

# FATIMA JINNAH WOMEN UNIVERSITY

*Department of Software Engineering*



---

## LAB #11

**SUBJECT: CLOUD COMPUTING**

**SUBMITTED TO: SIR MUHAMMAD SHOAIB**

**SUBMITTED BY: UMBER QASIM**

**REGISTRATION NO: 2023-BSE-066**

**CLASS: BSSE V-B**

## GH CLI Codespaces + AWS + Terraform: Variables, Collections, Sensitivity & EC2 Provisioning

### Task#0: Lab Setup (Codespace & GH CLI)

 Select Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\HP> gh --version
gh version 2.83.1 (2025-11-13)
https://github.com/cli/cli/releases/tag/v2.83.1
PS C:\Users\HP> gh auth login -s codespace
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 8EB5-91CA
Press Enter to open https://github.com/login/device in your browser...
② Authentication complete.
- gh config set -h github.com git_protocol https
③ Configured git protocol
③ Logged in as Umber-qasim
! You were already logged in to this account
PS C:\Users\HP> gh repo create CC_UmberQasim_066/Lab11 --public
HTTP 404: Not Found (https://api.github.com/users/CC_UmberQasim_066)
PS C:\Users\HP> gh repo create CC-UmberQasim-066/Lab11 --public
HTTP 404: Not Found (https://api.github.com/users/CC-UmberQasim-066)
PS C:\Users\HP> gh repo create Umber-qasim/Lab11 --public
③ Created repository Umber-qasim/Lab11 on github.com
  https://github.com/Umber-qasim/Lab11
PS C:\Users\HP>
```

```
PS C:\Users\HP> gh codespace create --repo Umber-qasim/Lab11
? Codespace usage for this repository is paid for by Umber-qasim
? Choose Machine Type: 2 cores, 8 GB RAM, 32 GB storage
shiny-doodle-4jpqqqr9p5pvc55p9
PS C:\Users\HP> gh codespace list
NAME          DISPLAY NAME      REPOSITORY      BRANCH STATE    CREATED AT
turbo-space-pancake-g4vxxxpwvgvxfv5xr  turbo space pancake  Umber-qasim/Lab9  main*  Shutdown  about 11 days ago
shiny-doodle-4jpqqqr9p5pvc55p9        shiny doodle       Umber-qasim/Lab11  main   Available less than a minute ago
PS C:\Users\HP> gh codespace ssh -c shiny-doodle-4jpqqqr9p5pvc55p9
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.

Umber-qasim @ /workspaces/Lab11 (main) $
```

### Task#01: Provider & Basic variable (variable precedence)

```
@Umber-qasim ② /workspaces/Lab11 (main) $ touch main.tf
```

#### ➤ Windows PowerShell

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

#### ➤ Windows PowerShell

```
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
}
```

```
@Umber-qasim ② /workspaces/Lab11 (main) $ terraform apply -auto-approve
var.subnet_cidr_block
  Enter a value:
```

#### ➤ Windows PowerShell

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

variable "subnet_cidr_block" {
  type    = string
  default = "10.0.0.0/24"
}

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

```
@Umber-qasim ② /workspaces/Lab11 (main) $ terraform apply -auto-approve

Changes to Outputs:
  ~ subnet_cidr_block_output = "" -> "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"
```

```
@Umber-qasim ② /workspaces/Lab11 (main) $ export TF_VAR_subnet_cidr_block=10.0.20.0/24
```

```
@Umber-qasim ② /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"
```

```
@lumber-qasim ② /workspaces/Lab11 (main) $ touch terraform.tfvars  
@lumber-qasim ② /workspaces/Lab11 (main) $ vim terraform.tfvars
```

#### Windows PowerShell

```
subnet_cidr_block = "10.0.30.0/24"  
~
```

```
@lumber-qasim ② /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"
```

```
@lumber-qasim ② /workspaces/Lab11 (main) $ terraform apply -auto-approve -var "subnet_cidr_block=10.0.40.0/24"
```

```
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"
```

```
@lumber-qasim ② /workspaces/Lab11 (main) $ printenv | grep TF_VAR_  
TF_VAR_subnet_cidr_block=10.0.20.0/24  
@lumber-qasim ② /workspaces/Lab11 (main) $ unset TF_VAR_subnet_cidr_block  
@lumber-qasim ② /workspaces/Lab11 (main) $ printenv | grep TF_VAR_
```

## Task#02: Variable validation & sensitive / ephemeral variables

#### Windows PowerShell

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

```
@lumber-qasim ② /workspaces/Lab11 (main) $ terraform apply -auto-approve -var "subnet_cidr_block=10.0.0"  
  
Changes to Outputs:  
  ~ subnet_cidr_block_output = "10.0.40.0/24" -> null  
  
You can apply this plan to save these new output values to the Terraform state, without changing any real infrastructure.  
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.  
@lumber-qasim ② /workspaces/Lab11 (main) $
```

```

[+] Windows PowerShell
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
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."
  }
}
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 letters, numbers, hyphens, or underscores."
  }
}
output "api_session_token_output" {
  value     = var.api_session_token
  sensitive = true
}

```

```

Almas-Qasim @ /workspaces/labil (main) $ terraform apply -auto-approve -var "api_session_token=my_API_session_Token_2024" -var "subnet_cidr_block=10.0.0.0/24"
Changes to Outputs:
  + api_session_token_output = (sensitive value)
  + subnet_cidr_block_output = "10.0.40.0/24" -> null

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>

```

```

Almas-Qasim @ /workspaces/labil (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 6,
  "lineage": "84f4460b-2ec2-a377-de08-491ce9dbcb8e",
  "outputs": {
    "api_session_token_output": {
      "value": "my_API_session_Token_2024",
      "type": "string",
      "sensitive": true
    }
  },
  "resources": [],
  "check_results": [
    {
      "object_kind": "var",
      "config_addr": "var.subnet_cidr_block",
      "status": "pass",
      "objects": [
        {
          "object_addr": "var.subnet_cidr_block",
          "status": "pass"
        }
      ]
    },
    {
      "object_kind": "var",
      "config_addr": "var.api_session_token",
      "status": "pass",
      "objects": [
        {
          "object_addr": "var.api_session_token",
          "status": "pass"
        }
      ]
    }
  ]
}

```

```
Qumber-qasim @ /workspaces/LabII (main) $ terraform apply -auto-approve
Error: Ephemeral value not allowed

  on main.tf line 33, in output "api_session_token_output":
  33:   value      = var.api_session_token

This output value is not declared as returning an ephemeral value, so it cannot be set to a result derived from an ephemeral value.
```

```
Windows PowerShell
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
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."
  }
}
variable "api_session_token" {
  type     = string
  default  = "my_API_Session_Token_Valid_Length_123"
  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 letters, numbers, hyphens, or underscores."
  }
}

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

```
Qumber-qasim @ /WORKSPACES/LabII (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>
```

