Terraform-Day3

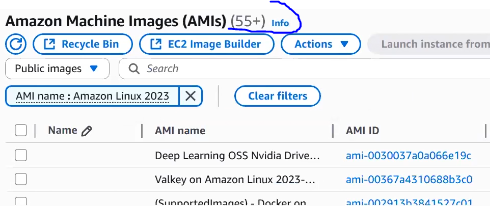
>> Terraform.tfstate becomes a local file at creation which gets transferred to remote backed S3. Tfstate file stores account information.

>> Tfstate file is stored in S3 when multiple users access the tfstate file. Each “Terraform apply” touches the tfstate file and locks that file during command execution, meaning no two users can execute the “terraform apply” command.

>> Always make it a habit to store tfstate file remote backend.

>> Understanding .gitignore

>> Data block in main.tf – make use of ready infrastructure and configure /read that config in our AWS infrastructure. At industry level, not hardcoding the ami value like practical ii. Is not enough, at industry level, like practical iv. Dynamically read the ami value



Practicals in terraform:

1. aws\_ec2\_aws\_vpc\_aws\_s3\_resource – main, resource stores hardcode of resource values
2. Day2\_ec2\_aws\_vpc\_valuesfromvariable – discontinue hardcode values. Tfvars stores the value of **var.** in variable
3. Backendblock – migrate tfstate file to remote backend
4. Datablock - ......critical & correct step

Goto >> VSCode >>Create folder datablock >> file provider.tf, main.tf, variable.tf, output.tf, terraform.tfvars

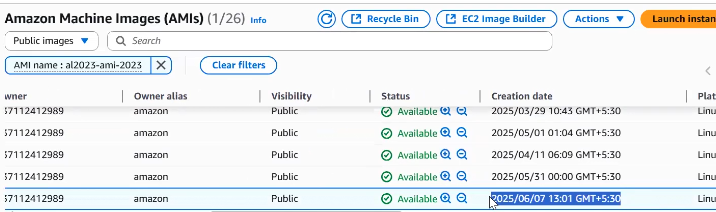


Table Creates a AMI based EC2 instance from the most recent image shown above having name al2023-ami-2023\*

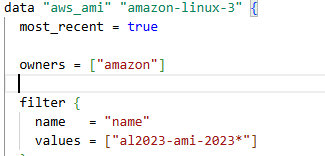
>> Add data {} block, resource {} block to main.tf with most\_recent argument

>> Add variable {} block to variable.tf

>> Add declatarations to terraform.tfvars

>> Add provider {} and terraform {} block to provide.tf

* Terraform fmt
* Terraform init
* Terraform plan ......note owners argument



* Terraform apply

