

Enter Details to create the org.



Organizations / New

Create a new organization

Organizations are privately shared spaces for teams to collaborate on infrastructure. Learn more about organizations in HCP Terraform.

Terraform organization name

- Must be unique.
- May contain valid characters including ASCII letters, numbers, spaces, as well as dashes (-), and underscores (_).

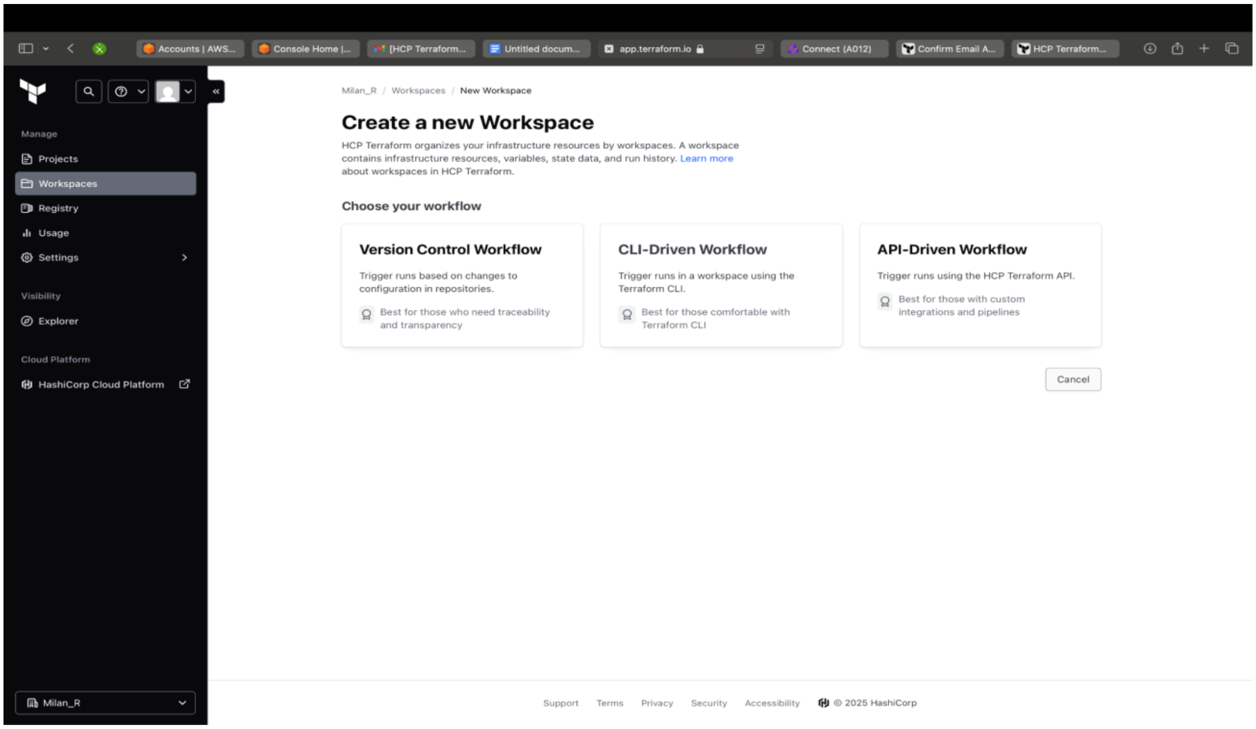
Milan_R

Email address

The organization email is used for any future notifications, such as billing alerts, and the organization avatar, via gravatar.com.

milanrana01+tf@gmail.com

It should land you the below page, where we will be using the CLI driven workflow for the sake of this workshop



Milan_R / Workspaces / New Workspace

Create a new Workspace

HCP Terraform organizes your infrastructure resources by workspaces. A workspace contains infrastructure resources, variables, state data, and run history. Learn more about workspaces in HCP Terraform.

Choose your workflow

Version Control Workflow

Trigger runs based on changes to configuration in repositories.

Best for those who need traceability and transparency

CLI-Driven Workflow

Trigger runs in a workspace using the Terraform CLI.

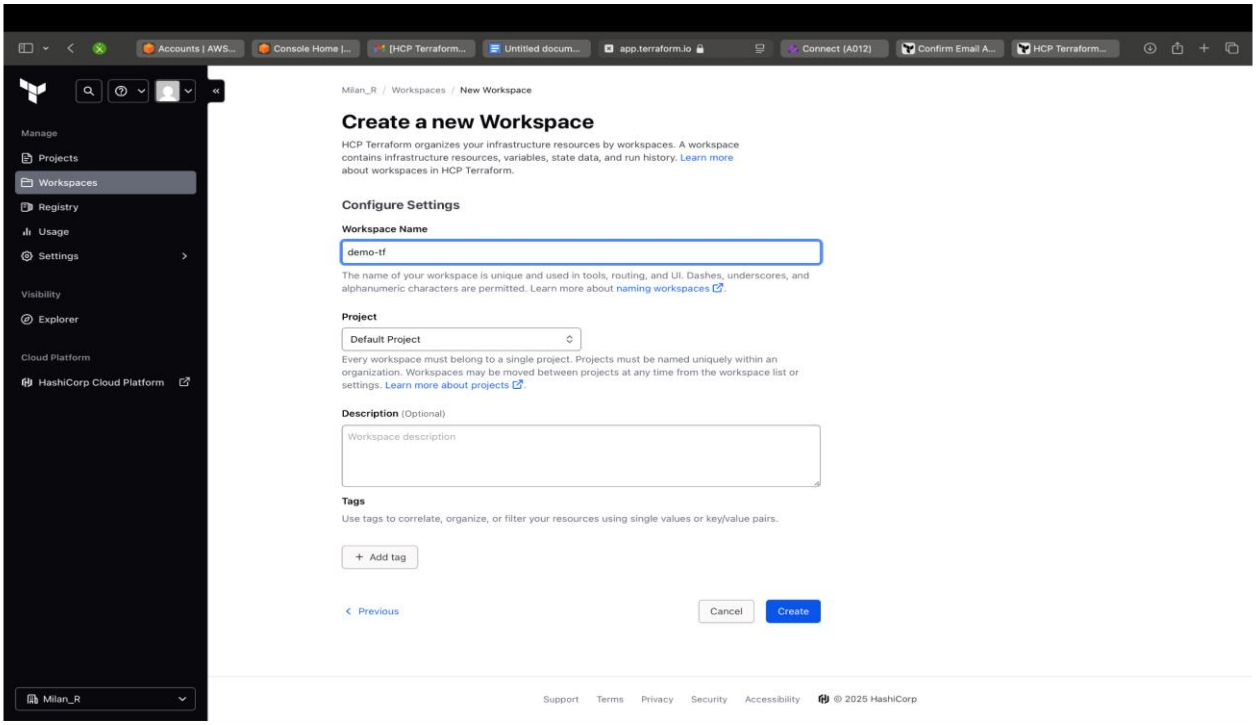
Best for those comfortable with Terraform CLI

API-Driven Workflow

Trigger runs using the HCP Terraform API.

Best for those with custom integrations and pipelines

Selecting the cli driven option should take you to the page to create a workspace, enter details and click enter



Milan_R / Workspaces / New Workspace

Create a new Workspace

HCP Terraform organizes your infrastructure resources by workspaces. A workspace contains infrastructure resources, variables, state data, and run history. Learn more about workspaces in HCP Terraform.

Configure Settings

Workspace Name: demo-tf

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: Default 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): Workspace description

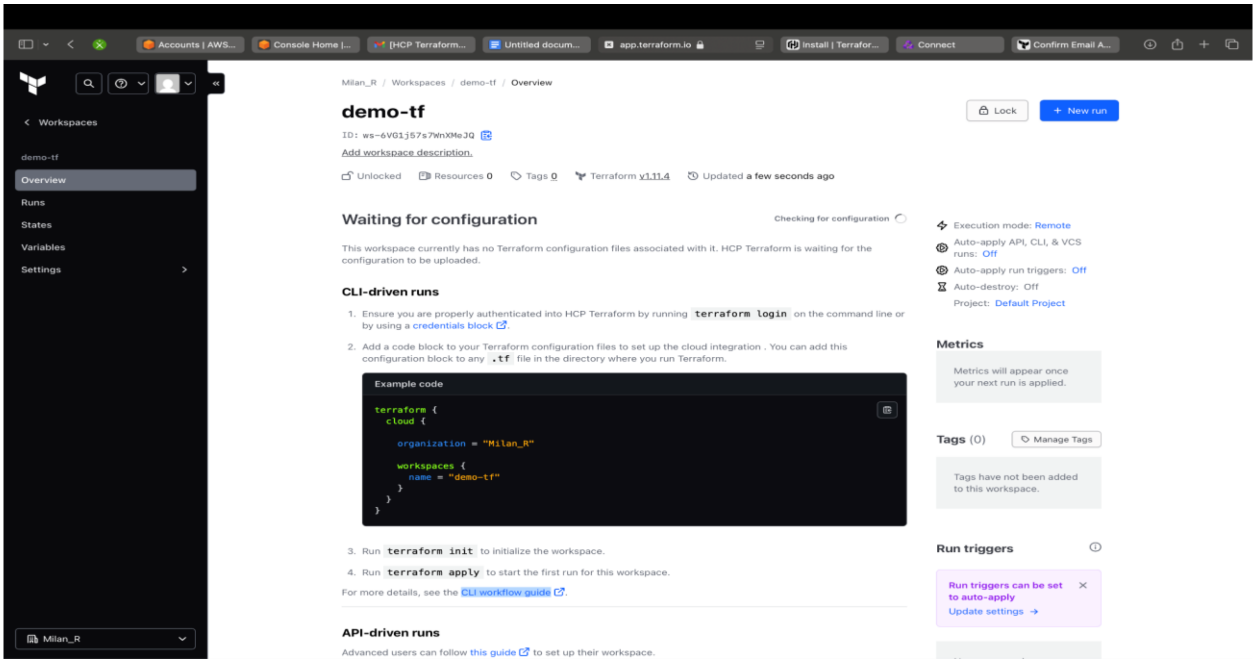
Tags

Use tags to correlate, organize, or filter your resources using single values or key/value pairs.

+ Add tag

< Previous   Cancel   Create

Once the Workspace gets created, you can follow further instructions in the below link to set up tf in your local and then we can connect our aws with tfe cloud.
Link-https://developer.hashicorp.com/terraform/install#darwin



Milan_R / Workspaces / demo-tf / Overview

demo-tf

ID: ws-6VG1j57s7WnXMeJQ

Add workspace description.

Unlocked   Resources 0   Tags 0   Terraform v1.11.4   Updated a few seconds ago

Waiting for configuration   Checking for configuration

This workspace currently has no Terraform configuration files associated with it. HCP Terraform is waiting for the configuration to be uploaded.

CLI-driven runs

1. Ensure you are properly authenticated into HCP Terraform by running `terraform login` on the command line or by using a credentials block.
2. Add a code block to your Terraform configuration files to set up the cloud integration . You can add this configuration block to any `.tf` file in the directory where you run Terraform.

Example code

```
terraform {
  cloud {

    organization = "Milan_R"

    workspaces {
      name = "demo-tf"
    }
  }
}
```

3. Run `terraform init` to initialize the workspace.
4. Run `terraform apply` to start the first run for this workspace.

For more details, see the CLI workflow guide.

API-driven runs

Advanced users can follow this guide to set up their workspace.

Execution mode: Remote
Auto-apply API, CLI, & VCS runs: Off
Auto-apply run triggers: Off
Auto-destroy: Off
Project: Default Project

Metrics

Metrics will appear once your next run is applied.

Tags (0)   Manage Tags

Tags have not been added to this workspace.

