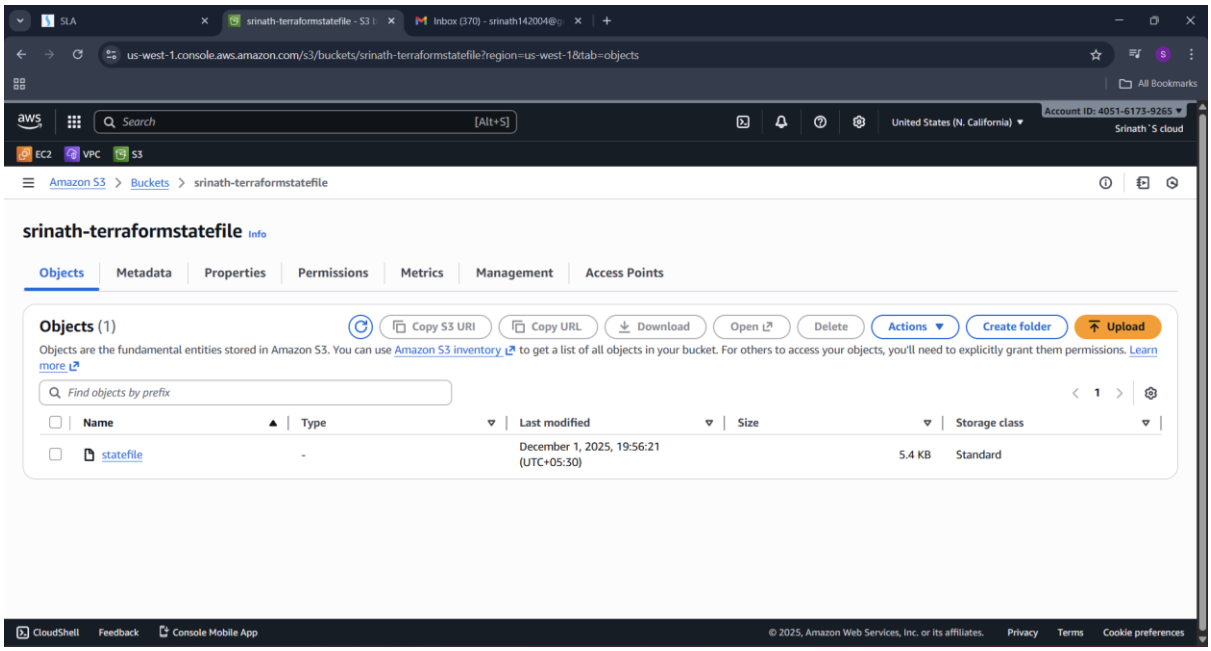
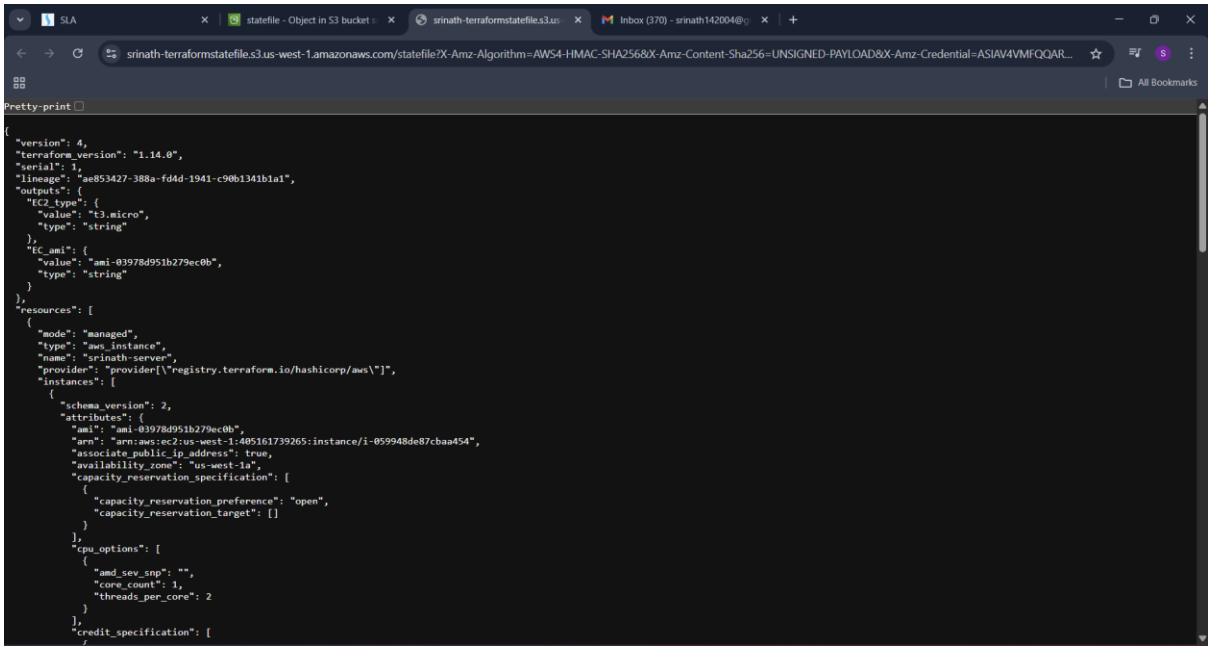
Task-9

