

Pulkit

50125835

Lab Exercise 16– Terraform Variables with Command Line Arguments

Objective:

Learn how to pass values to Terraform variables using command line arguments.

Prerequisites:

- Terraform installed on your machine.
- Basic knowledge of Terraform variables.

Steps:

1. Create a Terraform Directory:

```
mkdir terraform-cli-variables  
cd terraform-cli-variables
```

2. Create Terraform Configuration Files:

- Create a file named main.tf:

instance.tf

```
resource "aws_instance" "example" {  
  ami      = var.ami  
  instance_type = var.instance_type  
}
```

- Create a file named variables.tf:

variables.tf

```
variable "ami" {  
  description = "AMI ID"  
  default     = "ami-08718895af4dfa033"  
}
```

```
variable "instance_type" {
  description = "EC2 Instance Type"
  default     = "t2.micro"
}
```

3. Use Command Line Arguments:

- Open a terminal and navigate to your Terraform project directory.
- Run the terraform init command:

terraform init

- Run the terraform apply command with command line arguments to set variable values:

```
terraform plan -var="ami=ami-0522ab6e1ddcc7055" -var="instance_type=t3.micro"
```

```
PS C:\Me\DevSecOps\Terraform-demo\Exercise-11> terraform plan -var="my-ami=ami-02d26659fd82cf299" -var="my-instance-type=t3.micro"

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.web1 will be created
+ resource "aws_instance" "web1" {
  + ami                    = "ami-02d26659fd82cf299"
  + arn                   = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone      = (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)
  + force_destroy          = false
  + 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          = "t3.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)
}
```



```

PS C:\Me\DevSecOps\Terraform-demo\Exercise-11> terraform plan
var.my-ami
  Enter a value: ami-02d26659fd82cf299

var.my-instance-type
  Enter a value: t3.micro

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.web1 will be created
+ resource "aws_instance" "web1" {
  + ami                        = "ami-02d26659fd82cf299"
  + arn                       = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone          = (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)
  + force_destroy              = false
  + 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              = "t3.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)

```

- Adjust the values based on your preferences.

```

PS C:\Me\DevSecOps\Terraform-demo\Exercise-11> terraform apply -var="my-ami=ami-02d26659fd82cf299" -var="my-instance-type=t3.micro" -auto-approve

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.web1 will be created
+ resource "aws_instance" "web1" {
  + ami                        = "ami-02d26659fd82cf299"
  + arn                       = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone          = (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)
  + force_destroy              = false
  + 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              = "t3.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)
  + private_ip                  = (known after apply)
  + public_dns                  = (known after apply)
  + public_ip                   = (known after apply)

