Name: Zuha Irfan

Roll No: 2023-BSE-073

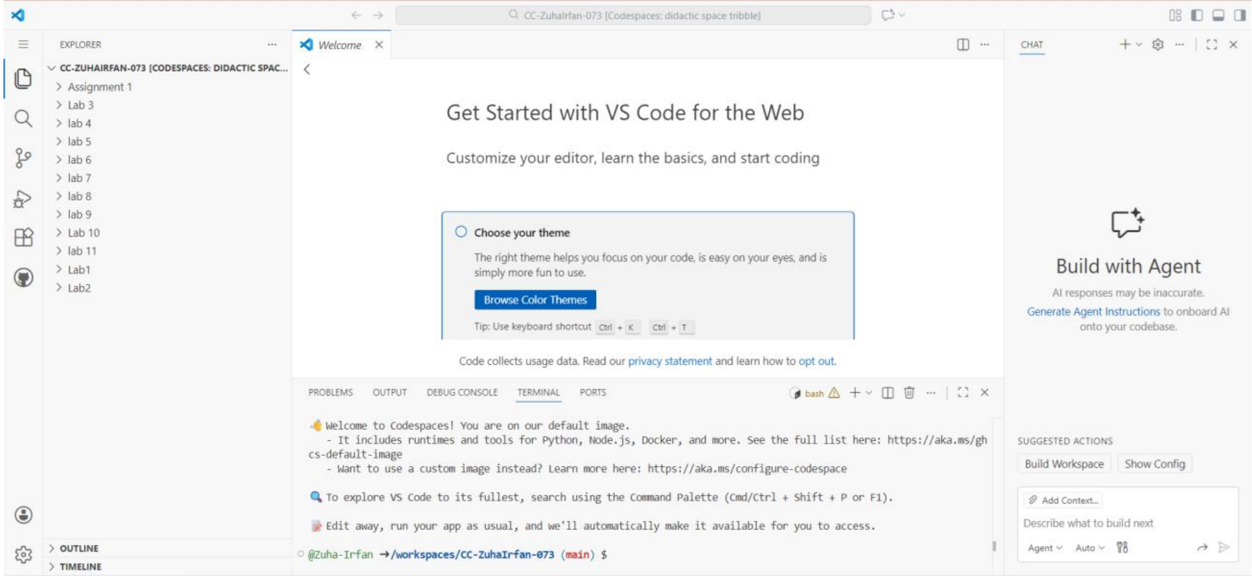
Subject: Cloud Computing

Semester & Section: V-B

Lab: 11

Task 0 Lab Setup (Codespace & GH CLI)

- taskA_codespace_create_and_list.png



Get Started with VS Code for the Web

Customize your editor, learn the basics, and start coding

○ Choose your theme

The right theme helps you focus on your code, is easy on your eyes, and is simply more fun to use.

Tip: Use keyboard shortcut Ctrl + K Ctrl + T

Code collects usage data. Read our privacy statement and learn how to opt out.

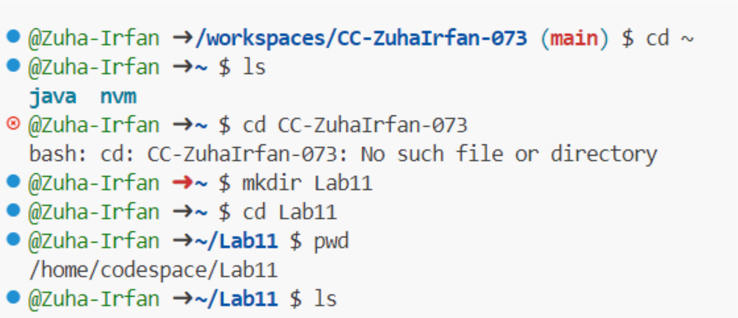
Build with Agent

AI responses may be inaccurate.

Generate Agent Instructions to onboard AI onto your codebase.

```
👋 Welcome to Codespaces! You are on our default image.
   - It includes runtimes and tools for Python, Node.js, Docker, and more. See the full list here: https://aka.ms/gh
cs-default-image
   - Want to use a custom image instead? Learn more here: https://aka.ms/configure-codespace

🔍 To explore VS Code to its fullest, search using the Command Palette (Cmd/Ctrl + Shift + P or F1).

📝 Edit away, run your app as usual, and we'll automatically make it available for you to access.

○ @Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073 (main) $
```

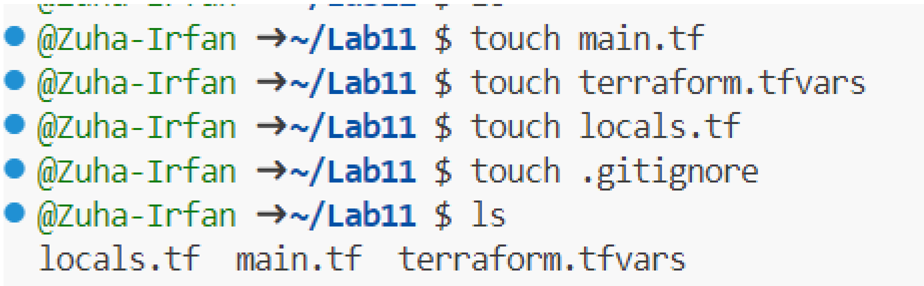
- taskA_codespace_ssh_connected.png



```
● @Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073 (main) $ cd ~
● @Zuha-Irfan ➜~ $ ls
  java  nvm
⊗ @Zuha-Irfan ➜~ $ cd CC-ZuhaIrfan-073
  bash: cd: CC-ZuhaIrfan-073: No such file or directory
● @Zuha-Irfan ➜~ $ mkdir Lab11
● @Zuha-Irfan ➜~ $ cd Lab11
● @Zuha-Irfan ➜~/Lab11 $ pwd
  /home/codespace/Lab11
● @Zuha-Irfan ➜~/Lab11 $ ls
```

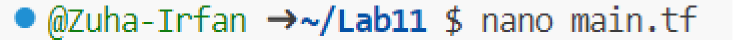
Task 1 — Provider & Basic variable (variable precedence)

- task1_touch_main_tf.png

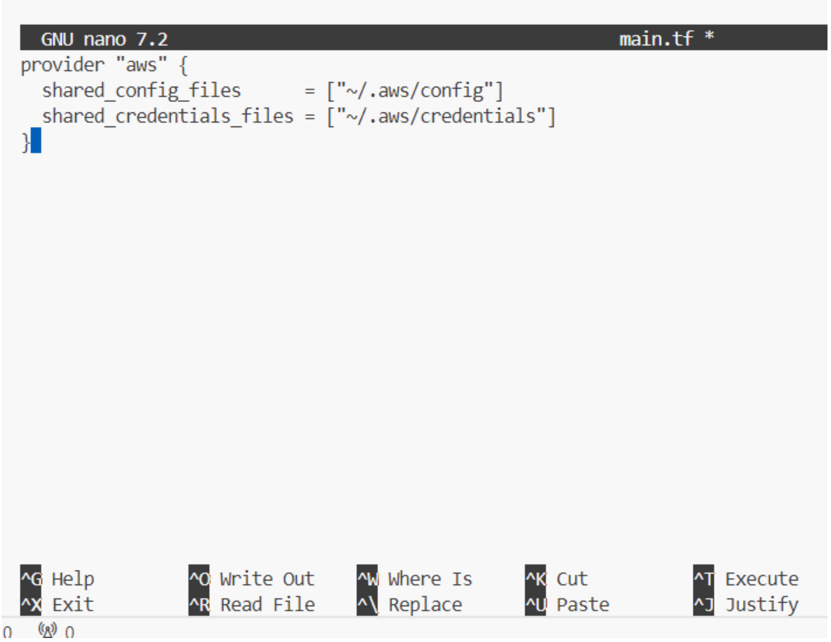


```
● @Zuha-Irfan ➜~/Lab11 $ touch main.tf
● @Zuha-Irfan ➜~/Lab11 $ touch terraform.tfvars
● @Zuha-Irfan ➜~/Lab11 $ touch locals.tf
● @Zuha-Irfan ➜~/Lab11 $ touch .gitignore
● @Zuha-Irfan ➜~/Lab11 $ ls
locals.tf  main.tf  terraform.tfvars
```

- task1_main_tf_provider.png

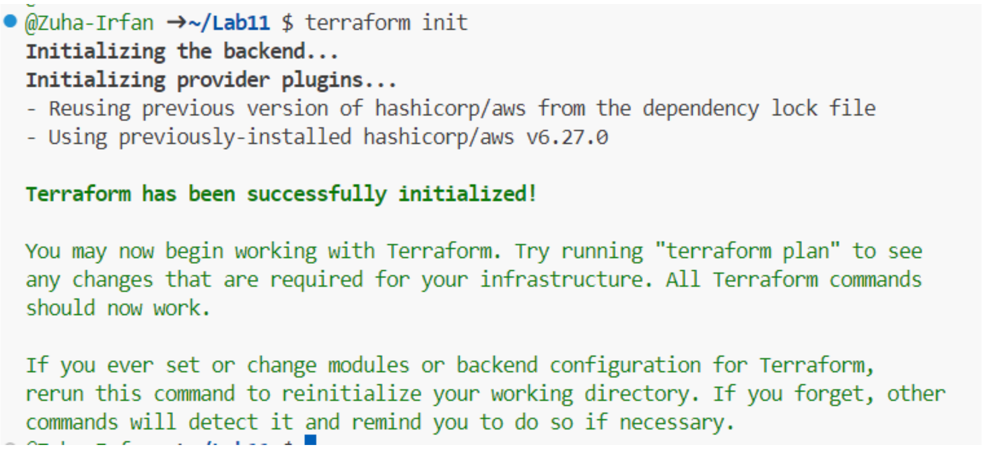


```
● @Zuha-Irfan ➜~/Lab11 $ nano main.tf
```



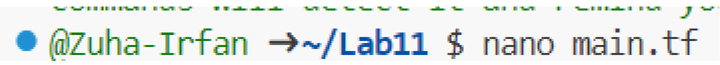
```
 GNU nano 7.2                                        main.tf *
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
```

- task1_terraform_init.png

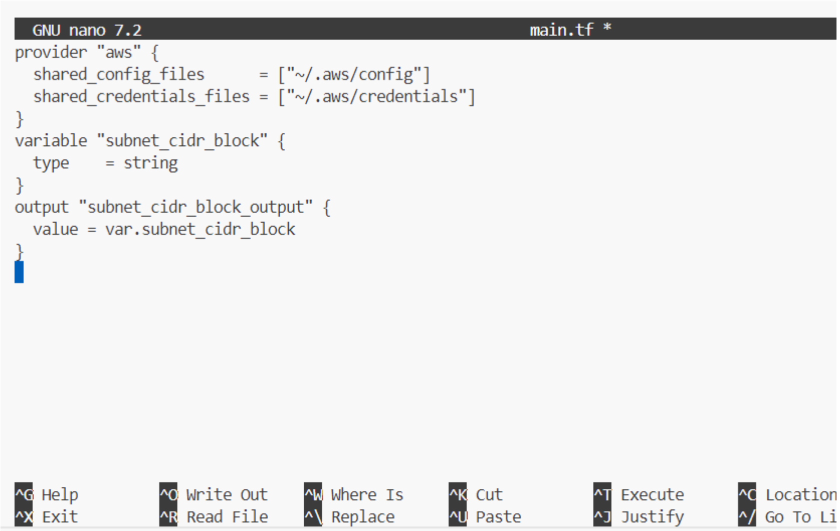


```
● @Zuha-Irfan ➜~/Lab11 $ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.27.0

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.
```

- task1_variable_and_output_added.png

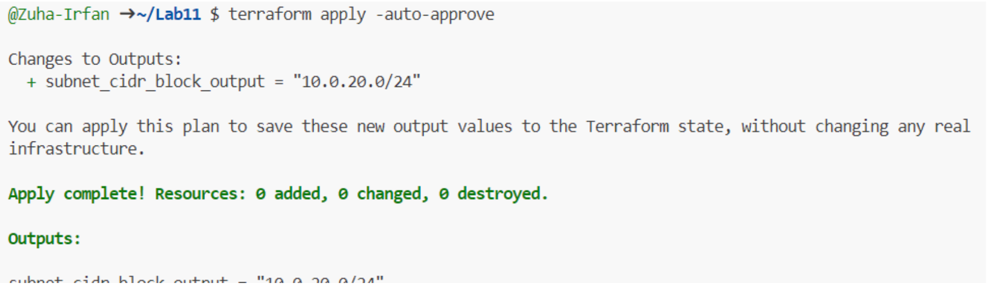


```
● @Zuha-Irfan →~/Lab11 $ nano main.tf
```



