Lesson 10 Lesson-End Project

Managing Terraform State Using Different Backends

Project agenda: To perform Terraform state management using different backends for storing and managing the state file securely and efficiently

Description: You work as a junior DevOps engineer in an IT firm. Your company is undertaking a project that involves migrating the Terraform state between various backends for better state management and collaboration. The project aims to leverage Amazon S3 for state storage and DynamoDB for state locking, followed by a migration to Terraform Cloud for enhanced team collaboration.

Tools required: Visual Studio Code

Prerequisites: Terraform Cloud account
Ensure you have created the AWS access key and secret key before starting this LEP. Refer to Lesson 08 Assisted Practice 02 for detailed steps.

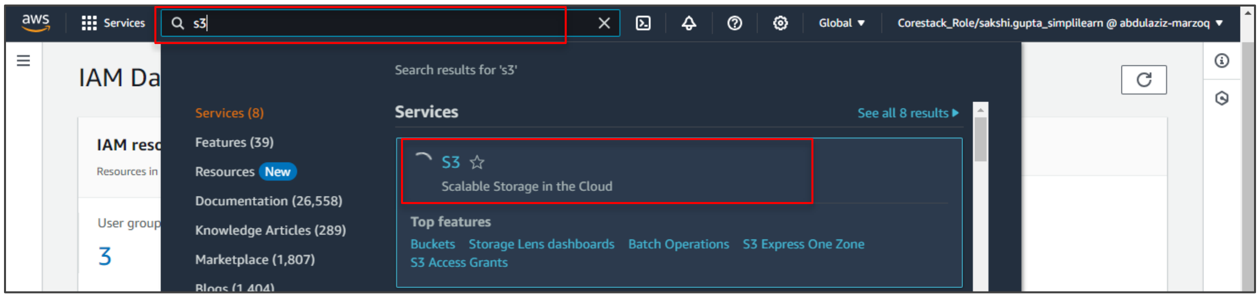
Expected deliverables: An operational Terraform state management mechanism across S3 and Terraform Cloud backends.

Steps to be followed:

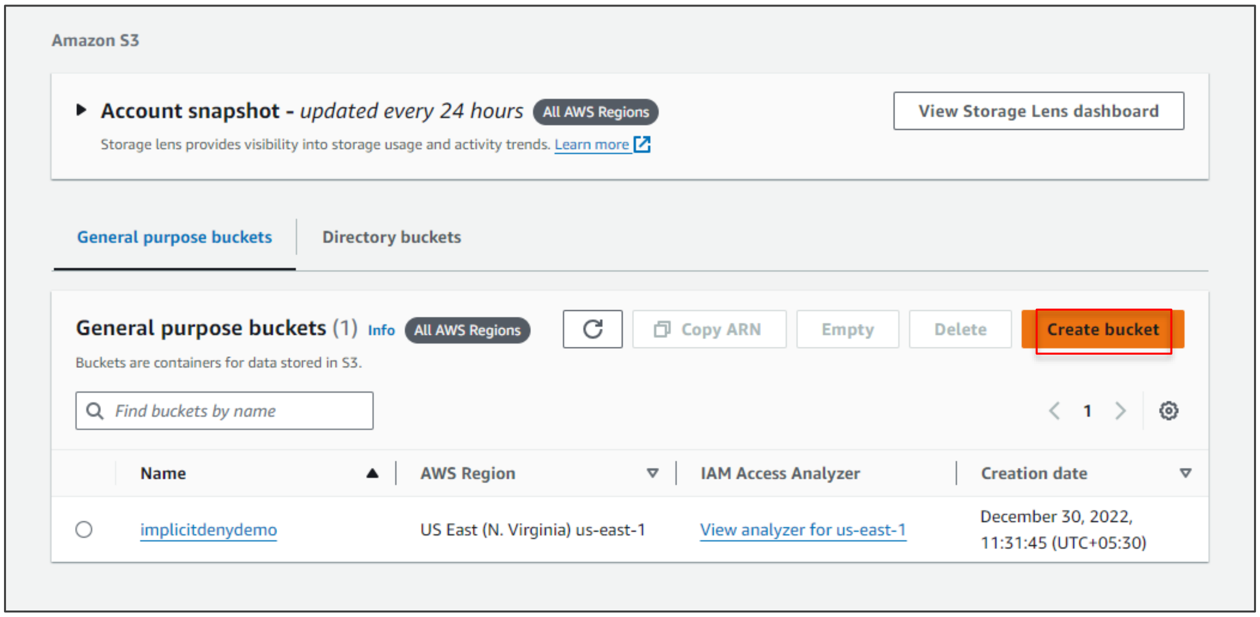
1. Configure S3 backend and DynamoDB
2. Update the Terraform configuration for S3 backend
3. Migrate state to remote backend with Terraform Cloud
4. Update the Terraform configuration for remote backend

Step 1: Configure S3 backend and DynamoDB

1.1 Log in to the AWS Management Console and navigate to the **S3** service using the search bar



1.2 Click on **Create bucket**

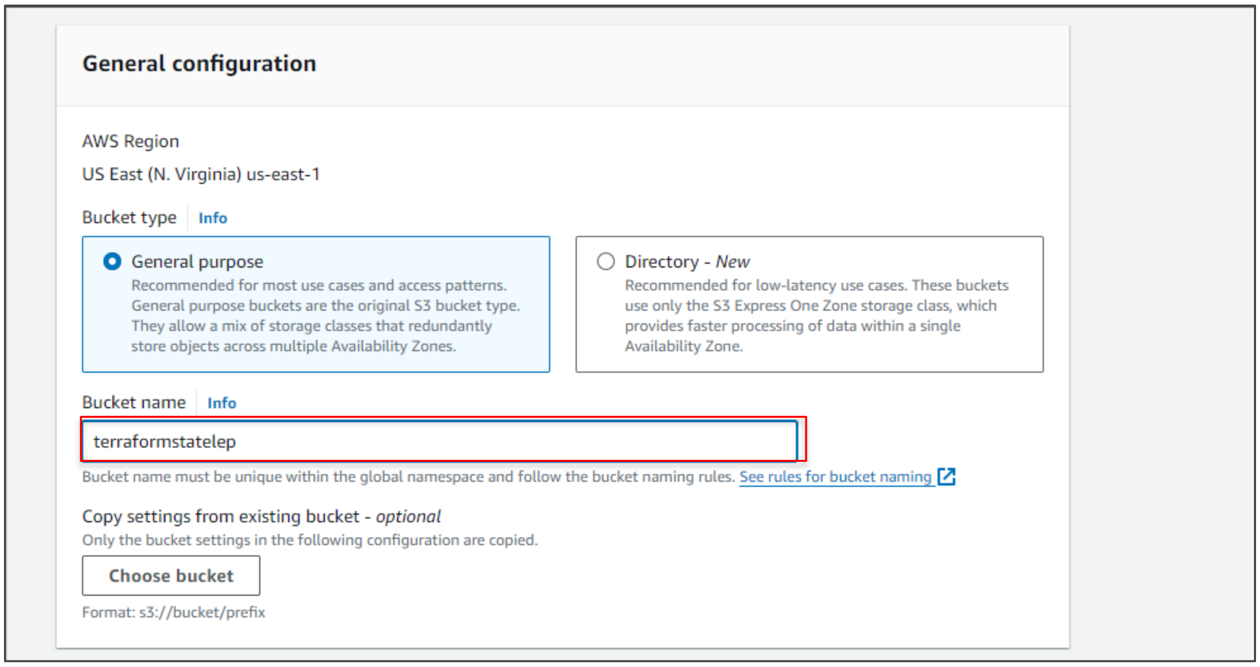


Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets are containers for data stored in S3.

	Name	AWS Region	IAM Access Analyzer	Creation date
○	implicitdenydemo	US East (N. Virginia) us-east-1	View analyzer for us-east-1	December 30, 2022, 11:31:45 (UTC+05:30)

1.3 Name it as **terraformstatelep**



General configuration

AWS Region

US East (N. Virginia) us-east-1

Bucket type Info

[✓ General purpose] Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Directory - *New* Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name Info

terraformstatelep

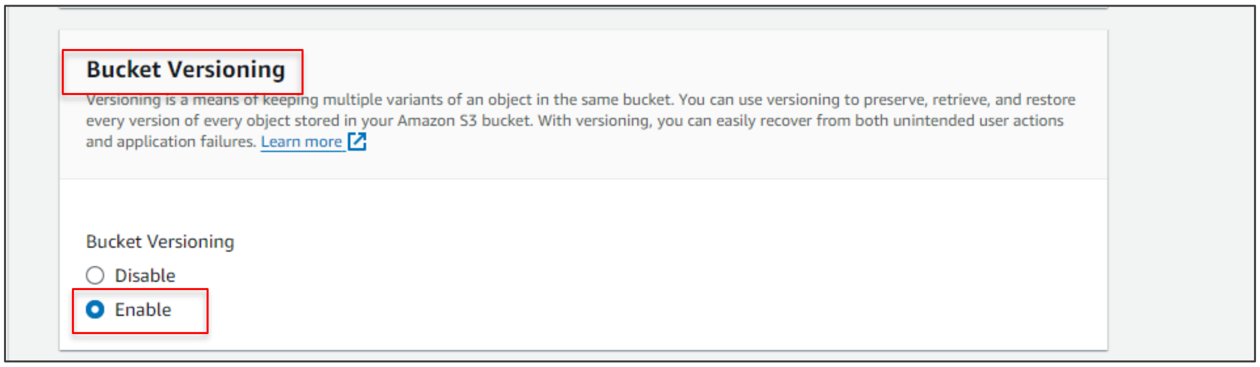
Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Format: s3://bucket/prefix

1.4 Scroll down to **Bucket Versioning** and click on **Enable**



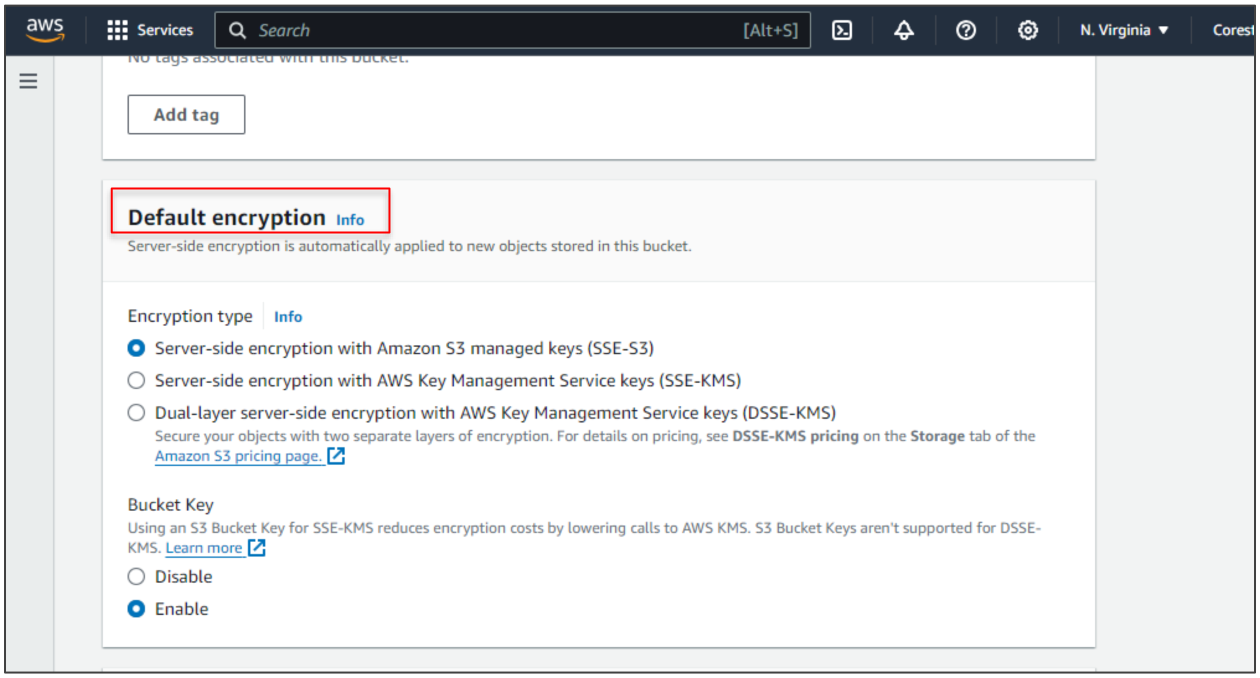
Bucket Versioning

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

Disable [✓ Enable]

1.5 Scroll down to **Default encryption** and make sure it is enabled



No tags associated with this bucket.

