LAB 11

NAME: SHUMAIL ZAHRA

REGISTRATION #: 2023-BSE-061

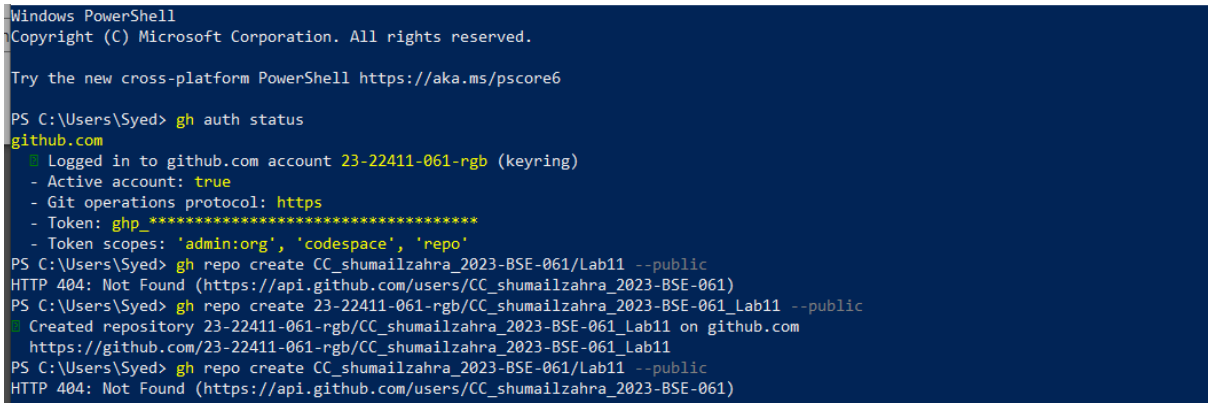
DEPARTMENT: BSE(5B)

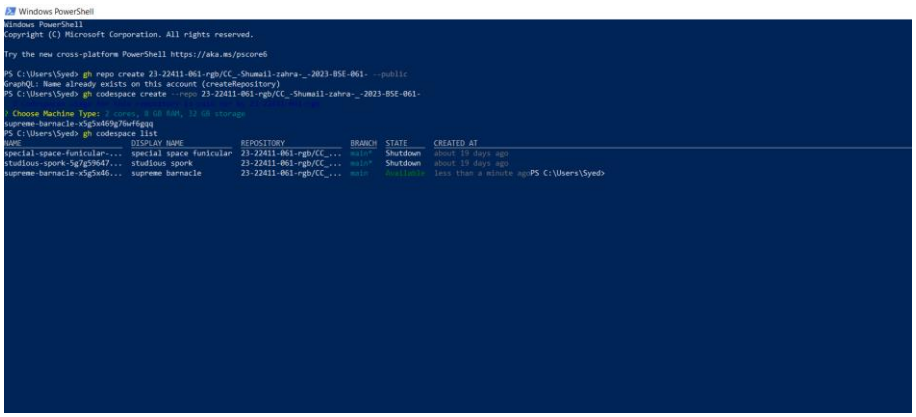
LAB TASK

Lab 11 – GH CLI Codespaces + AWS + Terraform: Variables, Collections, Sensitivity & EC2 Provisioning

Task 0 Lab Setup (Codespace & GH CLI)

taskA_codespace_create_and_list





taskA_codespace_ssh_connected

```
PS C:\Users\Syed> gh codespace ssh -c supreme-barnacle-x5g5x469g76wf6gqq
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.

@23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
```

Task 1 — Provider & Basic variable (variable precedence)

task1_touch_main_tf

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

@23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ touch main.tf
@23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
```

task1_main_tf_provider

task1_terraform_init

```
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ terraform init
-bash: terraform: command not found
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ sudo apt-get update
Get:1 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble InRelease [3600 B]
Get:2 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble/main all Packages [643 B]
Get:3 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble/main amd64 Packages [77.3 kB]
Get:4 https://dl.yarnpkg.com/debian stable InRelease
Get:5 https://repo.anaconda.com/pkgs/misc/debrepo/conda stable InRelease [3961 B]
Get:6 https://dl.yarnpkg.com/debian stable/main amd64 Packages [11.8 kB]
Get:7 https://dl.yarnpkg.com/debian stable/main all Packages [11.8 kB]
Get:8 https://repo.anaconda.com/pkgs/misc/debrepo/conda stable/main amd64 Packages [4557 B]
Get:9 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [33.1 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2898 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1183 kB]
Get:16 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:17 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1752 kB]
Get:18 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:19 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:20 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:21 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [35.9 kB]
Get:22 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [3059 kB]
Get:23 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1950 kB]
Get:24 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [2130 kB]
Get:25 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [49.5 kB]
Get:26 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [34.6 kB]
Fetched 35.5 MB in 6s (6233 kB/s)
Reading package lists... Done
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ sudo apt-get install -y gnupg software-properties-common curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnupg is already the newest version (2.4.4-2ubuntu17.3).
gnupg set to manually installed.
software-properties-common is already the newest version (0.99.49.3).
curl is already the newest version (8.5.0-2ubuntu10.6).
0 upgraded, 0 newly installed, 0 to remove and 51 not upgraded.
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.li
st.d/hashicorp.list
deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com noble main
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ sudo apt-get update && sudo apt-get install terraform -y
Hit:1 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble InRelease
Get:2 https://apt.releases.hashicorp.com noble InRelease [12.9 kB]
Hit:3 https://dl.yarnpkg.com/debian stable InRelease
Hit:4 https://repo.anaconda.com/pkgs/misc/debrepo/conda stable InRelease
Get:5 https://apt.releases.hashicorp.com noble/main amd64 Packages [266 kB]
Hit:6 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:7 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:8 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:9 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Fetched 279 kB in 1s (337 kB/s)
```


```
deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com noble main
```


```
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ sudo apt-get update && sudo apt-get install terraform -y
Hit:1 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble InRelease
Get:2 https://apt.releases.hashicorp.com noble InRelease [12.9 kB]
Hit:3 https://dl.yarnpkg.com/debian stable InRelease
Hit:4 https://repo.anaconda.com/pkgs/misc/debrepo/conda stable InRelease
Get:5 https://apt.releases.hashicorp.com noble/main amd64 Packages [266 kB]
Hit:6 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:7 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:8 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:9 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Fetched 279 kB in 1s (337 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  terraform
0 upgraded, 1 newly installed, 0 to remove and 51 not upgraded.
Need to get 30.6 MB of archives.
After this operation, 101 MB of additional disk space will be used.
Get:1 https://apt.releases.hashicorp.com noble/main amd64 terraform amd64 1.14.3-1 [30.6 MB]
Fetched 30.6 MB in 0s (73.6 MB/s)
Selecting previously unselected package terraform.
(Reading database ... 58629 files and directories currently installed.)
Preparing to unpack .../terraform_1.14.3-1_amd64.deb ...
Unpacking terraform (1.14.3-1) ...
Setting up terraform (1.14.3-1) ...
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ terraform -v
Terraform v1.14.3
on linux_amd64
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
```

```
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (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.
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
```

task1_variable_and_output_added

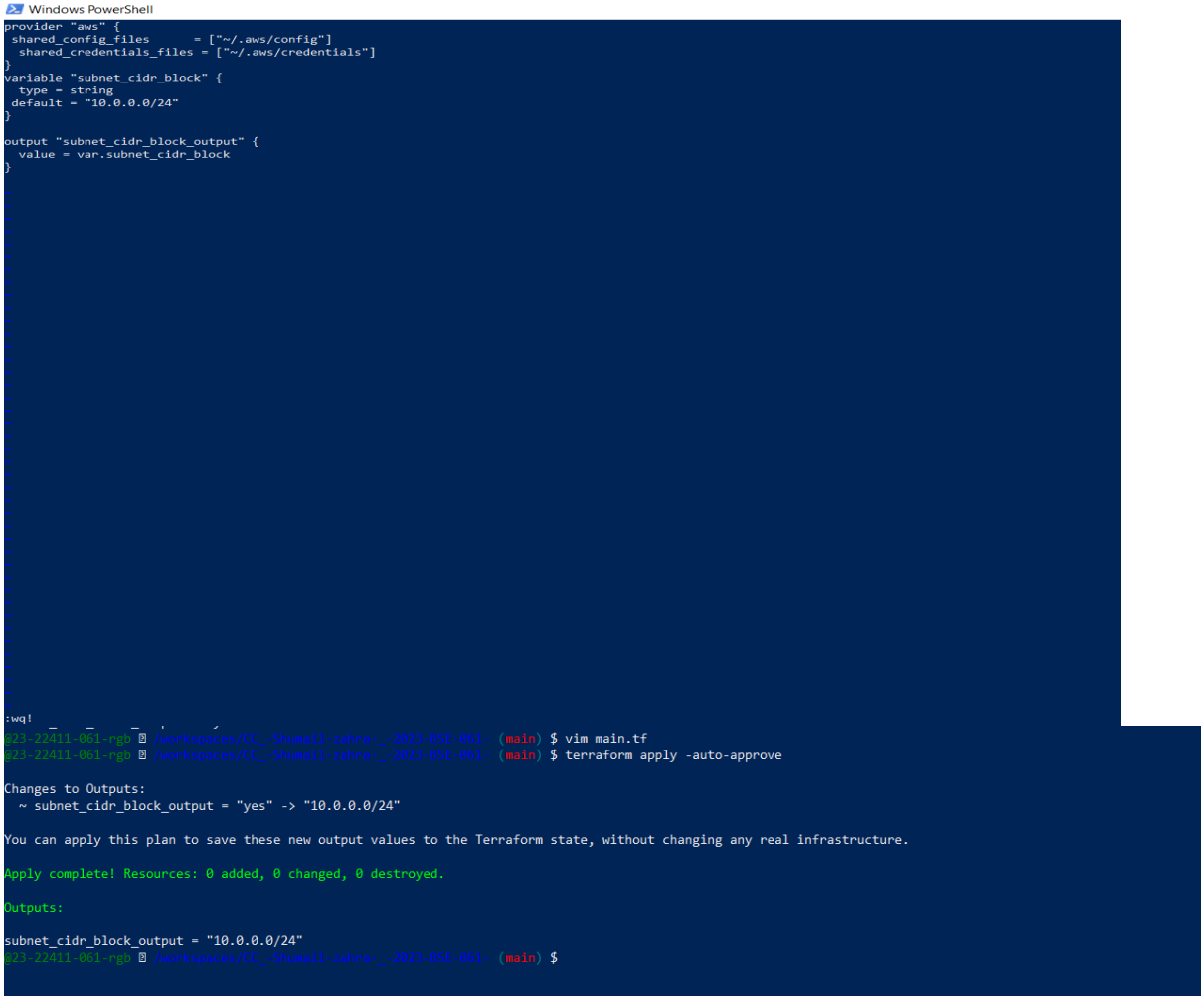
Windows PowerShell

