

TERRAFORM COMMANDS

CHEATSHEET

(A COMPLETE QUICK-REFERENCE GUIDE)



1. Terraform – Basics

Core commands to understand Terraform installation and usage.

- `terraform version` – Display installed Terraform version
- `terraform help` – Show general help
- `terraform -help` – List all available Terraform commands

2. Initialization

Prepare the working directory and backend.

- `terraform init` – Initialize Terraform working directory
- `terraform init -upgrade` – Upgrade providers & modules
- `terraform init -reconfigure` – Reconfigure backend settings
- `terraform init -backend=false` – Initialize without backend
- `terraform init -migrate-state` – Migrate state to a new backend

3. Validation & Formatting

Ensure configuration quality and consistency.

- `terraform validate` – Validate configuration files
- `terraform validate -json` – Validation output in JSON
- `terraform fmt` – Format Terraform files
- `terraform fmt -recursive` – Format files recursively
- `terraform fmt -check` – Check formatting without modifying
- `terraform fmt -diff` – Show formatting differences

4. Planning

Preview infrastructure changes before applying.

- `terraform plan` – Generate execution plan
- `terraform plan -out=tfplan` – Save plan to file
- `terraform plan -var="key=value"` – Pass variables at runtime
- `terraform plan -var-file=vars.tfvars` – Use variable file
- `terraform plan -target=resource.name` – Plan specific resource

- `terraform plan -destroy` – Preview destroy plan
- `terraform plan -refresh=false` – Skip state refresh

5. Apply

Provision or update infrastructure.

- `terraform apply` – Apply infrastructure changes
- `terraform apply tfplan` – Apply saved plan
- `terraform apply -auto-approve` – Skip confirmation
- `terraform apply -var-file=vars.tfvars` – Apply with variables
- `terraform apply -target=resource.name` – Apply specific resource

6. Destroy

Remove managed infrastructure safely.

- `terraform destroy` – Destroy all resources
- `terraform destroy -auto-approve` – Destroy without confirmation
- `terraform destroy -target=resource.name` – Destroy specific resource

7. State Management

Manage and troubleshoot Terraform state.

- `terraform state list` – List resources in state
- `terraform state show <resource>` – Show resource details
- `terraform state pull` – Download state from backend
- `terraform state push` – Upload state to backend
- `terraform state mv <old> <new>` – Move resource in state
- `terraform state rm <resource>` – Remove resource from state
- `terraform state replace-provider <old> <new>` – Replace provider

8. Outputs

Access output values from Terraform.

- `terraform output` – Show all outputs

- `terraform output <name>` – Show specific output
- `terraform output -json` – Output in JSON format
- `terraform output -raw <name>` – Display raw value

9. Variables & Console

Debug and test expressions interactively.

- `terraform console` – Open interactive Terraform console

10. Workspaces

Manage multiple environments (dev, stage, prod).

- `terraform workspace list` – List workspaces
- `terraform workspace show` – Show current workspace
- `terraform workspace new <name>` – Create workspace
- `terraform workspace select <name>` – Switch workspace
- `terraform workspace delete <name>` – Delete workspace

11. Providers & Modules

Handle providers and reusable modules.

- `terraform providers` – List required providers
- `terraform providers mirror <dir>` – Mirror providers locally
- `terraform get` – Download modules
- `terraform get -update` – Update modules

12. Import

Bring existing infrastructure under Terraform management.

- `terraform import <resource> <id>` – Import existing resource

13. Refresh & Show

Sync and inspect Terraform state.

- `terraform refresh` – Update state from real infrastructure

- `terraform show` – Display state or plan details
- `terraform show -json` – Output details in JSON

14. Graph

Visualize resource dependencies.

- `terraform graph` – Generate dependency graph

15. Debugging & Logs

Troubleshoot Terraform operations.

- `TF_LOG=TRACE terraform apply` – Enable trace-level logs
- `TF_LOG=DEBUG terraform plan` – Enable debug logs
- `TF_LOG_PATH=terraform.log terraform apply` – Save logs to file

16. Lock & Recovery

Handle state lock issues.

- `terraform force-unlock <LOCK_ID>` – Force unlock state

17. Advanced / Rare Commands

Used in special scenarios.

- `terraform taint <resource>` – Mark resource for recreation
- `terraform untaint <resource>` – Remove taint
- `terraform test` – Run Terraform tests (*new feature*)

18. Cleanup (Manual)

Local cleanup operations.

- `rm -rf .terraform` – Remove Terraform cache
- `rm terraform.tfstate` – Delete local state file

19. Terraform Workflow (One-Line Summary)

`init → validate → plan → apply → state → destroy`