



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Semester: V

Academic Year: 2024-25

Class / Branch: TE IT

Subject: Advanced Devops Lab (ADL)

Name of Instructor: Prof. Manjusha K.

Name of Student: Harsh Prajapati

Student ID: 22104188

EXPERIMENT NO. 06

Aim: To Build, change, and destroy AWS infrastructure Using Terraform.

Pre-requisites:

1. Install the AWS CLI version 2 on Linux

Follow these steps from the command line to install the AWS CLI on Linux.

Install curl on linux

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ sudo apt-get install curl
[sudo] password for apsit:
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.58.0-2ubuntu3.24).
0 upgraded, 0 newly installed, 0 to remove and 14 not upgraded.
```

vishal@apsit:~\$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"

```
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 57.9M 100 57.9M 0 0 12.6M 0 0:00:04 0:00:04 --:--:-- 13.3M
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit#
```

vishal@apsit:~\$ sudo apt install unzip



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



```
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit# sudo apt install unzip
Reading package lists... Done
Building dependency tree
Reading state information... Done
unzip is already the newest version (6.0-21ubuntu1.2).
0 upgraded, 0 newly installed, 0 to remove and 14 not upgraded.
```

vishal@apsit:~\$ sudo unzip awscliv2.zip

```
:~$ sudo unzip awscliv2.zip
```

vishal@apsit:~\$ sudo ./aws/install

```
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit# aws --version
aws-cli/2.13.11 Python/3.11.4 Linux/5.4.0-150-generic exe/x86_64.ubuntu.18 prompt/off
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit#
```

vishal@apsit:~\$ aws --version

it should display the below output.

aws-cli/2.1.29 Python/3.8.8 Linux/5.4.0-1038-aws exe/x86_64.ubuntu.18 prompt/off

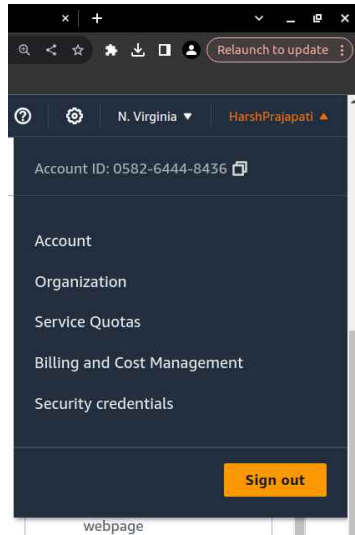
```
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit# aws --version
aws-cli/2.13.11 Python/3.11.4 Linux/5.4.0-150-generic exe/x86_64.ubuntu.18 prompt/off
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit#
```

2. Create a new access key if you don't have one. Make sure you download the keys in your local machine.

Login to AWS console, click on username and go to My security credentials.



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)



Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, see [AWS IAM Users](#). To learn more about the types of AWS credentials and how they're used, see [AWS Security Credentials](#).

▲ Password

▲ Multi-factor authentication (MFA)

▼ Access keys (access key ID and secret access key)

Use access keys to make programmatic calls to AWS from the AWS CLI, Tools for PowerShell, AWS SDKs, and other tools.

For your protection, you should never share your secret keys with anyone. As a best practice, we recommend that you rotate your secret keys regularly. **If you lose or forget your secret key, you cannot retrieve it. Instead, create a new access key and delete the old one.**

Created	Access Key ID	Last Used
---------	---------------	-----------

Continue on security credentials, click on access keys

Perform below commands in Linux where you have installed Terraform

First setup your access keys, secret keys and region code locally.

vishal@apsit:~\$aws configure

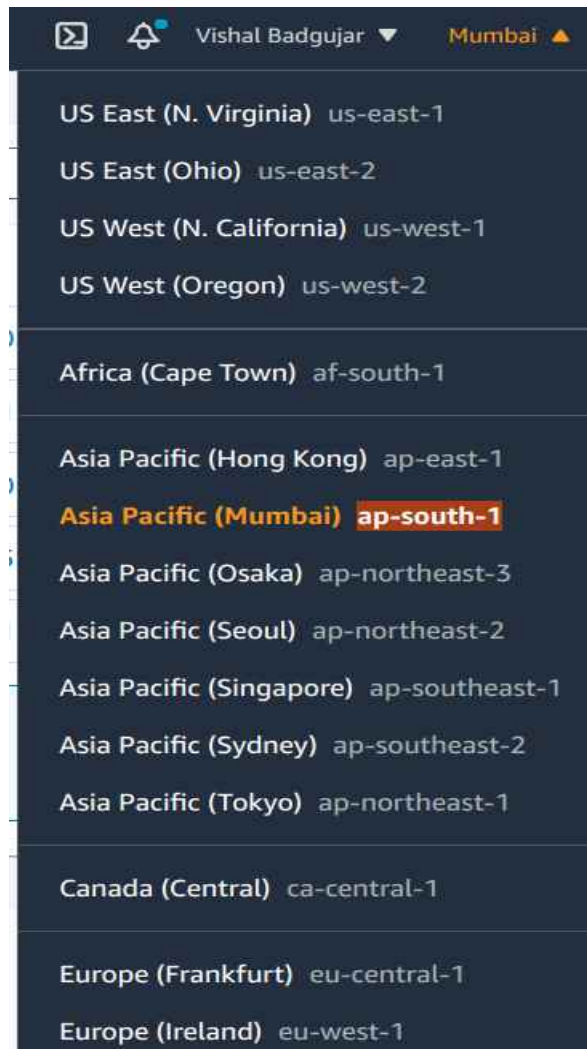
Created	Access Key ID	Last Used	Last Used Region	Last Used Service	Status
Jun 4th 2021	AKIATKYZJ6PMC2VF436	2021-07-04 21:26 UTC+0530	us-east-1	sts	Active
Aug 1st 2021	AKIATKYZJ6PMFLTCGGPV	N/A	N/A	N/A	Active



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)



You can check region as shown in below image :