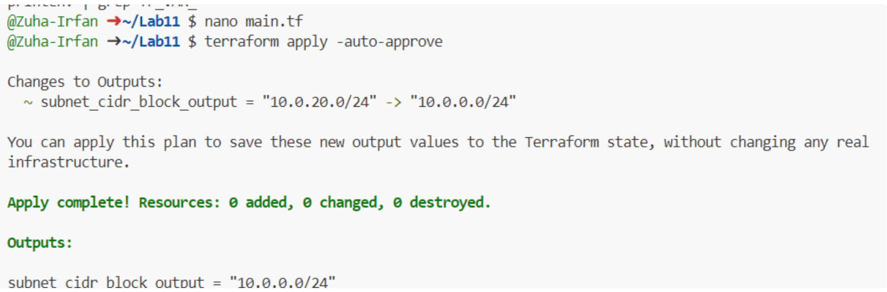
```
  GNU nano 7.2                                          main.tf *
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
variable "subnet_cidr_block" {
  type    = string
}
output "subnet_cidr_block_output" {
  value = var.subnet_cidr_block
}





^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Li
```

- task1_apply_prompt_for_var.png



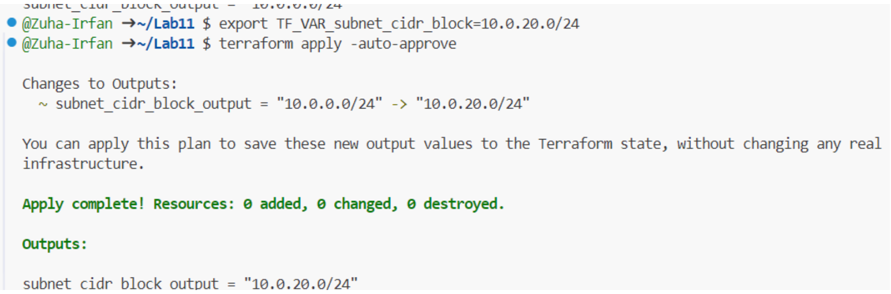
```
@Zuha-Irfan →~/Lab11 $ terraform apply -auto-approve

Changes to Outputs:
  + subnet_cidr_block_output = "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"
```

- task1_apply_with_default.png



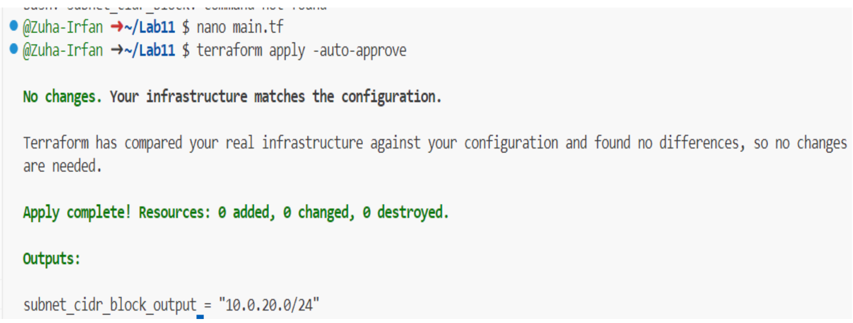
```
printenv | grep TF_VAR_
@Zuha-Irfan →~/Lab11 $ nano main.tf
@Zuha-Irfan →~/Lab11 $ terraform apply -auto-approve

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

- task1_env_var_set_and_apply.png



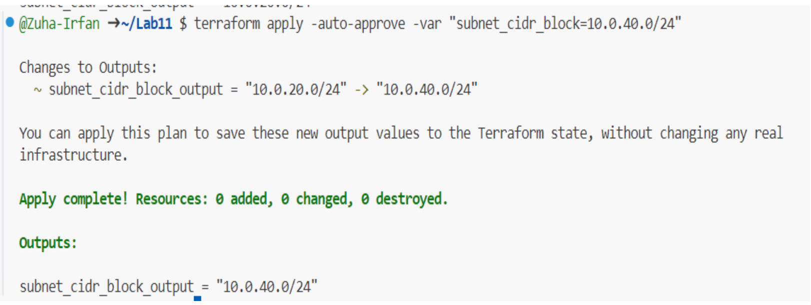
```
subnet_cidr_block_output = "10.0.0.0/24"
● @Zuha-Irfan →~/Lab11 $ export TF_VAR_subnet_cidr_block=10.0.20.0/24
● @Zuha-Irfan →~/Lab11 $ terraform apply -auto-approve

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"
```

- task1_terraform_tfvars_and_apply.png



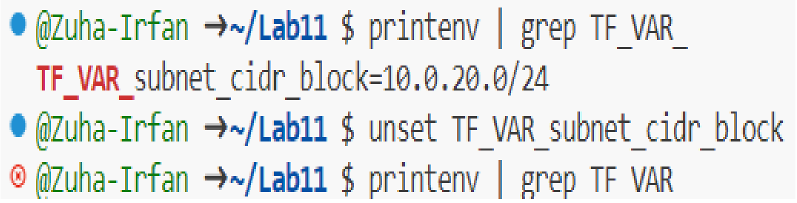
```
bash: subnet_cidr_block: command not found
● @Zuha-Irfan →~/Lab11 $ nano main.tf
● @Zuha-Irfan →~/Lab11 $ terraform apply -auto-approve

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:

subnet_cidr_block_output = "10.0.20.0/24"
```

- task1_var_override_with_dash_var.png



```
subnet_cidr_block_output = "10.0.20.0/24"
● @Zuha-Irfan →~/Lab11 $ terraform apply -auto-approve -var "subnet_cidr_block=10.0.40.0/24"

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

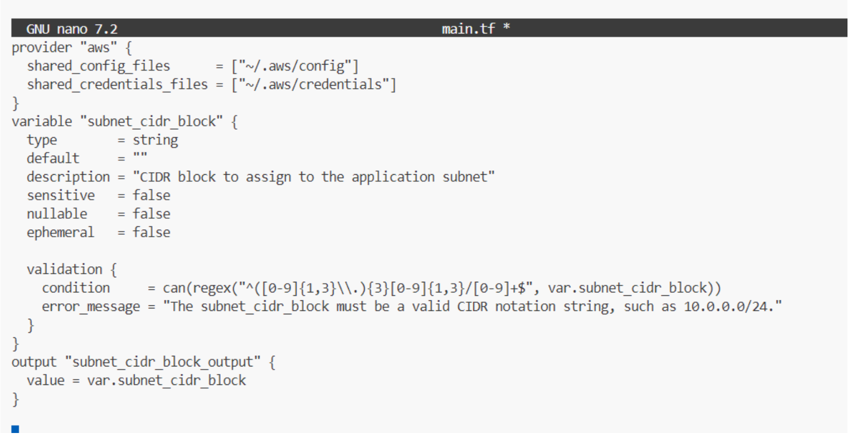
- task1_printenv_tf_var_and_unset.png



```
● @Zuha-Irfan →~/Lab11 $ printenv | grep TF_VAR_
  TF_VAR_subnet_cidr_block=10.0.20.0/24
● @Zuha-Irfan →~/Lab11 $ unset TF_VAR_subnet_cidr_block
⊗ @Zuha-Irfan →~/Lab11 $ printenv | grep TF_VAR
```

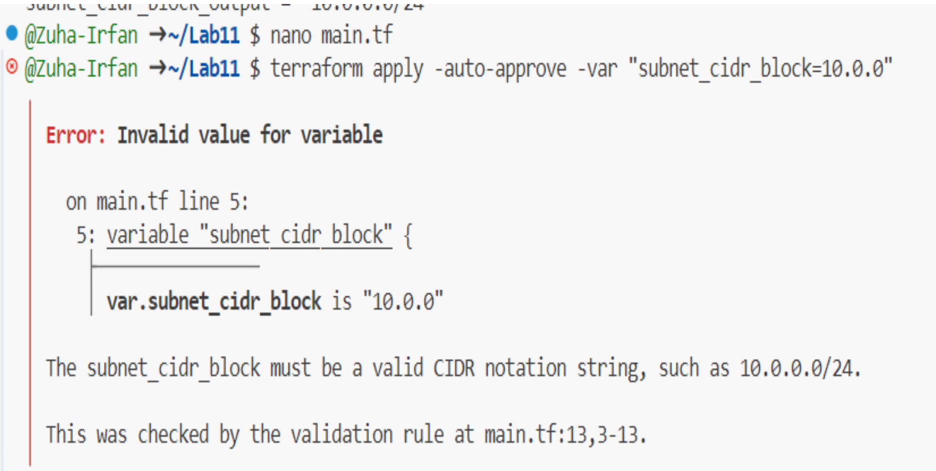
Task 2 — Variable validation & sensitive / ephemeral variables

- task2_subnet_variable_with_validation.png



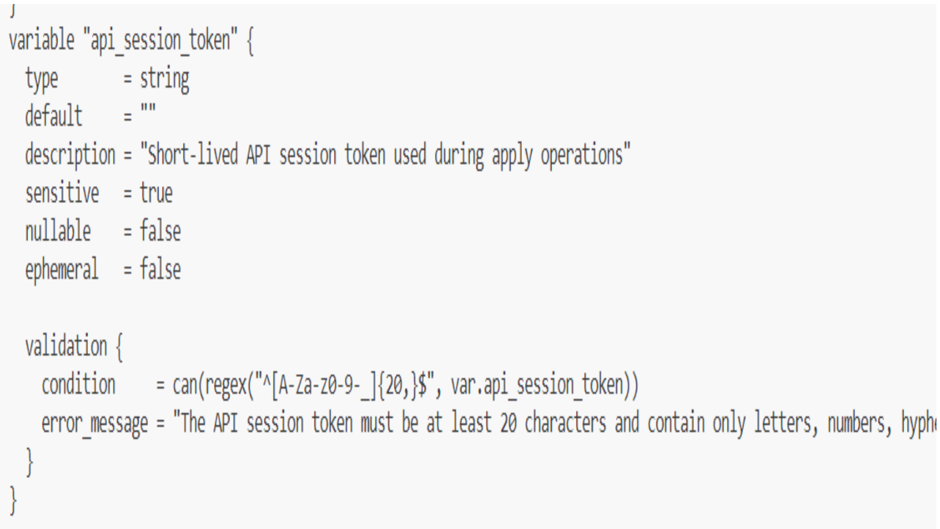
```
GNU nano 7.2                                              main.tf *
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
variable "subnet_cidr_block" {
  type        = string
  default     = ""
  description = "CIDR block to assign 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.png



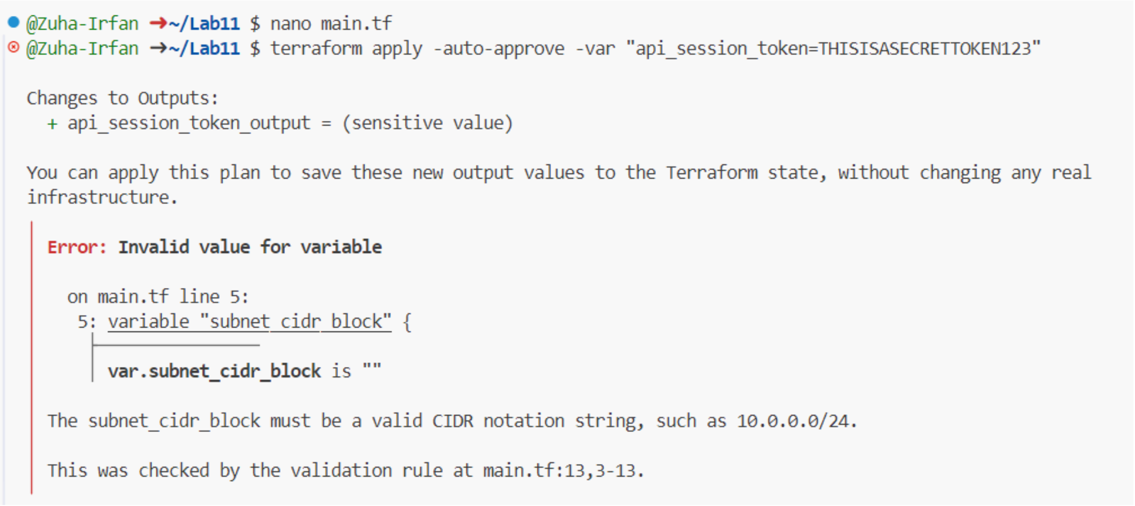
```
  subnet_cidr_block_output = "10.0.0.0/24"
● @Zuha-Irfan →~/Lab11 $ nano main.tf
⊗ @Zuha-Irfan →~/Lab11 $ terraform apply -auto-approve -var "subnet_cidr_block=10.0.0"