## Task#03: Project-level variables, locals, and outputs

```
➤ Windows PowerShell
provider "aws" {
    shared_config_files      = ["~/.aws/config"]
    shared_credentials_files = ["~/.aws/credentials"]
}
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}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}$", var.subnet_cidr_block))
        error_message = "The subnet_cidr_block must be a valid CIDR notation string, such as 10.0.0.0/24."
    }
}
variable "api_session_token" {
    type     = string
    default  = "my_API_session_Token_Valid_Length_123"
    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 letters, numbers, hyphens, or underscores."
    }
}
output "api_session_token_output" {
    value     = var.api_session_token
    sensitive = true
}
variable "environment" {}
variable "project_name" {}
variable "primary_subnet_id" {}
variable "subnet_count" {}
variable "monitoring" [ ]
```

```
@Umber-qasim ② /workspaces/Lab11 (main) $ aws ec2 describe-subnets \
> --filters "Name=availability-zone,Values=me-central-1a" \
> --query "Subnets[].SubnetId" \
> --output text
subnet-003aaaf45b7679f681
```

```
➤ Windows PowerShell
subnet_cidr_block = "10.0.30.0/24"
environment       = "dev"
project_name      = "lab_work"
primary_subnet_id = "subnet-003aaaf45b7679f681"
subnet_count      = 3
monitoring        = true
```

```
➤ Windows PowerShell
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
```

```
Windows PowerShell

validation {
    condition   = can(regex("^(0-9)(1,3)\\.(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."
}

variable "api_session_token" {
    type         = string
    default      = "my_API_session_Token_Valid_Length_123"
    description  = "ShortLived 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 letters, numbers, hyphens, or underscores."
    }
}

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

variable "environment" {}
variable "project_name" {}
variable "primary_subnet_id" {}
variable "subnet_count" {}
variable "monitoring" {}

output "resource_name" {
    value = local.resource_name
}

output "primary_public_subnet" {
    value = local.primary_public_subnet
}

output "subnet_count" {
    value = local.subnet_count
}

output "is_production" {
    value = local.is_production
}

output "monitoring_enabled" {
    value = local.monitoring_enabled
}
```

```
zaiden-qasim @ /workspaces/Labil (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.
zaiden-qasim @ /workspaces/Labil (main) $ terraform apply -auto-approve

Changes to Outputs:
+ is_production          = false
+ monitoring_enabled     = true
+ primary_public_subnet  = "subnet-003aaf45b7679f681"
+ 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-003aaf45b7679f681"
resource_name = "lab_work-dev"
subnet_count = 3
zaiden-qasim @ /workspaces/Labil (main) $
```

## Task#04: Maps and Objects

### PART 1: Map Variable (tags)

```
Windows PowerShell
type      = string
default   = "my_API_session_Token_Valid_Length_123"
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 letters, numbers, hyphens, or underscores."
}
}

output "api_session_token_output" {
    value   = var.api_session_token
    sensitive = true
}
variable "environment" {}
variable "project_name" {}
variable "primary_subnet_id" {}
variable "subnet_count" {}
variable "monitoring" {}
output "resource_name" {
    value = local.resource_name
}

output "primary_public_subnet" {
    value = local.primary_public_subnet
}

output "subnet_count" {
    value = local.subnet_count
}

output "is_production" {
    value = local.is_production
}

output "monitoring_enabled" {
    value = local.monitoring_enabled
}
variable "tags" {
    type = map(string)
}

output "tags" {
    value = var.tags
}
```

```
@Umber-gasim @ /workspaces/LabII (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-003aaaf45b7679f681"
resource_name = "lab_work-dev"
subnet_count = 3
tags = tomap({
  "Environment" = "dev"
  "Owner" = "platform-team"
  "Project" = "sample-app"
})
```

### PART 2: Object Variable (server\_config)

```
> Windows PowerShell
output "api_session_token_output" {
    value      = var.api_session_token
    sensitive = true
}
variable "environment" {}
variable "project_name" {}
variable "primary_subnet_id" {}
variable "subnet_count" {}
variable "monitoring" {}
output "resource_name" {
    value = local.resource_name
}

output "primary_public_subnet" {
    value = local.primary_public_subnet
}

output "subnet_count" {
    value = local.subnet_count
}

output "is_production" {
    value = local.is_production
}

output "monitoring_enabled" {
    value = local.monitoring_enabled
}
variable "tags" {
    type = map(string)
}

output "tags" {
    value = var.tags
}
variable "server_config" {
    type = object({
        name          = string
        instance_type = string
        monitoring    = bool
        storage_gb    = number
        backup_enabled = bool
    })
}
output "server_config" {
    value = var.server_config
}
```

```
> Windows PowerShell
subnet_cidr_block = "10.0.30.0/24"
environment        = "dev"
project_name       = "lab_work"
primary_subnet_id = "subnet-003aaaf45b7679f681"
subnet_count       = 3
monitoring         = true
tags = {
    Environment = "dev"
    Project     = "sample-app"
    Owner       = "platform-team"
}
server_config = {
    name          = "web-server"
    instance_type = "t3.micro"
    monitoring    = true
    storage_gb    = 20
    backup_enabled = false
}
```

```

dumber+qasim @ /workspaces/Lab01 (main) $ terraform apply -auto-approve

Changes to Outputs:
+ server_config          = {
    + backup_enabled = false
    + instance_type = "t3.micro"
    + monitoring    = true
    + name          = "web-server"
    + storage_gb   = 20
  }

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-003aaaf45b7679f681"
resource_name = "lab_work-dev"
server_config = {
  "backup_enabled" = false
  "instance_type" = "t3.micro"
  "monitoring" = true
  "name" = "web-server"
  "storage_gb" = 20
}
subnet_count = 3
tags = tomap({
  "Environment" = "dev"
  "Owner" = "platform-team"
  "Project" = "sample-app"
})

```

## Task#05: Collections: list, tuple, set & mutation via locals

### PART 1: Define Collections

```

➤ Windows PowerShell
value = local.is_production
}

output "monitoring_enabled" {
  value = local.monitoring_enabled
}
variable "tags" {
  type = map(string)
}

output "tags" {
  value = var.tags
}
variable "server_config" {
  type = object({
    name      = string
    instance_type = string
    monitoring    = bool
    storage_gb   = number
    backup_enabled = bool
  })
}

output "server_config" {
  value = var.server_config
}
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
  }
}

```

## PART 2: Observe behavior of Collections

```
Windows PowerShell
Pomber-qasim @ /workspaces/Lab11 (main) $ terraform apply -auto-approve

Changes to Outputs:
+ compare_collections      = {
    + list_example   = [
        + "web-2",
        + "web-1",
        + "web-2",
    ]
    + set_example    = [
        + "me-central-1a",
        + "me-central-1b",
    ]
    + tuple_example = [
        + "web-1",
        + 4,
        + true,
    ]
}

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>
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,
  ]
}
is_production = false
monitoring_enabled = true
primary_public_subnet = "subnet-003aaaf45b7679f681"
resource_name = "lab_work-dev"
server_config = {
  "backup_enabled" = false
  "instance_type" = "t3.micro"
}
```

