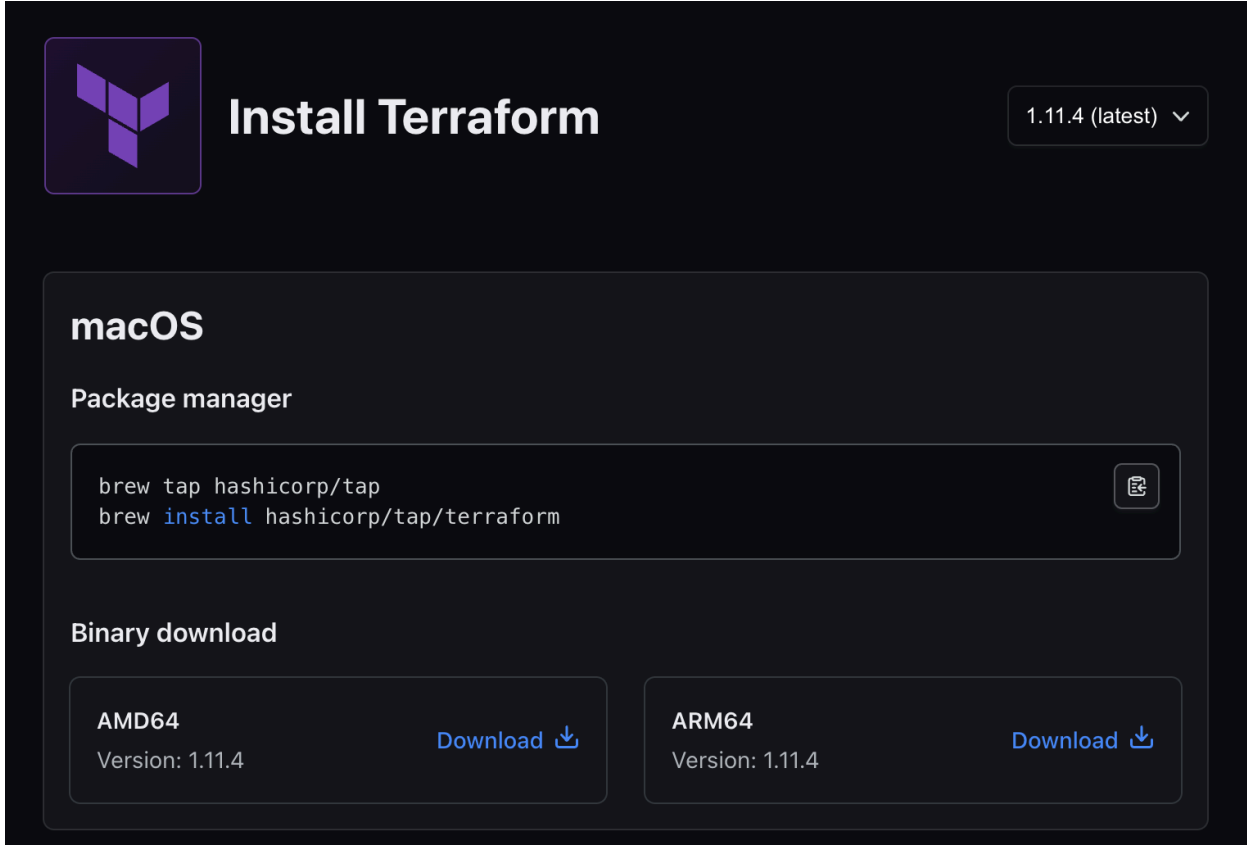


ASSIGNMENT NO.7

1. Install Terraform and verify with `terraform -v`



The image shows the Terraform installation page for macOS. At the top left is the Terraform logo, followed by the heading "Install Terraform". On the top right, there is a dropdown menu showing "1.11.4 (latest)". Below this, the "macOS" section is highlighted. Under "Package manager", there is a code block with the following commands:

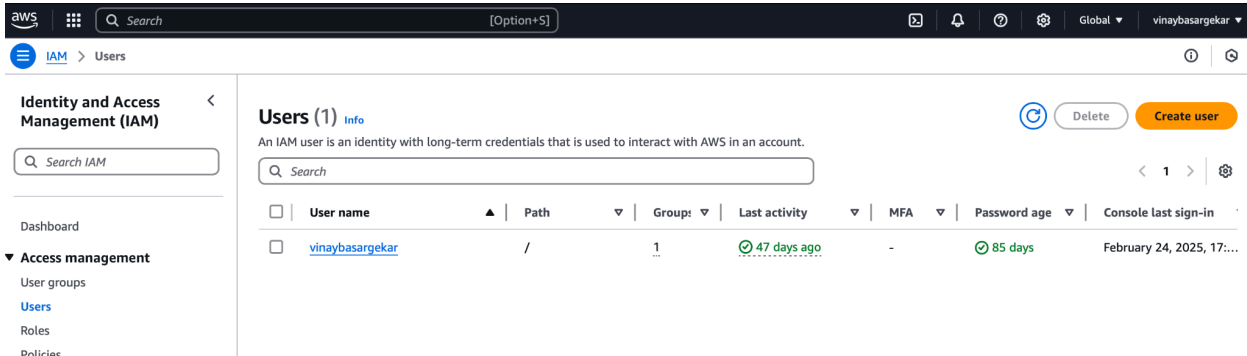
```
brew tap hashicorp/tap
brew install hashicorp/tap/terraform
```

