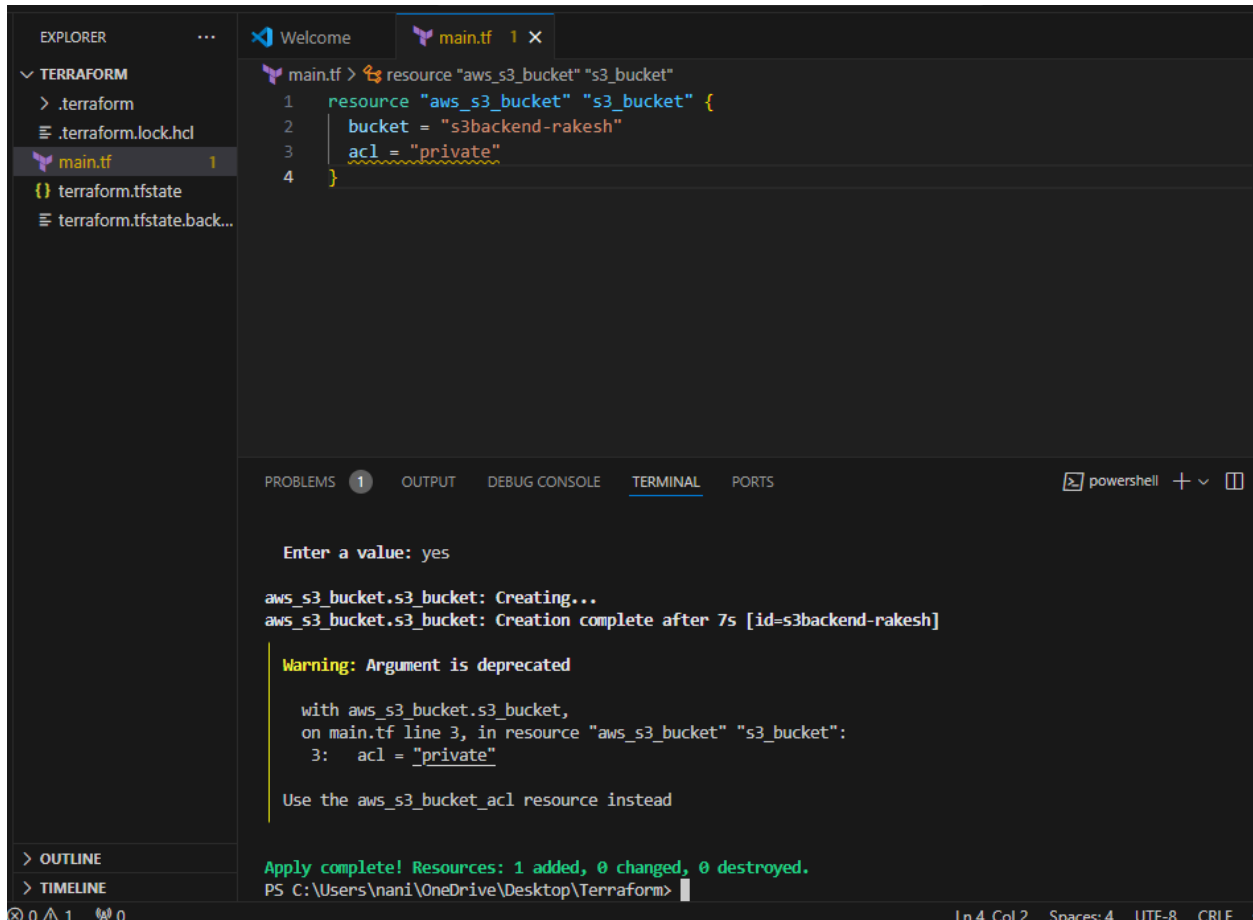


# JENKINS5&6

- 1) Watch terraform-05 video.
- 2) Execute the script shown in video.



