

Lesson 10 Demo 04

Configuring Terraform Remote State Backend

Objective: To configure and manage Terraform state using the remote enhanced backend with Terraform Cloud for ensuring efficient and collaborative state management

Tools required: Visual Studio Code

Prerequisites: Terraform Cloud account

Ensure you have created and implemented the AWS access key and secret key before starting this demo. Refer to Lesson 08 Assisted Practice 02 for detailed steps.

Note: The folder structure created in the previous demos is used here. It is also included in the resources section of LMS. Please refer to Lesson 10 Demo 01.

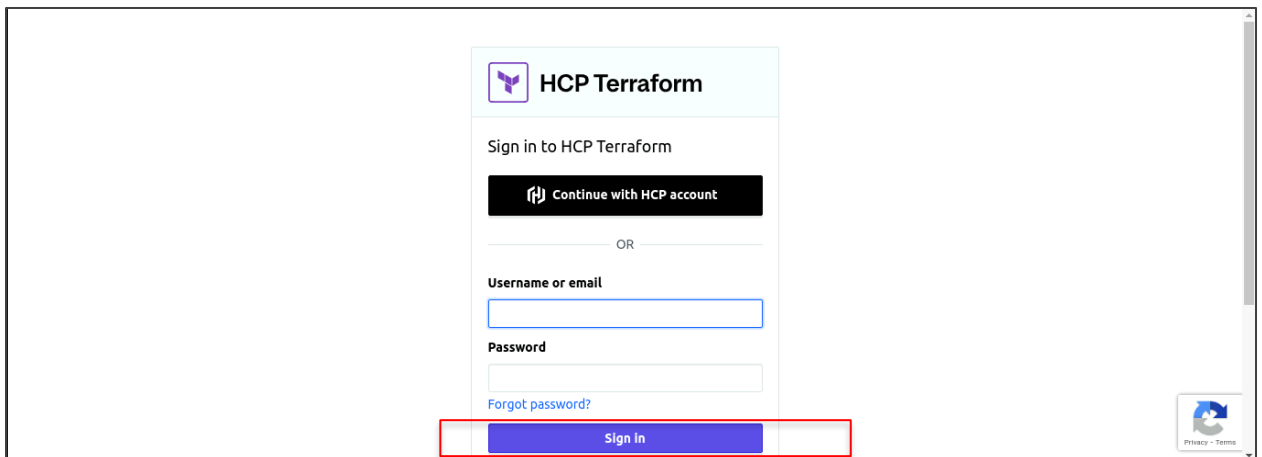
Steps to be followed:

1. Sign in to Terraform Cloud platform
2. Update the Terraform configuration to use remote enhanced backend
3. Re-initialize Terraform and validate the remote backend with Terraform Cloud
4. Provide secure credentials for remote runs
5. Remove existing resources with the terraform destroy command

Step 1: Sign in to Terraform Cloud platform

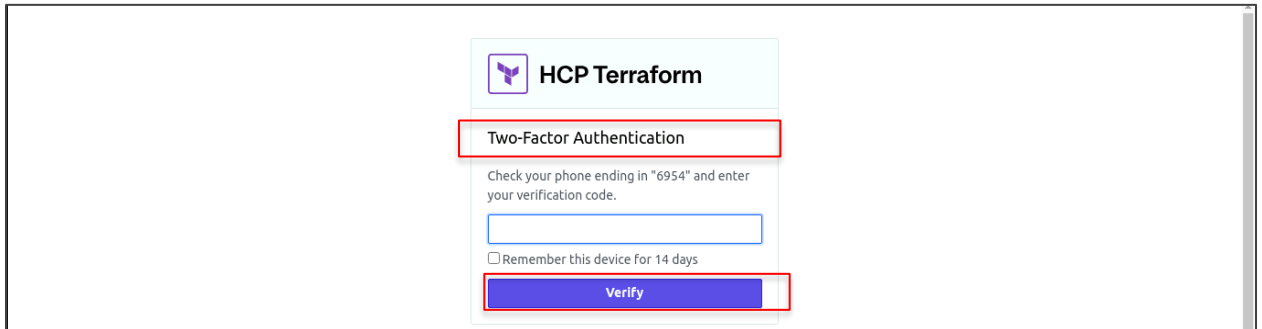
1.1 Enter the required details and click on **Sign In** by using the following URL:

<https://app.terraform.io/session>



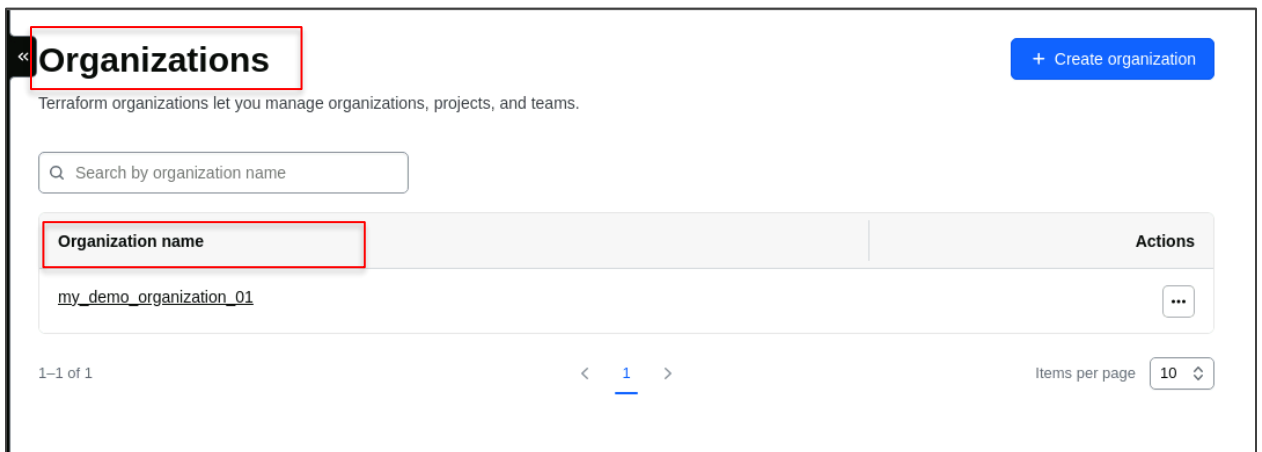
The screenshot displays the HCP Terraform sign-in interface. At the top, the HCP Terraform logo is visible. Below it, the text "Sign in to HCP Terraform" is shown. There is a button labeled "Continue with HCP account" with a key icon. Below this, the word "OR" is centered. The sign-in form includes two input fields: "Username or email" and "Password". A link for "Forgot password?" is located below the password field. At the bottom of the form, the "Sign In" button is highlighted with a red rectangular box. In the bottom right corner, there is a small icon for "Privacy - Terms".

1.2 Verify the **Two-Factor Authentication**



The screenshot shows the 'HCP Terraform' interface for Two-Factor Authentication. A red box highlights the 'Two-Factor Authentication' header. Below it, instructions state: 'Check your phone ending in "6954" and enter your verification code.' A text input field is provided for the code. A checkbox labeled 'Remember this device for 14 days' is present. A red box highlights the 'Verify' button at the bottom.

1.3 The **Organizations** menu will automatically open. Take note of the **Organization name**.




The screenshot displays the 'Organizations' page in the Terraform Cloud interface. A red box highlights the 'Organizations' header. A blue button '+ Create organization' is in the top right. Below the header, a search bar is labeled 'Search by organization name'. A table lists organizations with a red box highlighting the 'Organization name' column header. The table contains one entry: 'my_demo_organization_01'. The 'Actions' column has a three-dot menu icon. At the bottom, pagination shows '1-1 of 1' and 'Items per page 10'.

