Name: Maira Malik

Reg.no: 2023-BSE-040

Subject: Cloud Computing

LAB # 11

TASK 0:

1.

```
C:\Users\BOSS>gh repo create Maira222/Lab11 --public
GraphQL: Name already exists on this account (createRepository)

C:\Users\BOSS>gh codespace create --repo Maira222/Lab11
  Codespaces usage for this repository is paid for by Maira222
? Choose Machine Type: 2 cores, 8 GB RAM, 32 GB storage
turbo-sniffle-g4vqv6jwxqrjhw4g4

C:\Users\BOSS>gh codespace list
NAME                                DISPLAY NAME     REPOSITORY       BRANCH  STATE      CREATED AT
sturdy-sniffle-69xgxwq796q93rpp6    sturdy sniffle   Maira222/lab9    main*   Shutdown   about 12 hours ago
effective-lamp-q75q56jg7r6qf467w    effective lamp   Maira222/Lab10   main*   Available  about 1 hour ago
turbo-sniffle-g4vqv6jwxqrjhw4g4     turbo sniffle    Maira222/Lab11   main    Available  less than a minute ago
```

```
C:\Users\BOSS>gh codespace ssh -c turbo-sniffle-g4vqv6jwxqrjhw4g4
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-1030-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

@Maira222 /workspaces/Lab11 (main) $
```

TASK 1:

1.

```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
~
~
~
```

2.

```
@Maira222 /workspaces/Lab11 (main) $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.27.0...
- Installed hashicorp/aws v6.27.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
@Maira222 /workspaces/Lab11 (main) $
```

3.

```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials
}
variable "subnet_cidr_block" {
  type = string
}

output "subnet_cidr_block_output" {
  value = var.subnet_cidr_block
}
```

4.

```
@Maira222 /workspaces/Lab11 (main) $ terraform apply -auto-approve
subnet_cidr_blockvar.subnet_cidr_block
  Enter a value:
```

5.

```
@Maira222 /workspaces/Lab11 (main) $ terraform apply -auto-approve

Changes to Outputs:
  ~ subnet_cidr_block_output = "# it will prompt for the value of subnet_cidr_block" -> "10.0.0.0/24"

You can apply this plan to save these new output values to the Terraform state, without changing any real
infrastructure.

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

Outputs:

subnet_cidr_block_output = "10.0.0.0/24"
```

6.

```
@Maira222 /workspaces/Lab11 (main) $ export TF_VAR_subnet_cidr_block=10.0.20.0/24
@Maira222 /workspaces/Lab11 (main) $ terraform apply -auto-approve

Changes to Outputs:
  ~ subnet_cidr_block_output = "10.0.0.0/24" -> "10.0.20.0/24"

You can apply this plan to save these new output values to the Terraform state, without changing any real
infrastructure.

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

Outputs:

subnet_cidr_block_output = "10.0.20.0/24"
```

7.

```
@Maira222 /workspaces/Lab11 (main) $ vim terraform.tfvars
@Maira222 /workspaces/Lab11 (main) $ @Maira222 /workspaces/Lab11 (main) $ terraform apply -auto-approve

Changes to Outputs:
  ~ subnet_cidr_block_output = "10.0.20.0/24" -> "10.0.30.0/24"

You can apply this plan to save these new output values to the Terraform state, without changing any real
infrastructure.

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

Outputs:

subnet_cidr_block_output = "10.0.30.0/24"
```

8.

```
@Maira222 /workspaces/Lab11 (main) $ terraform apply -auto-approve -var "subnet_cidr_block=10.0.40.0/24"
# -var is highest precedence

Changes to Outputs:
  ~ subnet_cidr_block_output = "10.0.30.0/24" -> "10.0.40.0/24"

You can apply this plan to save these new output values to the Terraform state, without changing any real
infrastructure.

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

Outputs:

subnet_cidr_block_output = "10.0.40.0/24"
```

9.

```
@Maira222 /workspaces/Lab11 (main) $ printenv | grep TF_VAR_
_block
TF_VAR_subnet_cidr_block=10.0.20.0/24
@Maira222 /workspaces/Lab11 (main) $ unset TF_VAR_subnet_cidr_block
@Maira222 /workspaces/Lab11 (main) $ printenv | grep TF_VAR_
@Maira222 /workspaces/Lab11 (main) $
```

TASK 2:

1.

```
variable "subnet_cidr_block" {
  type        = string
  default     = ""
  description = "CIDR block to assign to the application subnet"
  sensitive   = false
  nullable    = false
  ephemeral   = false

  validation {
    condition     = can(regex("^([0-9]{1,3}\\.){3}[0-9]{1,3}/[0-9]+$", var.subnet_cidr_block))
    error_message = "The subnet_cidr_block must be a valid CIDR notation string, such as 10.0.0.0/24."
  }
}

output "subnet_cidr_block_output" {
  value = var.subnet_cidr_block
}
```

2.

```
@Maira222 /workspaces/Lab11 (main) $ terraform apply -auto-approve -var "subnet_cidr_block=10.0.0"

Error: Invalid value for variable

  on main.tf line 5:
   5: variable "subnet_cidr_block" {
    ├────────────────
    │ var.subnet_cidr_block is "10.0.0"

The subnet_cidr_block must be a valid CIDR notation string, such as 10.0.0.0/24.

This was checked by the validation rule at main.tf:13,3-13.
```

3.

```
}
variable "api_session_token" {
  type        = string
  default     = ""
  description = "Short-lived API session token used during apply operations"
  sensitive   = true
  nullable    = false
  ephemeral   = false

  validation {
    condition     = can(regex("^[A-Za-z0-9-_]{20,}$", var.api_session_token))
    error_message = "The API session token must be at least 20 characters and contain only
 underscores."
  }
}

output "api_session_token_output" {
  value     = var.api_session_token
  sensitive = true
}
```

4.

```
@Maira222 /workspaces/Lab11 (main) $ terraform apply -auto-approve -var "api_session_token=my_API_session_Token"

Changes to Outputs:
  + api_session_token_output = (sensitive value)
  ~ subnet_cidr_block_output = "10.0.40.0/24" -> "10.0.30.0/24"

You can apply this plan to save these new output values to the Terraform state, without changing any real
infrastructure.

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

Outputs:

api_session_token_output = <sensitive>
subnet_cidr_block_output = "10.0.30.0/24"
@Maira222 /workspaces/Lab11 (main) $
```

5.

```
@Maira222 /workspaces/Lab11 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 6,
  "lineage": "0eeac9cf-e1b8-84cb-afff-506e6e9716a1",
  "outputs": {
    "api_session_token_output": {
      "value": "my_API_session_Token",
      "type": "string",
      "sensitive": true
    },
```

6.

```
@Maira222 /workspaces/Lab11 (main) $ terraform apply -auto-approve

Error: Ephemeral value not allowed

  on main.tf line 37, in output "api_session_token_output":
  37:   value     = var.api_session_token

This output value is not declared as returning an ephemeral value, so it cannot be se
ephemeral value.

@Maira222 /workspaces/Lab11 (main) $
```

7.

```
@Maira222 /workspaces/Lab11 (main) $ terraform apply -auto-approve

Changes to Outputs:
  ~ api_session_token_output = (sensitive value)

You can apply this plan to save these new output values to the Terraform state, without changing any real infrastructure.

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

Outputs:

api_session_token_output = <sensitive>
subnet_cidr_block_output = "10.0.30.0/24"
```

TASK 3:

1.

```
}
variable "environment" {}
variable "project_name" {}
variable "primary_subnet_id" {}
variable "subnet_count" {}
variable "monitoring" {}
-- INSERT --
```

2.

```
@Maira222 /workspaces/Lab11 (main) $ aws ec2 describe-subnets   --filters "Name=availability-zone,V
"   --query "Subnets[].SubnetId"   --output text
subnet-0761224bffed3cb12
@Maira222 /workspaces/Lab11 (main) $
```

3.

```
subnet_cidr_block = "10.0.30.0/24"
environment = "dev"
project_name = "lab_work"
primary_subnet_id = "subnet-0761224bffed3cb12"
subnet_count = 3
monitoring = true
```

4.

```
locals {
  resource_name = "${var.project_name}-${var.environment}"
  primary_public_subnet = var.primary_subnet_id
  subnet_count          = var.subnet_count
  is_production         = var.environment == "prod"
  monitoring_enabled    = var.monitoring || local.is_production
}
```

5.

```
@Maira222 /workspaces/Lab11 (main) $ @Maira222 /workspaces/Lab11 (main) $ terraform apply -auto-approve

Changes to Outputs:
  + is_production           = false
  + monitoring_enabled      = true
  + primary_public_subnet   = "subnet-0761224bffed3cb12"
  + resource_name           = "lab_work-dev"
  + subnet_count            = 3

You can apply this plan to save these new output values to the Terraform state, without changing any real
infrastructure.

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

Outputs:

api_session_token_output = <sensitive>
is_production = false
monitoring_enabled = true
primary_public_subnet = "subnet-0761224bffed3cb12"
resource_name = "lab_work-dev"
subnet_cidr_block_output = "10.0.30.0/24"
subnet_count = 3
```

