Cloud Computing Lab

Nayab Khazin

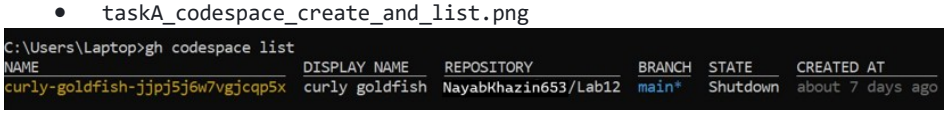
BSE-V B

2023-BSE-046

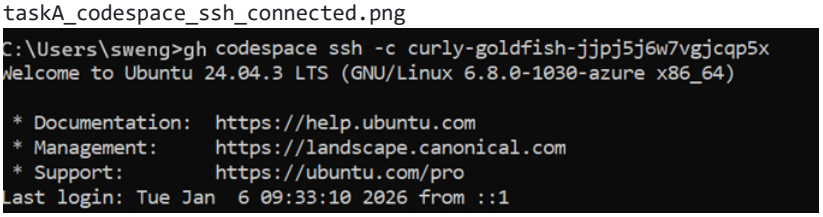
LAB 11

Task 0 Lab Setup (Codespace & GH CLI)

- taskA_codespace_create_and_list.png

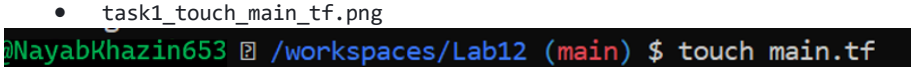


- taskA_codespace_ssh_connected.png

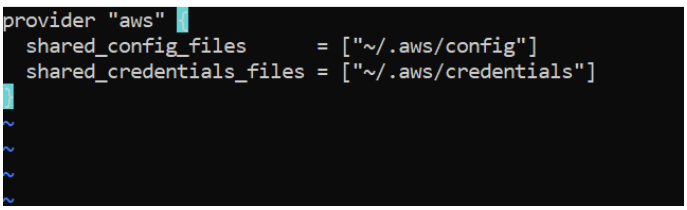


Task 1 — Provider & Basic variable (variable precedence)

- task1_touch_main_tf.png



- task1_main_tf_provider.png



- task1_terraform_init.png

```
@NayabKhazin653 /workspaces/Lab12 (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 made some changes to the provider dependency selections recorded
in the .terraform.lock.hcl file. Review those changes and commit them to your
version control system if they represent changes you intended to make.

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

```
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.png

```
@NayabKhazin653 /workspaces/Lab12 (main) $ terraform apply -auto-approve
var.subnet_cidr_block
  Enter a value: 0.0.0.0/0


Changes to Outputs:
  + subnet_cidr_block_output = "0.0.0.0/0"

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 = "0.0.0.0/0"
```

- task1_apply_with_default.png

```
@NayabKhazin653 /workspaces/Lab12 (main) $ terraform apply -auto-approve

hanges to Outputs:
 ~ subnet_cidr_block_output = "0.0.0.0/0" -> "10.0.0.0/24"

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

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

utputs:

ubnet_cidr_block_output = "10.0.0.0/24"
```

- task1_env_var_set_and_apply.png

```
@NayabKhazin653 /workspaces/Lab12 (main) $ export TF_VAR_subnet_cidr_block=10.0.20.0/24
@NayabKhazin653 /workspaces/Lab12 (main) $ terraform apply -auto-approve

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

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

```
NayabKhazin653 /workspaces/Lab12 (main) $ cat terraform.tfvars
ubnet_cidr_block = "10.0.30.0/24"
NayabKhazin653 /workspaces/Lab12 (main) $ terraform apply -auto-approve

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

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

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

utputs:

ubnet_cidr_block_output = "10.0.30.0/24"
```

- task1_var_override_with_dash_var.png

```
NayabKhazin653 /workspaces/Lab12 (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.png

```
NayabKhazin653 /workspaces/Lab12 (main) $ printenv | grep TF_VAR_
TF_VAR_subnet_cidr_block=10.0.20.0/24
NayabKhazin653 /workspaces/Lab12 (main) $ unset TF_VAR_subnet_cidr_block
NayabKhazin653 /workspaces/Lab12 (main) $ printenv | grep TF_VAR_
```