Step 2: Update the Terraform configuration to use remote enhanced backend

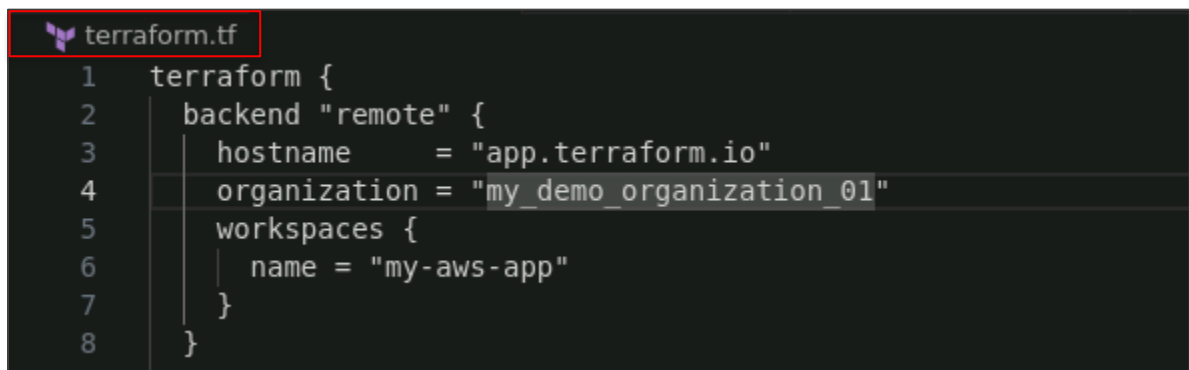
2.1 Edit your **terraform.tf** file to include the remote backend configuration by using the following code:

```
terraform {  
  backend "remote" {  
    hostname = "app.terraform.io"  
    # Replace " my_demo_organization_01" with your actual Terraform Cloud  
    organization name  
    organization = "my_demo_organization_01"  
    workspaces {  
      name = "my-aws-app"    }  
  }  
}
```

```
}  
}  
}
```



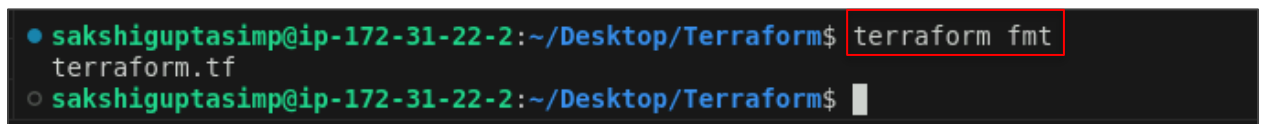
```
main.tf  terraform.tf x  variables.tf  terraform.tfstate  
terraform.tf  
1 terraform {  
2   backend "s3" {  
3     bucket = "myterraformstatedemo"  
4     key     = "prod/aws_infra"  
5     region = "us-east-1"  
6   }  
7   # Replace this with your DynamoDB table name!  
8   dynamodb_table = "terraform-locks"  
9   encrypt         = true  
10 }  
11  
12
```



```
terraform.tf  
1 terraform {  
2   backend "remote" {  
3     hostname     = "app.terraform.io"  
4     organization = "my_demo_organization_01"  
5     workspaces {  
6       name = "my-aws-app"  
7     }  
8   }  
9 }
```

2.2 Format the Terraform configuration file by using the following command:

terraform fmt



```
sakshiguptasimp@ip-172-31-22-2:~/Desktop/Terraform$ terraform fmt  
terraform.tf  
sakshiguptasimp@ip-172-31-22-2:~/Desktop/Terraform$
```

Step 3: Re-initialize Terraform and validate the remote backend with Terraform Cloud

3.1 Re-initialize the backend using the following command:

terraform init -reconfigure

```
• sakshiguptasimp@ip-172-31-22-2:~/Desktop/Terraform$ terraform init -reconfigure
Initializing modules...

Initializing the backend...

Successfully configured the backend "remote"! Terraform will automatically
use this backend unless the backend configuration changes.

Initializing provider plugins...
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  PORTS

- Reusing previous version of hashicorp/tls from the dependency lock file
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/http from the dependency lock file
- Using previously-installed hashicorp/random v3.1.0
- Using previously-installed hashicorp/local v2.1.0
- Using previously-installed hashicorp/tls v3.1.0
- Using previously-installed hashicorp/aws v3.76.1
- Using previously-installed hashicorp/http v2.1.0

Terraform has been successfully initialized!

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

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
• sakshiguptasimp@ip-172-31-22-2:~/Desktop/Terraform$
```

3.2 Apply the configuration to the remote backend using the following command:

terraform apply

```
• sakshiguptasimp@ip-172-31-22-2:~/Desktop/Terraform$ terraform apply
Running apply in the remote backend. 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/my_demo_organization_01/my-aws-app/runs/run-jGiC1V41yFLKvKxW

Waiting for the plan to start...

Terraform v1.1.6
on linux_amd64
Initializing plugins and modules...
```

3.3 When prompted, approve the changes by typing **yes**

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  PORTS

+ volume_size           = (known after apply)
+ volume_type           = (known after apply)
}
}

