Infstrusture as code

Def- provision infrasturre through software to achive consistent and predictable environments.

Every time its use, it is consistent…creating a predictable enviornemtn.

Base on configuration files.

Core concepts

Defined in code – config files – software and coding mechanism to defined your infrastructure – json, yaml, hashicorp config language

Store code in source control – github, bitbucket,

Declarative or imperative – define your code

Idempotent and consistent –

Push or pull

Declarative or imperative –

Imperative

Procedural in nature

#make me a taco…

Get ingredients

Get shell

Get beans

etc

Assemble in order

Beans on shell

Salso on bean

Etc.

Declarative

Use configuration language like hashi corp

#make me a taco

Define what you want…

Software has a predefine routine to get and assemble items requested.

Food taco ‘’bean-taco” {

Ingredients = [

“beans”, ‘shell’, ‘lettuce’

]

}

Idempotent

Terraform is itempotent – wont create again because it already exist. In the environment.

Consistent – same each time

Push – terraform is a push type config

Pull – pulling configs from source to updates..

Infrastructure as code benefits –

Automated deployment

Consistent environments

Repeatable process reusable components – don’t repeat yourself – DRY

Documents your architecture

Deploying your first terraform configurations

Automating infrsstrutue deployment

-provision resoueces

-planning updares

-using source control

-reusuing templates

Terraform components

-terraform executable

-terraform files – configuration files could be more than one.. ##.tf

-terrform plugins – aws

Terraform state – keeps track of created resources in this file. Compares new config file to state file based

Ability to define variables , pieces of information in the configurations.

THE CONFIG FILE

Variables

Provider – required for AWS

Data – ( your ec2 or vms …aws ami, availability zones)

Resources – aws instance – create the ec2 instance t2.micro

Output – get information from the ec2, like public IP address

DON’T TOUCH THE STATE FILE!!!

You can break it.

It is in json.

Contains resource mapping and metadata

Locking – state file is in a period of flux

Location – local, remote:aws s3, azure blob , nsf share, terraformcloud

Workspaces – in terraform has its own state file

Terraform state file is true

Make all changes in the Terraform.

Serial number must match statefile and plan. Before applying changes

Terraform planning –

Refresh and panning from state files.

Inspect the state

Dependency graph – ec3 depends on subnet.

Determines the order

Additions, updates, and deletions.

Parallel execution

Save the plan file.