Fatima Jinnah
Women University

CLOUD COMPUTING LAB: 11

Submitted By:

Seerat Fatima

Registration. No:

2023-BSE-060

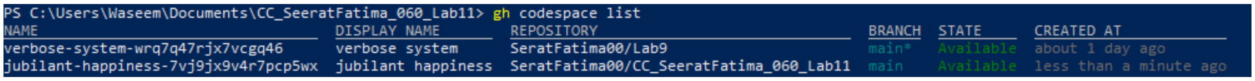
Submitted To:

Engr. Shoaib

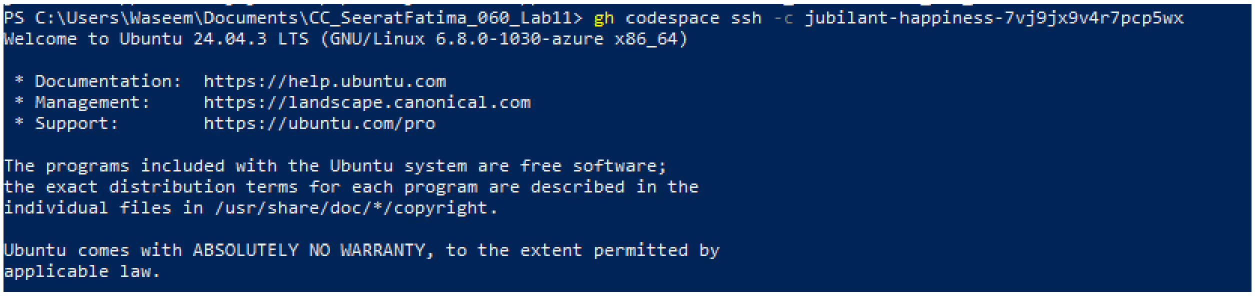
Section:

5B

Task 0 Lab Setup (Codespace & GH CLI)

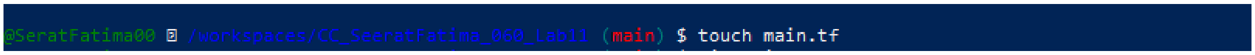


```
PS C:\Users\Waseem\Documents\CC_SeeratFatima_060_Lab11> gh codespace list
NAME                               DISPLAY NAME         REPOSITORY                                BRANCH  STATE      CREATED AT
verbose-system-wrq7q47rjx7vcgq46   verbose system       SeratFatima00/Lab9                        main*   Available  about 1 day ago
jubilant-happiness-7vj9jx9v4r7pcp5wx  jubilant happiness  SeratFatima00/CC_SeeratFatima_060_Lab11  main    Available  less than a minute ago
```

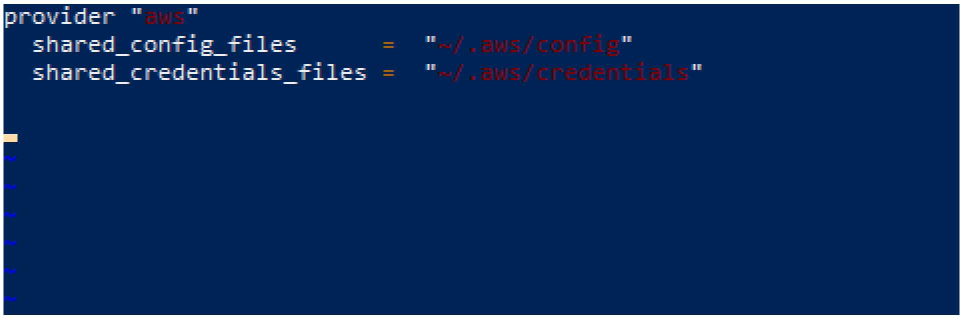


```
PS C:\Users\Waseem\Documents\CC_SeeratFatima_060_Lab11> gh codespace ssh -c jubilant-happiness-7vj9jx9v4r7pcp5wx
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-1030-azure x86_64)

 * 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.
```

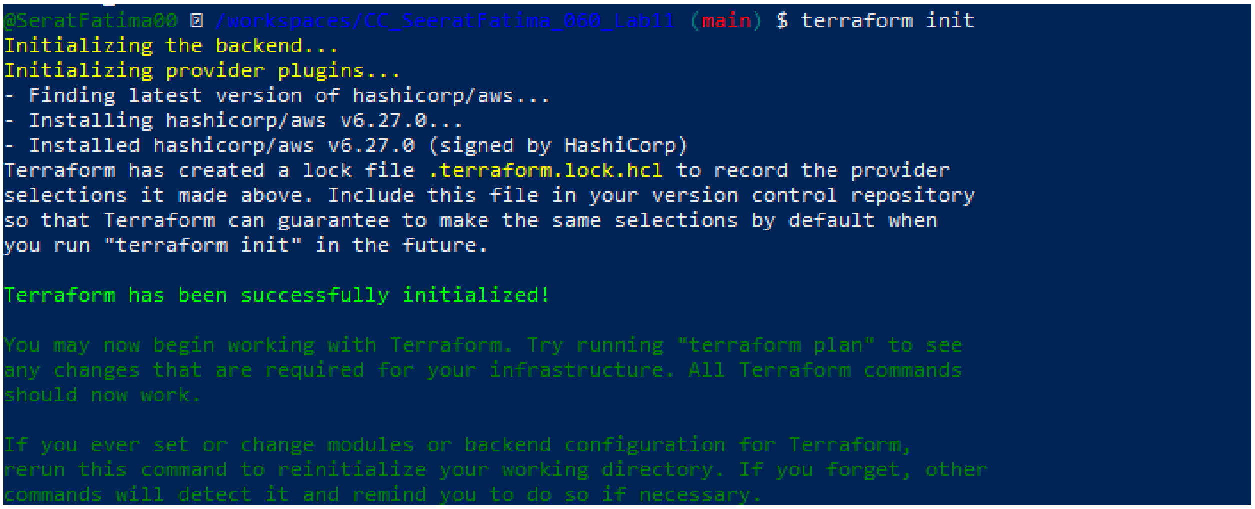
Task 1 — Provider & Basic variable (variable precedence)



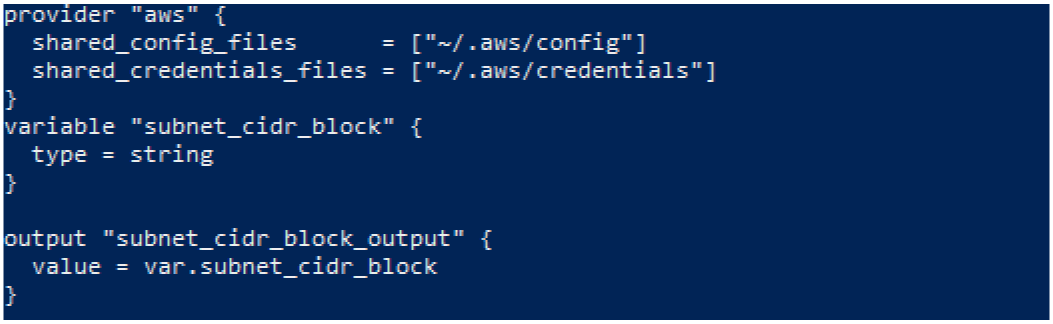
```
@SeratFatima00 ➜ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ touch main.tf
```



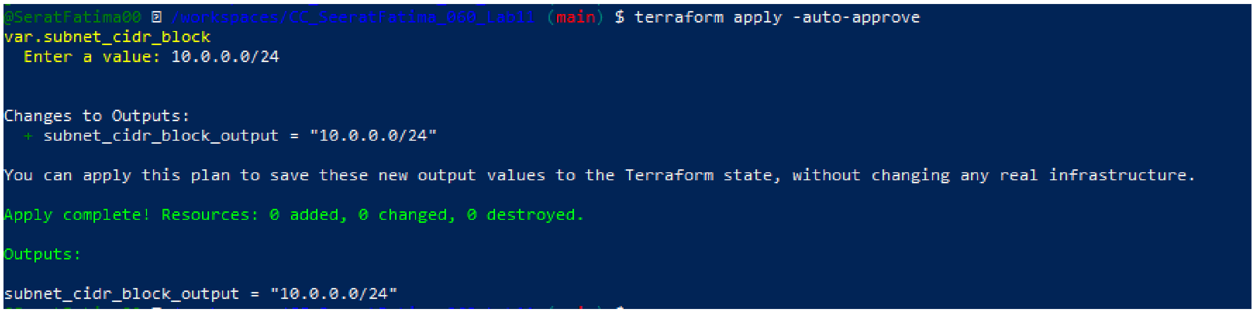
```
provider "aws"
  shared_config_files      =  "~/.aws/config"
  shared_credentials_files =  "~/.aws/credentials"


~
~
~
~
~
~
```



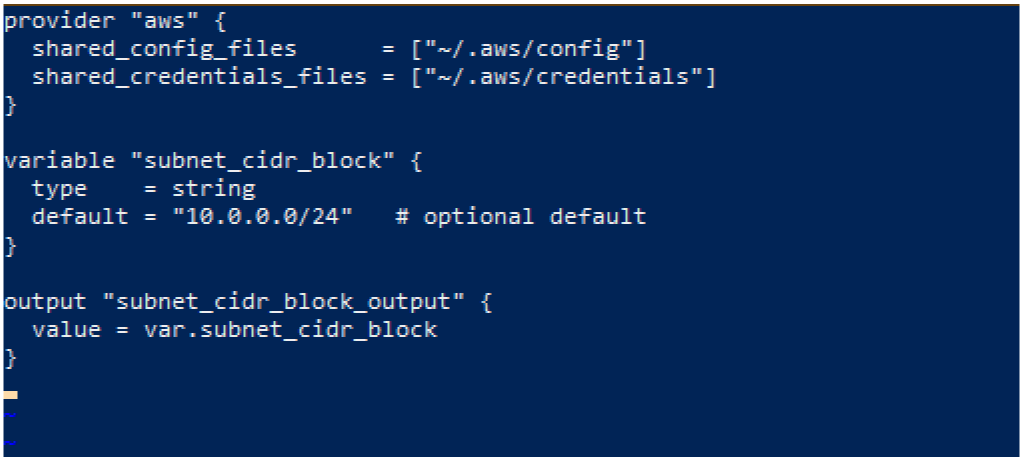
```
@SeratFatima00 ➜ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.27.0...
- Installed hashicorp/aws v6.27.0 (signed by HashiCorp)
Terraform has created 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.
```



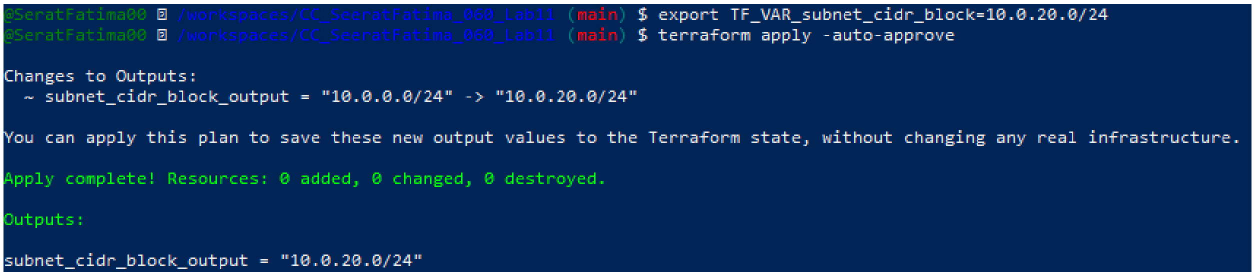
```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
variable "subnet_cidr_block" {
  type = string
}

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



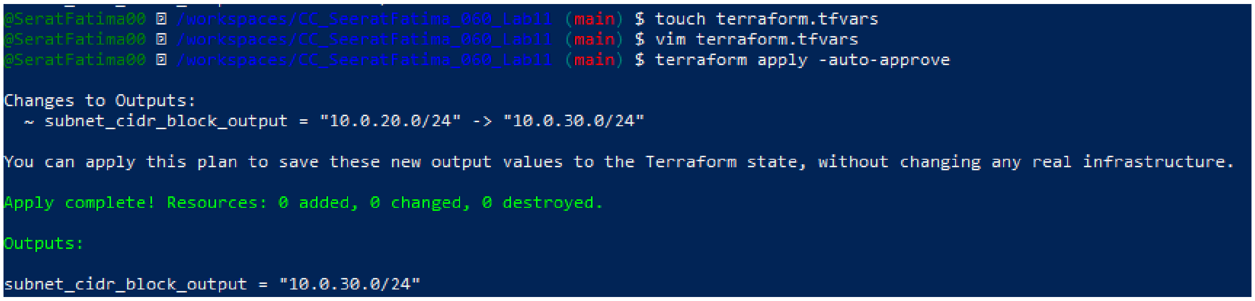
```
@SeratFatima00 ▣ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve
var.subnet_cidr_block
  Enter a value: 10.0.0.0/24


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"
```



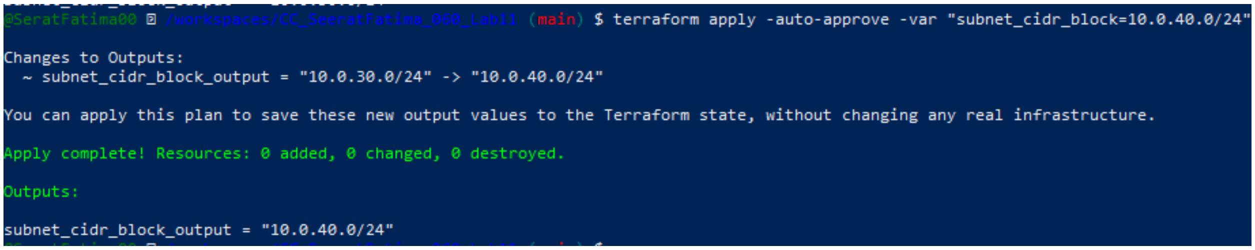
```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}

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

output "subnet_cidr_block_output" {
  value = var.subnet_cidr_block
}

~
~
```



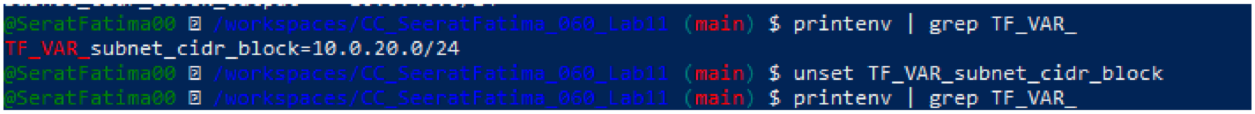
```
@SeratFatima00 ▣ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ export TF_VAR_subnet_cidr_block=10.0.20.0/24
@SeratFatima00 ▣ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve

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"
```



```
@SeratFatima00 ▣ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ touch terraform.tfvars
@SeratFatima00 ▣ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ vim terraform.tfvars
@SeratFatima00 ▣ /workspaces/CC_SeeratFatima_060_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"
```

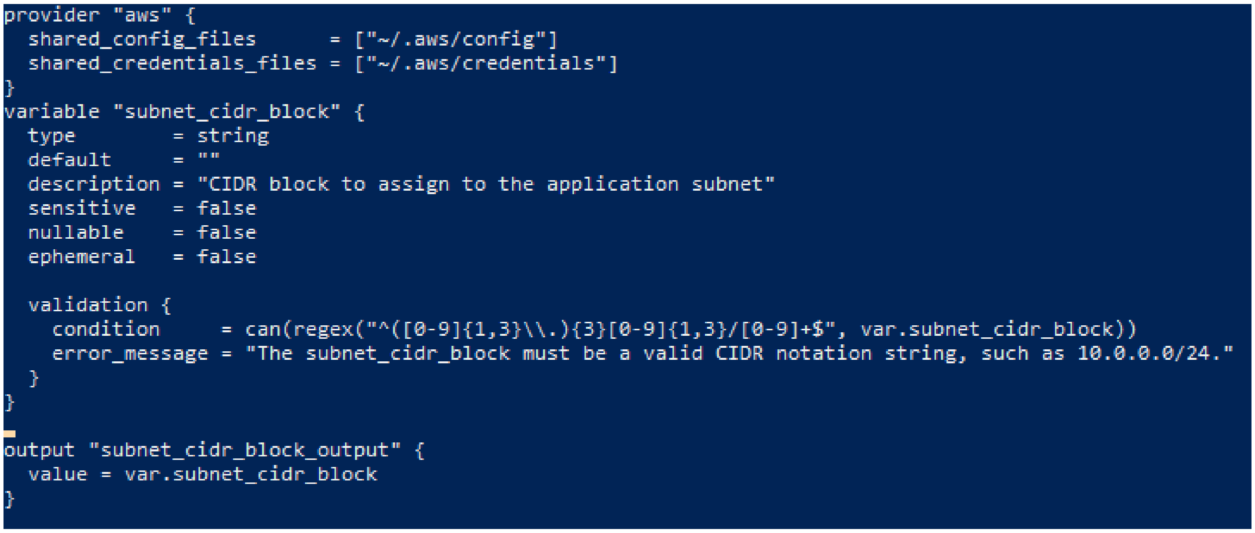


```
@SeratFatima00 ⊡ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve -var "subnet_cidr_block=10.0.40.0/24"

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"
```

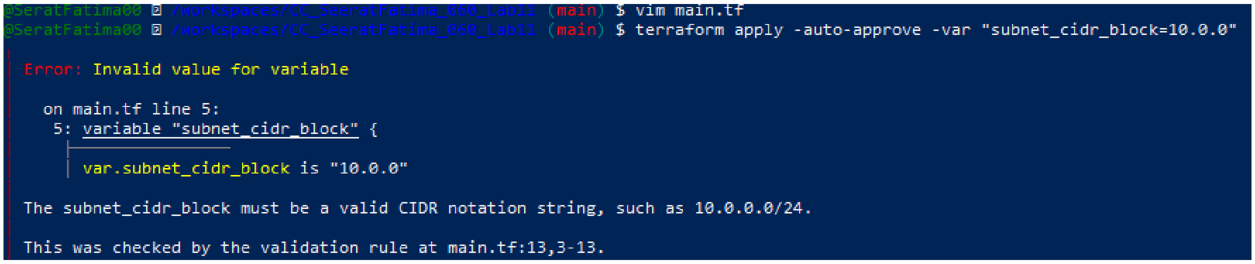