## PART 3: Mutations via locals

```
Windows PowerShell
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
}
locals {
  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"])
}
```

## PART 4: Mutation comparison output

➤ Windows PowerShell

```
type = map(string)
}

output "tags" {
    value = var.tags
}
variable "server_config" {
    type = object({
        name      = string
        instance_type = string
        monitoring   = bool
        storage_gb   = number
        backup_enabled = bool
    })
}

output "server_config" {
    value = var.server_config
}
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
    }
}

output "mutation_comparison" {
    value = {
        original_tuple = var.server_metadata
        mutated_tuple  = local.mutated_tuple
    }
}
```

## ➤ Windows PowerShell

```
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,
    ]
}
is_production = false
monitoring_enabled = true
mutation_comparison = {
    "mutated_tuple" = toset([
        "4",
        "true",
        "web-1",
        "web-2",
    ])
    "original_tuple" = [
        "web-1",
        4,
        true,
    ]
}
primary_public_subnet = "subnet-003aaaf45b7679f681"
resource_name = "lab_work-dev"
server_config = {
    "backup_enabled" = false
    "instance_type" = "t3.micro"
    "monitoring" = true
    "name" = "web-server"
    "storage_gb" = 20
}
subnet_count = 3
tags = tomap({
    "Environment" = "dev"
    "Owner" = "platform-team"
    "Project" = "sample-app"
})
@Umber-qasim ② /workspaces/Lab11 (main) $
```

## Task#06: Null, any type & dynamic values

### PART 1: Null variable (**optional\_tag**)

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

## PART 2: Merge logic in locals.tf

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

## PART 3: Add Output

```
output "optional_tag" {
  value = local.server_tags
}
```

## PART 4: Apply WITHOUT optional\_tag

```
➤ Windows PowerShell
subnet_cidr_block = "10.0.30.0/24"
environment        = "dev"
project_name       = "lab_work"
primary_subnet_id = "subnet-003aaf45b7679f681"
subnet_count       = 3
monitoring         = true
tags = {
  Environment = "dev"
  Project     = "sample-app"
  Owner       = "platform-team"
}
server_config = {
  name          = "web-server"
  instance_type = "t3.micro"
  monitoring    = true
  storage_gb    = 20
  backup_enabled = false
}
```

```

➤ Windows PowerShell
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,
    ]
}
is_production = false
monitoring_enabled = true
mutation_comparison = {
    "mutated_tuple" = toset([
        "4",
        "true",
        "web-1",
        "web-2",
    ])
    "original_tuple" = [
        "web-1",
        4,
        true,
    ]
}
optional_tag = {
    "Name" = "web-server"
}
primary_public_subnet = "subnet-003aaf45b7679f681"
resource_name = "lab_work-dev"
server_config = {
    "backup_enabled" = false
    "instance_type" = "t3.micro"
    "monitoring" = true
    "name" = "web-server"
    "storage_gb" = 20
}
subnet_count = 3
tags = tomap({
    "Environment" = "dev"
    "Owner" = "platform-team"
    "Project" = "sample-app"
})
@Umber-qasim ② /workspaces/Lab11 (main) $
```

## PART 5: Apply WITH optional\_tag

```

➤ Windows PowerShell
subnet_cidr_block = "10.0.30.0/24"
environment      = "dev"
project_name     = "lab_work"
primary_subnet_id = "subnet-003aaf45b7679f681"
subnet_count     = 3
monitoring       = true
tags = {
    Environment = "dev"
    Project     = "sample-app"
    Owner       = "platform-team"
}
server_config = {
    name          = "web-server"
    instance_type = "t3.micro"
    monitoring    = true
    storage_gb   = 20
    backup_enabled = false
}
optional_tag = "dev"
```

```

➤ Windows PowerShell
"list_example" = tolist([
    "web-2",
    "web-1",
    "web-2",
])
"set_example" = toset([
    "me-central-1a",
    "me-central-1b",
])
"tuple_example" = [
    "web-1",
    4,
    true,
]
}
is_production = false
monitoring_enabled = true
mutation_comparison = {
    "mutated_tuple" = toset([
        "4",
        "true",
        "web-1",
        "web-2",
    ])
    "original_tuple" = [
        "web-1",
        4,
        true,
    ]
}
optional_tag = {
    "Custom" = "dev"
    "Name" = "web-server"
}
primary_public_subnet = "subnet-003aaf45b7679f681"
resource_name = "lab_work-dev"
server_config = {
    "backup_enabled" = false
    "instance_type" = "t3.micro"
    "monitoring" = true
    "name" = "web-server"
    "storage_gb" = 20
}
subnet_count = 3
tags = tomap({
    "Environment" = "dev"
    "Owner" = "platform-team"
    "Project" = "sample-app"
})
@Umber-qasim @ /workspaces/Lab11 (main) $
```

## PART 6: any type variable (dynamic\_value)

```

variable "dynamic_value" {
    type      = any
    description = "A variable that can accept any data type"
    default    = null
}

output "value_received" {
    value = var.dynamic_value
}
```

## Dynamic Testing in terraform.tfvars:

### **String test:**

```
dynamic_value = "hello"
```

#### ➤ Windows PowerShell

```
"web-2",
"web-1",
"web-2",
])
"set_example" = toset([
"me-central-1a",
"me-central-1b",
])
"tuple_example" = [
"web-1",
4,
true,
]
}
is_production = false
monitoring_enabled = true
mutation_comparison = {
"mutated_tuple" = toset([
"4",
"true",
"web-1",
"web-2",
])
"original_tuple" = [
"web-1",
4,
true,
]
}
optional_tag = {
"Custom" = "dev"
"Name" = "web-server"
}
primary_public_subnet = "subnet-003aaaf45b7679f681"
resource_name = "lab_work-dev"
server_config = {
"backup_enabled" = false
"instance_type" = "t3.micro"
"monitoring" = true
"name" = "web-server"
"storage_gb" = 20
}
subnet_count = 3
tags = tomap({
"Environment" = "dev"
"Owner" = "platform-team"
"Project" = "sample-app"
})
value_received = "hello"
```

#### **Number test:**

```
dynamic_value = 42
```

```
    "instance_type" = "t3.micro"
    "monitoring" = true
    "name" = "web-server"
    "storage_gb" = 20
}
subnet_count = 3
tags = tomap({
    "Environment" = "dev"
    "Owner" = "platform-team"
    "Project" = "sample-app"
})
value_received = 42
@Umber-qasim ② /workspaces/Lab11 (main) $
```

### **List test:**

```
dynamic_value = ["a", "b", "c"]
```

```
subnet_count = 3
tags = tomap({
    "Environment" = "dev"
    "Owner" = "platform-team"
    "Project" = "sample-app"
})
value_received = [
    "a",
    "b",
    "c",
]
```

