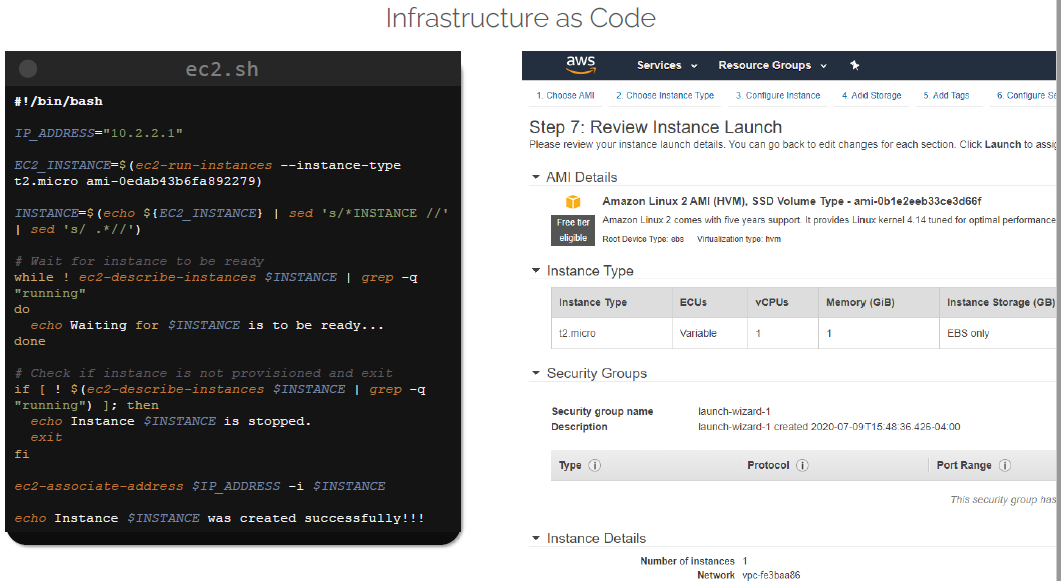
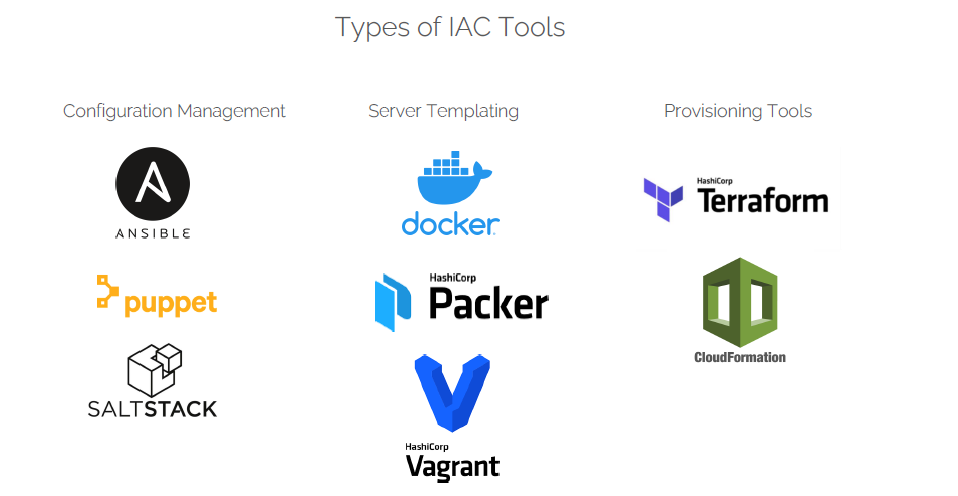
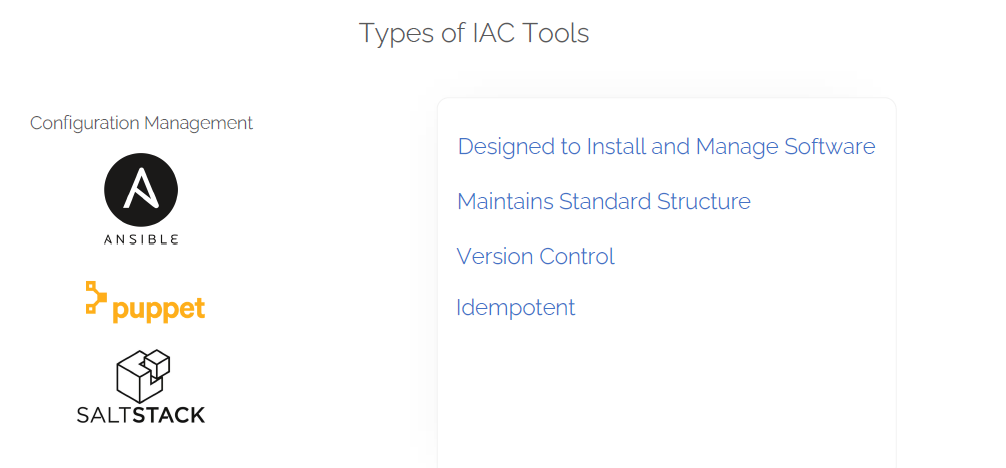
Terraform













HashiCorp Language Syntax:

resource “aws\_instance” “demo” {

ami = "ami-0c2f25c1f66a1ff4d"

instance\_type="t2.micro"

}

Here resource (BlockName) “aws\_instance” (ResourceType) “demo”(ResourceNameAnything of ourchoice) {

ami = "ami-0c2f25c1f66a1ff4d" (Argument 1)

instance\_type="t2.micro" (Argument 2)

}

**Note: Resource type is always in this format providername\_resourcewewanttoprovisioned**

**Ex: aws(ProviderName)\_instance(ResouceWeWantToProvisioned)**

**Terraform Commands:**

Command1: terraform init ( To initiate the process it will also download the required plugins for the provider for example if we are using AWS as our infrastructure it will download all the plugins require by AWS)

Command2: terraform validate ( It will validate our script for any syntax errors)

Command3: terraform fmt (This command not only validate our script but also give us suggestion to fix the issues and also format our code)

Command4: terraform plan (It is like a dry run in ansible. Shows the plan but doesn’t make any implementation.

Command5: terraform apply ( To provisioned the resources)

Command6: terraform show (It will show the current resources provisioned using terraform)

Command7: terraform destroy ( It will destroy all the resources provisioned using terraform)

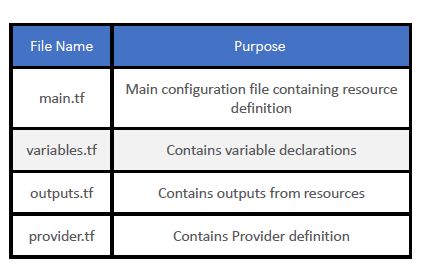
Command8: terraform destroy - - target ResourceName(To destroy specific resource)

Command9: terraform refresh (It will update the .tfstate file if there was any manual provisioned done on the infrastructure)

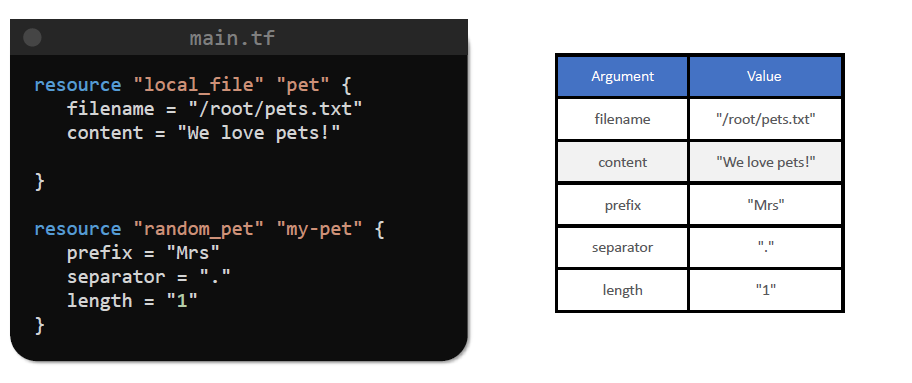
Command10: terraform graph (It will gives a graphical representation of our terraform provisioning)

Command11: terraform output ( It will print the output variable if we have defined any

