



- ❖ **Step 1: Download the Terraform** in to your PC. Install Chocolatey recommended for windows, Open power shell as administrator then paste the below command.

- ❖ First install Chocolatey (if not already installed),

```
Set-ExecutionPolicy Bypass -Scope Process -Force
[System.Net.ServicePointManager]::SecurityProtocol =
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072
iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))
```

- ❖ Install Terraform

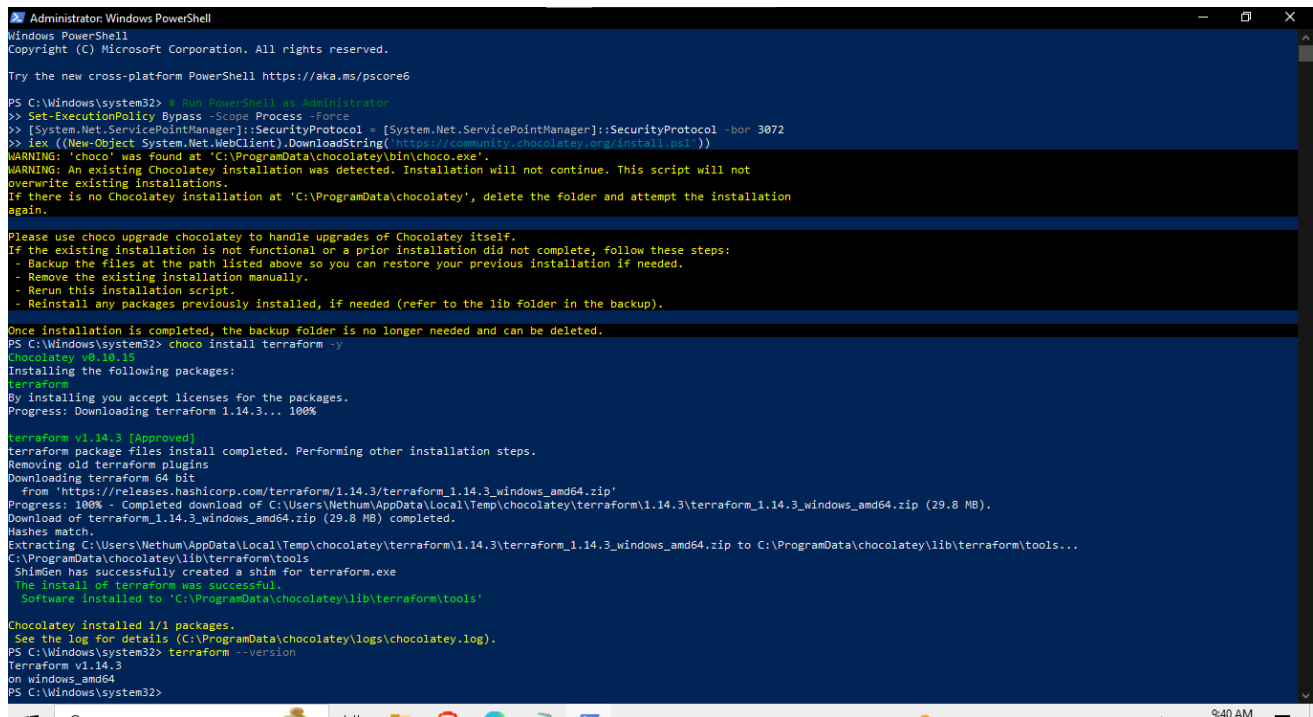
```
choco install terraform
```

- ❖ Option 2: Go to the official Terraform website, select your operating system (Windows or macOS), choose the version you need, and download the package

<https://developer.hashicorp.com/terraform/install>

After Installation done go to PowerShell or MacOs terminal just type below command then your Terraform version show like that,

```
terraform --version
```



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

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

PS C:\Windows\system32> Set-ExecutionPolicy Bypass -Scope Process -Force
> [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072
> iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))
WARNING: 'choco' was found at 'C:\ProgramData\chocolatey\bin\choco.exe'.
WARNING: An existing Chocolatey installation was detected. Installation will not continue. This script will not
overwrite existing installations.
If there is no Chocolatey installation at 'C:\ProgramData\chocolatey', delete the folder and attempt the installation
again.

