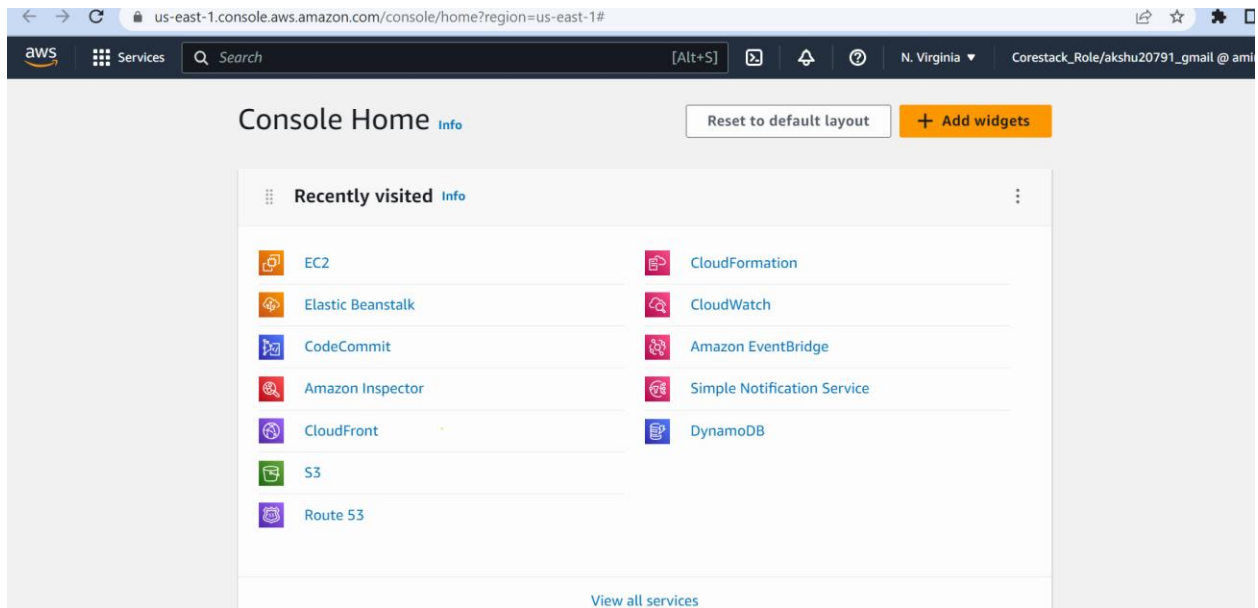
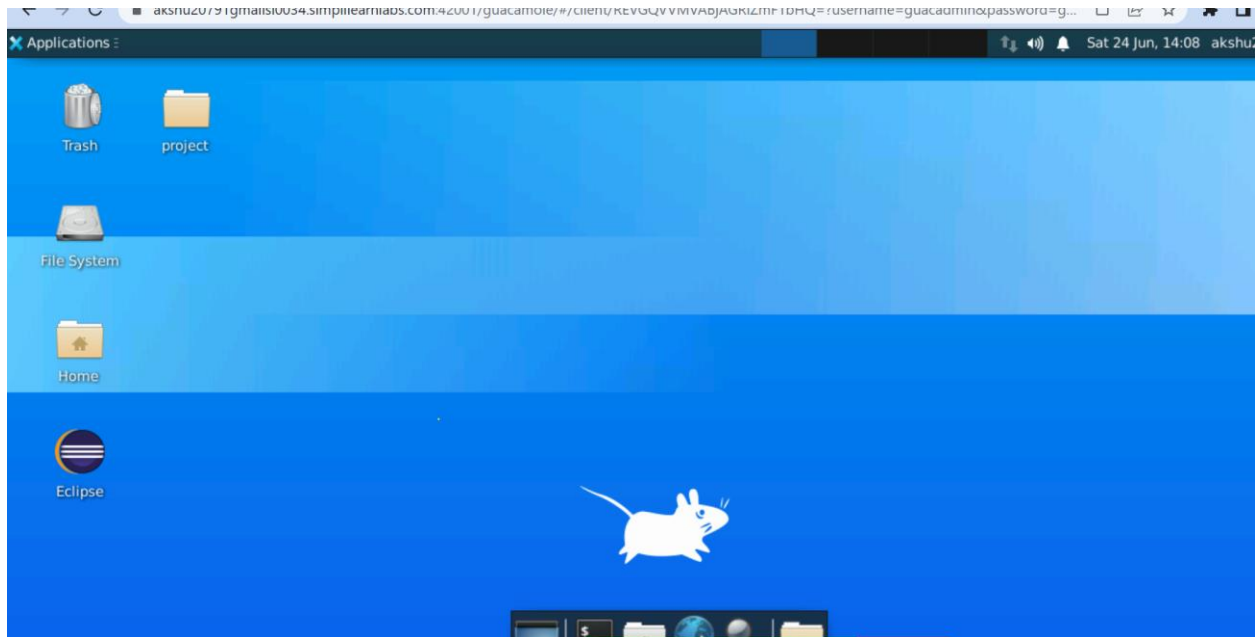