Task 2 — Variable validation & sensitive / ephemeral variables

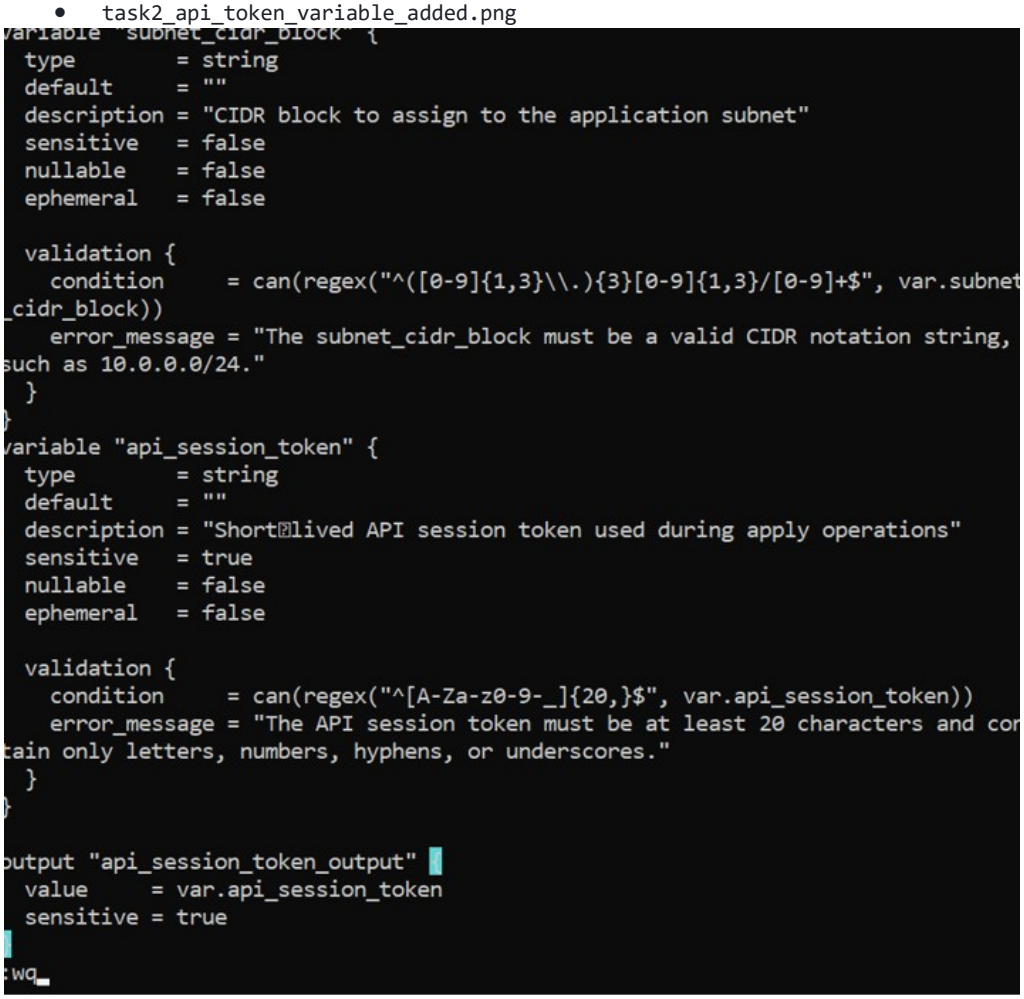
- task2_subnet_variable_with_validation.png

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

- task2_subnet_validation_error.png

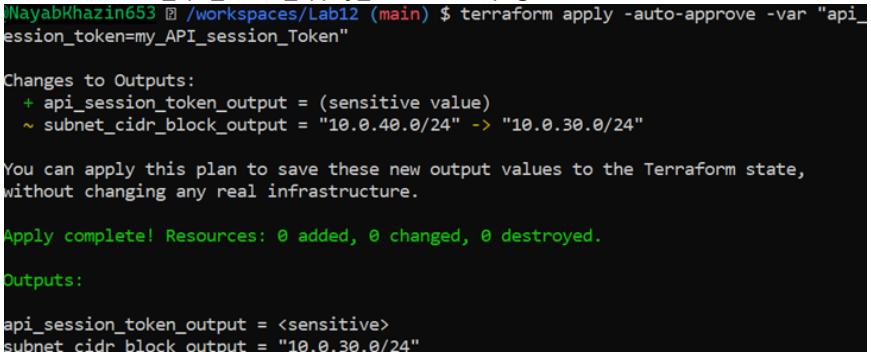
```
NayabKhazin653 /workspaces/Lab12 (main) $ terraform apply -auto-approve -var "
subnet_cidr_block=10.0.0"

Error: Unsupported argument

  on main.tf line 14, in variable "subnet_cidr_block":
  14:   ephemeral   = false

An argument named "ephemeral" is not expected here.
```

