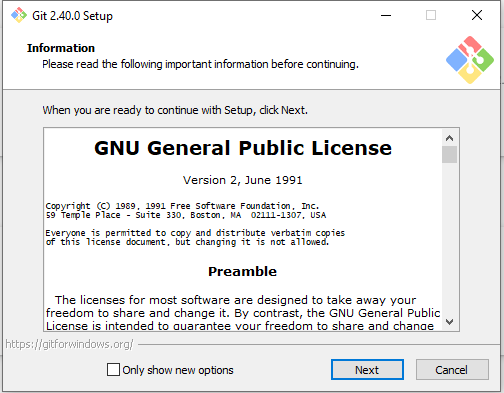
Download, Install and configure Git, Terraform, VScode, AWS CLI, Packer, AWS STS User and GitHub Actions secrets & variables and authentication terraform with AWS using AWS Access key.

- To Download, Install and configure Git:

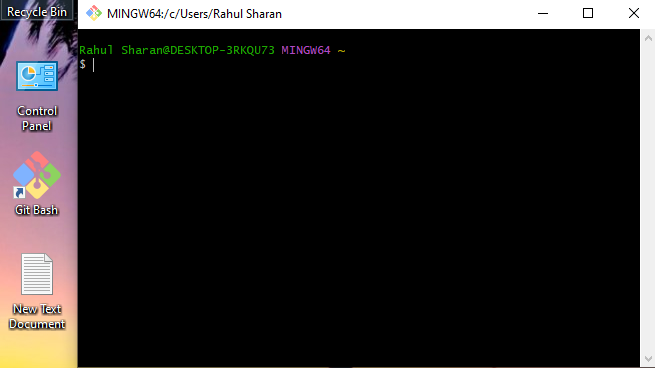
<https://git-scm.com/download/win>



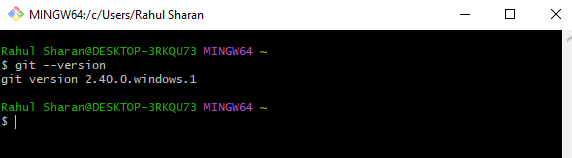
To install git on windows machine:



-Open the Git bash:



-To verify the git version:



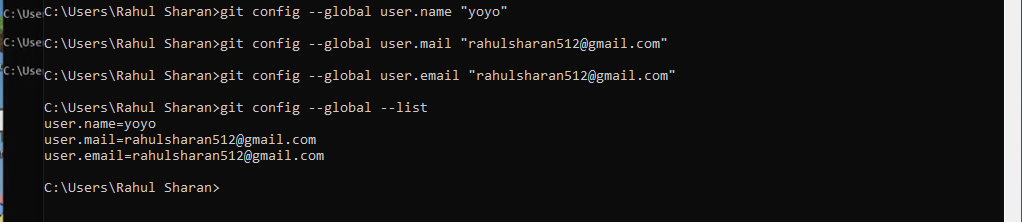


-To configure the Git:

> git config --global user.name "rahuls512"

> git config --global user.email [rahulsharan512@gmail.com](mailto:rahulsharan512@gmail.com)

> git config --global --list

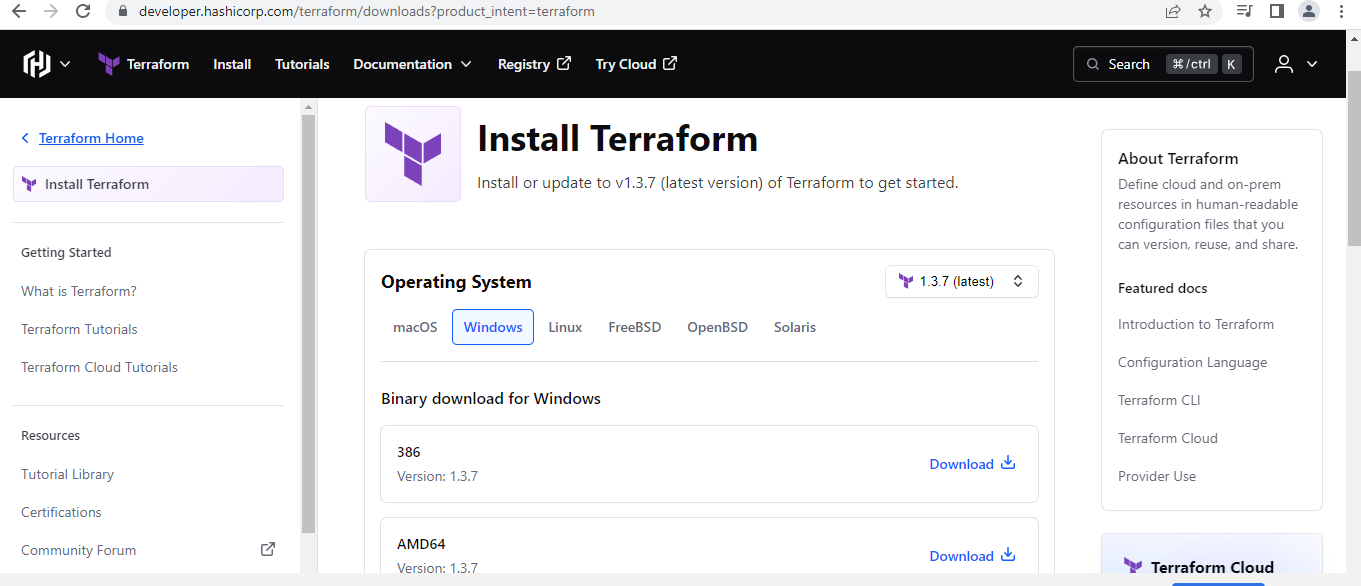


# **To Download, Install and configure terraform and Visual studio code.**

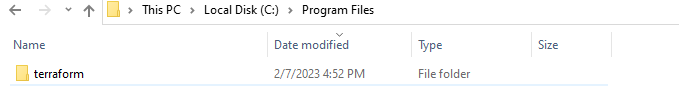
## -Download, Install and configure terraform:

To download terraform by using the following link:

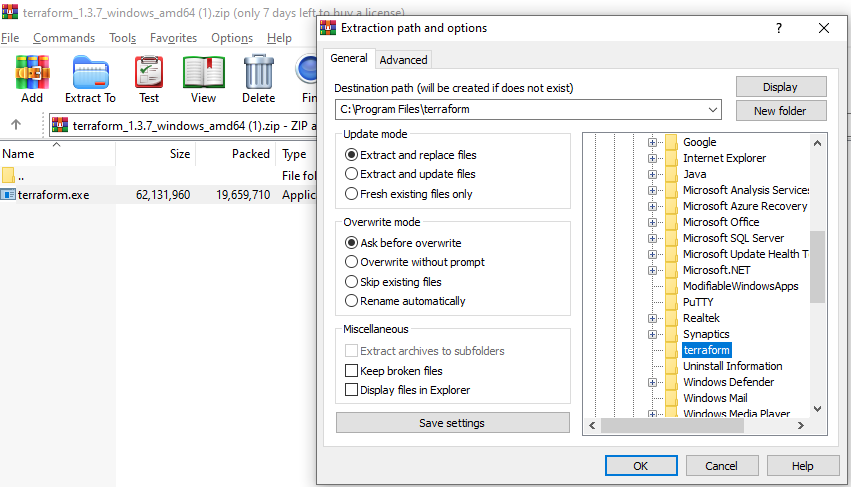
- <https://developer.hashicorp.com/terraform/downloads>