Plan: 30 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ public_dns             = (known after apply)
+ public_dns_server_subnet_1 = (known after apply)
+ public_ip              = (known after apply)
+ public_ip_server_subnet_1 = (known after apply)
+ size                   = "t2.micro"

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

Enter a value: yes
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  PORTS  terraform + ~

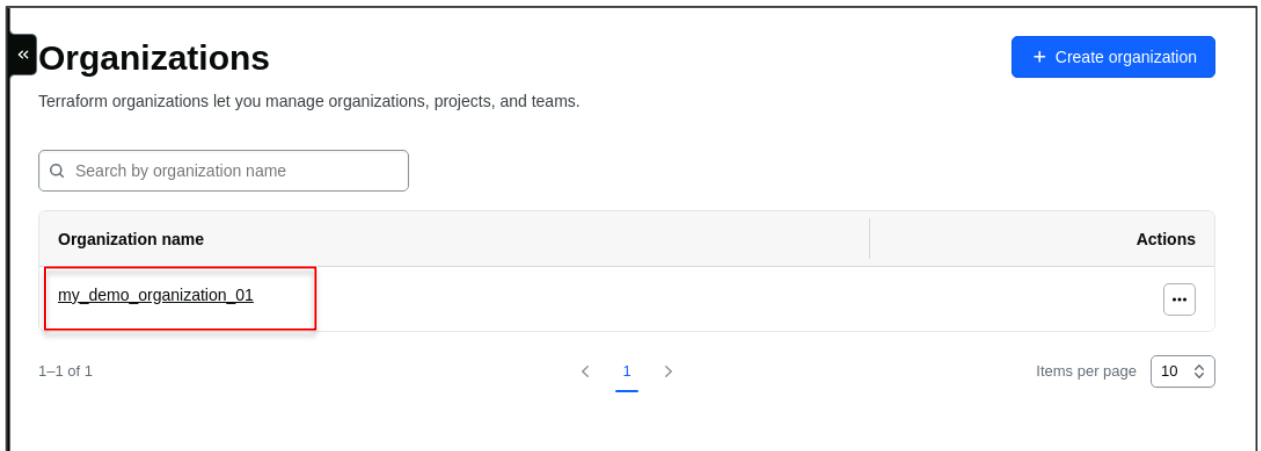
aws_subnet.private_subnets["private_subnet_1"]: Creating...
aws_subnet.private_subnets["private_subnet_2"]: Creating...
aws_subnet.public_subnets["public_subnet_1"]: Creating...
aws_subnet.public_subnets["public_subnet_3"]: Creating...
aws_security_group.ingress-ssh: Creating...
aws_security_group.vpc-ping: Creating...
aws_security_group.vpc-web: Creating...
aws_internet_gateway.internet_gateway: Creation complete after 0s [id=igw-0935512912e66c367]
aws_eip.nat_gateway_eip: Creating...
aws_subnet.private_subnets["private_subnet_3"]: Creation complete after 0s [id=subnet-0285e954d5276877b]
aws_route_table.public_route_table: Creating...
aws_subnet.private_subnets["private_subnet_2"]: Creation complete after 0s [id=subnet-0c9cce476fd244ce5]
aws_subnet.private_subnets["private_subnet_1"]: Creation complete after 0s [id=subnet-01567f09b7d17aeb7]
aws_eip.nat_gateway_eip: Creation complete after 1s [id=eipalloc-000a281852ff09f80]
aws_route_table.public_route_table: Creation complete after 1s [id=rtb-0c28c5ef816667ff7]
aws_security_group.vpc-ping: Creation complete after 2s [id=sg-04d7b6e8a8df09643]
aws_security_group.vpc-web: Creation complete after 2s [id=sg-0ba47019920676600]
aws_security_group.ingress-ssh: Creation complete after 2s [id=sg-0b8dc2374b6814fea]
```

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS
aws_route_table.private_route_table: Creating...
aws_route_table.private_route_table: Creation complete after 1s [id=rtb-06adcf5d46a0a60de]
aws_route_table_association.private["private_subnet_1"]: Creating...
aws_route_table_association.private["private_subnet_2"]: Creating...
aws_route_table_association.private["private_subnet_3"]: Creating...
aws_route_table_association.private["private_subnet_2"]: Creation complete after 0s [id=rtbassoc-01989bdfb41e77f2c]
aws_route_table_association.private["private_subnet_3"]: Creation complete after 1s [id=rtbassoc-0f9b7f543dfcfa244]
aws_route_table_association.private["private_subnet_1"]: Creation complete after 1s [id=rtbassoc-00d28a447954f964f]

Apply complete! Resources: 30 added, 0 changed, 0 destroyed.

Outputs:
public_dns = "ec2-44-220-82-222.compute-1.amazonaws.com"
public_dns_server_subnet_1 = "ec2-3-95-172-42.compute-1.amazonaws.com"
public_ip = "44.220.82.222"
public_ip_server_subnet_1 = "3.95.172.42"
size = "t2.micro"
sakshiguptasimp@ip-172-31-22-2:~/Desktop/Terraform$
```

