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"	Initalized repo for Terraform Pulls and updates all the Terraform modules locally Creates a backend in consul to store the terraform state file
Plan	terraform plan -out plan.out 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 generates a plan to destroy all the known resources Generate 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.RE-SOURCE_TYPE.NAME	Apply is the step that will go out to the provider, AWS/GCP, and start creating the infrastructure -auto-approve allows TF to be ran in a CICD pipeline Apply changes to only named resource and not the whole infrastructure
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 TF Destroy specifc Resource while leaving the others intact Destroys resources in a specific module
Variable	terraform apply -var 'foo=bar' terraform apply -var-file FILE_NAME.tfvars terraform.tfvars export TF_VAR_token=your-token-value	Populate variable foo with the value bar Populate variable from a file named file_name All variables will automatically be loaded from terraform.tfvars Terraform will read in all 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.al-low_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 Force an update to remote state from a local change Remove instances from the state Show a resource in the state Refresh 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 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
Work-space	terraform workspace new NAME terraform workspace select NAME terraform workspace list terraform workspace show	Initalized repo for Terraform Switch the named workspace List all available workspaces Shows the current workspace you are executing against
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