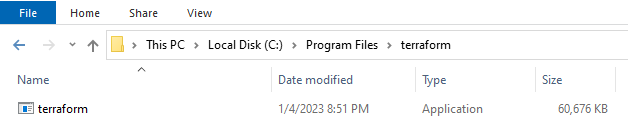
To create a folder terraform in c :> program files:



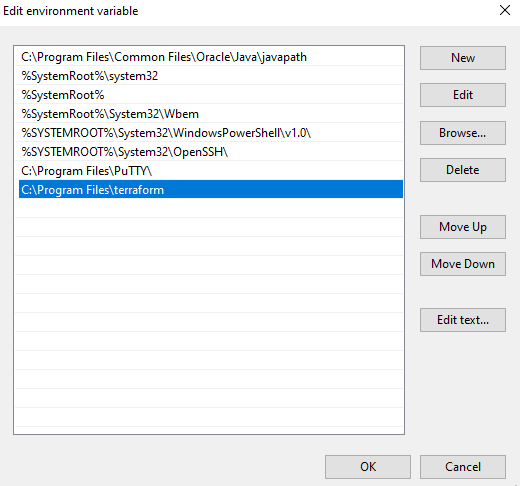
Terraform.exe file extract in c:>program files/terraform folder:



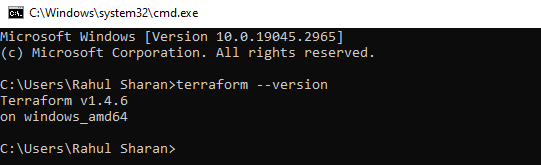
-To check the executable file in c :> program files>terraform:



## -Add the environment variable: Path- C:\Program Files\terraform



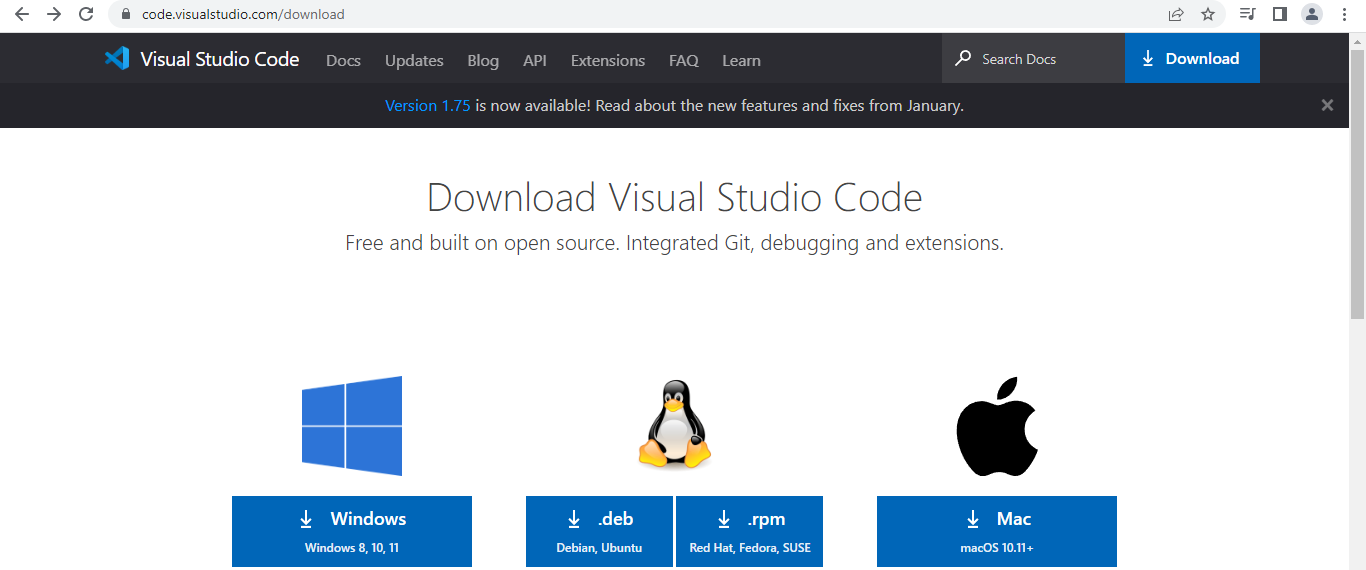
## -Open command prompt and check the terraform version by using the following the command:



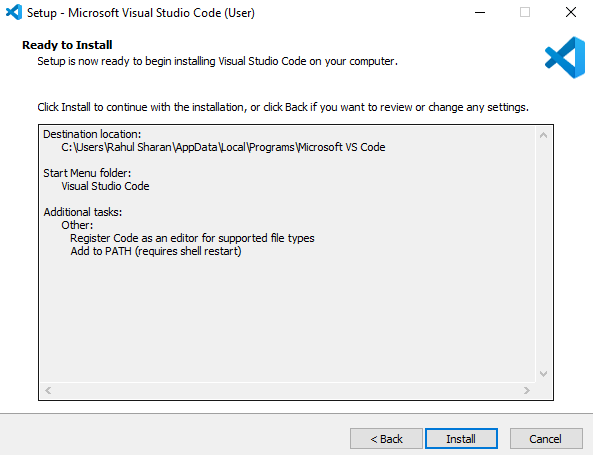
## -Download, Install and configure Visual studio code:

To download Visual studio code by using the following link:

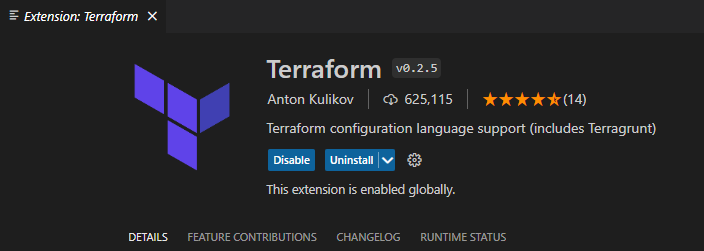
-https://code.visualstudio.com/download



To install Visual studio code:



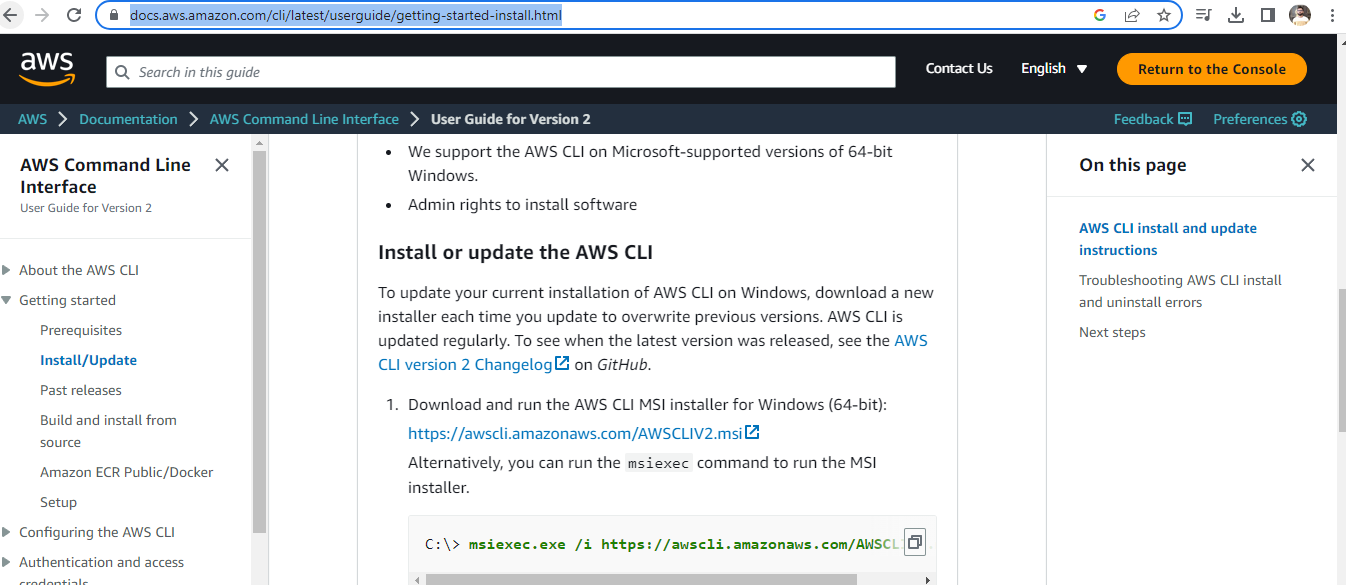
## -Add extensions in VScode -terraform and AWS terraform:

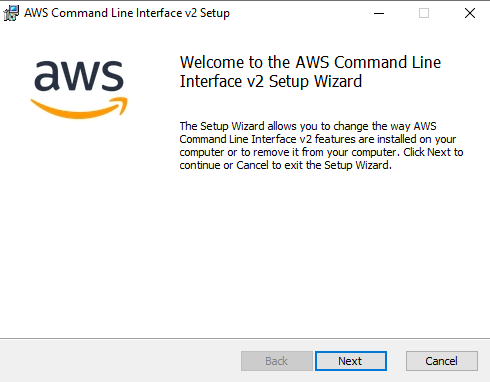


# **To Authentication terraform with AWS using AWS Access key:**

## -Download and install the AWS CLI for windows by using the following link:

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

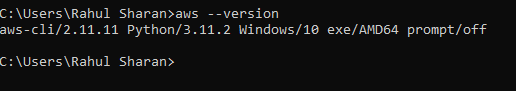




## - Check the Aws version on windows system:

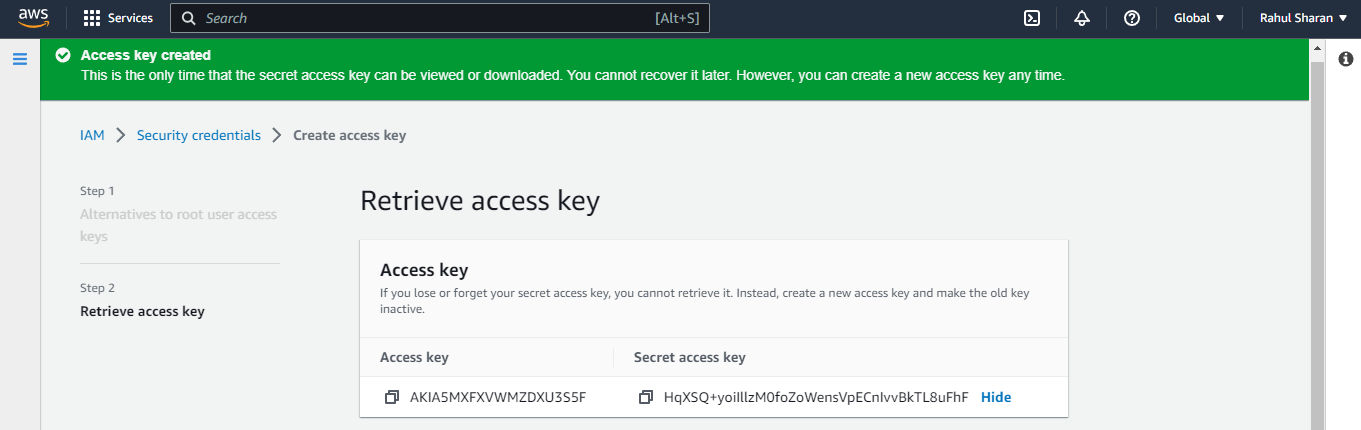
Open the command prompt and check the Aws version by using the following command:

>aws --version

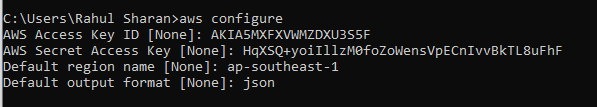


## -Create Access Key using IAM service:

IAM>Security Credentials>Access key: Create access key.

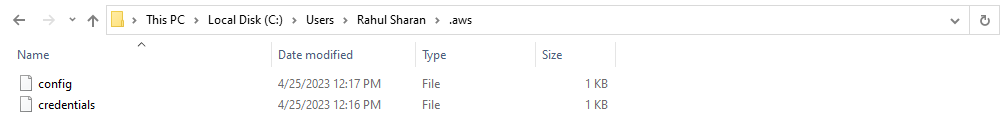


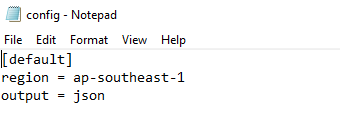
## -Configure the aws access key with terraform on windows:

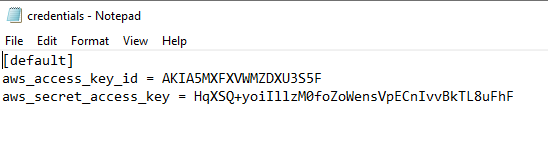


## -Check the created .aws folder and config and credentials files by using following path:

C:\Users\Rahul Sharan\.aws

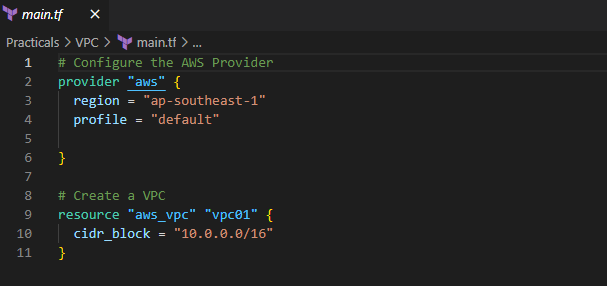






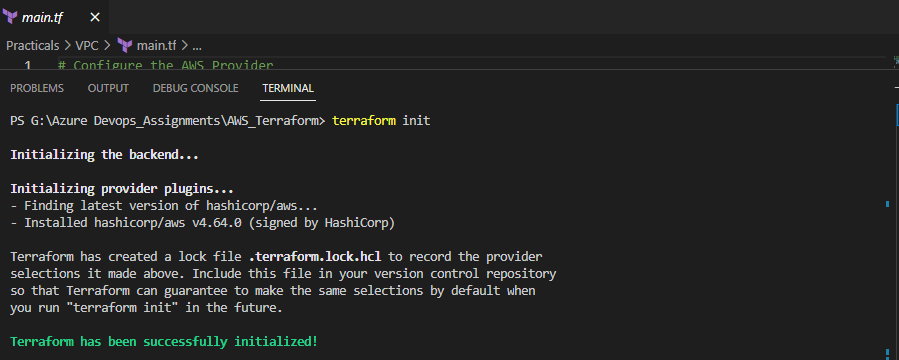
# **To create virtual private cloud by using the Aws Terraform:**

## -Configure the VPC file by using the HCL.



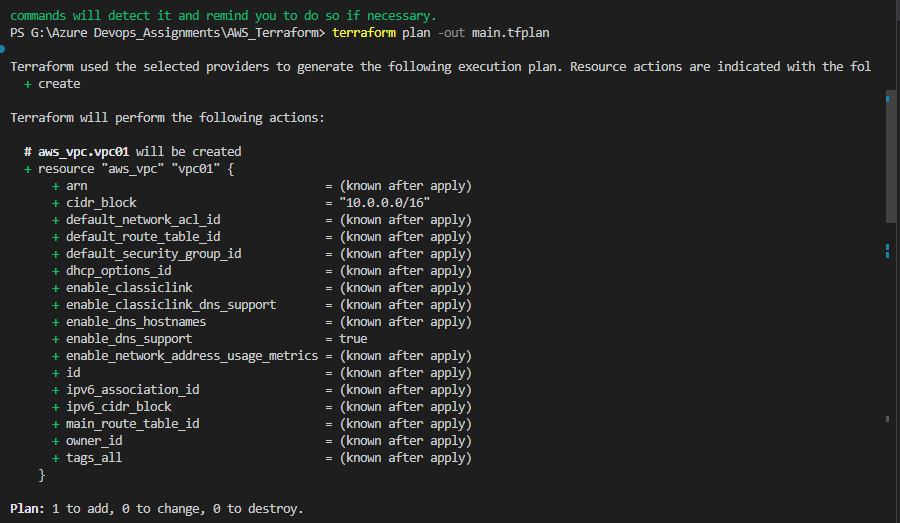
## -Initialize the terraform configuration file using the following command:

>terraform init



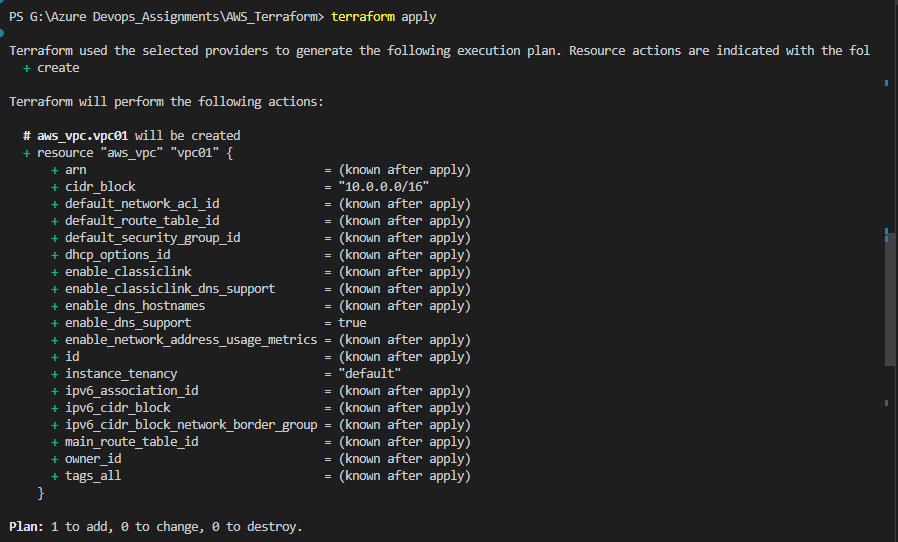
## -Create an execution plan for resource group by using the following command:

>terraform plan –out main.tfplan

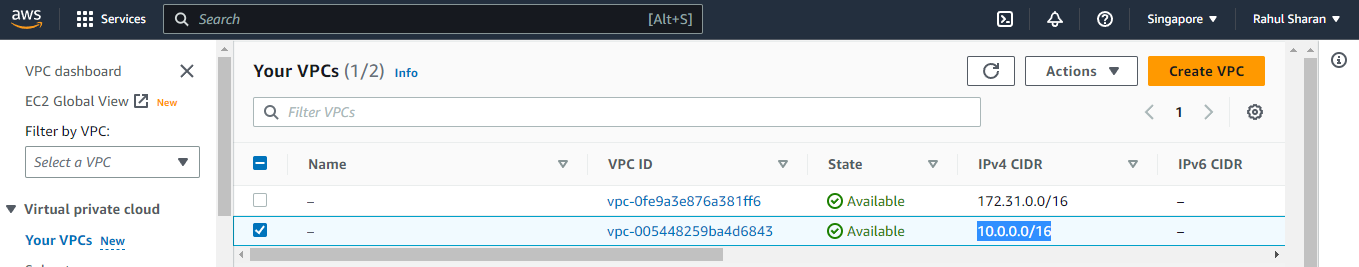


## -Execute the resource group by using the following command:

>terraform apply



## -Verify the VPC in AWS portal:

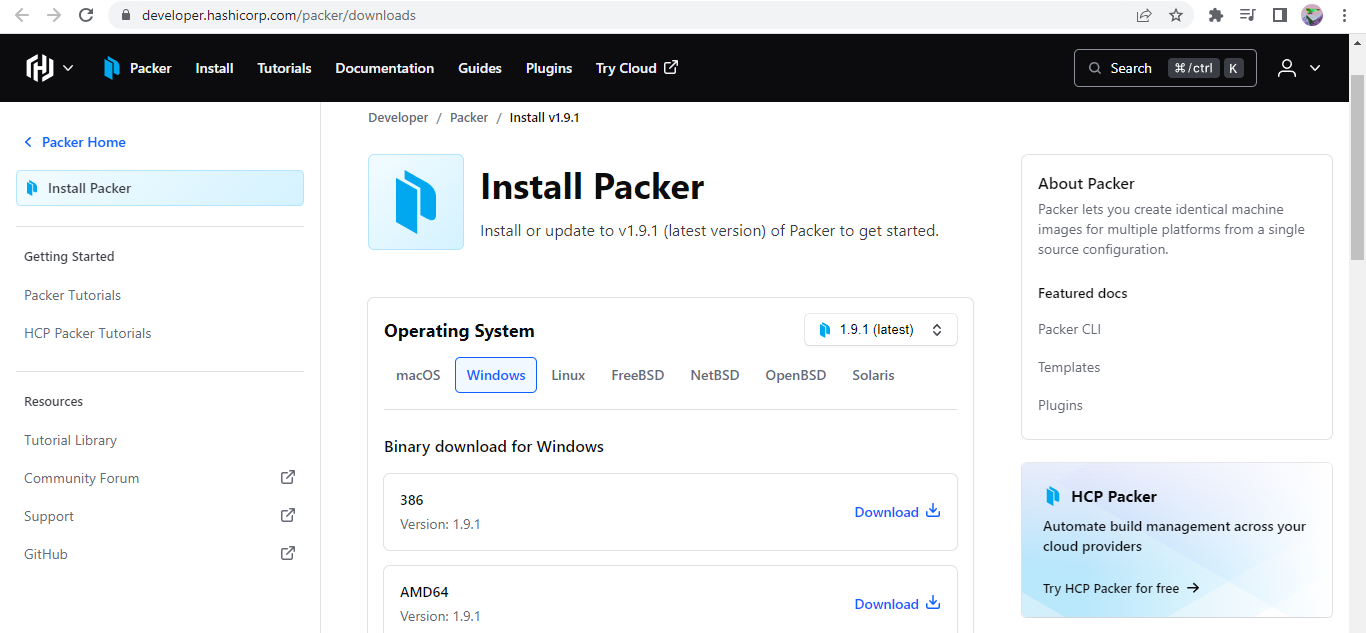


# **To Download, Install and configure Packer.**

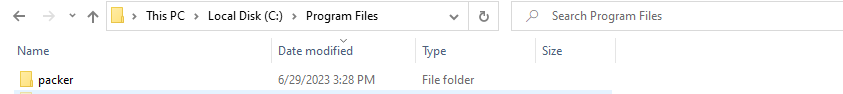
## -Download, Install and configure packer:

To download packer by using the following link:

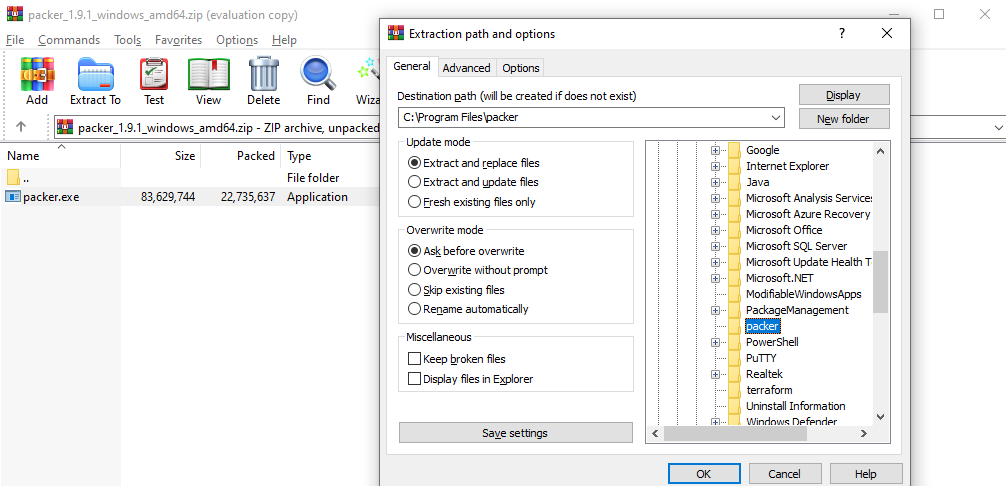
- (<https://www.packer.io/downloads>)



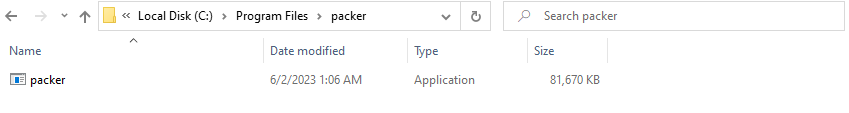
To create a folder packer in c :> program files:



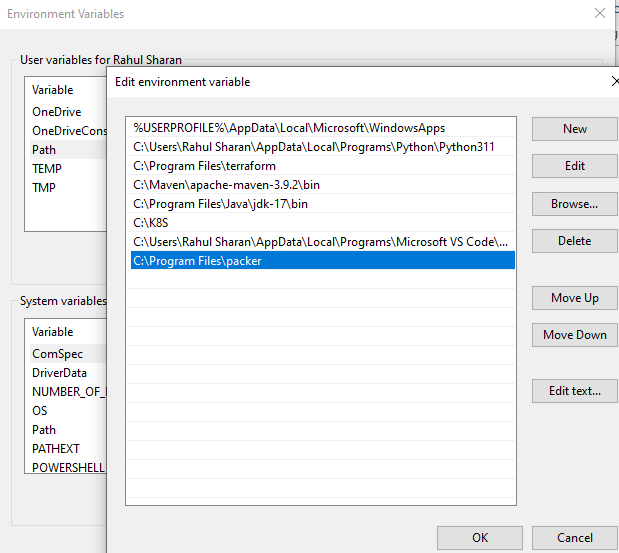
Packer.exe file extract in c :> program files/packer folder:



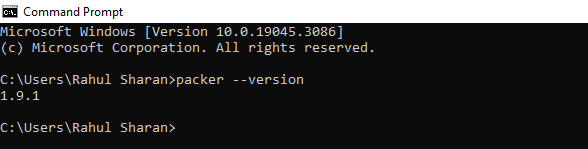
To check the executable file in c :> program files>packer:



## -Add the environment variable: Path- C:\Program Files\packer

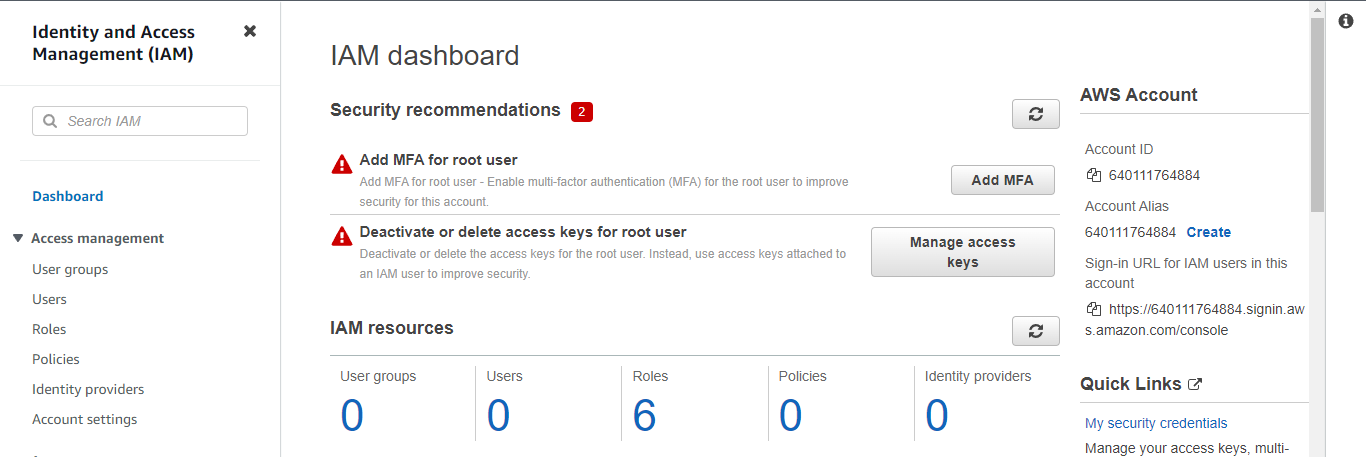


## -Open command prompt and check the packer version by using the following the command:



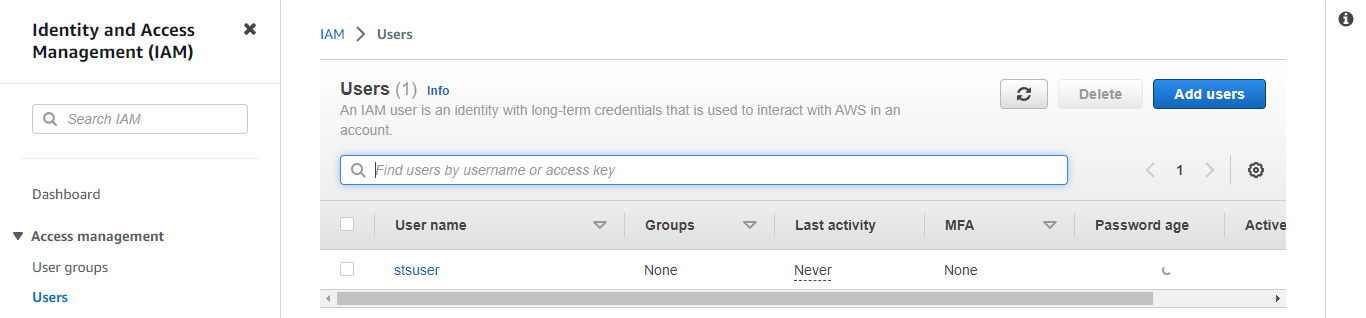
# **To configure the AWS Security Token Service (STS) using IAM (Identity and Access Management), you can follow these steps:**

### -Open the IAM console:



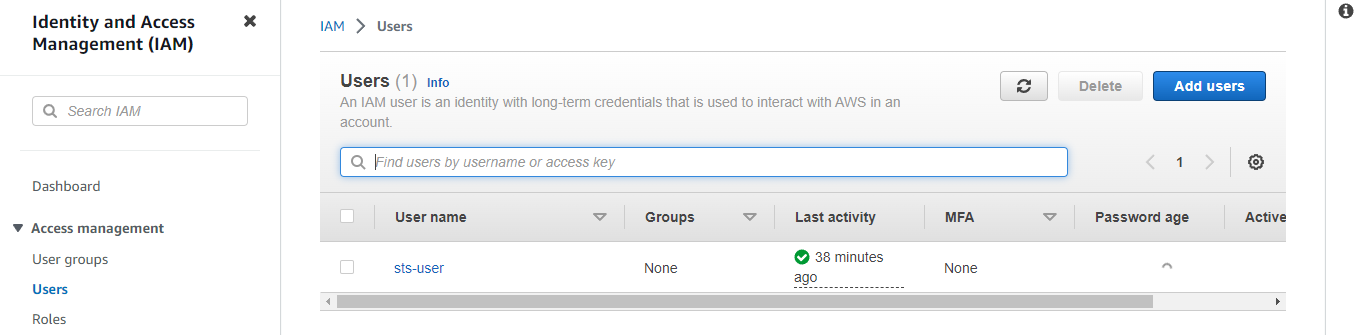
### -Create a user using IAM:

In the navigation pane, choose "user"->Set permission: Attach policies directly->next->create



### -Create access key:

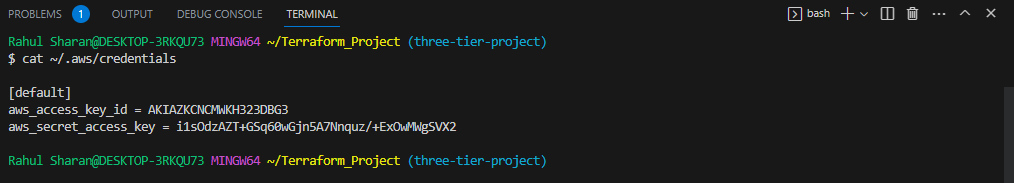
Click on “sts-user” to start creating an access key->Security credentials-> Access keys:Create Access key: local code->next->create access key->Done



