## Assignment - 5:-

## Q) By Using Provisioners copy file from desktop to Ec2 Instance?

- **A.)** <u>Terraform:</u> It is an infrastructure as code tool that enables you to safely and predictably provision and manage infrastructure in any cloud. They are 8 blocks in terrafor;
  - 1. **Terraform Block** It is used to define global configuration and behaviour for terraform execution.
  - 2. **Provider Block** Configuring the provider for a specific cloud or infrastructure platform.
  - 3. **Data Block** This block is used to fetch data from external sources or existing resources.
  - 4. **Resource Block** It is used to declare and define the provider for a specific cloud or infrastructure program.
  - 5. **Module Block** Defining and configuring reusable modules to encapsulate and manage infrastructure components.
  - 6. **Variable Block** Declaring input variables that can be provided during Terraform execution for flexible configurations.
  - 7. **Output Block** Defining values that are displayed as output after executing terraform apply or terraform output commands.
  - 8. **Locals Block** Declaring local variables within the Terraform configuration for easier code readability and reusability.

<u>Provisioner's:</u> provisioner is a powerful feature that allows you to execute scripts or perform specific actions during the resource creation or destruction process. These provisioners enable additional configuration and setup tasks that can't be accomplished with Terraform's declarative syntax alone.

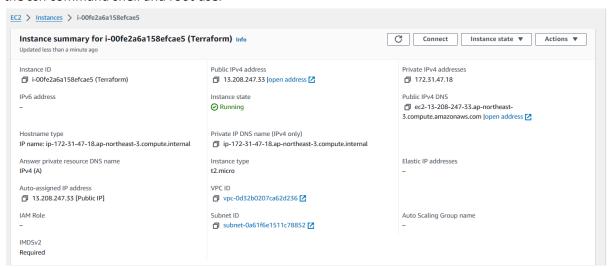
## **Types of Provisioners:**

Terraform includes several built-in provisioners, such as:

- **File provisioner**: Copies files to a remote machine.
- Remote-exec provisioner: Executes commands on a remote machine over SSH or WinRM.
- Local-exec provisioner: Executes commands on the local machine.
- You can also use third-party provisioners by placing them in specific directories.

## **Practical:**

 Let's launch an Ec2 instance with a key pair, network settings, name and storage connect to the ssh command shell and root user



```
ubuntu@ip-172-31-47-18:~$ sudo -i
root@ip-172-31-47-18:~# []
```

Now, update the packages

```
root@ip-172-31-47-18:~# apt update -y
Hit:1 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:5 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy/miverse amd04 rackages [141 Mb]
Get:6 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
  et:9 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
  set:10 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Set:11 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1518 kB]
Set:12 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [293 kB]
                                                                                                                                jammy-updates/main Tiansiation-en [293 kB]
jammy-updates/restricted amd64 Packages [1644 kB]
jammy-updates/restricted Translation-en [274 kB]
jammy-updates/universe amd64 Packages [1060 kB]
jammy-updates/universe Translation-en [241 kB]
   et:13 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu
  Get:14 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu
  et:15 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu
  et:16 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu
                                                                                                                                jammy-updates/universe amd64 c-n-f Metadata [22.1 kB]
jammy-updates/multiverse amd64 Packages [49.6 kB]
  et:17 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu
  et:18 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu
                                                                                                                                jammy-updates/multiverse Translation-en [12.0 kB]
jammy-updates/multiverse Translation-en [12.0 kB]
jammy-updates/multiverse amd64 c-n-f Metadata [47
jammy-backports/main amd64 Packages [67.1 kB]
jammy-backports/main Translation-en [11.0 kB]
  Get:19 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu
  Get:20 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu
  Set:21 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu
Set:22 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu
                                                                                                                                jammy-backports/main amd64 c-n-f Metadata [388 B]
jammy-backports/restricted amd64 c-n-f Metadata [116 B]
jammy-backports/universe amd64 Packages [28.4 kB]
 Get:23 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu
 Get:24 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu
Get:25 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu
Get:26 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu
                                                                                                                                        my-backports/universe Translation-
```

