



HashiCorp

# Terraform



# GitHub



Amazon EC2



Amazon S3

## TASK 1:- LAUNCHING OF AWS SERVICES USING TERRAFORM

```
provider "aws" {  
    region    = "ap-south-1"  
    access_key = "my-access-key"  
    secret_key = "my-secret-key"  
}  
  
resource "aws_key_pair" "mykey11" {  
    key_name = "mykey11"  
    public_key = "my-public-key"  
}  
  
resource "aws_security_group" "terra1" {  
    name      = "terra1"  
    description = "Allow TCP inbound traffic"  
    vpc_id    = "vpc-26e3ff4e"  
  
    ingress {  
        description = "HTTP"  
        from_port   = 80  
        to_port     = 80  
        protocol    = "tcp"  
        cidr_blocks = ["0.0.0.0/0"]  
    }  
  
    ingress {  
        description = "SSH"  
        from_port   = 22
```

```
to_port    = 22

protocol   = "tcp"

cidr_blocks = ["0.0.0.0/0"]

}

egress {

    from_port = 0

    to_port = 0

    protocol = "-1"

    cidr_blocks = ["0.0.0.0/0"]

}

tags = {

    Name = "mypratik"

}

}
```

```
resource "aws_instance" "mypratik" {

    ami = "ami-0447a12f28fdbb066"

    instance_type = "t2.micro"

    key_name = "mykey11"

    security_groups = ["terra1"]

    user_data = <<-EOF

        #!/bin/bash

        sudo yum install httpd -y

        sudo systemctl start httpd

        sudo systemctl enable httpd

        sudo yum install git -y

    >>>

}
```

```
mkfs.ext4 /dev/df1
```

```
mount /dev/df1 /var/www/html
```

```
cd /var/www/html
```

```
git clone https://github.com/Pratikkohad1999/multicloud.git
```

```
EOF
```

```
tags = {
```

```
    Name = "mypratik"
```

```
}
```

```
}
```

```
resource "aws_ebs_volume" "my lw pendrive" {
```

```
    availability_zone = aws_instance.mypratik.availability_zone
```

```
    size              = 1
```

```
tags = {
```

```
    Name = "my lw pendrive"
```

```
}
```

```
}
```

```
resource "aws_volume_attachment" "EBSattach" {
```

```
    device_name = "/dev/sdh"
```

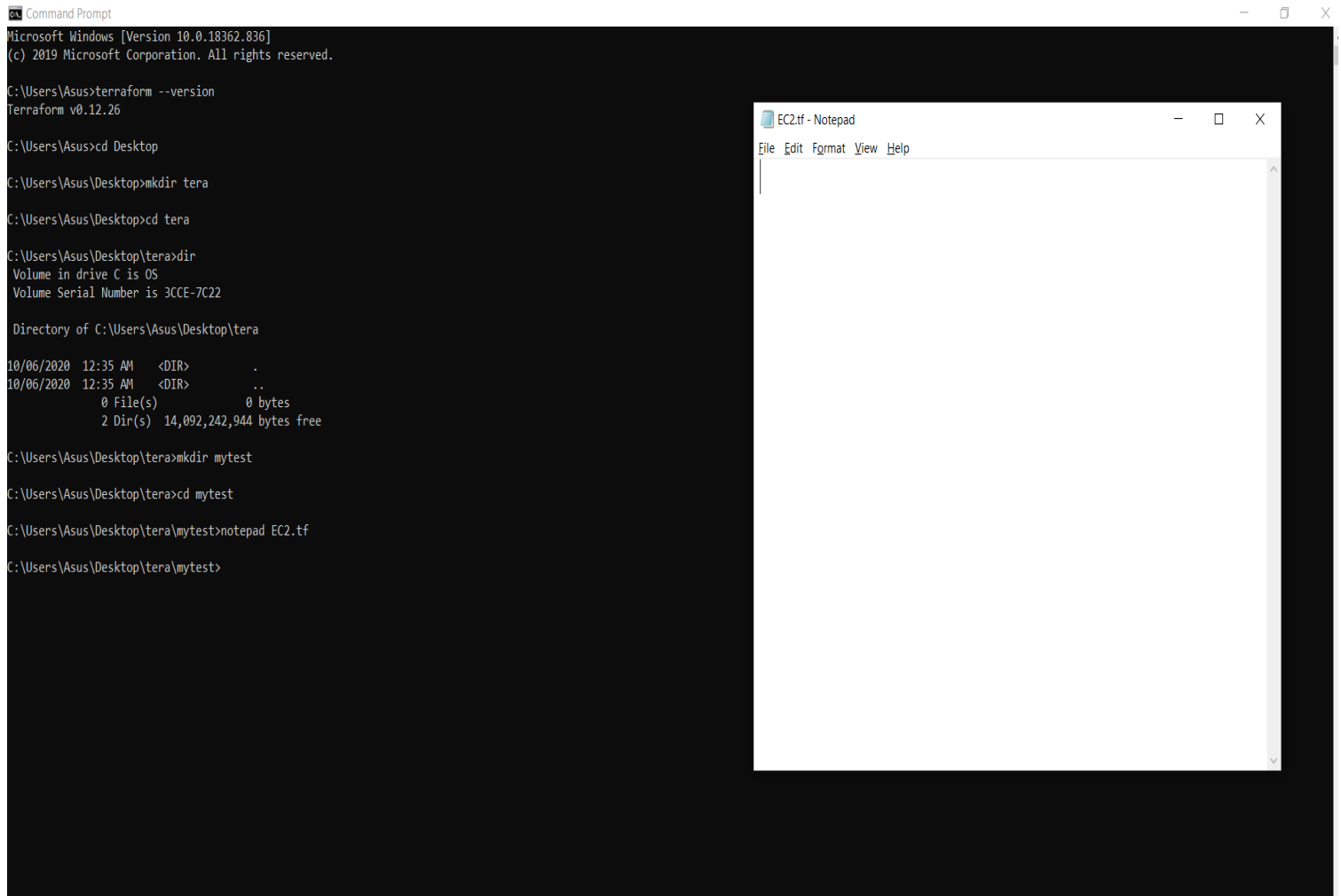
```
    volume_id   = aws_ebs_volume.my lw pendrive.id
```

```
    instance_id = aws_instance.mypratik.id
```

```
}
```

```
resource "aws_s3_bucket" "terra-s1" {  
  
    bucket = "terra-s1"  
  
}
```

```
resource "aws_s3_bucket_public_access_block" "access" {  
  
    bucket = "${aws_s3_bucket.terra-s1.id}"  
  
    block_public_acls = true  
  
    block_public_policy = true  
  
}
```



```

}
aws_instance.web (remote-exec): perl-Git.noarch 0:2.23.3-1.amzn2.0.1
aws_instance.web (remote-exec): perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2
aws_instance.web (remote-exec): php-cli.x86_64 0:5.4.16-46.amzn2.0.2
aws_instance.web (remote-exec): php-common.x86_64 0:5.4.16-46.amzn2.0.2
resource "aws_instance.web" (remote-exec): Complete!
aws_instance.web (remote-exec): Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
}
aws_instance.web: Creation complete after 52s [id=i-0c580a4c218ea24b5]
}
null_resource.nulllocal2: Creating...
aws_ebs_volume.esb1: Creating...
null_resource.nulllocal2: Provisioning with 'local-exec'...
null_resource.nulllocal2 (local-exec): Executing: ["cmd" "/C" "echo 13.234.17.212 > publicip.txt"]
resource "null_resource.nulllocal2": Creation complete after 0s [id=7155686714059548910]
aws_ebs_volume.esb1: Still creating... [10s elapsed]
aws_ebs_volume.esb1: Creation complete after 11s [id=vol-05450ac36a13a2c5d]
depends_on aws_volume_attachment.ebs_att: Creating...
aws_volume_attachment.ebs_att: Still creating... [10s elapsed]

```

Command Prompt - terraform apply -auto-approve

```

null_resource.nullremote3 (remote-exec): OS type: Linux
null_resource.nullremote3 (remote-exec): Block size=4096 (log=2)
null_resource.nullremote3 (remote-exec): Fragment size=4096 (log=2)
null_resource.nullremote3 (remote-exec): Stride=0 blocks, Stripe width=0 blocks
null_resource.nullremote3 (remote-exec): 65536 inodes, 262144 blocks
null_resource.nullremote3 (remote-exec): 13107 blocks (5.00%) reserved for the super user
null_resource.nullremote3 (remote-exec): First data block=0
null_resource.nullremote3 (remote-exec): Maximum filesystem blocks=268435456
null_resource.nullremote3 (remote-exec): 8 block groups
null_resource.nullremote3 (remote-exec): 32768 blocks per group, 32768 fragments per group
null_resource.nullremote3 (remote-exec): 8192 inodes per group
null_resource.nullremote3 (remote-exec): Superblock backups stored on blocks:
null_resource.nullremote3 (remote-exec): 32768, 98304, 163840, 229376

null_resource.nullremote3 (remote-exec): Allocating group tables: done
null_resource.nullremote3 (remote-exec): Writing inode tables: done
null_resource.nullremote3 (remote-exec): Creating journal (8192 blocks): done
null_resource.nullremote3 (remote-exec): Writing superblocks and filesystem accounting information: done

null_resource.nullremote3 (remote-exec): Cloning into '/var/www/html'...

```

```
Command Prompt - terraform destroy -auto-approve
aws_instance.web: Refreshing state... [id=i-0c580a4c218ea24b5]
null_resource.nulllocal2: Refreshing state... [id=7155686714059548910]
aws_ebs_volume.esb1: Refreshing state... [id=vol-05450ac36a13a2c5d]
aws_volume_attachment.ebs_att: Refreshing state... [id=vai-4276427842]
null_resource.nullremote3: Refreshing state... [id=8374154258520146834]
null_resource.nulllocal1: Refreshing state... [id=8902102633353659582]
null_resource.nulllocal1: Destroying... [id=8902102633353659582]
null_resource.nulllocal2: Destroying... [id=7155686714059548910]
null_resource.nulllocal2: Destruction complete after 0s
null_resource.nulllocal1: Destruction complete after 0s
null_resource.nullremote3: Destroying... [id=8374154258520146834]
null_resource.nullremote3: Destruction complete after 0s
aws_volume_attachment.ebs_att: Destroying... [id=vai-4276427842]
aws_volume_attachment.ebs_att: Still destroying... [id=vai-4276427842, 10s elapsed]
aws_volume_attachment.ebs_att: Still destroying... [id=vai-4276427842, 20s elapsed]
aws_volume_attachment.ebs_att: Still destroying... [id=vai-4276427842, 30s elapsed]
aws_volume_attachment.ebs_att: Still destroying... [id=vai-4276427842, 40s elapsed]
aws_volume_attachment.ebs_att: Destruction complete after 41s
aws_ebs_volume.esb1: Destroying... [id=vol-05450ac36a13a2c5d]
aws_ebs_volume.esb1: Destruction complete after 1s
aws_instance.web: Destroying... [id=i-0c580a4c218ea24b5]
```

Services ▾ Resource Groups ▾ 🔍

Pratik ▾ Mumbai ▾ Support ▾

New EC2 Experience

Tell us what you think

Launch Instance ▾

Connect

Actions ▾

Tags

Reports

Limits

▼ INSTANCES

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts New

Capacity Reservations

▼ IMAGES

AMIs

Bundle Tasks

▼ ELASTIC BLOCK

Filter by tags and attributes or search by keyword

1 to 2 of 2

<input type="checkbox"/>	Name ▾	Instance ID ▾	Instance Type ▾	Availability Zone ▾	Instance State ▾	Status Checks ▾	Alarm Status	Public DNS (IPv4) ▾	IPv4 Public IP ▾	IPv6 Public IP ▾
<input checked="" type="checkbox"/>	mypratik	i-095d08262fa300a5f	t2.micro	ap-south-1a	running	Initializing	None	ec2-35-154-147-215.ap...	35.154.147.215	-
<input type="checkbox"/>	Linux O S	i-004b9b1604e73a29c	t2.micro	ap-south-1a	stopped		None		-	-





**Congratulations on your success**

Welcome to the First Task of Hybrid multi cloud

LinuxWorld

Summary on my task:-

- 1.Create a key-Pair.
- 2.Create one security group allowing port number 80.
- 3.Launch the EC2 instance by using the key-pair and the security group that we have created.
- 4.Launch one volume and mount this volume into `/var/www/html`.
- 5.Install httpd, php, and git
- 6.Developer have uploaded the code into github repo and the repo has some images.
- 7.Copy the github repo code into `/var/www/html`.
- 8.Create S3 bucket and copy/deploy the image from github repo into the S3 bucket and change the permission to public readable.
- 9.Create the cloudfront using S3 bucket(which contains the image) and use the cloudfront URL to update in the code in `/var/www/html`.

**Thanks for reading!!!**