Run triggers

Run triggers can be set to auto-apply
Update settings →

Manual Terraform Installation

1. Download Terraform –

Visit the official download page: **https://developer.hashicorp.com/terraform/downloads**

Choose the Correct Version

- **Apple Silicon (M1/M2/M3)**: Select arm64 architecture
- **Intel Macs**: Select amd64 architecture

2. Make Terraform System-Wide Accessible

- Move the downloaded binary to your system PATH: -sudo mv terraform /usr/local/bin/

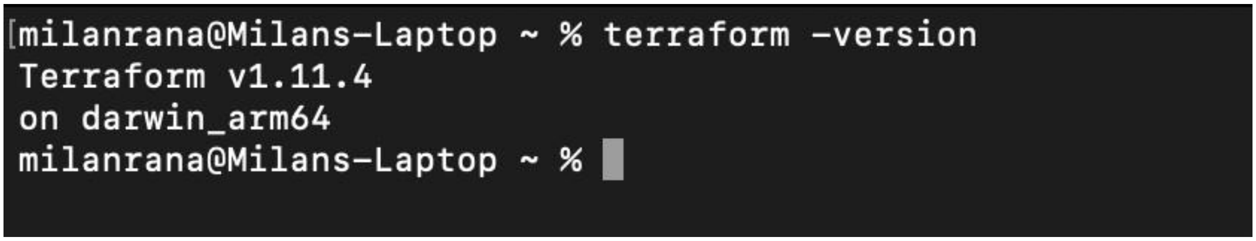
3. Set Execute Permissions

- Ensure the binary has execute permissions: - chmod +x /usr/local/bin/terraform

Note: Usually already has execute permissions, but this ensures it works

4. Verify Installation

Test that Terraforms is properly installed:



```
[milanrana@Milans-Laptop ~ % terraform -version
Terraform v1.11.4
on darwin_arm64
milanrana@Milans-Laptop ~ %
```

Alternative Verification Commands

- Check version (short form)

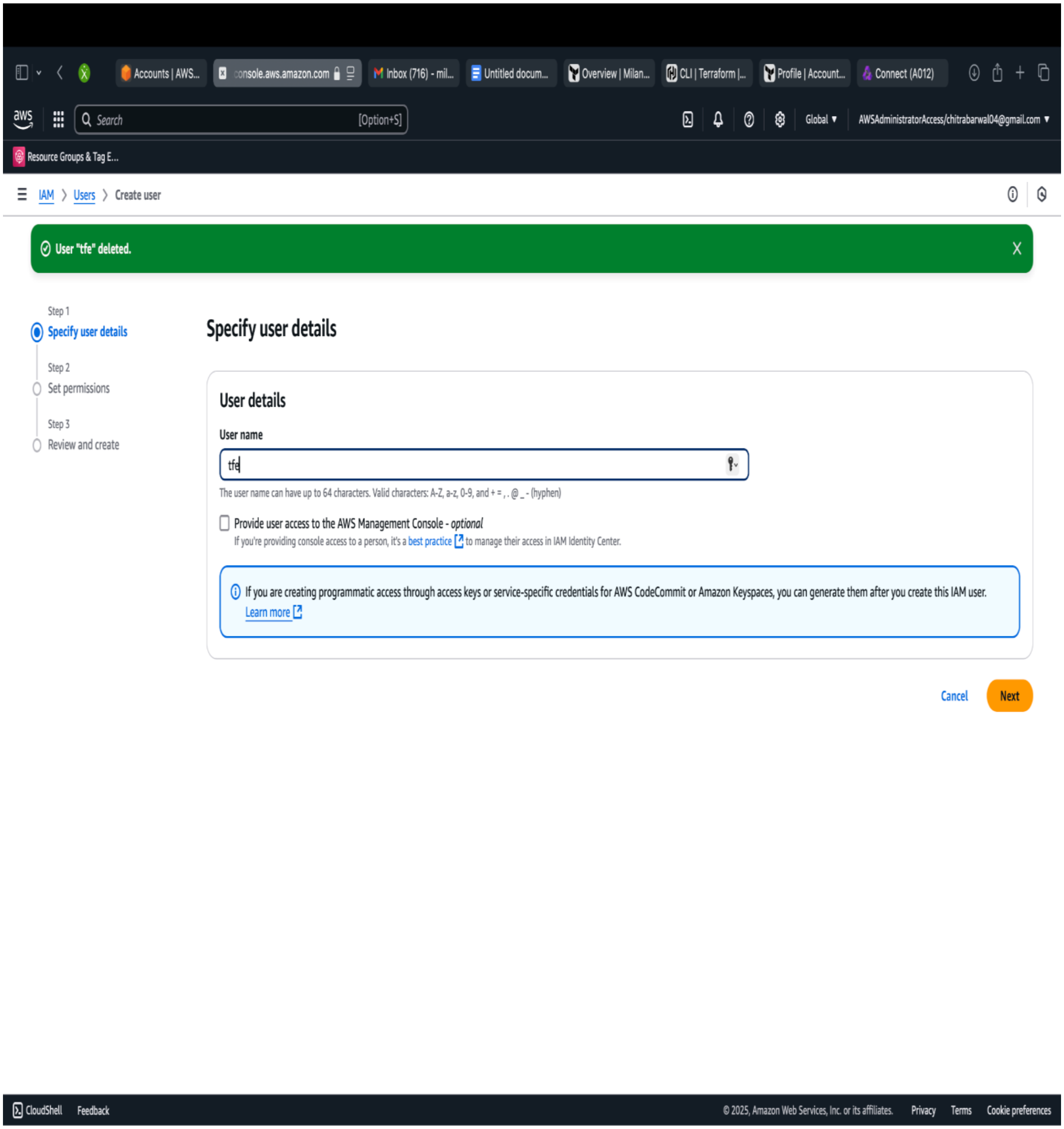
  terraform -v

- Show help to confirm it's working

  terraform --help

Set Up at AWS end-

Create a tfe user in your AWS account with AWS managed admin Policy



User "tfe" deleted.

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Specify user details

User details

User name

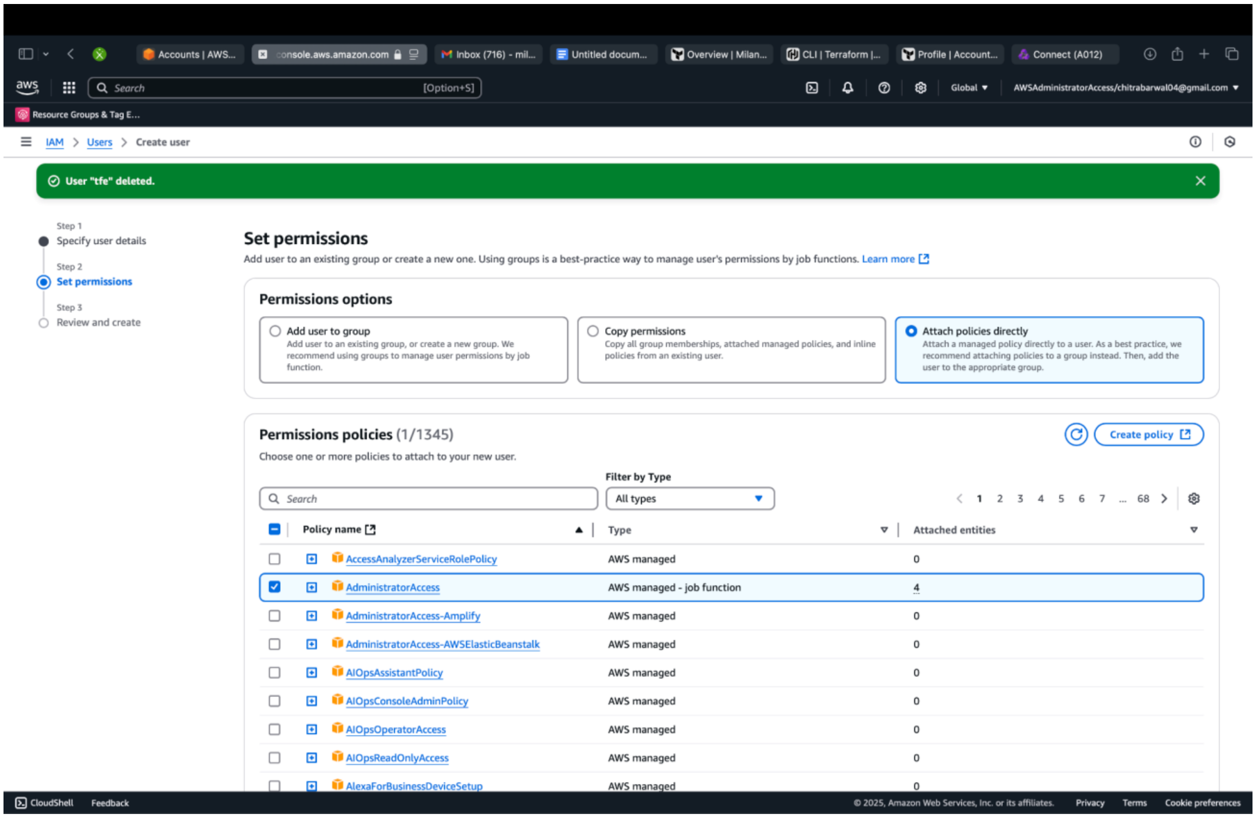
tfe

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

□ Provide user access to the AWS Management Console - *optional*

If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more



IAM > Users > Create user

User "tfe" deleted.

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more

Permissions options

○ Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

○ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

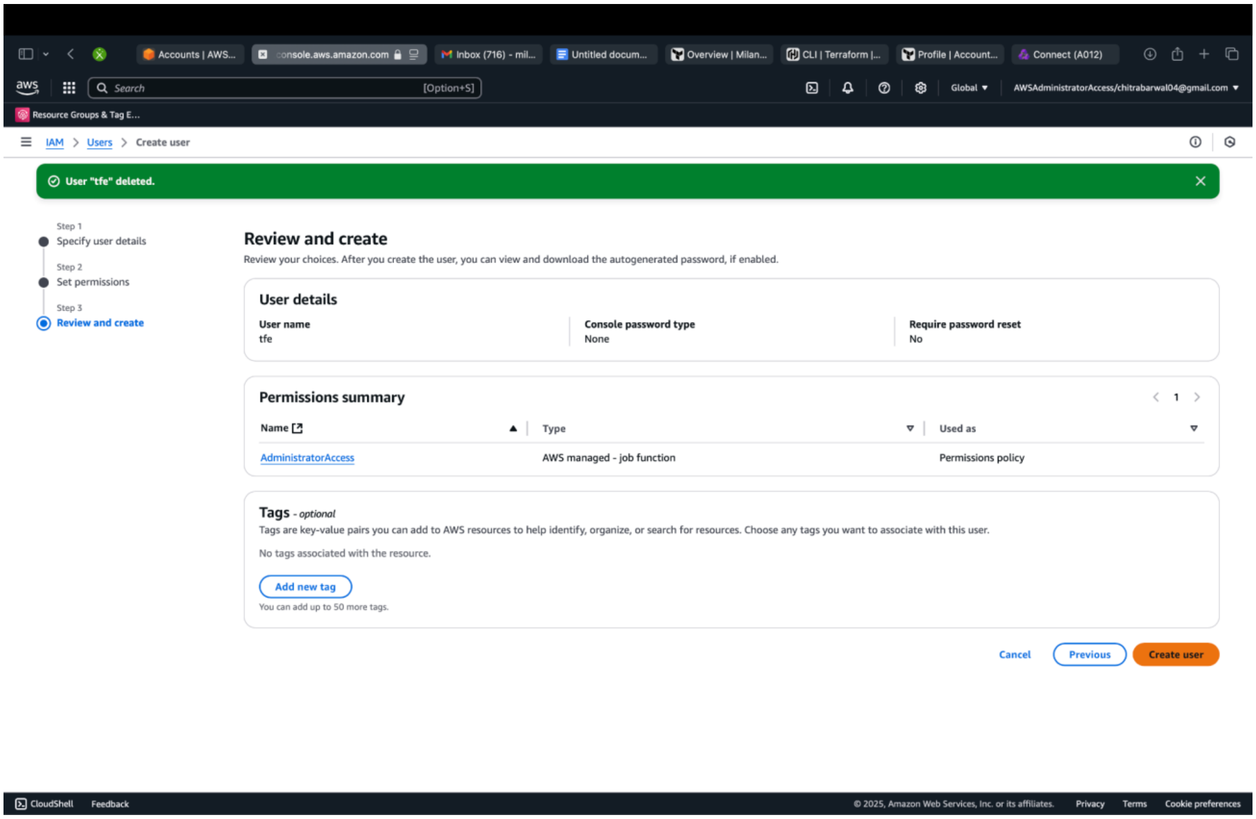
● Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1345)

Choose one or more policies to attach to your new user.