```
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
}
```

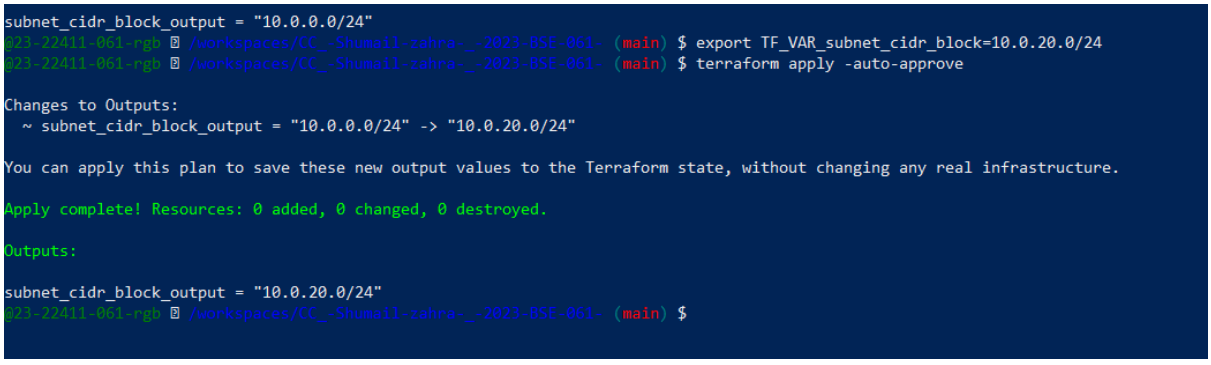
task1_apply_prompt_for_var

```
@23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ terraform apply -auto-approve
var.subnet_cidr_block
  Enter a value: yes


Changes to Outputs:
  + subnet_cidr_block_output = "yes"

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 = "yes"
@23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
```

task1_apply_with_default



task1_env_var_set_and_apply



task1_terraform_tfvars_and_apply

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

```
@23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ touch terraform.tfvars
@23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ subnet_cidr_block = "10.0.30.0/24"
-bash: subnet_cidr_block: command not found
@23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ vim terraform.tfvars
@23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (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"
@23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
```

task1_var_override_with_dash_var

```
@23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (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"
```

task1_printenv_tf_var_and_unset

```
subnet_cidr_block_output = "10.0.40.0/24"
@23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ printenv | grep TF_VAR_
TF_VAR_subnet_cidr_block=10.0.20.0/24
@23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ unset TF_VAR_subnet_cidr_block
@23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ printenv | grep TF_VAR_
@23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
```

Task 2 — Variable validation & sensitive / ephemeral variables

task2_subnet_variable_with_validation

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

task2_subnet_validation_error

```
@23-22411-061-rgb ➜ /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ vim main.tf
@23-22411-061-rgb ➜ /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (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.

@23-22411-061-rgb ➜ /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
```

task2_api_token_variable_added

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

task2_api_token_apply_sensitive

```
$ 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" -> null

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

Warning: Value for undeclared variable

The root module does not declare a variable named "subnet_cidr_block" 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, use the -compact-warnings option.

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

Outputs:

api_session_token_output = <sensitive>
```

task2_check_terraform_state_api_token

```
$ cat terraform.tfstate | grep -A 5 "api_session_token_output"
    "api_session_token_output": {
      "value": "my_API_session_Token",
      "type": "string",
      "sensitive": true
    }
  },
```

task2_api_token_ephemeral_error

Windows PowerShell

```
provider "aws" {
 shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
variable "api_session_token" {
  type        = string
  default     = ""
  description = "Short-lived API session token used during apply operations"
  sensitive   = true
  nullable    = false
  ephemeral   = true

  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
}
```

```
$ terraform apply -auto-approve -var "api_session_token=my_API_session_Token"

Warning: Value for undeclared variable

The root module does not declare a variable named "subnet_cidr_block" 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, use the -compact-warnings option.

Error: Ephemeral value not allowed

  on main.tf line 20, in output "api_session_token_output":
  20:   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.
```

task2_api_token_default_apply

Windows PowerShell

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

```
$ vim main.tf
$ 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.

  Warning: Value for undeclared variable

  The root module does not declare a variable named "subnet_cidr_block" 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, use the -compact-warnings option.

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

Outputs:

api_session_token_output = <sensitive>
(main) $
```

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

task3_variables_added

Windows PowerShell

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

task3_terraform_tfvars_populated

Windows PowerShell

```
subnet_cidr_block = "10.0.30.0/24"
environment = "dev"
project_name = "lab_work"
primary_subnet_id = "subnet-05bc73f40fefb9fe1"
subnet_count = 3
monitoring = true
```

task3_locals_tf_created

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
}
```

```
:wq!
```

task3_outputs_apply

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

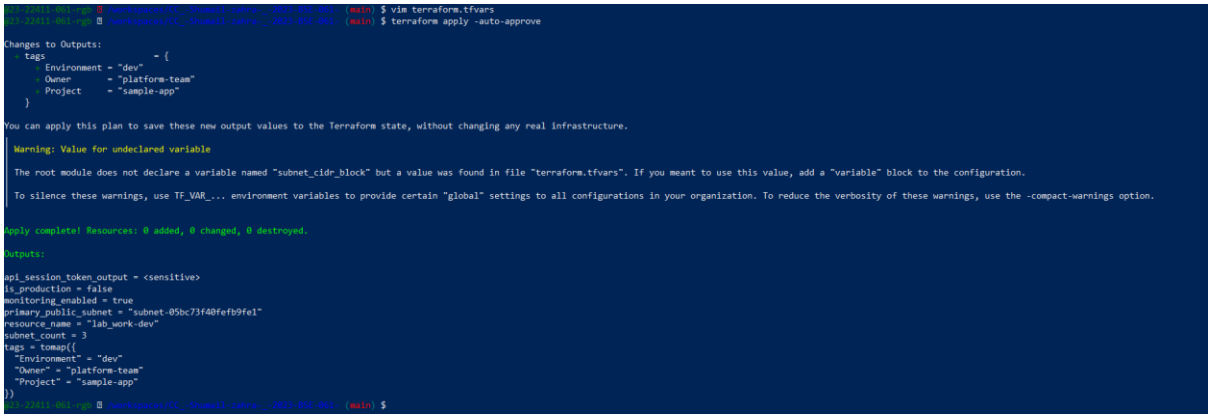
```
$ terraform apply -auto-approve

Changes to Outputs:
  + is_production           = false
  + monitoring_enabled      = true
  + primary_public_subnet   = "subnet-05bc73f40fefb9fe1"
  + 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.

╷
│ Warning: Value for undeclared variable
│
│ The root module does not declare a variable named "subnet_cidr_block" 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, use the -compact-warnings option.
╵

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-05bc73f40fefb9fe1"
resource_name = "lab_work-dev"
subnet_count = 3
```