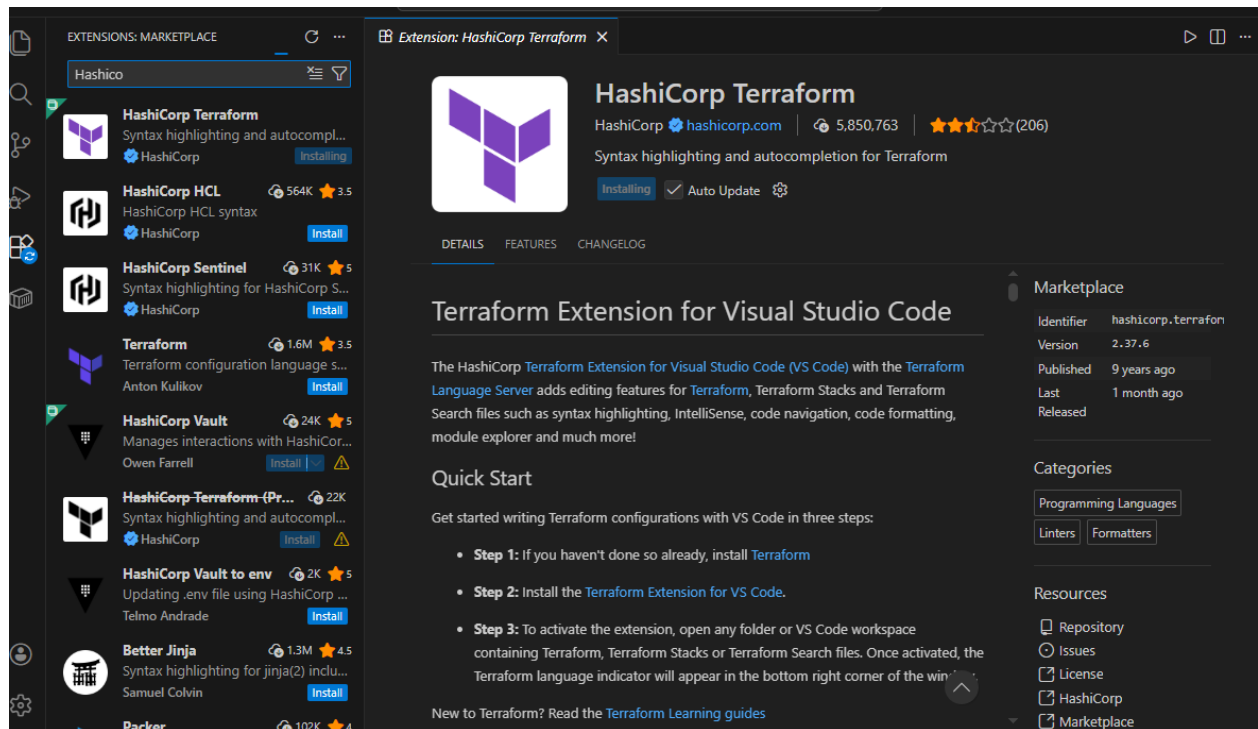
Please use choco upgrade chocolatey to handle upgrades of Chocolatey itself.
If the existing installation is not functional or a prior installation did not complete, follow these steps:
- Backup the files at the path listed above so you can restore your previous installation if needed.
- Remove the existing installation manually.
- Rerun this installation script.
- Reinstall any packages previously installed, if needed (refer to the lib folder in the backup).

Once installation is completed, the backup folder is no longer needed and can be deleted.
PS C:\Windows\system32> choco install terraform -y
Chocolatey v0.10.15
Installing the following packages:
terraform
By installing you accept licenses for the packages.
Progress: Downloading terraform 1.14.3... 100%