│ 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.
```

- task2_api_token_variable_added.png



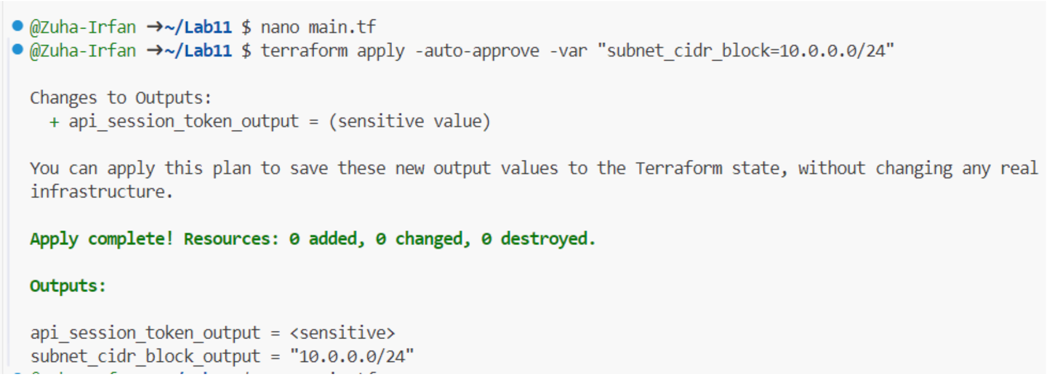
```
}
variable "api_session_token" {
  type        = string
  default     = ""
  description = "Short-lived API session token used during apply operations"
  sensitive   = true
  nullable    = false
  ephemeral   = false

  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, hyph
  }
}
```

- task2_api_token_apply_sensitive.png



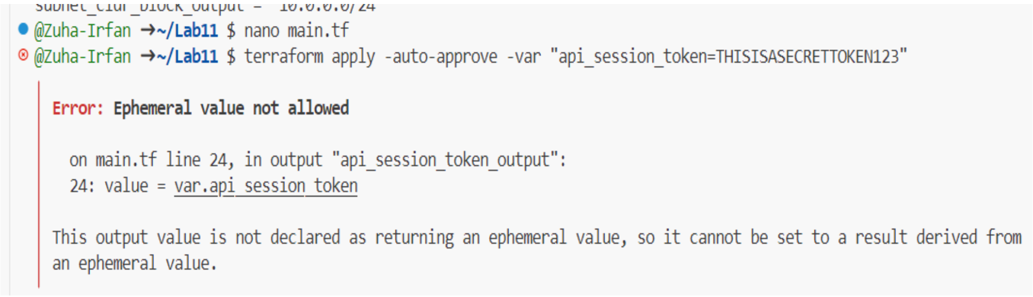
```
● @Zuha-Irfan →~/Lab11 $ nano main.tf
⊗ @Zuha-Irfan →~/Lab11 $ terraform apply -auto-approve -var "api_session_token=THISISASECRETTOKEN123"

Changes to Outputs:
  + api_session_token_output = (sensitive value)

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

  Error: Invalid value for variable

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

  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.
```

- task2_check_terraform_state_api_token.png



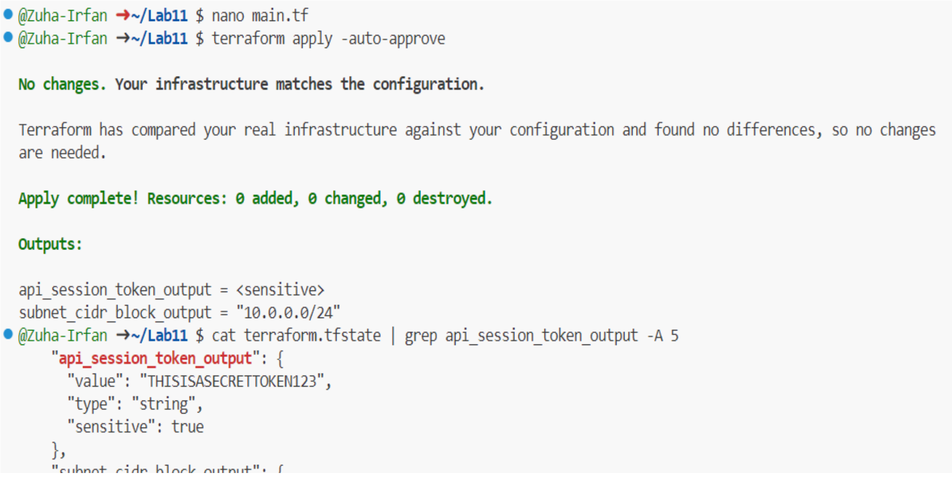
```
● @Zuha-Irfan →~/Lab11 $ nano main.tf
● @Zuha-Irfan →~/Lab11 $ terraform apply -auto-approve -var "subnet_cidr_block=10.0.0.0/24"

Changes to Outputs:
  + api_session_token_output = (sensitive value)

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

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

Outputs:

api_session_token_output = <sensitive>
subnet_cidr_block_output = "10.0.0.0/24"
```

- task2_api_token_ephemeral_error.png



```
subnet_cidr_block_output = "10.0.0.0/24"
● @Zuha-Irfan →~/Lab11 $ nano main.tf
⊗ @Zuha-Irfan →~/Lab11 $ terraform apply -auto-approve -var "api_session_token=THISISASECRETTOKEN123"

  Error: Ephemeral value not allowed

    on main.tf line 24, in output "api_session_token_output":
    24: 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.
```

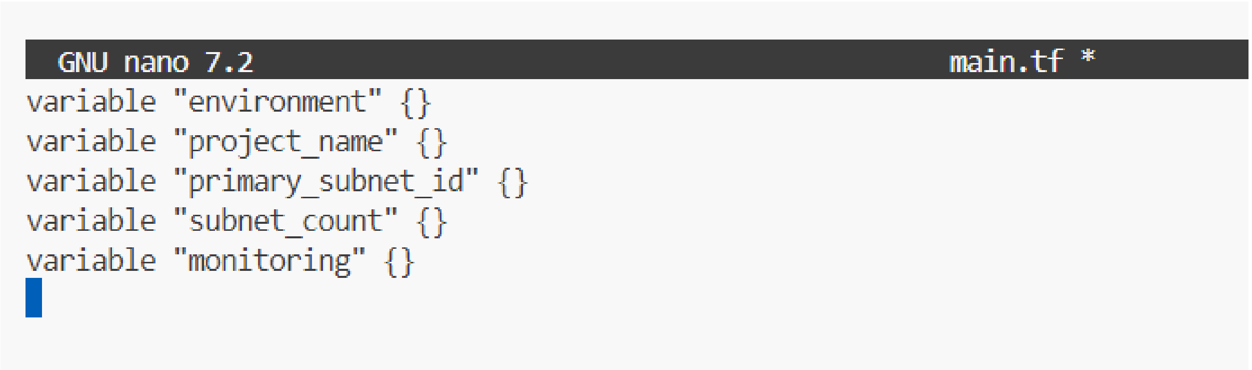
- task2_api_token_default_apply.png



```
@Zuha-Irfan →~/Lab11 $ nano main.tf
@Zuha-Irfan →~/Lab11 $ 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 = "10.0.0.0/24"
@Zuha-Irfan →~/Lab11 $ cat terraform.tfstate | grep api_session_token_output -A 5
    "api_session_token_output": {
      "value": "THISISASECRETTOKEN123",
      "type": "string",
      "sensitive": true
    },
    "subnet_cidr_block_output": {
```

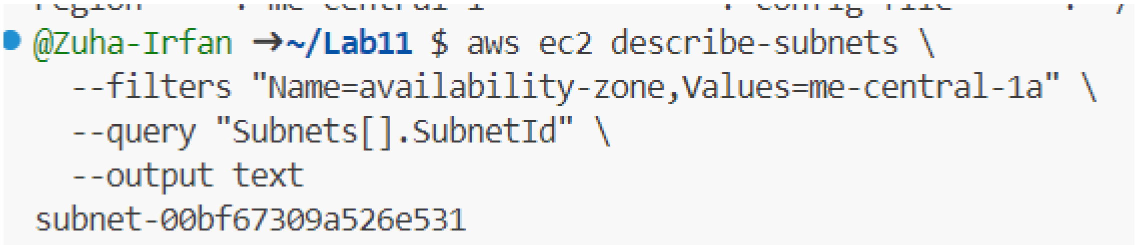
Task 3 — Project-level variables, locals, and outputs

- task3_variables_added.png

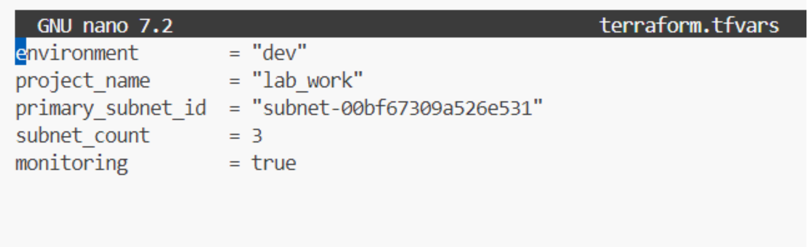


```
 GNU nano 7.2                                        main.tf *
variable "environment" {}
variable "project_name" {}
variable "primary_subnet_id" {}
variable "subnet_count" {}
variable "monitoring" {}
```

- task3_terraform_tfvars_populated.png

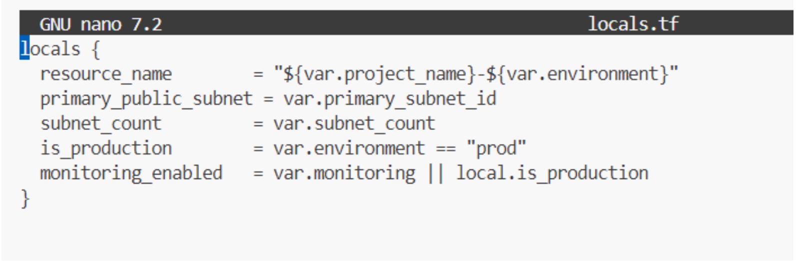