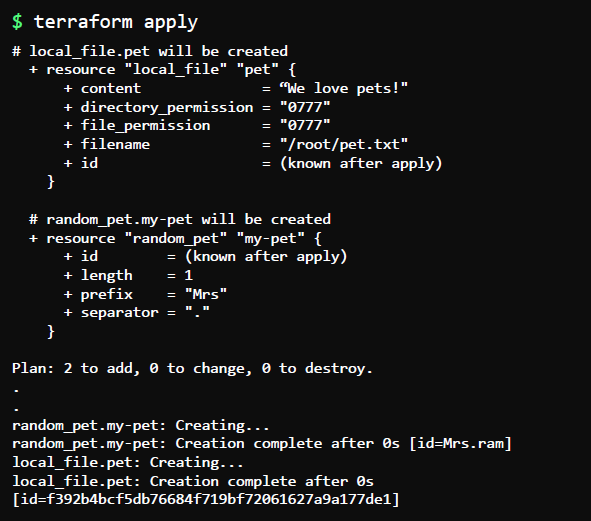
**Configuration Directory:**

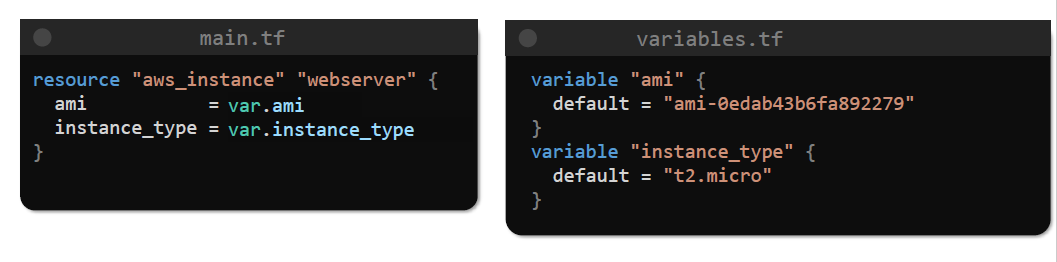


**Define Input Variables In Terraform:**

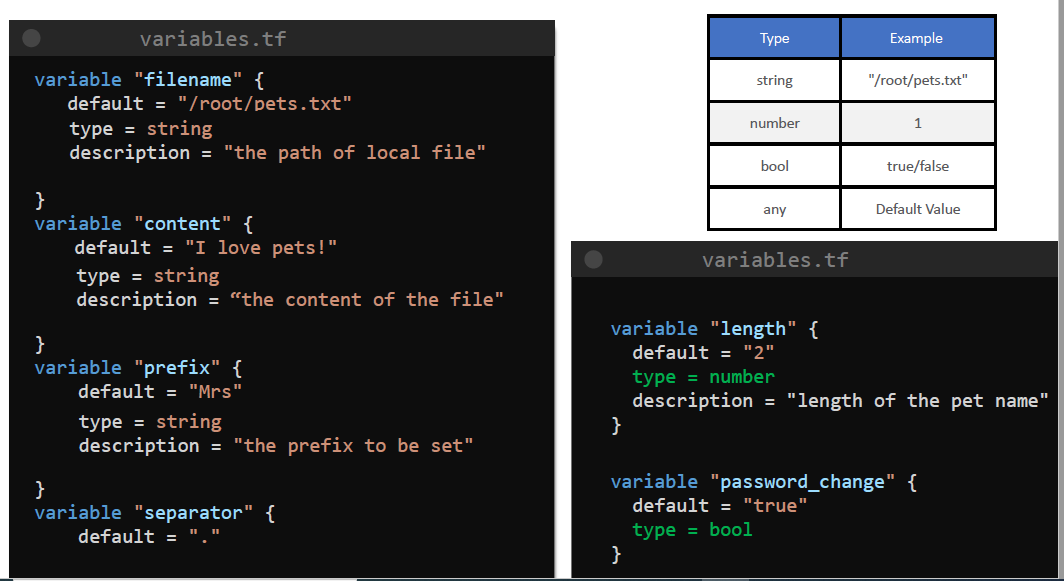


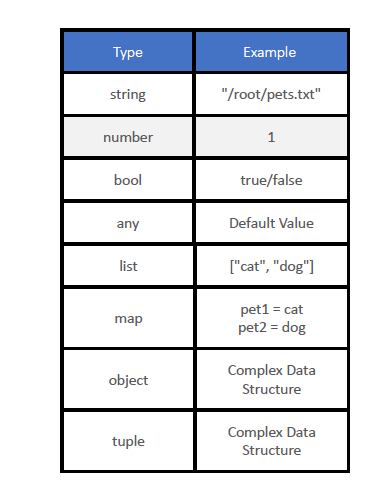




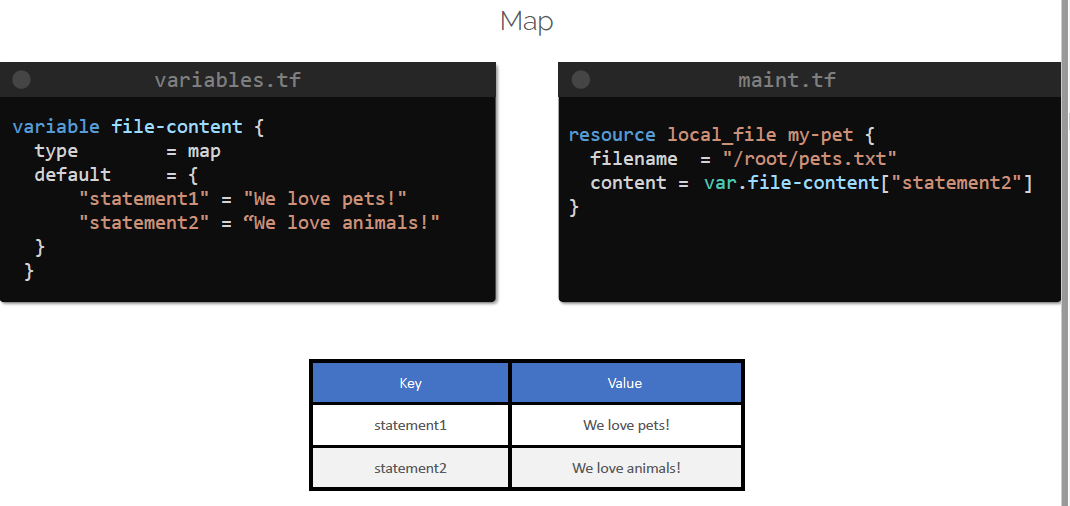


**Different Types of Variables in Terraform**

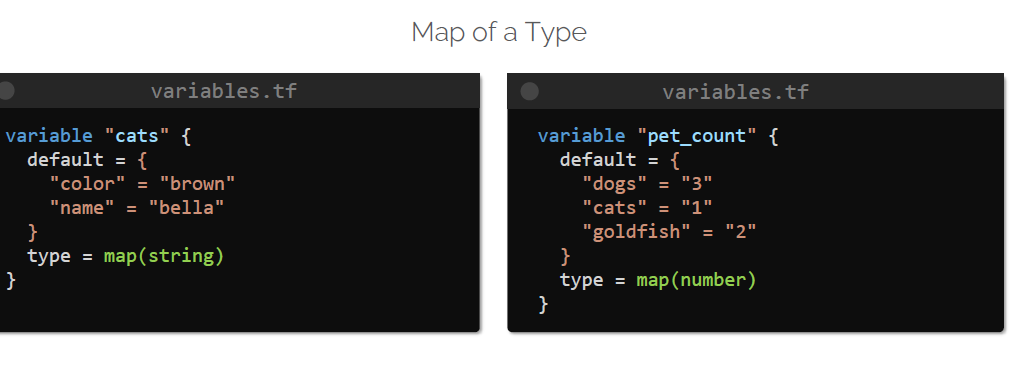




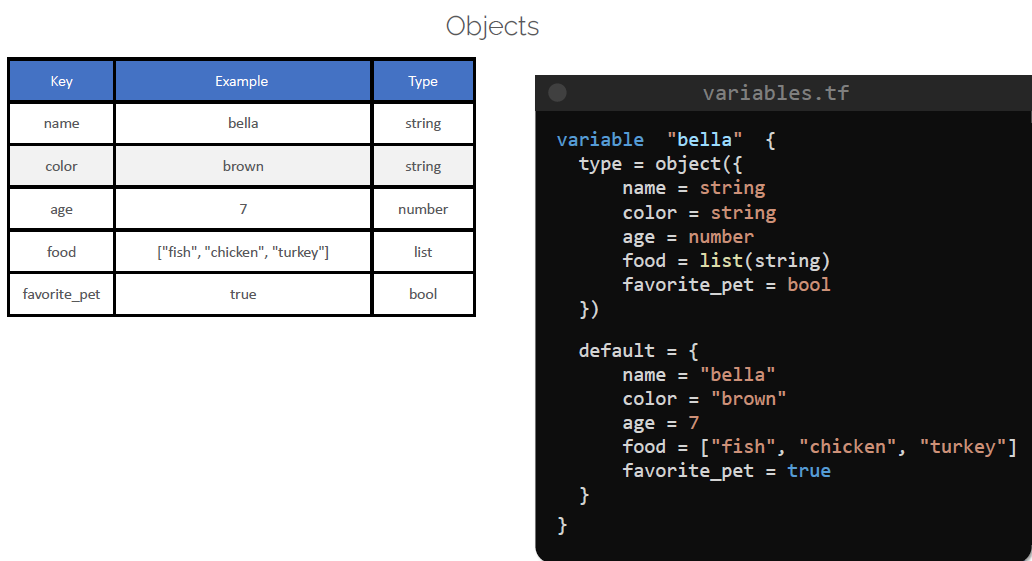


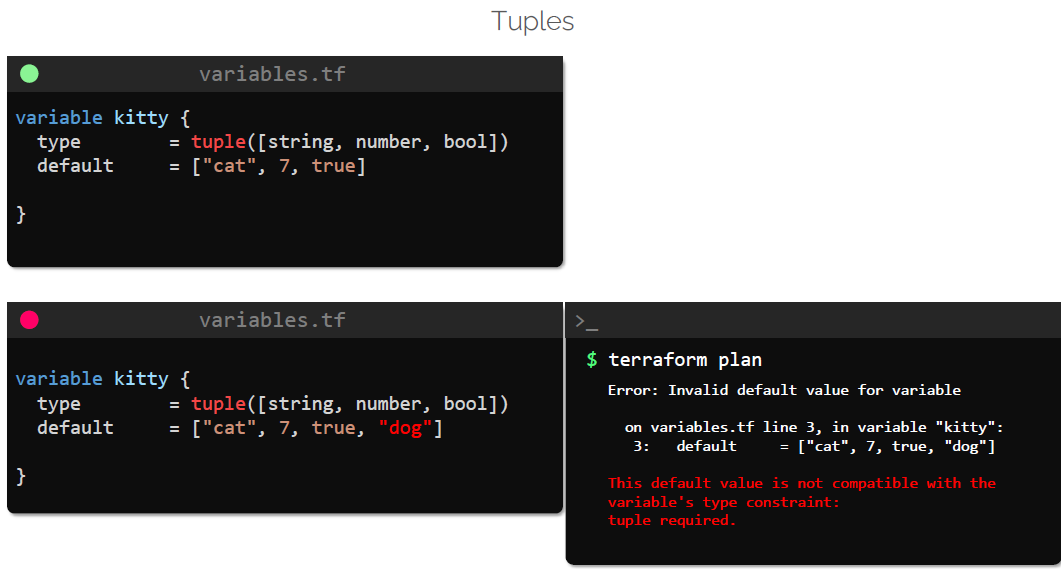












**Using Variables in Terraform:**

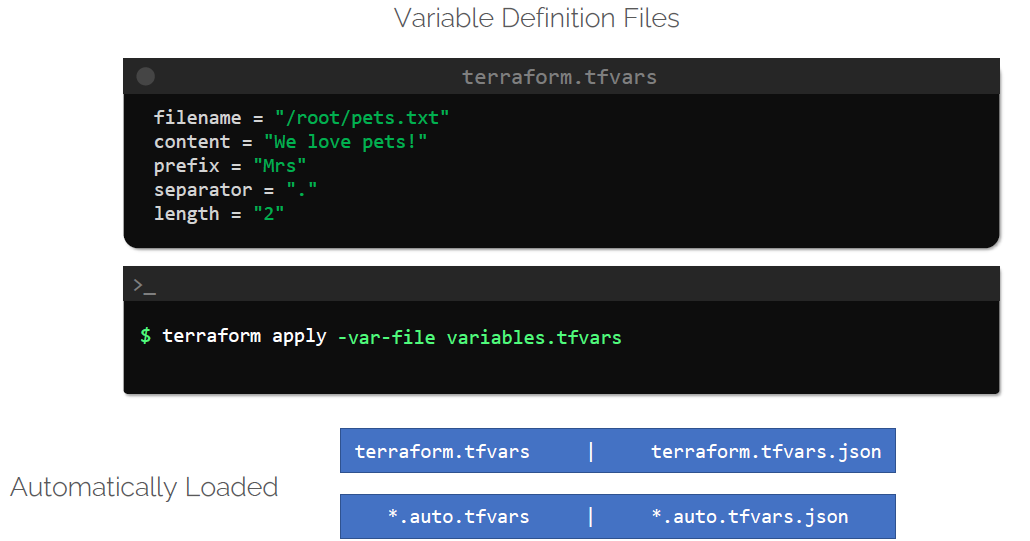


