TECHNICAL HUB

VIRTUAL PRIVATE CLOUD USING TERRAFORM

SUBMITTED BY

TERRIFIC TERRAFORMS (77)

B.V.S.S.SUSHMA(TEAM LEADER)

T.KAVYA SRI

V.N.S.SRINITHA

Y.PUSHPA MOULIKA

UNDER THE ESTEEMED GUIDANCE OF

BOBBY SIR

SURYA ASHOK SIR

INTRODUCTION:

Terraform and VPC allow you to create and manage your VPC resources in a programmatic and automated way. This allows you to quickly and easily provision and manage VPC resources, while also reducing the risk of human error and enabling greater consistency across your infrastructure.

A Virtual Private Cloud (VPC) is a virtual network environment that provides a secure and isolated space for resources within the Amazon Web Services (AWS) cloud. VPCs can be customized to meet specific needs, and can be connected to on-premises data centers or other VPCs using VPN or Direct Connect.

Terraform is an open-source infrastructure as code (IaC) tool that allows you to define and manage your infrastructure as code. Terraform enables you to create, modify, and delete infrastructure resources in a safe and repeatable way, ensuring consistency and reducing the potential for human error.

Subnet: A subnet, or subnetwork, is a segmented piece of a larger network. More specifically, subnets are a logical partition of an IP network into multiple, smaller network segments. The Internet Protocol (IP) is the method for sending data from one computer to another over the internet. Each computer, or host, on the internet has at least one IP address as a unique identifier**.**

Routetable: A route table contains a set of rules, called routes, that determine where network traffic from your subnet or gateway is directed**.**

Internet Gateway: An Internet gateway is a network "node" that connects two different networks that use different protocols (rules) for communicating. In the most basic terms, an Internet gateway is where data stops on its way to or from other networks.

NAT Gateway: A NAT gateway is a Network Address Translation (NAT) service. You can use a NAT gateway so that instances in a private subnet can connect to services outside your VPC but external services cannot initiate a connection with those instances**.**

In essence, using Terraform to create and manage a VPC enables you to define your infrastructure as code, enabling you to rapidly and consistently create and manage your cloud infrastructure.

SCOPE OF THE PROJECT

The scope of a Virtual Private Cloud (VPC) using Terraform project is to create a secure and isolated network environment in the cloud. The project includes configuring the VPC, subnets, route tables, internet gateways, security groups, NAT gateways, and VPN connections using infrastructure as code. The goal is to provide a reliable, scalable, and customizable infrastructure that can be easily managed and updated. Overall, the project scope is to create a flexible and secure cloud-based networking solution that can meet the requirements of various applications and workloads.

PURPOSE OF THE PROJECT

The purpose of a Virtual Private Cloud (VPC) using Terraform project is to create a secure, flexible, and cost-effective networking infrastructure in the cloud for running applications and workloads. The project achieves this by defining and managing the VPC resources as code, which enables easy and efficient deployment, management, and updating of the infrastructure.

TOOLS USED

There are several tools used in a Virtual Private Cloud (VPC) using Terraform project to create and manage the networking infrastructure in the cloud. These tools include

1.Terraform: Terraform is an open-source infrastructure as code tool used to create, manage, and update the VPC and its resources.

2.Amazon Virtual Private Cloud (VPC): Amazon VPC is a service that allows you to create a virtual network environment in the cloud.

3.AWS CLI: It allows users to interact with AWS services and resources using commands in the terminal. This can be useful for tasks such as configuring AWS credentials, managing EC2 instances, and performing other AWS resource management tasks.

4.Visual Studio Code (VS Code): Visual studio code can be used for creating and managing a Virtual Private Cloud (VPC) using Terraform.