```

```

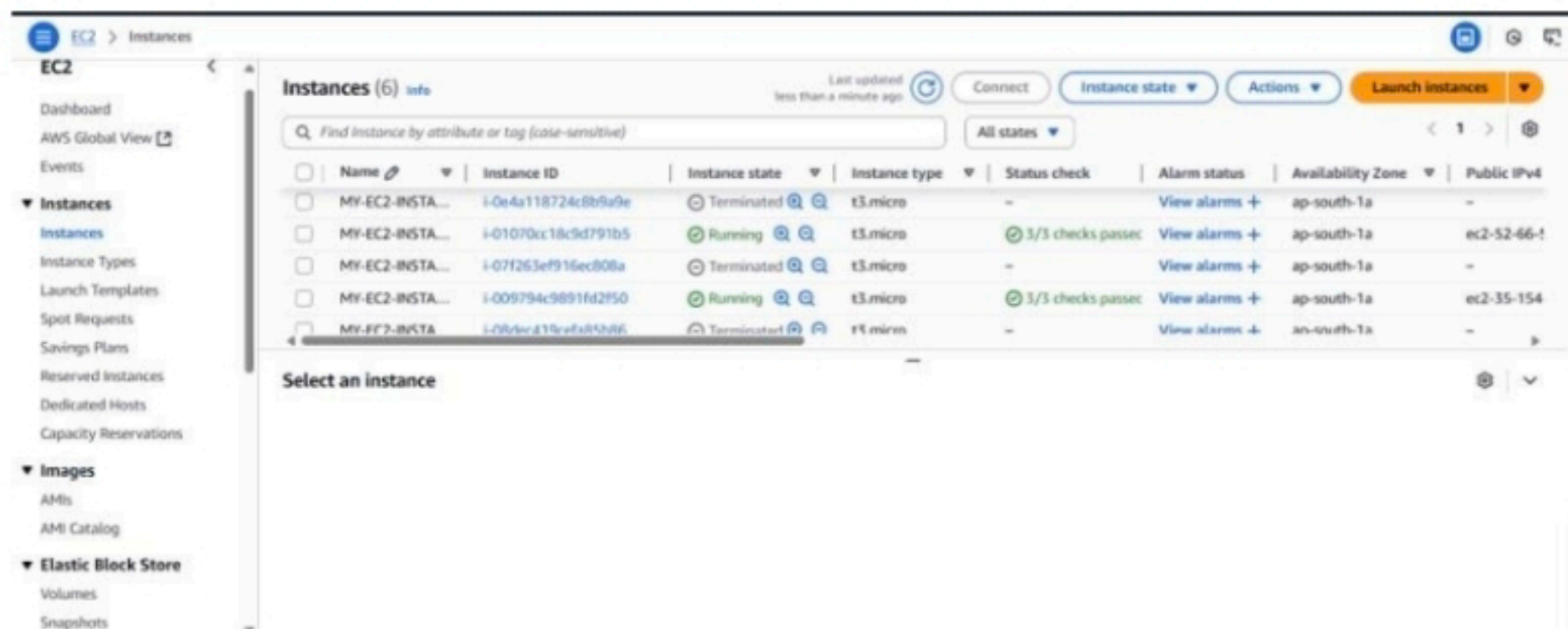
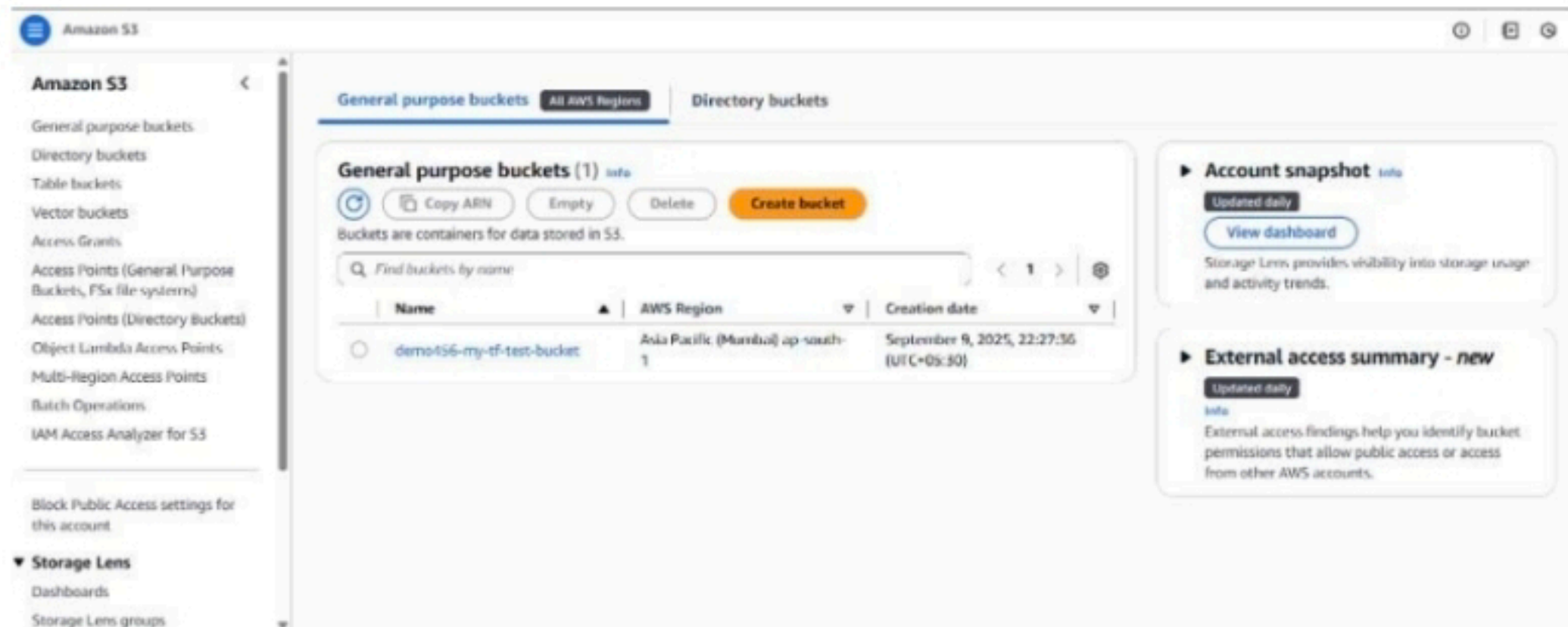
Plan: 3 to add, 0 to change, 0 to destroy.
aws_instance.web2: Creating...
aws_s3_bucket.example: Creating...
aws_instance.web1: Creating...
aws_s3_bucket.example: Creation complete after 3s [id=demo456-my-tf-test-bucket]
aws_instance.web2: Still creating... [00m10s elapsed]
aws_instance.web1: Still creating... [00m10s elapsed]
aws_instance.web1: Creation complete after 13s [id=i-01070cc18c9d791b5]
aws_instance.web2: Creation complete after 13s [id=i-009794c9891fd2f50]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
PS C:\Me\DevSecOps\Terraform-demo\Exercise-11> |

```

4. Test and Verify:

- Observe how the command line arguments dynamically set the variable values during the apply process.
- Access the AWS Management Console or use the AWS CLI to verify the creation of resources in the specified region.



