# Terraform

- init
- plan, terraform plan - - refresh=false
- apply, can be used with 'refresh' → terraform apply -refresh-only
- destroy
- show, can be used with a -json flag → `terraform show -json`
- output usage (to print output vars) → terraform output or terraform output {outputVar_name}
- validate — checks if the configuration file is valid or not
- fmt
- providers, can be used with 'mirror' → `terraform providers mirror {new_path}`
- graph, usage → `terraform graph | dot -Tsvg > graph.svg`
- taint
- untaint
- import - terraform import <rsrc type>.<resrc name> <attr>
- get - gets module from terraform registry
- console

## Terraform Input Variables

- Variables can be stored in variables.tf and used in main.tf
- They can also be passed interactively → terraform apply -var="image_id=ami-abc123"
- Can be declared as env vars → export TF_VAR_${var_name}="value"
- Can be passed via CLI
- Variable definition files - terraform.tfvars or terraform.tfvars.json or `*.auto.tfvars` or `*.auto.tfvars.json` - Same syntax as HCL file

- terraform apply -var-file ${randomName}.tfvars
- Var precedence → env vars < terraform.tvars < *.auto.tfvars (alphabetical order) < cli flag

## Output Variables

https://developer.hashicorp.com/terraform/language/values/outputs

## Resource attributes

syntax → ${`resource_type.resource_name.attribute`}

## Resource dependencies

Explicit dependency → `depends_on = [resourceType.resourceName]`

## Lifecycle rules

lifecycle block →

## Datasources

## Meta-Arguments

- depends_on - Specify explicit dependencies
- lifecycle - Control resource lifecycle behavior
- count - Create multiple resource instances → '*`length`*' function is useful
- for_each → works with only sets or maps → '`toset`' function is useful

## Terraform state

terraform state <subcommand> [options] [args]

terraform state list

terraform state show - detailed info

terraform state mv [options] src dest

terraform state pull [options] src dest

terraform state rm address

## Terraform Provisioners

remote-exec

local-exec

destroy time provisioner

on_failure = fail/continue

## Debugging

export TF_LOG = <log-lvl> (TRACE)

export TF_LOG_PATH = <file-path>

## Modules

## Functions

**Numeric** - expansion, example → max(var.num...)

examples - min, max, ceil, floor

**String -** split(”,”var.ami),lower(var.ami), upper(var.ami), title, substr, join

**Collection** - length, index, element, contains, keys, values, lookup(var.ami,key,default value)

## Operators

condition ? true_val : false_val

## Workspaces

- workspace new

- workspace list
- terraform.worspace
- terraform workspace select <ws name>
- state files stored in - `terraform.tfstate.d` directory