**Dealing with different env: -**

1. tfvars
2. Workspasec
3. Different repos for different env (Dev, Prod)

Terraform workspace show – This command will show the workspace which we are in

Terraform workspace select dev – This command will select and open the dev env

Terraform workspace list – This will list all workspace list

Terraform workspace new prod – This will create new work space with prod

Terraform init -reconfigure – This will use once we are in one env (dev) and want to plan and deploy other env (prod) we need to use this command

terraform init -backend-config=dev/backend.tf – when we created new workspace with dev name and need to init we need to use this command

terraform plan -var-file=dev/dev.tfvars – This is used after the backend-config is done for dev workspace

terraform apply -var-file=dev/dev.tfvars – This is used to apply to create infra with dev

terraform destroy -var-file=prod/prod.tfvars – This is used to destroy infra from dev workspace

terraform init -reconfigure -backend-config=dev/backend.tf – If we are switching from prod workspace to dev workspace we need to use this command

**Provisioners : -**

Infra creation is taken care by terraform and configuration management is taken care by ansible, we need to integrate terraform and ansible with provisioners

**Terraform modules : -**

Reuse

DRY- Don’t repeat yourself

Code reuse , no need to write code for

Data types : - when we want to ask aws to provide details we will use data types to query the resources as below

data "aws\_availability\_zones" "azs" {

    #all\_availability\_zones = true

    state = "available"

}

In the above with data we are querying awz availability zones

Local.tf : - in local we can keep the common code and we can call it as below

locals {

  name = "${var.project\_name}-${var.environment}"

  az\_names = slice(data.aws\_availability\_zones.azs.names,0,2)

}

Local.name

Local.az\_names

Merge : -

We can merge any two tags as below

 tags = merge(

        var.common\_tags,

        var.vpc\_tags,

        {

            Name = local.name

        }

    )

Slice : -

With slice we can add the data fetched from aws and we can select the first two outputs provided by aws with 0,2 as in below command

az\_names = slice(data.aws\_availability\_zones.azs.names,0,2)