

Section	Command	Description
Start	<pre>terraform init terraform get -update=true terraform init -backend-config="address=demo.consul.io" \ -backend-config="path=example_app/terraform_state" \ -backend-config="scheme=https"</pre>	<p>`init` installs the required plugins for referenced providers</p> <p>`get` pulls modules down locally and updates any remote changes</p> <p>Creates a backend in consul to store the terraform state file</p>
Plan	<pre>terraform plan -out plan.out terraform plan -destroy terraform plan -out plan.out -target=RESOURCE_ TYPE.NAME</pre>	<p>`plan` creates a configuration check. It builds the graph of infrastructure that is to be deployed and stores it in plan.out</p> <p>Generates a plan to destroy all the known resources</p> <p>Generates a plan against a specific resource</p>
Apply	<pre>terraform apply plan.out terraform apply plan.out -auto-approve terraform apply -target=RESOURCE_TYPE.NAME terraform apply -target=moduleA.moduleB.RESOURCE _TYPE.NAME</pre>	<p>`apply` calls the provider (AWS/GCP/Azure/...) and creates the infrastructure from the specific plan.out</p> <p>-auto-approve allows TF to be ran in a CICD pipeline</p> <p>Executes changes to only named resource and not the whole infrastructure</p> <p>Executes changes to resources in the specified (nested) module</p>
Destroy	<pre>terraform destroy terraform destroy -target RESOURCE_TYPE.NAME terraform destroy -target=moduleA.moduleB.RE- SOURCE_TYPE.NAME</pre>	<p>`destroy` deletes all the objects managed by Terraform</p> <p>Removes specified resource while leaving the others intact</p> <p>Removes the resource in the specified (nested) module</p>
Variable	<pre>terraform apply -var 'foo=bar' terraform apply -var-file FILE_NAME.tfvars terraform.tfvars export TF_VAR_token=your-token-value</pre>	<p>Populates variable `foo` with the value `bar` during the execution of the plan</p> <p>Populates variables from the file FILE_NAME.tfvars</p> <p>All variables will automatically be populated from terraform.tfvars</p> <p>Terraform will read in all environment variables prefixed with TF_VAR</p>

Section	Command	Description
State	<code>terraform state list</code> <code>terraform state mv</code> <code>terraform state pull > terraform.tfstate</code> <code>terraform state push</code> <code>terraform state rm ADDRESS</code> <code>terraform state show module.aws_security_group.al-low_https</code> <code>terraform refresh</code> <code>terraform apply -state=path_to_file</code> <code>terraform taint aws_instance.foo</code> <code>terraform untaint aws_instance.foo</code> <code>terraform import aws_instance.web i-0e01ec-cb9121f8264</code>	<p>List resources in the state</p> <p>Move an item in the state</p> <p>Create a local state copy</p> <p>Forces an update to remote state from a local change</p> <p>Remove instances from the state</p> <p>Show a resource in the state</p> <p>Updates the state with what is actually provisioned</p> <p>Path to the state file. Defaults to "terraform.tfstate"</p> <p>Marks resource for deletion on next apply</p> <p>Manually unmarks a Terraform-managed resource as tainted</p> <p>Import will bring a resource under TF management and add it to the state file</p>
Drift	<code>terraform show >before</code> <code>terraform refresh</code> <code>terraform show >after</code> <code>diff -u before after</code>	<p>Terraform doesn't have automatic drift detection. Terraform plan will create the graph of what it thinks the infrastructure should be based on the code, refresh updates the state of what actually is deployed and terraform reconciles the plan and the state</p>
Work-space	<code>terraform workspace new NAME</code> <code>terraform workspace select NAME</code> <code>terraform workspace list</code> <code>terraform workspace show</code>	<p>Creates new workspace</p> <p>Switches to the named workspace</p> <p>Lists all available workspaces</p> <p>Shows the current workspace you are executing against</p>
Providers	<code>terraform providers</code>	<p>List all the providers available/being used in this project</p>
Code lint	<code>terraform fmt</code> <code>terraform validate</code>	<p>Fmt formats all the code in the directory to the terraform standard</p> <p>Validate is a syntax check built into Terraform</p>