5. Clean Up:

After testing, you can clean up resources:

```
terraform destroy
```

Confirm the destruction by typing yes.


```

PS C:\Me\DevSecOps\Terraform-demo\Exercise-11> terraform destroy -auto-approve
aws_s3_bucket.example: Refreshing state... [id=demo456-my-tf-test-bucket]
aws_instance.web1: Refreshing state... [id=i-01070cc18c9d791b5]
aws_instance.web2: Refreshing state... [id=i-009794c9891fd2f50]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
- destroy

Terraform will perform the following actions:

# aws_instance.web1 will be destroyed
- resource "aws_instance" "web1" {
  - ami                  = "ami-02d26659fd82cf299" -> null
  - arn                  = "arn:aws:ec2:ap-south-1:060211753450:instance/i-01070cc18c9d791b5" -> null
  - associate_public_ip_address = true -> null
  - availability_zone     = "ap-south-1a" -> null
  - disable_api_stop      = false -> null
  - disable_api_termination = false -> null
  - ebs_optimized         = false -> null
  - force_destroy         = false -> null
  - get_password_data      = false -> null
  - hibernation            = false -> null
  - id                    = "i-01070cc18c9d791b5" -> null
  - instance_initiated_shutdown_behavior = "stop" -> null
  - instance_state        = "running" -> null
  - instance_type          = "t3.micro" -> null
  - ipv6_address_count     = 0 -> null
  - ipv6_addresses        = [] -> null
  - monitoring             = false -> null
  - placement_partition_number = 0 -> null
  - primary_network_interface_id = "eni-042c1a77d92ee2b16" -> null
  - private_dns            = "ip-172-31-41-116.ap-south-1.compute.internal" -> null
  - private_ip             = "172.31.41.116" -> null
  - public_dns             = "ec2-52-66-53-21.ap-south-1.compute.amazonaws.com" -> null
  - public_ip             = "52.66.53.21" -> null
  - region                = "ap-south-1" -> null
  - secondary_private_ips  = [] -> null
  - security_groups        = [
    - "default",
  ]

```

```

Plan: 0 to add, 0 to change, 3 to destroy.
aws_s3_bucket.example: Destroying... [id=demo456-my-tf-test-bucket]
aws_instance.web1: Destroying... [id=i-01070cc18c9d791b5]
aws_instance.web2: Destroying... [id=i-009794c9891fd2f50]
aws_s3_bucket.example: Destruction complete after 1s
aws_instance.web1: Still destroying... [id=i-01070cc18c9d791b5, 00m10s elapsed]
aws_instance.web2: Still destroying... [id=i-009794c9891fd2f50, 00m10s elapsed]
aws_instance.web1: Still destroying... [id=i-01070cc18c9d791b5, 00m20s elapsed]
aws_instance.web2: Still destroying... [id=i-009794c9891fd2f50, 00m20s elapsed]
aws_instance.web1: Still destroying... [id=i-01070cc18c9d791b5, 00m30s elapsed]
aws_instance.web2: Still destroying... [id=i-009794c9891fd2f50, 00m30s elapsed]
aws_instance.web1: Still destroying... [id=i-01070cc18c9d791b5, 00m40s elapsed]
aws_instance.web2: Still destroying... [id=i-009794c9891fd2f50, 00m40s elapsed]
aws_instance.web1: Still destroying... [id=i-01070cc18c9d791b5, 00m50s elapsed]
aws_instance.web2: Still destroying... [id=i-009794c9891fd2f50, 00m50s elapsed]
aws_instance.web2: Destruction complete after 52s
aws_instance.web1: Still destroying... [id=i-01070cc18c9d791b5, 01m00s elapsed]
aws_instance.web1: Still destroying... [id=i-01070cc18c9d791b5, 01m10s elapsed]
aws_instance.web1: Destruction complete after 1m12s

Destroy complete! Resources: 3 destroyed.
PS C:\Me\DevSecOps\Terraform-demo\Exercise-11> |

```

6. Conclusion:

This lab exercise demonstrates how to use command line arguments to set variable values dynamically during the terraform apply process. It allows you to customize your Terraform deployments without modifying the configuration files directly. Experiment with different variable values and observe how command line arguments impact the infrastructure provisioning process.