Terraform Lab Report: Creating Multiple EC2 Instances with Loops

Objective

In this lab, I learned how to create multiple EC2 instances dynamically using two types of loops in Terraform:

- count
- for_each

Steps I Followed

Part 1: count

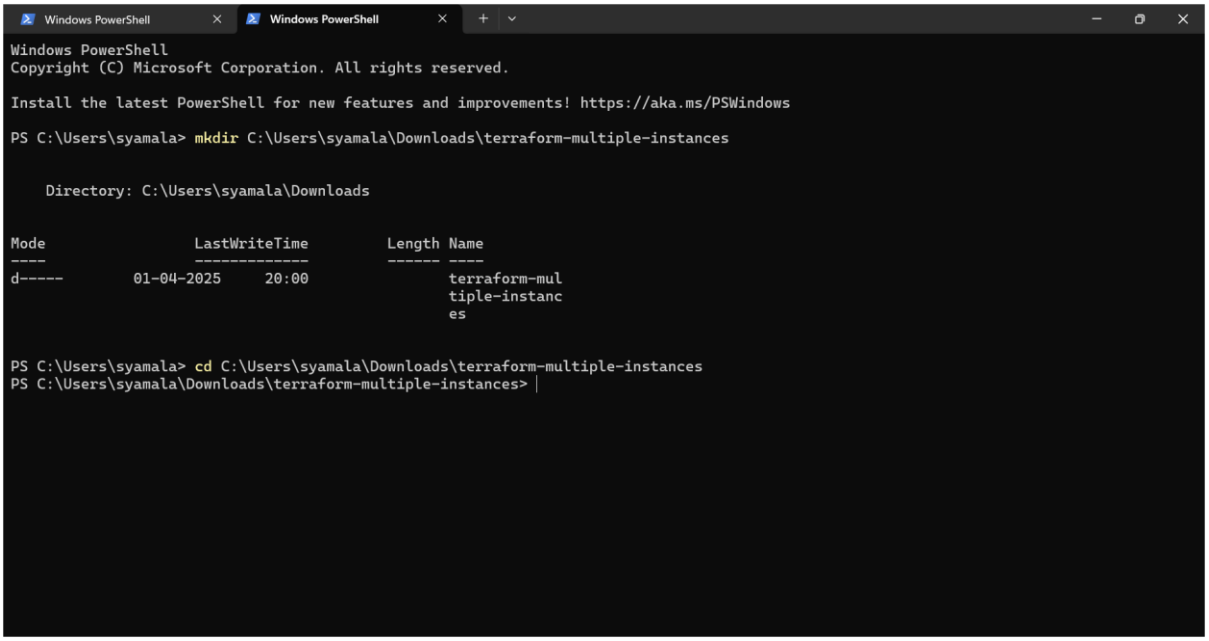
1. Created a folder named terraform-multiple-instances
2. Wrote a main.tf file to create 3 EC2 instances using count = 3
3. Used this AMI ID: ami-03c1aa37835df06a1 (Ubuntu 20.04 in us-east-1)
4. Ran terraform init and then terraform apply -auto-approve
5. Verified the 3 EC2 instances in the AWS Console
6. Destroyed the instances with terraform destroy -auto-approve