- task2_api_token_variable_added.png



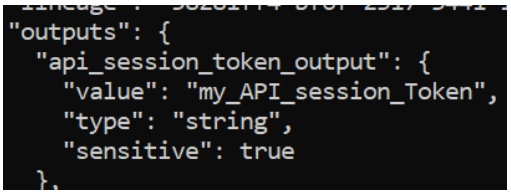
```
variable "subnet_cidr_block" {
  type        = string
  default     = ""
  description = "CIDR block to assign to the application subnet"
  sensitive   = false
  nullable    = false
  ephemeral   = false

  validation {
    condition     = can(regex("^([0-9]{1,3}\\.){3}[0-9]{1,3}/[0-9]+$", var.subnet
_cidr_block))
    error_message = "The subnet_cidr_block must be a valid CIDR notation string,
such as 10.0.0.0/24."
  }
}
variable "api_session_token" {
  type        = string
  default     = ""
  description = "Short-lived API session token used during apply operations"
  sensitive   = true
  nullable    = false
  ephemeral   = false

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

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

- task2_api_token_apply_sensitive.png



```
@NayabKhazin653 /workspaces/Lab12 (main) $ terraform apply -auto-approve -var "api_
session_token=my_API_session_Token"

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

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

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

Outputs:

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

- task2_check_terraform_state_api_token.png



```
"outputs": {
  "api_session_token_output": {
    "value": "my_API_session_Token",
    "type": "string",
    "sensitive": true
  },
```

- task2_api_token_ephemeral_error.png

```
Error: Invalid value for variable

  on main.tf line 8:
   8: 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:15,3-13.


Error: Invalid value for variable

  on main.tf line 20:
  20: variable "api_session_token" {
     ├────────────────
     │ var.api_session_token is ""

The API session token must be at least 20 characters and contain only letters,
numbers, hyphens, or underscores.

This was checked by the validation rule at main.tf:28,3-13.
```

- task2_api_token_default_apply.png

```
no differences, so no changes are needed.

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

Outputs:

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

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

- task3_variables_added.png

```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
variable "environment" {}
variable "project_name" {}
variable "primary_subnet_id" {}
variable "subnet_count" {}
variable "monitoring" {}
```

- task3_terraform_tfvars_populated.png

```
NayabKhazin653 /workspaces/Lab12 (main) $ aws ec2 describe-subnets \
   --filters "Name=availability-zone,Values=me-central-1a" \
   --query "Subnets[].SubnetId" \
   --output text
ubnet-0fb24971d181efddb
NayabKhazin653 /workspaces/Lab12 (main) $ vim terraform.tfvars
NayabKhazin653 /workspaces/Lab12 (main) $ @NayabKhazin653 /workspaces/Lab12 (mai
 $ cat terraform.tfvars
ubnet_cidr_block = "10.0.30.0/24"
nvironment = "dev"
roject_name = "lab_work"
rimary_subnet_id = "subnet-0fb24971d181efddb"
ubnet_count = 3
onitoring = true
```

- task3_locals_tf_created.png

```
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

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

Outputs:

is_production = false
monitoring_enabled = true
primary_public_subnet = "subnet-0fb24971d181efddb"
resource_name = "lab_work-dev"
subnet_count = 3
```

Task 4 — Maps and Objects

- task4_tags_variable_added.png

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

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

- task4_tags_output.png

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

Outputs:

is_production = false
monitoring_enabled = true
primary_public_subnet = "subnet-0fb24971d181efddb"
resource_name = "lab_work-dev"
subnet_count = 3
tags = tomap({
  "Environment" = "dev"
  "Owner" = "platform-team"
  "Project" = "sample-app"
})
```