```
@SeratFatima00 ⊡ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ printenv | grep TF_VAR_
TF_VAR_subnet_cidr_block=10.0.20.0/24
@SeratFatima00 ⊡ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ unset TF_VAR_subnet_cidr_block
@SeratFatima00 ⊡ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ printenv | grep TF_VAR_
```

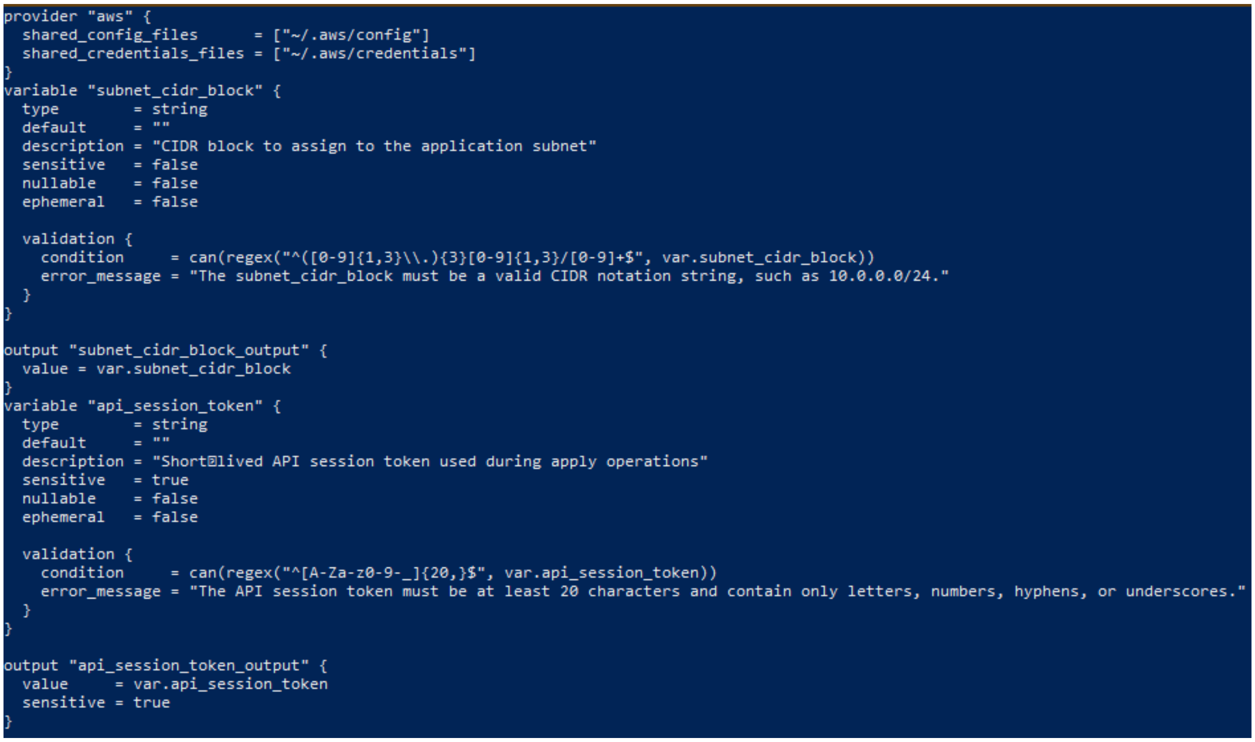
Task 2 — Variable validation & sensitive / ephemeral variables



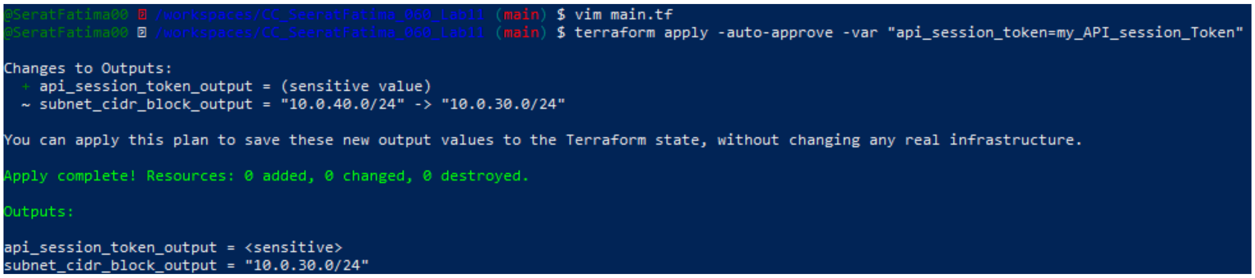
```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
variable "subnet_cidr_block" {
  type        = string
  default     = ""
  description = "CIDR block to assign to the application subnet"
  sensitive   = false
  nullable    = false
  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."
  }
}

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



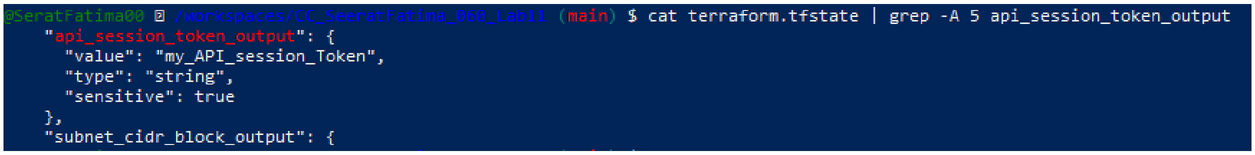
```
@SeratFatima00 ⊡ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ vim main.tf
@SeratFatima00 ⊡ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve -var "subnet_cidr_block=10.0.0"
│
│ Error: Invalid value for variable
│
│   on main.tf line 5:
│    5: variable "subnet_cidr_block" {
│     ├────────────────
│     │ 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.
```



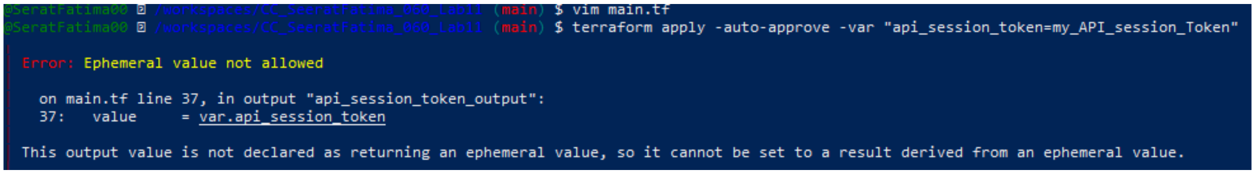
```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
variable "subnet_cidr_block" {
  type        = string
  default     = ""
  description = "CIDR block to assign to the application subnet"
  sensitive   = false
  nullable    = false
  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."
  }
}

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



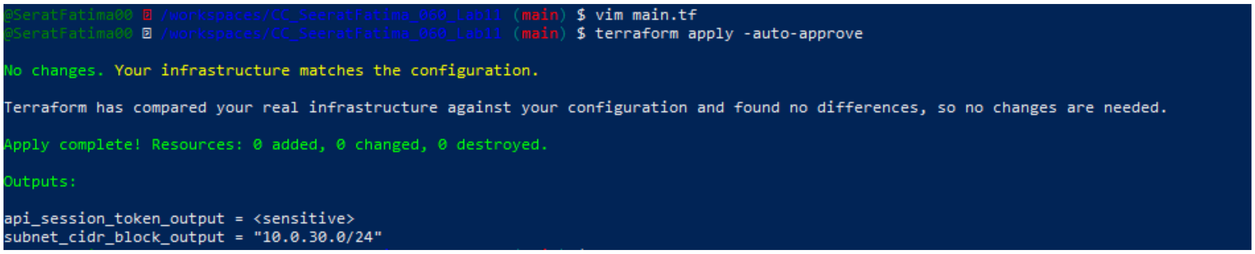
```
@SeratFatima00 ▯ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ vim main.tf
@SeratFatima00 ▯ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve -var "api_session_token=my_API_session_Token"

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"
```



```
@SeratFatima00 ▯ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ cat terraform.tfstate | grep -A 5 api_session_token_output
    "api_session_token_output": {
      "value": "my_API_session_Token",
      "type": "string",
      "sensitive": true
    },
    "subnet_cidr_block_output": {
```

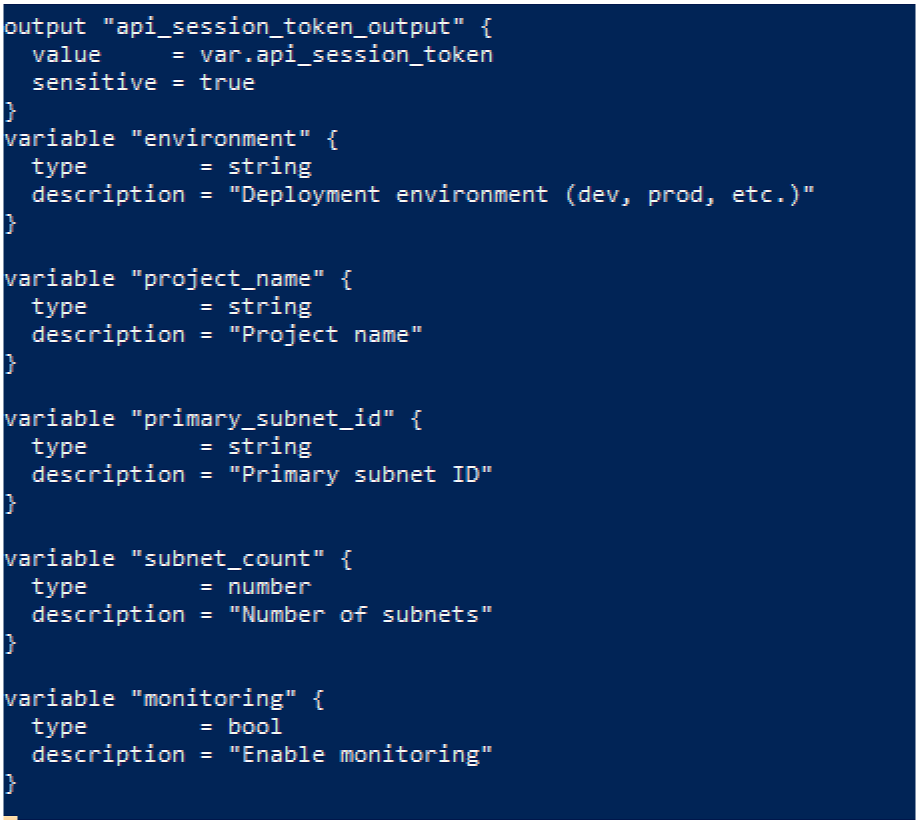


```
@SeratFatima00 ▯ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ vim main.tf
@SeratFatima00 ▯ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve -var "api_session_token=my_API_session_Token"

  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 set to a result derived from an ephemeral value.
```

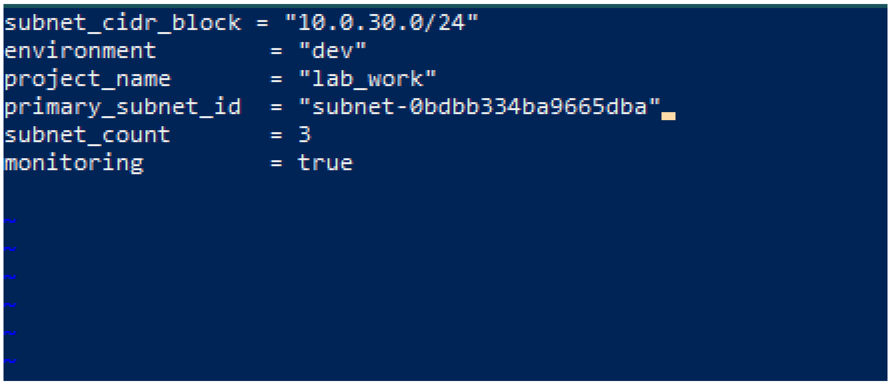


```
@SeratFatima00 ▯ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ vim main.tf
@SeratFatima00 ▯ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.

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

Outputs:

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

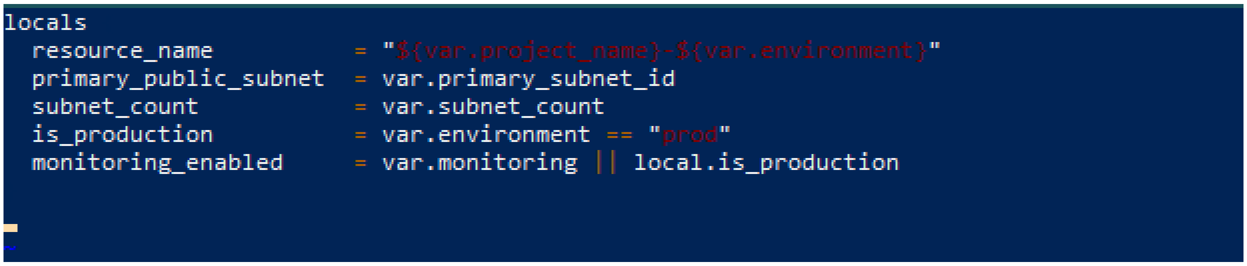
Task 3 — Project-level variables, locals, and outputs



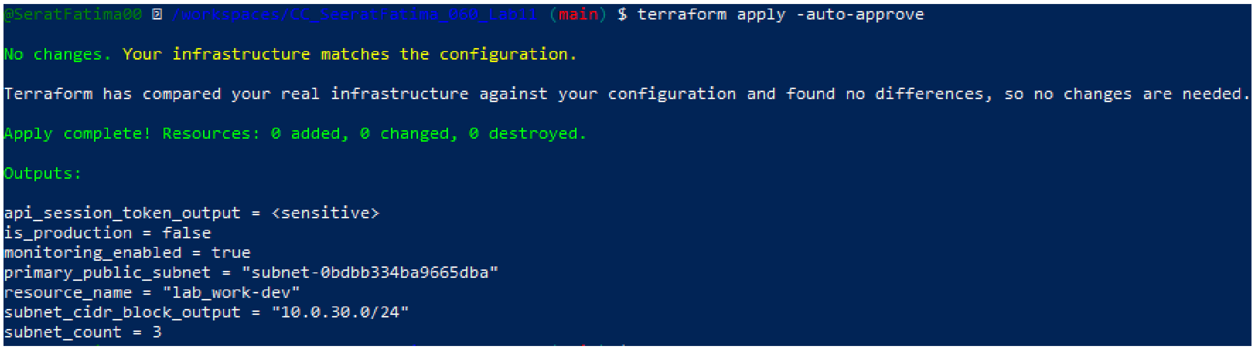
```
output "api_session_token_output" {
  value     = var.api_session_token
  sensitive = true
}
variable "environment" {
  type        = string
  description = "Deployment environment (dev, prod, etc.)"
}

variable "project_name" {
  type        = string
  description = "Project name"
}

variable "primary_subnet_id" {
  type        = string
  description = "Primary subnet ID"
}

variable "subnet_count" {
  type        = number
  description = "Number of subnets"
}

variable "monitoring" {
  type        = bool
  description = "Enable monitoring"
}
```



```
subnet_cidr_block = "10.0.30.0/24"
environment         = "dev"
project_name        = "lab_work"
primary_subnet_id   = "subnet-0bdbb334ba9665dba"
subnet_count        = 3
monitoring          = true

~
~
~
~
~
~
```

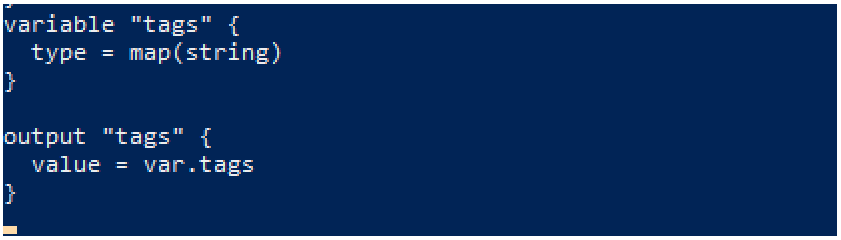


```
locals
  resource_name          = "${var.project_name}-${var.environment}"
  primary_public_subnet  = var.primary_subnet_id
  subnet_count           = var.subnet_count
  is_production          = var.environment == "prod"
  monitoring_enabled     = var.monitoring || local.is_production


~
```

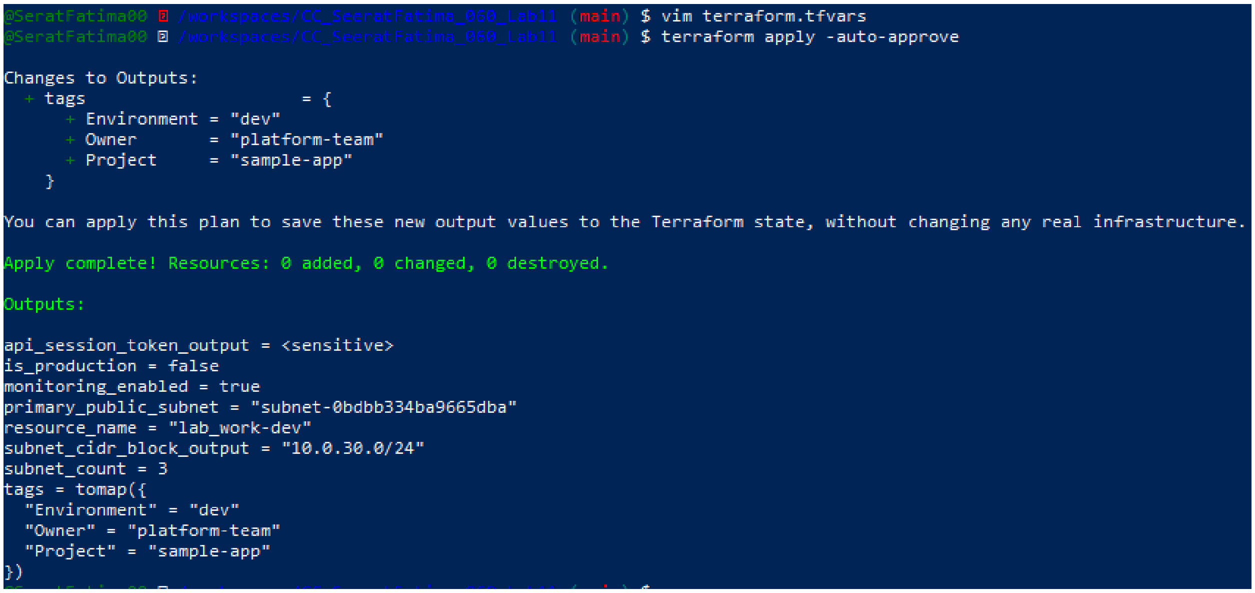


```
@SeratFatima00 ⍰ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.

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-0bdbb334ba9665dba"
resource_name = "lab_work-dev"
subnet_cidr_block_output = "10.0.30.0/24"
subnet_count = 3
```

