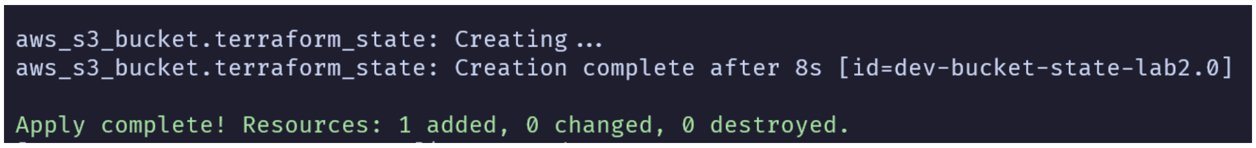
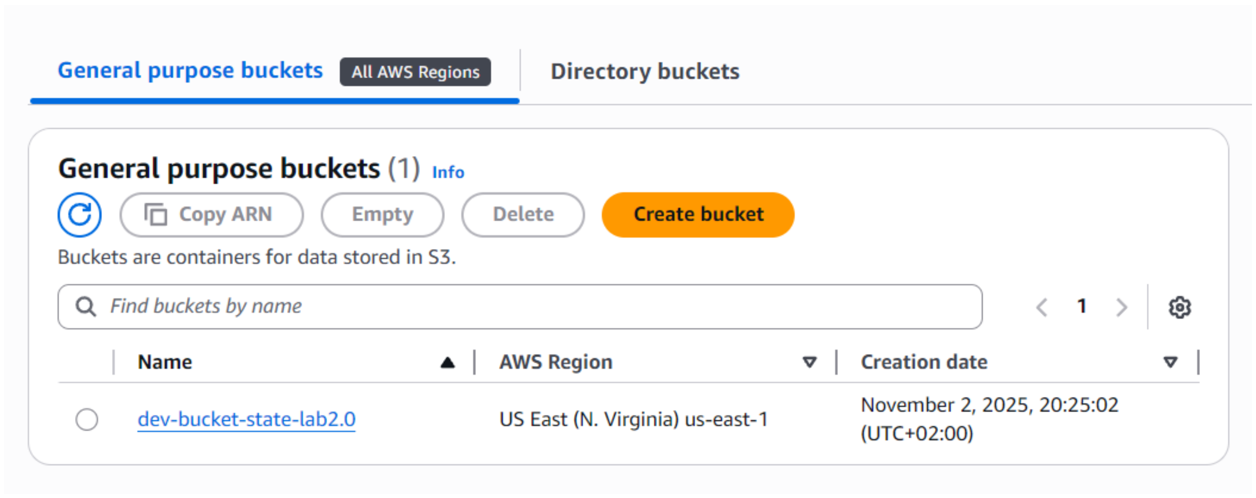
Day2

1- Creating s3 for state file by terraform



```
aws_s3_bucket.terraform_state: Creating...
aws_s3_bucket.terraform_state: Creation complete after 8s [id=dev-bucket-state-lab2.0]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

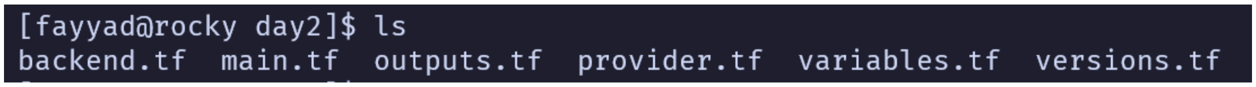


General purpose buckets (1)

Buckets are containers for data stored in S3.

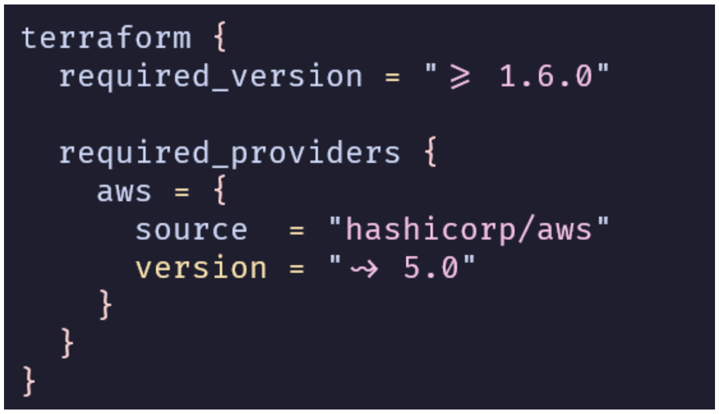
Name	AWS Region	Creation date
dev-bucket-state-lab2.0	US East (N. Virginia) us-east-1	November 2, 2025, 20:25:02 (UTC+02:00)

2- organize Terraform configurations into logical files (versions.tf, provider.tf, backend.tf, variables.tf, and main.tf).



```
[fayyad@rocky day2]$ ls
backend.tf  main.tf  outputs.tf  provider.tf  variables.tf  versions.tf
```

3- use the terraform block to define required providers and version constraints.



```
terraform {
  required_version = ">= 1.6.0"

  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}
```

4- use variables for EC2 instance configuration.

```
variable "aws_region" {
  description = "The AWS region to deploy the resources"
  type        = string
  default     = "us-east-1"
}

variable "instance_type" {
  description = "The EC2 instance type"
  type        = string
  default     = "t2.micro"
}

variable "instance_name" {
  description = "The Name tag for the EC2 instance"
  type        = string
  default     = "Terraform-Lab-EC2"
}

variable "instance_count" {
  description = "(Bonus) Number of EC2 instances to deploy"
  type        = number
  default     = 1
}
```

5- use a data source to dynamically retrieve the latest Amazon Linux 2 AMI ID.

```
# Data source to get the latest Amazon Linux 2 AMI

data "aws_ami" "amazon_linux_2" {
  most_recent = true
  owners      = ["amazon"]

  filter {
    name   = "name"
    values = ["amzn2-ami-hvm-*-x86_64-gp2"]
  }

  filter {
    name   = "virtualization-type"
    values = ["hvm"]
  }
}
```

6- configure an S3 remote backend with the use_lockfile flag for state management and locking.

```
terraform {
  backend "s3" {
    bucket       = "dev-bucket-state-lab2.0"
    key          = "ec2-deployment/terraform.tfstate"
    region       = "us-east-1"
    use_lockfile = "true"
  }
}
```

7- output important information such as instance public IP addresses.

```
output "instance_public_ips" {
  description = "Public IP addresses of the deployed EC2 instance(s)"
  value       = aws_instance.lab_instance.*.public_ip # Uses a splat expression to get all IPs
}

output "instance_ids" {
  description = "IDs of the deployed EC2 instance(s)"
  value       = aws_instance.lab_instance.*.id
}
```