Default encryption Info

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

Encryption type Info

[✓ Server-side encryption with Amazon S3 managed keys (SSE-S3)]

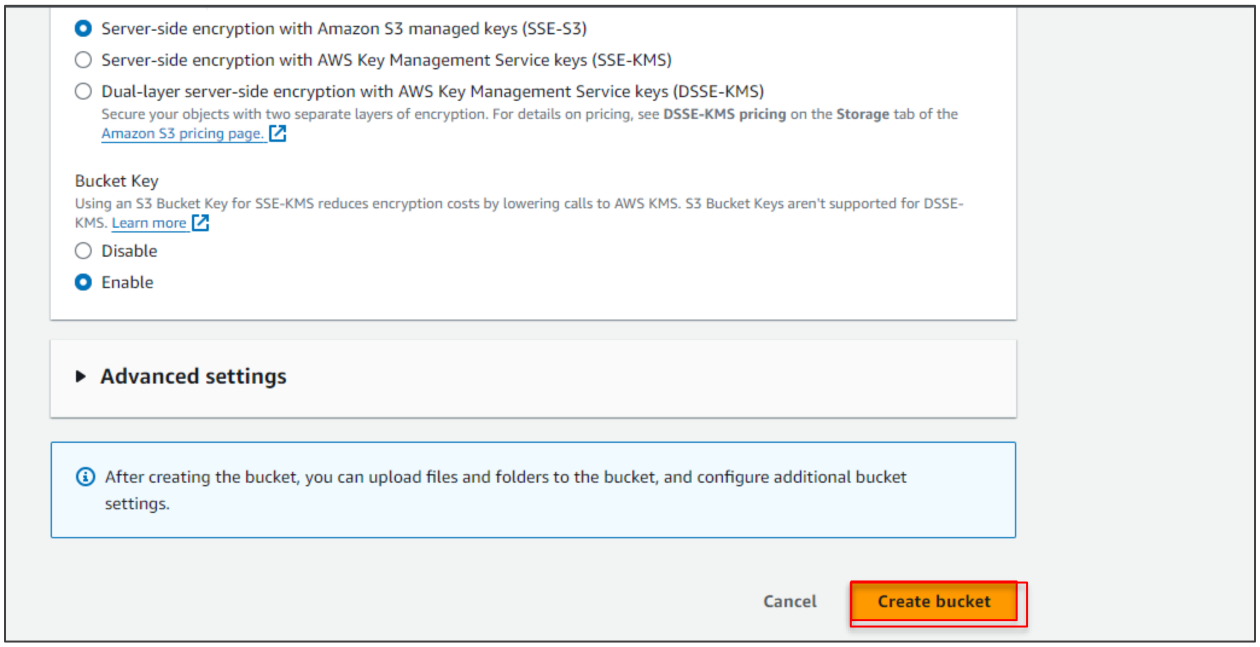
Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see **DSSE-KMS pricing** on the **Storage** tab of the Amazon S3 pricing page.

Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. Learn more

Disable [✓ Enable]

1.6 Retain all the other default configurations and click on **Create bucket**



[✓ Server-side encryption with Amazon S3 managed keys (SSE-S3)]

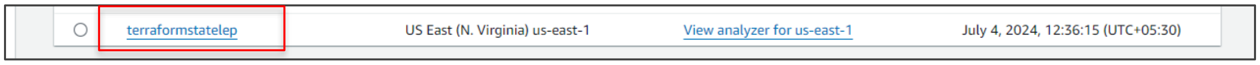
Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see **DSSE-KMS pricing** on the **Storage** tab of the Amazon S3 pricing page.

Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. Learn more

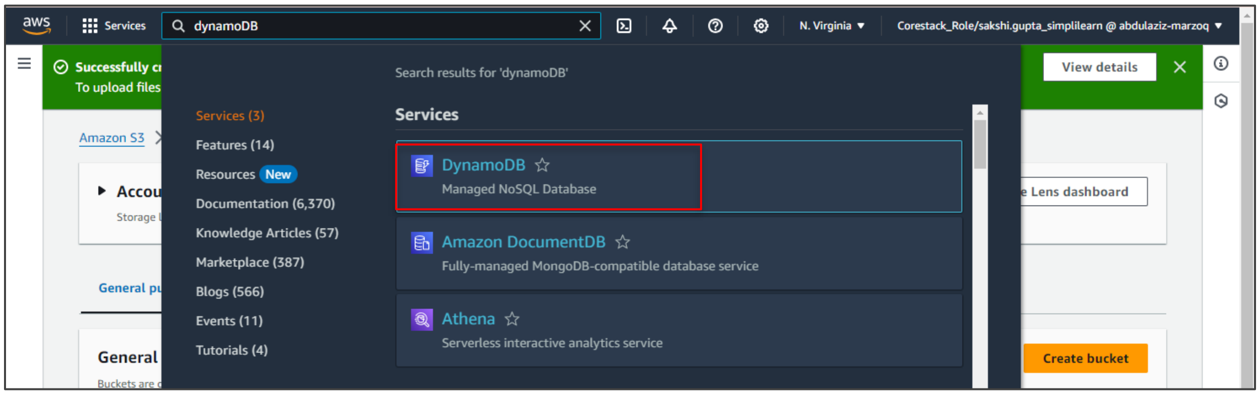
Disable [✓ Enable]

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

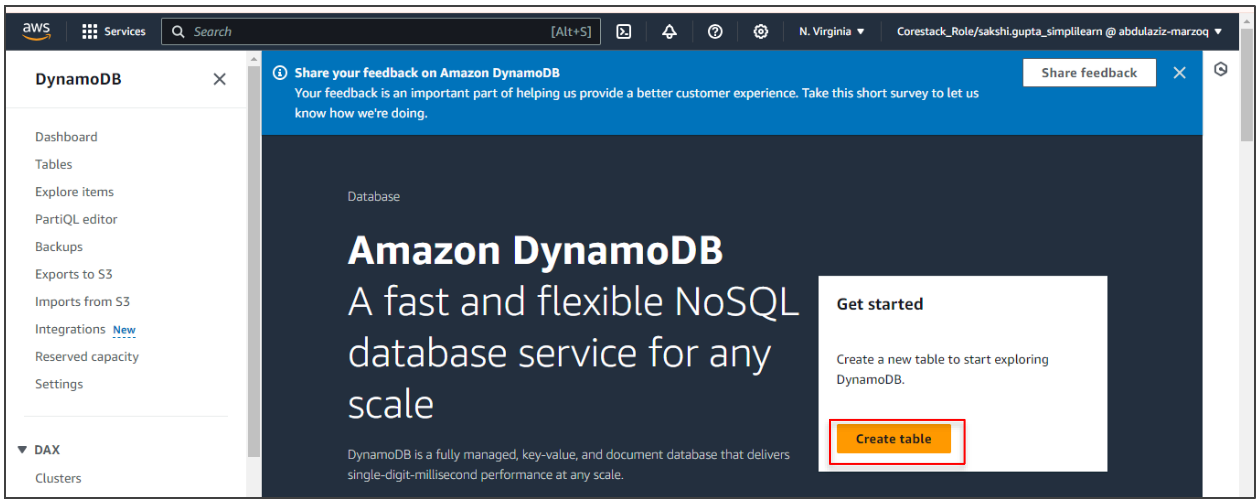


terraformstatelep	US East (N. Virginia) us-east-1	View analyzer for us-east-1	July 4, 2024, 12:36:15 (UTC+05:30)

1.7 Go to the search bar and search for **DynamoDB**



1.8 Click on **Create table**



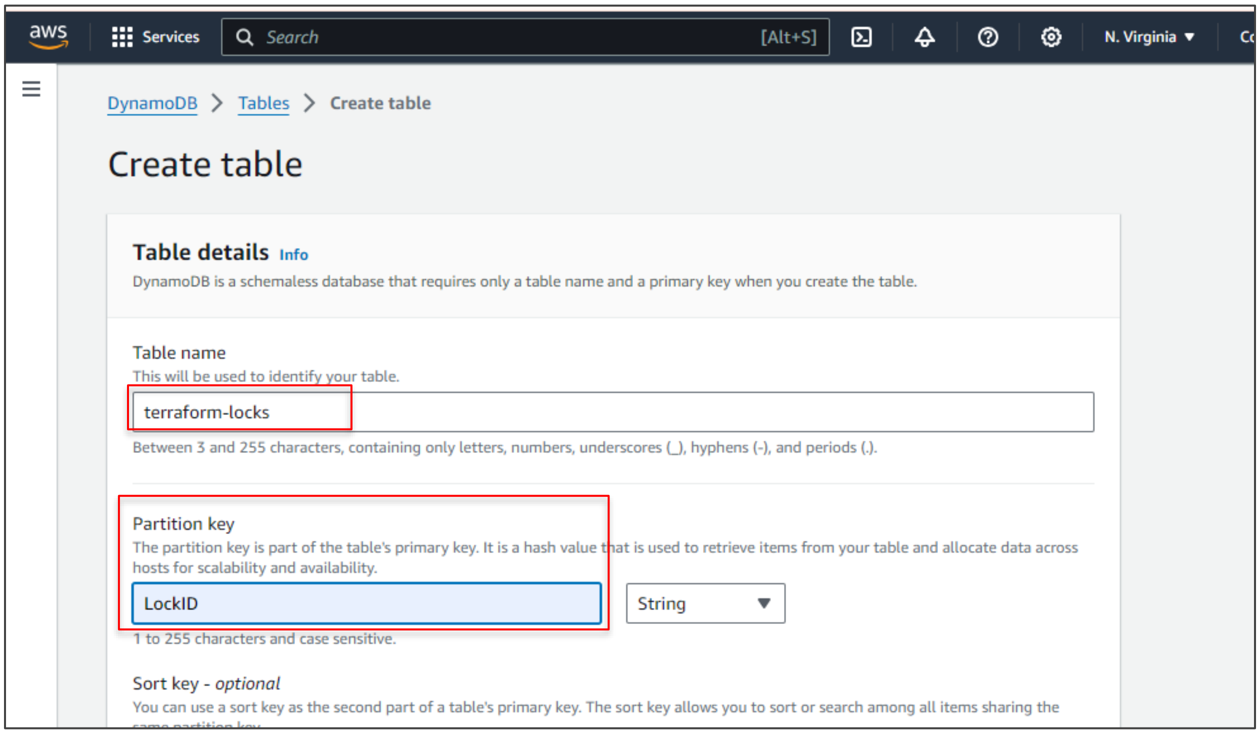
Share your feedback on Amazon DynamoDB
Your feedback is an important part of helping us provide a better customer experience. Take this short survey to let us know how we're doing.

Amazon DynamoDB
A fast and flexible NoSQL database service for any scale

DynamoDB is a fully managed, key-value, and document database that delivers single-digit-millisecond performance at any scale.

Create a new table to start exploring DynamoDB.

