Terraform Cloud with Google Cloud

Overview

This lab introduces Terraform Cloud and will demonstrate moving state storage and command execution off the local machine and into Terraform Cloud.

Lab Setup

For each lab, you are provided with a new Google Cloud project and a set of resources for a fixed time at no cost.

1. Note the lab's access time (for example, `1:15:00`), and make sure you can finish within that time. There is no pause feature. You can restart if needed, but you have to start at the beginning.

2. When ready, click **Start lab.**

3. Note your lab credentials (Username and Password). You will use them to sign in to the Google Cloud Console.

4. Click **Open Google Console**.

5. Click **Use another account** and copy/paste credentials for this lab into the prompts. If you use other credentials, you'll receive errors or incur charges.

6. Accept the terms and skip the recovery resource page.

Do not click **End Lab** unless you have finished the lab or want to restart it. This clears your work and removes the project.

Terraform and Cloud Shell Setup

1. In the Cloud Console, click Activate Cloud Shell >_

2. If prompted, click Continue.

3. Select the **Open in a new window** icon ↗

4. Select the **Open Editor** icon ✎

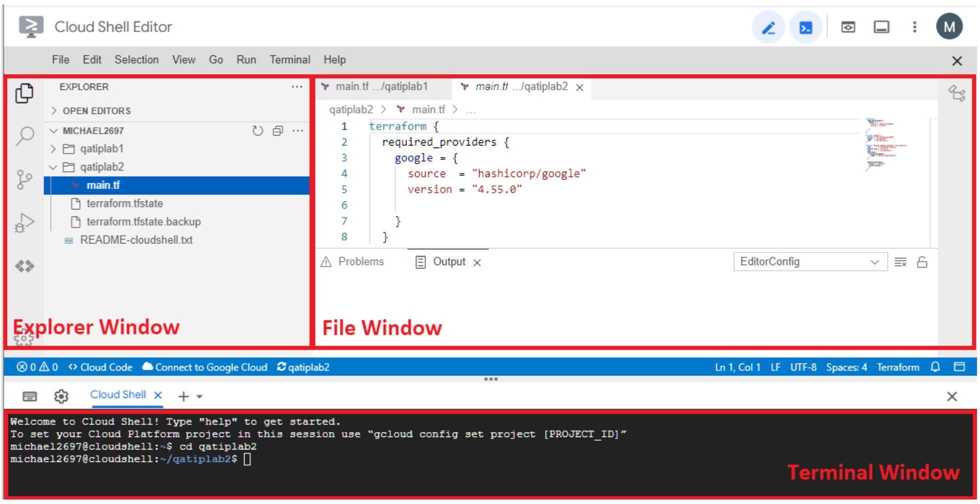
5. Select **Open Home Workspace** if explorer pane does not display.

6. Create a directory for your Terraform configuration and initial files by running the following commands in the terminal:

   ```
   cd ~
   ```

```
mkdir ./lab
cd ./lab
touch main.tf
```

7. In the Explorer, navigate to and click on main.tf to open the empty file for editing...



```
terraform {
  required_providers {
    google = {
      source  = "hashicorp/google"
      version = "4.55.0"

    }
  }
```

```
Welcome to Cloud Shell! Type "help" to get started.
To set your Cloud Platform project in this session use "gcloud config set project [PROJECT_ID]"
michael2697@cloudshell:~$ cd qatiplab2
michael2697@cloudshell:~/qatiplab2$ 
```

8. You now have 2 browser tabs open. The current tab will be referred to as the `IDE` throughout these instructions.

9. Switch to the Google console tab and close the cloud shell. This tab will be referred to as the `Console` throughout these instructions.

10. Switch back to the `IDE`

11. Execute all terraform commands from within the lab directory

12. Paste the following into **main.tf**, updating **line 11** with your **project id**, and save the changes

```
terraform {
  required_providers {
    google = {
      source = "hashicorp/google"
      version = "4.55.0"
    }
  }
}

provider "google" {
  project = "terraform-2697"
  region = "us-central1"
}

resource "google_compute_network" "lab-vpc" {
    name = "lab-vpc"
    auto_create_subnetworks = false
```

```
}

resource "google_compute_subnetwork" "public-subnet" {
  name = "public-subnet"
  ip_cidr_range = "10.0.1.0/24"
  region        = "us-central1"
  network = google_compute_network.lab-vpc.id
}

resource "google_compute_subnetwork" "private-subnet" {
  name = "private-subnet"
  ip_cidr_range = "10.0.2.0/24"
  region        = "us-central1"
  network = google_compute_network.lab-vpc.id
}

resource "google_compute_router" "lab-router" {
   name = "lab-router"
  network = google_compute_network.lab-vpc.id
}

resource "google_compute_router_nat" "lab-nat" {
  name = "lab-nat"
  router = google_compute_router.lab-router.name
  region = google_compute_router.lab-router.region
  nat_ip_allocate_option = "AUTO_ONLY"
  source_subnetwork_ip_ranges_to_nat = "LIST_OF_SUBNETWORKS"

  subnetwork {
    name = google_compute_subnetwork.private-subnet.id
    source_ip_ranges_to_nat = ["ALL_IP_RANGES"]
  }
}

resource "google_compute_instance" "priv-vm" {
  name         = "priv-vm"
  machine_type = "e2-small"
  zone         = "us-central1-a"
  allow_stopping_for_update = true
  tags = ["priv-subnet-vm"]

  boot_disk {
    initialize_params {
      image = "debian-cloud/debian-11"
    }
  }
  network_interface {
    subnetwork = google_compute_subnetwork.private-subnet.id
  }
}

resource "google_compute_instance" "pub-vm" {
  name         = "pub-vm"
  machine_type = "e2-small"
  zone         = "us-central1-a"
  allow_stopping_for_update = true
  tags = ["pub-subnet-vm"]

  boot_disk {
   initialize_params {
      image = "debian-cloud/debian-11"
    }
  }
  network_interface {
    subnetwork = google_compute_subnetwork.public-subnet.id

    access_config {
      // Ephemeral public IP
    }
  }
}

resource "google_compute_firewall" "lab-private-fw-rules" {
  name    = "lab-private-firewall"
  network = google_compute_network.lab-vpc.name

  allow {
    protocol = "all"
  }
  source_tags = ["pub-subnet-vm"]
  target_tags = ["priv-subnet-vm"]
}

resource "google_compute_firewall" "lab-public-fw-rules" {
  name    = "lab-public-firewall"
  network = google_compute_network.lab-vpc.name

  allow {
    protocol = "tcp"
    ports    = ["22"]
  }
  source_ranges =["0.0.0.0/0"]
  target_tags = ["pub-subnet-vm"]
}
```