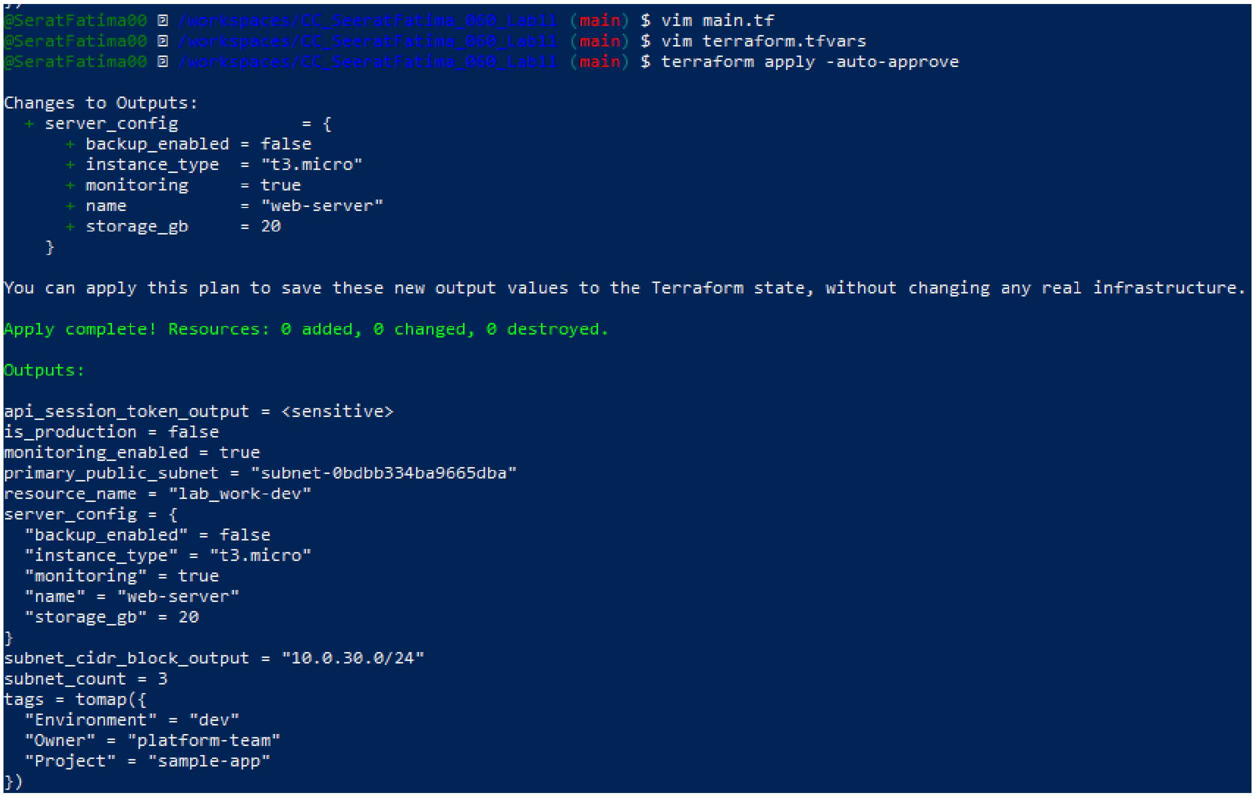
Task 4 — Maps and Objects



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

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

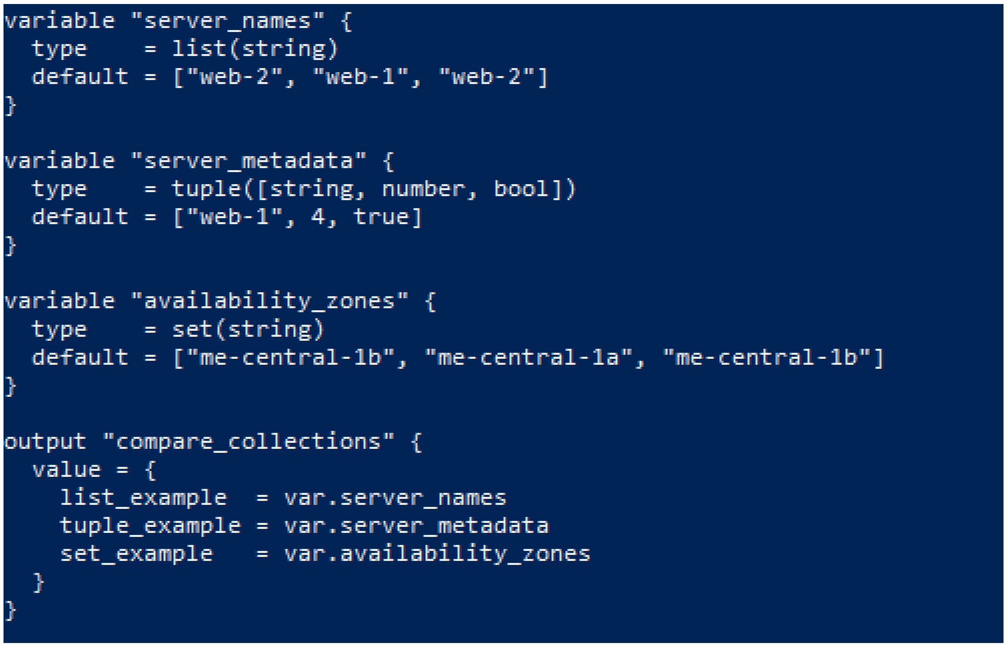


```
@SeratFatima00 ⊡ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ vim terraform.tfvars
@SeratFatima00 ⊡ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve

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-0bdbb334ba9665dba"
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"
})
```

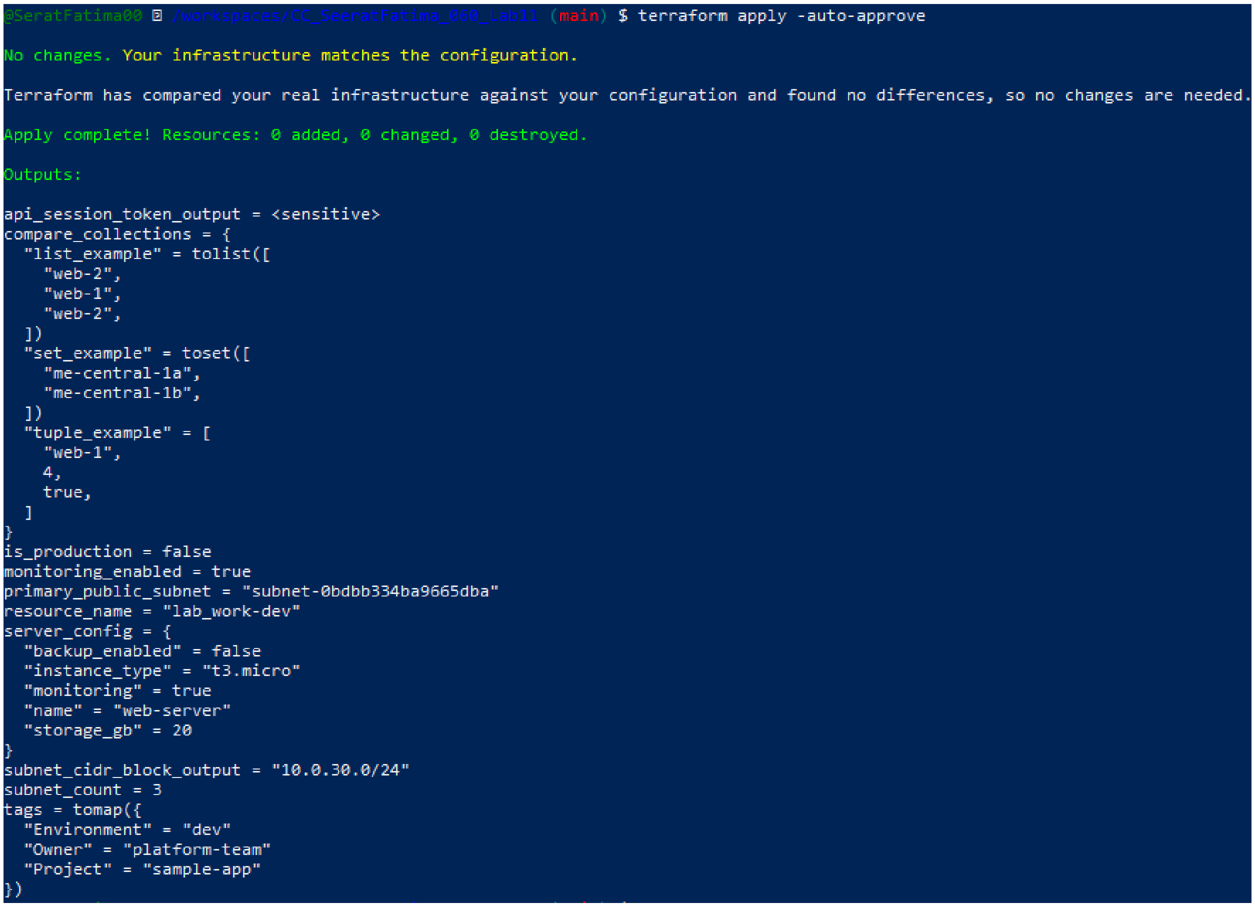


```
@SeratFatima00 ⊡ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ vim main.tf
@SeratFatima00 ⊡ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ vim terraform.tfvars
@SeratFatima00 ⊡ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve

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-0bdbb334ba9665dba"
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"
  "Project" = "sample-app"
})
```

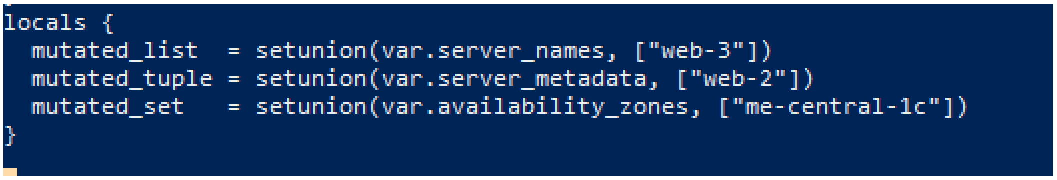
Task 5 — Collections: list, tuple, set & mutation via locals



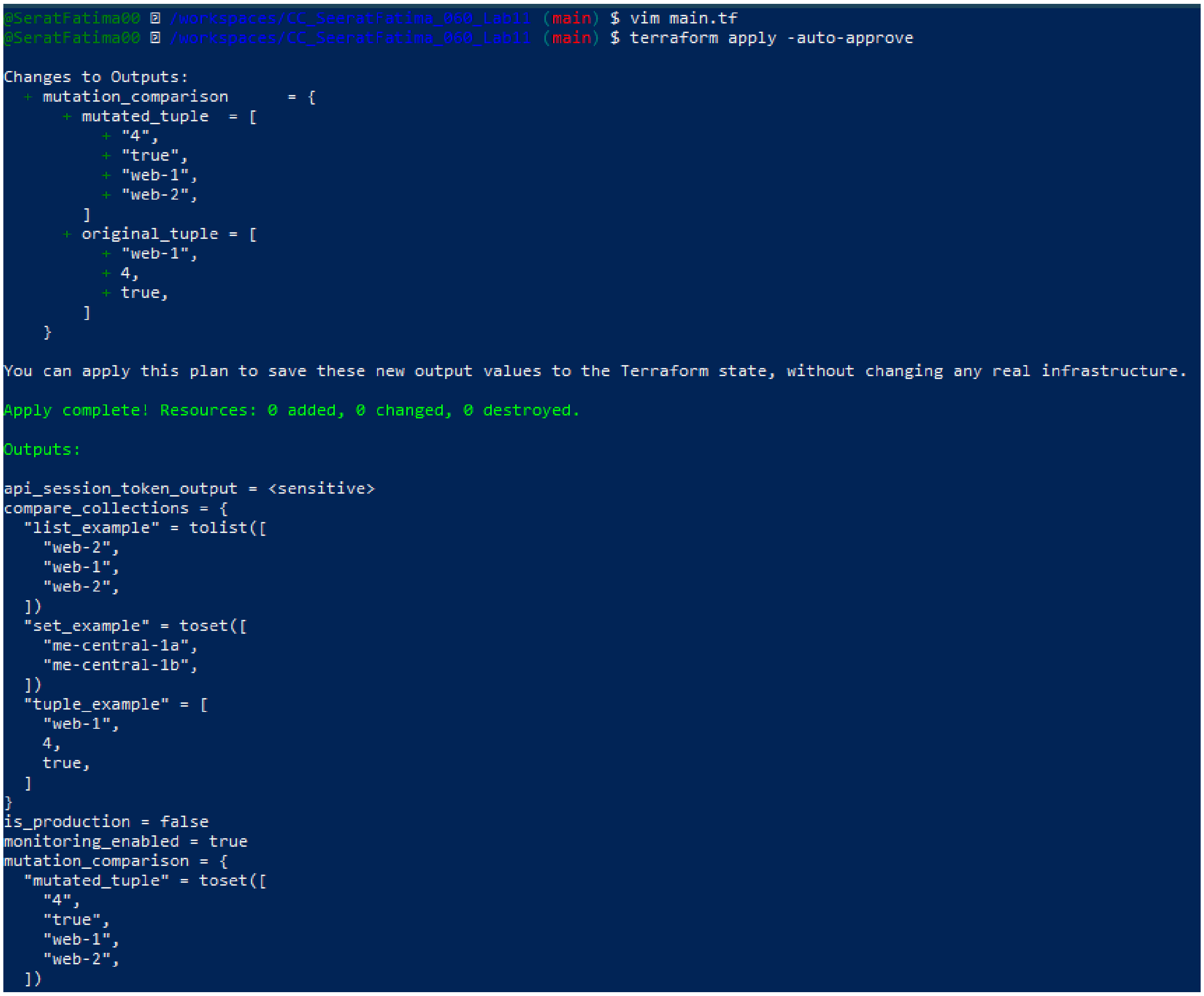
```
variable "server_names" {
  type    = list(string)
  default = ["web-2", "web-1", "web-2"]
}

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
  }
}
```



```
@SeratFatima00 ⊠ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.

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-0bdbb334ba9665dba"
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"
  "Project" = "sample-app"
})
```

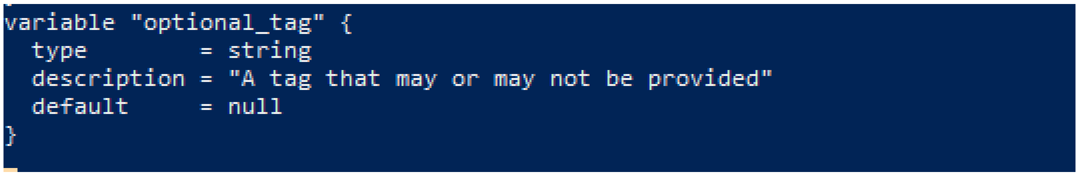


```
locals {
  mutated_list  = setunion(var.server_names, ["web-3"])
  mutated_tuple = setunion(var.server_metadata, ["web-2"])
  mutated_set   = setunion(var.availability_zones, ["me-central-1c"])
}
```

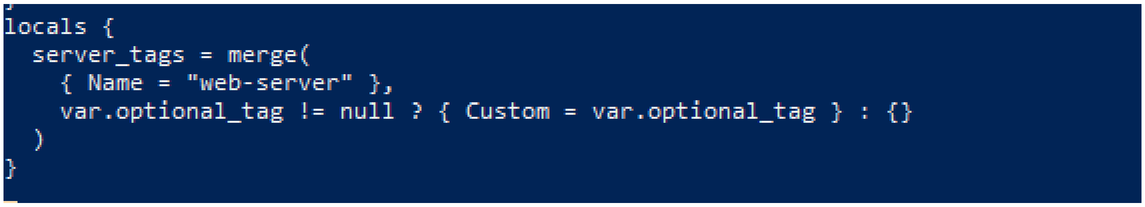


```
@SeratFatima00 ⊠ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ vim main.tf
@SeratFatima00 ⊠ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve

Changes to Outputs:
  + mutation_comparison      = {
      + mutated_tuple  = [
          + "4",
          + "true",
          + "web-1",
          + "web-2",
        ]
      + original_tuple = [
          + "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
mutation_comparison = {
  "mutated_tuple" = toset([
    "4",
    "true",
    "web-1",
    "web-2",
  ])
```