1.9 Name the table as **terraform-locks** and add a **Partition key** named **LockID**



Create table

Table details Info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name

This will be used to identify your table.

terraform-locks

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

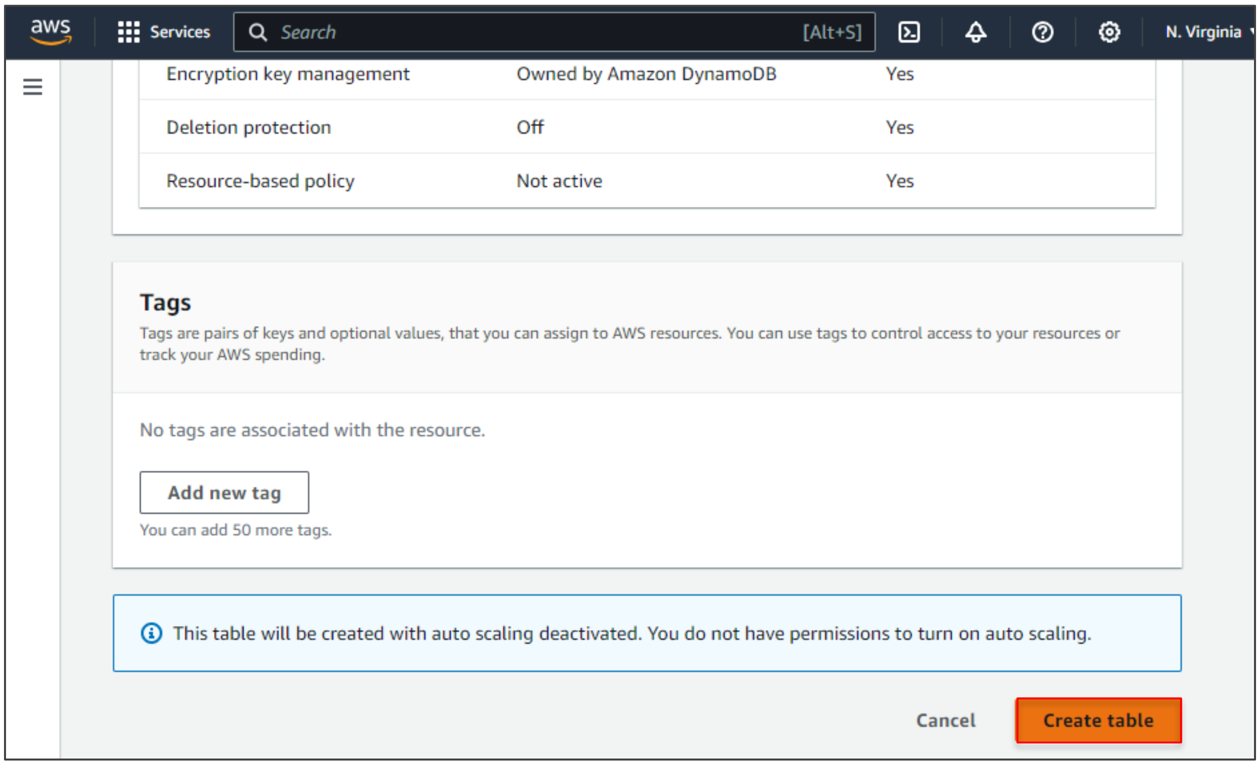
LockID | String

1 to 255 characters and case sensitive.

Sort key - *optional*

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the

1.10 Keep all the other configurations as default and click on **Create table**



Encryption key management	Owned by Amazon DynamoDB	Yes
Deletion protection	Off	Yes
Resource-based policy	Not active	Yes

Tags

Tags are pairs of keys and optional values, that you can assign to AWS resources. You can use tags to control access to your resources or track your AWS spending.

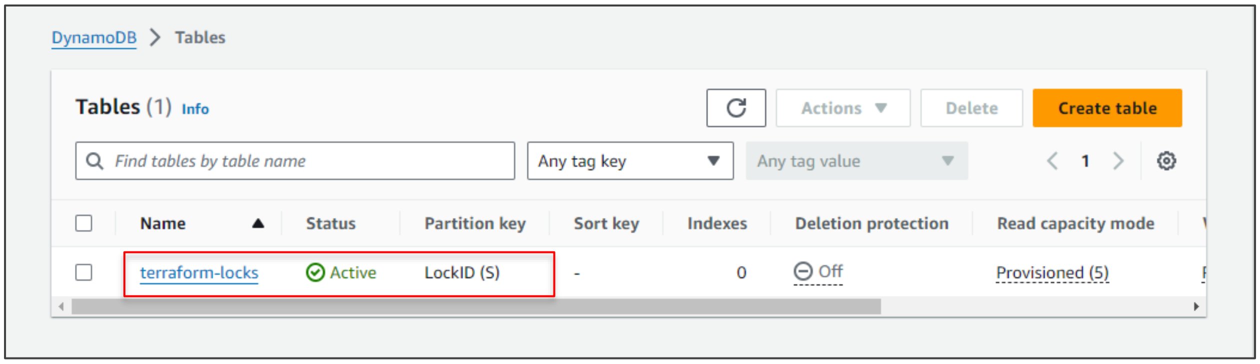
No tags are associated with the resource.

Add new tag

You can add 50 more tags.

This table will be created with auto scaling deactivated. You do not have permissions to turn on auto scaling.

Cancel | Create table

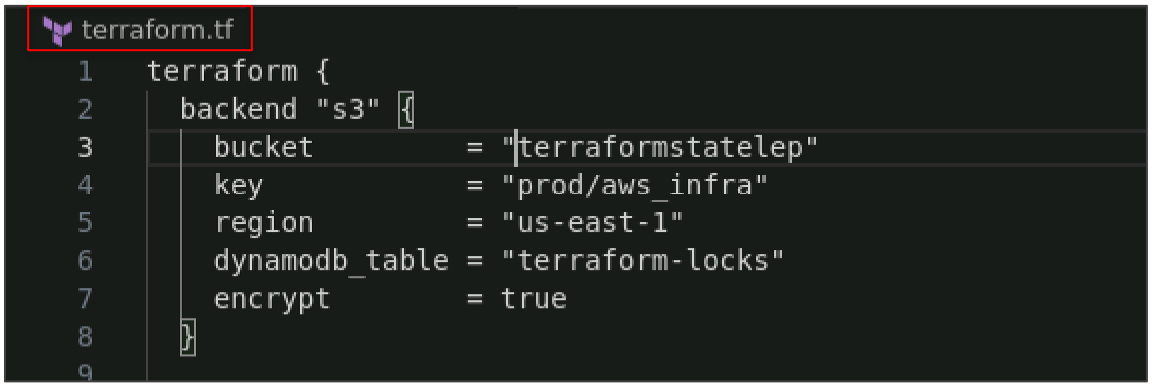


Name	Status	Partition key	Sort key	Indexes	Deletion protection	Read capacity mode
terraform-locks	Active	LockID (S)	-	0	Off	Provisioned (5)

Step 2: Update the Terraform configuration for S3 backend

2.1 Go to **terraform.tf** in your working directory, and update the terraform block using the following code:

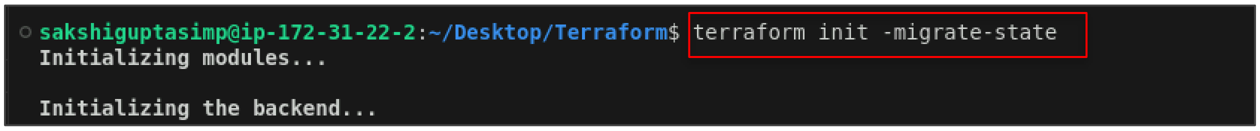
terraform {
 backend "s3" {
  bucket   = "terraformstatelep"
  key   = "prod/aws_infra"
  region   = "us-east-1"
  dynamodb_table = "terraform-locks"
  encrypt   = true
 }
}



terraform.tf

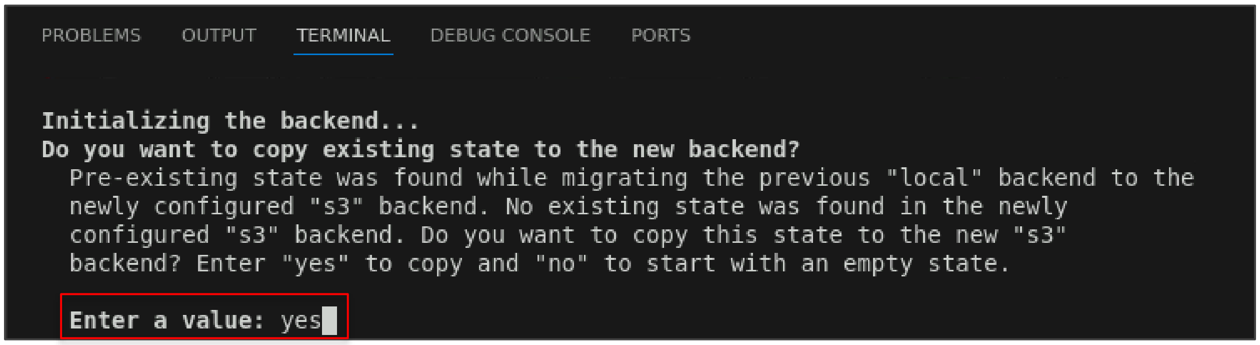
```
terraform {
  backend "s3" {
    bucket         = "terraformstatelep"
    key            = "prod/aws_infra"
    region         = "us-east-1"
    dynamodb_table = "terraform-locks"
    encrypt        = true
  }
```

2.2 Initialize and migrate state to S3 backend using the following command:
terraform init -migrate-state

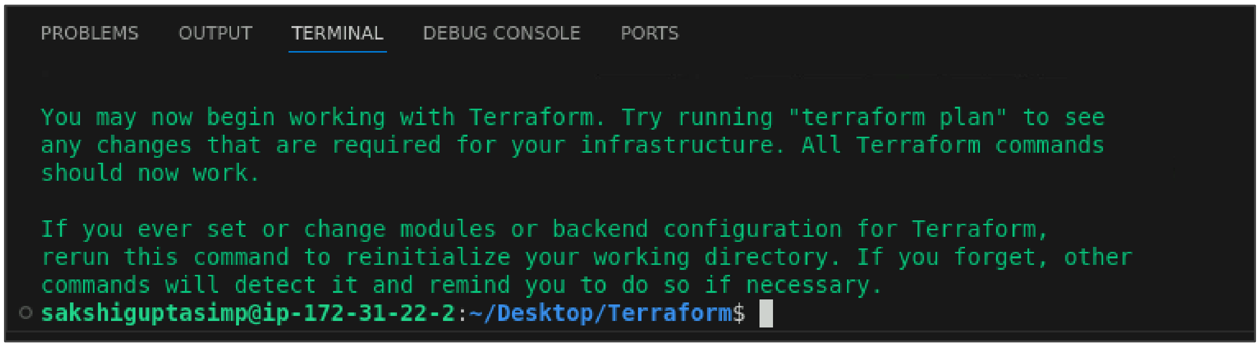


```
sakshiguptasimp@ip-172-31-22-2:~/Desktop/Terraform$ terraform init -migrate-state
Initializing modules...

Initializing the backend...
```

2.3 When prompted, approve the migration by typing **yes**

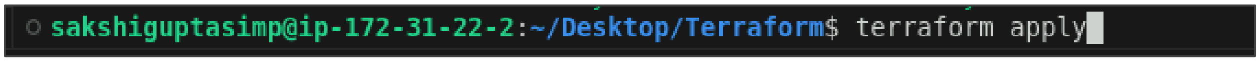


```
Initializing the backend...
Do you want to copy existing state to the new backend?
  Pre-existing state was found while migrating the previous "local" backend to the
  newly configured "s3" backend. No existing state was found in the newly
  configured "s3" backend. Do you want to copy this state to the new "s3"
  backend? Enter "yes" to copy and "no" to start with an empty state.

  Enter a value: yes
```



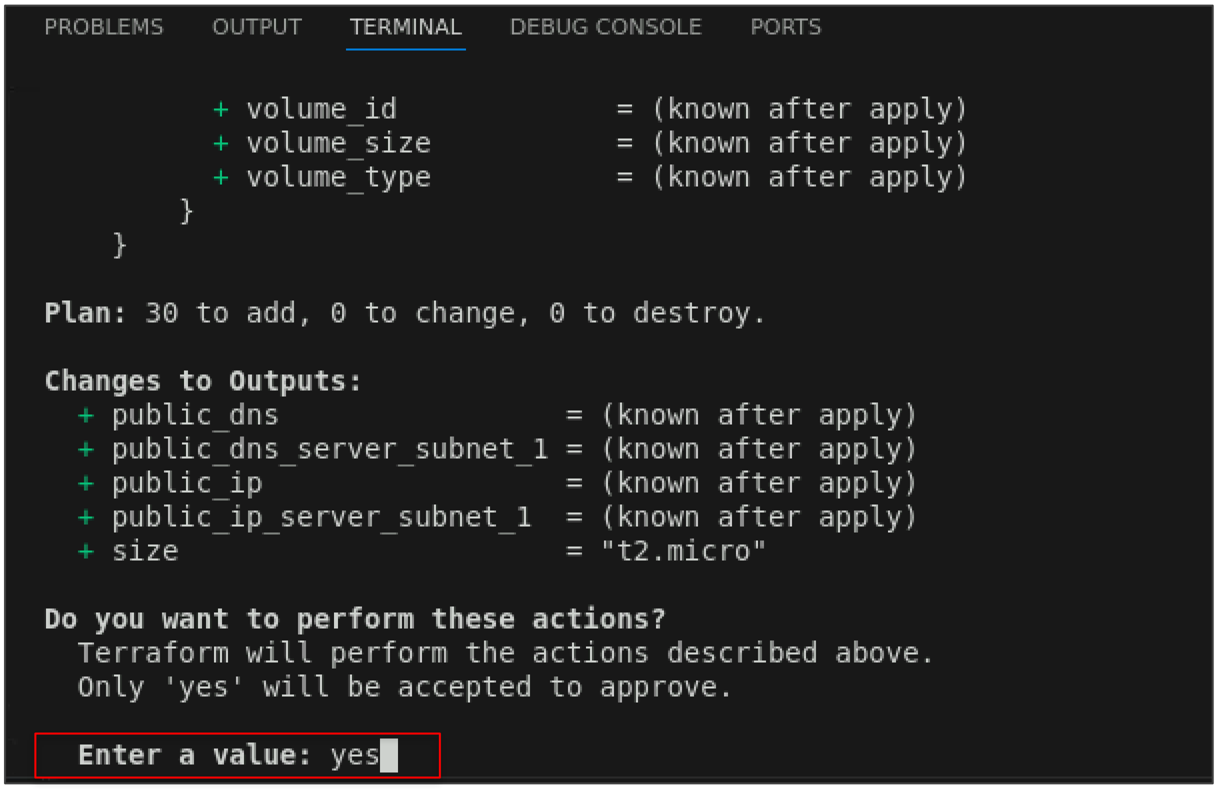
```
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

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$
```