3.4 Go to Terraform Cloud to validate the remote backend. Click on the organization in which you are working.



3.5 Click on the workspace **my-aws-app**

« my_demo_organization_01 / Workspaces

Workspaces

New ▾

Workspaces

0 0 0 0 1

Tags ▾ Status ▾ Clear all

Workspace Name ▾	Run Status	Repo	Latest Change ↕
my-aws-app Default Project	✓ Applied		a minute ago

1-1 of 1

< 1 >

« my_demo_organization_01 / Workspaces / my-aws-app / Overview

my-aws-app

ID: ws-EvHFQHFt5bwxTnbw

[Add workspace description.](#)

Unlocked Resources 33 Terraform v1.1.6 Updated a minute ago

Lock + New run

Latest Run

[View all runs](#)

Triggered via CLI

sakshiguptasimplilearn triggered a run 5 minutes ago via CLI

✓ Applied

⚡ Execution mode: Remote

⚙️ Auto-apply API, CLI, & VCS runs: Off

⚙️ Auto-apply run triggers: Off

Project: Default Project

3.6 Scroll down to the **Latest Run** section and click on **See details**

<<

Triggered via CLI

CURRENT

Applied

Plan & apply duration

3 minutes

Resources changed

+30 -0 -0

sakshiguptasimplelearn triggered a run from CLI 6 minutes ago

Run Details

Plan finished 6 minutes ago

Resources: 30 to add, 0 to change, 0 to destroy

Apply finished 2 minutes ago

Resources: 30 added, 0 changed, 0 destroyed

<<

Plan finished 6 minutes ago

Resources: 30 to add, 0 to change, 0 to destroy

Started 6 minutes ago > Finished 6 minutes ago

View raw log

Top

Bottom

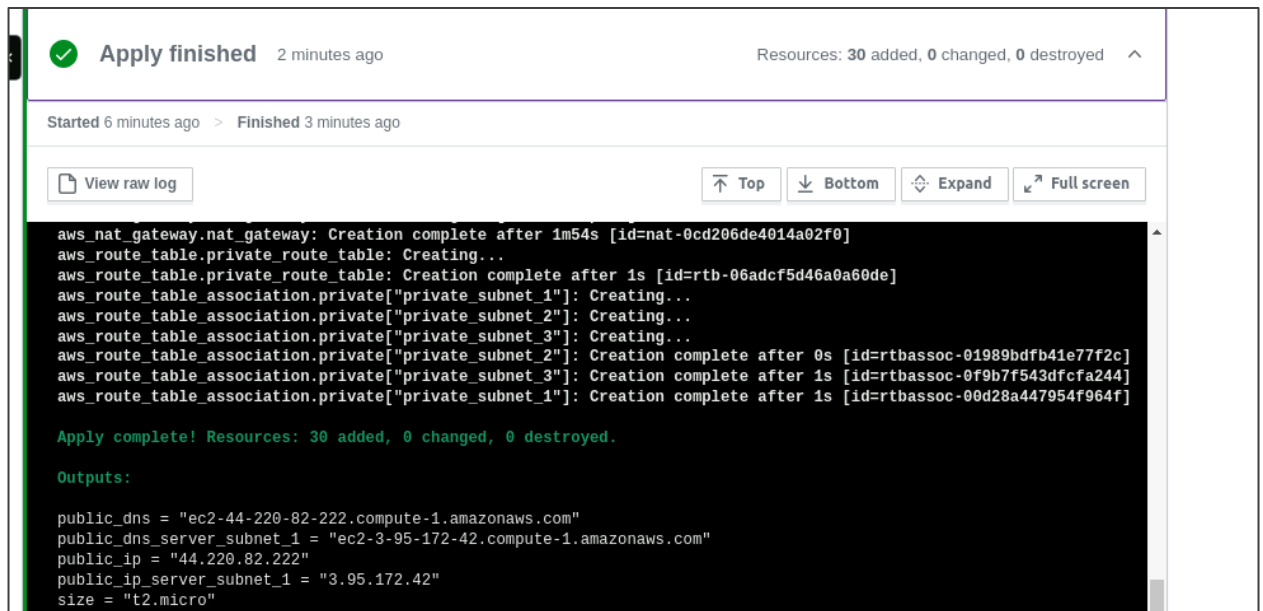
Expand

Full screen

```
+ device_name      = (known after apply)
+ encrypted        = (known after apply)
+ iops              = (known after apply)
+ kms_key_id       = (known after apply)
+ tags             = (known after apply)
+ throughput       = (known after apply)
+ volume_id        = (known after apply)
+ volume_size      = (known after apply)
+ volume_type      = (known after apply)
}

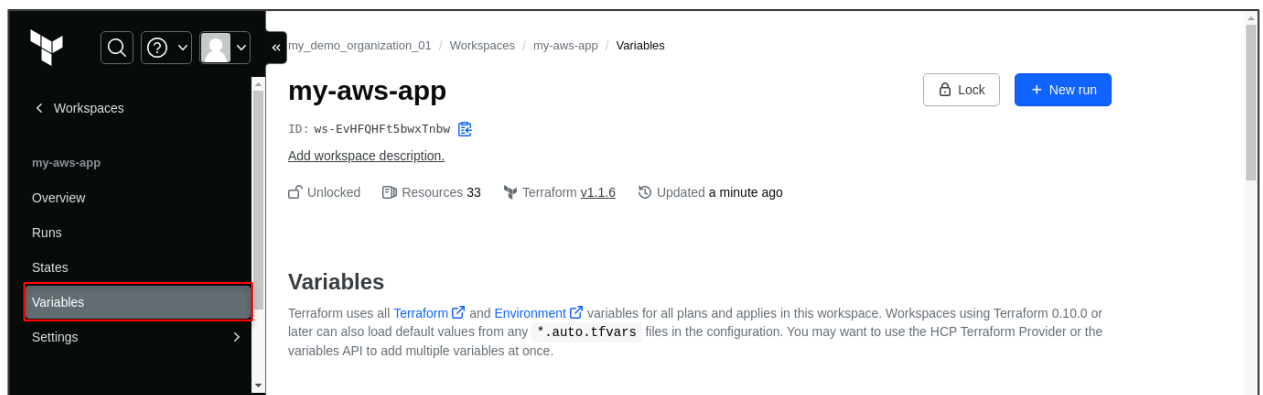