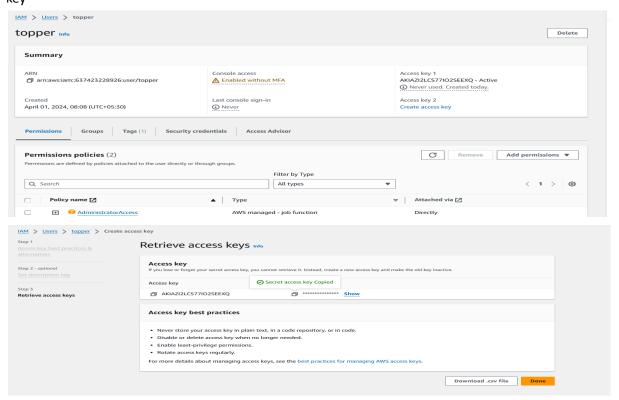
 Install awscli(amazon web services command line interface) for connecting to the aws resources

```
root@ip-172-31-47-18:-f apt install awscli -y
Reading package lists... Done
Reading state information... Done
Reading state information... Done
Reading state information... Done
Reading additional packages will be installed:
bzip2 docutils-common fontconfig fontconfig-config fonts-droid-fallback fonts-noto-mono fonts-urw-base35 ghostscript groff gsfonts hicolor-icon-theme imagemagick
imagemagick-6-common imagemagick-6-d.qfd libama3 libavahi-client3 libavahi-common-data libavahi-common3 libcairo2 libcups2 libdatriel libidavid5 libide265-0 libdeflate0
libdjvulibre-text libdjvulibre21 libfftw3-double3 libfontconfig1 libgomp1 libgraphite2-3 libgs9 libgs9-common libharfbuz20b libheif1 libice6 libidali2 libijs-0.35 libilmapseaptant0 libbjbjdj0 libipidg26c0 libipgeg-turbo8 libjpeg9 libjxr-tools libjxr0 liblcms2-2 liblcms2-2 libqcr-1-0 libltalid7 libmagickcore-6.qf6-6-extra
libmagickwand-6.qf6-6 libnetpkm10 libopenexr25 libopenjp2-7 libpango-1.0-0 libpangocatro-1.0-0 libpangot2-1.0-0 libpaper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-tibjpepper-t
```

Install terraform

```
rect@p.172-31-47.86.-4 wget -0- https://apt.releases.hashicorp.com/gpp | sudo gpg --dearmor -o /ustr/shart/keyrings/hashicorp-archive-keyring.gpg https://apt.releases.hashicorp.com $(lsb_release -cs) main* | sudo tee /etc/apt/sources.list.d/hashicorp.com (apt.releases.hashicorp.com (apt.releases.hashicorp.com (apt.releases.hashicorp.com) | sudo apt update 44 sudo apt install terraform --2024-04-01 02:50:58- https://apt.releases.hashicorp.com/gpg | Resolving apt.releases.hashicorp.com (apt.releases.hashicorp.com) | 3.163.198.36|:443... connected. | Connecting to apt.releases.hashicorp.com (apt.releases.hashicorp.com) | 3.163.198.36|:443... connected. | Connecting to apt.releases.hashicorp.com | 200 CK | Connecting to apt.releases.hashicorp.com | 200 CK | Connecting to apt.releases.hashicorp.com | 3.163.198.36|:443... connected. | Connecting to apt.releases.hashicorp.com | 3.163.198.36|:443... | Connecting to apt
```

 Create a lam user with attached policies (administration\_fullaccess) and also create access key



 Now, configure awscli and pass the access\_key and sceret\_access\_key to integrate with terraform

```
root@ip-172-31-47-18:~# aws configure
AWS Access Key ID [None]: AKIAZI2LCS77I02SEEXQ
AWS Secret Access Key [None]: VEEodOOEhLSBmOcdu8//a0lpMS7iMP4/sIa7Lv7A
Default region name [None]: ap-northeast-3
Default output format [None]: table
```

• Create a directory for terraform and change present working directory to terraform

```
root@ip-172-31-47-18:~# mkdir terraform
root@ip-172-31-47-18:~# cd terraform
root@ip-172-31-47-18:~/terraform# [
```

• Add a Terraform\_block.tf file and define the global configuration

```
required providers {
  aws = {
    source = "hashicorp/aws"
    version = "5.43.0"
}
}

terraform_block.tf" 9L, 116B
```

Add a Provider\_block.tf and define the provider and pass the access\_key

• Add a provisioner\_block.tf and define the source and destination path of the file

 Now, we can see that there will be new folder created in terraform directory from local system

```
coot@ip-172-31-47-18:~/terraform# ls
Provider_block.tf Provisioner_block.tf dheer terraform_block.tf
coot@ip-172-31-47-18:~/terraform# []
```