2.4 Apply the Terraform configuration using the following command:
terraform apply

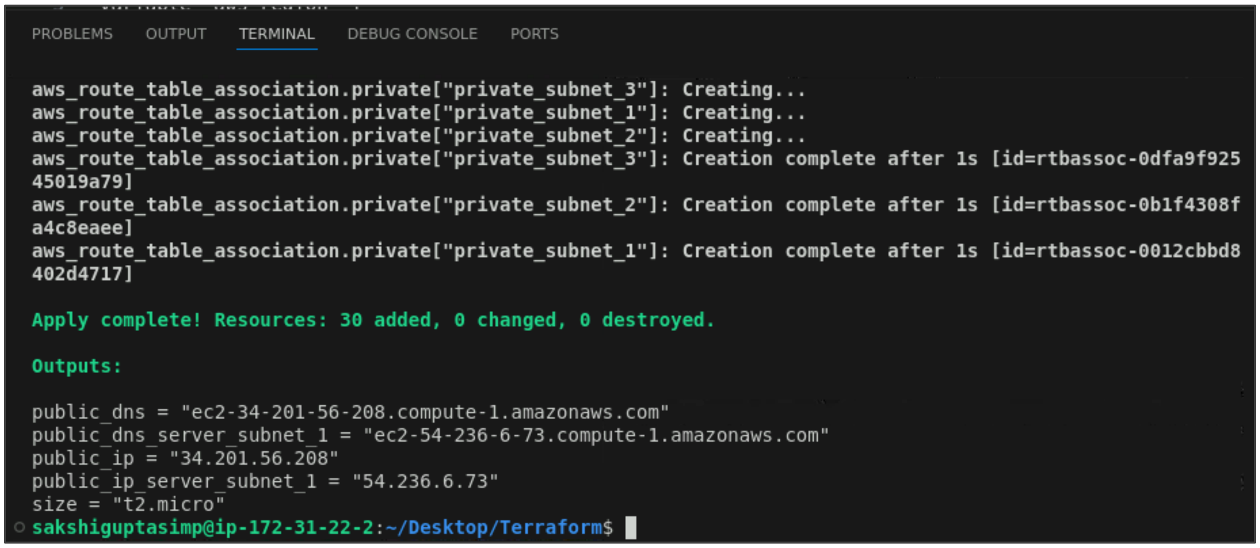


```
sakshiguptasimp@ip-172-31-22-2:~/Desktop/Terraform$ terraform apply
```

2.5 When prompted, approve the actions by typing **yes**

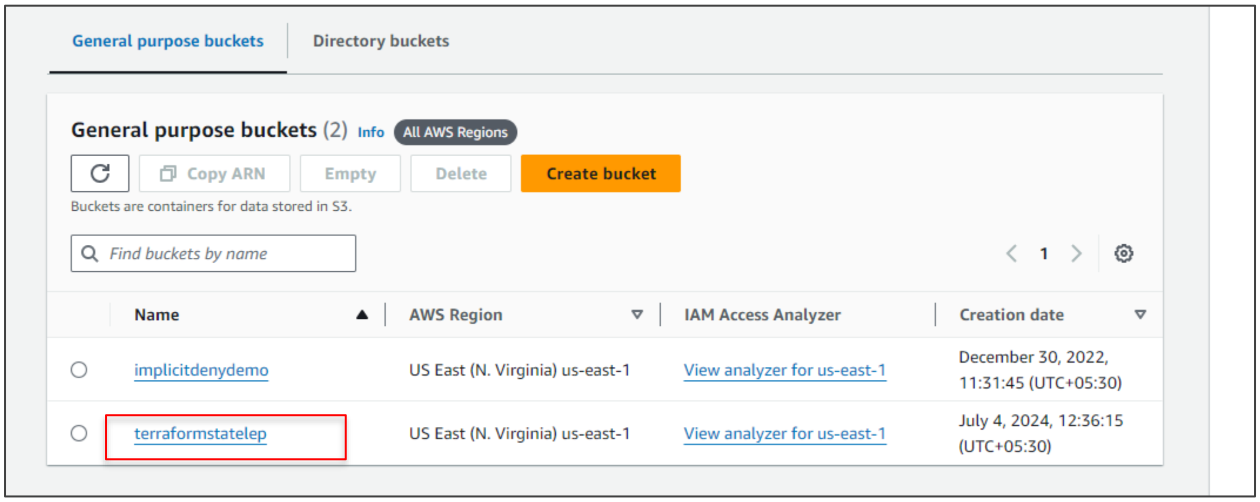


```
          + volume_id              = (known after apply)
          + volume_size            = (known after apply)
          + volume_type            = (known after apply)
        }
    }

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)
  + size                      = "t2.micro"

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes
```



```
aws_route_table_association.private["private_subnet_3"]: Creating...
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"]: Creation complete after 1s [id=rtbassoc-0dfa9f925
45019a79]
aws_route_table_association.private["private_subnet_2"]: Creation complete after 1s [id=rtbassoc-0b1f4308f
a4c8eaee]
aws_route_table_association.private["private_subnet_1"]: Creation complete after 1s [id=rtbassoc-0012cbbd8
402d4717]

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

Outputs:

public_dns = "ec2-34-201-56-208.compute-1.amazonaws.com"
public_dns_server_subnet_1 = "ec2-54-236-6-73.compute-1.amazonaws.com"
public_ip = "34.201.56.208"
public_ip_server_subnet_1 = "54.236.6.73"
size = "t2.micro"
sakshiguptasimp@ip-172-31-22-2:~/Desktop/Terraform$
```

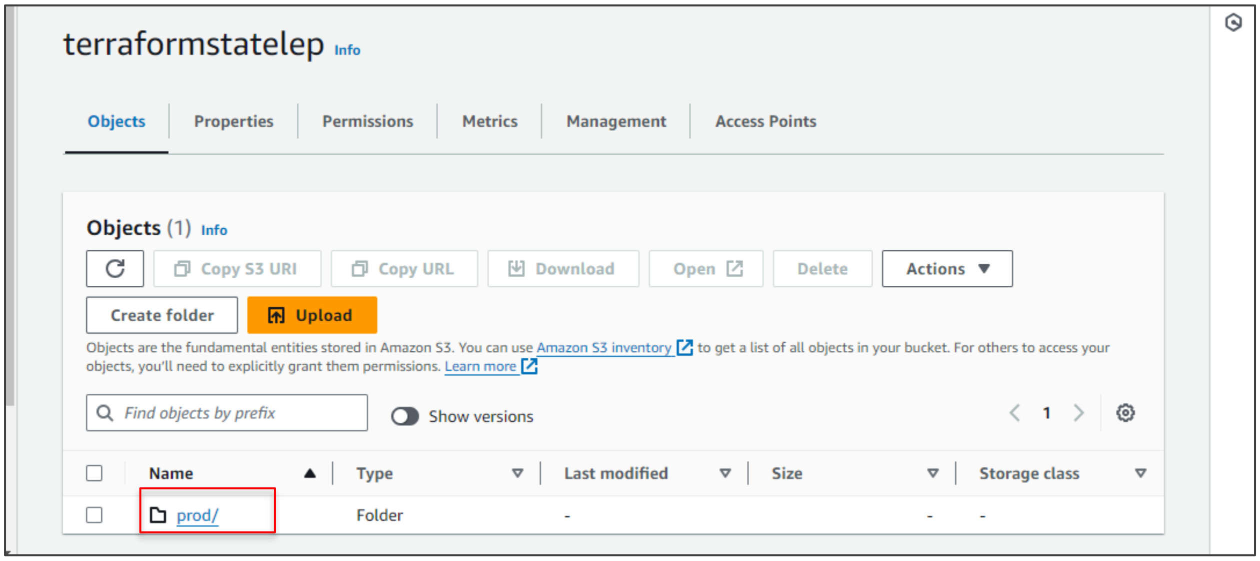
2.6 Go to the created bucket in the AWS Management Console and observe that the object has been added. Click on it to view it in detail.



Buckets are containers for data stored in S3.

	Name	AWS Region	IAM Access Analyzer	Creation date
○	implicitdenydemo	US East (N. Virginia) us-east-1	View analyzer for us-east-1	December 30, 2022, 11:31:45 (UTC+05:30)
○	terraformstatelep	US East (N. Virginia) us-east-1	View analyzer for us-east-1	July 4, 2024, 12:36:15 (UTC+05:30)

