



Riyaz Sayyad
AWS Cloud Expert

15 VITAL TERRAFORM COMMANDS FOR EVERYDAY USE

01

WHAT IS TERRAFORM?

Terraform is a popular open-source tool created by HashiCorp that helps manage and automate infrastructure as code (IaC). It allows users to define and deploy resources like servers, storage, and networks in a simple, human-readable configuration file.

02

COMMON USE CASES OF TERRAFORM

- Manage multi-cloud resources with ease.
- Automate and simplify infrastructure tasks.
- Standardize environments across dev, test, and prod.
- Enable disaster recovery with reusable configurations.
- Optimize costs with dynamic scaling.
- Collaborate effectively using shared templates.

03

terraform init

Purpose: Prepares a directory containing Terraform configuration files by initializing required backends and downloading provider plugins.

Example:



A terminal window with a dark background and a light gray header bar. The header bar has three small colored dots (red, yellow, green) on the left. The main terminal area contains the text "terraform init" in white.

Use Case: Run this command first in any new project or after adding a new provider.

Riyaz Sayyad
AWS Cloud Expert

04

terraform plan

Purpose: Creates an execution plan, showing what actions Terraform will take without actually performing them.

Example:



```
terraform plan -out=plan.tfplan
```

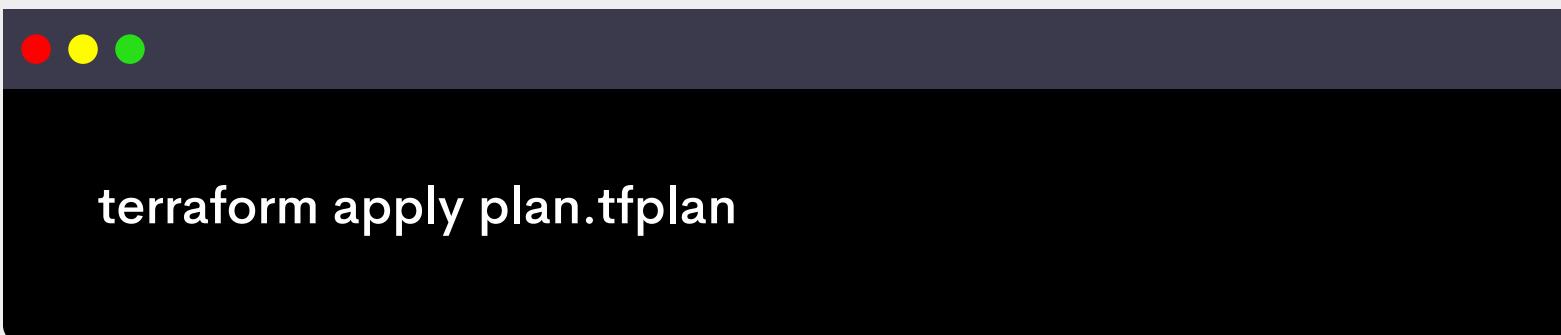
Use Case: Useful for reviewing changes before applying them to ensure accuracy.

05

terraform apply

Purpose: Executes the changes required to reach the desired state defined in the configuration files.

Example:



A screenshot of a terminal window. The title bar shows three colored dots (red, yellow, green). The main window displays the text "terraform apply plan.tfplan".