The Challenge

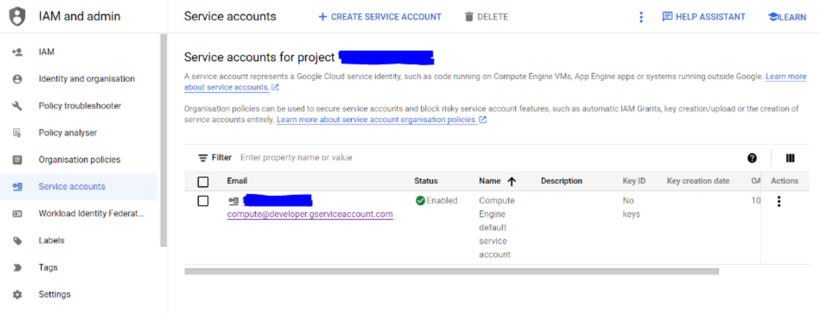
The terraform state file in currently held on a single computer from which all terraform commands are executed. Whilst possibly suitable for small/test/poc, production environments should have a more robust and resilient Terraform configuration. This lab will introduce Terraform Cloud and the challenge is to move state storage and command execution off the local machine and into Terraform Cloud.

This is a fully guided lab, but feel free to examine the Terraform Cloud enviroment you will be creating.

Exercise 1. Obtain correctly formatted JSON key file for Terraform Cloud authentication

1. Switch to the Console and navigate to **IAM and admin**

2. Click into the Compute Engine default service account (In production, a dedicated service account would normally be created)



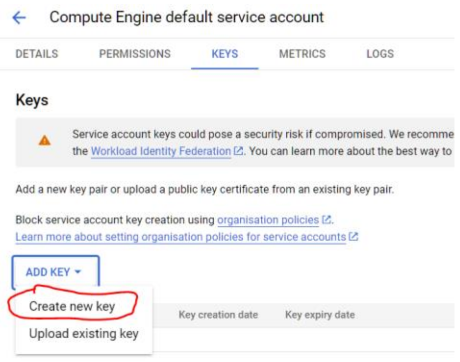
Service accounts for project [redacted]

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps or systems running outside Google. Learn more about service accounts.

Organisation policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload or the creation of service accounts entirely. Learn more about service account organisation policies.

□	Email	Status	Name ↑	Description	Key ID	Key creation date	OA	Actions
□	[redacted] compute@developer.gserviceaccount.com	✓ Enabled	Compute Engine default service account		No keys		10	⋮

3. Select KEYS, Choose Create a new key, JSON Key type...



Keys

Service account keys could pose a security risk if compromised. We recomme the Workload Identity Federation. You can learn more about the best way to

Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using organisation policies.
Learn more about setting organisation policies for service accounts

Key creation date	Key expiry date

4. A copy of the newly created key will be downloaded to your local machine

5. Navigate to the **IDE**, right-click over the lab folder in the explorer window and upload the json file from your local machine

6. Right-click on the json file and rename it to **key.json** (This is not strictly necessary but done here for lab instruction clarity)

7. Remove all line-breaks from the file by entering the following command in the terminal, ensuring you are in the lab directory...

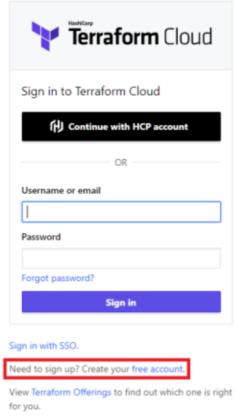
   ```
   cat key.json | jq -c >key.txt
   ```

8. Right-click and download **key.txt** to your local machine for later use.

Exercise 2. Signing up for a free Terraform Cloud account.

To successfully complete this lab, you will need an active email account. Consider creating a new burner email account that you can use purely for this lab.

1. Browse to https://app.terraform.io/..



Sign in to Terraform Cloud

Continue with HCP account

OR

Username or email

Password

Forgot password?

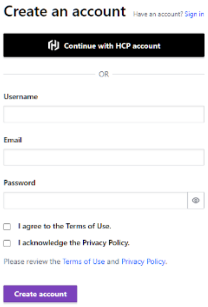
Sign in

Sign in with SSO.

Need to sign up? Create your free account.

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

2. Complete the enrolment form to create a new free account..



Create an account Have an account? Sign in

Continue with HCP account

OR

Username

Email

Password

□ I agree to the Terms of Use.

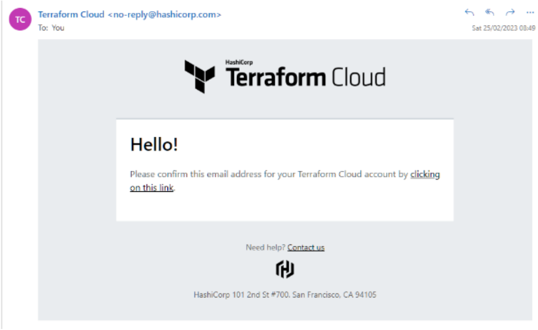
□ I acknowledge the Privacy Policy.

Please review the Terms of Use and Privacy Policy.

Create account

3. A confirmation email will be sent to you.

4. Acknowledge the email sent...



Terraform Cloud <no-reply@hashicorp.com>
To: You
Sat 25/02/2023 08:49

HashiCorp Terraform Cloud

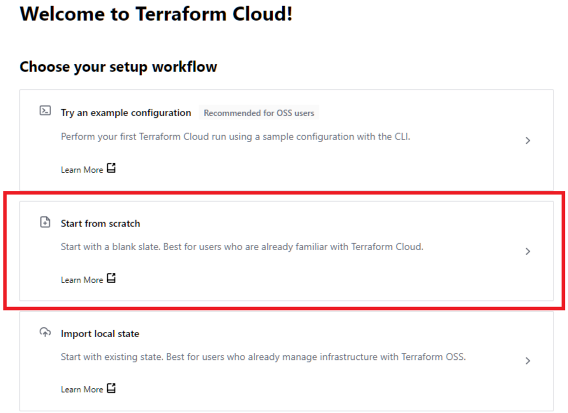
Hello!