Terraform : Infrastructure as a code service

Launch aws and Devops in aws v2 machine



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

Go to ubuntu machine:

```
wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o  
/usr/share/keyrings/hashicorp-archive-keyring.gpg
```

```
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg]  
https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee  
/etc/apt/sources.list.d/hashicorp.list
```

```
sudo apt update && sudo apt install terraform
```

```
terraform --version
```

```
akshu20791gmail@ip-172-31-83-7:~$ terraform --version  
Terraform v1.5.1  
on linux_amd64
```

Lets create a directory where we will work

```
$ mkdir akshatdir
```

```
akshu20791gmail@ip-172-31-83-7:~$ cd akshatdir  
akshu20791gmail@ip-172-31-83-7:~/akshatdir$
```

Open : <https://registry.terraform.io/providers/hashicorp/aws/latest/docs> these are complete documentation of any resource u want to create in terraform

optionally `token` , to the `aws` provider block.

Usage:

```
provider "aws" {  
  region = "us-west-2"  
  access_key = "my-access-key"  
  secret_key = "my-secret-key"  
}
```

I will be creating Infra on AWS

AWS datacenter

These are credentials we need to input to authenticate the access to AWS.

Other settings related to authorization can be configured, such as:

- `profile`
- `shared_config_files`

In the terminal of your ubuntu machine:

```
@ip-172-31-83-7:~/akshatdir$ vi ec2.tf
```

terraform file

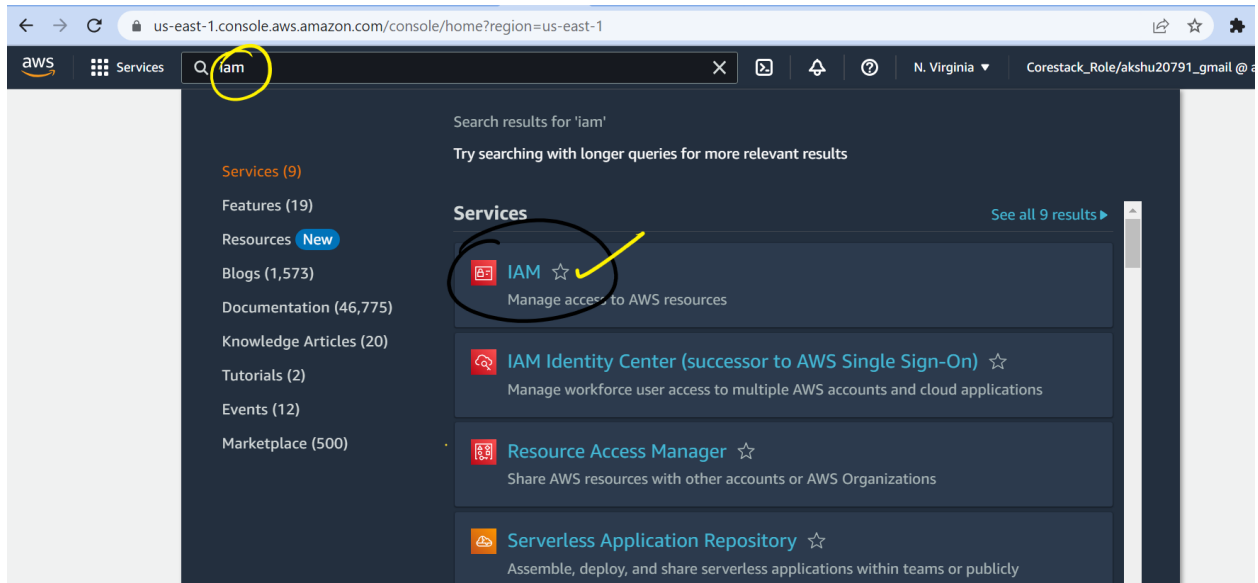
vi ec2.tf

extension for terraform is .tf

```
File Edit View Search Terminal Help
provider "aws" {
  region    = "us-east-1" #by default the resources would be created in north virginia of aws account
  access_key = "my-access-key"
  secret_key = "my-secret-key"
}
~
~
~
~
~
~
~
```

But from where
my access key & secret key
will come ??

