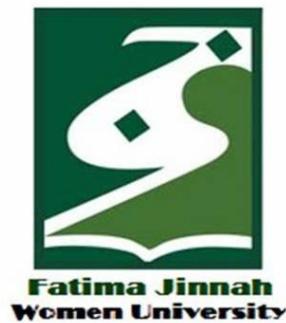


CLOUD COMPUTING



SUBMITTED TO
SIR WAQAS SALEEM & SIR SHOAIB

SUBMITTED BY
HAMNA MAHMOOD

2023-BSE-025

BSE V-A

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

Task 0 Lab Setup (Codespace & GH CLI)

- taskA_codespace_create_and_list.png

```
C:\Users\ABC\Downloads\gh_2.83.2_windows_amd64\bin>gh codespace list
NAME          DISPLAY NAME      REPOSITORY           BRANCH   STATE    CREATED AT
studious-journey-q7x57x9ww649...  studious journey  HamnaMahmood20/CC-Hamna-Mahmo...  main*   Shutdown  about 6 days ago
```

- taskA_codespace_ssh_connected.png

```
C:\Users\ABC\Downloads\gh_2.83.2_windows_amd64\bin>gh codespace ssh -c studious-journey-q7x57x9ww649hx57g
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
Last login: Tue Jan  6 17:07:52 2026 from ::1
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
Last login: Tue Jan  6 17:07:52 2026 from ::1
@hamna-mahmood ~ /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $
```

Task 1- Provider & Basic variable

- task1_touch_main_tf.png

```
@hamna-mahmood ~ /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ touch main.tf
```

- task1_main_tf_provider.png

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

- task1_terraform_init.png

```
@hamna-mahmood ~ /workspaces/CC-Hamna-Mahmood-25-BSE-VA (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.
```

- task1_variable_and_output_added.png

```
variable "subnet_cidr_block" {
  type = string
}

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

- task1_apply_prompt_for_var.png

```
@hamna-mahmood ✭ /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ terraform apply -auto-approve
subnet_cidr_blockvar.subnet_cidr_block
Enter a value:
```

- task1_apply_with_default.png

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

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

Outputs:

dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-region = "me-central-1a"
dev-subnet-tags_all = tomap({
  "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-region = "me-central-1"
dev-vpc-tags_all = tomap({
  "Name" = "development"
})
dev-vpc-tags_name = "development"
subnet_cidr_block_output = "10.0.0.0/24"
```

- task1_env_var_set_and_apply.png

```
@hamna-mahmood ✭ /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ export TF_VAR_subnet_cidr_block=10.0.20.0/24
@hamna-mahmood ✭ /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ terraform apply -auto-approve
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-00e2aa2a7430fc24c]
data.aws_vpc.existing_vpc: Read complete after 0s [id=vpc-0b412746b28b797e7]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-01f96be1fa93f203d]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0cf0259a916972e54]

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:

dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-region = "me-central-1a"
dev-subnet-tags_all = tomap({
  "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-region = "me-central-1"
dev-vpc-tags_all = tomap({
  "Name" = "development"
})
dev-vpc-tags_name = "development"
subnet_cidr_block_output = "10.0.20.0/24"
```

- task1_terraform_tfvars_and_apply.png

```
@hamna-mahmood 7 /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ cat terraform.tfvars
subnet_cidr_block = "10.0.30.0/24"
@hamna-mahmood 7 /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ terraform apply -auto-approve
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-00e2aa2a7430fc24c]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0b412746b28b797e7]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-01f96be1fa93f203d]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0cf0259a916972e54]

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:

dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-region = "me-central-1a"
dev-subnet-tags_all = tomap({
  "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-region = "me-central-1"
dev-vpc-tags_all = tomap({
  "Name" = "development"
})
dev-vpc-tags_name = "development"
subnet_cidr_block_output = "10.0.30.0/24"
```