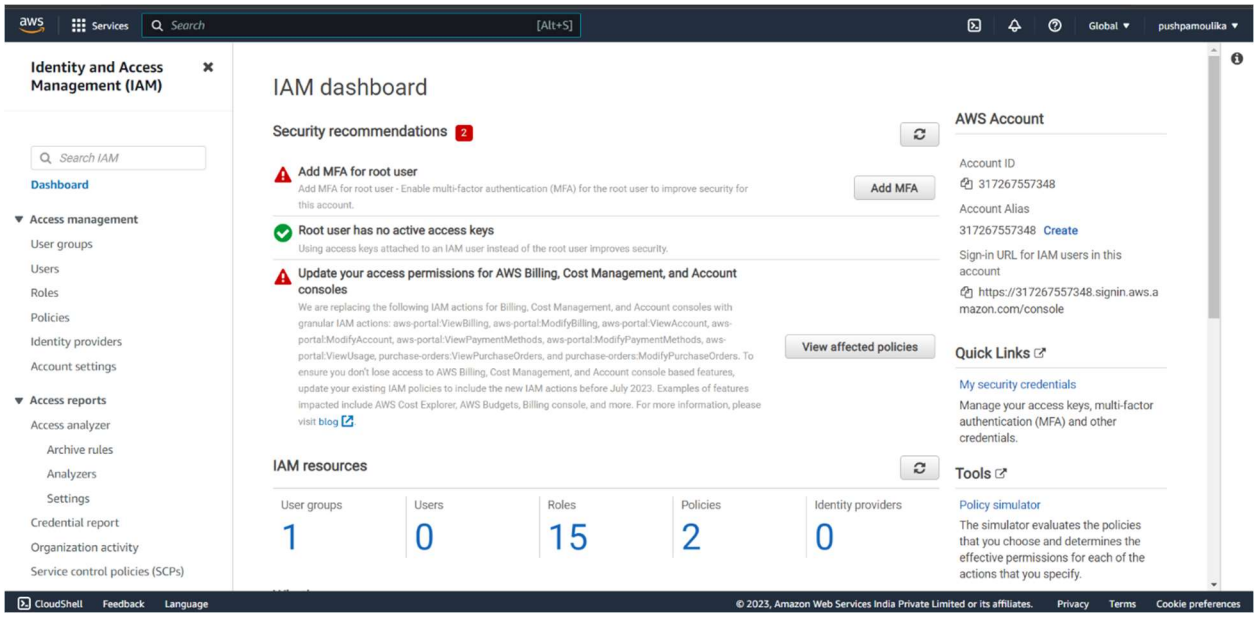
TECHNOLOGIES USED

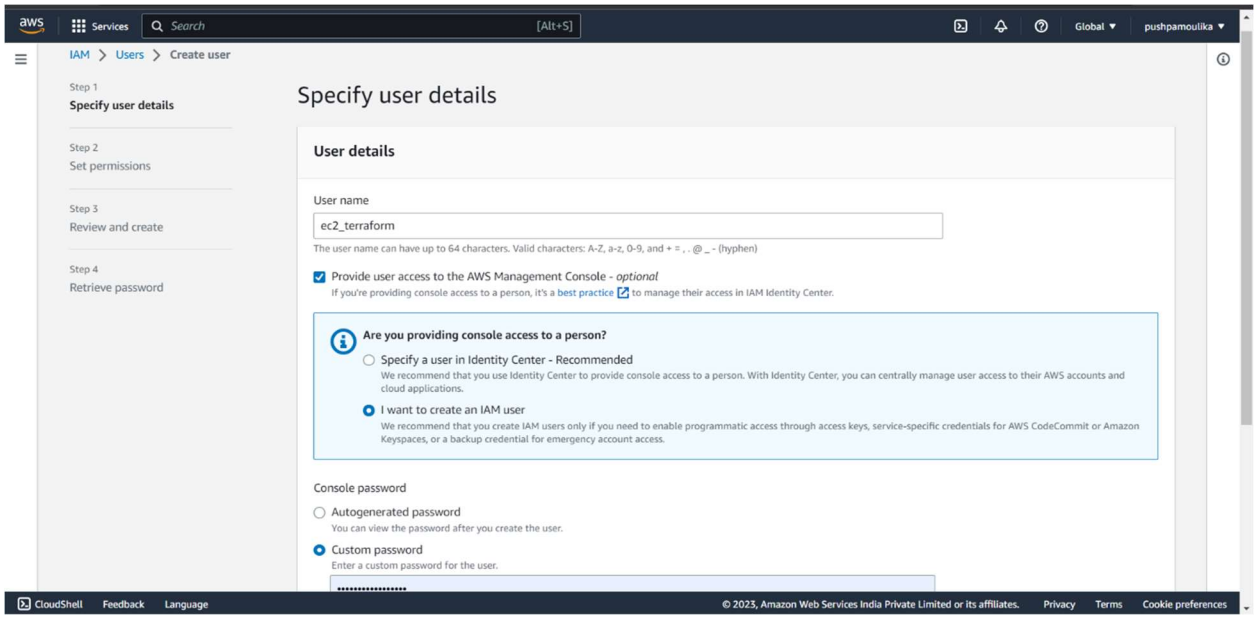
The technology used in a Virtual Private Cloud (VPC) using Terraform includes AWS, Amazon VPC and IAM. Terraform is used as an infrastructure as code (IaC) tool to define and manage the VPC and its resources. Other tools and technologies may also be used to automate the deployment and management of the VPC using Terraform.

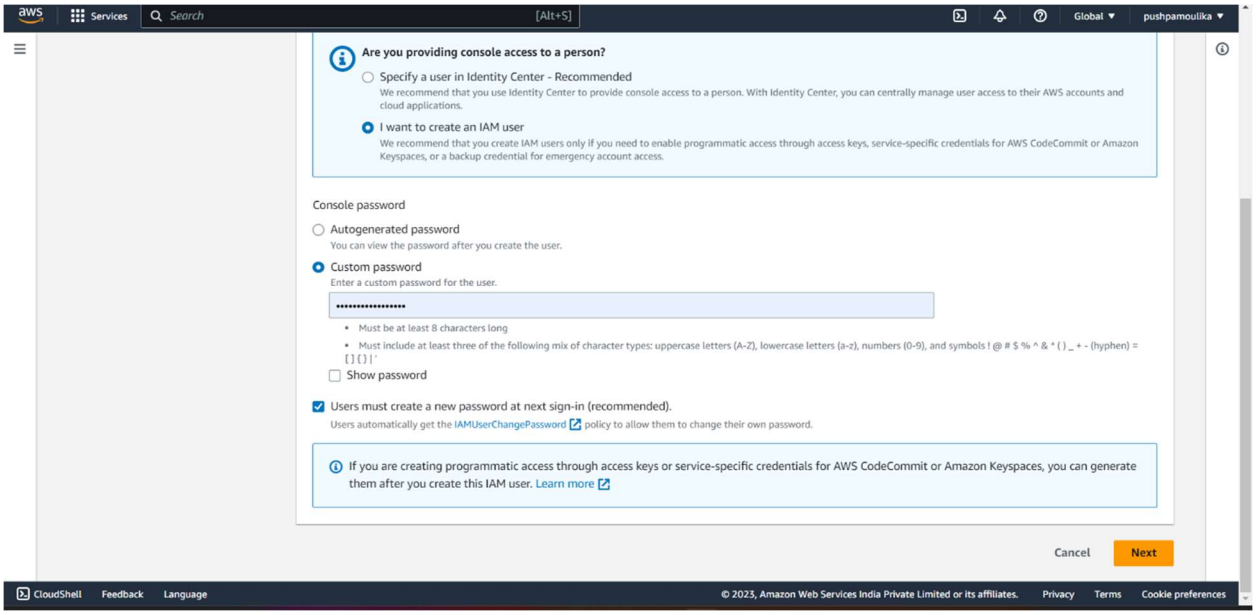
PROOF OF CONCEPT

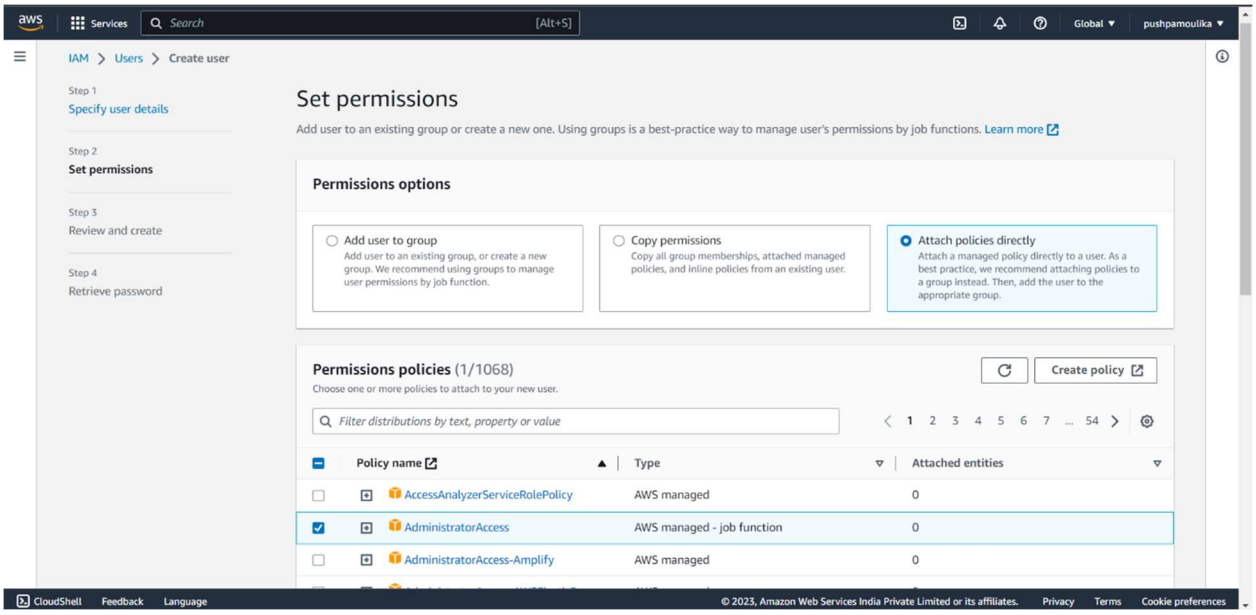
The proof of concept (POC) for a Virtual Private Cloud (VPC) using Terraform typically involves creating a simple VPC with a few resources to demonstrate the basic functionality and feasibility of using Terraform for infrastructure as code (IaC).

