

# terraform

## terraform plan

The `plan` will check the configuration files (basically all the `*.tf` files in the directory) and will show you the items or changes going to be made on target infrastructure or resources. Please note, this command will not actually perform the planned actions.

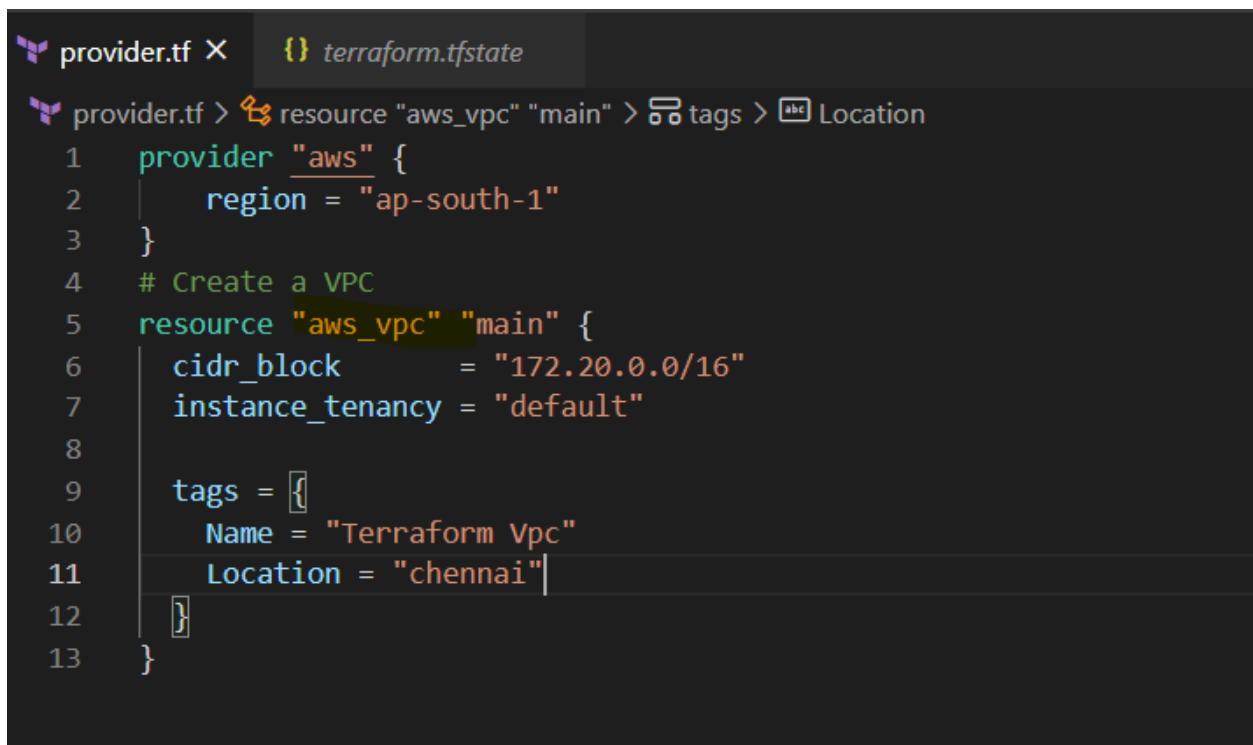
terraform plan

## terraform destroy

Warning: `destroy` will delete all resource but with confirmation

Destroying Resources in terraform

1. To delete all resources
  - a. terraform destroy
2. To delete a specific resource
  - a. Comment that resource code in terraform script. And run terraform apply
  - b. terraform destroy -target=resource



```
provider.tf X terraform.tfstate
provider.tf > resource "aws_vpc" "main" > tags > Location
1  provider "aws" {
2      region = "ap-south-1"
3  }
4  # Create a VPC
5  resource "aws_vpc" "main" {
6      cidr_block      = "172.20.0.0/16"
7      instance_tenancy = "default"
8
9      tags = {
10         Name = "Terraform Vpc"
11         Location = "chennai"
12     }
13 }
```