- task1_var_override_with_dash_var.png

```
@hamna-mahmood 7 /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ terraform apply -auto-approve -var "subnet_cidr_block=10.0.40.0/24"
# -var is hdata.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-00e2aa2a7430fc24c]
data.aws_vpc.existing_vpc: Read complete after 0s [id=vpc-0b412746b28b797e7]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-01f96be1fa93f203d]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0cf0259a916972e54]

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:

dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-region = "me-central-1a"
dev-subnet-tags_all = tomap({
  "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-region = "me-central-1"
dev-vpc-tags_all = tomap({
  "Name" = "development"
})
dev-vpc-tags_name = "development"
subnet_cidr_block_output = "10.0.40.0/24"
```

- task1_printenv_tf_var_and_unset.png

```
@hamna-mahmood 7 /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ printenv | grep TF_VAR_
TF_VAR_subnet_cidr_block=10.0.20.0/24
@hamna-mahmood 7 /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ unset TF_VAR_subnet_cidr_block
@hamna-mahmood 7 /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ printenv | grep TF_VAR_
@hamna-mahmood 7 /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $
```

Task 2- Variable validation & sensitive/ephemeral variables

- task2_subnet_variable_with_validation.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."  
    }  
}
```

- task2_subnet_validation_error.png

```
@hamna-mahmood eworkspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ terraform apply -auto-approve -var "subnet_cidr_block=1  
0.0.0"  
rdata.aws_vpc.existing_vpc: Reading...  
aws_vpc.development_vpc: Refreshing state... [id=vpc-00e2aa2a7430fc24c]  
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0b412746b28b797e7]  
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-01f96be1fa93f203d]  
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0cf0259a916972e54]  
  
Planning failed. Terraform encountered an error while generating this plan.  
  
Error: Invalid value for variable  
  
on main.tf line 58:  
58: 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:66,3-13.
```

- 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, hyphens, or underscores."  
    }  
}  
  
output "api_session_token_output" {  
    value      = var.api_session_token  
    sensitive  = true
```

- task2_api_token_apply_sensitive.png

```

@hamna-mahmood ② /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ terraform apply -auto-approve -var "api_session_token=my_API_session_Token"
  waiting for outputs...
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-00e2aa2a7430fc24c]
data.aws_vpc.existing_vpc: Read complete after 0s [id=vpc-0b412746b28b797e7]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-01f96be1fa93f203d]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0cf0259a916972e54]

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>
dev_subnet-cidr_block = "10.0.10.0/24"
dev-subnet-region = "me-central-1a"
dev-subnet-tags_all = tomap({
  "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-region = "me-central-1"
dev-vpc-tags_all = tomap({
  "Name" = "development"
})
dev-vpc-tags_name = "development"
subnet_cidr_block_output = "10.0.30.0/24"

```

- [task2_check_terraform_state_api_token.png](#)

```

"api_session_token_output": {
  "value": "my_API_session_Token",
  "type": "string",
  "sensitive": true
},
"subnet_cidr_block_output": "10.0.30.0/24"

```

- [task2_api_token_ephemeral_error.png](#)

```

variable "api_session_token" {
  ephemeral=true
}

@hamna-mahmood ② /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ terraform apply
Error: Error acquiring the state lock

Error message: 2 problems:

- Unsupported state file format: The state file could not be parsed as JSON: syntax error at byte offset 264.
- Unsupported state file format: The state file does not have a "version" attribute, which is required to identify the format version.

Terraform acquires a state lock to protect the state from being written by multiple users at the same time. Please resolve the issue above and try again. For most commands, you can disable locking with the "-lock=false" flag, but this is not recommended.

```

- [task2_api_token_default_apply.png](#)

```

hamna-mahmood ~ /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ terraform apply -auto-approve
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-00e2aa2a7430fc24c]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0b412746b28b797e7]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-01f96be1fa93f203d]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0cf0259a916972e54]

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>
dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-region = "me-central-1a"
dev-subnet-tags_all = tomap({
  "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-region = "me-central-1"
dev-vpc-tags_all = tomap({
  "Name" = "development"
})
dev-vpc-tags_name = "development"
subnet_cidr_block_output = "10.0.30.0/24"

```

Task3- Project level variables, locals & outputs

- task3_variables_added.png

```

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

- task3_terraform_tfvars_populated.png