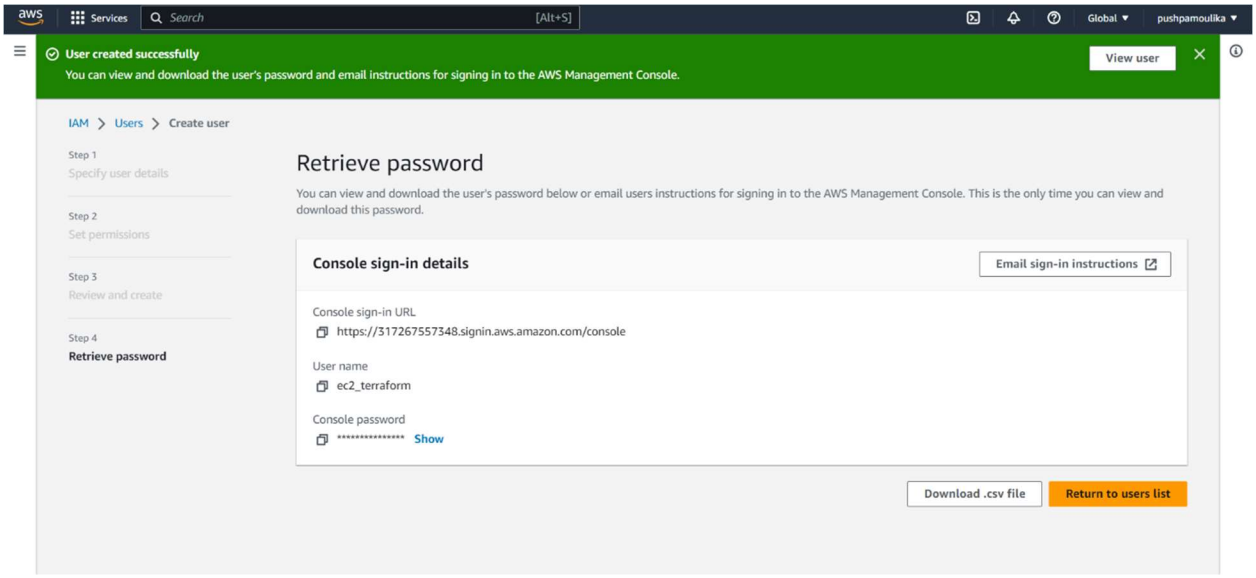
STEP1: first we need to install terraform and aws cli in our local system. Now open aws account and create IAM user with administrator access and go to security credentials and create access key and secret key and click on download csv file.

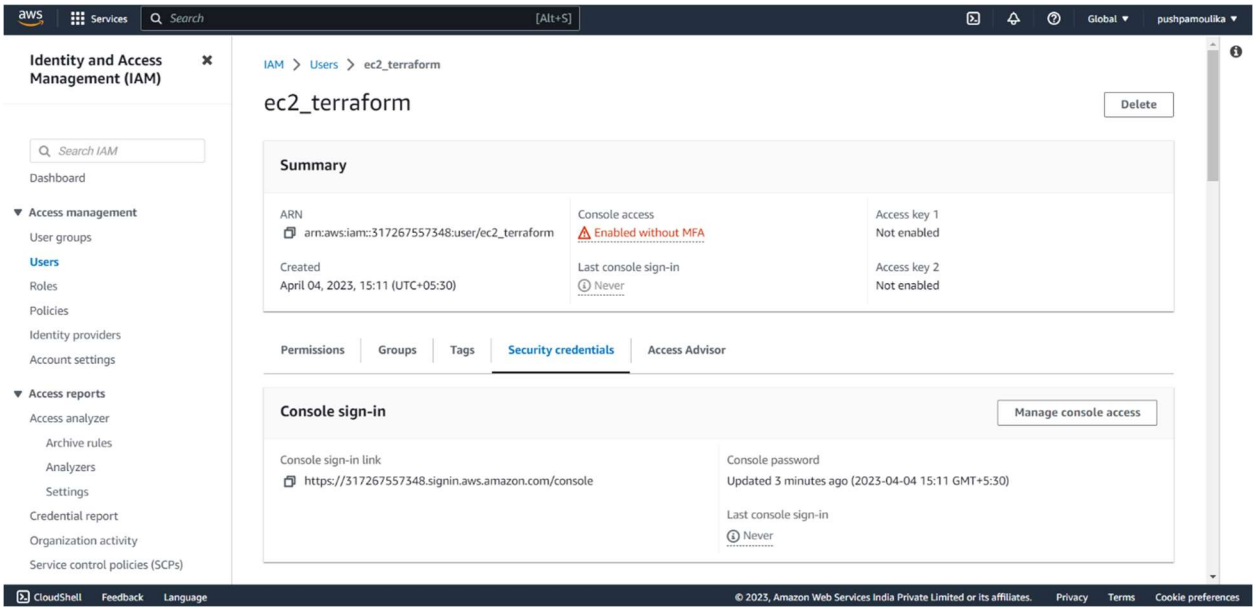

Services
Search
[Alt+S]
Global
pushpamoulika
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
Settings
Credential report
Organization activity
Service control policies (SCPs)
IAM dashboard
Security recommendations 2
Add MFA for root user
Add MFA for root user - Enable multi-factor authentication (MFA) for the root user to improve security for this account.
Add MFA
Root user has no active access keys
Using access keys attached to an IAM user instead of the root user improves security.
Update your access permissions for AWS Billing, Cost Management, and Account consoles
We are replacing the following IAM actions for Billing, Cost Management, and Account consoles with granular IAM actions: aws-portal:ViewBilling, aws-portal:ModifyBilling, aws-portal:ViewAccount, aws-portal:ModifyAccount, aws-portal:ViewPaymentMethods, aws-portal:ModifyPaymentMethods, aws-portal:ViewUsage, purchase-orders:ViewPurchaseOrders, and purchase-orders:ModifyPurchaseOrders. To ensure you don't lose access to AWS Billing, Cost Management, and Account console based features, update your existing IAM policies to include the new IAM actions before July 2023. Examples of features impacted include AWS Cost Explorer, AWS Budgets, Billing console, and more. For more information, please visit blog
View affected policies
IAM resources
User groups 1
Users 0
Roles 15
Policies 2
Identity providers 0
AWS Account
Account ID
317267557348
Account Alias
317267557348 Create
Sign-in URL for IAM users in this account
https://317267557348.signin.aws.amazon.com/console
Quick Links
My security credentials
Manage your access keys, multi-factor authentication (MFA) and other credentials.
Tools
Policy simulator
The simulator evaluates the policies that you choose and determines the effective permissions for each of the actions that you specify.
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