Problems I Faced & How I Solved Them

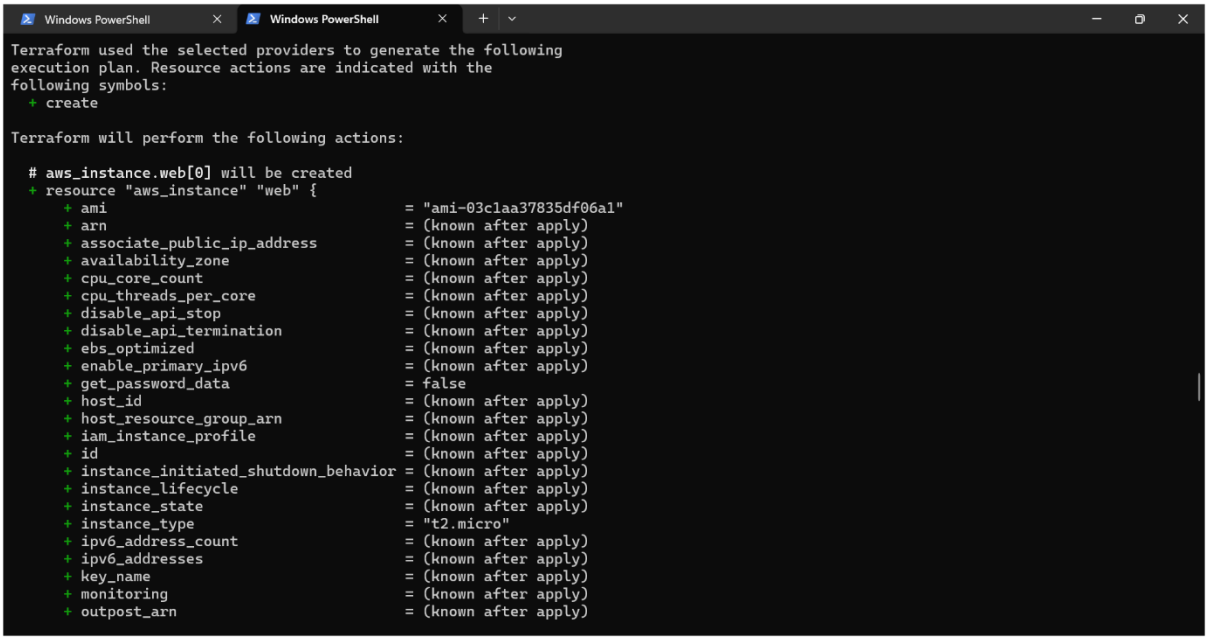
Problem	**Solution**
Old AMI ID not found	Searched for the latest AMI on Ubuntu AMI Locator
AMI not valid in the region	Verified region is us-east-1 and matched the AMI
File saved as .tf.txt	Fixed the filename to just main.tf