TASK 4:

1.

```
variable "tags" {
  type = map(string)
}

output "tags" {
  value = var.tags
}



-- INSERT --
```

2.

```
@Maira222 /workspaces/Lab11 (main) $ terraform apply -auto-approve

Changes to Outputs:
  + tags                    = {
      + Environment = "dev"
      + Owner       = "platform-team"
      + Project     = "sample-app"
    }

You can apply this plan to save these new output values to the Terraform state, without changing any real
infrastructure.

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

Outputs:

api_session_token_output = <sensitive>
is_production = false
monitoring_enabled = true
primary_public_subnet = "subnet-0761224bffed3cb12"
resource_name = "lab_work-dev"
subnet_cidr_block_output = "10.0.30.0/24"
subnet_count = 3
tags = tomap({
  "Environment" = "dev"
  "Owner" = "platform-team"
  "Project" = "sample-app"
})
@Maira222 /workspaces/Lab11 (main) $
```

3.

```
api_session_token_output = <sensitive>
is_production = false
monitoring_enabled = true
primary_public_subnet = "subnet-0761224bffed3cb12"
resource_name = "lab_work-dev"
server_config = {
  "backup_enabled" = false
  "instance_type" = "t3.micro"
  "monitoring" = true
  "name" = "web-server"
  "storage_gb" = 20
}
subnet_cidr_block_output = "10.0.30.0/24"
subnet_count = 3
tags = tomap({
  "Environment" = "dev"
  "Owner" = "platform-team"
```

TASK 5:

1.

```
variable "server_names" {
  type = list(string)
  default = ["web-2", "web-1", "web-2"]
}

variable "server_metadata" {
  type = tuple([string, number, bool])
  default = ["web-1", 4, true]
}

variable "availability_zones" {
  type = set(string)
  default = ["me-central-1b", "me-central-1a", "me-central-1b"]
}

output "compare_collections" {
  value = {
    list_example  = var.server_names
    tuple_example = var.server_metadata
    set_example   = var.availability_zones
  }
}
```

2.

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

Outputs:

api_session_token_output = <sensitive>
compare_collections = {
  "list_example" = tolist([
    "web-2",
    "web-1",
    "web-2",
  ])
  "set_example" = toset([
    "me-central-1a",
    "me-central-1b",
  ])
  "tuple_example" = [
    "web-1",
    4,
    true,
  ]
}
```

3.

```
locals {
  resource_name = "${var.project_name}-${var.environment}"
  primary_public_subnet = var.primary_subnet_id
  subnet_count          = var.subnet_count
  is_production         = var.environment == "prod"
  monitoring_enabled    = var.monitoring || local.is_production
  mutated_list  = setunion(var.server_names, ["web-3"])
  mutated_tuple = setunion(var.server_metadata, ["web-2"])
  mutated_set   = setunion(var.availability_zones, ["me-central-1c"])
}
~
~
```

4.

```
monitoring_enabled = true
mutation_comparison = {
  "mutated_tuple" = toset([
    "4",
    "true",
    "web-1",
    "web-2",
  ])
  "original_tuple" = [
    "web-1",
    4,
    true,
  ]
}
primary_public_subnet = "subnet-0761224b
resource_name = "lab_work-dev"
server_config = {
```

TASK 6:

1.

```
variable "optional_tag" {
  type        = string
  description = "A tag that may or may not be provided"
  default     = null
}
```

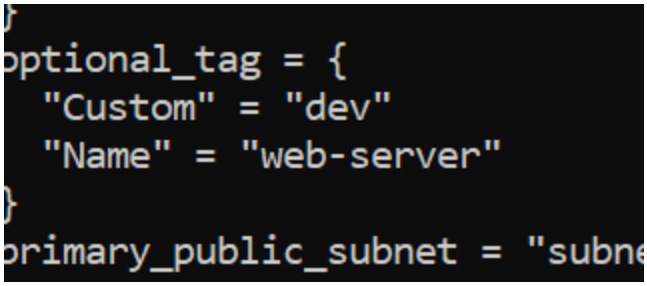
2.

