

## Task-9

### Terraform S3 Backend:

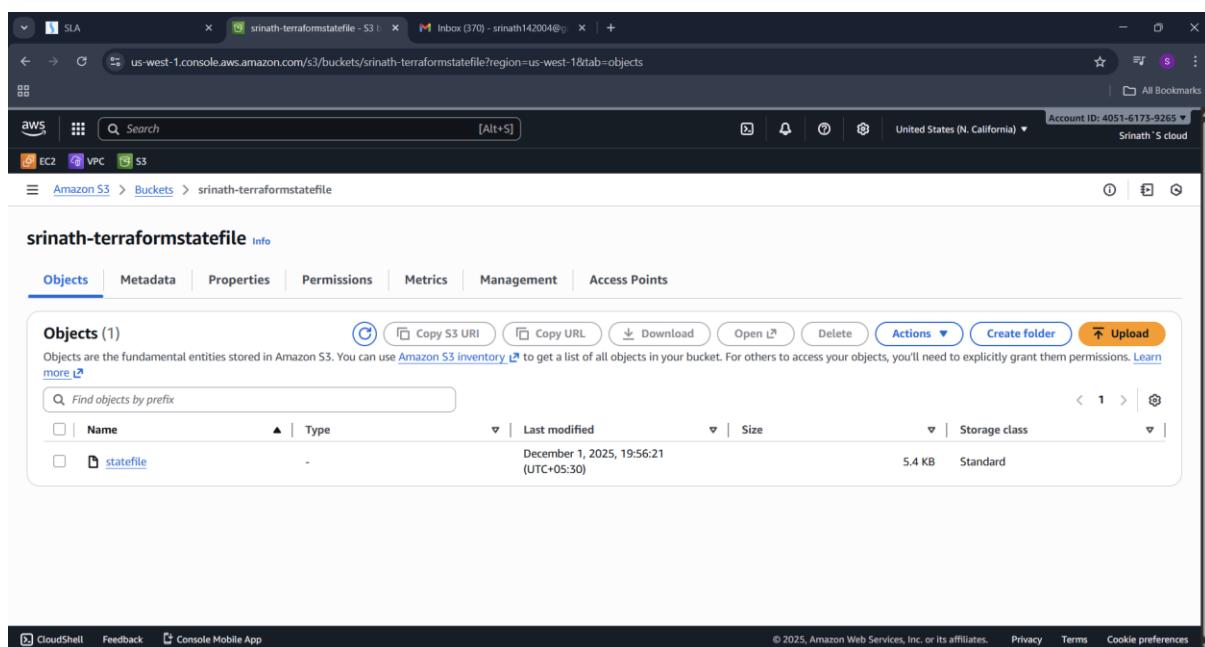
The S3 backend in Terraform is a remote backend type used to store the Terraform state file in an Amazon S3 bucket. This approach is widely adopted for managing Terraform state in collaborative environments and CI/CD pipelines due to its benefits in terms of collaboration, state locking, versioning,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

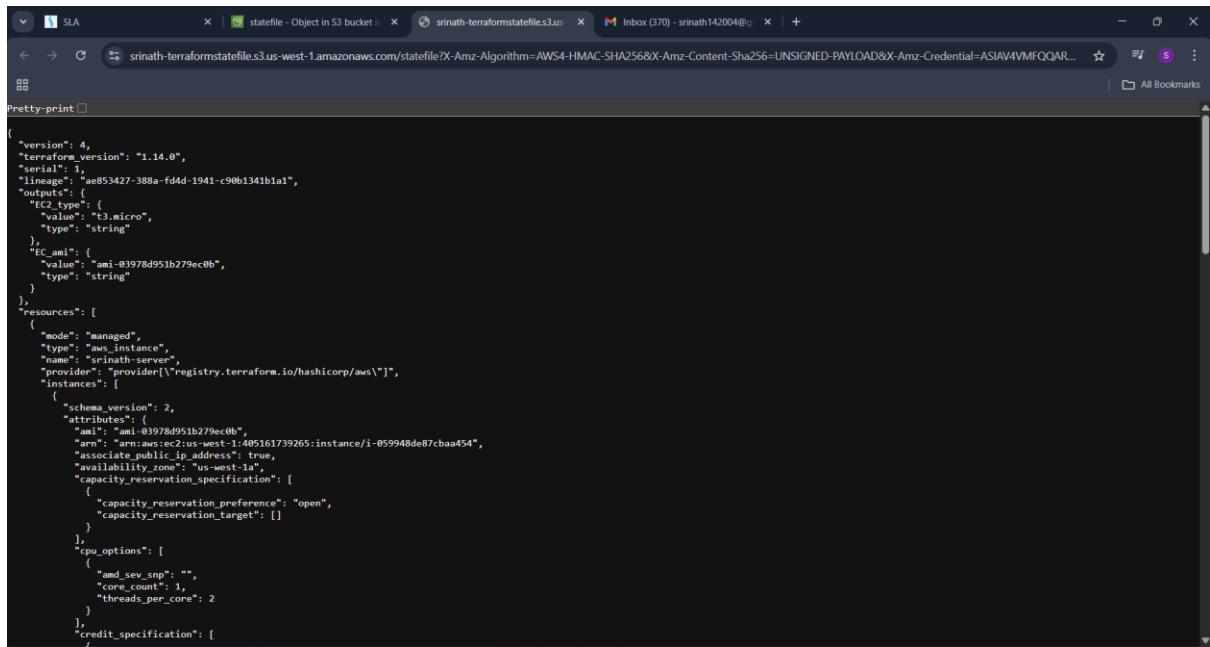
### Screenshots:



```
provider "aws" {
  region = "us-west-1"
}

terraform {
  backend "s3" {
    bucket = "statefile568765"
    key    = "statefile/terraform.tfstate"
    region = "us-west-1"
  }
}
```





