

Lab 2

1-restructure your code to use variables

```
variables.tf
1 variable "instance_type" {
2   type = string
3 }
4
5 variable "region" {
6   type    = string
7   default = "us-east-1"
8 }
9
10
11 variable "profile" {
12   type    = string
13   default = "default"
14 }
15
16 }
17
18 variable "vpc_cidr_block" {
19   type = string
20 }
21
22
23 variable "subnets" {
24   type = list(object({
25     name      = string
26     cidr_block = string
27     type      = string
28     az        = string
29   }))
30 }
31
32
33 variable "instances" {
34   type = list(object({
35     name    = string
36     subnet  = string
37     type    = string
38   }))
39 }
40 }
```

2-create all subnets with single resource using for_each

```
variables.tf  subnet.tf x
network > subnet.tf
1 resource "aws_subnet" "subnets" {
2   for_each      = { for subnet in var.subnets : subnet.name => subnet }
3   vpc_id        = aws_vpc.main.id
4   cidr_block    = each.value.cidr_block
5   map_public_ip_on_launch = each.value.type == "public" ? true : false
6   availability_zone = "${var.region}${each.value.az}"
7   tags          = { "Name" = each.value.name }
8 }
```

3-make condition on subnet resource based on type (public or private) to control map_public_ip_on_launch

```
map_public_ip_on_launch = each.value.type == "public" ? true : false
```

4-create all ec2s with single resource using count

```
variables.tf • subnet.tf • ec2.tf •
ec2.tf
1 resource "aws_instance" "instances" {
2   ami           = data.aws_ami.amazon_linux.id
3   instance_type = var.instance_type
4
5   count          = 2
6   subnet_id      = module.network_module.subnets["${var.instances[count.index].subnet}"].id
7
8   tags = {
9     Name = var.instances[count.index].name
10  }
11  vpc_security_group_ids = [var.instances[count.index].type == "public" ? module.network_module.public_sg.id : module.network_module.private_sg.id]
12
13  provisioner "local-exec" {
14    command = var.instances[count.index].name == "bastion" ? "echo The server ${self.tags.Name} IP address is ${self.public_ip} > servers_ips" : ""
15  }
16
17 }
```

5-create two workspaces dev and prod

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
bash + v [ ] [ ] ... ^ x

Mon Mar  3 08:13:43 PM EET 2025
• [engy@localhost Lab 2]$ terraform workspace new dev
Created and switched to workspace "dev"!

You're now on a new, empty workspace. Workspaces isolate their state,
so if you run "terraform plan" Terraform will not see any existing state
for this configuration.
• [engy@localhost Lab 2]$ terraform workspace new prod
Created and switched to workspace "prod"!

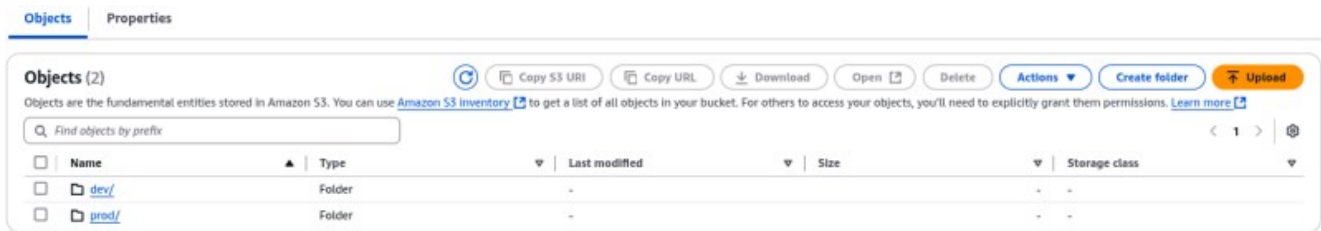
You're now on a new, empty workspace. Workspaces isolate their state,
so if you run "terraform plan" Terraform will not see any existing state
for this configuration.
• [engy@localhost Lab 2]$ terraform workspace list
default
dev
* prod
```

6-create two variable definition files(.tfvars) for the two environments

```
variables.tf • subnet.tf • ec2.tf • prod.tfvars •
prod.tfvars
1 instance_type = "t2.micro"
2 region       = "eu-central-1"
3 profile      = "default"
4 vpc_cidr_block = "10.0.0.0/16"
5 subnets = [
6   {
7     name       = "pub_subnet_1"
8     cidr_block = "10.0.0.0/24"
9     type       = "public"
10  },
11  {
12    name       = "pub_subnet_2"
13    cidr_block = "10.0.1.0/24"
14    type       = "public"
15  },
16  {
17    name       = "priv_subnet_1"
18    cidr_block = "10.0.2.0/24"
19    type       = "private"
20  },
21  {
22    name       = "priv_subnet_2"
23    cidr_block = "10.0.3.0/24"
24    type       = "private"
25  }
26 ]
27
28 instances = [
29   {
30     name       = "bastion"
31     subnet     = "pub_subnet_1",
32     type       = "public"
33   },
34   {
35     name       = "application",
36     subnet     = "priv_subnet_1",
37     type       = "private"
38   }
39 ]
40
41
42
43
44
```