Filter by Type: All types

	Policy name	Type	Attached entities
□	AccessAnalyzerServiceRolePolicy	AWS managed	0
☑	AdministratorAccess	AWS managed - job function	4
□	AdministratorAccess-Amplify	AWS managed	0
□	AdministratorAccess-AWSElasticBeanstalk	AWS managed	0
□	AIOpsAssistantPolicy	AWS managed	0
□	AIOpsConsoleAdminPolicy	AWS managed	0
□	AIOpsOperatorAccess	AWS managed	0
□	AIOpsReadOnlyAccess	AWS managed	0
□	AlexaForBusinessDeviceSetup	AWS managed	0



IAM > Users > Create user

User "tfe" deleted.

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name	Console password type	Require password reset
tfe	None	No

Permissions summary

Name	Type	Used as
AdministratorAccess	AWS managed - job function	Permissions policy

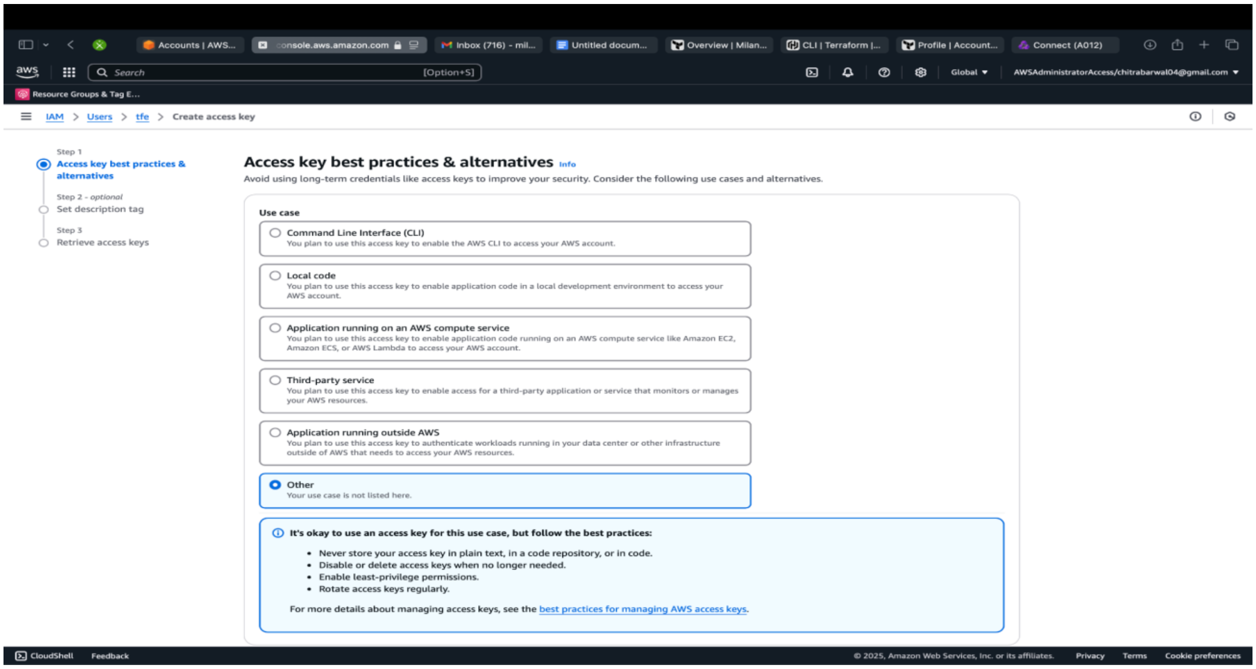
Tags - *optional*

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.



Access key best practices & alternatives Info

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

○ Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.

○ Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.

○ Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

○ Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

○ Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.

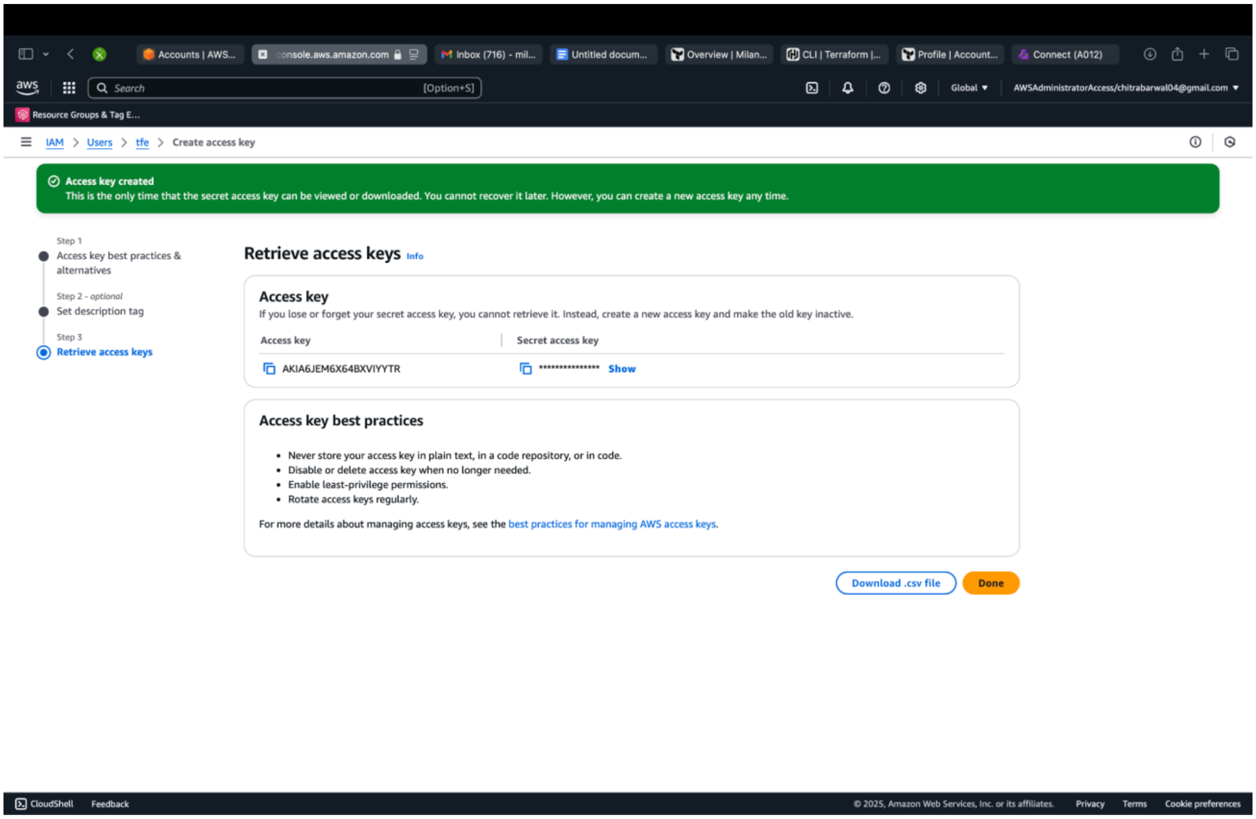
● Other
Your use case is not listed here.

It's okay to use an access key for this use case, but follow the best practices:

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access keys when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the best practices for managing AWS access keys.

Note the secret Key and access key



Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Retrieve access keys Info

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIA6JEM6X64BXVIYYTR	*************** Show

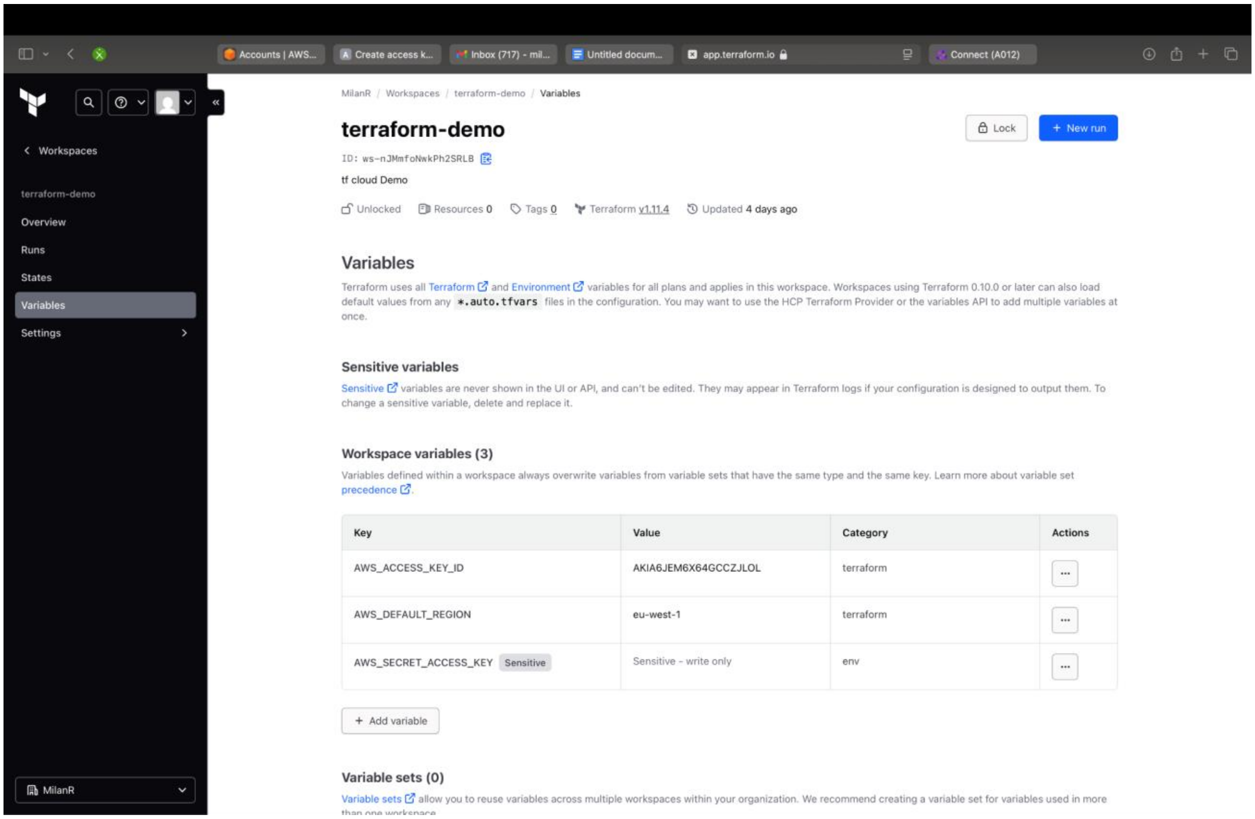
Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the best practices for managing AWS access keys.

Go to your variable in workspace and create the following variables as shown.

You would need to enter the values of the access key and secret Key created in last step Here. This step basically links your AWS account and terraform.



MilanR / Workspaces / terraform-demo / Variables

terraform-demo

ID: ws-nJMmfoNwkPh2SRLB

tf cloud Demo

Unlocked   Resources 0   Tags 0   Terraform v1.11.4   Updated 4 days ago

Variables

Terraform uses all Terraform and Environment variables for all plans and applies in this workspace. Workspaces using Terraform 0.10.0 or later can also load default values from any `*.auto.tfvars` files in the configuration. You may want to use the HCP Terraform Provider or the variables API to add multiple variables at once.

Sensitive variables

Sensitive variables are never shown in the UI or API, and can't be edited. They may appear in Terraform logs if your configuration is designed to output them. To change a sensitive variable, delete and replace it.