	Name	VPC ID	State	IPv4 CIDR
<input type="checkbox"/>	-	vpc-b95399d2	Available	172.31.0.0/
<input checked="" type="checkbox"/>	Terraform Vpc	vpc-089263ab2916f93c0	Available	172.20.0.0/

---

### Tags

Key	Value
Location	chennai
Name	Terraform Vpc

```

C:\Users\Admin\Desktop\iac>terraform destroy -target=aws_vpc.main
[0m[1maws_vpc.main: Refreshing state... [id=vpc-089263ab2916f93c0][0m

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
[31m-[0m destroy
[0m
Terraform will perform the following actions:

[1m # aws_vpc.main[0m will be [1m[31mdestroyed[0m[0m
[0m [31m-[0m[0m resource "aws_vpc" "main" {
  [31m-[0m [0m[0m[1m[0marn[0m[0m = "arn:aws:ec2:ap-south-1:615086145317:vpc/vpc-089263ab2916f93c0" [90m->[0m [0m[0m[4
  [31m-[0m [0m[0m[1m[0massign_generated_ipv6_cidr_block[0m[0m = false [90m->[0m [0m[0m[90mnull[0m[0m[0m
  [31m-[0m [0m[0m[1m[0mcidr_block[0m[0m = "172.20.0.0/16" [90m->[0m [0m[0m[90mnull[0m[0m[0m
  [31m-[0m [0m[0m[1m[0mdefault_network_acl_id[0m[0m = "acl-091212d66b722abf0" [90m->[0m [0m[0m[90mnull[0m[0m[0m
  [31m-[0m [0m[0m[1m[0mdefault_route_table_id[0m[0m = "rtb-0dbaa50c6ad356273" [90m->[0m [0m[0m[90mnull[0m[0m[0m
  [31m-[0m [0m[0m[1m[0mdefault_security_group_id[0m[0m = "sg-00d8771ac440a5d44" [90m->[0m [0m[0m[90mnull[0m[0m[0m
  [31m-[0m [0m[0m[1m[0mdhcp_options_id[0m[0m = "dopt-3e289955" [90m->[0m [0m[0m[90mnull[0m[0m[0m
  [31m-[0m [0m[0m[1m[0menable_dns_hostnames[0m[0m = false [90m->[0m [0m[0m[90mnull[0m[0m[0m
  [31m-[0m [0m[0m[1m[0menable_dns_support[0m[0m = true [90m->[0m [0m[0m[90mnull[0m[0m[0m
  [31m-[0m [0m[0m[1m[0mid[0m[0m = "vpc-089263ab2916f93c0" [90m->[0m [0m[0m[90mnull[0m[0m[0m
  [31m-[0m [0m[0m[1m[0minstance_tenancy[0m[0m = "default" [90m->[0m [0m[0m[90mnull[0m[0m[0m
  [31m-[0m [0m[0m[1m[0main_route_table_id[0m[0m = "rtb-0dbaa50c6ad356273" [90m->[0m [0m[0m[90mnull[0m[0m[0m
  [31m-[0m [0m[0m[1m[0mowner_id[0m[0m = "615086145317" [90m->[0m [0m[0m[90mnull[0m[0m[0m
  [31m-[0m [0m[0m[1m[0mtags[0m[0m = {
    [31m-[0m [0m[0m"Location" = "chennai"
    [31m-[0m [0m[0m"Name" = "Terraform Vpc"
  } [90m->[0m [0m[0m[90mnull[0m[0m[0m
  [31m-[0m [0m[0m[1m[0mtags_all[0m[0m = {
    [31m-[0m [0m[0m"Location" = "chennai"
    [31m-[0m [0m[0m"Name" = "Terraform Vpc"
  } [90m->[0m [0m[0m[90mnull[0m[0m[0m
}