```
@Zuha-Irfan →~/Lab11 $ aws ec2 describe-subnets \
  --filters "Name=availability-zone,Values=me-central-1a" \
  --query "Subnets[].SubnetId" \
  --output text
subnet-00bf67309a526e531
```



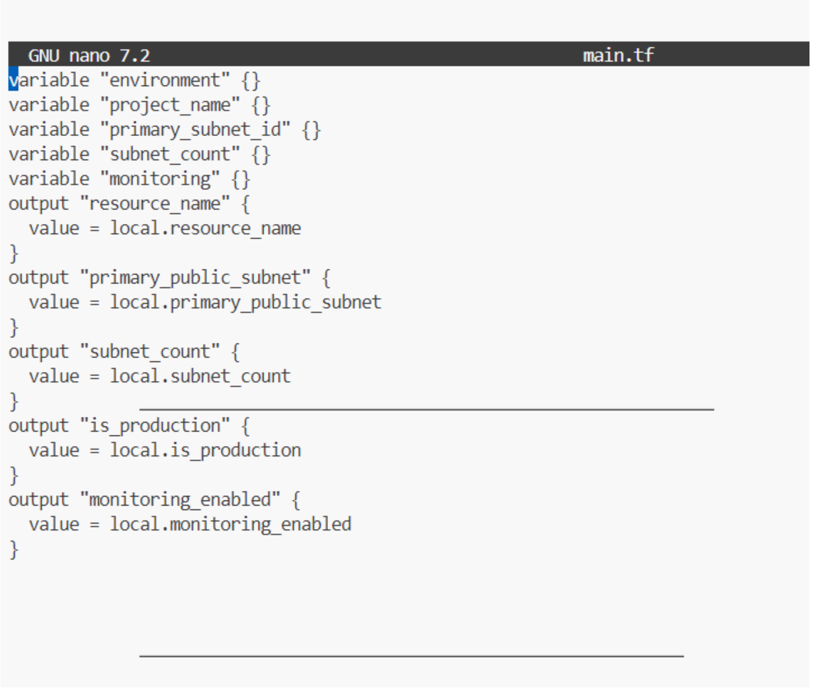
```
 GNU nano 7.2                               terraform.tfvars
environment        = "dev"
project_name       = "lab_work"
primary_subnet_id  = "subnet-00bf67309a526e531"
subnet_count       = 3
monitoring         = true
```

- task3_locals_tf_created.png

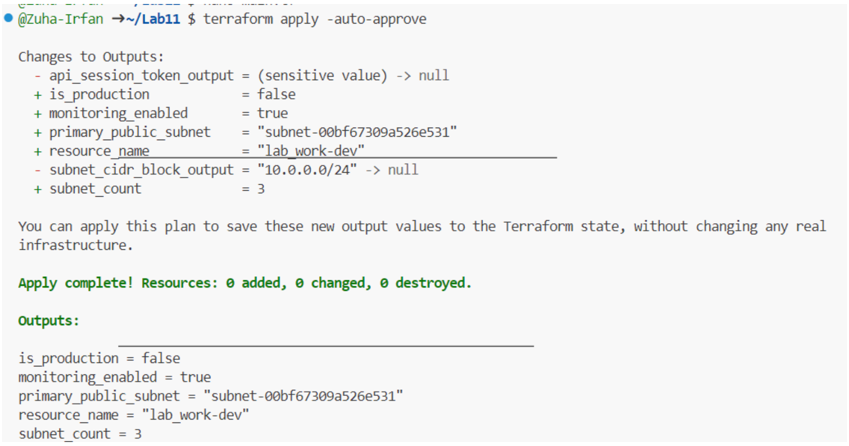


```
 GNU nano 7.2                                          locals.tf
locals {
  resource_name         = "${var.project_name}-${var.environment}"
  primary_public_subnet = var.primary_subnet_id
  subnet_count          = var.subnet_count
  is_production         = var.environment == "prod"
  monitoring_enabled    = var.monitoring || local.is_production
}
```

- task3_outputs_apply.png



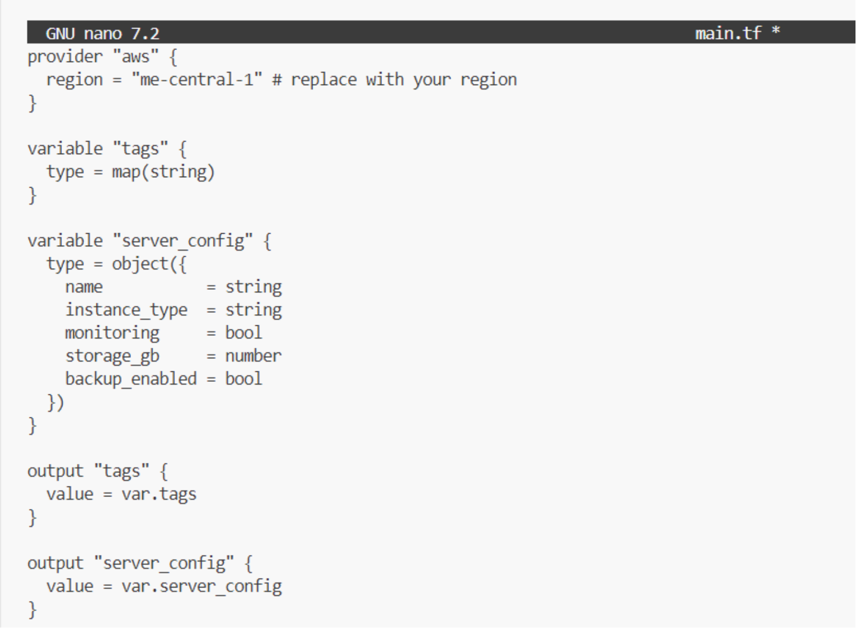
```
 GNU nano 7.2                                          main.tf
variable "environment" {}
variable "project_name" {}
variable "primary_subnet_id" {}
variable "subnet_count" {}
variable "monitoring" {}
output "resource_name" {
  value = local.resource_name
}
output "primary_public_subnet" {
  value = 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
}
```



```
● @Zuha-Irfan →~/Lab11 $ terraform apply -auto-approve

Changes to Outputs:
  - api_session_token_output = (sensitive value) -> null
  + is_production            = false
  + monitoring_enabled       = true
  + primary_public_subnet    = "subnet-00bf67309a526e531"
  + resource_name            = "lab_work-dev"
  - subnet_cidr_block_output = "10.0.0.0/24" -> null
  + subnet_count             = 3

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

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

Outputs:

is_production = false
monitoring_enabled = true
primary_public_subnet = "subnet-00bf67309a526e531"
resource_name = "lab_work-dev"
subnet_count = 3
```

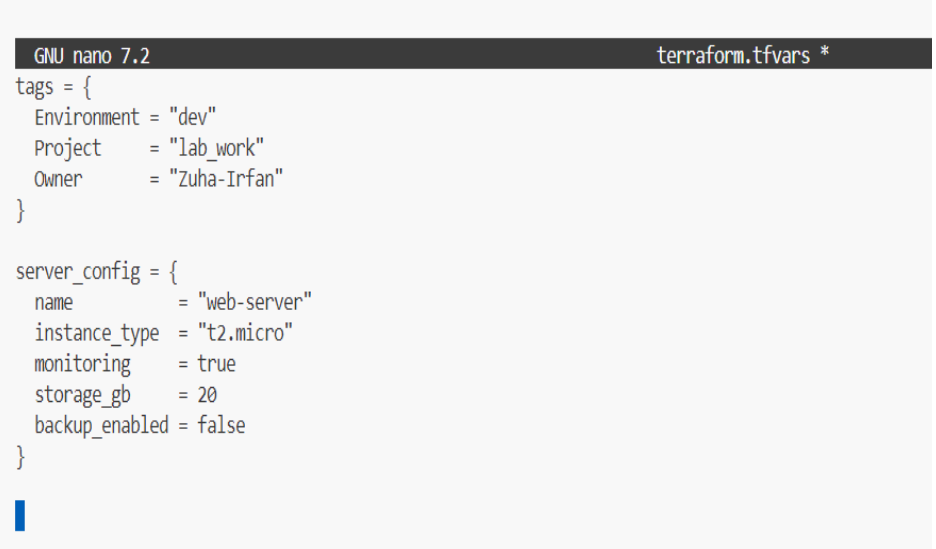
Task 4 — Maps and Objects

- task4_tags_variable_added.png



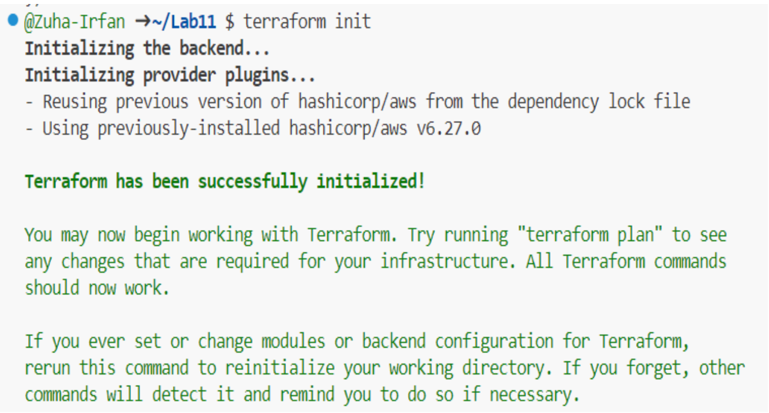
```
provider "aws" {
  region = "me-central-1" # replace with your region
}

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

variable "server_config" {
  type = object({
    name          = string
    instance_type  = string
    monitoring     = bool
    storage_gb     = number
    backup_enabled = bool
  })
}

output "tags" {
  value = var.tags
}

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

- task4_tags_output.png

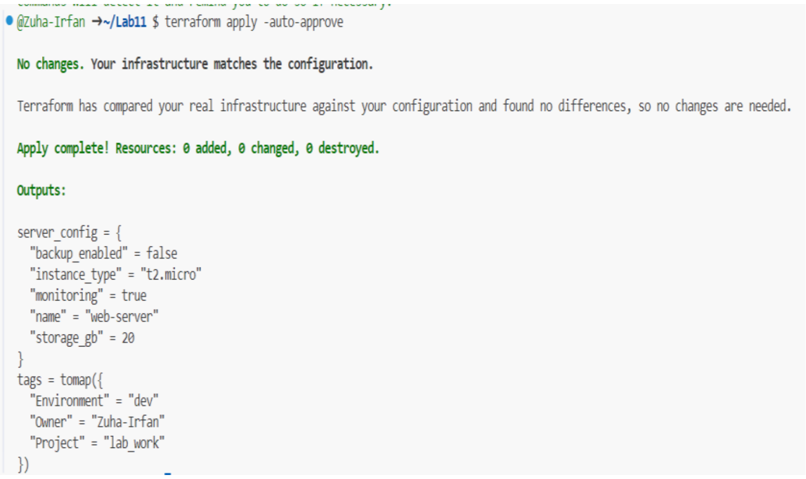


```
tags = {
  Environment = "dev"
  Project     = "lab_work"
  Owner       = "Zuha-Irfan"
}

server_config = {
  name           = "web-server"
  instance_type  = "t2.micro"
  monitoring     = true
  storage_gb     = 20
  backup_enabled = false
}
```

- task4_server_config_output.png



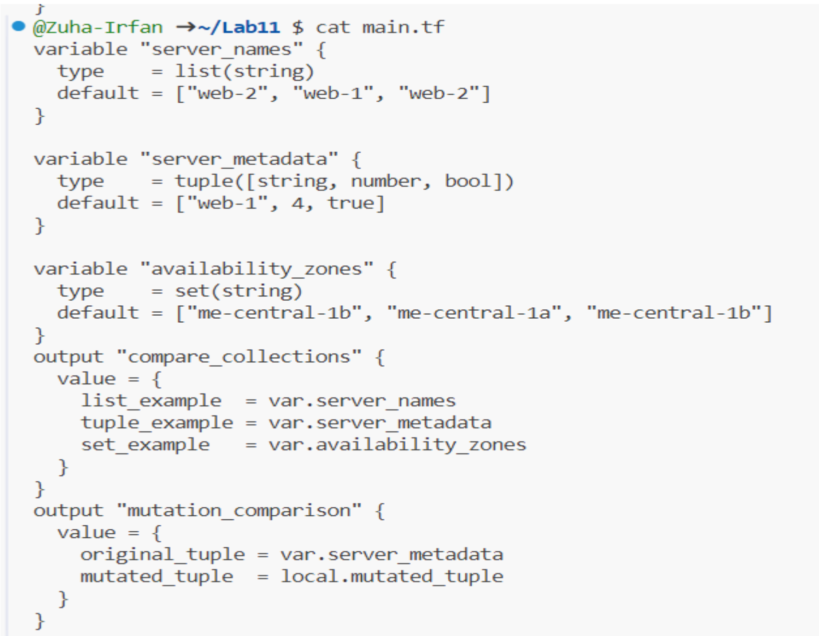
```
● @Zuha-Irfan →~/Lab11 $ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.27.0

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.
```



```
● @Zuha-Irfan →~/Lab11 $ 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:

server_config = {
  "backup_enabled" = false
  "instance_type" = "t2.micro"
  "monitoring" = true
  "name" = "web-server"
  "storage_gb" = 20
}
tags = tomap({
  "Environment" = "dev"
  "Owner" = "Zuha-Irfan"
  "Project" = "lab_work"
})
```