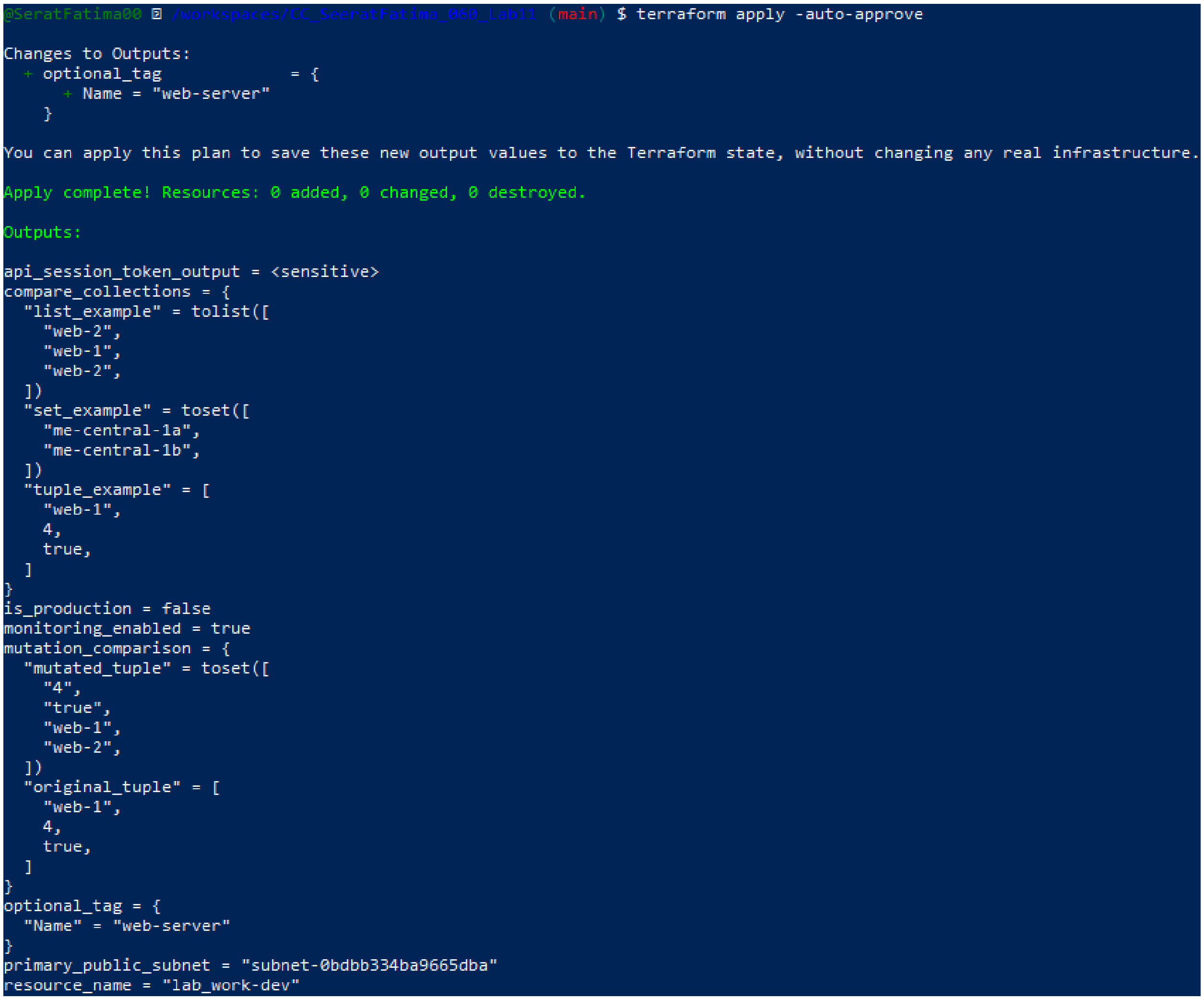
Task 6 — Null, any type & dynamic values



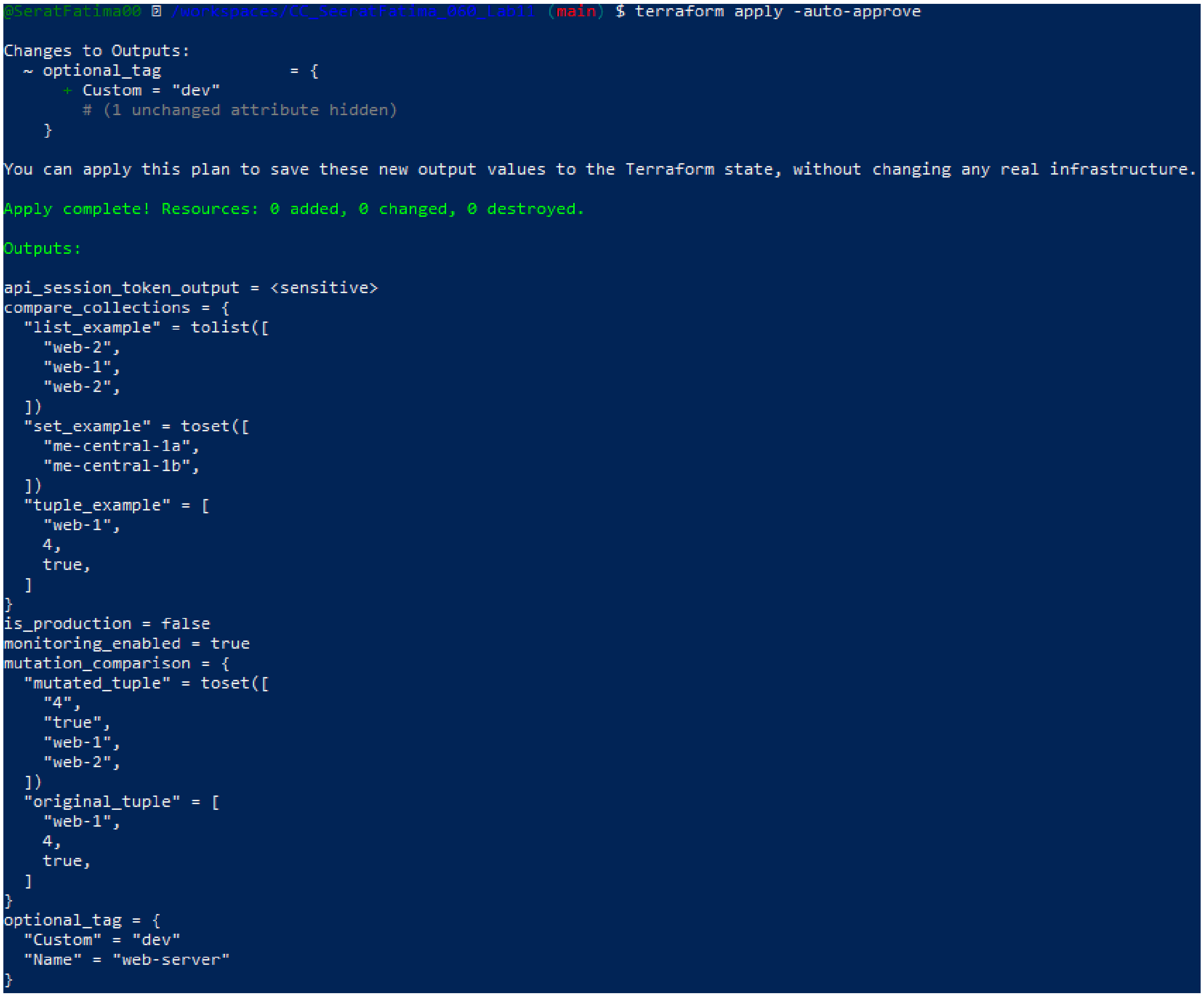
```
variable "optional_tag" {
  type        = string
  description = "A tag that may or may not be provided"
  default     = null
}
```



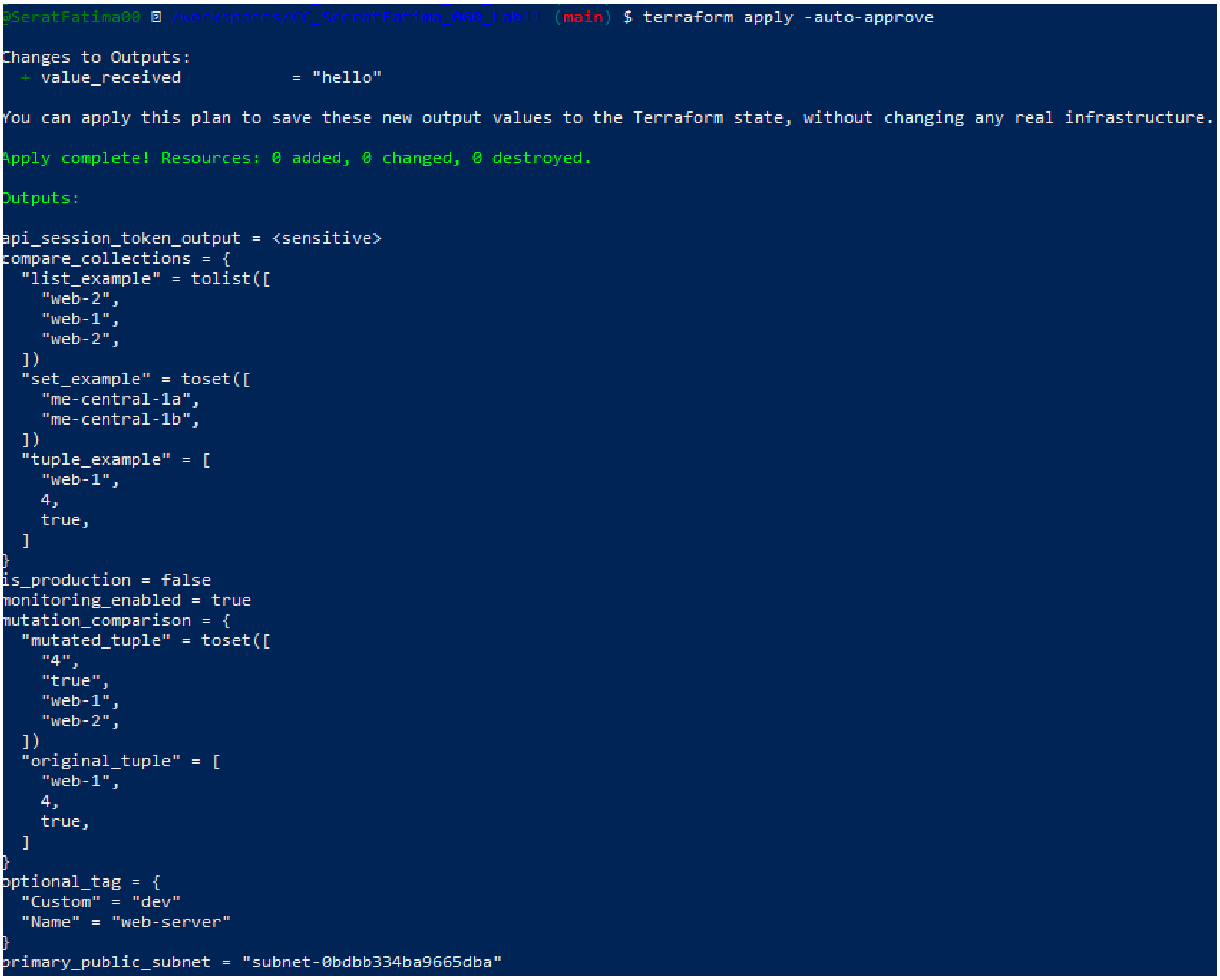
```
locals {
  server_tags = merge(
    { Name = "web-server" },
    var.optional_tag != null ? { Custom = var.optional_tag } : {}
  )
}
```



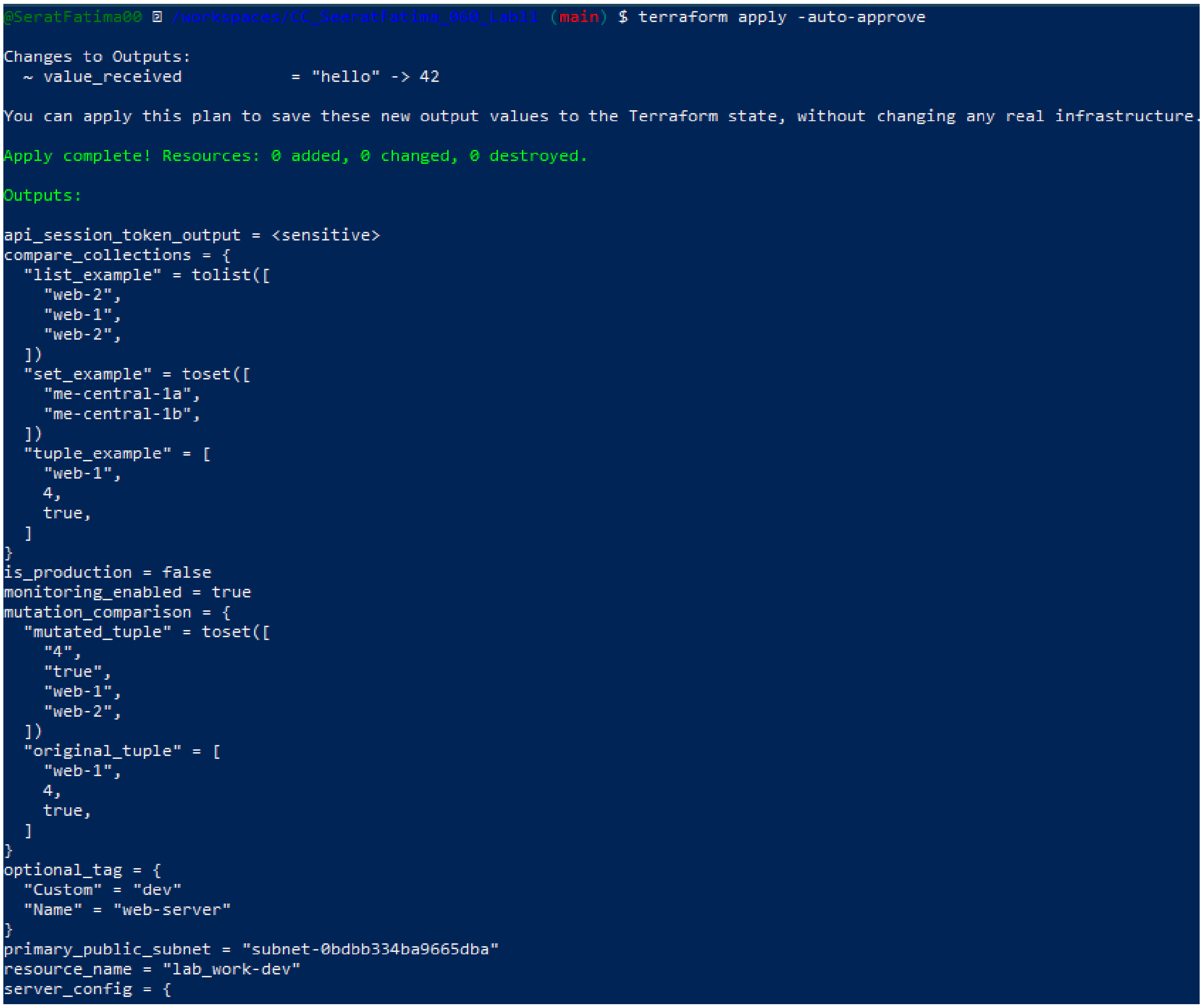
```
@SeratFatima00 ➜ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve

Changes to Outputs:
  + optional_tag              = {
      + Name = "web-server"
    }

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
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-0bdbb334ba9665dba"
resource_name = "lab_work-dev"
```



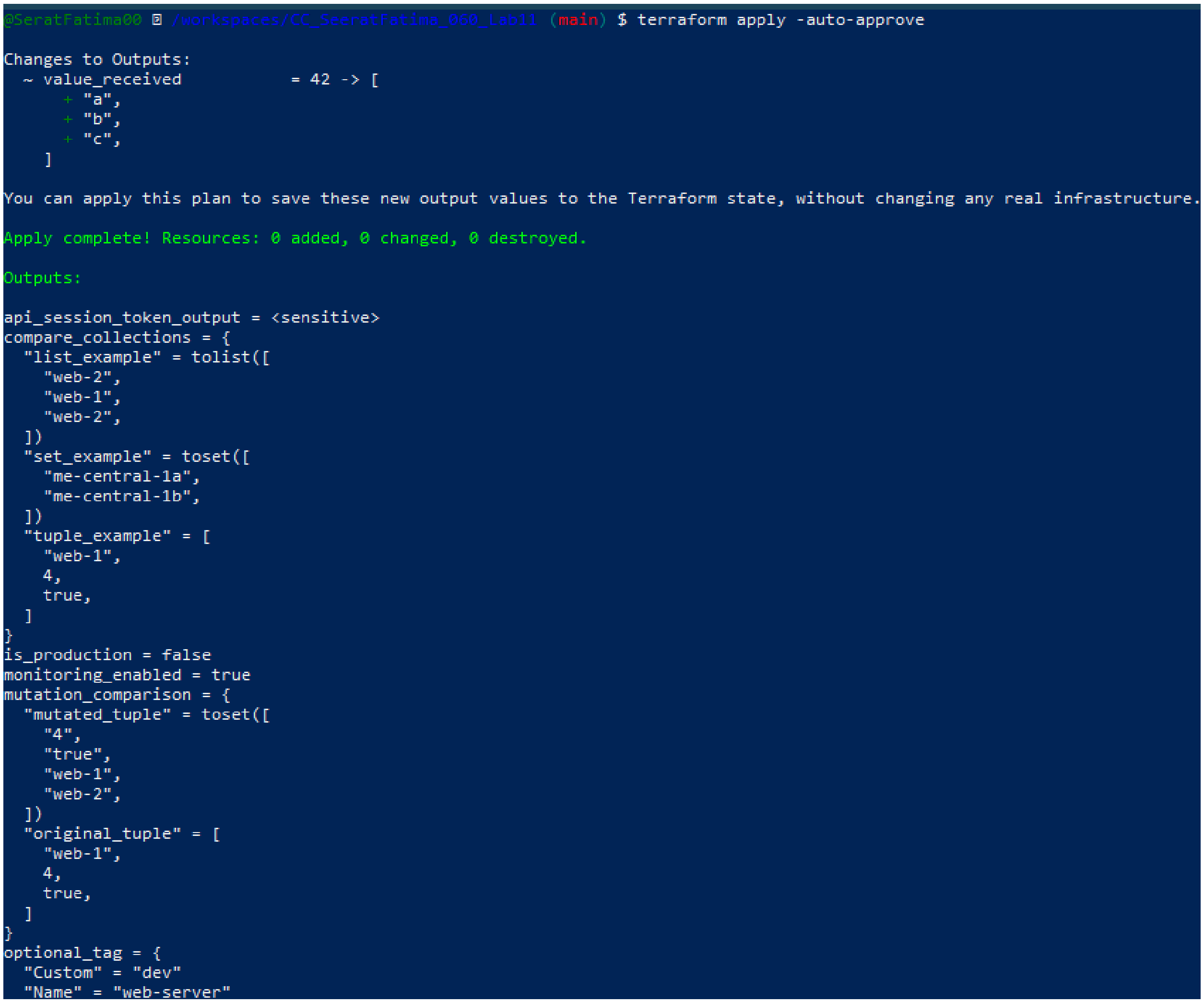
```
@SeratFatima00 ⍰ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve

Changes to Outputs:
  ~ optional_tag              = {
      + Custom = "dev"
        # (1 unchanged attribute hidden)
    }

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
mutation_comparison = {
  "mutated_tuple" = toset([
    "4",
    "true",
    "web-1",
    "web-2",
  ])
  "original_tuple" = [
    "web-1",
    4,
    true,
  ]
}
optional_tag = {
  "Custom" = "dev"
  "Name" = "web-server"
}
```