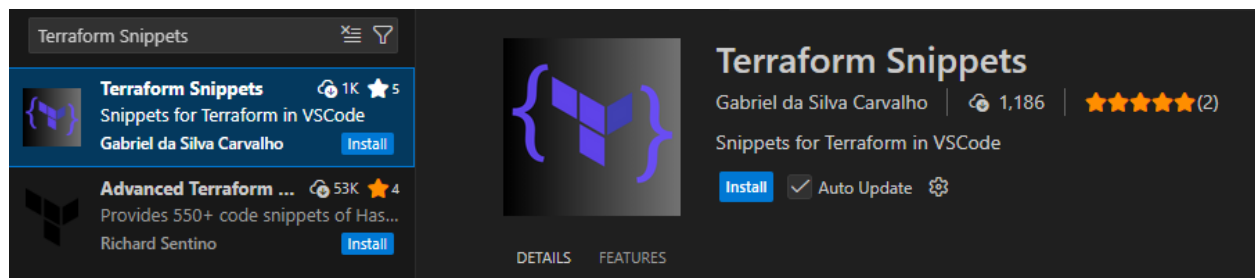
terraform v1.14.3 [Approved]
terraform package files install completed. Performing other installation steps.
Removing old terraform plugins
Downloading terraform 64 bit
from 'https://releases.hashicorp.com/terraform/1.14.3/terraform_1.14.3_windows_amd64.zip'
Progress: 100% - Completed download of C:\Users\Nethum\AppData\Local\Temp\chocolatey\terraform\1.14.3\terraform_1.14.3_windows_amd64.zip (29.8 MB).
Download of terraform_1.14.3_windows_amd64.zip (29.8 MB) completed.
Hashes match.
Extracting C:\Users\Nethum\AppData\Local\Temp\chocolatey\terraform\1.14.3\terraform_1.14.3_windows_amd64.zip to C:\ProgramData\chocolatey\lib\terraform\tools...
C:\ProgramData\chocolatey\lib\terraform\tools
Shimden has successfully created a shim for terraform.exe
The install of terraform was successful.
Software installed to 'C:\ProgramData\chocolatey\lib\terraform\tools'