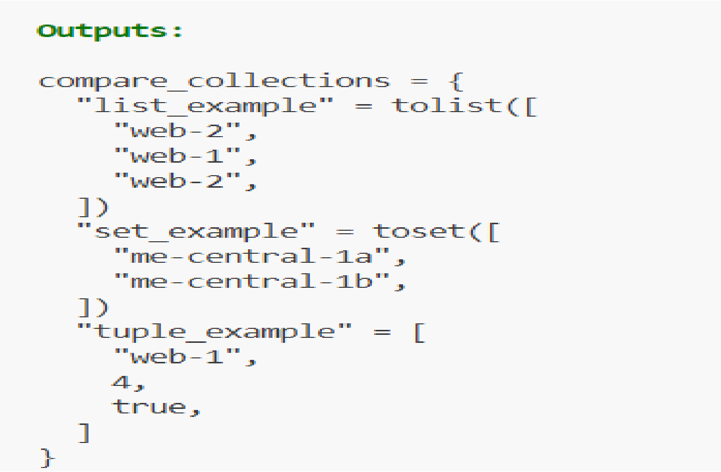
Task 5 — Collections: list, tuple, set & mutation via locals

- task5_collections_defined.png



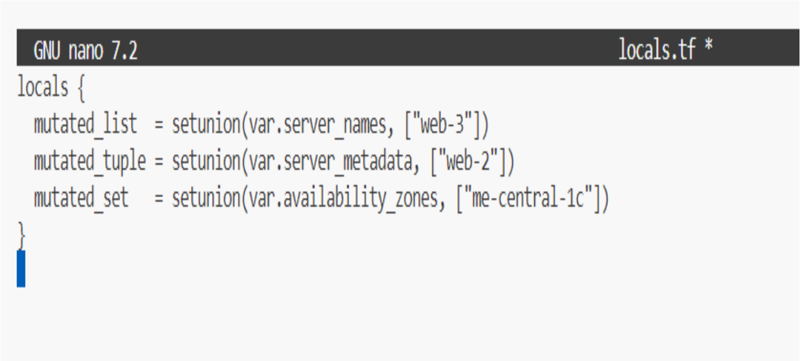
```
● @Zuha-Irfan →~/Lab11 $ cat main.tf
variable "server_names" {
  type    = list(string)
  default = ["web-2", "web-1", "web-2"]
}

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

variable "availability_zones" {
  type    = set(string)
  default = ["me-central-1b", "me-central-1a", "me-central-1b"]
}
output "compare_collections" {
  value = {
    list_example  = var.server_names
    tuple_example = var.server_metadata
    set_example   = var.availability_zones
  }
}
output "mutation_comparison" {
  value = {
    original_tuple = var.server_metadata
    mutated_tuple  = local.mutated_tuple
  }
}
```

- task5_compare_collections.png



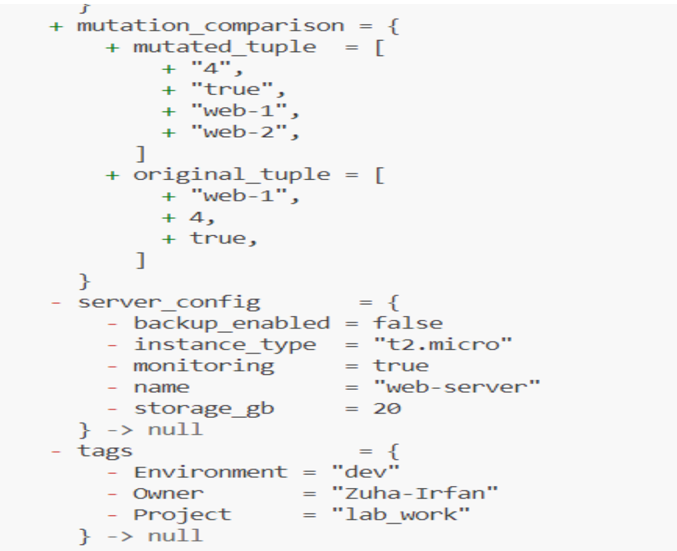
```
Outputs:

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,
  ]
}
```

- task5_locals_mutations.png



```
GNU nano 7.2                                              locals.tf *
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"])
}
```

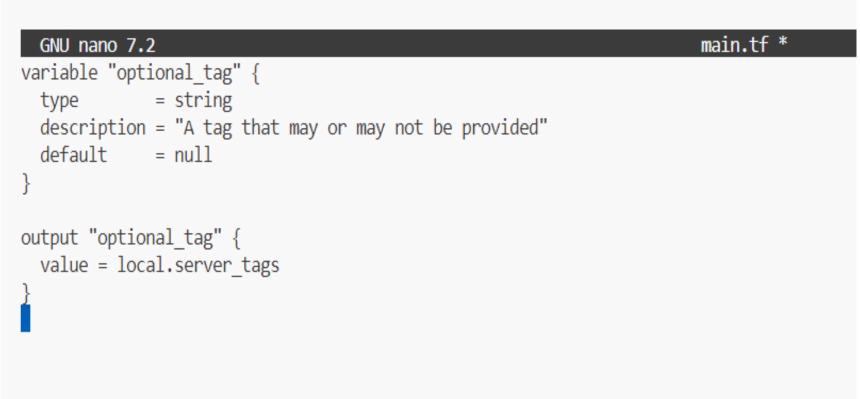
- task5_mutation_comparison.png



```
      }
    + mutation_comparison = {
        + mutated_tuple  = [
            + "4",
            + "true",
            + "web-1",
            + "web-2",
          ]
        + original_tuple = [
            + "web-1",
            + 4,
            + true,
          ]
      }
    - server_config        = {
        - backup_enabled = false
        - instance_type  = "t2.micro"
        - monitoring     = true
        - name           = "web-server"
        - storage_gb     = 20
      } -> null
    - tags                 = {
        - Environment = "dev"
        - Owner       = "Zuha-Irfan"
        - Project     = "lab_work"
      } -> null
```

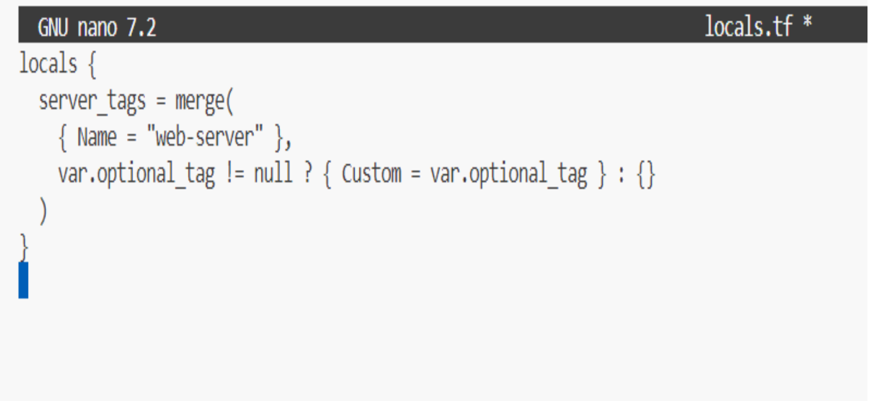
Task 6 — Null, any type & dynamic values

- task6_optional_tag_variable.png



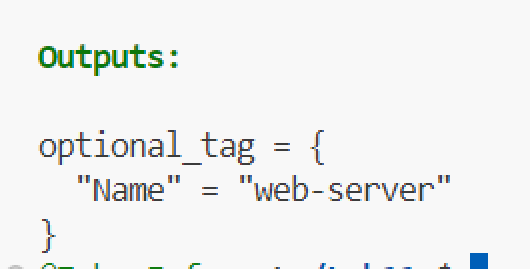
```
GNU nano 7.2                                              main.tf *
variable "optional_tag" {
  type        = string
  description = "A tag that may or may not be provided"
  default     = null
}

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

- task6_locals_merge.png



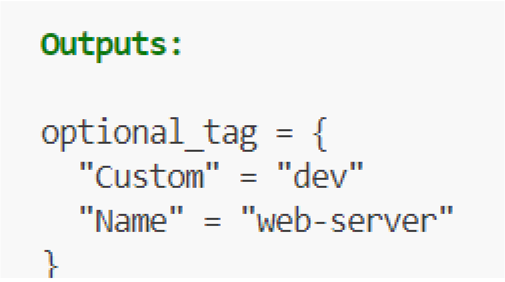
```
GNU nano 7.2                                              locals.tf *
locals {
  server_tags = merge(
    { Name = "web-server" },
    var.optional_tag != null ? { Custom = var.optional_tag } : {}
  )
}
```

- task6_optional_tag_no_value.png



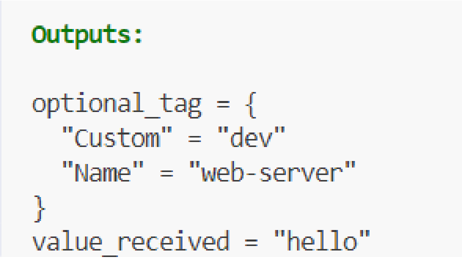
```
Outputs:

optional_tag = {
  "Name" = "web-server"
}
```

- task6_optional_tag_with_value.png



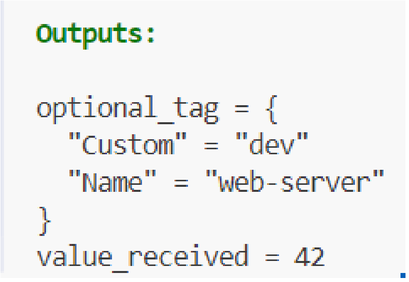
```
Outputs:

optional_tag = {
  "Custom" = "dev"
  "Name" = "web-server"
}
```

- task6_dynamic_value_string.png



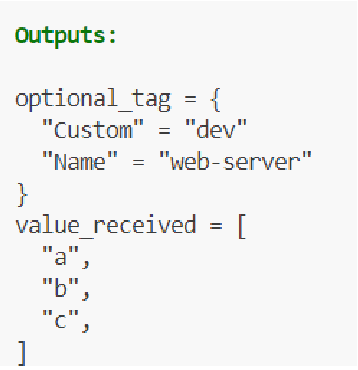
```
Outputs:

optional_tag = {
  "Custom" = "dev"
  "Name" = "web-server"
}
value_received = "hello"
```

- task6_dynamic_value_number.png



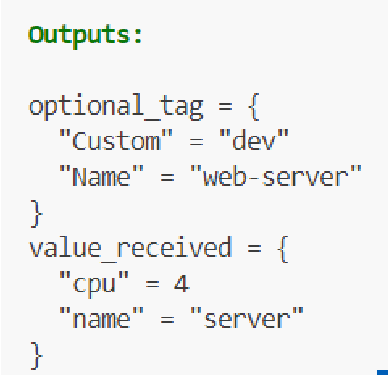
```
Outputs:

optional_tag = {
  "Custom" = "dev"
  "Name" = "web-server"
}
value_received = 42
```

- task6_dynamic_value_list.png



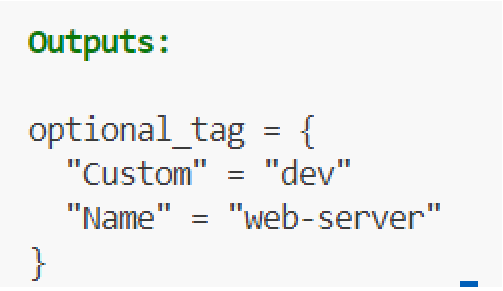
```
Outputs:

optional_tag = {
  "Custom" = "dev"
  "Name" = "web-server"
}
value_received = [
  "a",
  "b",
  "c",
]
```

- task6_dynamic_value_map.png



```
Outputs:

optional_tag = {
  "Custom" = "dev"
  "Name" = "web-server"
}
value_received = {
  "cpu" = 4
  "name" = "server"
}
```

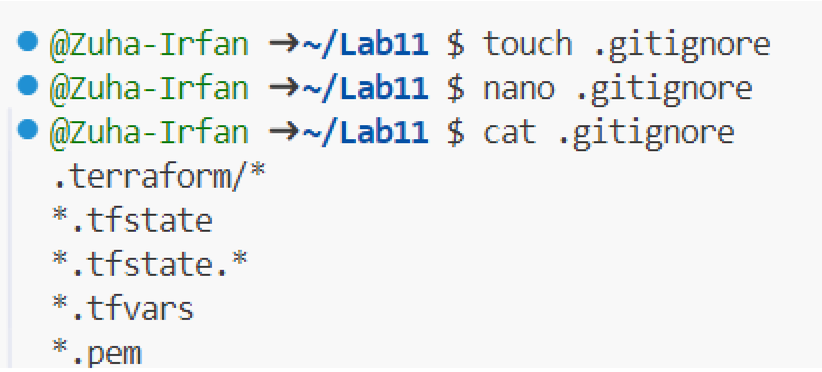
- task6_dynamic_value_null.png