```
server_tags = merge(
  { Name = "web-server" },
  var.optional_tag != null ? { Custom = var.optional_tag } : {}
)
```

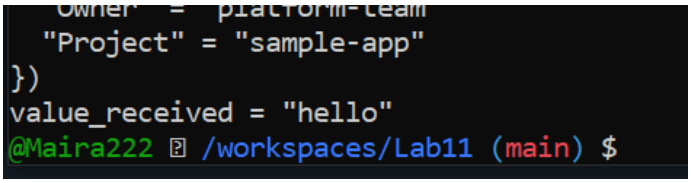
3.

```
  ]
}
optional_tag = {
  "Name" = "web-server"
}
primary_public_subnet = "subnet-0761224bffed3cb12"
resource_name = "lab_work-dev"
server_config = {
  "backup_enabled" = false
  "instance_type" = "t3.micro"
  "monitoring" = true
  "name" = "web-server"
  "storage_gb" = 20
}
subnet_cidr_block_output = "10.0.30.0/24"
subnet_count = 3
tags = tomap({
  "Environment" = "dev"
  "Owner" = "platform-team"
```

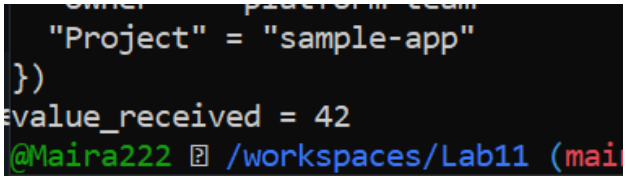
4.



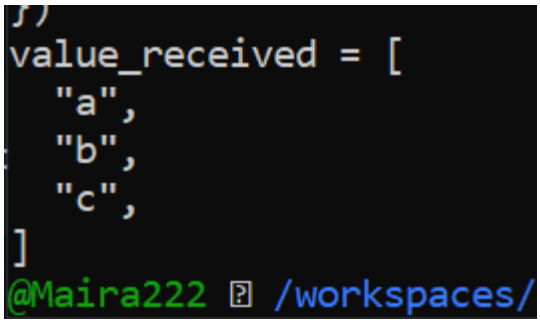
5.



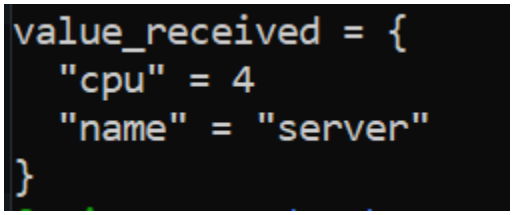
6.



7.

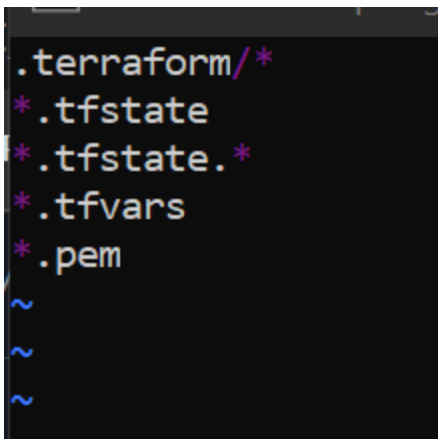


8.



TASK 7:

1.



TASK 8:

1.

```
@Maira222 /workspaces/Lab11 (main) $ ls -1
README.md
aws
awscliv2.zip
main.tf
terraform.tfstate
terraform.tfstate.backup
```

```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
```

2.

```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
variable "vpc_cidr_block" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
```

3.

```
variable "env_prefix" {}
resource "aws_vpc" "myapp_vpc" {
  cidr_block = var.vpc_cidr_block
  tags = {
    Name = "${var.env_prefix}-vpc"
  }
}
```

4.

```
resource "aws_subnet" "myapp_subnet_1" {
  vpc_id            = aws_vpc.myapp_vpc.id
  cidr_block        = var.subnet_cidr_block
  availability_zone = var.availability_zone
  tags = {
    Name = "${var.env_prefix}-subnet-1"
  }
}
```

5.

```
vpc_cidr_block     = "10.0.0.0/16"
subnet_cidr_block  = "10.0.10.0/24"
availability_zone  = "me-central-1a"
env_prefix         = "dev"
```

6.