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

Need help? Contact us

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

5. A new browser session will open. You can safely close the first session tab. There are 3 setup workflow options. We will begin with **Start from scratch**...



Welcome to Terraform Cloud!

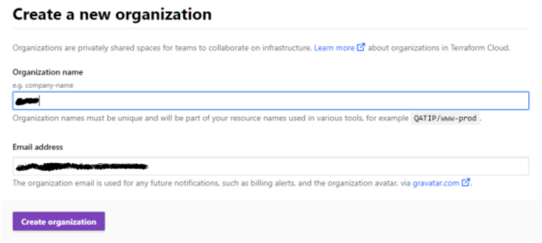
Choose your setup workflow

Try an example configuration Recommended for OSS users
Perform your first Terraform Cloud run using a sample configuration with the CLI.
Learn More

Start from scratch
Start with a blank slate. Best for users who are already familiar with Terraform Cloud.
Learn More

Import local state
Start with existing state. Best for users who already manage infrastructure with Terraform OSS.
Learn More

6. As this is a new account you are required to create a new organization. Chose an organization name and complete the form...



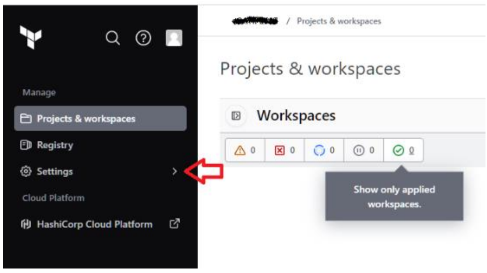
Create a new organization

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

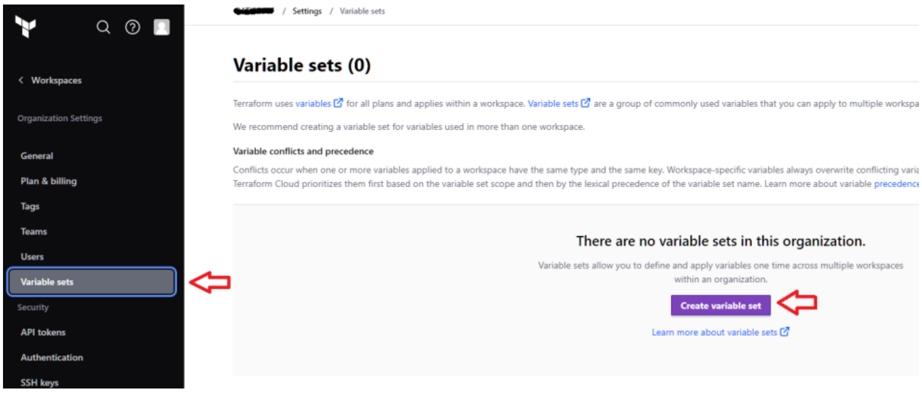
Organization name
e.g. company-name
[redacted]
Organization names must be unique and will be part of your resource names used in various tools, for example `QATIP/www-prod`.

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

7. Before creating our first workspace we must consider the credentials that Terraform Cloud will use as its authority to perform tasks in Google Cloud on our behalf. We will create an environmental variable with our Google Cloud credentials file **keys.txt**. In the Projects and workspaces view, select `Settings`...



8. This will take us to the Organization menu. Select `Variable Sets` and then `Create variable set`...



Variable sets (0)

Terraform uses variables for all plans and applies within a workspace. Variable sets are a group of commonly used variables that you can apply to multiple workspa

We recommend creating a variable set for variables used in more than one workspace.

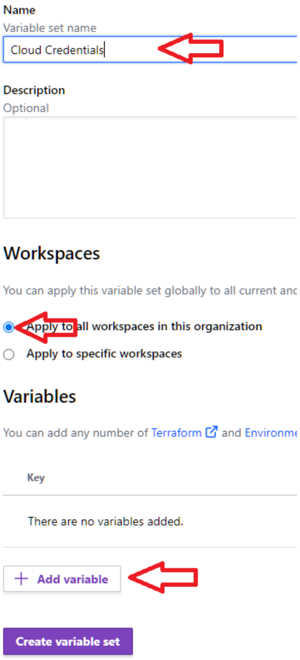
Variable conflicts and precedence

Conflicts occur when one or more variables applied to a workspace have the same type and the same key. Workspace-specific variables always overwrite conflicting varia
Terraform Cloud prioritizes them first based on the variable set scope and then by the lexical precedence of the variable set name. Learn more about variable precedence

There are no variable sets in this organization.

Variable sets allow you to define and apply variables one time across multiple workspaces within an organization.

9. Name the set **Cloud Credentials**, apply the set globally across all workspaces, and click to add your first variable...


Name
Variable set name
Cloud Credentials

Description
Optional

Workspaces

You can apply this variable set globally to all current anc

● Apply to all workspaces in this organization
○ Apply to specific workspaces

Variables

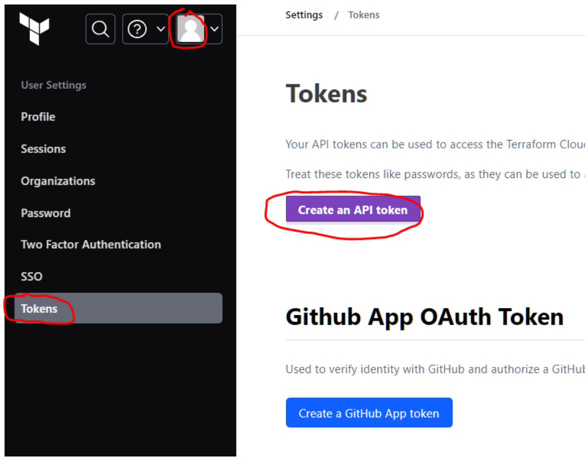
You can add any number of Terraform and Environme

Key
There are no variables added.

10. Select **Environment variable**, and mark as **sensitive** Enter **GOOGLE_CREDENTIALS** as the key. Paste in the contents of **keys.txt** created earlier.

11. We also need to generate an api token within Terraform Cloud that will be used as authentication credentials from the IDE. Navigate to your **User Settings** and select **Tokens**..