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\Windows\system32> terraform --version
Terraform v1.14.3
on windows_amd64
PS C:\Windows\system32>
```

- ❖ **Step 2: Open VS code** and go to extensions then download supportive tools for properly run Terraform file,
 - ❖ Install Hashicope Terraform



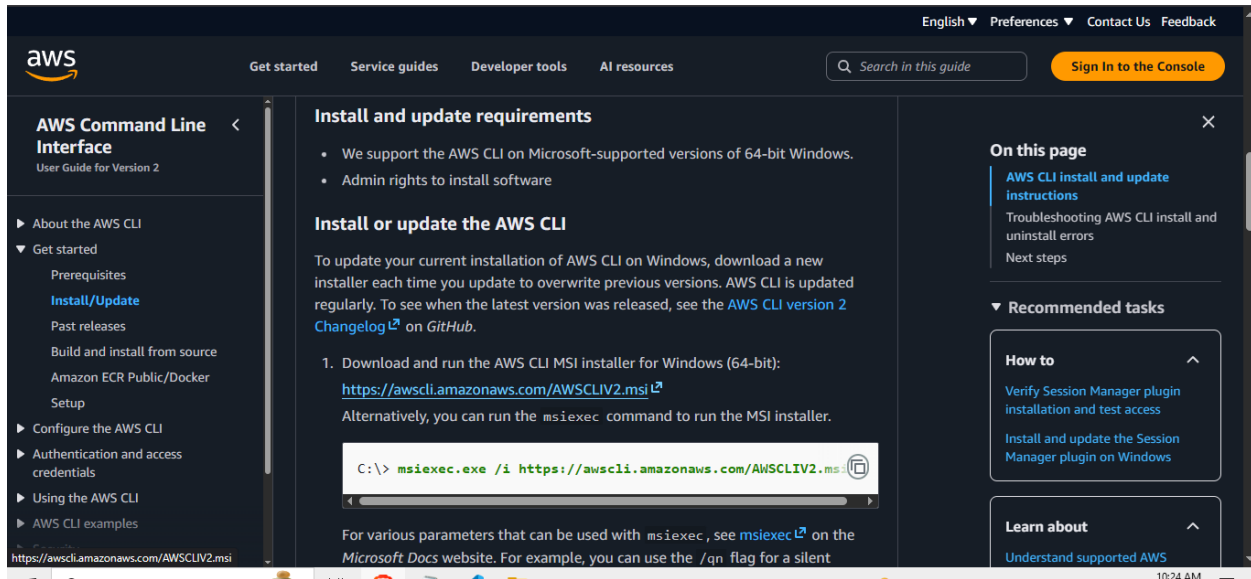
- ❖ Install Terraform Snippets,



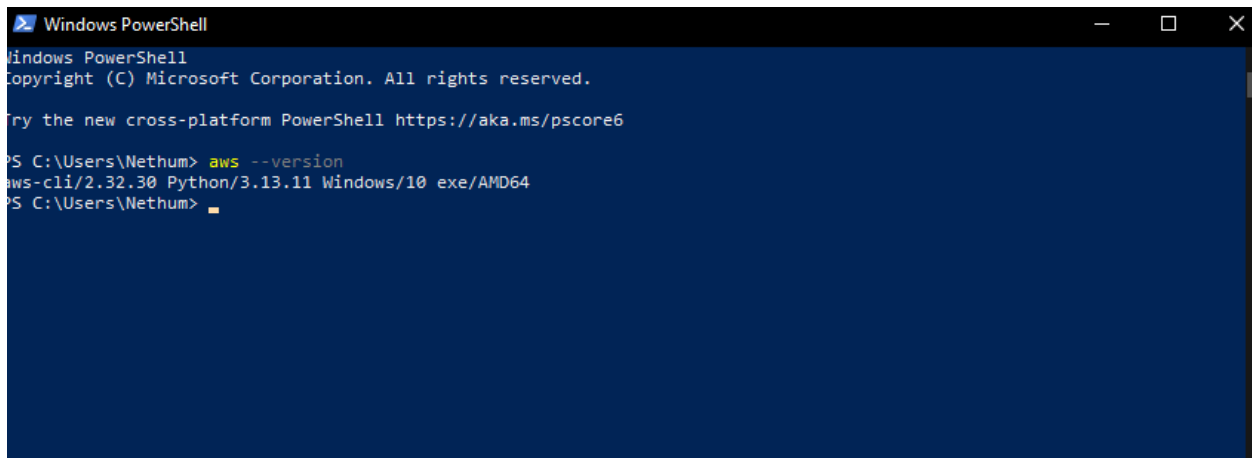
- ❖ **Step 3: Download AWS CLI** : Go to the official AWS website, download the latest version to your PC, and verify the download was successful.

<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

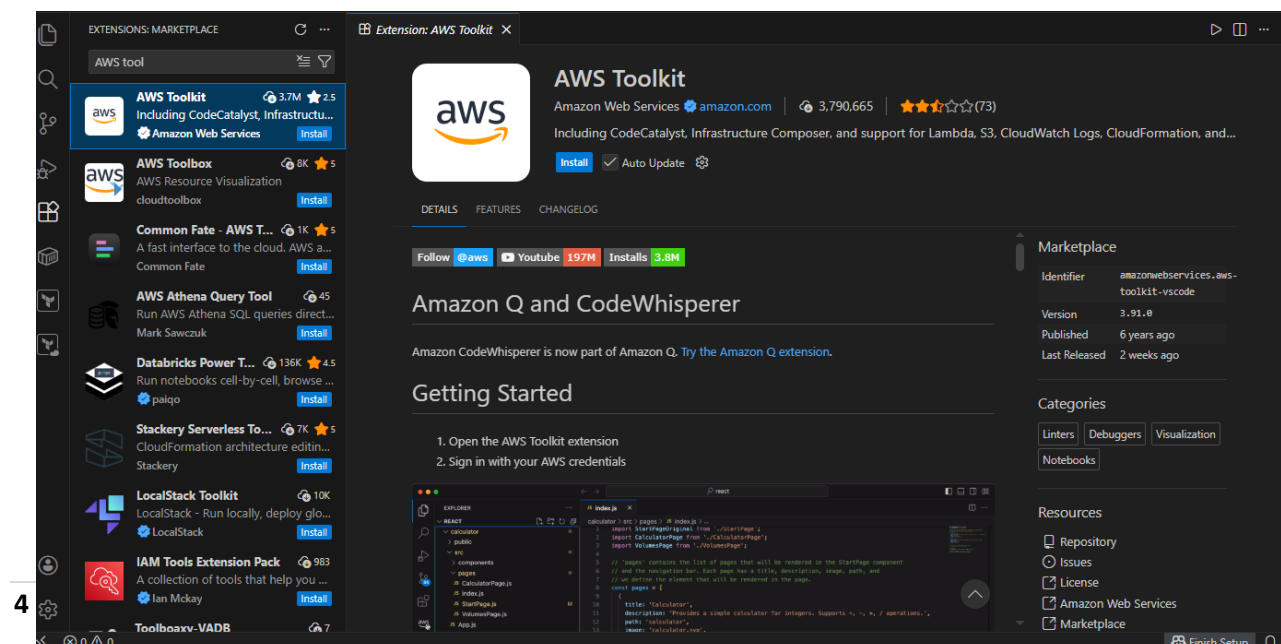
check whether AWS status :- `aws --version`



❖ AWS Status



❖ Open VS code and Install AWS tool kit to configure AWS service.



- ❖ **Step 4 Create an IAM user.** Under Permissions policies, attach the policies AmazonS3FullAccess and AmazonEC2FullAccess. Then, go to Security credentials, select Create access key, and generate an access key ID and a secret access key.

❖ IAM User Creation

The screenshot shows the 'Specify user details' step in the AWS IAM console. The breadcrumb trail is IAM > Users > Create user. On the left, a progress bar shows Step 1 (Specify user details) as the active step, followed by Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Review and create). The main content area is titled 'Specify user details' and contains a 'User details' section. The 'User name' field is populated with 'Terraform-user'. Below this, there is a checkbox for 'Provide user access to the AWS Management Console - optional'. A blue callout box contains a note: 'If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)'. At the bottom right, there are 'Cancel' and 'Next' buttons.

Step 1
● Specify user details
○ Step 2
○ Set permissions
○ Step 3
○ Review and create

Specify user details

User details

User name

Terraform-user

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and +, =, ., @, _ - (hyphen)

☐ Provide user access to the AWS Management Console - *optional*
In addition to console access, users with `SignInLocalDevelopmentAccess` permissions can use the same console credentials for programmatic access without the need for access keys.

① If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel Next

- ❖ Add permission and policy in to IAM user,

The screenshot shows the 'Review' step in the AWS IAM console. The breadcrumb trail is IAM > Users > Terraform-user > Add permissions. On the left, a progress bar shows Step 1 (Add permissions) and Step 2 (Review) as the active step. The main content area is titled 'Review' and contains a 'User details' section with 'User name' as 'Terraform-user'. Below this is a 'Permissions summary (2)' section. It contains a table with two rows of permissions. At the bottom right, there are 'Cancel', 'Previous', and 'Add permissions' buttons.

Step 1
● Add permissions
○ Step 2
● Review

Review

The following policies will be attached to this user. [Learn more](#)

User details

User name

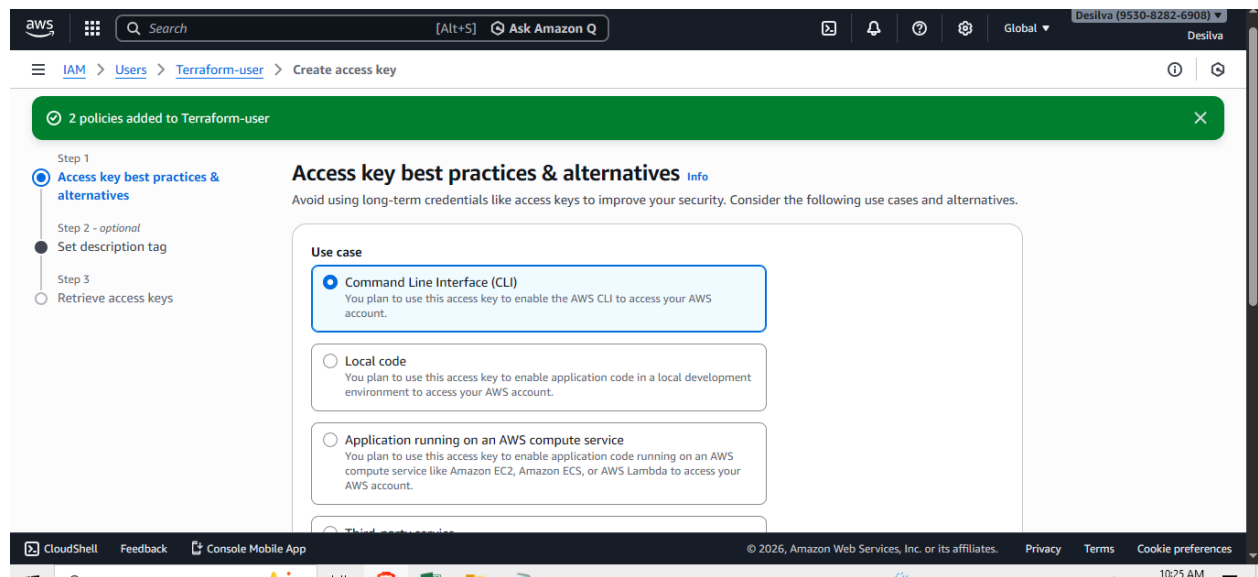
Terraform-user

Permissions summary (2)

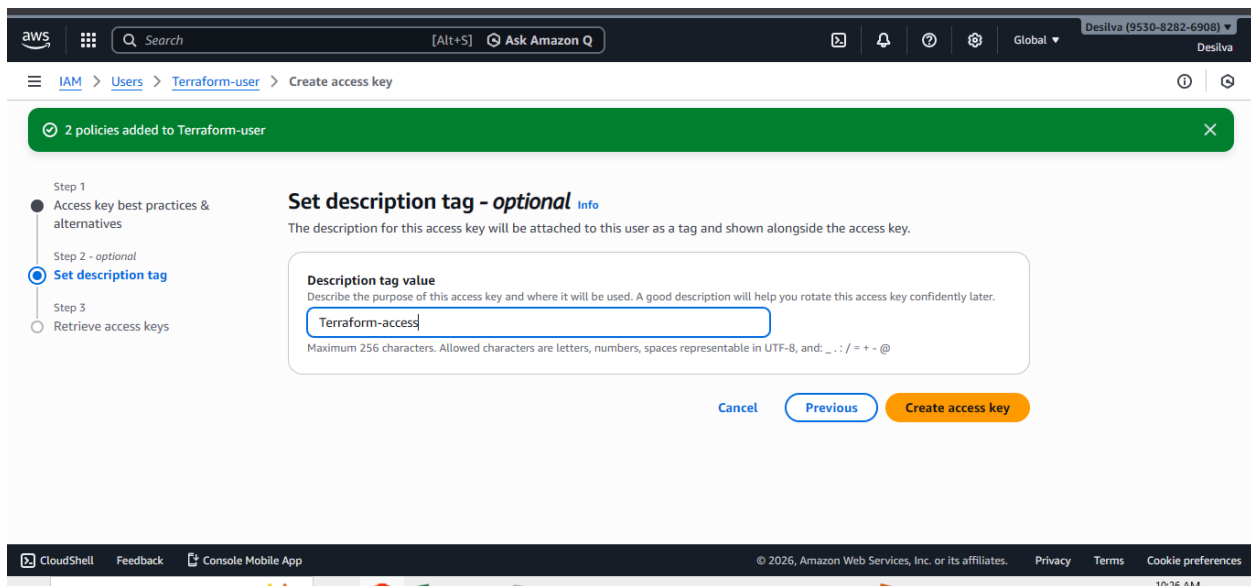
Name	Type	Used as
AmazonS3FullAccess	AWS managed	Permissions policy
AmazonEC2FullAccess	AWS managed	Permissions policy

Cancel Previous Add permissions

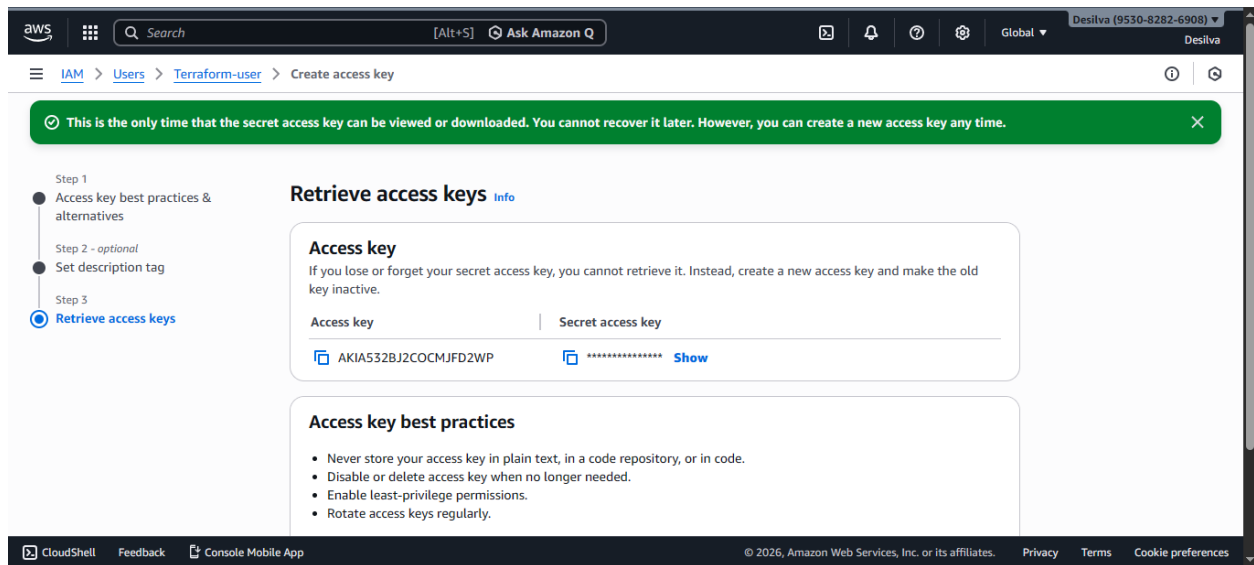
❖ Access Key Creation,



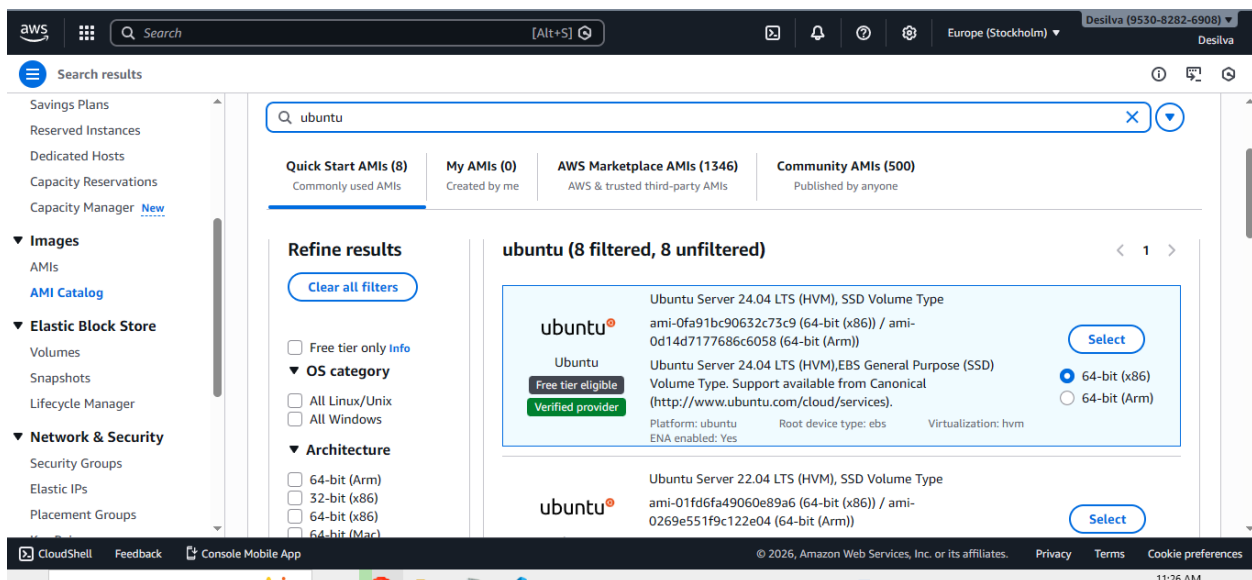
❖ Provide tag name for the Access Key creation,



❖ Generated the Access key and secret key,



❖ **Step 5: Create a Terraform file in VS Code to define an EC2 instance.** Provide the required AMI ID from the AWS Management Console's AMI Catalog.



- ❖ **Step 6: Configure AWS CLI:** Open your terminal and configure the AWS CLI by providing your IAM user's access key, secret key, region, and output format. Then, create separate Terraform files for an S3 bucket and an EC2 instance,