```
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit# aws configure
AWS Access Key ID [*****3IHV]: AKIAQ3EGU5W2L5THGTUK
AWS Secret Access Key [*****cdhu]: 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
Default region name [us-east-1]: us-east-1
Default output format [None]:
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit#
```

Create one Directory for Terraform project in which all files of terraform we can save

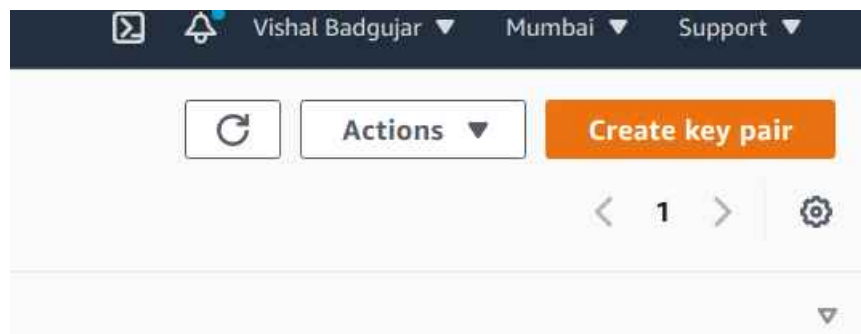


```
vishal@apsit:~$ cd ~  
vishal@apsit:~$ mkdir project-terraform  
vishal@apsit:~$ cd project-terraform
```

```
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit# mkdir HP_terraformexp6  
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit# cd Hp_terraformexp6  
bash: cd: Hp_terraformexp6: No such file or directory  
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit# cd HP_terraformexp6  
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit/HP_terraformexp6#
```

Create Terraform Files

In order to provide key name in variables first create key pair as shown:



Give name to key pair file as **terraform**



Create key pair

Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Private key file format

- ☒ .pem
For use with OpenSSH
- ☐ .ppk
For use with PuTTY

Tags (Optional)

No tags associated with the resource.

Add tag

You can add 50 more tags.

Cancel

Create key pair

Key pair is generated

<input type="checkbox"/>	terraform	d4:aa:d4:24:a8:f5:a2:2a:28:59:e6:38:d...	key-080872ef28d76fe24
--------------------------	-----------	--	-----------------------

Use your Region and Key name in variable.tf as shown and provide instance type which you want to create.