```
variables.tf • dev.tfvars × subnet.tf • ec2.tf • prod.tfvars •
dev.tfvars
1 instance_type = "t2.micro"
2 region       = "us-east-1"
3 profile      = "default"
4 vpc_cidr_block = "10.0.0.0/16"
5 subnets = [
6   {
7     name       = "pub_subnet_1"
8     cidr_block = "10.0.0.0/24"
9     type       = "public"
10    az         = "a"
11  },
12  {
13    name       = "pub_subnet_2"
14    cidr_block = "10.0.1.0/24"
15    type       = "public"
16    az         = "b"
17  },
18  {
19    name       = "priv_subnet_1"
20    cidr_block = "10.0.2.0/24"
21    type       = "private"
22    az         = "a"
23  },
24  {
25    name       = "priv_subnet_2"
26    cidr_block = "10.0.3.0/24"
27    type       = "private"
28    az         = "b"
29  }
30 ]
31
32 instances = [
33   {
34     name       = "bastion"
35     subnet     = "pub_subnet_1",
36     type       = "public"
37   },
38   {
39     name       = "application",
40     subnet     = "priv_subnet_1",
41     type       = "private"
42   }
43 ]
44
45
46
```

Ln 25, Col 1 Spaces: 4 UTF-8 LF Plain Text



7-apply your code to create two environments one in us-east-1 and eu-central-1

```
aws_instance.instances[1]: Creation complete after 14s [id=i-03907691f3a365393]
aws_instance.instances[0]: Provisioning with 'local-exec'...
aws_instance.instances[0] (local-exec): Executing: ["/bin/sh" "-c" "echo The serv
aws_instance.instances[0]: Creation complete after 15s [id=i-0d5ecc2656c39d508]

Apply complete! Resources: 21 added, 0 changed, 0 destroyed.
```

The screenshot shows the AWS Management Console 'Instances' page. It includes a search bar, a filter dropdown set to 'All states', and a table of instances. The table has columns: Name, Instance ID, Instance state, Instance type, Status check, and Alarm s. Two instances are listed: 'bastion' and 'application'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm s
bastion	i-0d5ecc2656c39d508	Running	t2.micro	Initializing	View al
application	i-03907691f3a365393	Running	t2.micro	Initializing	View al

8-run local-exec provisioner to print the public_ip of bastion ec2

```
# Print Bastion Public IP
provisioner "local-exec" {
  command = "echo Bastion Public IP: ${self.public_ip} >> EC2_Public_IPs.txt"
}
```

```
servers_ips x
servers_ips
1 The server bastion IP address is 3.121.160.115
```

9- upload infrastructure code on github project