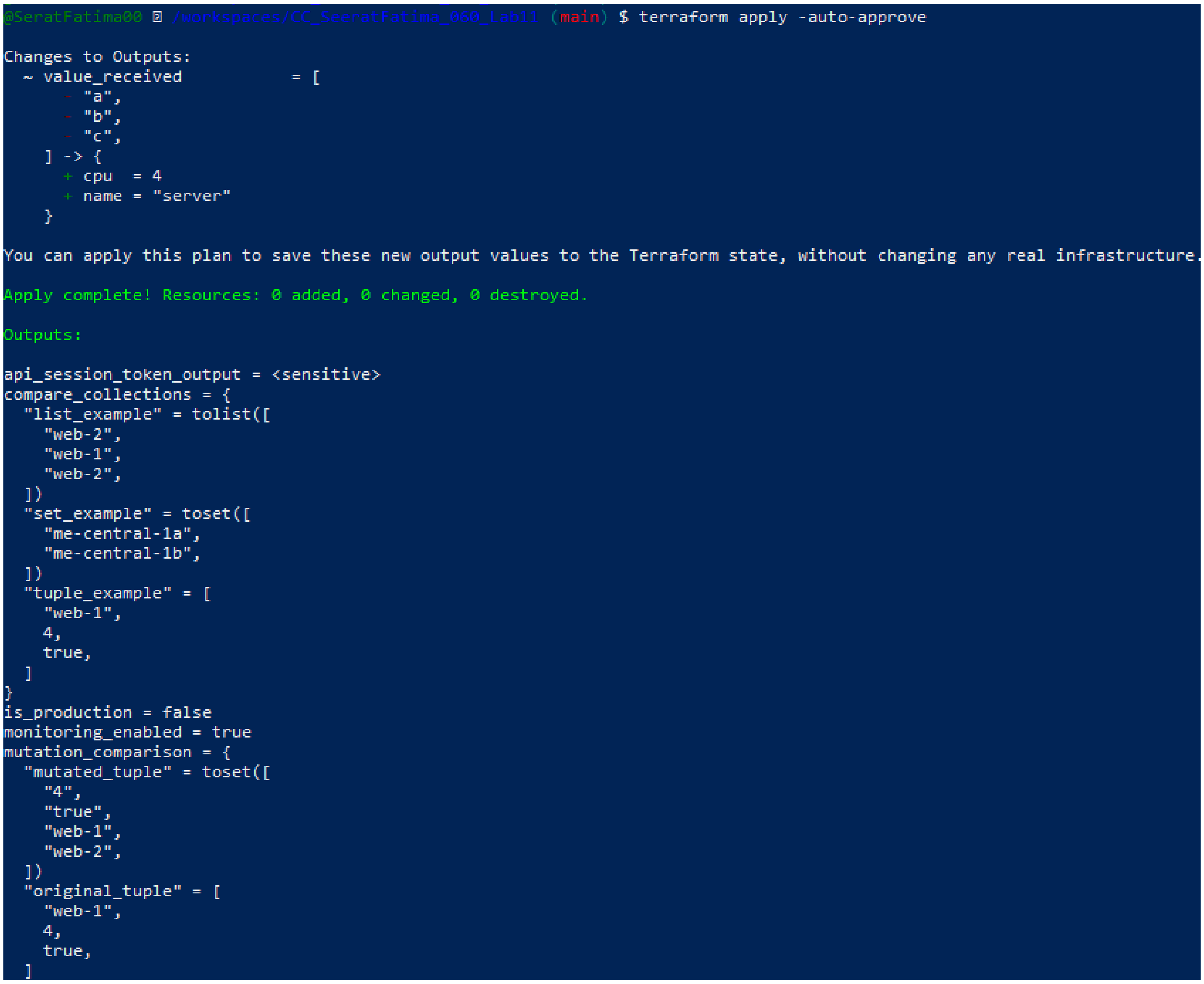
```
@SeratFatima00 ⊡ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve

Changes to Outputs:
  + value_received           = "hello"

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
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-0bdbb334ba9665dba"
```



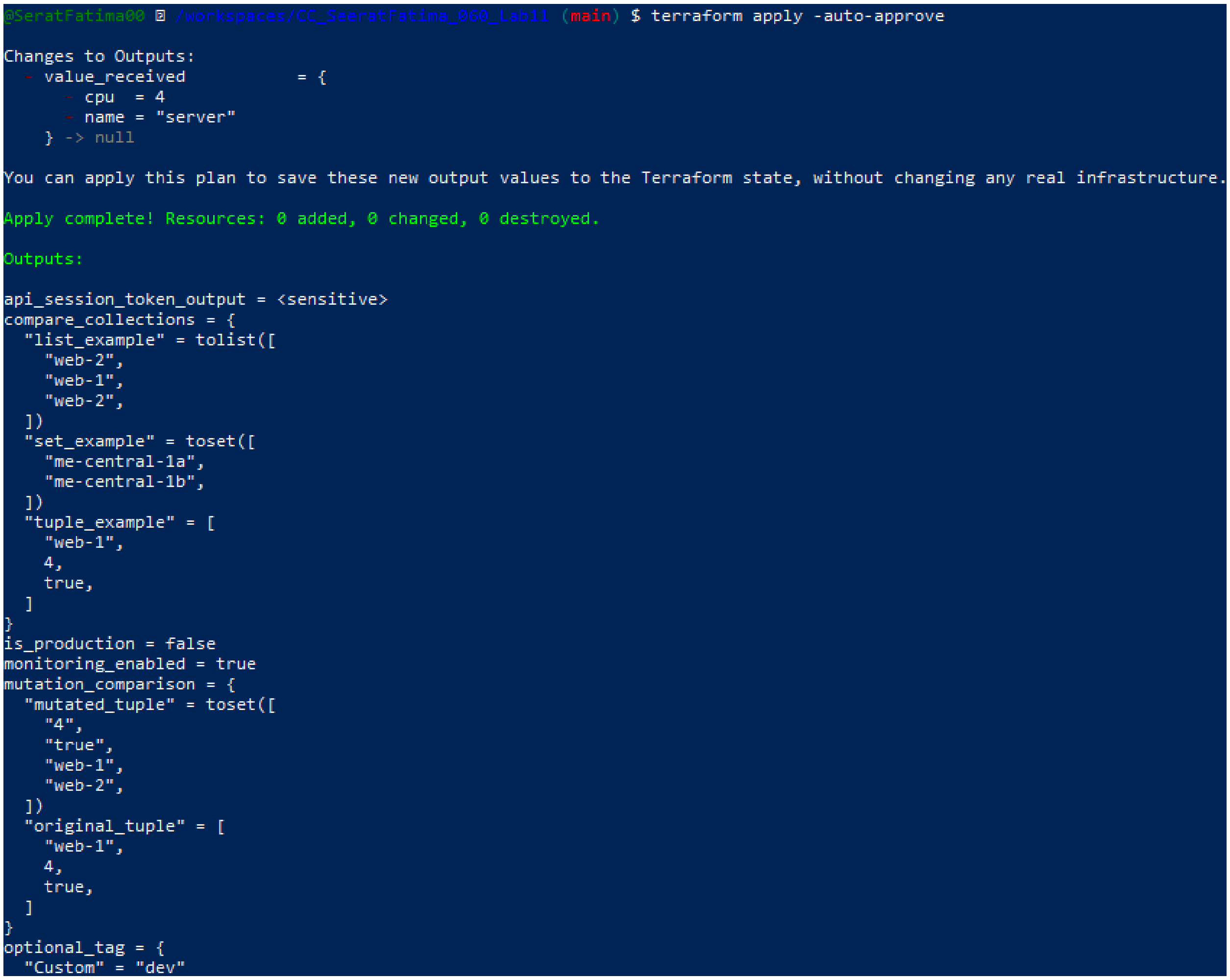
```
@SeratFatima00 ⍰ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve

Changes to Outputs:
  ~ value_received           = "hello" -> 42

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
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-0bdbb334ba9665dba"
resource_name = "lab_work-dev"
server_config = {
```



```
@SeratFatima00 ⍰ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve

Changes to Outputs:
  ~ value_received           = 42 -> [
      + "a",
      + "b",
      + "c",
    ]

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
mutation_comparison = {
  "mutated_tuple" = toset([
    "4",
    "true",
    "web-1",
    "web-2",
  ])
  "original_tuple" = [
    "web-1",
    4,
    true,
  ]
}
optional_tag = {
  "Custom" = "dev"
  "Name" = "web-server"
```

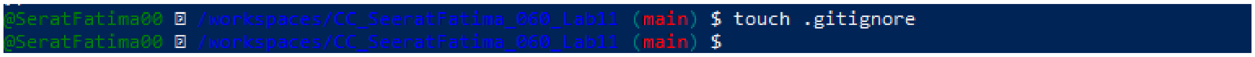


```
@SeratFatima00 ⍰ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve

Changes to Outputs:
  ~ value_received           = [
      - "a",
      - "b",
      - "c",
    ] -> {
      + cpu  = 4
      + name = "server"
    }

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
mutation_comparison = {
  "mutated_tuple" = toset([
    "4",
    "true",
    "web-1",
    "web-2",
  ])
  "original_tuple" = [
    "web-1",
    4,
    true,
  ]
```

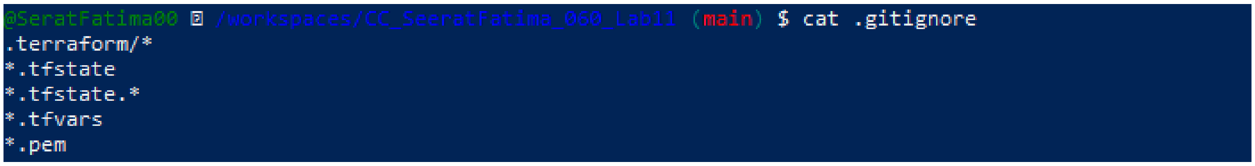


```
@SeratFatima00 ⍰ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve

Changes to Outputs:
  - value_received           = {
      - cpu  = 4
      - name = "server"
    } -> 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>
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 = {
  "Custom" = "dev"
```

Task 7 — Git ignore

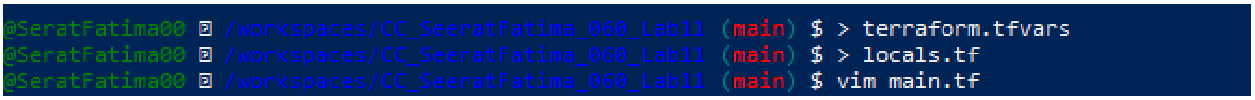


```
@SeratFatima00 ⍰ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ touch .gitignore
@SeratFatima00 ⍰ /workspaces/CC_SeeratFatima_060_Lab11 (main) $
```

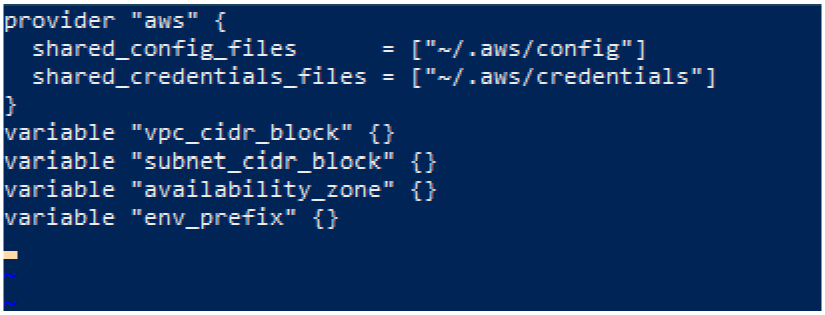


```
@SeratFatima00 ⍰ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ cat .gitignore
.terraform/*
*.tfstate
*.tfstate.*
*.tfvars
*.pem
```

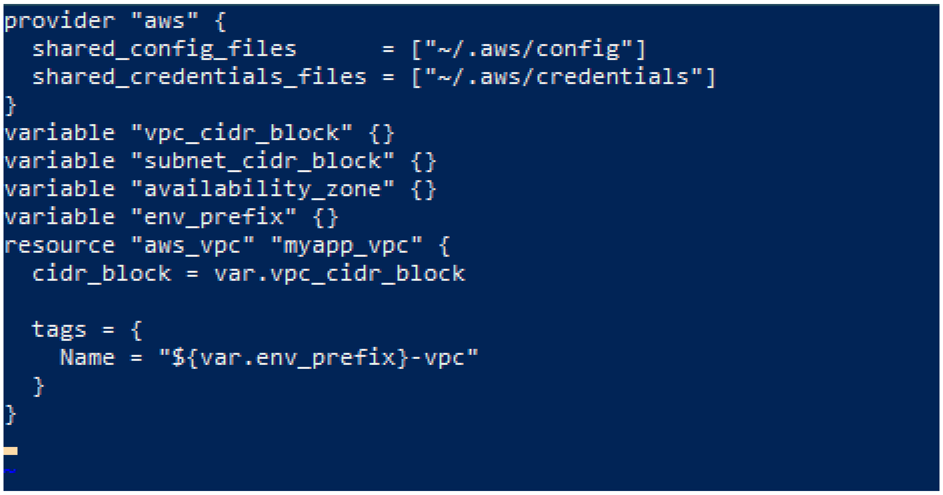
Task 8 — Clean-up then build real infra (VPC, Subnet, IGW, routing, default route table)



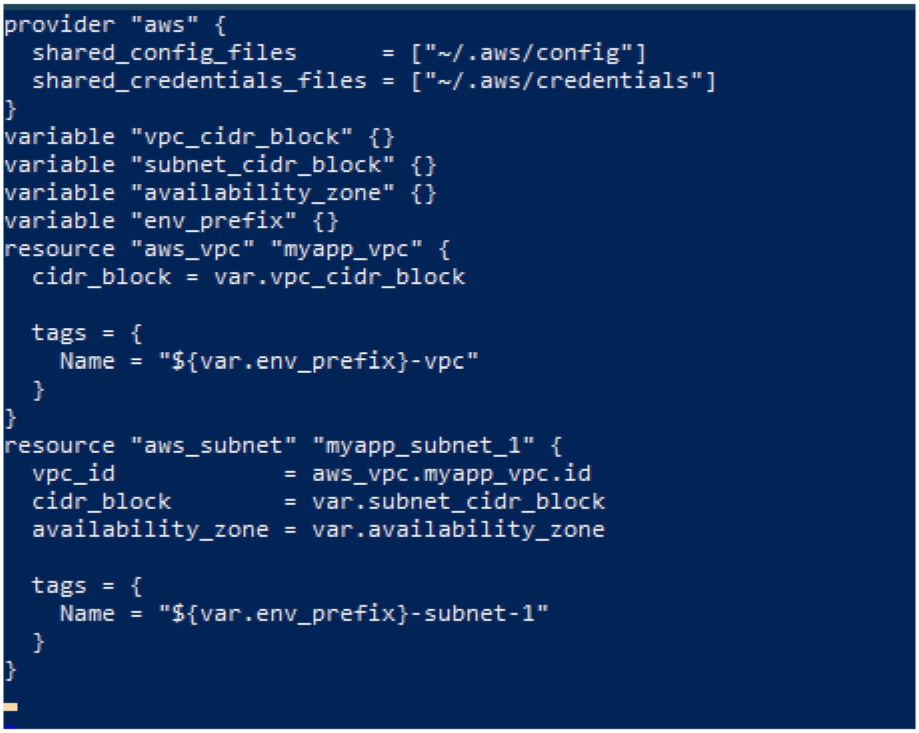
```
@SeratFatima00 □ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ > terraform.tfvars
@SeratFatima00 □ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ > locals.tf
@SeratFatima00 □ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ vim main.tf
```



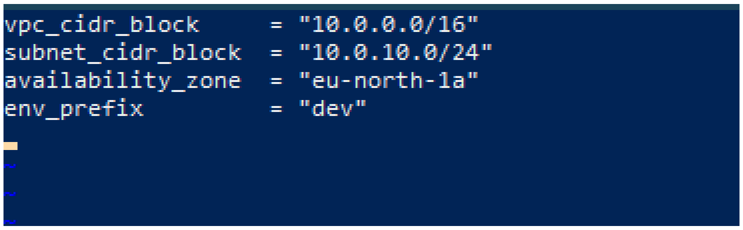
```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
variable "vpc_cidr_block" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}

~
~
```



