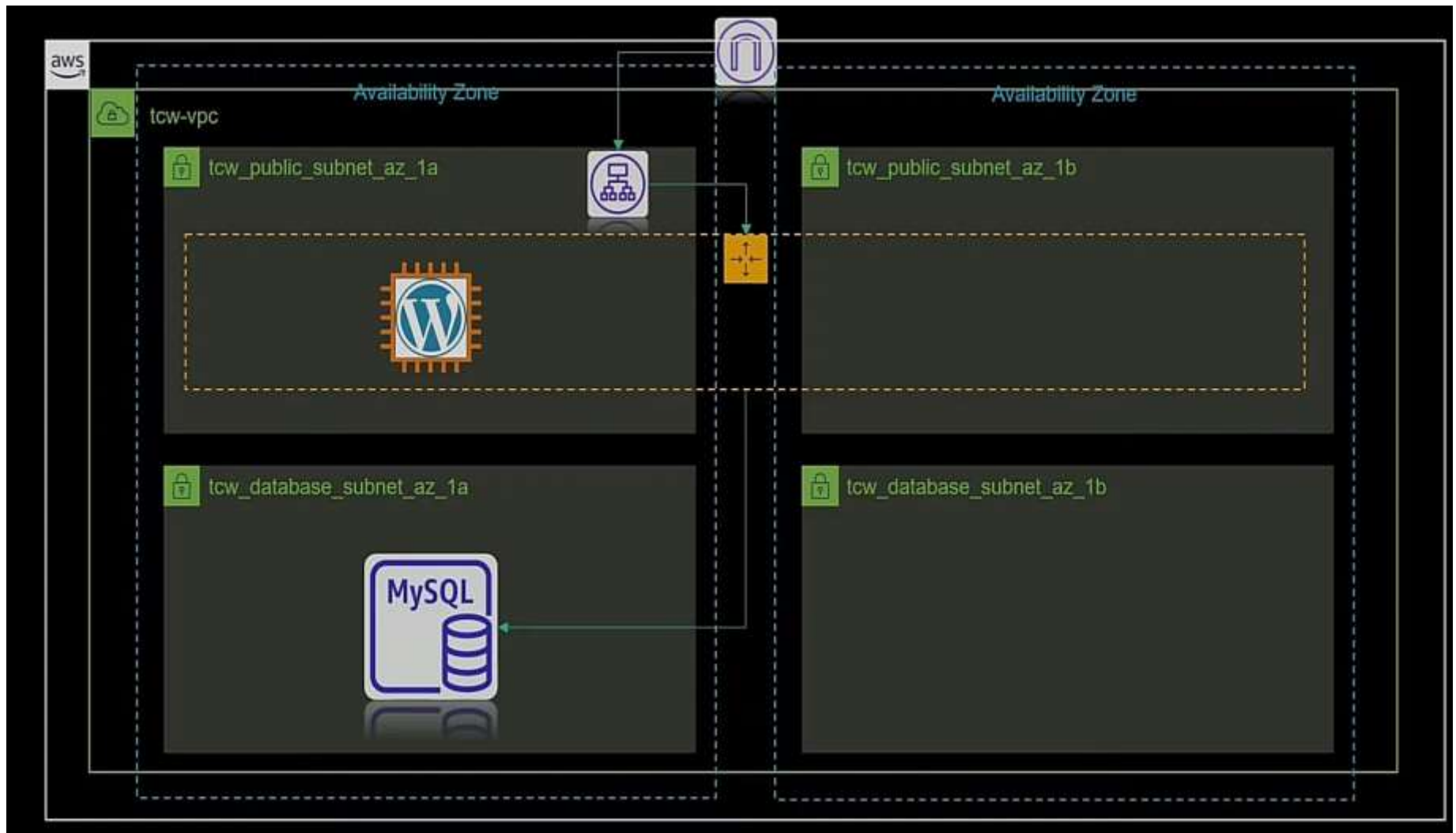


Project

This terraform projects creates Vpc, LoadBalancer, Autoscaling, RDS databse with mysql
#EC2 instance with wordpress installed and configured

- Architecture of project look like this:--



Prerequisite

- AWS account
 - Terraform
- Amazon cli if having window

Continue-----

Route Table for Application

ROUTE	CIDR_BLOCK
Local	192.168.0.0/16
Internet_Gateway	0.0.0.0/0

Route Table for Database

ROUTE	CIDR_BLOCK
Local	192.168.0.0/16

Downloading & Installing Prerequisite

- Go to google and type download terraform

- → ↻ 🔒 google.com/search?q=download+terraform&rlz=1C1RXQR_enIN1019IN1019&oq=download+&aqs=chrome.0.69i59l3j69i57j69i60l4.370

Google

download terraform

🔍 All 📺 Videos 📰 News 🖼️ Images 📍 Maps ⋮ More Tools

About 99,30,000 results (0.39 seconds)

<https://www.terraform.io> > downloads ⋮

Downloads | Terraform by HashiCorp

Download Terraform. macOS. Windows Linux FreeBSD OpenBSD. Solaris. Package Manager. Ubuntu/Debian CentOS/RHEL Fedora Amazon Linux Homebrew.

You've visited this page 4 times. Last visit: 24/8/22

<https://www.terraform.io> ⋮

Terraform by HashiCorp

Terraform is an open-source infrastructure as code software tool that enables you ... **Download.**

Download the open source **Terraform** binary and run locally or ...