Workspace variables (3)

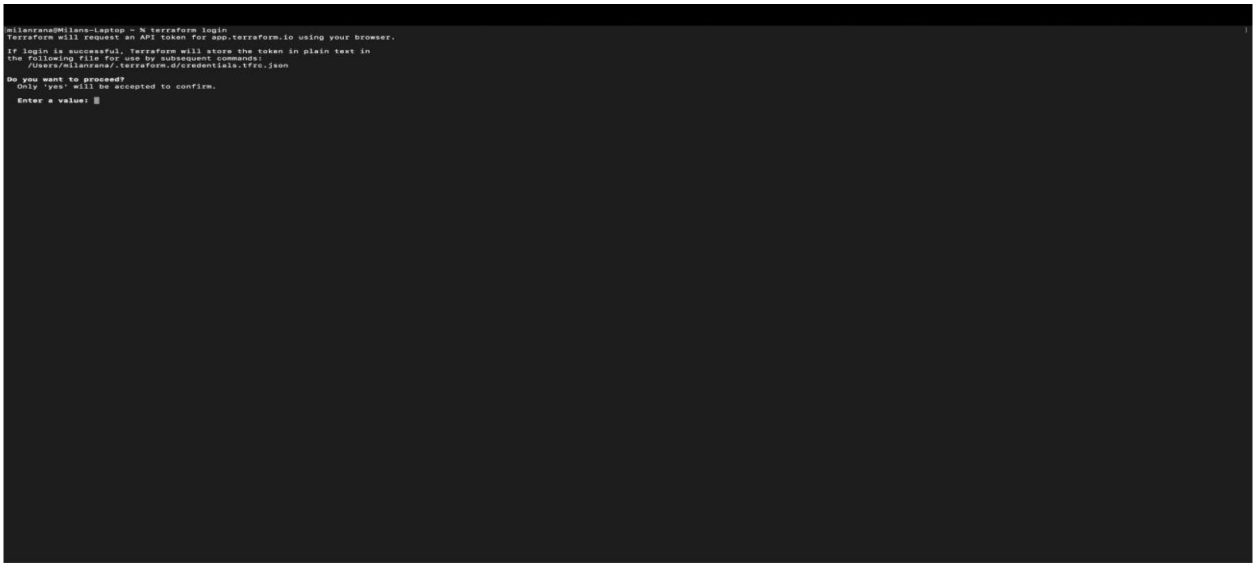
Variables defined within a workspace always overwrite variables from variable sets that have the same type and the same key. Learn more about variable set precedence.

Key	Value	Category	Actions
AWS_ACCESS_KEY_ID	AKIA6JEM6X64GCCZJLOL	terraform	...
AWS_DEFAULT_REGION	eu-west-1	terraform	...
AWS_SECRET_ACCESS_KEY Sensitive	Sensitive - write only	env	...

Variable sets (0)

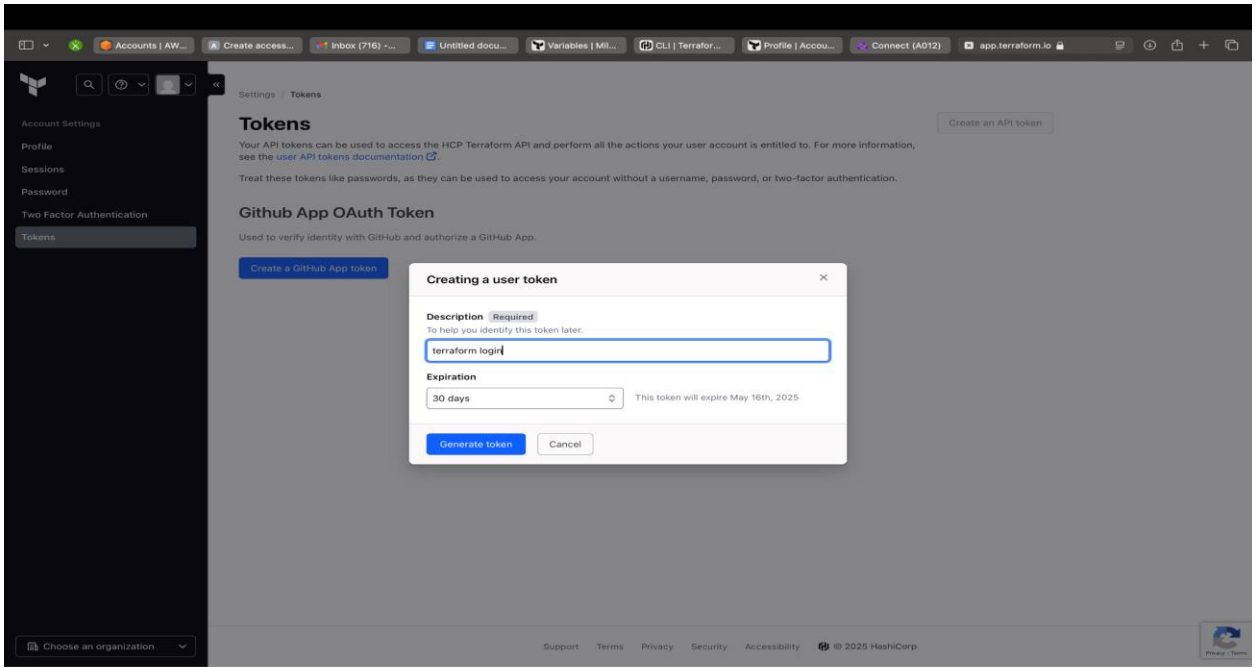
Variable sets allow you to reuse variables across multiple workspaces within your organization. We recommend creating a variable set for variables used in more than one workspace

On terminal /cmd use command terraform login and yes to confirm



```
milanrana@Milans-Laptop ~ % terraform login
Terraform will request an API token for app.terraform.io using your browser.

If login is successful, Terraform will store the token in plain text in
the following file for use by subsequent commands:
    /Users/milanrana/.terraform.d/credentials.tfrc.json

Do you want to proceed?
  Only 'yes' will be accepted to confirm.

  Enter a value: 
```

It should take u to generate a token on your tfe.



Settings / Tokens

Tokens

Your API tokens can be used to access the HCP Terraform API and perform all the actions your user account is entitled to. For more information, see the user API tokens documentation.

Treat these tokens like passwords, as they can be used to access your account without a username, password, or two-factor authentication.

Github App OAuth Token

Used to verify identity with GitHub and authorize a GitHub App.

Creating a user token

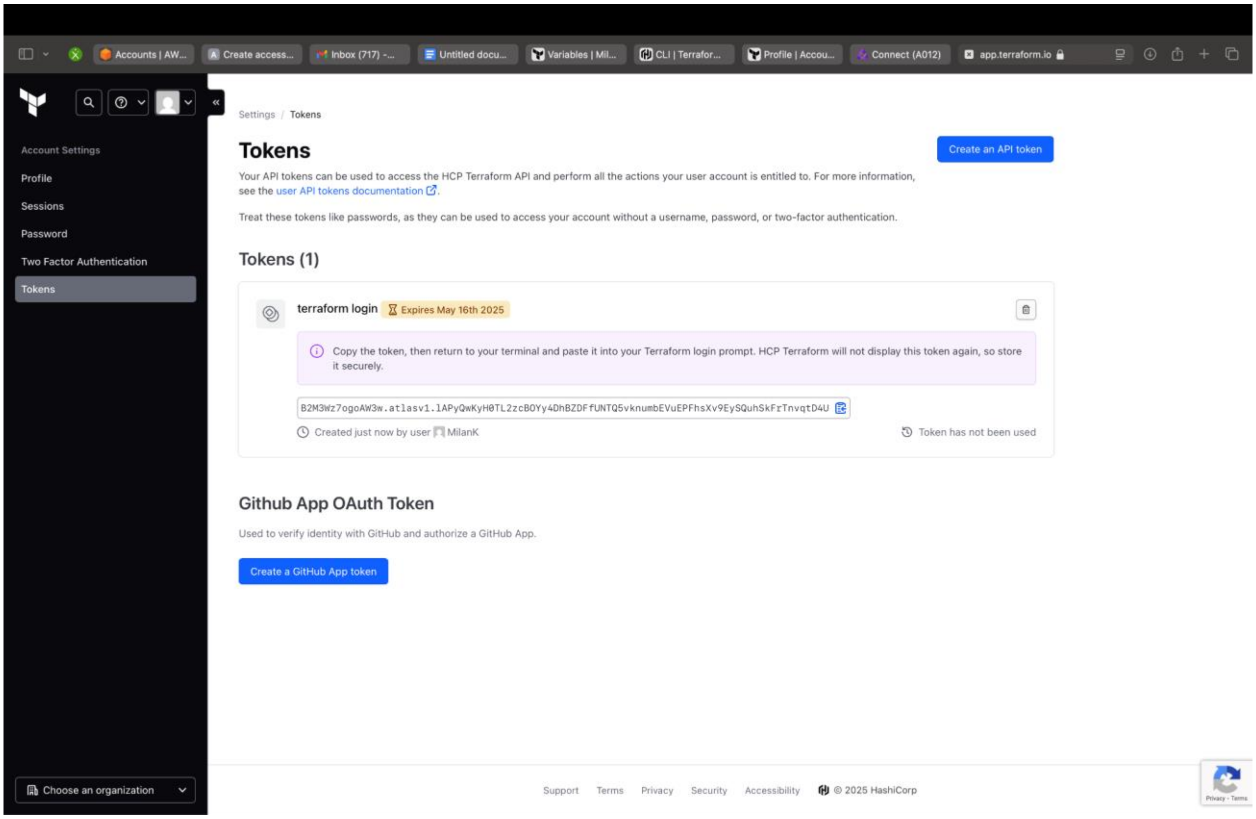
Description Required
To help you identify this token later.
terraform login

Expiration
30 days
This token will expire May 16th, 2025

Generate token Cancel

© 2025 HashiCorp

Copy the token generated and put it in termial/cmd



Settings / Tokens

Tokens

Your API tokens can be used to access the HCP Terraform API and perform all the actions your user account is entitled to. For more information, see the user API tokens documentation.

Treat these tokens like passwords, as they can be used to access your account without a username, password, or two-factor authentication.

Tokens (1)

terraform login Expires May 16th 2025

Copy the token, then return to your terminal and paste it into your Terraform login prompt. HCP Terraform will not display this token again, so store it securely.

```
B2M3Wz7ogoAW3w.atlasv1.lAPyQwKyH0TL2zcBOYy4DhBZDFfUNTQ5vknumbEVuEPFhsXv9EySQuhSkFrTnvqtD4U
```

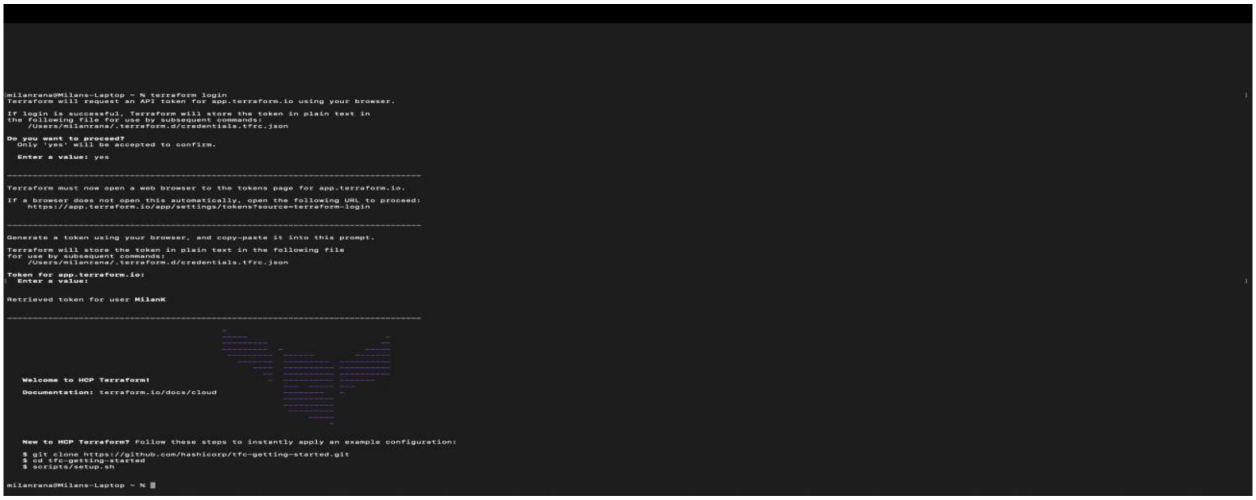
Created just now by user MilanK

Token has not been used

Github App OAuth Token

Used to verify identity with GitHub and authorize a GitHub App.