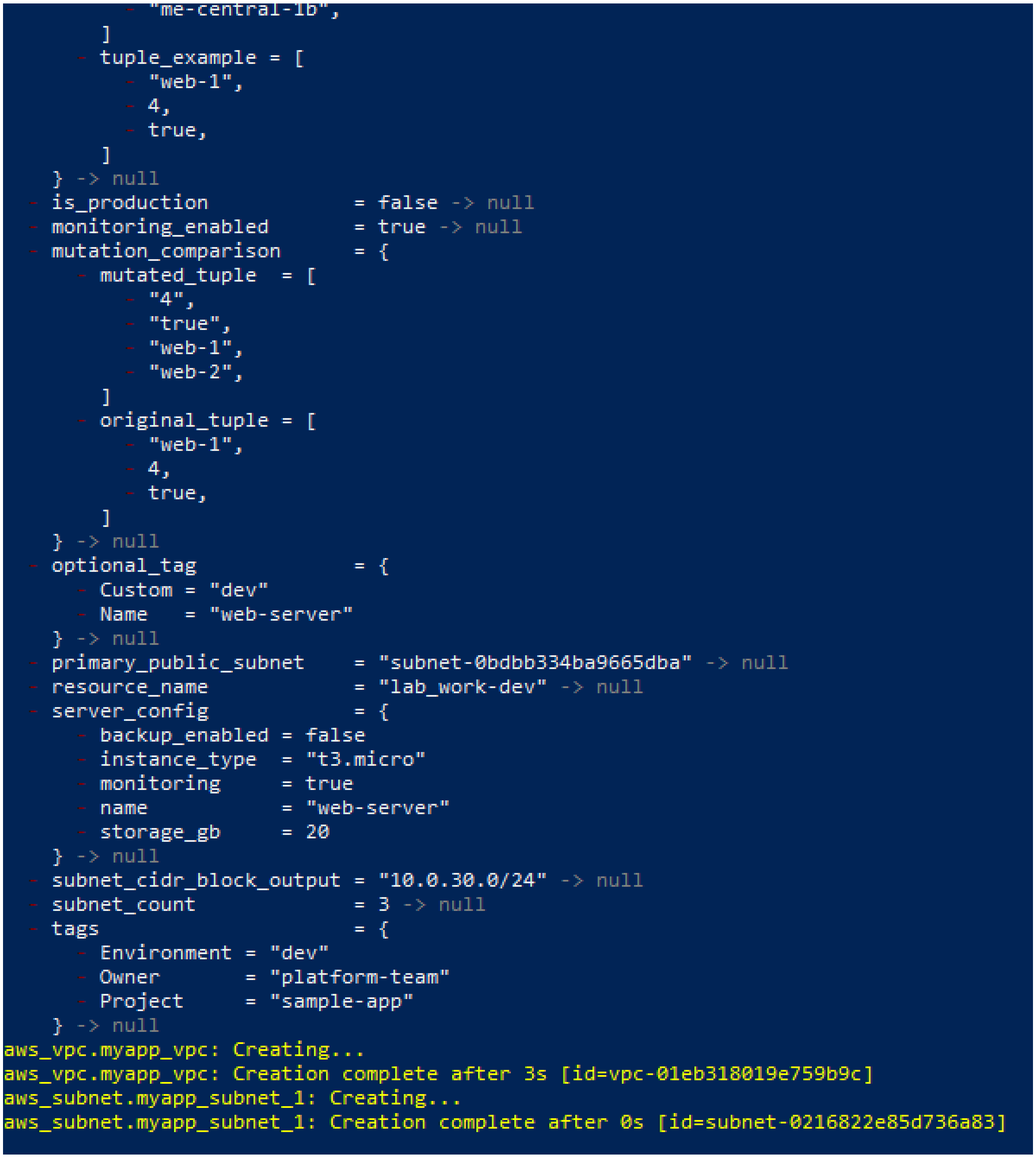
```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
variable "vpc_cidr_block" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
resource "aws_vpc" "myapp_vpc" {
  cidr_block = var.vpc_cidr_block

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

~
```



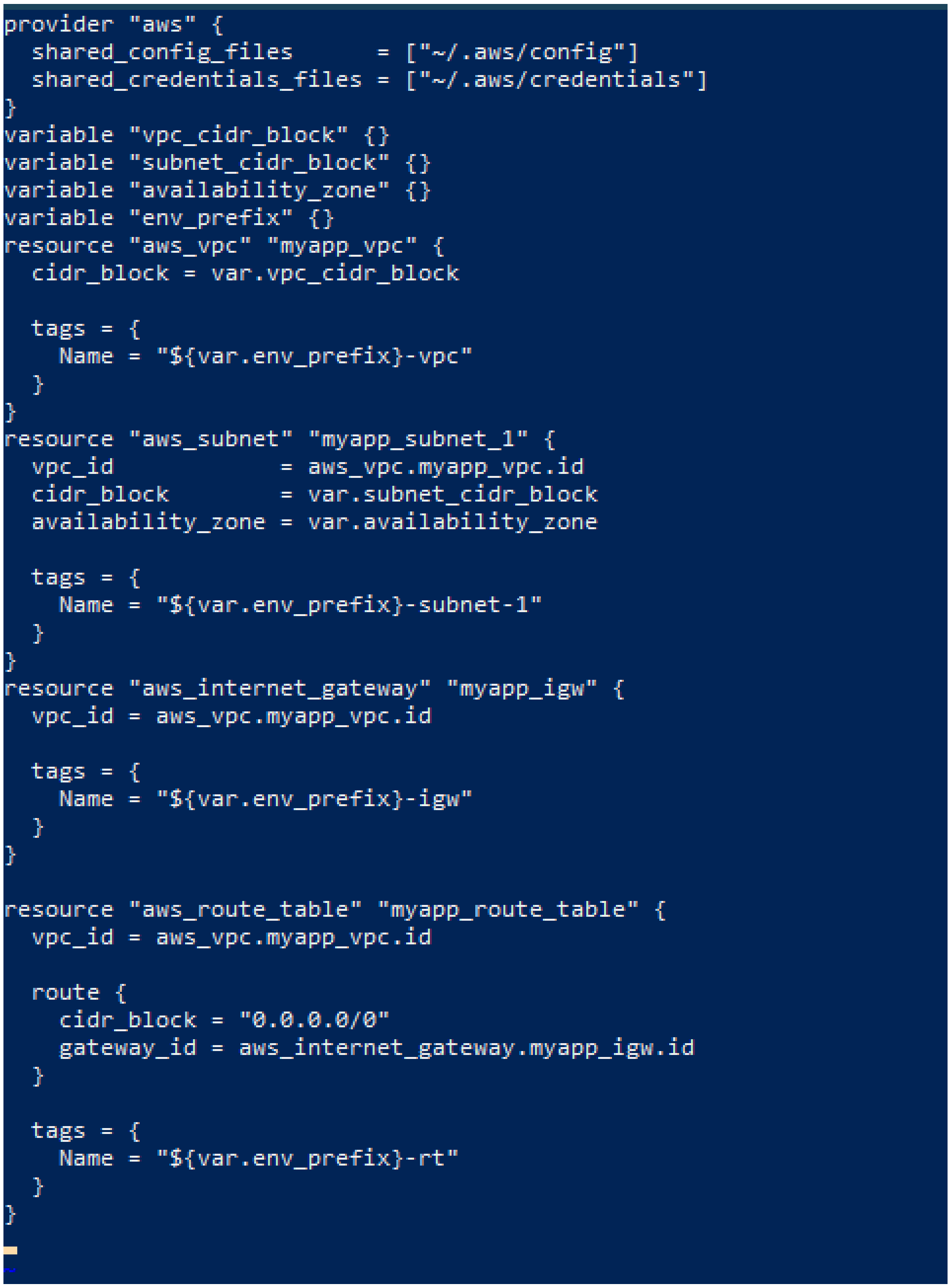
```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
variable "vpc_cidr_block" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
resource "aws_vpc" "myapp_vpc" {
  cidr_block = var.vpc_cidr_block

  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"
  }
}

~
```



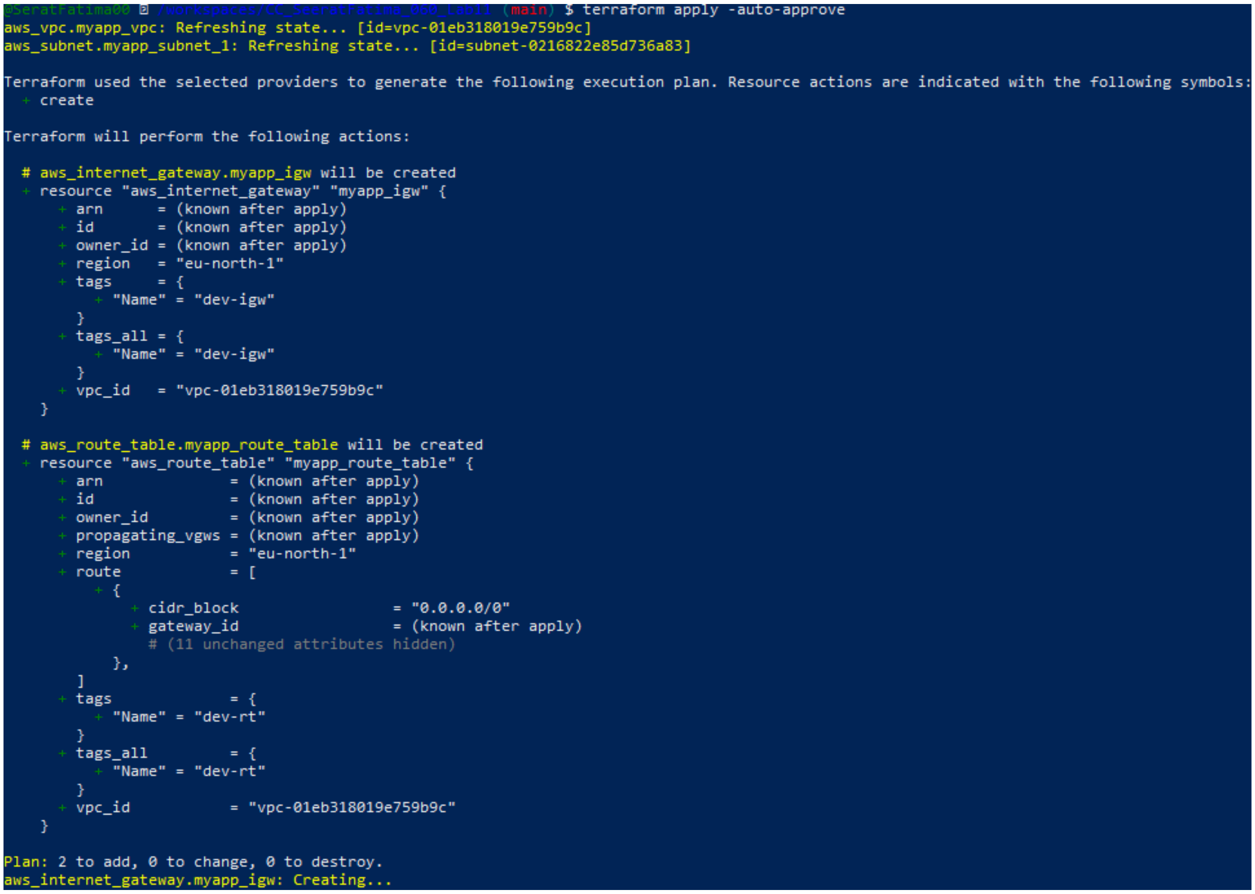
```
vpc_cidr_block     = "10.0.0.0/16"
subnet_cidr_block  = "10.0.10.0/24"
availability_zone  = "eu-north-1a"
env_prefix         = "dev"

~
~
~
```



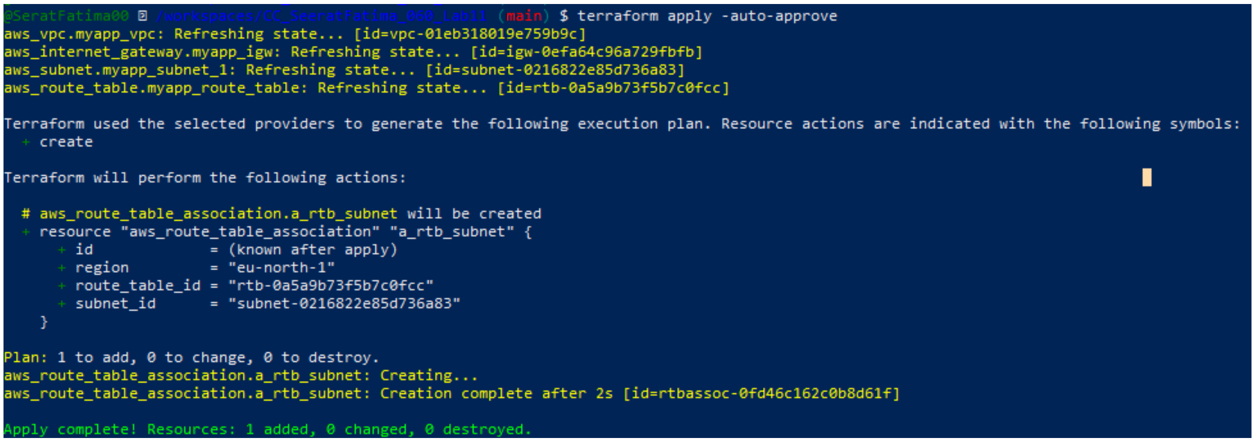
```
          - "me-central-1b",
        ]
      - tuple_example = [
          - "web-1",
          - 4,
          - true,
        ]
    } -> null
  - is_production            = false -> null
  - monitoring_enabled       = true -> null
  - mutation_comparison      = {
      - mutated_tuple  = [
          - "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-0bdbb334ba9665dba" -> 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_cidr_block_output = "10.0.30.0/24" -> 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 3s [id=vpc-01eb318019e759b9c]
aws_subnet.myapp_subnet_1: Creating...
aws_subnet.myapp_subnet_1: Creation complete after 0s [id=subnet-0216822e85d736a83]
```



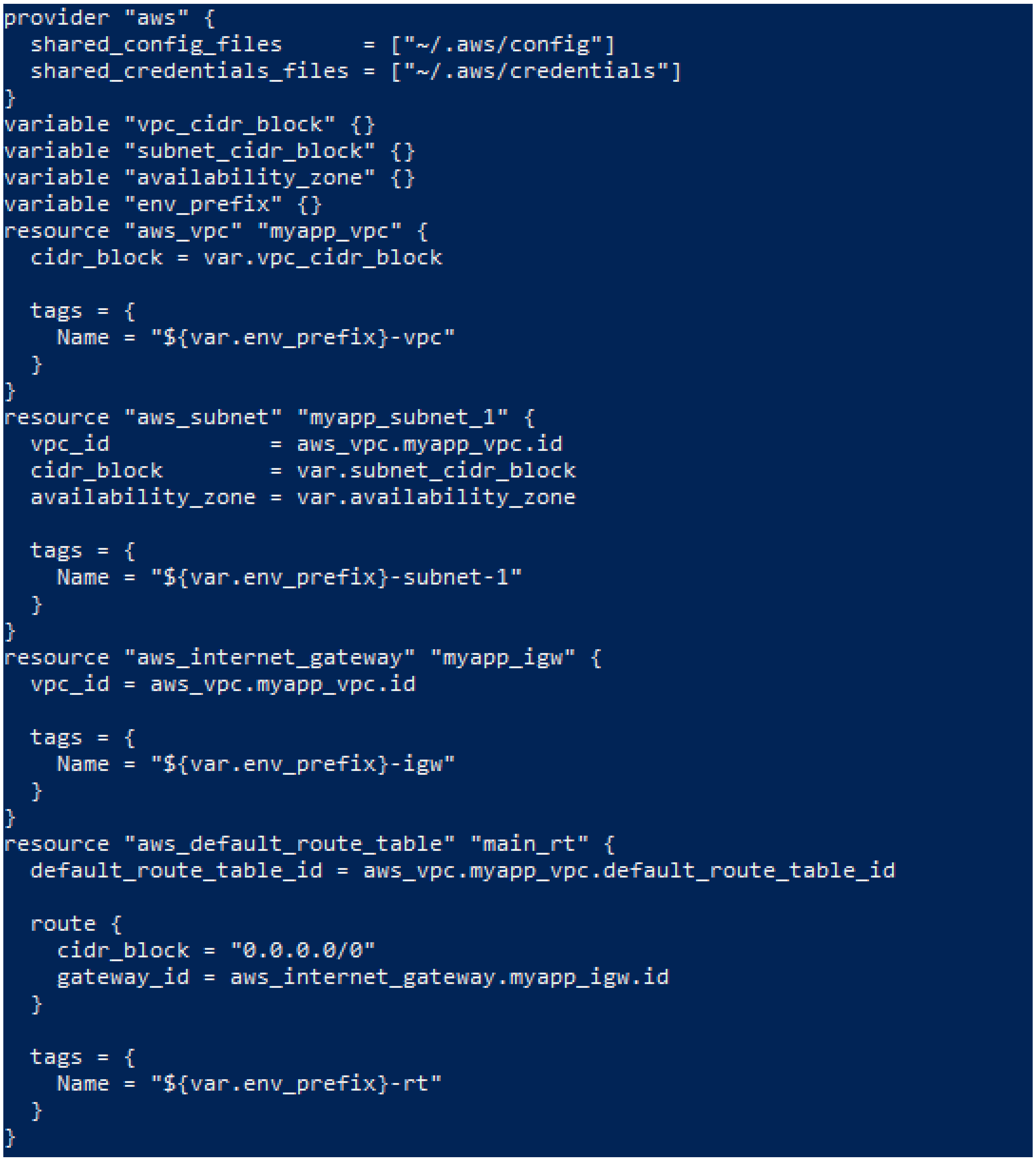
```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
variable "vpc_cidr_block" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
resource "aws_vpc" "myapp_vpc" {
  cidr_block = var.vpc_cidr_block

  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"
  }
}
```