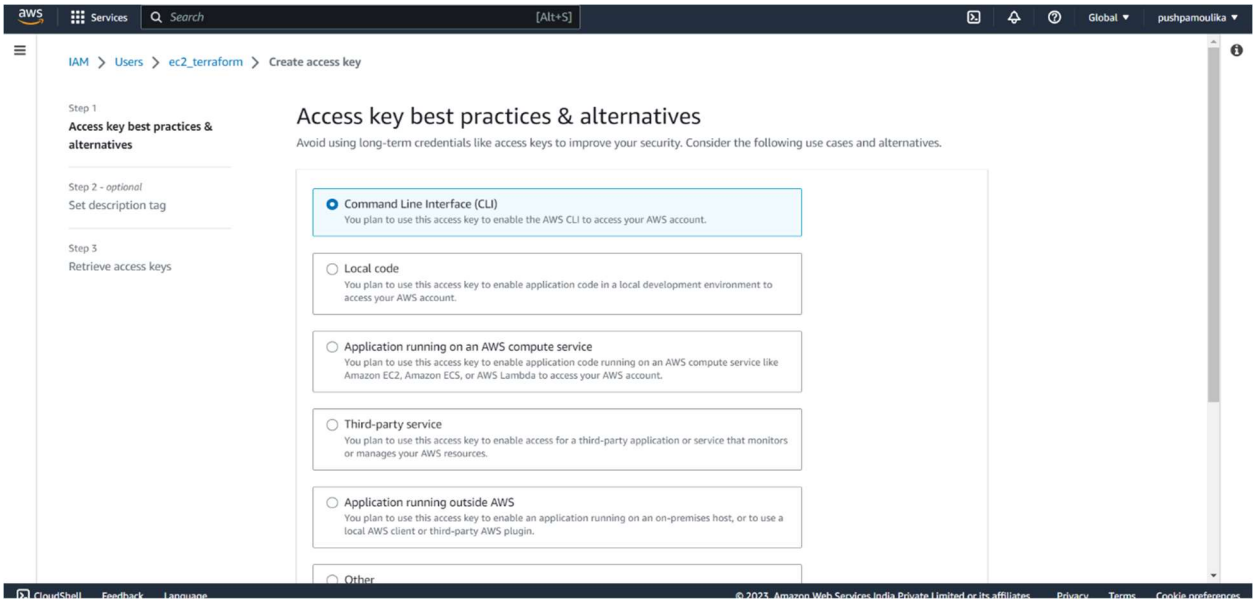

Services
Search
[Alt+S]
Global
pushpamoulika
IAM > Users > Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
Specify user details
User details
User name
ec2_terraform
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)
Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.
Are you providing console access to a person?
Specify a user in Identity Center - Recommended
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.
I want to create an IAM user
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.
Console password
Autogenerated password
You can view the password after you create the user.
Custom password
Enter a custom password for the user.
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

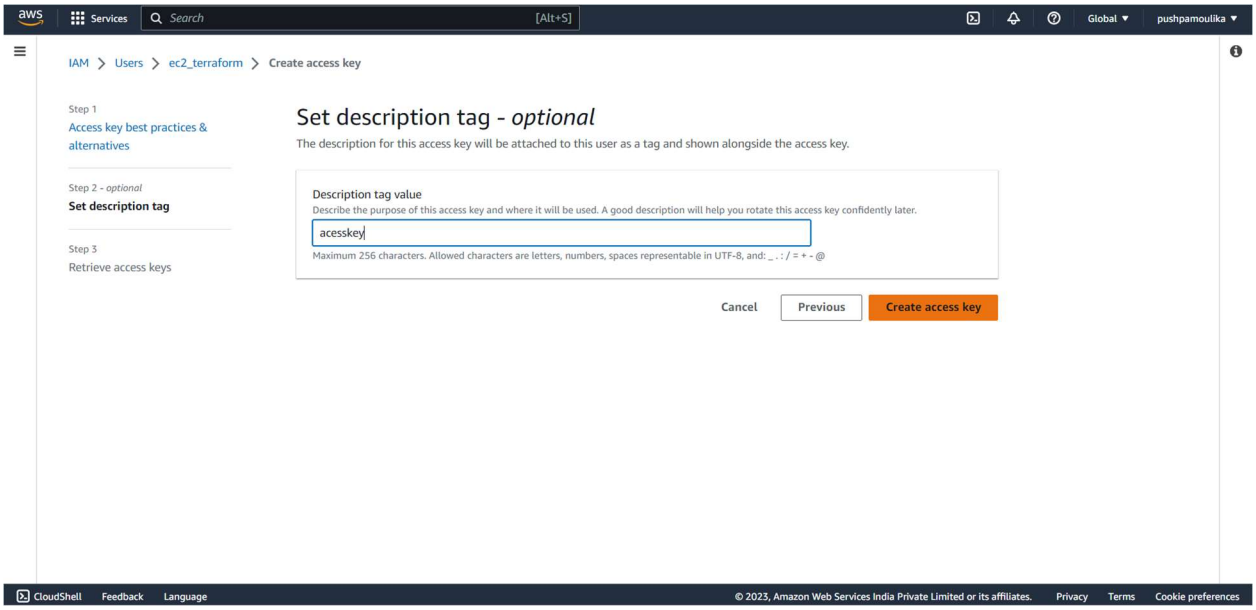

Are you providing console access to a person?
Specify a user in Identity Center - Recommended
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.
I want to create an IAM user
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.
Console password
Autogenerated password
You can view the password after you create the user.
Custom password
Enter a custom password for the user.
Must be at least 8 characters long
Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # $ % ^ & * ( ) _ + - (hyphen) = [ ] { } | '
Show password
Users must create a new password at next sign-in (recommended).
Users automatically get the IAMUserChangePassword policy to allow them to change their own password.
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

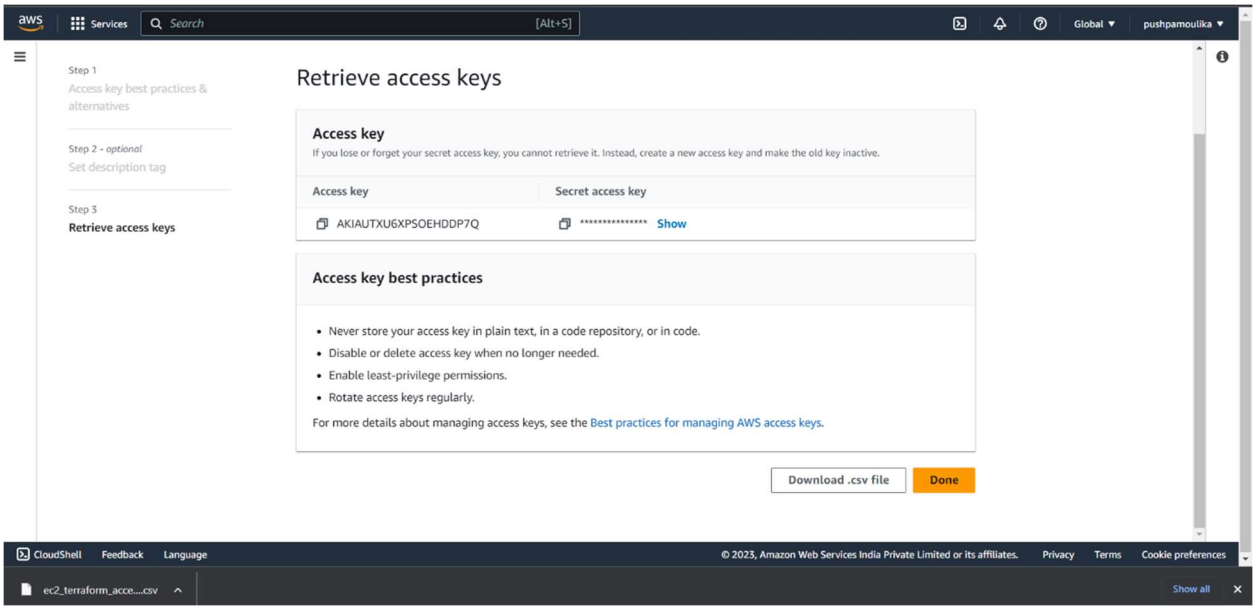

IAM > Users > Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
Set permissions
Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more
Permissions options
Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
Attach policies directly
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.
Permissions policies (1/1068)
Choose one or more policies to attach to your new user.
Create policy
Filter distributions by text, property or value
Policy name
Type
Attached entities
AccessAnalyzerServiceRolePolicy
AWS managed
0
AdministratorAccess
AWS managed - job function
0
AdministratorAccess-Amplify
AWS managed
0