```

environment="dev"
project_name="lab_work"
primary_subnet_id="subnet-01f96be1fa93f203d "
subnet_count=3
monitoring=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:

api_session_token_output = <sensitive>
dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-region = "me-central-1a"
dev-subnet-tags_all = tomap({
    "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-region = "me-central-1"
dev-vpc-tags_all = tomap({
    "Name" = "development"
})
dev-vpc-tags_name = "development"
is_production = false
monitoring_enabled = true
primary_public_subnet = "subnet-01f96be1fa93f203d "
resource_name = "lab_work-dev"
subnet_cidr_block_output = "10.0.30.0/24"
subnet_count = 3
```

Task 4 — Maps and Objects

- task4_tags_variable_added.png

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

output "tags" [
  value = var.tags
]
```

- task4_tags_output.png

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

Outputs:

api_session_token_output = <sensitive>
dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-region = "me-central-1a"
dev-subnet-tags_all = tomap({
    "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-region = "me-central-1"
dev-vpc-tags_all = tomap({
    "Name" = "development"
})
dev-vpc-tags_name = "development"
is_production = false
monitoring_enabled = true
primary_public_subnet = "subnet-01f96be1fa93f203d "
resource_name = "lab_work-dev"
subnet_cidr_block_output = "10.0.30.0/24"
subnet_count = 3
tags = tomap({
    "Environment" = "dev"
    "Owner" = "platform-team"
    "Project" = "sample-app"
})
```

- task4_server_config_output.png

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

Outputs:

api_session_token_output = <sensitive>
dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-region = "me-central-1a"
dev-subnet-tags_all = tomap({
    "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-region = "me-central-1"
dev-vpc-tags_all = tomap({
    "Name" = "development"
})
dev-vpc-tags_name = "development"
is_production = false
monitoring_enabled = true
primary_public_subnet = "subnet-01f96be1fa93f203d"
resource_name = "lab_work-dev"
server_config = {
    "backup_enabled" = false
    "instance_type" = "t3.micro"
    "monitoring" = true
    "name" = "web-server"
    "storage_gb" = 20
}
subnet_cidr_block_output = "10.0.30.0/24"
subnet_count = 3
tags = tomap({
    "Environment" = "dev"
    "Owner" = "platform-team"
    "Project" = "sample-app"
})
```

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

- 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
  }
}
```

- task5_compare_collections.png

```

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,
  ]
}
dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-region = "me-central-1a"
dev-subnet-tags_all = tomap({
  "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-region = "me-central-1"
dev-vpc-tags_all = tomap({
  "Name" = "development"
})
dev-vpc-tags_name = "development"
is_production = false
monitoring_enabled = true
primary_public_subnet = "subnet-01f96be1fa93f203d"
resource_name = "lab_work-dev"
server_config = {
  "backup_enabled" = false
  "instance_type" = "t3.micro"
}

```

- task5_locals_mutations.png

```

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

```

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,
    ]
}
dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-region = "me-central-1a"
dev-subnet-tags_all = tomap({
    "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-region = "me-central-1"
dev-vpc-tags_all = tomap({
    "Name" = "development"
})
dev-vpc-tags_name = "development"
is_production = false
monitoring_enabled = true
mutation_comparison = {
    "mutated_tuple" = toset([
        "4",
        "true",
        "web-1",
        "web-2",
    ])
    "original_tuple" = [
        "web-1",
        4,
        true,
    ]
}
primary_public_subnet = "subnet-01f96be1fa93f203d"
resource_name = "lab_work-dev"
server_config = {
    "backup_enabled" = false
    "instance_type" = "t3.micro"
    "monitoring" = true
    "name" = "web-server"
    "storage_gb" = 20
}
subnet_cidr_block_output = "10.0.30.0/24"
subnet_count = 3
tags = tomap({
    "Environment" = "dev"
    "Owner" = "platform-team"
    "Project" = "sample-app"
})

```

Task 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
]

```

- task6_locals_merge.png

```

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

```

- task6_optional_tag_no_value.png

```

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,
    ]
}
dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-region = "me-central-1a"
dev-subnet-tags_all = tomap({
    "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-region = "me-central-1"
dev-vpc-tags_all = tomap({
    "Name" = "development"
})
dev-vpc-tags_name = "development"
is_production = false

```

- task6_optional_tag_with_value.png