People also ask ⋮

How do I download and install Terraform? ▾

How do I download Terraform version? ▾

Is there a free version of Terraform? ▾

Click to Download | Terraform by HasiCorp

Download Terraform

macOS

Windows

Linux

FreeBSD

OpenBSD

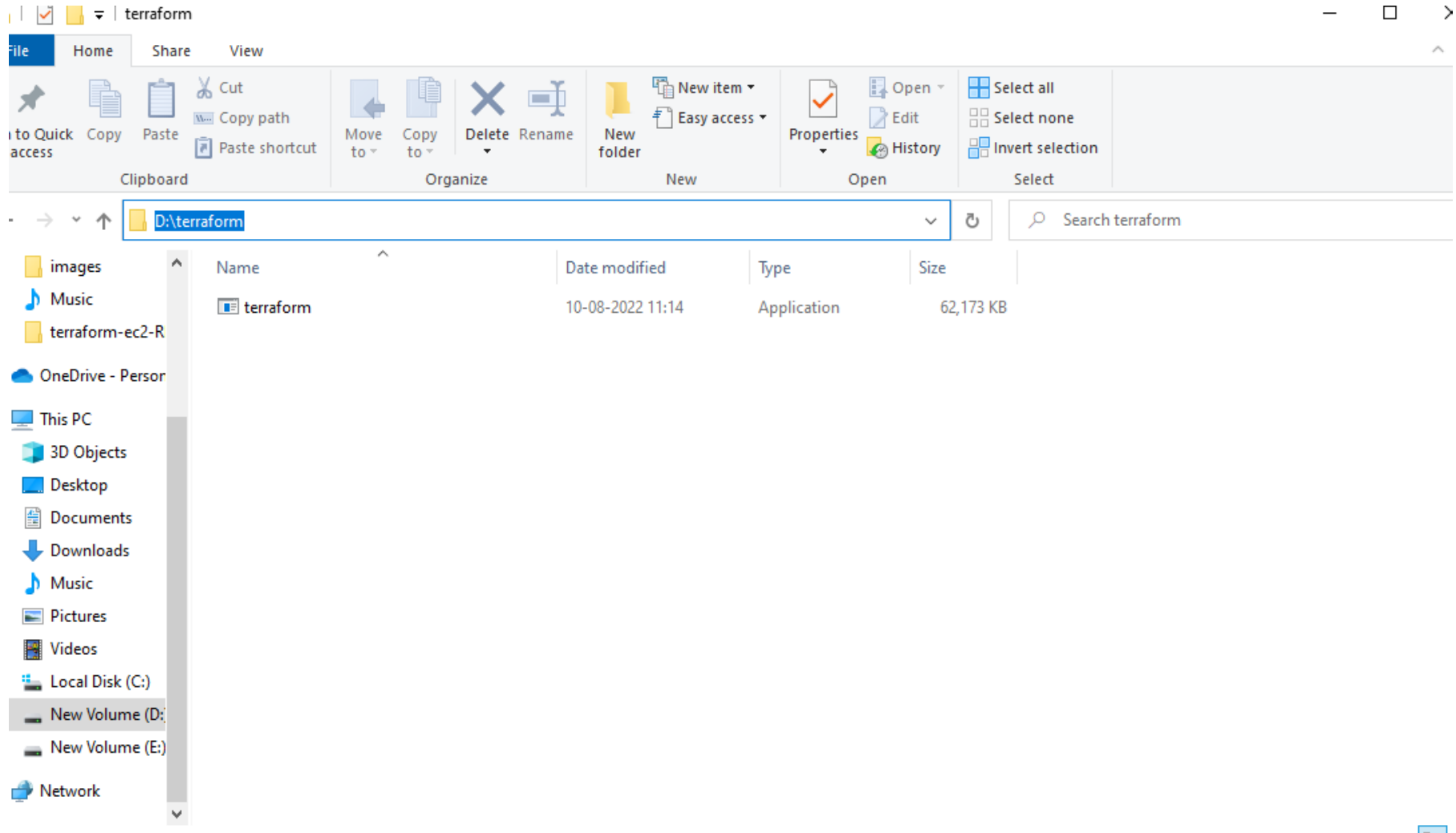
Solaris

WINDOWS BINARY DOWNLOAD

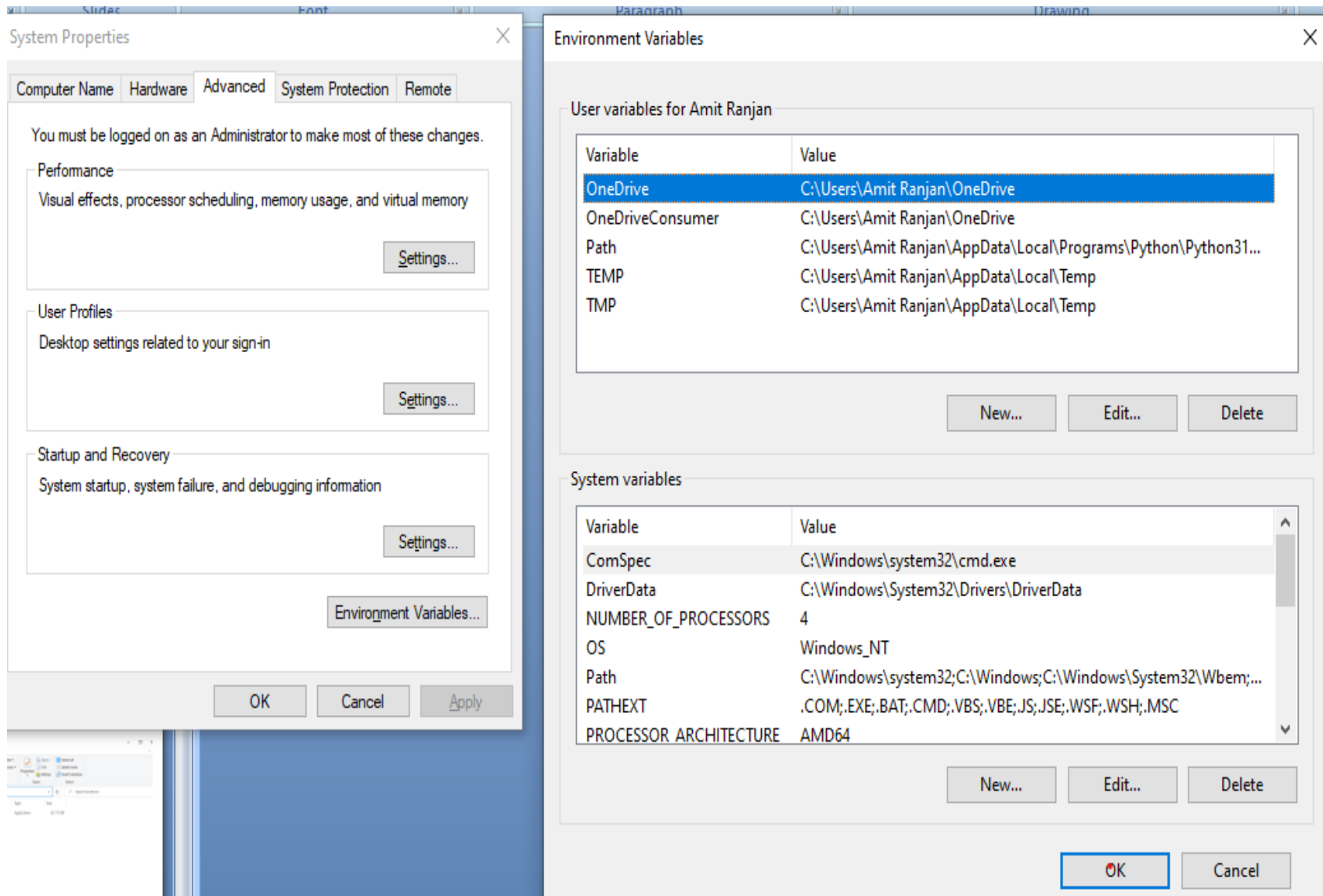


[386](#) [Amd64](#)

Copy the path where terraform is located



• Paste it to environment variable



- Download Aws cli
- Url:-- <https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>
