Terragrunt Terraform orchestration

CONCEPTS

Terragrunt is a wrapper around OpenTofu/Terraform. A unit is a directory with a terragrunt.hcl; a stack is a tree of units run together. Unknown commands such as plan pass through to tofu/terraform.

Discover find units in tree
↓ filter
Order DAG from dependencies
↓ per unit
Init + run auto-init, then tofu cmd
↓ state
Result summary of all units

Use -- to separate Terragrunt flags from tofu/terraform arguments: terragrunt run -- plan -no-color. The old run-all and hclfmt forms are deprecated in favour of run --all and hcl fmt.

--all adds -auto-approve to apply and destroy automatically. Add --no-auto-approve if you want the prompt back.

RUN

terragrunt run plan
Plan the unit in this directory

terragrunt run apply
Apply the unit in this directory

terragrunt run init
Initialise the unit

terragrunt run destroy
Destroy the unit

terragrunt run output
Show unit outputs

terragrunt run --all plan
Plan all units below here

terragrunt run --all apply
Apply all units in order

SELECT UNITS

--filter '<*glob*>'
Only units matching the path

--filter '!<*glob*>'
Exclude matching units

CONFIG BLOCKS

Block purpose

Block	Use
`include`	Inherit a parent file
`inputs`	Variables passed to the module
`dependency`	Outputs of another unit
`dependencies`	Ordering without outputs
`remote_state`	Backend setup
`generate`	Write helper files
`locals`	Named values in the file
`feature`	Flags set with --feature

mock_outputs let plan work before the dependency has been applied. Without them, plan on a fresh stack fails on missing outputs.

FUNCTIONS

find_in_parent_folders()
Nearest parent terragrunt file

path_relative_to_include()
Unit path relative to parent

DISCOVER

terragrunt find
List units and stacks

FORMAT AND VALIDATE

terragrunt hcl fmt
Format HCL files in place

STACKS AND BACKEND

terragrunt stack generate
Create units from stack.hcl

terragrunt backend bootstrap
Create the state backend