© 2025 HashiCorp



```
milanrana@Milans-Laptop ~ % terraform login
Terraform will request an API token for app.terraform.io using your browser.

If login is successful, Terraform will store the token in plain text in
the following file for use by subsequent commands:
    /Users/milanrana/.terraform.d/credentials.tfrc.json

Do you want to proceed?
  Only 'yes' will be accepted to confirm.

  Enter a value: yes

---------------------------------------------------------------------------------

Terraform must now open a web browser to the tokens page for app.terraform.io.

If a browser does not open this automatically, open the following URL to proceed:
    https://app.terraform.io/app/settings/tokens?source=terraform-login

---------------------------------------------------------------------------------

Generate a token using your browser, and copy-paste it into this prompt.

Terraform will store the token in plain text in the following file
for use by subsequent commands:
    /Users/milanrana/.terraform.d/credentials.tfrc.json

Token for app.terraform.io:
  Enter a value:

Retrieved token for user MilanK

---------------------------------------------------------------------------------

   Welcome to HCP Terraform!

   Documentation: terraform.io/docs/cloud

   New to HCP Terraform? Follow these steps to instantly apply an example configuration:

   $ git clone https://github.com/hashicorp/tfc-getting-started.git
   $ cd tfc-getting-started
   $ scripts/setup.sh

milanrana@Milans-Laptop ~ %
```

Create a folder for this demo and navigate into it

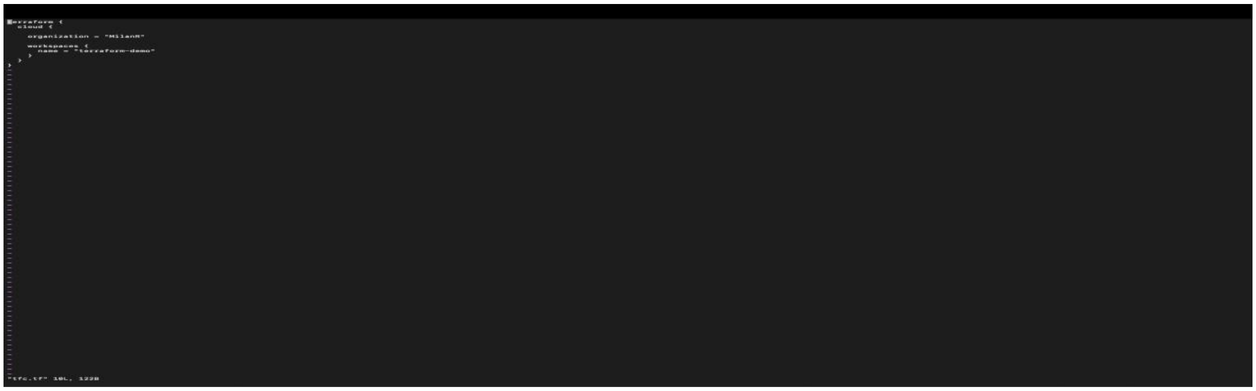
- Create a Terraform configuration file as per TFE workspace instructions

- Use vim <filename.tf> to create the file

- Press i to enter insert mode

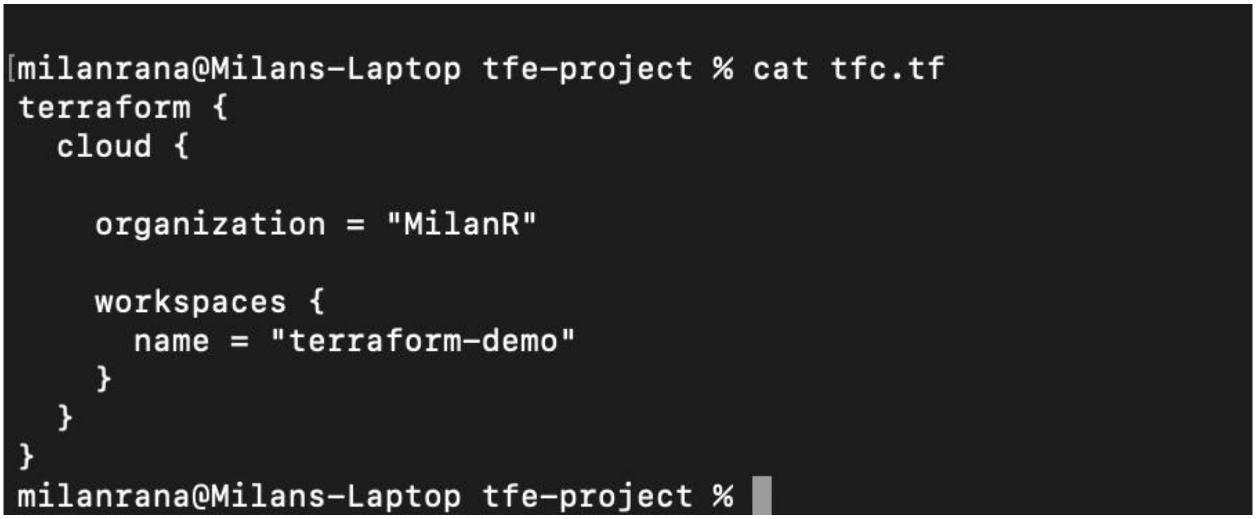
- Paste the required configuration code

- Press Esc to exit insert mode
- Type :wq! to save and exit the file



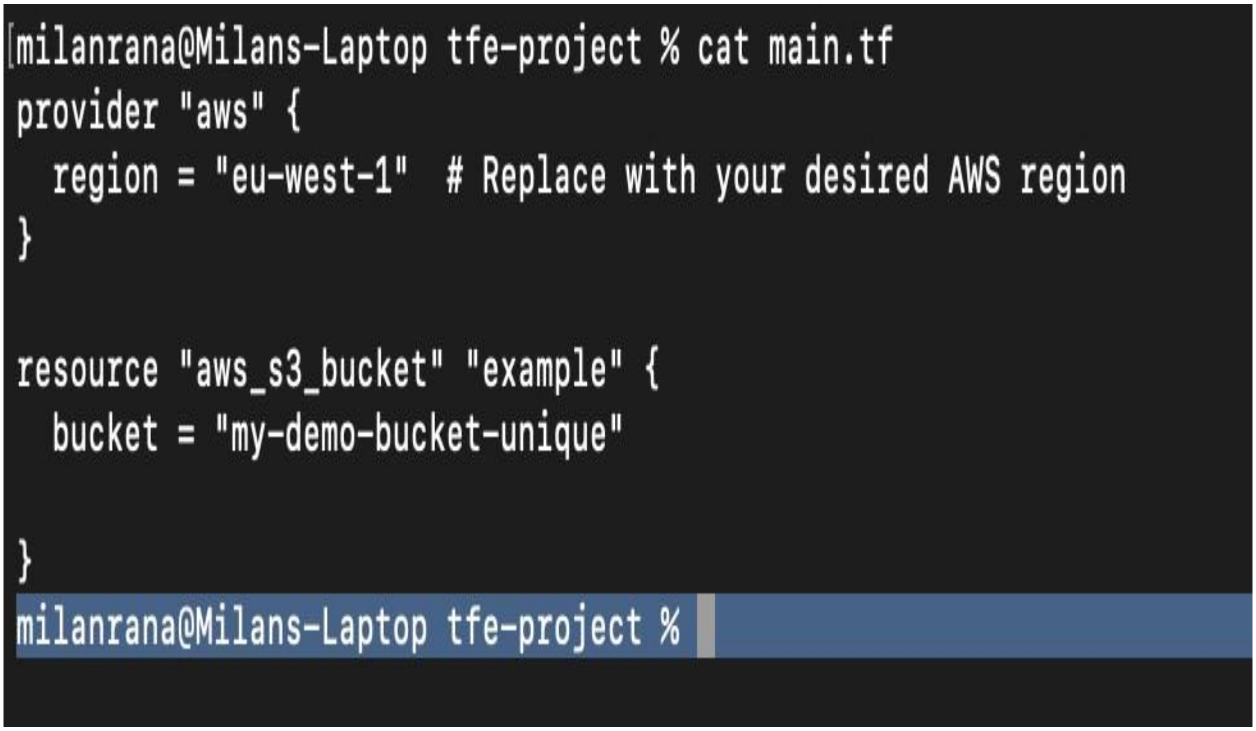
```
terraform {
  cloud {

    organization = "MilanR"

    workspaces {
      name = "terraform-demo"
    }
  }
}
"tfc.tf" 10L, 122B
```

Once saved u can check the content of file using cat



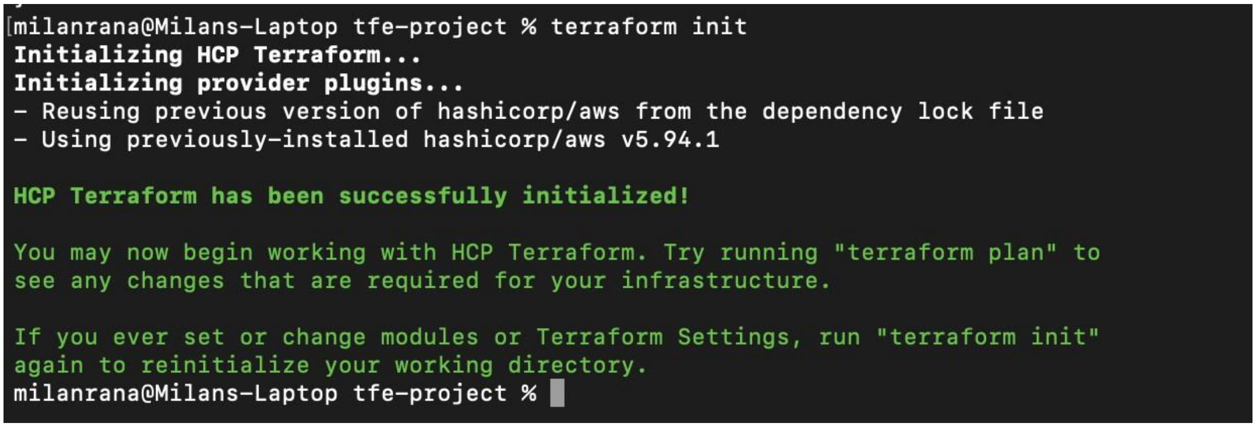
```
[milanrana@Milans-Laptop tfe-project % cat tfc.tf
terraform {
  cloud {

    organization = "MilanR"

    workspaces {
      name = "terraform-demo"
    }
  }
}
milanrana@Milans-Laptop tfe-project %
```

Similarly create a main.tf file to create your first resource as below and save the file and do cat to see if it is saved as per expectations



