DEPARTMENT OF SOFTWARE ENGINEERING

LAB#12

SUBMITTED TO:

ENGR. MUHAMMAD SHOAIB

ENGR. WAQAS SALEEM

SUBMITTED BY: NEHA AMJAD

REG NO: 2021-BSE-024

README.md | variables.tf U | outputs.tf U | locals.tf U | terraform.tfvars U | entry-script.sh U

Public > Lab12 > $ entry-script.sh

```
#!/bin/bash
set -e
yum update -y
yum install -y nginx
systemctl start nginx
systemctl enable nginx
```

```
@neha-121 →/workspaces/cc_-nehaamjad-_-2021-BSE-024-/Public/Lab12 (main) $ gh --version
gh auth status
  - Git operations protocol: https
  - Token: ghu_**********************************
@neha-121 →/workspaces/cc_-nehaamjad-_-2021-BSE-024-/Public/Lab12 (main) $ mkdir -p modules
touch main.tf variables.tf outputs.tf locals.tf terraform.tfvars entry-script.sh
ls -la
total 12
drwxrwxrwx+ 3 codespace codespace 4096 Jan 30 09:06 .
drwxrwxrwx+ 3 codespace codespace 4096 Jan 30 08:49 ..
-rw-rw-rw-  1 codespace codespace    0 Jan 30 09:06 entry-script.sh
-rw-rw-rw-  1 codespace codespace    0 Jan 30 09:06 locals.tf
-rw-rw-rw-  1 codespace codespace    0 Jan 30 09:06 main.tf
drwxrwxrwx+ 2 codespace codespace 4096 Jan 30 09:06 modules
-rw-rw-rw-  1 codespace codespace    0 Jan 30 09:06 outputs.tf
-rw-rw-rw-  1 codespace codespace    0 Jan 30 09:06 terraform.tfvars
-rw-rw-rw-  1 codespace codespace    0 Jan 30 09:06 variables.tf
@neha-121 →/workspaces/cc_-nehaamjad-_-2021-BSE-024-/Public/Lab12 (main) $
```

[Preview] README.md | variables.tf U | outputs.tf U | locals.tf U

Public > Lab12 > locals.tf

```
locals {
  my_ip = "${chomp(data.http.my_ip.response_body)}/32"
}

data "http" "my_ip" {
  url = "https://icanhazip.com"
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS bash - Lab12

variables.tf U | outputs.tf U | locals.tf U | terraform.tfvars U | entry-script.sh U | main.tf U

Public > Lab12 > main.tf

```
33  resource "aws_default_security_group" "default_sg" {
43    ingress {
47      cidr_blocks = ["0.0.0.0/0"]
48    }
49
50    egress {
51      from_port   = 0
52      to_port     = 0
53      protocol    = "-1"
54      cidr_blocks = ["0.0.0.0/0"]
55    }
56  }
57
58  resource "aws_key_pair" "ssh-key" {
59    key_name   = "serverkey"
60    public_key = file(var.public_key)
61  }
62
63  resource "aws_instance" "myapp-server" {
64    ami                         = "ami-05524d6658fcf35b6"
65    instance_type               = var.instance_type
66    subnet_id                   = aws_subnet.myapp_subnet_1.id
67    security_groups             = [aws_default_security_group.default_sg.id]
68    availability_zone           = var.availability_zone
69    associate_public_ip_address = true
70    key_name                    = aws_key_pair.ssh-key.key_name
71
72    user_data = file("./entry-script.sh")
73
74    tags = {
75      Name = "${var.env_prefix}-ec2-instance"
```

[Preview] README.md | variables.tf U | outputs.tf U

Public > Lab12 > outputs.tf

```
1  output "aws_instance_public_ip" {
2    value = aws_instance.myapp-server.public_ip
3  }
4
```

```
@neha-121 →/workspaces/cc_-nehaamjad-_-2021-BSE-024-/Public/Lab12 (main) $ ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519
 -N ""
Generating public/private ed25519 key pair.
Created directory '/home/codespace/.ssh'.
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:qnsrNf3bDVFCB/8PDS2tEdDEyJfZCSwgIWS80yeAY8A codespace@codespaces-bc92a9
The key's randomart image is:
+--[ED25519 256]--+
|... ++ o....=O=+.|
| E +.o. .  oo+*=.|
|  . . +     o.* o|
|     o o .   o B |
|      ..S   . o o|
|      o..    . ..|
|     ... .  .   .|
|    ...   .. o   |
|    o+..  ... .  |
+----[SHA256]-----+
```

```
@neha-121 →/workspaces/cc_-nehaamjad-_-2021-BSE-024-/Public/Lab12 (main) $ terraform init
terraform apply -auto-approve
- Installed hashicorp/http v3.5.0 (signed by HashiCorp)
- Installing hashicorp/aws v6.30.0...
- Installed hashicorp/aws v6.30.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
data.http.my_ip: Reading...
data.http.my_ip: Read complete after 0s [id=https://icanhazip.com]
```

[Preview] README.md | variables.tf U | outputs.tf U | locals.tf U | terraform.tfvars U

Public > Lab12 > 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"
instance_type        = "t3.micro"
public_key           = "~/.ssh/id_ed25519.pub"
private_key          = "~/.ssh/id_ed25519"

```

[Preview] README.md | variables.tf U

Public > Lab12 > variables.tf

```
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
variable "instance_type" {}
variable "public_key" {}
variable "private_key" {}