- task4_server_config_output.png

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

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

Outputs:

is_production = false
monitoring_enabled = true
primary_public_subnet = "subnet-0fb24971d181efddb"
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"
  "Project" = "lab_work"
})
```

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

- task5_collections_defined.png

```
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
  }
}
"main.tf" 59L, 1435B
```

- task5_compare_collections.png

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

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,
  ]
}
is_production = false
monitoring_enabled = true
primary_public_subnet = "subnet-0fb24971d181efddb"
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"
  "Project" = "lab_work"
})
```

- task5_locals_mutations.png

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

- task5_mutation_comparison.png

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

Task 6 — Null, any type & dynamic values

- task6_optional_tag_variable.png

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

- task6_locals_merge.png

```
  mutated_set   = setunion(var.availability_zones, ["me-central-1c"])
  server_tags = merge(
    { Name = "web-server" },
    var.optional_tag != null ? { Custom = var.optional_tag } : {}
  )
}
```

- task6_optional_tag_no_value.png

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

- task6_optional_tag_with_value.png

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

- task6_dynamic_value_string.png

```
  "Project" = "lab_work"
})
value_received = "hello"
```

- task6_dynamic_value_number.png

```
})
value_received = 42
```

- task6_dynamic_value_list.png

```
})
value_received = [
  "a",
  "b",
  "c",
]
```

- task6_dynamic_value_map.png

```
})
value_received = {
  "cpu" = 4
  "name" = "server"
}
```

- task6_dynamic_value_null.png(previous outputs only)

```
subnet_count = 3
tags = tomap({
  "Environment" = "dev"
  "Project" = "lab_work"
})
```

Task 7 — Git ignore

- task7_gitignore_created.png

```
.terraform/*
*.tfstate
*.tfstate.*
*.tfvars
*.pem
~
~
~
```

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

- task8_clean_files.png

```
@NayabKhazin653 /workspaces/Lab12 (main) $ cat main.tf
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
@NayabKhazin653 /workspaces/Lab12 (main) $ cat locals.tf

@NayabKhazin653 /workspaces/Lab12 (main) $ cat terraform.tfvars
```

- task8_variables_recreated.png

```
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" {}
```

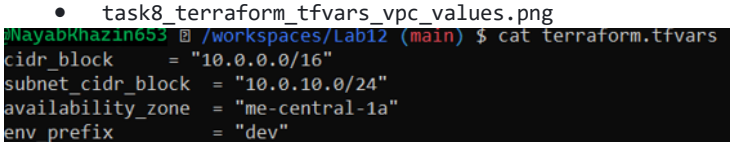
- task8_vpc_resources_added.png

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

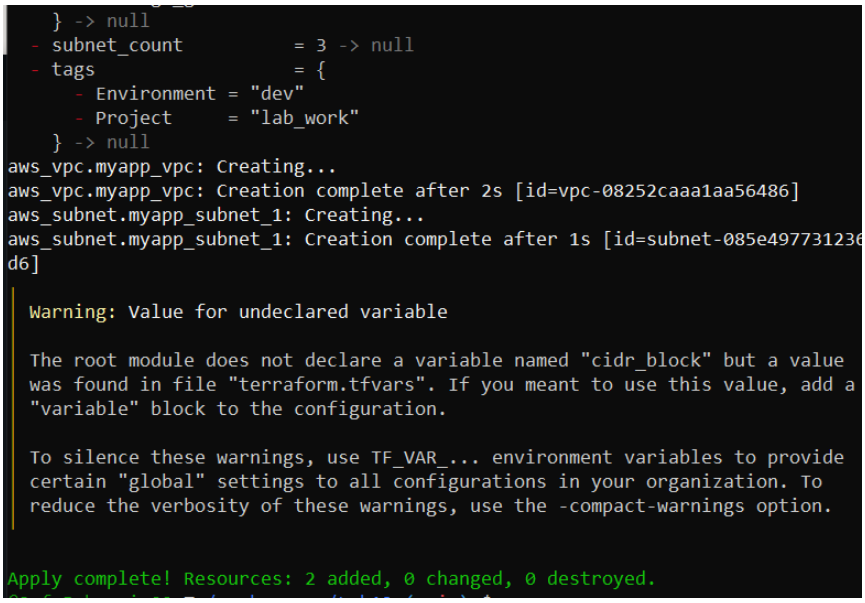
- task8_subnet_resources_added.png

```
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"
  }
}
```

