Advance DevOps Practical Examination Case Study Assignment

Prajjwal Pandey
D15A / 33

13. Basic Infrastructure Management with Terraform

● **Concepts Used:** Terraform, AWS EC2, and S3.

● **Problem Statement:** "Use Terraform to create an EC2 instance and an S3 bucket. Store the EC2 instance's IP address in the S3 bucket."

● **Tasks:**
○ Write a simple Terraform script to provision an EC2 instance and an S3 bucket.

○ Use Terraform outputs to extract the EC2 instance's IP address.

○ Store the IP address in a text file in the S3 bucket.

1. Introduction

● **Case Study Overview:**
The case study focuses on implementing basic infrastructure management using Terraform, where the objective is to create an EC2 instance and an S3 bucket on AWS. Additionally, the IP address of the EC2 instance is extracted and stored in the S3 bucket, showcasing the automation of cloud infrastructure deployment and resource management.

● **Key Feature and Application:**
A key feature of this case study is the integration of Terraform with AWS services like EC2 and S3. Terraform simplifies the provisioning of infrastructure by using code to automate resource creation. Its practical use lies in managing infrastructure as code (IaC), allowing for reproducibility, scalability, and faster deployments in real-world environments where managing cloud resources manually would be tedious and error-prone.

● **Third-Year Project Integration (Optional):**

My OCR project can relate to this Terraform case study through the use of cloud infrastructure for deployment and scalability. Here's how:

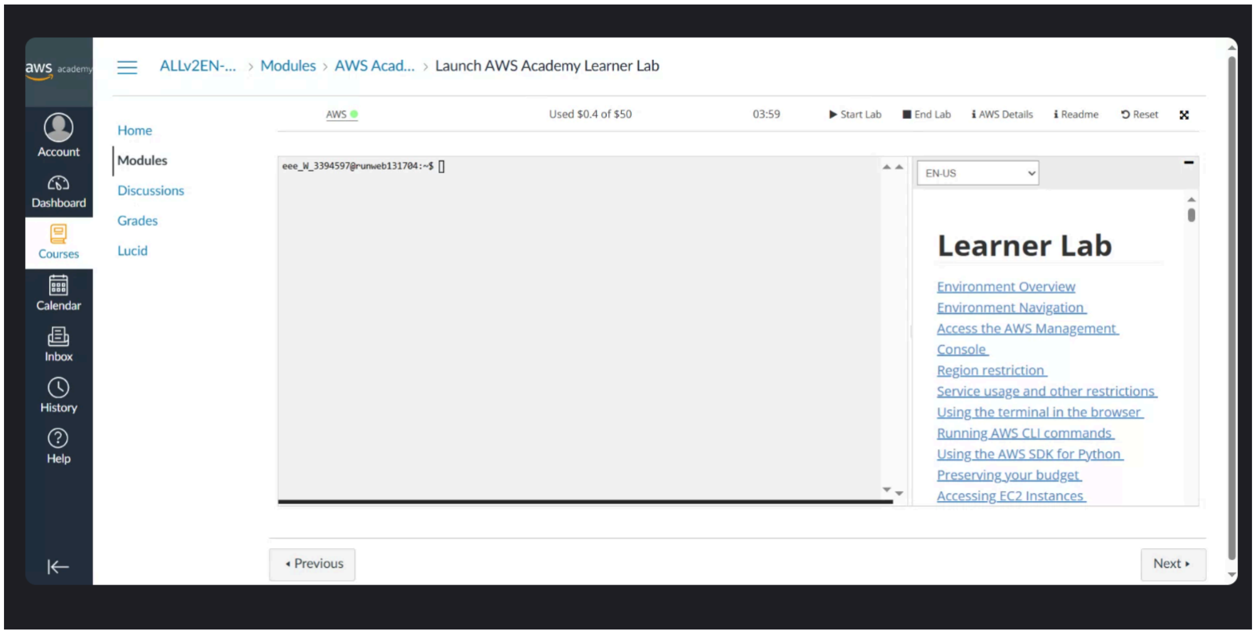
1. Cloud Infrastructure: Just like the EC2 instance and S3 bucket in the case study, you can use EC2 to deploy and run your OCR application. Instead of running it locally, your OCR processing could happen on an EC2 instance, which can be scaled as needed depending on the volume of data.
2. Storage of Data: Your OCR project likely processes text or images, which could be stored in S3 buckets, similar to how the case study stores the EC2 instance's IP. You could use an S3 bucket to store images to be processed or the OCR output data, enabling scalability and better data management.
3. Automation: Just like the Terraform script automates the creation of infrastructure, you could automate the deployment of your OCR application using Terraform. This ensures your application can be easily deployed in multiple environments or updated as required without manual intervention.

In essence, integrating Terraform with your OCR project would allow you to automate the deployment of the infrastructure (EC2 for running the OCR engine, S3 for storage) and ensure that it scales easily in the cloud, much like the automation demonstrated in the case study.

2. Step-by-Step Explanation

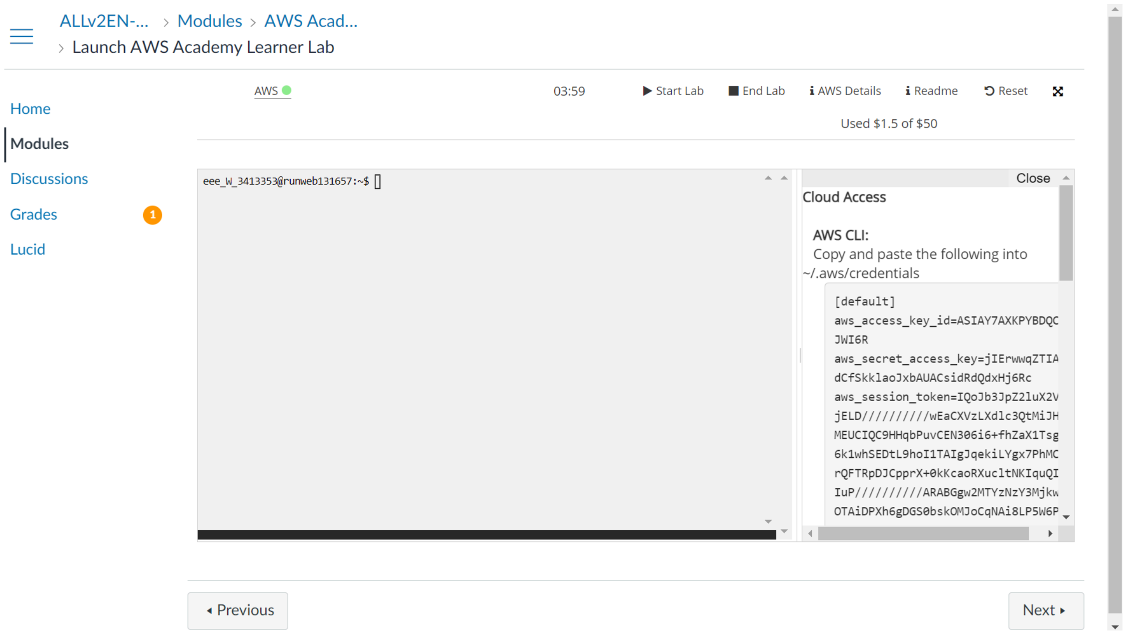
A. Creating EC2 instance using Terraform (IaC)