The screenshot shows a Visual Studio Code editor with a Terraform script in a file named `main.tf`. The script defines an `aws_s3_bucket` resource named `s3_bucket` with the following configuration:

```
1 resource "aws_s3_bucket" "s3_bucket" {  
2   bucket = "s3backend-rakesh"  
3   acl = "private"  
4 }
```


The terminal output shows the execution of the script, including a warning about a deprecated argument and a successful completion message.

```
Enter a value: yes  
  
aws_s3_bucket.s3_bucket: Creating...  
aws_s3_bucket.s3_bucket: Creation complete after 7s [id=s3backend-rakesh]  
  
Warning: Argument is deprecated  
with aws_s3_bucket.s3_bucket,  
on main.tf line 3, in resource "aws_s3_bucket" "s3_bucket":  
3:   acl = "private"  
  
Use the aws_s3_bucket_acl resource instead  
  
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.  
PS C:\Users\nani\OneDrive\Desktop\Terraform>
```

General purpose buckets (4) [Info](#) [All AWS Regions](#) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
<a href="#">rakeshpagidimarri.shop</a>	Europe (Paris) eu-west-3	<a href="#">View analyzer for eu-west-3</a>	November 18, 2024, 17:42:19 (UTC+05:30)
<a href="#">load-logs-1</a>	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	November 20, 2024, 17:55:52 (UTC+05:30)
<a href="#">my-s3-naresh</a>	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	November 26, 2024, 15:54:03 (UTC+05:30)
<a href="#">s3backend-rakesh</a>	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	December 9, 2024, 16:51:06 (UTC+05:30)



The screenshot shows a VS Code editor with a file named `main.tf` open. The file contains Terraform configuration for an AWS S3 bucket and a DynamoDB table. The S3 bucket resource is named `aws_s3_bucket` and has a bucket name of `s3backend-rakesh` and an ACL of `private`. The DynamoDB table resource is named `aws_dynamodb_table` and has a name of `terraform-state-lock-dynamo`, a hash key of `LockID`, and a read capacity of 20. The table also has an attribute named `LockID` of type `S`.

The terminal output shows the Terraform command being executed: `terraform apply`. The output indicates that the `aws_s3_bucket_acl` resource is being used instead of the `aws_s3_bucket` resource. The output also shows the creation of the `aws_dynamodb_table.dynamodb-terraform-state-lock` resource, which is still creating after 10 seconds. The output then shows the creation of the `aws_dynamodb_table.dynamodb-terraform-state-lock` resource, which is complete after 11 seconds. The output ends with the message: `Apply complete! Resources: 1 added, 0 changed, 0 destroyed.`

**DynamoDB**

Dashboard  
Tables  
Explore items  
 PartiQL editor  
Backups  
Exports to S3  
Imports from S3  
Integrations [New](#)

**Tables (1)** Info

Find tables Any tag key Any tag value 1 match < 1 >

<input type="checkbox"/>	Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection
<input type="checkbox"/>	<a href="#">terraform-state-lock-dynamo</a>	Active	LockID (S)	-	0	0	Off

Amazon S3 Buckets s3backend-rakesh

**s3backend-rakesh** Info

Objects | Metadata - Preview | Properties | Permissions | Metrics | Management | Access Points

---

**Objects (1)** Info

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Name	Type	Last modified	Size	Storage class
terraform.tfstate	tfstate	December 9, 2024, 17:14:45 (UTC+05:30)	4.5 KB	Standard

The screenshot shows a VS Code editor with a Terraform configuration file `main.tf` and its application output in the terminal.

```
resource "aws_dynamodb_table" "dynamodb-terraform-state-lock" {
  attribute {
    name = "LockID"
    type = "S"
  }
  terraform {
    backend "s3" {
      bucket = "s3backend-rakesh"
      dynamodb_table = "terraform-state-lock-dynamo"
      key = "terraform.tfstate"
      region = "us-east-1"
    }
  }
}
```

The terminal output shows the following messages:

```
Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.

Warning: Argument is deprecated
with aws_s3_bucket.s3_bucket,
on main.tf line 3, in resource "aws_s3_bucket" "s3_bucket":
3:   acl = "private"

Use the aws_s3_bucket_acl resource instead
(and one more similar warning elsewhere)

Releasing state lock. This may take a few moments...

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
PS C:\Users\nani\OneDrive\Desktop\Terraform>
```

The screenshot shows a VS Code editor with a Terraform configuration file `main.tf` and its application output in the terminal.

```
resource "aws_instance" "s3-test-server" {
  ami = "ami-0453ec754f44f9a4a"
  instance_type = "t2.micro"
  user_data = <<-EOF
  #!/bin/bash
  yum update -y
  yum install -y httpd
  systemctl start httpd
  systemctl enable httpd
  echo "<h1>Welcome to My Web Server</h1>" > /var/www/html/index.html
  EOF
  tags = {
    name = "s3-test-server"
  }
}
```

The terminal output shows the following messages:

```
+ root_block_device (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.

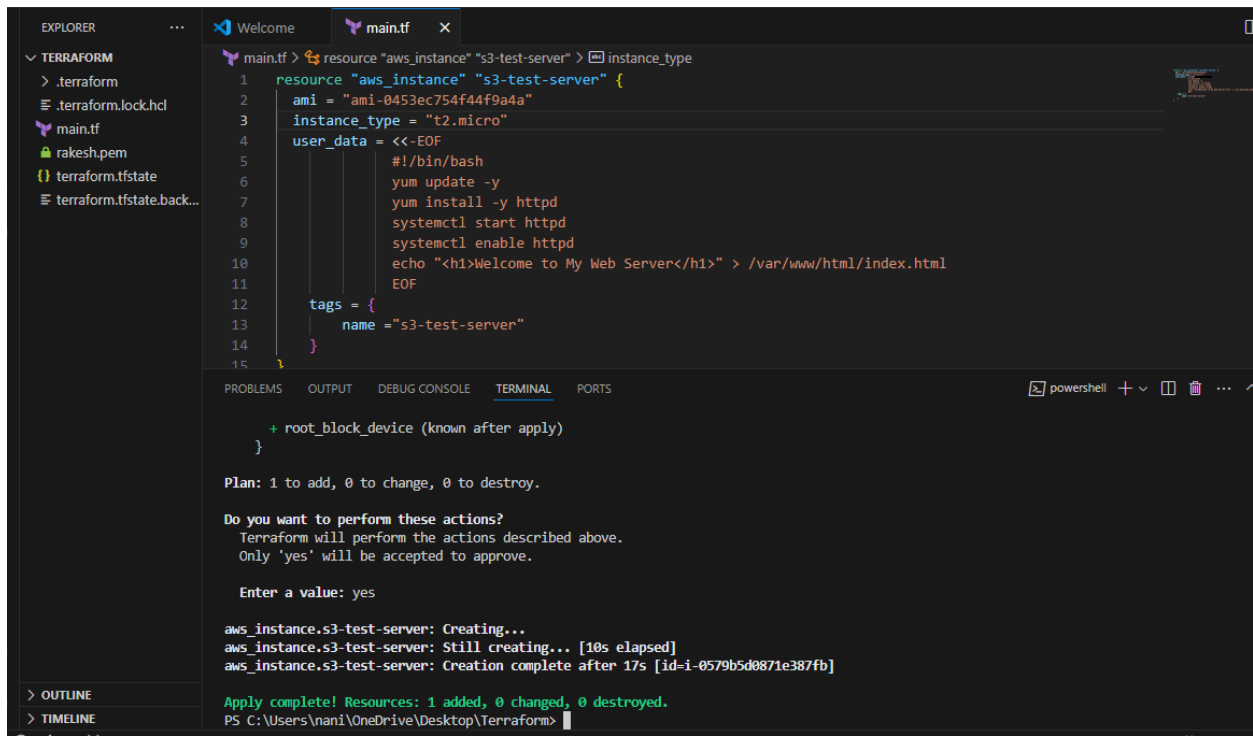
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_instance.s3-test-server: Creating...
aws_instance.s3-test-server: Still creating... [10s elapsed]
aws_instance.s3-test-server: Creation complete after 17s [id=i-0579b5d0871e387fb]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\nani\OneDrive\Desktop\Terraform>
```

3) Create one ec2 instance with httpd installed using terraform script.



The screenshot shows the VS Code interface with a Terraform configuration file named `main.tf` open. The configuration defines an AWS instance resource `s3-test-server` with the following details:

- Resource: `aws_instance "s3-test-server"`
- AMI: `ami-0453ec754f44f9a4a`
- Instance Type: `t2.micro`
- User Data (EOF):

```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<h1>Welcome to My Web Server</h1>" > /var/www/html/index.html
```
- Tags: `{ name = "s3-test-server" }`

The terminal output shows the Terraform plan and apply process:

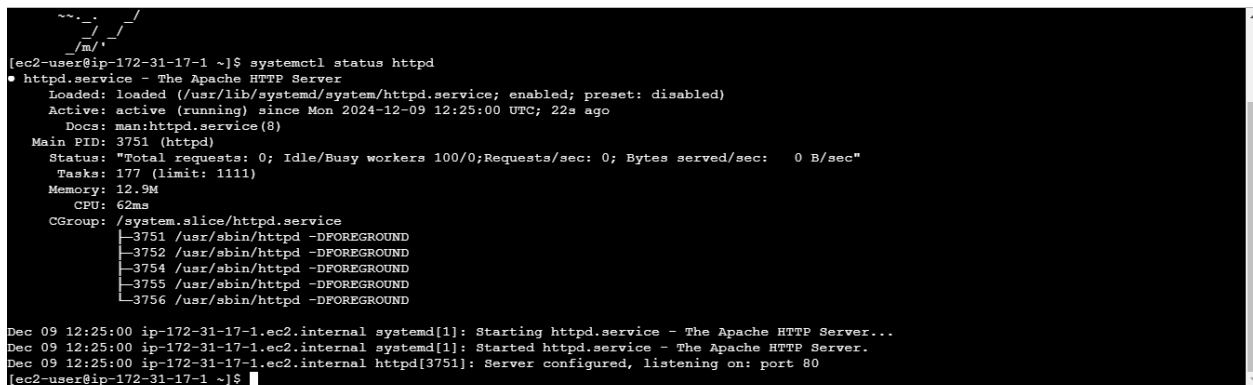
```
Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.s3-test-server: Creating...
aws_instance.s3-test-server: Still creating... [10s elapsed]
aws_instance.s3-test-server: Creation complete after 17s [id=i-0579b5d0871e387fb]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\nan1\OneDrive\Desktop\Terraform>
```



The terminal output shows the status of the `httpd.service` and its logs:

```
[ec2-user@ip-172-31-17-1 ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Mon 2024-12-09 12:25:00 UTC; 22s ago
     Docs: man:httpd.service(8)
  Main PID: 3751 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
    Tasks: 177 (limit: 1111)
  Memory: 12.9M
     CPU: 62ms
  CGroup: /system.slice/httpd.service
          └─3751 /usr/sbin/httpd -DFOREGROUND
            └─3752 /usr/sbin/httpd -DFOREGROUND
              └─3754 /usr/sbin/httpd -DFOREGROUND
                └─3755 /usr/sbin/httpd -DFOREGROUND
                  └─3756 /usr/sbin/httpd -DFOREGROUND

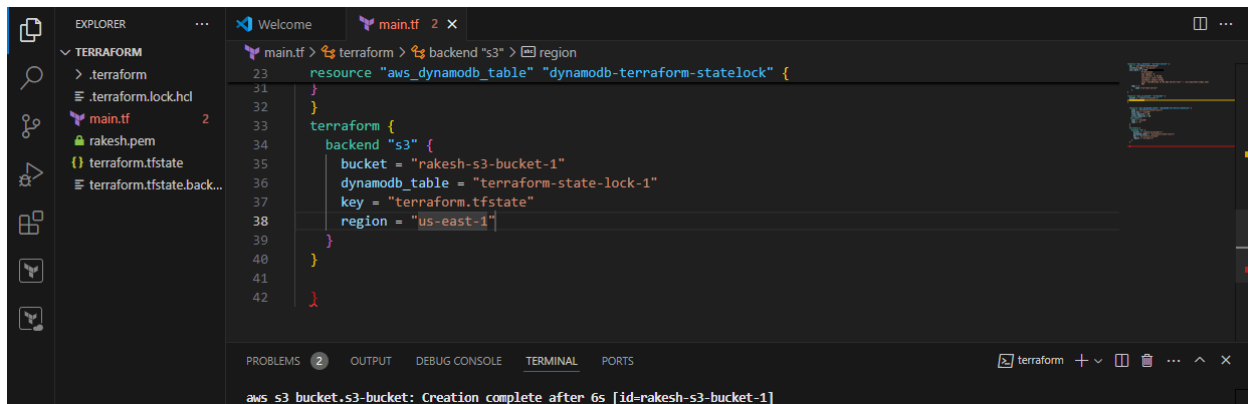
Dec 09 12:25:00 ip-172-31-17-1.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Dec 09 12:25:00 ip-172-31-17-1.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Dec 09 12:25:00 ip-172-31-17-1.ec2.internal httpd[3751]: Server configured, listening on: port 80
[ec2-user@ip-172-31-17-1 ~]$
```

i-0579b5d0871e387fb

PublicIPs: 54.145.224.219 PrivateIPs: 172.31.17.1

4) Setup s3 as backend to the task 3.



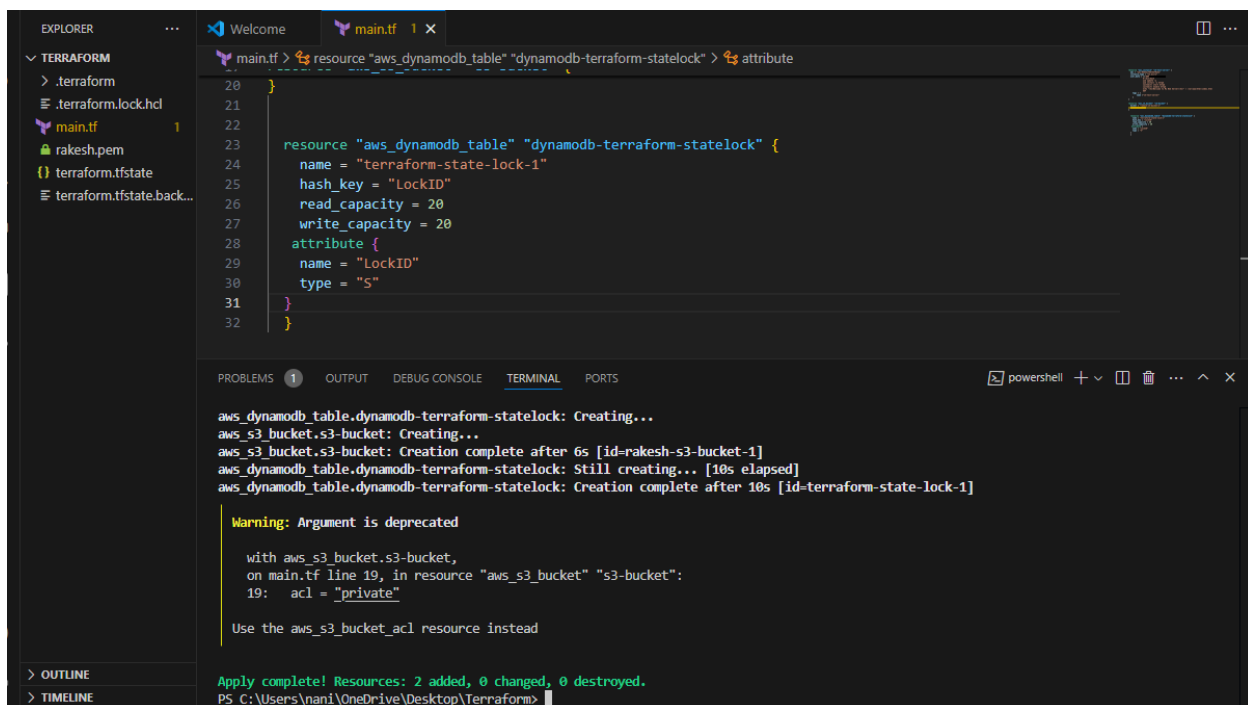


The screenshot shows the VS Code interface with the Explorer pane on the left displaying the Terraform project structure. The main editor shows the `main.tf` file with the following configuration:

```
23 resource "aws_dynamodb_table" "dynamodb-terraform-statelock" {
31 }
32 }
33 terraform {
34   backend "s3" {
35     bucket = "rakesh-s3-bucket-1"
36     dynamodb_table = "terraform-state-lock-1"
37     key = "terraform.tfstate"
38     region = "us-east-1"
39   }
40 }
41
42 }
```

The bottom status bar indicates the execution progress: `aws s3 bucket.s3-bucket: Creation complete after 6s [id=rakesh-s3-bucket-1]`.

5) Setup dynamo db locking for task3.



The screenshot shows the VS Code interface with the Explorer pane on the left. The main editor shows the `main.tf` file with the following configuration:

```
20 }
21
22
23 resource "aws_dynamodb_table" "dynamodb-terraform-statelock" {
24   name = "terraform-state-lock-1"
25   hash_key = "LockID"
26   read_capacity = 20
27   write_capacity = 20
28   attribute {
29     name = "LockID"
30     type = "S"
31   }
32 }
```

The bottom terminal pane shows the execution progress:

```
aws_dynamodb_table.dynamodb-terraform-statelock: Creating...
aws_s3_bucket.s3-bucket: Creating...
aws_s3_bucket.s3-bucket: Creation complete after 6s [id=rakesh-s3-bucket-1]
aws_dynamodb_table.dynamodb-terraform-statelock: Still creating... [10s elapsed]
aws_dynamodb_table.dynamodb-terraform-statelock: Creation complete after 10s [id=terraform-state-lock-1]
```

A warning message is displayed:

```
Warning: Argument is deprecated

with aws_s3_bucket.s3-bucket,
on main.tf line 19, in resource "aws_s3_bucket" "s3-bucket":
19:   acl = "private"

Use the aws_s3_bucket_acl resource instead
```

The terminal also shows the final status: `Apply complete! Resources: 2 added, 0 changed, 0 destroyed.` and the command prompt: `PS C:\Users\nani\OneDrive\Desktop\Terraform>`

The top part of the image shows a VS Code editor with a Terraform configuration file. The configuration defines an AWS DynamoDB table named 'dynamodb-terraform-statelock' with a backend 's3' pointing to a bucket 'rakesh-s3-bucket-1'. The table has a partition key 'terraform.tfstate' and is located in the 'us-east-1' region.

```
23 resource "aws_dynamodb_table" "dynamodb-terraform-statelock" {
31 }
32 }
33 terraform {
34   backend "s3" {
35     bucket = "rakesh-s3-bucket-1"
36     dynamodb_table = "terraform-state-lock-1"
37     key = "terraform.tfstate"
38     region = "us-east-1"
39   }
40 }
41 }
42 }
```

The bottom part of the image shows the AWS Management Console 'DynamoDB' console. It displays a table with 2 matches for 'terraform-state-lock-1' and 'terraform-state-lock-dynamo'. Both tables are 'Active' and have 'LockID (S)' as the partition key.

Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protect
<a href="#">terraform-state-lock-1</a>	Active	LockID (S)	-	0	0	Off
<a href="#">terraform-state-lock-dynamo</a>	Active	LockID (S)	-	0	0	Off

- 6) Watch terraform-06 video.
- 7) Execute the script shown in video.
- 8) Provision ec2,s3 and vpc using Terraform modules.
- 9) Provision ec2 for 3 different environments (Dev, Staging and Prod) using terraform workspaces.