Terraform

>> Terraform uses 5000+ providers to make available via its releases. It makes the provider available with Provider block

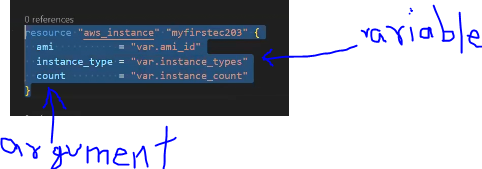
>> example with definition of terraform blocks mandatorily in use

Terraform commands

* Terraform fmt
* Terraform init
* Terraform validate
* Terraform plan
* Terraform apply –auto-approve
* Terraform destroy –auto-approve
* Terraform state list

Datatypes in Terraform

String, Number, Boolean, List, Map, Object, Tuples, Sets, Null



^^Argument name != variable name (instance\_type != instance\_types)

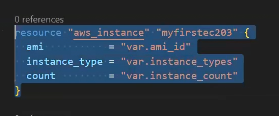
^^map datatype being used to specify arguments brackets

Steps to work with Terraform

>> Hardcoding of terraform code is not fit for SOLID principles

Goto VScode >> Folder “Day2\_ec2\_aws\_vpc\_valuesfromvariable”

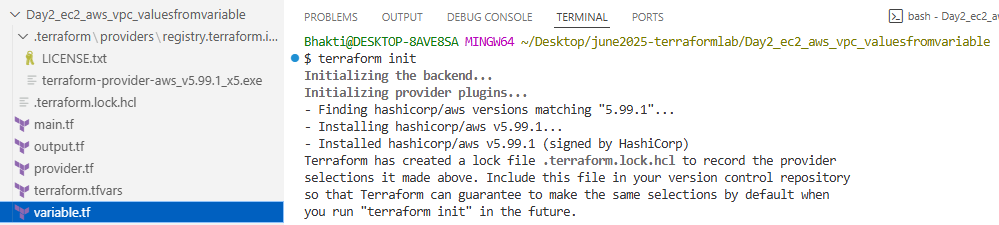
* File “main.tf”, “provider.tf” ......file-naming is not mandatory. Provider.tf is required to maintain file structure and ease to troubleshoot
* Add provider block & terraform block in provider.tf; resource block in main.tf
* File “variable.tf”, “output.tf”, “terraform.tfvars”
* Add variable block in variable.tf
* Add resource arguments in resource block



* Add all in terraform.tfvars
* Terminal cd to current folder

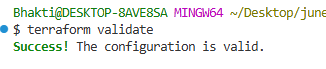


* Terraform fmt
* Terraform init ......validates tf files and installs terraform, creates terraform lockfiles

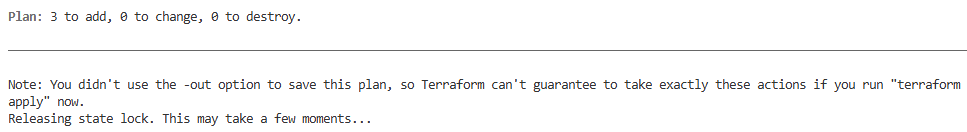


>> If you ever set or change modules or backend configuration for Terraform, rerun this command to reinitialize your working directory. If you forget, other rerun this command to reinitialize your working directory. If you forget, other commands will detect it and remind you to do so if necessary.

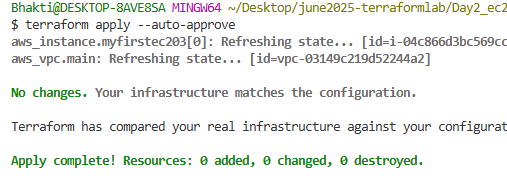
* Terraform validate ......needed when size goes beyond



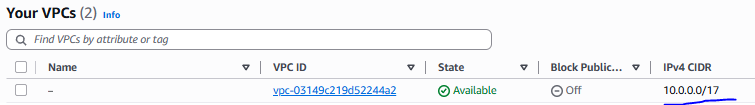
* Terraform plan



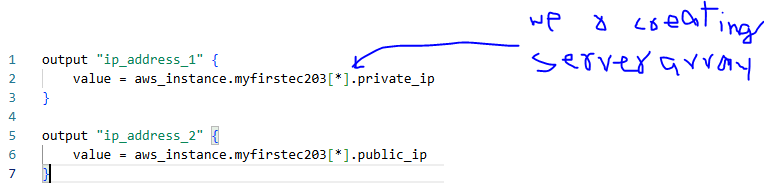
* Terraform state list ......infrastructure state
* Terraform apply –auto-approve



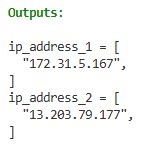
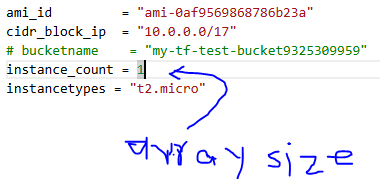




* Add output block to output.tf

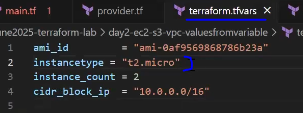
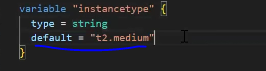


* Terraform apply –auto-approve ......output block reads output from recent applied changes

* Terraform destroy –auto-approve

>> Precedence challenges and how terraform overcomes them while calling **terraform.tfvars** and **variable.tf** file

* Terraform plan 

>> Picks terraform.tfvars value which is set as input. Leave the instancetypes blank then default t2.medium value is used which is shown in plan result

* Terraform plan without terraform.tfvars value 

>> Deleting the instancetypes value in both files results in interactive command prompt at terraform runtime which then reads keyboard input as var.instancetypes

>> Within industry tfvars file is independent at environment level. Utilize the file accordingly during plan

