OpenTofu Open-source infrastructure as code

CONCEPTS

OpenTofu is the open-source fork of Terraform: same HCL and workflow, command is tofu. Configuration and state stay compatible for typical setups.

Config .tf files
↓ `init`
Working dir providers, modules
↓ `plan`
Plan diff vs. state
↓ `apply`
Infrastructure state updated

Always read the plan before applying. Saving it with -out guarantees apply does exactly what you reviewed.

State may hold secrets. Keep it out of git, use a remote backend, and consider OpenTofu's built-in state encryption.

WORKFLOW

`tofu init`
Prepare the working directory

`tofu init -upgrade`
Update providers and modules

`tofu validate`
Check configuration syntax

`tofu plan`
Preview changes

`tofu plan -out=<file>`
Save the plan to a file

`tofu apply`
Create or update infrastructure

`tofu apply <planfile>`
Apply a saved plan

`tofu apply -auto-approve`
Skip the confirmation prompt

`tofu destroy`
Remove all managed resources

`tofu plan -destroy`
Preview a destroy

`tofu plan -refresh-only`
Preview drift against reality

WORKFLOW (cont.)

`tofu fmt`
Format files in place

`tofu fmt -recursive -check`
Check formatting in CI

TARGETING & PLANS

`tofu plan -target=<addr>`
Limit to one resource

`tofu apply -replace=<addr>`
Force recreating a resource

`tofu show <planfile>`
Read a saved plan

VARIABLES & OUTPUTS

`tofu plan -var '<name>=<value>'`
Set one variable

`tofu plan -var-file=<file>`
Load variables from a file

`TF_VAR_<name>=<value>`
Set a variable via environment

`terraform.tfvars`
Auto-loaded variable file

`tofu output`
Show all outputs

`tofu output <name>`
Show one output

`tofu output -raw <name>`
Plain value for scripts

`tofu output -json`
All outputs as JSON

STATE

`tofu state list`
List tracked resources

`tofu state show <addr>`
Show one resource's attributes

`tofu import <addr> <id>`
Adopt an existing resource

TESTING & MODULES

`tofu test`
Run *.tftest.hcl test files

`tofu version`
Show the OpenTofu version