```
@Maira222 /workspaces/Lab11 (main) $ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.27.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
@Maira222 /workspaces/Lab11 (main) $
```

```
@Maira222 /workspaces/Lab11 (main) $ terraform apply -auto-approve

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
  + create

Terraform will perform the following actions:

  # aws_subnet.myapp_subnet_1 will be created
  + resource "aws_subnet" "myapp_subnet_1" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "me-central-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.10.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = false
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + region                                         = "me-central-1"
      + tags                                           = {
          + "Name" = "dev-subnet-1"
        }
      + tags_all                                       = {
          + "Name" = "dev-subnet-1"
        }
      + vpc_id                                         = (known after apply)
    }

  # aws_vpc.myapp_vpc will be created
  + resource "aws_vpc" "myapp_vpc" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = (known after apply)
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
```

7.

```
resource "aws_internet_gateway" "myapp_igw" {
  vpc_id = aws_vpc.myapp_vpc.id
  tags = {
     Name = "${var.env_prefix}-igw"
  }
}

resource "aws_route_table" "myapp_route_table" {
  vpc_id = aws_vpc.myapp_vpc.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.myapp_igw.id
  }

  tags = {
     Name = "${var.env_prefix}-rt"
  }
}
-- INSERT --
```

```
  # aws_route_table.myapp_route_table will be created
  + resource "aws_route_table" "myapp_route_table" {
      + arn              = (known after apply)
      + id               = (known after apply)
      + owner_id         = (known after apply)
      + propagating_vgws = (known after apply)
      + region           = "me-central-1"
      + route            = [
          + {
              + cidr_block                = "0.0.0.0/0"
              + gateway_id                = (known after apply)
                # (11 unchanged attributes hidden)
            },
        ]
      + tags             = {
          + "Name" = "dev-rt"
        }
      + tags_all         = {
          + "Name" = "dev-rt"
        }
      + vpc_id           = "vpc-0e9a9afd68c42c19d"
    }

Plan: 2 to add, 0 to change, 0 to destroy.
aws_internet_gateway.myapp_igw: Creating...
aws_internet_gateway.myapp_igw: Creation complete after 1s [id=igw-09472db6257328c0d]
aws_route_table.myapp_route_table: Creating...
aws_route_table.myapp_route_table: Creation complete after 1s [id=rtb-04b6700356c394188]
```

8.

```
resource "aws_route_table_association" "a_rtb_subnet" {
  subnet_id      = aws_subnet.myapp_subnet_1.id
  route_table_id = aws_route_table.myapp_route_table.id
}
```

```
# aws_route_table_association.a_rtb_subnet will be created
+ resource "aws_route_table_association" "a_rtb_subnet" {
    + id             = (known after apply)
    + region         = "me-central-1"
    + route_table_id = "rtb-04b6700356c394188"
    + subnet_id      = "subnet-081a8591bf555c3b7"
  }

lan: 1 to add, 0 to change, 0 to destroy.
ws_route_table_association.a_rtb_subnet: Creating...
ws_route_table_association.a_rtb_subnet: Creation complete after 1
```

Clean up:

```
Plan: 0 to add, 0 to change, 5 to destroy.
aws_route_table_association.a_rtb_subnet: Destroying... [id=rtbassoc-0438ef2110ec31d16]
aws_route_table_association.a_rtb_subnet: Destruction complete after 0s
aws_subnet.myapp_subnet_1: Destroying... [id=subnet-081a8591bf555c3b7]
aws_route_table.myapp_route_table: Destroying... [id=rtb-04b6700356c394188]
aws_subnet.myapp_subnet_1: Destruction complete after 1s
aws_route_table.myapp_route_table: Destruction complete after 1s
aws_internet_gateway.myapp_igw: Destroying... [id=igw-09472db6257328c0d]
aws_internet_gateway.myapp_igw: Destruction complete after 0s
aws_vpc.myapp_vpc: Destroying... [id=vpc-0e9a9afd68c42c19d]
aws_vpc.myapp_vpc: Destruction complete after 1s
```

```
@Maira222 /workspaces/Lab11 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 32,
  "lineage": "0eeac9cf-e1b8-84cb-afff-506e6e9716a1",
  "outputs": {},
  "resources": [],
  "check_results": null
}
```

```
@Maira222 /workspaces/Lab11 (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 26,
  "lineage": "0eeac9cf-e1b8-84cb-afff-506e6e9716a1",
  "outputs": {},
  "resources": [
    {
      "mode": "managed",
      "type": "aws_internet_gateway",
      "name": "myapp_igw",
      "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",
      "instances": [
        {
          "schema_version": 0,
          "attributes": {
            "arn": "arn:aws:ec2:me-central-1:737230811842:internet-gatewa
```