

Advance DevOps Practical Examination Case Study Assignment

NAME-SPANDAN DEB

CLASS-D15A

ROLL NO-13

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: This case study focuses on automating cloud infrastructure provisioning using Terraform, specifically for AWS EC2 and S3 services. The primary goal is to efficiently manage cloud resources with Infrastructure as Code (IaC) principles. By using Terraform, we can deploy and configure an EC2 instance and an S3 bucket in AWS, ensuring consistency and automation in resource management.

Key Feature and Application: The unique feature of this case study is the seamless automation of infrastructure provisioning and configuration using Terraform. This automation eliminates the need for manual cloud resource management, allowing users to define and manage infrastructure as code. In practical terms, this enables developers and cloud engineers to create scalable, repeatable cloud environments. The EC2 instance's public IP address is extracted and stored in an S3 bucket, showcasing how cloud resources can be dynamically interconnected through automation.

Third-Year Project Integration (Optional):

1)Automated Provisioning of OCR Processing Environment: Terraform can be used to automatically create and manage the EC2 instances required to run OCR tasks. This removes the need for manual configuration, allowing the environment to be set up quickly when new OCR jobs are required.

2)Scalability: As OCR tasks can be resource-intensive, Terraform allows the infrastructure to scale automatically. More EC2 instances can be provisioned based on the load, ensuring the OCR solution can handle large batches of images efficiently.

3) Data Storage and Retrieval with S3: AWS S3 can be utilized to store images for OCR processing and the extracted text. Terraform can automate the creation of S3 buckets to manage this data flow seamlessly, storing both input and output data in an organized and secure manner.

4) Cost Optimization: By using Terraform, resources such as EC2 instances can be spun up only when needed and terminated after use. This ensures that OCR tasks are performed without incurring unnecessary costs, as the infrastructure is dynamically managed.

5) Consistency and Repeatability: Terraforms Infrastructure as Code (IaC) approach ensures that the OCR environment is consistently set up each time, with the same configurations and dependencies. This is critical in OCR projects, where consistent results depend on identical environments.

2.Step-by-Step Explanation

Step 1: Provide a detailed explanation of the initial setup or configuration.

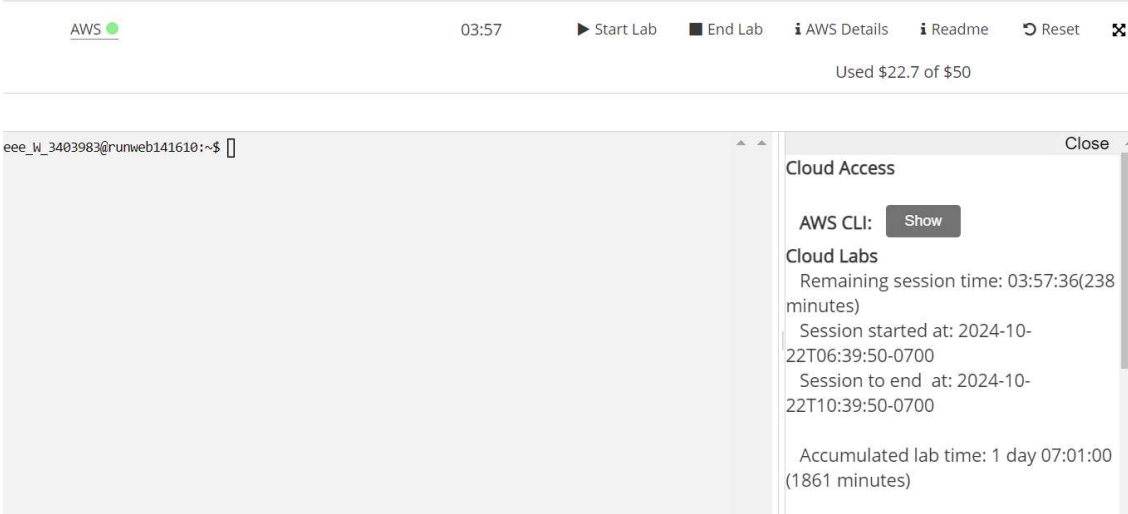
1.Install Terraform: The first step is to install Terraform on your local machine. Terraform is the tool that helps automate the process of creating and managing cloud resources. Confirm installation with terraform -v command, this should display the version of Terraform installed, confirming that it is ready for use.

2.Create AWS Account: AWS CLI is required by Terraform to communicate with your AWS account so having an AWS account is a necessity.

3.Create a Project Folder and Configure AWS CLI: On your local machine, create a new folder and open it on VS Code, this folder will hold the Terraform configuration files .Get the access key id, secret access key and session token from AWS CLI and paste it in the terminal of AWS .

Step 2: Continue with subsequent steps, including any code snippets or commands.

Go to AWS Learners Lab click start and then click on AWS details



The screenshot displays the AWS Learners Lab interface. At the top, there is a header bar with the AWS logo, a timer showing 03:57, and buttons for Start Lab, End Lab, AWS Details, Readme, Reset, and a close icon. Below the header, a status bar indicates "Used \$22.7 of \$50". The main area is divided into two sections. On the left is a terminal window with the prompt "eee_w_3403983@runweb141610:~\$". On the right is a sidebar with a "Close" button at the top. The sidebar contains two sections: "Cloud Access" with an "AWS CLI:" label and a "Show" button, and "Cloud Labs" which displays session information: "Remaining session time: 03:57:36(238 minutes)", "Session started at: 2024-10-22T06:39:50-0700", "Session to end at: 2024-10-22T10:39:50-0700", and "Accumulated lab time: 1 day 07:01:00 (1861 minutes)".

We can see AWS CLI click show.

Cloud Access

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

[default]
aws_access_key_id=ASIAYB4A6DUVO6MUERSD
aws_secret_access_key=/eY7fRS12HIWCSiZDPrO4WWe3K
jXbUUhO2k3F3q0
aws_session_token=IQoJb3JpZ2luX2VjED4aCXVzLXd1c3
QtMiJHMEUCIHUP27hv9NIv/Hc0pUjYr71gXMIIdHFz3uSdfQi
H1mxJyAiEA1vaUnbl2e2lND1N/J8aNbmEUfNkYd4Jpm+rSv6
d1j7oqrwIIP//////////ARAAGgw1NTM3ODQzMtKyNzQiDJ
w0hfQps+1fktgM9SqDAK2o/WJefi6EuokBERhi3wInOmgANa
sI870DDDeojmV86wyIkrRS+6vXHiJqks0vZLQ0woJ2SNgtVL
uTMOPtrVUh2UdTbf4kToRFGZDRUk2+imr/C9//wMvjuoQcS
UYCS8c4ChqyMTY9qKMB8dm1zSuzo94Dt8axXqoG1e3tSZE

These are the credentials copy paste these in a file in VS code. Add export keyword put the values in quotes and copy paste it in the Learner Lab.

```
eee_w_3403983@runweb141610:~$  
eee_w_3403983@runweb141610:~$ export AWS_ACCESS_KEY_ID="ASIAYB4A6DUVO6MUERSD"eee_w_3  
403983@runweb141610:~$ export AWS_SECRET_ACCESS_KEY="/eY7fRS12HIWCSiZDPrO4WWe3KjXbUU  
hO2k3F3q0"  
eee_w_3403983@runweb141610:~$ export AWS_SESSION_TOKEN="IQoJb3JpZ2luX2VjED4aCXVzLXd1  
c3QtMiJHMEUCIHUP27hv9NIv/Hc0pUjYr71gXMIIdHFz3uSdfQiH1mxJyAiEA1vaUnbl2e2lND1N/J8aNbmEU  
fNkYd4Jpm+rSv6d1j7oqrwIIP//////////ARAAGgw1NTM3ODQzMtKyNzQiDJw0hfQps+1fktgM9SqDAK2o  
/WJefi6EuokBERhi3wInOmgANasI870DDDeojmV86wyIkrRS+6vXHiJqks0vZLQ0woJ2SNgtVLuuIMQPtnVU  
h2UdTbf4kIoRFGZDRUk2+imr/C9//wMvjuoQcSUYCS8c4ChqyMTY9qKMB8dm1zSuzo94Dt8axXqoG1e3tSZE  
Z1Jd+rfk6Hddc2MoIsEbc3x/lHQ1BgCXg2SAXP9RAQKdfufelCGN63dTm9LnhyjvtKu3c131tu/Z8Ww+qEyX  
2LqHq5V1v6CpfkFpBQRnIXCYKia/205012A6BUD0mVd/EcdF8Uc5PfC0ZC4x59TGXrVIRgn7IFP9M76FSxhn  
a0oJgvcwp9beuAY6nQEpxPKkbEtKxCSZHPEp3WacAcdrU93vfsk/3qByExOeSCcouxu/q20j3NnJTKUJmnL  
vMRLU8MKFESZDDu4emerJUGbVRx7ULZ3rBigQaq227nmakH5dw/hqA00z3lYkhkFZPhD8KLQeKPRdKsMFbOQ  
Tgeq20iPiG0giReBm/XPhr2iH92h0y1oY/5GJAdBQfXUqSY55rRoJE4nPaAh"  
eee_w_3403983@runweb141610:~$
```

Create a provider.tf file and add this code. This Terraform code block defines the AWS provider with specific credentials (access key, secret key, and token) to authenticate and interact with the AWS services.

```
Welcome x provider.tf x
aws-ec2 > provider.tf > provider "aws"
1 provider "aws" {
2   access_key = "ASIAYB4A6DUVO6MUERSD"
3   secret_key = "/eY7fRS12HIWCSiZDPrO4WMe3KjXbUUhO2k3F3q0"
4   token = "IQoJb3JpZ2luX2VjED4aCXVzLXdlc3QtMiJHMEUCIHUP27hv9NIv/Hc0pUjYr71gXMIHfz3uSDFQiH1mx3yAiEA1vaUnb12e21ND1N/
J8aNBmEUfNkYd4Jpm+rSv6d1j7oqrwIip//////////ARAAAGw1NTM3ODQzMTkyNzQ1Djw0hfQps+1fktgM9SdDAk2o/
WJefi6EuokBERhi3wInOmgaNAsI870DDeoJmV86wyIkrRS+6vXHiJqks0vZLQ0woJ2SNgtVLuuIMQPtNvUu2UdThf4kIoRFGZDRUk2+imr/C9//
wMlvjuoQcSUYcS8c4ChqyMtY9qKMB8dm1zSuzo94Dt8axXqoG1e3tSZEZ1Jd+rfk6Hddc2MoIsEbc3x/
1HQ1BgCXg2SAXP9RAQKdFuFe1CGN63dTM9LnhYjvtKu3c131tu/Z8WW+qEyX2LqHq5V1v6CpFkFpBQrNIxCYKIA/205012A6BUD0mVd/
EcdF8Uc5PfC0ZC4x59TGXrVlrgn7IFP9M76FSxhnaOoJgvcwp9beuAY6nQEpXpKkbEtKxCSZHPEp3WacAcDRU93vfsk/3qByExOeSCcouxu/
q20j3NnNjTKuJmnLVmRLU8MKFESzDDu4emeRJUGbVRx7ULZ3rBigQaq227nmakH5dw/
hqAO0z31YhkhFZPhD8KLQeKPRdKsMFbOQTgeq20iPiG0giReBm/XPPr2iH92hOy1oY/5GJAdBQfXuqSY55rRoJE4nPaAh"
5   region = "us-east-1"
6 }
```

Now Create a new file main.tf in the folder. Click on AWS in learner lab it will route you to the dashboard, search EC2, click on launch instance. Select windows AMI and copy its AMI ID.

Recents

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE Li

SUSE

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Microsoft Windows Server 2022 Base

ami-0324a83b82023f0b3 (64-bit (x86))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Microsoft Windows 2022 Datacenter edition. [English]

Microsoft Windows Server 2022 Full Locale English AMI provided by Amazon

Architecture

AMI ID

Username

64-bit (x86)

ami-0324a83b82023f0b3

root

Verified provider

Feedback

Go to the terminal check whether its in current directory if not cd over it. In terminal initialize terraform.

```
HOME@LAPTOP-9JIMM8I3 MINGW64 ~/OneDrive/Desktop/Adv Devops Case Study/aws-ec2
● $ terraform init
  Initializing the backend...
  Initializing provider plugins...
  - Finding latest version of hashicorp/aws...
  - Installing hashicorp/aws v5.72.1...
  - Installed hashicorp/aws v5.72.1 (signed by HashiCorp)
  Terraform has created a lock file .terraform.lock.hcl to record the provider
  selections it made above. Include this file in your version control repository
  so that Terraform can guarantee to make the same selections by default when
  you run "terraform init" in the future.

  Terraform has been successfully initialized!
```

Create two files output.tf and variable.tf

Use this code in variables.tf. It is used to set the EC2 instance type

```
aws-ec2 > variables.tf > variable "instance_type"
1  variable "instance_type" {
2    description = "value"
3    type = string
4    default = "t2.micro"
5  }
```

In main.tf change the value of instance_type to var.instance_type

```
ec2 > main.tf > ...
  ✓ resource "aws_instance" "myserver" {
    ami = "ami-0324a83b82023f0b3"
    instance_type = var.instance_type
  ✓   tags = {
      Name = "sample server"
    }
  }
|
```

Validate the configurations by using terraform validate

```
HOME@LAPTOP-9JIMM8I3 MINGW64 ~/OneDrive/Desktop/Adv Devops C
● $ terraform validate
Success! The configuration is valid.

HOME@LAPTOP-9JIMM8I3 MINGW64 ~/OneDrive/Desktop/Adv Devops C
```

Use this code in output.tf. This is used to see key information of the infrastructure the public Ip of ec2 instance and the instance type.

```
s-ec2 > output.tf > ...
1  ✓ output "ec2_public_ip" {
2    |   value = aws_instance.myserver.public_ip
3    | }
4
5  ✓ output "ec2_instance_type" {
6    |   value = aws_instance.myserver.instance_type
7    | }
8  |
```


In terminal write terraform plan and apply. You will be able to see the IP address of the EC2 instance Copy it as it will be stored in a text file and the file will be stored in S3 bucket.

```
HOME@LAPTOP-9JIMM8I3 MINGW64 ~/OneDrive/Desktop/Adv Devops Case Study/aws-ec2
$ terraform plan

Terraform used the selected providers to generate the following execution plan.
+ create

Terraform will perform the following actions:

# aws_instance.myserver will be created
+ resource "aws_instance" "myserver" {
  + ami              = "ami-0324a83b82023f0b3"
  + arn              = (known after apply)
```

```
HOME@LAPTOP-9JIMM8I3 MINGW64 ~/OneDrive/Desktop/Adv Devops Case Study/aws-ec2
$ terraform apply

Terraform used the selected providers to generate the following execution plan. Re
+ create

Terraform will perform the following actions:

# aws_instance.myserver will be created
+ resource "aws_instance" "myserver" {
  + ami              = "ami-0324a83b82023f0b3"
```

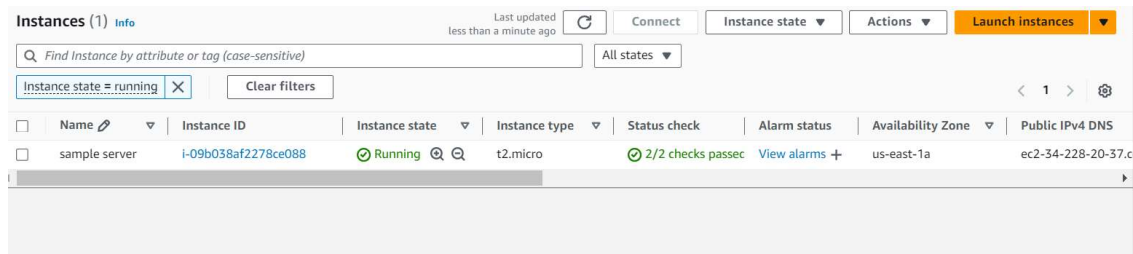
```
aws_instance.myserver: Creating...
aws_instance.myserver: Still creating... [10s elapsed]
aws_instance.myserver: Creation complete after 17s [id=i-09b038af2278ce088]
```

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

Outputs:

```
ec2_instance_type = "t2.micro"
ec2_public_ip     = "34.228.20.37"
```

To check if the instance is being created go to AWS



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	sample server	i-09b038af2278ce088	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-34-228-20-37.c

Now we have to create an S3 bucket using terraform create a new folder aws-s3 and copy the provider.tf file as the credentials is going to remain the same.

Create main.tf use this code to create bucket

```
aws-s3 > main.tf > terraform > required_providers > aws
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.64.0"
6     }
7   }
8 }
9 resource "aws_s3_bucket" "demo-bucket" {
10  bucket = "sd13-bucket-advdevops-practical-2024"
11 }
12
```

Initialize, plan and apply terraform

```
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_s3_bucket.demo-bucket: Creating...
aws_s3_bucket.demo-bucket: Creation complete after 6s [id=sd13-bucket-advdevops-practical-2024]

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

Create a new text file and store the ec2 instance public IP address

```
aws-s3 > ⚡ sd13.txt
1   ec2_public_ip = "34.228.20.37"
2   |
```

Add this code in the main.tf to upload the file in S3 bucket.

```
resource "aws_s3_object" "bucket-data" {
  bucket = aws_s3_bucket.demo-bucket.bucket
  source = "./sd13.txt"
  key = "newfile.txt"
}
```

Terraform apply in the terminal

```
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_s3_object.bucket-data: Creating...
aws_s3_object.bucket-data: Creation complete after 2s [id=newfile.txt]

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

HOME@LAPTOP-9JIMM8I3 MINGW64 ~/OneDrive/Desktop/Adv Devops Case Study/aws-s3
$
```

Bucket has been created in AWS

Amazon S3 > Buckets > sd13-bucket-advdevops-practical-2024

sd13-bucket-advdevops-practical-2024 [Info](#)

Objects

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](#)

< 1 >

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	newfile.txt	txt	October 22, 2024, 21:07:12 (UTC+05:30)	32.0 B	Standard

If we download the text file, we can see the IP address stored

 newfile - Notepad

File Edit Format View Help

```
ec2_public_ip = "34.228.20.37"
```