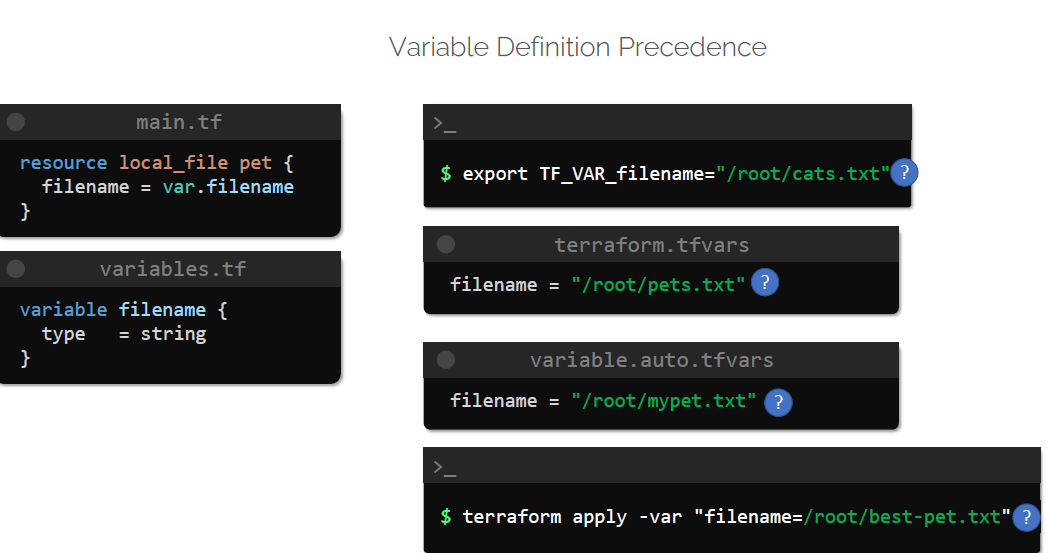


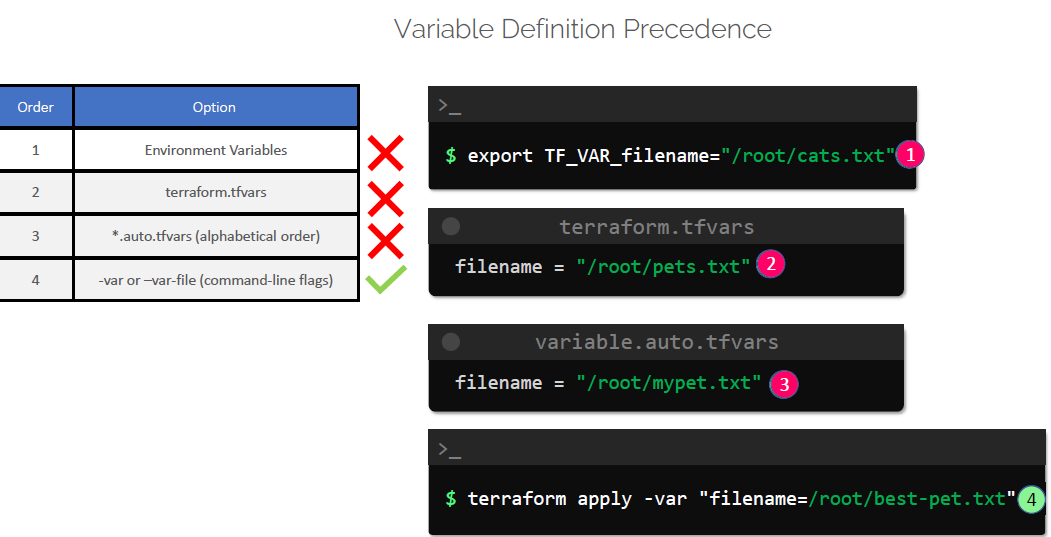




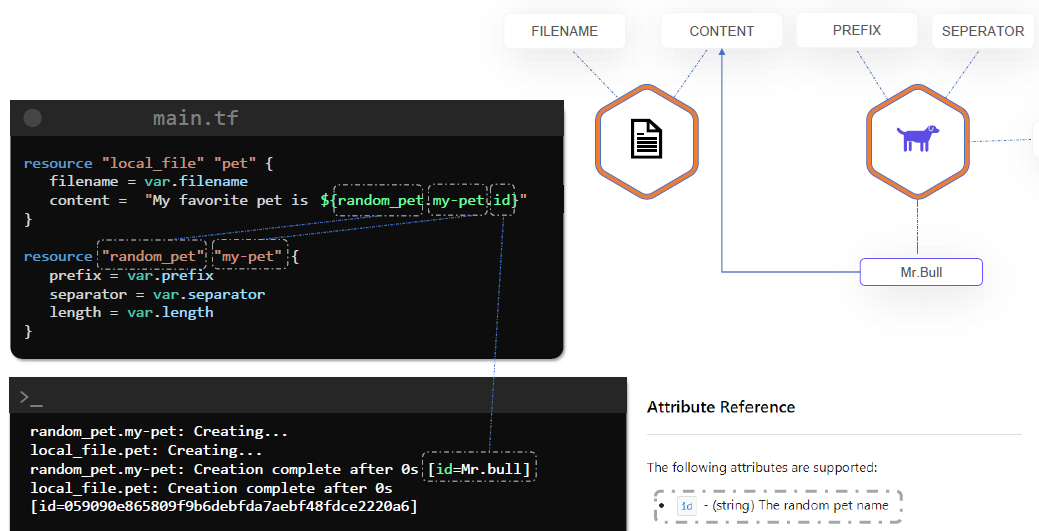


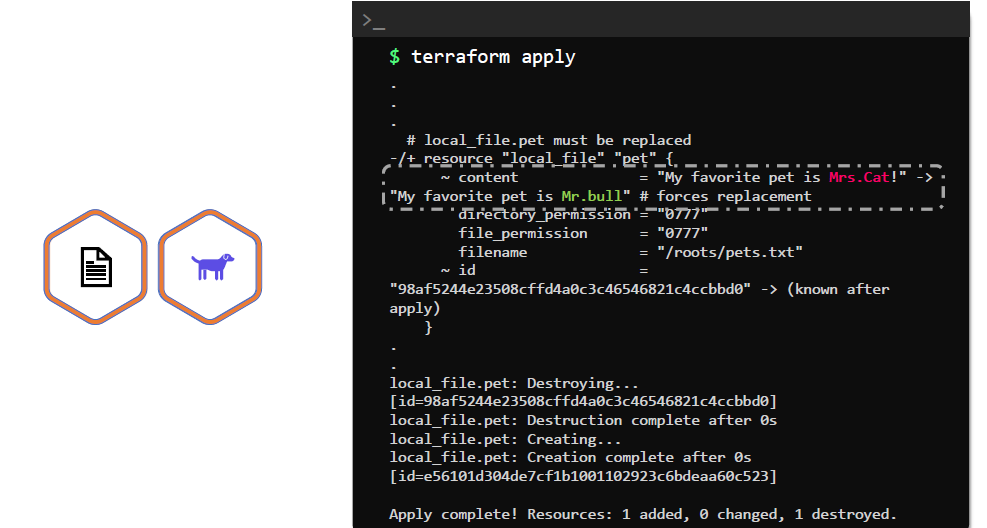


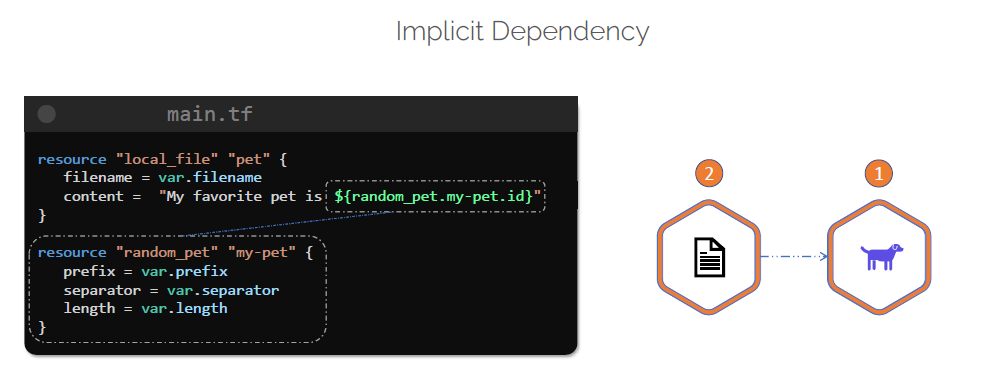


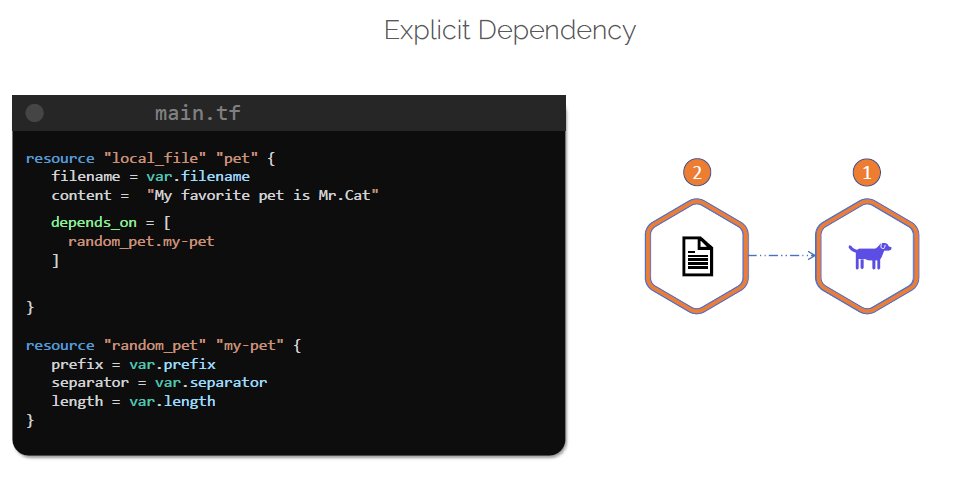


**Resource Attribute Reference:**







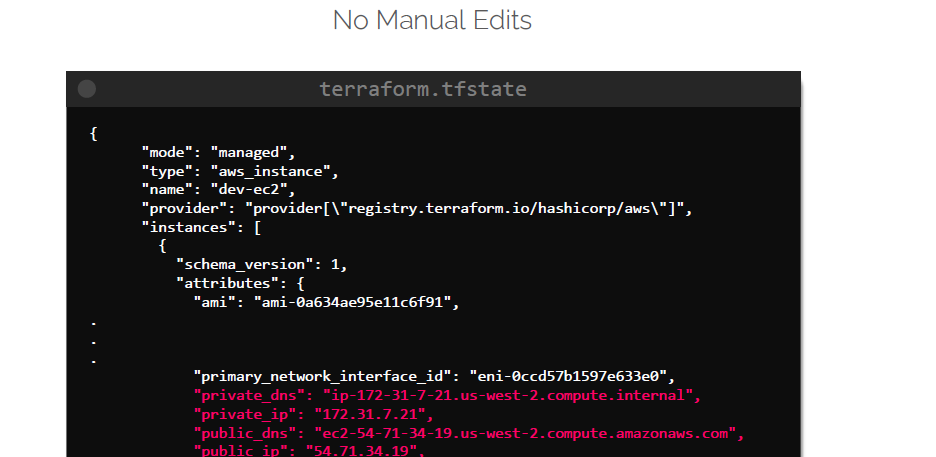


OutPut Variable In Terraform:



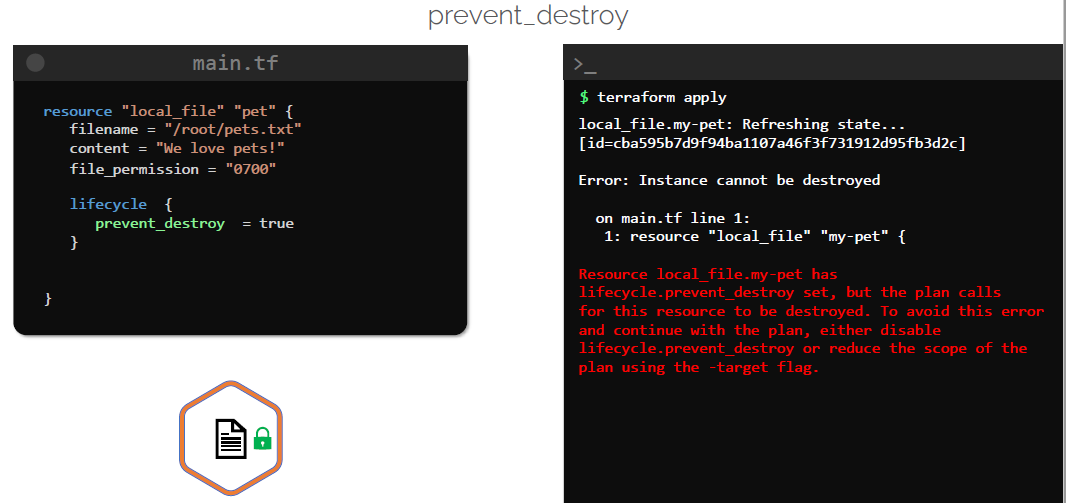


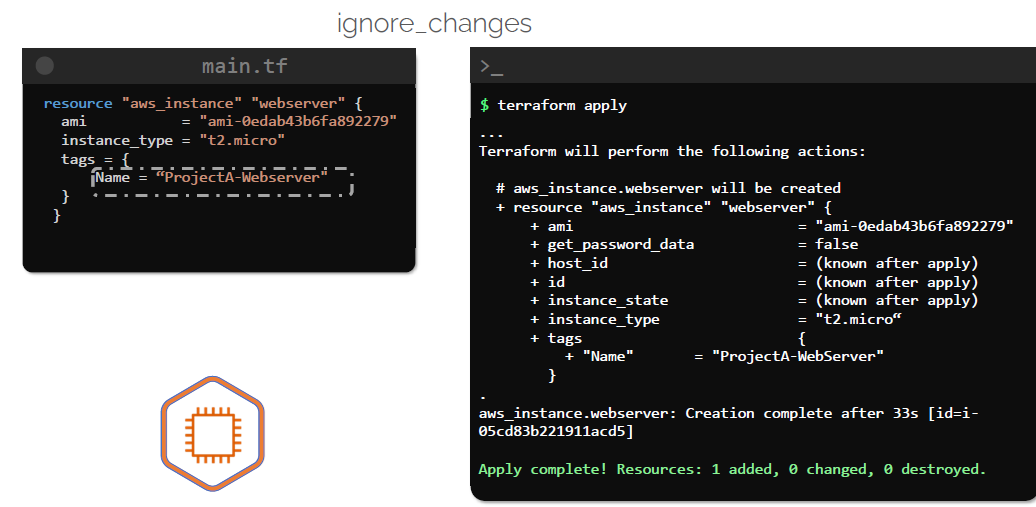
Terraform State file (terraform.tfstatefile) It is like a blue print of our infrastrucutre provisioned using terraform.