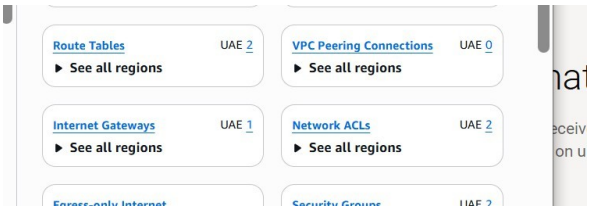
- task8_terraform_tfvars_vpc_values.png



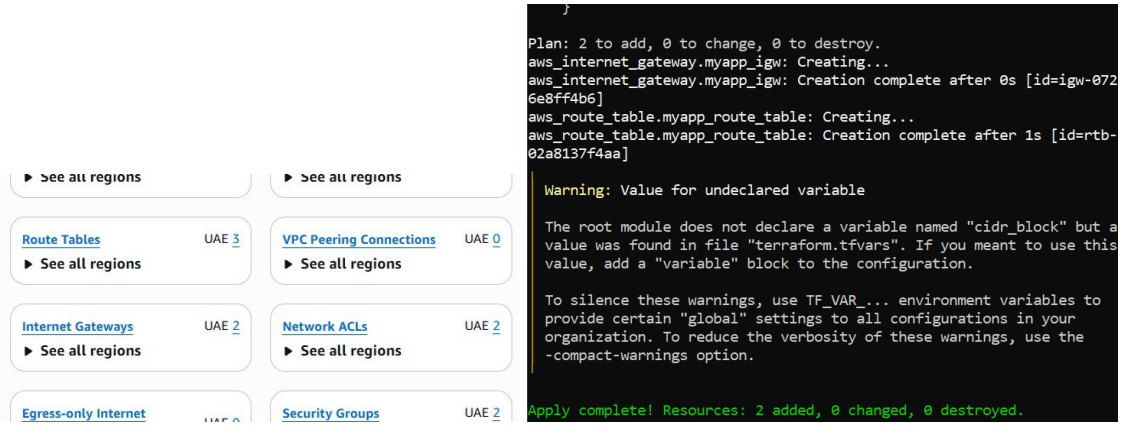
- task8_vpc_subnet_apply.png



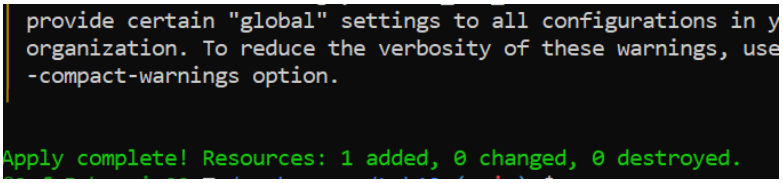
- task8_igw_route_table_before_apply.png



- task8_igw_route_table_after_apply.png



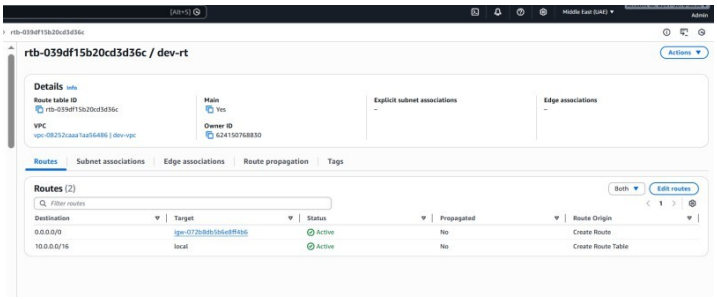
- task8_association_apply.png



- task8_default_route_table.png

```
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"
  }
}
~
~
~
~
:wq
```