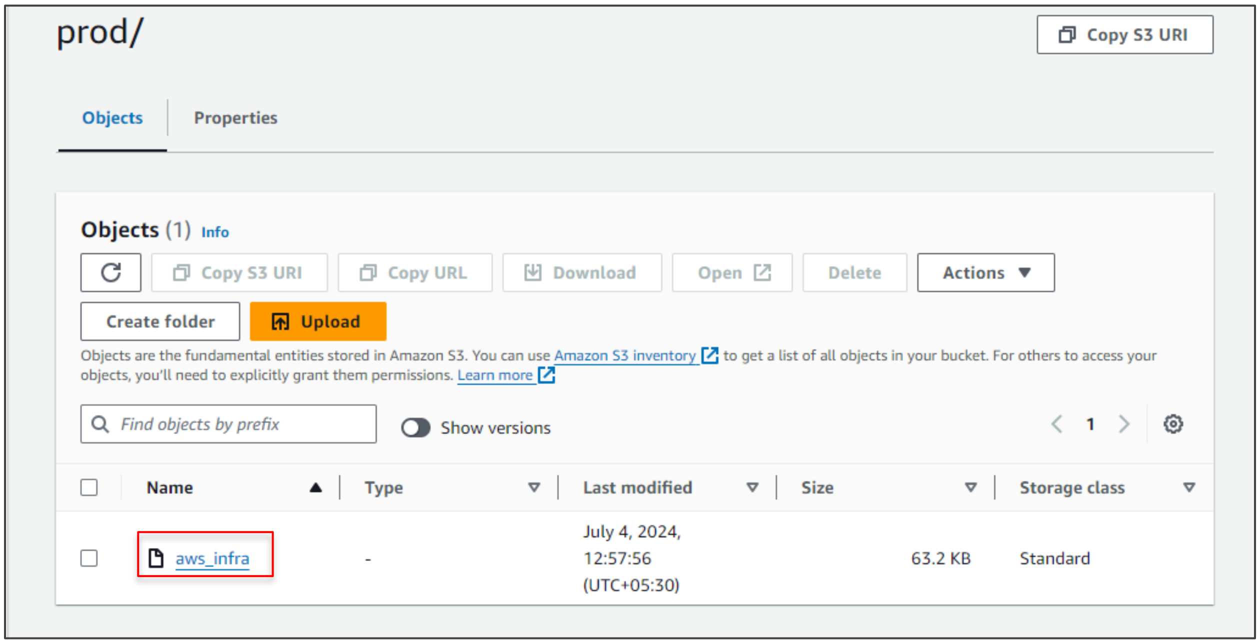
2.7 Click on **prod/**



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

	Name	Type	Last modified	Size	Storage class
☐	prod/	Folder	-	-	-

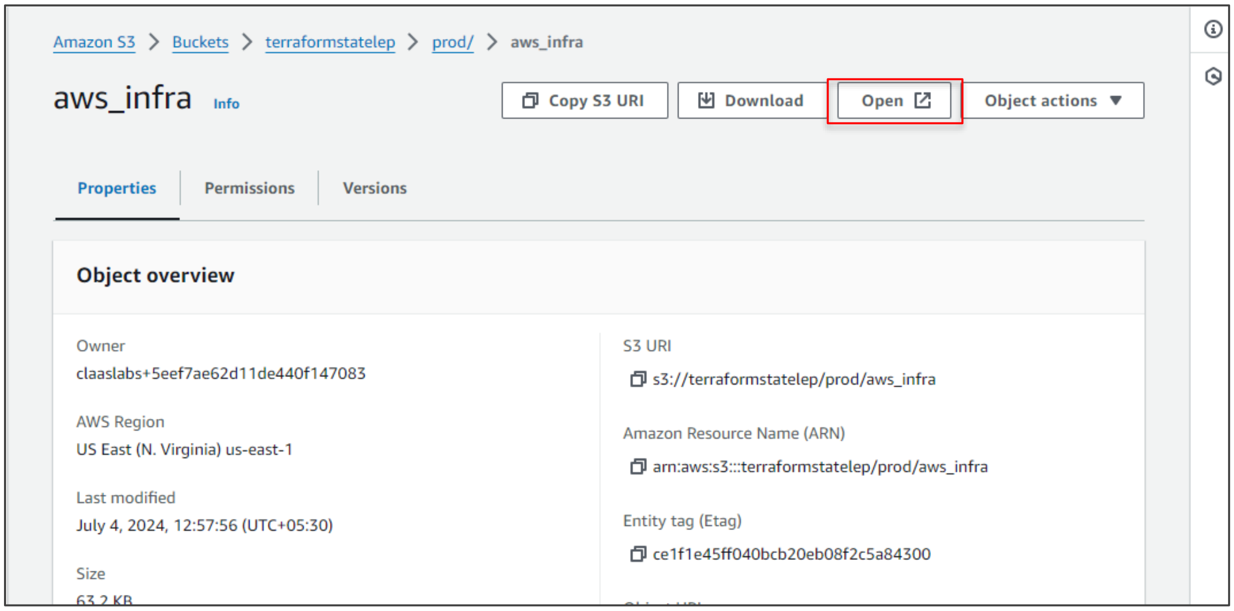
2.8 Click on **aws_infra**



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
aws_infra	-	July 4, 2024, 12:57:56 (UTC+05:30)	63.2 KB	Standard

2.9 Click on **Open**, and you will be able to see that your Terraform state file has been successfully migrated to the S3 backend



Owner
claaslabs+5eef7ae62d11de440f147083

AWS Region
US East (N. Virginia) us-east-1

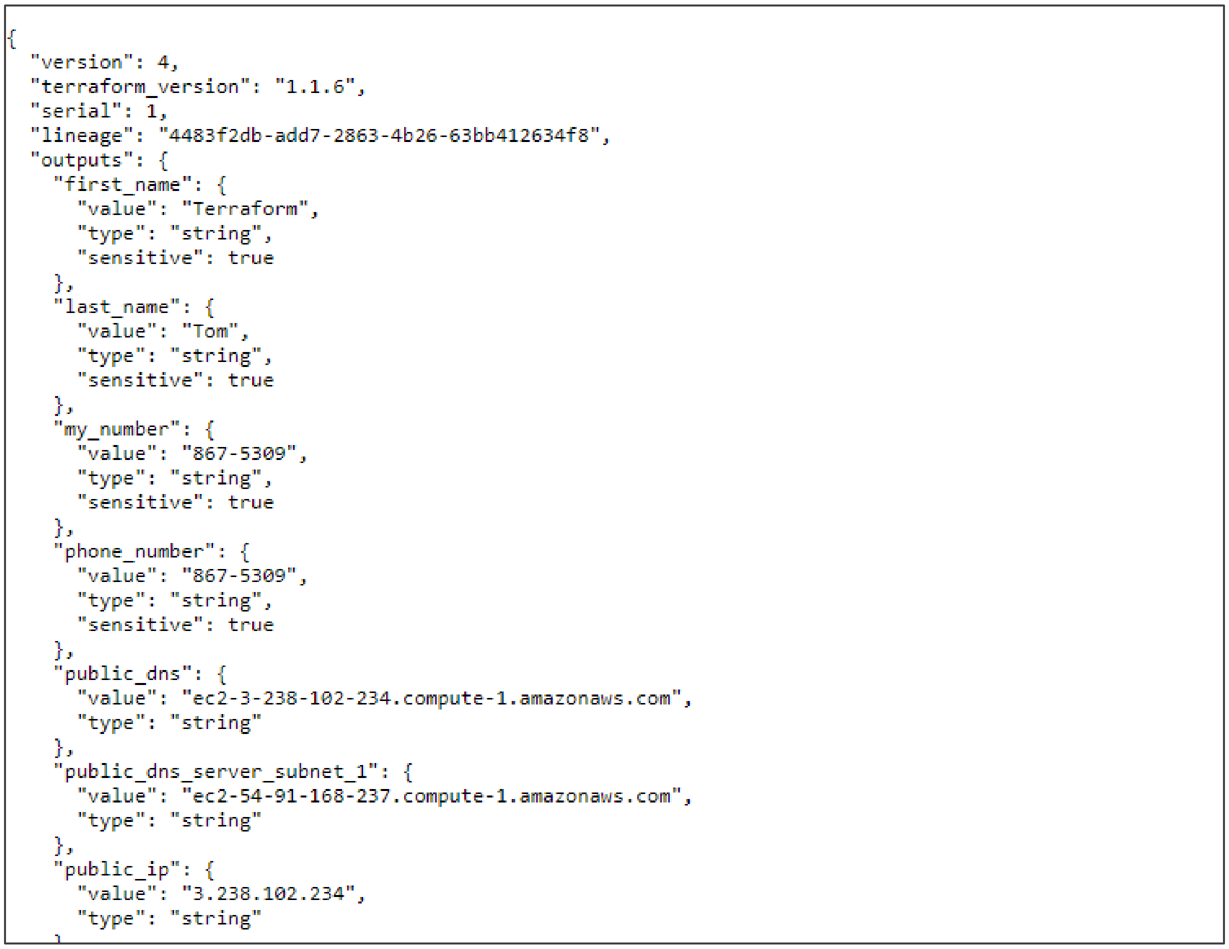
Last modified
July 4, 2024, 12:57:56 (UTC+05:30)

Size
63.2 KB

S3 URI
s3://terraformstatelep/prod/aws_infra

Amazon Resource Name (ARN)
arn:aws:s3:::terraformstatelep/prod/aws_infra

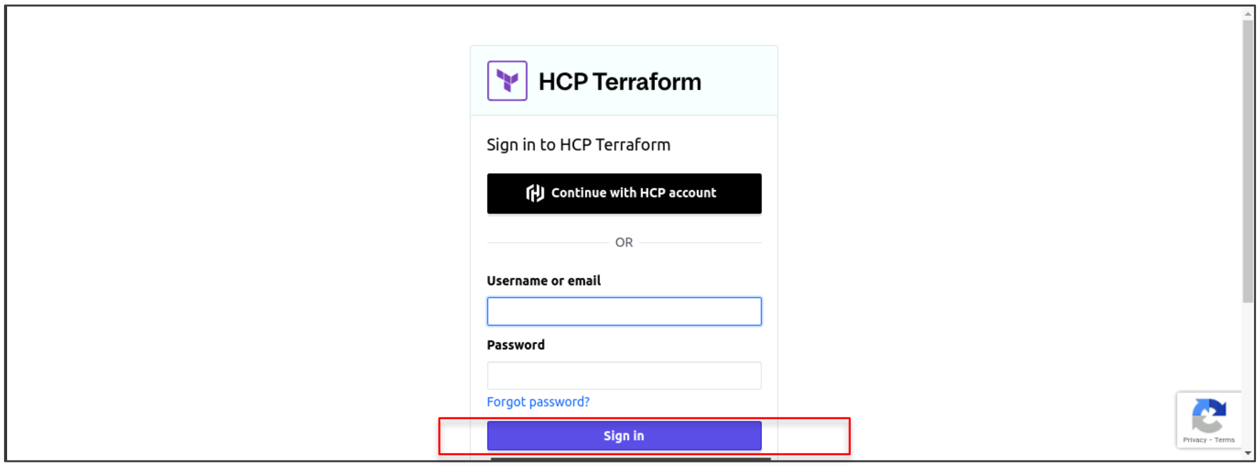
Entity tag (Etag)
ce1f1e45ff040bcb20eb08f2c5a84300



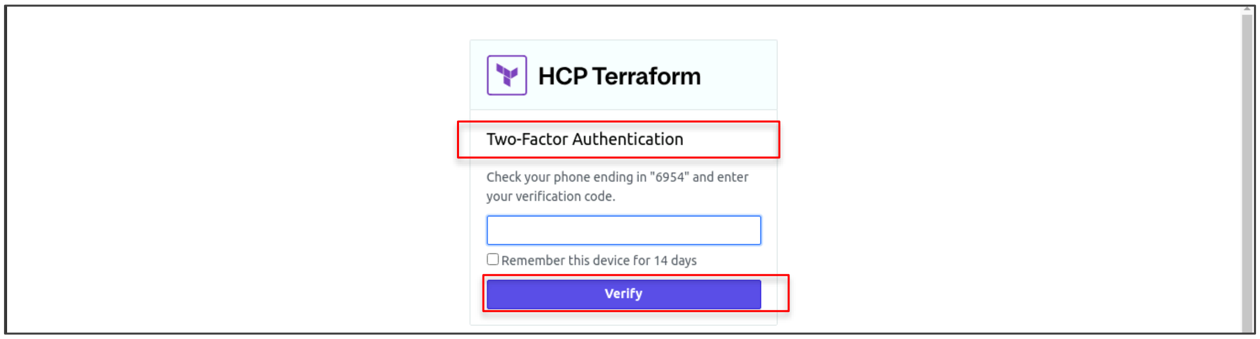
```
{
  "version": 4,
  "terraform_version": "1.1.6",
  "serial": 1,
  "lineage": "4483f2db-add7-2863-4b26-63bb412634f8",
  "outputs": {
    "first_name": {
      "value": "Terraform",
      "type": "string",
      "sensitive": true
    },
    "last_name": {
      "value": "Tom",
      "type": "string",
      "sensitive": true
    },
    "my_number": {
      "value": "867-5309",
      "type": "string",
      "sensitive": true
    },
    "phone_number": {
      "value": "867-5309",
      "type": "string",
      "sensitive": true
    },
    "public_dns": {
      "value": "ec2-3-238-102-234.compute-1.amazonaws.com",
      "type": "string"
    },
    "public_dns_server_subnet_1": {
      "value": "ec2-54-91-168-237.compute-1.amazonaws.com",
      "type": "string"
    },
    "public_ip": {
      "value": "3.238.102.234",
      "type": "string"
```

Step 3: Migrate state to remote backend with Terraform Cloud

3.1 Use the following URL to go to Terraform Cloud. Enter the required details and click on **Sign In**:
https://app.terraform.io/session



3.2 Click **Verify** after entering your verification code


HCP Terraform

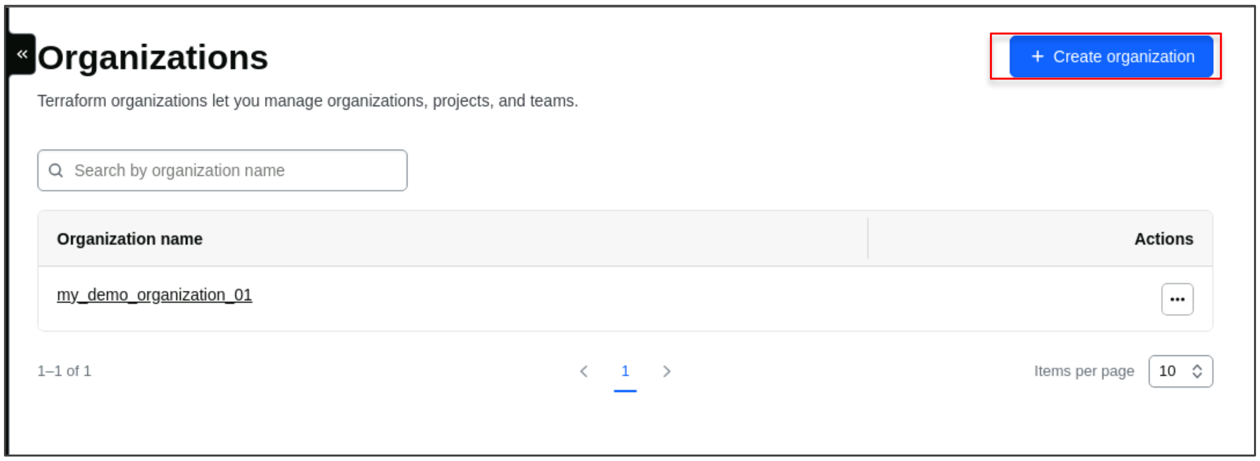
Two-Factor Authentication

Check your phone ending in "6954" and enter your verification code.