A screenshot of a web browser displaying a Terraform state file. The file is a large JSON object with various nested properties. Key sections include "version", "terraform\_version", "serial", "lineage", "outputs", "resources", and "instances". The "instances" section contains detailed information about an AWS Lambda function, including its schema version, arn, associate\_public\_ip\_address, availability\_zone, capacity\_reservation\_specification, cpu\_options, and credit\_specification.

```
{
  "version": 4,
  "terraform_version": "1.14.0",
  "serial": 1,
  "lineage": "ae853427-388a-fd4d-1941-c90b1341b1a1",
  "outputs": {
    "EC2_type": {
      "value": "t3.micro",
      "type": "string"
    },
    "EC2 ami": {
      "value": "ami-03978a951b279ec0b",
      "type": "string"
    }
  },
  "resources": [
    {
      "mode": "managed",
      "type": "aws_lambda_function",
      "name": "srinath-server",
      "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",
      "instances": [
        {
          "schema_version": 2,
          "arn": "arn:aws:lambda:us-west-1:405161739265:function:i-059948de87cbaa454",
          "associate_public_ip_address": true,
          "availability_zone": "us-west-1a",
          "capacity_reservation_specification": [
            {
              "capacity_reservation_preference": "open",
              "capacity_reservation_target": []
            }
          ],
          "cpu_options": [
            {
              "end_sov": "",
              "core_count": 1,
              "threads_per_core": 2
            }
          ],
          "credit_specification": [
            ...
          ]
        }
      ]
    }
  ]
}
```

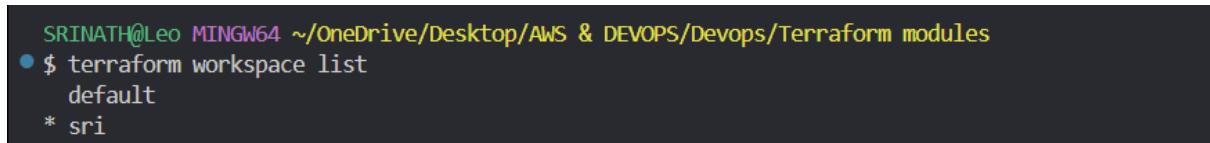
## Terraform workspace:

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
```

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER** view: Shows the project structure with files like provider.tf, sri.tvars, and terraform.tfvars.
- TERMINAL** tab:
  - Output from the terminal shows the current workspace: \$ terraform workspace show
  - Output from \$ terraform workspace list
  - Output from \$ terraform plan --var-file=sri.tvars
  - Information about the execution plan: Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols: + create
- SIDE BAR**: Shows a list of providers: powershell, powershell, and bash.

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER** view: Shows the project structure with files like provider.tf, sri.tvars, and terraform.tfvars.
- TERMINAL** tab:
  - Output from the terminal shows the execution of the apply command: \$ terraform apply --var-file=sri.tvars
  - Output from aws instance.srinath-server: Creating... and aws instance.srinath-server: Still creating... [00m10s elapsed]
  - Output from aws instance.srinath-server: Creation complete after 17s [id=i-06591bc8be7dd3113]
  - Output from Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
  - Output from Outputs:
    - EC2\_type = "t3.micro"
    - EC\_ami = "ami-03978d951b279ec0b"
- SIDE BAR**: Shows a list of providers: powershell, powershell, and bash.