```

```

}
[0m][1mPlan:[0m 0 to add, 0 to change, 1 to destroy.
[0m][3m][0m [0m][1m[3mWarning: [0m][0m][1mResource targeting is in effect[0m
[3m][0m [0m
[3m][0m [0m[0mYou are creating a plan with the -target option, which means that the result of this plan may not represent all of the changes requested by the current configuration.
[3m][0m [0m
[3m][0m [0m[0mThe -target option is not for routine use, and is provided only for exceptional situations such as recovering from errors or mistakes, or when Terraform specifically suggests to use it as part of an error
[3m][0m [0m[0mmessage.
[3m][0m [0m[0m
[0m][1m
Do you really want to destroy all resources?[0m
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

[1mEnter a value:[0m [0m[0myes
[0m][1maws_vpc.main: Destroying... [id=vpc-089263ab2916f93c0][0m[0m
[0m][1maws_vpc.main: Destruction complete after 0s[0m
[3m][0m [0m[0m
[3m][0m [0m [0m[1m[3mWarning: [0m][0m][1mApplied changes may be incomplete[0m
[3m][0m [0m [0m
[3m][0m [0m [0m[0mThe plan was created with the -target option in effect, so some changes requested in the configuration may have been ignored and the output values may not be fully updated. Run the following command to
[3m][0m [0m [0m[0mverify that no other changes are pending:
[3m][0m [0m [0m terraform plan
[3m][0m [0m [0m
[3m][0m [0m [0m[0mnote that the -target option is not suitable for routine use, and is provided only for exceptional situations such as recovering from errors or mistakes, or when Terraform specifically suggests to use it as
[3m][0m [0m [0m[0mpart of an error message.
[3m][0m [0m [0m
[0m][1m[3m
Destroy complete! Resources: 1 destroyed.
[0m
C:\Users\Admin\Desktop\iac>

```

Your VPCs (1/1) [Info](#) 🔄 Actions ▼ Create

🔍 Filter VPCs 1

<input checked="" type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
<input checked="" type="checkbox"/>	- [REDACTED]	vpc-b95399d2	🟢 Available	172.31.0.0/16	-

## Terraform Auto Approve

- In automation we do not use prompts
  - But terraform by default prompts for your confirmations, if you want to auto confirm use following command
- terraform apply --auto-approve

```
C:\Users\Admin\Desktop\iac>terraform apply --auto-approve

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_vpc.main will be created
resource "aws_vpc" "main" {
  enable_classiclink           = (known after apply)
  enable_classiclink_dns_support = (known after apply)
  enable_dns_hostnames         = (known after apply)
  enable_dns_support           = true
  instance_tenancy              = "default"
  ipv6_address_assignment_enabled = (known after apply)
  ipv6_cidr_block                = (known after apply)
  main_route_table_id           = (known after apply)
  owner_id                      = (known after apply)
  tags                          = {
    "Location" = "chennai"
    "Name"      = "Terraform Vpc"
  }
}

Plan: 1 to add, 0 to change, 0 to destroy.
aws_vpc.main: Creating...
aws_vpc.main: Creation complete after 2s [id=vpc-09d59fe18330e41a0]
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
C:\Users\Admin\Desktop\iac>
```

## Terraform State File

In a multi developer environment we need to keep the state file in a central location where all developers point to the same file.

There are several options to maintain state files in remote locations.

The best and most common option is S3.

```
C:\Users\Admin\Desktop\iac>terraform state list
aws_vpc.main
```

```
Subcommands:
  list      List resources in the state
  mv        Move an item in the state
  pull      Pull current state and output to stdout
  push      Update remote state from a local state file
  replace-provider Replace provider in the state
  rm        Remove instances from the state
  show      Show a resource in the state
```

```
C:\Users\Admin\Desktop\iac>terraform state show aws_vpc.main
# aws_vpc.main:
resource "aws_vpc" "main" {
  arn = "arn:aws:ec2:ap-south-1:615086145317:vpc/vpc-09d59fe18330e41a0"
  assign_generated_ipv6_cidr_block = false
  cidr_block = "172.20.0.0/16"
  default_network_acl_id = "acl-0b1cd9a2f12128d80"
  default_route_table_id = "rtb-0ef952c5f47fad214"
  default_security_group_id = "sg-0fa580eb3679a48c2"
  dhcp_options_id = "dopt-3e289955"
  enable_dns_hostnames = false
  enable_dns_support = true
  id = "vpc-09d59fe18330e41a0"
  instance_tenancy = "default"
  main_route_table_id = "rtb-0ef952c5f47fad214"
  owner_id = "615086145317"
  tags = {
    "Location" = "chennai"
    "Name" = "Terraform Vpc"
  }
  tags_all = {
    "Location" = "chennai"
    "Name" = "Terraform Vpc"
  }
}
```