Remember this device for 14 days

Verify

3.3 Click on **Create organization**

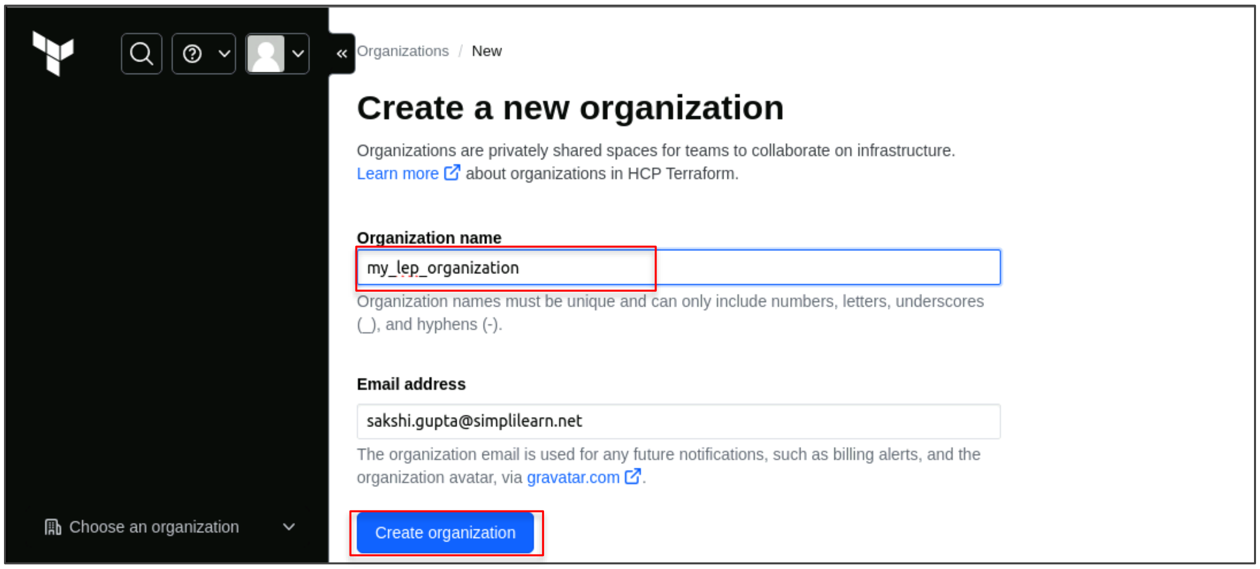

Organizations

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

Organization name	Actions
my_demo_organization_01	•••

1–1 of 1

3.4 Name the organization as **my_lep_organization** and click on **Create organization**


Organizations / New

Create a new organization

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

Organization name

my_lep_organization

Organization names must be unique and can only include numbers, letters, underscores (_), and hyphens (-).

Email address

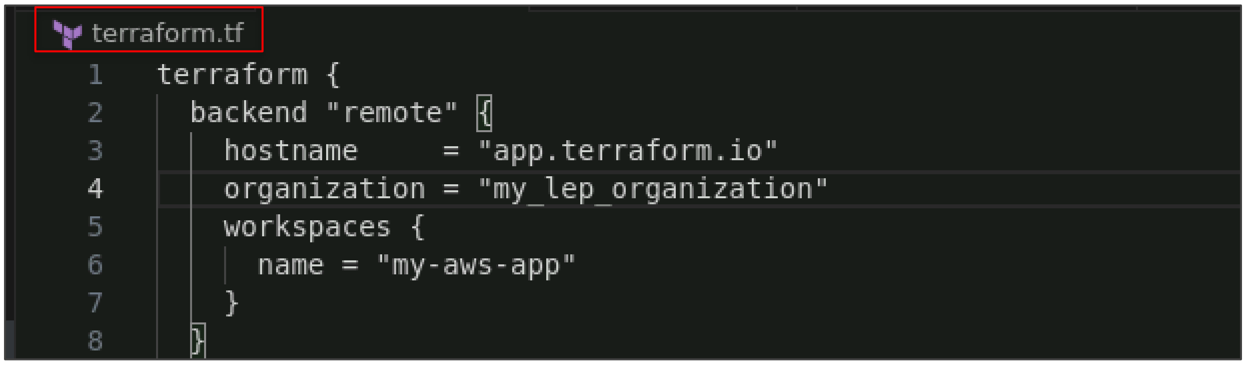
sakshi.gupta@simplilearn.net

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

Step 4: Update the Terraform configuration for remote backend

4.1 Go to **terraform.tf** in your working directory and update the terraform block using the following code:

```
terraform {
 backend "remote" {
  hostname     = "app.terraform.io"
  organization = "my_lep_organization"
  workspaces {
   name = "my-aws-app"
  }
 }
```

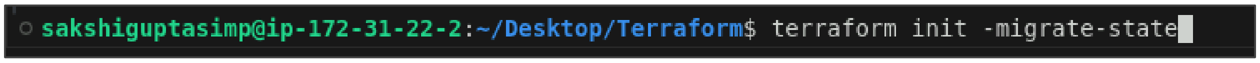


terraform.tf

```
terraform {
  backend "remote" {
    hostname     = "app.terraform.io"
    organization = "my_lep_organization"
    workspaces {
      name = "my-aws-app"
    }
  }
```

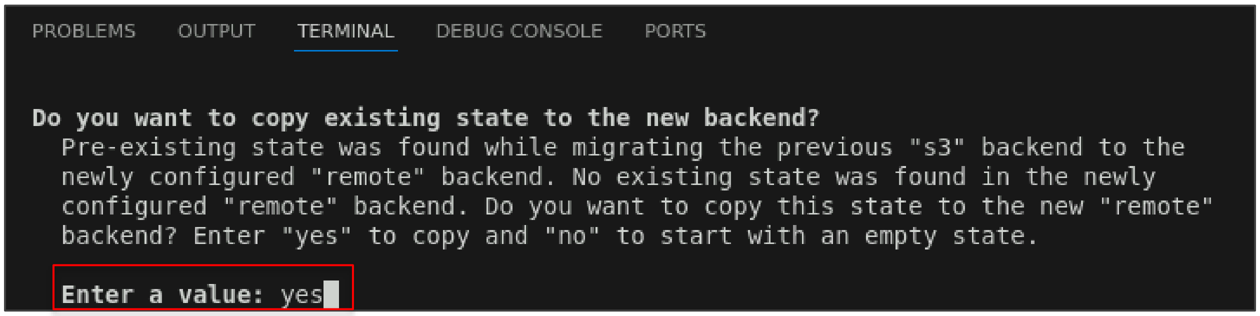
4.2 Initialize and migrate state to Terraform Cloud using the following command:

terraform init -migrate-state

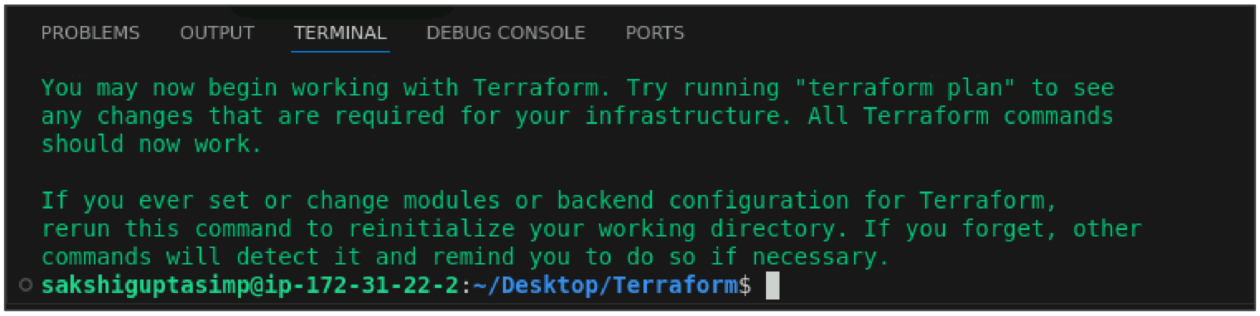


```
sakshiguptasimp@ip-172-31-22-2:~/Desktop/Terraform$ terraform init -migrate-state
```

4.3 When prompted, approve the initialization by typing **yes**

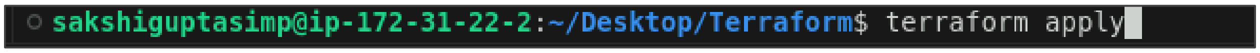


```
Do you want to copy existing state to the new backend?
  Pre-existing state was found while migrating the previous "s3" backend to the
  newly configured "remote" backend. No existing state was found in the newly
  configured "remote" backend. Do you want to copy this state to the new "remote"
  backend? Enter "yes" to copy and "no" to start with an empty state.

  Enter a value: yes
```

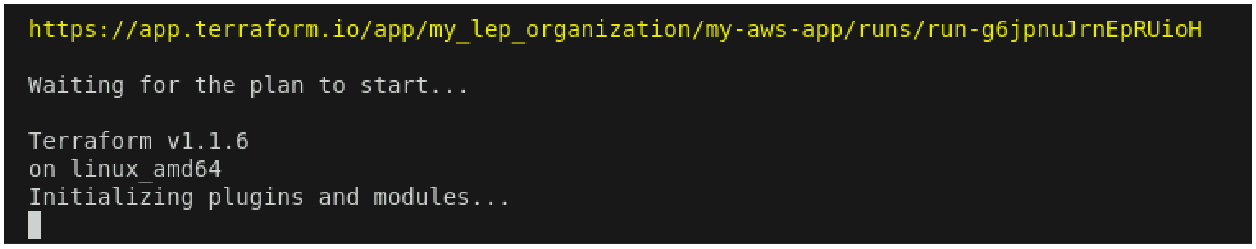


```
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

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$
```

4.4 Apply the Terraform configuration using the following command:
terraform apply

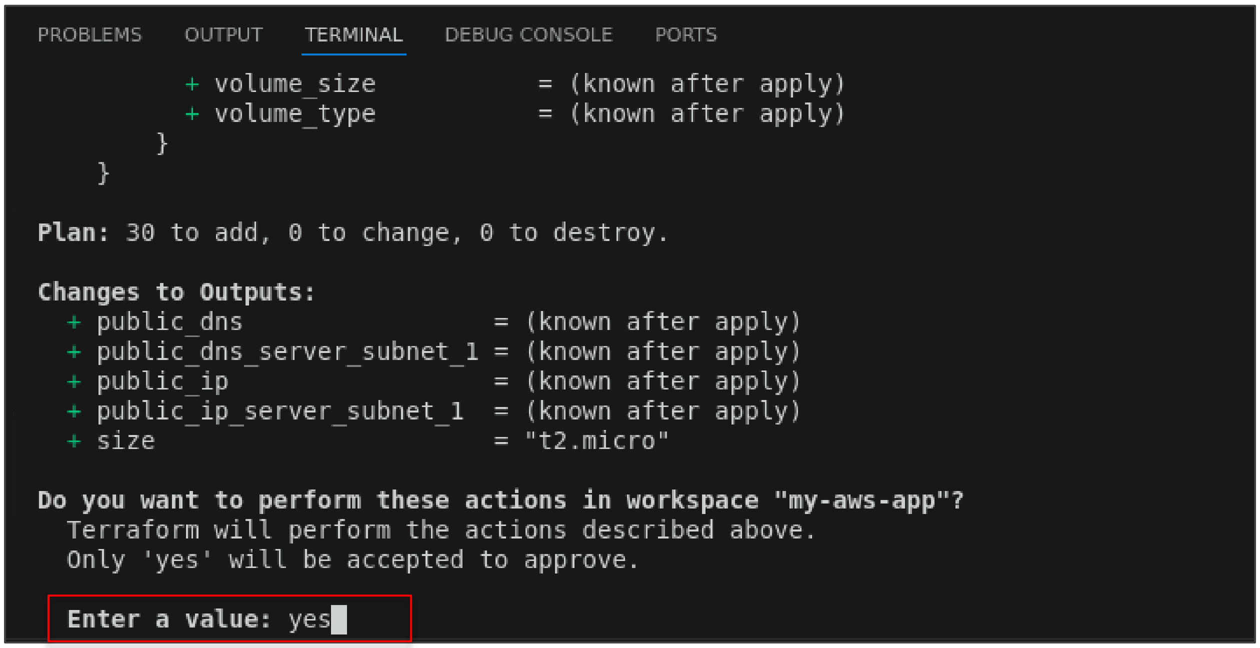


```
sakshiguptasimp@ip-172-31-22-2:~/Desktop/Terraform$ terraform apply
```

