

[Skip to main content](#)


# 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                                       |
|--------------------------------------|---|
| <code>terraform init</code>          | Prepare your working directory for other commands |
| <code>terraform init -upgrade</code> | Upgrade all modules to their latest versions      |
| <code>terraform get</code>           | Only download and install modules                 |

## Making infrastructure changes

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

## Inspecting output values

| Command   | Description                                 |
|---|---|
| <code>terraform output</code>                   | Show all output values                      |
| <code>terraform output -json</code>             | Show all output values in JSON format       |
| <code>terraform output &lt;name&gt;</code>      | Show a specific output value                |
| <code>terraform output -raw &lt;name&gt;</code> | Show a specific output value without quotes |

## Managing state

| Command   | Description                                   |
|---|---|
| <code>terraform refresh</code>  | Update the state to match remote systems      |
| <code>terraform show</code>   | Show the current state                        |
| <code>terraform show &lt;file&gt;</code>                              | Show a saved plan                             |
| <code>terraform show -json &lt;file&gt;</code>                        | Show the plan in JSON format                  |
| <code>terraform state list</code>                                     | List all resources in the state file          |
| <code>terraform state show &lt;resource&gt;</code>                    | Show details about a resource                 |
| <code>terraform state mv &lt;source&gt; &lt;dest&gt;</code>           | Rename a resource in the state file           |
| <code>terraform state rm &lt;resource&gt;</code>                      | Remove a resource from the state file         |
| <code>terraform state replace-provider &lt;from&gt; &lt;to&gt;</code> | Replace provider for resources in the state   |
| <code>terraform import &lt;resource&gt; &lt;remote_id&gt;</code>      | Import existing infrastructure into Terraform |
| <code>terraform state pull</code>                                     | Pull current state and output to stdout       |

## Tainting resources

| Command   | Description                                 |
|---|---|
| <code>terraform taint &lt;resource&gt;</code>   | (Deprecated) Mark a resource to be replaced |
| <code>terraform untaint &lt;resource&gt;</code> | Mark a resource as no longer tainted        |

## Destroying infrastructure

| Command   | Description                                 |
|---|---|
| <code>terraform destroy</code>                          | Destroy infrastructure managed by Terraform |
| <code>terraform destroy -target &lt;resource&gt;</code> | Destroy a specific resource                 |

## Formatting and validation

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

© [Nic Wortel](#), Software Consultant - Updated on Wednesday 16 August 2023 - Find more cheat sheets at <https://nicwortel.nl/cheat-sheets>