12. Select **Create an API token**, use **terraform login** as the description and click on **Generate token**...



Tokens

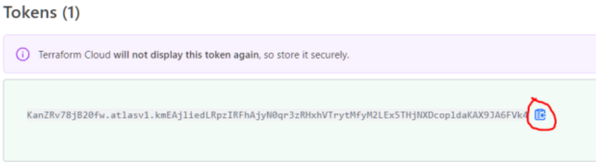
Your API tokens can be used to access the Terraform Clou

Treat these tokens like passwords, as they can be used to .

Github App OAuth Token

Used to verify identity with GitHub and authorize a GitHul

13. A new token is generated...



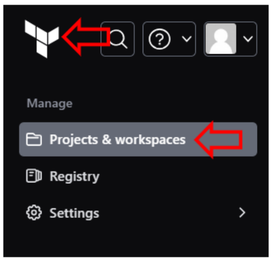
Tokens (1)

Terraform Cloud **will not display this token again**, so store it securely.

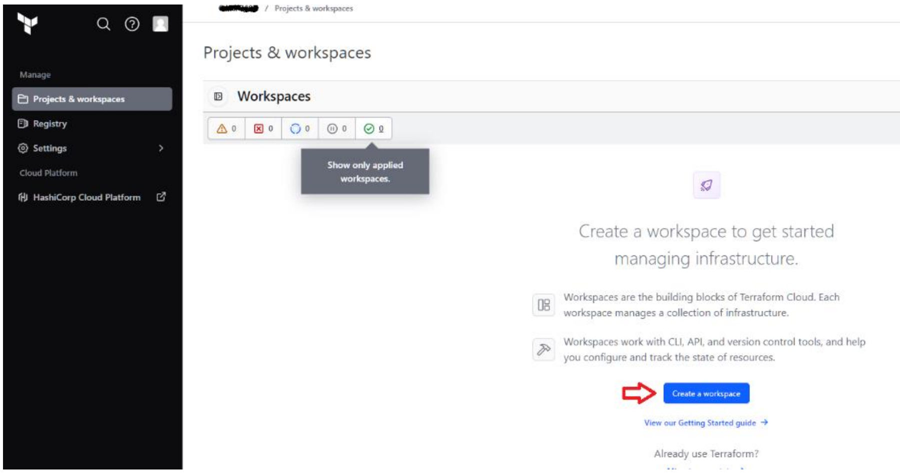
```
KanZRv78jB20fw.atlasv1.kmEAjliedLRpzIRFhAjyN0qr3zRHxhVTrytMfyM2LEx5THjNXDcopldaKAX9JA6FVk4
```

14. Create a local text file and paste the token in for later use.

15. We can now create our first workspace. Select `Workspaces`...



16. Select **Create a workspace**...



[redacted] / Projects & workspaces

Projects & workspaces

Workspaces

⚠ 0 ☒ 0 ○ 0 ⏸ 0 ✓ 0

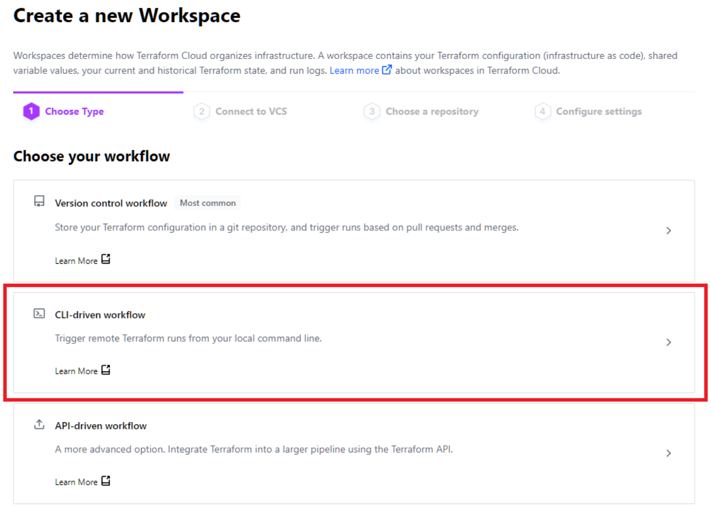
Create a workspace to get started managing infrastructure.

Workspaces are the building blocks of Terraform Cloud. Each workspace manages a collection of infrastructure.

Workspaces work with CLI, API, and version control tools, and help you configure and track the state of resources.

Already use Terraform?

17. Our workspace will be used to generate **CLI-driven workflow**...



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.

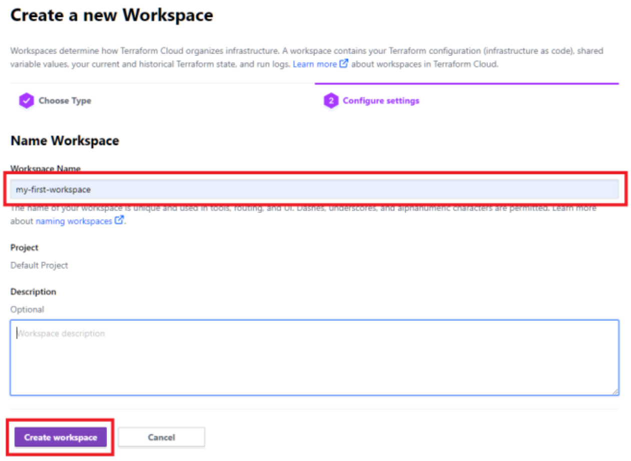
Choose your workflow

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

18. Name the new workspace **my-first-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.

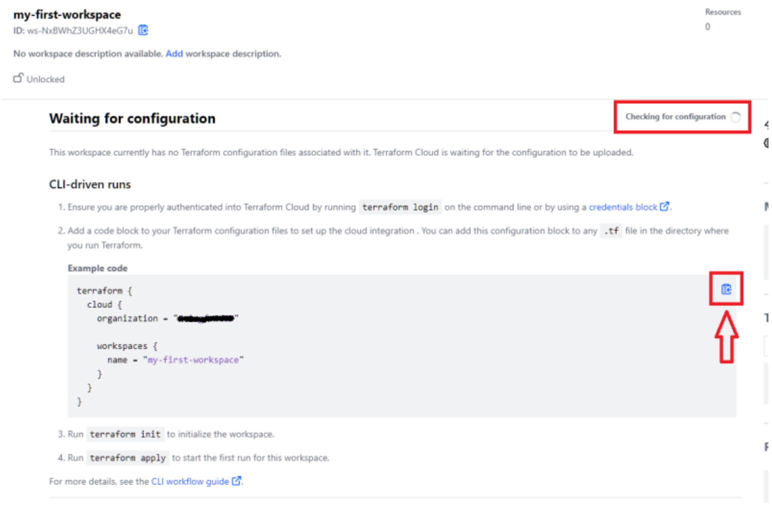
Name Workspace

Workspace Name
my-first-workspace
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

Description
Optional
Workspace description

19. The workspace now exists but has no configuration files associated with it. To associate our local terraform files with Terraform Cloud we need to add the command block shown on-screen, pre-populated with your organization and workspace name, to our configuration...



my-first-workspace

ID: ws-NxBWhZ3UGHX4eG7u

Resources
0

No workspace description available. Add workspace description.