### -Delete the Credentials:

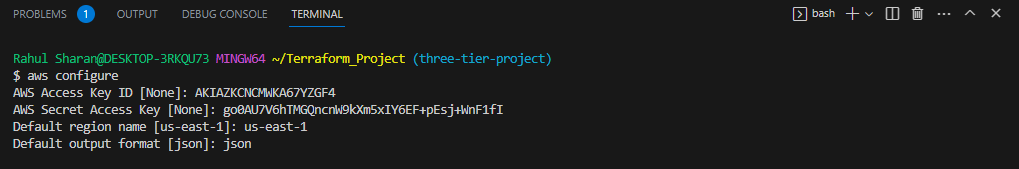
>cat ~/.aws/credentials

>rm –rf ~/.aws/credentials



### -Configure the access key with terraform using the following command:

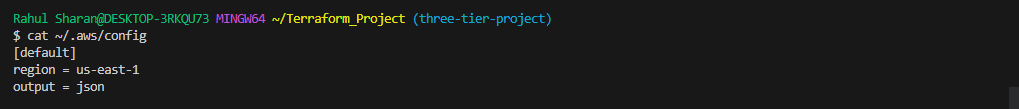
>aws configure

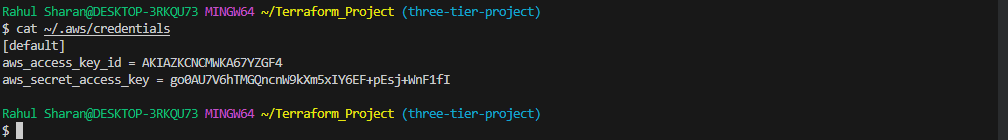


### -Check the .aws configuration file:

> cat ~/.aws/config and

>cat ~/.aws/credentials



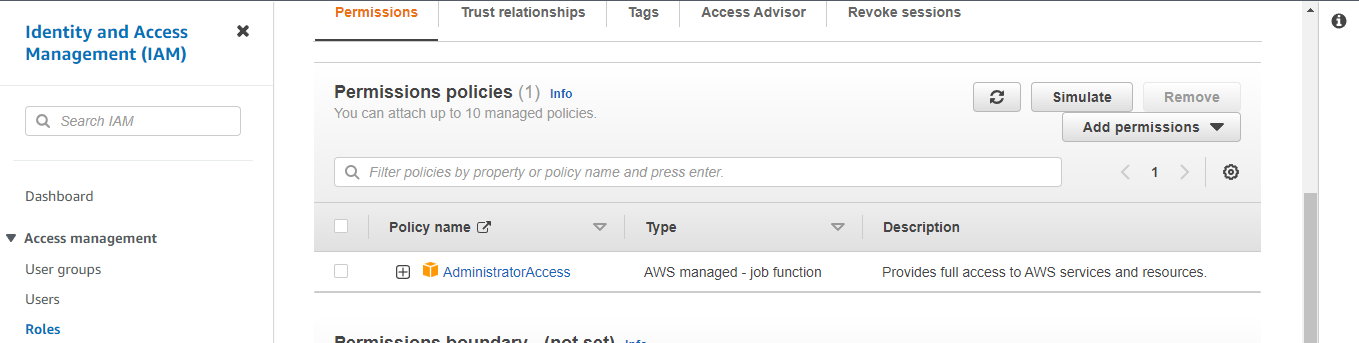


### -Create role:

Click on "Create role" to start creating a new role.

Trusted type->AWS account: use case->EC2->AWS Account: This account->Next->Add permission: Administrator access->next->Role name: “stsassume-role” ->Create role





### -Add permission to user:

Create the inline policy by Configuring using a role by using below link:

<https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-role.html>

IAM->User: sts-user->permission: add permission: Create inline policy: JSON:->Review policy and Create: name it as "stsassumerole-policy"

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

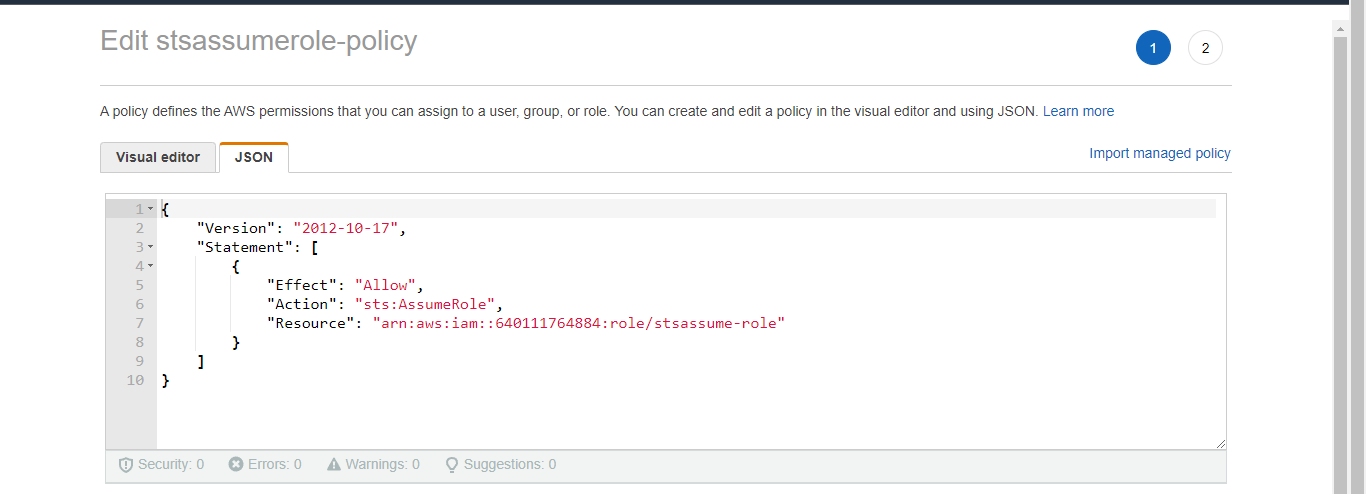
"Action": "sts:AssumeRole",

"Resource": "arn:aws:iam::640111764884:role/stsassume-role"

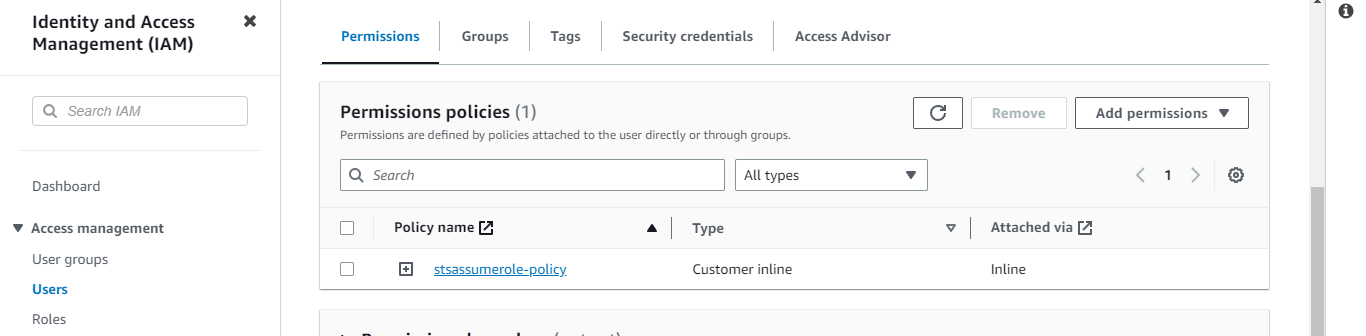
}

]

}



### -Successfully added the permission:



### -Configure the profile sts in configuration file:

>code ~/.aws/config

[profile sts]

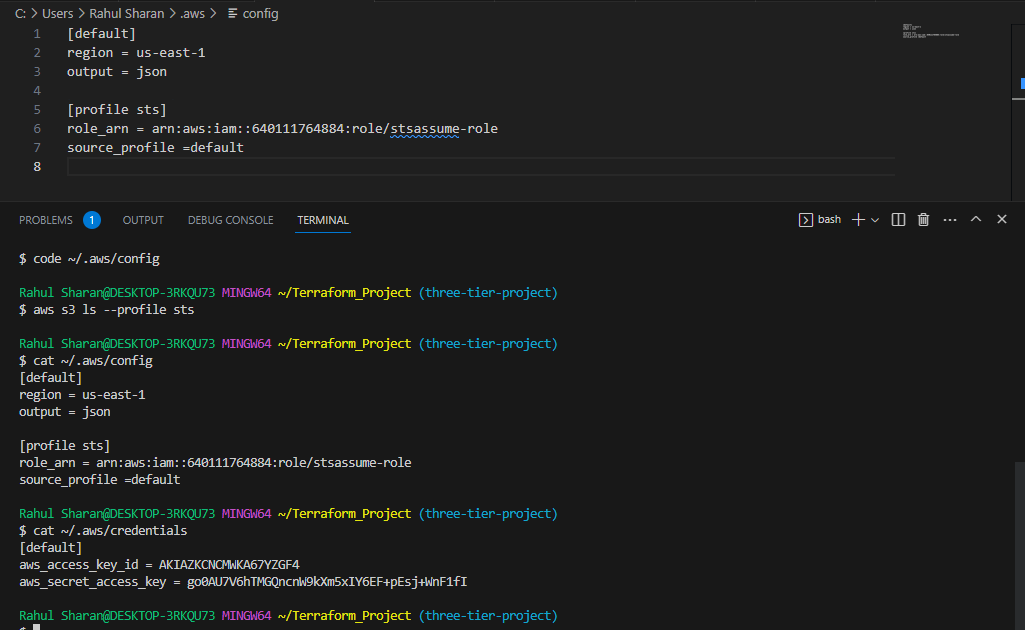
role\_arn = arn:aws:iam::640111764884:role/stsassume-role