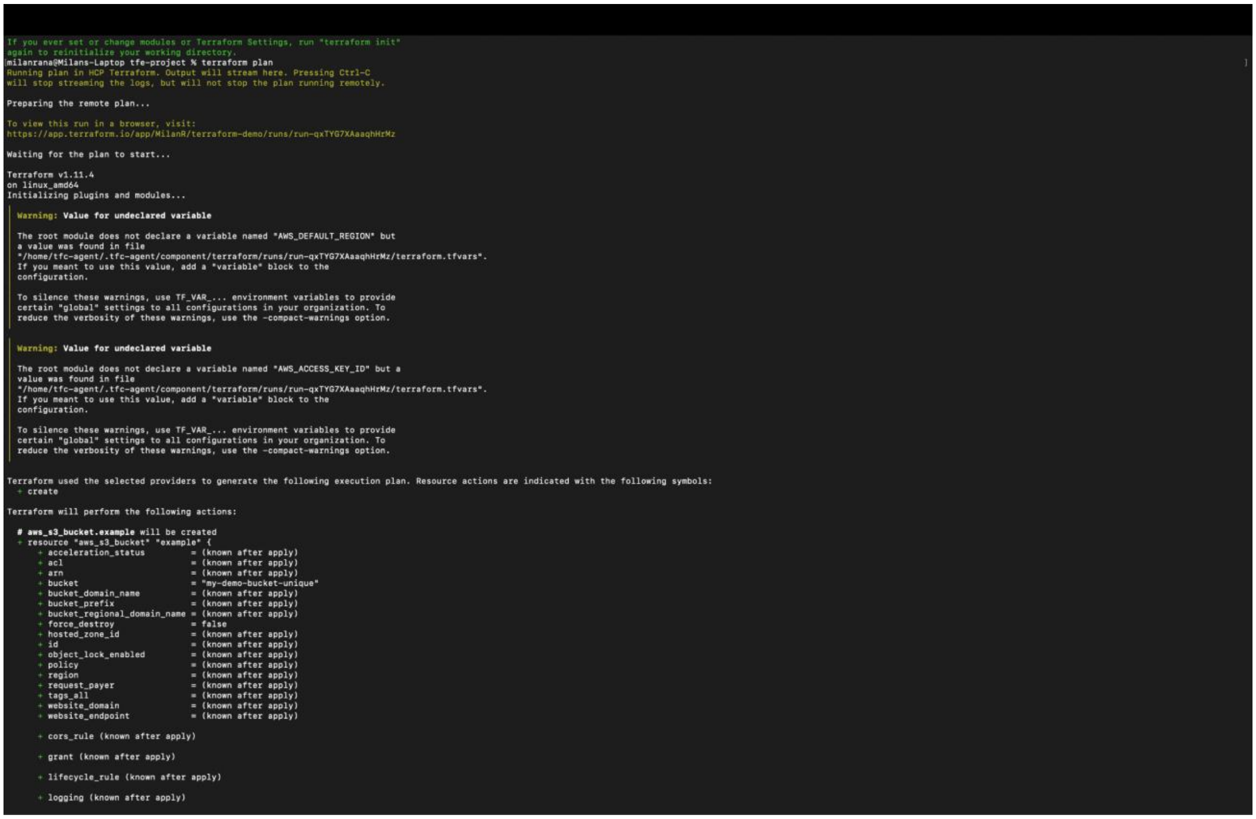
```
[milanrana@Milans-Laptop tfe-project % cat main.tf
provider "aws" {
  region = "eu-west-1"  # Replace with your desired AWS region
}


resource "aws_s3_bucket" "example" {
  bucket = "my-demo-bucket-unique"


}
milanrana@Milans-Laptop tfe-project %
```

Run terraform init to initialize



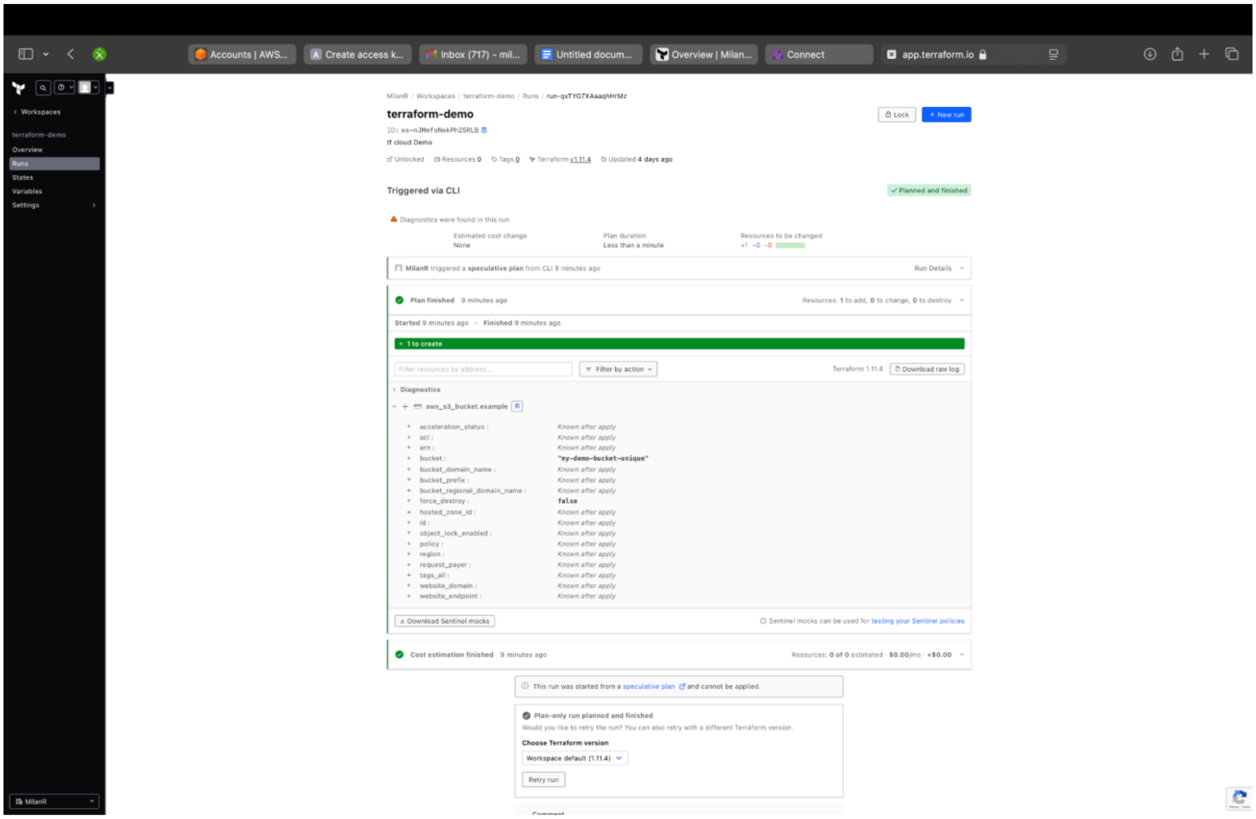
```
[milanrana@Milans-Laptop tfe-project % terraform init
Initializing HCP Terraform...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.94.1

HCP Terraform has been successfully initialized!

You may now begin working with HCP Terraform. Try running "terraform plan" to
see any changes that are required for your infrastructure.

If you ever set or change modules or Terraform Settings, run "terraform init"
again to reinitialize your working directory.
milanrana@Milans-Laptop tfe-project %
```

Followed by tfe plan



```
If you ever set or change modules or Terraform Settings, run "terraform init"
again to reinitialize your working directory.
milanrana@Milans-Laptop tfe-project % terraform plan
Running plan in HCP Terraform. Output will stream here. Pressing Ctrl-C
will stop streaming the logs, but will not stop the plan running remotely.

Preparing the remote plan...

To view this run in a browser, visit:
https://app.terraform.io/app/MilanR/terraform-demo/runs/run-qxTYG7XAaaqhHrMz

Waiting for the plan to start...

Terraform v1.11.4
on linux_amd64
Initializing plugins and modules...
│
│ Warning: Value for undeclared variable
│
│ The root module does not declare a variable named "AWS_DEFAULT_REGION" but
│ a value was found in file
│ "/home/tfc-agent/.tfc-agent/component/terraform/runs/run-qxTYG7XAaaqhHrMz/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 "AWS_ACCESS_KEY_ID" but a
│ value was found in file
│ "/home/tfc-agent/.tfc-agent/component/terraform/runs/run-qxTYG7XAaaqhHrMz/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.
│

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_s3_bucket.example will be created
  + resource "aws_s3_bucket" "example" {
      + acceleration_status         = (known after apply)
      + acl                         = (known after apply)
      + arn                         = (known after apply)
      + bucket                      = "my-demo-bucket-unique"
      + bucket_domain_name          = (known after apply)
      + bucket_prefix               = (known after apply)
      + bucket_regional_domain_name = (known after apply)
      + force_destroy               = false
      + hosted_zone_id              = (known after apply)
      + id                          = (known after apply)
      + object_lock_enabled         = (known after apply)
      + policy                      = (known after apply)
      + region                      = (known after apply)
      + request_payer               = (known after apply)
      + tags_all                    = (known after apply)
      + website_domain              = (known after apply)
      + website_endpoint            = (known after apply)

      + cors_rule (known after apply)

      + grant (known after apply)

      + lifecycle_rule (known after apply)

      + logging (known after apply)
```

You can view the same in your workspace



MilanR / Workspaces / terraform-demo / Runs / run-qxTYG7XAaaqhHrMz

terraform-demo

ID: ws-nJMmfoNwkPh2SRLB

tf cloud Demo

Unlocked · Resources 0 · Tags 0 · Terraform v1.11.4 · Updated 4 days ago

Triggered via CLI ✓ Planned and finished

Diagnostics were found in this run

Estimated cost change	Plan duration	Resources to be changed
None	Less than a minute	+1 ~0 -0

MilanR triggered a **speculative plan** from CLI 9 minutes ago

Plan finished 9 minutes ago — Resources: **1** to add, **0** to change, **0** to destroy

Started 9 minutes ago > **Finished** 9 minutes ago

+ 1 to create

Terraform 1.11.4

Diagnostics

aws_s3_bucket.example

+ acceleration_status :	*Known after apply*
+ acl :	*Known after apply*
+ arn :	*Known after apply*
+ bucket :	**"my-demo-bucket-unique"**
+ bucket_domain_name :	*Known after apply*
+ bucket_prefix :	*Known after apply*
+ bucket_regional_domain_name :	*Known after apply*
+ force_destroy :	**false**
+ hosted_zone_id :	*Known after apply*
+ id :	*Known after apply*
+ object_lock_enabled :	*Known after apply*
+ policy :	*Known after apply*
+ region :	*Known after apply*
+ request_payer :	*Known after apply*
+ tags_all :	*Known after apply*
+ website_domain :	*Known after apply*
+ website_endpoint :	*Known after apply*

Sentinel mocks can be used for testing your Sentinel policies

Cost estimation finished 9 minutes ago — Resources: **0 of 0** estimated · **$0.00**/mo · **+$0.00**

This run was started from a speculative plan and cannot be applied.

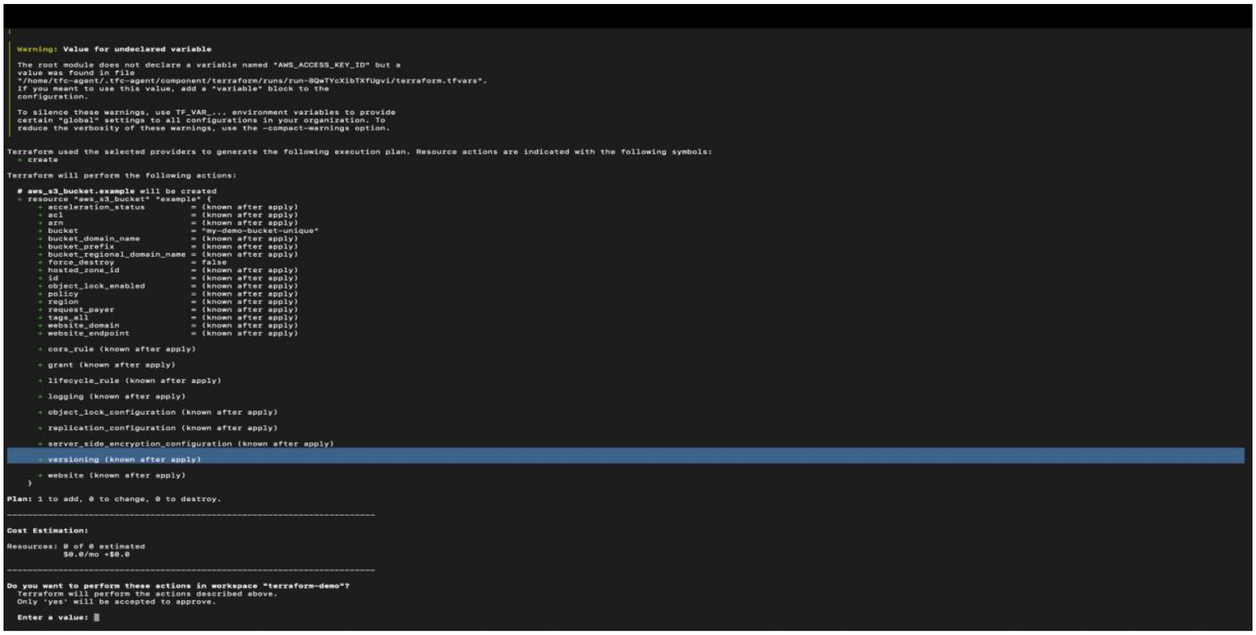
Plan-only run planned and finished

Would you like to retry the run? You can also retry with a different Terraform version.