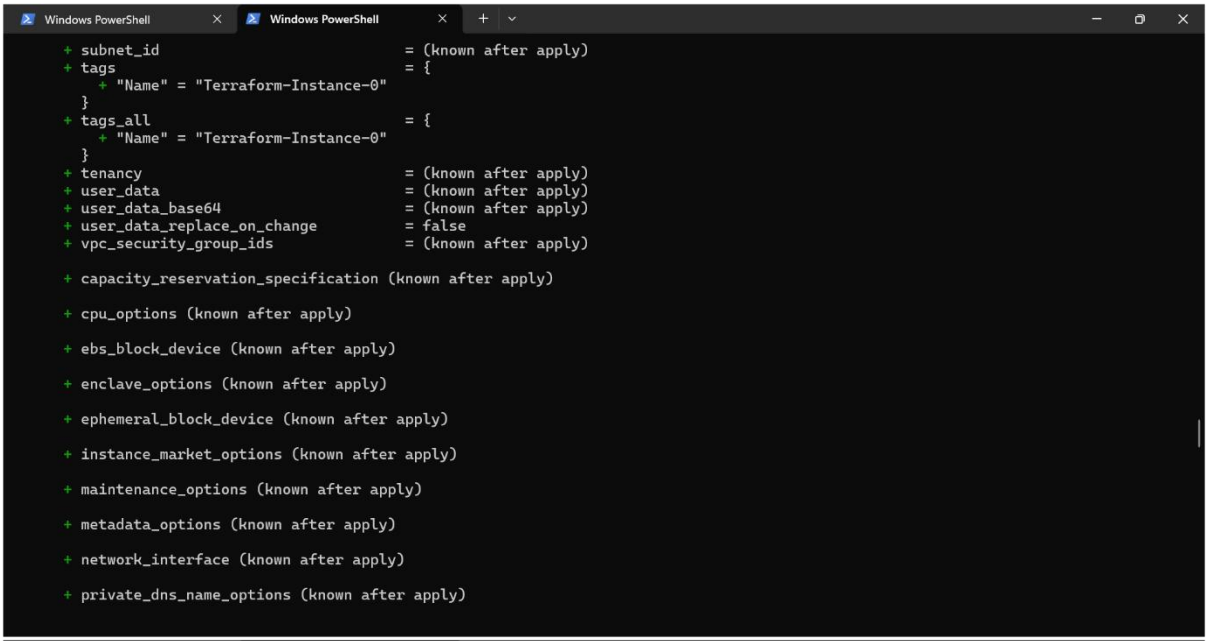
Screenshots Taken

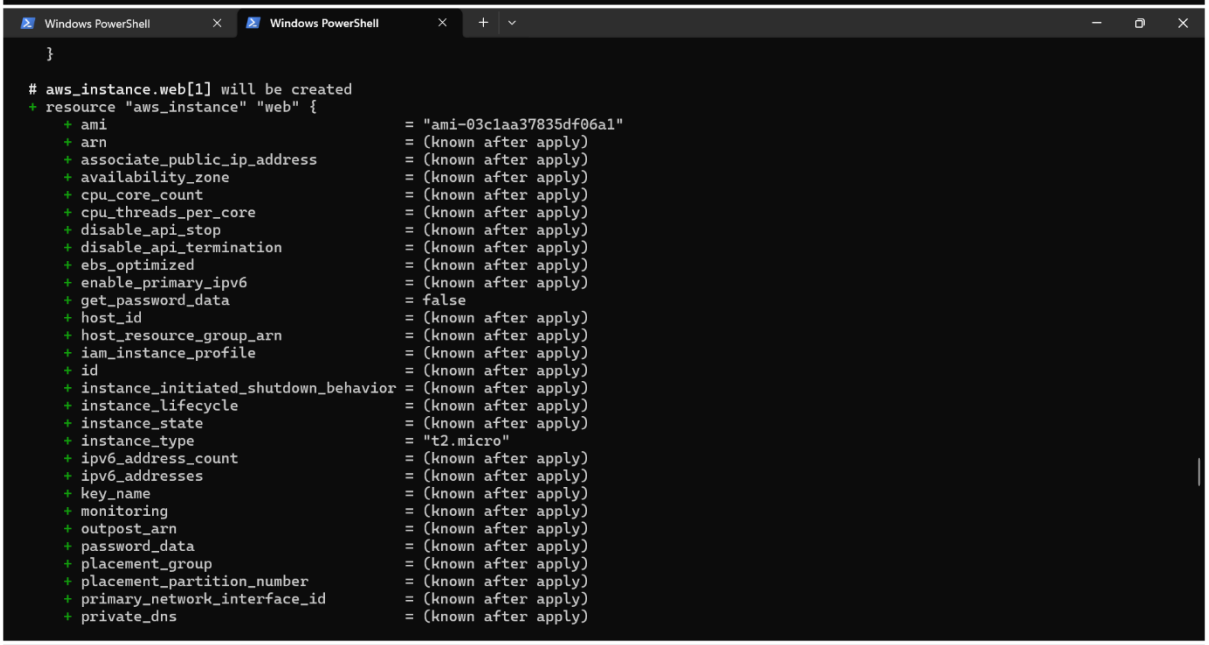
- Terraform init output
- Terraform apply output
- AWS Console showing EC2 instances
- Terraform destroy output