Task 4 — Maps and Objects

task4_tags_variable_added

```
 shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
variable "api_session_token" {
  type        = string
  default     = "my_API_session_Token"
  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
}
```

task4_tags_output



task4_server_config_output

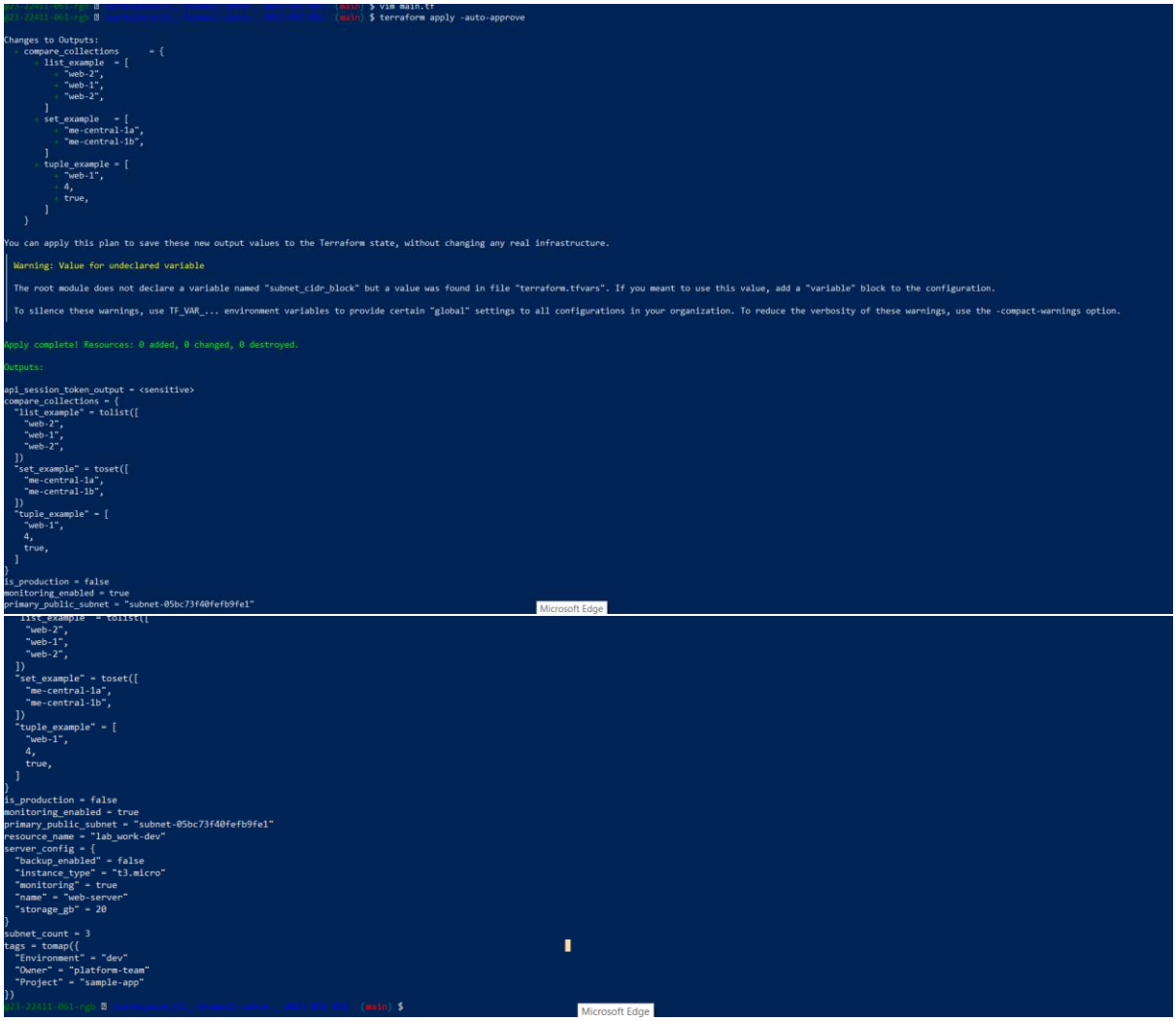
```
$ terraform apply -auto-approve

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.

Warning: Value for undeclared variable

The root module does not declare a variable named "subnet_cidr_block" 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, use the -compact-warnings option.

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-05bc73f40fefb9fe1"
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 5 — Collections: list, tuple, set & mutation via locals

task5_collections_defined

```
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
  }
}
```

task5_compare_collections

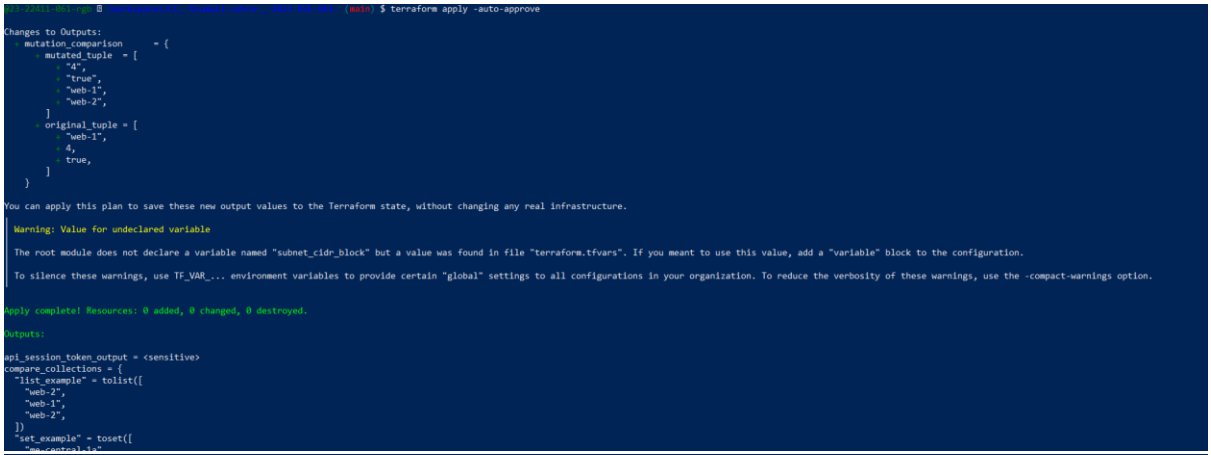


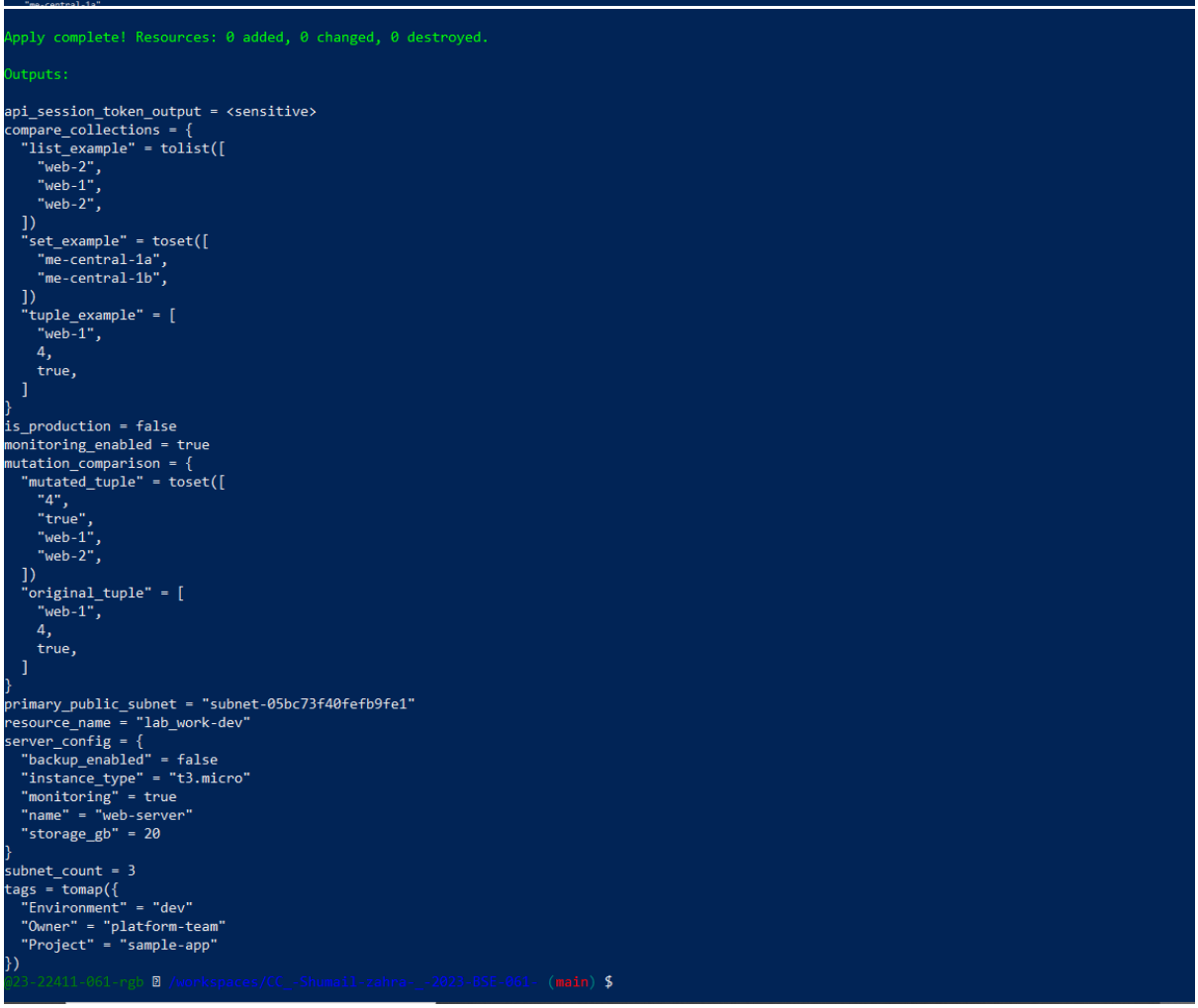
task5_locals_mutations

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"])
}
```