```
File Edit View Search Terminal Help
GNU nano 2.9.3 variables.tf Modified

variable "aws_region" {
  description = "The AWS region to create things in."
  default     = "ap-south-1"
}

variable "key_name" {
  description = "SSH keys to connect to ec2 instance"
  default     = "terraform"
}

variable "instance_type" {
  description = "instance type for ec2"
  default     = "t2.micro"
}

^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line  M-E Redo
```

After creating variable terraform file note down the AMI ID of instance which u want to create which we will use to configure our instance in main.tf file.



Amazon Linux 2 AMI (HVM), SSD Volume Type - `ami-04db49c0fb2215364` (64-bit x86) / ami

Amazon Linux
Free tier eligible

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance (Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux 2020 and has been removed from this wizard.

Now create main.tf file:

```
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit/HP_terraformexp6# sudo nano variables.tf
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit/HP_terraformexp6# sudo nano main.tf
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit/HP_terraformexp6#
```

```
provider "aws" {
  region = var.aws_region
}
```



#Create security group with firewall rules

```
resource "aws_security_group" "security_jenkins_port" {  
  name      = "security_jenkins_port"  
  description = "security group for jenkins"
```

```
  ingress {  
    from_port = 8080  
    to_port   = 8080  
    protocol  = "tcp"  
    cidr_blocks = ["0.0.0.0/0"]  
  }  
}
```

```
  ingress {  
    from_port = 22  
    to_port   = 22  
    protocol  = "tcp"  
    cidr_blocks = ["0.0.0.0/0"]  
  }  
}
```

outbound from jenkins server

```
  egress {  
    from_port = 0  
    to_port   = 65535  
    protocol  = "tcp"  
    cidr_blocks = ["0.0.0.0/0"]  
  }  
}
```




```
tags= {  
    Name = "security_jenkins_port"  
}  
}  
  
resource "aws_instance" "myFirstInstance" {  
    ami      = "ami-0b9064170e32bde34"  
    key_name = var.key_name  
    instance_type = var.instance_type  
    security_groups= [ "security_jenkins_port"]  
    tags= {  
        Name = "jenkins_instance"  
    }  
}  
  
# Create Elastic IP address  
resource "aws_eip" "myFirstInstance" {  
    vpc      = true  
    instance = aws_instance.myFirstInstance.id  
    tags= {  
        Name = "jenkins_elstic_ip"  
    }  
}
```

Put AMI-ID in above highlighted space and Now execute the below command:



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



```
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit/HP_terraformexp6# terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.62.0...
- Installed hashicorp/aws v5.62.0 (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.

```
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit/HP_terraformexp6#
```

you should see like below screenshot.

Execute the below command

```
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit/HP_terraformexp6# terraform plan
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_elb.myFirstInstance will be created
+ resource "aws_elb" "myFirstInstance" {
+   allocation_id      = (known after apply)
+   arn                 = (known after apply)
+   association_id     = (known after apply)
+   carrier_ip         = (known after apply)
+   customer_owned_ip  = (known after apply)
+   domain              = (known after apply)
+   id                  = (known after apply)
+   instance            = (known after apply)
+   network_border_group = (known after apply)
+   network_interface  = (known after apply)
+   private_dns        = (known after apply)
+   ptr_record          = (known after apply)
+   public_dns          = (known after apply)
+   public_ip           = (known after apply)
+   public_ipv4_pool    = (known after apply)
+   tags                = {
+     "Name" = "jenkins_elstic_ip"
+   }
+   tags_all            = {
+     "Name" = "jenkins_elstic_ip"
+   }
+   vpc                  = true
}
```

the above command will show how many resources will be added.