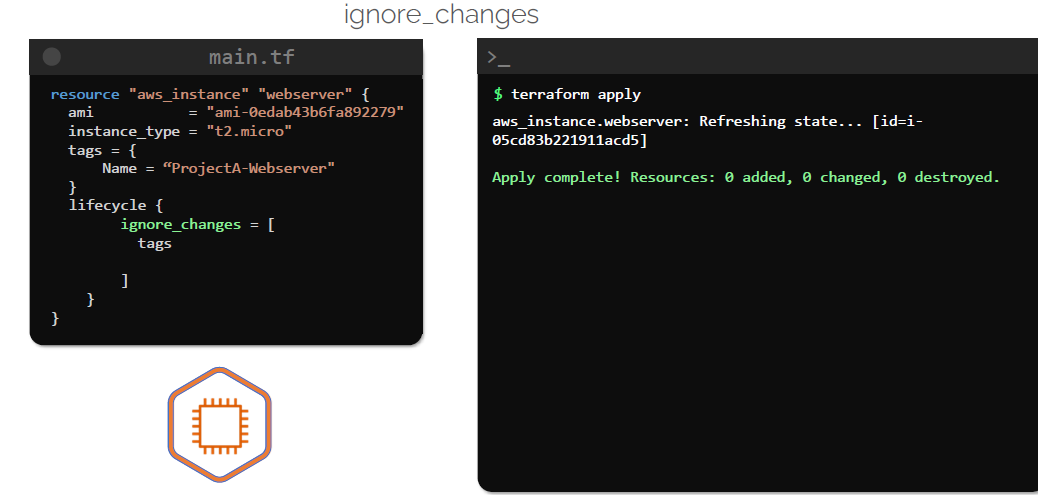


Lifecycle in Terraform:

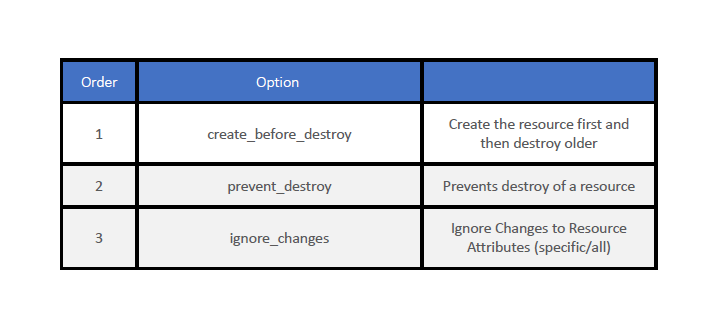






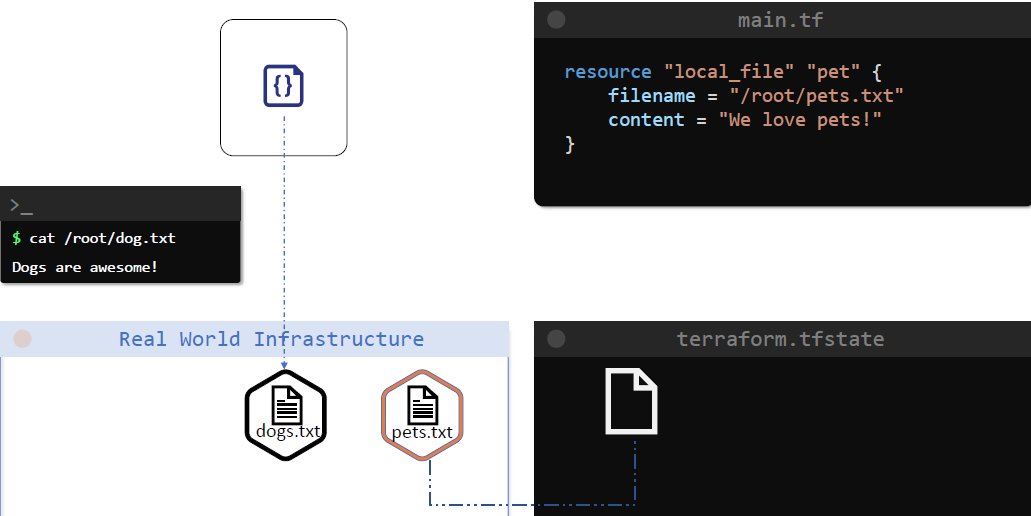


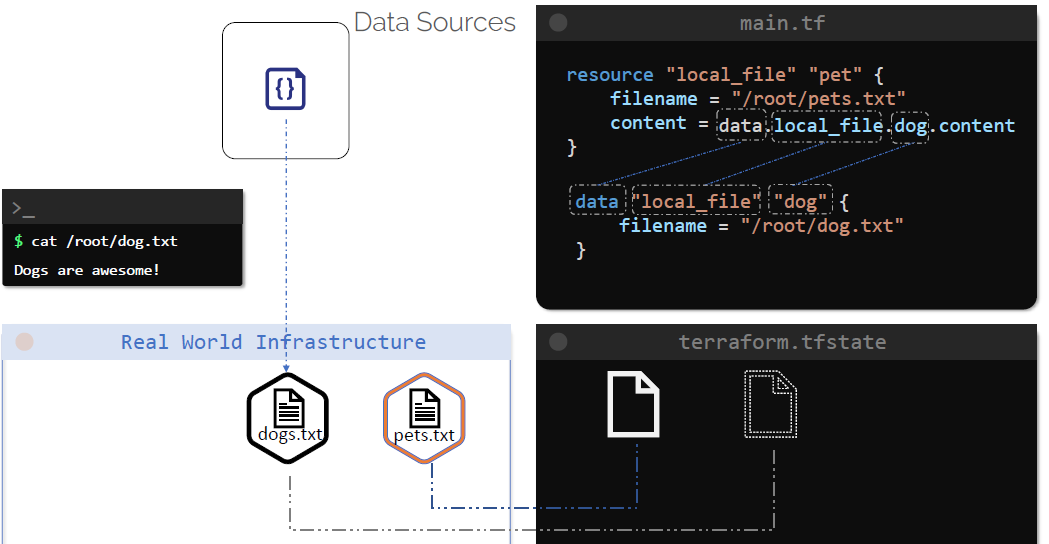


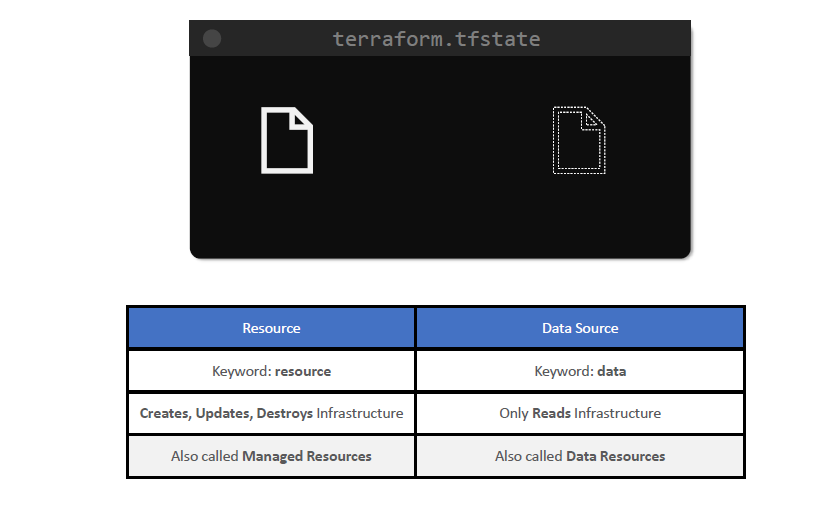


Datasources in Terraform:

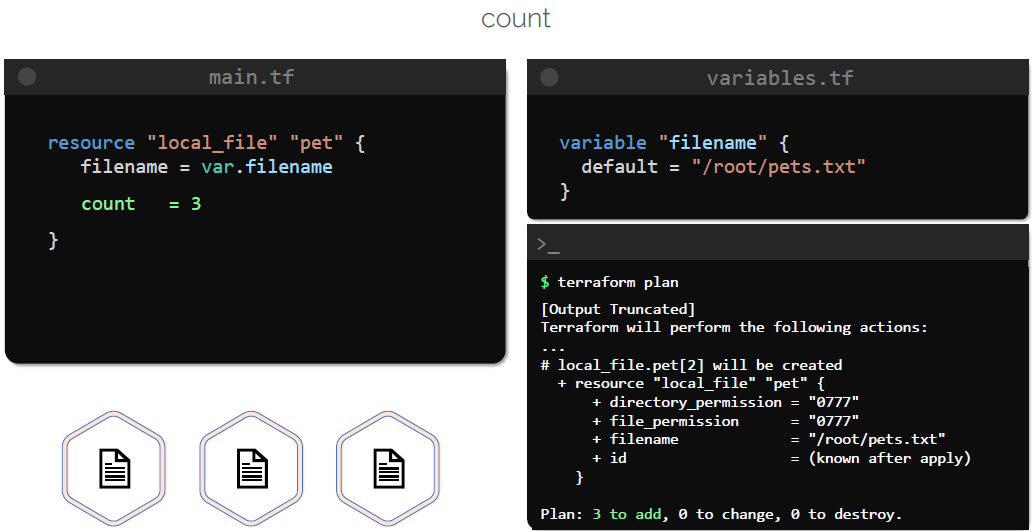
It is used when we create any resource without using Terraform but we want to utilized that specific resource.