Choose Terraform version

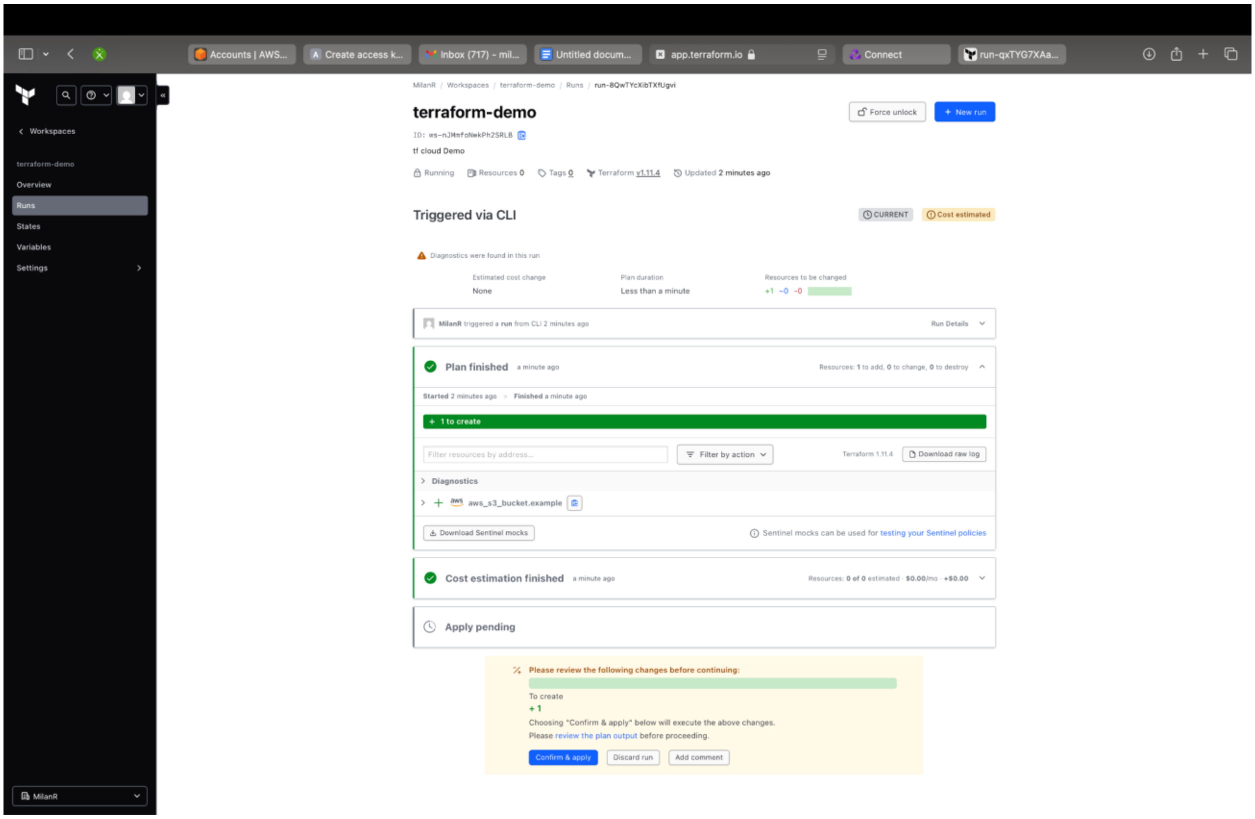
Workspace default (1.11.4)

This will be followed by terraform apply



```
Warning: Value for undeclared variable

The root module does not declare a variable named "AWS_ACCESS_KEY_ID" but a
value was found in file
"/home/tfc-agent/.tfc-agent/component/terraform/runs/run-8QwTYcXibTXfUgvi/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.

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_s3_bucket.example will be created
  + resource "aws_s3_bucket" "example" {
      + acceleration_status         = (known after apply)
      + acl                         = (known after apply)
      + arn                         = (known after apply)
      + bucket                      = "my-demo-bucket-unique"
      + bucket_domain_name          = (known after apply)
      + bucket_prefix               = (known after apply)
      + bucket_regional_domain_name = (known after apply)
      + force_destroy               = false
      + hosted_zone_id              = (known after apply)
      + id                          = (known after apply)
      + object_lock_enabled         = (known after apply)
      + policy                      = (known after apply)
      + region                      = (known after apply)
      + request_payer               = (known after apply)
      + tags_all                    = (known after apply)
      + website_domain              = (known after apply)
      + website_endpoint            = (known after apply)

      + cors_rule (known after apply)

      + grant (known after apply)

      + lifecycle_rule (known after apply)

      + logging (known after apply)

      + object_lock_configuration (known after apply)

      + replication_configuration (known after apply)

      + server_side_encryption_configuration (known after apply)

      + versioning (known after apply)

      + website (known after apply)
    }

Plan: 1 to add, 0 to change, 0 to destroy.

------------------------------------------------------------------------

Cost Estimation:

Resources: 0 of 0 estimated
           $0.0/mo +$0.0

------------------------------------------------------------------------

Do you want to perform these actions in workspace "terraform-demo"?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: 
```

Similar view in UI, where you can confirm apply



MilanR / Workspaces / terraform-demo / Runs / run-8QwTYcXibTXfUgvi

terraform-demo

Force unlock | + New run

ID: ws-nJHmfoNwkPh2SRLB

tf cloud Demo

Running | Resources 0 | Tags 0 | Terraform v1.11.4 | Updated 2 minutes ago

Triggered via CLI

CURRENT | Cost estimated

Diagnostics were found in this run

Estimated cost change	Plan duration	Resources to be changed
None	Less than a minute	+1 ~0 -0

MilanR triggered a run from CLI 2 minutes ago — Run Details

Plan finished a minute ago — Resources: 1 to add, 0 to change, 0 to destroy

Started 2 minutes ago > Finished a minute ago

+ 1 to create

Filter resources by address... | Filter by action | Terraform 1.11.4 | Download raw log

> Diagnostics

> + aws_s3_bucket.example

Download Sentinel mocks — Sentinel mocks can be used for testing your Sentinel policies

Cost estimation finished a minute ago — Resources: 0 of 0 estimated · $0.00/mo · +$0.00

Apply pending

Please review the following changes before continuing:

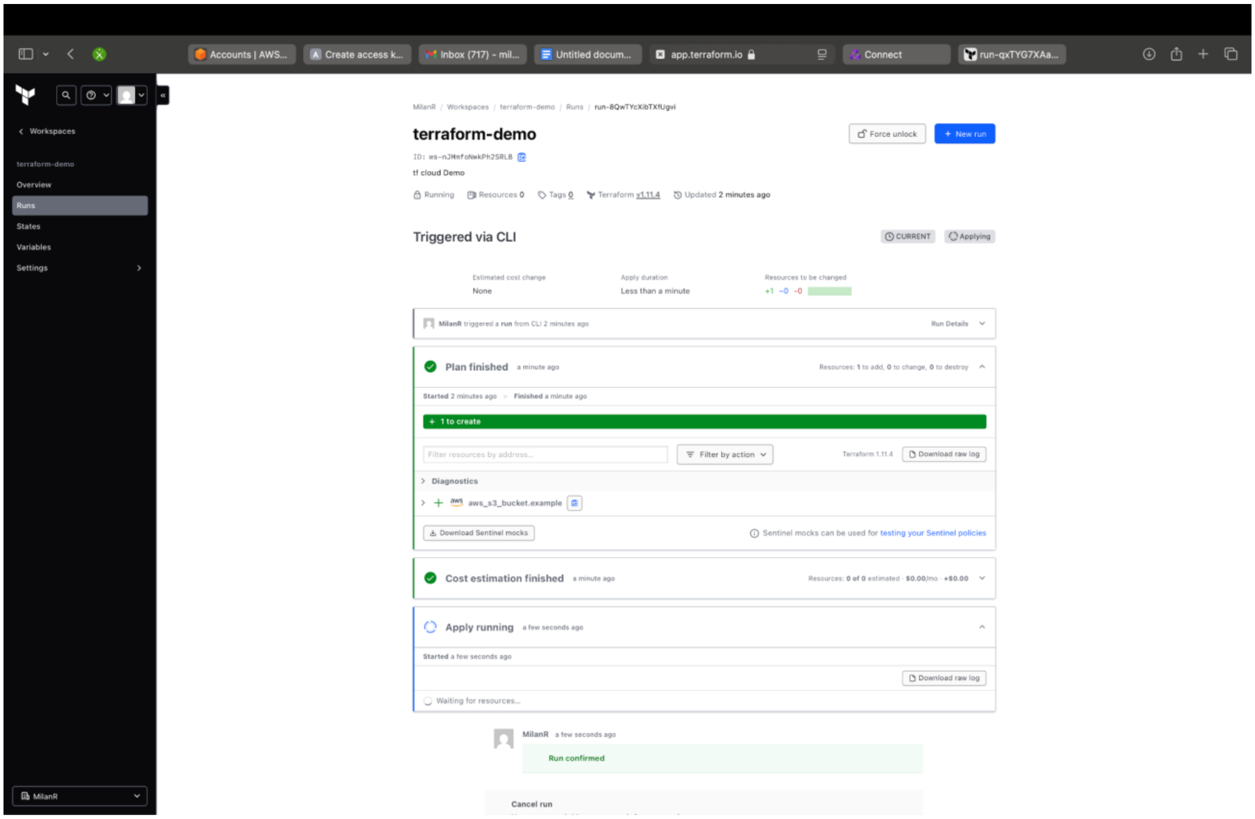
To create

+ 1

Choosing "Confirm & apply" below will execute the above changes.

Please review the plan output before proceeding.

Confirm & apply | Discard run | Add comment



MilanR / Workspaces / terraform-demo / Runs / run-8QwTYcXibTXfUgvi

terraform-demo

Force unlock | + New run

ID: ws-nJHmfoNwkPh2SRLB

tf cloud Demo

Running | Resources 0 | Tags 0 | Terraform v1.11.4 | Updated 2 minutes ago

Triggered via CLI

CURRENT | Applying

Estimated cost change	Apply duration	Resources to be changed
None	Less than a minute	+1 ~0 -0

MilanR triggered a run from CLI 2 minutes ago — Run Details

Plan finished a minute ago — Resources: 1 to add, 0 to change, 0 to destroy

Started 2 minutes ago > Finished a minute ago

+ 1 to create

Filter resources by address... | Filter by action | Terraform 1.11.4 | Download raw log

> Diagnostics

> + aws_s3_bucket.example

Download Sentinel mocks — Sentinel mocks can be used for testing your Sentinel policies

Cost estimation finished a minute ago — Resources: 0 of 0 estimated · $0.00/mo · +$0.00