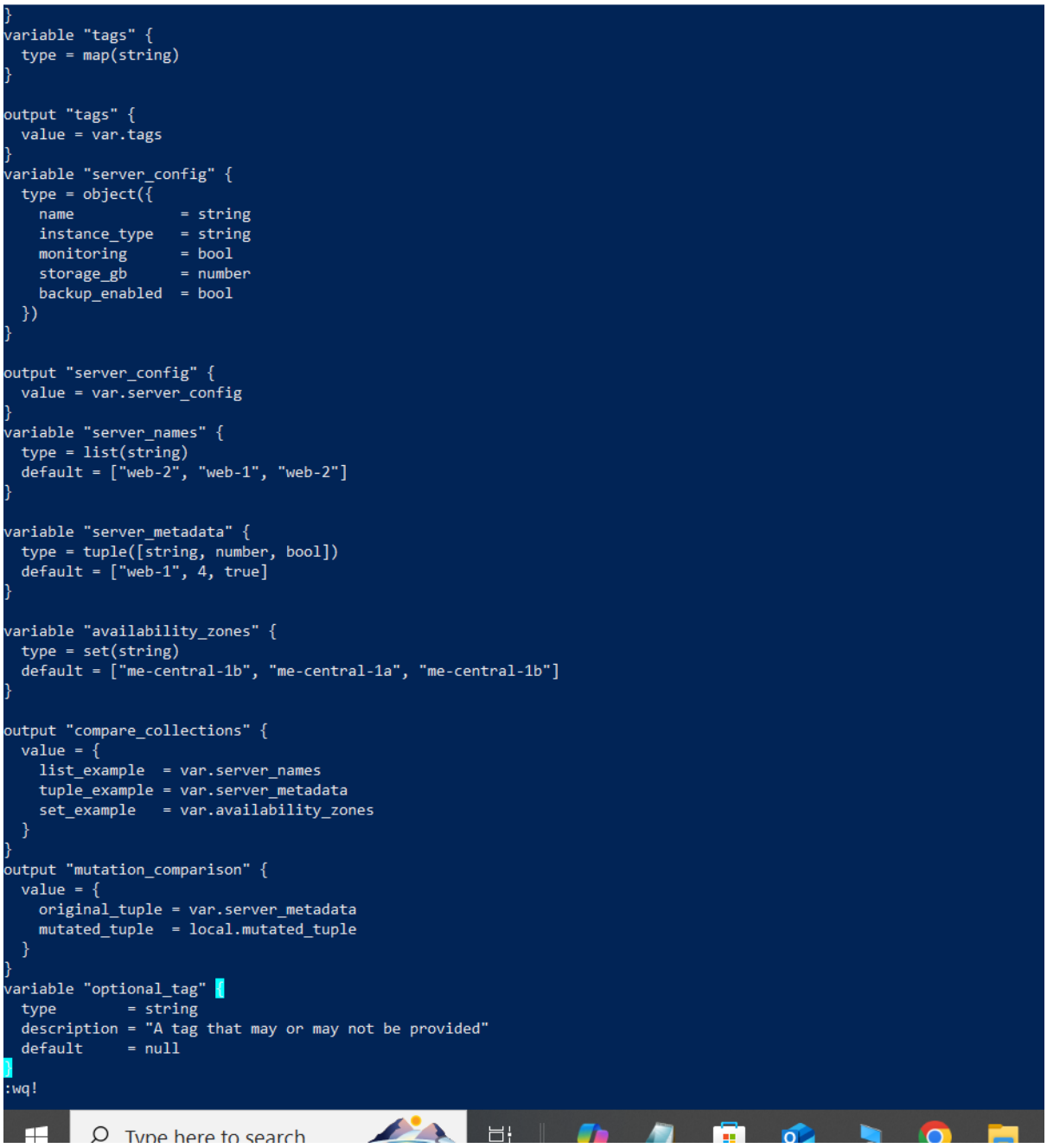
task5_mutation_comparison





Task 6 — Null, any type & dynamic values

task6_optional_tag_variable



task6_locals_merge

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

task6_optional_tag_no_value

```
(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.

Warning: Value for undeclared variable

The root module does not declare a variable named "subnet_cidr_block" 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, use the -compact-warnings option.

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",
```

```
    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-05bc73f40fefb9fe1"
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"
})
@23-22411-061-rgb /workspaces/CC_-Shumail-zabra-_-2023-BSE-061- (main) $
```

task6_optional_tag_with_value

```
23-22411-061-rgb /workspaces/CC_-Shumail-zabra-_-2023-BSE-061- (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.

╷
│ Warning: Value for undeclared variable
│
│ The root module does not declare a variable named "subnet_cidr_block" 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, use the -compact-warnings option.
╵

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,
```

```
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-05bc73f40fefb9fe1"
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"
})
@23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
@23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
```

task6_dynamic_value_string

```
    mutated_tuple  = local.mutated_tuple
  }
}
variable "optional_tag" {
  type        = string
  description = "A tag that may or may not be provided"
  default     = null
}
variable "dynamic_value" {
  type        = any
  description = "A variable that can accept any data type"
  default     = null
}

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

```
  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"
dynamic_value = "hello"
```

```
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.

╷
│ Warning: Value for undeclared variable
│
│ The root module does not declare a variable named "subnet_cidr_block" 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, use the -compact-warnings option.
╵

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"
}
```

```
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-05bc73f40fefb9fe1"
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"
g23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
```

task6_dynamic_value_number

```
subnet_cidr_block = "10.0.30.0/24"
environment = "dev"
project_name = "lab_work"
primary_subnet_id = "subnet-05bc73f40fefb9fe1"
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"
dynamic_value = 42

~
~
~
~
```

```
$ vim terraform.tfvars
$ 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.

╷
│ Warning: Value for undeclared variable
│
│ The root module does not declare a variable named "subnet_cidr_block" 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, use the -compact-warnings option.
╵

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",
```

task6_dynamic_value_list

Windows PowerShell

```
subnet_cidr_block = "10.0.30.0/24"
environment = "dev"
project_name = "lab_work"
primary_subnet_id = "subnet-05bc73f40fefb9fe1"
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"
dynamic_value = ["a", "b", "c"]
```

```
g23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061-  (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.

 Warning: Value for undeclared variable

 The root module does not declare a variable named "subnet_cidr_block" 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, use the -compact-warnings option.


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([
```

```
}
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-05bc73f40fefb9fe1"
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 = [
  "a",
  "b",
  "c",
]
g23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061-  (main) $
```

task6_dynamic_value_map

```
subnet_cidr_block = "10.0.30.0/24"
environment = "dev"
project_name = "lab_work"
primary_subnet_id = "subnet-05bc73f40fefb9fe1"
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"
dynamic_value = {
  name = "server"
  cpu  = 4
}
```

```
(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.

 Warning: Value for undeclared variable

 The root module does not declare a variable named "subnet_cidr_block" 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, use the -compact-warnings option.

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" = [
```

```
  "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-05bc73f40fefb9fe1"
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 = {
  "cpu" = 4
  "name" = "server"
}
```

task6_dynamic_value_null

Windows PowerShell

```
subnet_cidr_block = "10.0.30.0/24"
environment = "dev"
project_name = "lab_work"
primary_subnet_id = "subnet-05bc73f40fefb9fe1"
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"
dynamic_value = null
```

```
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.

 Warning: Value for undeclared variable

 The root module does not declare a variable named "subnet_cidr_block" 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, use the -compact-warnings option.

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,
  ]
```

```
}
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-05bc73f40fefb9fe1"
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"
})
023-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
```

Task 7 — Git ignore

task7_gitignore_created

```
eeeeeiles
Dociiles
.terraform
 .tfstate
 .tfstate.
 .tfvars
 .pem
```

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

task8_clean_files


"locals.tf" 1L, 2B


"terraform.tfvars" 1L, 1B

task8_variables_recreated

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" {}
```