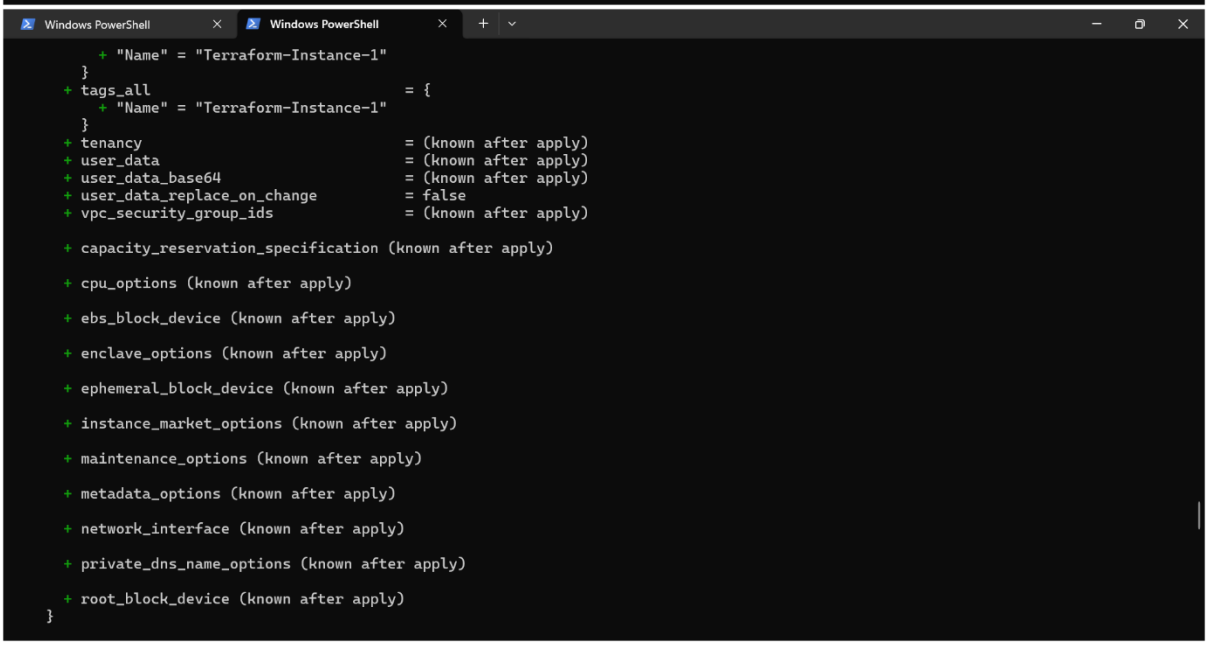

Windows PowerShell
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\syamala> mkdir C:\Users\syamala\Downloads\terraform-multiple-instances
Directory: C:\Users\syamala\Downloads
Mode LastWriteTime Length Name
d----- 01-04-2025 20:00 terraform-mul
tiple-instanc
es
PS C:\Users\syamala> cd C:\Users\syamala\Downloads\terraform-multiple-instances
PS C:\Users\syamala\Downloads\terraform-multiple-instances>

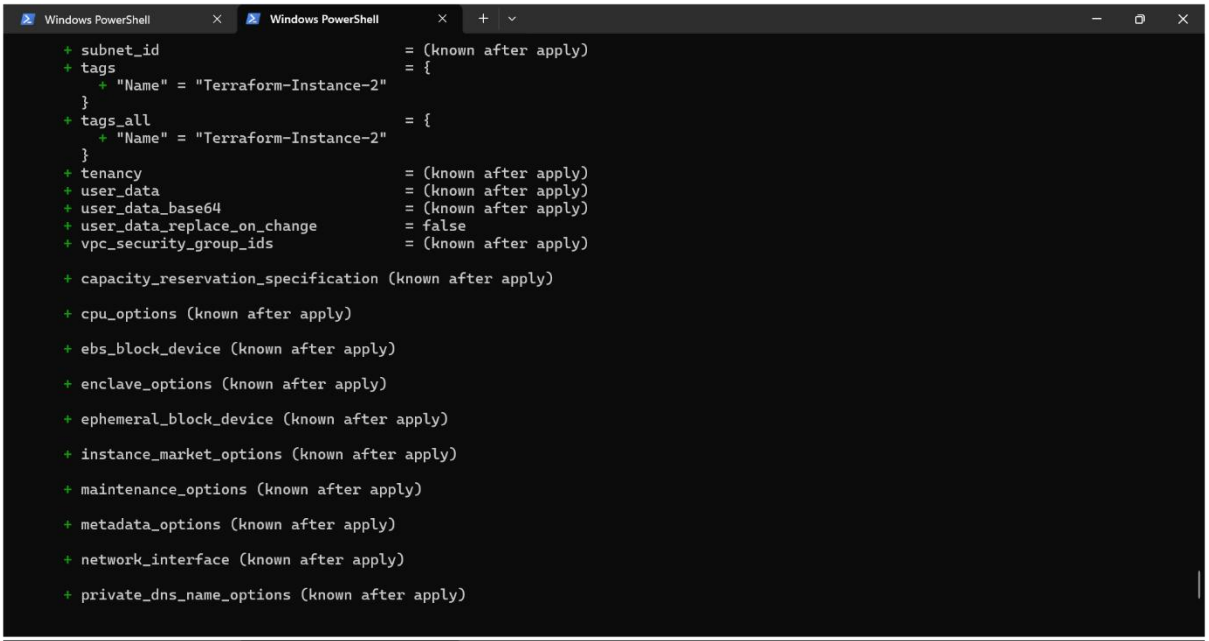

PS C:\Users\syamala\Downloads\terraform-multiple-instances> & "C:\Users\syamala\Downloads\terraform.exe" init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.93.0...

