Terraform

CONTINO

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

Terraform

CONTINO

Section	Command	Description
State	terraform state list terraform state mv terraform state pull > terraform.tfstate terraform state push terraform state rm ADDRESS terraform state show module.aws_security_group.allow_https terraform refresh terraform apply -state=path_to_file terraform taint aws_instance.foo terraform untaint aws_instance.foo terraform import aws_instance.web i-0e01ec-cb9121f8264	List resources in the state Move an item in the state Create a local state copy Forces an update to remote state from a local change Remove instances from the state Show a resource in the state Updates the state with what is actually provisioned Path to the state file. Defaults to "terraform.tfstate" Marks resource for deletion on next apply Manually unmarks a Terraform-managed resource as tainted Import will bring a resource under TF management and add it to the state file
Drift	terraform show >before terraform refresh terraform show >after diff -u before after terraform show -json	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 Machine-readable output from Terraform show
Work-space	terraform workspace new NAME terraform workspace select NAME terraform workspace list terraform workspace show terraform workspace delete NAME	Creates new workspace Switches to the named workspace Lists all available workspaces Shows the current workspace you are executing against Deleted specified workspace
Providers	terraform providers	List all the providers available/being used in this project
Code lint	terraform fmt terraform validate	Fmt formats all the code in the directory to the terraform standard Validate is a syntax check built into Terraform