```
Outputs:

optional_tag = {
  "Custom" = "dev"
  "Name" = "web-server"
}
```

Task 7 — Git ignore

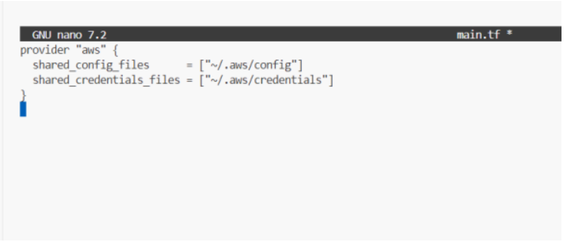
- task7_gitignore_created.png



```
@Zuha-Irfan →~/Lab11 $ touch .gitignore
@Zuha-Irfan →~/Lab11 $ nano .gitignore
@Zuha-Irfan →~/Lab11 $ cat .gitignore
.terraform/*
*.tfstate
*.tfstate.*
*.tfvars
*.pem
```

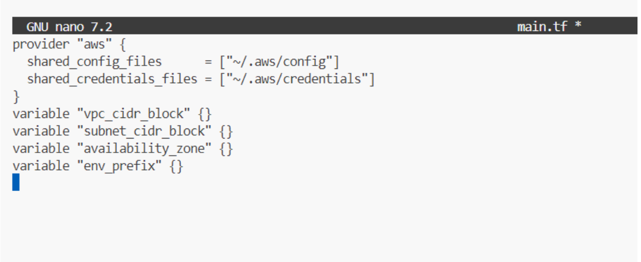
Task 8 — Clean-up then build real infra (VPC, Subnet, IGW, routing, default route table)

- task8_clean_files.png



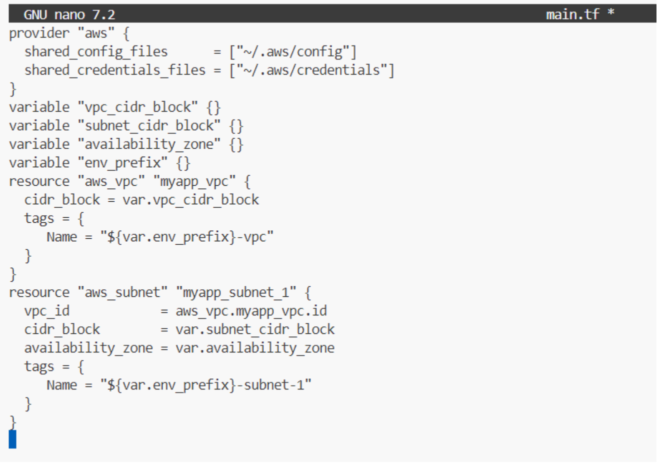
```
  GNU nano 7.2                                           main.tf *
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
```

- task8_variables_recreated.png



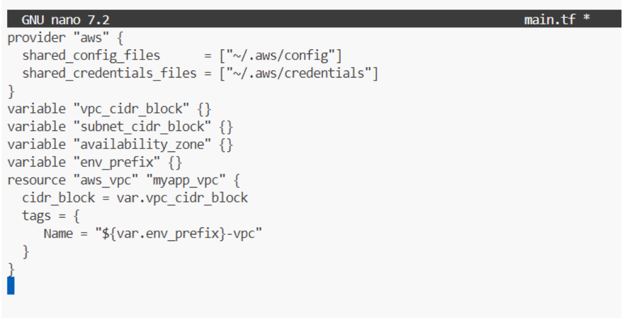
```
  GNU nano 7.2                                                  main.tf *
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
variable "vpc_cidr_block" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
```

- task8_vpc_resources_added.png



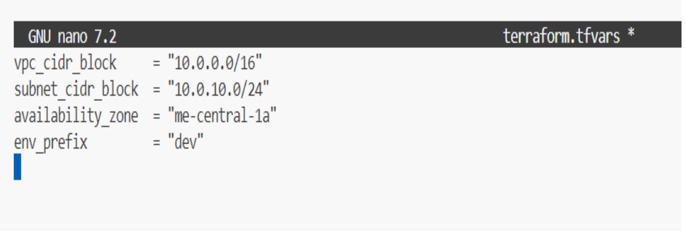
```
GNU nano 7.2                                                    main.tf *
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
variable "vpc_cidr_block" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
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_subnet_resources_added.png



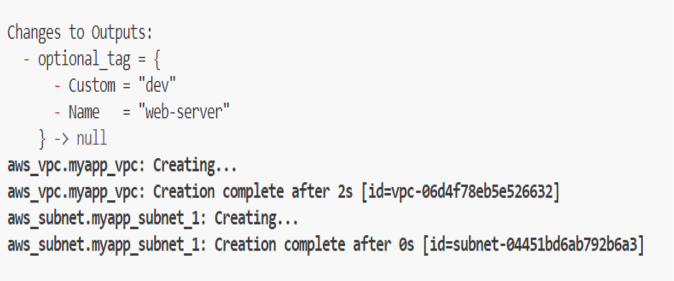
```
GNU nano 7.2                                                    main.tf *
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
variable "vpc_cidr_block" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
resource "aws_vpc" "myapp_vpc" {
  cidr_block = var.vpc_cidr_block
  tags = {
     Name = "${var.env_prefix}-vpc"
  }
}
```

- task8_terraform_tfvars_vpc_values.png



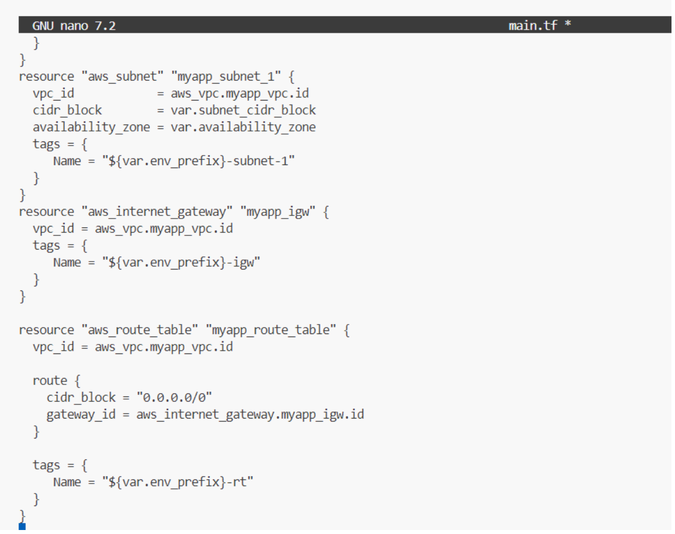
```
GNU nano 7.2                                          terraform.tfvars *
vpc_cidr_block     = "10.0.0.0/16"
subnet_cidr_block  = "10.0.10.0/24"
availability_zone  = "me-central-1a"
env_prefix         = "dev"
```

- task8_vpc_subnet_apply.png



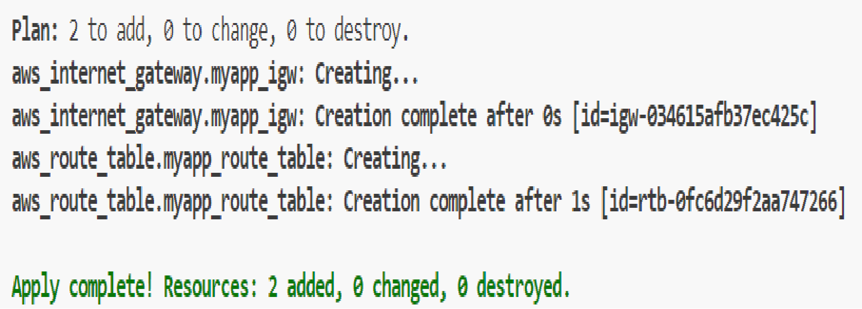
```
Changes to Outputs:
  - optional_tag = {
      - Custom = "dev"
      - Name   = "web-server"
    } -> null
aws_vpc.myapp_vpc: Creating...
aws_vpc.myapp_vpc: Creation complete after 2s [id=vpc-06d4f78eb5e526632]
aws_subnet.myapp_subnet_1: Creating...
aws_subnet.myapp_subnet_1: Creation complete after 0s [id=subnet-04451bd6ab792b6a3]
```

- task8_igw_route_table_before_apply.png



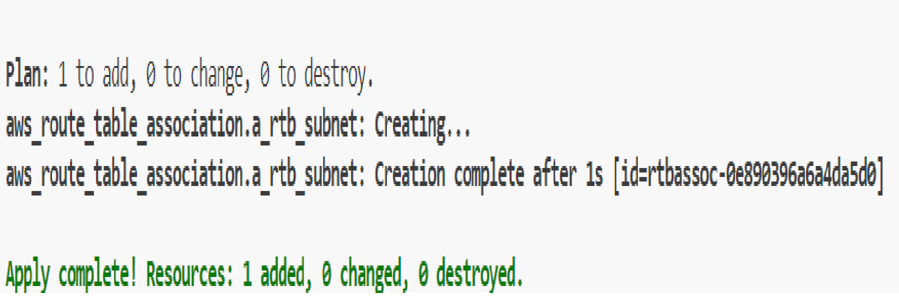
```
  GNU nano 7.2                                          main.tf *
  }
}
resource "aws_subnet" "myapp_subnet_1" {
  vpc_id            = aws_vpc.myapp_vpc.id
  cidr_block        = var.subnet_cidr_block
  availability_zone = var.availability_zone
  tags = {
     Name = "${var.env_prefix}-subnet-1"
  }
}
resource "aws_internet_gateway" "myapp_igw" {
  vpc_id = aws_vpc.myapp_vpc.id
  tags = {
     Name = "${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.png



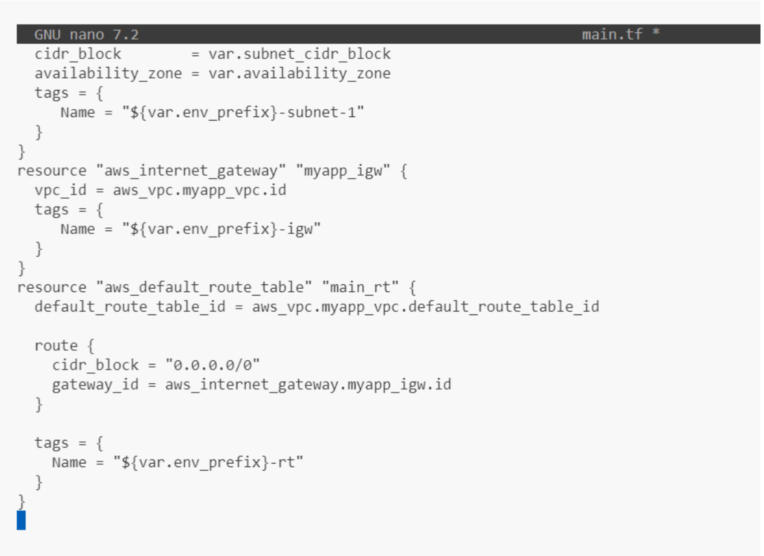
```
Plan: 2 to add, 0 to change, 0 to destroy.
aws_internet_gateway.myapp_igw: Creating...
aws_internet_gateway.myapp_igw: Creation complete after 0s [id=igw-034615afb37ec425c]
aws_route_table.myapp_route_table: Creating...
aws_route_table.myapp_route_table: Creation complete after 1s [id=rtb-0fc6d29f2aa747266]

