**Terraform Setup and Provisioning**

Download terraform from <https://www.terraform.io/downloads.html>

Set up the below

**on linux**

*mkdir terraform*

*unzip the .zip file*

*export PATH=/e/terraform/:$PATH*

*echo $PATH*

or **on windows** set in environment variables

then you should be able to trigger the command terraform from any location.

*terraform –-version*

get the IAM access keys from AWS and keep it ready.

mkdir my\_project

cd my\_project

**create an AWS instance using Terraform**

**create\_inst.tf**

==============================================================

provider "aws" {

access\_key = "xxxxxxxxxxxxxxxxxxxxxxxxx"

secret\_key = "xxxxxxxxxxxxxxxxx"

region = "ap-south-1"

}

resource "aws\_instance" "terraform\_example" {

ami = "ami-xxxxxxxxxxxxxxxx"

instance\_type = "t2.micro"

}

==================================================================

terraform init

To initialize the terraform projet

terraform plan

will validate the templates for any errors.

terraform apply

will create an instance on your specified region in AWS.

We can check the same on AWS console.

terraform destroy -- will deystory what it has created.

**Using Variables and separating the code into multiplpe files.**

**provider.tf**

========================================

provider "aws" {

access\_key = "${var.AWS\_ACCESS\_KEY}"

secret\_key = "${var.AWS\_SECRET\_KEY}"

region = "${var.AWS\_REGION}"

}

**vars.tf**

variable “AWS\_ACCESS\_KEY” {}

variable “AWS\_SECRET\_KEY” {}

variable “AWS\_REGION” {

default = “ap-south-1”

}

**terraform.tfvars**

AWS\_ACCESS\_KEY = ””

AWS\_SECRET\_KEY = ””

**create\_inst.tf**

resource "aws\_instance" "terraform\_example" {

ami = "ami-xxxxxxxxxxxxxxxx"

instance\_type = "t2.micro"

}

**Software Provisioning using Terraform**

Use the files provider.tf , terrafprm.tfvars from above and

Create a file called my\_script.sh with your own linux commands.

Create create\_inst.tf and vars.tf files as below

**create\_inst.tf**

======================================================================================

resource "aws\_instance" "terraform\_example" {

ami = "ami-0912f71e06545ad88"

instance\_type = "t2.micro"

key\_name = “<your key name on AWS> "

provisioner "file" {

source = "script.sh"

destination = "/tmp/script.sh"

}

connection {

user = "${var.INSTANCE\_USERNAME}"

private\_key = "${file("${var.PATH\_TO\_PVT\_KEY}")}"

}

}

vars.tf

variable AWS\_ACCESS\_KEY {}

variable AWS\_SECRET\_KEY {}

variable AWS\_REGION {

default = "ap-south-1"

}

variable PATH\_TO\_PVT\_KEY {

default = "< path of your pvt key on your machine( .pem) >"

}

variable INSTANCE\_USERNAME {

default = "ec2-user"

}

**Windows Server Provisioning using terraform**

**create\_inst.tf**

resource "aws\_instance" "terraform\_example" {

ami = "ami-0912f71e06545ad88"

instance\_type = "t2.micro"

key\_name = "<your key name on AWS> "

userdata = <<EOF

<powershell>

net user ${var.INSTANCE\_USERNAME} ${var.INSTANCE\_PASSWORD} /add

net localgroup administrators ${var.INSTANCE\_USERNAME} /add

winrm quickconfig –q

winrm set winrm/config/winrs ‘@{MaxMemoryPerShellMB=”300”}’

winrm set winrm/config ‘@{MaxTimeoutMS=”180000”}’

winrm set winrm/config/service ‘@{AllowUnencrypted=”true”}’

winrm set winrm/config/service/auth ‘@{Basic=”true”}’

netsh advfirewall firewall add rule name=”WinRM 5985” protocol=TCP dir=in localport=5985 action=allow

netsh advfirewall firewall add rule name=”WinRM 5986” protocol=TCP dir=in localport=5986 action=allow

net stop winrm

sc.exe config winrm start=auto

net start winrm

</powershell>

EOF

connection {

type = “winrm”

user = "${var.INSTANCE\_USERNAME}"

password = "${var.INSTANCE\_PASSWORD" }

}

**vars.tf**

variable AWS\_ACCESS\_KEY {}

variable AWS\_SECRET\_KEY {}

variable AWS\_REGION {

default = "ap-south-1"

}

variable PATH\_TO\_PVT\_KEY {

default = "< path of your pvt key on your machine( .pem)>"

}

variable INSTANCE\_USERNAME {

default = "administrator"

}

variable INSTANCE\_PASSWORD { }