### **Map / Object test:**

```
dynamic_value = {
    name = "server"
    cpu  = 4
}
```

```
subnet_count = 3
tags = tomap({
    "Environment" = "dev"
    "Owner" = "platform-team"
    "Project" = "sample-app"
})
value_received = {
    "cpu" = 4
    "name" = "server"
}
```

### **Null test:**

```
dynamic_value = null
```

```
➤ Windows PowerShell
"list_example" = tolist([
    "web-2",
    "web-1",
    "web-2",
])
"set_example" = toset([
    "me-central-1a",
    "me-central-1b",
])
"tuple_example" = [
    "web-1",
    4,
    true,
]
}
is_production = false
monitoring_enabled = true
mutation_comparison = {
    "mutated_tuple" = toset([
        "4",
        "true",
        "web-1",
        "web-2",
    ])
    "original_tuple" = [
        "web-1",
        4,
        true,
    ]
}
optional_tag = {
    "Custom" = "dev"
    "Name" = "web-server"
}
primary_public_subnet = "subnet-003aaaf45b7679f681"
resource_name = "lab_work-dev"
server_config = {
    "backup_enabled" = false
    "instance_type" = "t3.micro"
    "monitoring" = true
    "name" = "web-server"
    "storage_gb" = 20
}
subnet_count = 3
tags = tomap({
    "Environment" = "dev"
    "Owner" = "platform-team"
    "Project" = "sample-app"
})
@Umber-qasim ② /workspaces/Lab11 (main) $
```

## Task#07: Git ignore

```
@Umber-qasim ② /workspaces/Lab11 (main) $ touch .gitignore
@Umber-qasim ② /workspaces/Lab11 (main) $ vim .gitignore
@Umber-qasim ② /workspaces/Lab11 (main) $ cat .gitignore
.terraform/*
*.tfstate
*.tfstate.*
*.tfvars
*.pem
```

## Task#08: Clean-up then build real infra (VPC, Subnet, IGW, routing, default route table)

```

@Umber-qasim ② /workspaces/Lab11 (main) $ > terraform.tfvars
@Umber-qasim ② /workspaces/Lab11 (main) $ vim terraform.tfvars
@Umber-qasim ② /workspaces/Lab11 (main) $ > locals.tf
@Umber-qasim ② /workspaces/Lab11 (main) $ vim locals.tf
@Umber-qasim ② /workspaces/Lab11 (main) $ cat > main.tf <<EOF
> provider "aws" {
>   shared_config_files      = ["~/.aws/config"]
>   shared_credentials_files = ["~/.aws/credentials"]
> }
> EOF
@Umber-qasim ② /workspaces/Lab11 (main) $ vim main.tf
@Umber-qasim ② /workspaces/Lab11 (main) $

```

## Add variables:

➤ Windows PowerShell

```

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" {}

```

## Create VPC:

➤ Windows PowerShell

```

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

```

## Create Subnet:

➤ Select Windows PowerShell

```

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

```

## Populate terraform.tfvars:

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

## Initialize and apply:

```
❯ Windows PowerShell
- "me-central-1a",
- "me-central-1b",
]
- tuple_example = [
- "web-1",
- 4,
- true,
]
} -> null
- is_production      = false -> null
- monitoring_enabled = true -> null
- mutation_comparison = {
- mutated_tuple = [
- "4",
- "true",
- "web-1",
- "web-2",
]
- original_tuple = [
- "web-1",
- 4,
- true,
]
} -> null
- optional_tag        = {
- Custom = "dev"
- Name   = "web-server"
} -> null
- primary_public_subnet = "subnet-003aaaf45b7679f681" -> null
- resource_name       = "lab_work-dev" -> null
- server_config        = {
- backup_enabled = false
- instance_type  = "t3.micro"
- monitoring     = true
- name           = "web-server"
- storage_gb    = 20
} -> null
- subnet_count         = 3 -> null
- tags                 = {
- Environment = "dev"
- Owner       = "platform-team"
- Project     = "sample-app"
} -> null
aws_vpc.myapp_vpc: Creating...
aws_vpc.myapp_vpc: Creation complete after 2s [id=vpc-0010c57fdbd136756f]
aws_subnet.myapp_subnet_1: Creating...
aws_subnet.myapp_subnet_1: Creation complete after 1s [id=subnet-00cb636e3fad4fba5]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
@Umber-qasim ② /workspaces/LabII (main) $
```

## Add Internet Gateway & Custom Route Table:

Σ Windows PowerShell

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

## Windows PowerShell

```
+ create

Terraform will perform the following actions:

# aws_internet_gateway.myapp_igw will be created
+ resource "aws_internet_gateway" "myapp_igw" {
    + arn      = (known after apply)
    + id       = (known after apply)
    + owner_id = (known after apply)
    + region   = "me-central-1"
    + tags     = {
        + "Name" = "dev-igw"
    }
    + tags_all = {
        + "Name" = "dev-igw"
    }
    + vpc_id   = "vpc-0010c57fdbd136756f"
}

# 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-0010c57fdbd136756f"
}

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-0f73f62c7bf1e37aa]
aws_route_table.myapp_route_table: Creating...
aws_route_table.myapp_route_table: Creation complete after 2s [id=rtb-05780a65a9510a7f0]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
@Umber-qasim ② /workspaces/Lab11 (main) $
```

## Associate Route Table with Subnet:

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

```

plumber-qasim @ /workspaces/Lab11 (main) $ terraform apply -auto-approve
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-0010c57fdbd136756f]
aws_internet_gateway.myapp_igw: Refreshing state... [id=igw-0f73f62c7bf1e37aa]
aws_subnet.myapp_subnet_1: Refreshing state... [id=subnet-00cb636e3fad4fba5]
aws_route_table.myapp_route_table: Refreshing state... [id=rtb-05780a65a9510a7f0]

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_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-05780a65a9510a7f0"
  + subnet_id   = "subnet-00cb636e3fad4fba5"
}

Plan: 1 to add, 0 to change, 0 to destroy.
aws_route_table_association.a_rtb_subnet: Creating...
aws_route_table_association.a_rtb_subnet: Creation complete after 0s [id=rtbassoc-0a0696b4f4082c916]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
plumber-qasim @ /workspaces/Lab11 (main) $

```

## Switch to Default Route Table:

```

❯ Windows PowerShell
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" {}
resource "aws_vpc" "myapp_vpc" {
    cidr_block = var.vpc_cidr_block
    tags = {
        Name = "${var.env_prefix}-vpc"
    }
}
resource "aws_subnet" "myapp_subnet_1" {
    vpc_id          = aws_vpc.myapp_vpc.id
    cidr_block      = var.subnet_cidr_block
    availability_zone = var.availability_zone
    tags = {
        Name = "${var.env_prefix}-subnet-1"
    }
}
resource "aws_internet_gateway" "myapp_igw" {
    vpc_id = aws_vpc.myapp_vpc.id
    tags = {
        Name = "${var.env_prefix}-igw"
    }
}

resource "aws_default_route_table" "main_rt" {
    default_route_table_id = aws_vpc.myapp_vpc.default_route_table_id

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

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

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
}