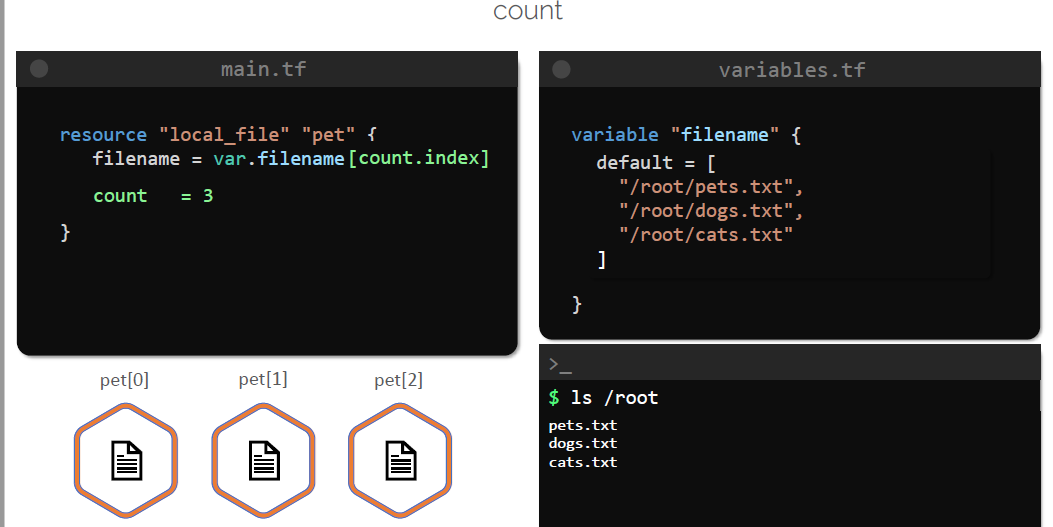




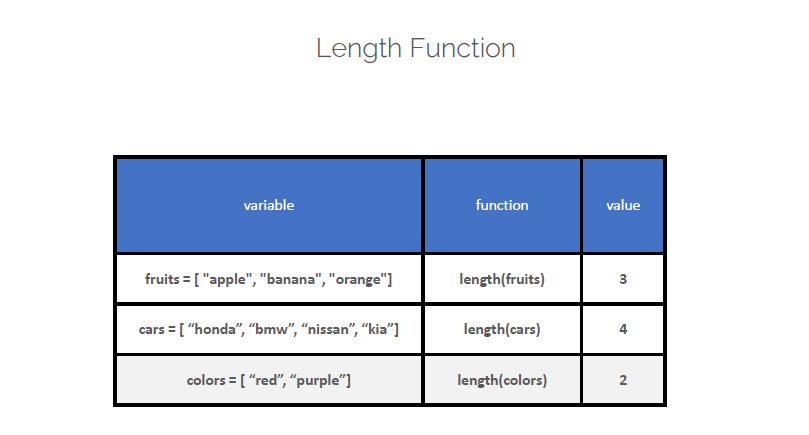
**MetaArguments In Terraform:**

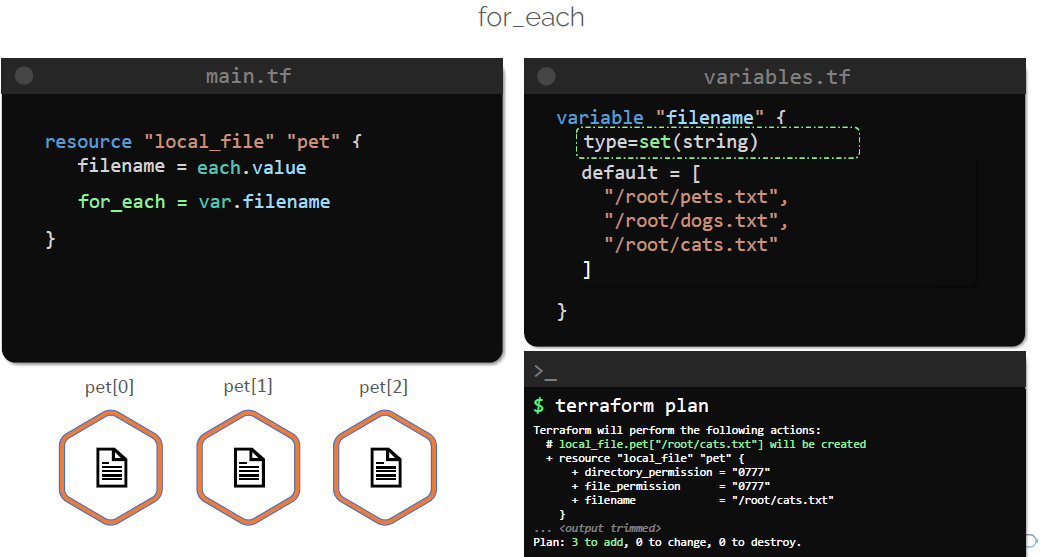


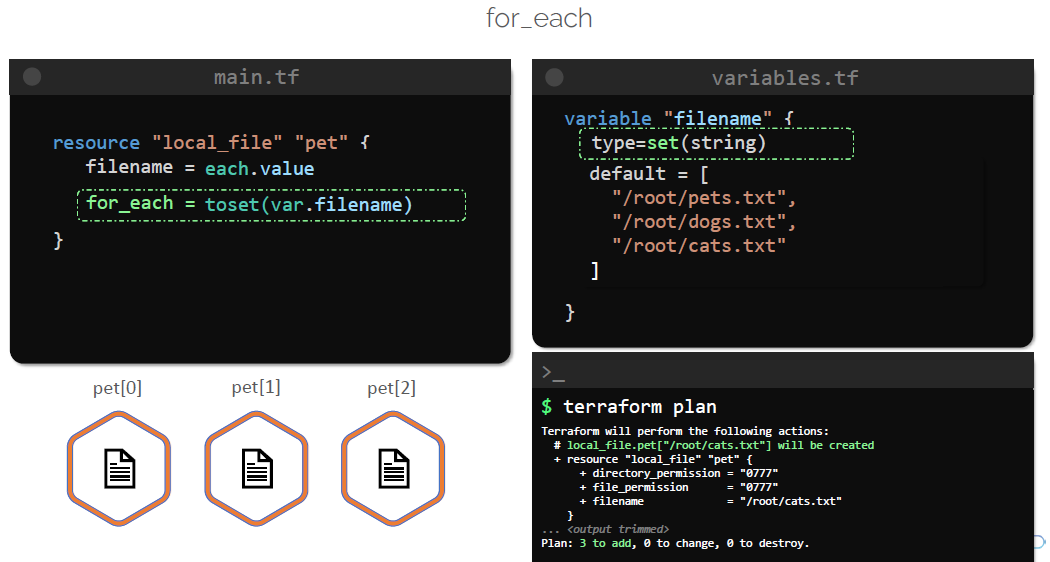


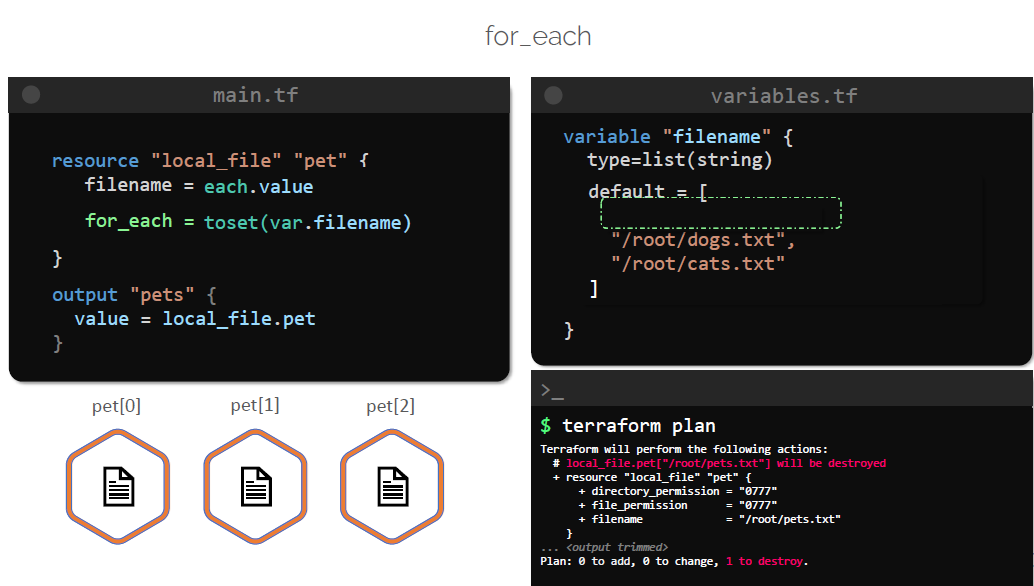


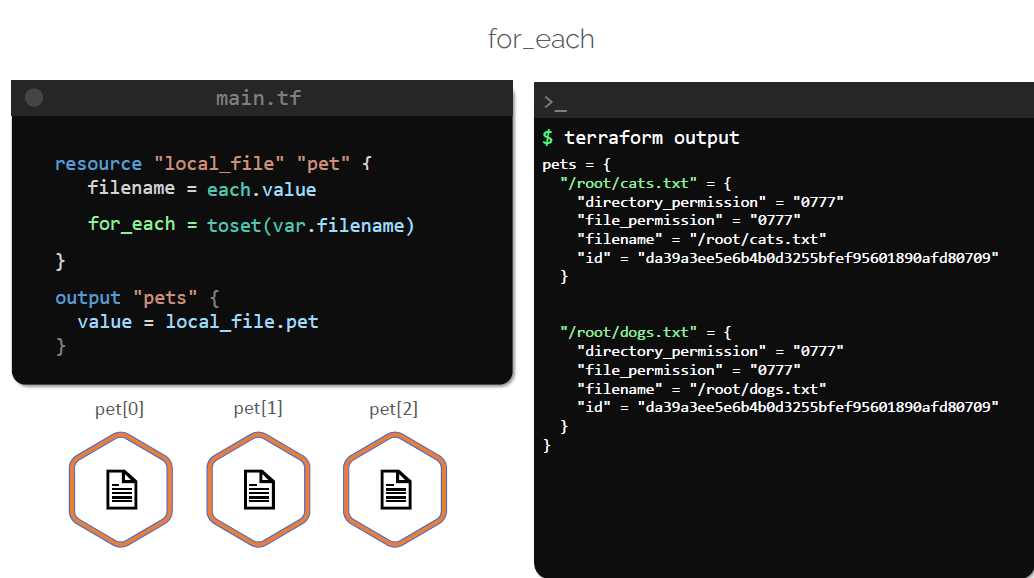


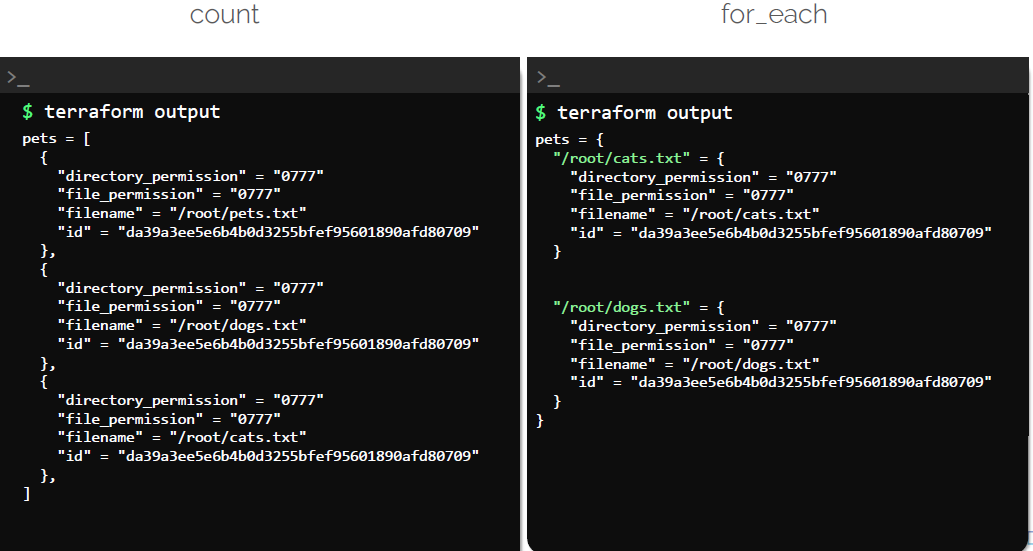