Plan: 3 to add, 0 to change, 0 to destroy.



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)



```
Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_eip.myfirstinstance1: Destroying... [id=eipalloc-046d7454608bc8f35]
aws_security_group.security_jenkins_port: Destroying... [id=sg-02b0c3290c31bc518]
aws_eip.myfirstinstance1: Destruction complete after 6s
aws_instance.myfirstinstance1: Destroying... [id=i-0f3b6ff88a3471ec5]
aws_security_group.security_jenkins_port: Still destroying... [id=sg-02b0c3290c31bc518, 10s elapsed]
aws_instance.myfirstinstance1: Still destroying... [id=i-0f3b6ff88a3471ec5, 10s elapsed]
aws_security_group.security_jenkins_port: Still destroying... [id=sg-02b0c3290c31bc518, 20s elapsed]
aws_instance.myfirstinstance1: Still destroying... [id=i-0f3b6ff88a3471ec5, 20s elapsed]
aws_security_group.security_jenkins_port: Still destroying... [id=sg-02b0c3290c31bc518, 30s elapsed]
aws_instance.myfirstinstance1: Still destroying... [id=i-0f3b6ff88a3471ec5, 30s elapsed]
aws_security_group.security_jenkins_port: Still destroying... [id=sg-02b0c3290c31bc518, 40s elapsed]
aws_security_group.security_jenkins_port: Destruction complete after 44s
aws_instance.myfirstinstance1: Still destroying... [id=i-0f3b6ff88a3471ec5, 40s elapsed]
aws_instance.myfirstinstance1: Destruction complete after 46s

Destroy complete! Resources: 3 destroyed.
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit/HP_terraformexp6#
```

Execute the below command



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



```
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.myfirstinstance1: Creating...
aws_instance.myfirstinstance1: Still creating... [10s elapsed]
aws_instance.myfirstinstance1: Still creating... [20s elapsed]
aws_instance.myfirstinstance1: Still creating... [30s elapsed]
aws_instance.myfirstinstance1: Creation complete after 36s [id=i-0f3b8ff88a3471ec5]
aws_eip.myfirstinstance1: Creating...
aws_eip.myfirstinstance1: Creation complete after 3s [id=eipalloc-046d7454608bc8f35]

Warning: Argument is deprecated

with aws_eip.myfirstinstance1,
on main.tf line 44, in resource "aws_eip" "myfirstinstance1":
44: vpc = true

use domain attribute instead

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit/HP_terraformexp6#
```

Provide the value as Yes for applying terraform

Plan: 3 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

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.