```

Σ Windows PowerShell

```
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" {}
resource "aws_vpc" "myapp_vpc" {
    cidr_block = var.vpc_cidr_block
    tags = {
        Name = "${var.env_prefix}-vpc"
    }
}
resource "aws_subnet" "myapp_subnet_1" {
    vpc_id          = aws_vpc.myapp_vpc.id
    cidr_block     = var.subnet_cidr_block
    availability_zone = var.availability_zone
    tags = {
        Name = "${var.env_prefix}-subnet-1"
    }
}
resource "aws_internet_gateway" "myapp_igw" {
    vpc_id = aws_vpc.myapp_vpc.id
    tags = {
        Name = "${var.env_prefix}-igw"
    }
}
resource "aws_default_route_table" "main_rt" {
    default_route_table_id = aws_vpc.myapp_vpc.default_route_table_id

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

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

```

❯ Windows PowerShell
+ tags_all           = {
+   + "Name" = "dev-rt"
}
+ vpc_id             = (known after apply)
}

# aws_route_table.myapp_route_table will be destroyed
# (because aws_route_table.myapp_route_table is not in configuration)
- resource "aws_route_table" "myapp_route_table" {
-   arn          = "arn:aws:ec2:me-central-1:458862189705:route-table/rtb-05780a65a9510a7f0" -> null
-   id           = "rtb-05780a65a9510a7f0" -> null
-   owner_id     = "458862189705" -> null
-   propagating_vgws = [] -> null
-   region       = "me-central-1" -> null
-   route        = [
-     {
-       - {
-         - cidr_block      = "0.0.0.0/0"
-         - gateway_id     = "igw-0f73f62c7bf1e37aa"
-         # (11 unchanged attributes hidden)
-       },
-     ] -> null
-   tags          = {
-     - "Name" = "dev-rt"
-   } -> null
-   tags_all      = {
-     - "Name" = "dev-rt"
-   } -> null
-   vpc_id        = "vpc-0010c57fb136756f" -> null
}

# aws_route_table_association.a_rtb_subnet will be destroyed
# (because aws_route_table_association.a_rtb_subnet is not in configuration)
- resource "aws_route_table_association" "a_rtb_subnet" {
-   id          = "rtbassoc-0a0696b4f4082c916" -> null
-   region      = "me-central-1" -> null
-   route_table_id = "rtb-05780a65a9510a7f0" -> null
-   subnet_id    = "subnet-00cb636e3fad4fba5" -> null
-   # (1 unchanged attribute hidden)
}

Plan: 1 to add, 0 to change, 2 to destroy.
aws_route_table_association.a_rtb_subnet: Destroying... [id=rtbassoc-0a0696b4f4082c916]
aws_default_route_table.main_rt: Creating...
aws_route_table_association.a_rtb_subnet: Destruction complete after 1s
aws_route_table.myapp_route_table: Destroying... [id=rtb-05780a65a9510a7f0]
aws_default_route_table.main_rt: Creation complete after 2s [id=rtb-0a288f0829226c7e9]
aws_route_table.myapp_route_table: Destruction complete after 1s

Apply complete! Resources: 1 added, 0 changed, 2 destroyed.

```

## Task#08: Security Group, Key Pair, EC2 Instance, user\_data & nginx

```
variable "my_ip" {}
```

```
NUMBER-QASIM @ /workspaces/Lab11 (main) $ curl icanhazip.com
4.240.18.224
```

```

❯ Windows PowerShell
vpc_cidr_block      = "10.0.0.0/16"
subnet_cidr_block   = "10.0.10.0/24"
my_ip = "4.240.18.224/32"
instance_type = "t3.micro"
availability_zone = "me-central-1a"
env_prefix = "dev"

```

Σ Windows PowerShell

```
}

resource "aws_internet_gateway" "myapp_igw" {
  vpc_id = aws_vpc.myapp_vpc.id
  tags = [
    Name = "${var.env_prefix}-igw"
  ]
}
resource "aws_default_route_table" "main_rt" {
  default_route_table_id = aws_vpc.myapp_vpc.default_route_table_id

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

  tags = [
    Name = "${var.env_prefix}-rt"
  ]
}
variable "my_ip" {}
resource "aws_default_security_group" "myapp_sg" {
  vpc_id = aws_vpc.myapp_vpc.id

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = [var.my_ip]
  }

  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = [
    Name = "${var.env_prefix}-sg"
  ]
}
```

```
Windows PowerShell
+ prefix_list_ids = []
+ protocol        = "tcp"
+ security_groups = []
+ self            = false
+ to_port         = 80
# (1 unchanged attribute hidden)
},
{
+ cidr_blocks     = [
+ "4.240.18.224/32",
]
+ from_port       = 22
+ ipv6_cidr_blocks = []
+ prefix_list_ids = []
+ protocol        = "tcp"
+ security_groups = []
+ self            = false
+ to_port         = 22
# (1 unchanged attribute hidden)
},
{
+ name           = (known after apply)
+ name_prefix    = (known after apply)
+ owner_id       = (known after apply)
+ region         = "me-central-1"
+ revoke_rules_on_delete = false
+ tags           = {
+   "Name" = "dev-sg"
}
+ tags_all       = {
+   "Name" = "dev-sg"
}
+ vpc_id         = "vpc-0010c57fb136756f"
}

Plan: 1 to add, 0 to change, 0 to destroy.
aws_default_security_group.myapp_sg: Creating...
aws_default_security_group.myapp_sg: Creation complete after 3s [id=sg-004b330147056b771]

Warning: Value for undeclared variable
The root module does not declare a variable named "instance_type" but a value was found in file "terraform.tfvars". If you meant to use this value, add a "variable" block to the configuration.
To silence these warnings, use TF_VAR_... environment variables to provide certain "global" settings to all configurations in your organization. To reduce the verbosity of these warnings, Activate Windows
Go to Settings to activate Windows.

apply complete! Resources: 1 added, 0 changed, 0 destroyed.
@Umber-qasim ~ /workspaces/lambda (main) $
```

```
@Umber-qasim ~ /workspaces/Lambda (main) $ aws ec2 create-key-pair \
> --key-name MyED25519Key \
> --key-type ed25519 \
> --key-format pem \
> --query 'KeyMaterial' \
> --output text > MyED25519Key.pem
9Key.pem
@Umber-qasim ~ /workspaces/Lambda (main) $ 
@Umber-qasim ~ /workspaces/Lambda (main) $ chmod 600 MyED25519Key.pem
@Umber-qasim ~ /workspaces/Lambda (main) $ cat .gitignore
.terraform/*
*.tfstate
*.tfstate.*
*.tfvars
*.pem
@Umber-qasim ~ /workspaces/Lambda (main) $
```

Σ Windows PowerShell