```

```
@neha-121 ➜ /workspaces/cc_-nehaamjad-_-2021-BSE-024-/Lab11 (main) $ terraform apply -auto-approve -var "api_session_token=my_API_session_Token_12345"
terraform apply -auto-approve -var "api_session_token=my_API_session_Token_12345"
  ~ 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"
@neha-121 ➜ /workspaces/cc_-nehaamjad-_-2021-BSE-024-/Lab11 (main) $ terraform apply -auto-approve

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 = "10.0.30.0/24"
```

```
@neha-121 →/workspaces/cc_-nehaamjad-_-2021-BSE-024-/Lab11 (main) $ terraform apply -auto-approve

Error: Ephemeral value not allowed

  on main.tf line 35, in output "api_session_token_output":
  35:   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.
```

```
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 let
  }
}

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

```
--output text
@neha-121 →/workspaces/cc_-nehaamjad-_-2021-BSE-024-/Lab11 (main) $ aws configure
AWS Access Key ID [None]: AKIAYGZVAOQUXB7AFA4K
AWS Secret Access Key [None]: 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
Default region name [None]: ap-east-1
Default output format [None]: json
@neha-121 →/workspaces/cc_-nehaamjad-_-2021-BSE-024-/Lab11 (main) $
```

```
@neha-121 ➜/workspaces/cc_-nehaamjad-_-2021-BSE-024-/Lab11 (main) $ terraform init
terraform apply -auto-approve

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>
environment = "dev"
monitoring_enabled = true
primary_subnet_id = "subnet-00b47a6f034c9a932"
project_name = "lab_work"
server_config = {
  "backup_enabled" = false
  "instance_type" = "t3.micro"
  "monitoring" = true
  "name" = "web-server"
  "storage_gb" = 20
}
subnet_cidr_block_output = "10.0.0.0/24"
subnet_count = 3
tags = tomap({
  "Environment" = "dev"
  "Owner" = "platform-team"
  "Project" = "sample-app"
})
```

b11 > terraform.tfvars

```
subnet_cidr_block = "10.0.30.0/24"
environment       = "dev"
project_name      = "lab_work"
primary_subnet_id = "subnet-00b47a6f034c9a932"
subnet_count      = 3
monitoring        = true
```

```
variable "subnet_cidr_block" {
  type        = string
  default     = ""
  description = "CIDR block to assign to the application subnet"
  sensitive   = false
  nullable    = false
  ephemeral   = false
}
variable "environment" {}
variable "project_name" {}
variable "primary_subnet_id" {}
variable "subnet_count" {}
variable "monitoring" {}
```

```
}
@neha-121 →/workspaces/Lab10 (main) $ aws ec2 describe-vpcs --filter "Name=vpc-id,Values=vpc-00c235734dab9fbf4"
{
    "Vpcs": [
        {
            "OwnerId": "564362507305",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-0c58907c95e76d72d",
                    "CidrBlock": "10.0.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                }
            ],
            "IsDefault": false,
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "VpcId": "vpc-00c235734dab9fbf4",
            "State": "available",
            "CidrBlock": "10.0.0.0/16",
            "DhcpOptionsId": "dopt-0272c2329c23c4965"
        }
    ]
}
Plan: 3 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_vpc.development_vpc: Creating...
aws_subnet.dev_subnet_1_existing: Creating...
aws_subnet.dev_subnet_1_existing: Creation complete after 1s [id=subnet-0d75e318b4639a1b1]
aws_vpc.development_vpc: Creation complete after 2s [id=vpc-0f1d5875ce426d30f]
aws_subnet.dev_subnet_1: Creating...
aws_subnet.dev_subnet_1: Creation complete after 1s [id=subnet-0847aea67997ca257]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
@neha-121 →/workspaces/Lab10 (main) $ vim main.tf
@neha-121 →/workspaces/Lab10 (main) $
```

```
@neha-121 →/workspaces/Lab10 (main) $ terraform apply
      + default_security_group_id           = (known after apply)
      + dhcp_options_id                     = (known after apply)
      + enable_dns_hostnames                = (known after apply)
      + enable_dns_support                  = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                  = (known after apply)
      + instance_tenancy                    = "default"
      + ipv6_association_id                 = (known after apply)
      + ipv6_cidr_block                     = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                 = (known after apply)
      + owner_id                            = (known after apply)
      + region                              = "me-central-1"
      + tags_all                            = (known after apply)
    }

Plan: 3 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_vpc.development_vpc: Creating...
aws_subnet.dev_subnet_1_existing: Creating...
aws_subnet.dev_subnet_1_existing: Creation complete after 1s [id=subnet-0d75e318b4639a1b1]
aws_vpc.development_vpc: Creation complete after 2s [id=vpc-0f1d5875ce426d30f]
aws_subnet.dev_subnet_1: Creating...
aws_subnet.dev_subnet_1: Creation complete after 1s [id=subnet-0847aea67997ca257]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                bash

@neha-121 →/workspaces/Lab10 (main) $ terraform plan
      + tags_all                                   = (known after apply)
      + vpc_id                                     = "vpc-0091f33e6aff43d6f"
    }

  # aws_vpc.development_vpc will be created
  + resource "aws_vpc" "development_vpc" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = (known after apply)
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + region                               = "me-central-1"
      + tags_all                             = (known after apply)
    }

Plan: 3 to add, 0 to change, 0 to destroy.

─────────────────────────────────────────────────────────────────────────────

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if
you run "terraform apply" now.
@neha-121 →/workspaces/Lab10 (main) $ terraform plan
terraform apply -auto-approve

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes
are needed.
aws_vpc.development_vpc: Refreshing state... [id=vpc-0f1d5875ce426d30f]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0847aea67997ca257]

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.
@neha-121 →/workspaces/Lab10 (main) $
```

```
@neha-121 →/workspaces/Lab10 (main) $ terraform refresh
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-00c235734dab9fbf4]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0091f33e6aff43d6f]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0fb584d597fa7e387]
```