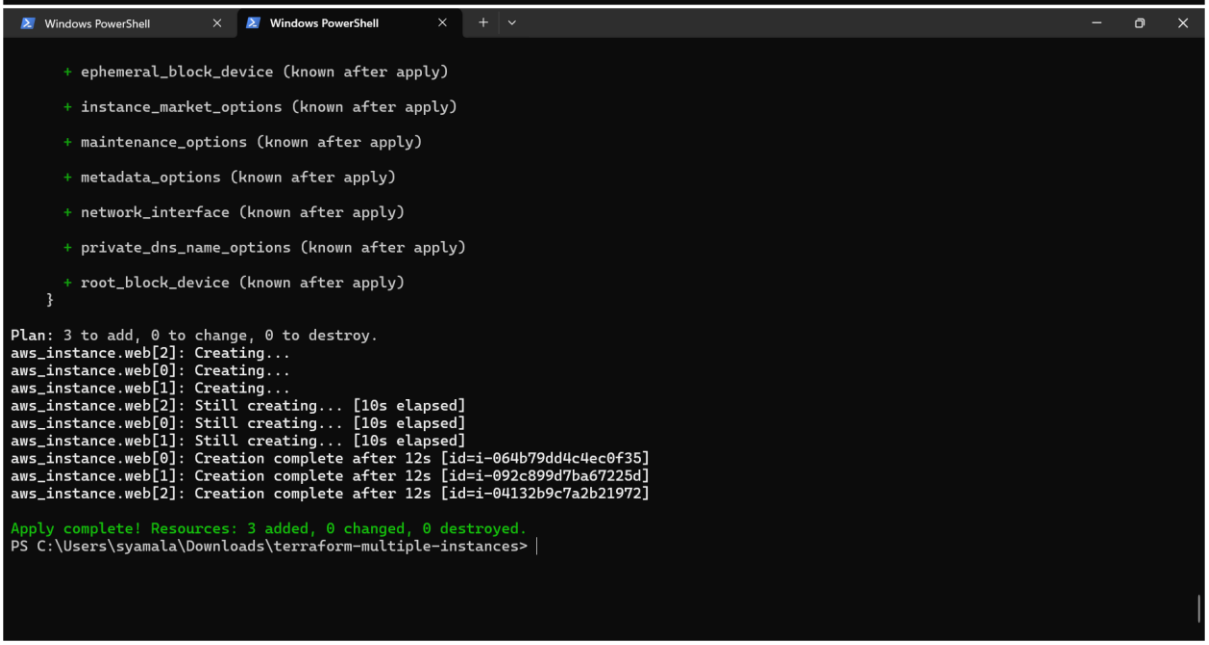

Windows PowerShell
Windows PowerShell
Terraform used the selected providers to generate the following
execution plan. Resource actions are indicated with the
following symbols:
+ create
Terraform will perform the following actions:
# aws_instance.web[0] will be created
+ resource "aws_instance" "web" {
+ ami = "ami-03c1aa37835df06a1"
+ arn = (known after apply)
+ associate_public_ip_address = (known after apply)
+ availability_zone = (known after apply)
+ cpu_core_count = (known after apply)
+ cpu_threads_per_core = (known after apply)
+ disable_api_stop = (known after apply)
+ disable_api_termination = (known after apply)
+ ebs_optimized = (known after apply)
+ enable_primary_ipv6 = (known after apply)
+ get_password_data = false
+ host_id = (known after apply)
+ host_resource_group_arn = (known after apply)
+ iam_instance_profile = (known after apply)
+ id = (known after apply)
+ instance_initiated_shutdown_behavior = (known after apply)
+ instance_lifecycle = (known after apply)
+ instance_state = (known after apply)
+ instance_type = "t2.micro"
+ ipv6_address_count = (known after apply)
+ ipv6_addresses = (known after apply)
+ key_name = (known after apply)
+ monitoring = (known after apply)
+ outpost_arn = (known after apply)