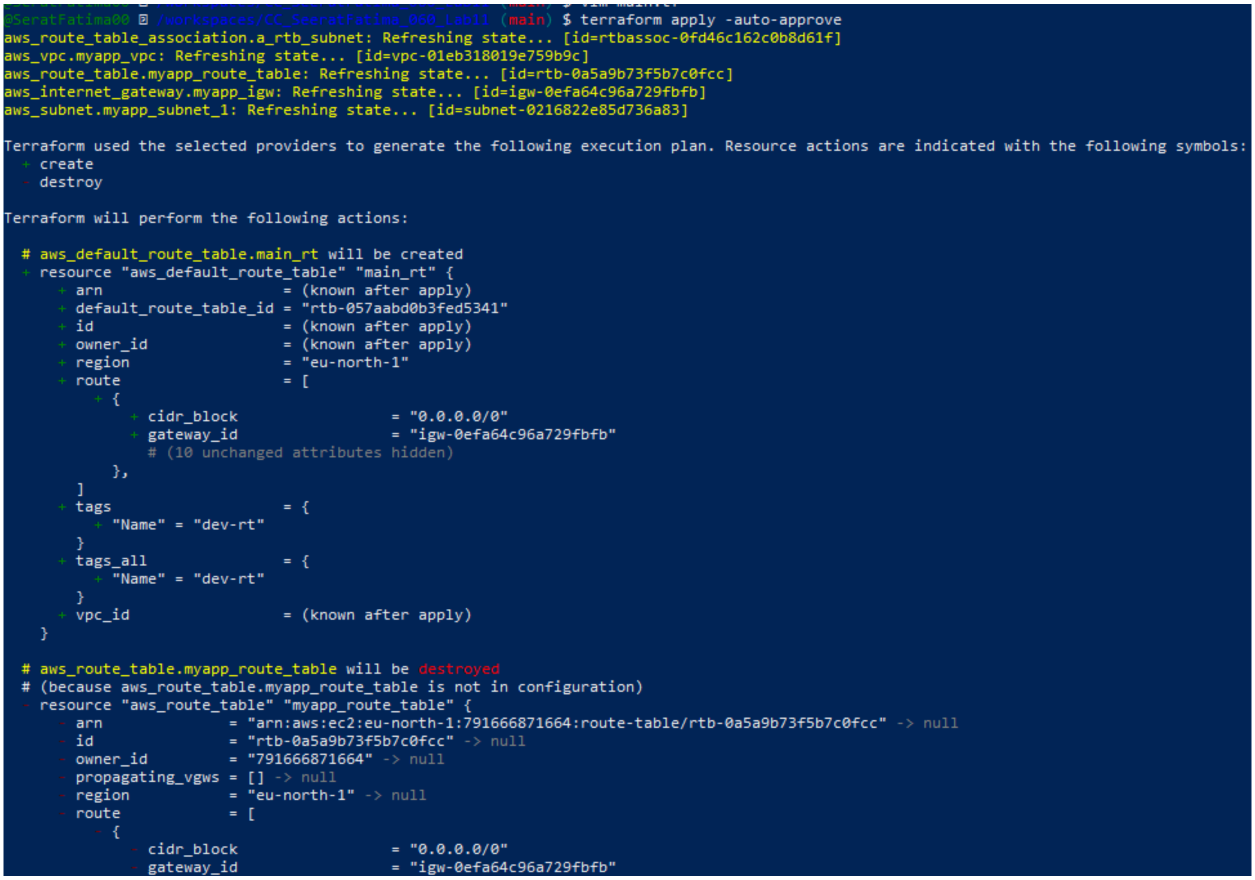
```
@SeratFatima00 ⊡ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-01eb318019e759b9c]
aws_subnet.myapp_subnet_1: Refreshing state... [id=subnet-0216822e85d736a83]

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_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   = "eu-north-1"
      + tags     = {
          + "Name" = "dev-igw"
        }
      + tags_all = {
          + "Name" = "dev-igw"
        }
      + vpc_id   = "vpc-01eb318019e759b9c"
    }

  # 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           = "eu-north-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-01eb318019e759b9c"
    }

Plan: 2 to add, 0 to change, 0 to destroy.
aws_internet_gateway.myapp_igw: Creating...
```



```
@SeratFatima00 ⊡ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-01eb318019e759b9c]
aws_internet_gateway.myapp_igw: Refreshing state... [id=igw-0efa64c96a729fbfb]
aws_subnet.myapp_subnet_1: Refreshing state... [id=subnet-0216822e85d736a83]
aws_route_table.myapp_route_table: Refreshing state... [id=rtb-0a5a9b73f5b7c0fcc]

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         = "eu-north-1"
      + route_table_id = "rtb-0a5a9b73f5b7c0fcc"
      + subnet_id      = "subnet-0216822e85d736a83"
    }

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 2s [id=rtbassoc-0fd46c162c0b8d61f]

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

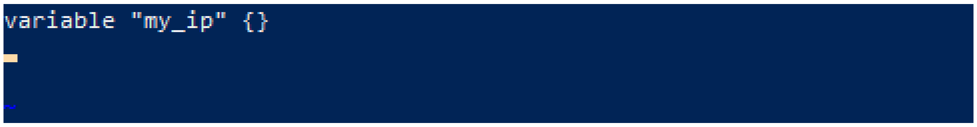


```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
variable "vpc_cidr_block" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
resource "aws_vpc" "myapp_vpc" {
  cidr_block = var.vpc_cidr_block

  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"
  }
}
```

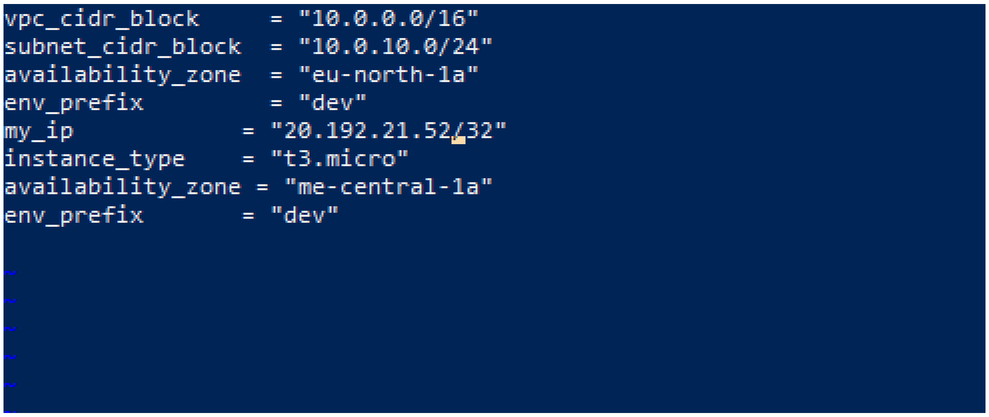


```
@SeratFatima00 ⍰ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve
aws_route_table_association.a_rtb_subnet: Refreshing state... [id=rtbassoc-0fd46c162c0b8d61f]
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-01eb318019e759b9c]
aws_route_table.myapp_route_table: Refreshing state... [id=rtb-0a5a9b73f5b7c0fcc]
aws_internet_gateway.myapp_igw: Refreshing state... [id=igw-0efa64c96a729fbfb]
aws_subnet.myapp_subnet_1: Refreshing state... [id=subnet-0216822e85d736a83]

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

Terraform will perform the following actions:

  # aws_default_route_table.main_rt will be created
  + resource "aws_default_route_table" "main_rt" {
      + arn                    = (known after apply)
      + default_route_table_id = "rtb-057aabd0b3fed5341"
      + id                     = (known after apply)
      + owner_id               = (known after apply)
      + region                 = "eu-north-1"
      + route                  = [
          + {
              + cidr_block                 = "0.0.0.0/0"
              + gateway_id                 = "igw-0efa64c96a729fbfb"
                # (10 unchanged attributes hidden)
            },
        ]
      + tags                   = {
          + "Name" = "dev-rt"
        }
      + 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:eu-north-1:791666871664:route-table/rtb-0a5a9b73f5b7c0fcc" -> null
      - id               = "rtb-0a5a9b73f5b7c0fcc" -> null
      - owner_id         = "791666871664" -> null
      - propagating_vgws = [] -> null
      - region           = "eu-north-1" -> null
      - route            = [
          - {
              - cidr_block                 = "0.0.0.0/0"
              - gateway_id                 = "igw-0efa64c96a729fbfb"
```

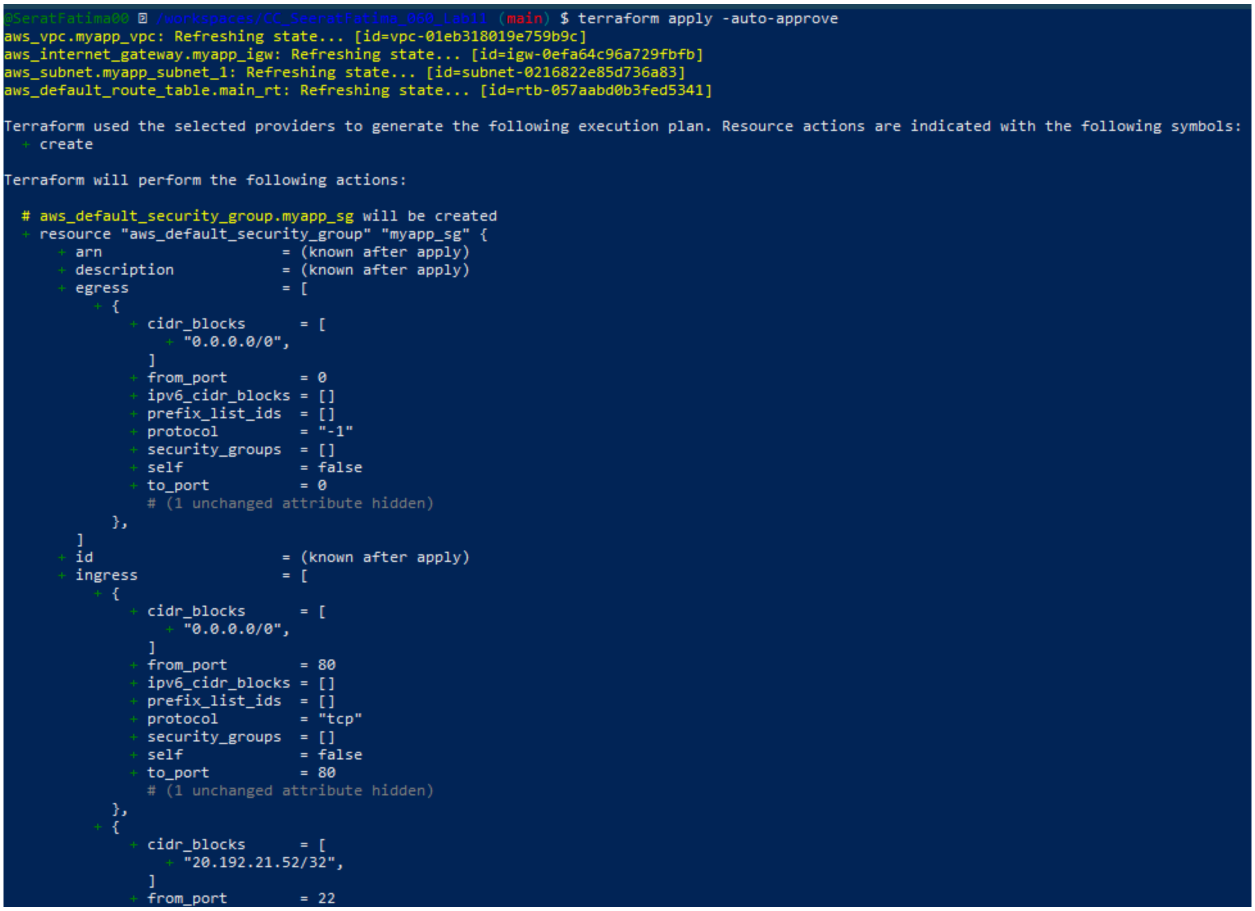
Task 9 — Security Group, Key Pair, EC2 Instance, user_data & nginx



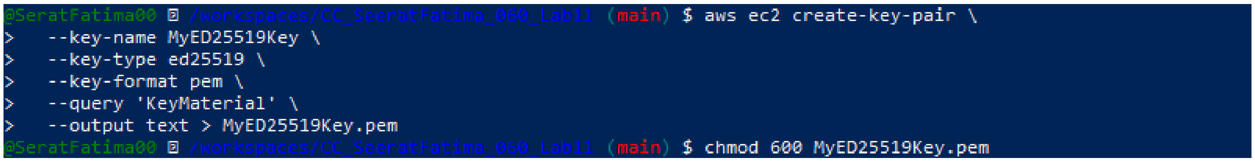
```
variable "my_ip" {}

~
```



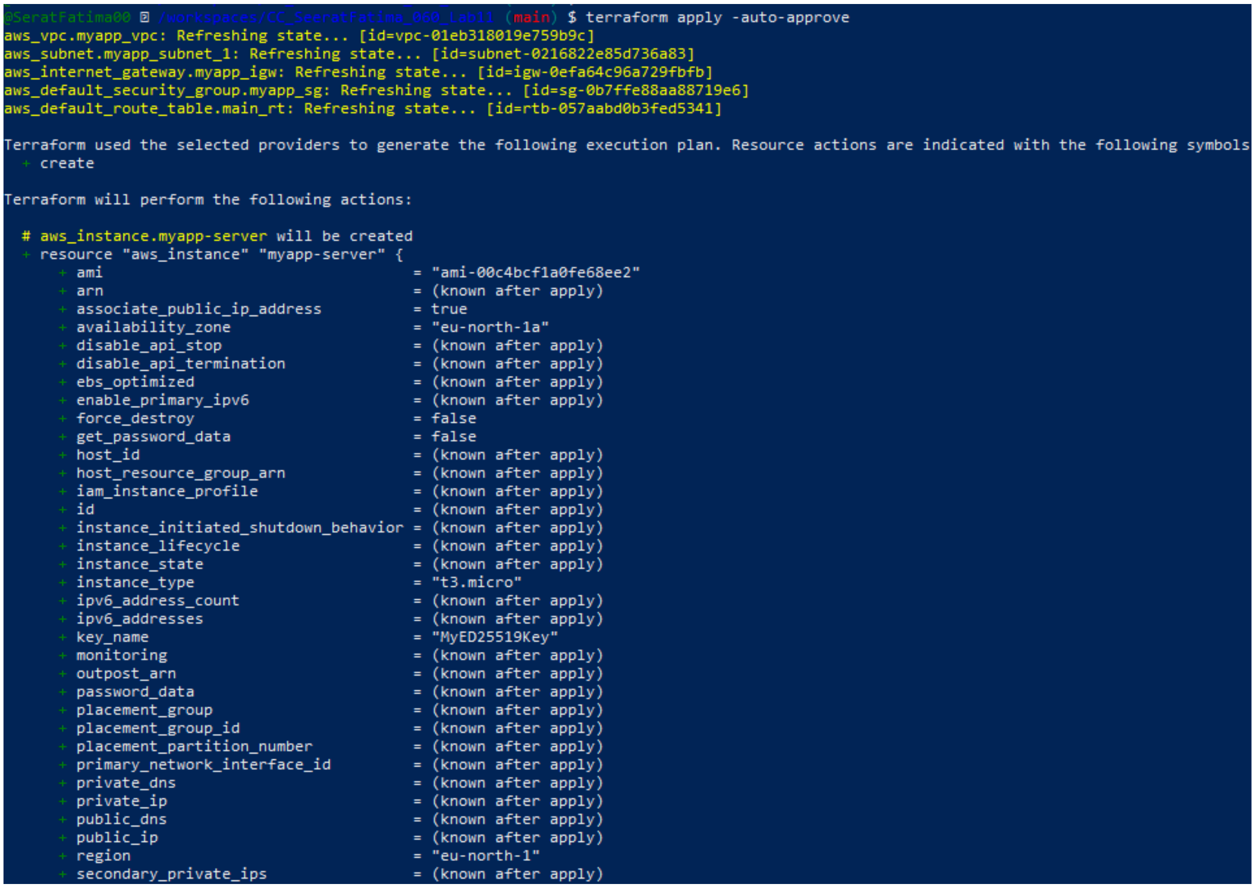
```
vpc_cidr_block      = "10.0.0.0/16"
subnet_cidr_block   = "10.0.10.0/24"
availability_zone   = "eu-north-1a"
env_prefix          = "dev"
my_ip             = "20.192.21.52/32"
instance_type     = "t3.micro"
availability_zone = "me-central-1a"
env_prefix        = "dev"

~
~
~
~
~
~
```



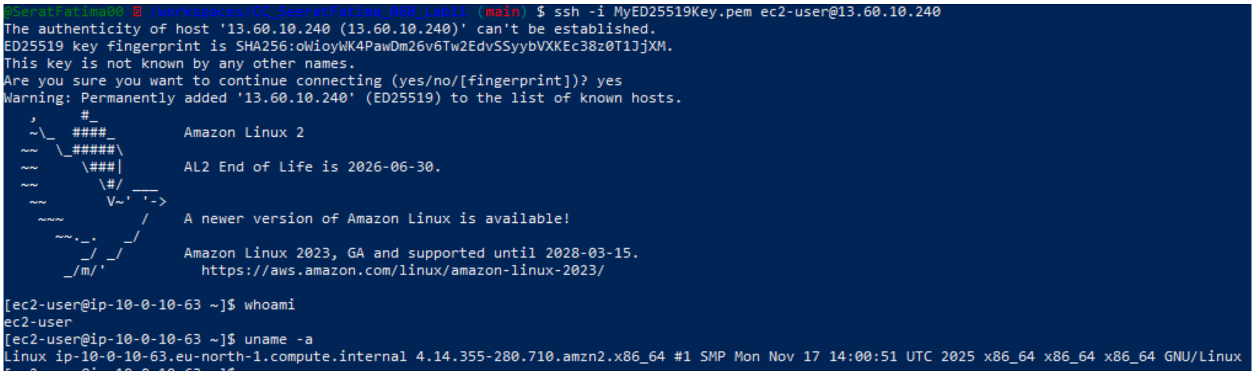
```
@SeratFatima00 ⍰ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-01eb318019e759b9c]
aws_internet_gateway.myapp_igw: Refreshing state... [id=igw-0efa64c96a729fbfb]
aws_subnet.myapp_subnet_1: Refreshing state... [id=subnet-0216822e85d736a83]
aws_default_route_table.main_rt: Refreshing state... [id=rtb-057aabd0b3fed5341]