Services
Search
[Alt+S]
Global
pushpamoulika
User created successfully
You can view and download the user's password and email instructions for signing in to the AWS Management Console.
View user
IAM > Users > Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
Retrieve password
You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.
Console sign-in details
Email sign-in instructions
Console sign-in URL
https://317267557348.signin.aws.amazon.com/console
User name
ec2_terraform
Console password
********** Show
Download .csv file
Return to users list


Services
Search
[Alt+S]
Global
pushpamoulika
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
Settings
Credential report
Organization activity
Service control policies (SCPs)
IAM > Users > ec2_terraform
ec2_terraform
Delete
Summary
ARN
arn:aws:iam::317267557348:user/ec2_terraform
Console access
Enabled without MFA
Access key 1
Not enabled
Created
April 04, 2023, 15:11 (UTC+05:30)
Last console sign-in
Never
Access key 2
Not enabled
Permissions
Groups
Tags
Security credentials
Access Advisor
Console sign-in
Manage console access
Console sign-in link
https://317267557348.signin.aws.amazon.com/console
Console password
Updated 3 minutes ago (2023-04-04 15:11 GMT+5:30)
Last console sign-in
Never
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences


Services
Search
[Alt+S]
Global
pushpamoulika
IAM > Users > ec2_terraform > Create access key
Step 1
Access key best practices & alternatives
Step 2 - optional
Set description tag
Step 3
Retrieve access keys
Access key best practices & alternatives
Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.
Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.
Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.
Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
Application running outside AWS
You plan to use this access key to enable an application running on an on-premises host, or to use a local AWS client or third-party AWS plugin.
Other
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences


Services
Search
[Alt+S]
Global
pushpamoulika
IAM > Users > ec2_terraform > Create access key
Step 1
Access key best practices & alternatives
Step 2 - optional
Set description tag
Step 3
Retrieve access keys
Set description tag - optional
The description for this access key will be attached to this user as a tag and shown alongside the access key.
Description tag value
Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.
acesskey
Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: _ . : / = + - @
Cancel
Previous
Create access key
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences



STEP2: Now open PowerShell with run as administrator and give the command aws configure.

It asks for access key and secret key give them later it asks for region give default.

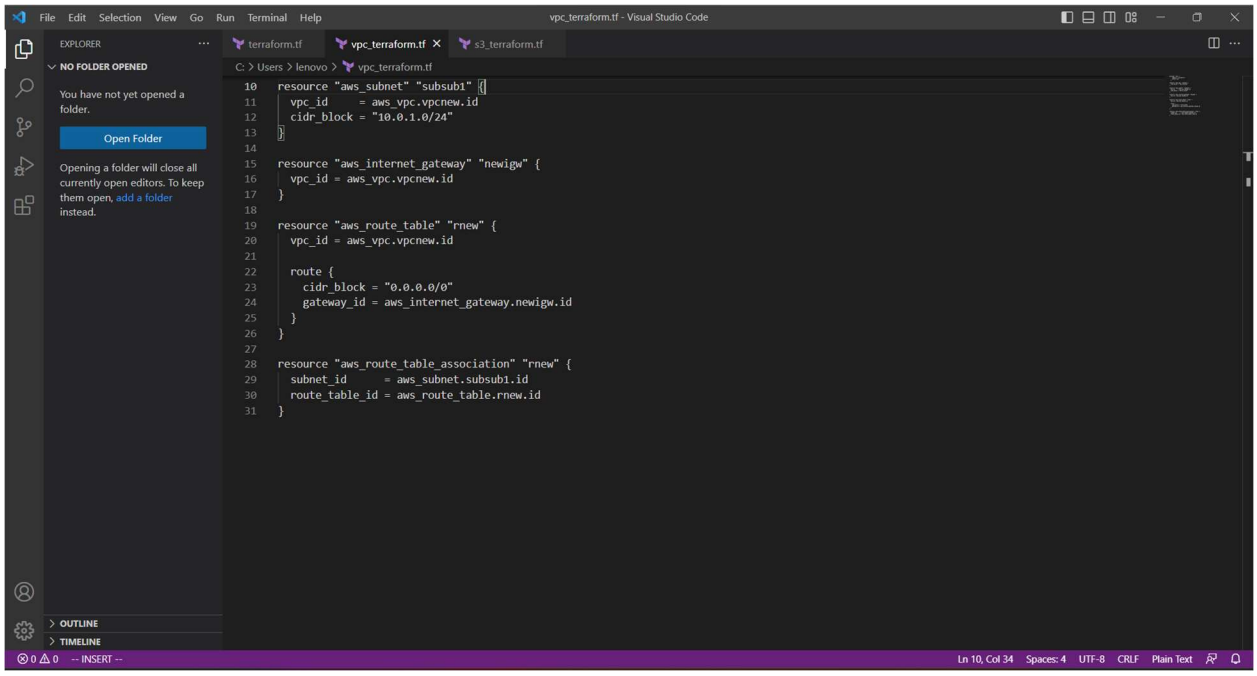
```
PS C:\Users\lenovo> aws configure
AWS Access Key ID [None]: AKIAUTXU6XPSOEHDDP7Q
AWS Secret Access Key [None]: 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
```

```
PS C:\Users\lenovo> aws configure
AWS Access Key ID [None]: AKIAUTXU6XPSOEHDDP7Q
AWS Secret Access Key [None]: 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
Default region name [None]: default
Default output format [None]: default
PS C:\Users\lenovo>
```

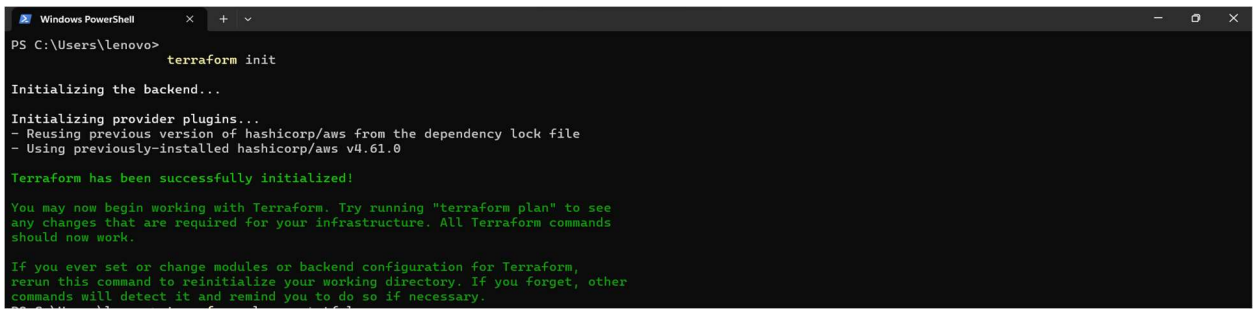
To generate the SSH key give the command ssh-keygen.

```
PS C:\Users\lenovo> ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\lenovo/.ssh/id_rsa): sshfile.txt
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in sshfile.txt
Your public key has been saved in sshfile.txt.pub
The key fingerprint is:
SHA256:TtahFL3QAHDZ4b7qhDHQqHnvxgCcSypZ3Frkgu9tBxg lenovo@LAPTOP-FOOK5AP6
The key's randomart image is:
+---[RSA 3072]----+
|     ..o++=      |
|    o....o.o     |
|.o++.    o...    |
|o*E.+  o o..     |
|==o*o   S .      |
|+o=..+ + .       |
|.. +o.. o        |
|  ..=...         |
|   o.oo          |
+----[SHA256]-----+
PS C:\Users\lenovo>
```