- Click on the <https://awscli.amazonaws.com/AWSCLIV2.msi> as shown below .
- After downloading click on install

▼ Windows

Installation requirements

- We support the AWS CLI on Microsoft-supported versions of 64-bit Windows.
- Admin rights to install software

Install or update the AWS CLI

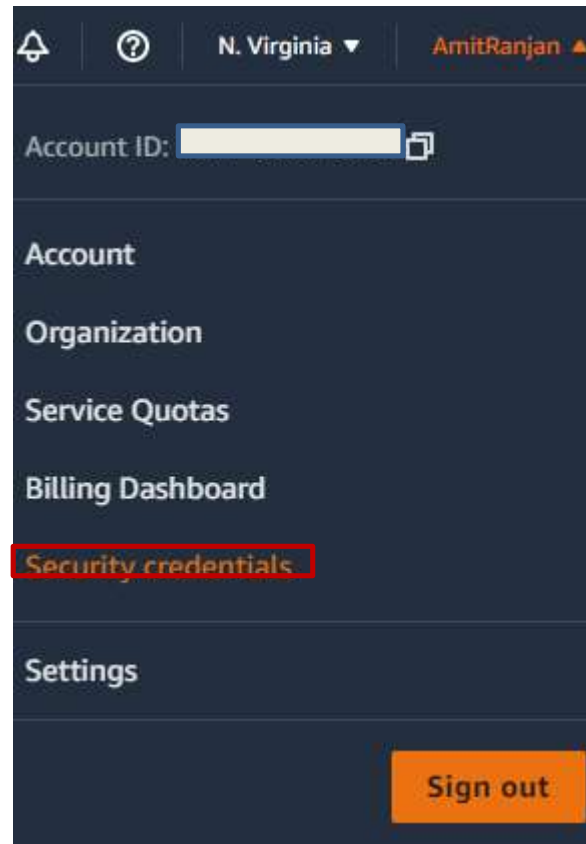
To update your current installation of AWS CLI on Windows, download a new installer each time you update to overwrite previous versions. AWS CLI is updated regularly. To see when the latest version was released, see the [AWS CLI changelog](#) on *GitHub*.