```
[engy@localhost Terraform]$ git add .
[engy@localhost Terraform]$ git commit -m "Lab 2"
[main 33cd519] Lab 2
 26 files changed, 1102 insertions(+)
 create mode 100644 ~/.lock.Lab 1.odt#
 create mode 100644 ~/.lock.Lab 2.odt#
 create mode 100644 Lab 1/Lab 1.pdf
 create mode 100644 Lab 1/Lab1.odt
 create mode 100644 Lab 2/ami.tf
 create mode 100644 Lab 2/backend.tf
 create mode 100644 Lab 2/dev.tfvars
 create mode 100644 Lab 2/ec2.tf
 create mode 100644 Lab 2/gitignore
 create mode 100644 Lab 2/network.tf
 create mode 100644 Lab 2/network/inputs.tf
 create mode 100644 Lab 2/network/internet-gateway.tf
 create mode 100644 Lab 2/network/outputs.tf
 create mode 100644 Lab 2/network/route-table-association.tf
 create mode 100644 Lab 2/network/route-tables.tf
 create mode 100644 Lab 2/network/security-groups.tf
 create mode 100644 Lab 2/network/subnet.tf
 create mode 100644 Lab 2/network/vpc.tf
 create mode 100644 Lab 2/prod.tfvars
 create mode 100644 Lab 2/provider.tf
 create mode 100644 Lab 2/servers_ips
 create mode 100644 Lab 2/subnets.tf
 create mode 100644 Lab 2/terraform.lock.hcl
 create mode 100644 Lab 2/terraform.tfstate
 create mode 100644 Lab 2/terraform.tfstate.backup
 create mode 100644 Lab 2/variables.tf
[engy@localhost Terraform]$ git push origin main
Enumerating objects: 32, done.
Counting objects: 100% (32/32), done.
Delta compression using up to 8 threads
Compressing objects: 100% (28/28), done.
Writing objects: 100% (30/30), 3.91 MiB | 654.00 KiB/s, done.
Total 30 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:EngyElhamzawy/Terraform_Labs.git
   ea3a302..33cd519  main -> main
[engy@localhost Terraform]$
```

EngyElhamzawy / Terraform_Labs

Type to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Files

main +

Go to file t

> Lab 1

Lab 2

> network

ami.tf

backend.tf

dev.tfvars

ec2.tf

gitignore

network.tf

prod.tfvars

provider.tf

servers_ips

subnets.tf

terraform.lock.hcl

terraform.tfstate

terraform.tfstate.backup

variables.tf

.gitignore

.terraform.lock.hcl

~.lock.Lab 1.odt#

~.lock.Lab 2.odt#

README.md

Terraform_Labs / Lab 2

Add file ...

EngyElhamzawy Lab 2 33cd519 · 1 minute ago History

Name	Last commit message	Last commit date
..		
network	Lab 2	1 minute ago
ami.tf	Lab 2	1 minute ago
backend.tf	Lab 2	1 minute ago
dev.tfvars	Lab 2	1 minute ago
ec2.tf	Lab 2	1 minute ago
gitignore	Lab 2	1 minute ago
network.tf	Lab 2	1 minute ago
prod.tfvars	Lab 2	1 minute ago
provider.tf	Lab 2	1 minute ago
servers_ips	Lab 2	1 minute ago
subnets.tf	Lab 2	1 minute ago
terraform.lock.hcl	Lab 2	1 minute ago
terraform.tfstate	Lab 2	1 minute ago
terraform.tfstate.backup	Lab 2	1 minute ago
variables.tf	Lab 2	1 minute ago

10- create rds(mysql) in private subnet

```
terraform.tfstate terraform.tfstate.backup terraform.lock.hcl gitignore rds(mysql).tf ...
rds(mysql).tf
1 resource "aws_db_subnet_group" "db_subnet_group" {
2   name = "my-db-subnet-group"
3   subnet_ids = [aws_subnet.subnets["priv_subnet_1"].id]
4
5   tags = {
6     Name = "DBGroup"
7   }
8 }
9
10 resource "aws_db_instance" "mysql_instance" {
11   identifier = "my-mysql-db"
12   engine = "mysql"
13   engine_version = "8.0"
14   instance_class = "db.t2.micro"
15   db_name = "mydatabase"
16   username = "admin"
17   password = "mysecretpassword"
18   allocated_storage = 20
19   db_subnet_group_name = aws_db_subnet_group.db_subnet_group.name
20   multi_az = false
21   storage_encrypted = true
22   publicly_accessible = false
23   backup_retention_period = 7
24   vpc_security_group_ids = [aws_security_group.sg_vpc_cidr_only.id]
25   tags = {
26     Name = "MySQL"
27   }
28 }
```

Databases (1)

Filter by databases

Group resources Modify Actions Restore from S3 Create database

DB Identifier	Status	Role	Engine	Region &...	Size	Recommendations	CPU	Current activity	Mainten...	VPC	Multi-AZ
terraform-20250303071718678600000	Available	Instance	MySQL Co...	us-east-1a	db.t3.micro				none	vpc-060d54627a405c9af	No

11- create elastic cache redis in private subnet

```
elastic-cache.tf
elastic-cache.tf
1
2
3 resource "aws_elasticache_subnet_group" "elasticache_subnet_group" {
4   name = "tf-test-cache-subnet"
5   subnet_ids = [module.network_module.subnets["priv_subnet_1"].id, module.network_module.subnets["priv_subnet_2"].id]
6 }
7
8
9 resource "aws_elasticache_cluster" "elasticache_cluster" {
10   cluster_id = "cluster-example"
11   engine = "redis"
12   node_type = "cache.t3.micro"
13   num_cache_nodes = 1
14   parameter_group_name = "default.redis7"
15   port = 6379
16   subnet_group_name = aws_elasticache_subnet_group.elasticache_subnet_group.name
17 }
```

Cache name	Status	Description	Engine version	Configuration
cluster-example	Available	-	7.1.0	cache.t3.micro

Finally , that's all ! :)

You can access my github repo for this Lab through :

https://github.com/EngyElhamzawy/Terraform_Labs/tree/main/Lab%202