

Assignment - 5:-

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

A.) Terraform: 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 terraform;

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.

Provisioner's: 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

EC2 > Instances > i-00fe2a6a158efcae5

Instance summary for i-00fe2a6a158efcae5 (Terraform) [Info](#)

Updated less than a minute ago

[Refresh](#) [Connect](#) [Instance state ▼](#) [Actions ▼](#)

Instance ID i-00fe2a6a158efcae5 (Terraform)	Public IPv4 address 13.208.247.33 open address	Private IPv4 addresses 172.31.47.18
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-13-208-247-33.ap-northeast-3.compute.amazonaws.com open address
Hostname type IP name: ip-172-31-47-18.ap-northeast-3.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-47-18.ap-northeast-3.compute.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	Auto Scaling Group name -
Auto-assigned IP address 13.208.247.33 [Public IP]	VPC ID vpc-0d32b0207ca62d236	
IAM Role -	Subnet ID subnet-0a61f6e1511c78852	
IMDSv2 Required		

```
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://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
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]
Get:9 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1518 kB]
Get:12 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [293 kB]
Get:13 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1644 kB]
Get:14 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [274 kB]
Get:15 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1060 kB]
Get:16 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [241 kB]
Get:17 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [22.1 kB]
Get:18 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [49.6 kB]
Get:19 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [12.0 kB]
Get:20 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [472 B]
Get:21 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [67.1 kB]
Get:22 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [11.0 kB]
Get:23 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:24 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:25 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [28.4 kB]
Get:26 http://ap-northeast-3.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.2 kB]
```


- Now, configure awscli and pass the access_key and secret_access_key to integrate with terraform

```
root@ip-172-31-47-18:~# aws configure
AWS Access Key ID [None]: AKIAZI2LCS77IO2SEEXQ
AWS Secret Access Key [None]: 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
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

```
region = "ap-northeast-3"
profile = "default"
}
```

"Provider_block.tf" 5L, 71B

- Add a provisioner_block.tf and define the source and destination path of the file

```
provisioner "file" {
  source      = "C:\Users\dheer"
  destination = "/root/terraform"
}
```

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

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