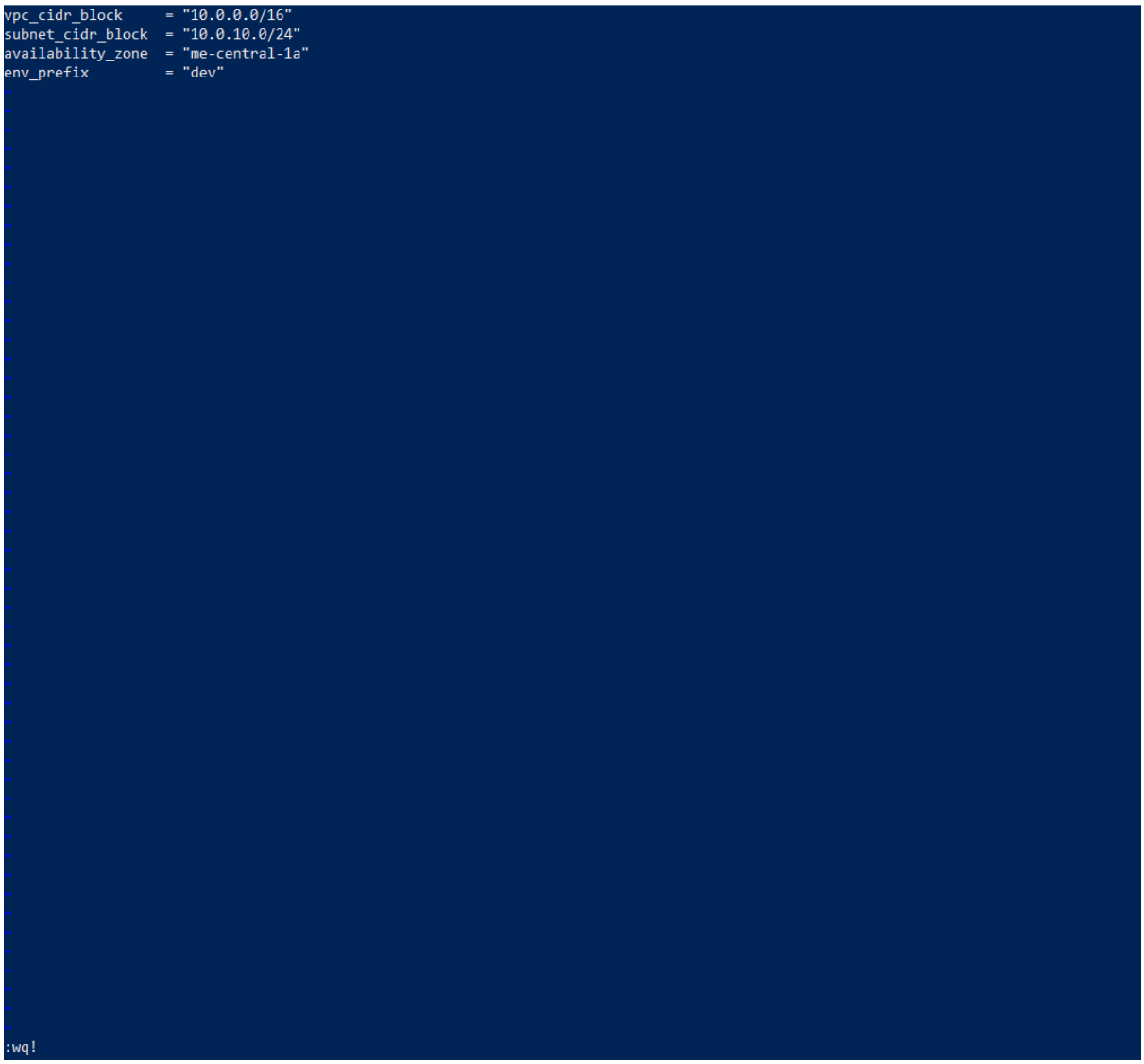
task8_vpc_resources_added

```
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"
  }
}
```

task8_subnet_resources_added

```
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"
  }
}
~
~
~
~
~
~
```

task8_terraform_tfvars_vpc_values



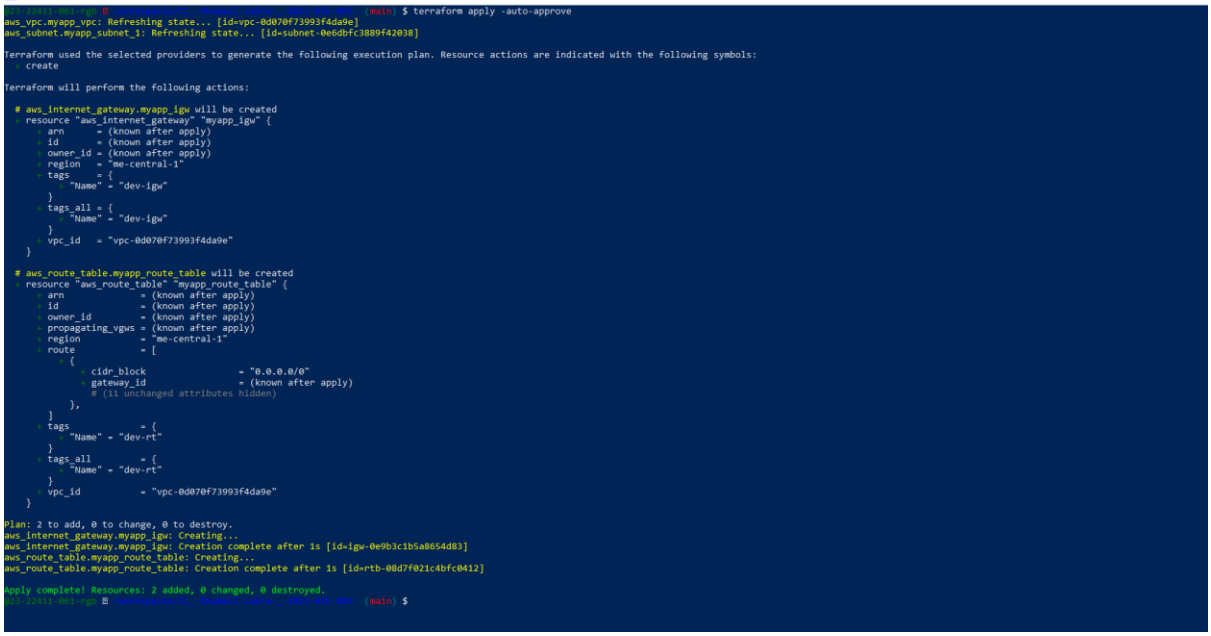
task8_vpc_subnet_apply

```
023-22411-061-rgb 🗄 /workspaces/CC_Shumail-zahra_2023-BSE-061 (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.
023-22411-061-rgb 🗄 /workspaces/CC_Shumail-zahra_2023-BSE-061 (main) $ terraform apply -auto-approve
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-0d070f73993f4da9e]

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

Terraform will perform the following actions:

  # aws_subnet.myapp_subnet_1 will be created
  + resource "aws_subnet" "myapp_subnet_1" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "me-central-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.10.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = false
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + region                                         = "me-central-1"
      + tags                                           = {
          + "Name" = "dev-subnet-1"
        }
      + tags_all                                       = {
          + "Name" = "dev-subnet-1"
        }
      + vpc_id                                         = "vpc-0d070f73993f4da9e"
    }

Plan: 1 to add, 0 to change, 0 to destroy.
aws_subnet.myapp_subnet_1: Creating...
aws_subnet.myapp_subnet_1: Creation complete after 1s [id=subnet-0e6dbfc3889f42038]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
023-22411-061-rgb 🗄 /workspaces/CC_Shumail-zahra_2023-BSE-061 (main) $
```

task8_igw_route_table_before_apply

```
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"
  }
}
```

task8_igw_route_table_after_apply



```
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-0d070f73993f4da9e]
aws_subnet.myapp_subnet_1: Refreshing state... [id=subnet-0e6dbfc3889f42038]

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   = "me-central-1"
      + tags     = {
          + "Name" = "dev-igw"
        }
      + tags_all = {
          + "Name" = "dev-igw"
        }
      + vpc_id   = "vpc-0d070f73993f4da9e"
    }

  # 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-0d070f73993f4da9e"
    }

Plan: 2 to add, 0 to change, 0 destroy.
aws_internet_gateway.myapp_igw: Creating...
aws_internet_gateway.myapp_igw: Creation complete after 1s [id=igw-0e9b3c1b5a8654d83]
aws_route_table.myapp_route_table: Creating...
aws_route_table.myapp_route_table: Creation complete after 1s [id=rtb-08d7f021c4bfc0412]

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

task8_association_apply

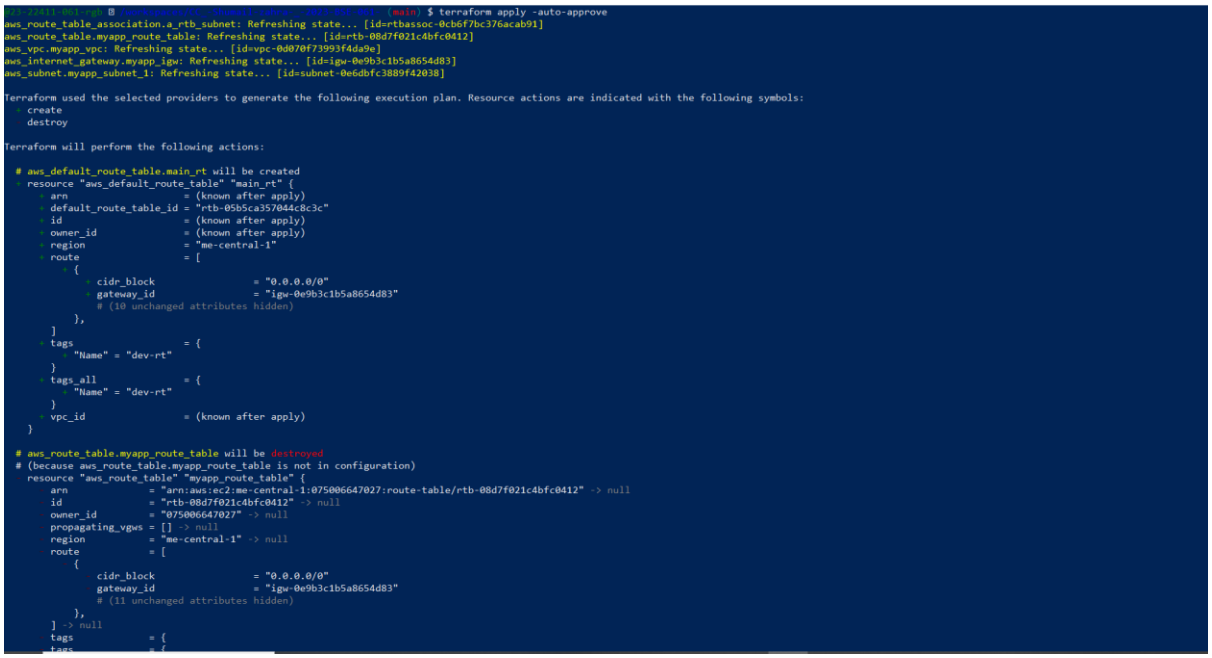
```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
variable "vpc_cidr_block" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
resource "aws_vpc" "myapp_vpc" {
  cidr_block = var.vpc_cidr_block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

```
023-22411-061-rgb /workspaces/CC_Shumail-zahra_2023-BSE-061 (main) $ vim -f main.tf
023-22411-061-rgb /workspaces/CC_Shumail-zahra_2023-BSE-061 (main) $ @23-22411-061-rgb /workspaces/CC_Shumail-zahra_2023-BSE-061 (main) $ terraform apply -auto-approve

aws_vpc.myapp_vpc: Refreshing state... [id=vpc-0d070f73993f4da9e]
aws_internet_gateway.myapp_igw: Refreshing state... [id=igw-0e9b3c1b5a8654d83]
aws_subnet.myapp_subnet_1: Refreshing state... [id=subnet-0e6dbfc3889f42038]
aws_route_table.myapp_route_table: Refreshing state... [id=rtb-08d7f021c4bfc0412]

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-08d7f021c4bfc0412"
      + subnet_id      = "subnet-0e6dbfc3889f42038"
    }

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 1s [id=rtbassoc-0cb6f7bc376acab91]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
023-22411-061-rgb /workspaces/CC_Shumail-zahra_2023-BSE-061 (main) $
023-22411-061-rgb /workspaces/CC_Shumail-zahra_2023-BSE-061 (main) $
```

