Terraform Cheat Sheet

Installation

See https://developer.hashicorp.com/terraform/downloads for installation instructions for your platform. Then run `terraform -install-autocomplete` to enable shell autocompletion.

Initializing Terraform

Command	Description
`terraform init`	Prepare your working directory for other commands
`terraform init -upgrade`	Upgrade modules/providers to the latest allowed versions
`terraform get`	Only download and install modules

Making infrastructure changes

Command	Description
`terraform plan`	Show changes required by the current configuration
`terraform plan -out=<file>`	Write the plan to a file to apply it later
`terraform plan -target <resource>`	Create a plan for a specific module or resource
`terraform plan -replace <resource>`	Force the plan to replace a specific resource
`terraform plan -var '<key>=<value>'`	Set a value for one of the input variables
`terraform plan -refresh-only`	Inspect resource drift without updating the state file
`terraform apply`	Create or update infrastructure
`terraform apply <file>`	Create or update infrastructure using a plan file
`terraform apply -target <resource>`	Create or update a specific resource
`terraform apply -replace <resource>`	Force the replacement of a specific resource
`terraform apply -auto-approve`	Skip interactive approval of plan before applying

Inspecting output values

Command	Description
`terraform output`	Show all output values
`terraform output -json`	Show all output values in JSON format
`terraform output <name>`	Show a specific output value
`terraform output -raw <name>`	Show a specific output value without quotes

Managing state

Command	Description
`terraform show`	Show the current state in human-readable form
`terraform show <file>`	Show a human-readable state or plan file
`terraform show -json <file>`	Show a state or plan file in JSON format
`terraform state list`	List all resources in the state file
`terraform state show <resource>`	Show details about a resource
`terraform state mv <source> <dest>`	Rename a resource in the state file
`terraform state rm <resource>`	Remove a resource from the state file
`terraform state replace-provider <from> <to>`	Replace provider for resources in the state
`terraform import <resource> <remote_id>`	Import existing infrastructure into Terraform
`terraform state pull`	Pull current state and output to stdout
`terraform refresh`	(Deprecated) Update state to match reality

Tainting resources

Command	Description
`terraform taint <resource>`	(Deprecated) Mark a resource to be replaced
`terraform untaint <resource>`	Mark a resource as no longer tainted

Destroying infrastructure

Command	Description
`terraform destroy`	Destroy infrastructure managed by Terraform
`terraform destroy -target <resource>`	Destroy a specific resource

Formatting and validation

Command	Description
`terraform validate`	Check whether the configuration is valid
`terraform fmt`	Reformat your configuration in the standard style
`terraform fmt -check`	Check whether the configuration is formatted correctly, return non-zero exit code if not

Terraform Cloud / Remote Authentication

Command	Description
terraform login	Log in to Terraform Cloud
terraform login <hostname>	Log in to a different host
terraform logout	Log out of Terraform Cloud
terraform logout <hostname>	Log out of a different host

Managing workspaces

Command	Description
terraform workspace list	List all existing workspaces
terraform workspace show	Show the name of the current workspace
terraform workspace select <name>	Select a different workspace
terraform workspace new <name>	Create a new workspace
terraform workspace delete <name>	Delete an existing workspace

Other commands

Command	Description
terraform providers	Show the providers required for this configuration
terraform force-unlock <lock-id>	Release a stuck lock
terraform console	Try Terraform expressions at an interactive prompt
terraform graph | dot -Tpng > graph.png	Generate a visual graph of Terraform resources
terraform version	Show the current Terraform version
terraform -help	Show help output for Terraform
terraform -help <command>	Show help output for a specific Terraform command

Referencing Named Values

Syntax	Description
<RESOURCE_TYPE>.<NAME>	Reference to a managed resource
var.<NAME>	Reference to an input variable
local.<NAME>	Reference to a local value
module.<MODULE NAME>	Reference to a child module
data.<DATA TYPE>.<NAME>	Reference to a data source

Conditional Expressions

Syntax	Description
condition ? true : false	If condition is true, return true, otherwise return false

Splat Expressions

Syntax	Description
<RESOURCE_TYPE>.<NAME>[*].<ATTRIBUTE>	Return a list of values for the given attribute of all instances of a resource

Resource Meta-Arguments

Argument	Description
depends_on	Explicitly specify resource dependencies
count	Create multiple instances of a resource
for_each	Create an instance of a resource for each element in a map or set
provider	Specify a provider configuration block to use for this resource
lifecycle	Configure the behavior of a resource over its lifetime

Lifecycle Meta-Argument Attributes

Argument	Description
create_before_destroy	Create the new resource before destroying the old one
prevent_destroy	Prevent Terraform from destroying the resource
ignore_changes	Ignore changes to specific resource attributes