Apply running a few seconds ago

Started a few seconds ago

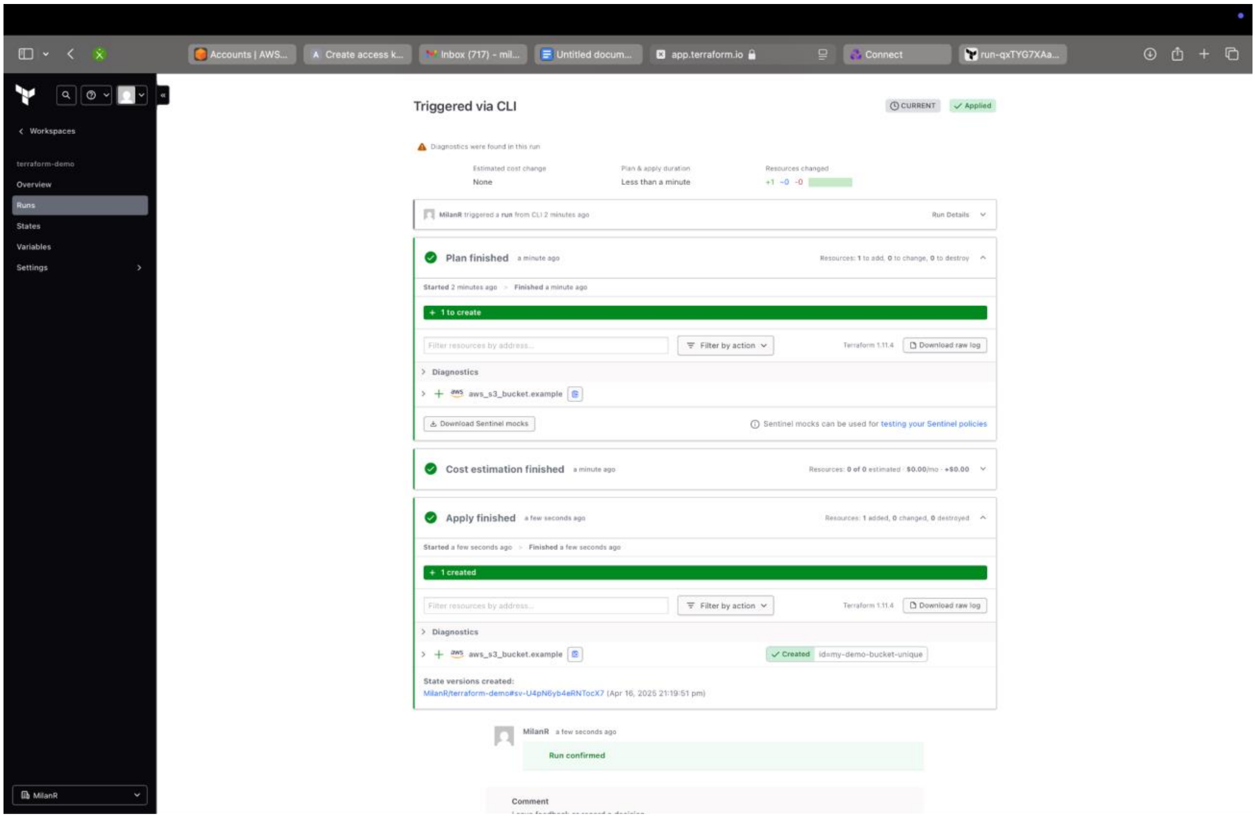
Download raw log

Waiting for resources...

MilanR a few seconds ago

Run confirmed

Cancel run



Triggered via CLI

CURRENT ✓ Applied

Diagnostics were found in this run

Estimated cost change	Plan & apply duration	Resources changed
None	Less than a minute	+1 ~0 -0

MilanR triggered a run from CLI 2 minutes ago — Run Details

Plan finished a minute ago — Resources: 1 to add, 0 to change, 0 to destroy

Started 2 minutes ago > Finished a minute ago

+ 1 to create

Filter resources by address... | Filter by action | Terraform 1.11.4 | Download raw log

Diagnostics

+ aws_s3_bucket.example

Download Sentinel mocks — Sentinel mocks can be used for testing your Sentinel policies

Cost estimation finished a minute ago — Resources: 0 of 0 estimated · $0.00/mo · +$0.00

Apply finished a few seconds ago — Resources: 1 added, 0 changed, 0 destroyed

Started a few seconds ago > Finished a few seconds ago

+ 1 created

Filter resources by address... | Filter by action | Terraform 1.11.4 | Download raw log

Diagnostics

+ aws_s3_bucket.example ✓ Created id=my-demo-bucket-unique

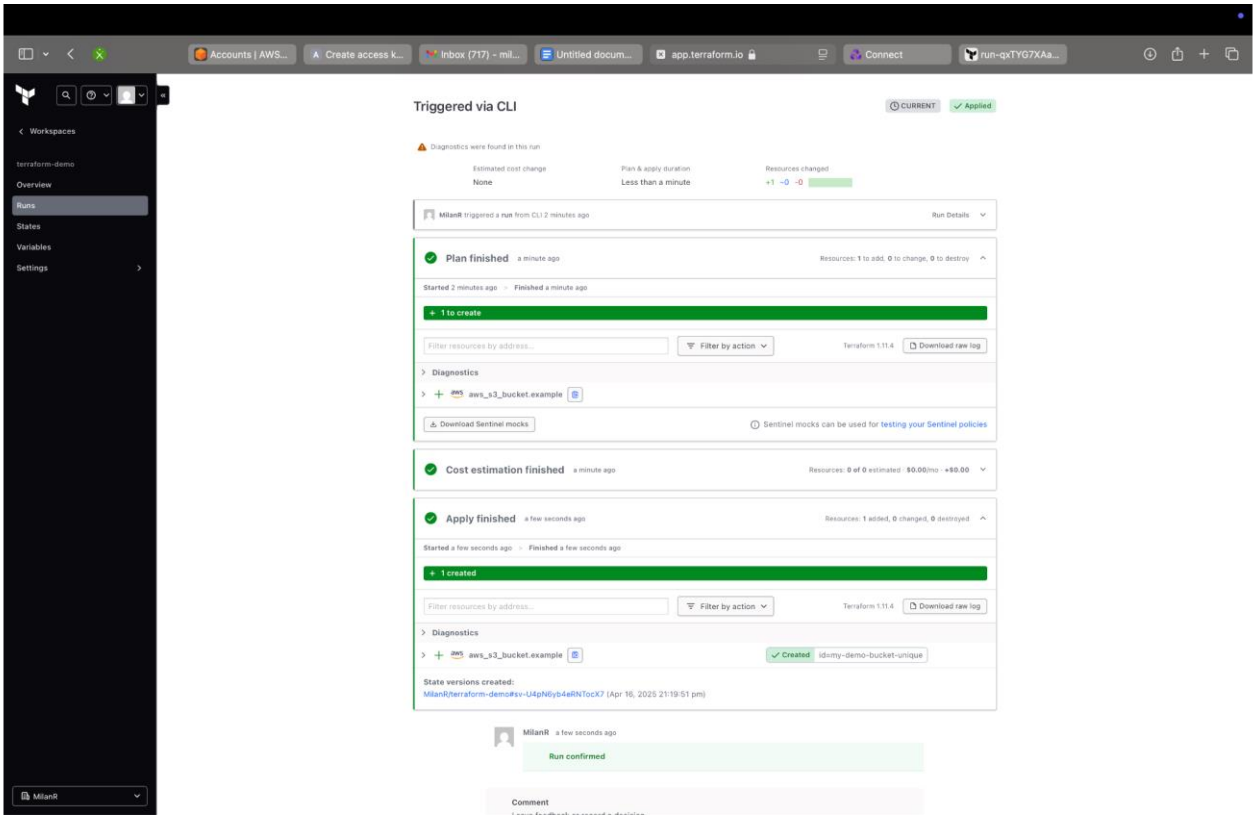
State versions created:
MilanR/terraform-demo#sv-U4pN6yb4eRNTocX7 (Apr 16, 2025 21:19:51 pm)

MilanR a few seconds ago

Run confirmed

Comment

You can verify the same resource created in your AWS account by going to s3


Triggered via CLI

CURRENT ✓ Applied

Diagnostics were found in this run

Estimated cost change: None — Plan & apply duration: Less than a minute — Resources changed: +1 ~0 -0

MilanR triggered a run from CLI 2 minutes ago — Run Details

Plan finished a minute ago — Resources: 1 to add, 0 to change, 0 to destroy

Started 2 minutes ago > Finished a minute ago

+ 1 to create

Diagnostics

+ aws_s3_bucket.example

Download Sentinel mocks — Sentinel mocks can be used for testing your Sentinel policies

Cost estimation finished a minute ago — Resources: 0 of 0 estimated · $0.00/mo · +$0.00

Apply finished a few seconds ago — Resources: 1 added, 0 changed, 0 destroyed

Started a few seconds ago > Finished a few seconds ago

+ 1 created

Diagnostics

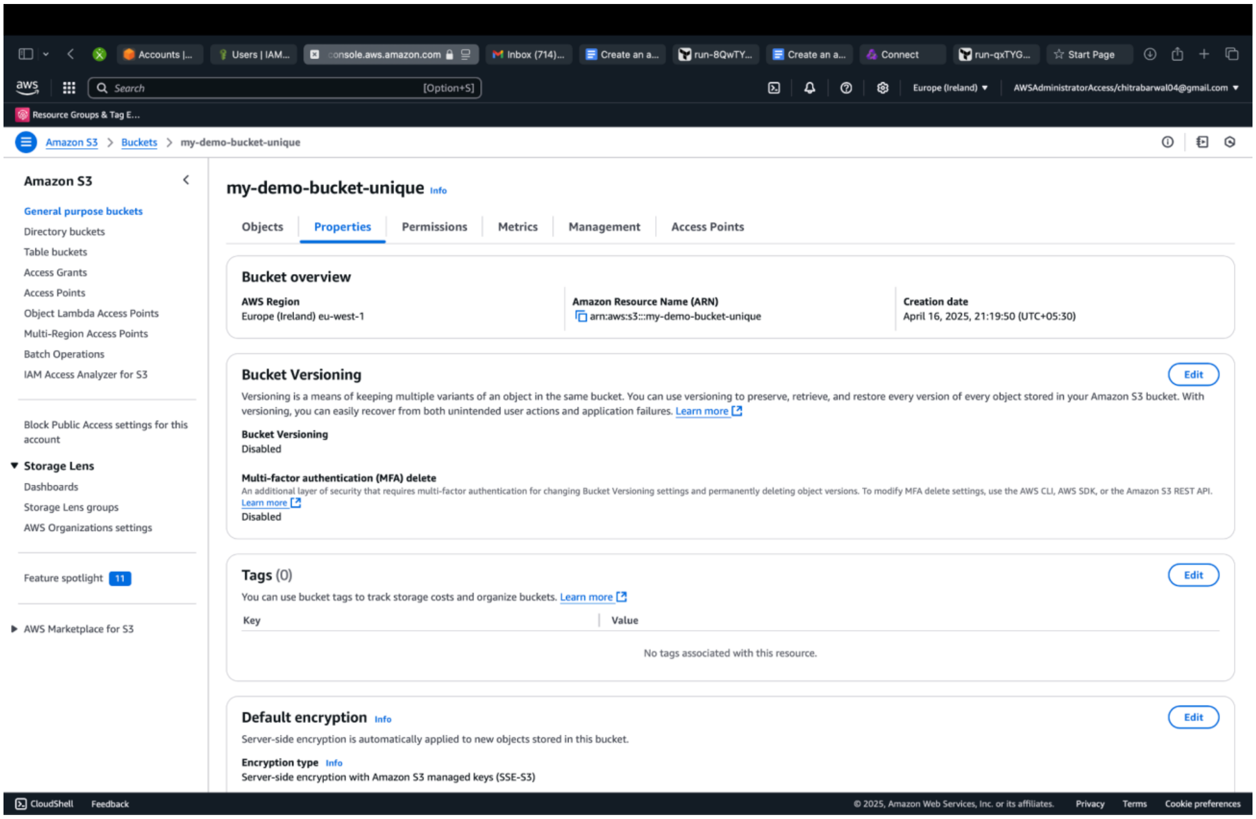
+ aws_s3_bucket.example ✓ Created id=my-demo-bucket-unique

State versions created:
MilanR/terraform-demo#sv-U4pN6yb4eRNTocX7 (Apr 16, 2025 21:19:51 pm)

MilanR a few seconds ago

Run confirmed

Comment



my-demo-bucket-unique

Bucket overview

AWS Region
Europe (Ireland) eu-west-1

Amazon Resource Name (ARN)
arn:aws:s3:::my-demo-bucket-unique

Creation date
April 16, 2025, 21:19:50 (UTC+05:30)

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning
Disabled

Multi-factor authentication (MFA) delete
An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. Learn more
Disabled

Tags (0)

You can use bucket tags to track storage costs and organize buckets. Learn more

Key	Value

No tags associated with this resource.

Default encryption

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type
Server-side encryption with Amazon S3 managed keys (SSE-S3)