Unlocked

Waiting for configuration

Checking for configuration

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

CLI-driven runs

1. Ensure you are properly authenticated into Terraform Cloud 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.

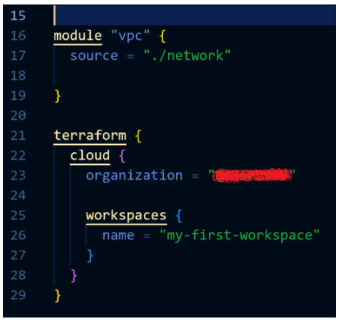
Example code

```
terraform {
  cloud {
    organization = "[redacted]"

    workspaces {
      name = "my-first-workspace"
    }
  }
}
```

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

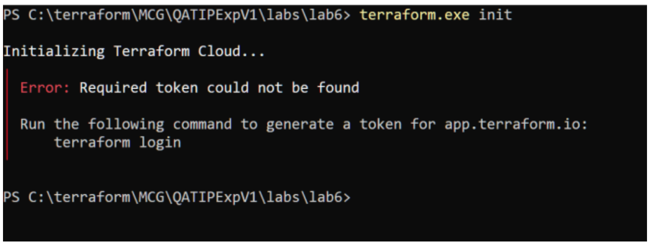
For more details, see the CLI workflow guide.

20. Copy the block, paste it into **main.tf** and save the changes..



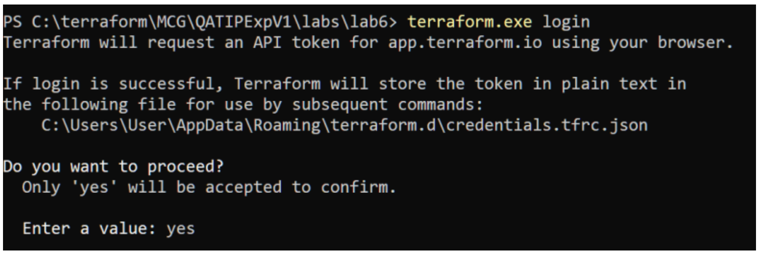
```

module "vpc" {
  source = "./network"

}

terraform {
  cloud {
    organization = "[redacted]"

    workspaces {
      name = "my-first-workspace"
    }
  }
}
```

21. Run **terraform init** ....



```
PS C:\terraform\MCG\QATIPExpV1\labs\lab6> terraform.exe init

Initializing Terraform Cloud...

  Error: Required token could not be found

  Run the following command to generate a token for app.terraform.io:
      terraform login


PS C:\terraform\MCG\QATIPExpV1\labs\lab6>
```

22. You need credentials on the IDE to identify yourself to Terraform Cloud. This is a token generated during the terraform login process. Run `terraform login` and enter `yes`...

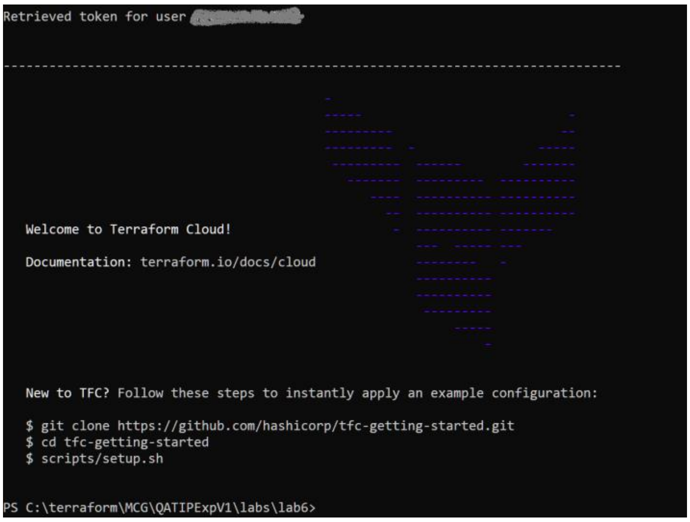


```
PS C:\terraform\MCG\QATIPExpV1\labs\lab6> terraform.exe login
Terraform will request an API token for app.terraform.io using your browser.

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

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

  Enter a value: yes
```

23. The cloud shell instances will fail to create a link to the Terraform Cloud website. Enter **q** then **y*** to return to the prompt awaiting a token.

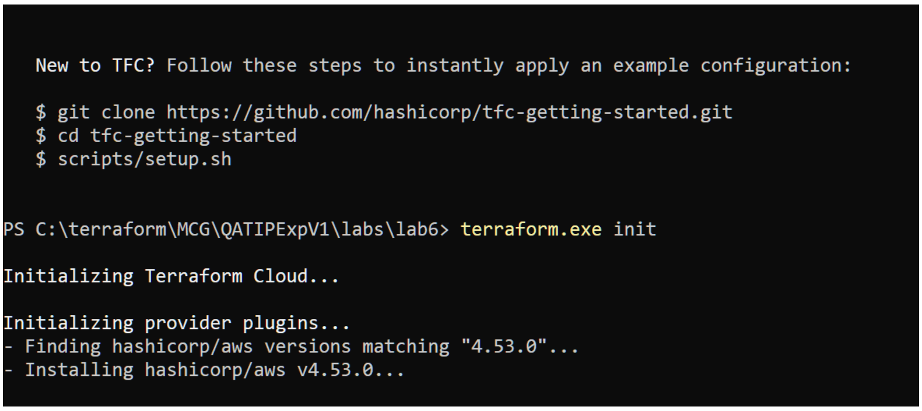
24. Copy the API key from your local text file into your clipboard then click into the terminal to gain focus and paste using **Ctrl+v**