task8_default_route_table

```
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"
  }
}
```

task8_default_route_table_apply



Windows PowerShell

```
  # 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-1:075006647027:route-table/rtb-08d7f021c4bfc0412" -> null
      - id               = "rtb-08d7f021c4bfc0412" -> null
      - owner_id         = "075006647027" -> null
      - propagating_vgws = [] -> null
      - region           = "me-central-1" -> null
      - route            = [
          - {
              - cidr_block                = "0.0.0.0/0"
              - gateway_id                = "igw-0e9b3c1b5a8654d83"
                # (11 unchanged attributes hidden)
            },
        ] -> null
      - tags             = {
          - "Name" = "dev-rt"
        } -> null
      - tags_all         = {
          - "Name" = "dev-rt"
        } -> null
      - vpc_id           = "vpc-0d070f73993f4da9e" -> 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-0cb6f7bc376acab91" -> null
      - region         = "me-central-1" -> null
      - route_table_id = "rtb-08d7f021c4bfc0412" -> null
      - subnet_id      = "subnet-0e6dbfc3889f42038" -> 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-0cb6f7bc376acab91]
aws_default_route_table.main_rt: Creating...
aws_route_table_association.a_rtb_subnet: Destruction complete after 0s
aws_route_table.myapp_route_table: Destroying... [id=rtb-08d7f021c4bfc0412]
aws_default_route_table.main_rt: Creation complete after 1s [id=rtb-05b5ca357044c8c3c]
aws_route_table.myapp_route_table: Destruction complete after 1s

Apply complete! Resources: 1 added, 0 changed, 2 destroyed.
@23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
```

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

task9_my_ip_variable_added

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



~
~
~
~
~
~
~
:wq!
```

task9_public_ip_curl

```
@23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ curl icanhazip.com
20.192.21.52
@23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ vim terraform.tfvars
```

```
vpc_cidr_block     = "10.0.0.0/16"
subnet_cidr_block  = "10.0.10.0/24"
my_ip = "20.192.21.52/32"
instance_type = "t3.micro"
availability_zone = "me-central-1a"   # or your chosen AZ
env_prefix = "dev"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

task9_security_group_apply

```
  }
}
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"]
    prefix_list_ids = []
  }

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

task9_keypair_created_and_saved

```
@23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ aws ec2 create-key-pair \
>   --key-name MyED25519Key \
>   --key-type ed25519 \
>   --key-format pem \
>   --query 'KeyMaterial' \
>   --output text > MyED25519Key.pem
ED25519Key.pem
@23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
@23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ chmod 600 MyED25519Key.pem
@23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ aws ec2 create-key-pair \
>   --key-name MyED25519Key \
>   --key-type ed25519 \
>   --key-format pem \
>   --query 'KeyMaterial' \
>   --output text > MyED25519Key.pem

An error occurred (InvalidKeyPair.Duplicate) when calling the CreateKeyPair operation: The keypair already exists
@23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ chmod 600 MyED25519Key.pem
@23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
```

```
>   --output text > MyED25519Key.pem

An error occurred (InvalidKeyPair.Duplicate) when calling the CreateKeyPair operation: The keypair already exists
@23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ chmod 600 MyED25519Key.pem
@23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ *.pem
-bash: MyED25519Key.pem: command not found
@23-22411-061-rgb /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ echo "*.pem" >> .gitignore
```

task9_instance_type_set

```
  }

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

  tags = {
    Name = "${var.env_prefix}-sg"
  }
}
resource "aws_instance" "myapp-server" {
  ami                         = "ami-05524d6658fcf35b6" # Amazon Linux 2023
  instance_type               = var.instance_type
  subnet_id                   = aws_subnet.myapp_subnet_1.id
  security_groups             = [aws_default_security_group.default_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 --
```

task9_ec2_apply_and_public_ip

```
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-0d070f73993f4da9e]
aws_internet_gateway.myapp_igw: Refreshing state... [id=igw-0e9b3c1b5a8654d83]
aws_subnet.myapp_subnet_1: Refreshing state... [id=subnet-0e6dbfc3889f42038]
aws_default_security_group.myapp_sg: Refreshing state... [id=sg-098d71ff41f0ec305]
aws_default_route_table.main_rt: Refreshing state... [id=rtb-05b5ca357044c8c3c]

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-05524d6658fcf35b6"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = true
      + availability_zone                    = "me-central-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                               = "me-central-1"
      + secondary_private_ips                = (known after apply)
      + security_groups                      = [
          + "sg-098d71ff41f0ec305",
        ]
```

```
        ]
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = "subnet-0e6dbfc3889f42038"
      + 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 12s [id=i-026d09ed753a975e0]

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

Outputs:

aws_instance_public_ip = "158.252.32.21"
```

```
@23-22411-061-rgb ➜ /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ vim main.tf
@23-22411-061-rgb ➜ /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ terraform apply -auto-approve
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-0d070f73993f4da9e]
aws_internet_gateway.myapp_igw: Refreshing state... [id=igw-0e9b3c1b5a8654d83]
aws_subnet.myapp_subnet_1: Refreshing state... [id=subnet-0e6dbfc3889f42038]
aws_default_security_group.myapp_sg: Refreshing state... [id=sg-098d71ff41f0ec305]
aws_default_route_table.main_rt: Refreshing state... [id=rtb-05b5ca357044c8c3c]

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-05524d6658fcf35b6"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = true
      + availability_zone                    = "me-central-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                             = "serverkey"
      + 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                               = "me-central-1"
      + secondary_private_ips                = (known after apply)
      + security_groups                      = [
          + "sg-098d71ff41f0ec305",
        ]
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = "subnet-0e6dbfc3889f42038"
```

```
      + 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)
    }

  # 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 AAAAC3NzaC1lZDI1NTE5AAAAID2FGFhgLoP8nFhitLFwErF+e9TuaMRTHrD94g/UGKYp codespace@codespaces-4f2106"
      + region          = "me-central-1"
      + tags_all        = (known after apply)
    }

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

Changes to Outputs:
  ~ aws_instance_public_ip = "158.252.32.21" -> (known after apply)
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 13s [id=i-08e2ff922169c882a]

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

Outputs:

aws_instance_public_ip = "158.252.72.87"
@23-22411-061-rgb ➜ /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
```

task9_ssh_into_ec2

```
aws_instance_public_ip = "158.252.32.21"
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ ssh -i MyED25519Key.pem ec2-user@158.252.32.21
The authenticity of host '158.252.32.21 (158.252.32.21)' can't be established.
ED25519 key fingerprint is SHA256:hiSCZnx4m5XgwCRL0NtPD2sxHzeYgmn1TcrnhhPtf/w.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '158.252.32.21' (ED25519) to the list of known hosts.
Load key "MyED25519Key.pem": error in libcrypto
ec2-user@158.252.32.21: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ ssh-keygen -p -m PEM -f MyED25519Key.pem
Failed to load key MyED25519Key.pem: error in libcrypto
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ ssh -i MyED25519Key.pem ec2-user@20.192.21.52


ssh: connect to host 20.192.21.52 port 22: Connection timed out
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ ssh -i MyED25519Key.pem ec2-user@158.252.32.21
```

task9_ssh_keypair_and_ssh

