Terraform Cheat Sheet

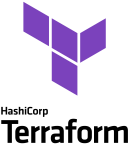
**Installation**

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

**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

### Command Description

terraform init Prepare your working directory for other commands

**Initializing Terraform**

terraform state list List all resources in the state ﬁle

terraform state show <resource> Show details about a resource

terraform init -upgrade Upgrade modules/providers to the latest allowed versions

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> Replace provider for resources in the state

terraform get Only download and install modules

<to>

terraform import <resource> <remote\_id> Import existing infrastructure into Terraform

**Making infrastructure changes**

### Command Description

terraform plan Show changes required by the current conﬁguration

terraform state pull Pull current state and output to stdout

terraform plan -out=<file> Write the plan to a ﬁle to apply it later

**Tainting resources**

terraform refresh (Deprecated) Update state to match reality

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 taint <resource> (Deprecated) Mark a resource to be replaced

### Command Description

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 untaint <resource> Mark a resource as no longer tainted

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

**Destroying infrastructure**

### Command Description

terraform destroy Destroy infrastructure managed by Terraform

terraform destroy -target <resource> Destroy a speciﬁc resource

## 

**Inspecting output values**

**Formatting and validation**

**Command Description**

terraform fmt -check

Check whether the conﬁguration is formatted correctly, return non-zero exit code if not

**Command Description**

## 

terraform output Show all output values

terraform validate Check whether the conﬁguration is valid

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

terraform fmt Reformat your conﬁguration in the standard style

**Command Description**

**Terraform Cloud / Remote Authentication**

# Conditional Expressions

### Syntax Description

condition ? true : false

If condition is true, return true, otherwise return false

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

# Splat Expressions

**Syntax Description**

**Managing workspaces**

<RESOURCE\_TYPE>.<NAME>[\*].<ATTRIBUTE> Return a list of values for the given attribute of all

instances of a resource

### Command Description

terraform workspace list List all existing workspaces

terraform workspace show Show the name of the current workspace

# Resource Meta-Arguments

**Argument Description**

terraform workspace select <name> Select a different workspace

depends\_on Explicitly specify resource dependencies

terraform workspace new <name> Create a new workspace count Create multiple instances of a resource

## 

terraform workspace delete <name> Delete an existing workspace

for\_each

Create an instance of a resource for each element in a map or set

### Command Description

**Other commands**

provider Specify a provider conﬁguration block to use for this resource

terraform providers Show the providers required for this conﬁguration

terraform force-unlock <lock-id> Release a stuck lock

lifecycle

Conﬁgure the behavior of a resource over its lifetime

terraform console Try Terraform expressions at an interactive prompt

terraform graph | dot -Tpng > graph.png Generate a visual graph of Terraform resources

# Lifecycle Meta-Argument Attributes

### Argument Description

create\_before\_destroy

Create the new resource before destroying the old one

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

prevent\_destroy Prevent Terraform from destroying the resource

ignore\_changes Ignore changes to speciﬁc resource attributes

**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