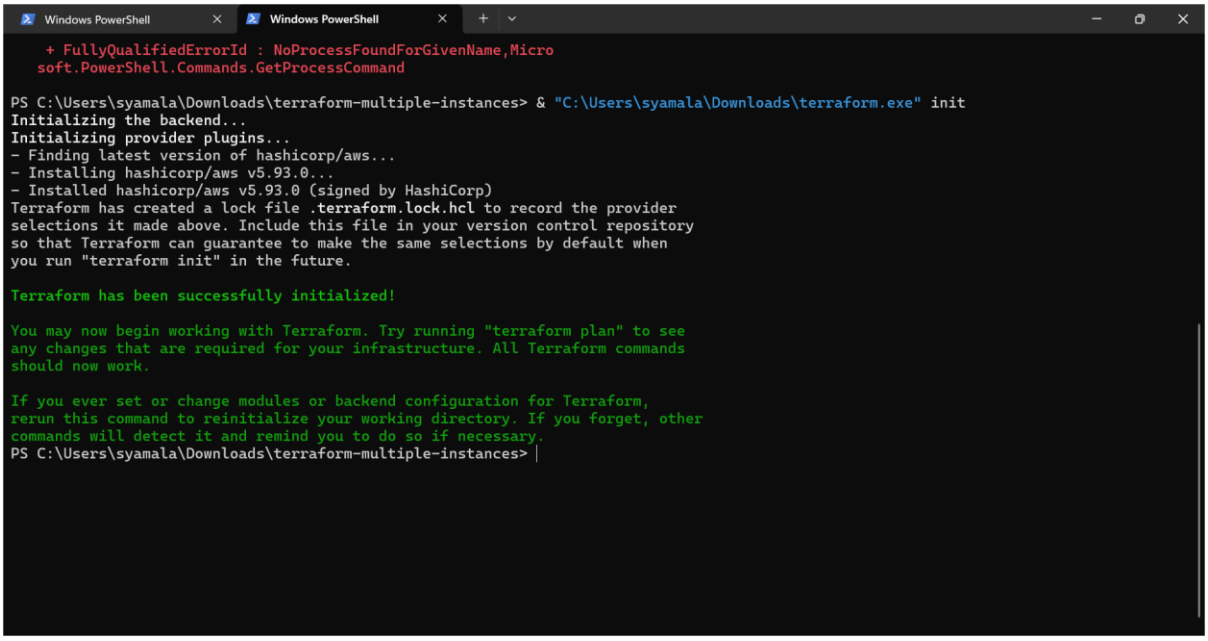

Windows PowerShell
Windows PowerShell
+ subnet_id = (known after apply)
+ tags = {
+ "Name" = "Terraform-Instance-0"
}
+ tags_all = {
+ "Name" = "Terraform-Instance-0"
}
+ tenancy = (known after apply)
+ user_data = (known after apply)
+ user_data_base64 = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
+ capacity_reservation_specification (known after apply)
+ cpu_options (known after apply)
+ ebs_block_device (known after apply)
+ enclave_options (known after apply)
+ ephemeral_block_device (known after apply)
+ instance_market_options (known after apply)
+ maintenance_options (known after apply)
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)