To get access key and secret access key ...go to your aws account



aws

Services

Search

[Alt+S]

Global

Corestack_Role/akshu2

Identity and Access Management (IAM)

Unable to load search

Dashboard

Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

Access reports

IAM dashboard

Security recommendations 2

Add MFA for root user

Sign in as the root user (or contact your administrator) and register a multi-factor authentication (MFA) device for the root user to improve security for this account.

Update your access permissions for AWS Billing, Cost Management, and Account consoles.

The following IAM actions for AWS Billing, Cost Management, and Account consoles will reach the end of standard support: 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. These actions will be replaced with granular IAM actions. Examples of impacted features include AWS Cost Explorer, AWS Budgets, Billing console, and more. To ensure you don't lose access, [update your policies](#) before July 2023 or contact your access administrator to complete your action.
06/06/23 update- You can now use the [Bulk Policy Migrator](#) to mass update policies from your Paver account (if using AWS Organization). The updates need to be performed before

View affected policies

AW:
Acc
:
Acc
amir
Sign
acc
I
maz
Too
Poli
The

aws

Services

Search

[Alt+S]

Global

Corestack_Role/akshu20791_gmail @ am

Identity and Access Management (IAM)

Unable to load search

Dashboard

Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

Access reports

IAM > Users

Users (1) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Find users by username or access key

	User name	Groups	Last activity	MFA	Password age
<input type="checkbox"/>	corestack-4f3f2	AdministratorGroup	Never	None	1028 days ago

Add users

IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Specify user details

User details

User name

user1

The username 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.

i 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**

Services Search [Alt+S] Global Corestack_Role/akshu20791_gmail @ amin15

Step 3
Review and create

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 (1/1104)

Choose one or more policies to attach to your new user.

Search Filter by Type All types

	Policy name	Type	Attached entities
<input type="checkbox"/>	AccessAnalyzerService...	AWS managed	0
<input checked="" type="checkbox"/>	AdministratorAccess	AWS managed - job function	3
<input type="checkbox"/>	AdministratorAccess-A...	AWS managed	0

Next

Create the user

Identity and Access Management (IAM)

Unable to load search

Dashboard

▼ Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

▼ Access reports

Access analyzer

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

Users (2) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Find users by username or access key

User name

Groups

Last activity

MFA

Password

☐

corestack-4f3f2

AdministratorGroup

Never

None

1028 da

☐

user1

None

Never

None

None

click

us-east-1.console.aws.amazon.com/iamv2/home/region=us-east-1#/users/details/user1/section=permissions

aws Services Search [Alt+S] Global Corestack_Role/akshu20791_gmail@amin

Identity and Access Management (IAM)

Unable to load search

Dashboard

▼ Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

▼ Access reports

Access analyzer

Archive rules

Summary

ARN

Console access

Access key 1

Created

Last console sign-in

Access key 2

Permissions

Groups

Tags

Security credentials

Access Advisor

Permissions policies

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type

Search

All types

☐

Policy name

Type

Attached via

Scroll down

RemoveResyncAssign MFA device

Device type	Identifier	Certifications	Created on
No MFA devices. Assign an MFA device to improve the security of your AWS environment			
Assign MFA device			

Access keys (0)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)

Create access key

No access keys

As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. [Learn more](#)

Create access key

Access key best practices & alternatives

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

☒ 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.

Search [Alt+S] Global Corestack_Role/akshu20791

☐ 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
Your use case is not listed here.

 **Alternatives recommended**

- Use [AWS CloudShell](#), a browser-based CLI, to run commands. [Learn more](#)
- Use the [AWS CLI V2](#) and enable authentication through a user in IAM Identity Center. [Learn more](#)

Confirmation

☒ I understand the above recommendation and want to proceed to create an access key.

Next

aws Services Search [Alt+S] Global Corestack_Role/akshu20791_gmail @

IAM > Users > user1 > 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.

Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: _ . : / = + - @

Cancel Previous **Create access key**

☑ Access key created

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.



alternatives

Step 2 - optional
Set description tag

Step 3
Retrieve access keys

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
 AKIAU6GUCH53PA5GSBE4	 odXgeLXbF3CDU+DstHTR+X9+Ix/xd09u+VRd0ZZI Hide

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.

Now copy them one by one and paste it in your code

```
aksnu2079igman@ip-172-31-85-7: ~/aksnat01r
File Edit View Search Terminal Help
provider "aws" {
  region = "us-east-1" #by default the resources would be created in north virginia of aws account
  access_key = "AKIAU6GUCH53PA5GSBE4"
  secret_key = "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"
}
~
~
~
~
~
~
~
~
~
~
```

paste access key

↓
paste secret key

Save and quit
press esc :wq