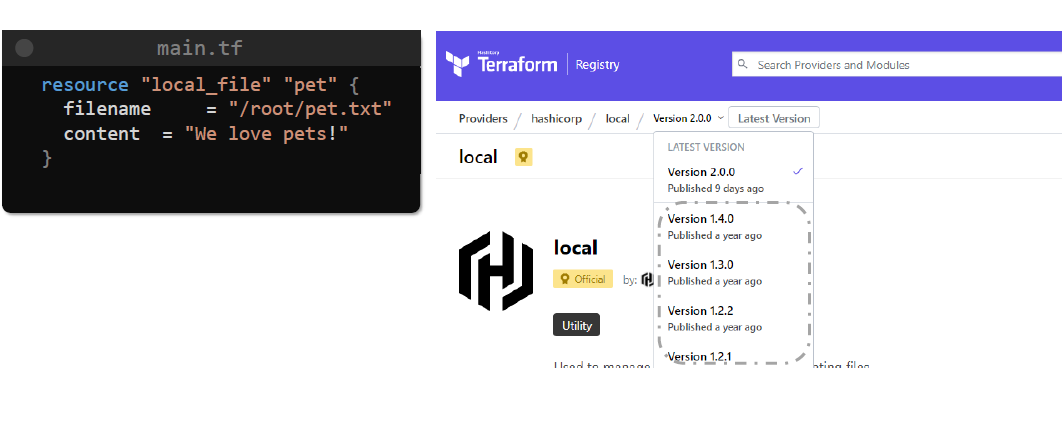






**Version Constraints in Terraform:**

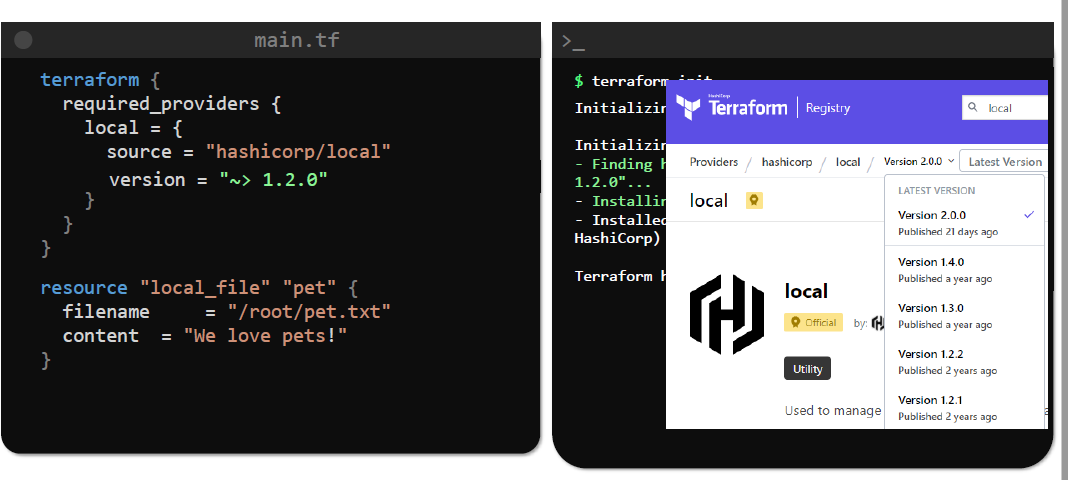




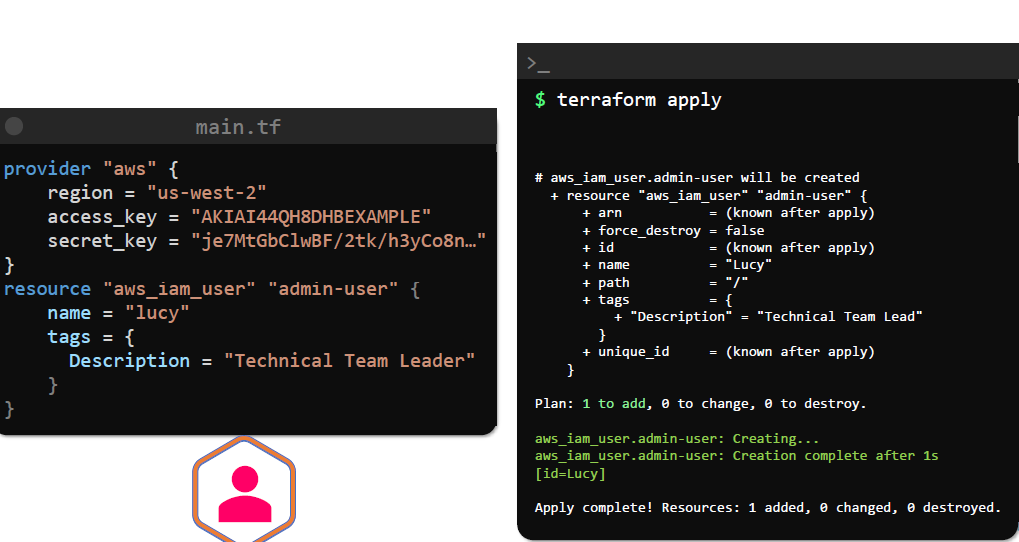






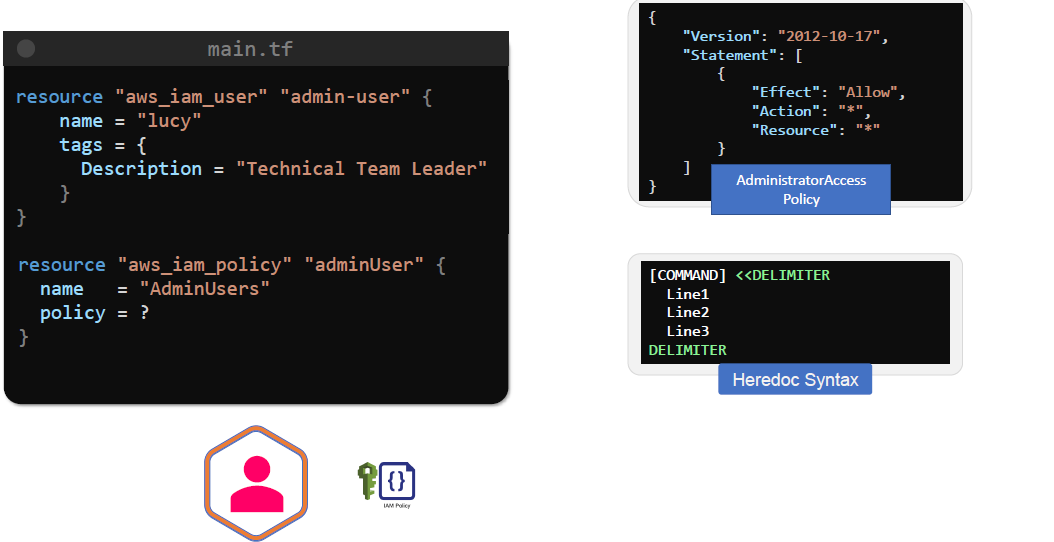


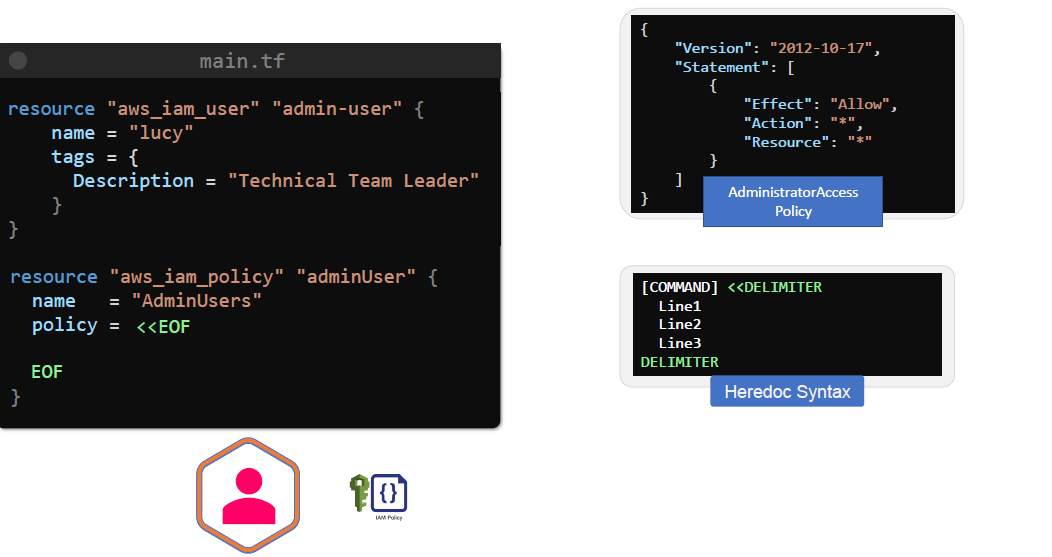
**Provisioning Resources Using Terraform:**