## Configure Remote state file using S3

- Create S3 bucket for storing statefile, this cannot be created using the same terraform project.
- Configure remote state in terraform

► **Account snapshot**

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

**Buckets (1)** [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

[Refresh](#) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

	Name ▲	AWS Region ▼	Access ▼	Creation date ▼
<input type="radio"/>	tfstatefiletest	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	November 28, 2021, 11:32:14 (UTC+05:30)

Ref:

<https://www.terraform.io/docs/language/settings/backends/s3.html>

provider.tf > terraform > backend "s3" > key

```
1  provider "aws" {
2    region = "ap-south-1"
3  }
4  # Create a VPC
5  resource "aws_vpc" "main" {
6    cidr_block      = "172.20.0.0/16"
7    instance_tenancy = "default"
8
9    tags = {
10      Name = "Terraform Vpc"
11      Location = "chennai"
12    }
13  }
14
15  terraform {
16    backend "s3" {
17      bucket = "tfstatefiletest"
18      key    = "terraform.tfstate"
19      region = "ap-south-1"
20    }
21  }
```

```
C:\Users\Admin\Desktop\iac>terraform init

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

Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v3.67.0

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.

C:\Users\Admin\Desktop\iac>
```

## tfstatefiletest [Info](#)

[Objects](#)
[Properties](#)
[Permissions](#)
[Metrics](#)
[Management](#)
[Access Points](#)

### Objects (1)

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](#)

☒ Show versions
 

< 1 >

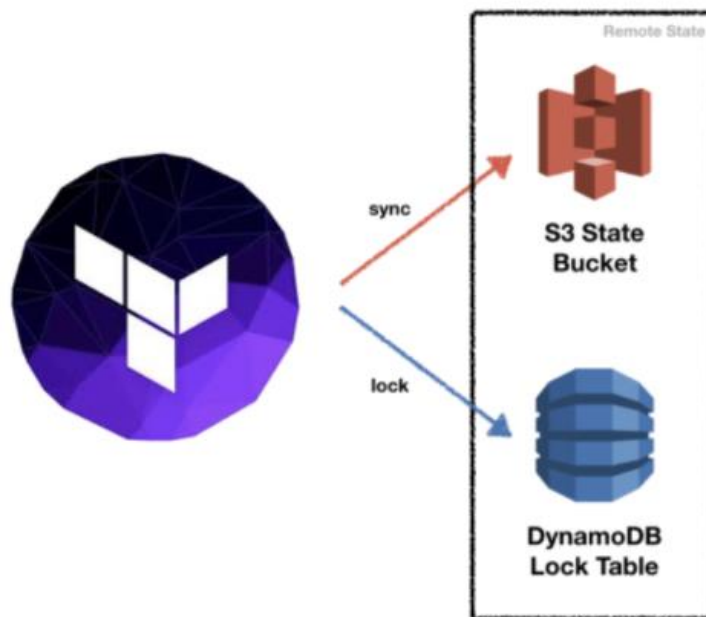
<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	terraform.tfstate	tfstate	November 28, 2021, 11:43:43 (UTC+05:30)	1.6 KB	Standard

How do you lock a state?