25. If successful, the **Welcome to Terraform Cloud** banner will show...



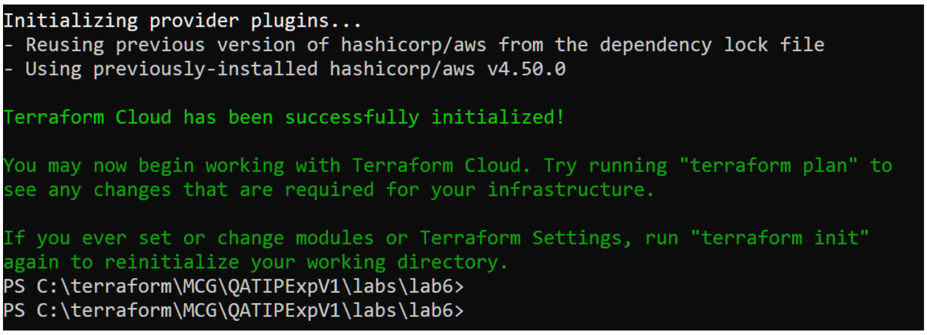
```
Retrieved token for user [redacted]


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

                                          -
                                          -----                             -
                                          ---------                        --
                                          ---------   -                 -----
                                           ---------   ------         -------
                                             -------   ---------   ----------
                                                ----   ---------- ----------
                                                  --   ---------- ----------
   Welcome to Terraform Cloud!                     -   ---------- -------
                                                       ---  ----- ---
   Documentation: terraform.io/docs/cloud              --------   -
                                                       ----------
                                                       ----------
                                                        ---------
                                                            -----
                                                                -


   New to TFC? 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


PS C:\terraform\MCG\QATIPExpV1\labs\lab6>
```

26. Run `terraform init`..



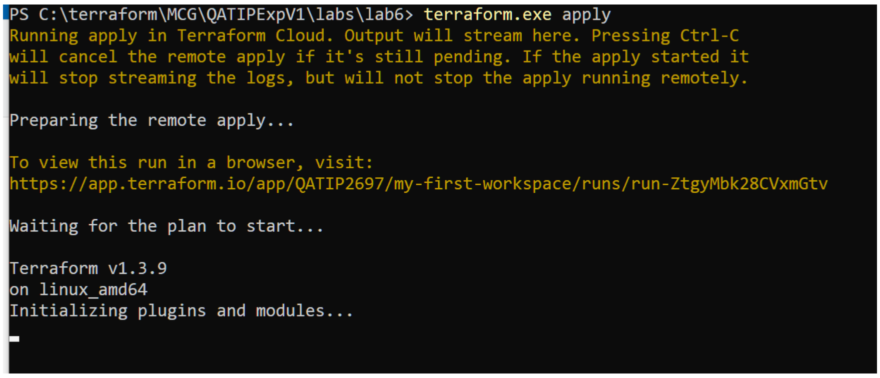
```

   New to TFC? 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


PS C:\terraform\MCG\QATIPExpV1\labs\lab6> terraform.exe init

Initializing Terraform Cloud...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "4.53.0"...
- Installing hashicorp/aws v4.53.0...
```

27. You will be notified that the Terraform Cloud initialization completes successfully...



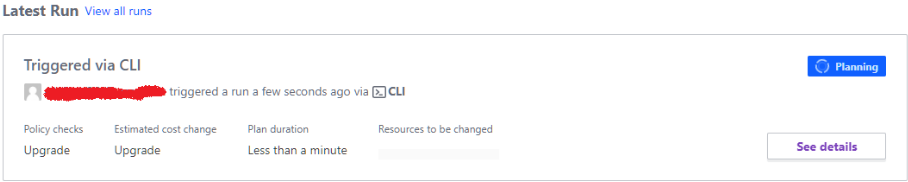
```
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v4.50.0

Terraform Cloud has been successfully initialized!

You may now begin working with Terraform Cloud. 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.
PS C:\terraform\MCG\QATIPExpV1\labs\lab6>
PS C:\terraform\MCG\QATIPExpV1\labs\lab6>
```

28. Run **terraform apply**...



```
PS C:\terraform\MCG\QATIPExpV1\labs\lab6> terraform.exe apply
Running apply in Terraform Cloud. Output will stream here. Pressing Ctrl-C
will cancel the remote apply if it's still pending. If the apply started it
will stop streaming the logs, but will not stop the apply running remotely.

Preparing the remote apply...

To view this run in a browser, visit:
https://app.terraform.io/app/QATIP2697/my-first-workspace/runs/run-ZtgyMbk28CVxmGtv

Waiting for the plan to start...

Terraform v1.3.9
on linux_amd64
Initializing plugins and modules...
```

29. Switch to TFC page. Planning should begin...



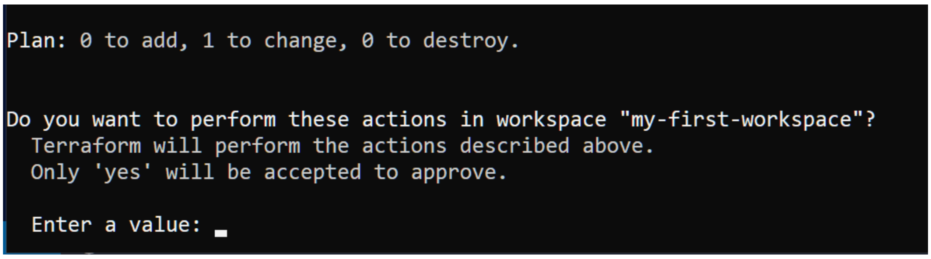
Latest Run View all runs

Triggered via CLI

[redacted] triggered a run a few seconds ago via CLI

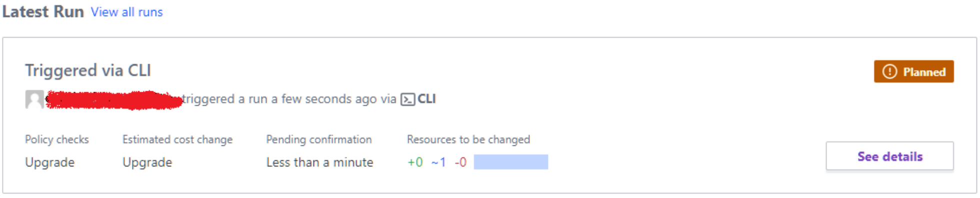
Policy checks	Estimated cost change	Plan duration	Resources to be changed
Upgrade	Upgrade	Less than a minute	