To the right of the code block is a copy icon. Below the package manager section is the "Binary download" section, which contains two boxes for "AMD64" and "ARM64". Each box shows the version "1.11.4" and a "Download" button with a download icon.

2. Install AWS CLI and configure with `aws configure`

- `brew install awscli`
- `aws --version`

3. Go to your AWS console and create a IAM user



The image shows the AWS IAM console. The top navigation bar includes the AWS logo, a search bar, and the text "[Option+S]". On the right, there are icons for email, notifications, and a user profile dropdown showing "Global" and "vinaybasargekar". The left sidebar shows the "IAM > Users" path. The main content area is titled "Users (1)" with an "Info" link. Below the title is a description: "An IAM user is an identity with long-term credentials that is used to interact with AWS in an account." There is a search bar and a table of users. The table has columns for "User name", "Path", "Group", "Last activity", "MFA", "Password age", and "Console last sign-in". One user is listed: "vinaybasargekar" with a path of "/", a group of "1", last activity of "47 days ago", MFA status of "-", password age of "85 days", and console last sign-in of "February 24, 2025, 17:...". At the top right of the table are "Delete" and "Create user" buttons.

User name	Path	Group	Last activity	MFA	Password age	Console last sign-in
vinaybasargekar	/	1	47 days ago	-	85 days	February 24, 2025, 17:...

4. Make sure to add permission for administrative access in “attach policies directly”

The screenshot shows the 'Set permissions' step in the AWS IAM console. On the left, a progress bar indicates three steps: 'Specify user details', 'Set permissions' (current), and 'Review and create'. The 'Permissions options' section has three radio buttons: 'Add user to group', 'Copy permissions', and 'Attach policies directly' (selected). The 'Attach policies directly' option includes a description: 'Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.' Below this, the 'Permissions policies (1/1337)' section shows a search bar and a table of policies. The table has columns for 'Policy name', 'Type', and 'Attached entities'. The 'AdministratorAccess' policy is selected, showing it is 'AWS managed - job function' and has 1 attached entity.

Policy name	Type	Attached entities
AccessAnalyzerServiceRolePolicy	AWS managed	0
AdministratorAccess	AWS managed - job function	1
AdministratorAccess_Amaliku	AWS managed	0

The screenshot shows the 'Users' page in the AWS IAM console. A green banner at the top states 'User created successfully' and provides a link to 'View user'. Below the banner, the 'Users (2)' section shows a list of users. The table has columns for 'User name', 'Path', 'Group', 'Last activity', 'MFA', 'Password age', and 'Console last sign-in'. The 'terraformUser' and 'vinaybasargekar' users are listed. The 'vinaybasargekar' user is highlighted, showing a last activity of '47 days ago' and a password age of '85 days'.

User name	Path	Group	Last activity	MFA	Password age	Console last sign-in
terraformUser	/	0	-	-	-	-
vinaybasargekar	/	1	47 days ago	-	85 days	February 24, 2025, 17:...

5. Navigate to Security Credentials and Generate a new Access Key

The screenshot shows the 'Retrieve access keys' page in the AWS IAM console for the 'terraformUser'. A green banner at the top states 'Access key created' and provides a warning: 'This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.' Below the banner, the 'Access key' section shows the 'Access key' and 'Secret access key' fields. The 'Secret access key' field is highlighted, showing the value 'af168BdRdByCtneFMOBRTXhXgRhGu2dWXAxdnaV'. The 'Access key best practices' section lists several recommendations: 'Never store your access key in plain text, in a code repository, or in code.', 'Disable or delete access key when no longer needed.', 'Enable least-privilege permissions.', and 'Rotate access keys regularly.' At the bottom, there are buttons for 'Download .csv file' and 'Done'.