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_default_security_group.myapp_sg will be created
  + resource "aws_default_security_group" "myapp_sg" {
      + arn                    = (known after apply)
      + description            = (known after apply)
      + egress                 = [
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + from_port        = 0
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "-1"
              + security_groups  = []
              + self             = false
              + to_port          = 0
                # (1 unchanged attribute hidden)
            },
        ]
      + id                     = (known after apply)
      + ingress                = [
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + from_port        = 80
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 80
                # (1 unchanged attribute hidden)
            },
          + {
              + cidr_blocks      = [
                  + "20.192.21.52/32",
                ]
              + from_port        = 22
```



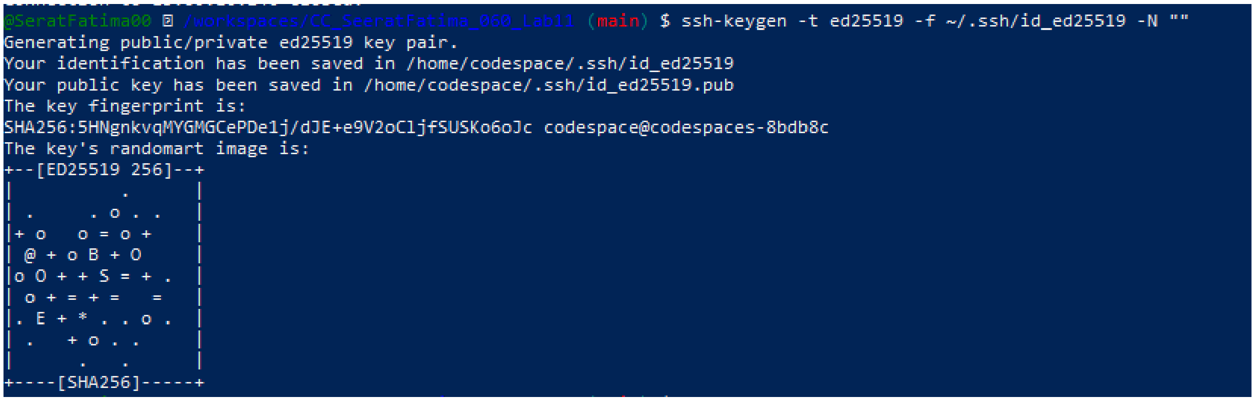
```
@SeratFatima00 ⍰ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ aws ec2 create-key-pair \
>   --key-name MyED25519Key \
>   --key-type ed25519 \
>   --key-format pem \
>   --query 'KeyMaterial' \
>   --output text > MyED25519Key.pem
@SeratFatima00 ⍰ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ chmod 600 MyED25519Key.pem
```



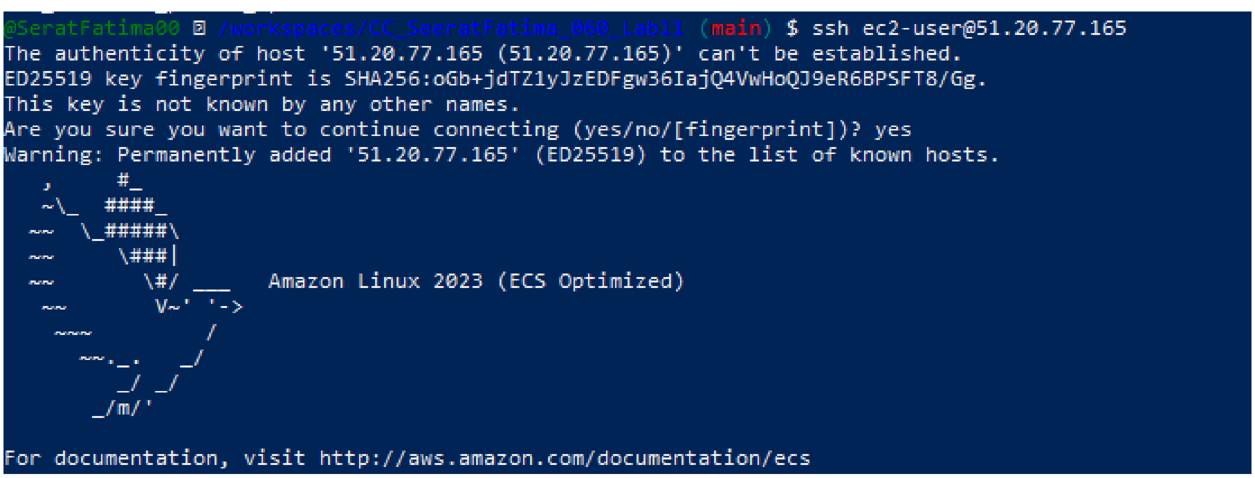
```
@SeratFatima00 ➜ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ terraform apply -auto-approve
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-01eb318019e759b9c]
aws_subnet.myapp_subnet_1: Refreshing state... [id=subnet-0216822e85d736a83]
aws_internet_gateway.myapp_igw: Refreshing state... [id=igw-0efa64c96a729fbfb]
aws_default_security_group.myapp_sg: Refreshing state... [id=sg-0b7ffe88aa88719e6]
aws_default_route_table.main_rt: Refreshing state... [id=rtb-057aabd0b3fed5341]

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_instance.myapp-server will be created
  + resource "aws_instance" "myapp-server" {
      + ami                                  = "ami-00c4bcf1a0fe68ee2"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = true
      + availability_zone                    = "eu-north-1a"
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + force_destroy                        = false
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t3.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "MyED25519Key"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_group_id                   = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + region                               = "eu-north-1"
      + secondary_private_ips                = (known after apply)
```



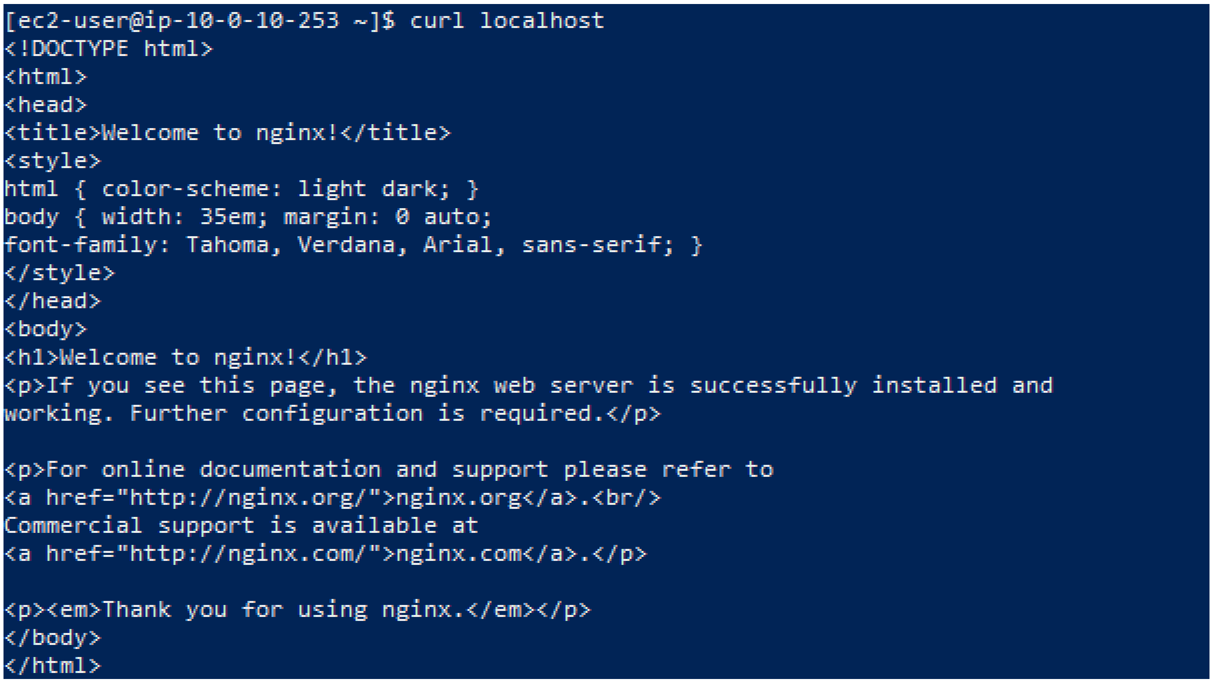
```
@SeratFatima00 ➜ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ ssh -i MyED25519Key.pem ec2-user@13.60.10.240
The authenticity of host '13.60.10.240 (13.60.10.240)' can't be established.
ED25519 key fingerprint is SHA256:oWioyWK4PawDm26v6Tw2EdvSSyybVXKEc38z0T1JjXM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.60.10.240' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-0-10-63 ~]$ whoami
ec2-user
[ec2-user@ip-10-0-10-63 ~]$ uname -a
Linux ip-10-0-10-63.eu-north-1.compute.internal 4.14.355-280.710.amzn2.x86_64 #1 SMP Mon Nov 17 14:00:51 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
```



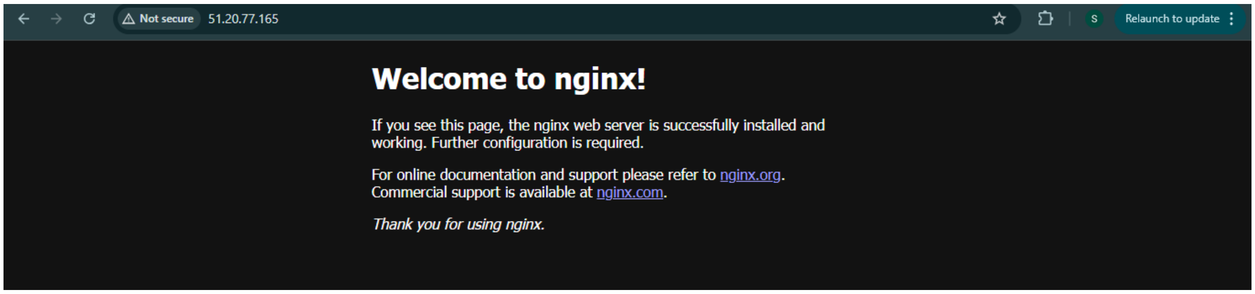
```
@SeratFatima00 ➜ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519 -N ""
Generating public/private ed25519 key pair.
Your identification has been saved in /home/codespace/.ssh/id_ed25519
Your public key has been saved in /home/codespace/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:5HNgnkvqMYGMGCePDe1j/dJE+e9V2oCljfSUSKo6oJc codespace@codespaces-8bdb8c
The key's randomart image is:
+--[ED25519 256]--+
|          .      |
| .     . o . .   |
|+ o   o = o +    |
| @ + o B + O     |
|o O + + S = + .  |
| o + = + =   =   |
|. E + * . . o .  |
| .   + o . .     |
|      .   .      |
+----[SHA256]-----+
```



```
@SeratFatima00 ⊠ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ ssh ec2-user@51.20.77.165
The authenticity of host '51.20.77.165 (51.20.77.165)' can't be established.
ED25519 key fingerprint is SHA256:oGb+jdTZ1yJzEDFgw36IajQ4VwHoQJ9eR6BPSFT8/Gg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '51.20.77.165' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   Amazon Linux 2023 (ECS Optimized)
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

For documentation, visit http://aws.amazon.com/documentation/ecs
```



```
[ec2-user@ip-10-0-10-253 ~]$ 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>
```



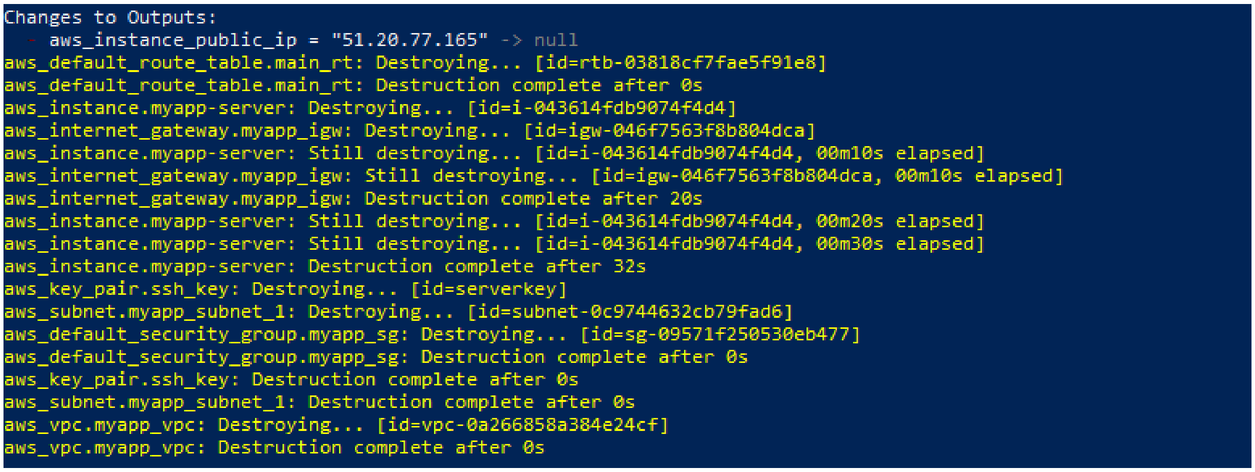
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

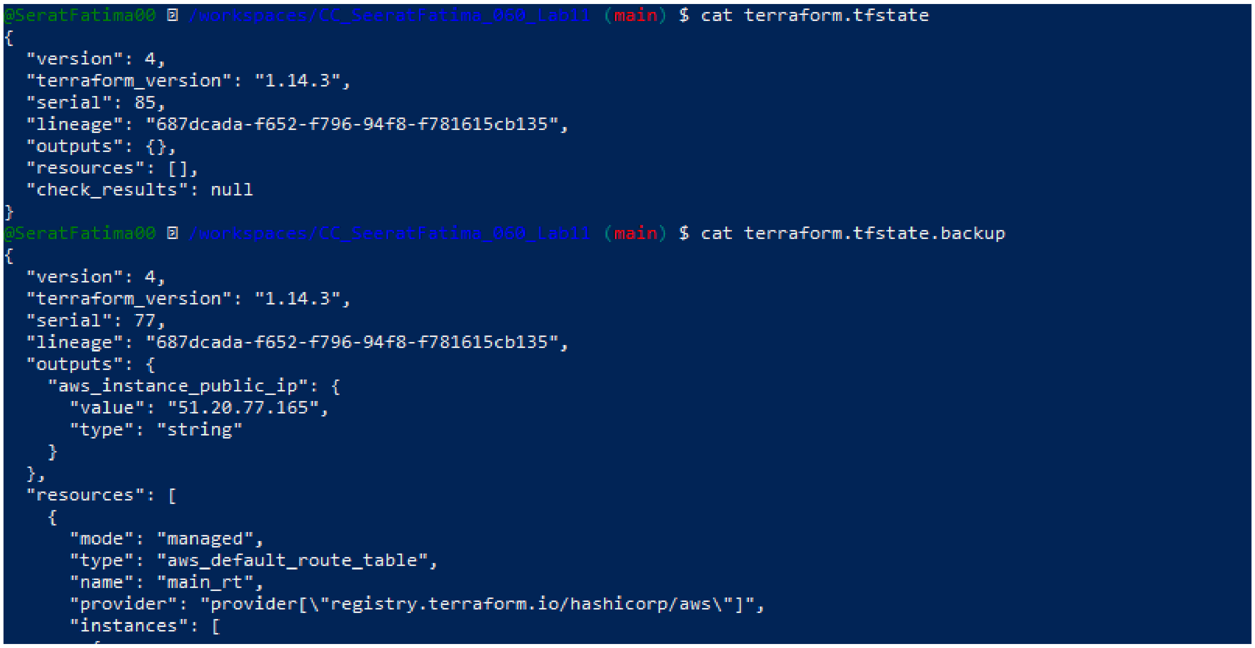
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

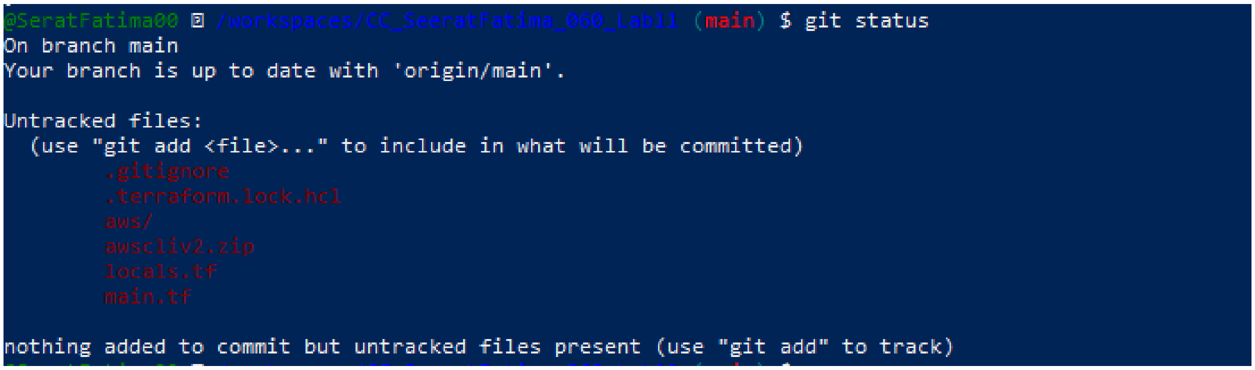
Cleanup



```
Changes to Outputs:
  - aws_instance_public_ip = "51.20.77.165" -> null
aws_default_route_table.main_rt: Destroying... [id=rtb-03818cf7fae5f91e8]
aws_default_route_table.main_rt: Destruction complete after 0s
aws_instance.myapp-server: Destroying... [id=i-043614fdb9074f4d4]
aws_internet_gateway.myapp_igw: Destroying... [id=igw-046f7563f8b804dca]
aws_instance.myapp-server: Still destroying... [id=i-043614fdb9074f4d4, 00m10s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-046f7563f8b804dca, 00m10s elapsed]
aws_internet_gateway.myapp_igw: Destruction complete after 20s
aws_instance.myapp-server: Still destroying... [id=i-043614fdb9074f4d4, 00m20s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-043614fdb9074f4d4, 00m30s elapsed]
aws_instance.myapp-server: Destruction complete after 32s
aws_key_pair.ssh_key: Destroying... [id=serverkey]
aws_subnet.myapp_subnet_1: Destroying... [id=subnet-0c9744632cb79fad6]
aws_default_security_group.myapp_sg: Destroying... [id=sg-09571f250530eb477]
aws_default_security_group.myapp_sg: Destruction complete after 0s
aws_key_pair.ssh_key: Destruction complete after 0s
aws_subnet.myapp_subnet_1: Destruction complete after 0s
aws_vpc.myapp_vpc: Destroying... [id=vpc-0a266858a384e24cf]
aws_vpc.myapp_vpc: Destruction complete after 0s
```



```
@SeratFatima00 ⍰ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 85,
  "lineage": "687dcada-f652-f796-94f8-f781615cb135",
  "outputs": {},
  "resources": [],
  "check_results": null
}
@SeratFatima00 ⍰ /workspaces/CC_SeeratFatima_060_Lab11 (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 77,
  "lineage": "687dcada-f652-f796-94f8-f781615cb135",
  "outputs": {
    "aws_instance_public_ip": {
      "value": "51.20.77.165",
      "type": "string"
    }
  },
  "resources": [
    {
      "mode": "managed",
      "type": "aws_default_route_table",
      "name": "main_rt",
      "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",
      "instances": [
```



```
@SeratFatima00 ⍰ /workspaces/CC_SeeratFatima_060_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
        .terraform.lock.hcl
        aws/
        awscliv2.zip
        locals.tf
        main.tf

nothing added to commit but untracked files present (use "git add" to track)
```