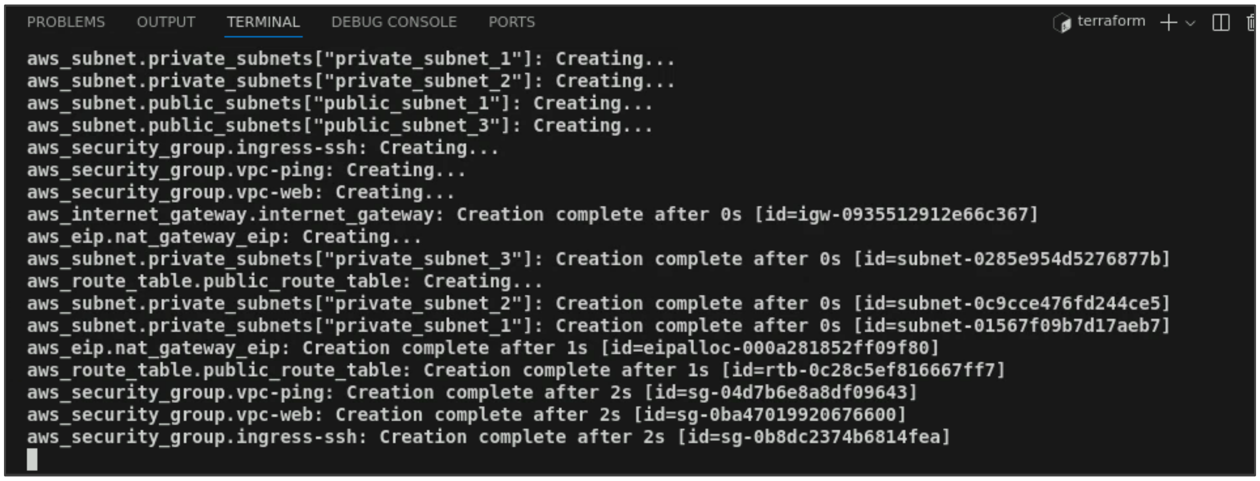


```
https://app.terraform.io/app/my_lep_organization/my-aws-app/runs/run-g6jpnuJrnEpRUioH

Waiting for the plan to start...

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

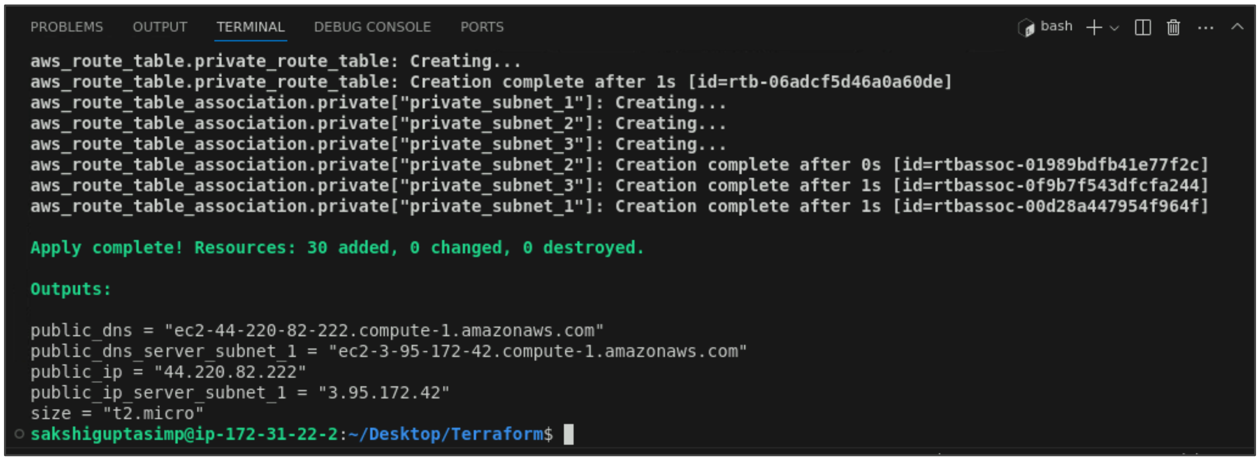
4.5 When prompted, approve the changes by typing **yes**



```
          + 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
```

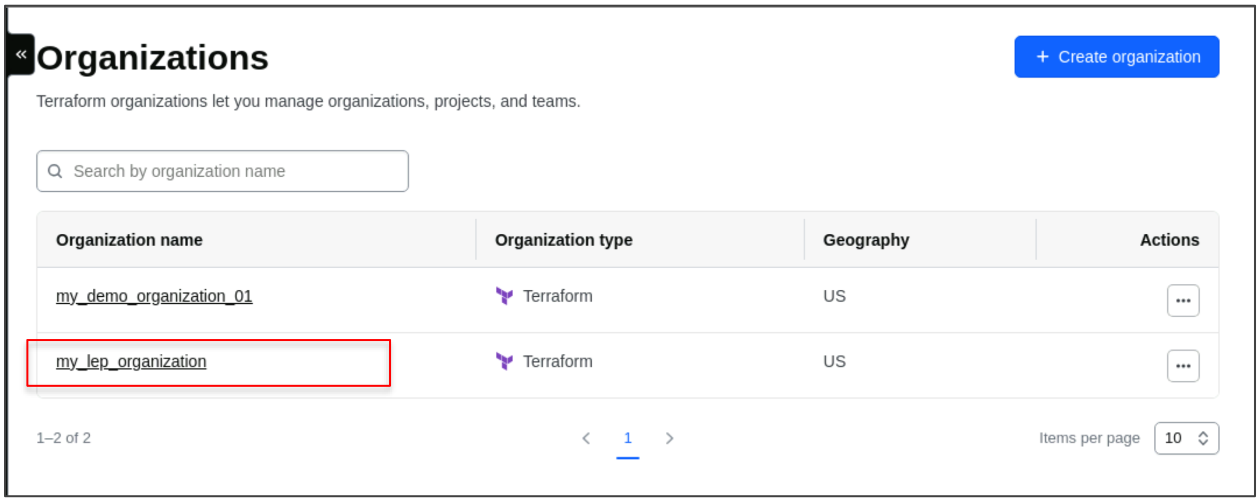


```
aws_subnet.private_subnets["private_subnet_1"]: Creating...
aws_subnet.private_subnets["private_subnet_2"]: Creating...
aws_subnet.public_subnets["public_subnet_1"]: Creating...
aws_subnet.public_subnets["public_subnet_3"]: Creating...
aws_security_group.ingress-ssh: Creating...
aws_security_group.vpc-ping: 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]
```