Plan: 30 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ public_dns                = (known after apply)
+ public_dns_server_subnet_1 = (known after apply)
+ public_ip                 = (known after apply)
+ public_ip_server_subnet_1 = (known after apply)
```



Step 4: Provide secure credentials for remote runs

4.1 Go to the **Variables** section of your workspace



4.2 Scroll down to the **Workspace variables** and click on **Add variable**

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 (0)

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
There are no variables added.		

+ Add variable

4.3 Select variable category as **Environment variable**

There are no variables added.

<<

Select variable category

☐ **Terraform variable**
These variables should match the declarations in your configuration. Click the HCL box to use interpolation or set a non-string value.

☒ **Environment variable**
These variables are available in the Terraform runtime environment.

Key

key

Value

value

☐ **Sensitive** ⓘ

Description (Optional)

description (optional)

Add variable

Cancel

4.4 Add the details for your access key and secret key as shown in the screenshots. After this, click on **Save variable**.

Key

Value

Category

Select variable category

☐ Terraform variable

☒ Environment variable

These variables should match the declarations in your configuration. Click the HCL box to use interpolation or set a non-string value.

These variables are available in the Terraform runtime environment.

Key

Value

☐ Sensitive ⓘ

AWS_ACCESS_KEY_ID

AKIA5FR7SGZJAHDS6GFD

Description (Optional)

description (optional)

Save variable

Cancel

Key

Value

Category

Select variable category

☐ Terraform variable

☒ Environment variable

These variables should match the declarations in your configuration. Click the HCL box to use interpolation or set a non-string value.