```
}

variable "my_ip" {}

resource "aws_default_security_group" "myapp_sg" {
  vpc_id = aws_vpc.myapp_vpc.id

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = [var.my_ip]
  }

  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "${var.env_prefix}-sg"
  }
}

resource "aws_instance" "myapp-server" {
  ami           = "ami-05524d6658fcf35b6"
  instance_type = var.instance_type
  subnet_id     = aws_subnet.myapp_subnet_1.id
  security_groups = [aws_default_security_group.myapp_sg.id]
  availability_zone = var.availability_zone
  associate_public_ip_address = true
  key_name      = "MyED25519Key"

  tags = {
    Name = "${var.env_prefix}-ec2-instance"
  }
}

output "aws_instance_public_ip" {
  value = aws_instance.myapp-server.public_ip
}

-- INSERT --
```

```

➤ Select Windows PowerShell
+ tags                                = {
  + "Name"   = "dev-ec2-instance"
}
+ tags_all                             = {
  + "Name"   = "dev-ec2-instance"
}
+ tenancy                               = (known after apply)
+ user_data_base64                     = (known after apply)
+ user_data_replace_on_change          = false
+ vpc_security_group_ids               = (known after apply)

+ capacity_reservation_specification (known after apply)
+ cpu_options (known after apply)
+ ebs_block_device (known after apply)
+ enclave_options (known after apply)
+ ephemeral_block_device (known after apply)
+ instance_market_options (known after apply)
+ maintenance_options (known after apply)
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ primary_network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ aws_instance_public_ip = (known after apply)
aws_instance.myapp-server: Creating...
aws_instance.myapp-server: Still creating... [00m10s elapsed]
aws_instance.myapp-server: Creation complete after 14s [id=i-0e22c8905b589e82a]

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

Outputs:
aws_instance_public_ip = "40.172.88.155"

```

```

@lumber-qasim ② /workspaces/Lab11 (main) $ ssh -i MyED25519Key.pem ec2-user@40.172.88.155
The authenticity of host '40.172.88.155 (40.172.88.155)' can't be established.
ED25519 key fingerprint is SHA256:4vB3rT6TW5ShQDM7vGENiE37nDUJHZJPhERfttLnXE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '40.172.88.155' (ED25519) to the list of known hosts.

      #_
      ~\_ #####      Amazon Linux 2023
      ~~ \#####\
      ~~  \###|
      ~~   \#/      https://aws.amazon.com/linux/amazon-linux-2023
      ~~    V~'__->
      ~~~_.  _/
      ~~  _/_/
      _/m'

[ec2-user@ip-10-0-10-108 ~]$ exit
logout
Connection to 40.172.88.155 closed.
@lumber-qasim ② /workspaces/Lab11 (main) $

```

```
jumber-qasim @ /workspaces/LabII (main) $ ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519 -N ""  
Generating public/private ed25519 key pair.  
Your identification has been saved in /home/codespace/.ssh/id_ed25519  
Your public key has been saved in /home/codespace/.ssh/id_ed25519.pub  
The key fingerprint is:  
SHA256:GDcnAOeXmC48yrryK9/1P3cZSNh6bEFotNSd+FLnrZM codespace@codespaces-5e1199  
The key's randomart image is:  
+--[ED25519 256]--+  
| ..o .oo o . |  
| o + oo.+ + . |  
| = * .+ o o. |  
| . . = +. = . o |  
| + o S + + o |  
| . . o . . = E |  
| o . . o + |  
| o. . . . . o |  
|+=+... ...o . |  
+---[SHA256]-----+
```

```
[x] ec2-user@ip-10-0-10-108:~  
vpc_id = aws_vpc.myapp_vpc.id  
  
ingress {  
    from_port  = 22  
    to_port    = 22  
    protocol   = "tcp"  
    cidr_blocks = [var.my_ip]  
}  
  
ingress {  
    from_port  = 80  
    to_port    = 80  
    protocol   = "tcp"  
    cidr_blocks = ["0.0.0.0/0"]  
}  
  
egress {  
    from_port  = 0  
    to_port    = 0  
    protocol   = "-1"  
    cidr_blocks = ["0.0.0.0/0"]  
}  
  
tags = {  
    Name = "${var.env_prefix}-sg"  
}  
}  
resource "aws_instance" "myapp-server" {  
    ami                  = "ami-05524d6658fcf35b6"  
    instance_type        = var.instance_type  
    subnet_id           = aws_subnet.myapp_subnet_1.id  
    security_groups      = [aws_default_security_group.myapp_sg.id]  
    availability_zone   = var.availability_zone  
    associate_public_ip_address = true  
    key_name             = aws_key_pair.ssh_key.key_name  
  
    tags = {  
        Name = "${var.env_prefix}-ec2-instance"  
    }  
}  
  
output "aws_instance_public_ip" {  
    value = aws_instance.myapp-server.public_ip  
}  
resource "aws_key_pair" "ssh_key" {  
    key_name   = "serverkey"  
    public_key = file("~/ssh/id_ed25519.pub")  
}
```

```

[ec2-user@ip-10-0-10-108:~]
- delete_on_termination = true -> null
- device_name          = "/dev/xvda" -> null
- encrypted            = false -> null
- iops                 = 3000 -> null
- tags                = {} -> null
- tags_all             = {} -> null
- throughput           = 125 -> null
- volume_id            = "vol-03fc773ca3975096f" -> null
- volume_size           = 8 -> null
- volume_type           = "gp3" -> null
# (1 unchanged attribute hidden)
}

# aws_key_pair.ssh_key will be created
+ resource "aws_key_pair" "ssh_key" {
+   arn              = (known after apply)
+   fingerprint      = (known after apply)
+   id               = (known after apply)
+   key_name         = "serverkey"
+   key_name_prefix  = (known after apply)
+   key_pair_id     = (known after apply)
+   key_type         = (known after apply)
+   public_key       = "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIHI1mM8v5sBe905AsrupRGrAr5z6gPqL4qohlw6efkm2SR codespace@codespaces-5e1199"
+   region          = "me-central-1"
+   tags_all        = (known after apply)
}

Plan: 2 to add, 0 to change, 1 to destroy.

Changes to Outputs:
~ aws_instance_public_ip = "40.172.88.155" -> (known after apply)
aws_instance.myapp-server: Destroying... [id=i-0e22c8905b589e82a]
aws_instance.myapp-server: Still destroying... [id=i-0e22c8905b589e82a, 00m10s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0e22c8905b589e82a, 00m20s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0e22c8905b589e82a, 00m30s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0e22c8905b589e82a, 00m40s elapsed]
aws_instance.myapp-server: Destruction complete after 41s
aws_key_pair.ssh_key: Creating...
aws_key_pair.ssh_key: Creation complete after 0s [id=serverkey]
aws_instance.myapp-server: Creating...
aws_instance.myapp-server: Still creating... [00m10s elapsed]
aws_instance.myapp-server: Creation complete after 14s [id=i-01712b6c60330f8ab]

Apply complete! Resources: 2 added, 0 changed, 1 destroyed.