- task8_default_route_table_apply.png



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

- task9_my_ip_variable_added.png

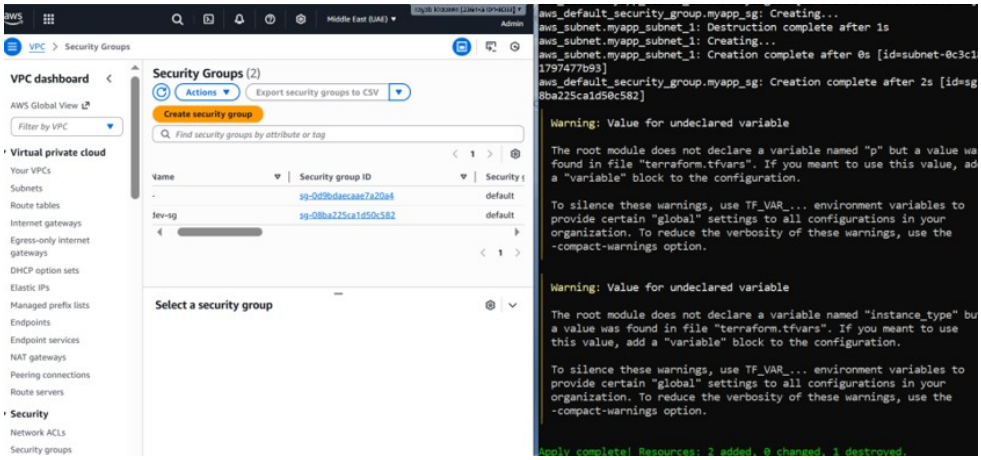
```
}
variable "my_ip" {}
```

- task9_public_ip_curl.png

```
@NayabKhazin653 /workspaces/Lab12 (main) $ curl icanhazip.com
4.240.18.227
@NayabKhazin653 /workspaces/Lab12 (main) $ vim terraform.tfvars
@NayabKhazin653 /workspaces/Lab12 (main) $ @NayabKhazin653 /wor
es/Lab12 (main) $ vim terraform.tfvars
@NayabKhazin653 /workspaces/Lab12 (main) $ cat terraform.tfvars
p = "4.240.18.227/32"
instance_type = "t3.micro"
availability_zone = "me-central-1a"
env_prefix = "dev"
```

- task9_security_group_apply.png

```
}
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"
  }
}
:wq
```



- task9_keypair_created_and_saved.png

```
NayabKhazin653 /workspaces/Lab12 (main) $ aws ec2 create-key-pair \
> --key-name MyED25519Key \
> --key-type ed25519 \
> --key-format pem \
> --query 'KeyMaterial' \
> --output text > MyED25519Key.pem
NayabKhazin653 /workspaces/Lab12 (main) $ chmod 600 MyED25519Key.pem
NayabKhazin653 /workspaces/Lab12 (main) $ ls -l MyED25519Key.pem && h
ead -n 3 MyED25519Key.pem
-rw------- 1 codespace codespace 388 Jan 11 05:25 MyED25519Key.pem
```

```
@NayabKhazin653 /workspaces/Lab12 (main) $ cat .gitignor
.terraform/*
*.tfstate
*.tfstate.*
*.tfvars
*.pem
```

- task9_instance_type_set.png

```
}

resource "aws_instance" "myapp-server" {
  ami                         = "ami-05524d6658fcf35b6" # Amazon Linux 2
023
  instance_type               = var.instance_type
  subnet_id                   = aws_subnet.myapp_subnet_1.id
  security_groups             = [aws_default_security_group.default_sg.i
d]
  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
}
:wq
```

- task9_ec2_apply_and_public_ip.png

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

Changes to Outputs:
  + aws_instance_public_ip = (known after apply)
aws_default_security_group.default_sg: Creating...
aws_default_security_group.default_sg: Creation complete after 3s [
-08ba225ca1d50c582]
aws_instance.myapp-server: Creating...
aws_instance.myapp-server: Still creating... [10s elapsed]
aws_instance.myapp-server: Creation complete after 13s [id=i-092574
a5e362]

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

Outputs:

aws_instance_public_ip = "3.29.123.61"
```

```
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... [10s elapsed]
aws_instance.myapp-server: Creation complete after 13s [id=i-0aa92f3b127
08c624]

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

Outputs:

aws_instance_public_ip = "3.29.123.215"
```

- task9_ssh_into_ec2.png

```
@NayabKhazin653 /workspaces/Lab12 (main) $ ssh -i MyED25519Key.pem ec2
-user@3.29.123.61
The authenticity of host '3.29.123.61 (3.29.123.61)' can't be establishe
d.
ED25519 key fingerprint is SHA256:PkD5Q4sR+XS5iWx26VmfGNfIDijeA7QYu8JLoj
DBXJI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.29.123.61' (ED25519) to the list of known
hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-118-31 ~]$ exit
logout
Connection to 3.29.123.61 closed.
```

- task9_ssh_keypair_and_ssh.png