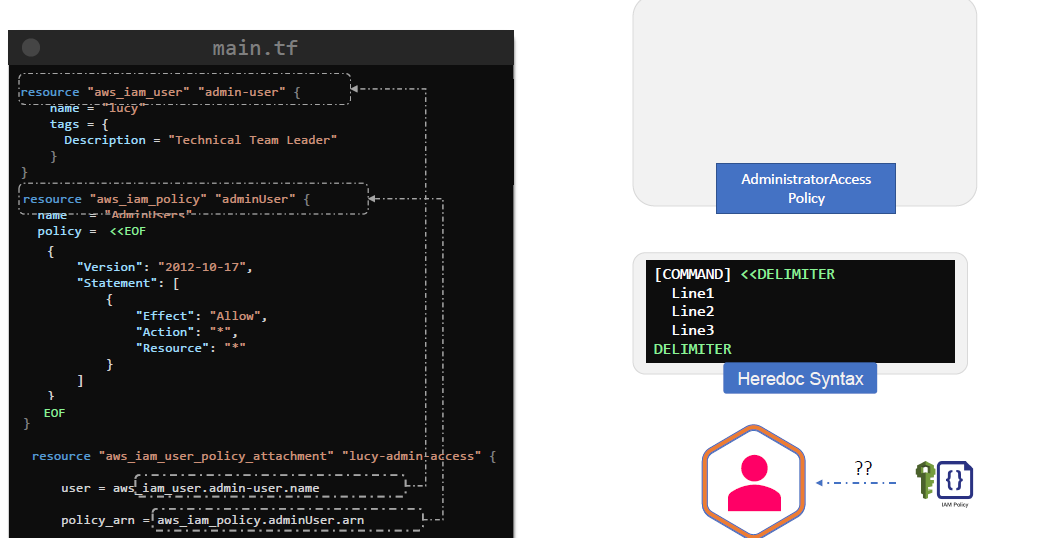








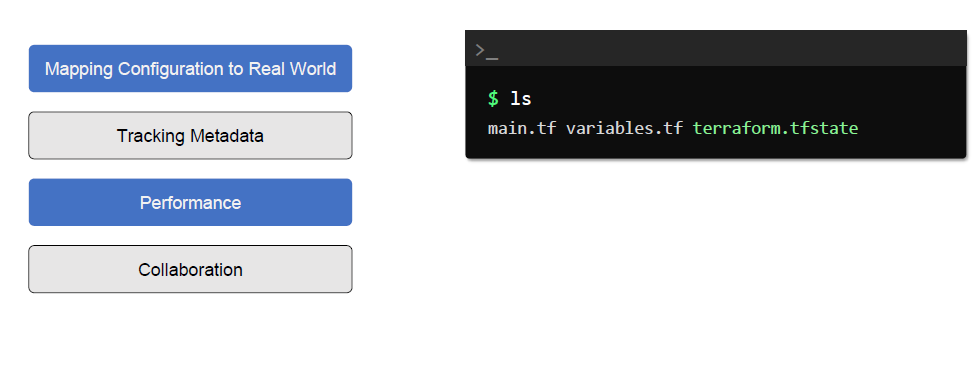


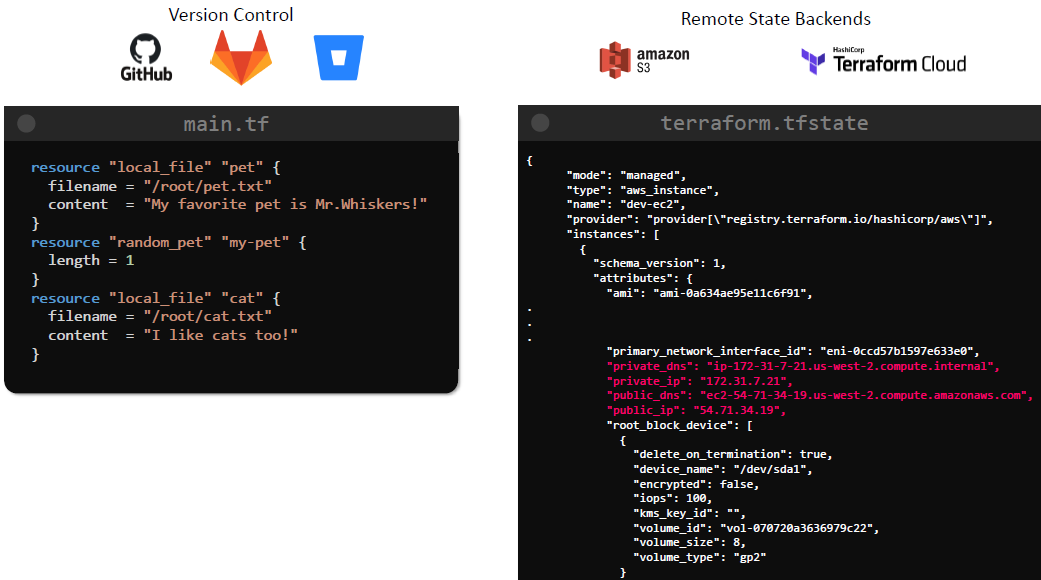


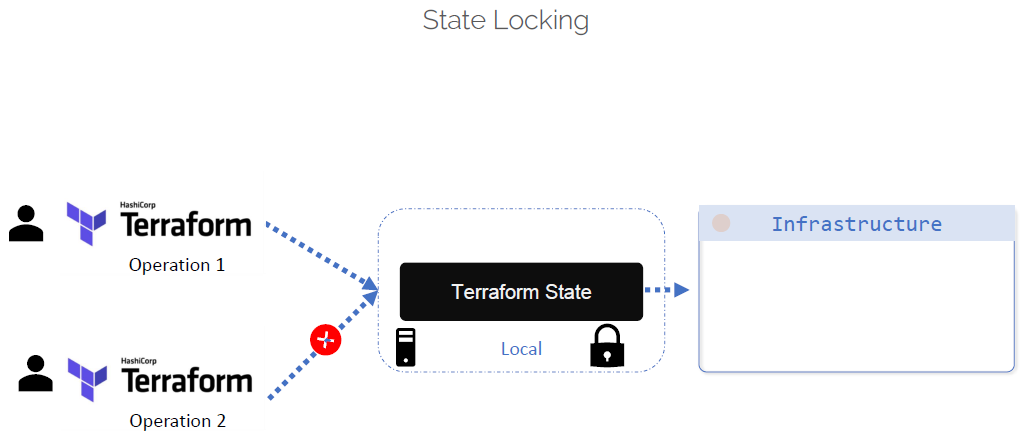


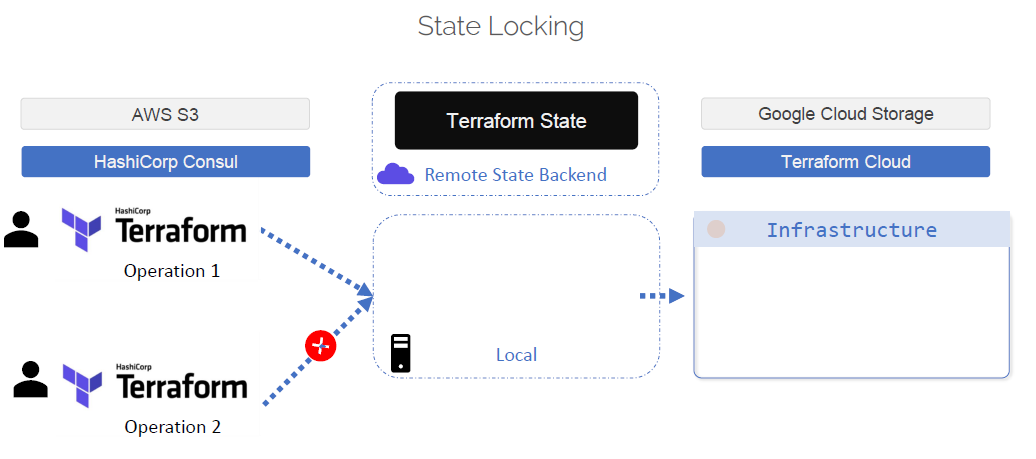


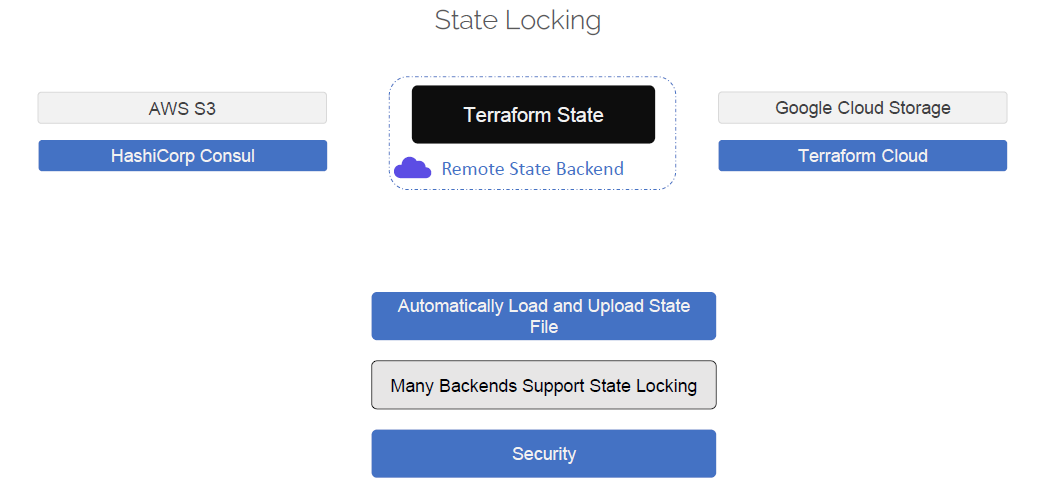
**Remote State in Terraform:**

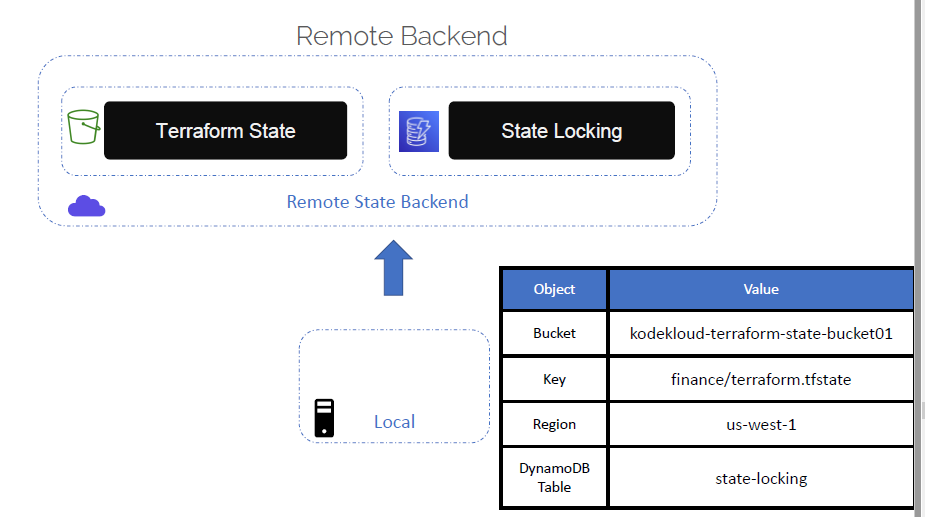












**Main.tf**

resource"local\_file""pet"{​

filename ="/root/pets.txt"​

content ="We love pets!"​

}

**terraform.tf**

terraform{

backend"s3"{

bucket="BucketNameCreated "

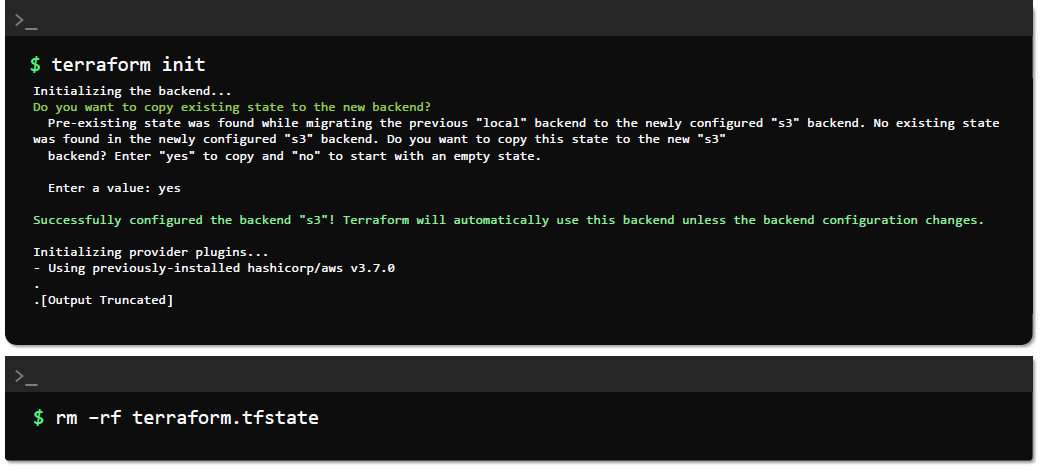
key="folderWecreated/terraform.tfstate"

region ="us-west-1"

dynamodb\_table= "state-locking"

}

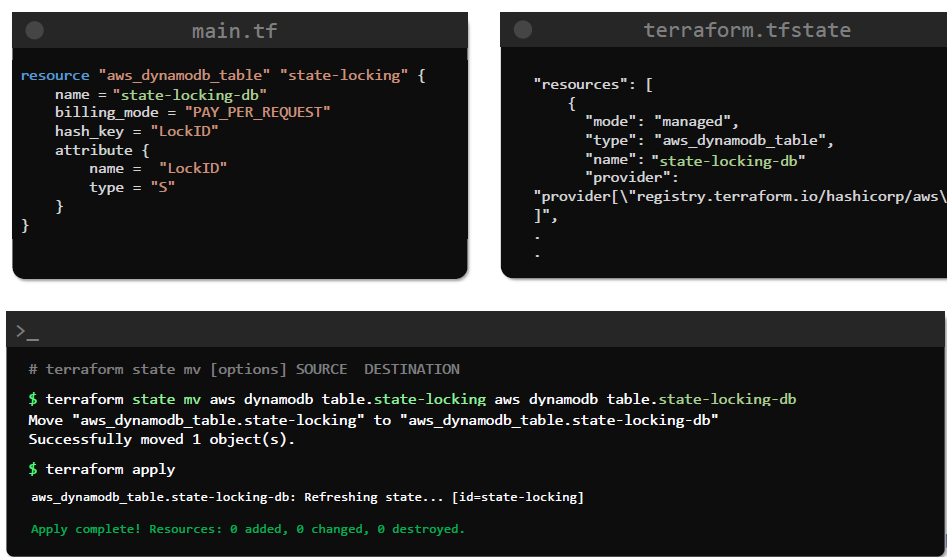
}



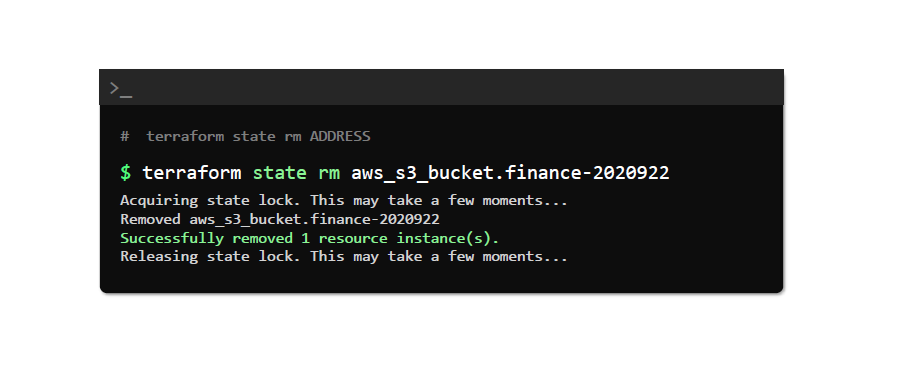
**Terraform State Command:**

**Command1: terraform state list ( To list)**

**Command2: terraform state show bucketname**







**Vi demo.tf**

**terraform {**

**required\_providers {**

**aws = {**

**source = "hashicorp/aws"**

**version = "~> 4.0"**

**}**

**}**

**}**

**provider "aws" {**

**region = "ap-south-1"**

**access\_key = "AKIAQCLYFQVNV7JYZY5Q"**

**secret\_key = "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"**

**}**

**resource "aws\_instance" "webserver" {**

**ami = "ami-07ffb2f4d65357b42"**

**instance\_type = "t2.micro"**

**tags = {**

**Name = "Webserver"**

**}**

**user\_data = file("/home/ec2-user/terraformdemo/script.sh")**

**key\_name = aws\_key\_pair.web.id**

**vpc\_security\_group\_ids = [aws\_security\_group.ssh-access.id]**

**}**

**resource "aws\_key\_pair" "web" {**

**public\_key = file("/home/ec2-user/.ssh/id\_rsa.pub")**

**}**

**resource "aws\_security\_group" "ssh-access" {**

**name = "ssh-access"**

**description = "allow ssh access"**

**ingress {**

**from\_port = 22**

**to\_port = 22**

**protocol = "tcp"**

**cidr\_blocks = ["0.0.0.0/0"]**

**}**

**}**

**resource "aws\_s3\_bucket" "demobucketterraform" {**

**bucket = "omerabbadi678"**

**tags = {**

**Description = " Demo bucket created using terraform"**

**}**

**}**

**resource "aws\_s3\_object" "demoobjectterraform" {**

**content = "/home/ec2-user/terraformdemo/script.sh"**

**key = "script.sh"**

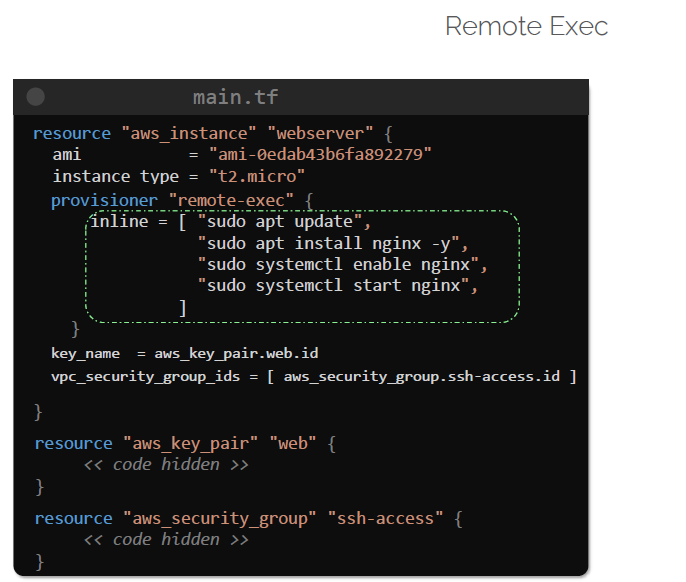
**bucket = aws\_s3\_bucket.demobucketterraform.id**