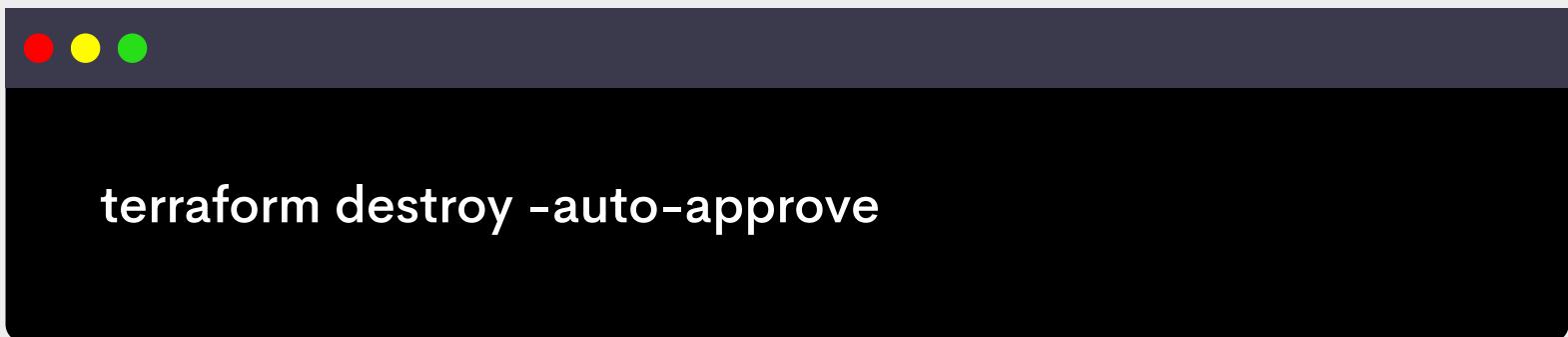
Use Case: Deploy or update infrastructure based on the reviewed execution plan.

06

terraform destroy

Purpose: Removes all resources defined in the current Terraform configuration.

Example:



```
● ● ●
terraform destroy -auto-approve
```

Use Case: Decommission environments when they're no longer needed, saving costs.

07

terraform fmt

Purpose: Formats Terraform configuration files for consistency and readability.

Example:



Use Case: Standardize code formatting for better collaboration and review.

Riyaz Sayyad
AWS Cloud Expert

08

terraform graph

Purpose: Generates a graph of the dependency relationships between resources.

Example:



```
terraform graph | dot -Tpng > graph.png
```

Use Case: Visualize dependencies for better understanding and debugging.

09

terraform refresh

Purpose: Updates the Terraform state file to match the actual infrastructure.

Example:



Use Case: Sync the state file after manual changes or external updates.

Riyaz Sayyad
AWS Cloud Expert

10

terraform import

Purpose: Imports existing infrastructure into Terraform management.

Example:

```
terraform import aws_instance.my_instance i-  
0abcd1234efgh5678
```

Use Case: Bring manually created resources under Terraform management without recreating them.

terraform untaint

Purpose: Removes the taint from a resource, ensuring it won't be recreated.

Example:

```
terraform untaint aws_instance.my_instance
```

Use Case: Undo a taint operation if recreation is no longer needed.

terraform taint

Purpose: Marks a resource for recreation during the next terraform apply.

Example:

```
terraform taint aws_instance.my_instance
```

Use Case: Force recreation of a resource to fix a specific issue or apply updates.



terraform state list

Purpose: Lists all resources tracked by Terraform in the current state file.

Example:



```
terraform taint aws_instance.my_instance
```

Use Case: Identify managed resources for troubleshooting or refactoring.

terraform output

Purpose: Extracts output variables from the Terraform state.

Example:

```
● ● ●  
terraform output instance_ip
```

Use Case: Retrieve critical information (e.g., IP addresses) for integration with other systems.

Riyaz Sayyad
AWS Cloud Expert

15

terraform validate

Purpose: Verifies if the configuration is syntactically valid and internally consistent.

Example:

```
● ● ●
terraform validate
```

Use Case: Run this before terraform plan to catch errors early.

terraform show

Purpose: Displays the contents of a Terraform state or plan file in a human-readable format.

Example:



```
terraform show plan.tfplan
```

Use Case: Understand what resources exist in the current state or what changes will occur.

17

terraform workspace

Purpose: Manage multiple workspaces for environment isolation (e.g., dev, staging, production).

Example:

Create a new workspace:



```
terraform graph | dot -Tpng > graph.png
```

Switch to an existing workspace:



```
terraform workspace select staging
```

Use Case: Separate configurations and states for different environments.



Was it helpful?

follow me for more Cloud related content



Like



Comment



Share



Save

For ACMP Check out: <https://needforcloud.com>