1. Start lab on aws learners lab



```
eee_W_3394597@runweb131704:~$
```

2. Click on show details then click on aws CLI



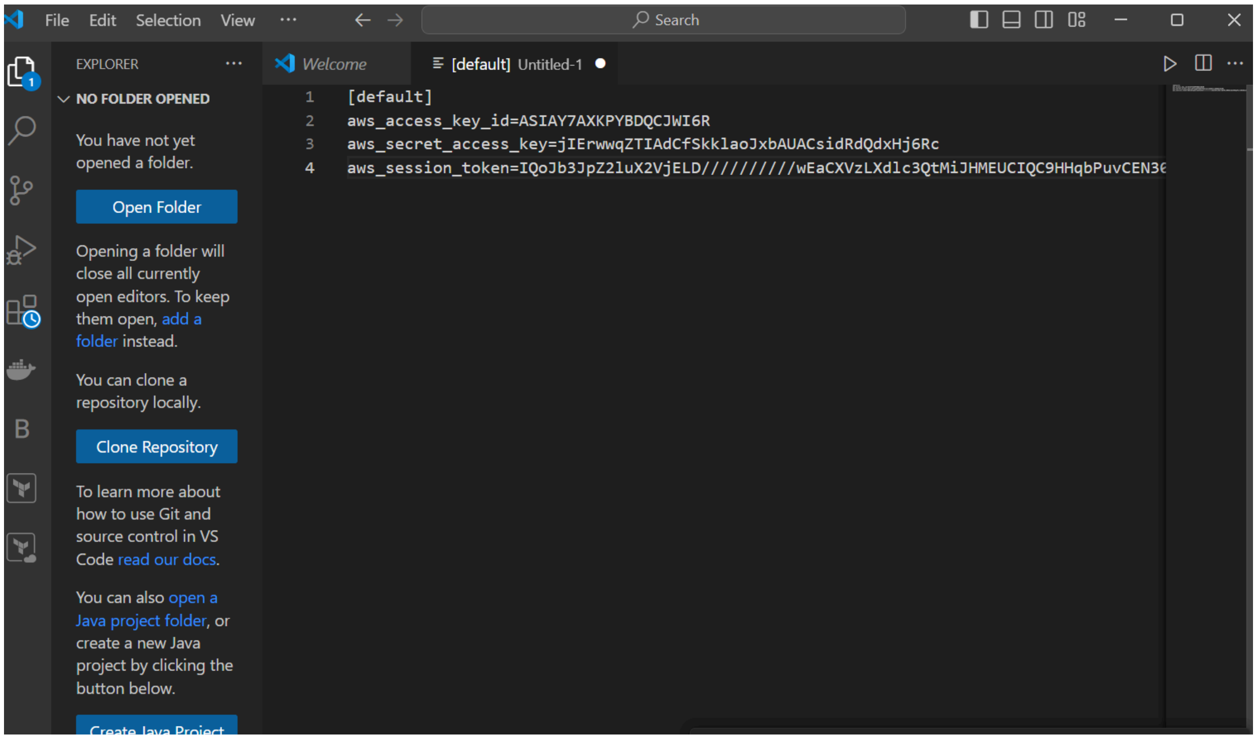
```
eee_W_3413353@runweb131657:~$
```

Cloud Access

AWS CLI:
Copy and paste the following into ~/.aws/credentials

```
[default]
aws_access_key_id=ASIAY7AXKPYBDQC
JWI6R
aws_secret_access_key=jIErwwqZTIA
dCfSkklaoJxbAUACsidRdQdxHj6Rc
aws_session_token=IQoJb3JpZ2luX2V
jELD//////////wEaCXVzLXdlc3QtMiJH
MEUCIQC9HHqbPuvCEN306i6+fhZaX1Tsg
6k1whSEDtL9hoI1TAIgJqekiLYgx7PhMC
rQFTRpDJCpprX+0kKcaoRXucltNKIquQI
IuP//////////ARABGgw2MTYzNzY3Mjkw
OTAiDPXh6gDGS0bskOMJoCqNAi8LP5W6P
```

3. Copy and paste AWS CLI in vsc



You have not yet opened a folder.

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

You can clone a repository locally.

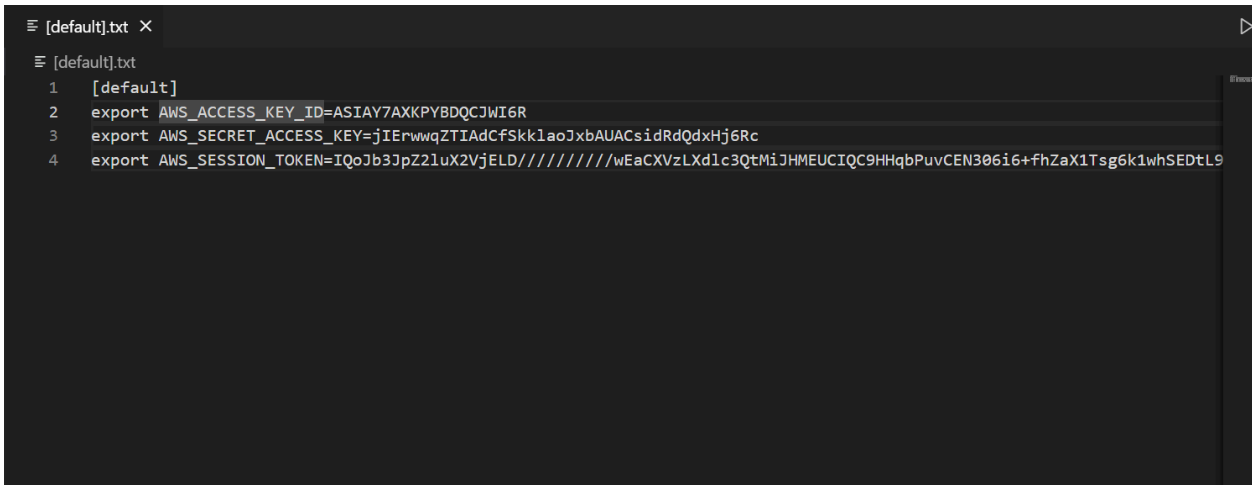
To learn more about how to use Git and source control in VS Code read our docs.

You can also open a Java project folder, or create a new Java project by clicking the button below.

```
[default]
aws_access_key_id=ASIAY7AXKPYBDQCJWI6R
aws_secret_access_key=jIErwwqZTIAdCfSkklaoJxbAUACsidRdQdxHj6Rc
aws_session_token=IQoJb3JpZ2luX2VjELD//////////wEaCXVzLXdlc3QtMiJHMEUCIQC9HHqbPuvCEN3
```

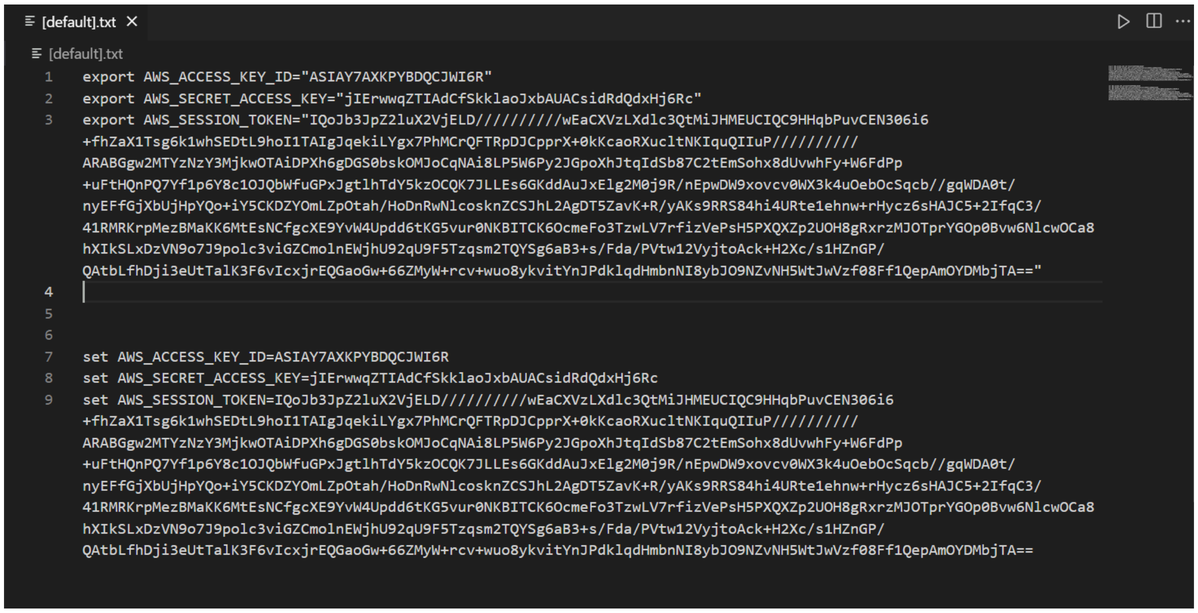
4. Make a folder on desktop as TERRAFORM-WORKSHOP and save this plain text

5. Transform LHS to uppercase.
6. Write the keyword export before AWS in all three expressions



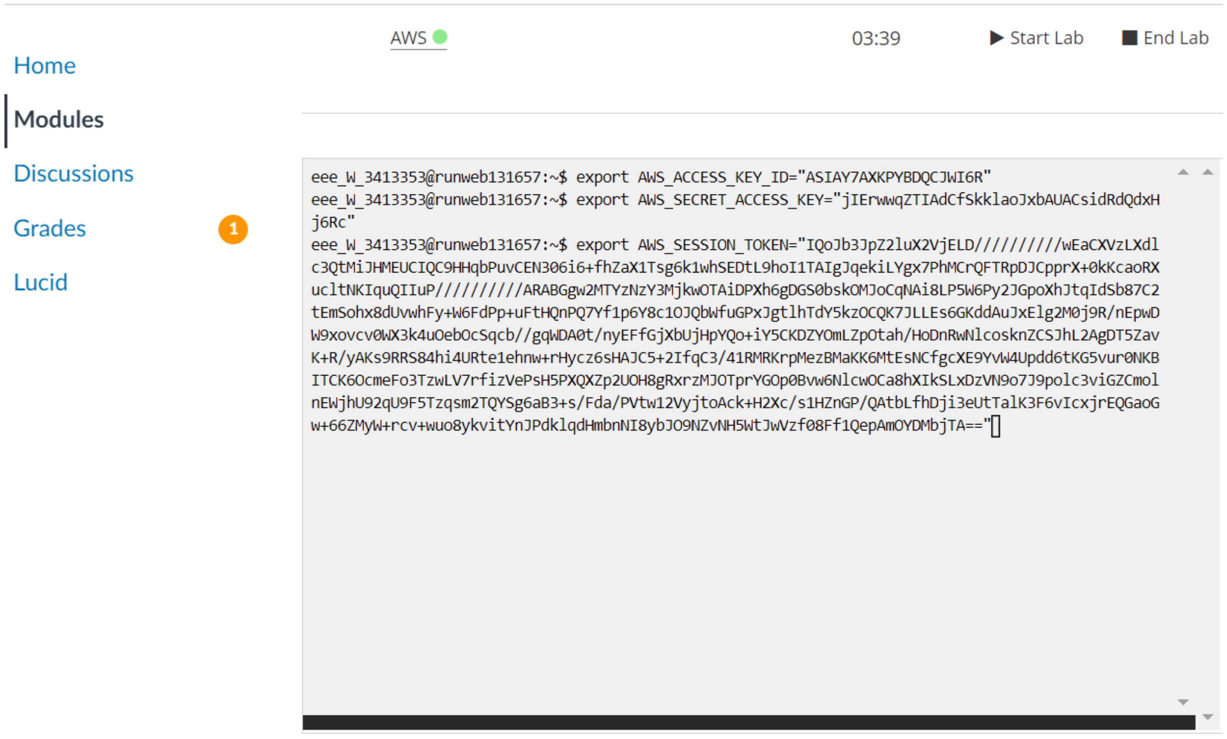
```
[default]
export AWS_ACCESS_KEY_ID=ASIAY7AXKPYBDQCJWI6R
export AWS_SECRET_ACCESS_KEY=jIErwwqZTIAdCfSkklaoJxbAUACsidRdQdxHj6Rc
export AWS_SESSION_TOKEN=IQoJb3JpZ2luX2VjELD//////////wEaCXVzLXdlc3QtMiJHMEUCIQC9HHqbPuvCEN306i6+fhZaX1Tsg6k1whSEDtL9
```

7. Copy paste the whole thing below and instead of export, write set. Then put RHS of first para in double quotes.



```
export AWS_ACCESS_KEY_ID="ASIAY7AXKPYBDQCJWI6R"
export AWS_SECRET_ACCESS_KEY="jIErwwqZTIAdCfSkklaoJxbAUACsidRdQdxHj6Rc"
export AWS_SESSION_TOKEN="IQoJb3JpZ2luX2VjELD//////////wEaCXVzLXdlc3QtMiJHMEUCIQC9HHqbPuvCEN306i6
+fhZaX1Tsg6k1whSEDtL9hoI1TAIgJqekiLYgx7PhMCrQFTRpDJCpprX+0kKcaoRXucltNKIquQIIuP//////////
ARABGgw2MTYzNzY3MjkwOTAiDPXh6gDGS0bskOMJoCqNAi8LP5W6Py2JGpoXhJtqIdSb87C2tEmSohx8dUvwhFy+W6FdPp
+uFtHQnPQ7Yf1p6Y8c1OJQbWfuGPxJgtlhTdY5kzOCQK7JLLEs6GKddAuJxElg2M0j9R/nEpwDW9xovcv0WX3k4uOebOcSqcb//gqWDA0t/
nyEFfGjXbUjHpYQo+iY5CKDZYOmLZpOtah/HoDnRwNlcosknZCSJhL2AgDT5ZavK+R/yAKs9RRS84hi4URte1ehnw+rHycz6sHAJC5+2IfqC3/
41RMRKrpMezBMaKK6MtEsNCfgcXE9YvW4Updd6tKG5vur0NKBITCK6OcmeFo3TzwLV7rfizVePsH5PXQXZp2UOH8gRxrzMJOTprYGOp0Bvw6NlcwOCa8
hXIkSLxDzVN9o7J9polc3viGZCmolnEWjhU92qU9F5Tzqsm2TQYSg6aB3+s/Fda/PVtw12VyjtoAck+H2Xc/s1HZnGP/
QAtbLfhDji3eUtTalK3F6vIcxjrEQGaoGw+66ZMyW+rcv+wuo8ykvitYnJPdklqdHmbnNI8ybJO9NZvNH5WtJwVzf08Ff1QepAmOYDMbjTA=="



set AWS_ACCESS_KEY_ID=ASIAY7AXKPYBDQCJWI6R
set AWS_SECRET_ACCESS_KEY=jIErwwqZTIAdCfSkklaoJxbAUACsidRdQdxHj6Rc
set AWS_SESSION_TOKEN=IQoJb3JpZ2luX2VjELD//////////wEaCXVzLXdlc3QtMiJHMEUCIQC9HHqbPuvCEN306i6
+fhZaX1Tsg6k1whSEDtL9hoI1TAIgJqekiLYgx7PhMCrQFTRpDJCpprX+0kKcaoRXucltNKIquQIIuP//////////
ARABGgw2MTYzNzY3MjkwOTAiDPXh6gDGS0bskOMJoCqNAi8LP5W6Py2JGpoXhJtqIdSb87C2tEmSohx8dUvwhFy+W6FdPp
+uFtHQnPQ7Yf1p6Y8c1OJQbWfuGPxJgtlhTdY5kzOCQK7JLLEs6GKddAuJxElg2M0j9R/nEpwDW9xovcv0WX3k4uOebOcSqcb//gqWDA0t/
nyEFfGjXbUjHpYQo+iY5CKDZYOmLZpOtah/HoDnRwNlcosknZCSJhL2AgDT5ZavK+R/yAKs9RRS84hi4URte1ehnw+rHycz6sHAJC5+2IfqC3/
41RMRKrpMezBMaKK6MtEsNCfgcXE9YvW4Updd6tKG5vur0NKBITCK6OcmeFo3TzwLV7rfizVePsH5PXQXZp2UOH8gRxrzMJOTprYGOp0Bvw6NlcwOCa8
hXIkSLxDzVN9o7J9polc3viGZCmolnEWjhU92qU9F5Tzqsm2TQYSg6aB3+s/Fda/PVtw12VyjtoAck+H2Xc/s1HZnGP/
QAtbLfhDji3eUtTalK3F6vIcxjrEQGaoGw+66ZMyW+rcv+wuo8ykvitYnJPdklqdHmbnNI8ybJO9NZvNH5WtJwVzf08Ff1QepAmOYDMbjTA==
```

8. Copy ansd paste 1st para in learner lab



```
eee_W_3413353@runweb131657:~$ export AWS_ACCESS_KEY_ID="ASIAY7AXKPYBDQCJWI6R"
eee_W_3413353@runweb131657:~$ export AWS_SECRET_ACCESS_KEY="jIErwwqZTIAdCfSkklaoJxbAUACsidRdQdxH
j6Rc"
eee_W_3413353@runweb131657:~$ export AWS_SESSION_TOKEN="IQoJb3JpZ2luX2VjELD//////////wEaCXVzLXdl
c3QtMiJHMEUCIQC9HHqbPuvCEN306i6+fhZaX1Tsg6k1whSEDtL9hoI1TAIgJqekiLYgx7PhMCrQFTRpDJCpprX+0kKcaoRX
ucltNKIquQIIuP//////////ARABGgw2MTYzNzY3MjkwOTAiDPXh6gDGS0bskOMJoCqNAi8LP5W6Py2JGpoXhJtqIdSb87C2
tEmSohx8dUvwhFy+W6FdPp+uFtHQnPQ7Yf1p6Y8c1OJQbWfuGPxJgtlhTdY5kzOCQK7JLLEs6GKddAuJxElg2M0j9R/nEpwD
W9xovcv0WX3k4uOebOcSqcb//gqWDA0t/nyEFfGjXbUjHpYQo+iY5CKDZYOmLZpOtah/HoDnRwNlcosknZCSJhL2AgDT5Zav
K+R/yAKs9RRS84hi4URte1ehnw+rHycz6sHAJC5+2IfqC3/41RMRKrpMezBMaKK6MtEsNCfgcXE9YvW4Updd6tKG5vur0NKB
ITCK6OcmeFo3TzwLV7rfizVePsH5PXQXZp2UOH8gRxrzMJOTprYGOp0Bvw6NlcwOCa8hXIkSLxDzVN9o7J9polc3viGZCmol
nEWjhU92qU9F5Tzqsm2TQYSg6aB3+s/Fda/PVtw12VyjtoAck+H2Xc/s1HZnGP/QAtbLfhDji3eUtTalK3F6vIcxjrEQGaoG
w+66ZMyW+rcv+wuo8ykvitYnJPdklqdHmbnNI8ybJO9NZvNH5WtJwVzf08Ff1QepAmOYDMbjTA=="
```

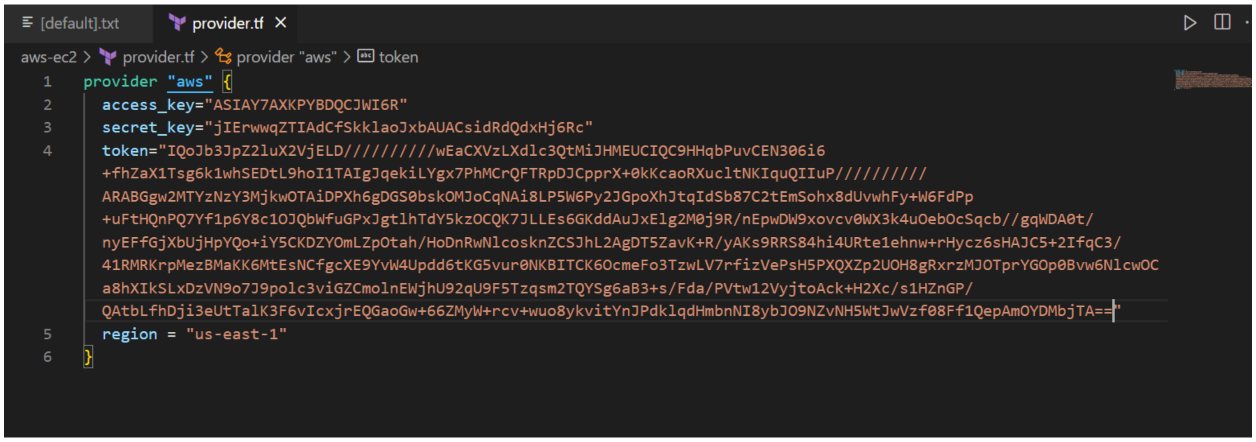
9. Create a aws-ec2 folder and create a file named provider.tf in it.

10. Copy 2nd para and paste it in VSC terminal command prompt.

11. Paste this code in provider.tf :

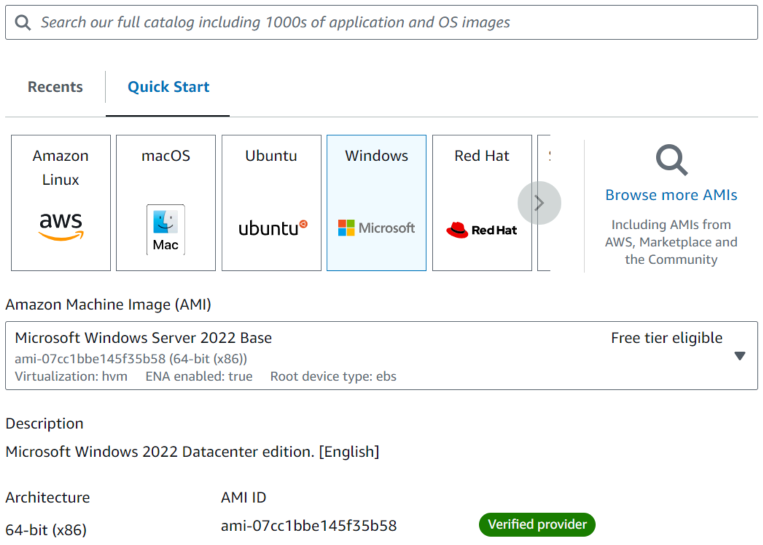
```
provider "aws" {
 access_key=""
 secret_key=""
 token=""
 region = "us-east-1"
}
```

12. Verify the region in above code with the one in AWS CLI on learner lab.
13. Copy paste values from 2nd para to code in provider.tf.



```
provider "aws" {
  access_key="ASIAY7AXKPYBDQCJWI6R"
  secret_key="jIErwwqZTIAdCfSkklaoJxbAUACsidRdQdxHj6Rc"
  token="IQoJb3JpZ2luX2VjELD//////////wEaCXVzLXdlc3QtMiJHMEUCIQC9HHqbPuvCEN306i6
  +fhZaX1Tsg6k1whSEDtL9hoI1TAIgJqekiLYgx7PhMCrQFTRpDJCpprX+0kKcaoRXucltNKIquQIIuP//////////
  ARABGgw2MTYzNzY3MjkwOTAiDPXh6gDGS0bskOMJoCqNAi8LP5W6Py2JGpoXhJtqIdSb87C2tEmSohx8dUvwhFy+W6FdPp
  +uFtHQnPQ7Yf1p6Y8c1OJQbWfuGPxJgtlhTdY5kzOCQK7JLLEs6GKddAuJxElg2M0j9R/nEpwDW9xovcv0WX3k4uOebOcSqcb//gqWDA0t/
  nyEFfGjXbUjHpYQo+iY5CKDZYOmLZpOtah/HoDnRwNlcosknZCSJhL2AgDT5ZavK+R/yAKs9RRS84hi4URte1ehnw+rHycz6sHAJC5+2IfqC3/
  41RMRKrpMezBMaKK6MtEsNCfgcXE9YvW4Upd6tKG5vur0NKBITCK6OcmeFo3TzwLV7rfizVePsH5PXQXZp2UOH8gRxrzMJOTprYGOp0Bvw6NlcwOC
  a8hXIkSLxDzVN9o7J9polc3viGZCmolnEWjhU92qU9F5Tzqsm2TQYSg6aB3+s/Fda/PVtw12VyjtoAck+H2Xc/s1HZnGP/
  QAtbLfhDji3eUtTalK3F6vIcxjrEQGaoGw+66ZMyW+rcv+wuo8ykvitYnJPdklqdHmbnNI8ybJO9NZvNH5WtJwVzf08Ff1QepAmOYDMbjTA=="
  region = "us-east-1"
}
```

14. Aws-ec2 is our module…click on that folder and create a new file main.tf.
15. Click on AWS on learner lab and open dashboard.
16. Search-select ec2
17. Click on launch instance
18. Click on windows and confirm changes.



Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
aws
macOS
Mac
Ubuntu
ubuntu
Windows
Microsoft
Red Hat
Red Hat
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Microsoft Windows Server 2022 Base — Free tier eligible
ami-07cc1bbe145f35b58 (64-bit (x86))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Microsoft Windows 2022 Datacenter edition. [English]

Architecture	AMI ID	
64-bit (x86)	ami-07cc1bbe145f35b58	Verified provider

19. Copy the AMI ID
20. Paste this code in main.tf and paste your AMI ID

```
resource "aws_instance" "myserver" {
  ami = "ami-07cc1bbe145f35b58"
  instance_type = "t2.micro"
  tags = {
    Name = "sample server"
  }
}
```

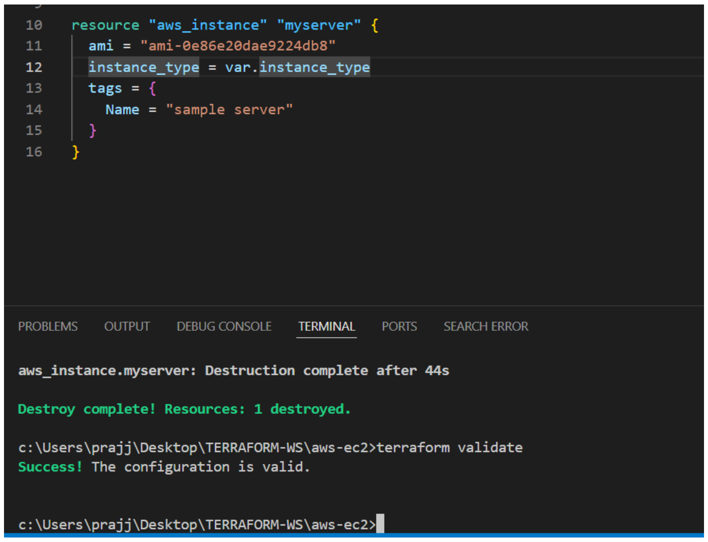
21. Create two files variables.tf and output.tf.

22. Paste this code in variables.tf:

```
variable "instance_type" {
  description = "value"
  type = string
  default = "t2.micro"
}
```

23. Go in main.tf and write var.instance_type in place of instance type.

24. On terminal write terraform validate.

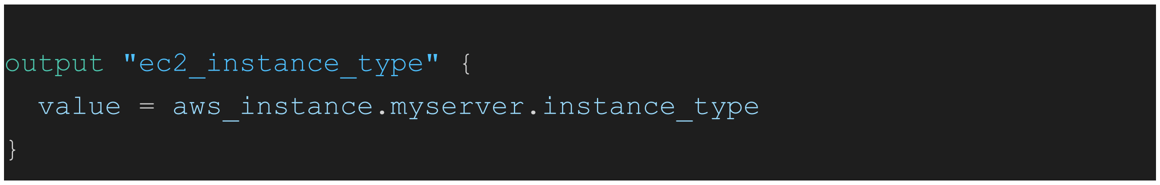


```
resource "aws_instance" "myserver" {
  ami = "ami-0e86e20dae9224db8"
  instance_type = var.instance_type
  tags = {
    Name = "sample server"
  }
}
```

```
aws_instance.myserver: Destruction complete after 44s

Destroy complete! Resources: 1 destroyed.

c:\Users\prajj\Desktop\TERRAFORM-WS\aws-ec2>terraform validate
Success! The configuration is valid.


c:\Users\prajj\Desktop\TERRAFORM-WS\aws-ec2>
```

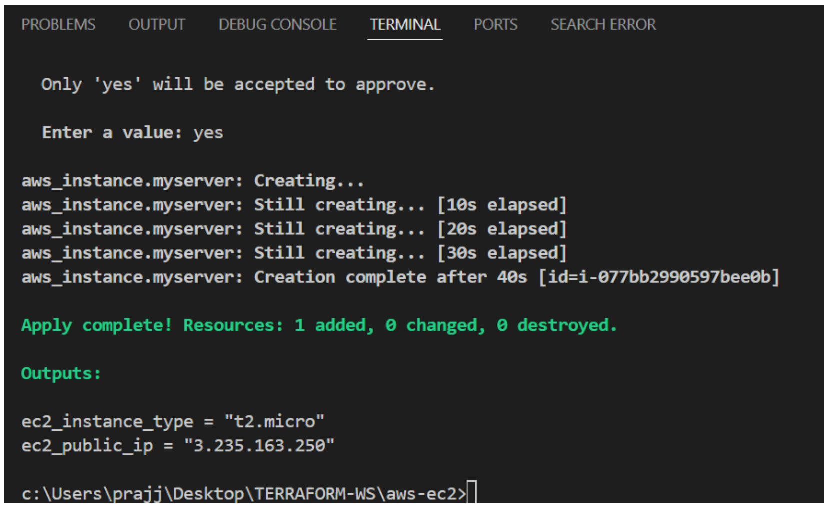
25. Go in output.tf and paste the below code:

```
output "ec2_public_ip" {
  value = aws_instance.myserver.public_ip
}
```



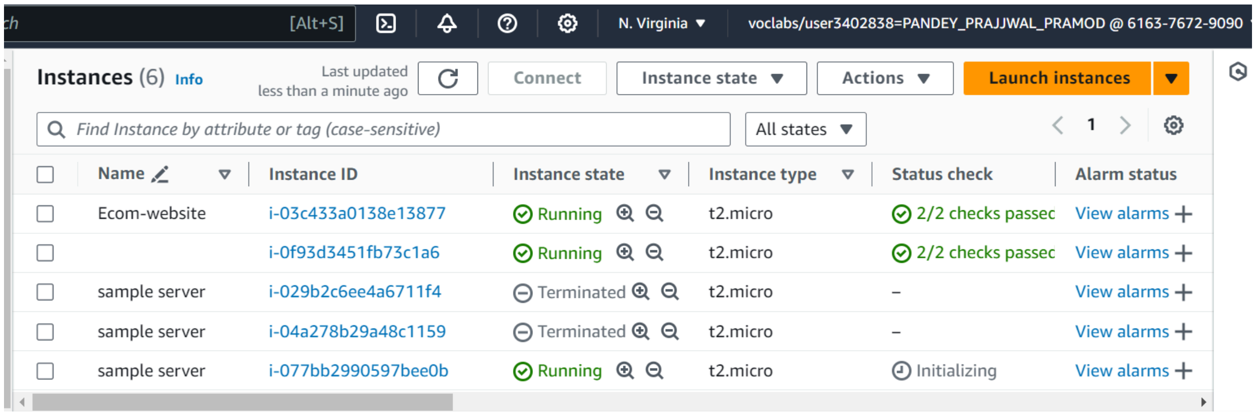
```
output "ec2_instance_type" {
  value = aws_instance.myserver.instance_type
}
```

26. In terminal write terraform plan and terraform apply. Copy the IP address on notepad.



```
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.myserver: Creating...
aws_instance.myserver: Still creating... [10s elapsed]
aws_instance.myserver: Still creating... [20s elapsed]
aws_instance.myserver: Still creating... [30s elapsed]
aws_instance.myserver: Creation complete after 40s [id=i-077bb2990597bee0b]

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

Outputs:

ec2_instance_type = "t2.micro"
ec2_public_ip = "3.235.163.250"

c:\Users\prajj\Desktop\TERRAFORM-WS\aws-ec2>
```

27. You can verify it on aws instances

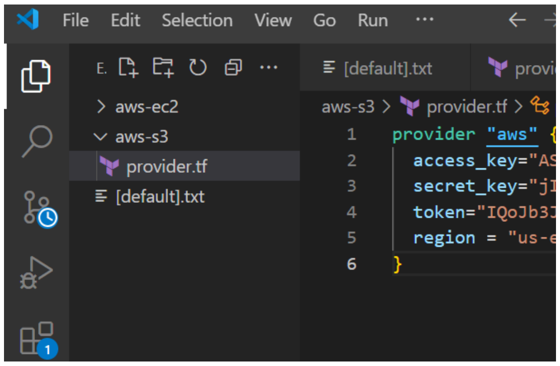

[Alt+S]
N. Virginia
voclabs/user3402838=PANDEY_PRAJJWAL_PRAMOD @ 6163-7672-9090
Instances (6) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
1

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
	Ecom-website	i-03c433a0138e13877	Running	t2.micro	2/2 checks passec	View alarms +
		i-0f93d3451fb73c1a6	Running	t2.micro	2/2 checks passec	View alarms +
	sample server	i-029b2c6ee4a6711f4	Terminated	t2.micro	–	View alarms +
	sample server	i-04a278b29a48c1159	Terminated	t2.micro	–	View alarms +
	sample server	i-077bb2990597bee0b	Running	t2.micro	Initializing	View alarms +

28. Write terraform destroy and terminate it.

B. Creating S3-bucket using terraform.

1. Make a folder aws-s3 and paste the provider.tf file from aws-ec2 in it.



```
provider "aws" {
  access_key="AS
  secret_key="jI
  token="IQoJb3J
  region = "us-e
}
```

2. Create main.tf in aws-s3 and paste the below code:

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.64.0"
    }
  }
}

resource "aws_s3_bucket" "demo-bucket" {
  bucket = "demo-bucket-advdevops-practical-2024"
}
```

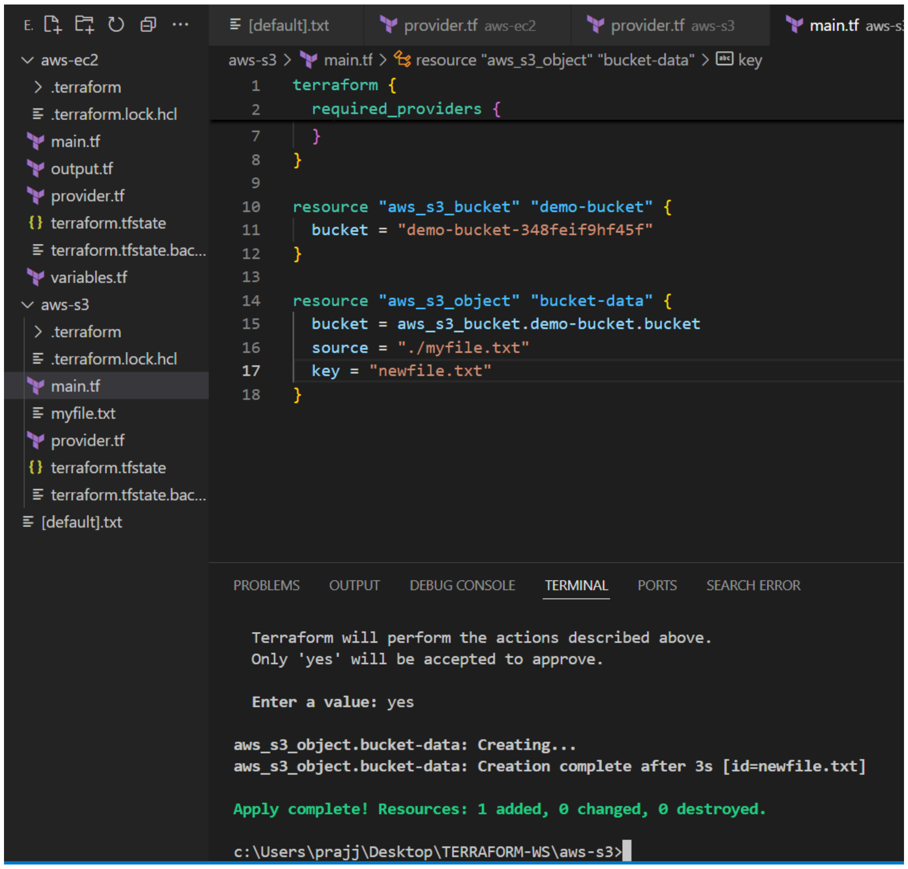
3. Cd over to aws-s3 and initialize terraform, terraform plan, terraform apply.

4. Make a new file named “myfile.txt” in aws-s3. Copy paste the EC2 public IP address in it and save it.

5. Go to main.tf and paste this code

```
resource "aws_s3_object" "bucket-data" {
  bucket = aws_s3_bucket.demo-bucket.bucket
  source = "./myfile.txt"
  key = "newfile.txt"
}
```

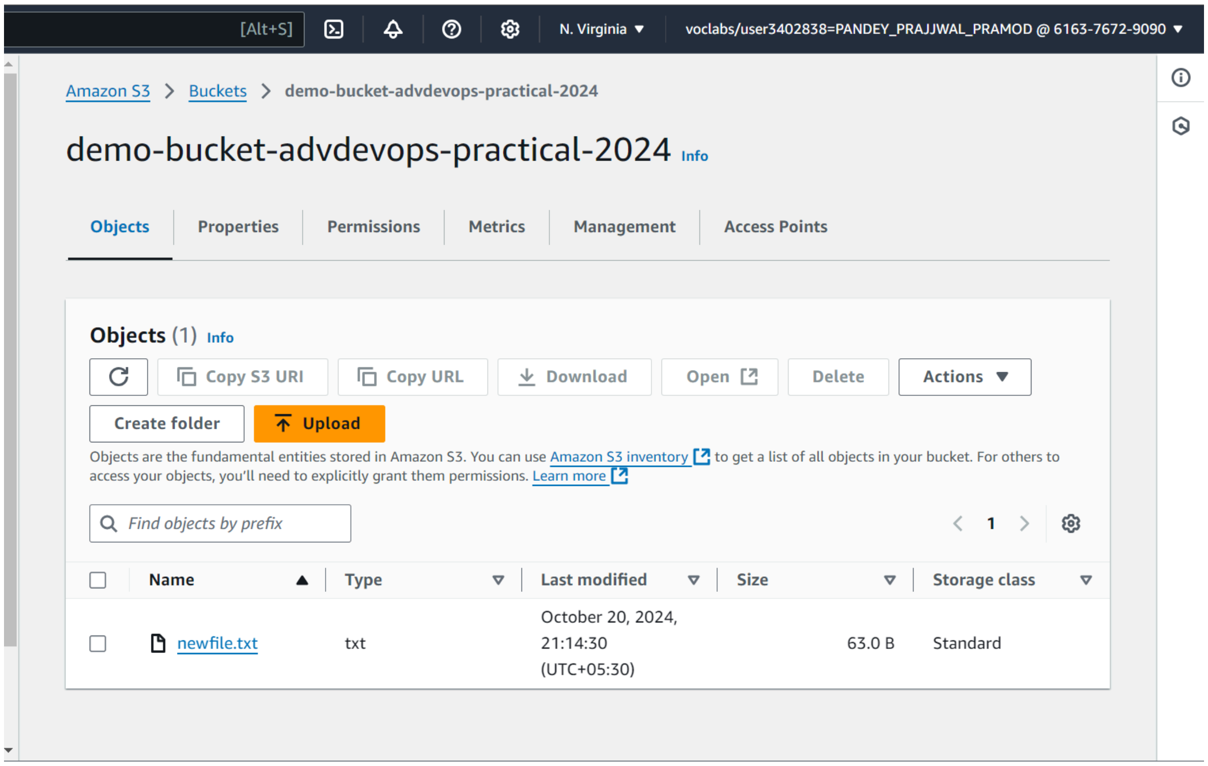
6. Terraform apply in the terminal



```
terraform {
  required_providers {
  }
}

resource "aws_s3_bucket" "demo-bucket" {
  bucket = "demo-bucket-348feif9hf45f"
}

resource "aws_s3_object" "bucket-data" {
  bucket = aws_s3_bucket.demo-bucket.bucket
  source = "./myfile.txt"
  key = "newfile.txt"
}
```

```
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_s3_object.bucket-data: Creating...
aws_s3_object.bucket-data: Creation complete after 3s [id=newfile.txt]

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

c:\Users\prajj\Desktop\TERRAFORM-WS\aws-s3>
```

7. Bucket has been successfully created.



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
newfile.txt	txt	October 20, 2024, 21:14:30 (UTC+05:30)	63.0 B	Standard

3. Demonstration Preparation

● **Key Points:**
Demonstrate how Terraform scripts are structured to create both an EC2 instance and an S3 bucket.
Focus on Terraform's ability to extract the EC2 instance's public IP address and store it in the S3 bucket, showcasing its use in automating data exchange between resources.
Highlight the automation benefits and how Terraform simplifies the creation and management of AWS infrastructure with minimal human intervention.

● **Practice:**
Practicing the execution of the Terraform scripts is essential to ensure a smooth demonstration. Test the code for any errors in creating the EC2 instance and S3 bucket. Familiarize yourself with troubleshooting common issues that may arise during Terraform execution, such as AWS authentication errors or conflicts in resource naming.