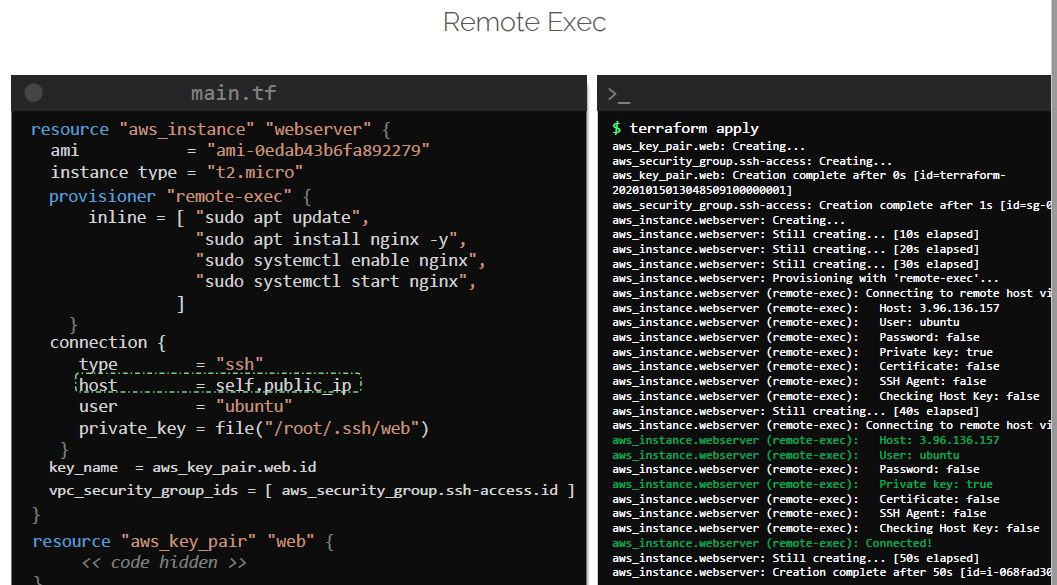
**}**

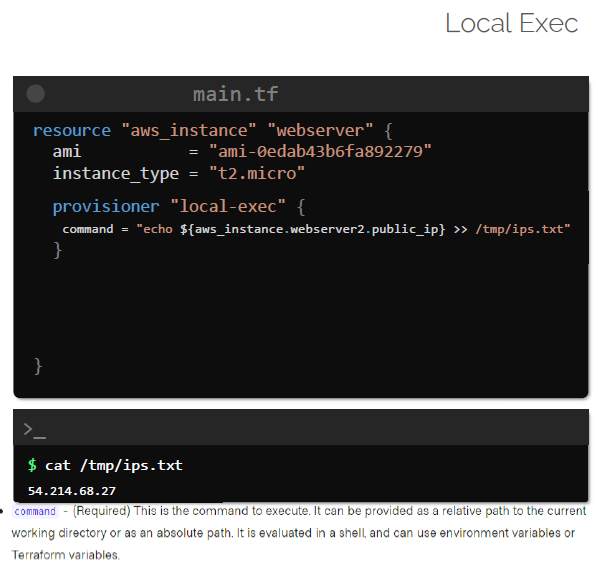
**output "publicip" {**

**value = aws\_instance.webserver.public\_ip**

**}**





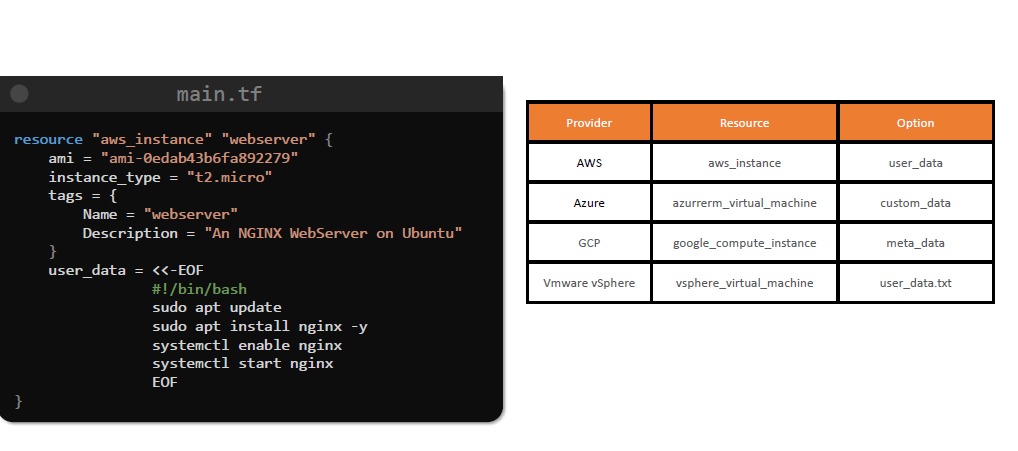


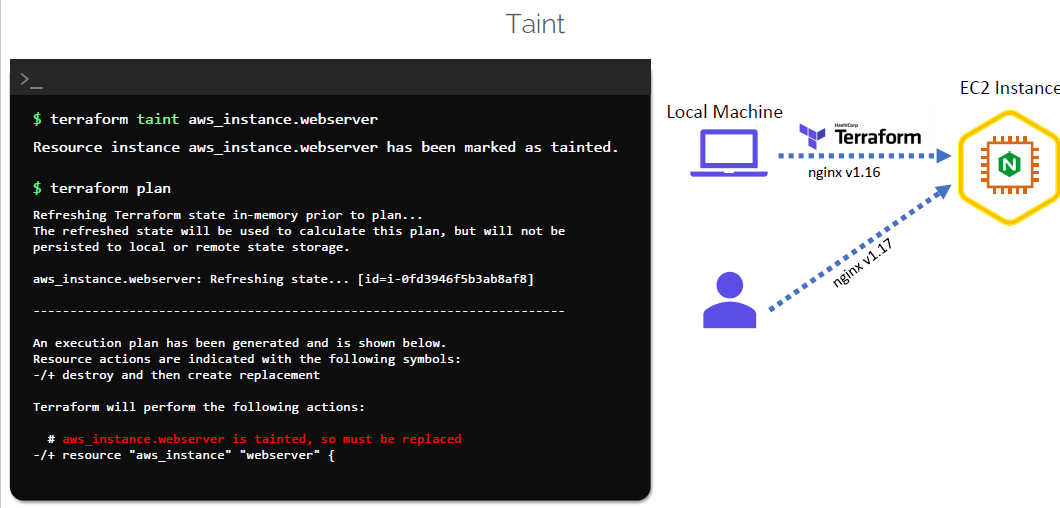








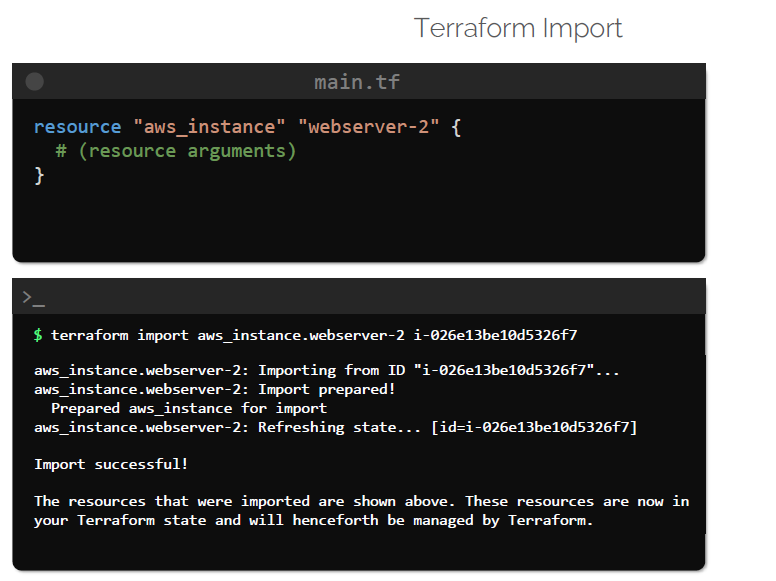


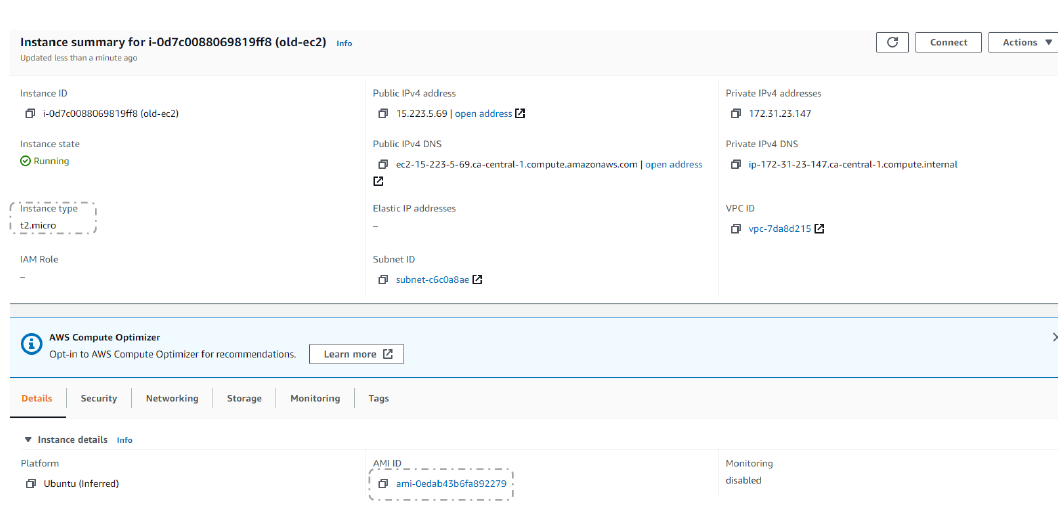


**Terraform Import:**

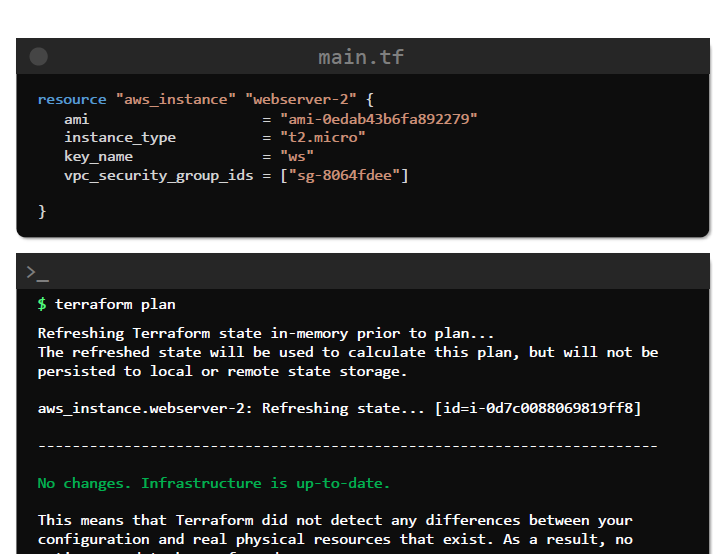










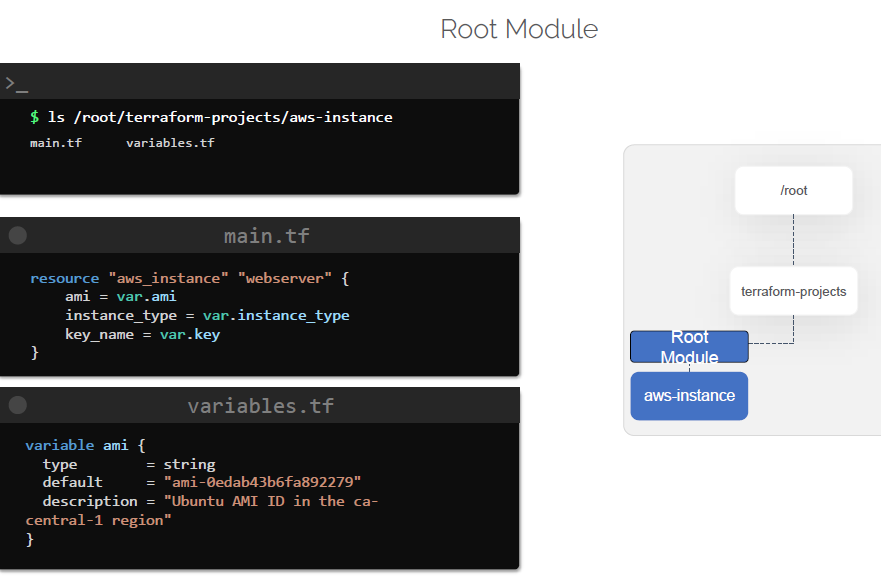


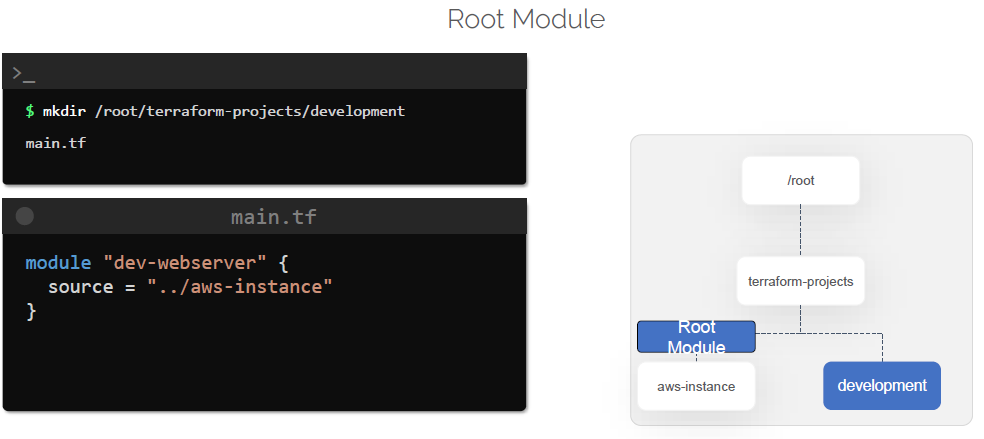
Terraform Modules:













Creating and Using Modules:

