

Name: Om Sanjay Bhavsar

Roll No.: 331069

PRN: 22120084

Batch: A3

Assignment No 5

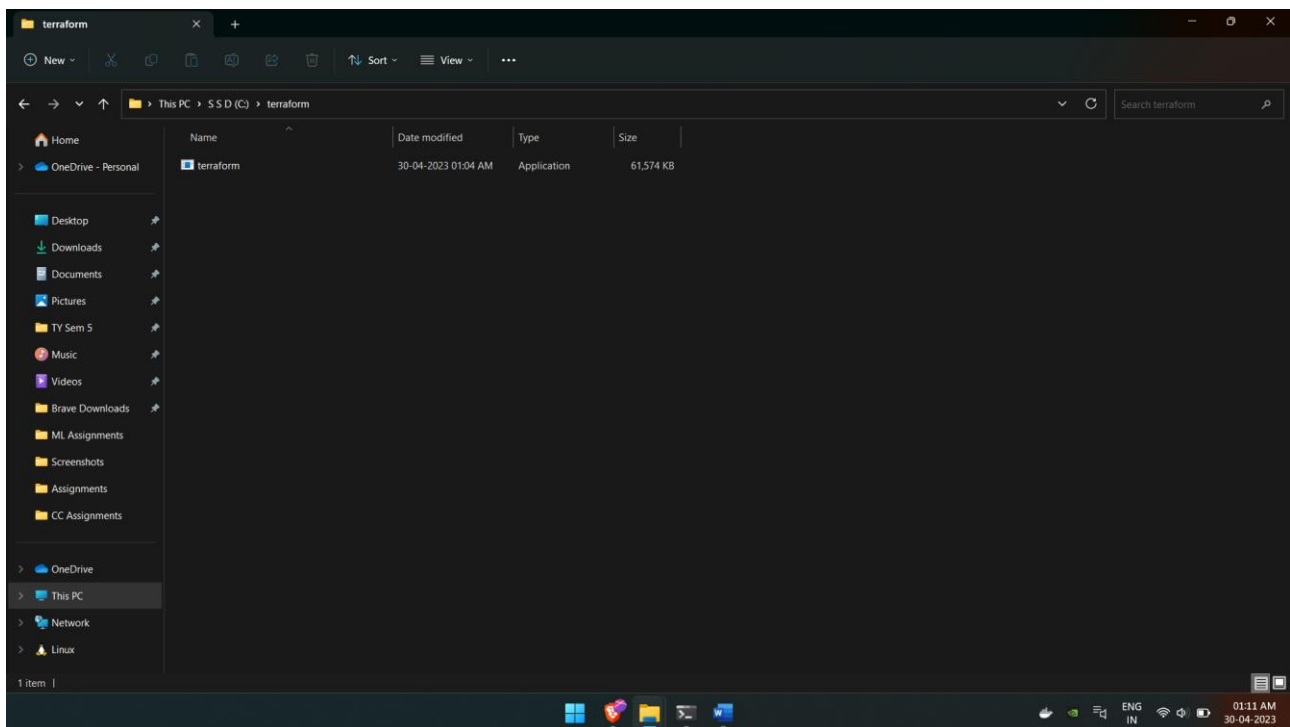
Title: Write IaC using terraform to create EC2 machine on aws (use modules, input and output)

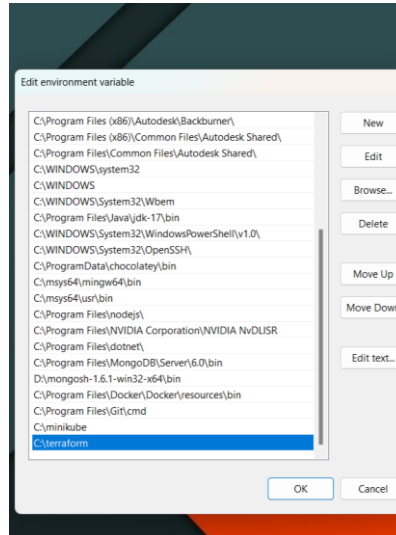
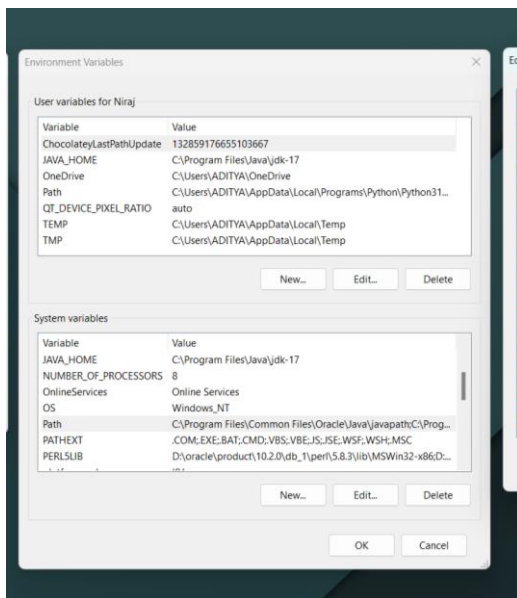
Theory:

- 1) What is Terraform?

Terraform is an infrastructure as code tool that lets you build, change, and version infrastructure safely and efficiently. This includes low-level components like compute instances, storage, and networking; and high-level components like DNS entries and SaaS features.

- 2) Step-by-step screenshot to install and configure Terraform.





```

Windows PowerShell
PS C:\Users\ADITYA> terraform --help
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init      Prepare your working directory for other commands
  validate  Check whether the configuration is valid
  plan      Show changes required by the current configuration
  apply     Create or update infrastructure
  destroy   Destroy previously-created infrastructure

All other commands:
  console   Try Terraform expressions at an interactive command prompt
  fmt       Reformat your configuration in the standard style
  force-unlock Release a stuck lock on the current workspace
  get       Install or upgrade remote Terraform modules
  graph     Generate a Graphviz graph of the steps in an operation
  import    Associate existing infrastructure with a Terraform resource
  login     Obtain and save credentials for a remote host
  logout    Remove locally-stored credentials for a remote host
  metadata  Metadata related commands
  output    Show output values from your root module
  providers Show the providers required for this configuration
  refresh   Update the state to match remote systems
  show      Show the current state or a saved plan
  state     Advanced state management
  taint     Mark a resource instance as not fully functional
  test      Experimental support for module integration testing
  untaint   Remove the 'tainted' state from a resource instance
  version   Show the current Terraform version
  workspace Workspace management

Global options (use these before the subcommand, if any):
  -chdir=DIR Switch to a different working directory before executing the
              given subcommand.
  -help      Show this help output, or the help for a specified subcommand.
  -version   An alias for the "version" subcommand.
  
```

3) Terraform script to create Infrastructure on any cloud platform (AWS or Azure or Google)

Gold - playlist by MRUDUL | Spotify Classes Dashboard How to Use and install AWS CLI On Vi Command Line Interface - AWS C

aws.amazon.com/cli/

aws

Contact Us Support English My Account Sign In Create an AWS Account

Products Solutions Pricing Documentation Learn Partner Network AWS Marketplace Customer Enablement Events Explore More

RESOURCES

AWS Command Line Interface

RELATED LINKS

Documentation

Tools

Release Notes

Get Started with AWS for Free

Create Free Account

AWS Command Line Interface

The AWS Command Line Interface (AWS CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

The AWS CLI v2 offers several [new features](#) including improved installers, new configuration options such as AWS IAM Identity Center (successor to AWS SSO), and various interactive features.

Windows
Download and run the [64-bit Windows installer](#).

MacOS
Download and run the [MacOS PKG installer](#).

Linux
Download, unzip, and then run the [Linux installer](#).

Amazon Linux
The AWS CLI comes pre-installed on [Amazon Linux AMI](#).

Release Notes
Check out the [Release Notes](#) for more information on the latest version.

1 Getting Started » AWS CLI Reference » GitHub Project » Community Forum »

aws-shell (Developer Preview)

<https://awscli.amazonaws.com/AWSCLIV2.msi>

aws-shell is a command-line shell program that provides convenience and productivity features to help both new and advanced users

10:32 PM 30-04-2023

File Edit Selection View Go Run ...

cc

EXPLORER

cc

terraform

terraform.lock.hcl

createEc2.tf

createEc2.tf

```
1 provider "aws" {
2   region = "us-east-1"
3 }
4
5 variable "instance_type" {
6   default = "t2.micro"
7 }
8
9 variable "ami_id" {
10  default = "ami-02396cdd13e9a1257"
11 }
12
13 resource "aws_instance" "example" {
14   ami = var.ami_id
15   instance_type = var.instance_type
16 }
17
```

OUTLINE

TIMELINE

Run Testcases 0 0 Connect

Ln 17, Col 1 Spaces: 4 UTF-8 CRLF Terraform Go Live Prettier

11:24 PM 30-04-2023

aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user2410046=om.22120084@vilit.ac.in @ 9941-3564-4410

EC2 > Instances > Launch an instance

Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name

[Add additional tags](#)

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Recents

Quick Start

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

S

Summary

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.0.2...[read more](#)

ami-02396cdd13e9a1257

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier

Cancel

Launch instance

[Review commands](#)

CloudShell

Feedback

Language

© 2023, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

aws

Services

Search

[Alt+S]

Global

voclabs/user2410046=om.22120084@vilit.ac.in @ 9941-3564-4410

IAM > Users > Create user

Step 1

Specify user details

Step 2

Set permissions

Step 3

Review and create

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name	Console password type	Require password reset
omb	None	No

Permissions summary

< 1 >

Name	Type	Used as
No resources		

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

CloudShell

Feedback

Language

© 2023, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

aws Services Search [Alt+S] N. Virginia voclabs/user2410046=om.22120084@vilit.ac.in @ 9941-3564-4410

New EC2 Experience Tell us what you think

EC2 Dashboard
EC2 Global View
Events
Limits

Instances

Instances

Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Scheduled Instances
Capacity Reservations

Images

AMIs
AMI Catalog

Elastic Block Store

Volumes

Instances (1/2) info

Find instance by attribute or tag (case-sensitive)

Instance state = running X Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
om007	i-08d37497331459640	Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-52-70-59-23
Bastion Host	i-0b4f94a82d16e0596	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-86-200-1

Instance: i-08d37497331459640 (om007)

Details Security Networking Storage Status checks Monitoring Tags

▼ Instance summary info

Instance ID i-08d37497331459640 (om007)	Public IPv4 address 52.70.59.235 open address	Private IPv4 addresses 172.31.92.213
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-52-70-59-235.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-92-213.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-92-213.ec2.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding
Auto-assigned IP address	VPC ID	

Waiting for us-east-1 console.aws.amazon.com... © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

File Edit Selection View Go Run ... CC

EXPLORER

cc

.terraform

terraform.lock.hcl

createEc2.tf

createEc2.tf X

```
1 provider "aws" {
2   region = "ap-south-1"
3 }
4
5 variable "instance_type" {
6   default = "t2.micro"
7 }
8
9 variable "ami_id" {
10  default = "ami-02396cdd13e9a1257"
11 }
12
13 resource "aws_instance" "example" {
14   ami = var.ami_id
15   instance_type = var.instance_type
16 }
```

Windows PowerShell

```
PS C:\Users\ADITYA\Desktop\CC> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.65.0...
- Installed hashicorp/aws v4.65.0 (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!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
PS C:\Users\ADITYA\Desktop\CC> |
```

Run Testcases 0.0.0 Connect In 16 Col 2 Spaces 4 UTF-8 CR LF Terraform Go Live Prettier