30. Switch back to IDE and wait for prompt to appear. **DO NOT TYPE yes**' yet...



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


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

  Enter a value: 
```

31. Switch to TFC page. Planning is now complete...



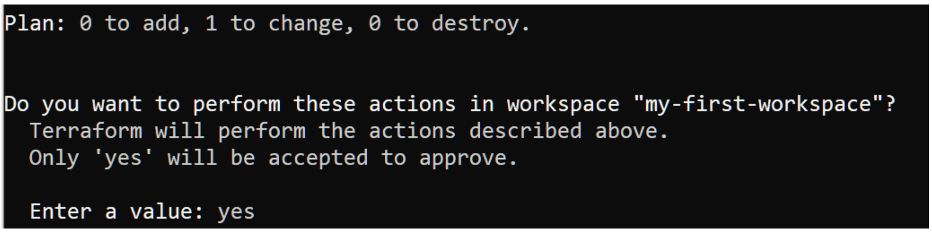
Latest Run View all runs

Triggered via CLI

[redacted] triggered a run a few seconds ago via CLI

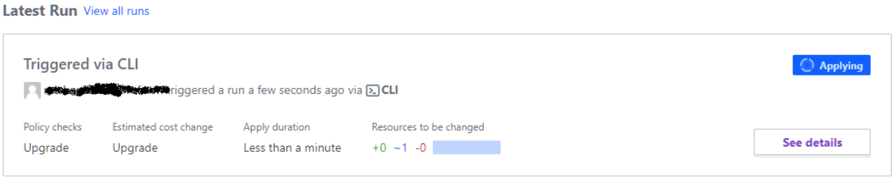
Policy checks	Estimated cost change	Pending confirmation	Resources to be changed
Upgrade	Upgrade	Less than a minute	+0 ~1 -0

32. Switch back to IDE and enter `yes`..



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


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

  Enter a value: yes
```

33. Switch back to TFC page. The plan is being applied..



Latest Run View all runs

Triggered via CLI

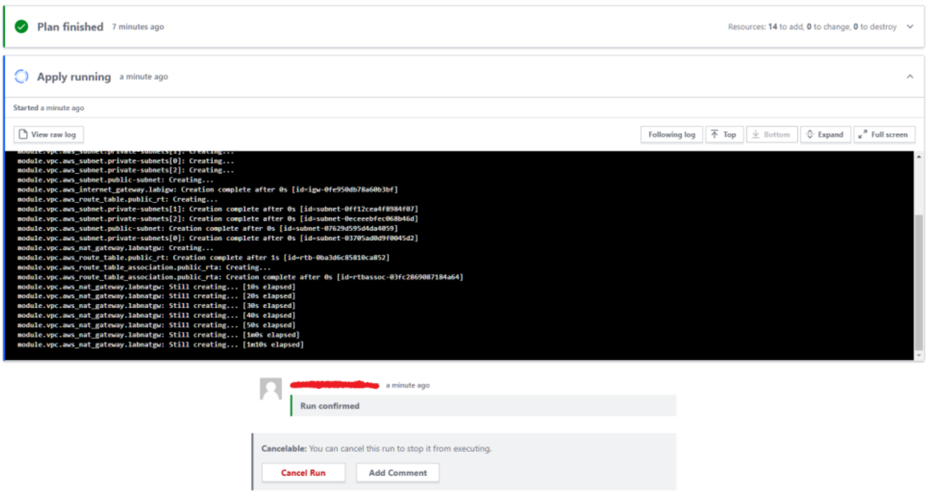
[redacted]riggered a run a few seconds ago via CLI

Applying

Policy checks	Estimated cost change	Apply duration	Resources to be changed
Upgrade	Upgrade	Less than a minute	+0 ~1 -0

See details

34. Click on `See details` to watch deployment progressing...



Plan finished 7 minutes ago

Resources: 14 to add, 0 to change, 0 to destroy

Apply running a minute ago

Started a minute ago

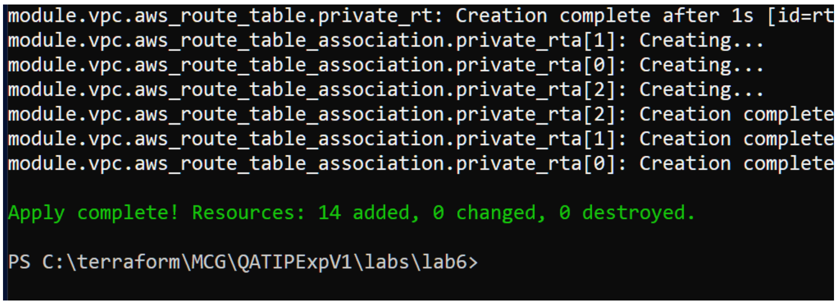
```
module.vpc.aws_subnet.private-subnets[1]: Creating...
module.vpc.aws_subnet.private-subnets[0]: Creating...
module.vpc.aws_subnet.private-subnets[2]: Creating...
module.vpc.aws_subnet.public-subnet: Creating...
module.vpc.aws_internet_gateway.labigw: Creation complete after 0s [id=igw-0fe950db78a60b3bf]
module.vpc.aws_route_table.public_rt: Creating...
module.vpc.aws_subnet.private-subnets[1]: Creation complete after 0s [id=subnet-0ff12cea4f8984f07]
module.vpc.aws_subnet.private-subnets[2]: Creation complete after 0s [id=subnet-0eceeebfec068b46d]
module.vpc.aws_subnet.public-subnet: Creation complete after 0s [id=subnet-07629d595d4da4059]
module.vpc.aws_subnet.private-subnets[0]: Creation complete after 0s [id=subnet-03705ad0d9f0045d2]
module.vpc.aws_nat_gateway.labnatgw: Creating...
module.vpc.aws_route_table.public_rt: Creation complete after 1s [id=rtb-0ba3d6c85810ca852]
module.vpc.aws_route_table_association.public_rta: Creating...
module.vpc.aws_route_table_association.public_rta: Creation complete after 0s [id=rtbassoc-03fc2869087184a64]
module.vpc.aws_nat_gateway.labnatgw: Still creating... [10s elapsed]
module.vpc.aws_nat_gateway.labnatgw: Still creating... [20s elapsed]
module.vpc.aws_nat_gateway.labnatgw: Still creating... [30s elapsed]
module.vpc.aws_nat_gateway.labnatgw: Still creating... [40s elapsed]
module.vpc.aws_nat_gateway.labnatgw: Still creating... [50s elapsed]
module.vpc.aws_nat_gateway.labnatgw: Still creating... [1m0s elapsed]
module.vpc.aws_nat_gateway.labnatgw: Still creating... [1m10s elapsed]
```

