Terraform

##
Terraform
##

-----*** Installing Terraform ***-----

```
$ sudo apt update
```

```
$ wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/hashicorp-archive-keyring.gpg
```

```
$ echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg]
https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee
/etc/apt/sources.list.d/hashicorp.list
```

```
$ sudo apt update && sudo apt install terraform
```

```
$ terraform -help
```

---** Create the first resource **---

```
$ mkdir test1
```

```
$ cd test1
```

```
$ nano main.tf
resource "local_file" "hello" {
   filename = "hello.txt"
   content = "Hello World"
}
```

```
$ terraform init
```

```
$ terraform fmt
$ terraform validate

$ terraform plan

$ terraform apply
```

---** Update & Destroy **---

```
$ nano main.tf
resource "local_file" "hello" {
   filename = "hello.txt"
   content = "Hello World"
   file_permission = "0700"
}

$ terraform plan

$ terraform apply

$ ls -l

$ terraform destroy
```

---** Using Multiple Providers **---

```
$ nano main.tf
resource "random_password" "password" {
  length          = 16
  special         = true
 }

$ terraform init

$ terraform plan

$ terraform apply
```

---** Using Variables **---

```
$ cd

$ mkdir test2

$ cd test2

$ nano variables.tf
variable "filename" {
    default = "hello.txt"
}
variable "content" {
    default = "Hello World"
}
variable "length" {
    default = "16"
}
variable "special" {
    default = "true"
}


$ nano main.tf
resource "local_file" "hello" {
  filename = var.filename
  content  = var.content
}

resource "random_password" "password" {
  length  = var.length
  special = var.special
}

$ terraform init

$ terraform plan

$ terraform apply
```

>>> Getting input from users

```
$ nano variables.tf
variable "filename" {

}
variable "content" {

}
variable "length" {

}
variable "special" {

}

$ terraform apply
```

---** Using Resource Attributes **---

```
$ mkdir test3

$ cd test3

$ nano main.tf
resource "local_file" "hello" {
  filename = "hello.txt"
  content  = "Password generated is ${random_password.password.result}"
}

resource "random_password" "password" {
  length  = 09
}

$ terraform init

$ terraform plan

$ terraform apply
```

---** Using Output Variables **---

```
$ nano main.tf
resource "local_file" "hello" {
  filename = "hello.txt"
  content  = "Password generated is ${random_password.password.result}"
}

resource "random_password" "password" {
  length  = 09
}

resource "random_id" "myid" {
     byte_length = 8
}

output id {
    value = random_id.myid.id
}
```

```
$ terraform plan
```

```
$ terraform apply
```

(Look for output)

```
$ terraform output
```

---** Working with AWS **---

>> Install awscli & configure credentials

```
$ mkdir aws-test
```

```
$ cd aws-test
```

```
$ nano main.tf
provider "aws" {
    region = "us-east-2"
    # access_key = ""
    # secret_key = ""
}
```

```
resource "aws_instance" "my-instance" {
   ami = "ami-05fb0b8c1424f266b"
   instance_type = "t2.micro"
   availability_zone = "us-east-2a"
   key_name = "ohio1"
   vpc_security_group_ids = ["sg-075e564816af00f1d"]
}
```

```
$ terraform init
```

```
$ terraform validate
```

```
$ terraform plan
```

```
$ terraform apply
```

---** Setting up Nifi with Terraform **---

```
$ mkdir nifi-test
```

```
$ cd nifi-test
```

```
$ nano main.tf
provider "aws" {
   region = "us-east-2"
}

resource "aws_instance" "nifi" {
   ami = "ami-05fb0b8c1424f266b"
   instance_type = "t2.medium"
   availability_zone = "us-east-2a"
   key_name = "ohio1"
   vpc_security_group_ids = ["sg-075e564816af00f1d"]

   user_data = <<-EOF
          #!/bin/bash
          sudo apt update
          sudo apt install openjdk-8-jdk openjdk-8-jre -y
          wget https://archive.apache.org/dist/nifi/1.9.0/nifi-1.9.0-bin.tar.gz
          tar -xzf nifi-1.9.0-bin.tar.gz -C /opt
```

```
            ln -s /opt/nifi-1.9.0 /opt/nifi
            /opt/nifi/bin/nifi.sh start
            EOF

}

output publicip {
   value = aws_instance.nifi.public.public_ip
}
```

```
$ terraform init

$ terraform validate

$ terraform plan

$ terraform apply
```

---** Trobleshooting & Debuggin in Terraform **---

```
$ export TF_LOG=TRACE
(TRACE, DEBUG, INFO, WARN or ERROR)

$ export TF_LOG_PATH=/tmp/terraform.log
```