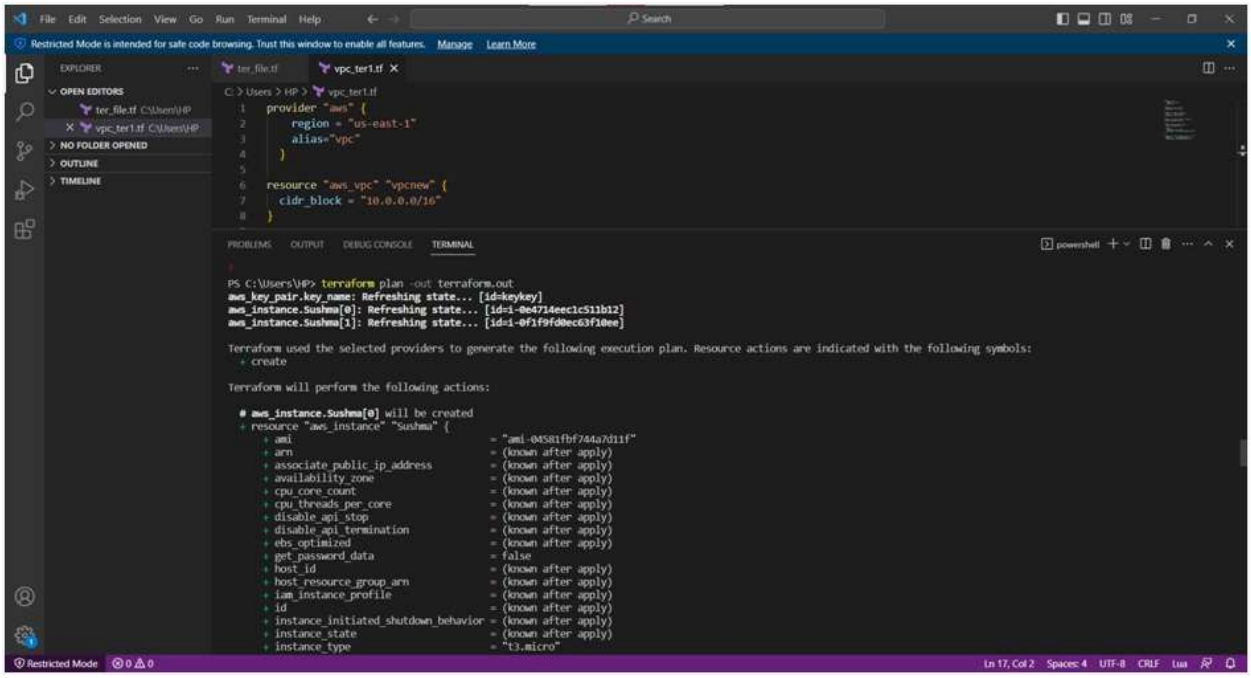
STEP3: Now create a folder in a desired path and open it in visual studio code and type the following code.



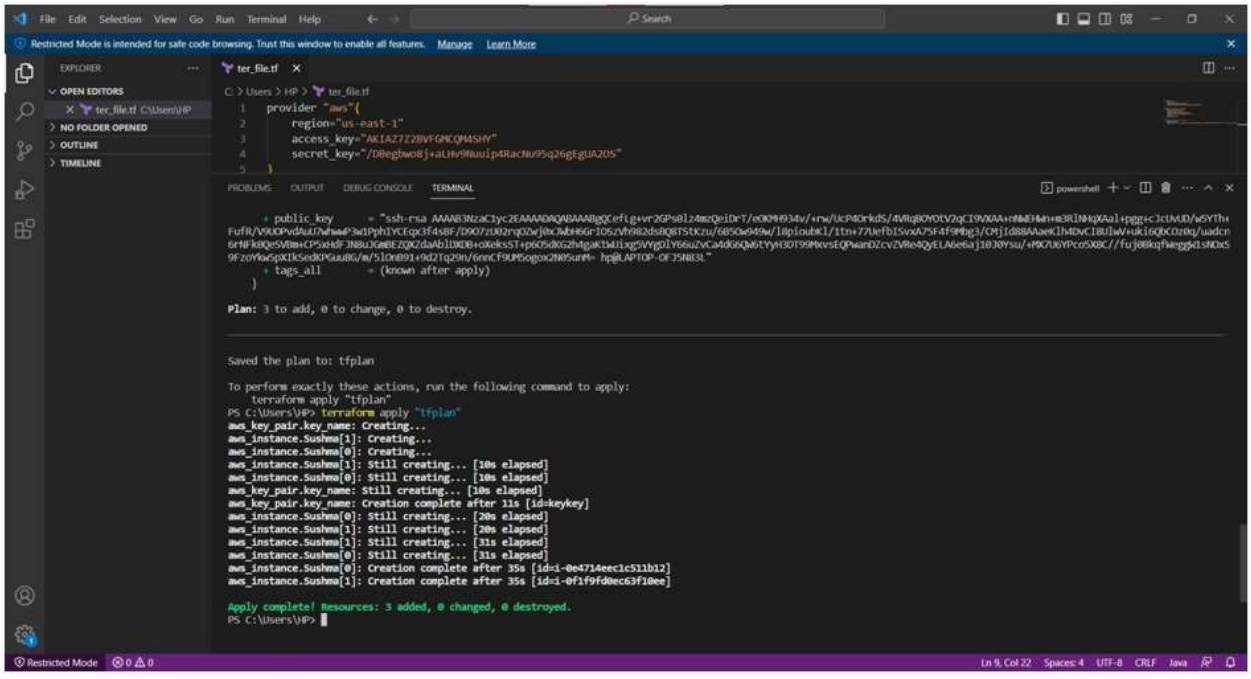
STEP 4: Now run the following command **“terraform init”** Terraform init command is used Initializes a Terraform working directory by downloading the necessary provider plugins and modules.



STEP5: Now after the terraform init command give the following command **“terraform plan”** terraform plan command is used generates an execution plan that shows what changes Terraform will make to your infrastructure. This is a useful command to run before applying changes to your infrastructure.



STEP6: Now after the terraform plan command give the following command **“terraform apply”** Applies the changes specified in the Terraform configuration to your infrastructure.



REPOSITORY DETAILS

main.tf: This file contains the main Terraform code that defines the VPC, subnets, route tables, and other resources.

variables.tf: This file defines the input variables that are used in the main.tf file. This allows you to customize the VPC configuration, such as the VPC CIDR block or the number of subnets.

outputs.tf: This file defines the output variables that are generated by the main.tf file, such as the VPC ID or the public and private subnets.

terraform.tfvars: This file contains the values for the input variables defined in variables.tf.