```

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,
    ]
}
dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-region = "me-central-1a"
dev-subnet-tags_all = tomap({
    "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-region = "me-central-1"
dev-vpc-tags_all = tomap({
    "Name" = "development"
})
dev-vpc-tags_name = "development"
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-01f96be1fa93f203d"
resource_name = "lab_work-dev"
server_config = {
    "backup_enabled" = false
    "instance_type" = "t3.micro"
    "monitoring" = true
    "name" = "web-server"
    "storage_gb" = 20
}
subnet_cidr_block_output = "10.0.30.0/24"
subnet_count = 3
tags = tomap({
    "Environment" = "dev"
    "Owner" = "platform-team"
    "Project" = "sample-app"
})

```

- task6_dynamic_value_string.png

```

Changes to Outputs:
+ value_received      = "hello"
You can apply this plan to save these new output values to []
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,
  ]
}
dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-region = "me-central-1a"
dev-subnet-tags_all = tomap({
  "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"

```

```

dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-region = "me-central-1"
dev-vpc-tags_all = tomap({
  "Name" = "development"
})
dev-vpc-tags_name = "development"
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-01f96be1fa93f203d"
resource_name = "lab_work-dev"
server_config = {
  "backup_enabled" = false
  "instance_type" = "t3.micro"
  "monitoring" = true
  "name" = "web-server"
  "storage_gb" = 20
}
subnet_cidr_block_output = "10.0.30.0/24"
subnet_count = 3
tags = tomap({
  "Environment" = "dev"
  "Owner" = "platform-team"
  "Project" = "sample-app"
})
value_received = "hello"

```

- task6_dynamic_value_number.png

```

~ value_received      = "hello" -> 42
You can apply this plan to save these new output values to []
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,
  ]
}
dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-region = "me-central-1a"
dev-subnet-tags_all = tomap({
  "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-region = "me-central-1"

```

```

dev-vpc-tags_all = tomap({
  "Name" = "development"
})
dev-vpc-tags_name = "development"
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-01f96be1fa93f203d"
resource_name = "lab_work-dev"
server_config = {
  "backup_enabled" = false
  "instance_type" = "t3.micro"
  "monitoring" = true
  "name" = "web-server"
  "storage_gb" = 20
}
subnet_cidr_block_output = "10.0.30.0/24"

```

```

subnet_count = 3
tags = tomap({
  "Environment" = "dev"
  "Owner" = "platform-team"
  "Project" = "sample-app"
})
value_received = 42

```

- task6_dynamic_value_list.png

```

Changes to Outputs:
~ value_received      = 42 -> [
  + "a",
  + "b",
  + "c",
]
You can apply this plan to save these new output values to []
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,
  ]
}
dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-region = "me-central-1a"
dev-subnet-tags_all = tomap({
  "Name" = "development"
})
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-region = "me-central-1"

```

```

  "Name" = "development"
)
dev-vpc-tags_name = "development"
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-01f96be1fa93f203d"
resource_name = "lab_work-dev"
server_config = {
  "backup_enabled" = false
  "instance_type" = "t3.micro"
  "monitoring" = true
  "name" = "web-server"
  "storage_gb" = 20
}
subnet_cidr_block_output = "10.0.30.0/24"

```

```

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

```

- task6_dynamic_value_map.png

```
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,
    ]
}
dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-region = "me-central-1a"
dev-subnet-tags_all = tomap({
    "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-region = "me-central-1"
dev-vpc-tags_all = tomap({
    "Name" = "development"
})
dev-vpc-tags_name = "development"
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-01f96be1fa93f203d"
resource_name = "lab_work-dev"
server_config = {
    "backup_enabled" = false
    "instance_type" = "t3.micro"
    "monitoring" = true
    "name" = "web-server"
    "storage_gb" = 20
}
subnet_cidr_block_output = "10.0.30.0/24"
subnet_count = 3
tags = tomap({
    "Environment" = "dev"
    "Owner" = "platform-team"
    "Project" = "sample-app"
})
```

- task6_dynamic_value_null.png