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

- task8_association_apply.png



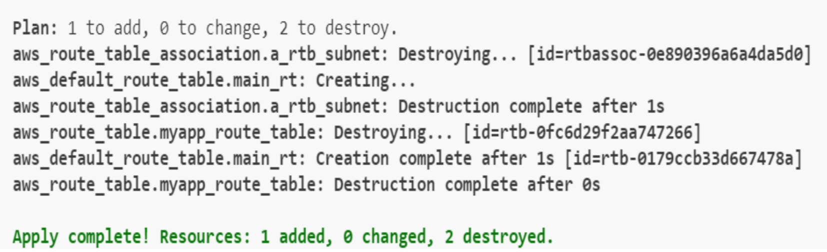
```
Plan: 1 to add, 0 to change, 0 to destroy.
aws_route_table_association.a_rtb_subnet: Creating...
aws_route_table_association.a_rtb_subnet: Creation complete after 1s [id=rtbassoc-0e890396a6a4da5d0]

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

- task8_default_route_table.png



```
 GNU nano 7.2                                                    main.tf *
  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"
  }
}
```

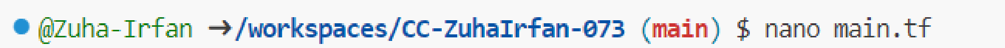
- task8_default_route_table_apply.png



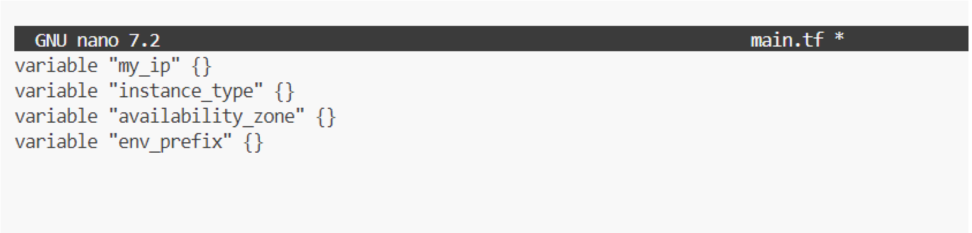
```
Plan: 1 to add, 0 to change, 2 to destroy.
aws_route_table_association.a_rtb_subnet: Destroying... [id=rtbassoc-0e890396a6a4da5d0]
aws_default_route_table.main_rt: Creating...
aws_route_table_association.a_rtb_subnet: Destruction complete after 1s
aws_route_table.myapp_route_table: Destroying... [id=rtb-0fc6d29f2aa747266]
aws_default_route_table.main_rt: Creation complete after 1s [id=rtb-0179ccb33d667478a]
aws_route_table.myapp_route_table: Destruction complete after 0s

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

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

- task9_my_ip_variable_added.png

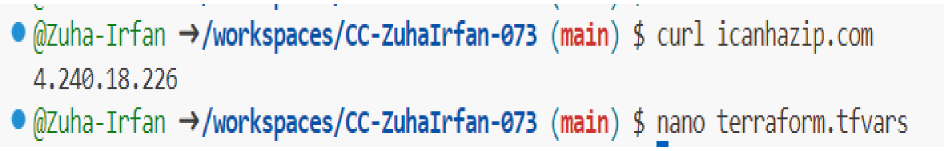


```
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ nano main.tf
```



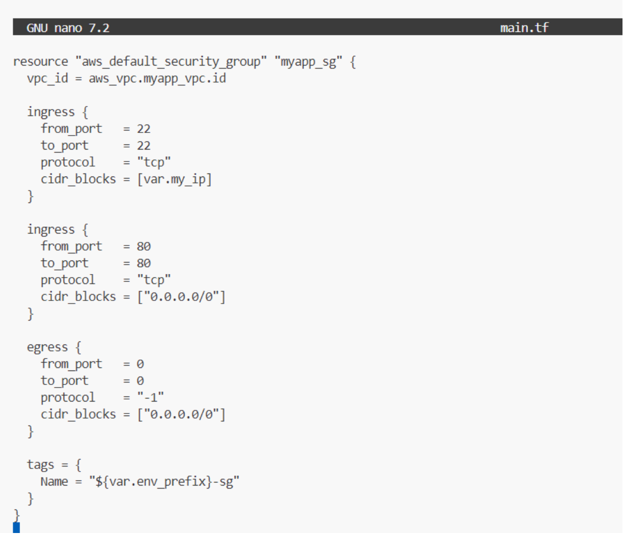
```
 GNU nano 7.2                                          main.tf *
variable "my_ip" {}
variable "instance_type" {}
variable "availability_zone" {}
variable "env_prefix" {}
```

- task9_public_ip_curl.png

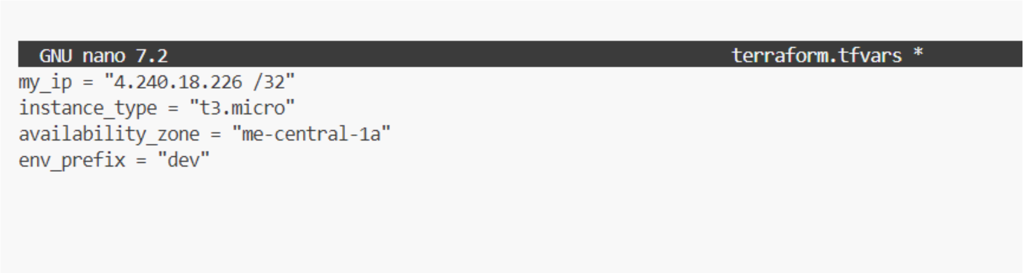


```
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ curl icanhazip.com
4.240.18.226
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ nano terraform.tfvars
```

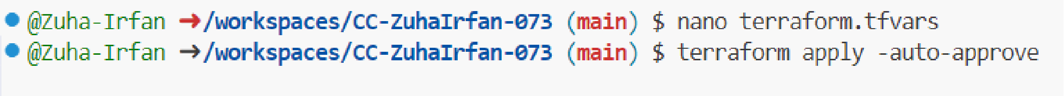
task9_security_group_apply.png



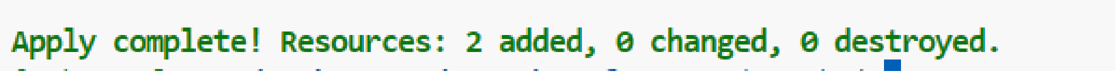
```
GNU nano 7.2                                                    main.tf

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

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

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

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

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



```
GNU nano 7.2                                   terraform.tfvars *
my_ip = "4.240.18.226 /32"
instance_type = "t3.micro"
availability_zone = "me-central-1a"
env_prefix = "dev"
```

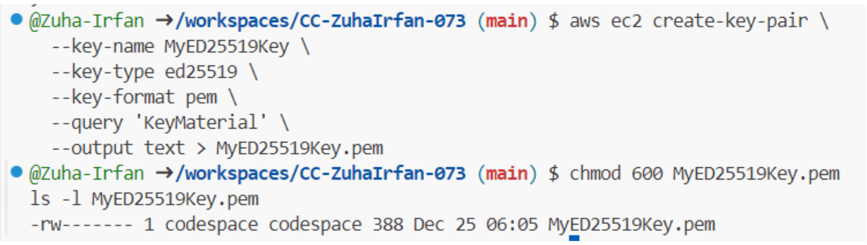


```
● @Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073 (main) $ nano terraform.tfvars
● @Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073 (main) $ terraform apply -auto-approve
```



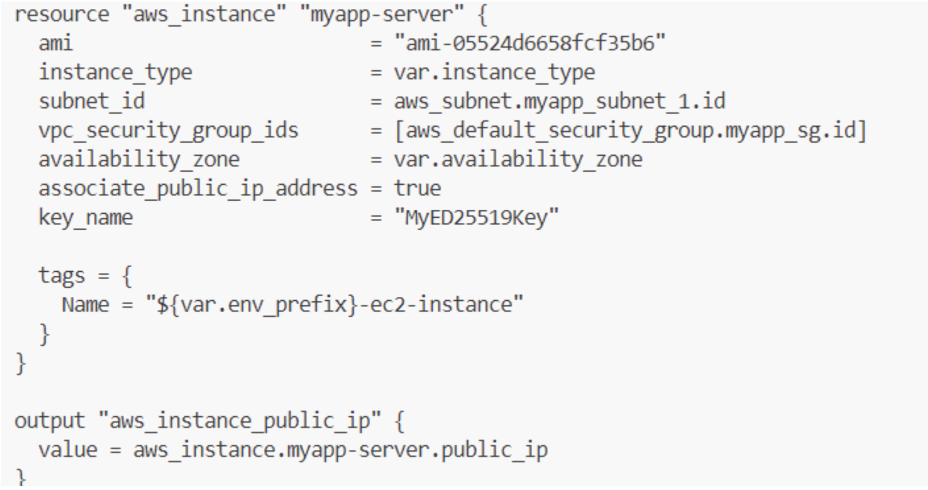
```
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

- task9_keypair_created_and_saved.png



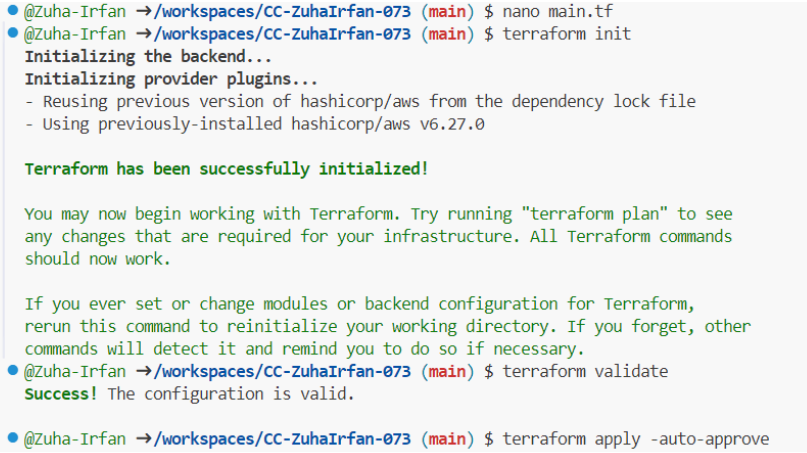
```
● @Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073 (main) $ aws ec2 create-key-pair \
    --key-name MyED25519Key \
    --key-type ed25519 \
    --key-format pem \
    --query 'KeyMaterial' \
    --output text > MyED25519Key.pem
● @Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073 (main) $ chmod 600 MyED25519Key.pem
  ls -l MyED25519Key.pem
  -rw------- 1 codespace codespace 388 Dec 25 06:05 MyED25519Key.pem
```

- task9_instance_type_set.png

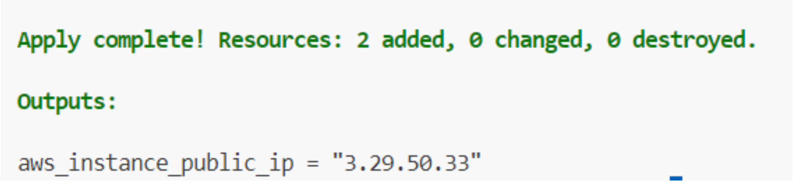


```
resource "aws_instance" "myapp-server" {
  ami                         = "ami-05524d6658fcf35b6"
  instance_type               = var.instance_type
  subnet_id                   = aws_subnet.myapp_subnet_1.id
  vpc_security_group_ids      = [aws_default_security_group.myapp_sg.id]
  availability_zone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

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

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

- task9_ec2_apply_and_public_ip.pn

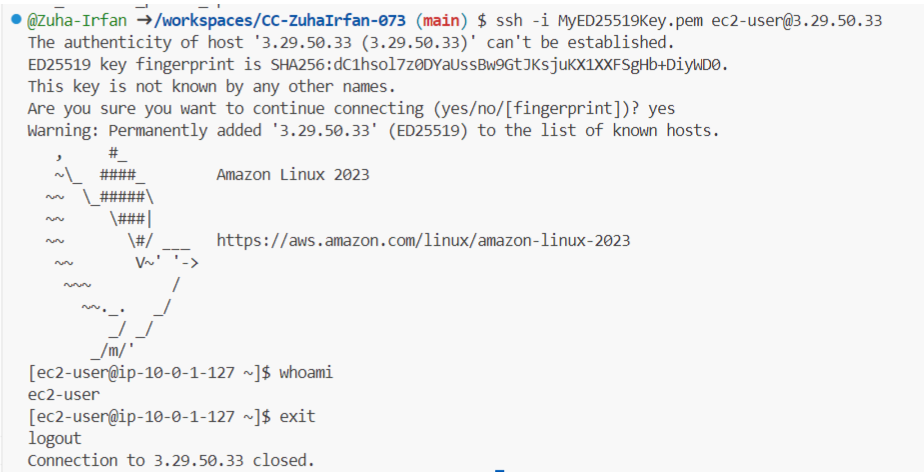


```
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ nano main.tf
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ terraform init
  Initializing the backend...
  Initializing provider plugins...
  - Reusing previous version of hashicorp/aws from the dependency lock file
  - Using previously-installed 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.
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ terraform validate
  Success! The configuration is valid.

● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ terraform apply -auto-approve
```



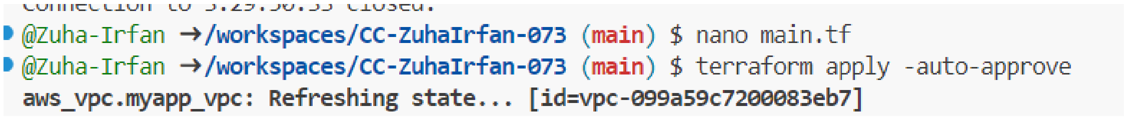
```
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

aws_instance_public_ip = "3.29.50.33"
```

- task9_ssh_into_ec2.png

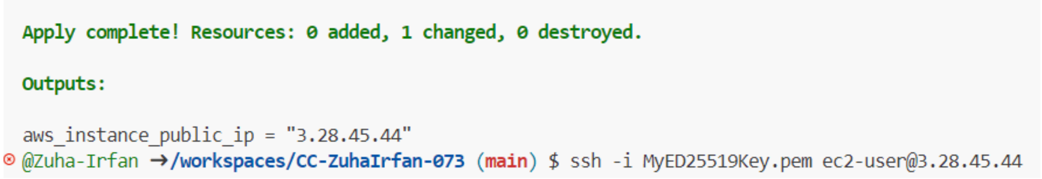


```
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ ssh -i MyED25519Key.pem ec2-user@3.29.50.33
  The authenticity of host '3.29.50.33 (3.29.50.33)' can't be established.
  ED25519 key fingerprint is SHA256:dC1hsol7z0DYaUssBw9GtJKsjuKX1XXFSgHb+DiyWD0.
  This key is not known by any other names.
  Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
  Warning: Permanently added '3.29.50.33' (ED25519) to the list of known hosts.
     ,     #_
     ~\_  ####_        Amazon Linux 2023
    ~~  \_#####\
    ~~     \###|
    ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
     ~~       V~' '->
      ~~~         /
        ~~._.   _/
           _/ _/
         _/m/'
  [ec2-user@ip-10-0-1-127 ~]$ whoami
  ec2-user
  [ec2-user@ip-10-0-1-127 ~]$ exit
  logout
  Connection to 3.29.50.33 closed.
```

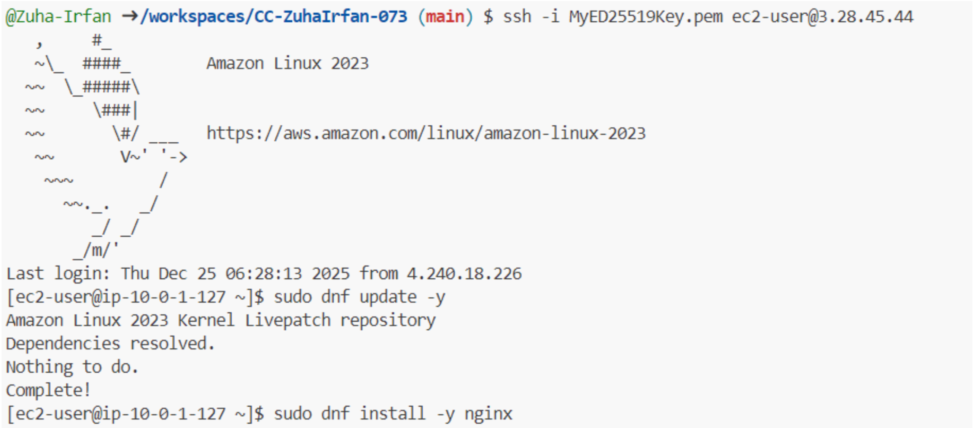
- task9_ssh_keypair_and_ssh.png



```
Connection to 3.29.50.53 closed.
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073 (main) $ nano main.tf
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073 (main) $ terraform apply -auto-approve
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-099a59c7200083eb7]
```

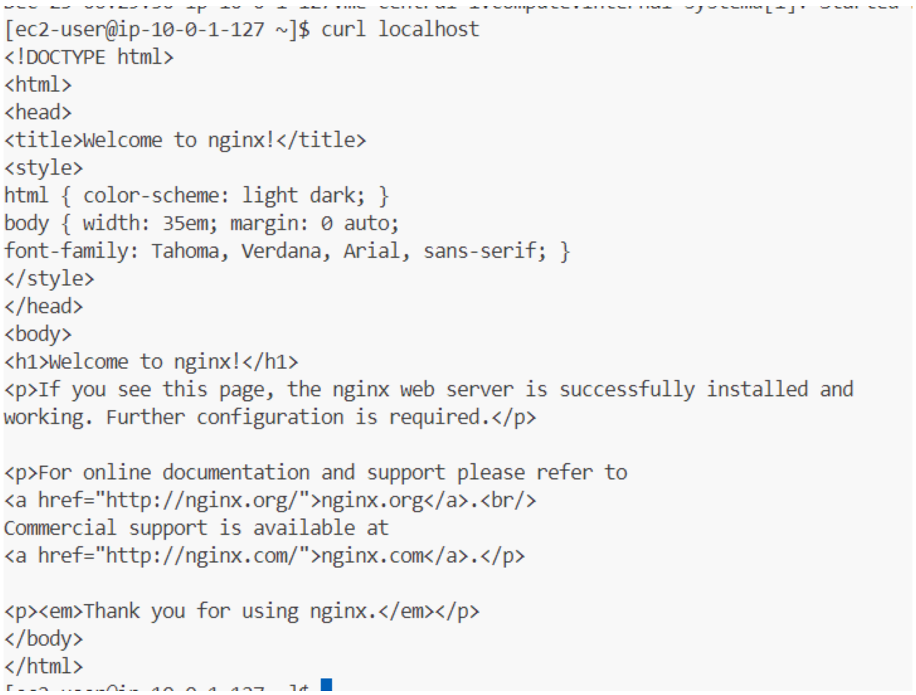


```
Apply complete! Resources: 0 added, 1 changed, 0 destroyed.

Outputs:

aws_instance_public_ip = "3.28.45.44"
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073 (main) $ ssh -i MyED25519Key.pem ec2-user@3.28.45.44
```



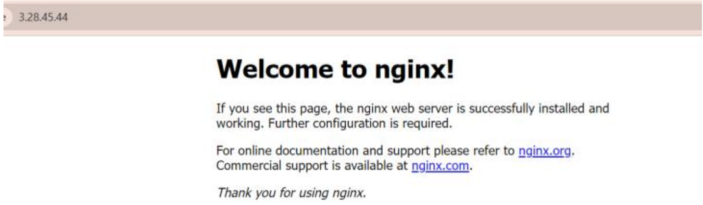
```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073 (main) $ ssh -i MyED25519Key.pem ec2-user@3.28.45.44
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Thu Dec 25 06:28:13 2025 from 4.240.18.226
[ec2-user@ip-10-0-1-127 ~]$ sudo dnf update -y
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-1-127 ~]$ sudo dnf install -y nginx
```

- task9_nginx_local_curl.png



```
Dec 25 06:29:50 ip-10-0-1-127.me-central-1.compute.internal systemd[1]: Started
[ec2-user@ip-10-0-1-127 ~]$ 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-10-0-1-127 ~]$
```

- task9_nginx_browser_page.png


3.28.45.44

Welcome to nginx!

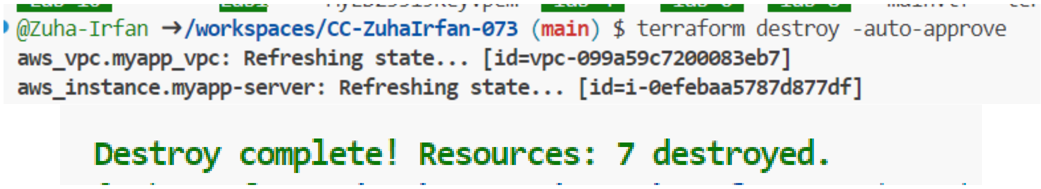
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

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

Thank you for using nginx.

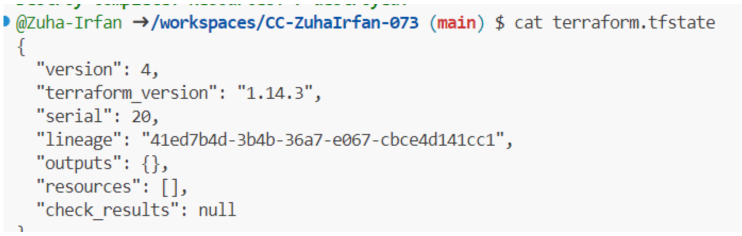
Cleanup

- cleanup_destroy.png

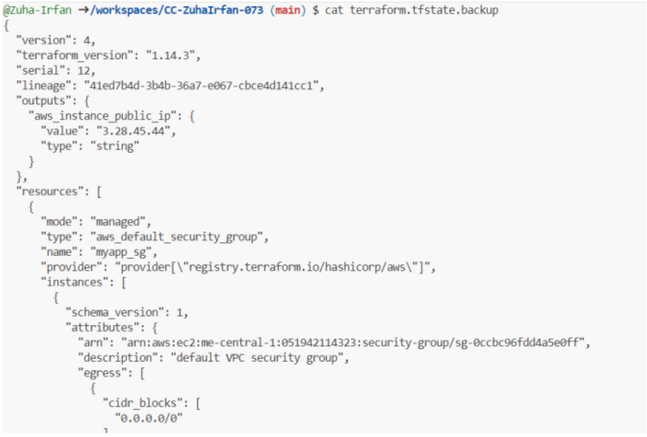


```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ terraform destroy -auto-approve
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-099a59c7200083eb7]
aws_instance.myapp-server: Refreshing state... [id=i-0efebaa5787d877df]

    Destroy complete! Resources: 7 destroyed.
```

- cleanup_state_files.png

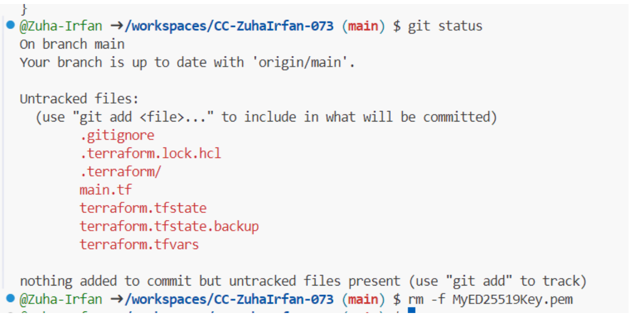


```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 20,
  "lineage": "41ed7b4d-3b4b-36a7-e067-cbce4d141cc1",
  "outputs": {},
  "resources": [],
  "check_results": null
```



```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 12,
  "lineage": "41ed7b4d-3b4b-36a7-e067-cbce4d141cc1",
  "outputs": {
    "aws_instance_public_ip": {
      "value": "3.28.45.44",
      "type": "string"
    }
  },
  "resources": [
    {
      "mode": "managed",
      "type": "aws_default_security_group",
      "name": "myapp_sg",
      "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",
      "instances": [
        {
          "schema_version": 1,
          "attributes": {
            "arn": "arn:aws:ec2:me-central-1:051942114323:security-group/sg-0ccbc96fdd4a5e0ff",
            "description": "default VPC security group",
            "egress": [
              {
                "cidr_blocks": [
                  "0.0.0.0/0"
```

- cleanup_verify_no_secrets.png



```
  }
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (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
          .terraform/
          main.tf
          terraform.tfstate
          terraform.tfstate.backup
          terraform.tfvars

  nothing added to commit but untracked files present (use "git add" to track)
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ rm -f MyED25519Key.pem
```