```

8  }
9
10 provider "aws" {
11   region = "us-east-1"
12 }
13

```

```

Do you want to run software from this untrusted publisher?
File C:\Users\Wethum\AppData\Local\Programs\Microsoft VS
Code\resources\app\out\vs\workbench\contrib\terminal\common\scripts\shellIntegration.ps1 is
published by CN=Microsoft Corporation, O=Microsoft Corporation, L=Redmond, S=Washington, C=US
and is not trusted on your system. Only run scripts from trusted publishers.
[V] Never run [D] Do not run [R] Run once [A] Always run [?] Help (default is "D"):
PS C:\Users\Wethum\Desktop\Terraform 2026> aws configure
AWS Access Key ID [None]: AKIA532B32C0CH0FD2NP
AWS Secret Access Key [None]: 1lf31ybnJv6ENC2TphWV2Ah120uFvkjggQx9G4n1
Default region name [None]: us-east-1
Default output format [None]: json
PS C:\Users\Wethum\Desktop\Terraform 2026>

```

- After Configure AWS CLI and create Terraform files, Finally Execute the terraform Commands,
- Command : terraform init

```

1 resource "aws_s3_bucket" "example_bucket" {
2   bucket = "my-unique-bucket-name-${random_id.bucket_suffix.hex}"
3   tags = {
4     Name       = "Example Bucket"
5     Environment = "Dev"
6   }
7 }
8
9 resource "aws_s3_bucket_versioning" "example_versioning" {
10  bucket = aws_s3_bucket.example_bucket.id
11  versioning_configuration {
12    status = "Enabled"
13  }
14 }

```