## DynamoDB State Locking

The following configuration is optional:

- `dynamodb_endpoint` - (Optional) Custom endpoint for the AWS DynamoDB API. This can also be sourced from the `AWS_DYNAMODB_ENDPOINT` environment variable.
- `dynamodb_table` - (Optional) Name of DynamoDB Table to use for state locking and consistency. The table must have a primary key named `LockID` with type of `string`. If not configured, state locking will be disabled.



DynamoDB

×

Dashboard

Tables

Items New

PartiQL editor New

Backups

Exports to S3

Reserved capacity

▼ DAX

Clusters

Subnet groups

The new DynamoDB console is now complete, and becomes your default experience

Following the preview phase in which we analyzed and incorporated your feedback, we have completed the new DynamoDB console, making it even easier for you to manage data and resources. Let us know what you think. You can still choose to return to the previous console from the navigation pane.

Database

# 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.

Get started

Create a new table to start exploring DynamoDB.

Create table

Pricing



## Table name

This will be used to identify your table.

terraform-lock

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 hosts for scalability and availability.

LockID

String

1 to 255 characters and case sensitive.

DynamoDB > Tables

Tables (1) Info



Actions ▼

Delete

Create table

Find tables by table name

Any table tag

< 1 > ⚙

<input type="checkbox"/>	Name ▲	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity mode
<input type="checkbox"/>	terraform-lock	Active	LockID (String)	-	0	Provisioned with auto scaling (5)	Provisioned with auto scaling (5)

```
C:\Users\Admin\Desktop\iac>terraform init
```

```
Initializing the backend...
```

```
Initializing provider plugins...
```

- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v3.67.0

```
Terraform has been successfully initialized!
```

```
C:\Users\Admin\Desktop\iac>terraform apply --auto-approve
```

```
Acquiring state lock. This may take a few moments...
```

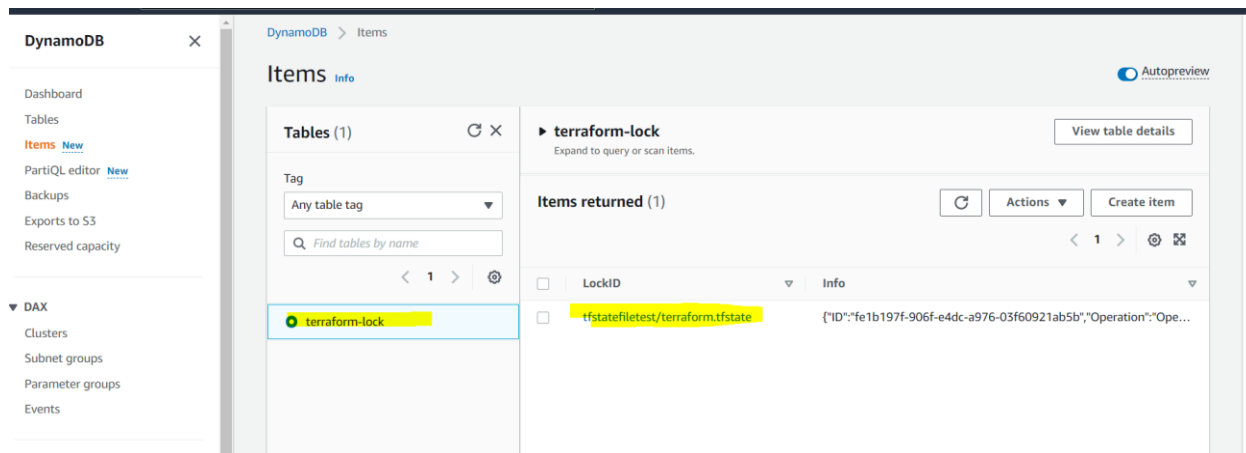
```
maws_vpc.main: Refreshing state... [id=vpc-09d59fe18330e41a0]
```

```
No changes. Your infrastructure matches the configuration.
```

```
Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.  
Releasing state lock. This may take a few moments...
```

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

```
C:\Users\Admin\Desktop\iac>
```



Ref:

<https://quileswest.medium.com/how-to-lock-terraform-state-with-s3-bucket-in-dynamodb-3ba7c4e637>