Outputs:

aws_instance_public_ip = "3.29.233.176"
@Umber-qasim @ /workspaces/Lab11 (main) $

```

```

aws_instance_public_ip = "3.29.233.176"
@Umber-qasim @ /workspaces/Lab11 (main) $ ssh ec2-user@3.29.233.176
The authenticity of host '3.29.233.176 (3.29.233.176)' can't be established.
ED25519 key fingerprint is SHA256:XrVI4sXF59UCita76HQmjaxI2H0aVTfY9dt0SZlmpkY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.29.233.176' (ED25519) to the list of known hosts.

,
#_
~\_\_ #####_      Amazon Linux 2023
~~ \_\#\#\#\`_
~~  \#\#\#
~~    \#/ __ > https://aws.amazon.com/linux/amazon-linux-2023
~~      V~' `->
~~      /
~~ ._. _/
~~ / _/
~~ /m/
[ec2-user@ip-10-0-10-4 ~]$

```

## Install nginx via inline user\_data

```

❯ Select ec2-user@ip-10-0-10-217:~
aws_instance.myapp-server: Creation complete after 13s [id=i-0958d588a56049329]

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

Outputs:

aws_instance_public_ip = "51.112.230.49"
@lumber-gasim @ /workspaces/labil (main) $ curl localhost
curl: (7) Failed to connect to localhost port 80 after 0 ms: Couldn't connect to server
@lumber-gasim @ /workspaces/labil (main) $ ssh -i ~/.ssh/id_ed25519 ec2-user@51.112.230.49
The authenticity of host '51.112.230.49 (51.112.230.49)' can't be established.
ED25519 key fingerprint is SHA256:/NxofH8bk4aBJ+GdXChg+ecpuAeIHB9iOCvRCQRR+Os.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '51.112.230.49' (ED25519) to the list of known hosts.

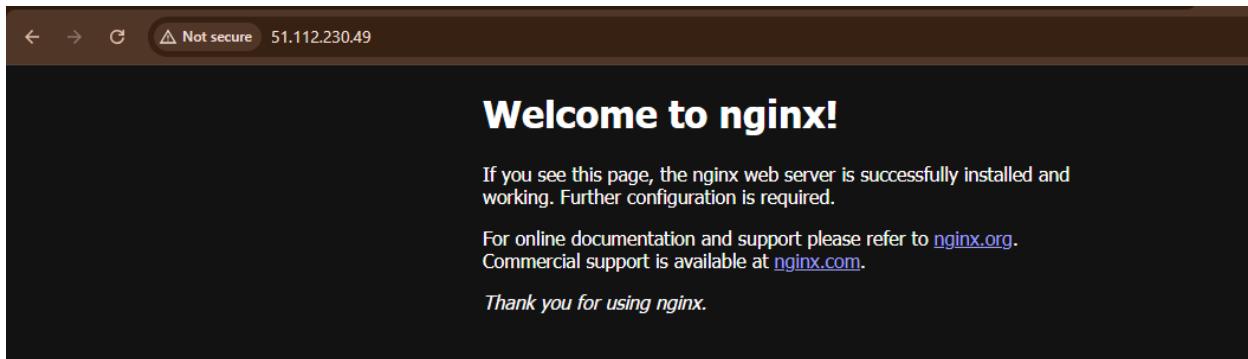
          _#
  ~\_\_ #####_      Amazon Linux 2023
  ~~ \_\#\#\#\|_
  ~~   \#\#|
  ~~     \#/  https://aws.amazon.com/linux/amazon-linux-2023
  ~~   V~` .->
  ~~~   /_
  ~~.~_. /_
  _/ _/
 _/m/'

[ec2-user@ip-10-0-10-217 ~]$ curl localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
[ec2-user@ip-10-0-10-217 ~]$

```



## Using external user\_data script

```
@Jumber-qasim ② /workspaces/Lab11 (main) $ cat > entry-script.sh <<'EOF'  
> #!/bin/bash  
> yum update -y  
> yum install -y nginx  
> systemctl start nginx  
> systemctl enable nginx  
> EOF
```

➤ Select Windows PowerShell

```
ingress {  
    from_port    = 80  
    to_port      = 80  
    protocol     = "tcp"  
    cidr_blocks  = ["0.0.0.0/0"]  
}  
  
egress {  
    from_port    = 0  
    to_port      = 0  
    protocol     = "-1"  
    cidr_blocks  = ["0.0.0.0/0"]  
}  
  
tags = {  
    Name = "${var.env_prefix}-sg"  
}  
}  
resource "aws_instance" "myapp-server" {  
    ami                      = "ami-05524d6658fcf35b6"  
    instance_type            = var.instance_type  
    subnet_id                = aws_subnet.myapp_subnet_1.id  
    security_groups          = [aws_default_security_group.myapp_sg.id]  
    availability_zone        = var.availability_zone  
    associate_public_ip_address = true  
    key_name = aws_key_pair.ssh_key.key_name  
  
    user_data = file("entry-script.sh")  
  
    tags = {  
        Name = "${var.env_prefix}-ec2-instance"  
    }  
}  
  
output "aws_instance_public_ip" {  
    value = aws_instance.myapp-server.public_ip  
}  
resource "aws_key_pair" "ssh_key" {  
    key_name    = "serverkey"  
    public_key  = file("~/ssh/id_ed25519.pub")  
}
```

```

❯ ec2-user@ip-10-0-10-249:~
aws_instance.myapp-server: Creating...
aws_instance.myapp-server: Still creating... [00m10s elapsed]
aws_instance.myapp-server: Creation complete after 13s [id=i-0a58917275ad6cbe4]

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

Outputs:

aws_instance_public_ip = "158.252.33.97"
@Umber-qasim ② /workspaces/Labil (main) $ ssh -i ~/.ssh/id_ed25519 ec2-user@158.252.33.97
The authenticity of host '158.252.33.97 (158.252.33.97)' can't be established.
ED25519 key fingerprint is SHA256:9/oF1Iyg5fKb2qG1lArR7nwn7QhjDMngJtKs7TzHdw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '158.252.33.97' (ED25519) to the list of known hosts.

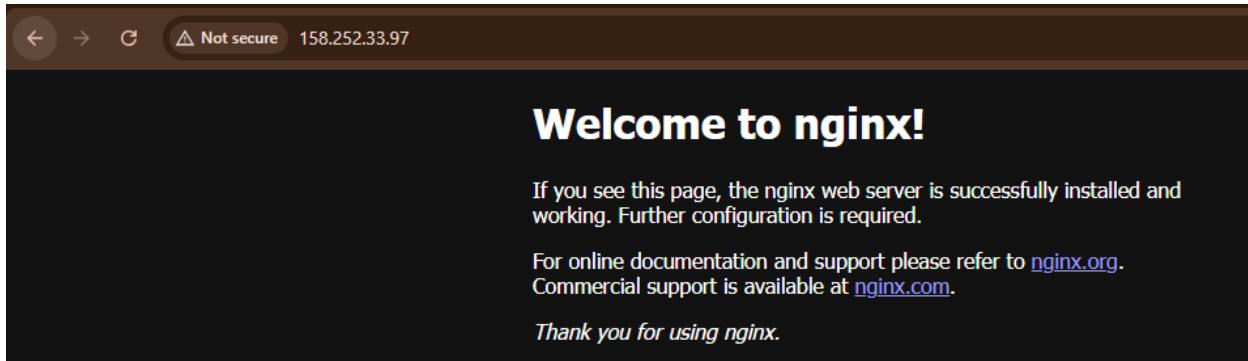
          #
 ~\_ ##### Amazon Linux 2023
 ~~ \#####\_
 ~~ \###|
 ~~ \#/ __ https://aws.amazon.com/linux/amazon-linux-2023
 ~~ V~' '-'>
 ~~ /
 ~~ ._. /_
 ~~ _/ _/
 _/m/'

[ec2-user@ip-10-0-10-249 ~]$ curl localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
[ec2-user@ip-10-0-10-249 ~]$