```

PS C:\Users\Wethum\Desktop\Terraform 2026> ls

Directory: C:\Users\Wethum\Desktop\Terraform 2026

Mode                LastWriteTime         Length Name
----                -
-a----             1/8/2026 11:28 AM             187 ec2.tf
-a----             1/8/2026 11:28 AM             172 provider.tf

PS C:\Users\Wethum\Desktop\Terraform 2026> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/random...

```


- Command :- terraform plan

```

rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
PS C:\Users\Nethum\Desktop\Terraform 2026> 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.ubuntu_server will be created
+ resource "aws_instance" "ubuntu_server" {
  + ami                    = "ami-053b0d53c279acc90"
  + 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)

```

- Command : terraform apply

The screenshot shows a VS Code editor with a file explorer on the left containing Terraform files: `.terraform`, `.terraform.lock.hcl`, `ec2.tf`, `s3.tf`, `terraform.tfstate`, and `terraform.tfstate.backup`. The main editor displays the `ec2.tf` file with the following content:

```

1 provider "aws" {
2   region = "us-east-1"
3 }
4
5 resource "aws_instance" "ubuntu_server" {
6   ami           = "ami-0885b1f6bd170450c"
7   instance_type = "t3.micro" # Changed from t2.micro to t3.micro
8
9   tags = {
10    Name = "ubuntu-free-tier-instance"
11  }
12 }
13

```

The terminal at the bottom shows the output of the `terraform apply` command:

```

Enter a value: yes
random_id.bucket_suffix: Creating...
random_id.bucket_suffix: Creation complete after 0s [id-b0x2Gg]
aws_s3_bucket.example_bucket: Creating...
aws_instance.ubuntu_server: Creating...
aws_instance.ubuntu_server: Still creating... [00m10s elapsed]
aws_s3_bucket.example_bucket: Still creating... [00m10s elapsed]
aws_s3_bucket.example_bucket: Creation complete after 11s [id-my-unique-bucket-name-6c3c591a]
aws_s3_bucket_versioning.example_versioning: Creating...
aws_s3_bucket_versioning.example_versioning: Creation complete after 2s [id-my-unique-bucket-name-6c3c591a]
aws_instance.ubuntu_server: Creation complete after 17s [id-i-0c011952bce448f84]

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.

Outputs:
my_Practice_terraform = "my-unique-bucket-name-6c3c591a"
PS C:\Users\Nethum\Desktop\Terraform 2026>

```

- After executing the Terraform commands, go to the AWS Management Console to verify whether the EC2 instance and S3 bucket were created successfully.

- EC2 Instance Status,

The screenshot shows the AWS Management Console for the 'Instances' page. The left sidebar contains navigation links for EC2, including Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, and Images. The main content area shows a list of instances with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability. The instance 'ubuntu-free-tier-instance' (i-0c011952bce448f84) is highlighted. Below the list, the details for this instance are shown, including its public and private IPv4 addresses. The console also displays the 'Instance summary' section with tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags.

- S3 bucket Status

The screenshot shows the AWS Management Console for the 'Amazon S3' page. The left sidebar contains navigation links for Amazon S3, including Buckets, Access management and security, and Storage management and insights. The main content area shows a list of buckets with columns for Name, AWS Region, and Creation date. The bucket 'my-unique-bucket-name-6c3c591a' is highlighted. Below the list, the details for this bucket are shown, including its creation date. The console also displays the 'Account snapshot' and 'External access summary' sections.

Terraform File Structure

Ec2.tf

```
provider "aws" {
  region = "us-east-1"
}

resource "aws_instance" "ubuntu_server" {
  ami          = "ami-0885b1f6bd170450c"
  instance_type = "t3.micro" # Changed from t2.micro to t3.micro

  tags = {
    Name = "ubuntu-free-tier-instance"
  }
}
```

S3.tf

```
resource "aws_s3_bucket" "example_bucket" {
  bucket = "my-unique-bucket-name-${random_id.bucket_suffix.hex}"
  tags = {
    Name          = "Example Bucket"
    Environment = "Dev"
  }
}

resource "aws_s3_bucket_versioning" "example_versioning" {
  bucket = aws_s3_bucket.example_bucket.id
  versioning_configuration {
    status = "Enabled"
  }
}

resource "random_id" "bucket_suffix" {
  byte_length = 4
}

output "my_Practice_terraform" {
  value = aws_s3_bucket.example_bucket.bucket
}
```