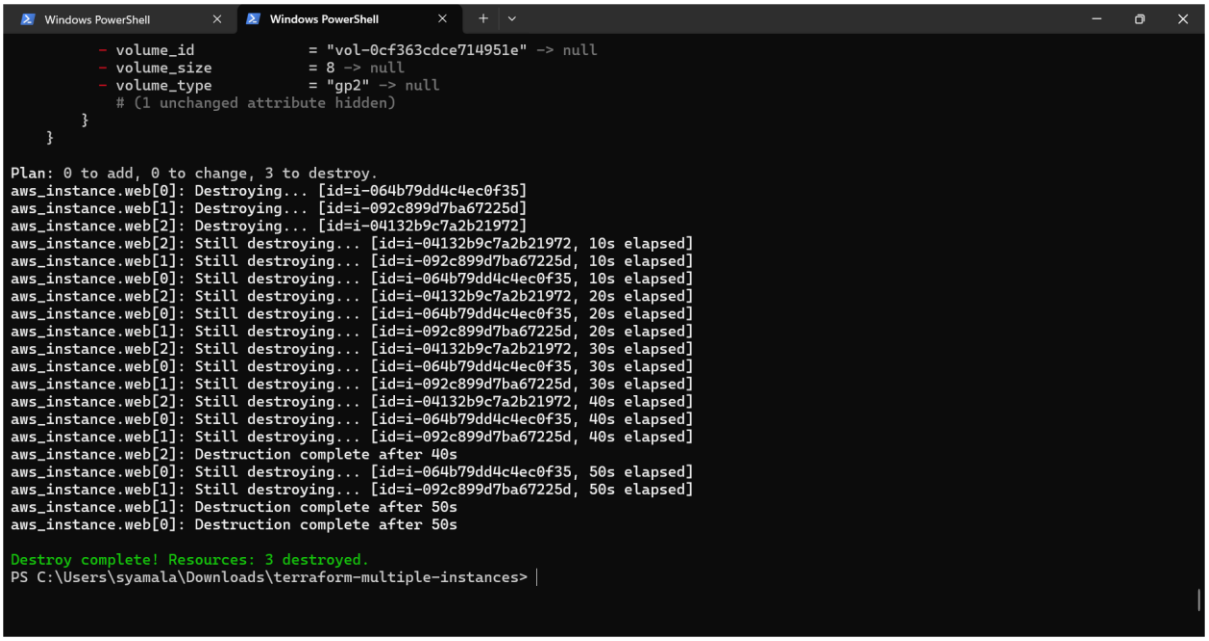

Windows PowerShell
Windows PowerShell
}
# aws_instance.web[1] will be created
+ resource "aws_instance" "web" {
+ ami = "ami-03c1aa37835df06a1"
+ arn = (known after apply)
+ associate_public_ip_address = (known after apply)
+ availability_zone = (known after apply)
+ cpu_core_count = (known after apply)
+ cpu_threads_per_core = (known after apply)
+ disable_api_stop = (known after apply)
+ disable_api_termination = (known after apply)
+ ebs_optimized = (known after apply)
+ enable_primary_ipv6 = (known after apply)
+ get_password_data = false
+ host_id = (known after apply)
+ host_resource_group_arn = (known after apply)
+ iam_instance_profile = (known after apply)
+ id = (known after apply)
+ instance_initiated_shutdown_behavior = (known after apply)
+ instance_lifecycle = (known after apply)
+ instance_state = (known after apply)
+ instance_type = "t2.micro"
+ ipv6_address_count = (known after apply)
+ ipv6_addresses = (known after apply)
+ key_name = (known after apply)
+ monitoring = (known after apply)
+ outpost_arn = (known after apply)
+ password_data = (known after apply)
+ placement_group = (known after apply)
+ placement_partition_number = (known after apply)
+ primary_network_interface_id = (known after apply)
+ private_dns = (known after apply)


Windows PowerShell
Windows PowerShell
+ "Name" = "Terraform-Instance-1"
}
+ tags_all = {
+ "Name" = "Terraform-Instance-1"
}
+ tenancy = (known after apply)
+ user_data = (known after apply)
+ user_data_base64 = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
+ capacity_reservation_specification (known after apply)
+ cpu_options (known after apply)
+ ebs_block_device (known after apply)
+ enclave_options (known after apply)
+ ephemeral_block_device (known after apply)
+ instance_market_options (known after apply)
+ maintenance_options (known after apply)
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}


Windows PowerShell
Windows PowerShell
+ subnet_id = (known after apply)
+ tags = {
+ "Name" = "Terraform-Instance-2"
}
+ tags_all = {
+ "Name" = "Terraform-Instance-2"
}
+ tenancy = (known after apply)
+ user_data = (known after apply)
+ user_data_base64 = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
+ capacity_reservation_specification (known after apply)
+ cpu_options (known after apply)
+ ebs_block_device (known after apply)
+ enclave_options (known after apply)
+ ephemeral_block_device (known after apply)
+ instance_market_options (known after apply)
+ maintenance_options (known after apply)
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)


Windows PowerShell
Windows PowerShell
+ ephemeral_block_device (known after apply)
+ instance_market_options (known after apply)
+ maintenance_options (known after apply)
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}
Plan: 3 to add, 0 to change, 0 to destroy.
aws_instance.web[2]: Creating...
aws_instance.web[0]: Creating...
aws_instance.web[1]: Creating...
aws_instance.web[2]: Still creating... [10s elapsed]
aws_instance.web[0]: Still creating... [10s elapsed]
aws_instance.web[1]: Still creating... [10s elapsed]
aws_instance.web[0]: Creation complete after 12s [id=i-064b79dd4c4ec0f35]
aws_instance.web[1]: Creation complete after 12s [id=i-092c899d7ba67225d]
aws_instance.web[2]: Creation complete after 12s [id=i-04132b9c7a2b21972]
Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
PS C:\Users\syamala\Downloads\terraform-multiple-instances>


Windows PowerShell
Windows PowerShell
+ FullyQualifiedErrorId : NoProcessFoundForGivenName,Micro
soft.PowerShell.Commands.GetProcessCommand
PS C:\Users\syamala\Downloads\terraform-multiple-instances> & "C:\Users\syamala\Downloads\terraform.exe" init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.93.0...
- Installed hashicorp/aws v5.93.0 (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.
PS C:\Users\syamala\Downloads\terraform-multiple-instances>



✅ Final Result

- ✔ Successfully completed the lab using count loop.
- ✔ All resources were created and destroyed using Terraform.