These variables are available in the Terraform runtime environment.

Key

Value

☐ Sensitive ⓘ

AWS_SECRET_ACCESS_KEY

K6LaS+Zz1apOQwuxVDW7pnsECG3cJB95ZvSQj5Yj

Description (Optional)

description (optional)

Save variable

Cancel

Workspace variables (2)			
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	
AWS_ACCESS_KEY_ID	AKIA5FR7SGZJAHDS6GFD	env	...
AWS_SECRET_ACCESS_KEY	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	env	...

4.5 Go to the terminal in Visual Studio Code and apply the Terraform configuration with the new credentials using the following command:

terraform apply

```
o sakshiguptasimp@ip-172-31-22-2:~/Desktop/Terraform$ terraform apply
Running apply in the remote backend. 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/my_demo_organization_01/my-aws-app/runs/run-hvpfGauWSyBDnb7M

Waiting for the plan to start...
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS

Terraform will perform the following actions:

```
# local_file.private_key_pem will be created
+ resource "local_file" "private_key_pem" {
  + content          = (sensitive)
  + directory_permission = "0777"
  + file_permission   = "0777"
  + filename         = "MyAWSKey.pem"
  + id               = (known after apply)
}
```

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

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

Enter a value: yes

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS

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

Enter a value: yes

local_file.private_key_pem: Creating...
local_file.private_key_pem: Creation complete after 0s [id=34cdb014049a9e0c76a19283afe30e2ef870b717]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

```
public_dns = "ec2-44-220-82-222.compute-1.amazonaws.com"
public_dns_server_subnet_1 = "ec2-3-95-172-42.compute-1.amazonaws.com"
public_ip = "44.220.82.222"
public_ip_server_subnet_1 = "3.95.172.42"
size = "t2.micro"
```

```
o sakshiguptasimp@ip-172-31-22-2:~/Desktop/Terraform$
```

Step 5: Remove existing resources with the terraform destroy command

5.1 Clean up all existing resources managed by Terraform using the following command:

terraform destroy

```
o sakshiguptasimp@ip-172-31-22-2:~/Desktop/Terraform$ terraform destroy
Running apply in the remote backend. 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/my_demo_organization_01/my-aws-app/runs/run-wii4eSoSK8w7s8f

Waiting for the plan to start...

Terraform v1.1.6
on linux amd64
Initializing plugins and modules...
█
```

5.2 When prompted, approve the destruction by typing **yes**

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  PORTS

- volume_size          = 8 -> null
- volume_type          = "gp2" -> null
}

Plan: 0 to add, 0 to change, 29 to destroy.

Changes to Outputs:
- public_dns            = "ec2-44-220-82-222.compute-1.amazonaws.com" -> null
- public_dns_server_subnet_1 = "ec2-3-95-172-42.compute-1.amazonaws.com" -> null
- public_ip             = "44.220.82.222" -> null
- public_ip_server_subnet_1 = "3.95.172.42" -> null
- size                  = "t2.micro" -> null

Do you really want to destroy all resources in workspace "my-aws-app"?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes█
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  PORTS

aws_subnet.public_subnets["public_subnet_3"]: Destroying... [id=subnet-031b4ccbab76352ca]
aws_subnet.public_subnets["public_subnet_2"]: Destroying... [id=subnet-0d3714b9ed6230472]
aws_subnet.public_subnets["public_subnet_3"]: Destruction complete after 0s
aws_subnet.public_subnets["public_subnet_1"]: Destruction complete after 0s
aws_subnet.public_subnets["public_subnet_2"]: Destruction complete after 0s
aws_eip.nat_gateway_eip: Destruction complete after 1s
aws_internet_gateway.internet_gateway: Destroying... [id=igw-0935512912e66c367]
aws_internet_gateway.internet_gateway: Destruction complete after 0s
aws_vpc.vpc: Destroying... [id=vpc-01c721fb010bc8ef8]
aws_vpc.vpc: Destruction complete after 1s

Apply complete! Resources: 0 added, 0 changed, 29 destroyed.

○ sakshiguptasimp@ip-172-31-22-2:~/Desktop/Terraform$
```

By following these steps, you have successfully configured and managed Terraform state using the remote enhanced backend with Terraform Cloud.