[redacted] a minute ago

Run confirmed

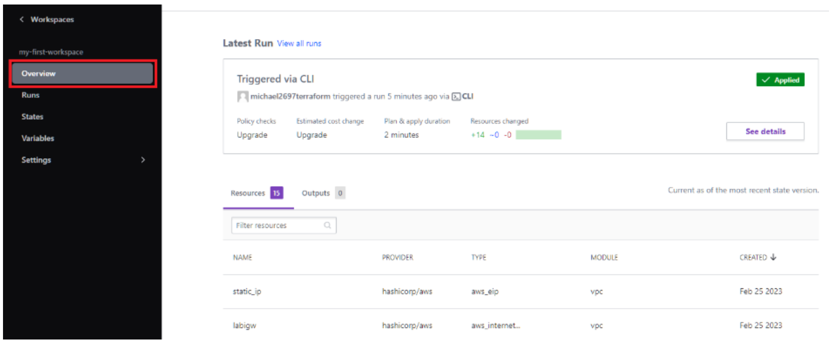
Cancelable: You can cancel this run to stop it from executing.

35. Switch back to IDE and wait for deployment to complete...



```
module.vpc.aws_route_table.private_rt: Creation complete after 1s [id=rt
module.vpc.aws_route_table_association.private_rta[1]: Creating...
module.vpc.aws_route_table_association.private_rta[0]: Creating...
module.vpc.aws_route_table_association.private_rta[2]: Creating...
module.vpc.aws_route_table_association.private_rta[2]: Creation complete
module.vpc.aws_route_table_association.private_rta[1]: Creation complete
module.vpc.aws_route_table_association.private_rta[0]: Creation complete

Apply complete! Resources: 14 added, 0 changed, 0 destroyed.

PS C:\terraform\MCG\QATIPExpV1\labs\lab6>
```

36. Switch back to TFC. Select `Overview` to see a summary of the applied job and the resources created...



Latest Run View all runs

Triggered via CLI

michael2697terraform triggered a run 5 minutes ago via CLI

Applied

Policy checks	Estimated cost change	Plan & apply duration	Resources changed
Upgrade	Upgrade	2 minutes	+14 ~0 -0

See details

Resources 15 Outputs 0

Current as of the most recent state version.

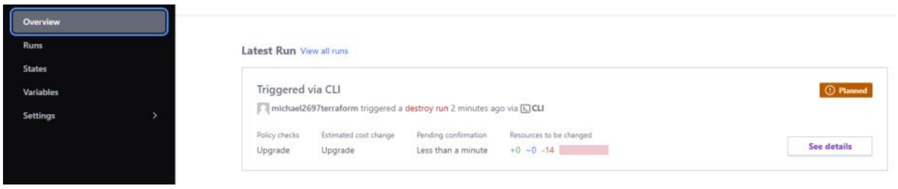
Filter resources

NAME	PROVIDER	TYPE	MODULE	CREATED ↓
static_ip	hashicorp/aws	aws_eip	vpc	Feb 25 2023
labigw	hashicorp/aws	aws_internet...	vpc	Feb 25 2023

37. In the IDE enter `terraform destroy` Do not confirm yet.

38. Switch to TFC and note the planning of the destroy is in progress.

39. Wait until the destroy run shows as `Planned` Click on `See details`

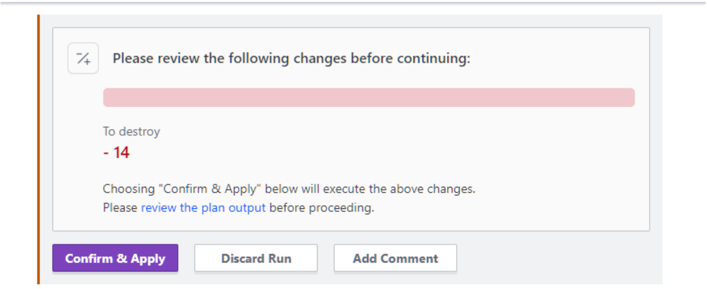


Triggered via CLI

michael2697terraform triggered a destroy run 2 minutes ago via CLI

Policy checks	Estimated cost change	Pending confirmation	Resources to be changed
Upgrade	Upgrade	Less than a minute	+0 ~0 -14

40. Scroll down and click on `Confirm and Apply`

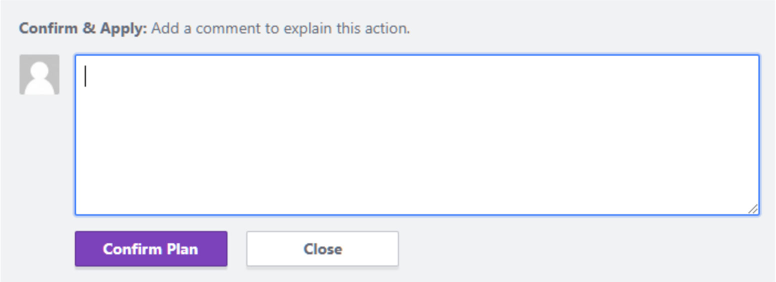


Please review the following changes before continuing:

To destroy
- 14

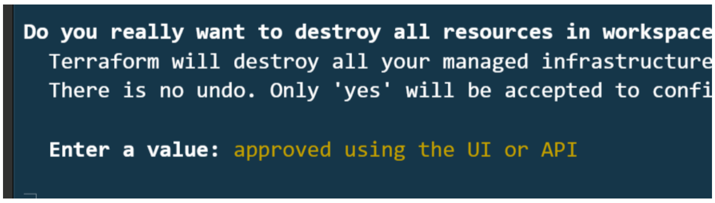
Choosing "Confirm & Apply" below will execute the above changes.
Please review the plan output before proceeding.

41. Click on `Confirm Plan`



Confirm & Apply: Add a comment to explain this action.

42. Switch back to IDE and, if timely, you will see a message indicating the apply has been approved on TFC. The destroy will then progress.



```
Do you really want to destroy all resources in workspace
  Terraform will destroy all your managed infrastructure
  There is no undo. Only 'yes' will be accepted to confi

  Enter a value: approved using the UI or API
```

Congratulations, you have completed this lab