```
@NayabKhazin653 /workspaces/Lab12 (main) $ ssh-keygen -t ed25519 -f ~/
.ssh/id_ed25519 -N ""
Generating public/private ed25519 key pair.
/home/codespace/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
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:H4M4AV/58lIBIXRSHnO43PsJRPMSN1A3HvDBiLMvOQ8 codespace@codespaces-
6c3a52
The key's randomart image is:
+--[ED25519 256]--+
|    ..+.O=o+o+=  |
|     o *o+B ++.+ |
|      o..= O .o  |
|       o+.B .    |
|      o S=o=     |
|       ...Eo.    |
|         ..B .   |
|            +    |
|                 |
+----[SHA256]-----+
```

```
  associate_public_ip_address = true
  key_name                    = key_name = aws_key_pair.ssh_key.key_nam

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

output "aws_instance_public_ip" {
  value = aws_instance.myapp-server.public_ip
}
resource "aws_default_security_group" "default_sg" {
  vpc_id = aws_vpc.myapp_vpc.id

  # Allow SSH (Port 22)
  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  # Allow all outbound traffic
  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
resource "aws_key_pair" "ssh_key" {
  key_name   = "serverkey"
  public_key = file("~/.ssh/id_ed25519.pub")
}
```

```
NayabKhazin653  /workspaces/Lab12 (main) $ ssh ec2-user@3.29.123.215
The authenticity of host '3.29.123.215 (3.29.123.215)' can't be establis
hed.
ED25519 key fingerprint is SHA256:cPzMEMrdW62K7maP1ps5kunq/2ZqXrdCBWAodx
YX6HQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.29.123.215' (ED25519) to the list of known
 hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-3-192 ~]$
```

- task9_nginx_local_curl.png

```
  ,     #_
  ~\_  ####_        Amazon Linux 2023
 ~~  \_#####\
 ~~     \###|
 ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
  ~~       V~' '->
   ~~~         /
     ~~._.   _/
        _/ _/
      _/m/'
[ec2-user@ip-10-0-171-16 ~]$ 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-10-0-171-16 ~]$
```

```
@NayabKhazin653 /workspaces/Lab12 (main) $ cat > entry-script.sh <<'EOF'
#!/bin/bash
yum update -y
yum install -y nginx
systemctl start nginx
systemctl enable nginx
EOF
```

- task9_nginx_browser_page.png

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

```
aws_instance.myapp-server: Destruction complete after 51s
aws_subnet.myapp_subnet_1: Destroying... [id=subnet-0c3c18a1797477b93]
aws_default_security_group.default_sg: Destroying... [id=sg-08ba225ca1d50c582]
aws_default_security_group.default_sg: Destruction complete after 0s
aws_key_pair.ssh_key: Destroying... [id=serverkey]
aws_key_pair.ssh_key: Destruction complete after 0s
aws_subnet.myapp_subnet_1: Destruction complete after 0s
aws_vpc.myapp_vpc: Destroying... [id=vpc-08252caaa1aa56486]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.
```

- cleanup_state_files.png

```
@NayabKhazin653 /workspaces/Lab12 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.9.5",
  "serial": 152,
  "lineage": "58281ff4-bf6f-2317-3441-125f5206a5a3",
  "outputs": {},
  "resources": [],
  "check_results": null
}
@NayabKhazin653 /workspaces/Lab12 (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.9.5",
  "serial": 144,
  "lineage": "58281ff4-bf6f-2317-3441-125f5206a5a3",
  "outputs": {
    "aws_instance_public_ip": {
      "value": "158.252.72.57",
      "type": "string"
    }
  },
  "resources": [
```

- cleanup_verify_no_secrets.png

```
@NayabKhazin653 /workspaces/Lab12 (main) $ cat .gitignore
.terraform/*
*.tfstate
*.tfstate.*
*.tfvars
*.pem
@NayabKhazin653 /workspaces/Lab12 (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.tfvars.swo
        .terraform.tfvars.swp
        :wq
        LICENSE.txt
        aws/
        awscliv2.zip
        entry-script.sh
        locals.tf
        locals.tf.backup
        main.tf
        terraform.tfvars.backup
        terraform_1.9.5_linux_amd64.zip

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