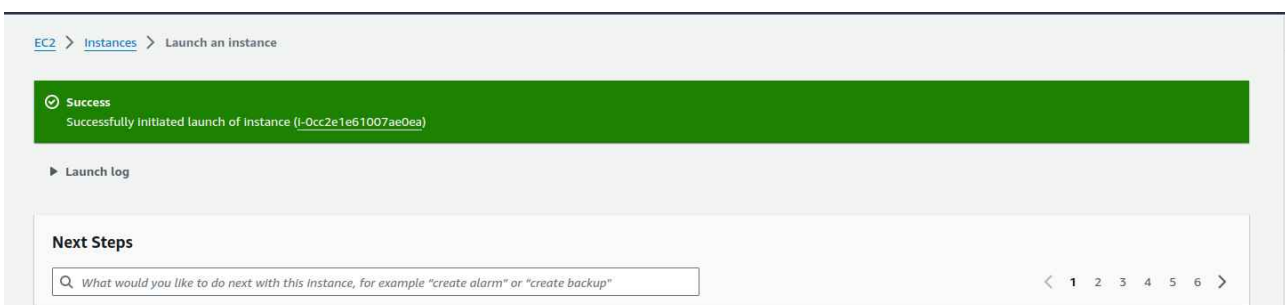
```
Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_eip.myfirstinstance1: Destroying... [id=eipalloc-046d7454608bc8f35]
aws_security_group.security_jenkins_port: Destroying... [id=sg-02b0c3290c31bc518]
aws_eip.myfirstinstance1: Destruction complete after 6s
aws_instance.myfirstinstance1: Destroying... [id=i-0f3b6ff88a3471ec5]
aws_security_group.security_jenkins_port: Still destroying... [id=sg-02b0c3290c31bc518, 10s elapsed]
aws_instance.myfirstinstance1: Still destroying... [id=i-0f3b6ff88a3471ec5, 10s elapsed]
aws_security_group.security_jenkins_port: Still destroying... [id=sg-02b0c3290c31bc518, 20s elapsed]
aws_instance.myfirstinstance1: Still destroying... [id=i-0f3b6ff88a3471ec5, 20s elapsed]
aws_security_group.security_jenkins_port: Still destroying... [id=sg-02b0c3290c31bc518, 30s elapsed]
aws_instance.myfirstinstance1: Still destroying... [id=i-0f3b6ff88a3471ec5, 30s elapsed]
aws_security_group.security_jenkins_port: Still destroying... [id=sg-02b0c3290c31bc518, 40s elapsed]
aws_security_group.security_jenkins_port: Destruction complete after 44s
aws_instance.myfirstinstance1: Still destroying... [id=i-0f3b6ff88a3471ec5, 40s elapsed]
aws_instance.myfirstinstance1: Destruction complete after 46s

Destroy complete! Resources: 3 destroyed.
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit/HP_terraformexp6#
```

Now login to EC2 console, to see the new instances up and running, you can see Jenkins_instance is up and running which we deploy from terraform.



You can also check the security group resource details which you created from terraform :

Terraform destroy

you can also destroy or delete your instance by using terraform destroy command :



```
Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_eip.myfirstinstance1: Destroying... [id=eipalloc-046d7454608bc8f35]
aws_security_group.security_jenkins_port: Destroying... [id=sg-02b0c3290c31bc518]
aws_eip.myfirstinstance1: Destruction complete after 6s
aws_instance.myfirstinstance1: Destroying... [id=i-0f3b6ff88a3471ec5]
aws_security_group.security_jenkins_port: Still destroying... [id=sg-02b0c3290c31bc518, 10s elapsed]
aws_instance.myfirstinstance1: Still destroying... [id=i-0f3b6ff88a3471ec5, 10s elapsed]
aws_security_group.security_jenkins_port: Still destroying... [id=sg-02b0c3290c31bc518, 20s elapsed]
aws_instance.myfirstinstance1: Still destroying... [id=i-0f3b6ff88a3471ec5, 20s elapsed]
aws_security_group.security_jenkins_port: Still destroying... [id=sg-02b0c3290c31bc518, 30s elapsed]
aws_instance.myfirstinstance1: Still destroying... [id=i-0f3b6ff88a3471ec5, 30s elapsed]
aws_security_group.security_jenkins_port: Still destroying... [id=sg-02b0c3290c31bc518, 40s elapsed]
aws_instance.myfirstinstance1: Still destroying... [id=i-0f3b6ff88a3471ec5, 40s elapsed]
aws_security_group.security_jenkins_port: Destruction complete after 44s
aws_instance.myfirstinstance1: Still destroying... [id=i-0f3b6ff88a3471ec5, 40s elapsed]
aws_instance.myfirstinstance1: Destruction complete after 46s

Destroy complete! Resources: 3 destroyed.
root@apsit-HP-280-Pro-G6-Microtower-PC:/home/apsit/HP_terraformexp6#
```

Now you can see instance which you created by using terraform is deleted successfully from aws console also you can check it will removed successfully:



All the Resources including Security groups, EC2 instances using terraform will be deleted. In this way we can automate infrastructure set up using terraform in aws cloud.

Conclusion: In this experiment we downloaded Curl and used terraform and created file with extension .tf and we initialized it and destroyed it using terraform command