source\_profile =default

>cat ~/.aws/config

>aws s3 ls --profile sts

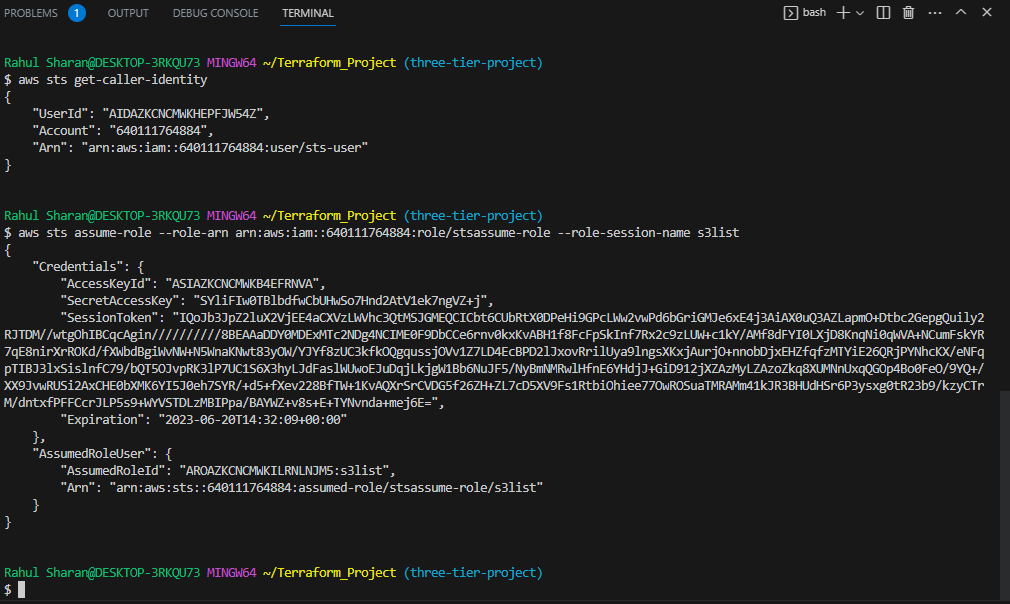
-> cat ~/.aws/credentials



### -Check the session token by using the following commands:

> aws sts get-caller-identity

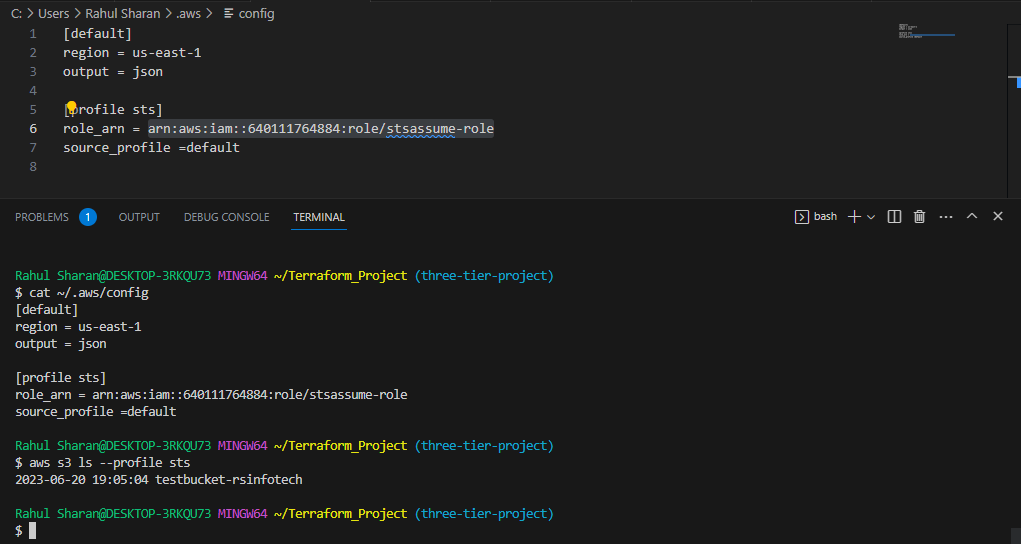
> aws sts assume-role --role-arn arn:aws:iam::640111764884:role/stsassume-role --role-session-name s3list



### -Check the S3 list using the following commands

> cat ~/.aws/config

> aws s3 ls --profile sts



### -Go to provider.tf and configure assume role:

provider "aws" {

  region = "us-east-1"

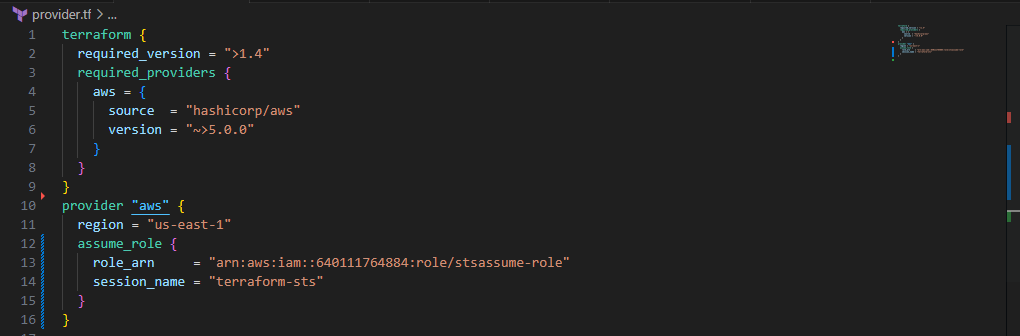
  assume\_role {

    role\_arn     = "arn:aws:iam::640111764884:role/stsassume-role"

    session\_name = "terraform-sts"

  }

}



## -**To Configure the Actions secrets and variables**

AWS\_ACCESS\_KEY and AWS\_SECRET\_KEY:

These are your AWS access credentials used for programmatic access to AWS services. The AWS\_ACCESS\_KEY is similar to a username, and the AWS\_SECRET\_KEY is like a password.

SSH\_PRIVATE\_KEY:

The SSH private key is part of a pair of keys used for secure authentication when connecting to remote servers via SSH (Secure Shell).