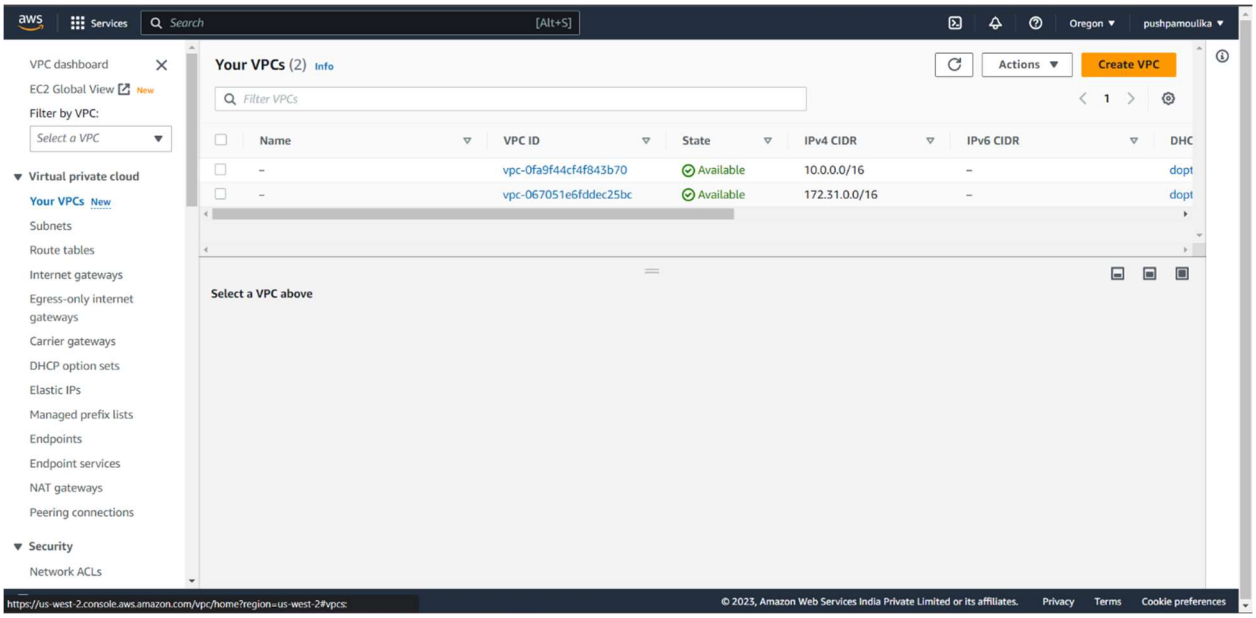
provider.tf: This file specifies the provider details, such as AWS credentials and region.

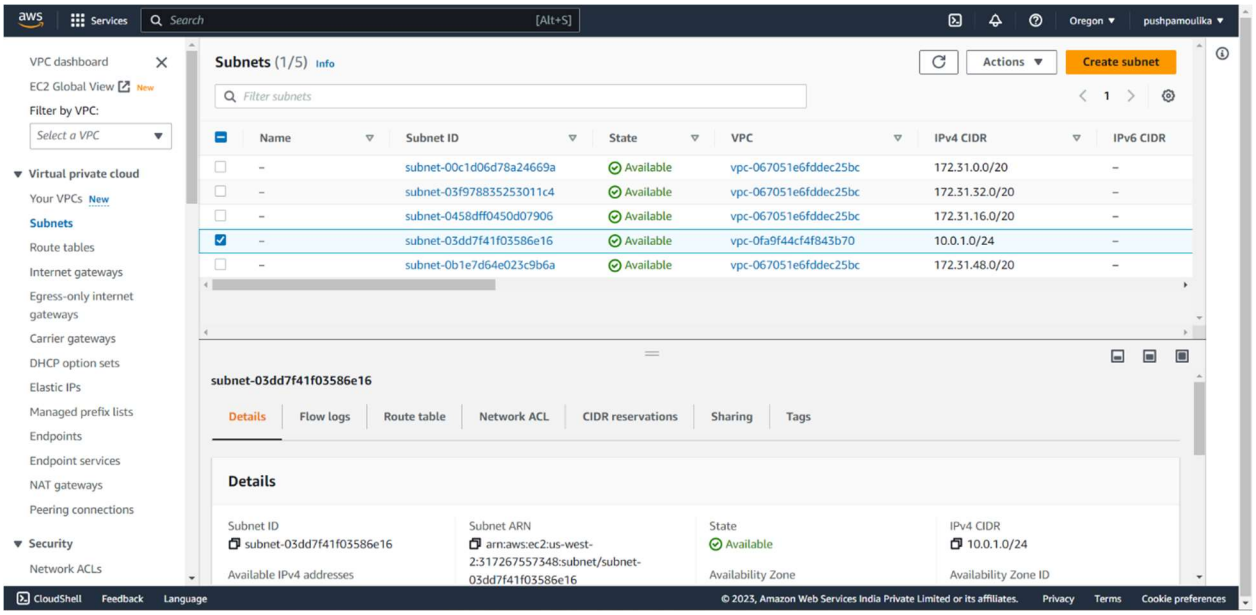
modules/: This directory contains any Terraform modules used in the project, such as a module for creating EC2 instances.

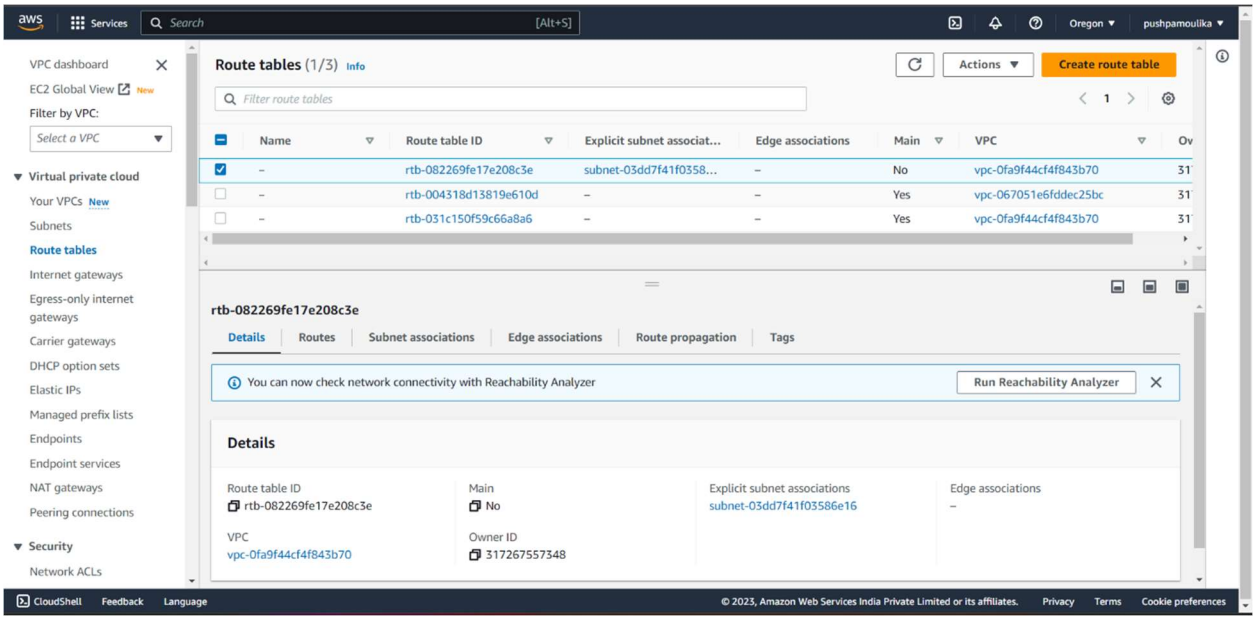
main.tf: This file contains the main Terraform code that defines the VPC, subnets, route tables, and other resources.