```
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,
    ]
}
dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-region = "me-central-1a"
dev-subnet-tags_all = tomap({
    "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-region = "me-central-1"
dev-vpc-tags_all = tomap({
    "Name" = "development"
})
dev-vpc-tags_name = "development"
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-01f96be1fa93f203d"
resource_name = "lab_work-dev"
server_config = {
    "backup_enabled" = false
    "instance_type" = "t3.micro"
    "monitoring" = true
    "name" = "web-server"
    "storage_gb" = 20
}
subnet_cidr_block_output = "10.0.30.0/24"
subnet_count = 3
tags = tomap({
    "Environment" = "dev"
    "Owner" = "platform-team"
    "Project" = "sample-app"
})
```

Task 7- Git Ignore

- task7_gitignore_created.png

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

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

- task8_clean_files.png

```
@hamna-mahmood ② /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ rm terraform.tfvars
@hamna-mahmood ② /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ rm main.tf
@hamna-mahmood ② /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ rm locals.tf
@hamna-mahmood ② /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ touch main.tf
@hamna-mahmood ② /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ vim main.tf

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

- task8_variables_recreated.png

```
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

```

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

The screenshot shows the AWS VPC console interface. At the top, there's a search bar labeled "Find VPCs by attribute or tag". Below it is a table for "Your VPCs" with three entries:

Name	VPC ID	State	Encryption c...	Encryption control ...	Block Public...	IPv4 CIDR	IPv6 CIDR
dev-vpc	vpc-05aa4ccb3cf969fa1	Available	-	-	Off	10.0.0.0/16	-
prod-vpc	vpc-09822723aed1d2815	Available	-	-	Off	10.0.0.0/16	-
-	vpc-0b412746b28b797e7	Available	-	-	Off	172.31.0.0/16	-

Below this is another table for "Subnets" with five entries:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR
-	subnet-078f1b79825a5fee0	Available	vpc-0b412746b28b797e7	Off	172.31.16.0/20	-
dev-subnet-1	subnet-082ba110498bf841	Available	vpc-05aa4ccb3cf969fa1 dev-vpc	Off	10.0.10.0/24	-
prod-public-subnet	subnet-0a2ce75f0ad691a6	Available	vpc-09822723aed1d2815 pro...	Off	10.0.10.0/24	-
-	subnet-03aec0612a1e2607f	Available	vpc-0b412746b28b797e7	Off	172.31.0.0/20	-
-	subnet-0f3fcff0e6b1899fc	Available	vpc-0b412746b28b797e7	Off	172.31.32.0/20	-

- task8_igw_route_table_before_apply.png

The screenshot shows the AWS VPC console interface. At the top, there's a search bar labeled "Find internet gateways by attribute or tag". Below it is a table for "Internet gateways" with one entry:

Name	Internet gateway ID	State	VPC ID
-	igw-042ad3ff5f6411b8b	Attached	vpc-04a77041d4db59245

Below this is another table for "Route tables" with one entry:

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
-	rtb-09ca838cfa1ea785b	-	-	Yes	vpc-0

- task8_igw_route_table_after_apply.png

The screenshot shows the AWS VPC console interface after changes have been applied. At the top, there's a search bar labeled "Find route tables by attribute or tag". Below it is a table for "Route tables" with five entries:

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Owner ID
-	rtb-000461310bb92433a	-	-	Yes	vpc-0b412746b28b797e7	542622959327
dev-rt	rtb-05c987809e89fcfc3	subnet-082ba110498bcf...	-	No	vpc-05aa4ccb3cf969fa1 dev-vpc	542622959327
-	rtb-083c4d05710186299	-	-	Yes	vpc-09822723aed1d2815 pro...	542622959327
prod-public-rt	rtb-0586238f2c56ea9ac	subnet-0a2ce75f0ad691...	-	No	vpc-09822723aed1d2815 pro...	542622959327
-	rtb-058cc44464c1c6742	-	-	Yes	vpc-05aa4ccb3cf969fa1 dev-vpc	542622959327

Below this is another table for "Internet gateways" with three entries:

Name	Internet gateway ID	State	VPC ID	Owner
prod-igw	igw-0468cfffd1df556b	Attached	vpc-09822723aed1d2815 prod-vpc	542622959327
dev-igw	igw-053d919d43bb235fb	Attached	vpc-05aa4ccb3cf969fa1 dev-vpc	542622959327
-	igw-0c6d9e84143733f82	Attached	vpc-0b412746b28b797e7	542622959327

- task8_association_apply.png

The screenshot shows two tables in the AWS CloudFormation console:

- Subnets (5) Info**: A table with columns: Name, Subnet ID, State, VPC, Block Public..., IPv4 CIDR, and IPv6 CIDR. It lists five subnets:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR
-	subnet-078f1b79825a5fee0	Available	vpc-0b412746b28b797e7	Off	172.31.16.0/20	-
dev-subnet-1	subnet-082ba110498bcf841	Available	vpc-05aa4ccb3cf969fa1 dev-vpc	Off	10.0.10.0/24	-
prod-public-subnet	subnet-0a2cee75f0ad691a6	Available	vpc-09822723aed1d2815 pro...	Off	10.0.10.0/24	-
-	subnet-05aee612a1c2e607f	Available	vpc-0b412746b28b797e7	Off	172.31.0.0/20	-
-	subnet-0f3fcff0e6b1889fc	Available	vpc-0b412746b28b797e7	Off	172.31.32.0/20	-
- Route tables (5) Info**: A table with columns: Name, Route table ID, Explicit subnet associa..., Edge associations. It lists five route tables:

Name	Route table ID	Explicit subnet associa...	Edge associations
-	rtb-000461310bb92433a	-	-
dev-rt	rtb-05c987809e89fcf3	subnet-082ba110498bcf...	-
-	rtb-083c4d05710186299	-	-
prod-public-rt	rtb-0586238f2c56ea9ac	subnet-0a2cee75f0ad691...	-
-	rtb-058ec44464c1c6742	-	-

On the right side, there is a terminal window showing the output of the apply command:

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

Outputs:

server_config = {
  "backup_enabled" = false
  "instance_type" = "t2.micro"
  "monitoring" = true
  "name" = "lab-server-1"
  "storage_gb" = 20
}
```

- 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"
  }
}
```

- task8_default_route_table_apply.png

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

Outputs:

server_config = {
  "backup_enabled" = false
  "instance_type" = "t2.micro"
  "monitoring" = true
  "name" = "lab-server-1"
  "storage_gb" = 20
}
```

Task 9- Security group, key apir, EC2 instance, user_data & nginx

- task9_my_ip_variable_added.png

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

- task9_public_ip_curl.png

```
@hamna-mahmood ② /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ curl icanhazip.com
20.192.21.54
```

```
my_ip = "20.192.21.54/32"
instance_type = "t3.micro"
availability_zone = "me-central-1a" # or your chosen AZ
env_prefix = "dev"

-- INSERT --
```

- task9_security_group_apply.png

```
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"
  }
}
```

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

```
Outputs:
```

```
server_config = {
  "backup_enabled" = false
  "instance_type" = "t2.micro"
  "monitoring" = true
  "name" = "lab-server-1"
  "storage_gb" = 20
}
```

```
@hamna-mahmood ② /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $
```

- task9_keypair_created_and_saved.png

```
00 MyE@hamna-mahmood ② /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ chmod 600 MyED25519Key.pem
@hamna-mahmood ② /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ aws ec2 create-key-pair --key-name MyED25519Key --key-type ed25519 --key-format pem --query 'KeyMaterial' --output text > MyED25519Key.pem
```

```
@hamna-mahmood ② /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ cat .gitignore
form/*
*.tfstate
*.tfstate.*
*.tfvars
*.pem
*.pem
```

- task9_instance_type_set.png

```
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
}
```

- task9_ec2_apply_and_public_ip.png

```
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 13s [id=i-06f2433df6a51e8e4]

Warning: Value for undeclared variable

The root module does not declare a variable named "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: 1 added, 0 changed, 0 destroyed.