Terraform S3 Backend:

The S3 backend in Terraform is a remote backend type used to store the Terraform state file in an Amazon S3 bucket. This approach is widely adopted for managing Terraform state in collaborative environments and CI/CD pipelines due to its benefits in terms of collaboration, state locking, versioning, and integration with AWS services.

Screenshots:





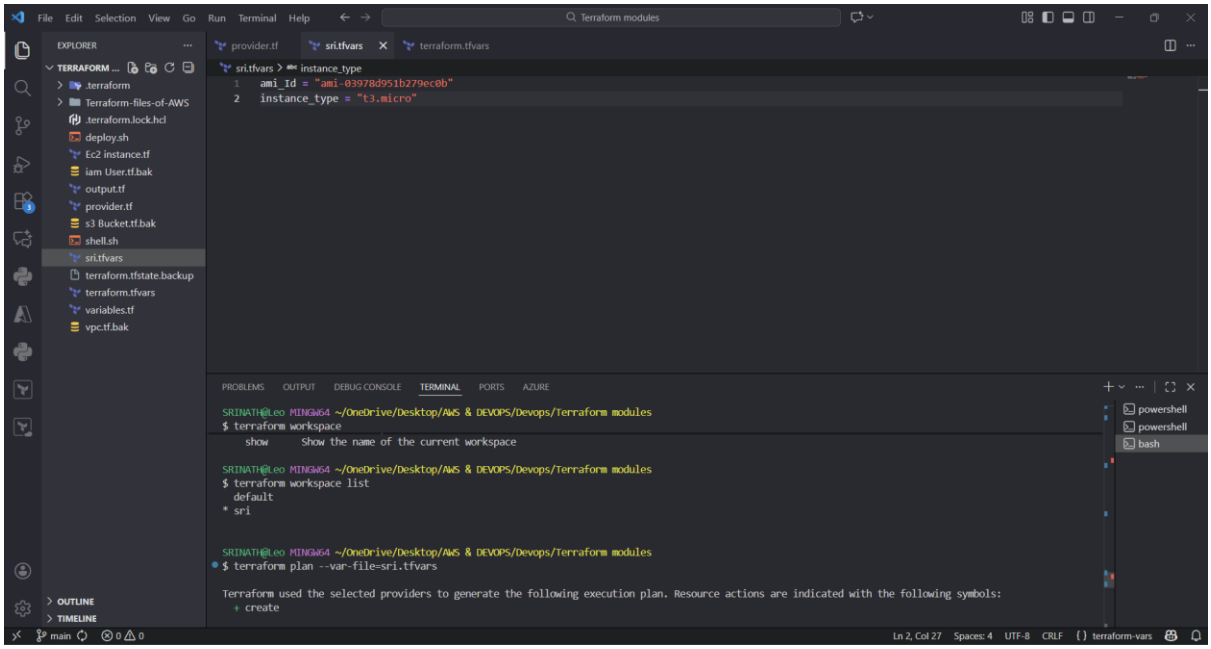
Terraform workspace:

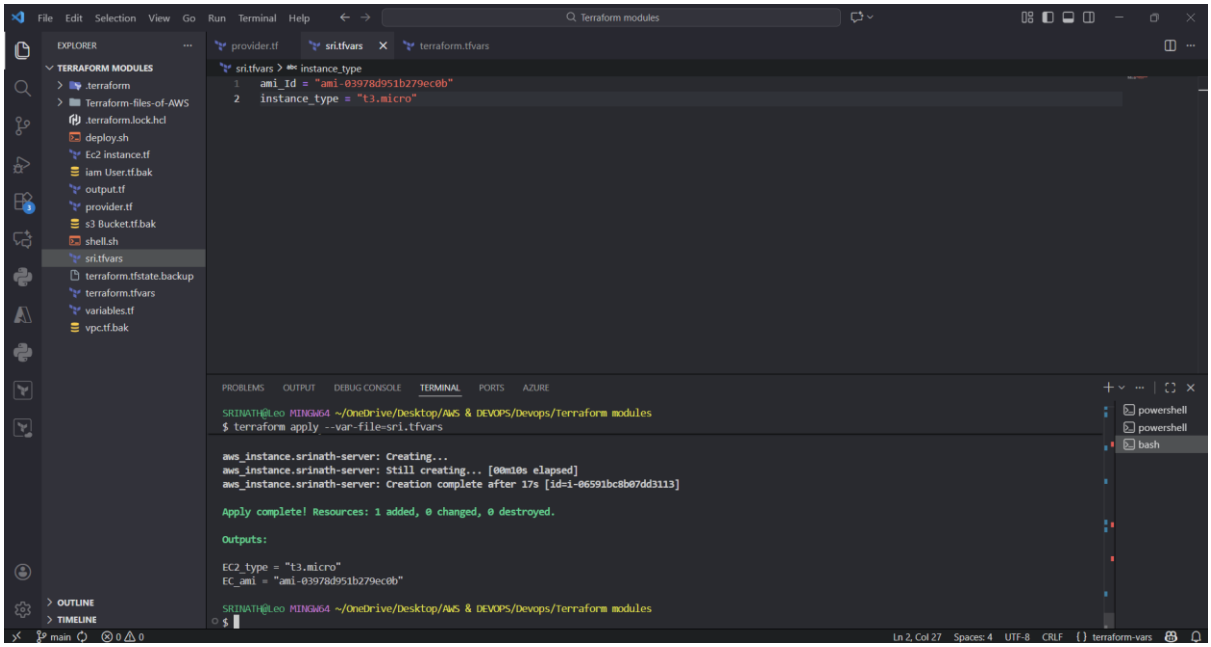
Terraform workspaces provide a mechanism to manage multiple, isolated instances of a Terraform configuration within the same working directory. Each workspace maintains its own state file, allowing for the deployment and management of separate environments (e.g., development, staging, production) using the same core infrastructure code.

Screenshots:

```
SRINATH@Leo MINGW64 ~/OneDrive/Desktop/AWS & DEVOPS/Devops/Terraform modules
$ terraform workspace new sri
Created and switched to workspace "sri"!
```

```
SRINATH@Leo MINGW64 ~/OneDrive/Desktop/AWS & DEVOPS/Devops/Terraform modules
$ terraform workspace list
  default
* sri
```


File Edit Selection View Go Run Terminal Help
Terraform modules
EXPLORER
TERRAFORM ...
.terraform
Terraform-files-of-AWS
.terraform.lock.hcl
deploy.sh
Ec2 instance.tf
iam User.tf.bak
output.tf
provider.tf
s3 Bucket.tf.bak
shell.sh
sri.tfvars
terraform.tfstate.backup
terraform.tfvars
variables.tf
vpc.tf.bak
provider.tf
sri.tfvars
terraform.tfvars
sri.tfvars > instance_type
1 ami_Id = "ami-03978d951b279ec0b"
2 instance_type = "t3.micro"
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
SRINATH@Leo MINGW64 ~/OneDrive/Desktop/AWS & DEVOPS/Devops/Terraform modules
$ terraform workspace
show Show the name of the current workspace
SRINATH@Leo MINGW64 ~/OneDrive/Desktop/AWS & DEVOPS/Devops/Terraform modules
$ terraform workspace list
default
* sri
SRINATH@Leo MINGW64 ~/OneDrive/Desktop/AWS & DEVOPS/Devops/Terraform modules
$ terraform plan --var-file=sri.tfvars
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create
powershell
powershell
bash
OUTLINE
TIMELINE
main
Ln 2, Col 27 Spaces: 4 UTF-8 CRLF terraform-vars


File Edit Selection View Go Run Terminal Help
Terraform modules
EXPLORER
TERRAFORM MODULES
.terraform
Terraform-files-of-AWS
.terraform.lock.hcl
deploy.sh
Ec2 instance.tf
iam User.tf.bak
output.tf
provider.tf
s3 Bucket.tf.bak
shell.sh
sri.tfvars
terraform.tfstate.backup
terraform.tfvars
variables.tf
vpc.tf.bak
provider.tf
sri.tfvars
terraform.tfvars
sri.tfvars > instance_type
1 ami_Id = "ami-03978d951b279ec0b"
2 instance_type = "t3.micro"
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
SRINATH@Leo MINGW64 ~/OneDrive/Desktop/AWS & DEVOPS/Devops/Terraform modules
$ terraform apply --var-file=sri.tfvars
aws_instance.srinath-server: Creating...
aws_instance.srinath-server: Still creating... [00m10s elapsed]
aws_instance.srinath-server: Creation complete after 17s [id=i-06591bc8b07dd3113]
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
Outputs:
EC2_type = "t3.micro"
EC_ami = "ami-03978d951b279ec0b"
SRINATH@Leo MINGW64 ~/OneDrive/Desktop/AWS & DEVOPS/Devops/Terraform modules
$
powershell
powershell
bash
OUTLINE
TIMELINE
main
Ln 2, Col 27 Spaces: 4 UTF-8 CRLF terraform-vars