Access key	Secret access key
AKIAZVMTU3776CQPELQX	af168BdRdByCtneFMOBRTXhXgRhGu2dWXAxdnaV

Access key : AKIAZVMTU3776CQPELQX

Secret access key : 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

6. Configure AWS CLI using `aws configure`

```
vinaybasargekar@Vinays-MacBook-Air ~ % aws configure

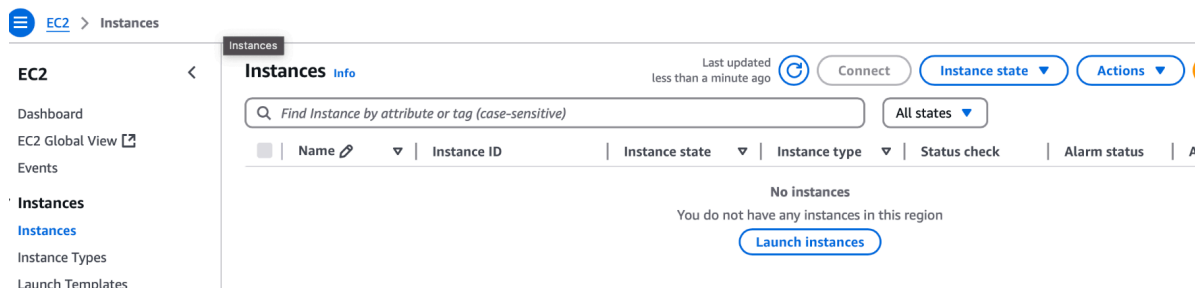
AWS Access Key ID [None]: AKIAZVMTU3776CQPELQX
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]: json
```

7. Create the Terraform Working Directory and Files

`main.tf` → The core configuration (provider + EC2 instance)

`variables.tf` → Input variables (dynamic values like AMI ID, instance type, etc.)

`outputs.tf` → Output values (like public IP, instance ID)



```
main.tf
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "~> 4.16"
6     }
7   }
8   required_version = "≥1.2.0"
9 }
10
11 provider "aws" {
12   region = var.region
13 }
14
15 resource "aws_instance" "my_ec2" {
16   ami           = var.ami_id
17   instance_type = var.instance_type
18
19   tags = {
20     Name = "Terraform-EC2"
21   }
22 }
```

```
variables.tf
1 variable "region" {
2   description = "AWS region"
3   type        = string
4   default     = "us-east-1"
5 }
6
7 variable "ami_id" {
8   description = "AMI ID for EC2"
9   type        = string
10 }
11
12 variable "instance_type" {
13   description = "EC2 instance type"
14   type        = string
15   default     = "t2.micro"
16 }
```

```

1 output "instance_id" {
2   description = "ID of the EC2 instance"
3   value       = aws_instance.my_ec2.id
4 }
5
6 output "public_ip" {
7   description = "Public IP address of the EC2 i
8   value       = aws_instance.my_ec2.public_ip
9 }
10

```

```

1 region       = "us-east-1"
2 ami_id       = "ami-00a929b66ed6e0de6"
3 instance_type = "t2.micro"
4

```

8. Initialize Terraform

```

vinaybasargekar@Vinays-MacBook-Air terraform-ec2-setup % terraform init

Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v4.67.0

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.

```

9. Plan terraform

```

vinaybasargekar@Vinays-MacBook-Air terraform-ec2-setup % 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_instance.my_ec2 will be created
+ resource "aws_instance" "my_ec2" {
  + ami                        = "ami-00a929b66ed6e0de6"
  + arn                       = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone          = (known after apply)
  + cpu_core_count             = (known after apply)
  + cpu_threads_per_core       = (known after apply)
  + disable_api_stop           = (known after apply)
  + disable_api_termination    = (known after apply)
  + ebs_optimized              = (known after apply)

```

10. Apply terraform

```
vinaybasargekar@Vinays-MacBook-Air terraform-ec2-setup % terraform apply

Terraform used the selected providers to generate the following execution plan. Resources to be created:
+ create

Terraform will perform the following actions:

# aws_instance.my_ec2 will be created
+ resource "aws_instance" "my_ec2" {
  + ami                        = "ami-00a929b66ed6e0de6"
  + arn                       = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone          = (known after apply)
  + cpu_core_count             = (known after apply)
  + cpu_threads_per_core       = (known after apply)
  + disable_api_stop           = (known after apply)
  + disable_api_termination    = (known after apply)
  + ebs_optimized              = (known after apply)
  + get_password_data          = false
  + host_id                    = (known after apply)
}
```

Changes to Outputs:

```
+ instance_id = (known after apply)
+ public_ip   = (known after apply)
```

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.my_ec2: Creating...

aws_instance.my_ec2: Still creating... [10s elapsed]

aws_instance.my_ec2: Creation complete after 16s [id=i-029f2eef2535dd310]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

instance_id = "i-029f2eef2535dd310"

public_ip = "54.242.17.51"

11. Check the instance running on AWS console

Instances

Search

[Option+S]

United States (N. Virginia)

vinaybasargekar

EC2

Instances

Instances (1) [Info](#)

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

< 1 >

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input type="checkbox"/>	Terraform-EC2	i-029f2eef2535dd310	Running	t2.micro	Initializing	View alarms +	us-east-1c