Outputs:

aws_instance_public_ip = "3.28.191.9"
server_config = {
  "backup_enabled" = false
  "instance_type" = "t2.micro"
  "monitoring" = true
  "name" = "lab-server-1"
  "storage_gb" = 20
}
@hamna-mahmood ② /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $
```

- task9_ssh_into_ec2.png

```
@hamna-mahmood ② /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ ssh -i MyED25519Key.pem ec2-user@3.28.191.9
The authenticity of host '3.28.191.9' (3.28.191.9) can't be established.
ED25519 key fingerprint is SHA256:Q/edvNjeVf1a/Llikda4JRq3iBftGokFQtrEopvY1bg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.28.191.9' (ED25519) to the list of known hosts.

      _#
     ~\_\#\#\#_      Amazon Linux 2023
     ~~\_\#####\
     ~~ \#\#\|
     ~~   \#/ _--> https://aws.amazon.com/linux/amazon-linux-2023
     ~~   V~' .-' /
     ~~~   / \
     ~~. / / \
     _/ / / \
[ec2-user@ip-10-0-10-188 ~]$
```

- task9_ssh_keypair_and_ssh.png

```
@hamna-mahmood ② /workspaces/CC-Hamna-Mahmood-25-BSE-VA (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:tYZlqo3cDRtHka2SJp4Unzf0jEkcal30xbB3ki4ugLQ codespace@codespaces-aa6536
The key's randomart image is:
+--[ED25519 256]--+
|          +o+ . |
|         . o B.. + |
|        + *=0 o . |
|       + *BB.+ . . |
|      + *S0+. o. |
|     .E=.B   o . |
|    + +... . . . |
|       . . . . . |
|      o... . . . |
+----[SHA256]-----+
```

- task9_ssh_keypair_and_shh2.png

```
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

resource "aws_key_pair" "ssh_key" {
  key_name     = "serverkey"
  public_key   = file("~/ssh/id_ed25519.pub")
}
```

- task9_ec2_apply_and_public_ip

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

Outputs:

aws_instance_public_ip = "51.112.229.108"
server_config = {
  "backup_enabled" = false
  "instance_type" = "t2.micro"
  "monitoring" = true
  "name" = "lab-server-1"
  "storage_gb" = 20
}
```

- task9_ssh_keypair_Ans_ssh3

```
@hamna-mahmood ② /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ ssh ec2-user@51.112.229.108
The authenticity of host '51.112.229.108' (51.112.229.108) can't be established.
ED25519 key fingerprint is SHA256:F6IX2Dlh2f53v82zUaP77+F38RNJxyv6T8uRYFP1wAY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '51.112.229.108' (ED25519) to the list of known hosts.

,
#_
~\_\#\#\#
~~\_#\#\#\#
~~\#\#\#
~~\#/`-->
~~V~,`->
~~`/
~~`-/`/
~~`/`/
[ec2-user@ip-10-0-10-254 ~]$
```

- task9_nginx_local_curl.png

```
[ec2-user@ip-10-0-10-62 ~]$ 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-10-0-10-62 ~]$
```

- task9_nginx_browser_page.png



Cleanup

- cleanup_destroy.png

```
Destroy complete! Resources: 7 destroyed.
@hamna-mahmood ② /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $
```

- cleanup_state_files.png

```
@hamna-mahmood [ /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 97,
  "lineage": "44daad15-ff50-9119-441e-f71bf3fee6a5",
  "outputs": {},
  "resources": [],
  "check_results": null
}
@hamna-mahmood [ /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 89,
  "lineage": "44daad15-ff50-9119-441e-f71bf3fee6a5",
  "outputs": {
    "aws_instance_public_ip": {
      "value": "52.1.1.1",
      "type": "string"
    },
    "server_config": {
      "value": {
        "backup_enabled": false,
        "instance_type": "t2.micro",
        "monitoring": true,
        "name": "lab-server-1",
        "storage_gb": 20
      }
    }
  }
}
```

- **cleanup_verify_no_secrets.png**

```
@hamna-mahmood ② /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    .main.tf.swo
    .main.tf.swp
    .terraform.lock.hcl
    .terraform/
    aws/
    awscliv2.zip
    "ec2 describe-vpcs \\"
    main.tf
    variables.tf

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

@hamna-mahmood ② /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ cat .gitignore
form/*
*.tfstate
*.tfstate.*
*.tfvars
*.pem
*.pem
```