


```
ebstf U x
ebstf > resource "aws_instance" "name" > 58 tags
1 provider "aws" {
2
3
4 }
5 resource "aws_ebs_volume" "example" {
6   availability_zone = "ap-south-1b"
7   size              = 1
8
9   tags = {
10    Name = "Web_Ebs"
11  }
12 }
13
14 resource "aws_instance" "name" {
15   ami="ami-05c179eced2eb9b5b"
16   instance_type = "t2.micro"
17   tags = {
18    Name="web2"
19  }
20 }
21
22 }
23
24 resource "aws_volume_attachment" "example" {
25   device_name = "/dev/sdh"
26   instance_id = aws_instance.name.id
27   volume_id   = aws_ebs_volume.example.id
28 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

# aws_volume_attachment.example will be created
+ resource "aws_volume_attachment" "example" {
+   device_name = "/dev/sdh"
+   id          = (known after apply)
+   instance_id = (known after apply)
+   volume_id   = (known after apply)
+ }
```

Successfully initiated termination (deletion) of i-047b7ab79873c49f3

Instances (1/1) Info

Last updated less than a minute ago

Connect Instance state Actions Launch instances

Find Instance by attribute or tag (case-sensitive) Running

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public
<input checked="" type="checkbox"/>	web2	i-0912809171e4c9d83	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-13

i-0912809171e4c9d83 (web2)

<input type="checkbox"/>	Volume ID	Device name	Volume size (GiB)	Volume State	Attachment status	Attachment time
<input checked="" type="checkbox"/>	vol-0d92cbe41104de5a3	/dev/xvda	8	In-use	Attached	2025/03/22 22:36 GMT+5:30
<input type="checkbox"/>	vol-02d5b65d4cd5144cf	/dev/sdh	1	In-use	Attached	2025/03/22 22:36 GMT+5:30

3.Creating of EDS volumes and attaching to Server.

```
ebs.tf  U x  terraform.lock.hcl  U
ebs.tf > ...
1  provider "aws" {
2    region = "ap-south-1"
3  }
4
5  # Create an EC2 instance with user data
6  resource "aws_instance" "example" {
7    ami           = "ami-0e35ddab05955cf57"
8    instance_type = "t2.micro"
9    key_name      = "key1"
10   # User data script
11   user_data = <<EOF
12   |#!/bin/bash
13   |apt update -y
14   |apt install -y apache2
15   |EOF
16 }
17
18
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.
aws_instance.example: Creating...
aws_instance.example: Still creating... [10s elapsed]
aws_instance.example: Creation complete after 15s [id=i-01f49678ff813460d]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```



4.Adding user data section to EC2 using terraform

```
ebs.tf  U X
ebs.tf > resource "aws_instance" "name" > tags > Name
1  provider "aws" {
2      region = var.region
3  }
4
5  variable "region" {
6      default = "ap-south-1"
7  }
8
9  variable "my_map" {
10     type = map(string)
11     default = {
12         ap-south-1 = "ami-0e35ddab05955cf57"
13         us-west-2 = "ami-0c94855ba95c71c8"
14         us-east-1 = "ami-084568db4383264d4"
15     }
16 }
17 resource "aws_instance" "name" {
18     ami = lookup(var.my_map, var.region)
19     instance_type = "t2.micro"
20     tags = {
21         Name = "web2"
22     }
23 }
24
25 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Terraform will perform the following actions:

```
# aws_instance.name will be created
+ resource "aws_instance" "name" {
+   ami                    = "ami-0e35ddab05955cf57"
+   arn                    = (known after apply)
+   associate_public_ip_address = (known after apply)
+   availability_zone       = (known after apply)
+   cpu_core_count          = (known after apply)
```

5.To provision instances based on their region and pick the respective AMI

```
s3.tf > resource "aws_s3_bucket_versioning" "versioning_example"
1 provider "aws" {
2   region = "ap-south-1"
3 }
4 }
5 resource "aws_s3_bucket" "example" {
6   bucket = "bucket22032025"
7 }
8 tags = {
9   Name       = "My bucket"
10  Environment = "Dev"
11 }
12 }
13 resource "aws_s3_bucket_versioning" "versioning_example" {
14   bucket = aws_s3_bucket.example.id
15   versioning_configuration {
16     status = "Enabled"
17   }
18 }
```

Plan: 2 to add, 0 to change, 0 to destroy.
aws_s3_bucket.example: Creating...
aws_s3_bucket.example: Creation complete after 9s [id=bucket22032025]
aws_s3_bucket_versioning.versioning_example: Creating...
aws_s3_bucket_versioning.versioning_example: Creation complete after 2s [id=bucket22032025]
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
DELL@TECH-SIRI MINGW64 ~/OneDrive/Desktop/terraform (master)

bucket22032025 Info

Objects Properties Permissions Metrics Management Access Points

Bucket overview

AWS Region Asia Pacific (Mumbai) ap-south-1	Amazon Resource Name (ARN) arn:aws:s3:::bucket22032025	Creation date March 22, 2025, 23:03:30 (UTC+05:30)
---	--	--

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning
Enabled

Multi-Factor authentication (MFA) delete

Creation of s3 bucket and enable versioning

Terraform GitHub repository

https://github.com/SirishaReddy1234/Terraform_.git