Lab 7 – Terraform Output Values (Beginner-Friendly Explanation)

Creator: Sandeep Kumar Sharma

Learning Objectives

- Understand what output values are in Terraform.
- Learn why outputs are important in real-world Terraform projects.
- Learn how to declare output values inside Terraform code.
- Learn how to fetch important information from Terraform-managed resources.
- Display EC2 details (public IP, instance ID, etc.) using outputs.

Learning Outcome

By the end of this lab, the learner will be able to: - Explain the purpose and role of output values in Terraform. - Create and use outputs in Terraform. - Access resource details after deployment. - Use outputs for chaining modules or CI/CD pipelines in future.

Concept Explanation (Natural Style)

Imagine you ordered a laptop online. After placing the order, you get a confirmation message with: - Order ID - Delivery date - Tracking number

These details help you track your order.

Terraform outputs work exactly the same way.

When you create AWS resources — EC2, S3, VPC — Terraform can show you important information at the end, such as: - EC2 public IP - EC2 instance ID - S3 bucket name - VPC ID - Subnet IDs

Terraform output values help you: - Pass values to other Terraform configurations - Pass values to CI/CD pipelines - Debug issues quickly - Share important IDs with teammates without opening AWS Console

Outputs are like the **summary report** of whatever Terraform created.

⭐Part 1: Project Setup

Step 1: Create Folder

```
mkdir terraform-lab7-outputs
cd terraform-lab7-outputs
```

Step 2: Create Files

```
touch main.tf
```

We will keep this simple with one file.

⭐Part 2: Write Terraform Code

main.tf

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

# Create EC2 instance
resource "aws_instance" "lab7_ec2" {
  ami           = "ami-052c08d70def0ac62"  # Amazon Linux 2
  instance_type = "t2.micro"

  tags = {
    Name = "Terraform-Lab7-EC2"
  }
}

# Output: Instance ID
output "instance_id" {
  description = "ID of the EC2 instance created"
  value       = aws_instance.lab7_ec2.id
}

# Output: Public IP
output "public_ip" {
  description = "Public IP address of EC2 instance"
  value       = aws_instance.lab7_ec2.public_ip
}
```

```
# Output: Availability Zone
output "az" {
  description = "Availability zone where EC2 is deployed"
  value       = aws_instance.lab7_ec2.availability_zone
}
```

Explanation of Outputs

🦝 `instance_id`

Shows the EC2’s unique identifier.

🦝 `public_ip`

Shows the public IP so you can SSH directly without checking AWS Console.

🦝 `availability_zone`

Shows where AWS deployed the instance.

Outputs help you avoid manually searching for details.

⭐Part 3: Initialize Terraform

```
terraform init
```

This installs AWS plugin.

⭐Part 4: Run Terraform Plan

```
terraform plan
```

Terraform will tell you: - EC2 will be created - Outputs will be displayed after apply

---

⭐Part 5: Apply the Configuration

```
terraform apply
```

Type **yes**.

After creation, Terraform will show:

```
Outputs:

instance_id = "i-0abc123xyz"
public_ip   = "15.207.10.20"
az          = "ap-south-1a"
```

This is your Terraform summary.

⭐Part 6: Validate in AWS Console

Go to: - **EC2 → Instances → Terraform-Lab7-EC2**

Confirm: - Instance ID - Public IP - Availability zone

Match with output displayed by Terraform.

⭐Part 7: Destroy (Optional)

```
terraform destroy
```

Type **yes**.

This removes the EC2 instance.

Summary

In this lab, you learned: - What output values are. - Why outputs help make Terraform more useful and professional. - How to create outputs to show EC2 instance ID, IP, and zone. - How outputs act as a final summary of deployed resources.

In the next lab, we will start building **Terraform reusable modules**, which is a major concept for real infrastructure projects.

End of Lab 7