```
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$
```

4.6 Go to Terraform Cloud to validate the migration. Click on the organization in which you are working.



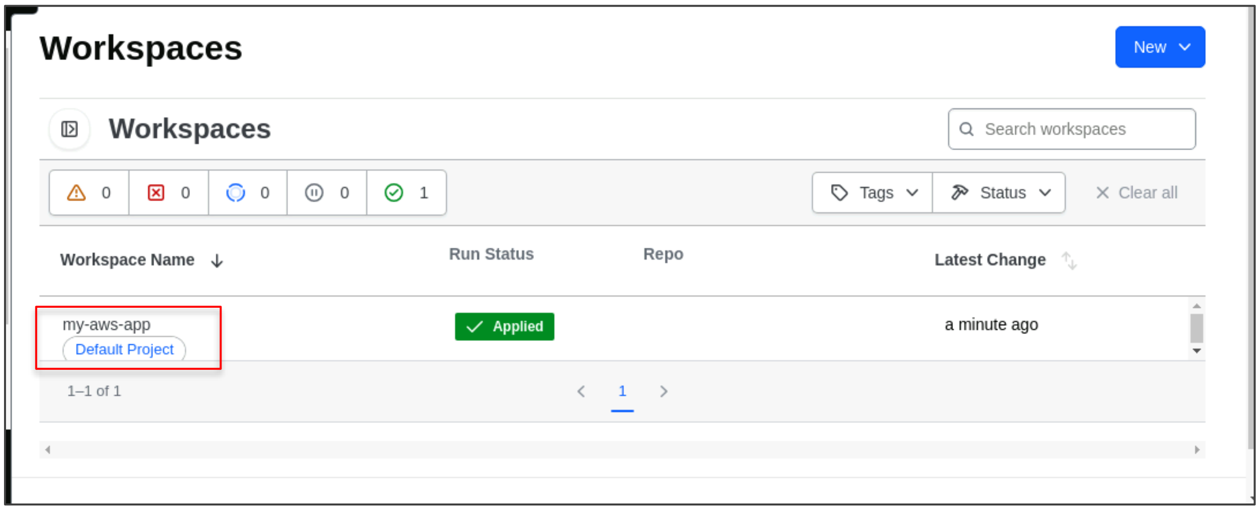
Organizations

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

Organization name	Organization type	Geography	Actions
my_demo_organization_01	Terraform	US	...
my_lep_organization	Terraform	US	...

1–2 of 2   1   Items per page 10

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



Workspaces

New

Workspaces

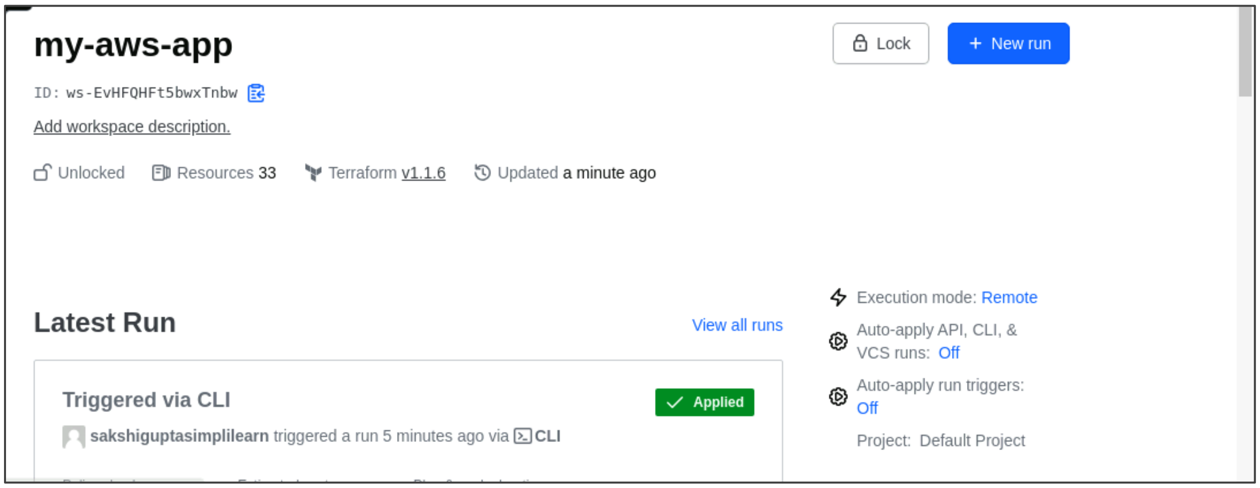
Search workspaces

⚠ 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



my-aws-app

Lock + New run

ID: ws-EvHFQHFt5bwxTnbw

Add workspace description.

Unlocked Resources 33 Terraform v1.1.6 Updated a minute ago

Latest Run

View all runs

Triggered via CLI ✓ Applied

sakshiguptasimplilearn triggered a run 5 minutes ago via CLI

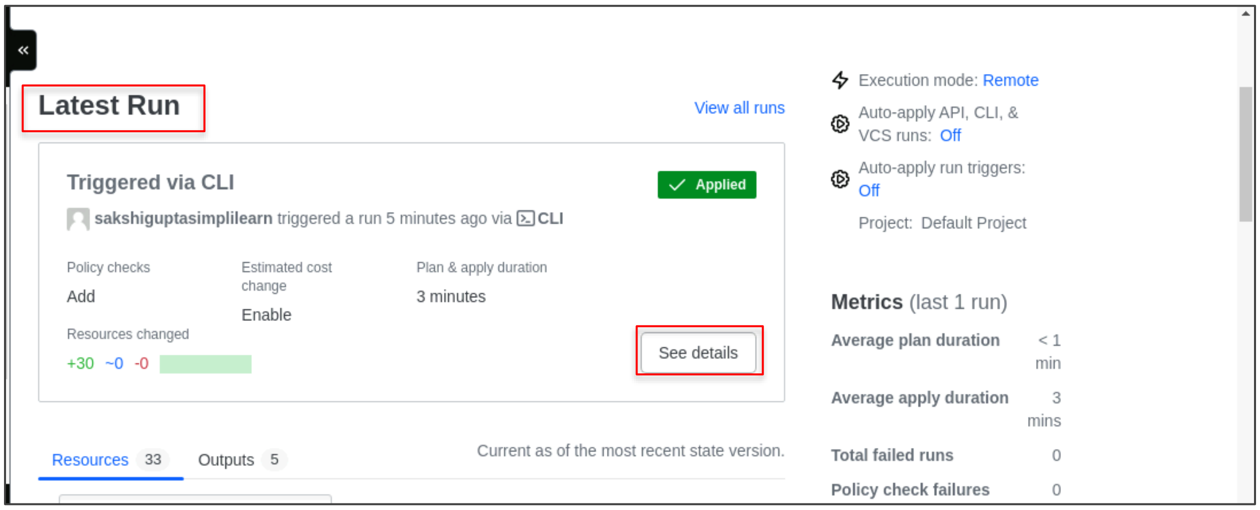
Execution mode: Remote

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

Auto-apply run triggers: Off

Project: Default Project

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



Latest Run

View all runs

Triggered via CLI

Applied

sakshiguptasimplilearn triggered a run 5 minutes ago via CLI

Policy checks	Estimated cost change	Plan & apply duration
Add	Enable	3 minutes

Resources changed

+30 ~0 -0

See details

Resources 33 Outputs 5

Current as of the most recent state version.

Execution mode: Remote

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

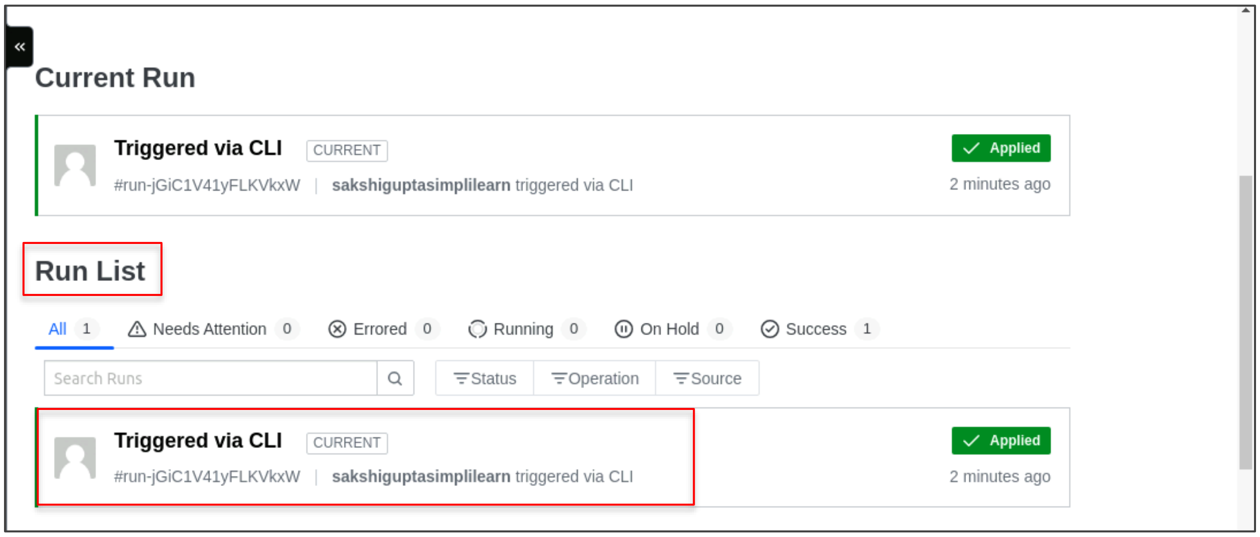
Auto-apply run triggers: Off

Project: Default Project

Metrics (last 1 run)

Average plan duration	< 1 min
Average apply duration	3 mins
Total failed runs	0
Policy check failures	0

4.9 Click on the latest **Triggered via CLI** under the **Run List**



Current Run

Triggered via CLI CURRENT

#run-jGiC1V41yFLKVkxW | sakshiguptasimplilearn triggered via CLI

Applied

2 minutes ago

Run List

All 1 Needs Attention 0 Errored 0 Running 0 On Hold 0 Success 1

Search Runs

Status Operation Source

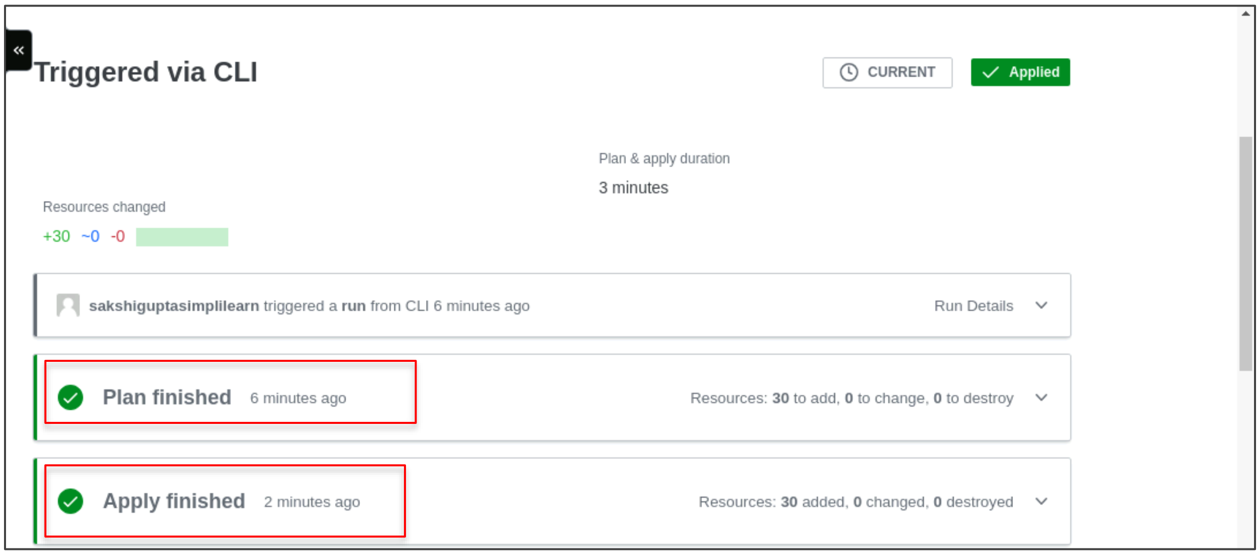
Triggered via CLI CURRENT

#run-jGiC1V41yFLKVkxW | sakshiguptasimplilearn triggered via CLI

Applied

2 minutes ago

4.10 Click on **Plan finished** and **Apply finished** to view the command line output on Terraform Cloud via the remote backend



Triggered via CLI

CURRENT Applied

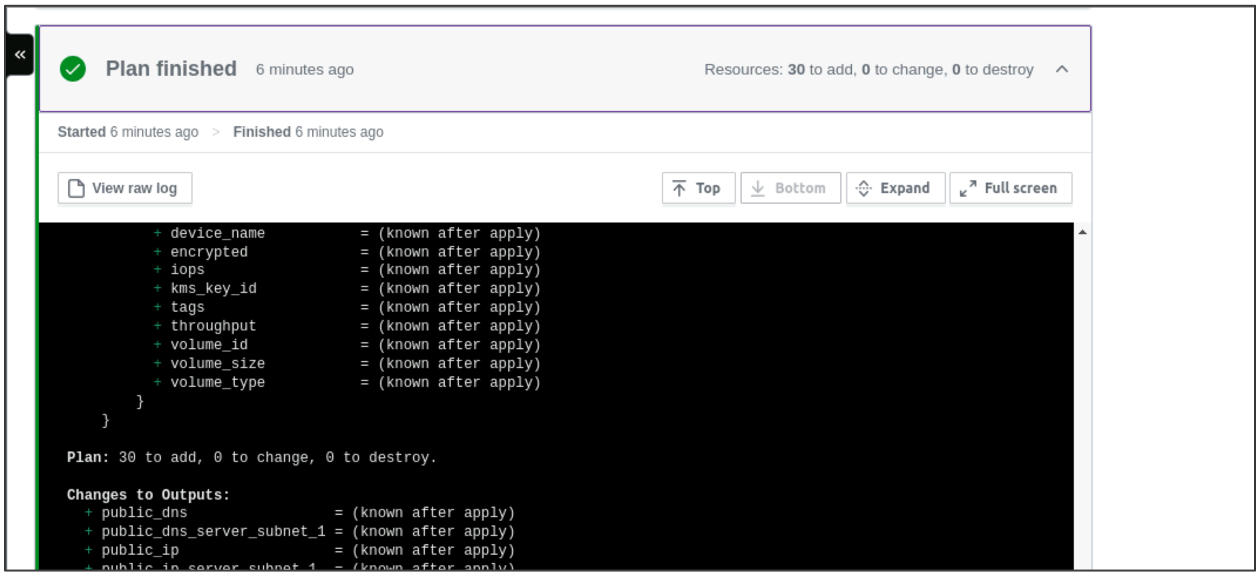
Plan & apply duration
3 minutes

Resources changed
+30 ~0 -0

sakshiguptasimplilearn 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

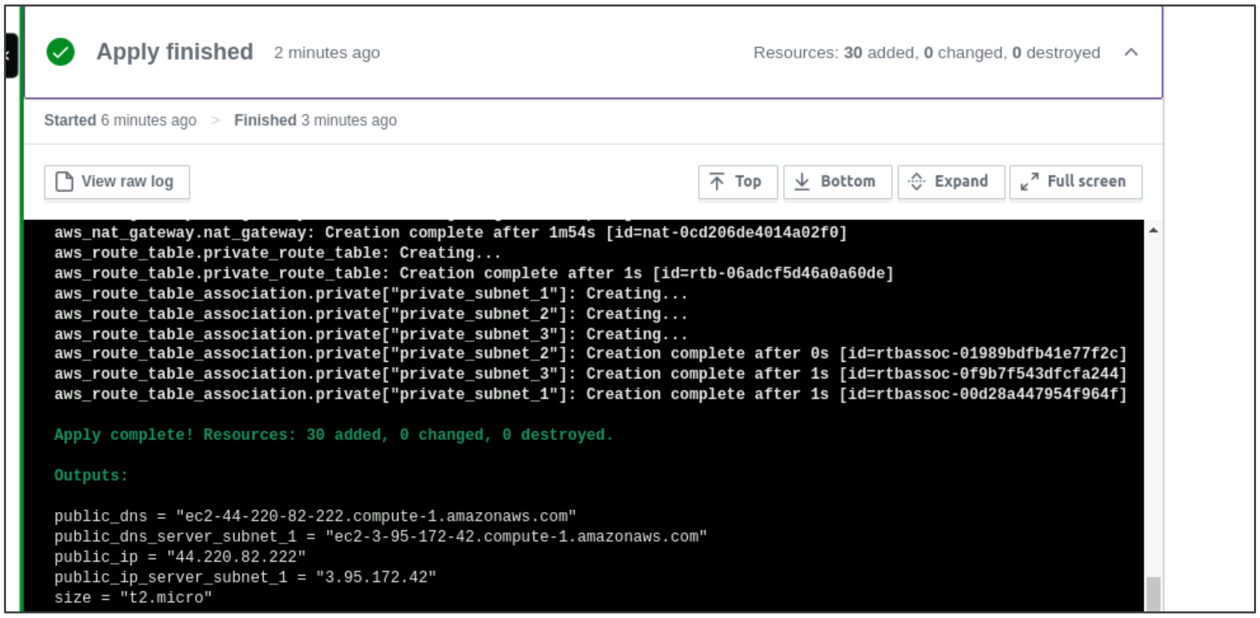


Plan finished 6 minutes ago Resources: 30 to add, 0 to change, 0 to destroy

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)
```


Apply finished
2 minutes ago
Resources: 30 added, 0 changed, 0 destroyed
Started 6 minutes ago > Finished 3 minutes ago
View raw log
Top
Bottom
Expand
Full screen

```
aws_nat_gateway.nat_gateway: Creation complete after 1m54s [id=nat-0cd206de4014a02f0]
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"
```

By following these steps, you have successfully performed Terraform state management using different backends for storing and managing the state file securely and efficiently.