Provisioners:

Till Now we have been working only on creation and destruction of infrastructure scenarios.

Lets take an Example:

We created a web server EC2 instance with Terraform

Problem: It is only an Ec2 instance, it does not have any software installed.

Welcome to Terraform Provisioners:

Provisioners are used to execute scripts on a local or remote machine as part of resource creation or destruction.

Example: On creation of webserver, execute a script which installs nginx server.

Types of Provisoners:

Terraform has capability to turn provionsers both at the time of resource creation as well as destruction.

-🡪local-Exec

-🡪Remote-exec

Local-exec: Provisoners allow us to invoke local executable after resource is created.

Ex:

resource “aws\_instance” “web” {

#....

Provisioner “local-exec” {

Command = “echo ${aws\_instance.web.private\_ip} >> private\_ip.txt

}

}

Remote-exec: Provisioners allow to invoke scripts directly on the remote server.

Ex:

Resource “aws\_instance” “web” {

#...

Provisioner “remote-exec” {

#.....

}

Dry Approach:

In software Engineering, don’t repeat yourself(DRY) is a principle of software development aimed at reducing repetition of software patterns.

In the earlier lecture,we were making static content into variable so that there can be single source of information.

172.31.236.60

Centralized Structure:

We can centralize the terraform resource and can call out from TF files whenever required.

--------------

Source module”source”

Resource “aws\_instance” “web” {

Ami = “ami-239807fen”

Instance\_type = “t2.micro”

Security\_group = [“default”]

}