```
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (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:
SHA256:3Zbbuj1mCzTmtQViYFoDce7r5q5vd1G2SG7STpDDaaQ codespace@codespaces-4f2106
The key's randomart image is:
+--[ED25519 256]--+
|        oo*      |
|         * +     |
|        . = = .  |
|         E X + .o|
|        S + & ooo|
|           B @.+ |
|          . O o. |
|         .o .==  |
|        .B=.o=oo |
+----[SHA256]-----+
@23-22411-061-rgb 🗄 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
```

```
    prefix_list_ids = []
  }

  tags = {
    Name = "${var.env_prefix}-sg"
  }
}
resource "aws_instance" "myapp-server" {
  ami                         = "ami-05524d6658fcf35b6" # Amazon Linux 2023
  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")
}

:wq!
```

```
aws_instance_public_ip = "158.252.72.87"
@23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ ssh ec2-user@158.252.72.87
The authenticity of host '158.252.72.87 (158.252.72.87)' can't be established.
ED25519 key fingerprint is SHA256:QP5P4M4n08qfAvbTNjXd1O1tN/bzyvVPtZbusWkxTm0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '158.252.72.87' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-10-107 ~]$
```

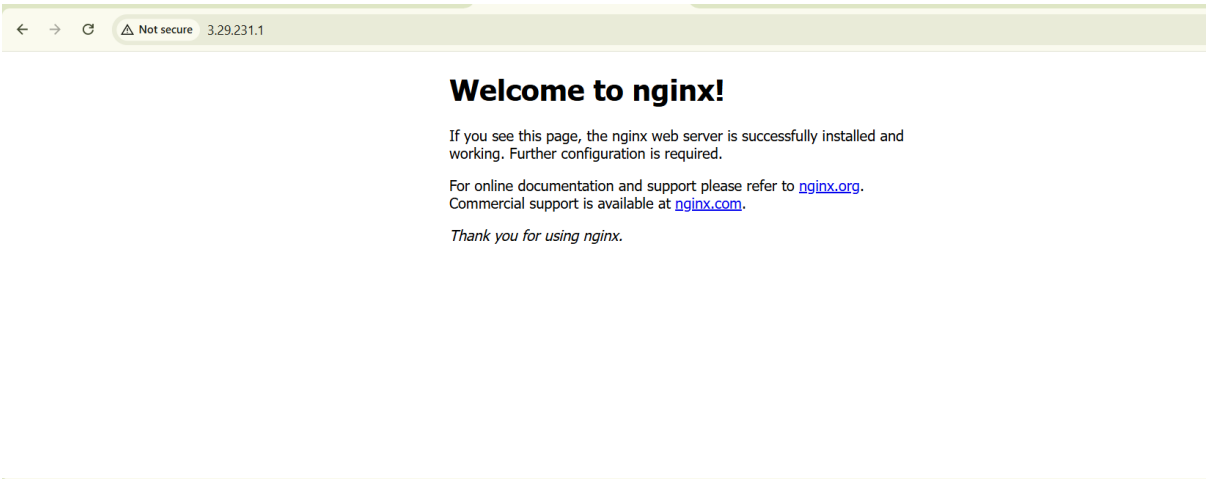
task9_nginx_local_curl

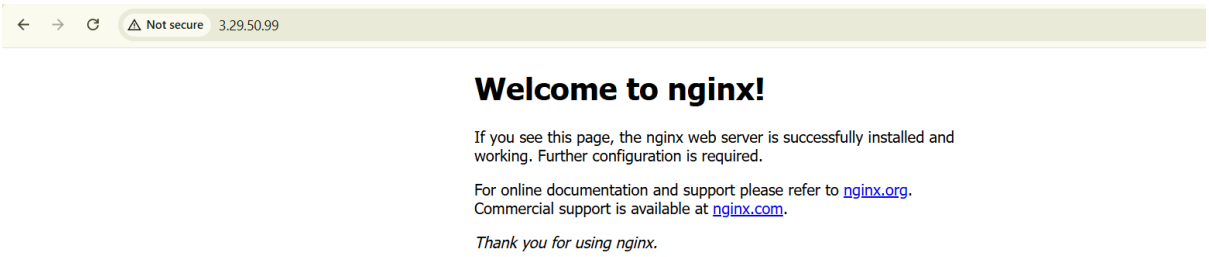
```
aws_instance_public_ip = "3.29.231.1"
@23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ ssh ec2-user@3.29.231.1
The authenticity of host '3.29.231.1 (3.29.231.1)' can't be established.
ED25519 key fingerprint is SHA256:HO7wcCzB74bhFyxq8osixZJEr1GwCRob0jvUPEUHCgU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.29.231.1' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-10-73 ~]$ 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-10-0-10-73 ~]$
```

```
@23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ cat > entry-script.sh <<'EOF'
> #!/bin/bash
> yum update -y
> yum install -y nginx
> systemctl start nginx
> systemctl enable nginx
> EOF
```

```
      - primary_network_interface {
          - delete_on_termination = true -> null
          - network_interface_id  = "eni-023706f0284ffd7c1" -> null
        }

      ~ private_dns_name_options (known after apply)
      - private_dns_name_options {
          - enable_resource_name_dns_a_record    = false -> null
          - enable_resource_name_dns_aaaa_record = false -> null
          - hostname_type                        = "ip-name" -> null
        }

      ~ root_block_device (known after apply)
      - root_block_device {
          - 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-01f0e1ba6fb4ebfd3" -> null
          - volume_size           = 8 -> null
          - volume_type           = "gp3" -> null
          # (1 unchanged attribute hidden)
        }
    }

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

Changes to Outputs:
  ~ aws_instance_public_ip = "3.29.231.1" -> (known after apply)
aws_instance.myapp-server: Destroying... [id=i-08048fab64dd20ed1]
aws_instance.myapp-server: Still destroying... [id=i-08048fab64dd20ed1, 00m10s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-08048fab64dd20ed1, 00m20s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-08048fab64dd20ed1, 00m30s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-08048fab64dd20ed1, 00m40s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-08048fab64dd20ed1, 00m50s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-08048fab64dd20ed1, 01m00s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-08048fab64dd20ed1, 01m10s elapsed]
aws_instance.myapp-server: Destruction complete after 1m10s
aws_instance.myapp-server: Creating...
aws_instance.myapp-server: Still creating... [00m10s elapsed]
aws_instance.myapp-server: Creation complete after 13s [id=i-0989920d6310e3f6e]

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

Outputs:

aws_instance_public_ip = "3.29.50.99"
```

task9_nginx_browser_page

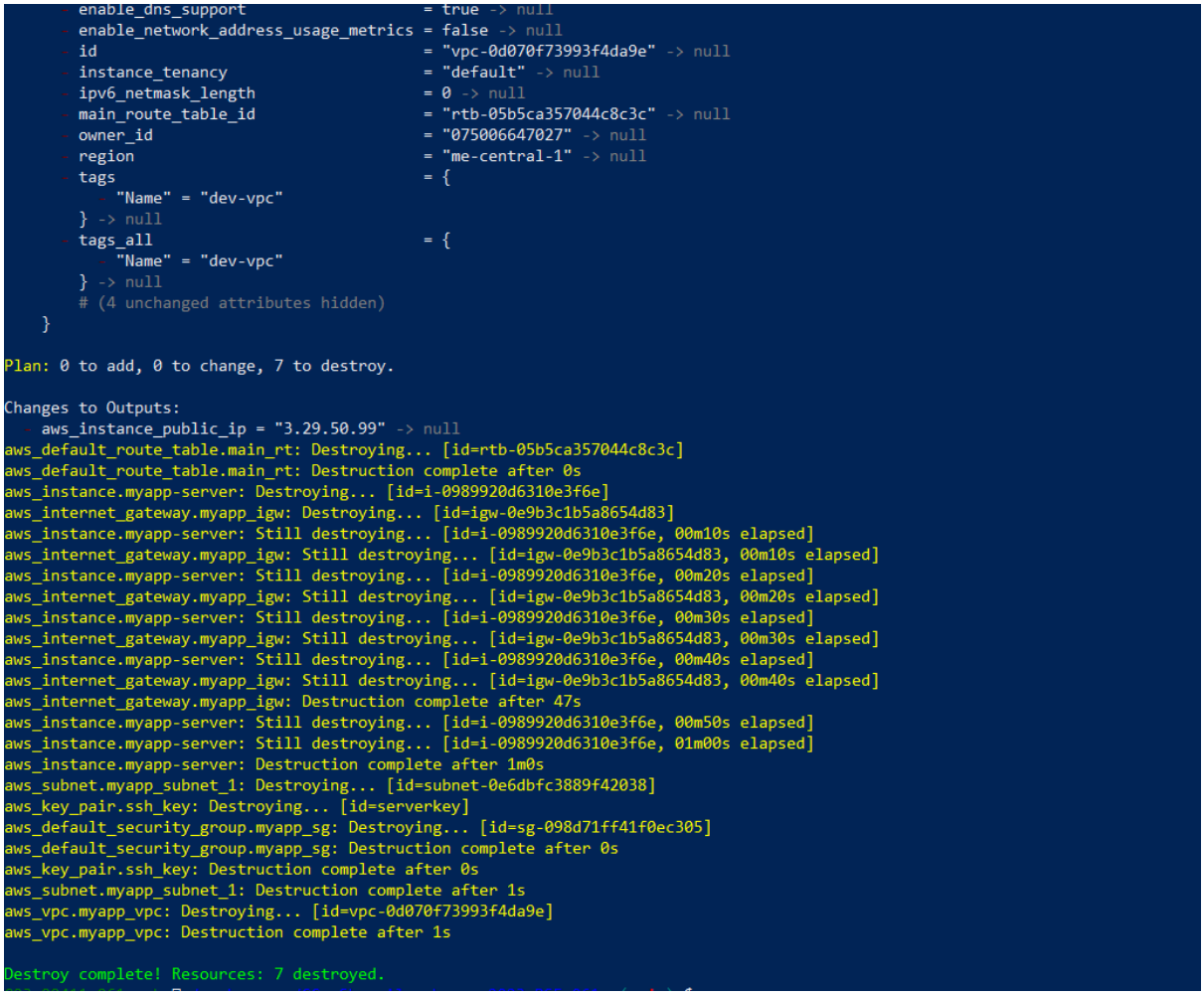




Cleanup

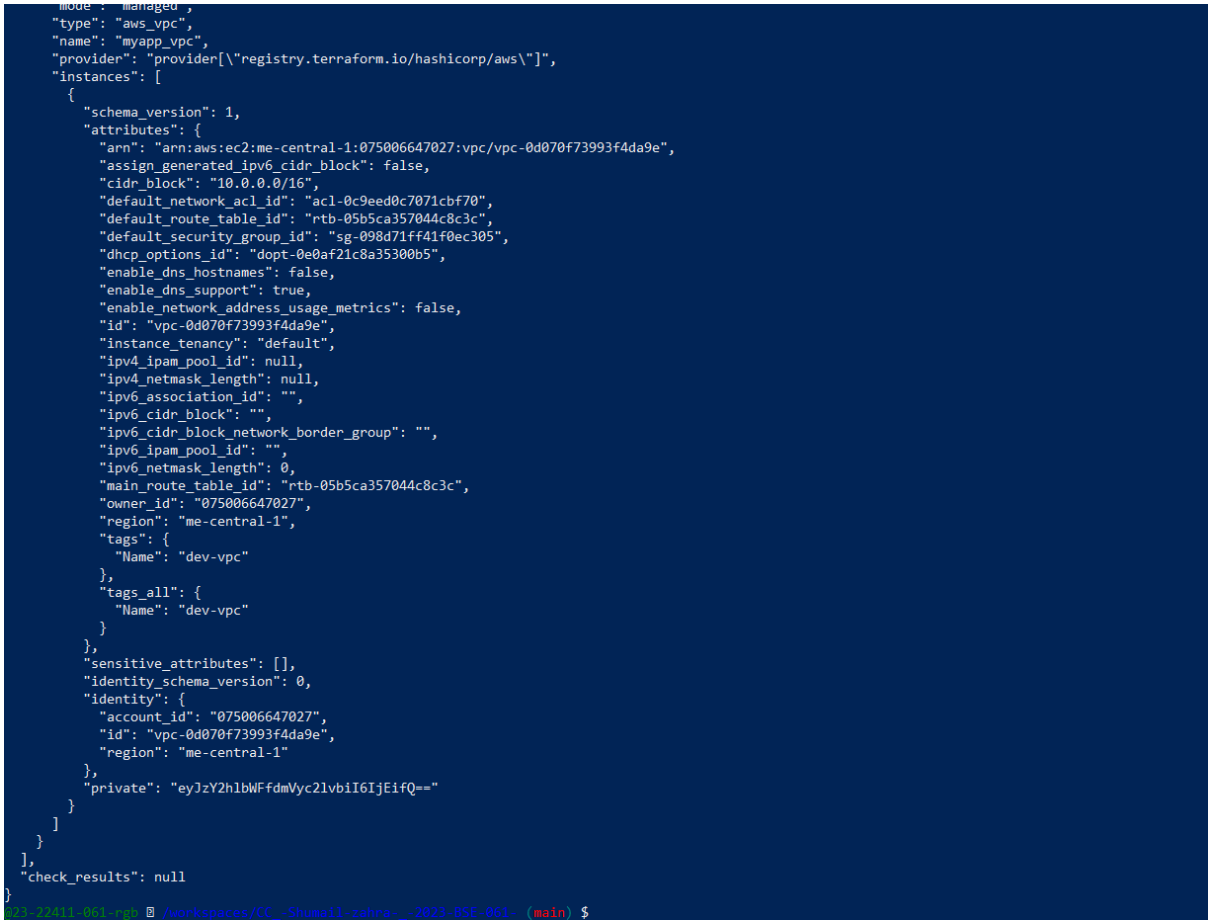
cleanup_destroy

```
23-22411-061-rgb 🗄 /workspaces/CC--Shumail-zahra--2023-BSE-061- (main) $ terraform destroy -auto-approve
aws_key_pair.ssh_key: Refreshing state... [id=serverkey]
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-0d070f73993f4da9e]
aws_subnet.myapp_subnet_1: Refreshing state... [id=subnet-0e6dbfc3889f42038]
aws_internet_gateway.myapp_igw: Refreshing state... [id=igw-0e9b3c1b5a8654d83]
aws_default_security_group.myapp_sg: Refreshing state... [id=sg-098d71ff41f0ec305]
aws_default_route_table.main_rt: Refreshing state... [id=rtb-05b5ca357044c8c3c]
aws_instance.myapp-server: Refreshing state... [id=i-0989920d6310e3f6e]

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

Terraform will perform the following actions:

  # aws_default_route_table.main_rt will be destroyed
  - resource "aws_default_route_table" "main_rt" {
      - arn                    = "arn:aws:ec2:me-central-1:075006647027:route-table/rtb-05b5ca357044c8c3c" -> null
      - default_route_table_id = "rtb-05b5ca357044c8c3c" -> null
      - id                     = "rtb-05b5ca357044c8c3c" -> null
      - owner_id               = "075006647027" -> null
      - propagating_vgws       = [] -> null
      - region                 = "me-central-1" -> null
      - route                  = [
          - {
              - cidr_block                = "0.0.0.0/0"
              - gateway_id                = "igw-0e9b3c1b5a8654d83"
              # (10 unchanged attributes hidden)
            },
        ] -> null
      - tags                   = {
          - "Name" = "dev-rt"
        } -> null
      - tags_all               = {
          - "Name" = "dev-rt"
        } -> null
      - vpc_id                 = "vpc-0d070f73993f4da9e" -> null
    }

  # aws_default_security_group.myapp_sg will be destroyed
  - resource "aws_default_security_group" "myapp_sg" {
      - arn                    = "arn:aws:ec2:me-central-1:075006647027:security-group/sg-098d71ff41f0ec305" -> null
      - description            = "default VPC security group" -> null
      - egress                 = [
          - {
              - cidr_blocks      = [
                  - "0.0.0.0/0",
                ]
              - from_port        = 0
              - ipv6_cidr_blocks = []
              - prefix_list_ids  = []
              - protocol         = "-1"
              - security_groups  = []
              - self             = false
```



cleanup_state_files

```
023-22411-061-rgb 🗎 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 63,
  "lineage": "4b0173a3-0061-e02d-88d2-58846293b1ac",
  "outputs": {},
  "resources": [],
  "check_results": null
}
023-22411-061-rgb 🗎 /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 55,
  "lineage": "4b0173a3-0061-e02d-88d2-58846293b1ac",
  "outputs": {
    "aws_instance_public_ip": {
      "value": "3.29.50.99",
      "type": "string"
    }
  },
  "resources": [
    {
      "mode": "managed",
      "type": "aws_default_route_table",
      "name": "main_rt",
      "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",
      "instances": [
        {
          "schema_version": 0,
          "attributes": {
            "arn": "arn:aws:ec2:me-central-1:075006647027:route-table/rtb-05b5ca357044c8c3c",
            "default_route_table_id": "rtb-05b5ca357044c8c3c",
            "id": "rtb-05b5ca357044c8c3c",
            "owner_id": "075006647027",
            "propagating_vgws": [],
            "region": "me-central-1",
            "route": [
              {
                "cidr_block": "0.0.0.0/0",
                "core_network_arn": "",
                "destination_prefix_list_id": "",
                "egress_only_gateway_id": "",
                "gateway_id": "igw-0e9b3c1b5a8654d83",
                "instance_id": "",
                "ipv6_cidr_block": "",
                "nat_gateway_id": "",
                "network_interface_id": "",
                "transit_gateway_id": "",
                "vpc_endpoint_id": "",
                "vpc_peering_connection_id": ""
              }
            ],
```



cleanup_verify_no_secrets

```
@23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .gitignore

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

no changes added to commit (use "git add" and/or "git commit -a")
@23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ .gitignore
-bash: /usr/local/nvs/.gitignore: Permission denied
@23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ vim .gitignore
@23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .gitignore

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

no changes added to commit (use "git add" and/or "git commit -a")
@23-22411-061-rgb  /workspaces/CC_-Shumail-zahra-_-2023-BSE-061- (main) $
```