```



## Destroy all resources

```
❯ ec2-user@ip-10-0-10-249:~  
- default_network_acl_id           = "acl-069b5aec5700fcfd25" -> null  
- default_route_table_id          = "rtb-0a288f0829226c7e9" -> null  
- default_security_group_id       = "sg-004b330147056b771" -> null  
- dhcp_options_id                = "dopt-0f6507644ddf11aeee" -> null  
- enable_dns_hostnames          = false -> null  
- enable_dns_support             = true -> null  
- enable_network_address_usage_metrics = false -> null  
- id                            = "vpc-0010c57fdbd136756f" -> null  
- instance_tenancy               = "default" -> null  
- ipv6_netmask_length            = 0 -> null  
- main_route_table_id            = "rtb-0a288f0829226c7e9" -> null  
- owner_id                       = "458862189705" -> null  
- region                         = "me-central-1" -> null  
- tags                           = {  
    - "Name" = "dev-vpc"  
} -> null  
- tags_all                       = {  
    - "Name" = "dev-vpc"  
} -> null  
# (4 unchanged attributes hidden)  
}  
  
Plan: 0 to add, 0 to change, 7 to destroy.  
  
Changes to Outputs:  
- aws_instance_public_ip = "158.252.33.97" -> null  
aws_default_route_table.main_rt: Destroying... [id=rtb-0a288f0829226c7e9]  
aws_instance.myapp-server: Destroying... [id=i-0a58917275ad6cbe4]  
aws_default_route_table.main_rt: Destruction complete after 0s  
aws_internet_gateway.myapp_igw: Destroying... [id=igw-0f73f62c7bf1e37aa]  
aws_instance.myapp-server: Still destroying... [id=i-0a58917275ad6cbe4, 00m10s elapsed]  
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0f73f62c7bf1e37aa, 00m10s elapsed]  
aws_instance.myapp-server: Still destroying... [id=i-0a58917275ad6cbe4, 00m20s elapsed]  
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0f73f62c7bf1e37aa, 00m20s elapsed]  
aws_instance.myapp-server: Still destroying... [id=i-0a58917275ad6cbe4, 00m30s elapsed]  
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0f73f62c7bf1e37aa, 00m30s elapsed]  
aws_internet_gateway.myapp_igw: Destruction complete after 39s  
aws_instance.myapp-server: Still destroying... [id=i-0a58917275ad6cbe4, 00m40s elapsed]  
aws_instance.myapp-server: Destruction complete after 41s  
aws_default_security_group.myapp_sg: Destroying... [id=sg-004b330147056b771]  
aws_key_pair.ssh_key: Destroying... [id=serverkey]  
aws_subnet.myapp_subnet_1: Destroying... [id=subnet-00cb636e3fad4fba5]  
aws_default_security_group.myapp_sg: Destruction complete after 0s  
aws_key_pair.ssh_key: Destruction complete after 1s  
aws_subnet.myapp_subnet_1: Destruction complete after 1s  
aws_vpc.myapp_vpc: Destroying... [id=vpc-0010c57fdbd136756f]  
aws_vpc.myapp_vpc: Destruction complete after 1s  
  
Destroy complete! Resources: 7 destroyed.  
@limer-qasim ② /workspaces/Leb11 (main) $
```

## Verify Terraform state files

```
➤ ec2-user@ip-10-0-10-249:~  
{"name": "myapp_vpc",  
 "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",  
 "instances": [  
     {  
         "schema_version": 1,  
         "attributes": {  
             "arn": "arn:aws:ec2:me-central-1:458862189705:vpc/vpc-0010c57fdbd136756f",  
             "assign_generated_ipv6_cidr_block": false,  
             "cidr_block": "10.0.0.0/16",  
             "default_network_acl_id": "acl-069b5aec5700fcfd25",  
             "default_route_table_id": "rtb-0a288f0829226c7e9",  
             "default_security_group_id": "sg-004b330147056b771",  
             "dhcp_options_id": "dopt-0f6507644ddf11aee",  
             "enable_dns_hostnames": false,  
             "enable_dns_support": true,  
             "enable_network_address_usage_metrics": false,  
             "id": "vpc-0010c57fdbd136756f",  
             "instance_tenancy": "default",  
             "ipv4_ipam_pool_id": null,  
             "ipv4_netmask_length": null,  
             "ipv6_association_id": "",  
             "ipv6_cidr_block": "",  
             "ipv6_cidr_block_network_border_group": "",  
             "ipv6_ipam_pool_id": "",  
             "ipv6_netmask_length": 0,  
             "main_route_table_id": "rtb-0a288f0829226c7e9",  
             "owner_id": "458862189705",  
             "region": "me-central-1",  
             "tags": {  
                 "Name": "dev-vpc"  
             },  
             "tags_all": {  
                 "Name": "dev-vpc"  
             }  
         },  
         "sensitive_attributes": [],  
         "identity_schema_version": 0,  
         "identity": {  
             "account_id": "458862189705",  
             "id": "vpc-0010c57fdbd136756f",  
             "region": "me-central-1"  
         },  
         "private": "eyJzY2hlbWFfdmVyc2lvbiI6IjEifQ=="  
     }  
 ]  
 ],  
 "check_results": null  
}  
@Umber-qasim ② /workspaces/LabII (main) $
```

## Ensure no sensitive files are committed

```
@Umber-qasim ② /workspaces/Lab11 (main) $ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    .main.tf.swp
    .terraform.lock.hcl
    aws/
    awscliv2.zip
    entry-script.sh
    locals.tf
    main.tf

nothing added to commit but untracked files present (use "git add" to track)
@Umber-qasim ② /workspaces/Lab11 (main) $ cat .gitignore
.terraform/*
*.tfstate
*.tfstate.*
*.tfvars
*.pem
@Umber-qasim ② /workspaces/Lab11 (main) $
```