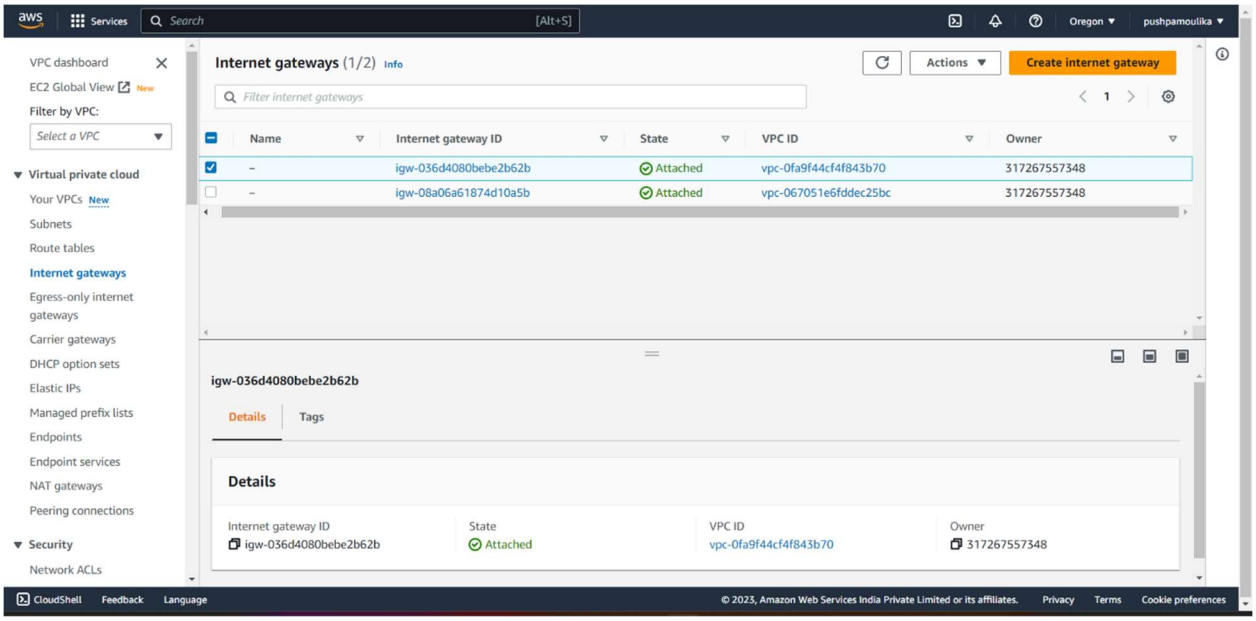
OUTCOME

The outcome of a VPC using Terraform project is a fully functional and customizable VPC infrastructure in Amazon Web Services (AWS). With the help of Terraform, you can quickly and easily create and manage VPC resources, including subnets, route tables, security groups, and EC2 instances. This allows you to set up a secure and isolated network environment that is tailored to the needs of your application. By using Terraform to define and manage your VPC infrastructure as code, you can ensure consistency and repeatability, as well as easy collaboration among team members. Moreover, Terraform enables you to easily update, modify, or delete VPC resources, and provides a clear view of the entire infrastructure state. Overall, the outcome of a VPC using Terraform project is a reliable, scalable, and maintainable VPC infrastructure that can support your application's growth and evolution.


aws
Services
Search
[Alt+S]
Oregon
pushpamoulika
VPC dashboard
EC2 Global View
New
Filter by VPC:
Select a VPC
Virtual private cloud
Your VPCs
New
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Peering connections
Security
Network ACLs
Your VPCs (2)
Info
Actions
Create VPC
Filter VPCs
Name
VPC ID
State
IPv4 CIDR
IPv6 CIDR
vpc-0fa9f44cf4f843b70
Available
10.0.0.0/16
vpc-067051e6fddec25bc
Available
172.31.0.0/16
Select a VPC above
https://us-west-2.console.aws.amazon.com/vpc/home?region=us-west-2#vpcs:
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences


aws
Services
Search
[Alt+S]
Oregon
pushpamoulika
VPC dashboard
EC2 Global View
New
Filter by VPC:
Select a VPC
Virtual private cloud
Your VPCs
New
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Peering connections
Security
Network ACLs
Subnets (1/5)
Info
Actions
Create subnet
Filter subnets
Name
Subnet ID
State
VPC
IPv4 CIDR
IPv6 CIDR
subnet-00c1d06d78a24669a
Available
vpc-067051e6fddec25bc
172.31.0.0/20
subnet-03f978835253011c4
Available
vpc-067051e6fddec25bc
172.31.32.0/20
subnet-0458dff0450d07906
Available
vpc-067051e6fddec25bc
172.31.16.0/20
subnet-03dd7f41f03586e16
Available
vpc-0fa9f44cf4f843b70
10.0.1.0/24
subnet-0b1e7d64e023c9b6a
Available
vpc-067051e6fddec25bc
172.31.48.0/20
subnet-03dd7f41f03586e16
Details
Flow logs
Route table
Network ACL
CIDR reservations
Sharing
Tags
Details
Subnet ID
subnet-03dd7f41f03586e16
Subnet ARN
arn:aws:ec2:us-west-2:317267557348:subnet/subnet-03dd7f41f03586e16
State
Available
IPv4 CIDR
10.0.1.0/24
Available IPv4 addresses
Availability Zone
Availability Zone ID
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences





At the end of the VPC using Terraform project, you should have a fully provisioned and functional VPC infrastructure in your chosen cloud provider (such as AWS, Azure, or Google Cloud Platform). This infrastructure should include all necessary resources such as VPC, subnets, routing tables, security groups, and any additional resources required for your application.

Once the infrastructure is set up, you can deploy your application and start using the VPC to host your application. You can also continue to use Terraform to manage and update the infrastructure over time, making it easier to scale and maintain your application.

The final outcome of the VPC using Terraform project is a reliable and flexible network infrastructure that enables your application to run efficiently and securely in the cloud.

CONCLUSION

In conclusion, using Terraform for creating and managing a Virtual Private Cloud (VPC) in Amazon Web Services (AWS) offers many benefits, such as improved infrastructure consistency, reduced human error, and increased automation. Terraform allows you to define your infrastructure as code, enabling you to rapidly and consistently create and manage your cloud infrastructure. With the help of Terraform, you can easily create and configure VPC resources, including subnets, route tables, security groups, and EC2 instances. Moreover, Terraform enables you to easily update, modify, or delete VPC resources, and provides a clear view of the entire infrastructure state. As such, Terraform is a powerful tool that can significantly enhance the management and provisioning of VPCs in AWS, making it an essential technology for cloud infrastructure management.