Lab 3 – Creating an EC2 Virtual Machine on AWS Using Terraform (No Variables)

Creator: Sandeep Kumar Sharma

Learning Objectives

- Understand how to create an EC2 instance using Terraform.
- Learn the minimum required Terraform configuration to launch a virtual machine on AWS.
- Understand provider configuration, resource creation, and AWS authentication.
- Deploy a real EC2 instance without using variables (simple and clean).

Learning Outcome

By the end of this lab, the learner will be able to: - Write a simple Terraform file that launches an EC2 instance. - Use Terraform workflow (init → plan → apply) for VM creation. - Understand how AMIs, instance types, and AWS providers work. - Verify the EC2 instance in AWS Console.

Concept Explanation (Natural Style)

In the previous labs, we created S3 buckets. Now it's time to move into real infrastructure — the **EC2 virtual machine**.

Think of EC2 as your laptop or server running in AWS. Terraform helps you create this machine automatically.

In this lab, we are going **very clean and simple**: - No variables - No separate files - Just one `main.tf` - Only the required configuration

This is perfect for beginners to understand the core components.

Later labs will introduce variables, modules, advanced concepts — but today we stay simple.

Prerequisites

- AWS CLI configured (`aws configure`)
- Terraform installed
- IAM user with EC2 permissions

Step-by-Step Hands-On Lab

Step 1: Create a New Lab Folder

```
mkdir terraform-lab3-ec2
cd terraform-lab3-ec2
```

Step 2: Create Your Terraform File

```
touch main.tf
```

Step 3: Add Provider Configuration

Open `main.tf` and paste the following:

```
provider "aws" {
  region = "ap-south-1"  # Mumbai region
}
```

This tells Terraform to use AWS and deploy in the Mumbai region.

Step 4: Add EC2 Resource Block

Now add the EC2 instance configuration below the provider:

```
resource "aws_instance" "lab3_ec2" {
  ami           = "ami-052c08d70def0ac62"  # Amazon Linux 2 AMI (Mumbai)
  instance_type = "t2.micro"               # Free-tier eligible

  tags = {
    Name = "Terraform-Lab3-EC2"
  }
}
```

This configuration: - Uses Amazon Linux 2 - Creates a `t2.micro` instance - Adds a tag "Terraform-Lab3-EC2" so you can identify it

Important: AMI IDs are region-specific. The AMI used above is for Mumbai region.

Step 5: Initialize Terraform

```
terraform init
```

This downloads the AWS plugin so Terraform can communicate with AWS.

Step 6: Preview the Changes (Plan)

```
terraform plan
```

Terraform will show you exactly what will be created.

Step 7: Apply the Configuration (Create EC2)

```
terraform apply
```

Type **yes** when prompted.

Terraform will now: - Connect to AWS - Create your EC2 instance - Show the created resource details

---

Step 8: Validate in AWS Console

1. Login to AWS Console.
2. Go to **EC2 → Instances**.
3. You will see an instance named **Terraform-Lab3-EC2**.
4. Check:
5. Instance ID
6. Public IP
7. Status: Running

Step 9: Terminate the EC2 Instance (Optional)

To avoid unnecessary AWS cost:

```
terraform destroy
```

Type **yes** to confirm.

Terraform will remove the EC2 instance safely.

Summary

In this lab, you: - Created your first EC2 virtual machine using Terraform. - Used the provider and resource blocks. - Understood AMI IDs and instance types. - Learned how Terraform automates virtual machine creation. - Validated and destroyed the resource.

This is the foundation for upcoming labs where we will make the configuration more powerful using variables, key pairs, security groups, and VPC setups.

End of Lab 3