1. Download and run the AWS CLI MSI installer for Windows (64-bit):

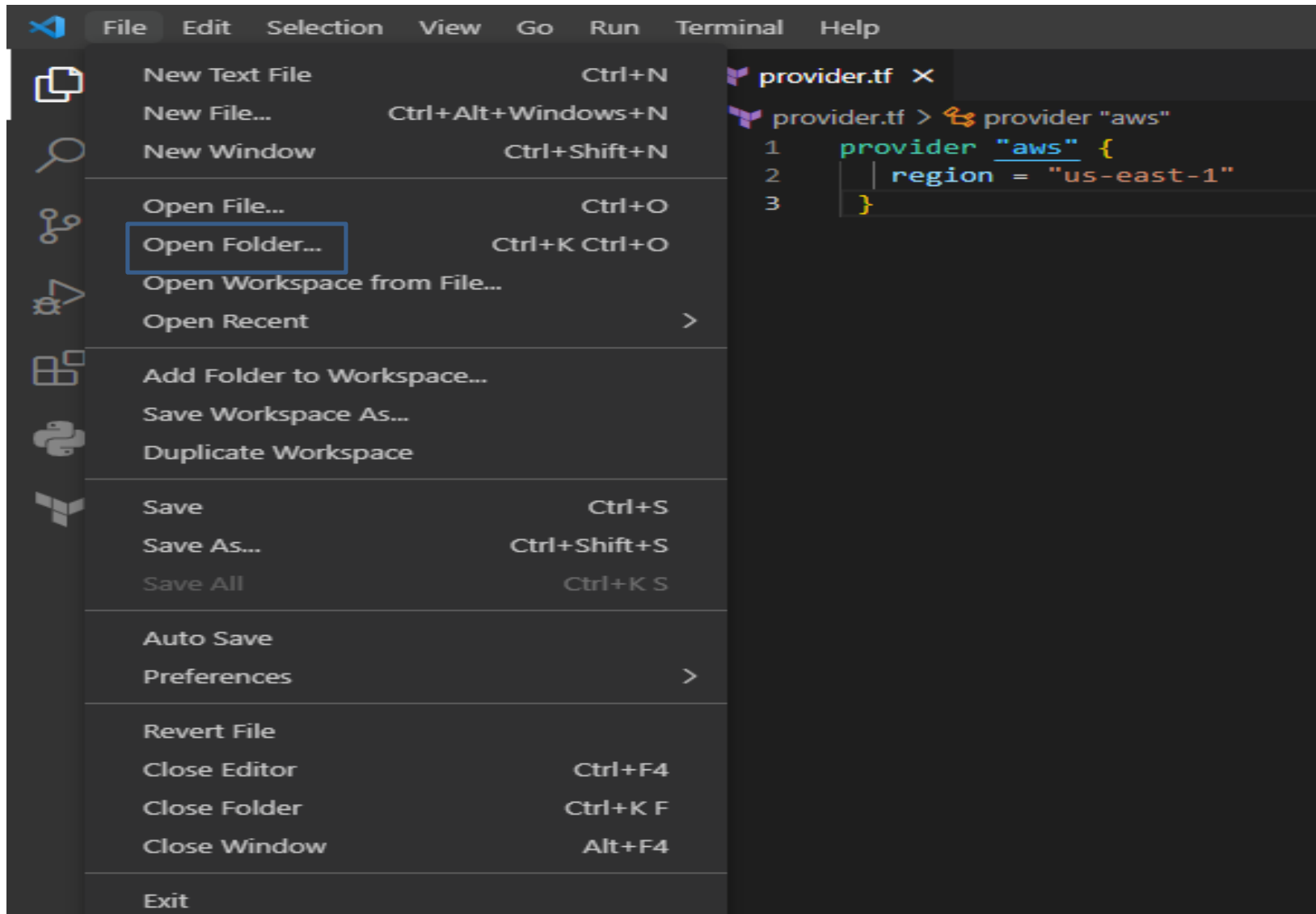
<https://awscli.amazonaws.com/AWSCLIV2.msi>

Alternatively, you can run the `msiexec` command to run the MSI installer.

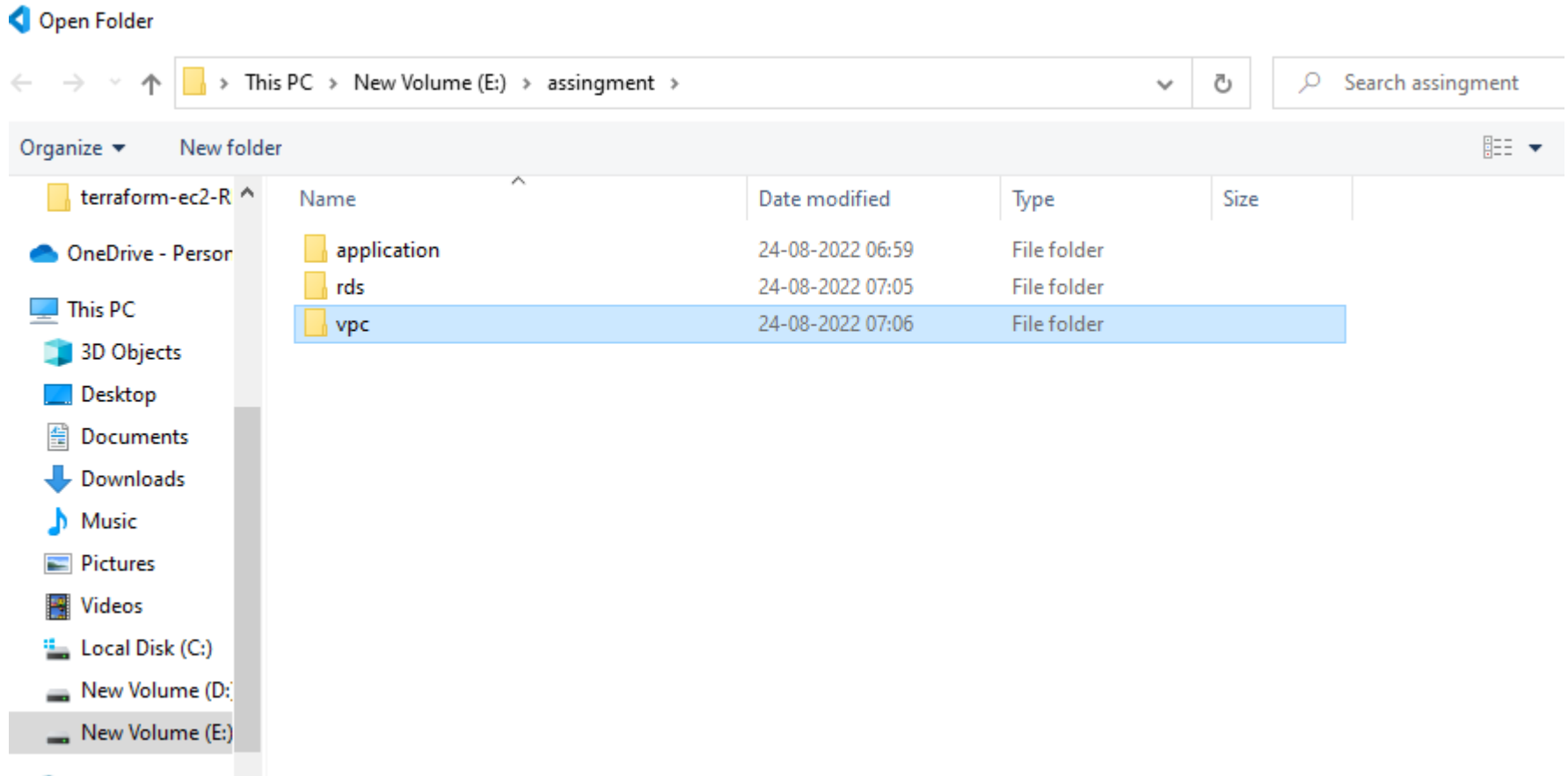
- After installing aws cli go to cmd and write aws configure and it will ask for access and secret key put public and secret key of aws console and go to security credential and click on Access keys (access key ID and secret access key) and then click on **Create New Access Key** and download the csv file



- Open vscode and go to



- Open vpc folder and click select folder on the right bottom side



- Now click on terminal in the vs code
- Commands you have to use in this project

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    JUPYTER

rs_security_group.sg: Destruction complete after 3s
rs_subnet.public_subnet_1: Destruction complete after 1s
rs_subnet.database_subnet_2: Destruction complete after 1s
rs_internet_gateway.myIGW: Destruction complete after 1s
rs_route_table.database_route_table: Destruction complete after 1s
rs_route_table.public_route_table: Destruction complete after 1s
rs_vpc.myVPC: Destroying... [id=vpc-0e8271fd00b4a8dc8]
rs_vpc.myVPC: Destruction complete after 1s

Destroy complete! Resources: 14 destroyed.
E:\assingment\vpc>
History restored

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

E:\assingment\vpc> terraform init
```

- Now click terraform plan and then terraform apply

```
PS D:\terraform project\Assingment\vpc_creation> terraform plan
data.aws_availability_zones.available: Reading...
data.aws_availability_zones.available: Read complete after 0s [id=us-east-1]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_internet_gateway.igw will be created
+ resource "aws_internet_gateway" "igw" {
  + arn          = (known after apply)
  + id          = (known after apply)
  + owner_id    = (known after apply)
  + tags        = {
    + "Name" = "tcw_igw"
  }
  + tags_all    = {
    + "Name" = "tcw_igw"
  }
  + vpc_id      = (known after apply)
}

# aws_route.public_internet_gateway will be created
+ resource "aws_route" "public_internet_gateway" {
  + destination_cidr_block = "0.0.0.0/0"
}
```

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply".




```
PS E:\assingment\application> terraform apply
data.template_file.user_data: Reading...
data.template_file.user_data: Read complete after 0s [id=67cb15a07cc39847133978765474b839b7d51932047b23beb4cd83ea4ef3a74f]
data.aws_subnets.available_db_subnet: Reading...
data.aws_security_group.tcw_sg: Reading...
data.aws_vpc.vpc_available: Reading...
data.aws_ami.amazon_linux_2: Reading...
data.aws_subnets.available_db_subnet: Read complete after 1s [id=us-east-1]
data.aws_security_group.tcw_sg: Read complete after 1s [id=sg-07669d2f51270c167]
data.aws_ami.amazon_linux_2: Read complete after 1s [id=ami-02538f8925e3aa27a]
data.aws_vpc.vpc_available: Read complete after 2s [id=vpc-0e8271fd00b4a8dc8]
```

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

- Same you have to do with the Rds creation and application all the variable are there in the variable.tf
- Command which you have to use i;e:-- terraform init , terraform plan, terraform apply

his PC > New Volume (E:) > assingment >				
Name	Date modified	Type	Size	
 application	24-08-2022 06:59	File folder		
 rds	24-08-2022 07:05	File folder		
 vpc	24-08-2022 07:06	File folder		

- After creation of vpc , Rds go to aws console and check once whether it is created or not as shown on the image below :--

[Create VPC](#) [Launch EC2 Instances](#)

Note: Your Instances will launch in the US East region.

Resources by Region

[Refresh Resources](#)

You are using the following Amazon VPC resources

VPCs See all regions ▼	US East 2	NAT Gateways See all regions ▼	US East 0
Subnets See all regions ▼	US East 10	VPC Peering Connections See all regions ▼	US East 0
Route Tables See all regions ▼	US East 4	Network ACLs See all regions ▼	US East 2
Internet Gateways See all regions ▼	US East 2	Security Groups See all regions ▼	US East 9
Egress-only Internet Gateways See all regions ▼	US East 0	Customer Gateways See all regions ▼	US East 0
DHCP option sets See all regions ▼	US East 1	Virtual Private Gateways See all regions ▼	US East 0
Elastic IPs See all regions ▼	US East 0	Site-to-Site VPN Connections See all regions ▼	US East 0
Endpoints See all regions ▼	US East 0	Running Instances See all regions ▼	US East 0

It will be like this as shown on this image:--

Your VPCs (2) Info

Filter VPCs

Actions Create VPC

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table
tcw_vpc	vpc-091faa145232a63a6	Available	192.168.0.0/16	-	dopt-09c0c9bf419d1d...	rtb-0169ebe0a6be145fc
-	vpc-010e8f2e96166aaaa	Available	172.31.0.0/16	-	dopt-09c0c9bf419d1d...	rtb-05d7121f93aee2ad0

Search for services, features, blogs, docs, and more [Alt+S]

Subnets (10) Info

Filter subnets

Actions Create subnet

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4 addresses	Avai
tcw_public_subnet_...	subnet-03402f5fee6fed0a5	Available	vpc-091faa145232a63a6 tcw...	192.168.2.0/24	-	251	US-E
tcw_public_subnet_...	subnet-0f372e52576b3dca1	Available	vpc-091faa145232a63a6 tcw...	192.168.1.0/24	-	251	US-E
tcw_database_subn...	subnet-0785059588cd1cff9	Available	vpc-091faa145232a63a6 tcw...	192.168.6.0/24	-	251	US-E
tcw_database_subn...	subnet-055dfb2eb4be56863	Available	vpc-091faa145232a63a6 tcw...	192.168.5.0/24	-	251	US-E
subnet_2	subnet-07abfe9af9bc4abb2	Available	vpc-010e8f2e96166aaaa	172.31.48.0/20	-	4091	US-E
subnet_1	subnet-07c024e9c4bba62ea	Available	vpc-010e8f2e96166aaaa	172.31.64.0/20	-	4091	US-E
-	subnet-0281236047ce6a16f	Available	vpc-010e8f2e96166aaaa	172.31.0.0/20	-	4091	US-E
-	subnet-0cd25b8262777649e	Available	vpc-010e8f2e96166aaaa	172.31.32.0/20	-	4091	US-E
-	subnet-024f7f6a315c52b82	Available	vpc-010e8f2e96166aaaa	172.31.80.0/20	-	4091	US-E
-	subnet-0b8c104991bc02786	Available	vpc-010e8f2e96166aaaa	172.31.16.0/20	-	4091	US-E

- Internet gateway and route table will be like this as shown below

Internet gateways (2) [Info](#)

Filter internet gateways

< 1 > ⚙

<input type="checkbox"/>	Name ▾	Internet gateway ID ▾	State ▾	VPC ID ▾	Owner ▾
<input type="checkbox"/>	-	igw-05cec8dc54cbefd33	Attached	vpc-010e8f2e96166aaaa	107190336424
<input type="checkbox"/>	tcw_igw	igw-0a75b094a60b3e28e	Attached	vpc-091faa145232a63a6 tcw_vpc	107190336424

Route tables (4) [Info](#)

Filter route tables

< 1 > ⚙

<input type="checkbox"/>	Name ▾	Route table ID ▾	Explicit subnet associat... ▾	Edge associations ▾	Main ▾	VPC ▾	Owner ID ▾
<input type="checkbox"/>	tcw_public_route_t...	rtb-0fb81db3a2b843575	2 subnets	-	No	vpc-091faa145232a63a6 tcw...	107190336424
<input type="checkbox"/>	tcw_database_rout...	rtb-0d075b1aff1fd77e2	2 subnets	-	No	vpc-091faa145232a63a6 tcw...	107190336424
<input type="checkbox"/>	-	rtb-0169ebe0a6be145fc	-	-	Yes	vpc-091faa145232a63a6 tcw...	107190336424
<input type="checkbox"/>	-	rtb-05d7121f93aee2ad0	-	-	Yes	vpc-010e8f2e96166aaaa	107190336424

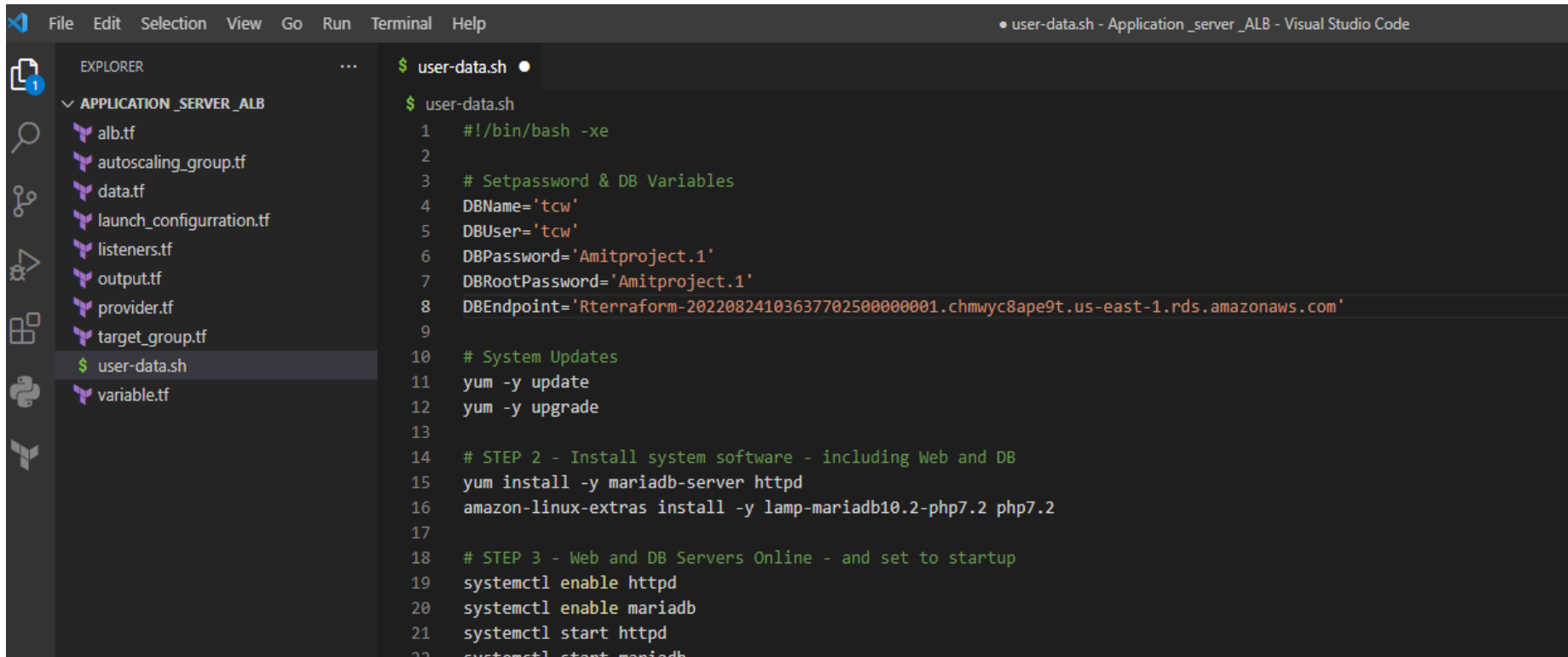
- To change user engine, name and password of the database go to dbs folder under there there is a variable.tf all the variable like database name password are there:-

```
EXPLORER
...
db_instance.tf U
data.tf U
db_subnet_group.tf U
output.tf U
variables.tf U X

RDS
> .terraform
.terraform.lock.hcl U
data.tf U
db_instance.tf U
db_subnet_group.tf U
output.tf U
provider.tf U
{} terraform.tfstate U
terraform.tfstate.backup U
variables.tf U

variables.tf > variable "pass" > default
1 variable "engine_name" {
2   description = "Enter the DB engine"
3   type        = string
4   default     = "mysql"
5 }
6
7
8 variable "db_name" {
9   description = "Enter the name of the database to be created inside DB Instance"
10  type        = string
11  default     = "tcw"
12 }
13 variable "user_name" {
14   description = "Enter the username for DB"
15   type        = string
16   default     = "tcw"
17 }
18 variable "pass" {
19   description = "Enter the username for DB"
20   type        = string
21   default     = "amit0987"
22 }
23 variable "multi_az_deployment" {
24   description = "Enable or disable multi-az deployment"
25   type        = bool
26   default     = false
27 }
28 variable "public_access" {
29   description = "Whether public access needed"
30   type        = bool
31   default     = false
32 }
33 variable "skip_finalSnapshot" {
34   type        = bool
35   default     = true
36 }
37 variable "delete_automated_backup" {
38   type        = bool
```

- After creating database lastly come to the application folder
- Now go to user-data.sh if you modified the database name, username and pass you have to put here the modified Dbname, Dbuser, DBPassword and for DBEndpoint you have to go to aws console search RDS then Copy the endpoint which is provided there



The screenshot shows the Visual Studio Code interface with the Explorer panel on the left displaying the file structure of the 'APPLICATION_SERVER_ALB' folder. The 'user-data.sh' file is selected. The main editor area shows the content of 'user-data.sh', which is a shell script for configuring an EC2 instance. The script includes comments for setting database variables, system updates, and installing web and database software.

```
File Edit Selection View Go Run Terminal Help
• user-data.sh - Application_server_ALB - Visual Studio Code

EXPLORER
APPLICATION_SERVER_ALB
  alb.tf
  autoscaling_group.tf
  data.tf
  launch_configuration.tf
  listeners.tf
  output.tf
  provider.tf
  target_group.tf
  user-data.sh
  variable.tf


$ user-data.sh
$ user-data.sh
1  #!/bin/bash -xe
2
3  # Setpassword & DB Variables
4  DBName='tcw'
5  DBUser='tcw'
6  DBPassword='Amitproject.1'
7  DBRootPassword='Amitproject.1'
8  DBEndpoint='Rterraform-20220824103637702500000001.chmwyc8ape9t.us-east-1.rds.amazonaws.com'
9
10 # System Updates
11 yum -y update
12 yum -y upgrade
13
14 # STEP 2 - Install system software - including Web and DB
15 yum install -y mariadb-server httpd
16 amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
17
18 # STEP 3 - Web and DB Servers Online - and set to startup
19 systemctl enable httpd
20 systemctl enable mariadb
21 systemctl start httpd
22 systemctl start mariadb
```

- Endpoint will be like this as shown below

RDS > Databases > terraform-20220824103637702500000001

terraform-20220824103637702500000001

Summary

DB identifier terraform-20220824103637702500000001	CPU <div><div></div></div> 8.79%	Status  Available
Role Instance	Current activity <div><div></div></div> 0 Connections	Engine MySQL Communi

[Connectivity & security](#) | [Monitoring](#) | [Logs & events](#) | [Configuration](#) | [Maintenance & backups](#) | [Tags](#)

Connectivity & security

Endpoint & port <div>Endpoint terraform-20220824103637702500000001.chmwyc8ape9t.us-east-1.rds.amazonaws.com</div> <div>Port 3306</div>	Networking <div>Availability Zone us-east-1b</div> <div>VPC tcw_vpc (vpc-091faa145232a63a6)</div> <div>Subnet group</div>
---	--

- After doing all this step in application which is in this folder type terraform apply
- And to remove simply type terraform destroy whatever you want to remove whether it is vpc, rds, or application server
- After apply go to the Aws console copy the public ip and put it on the browser address bar and click ok


The screenshot displays the AWS Management Console interface. The top navigation bar shows the AWS logo, a search bar, and the text "New EC2 Experience". The left sidebar contains a navigation menu with options like "EC2 Dashboard", "Instances", "Images", "Elastic Block Store", and "Network & Security". The main content area is titled "Instances (1/1) Info" and shows a table with one instance: "tcw-wordpress-app-server" with ID "i-06e930a66cd5960d3", state "Running", and type "t2.micro". Below this, the "Instance: i-06e930a66cd5960d3 (tcw-wordpress-app-server)" details are shown. The "Details" tab is active, displaying various attributes. The "Public IPv4 address" is highlighted with a red box, showing the value "44.198.184.10" and a link to "open address".

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
tcw-wordpress-app-server	i-06e930a66cd5960d3	Running	t2.micro	2/2 checks passed	No alarms

Attribute	Value
Instance ID	i-06e930a66cd5960d3 (tcw-wordpress-app-server)
IPv6 address	-
Hostname type	IP name: ip-192-168-1-76.ec2.internal
Answer private resource DNS name	-
Auto-assigned IP address	-
Public IPv4 address	44.198.184.10 open address
Instance state	Running
Private IP DNS name (IPv4 only)	ip-192-168-1-76.ec2.internal
Instance type	t2.micro
VPC ID	-

HURRAY! All Done it will look like this:---

← → ↻ ⚠ Not secure | 44.198.184.10/wp-admin/install.php



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title

Username

Username can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password

R\$goFp\$xCOk9tEChTA

Hide

Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email

Double-check your email address before continuing.

Search engine visibility

☐ Discourage search engines from indexing this site.
It is up to search engines to honor this request.

Install WordPress

tcw-terraform-cour...zip

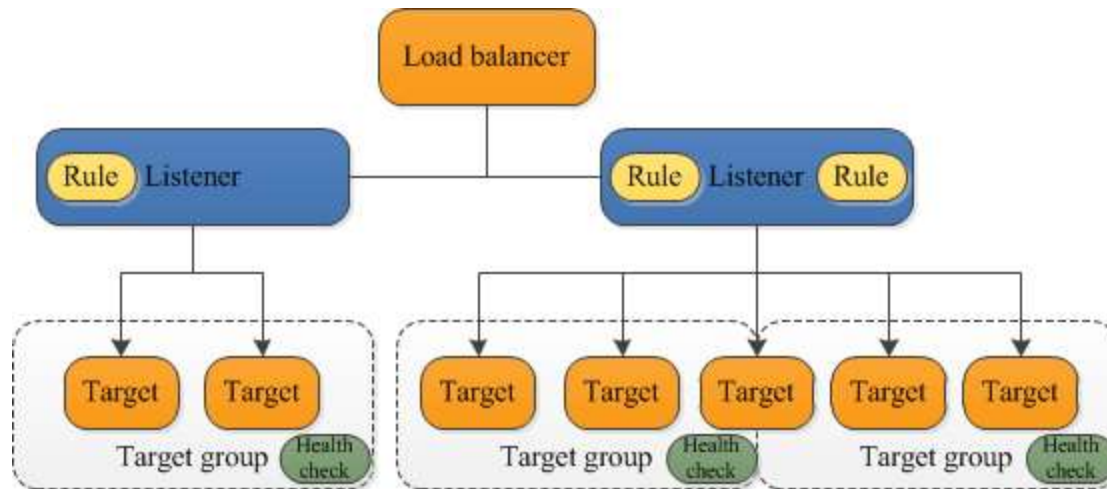
amit.pok

newkev.pem

Advantages:--

Of having Application load balancer and Autoscaling are as follows:--

The load balancer distributes incoming application traffic across multiple targets, such as EC2 instances, in multiple Availability Zones. This increases the availability of your application. You add one or more listeners to your load balancer.

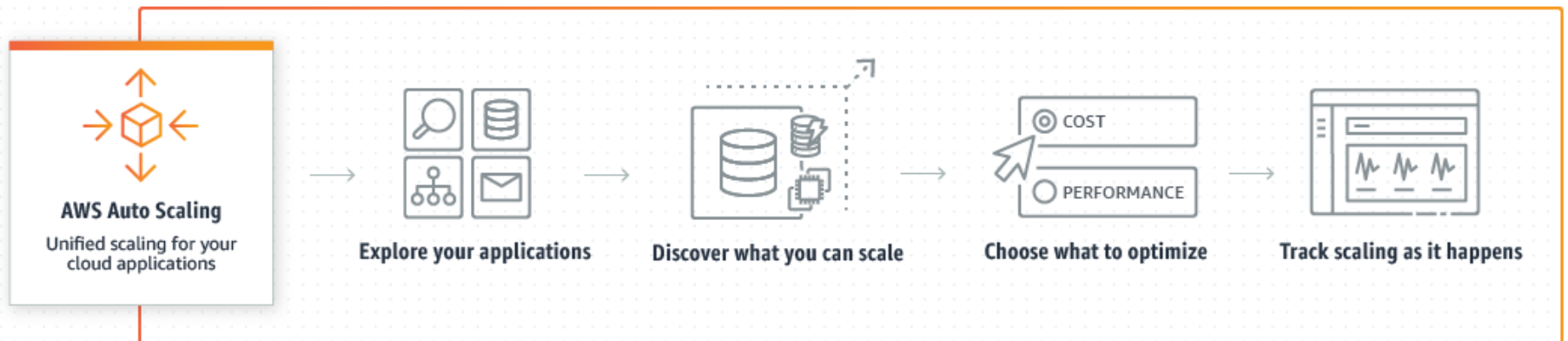


Autoscaling:-

- SETUP SCALING QUICKLY

- AUTOMATICALLY MAINTAIN PERFORMANCE

- AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. Using AWS Auto Scaling, it's easy to setup application scaling for multiple resources across multiple services in minutes. The service provides a simple, powerful user interface that lets you build scaling plans for resources including Amazon EC2 instances and Spot Fleets, Amazon ECS tasks, Amazon DynamoDB tables and indexes, and Amazon Aurora Replicas. AWS Auto Scaling makes scaling simple with recommendations that allow you to optimize performance, costs, or balance between them. If you're already using Amazon EC2 Auto Scaling to dynamically scale your Amazon EC2 instances, you can now combine it with AWS Auto Scaling to scale additional resources for other AWS services. With AWS Auto Scaling, your applications always have the right resources at the right time.



THANK YOU

For any further clarifications/patch assistance, please

contact:- 9113446714

email:- amit.ranjan.akr@gmail.com