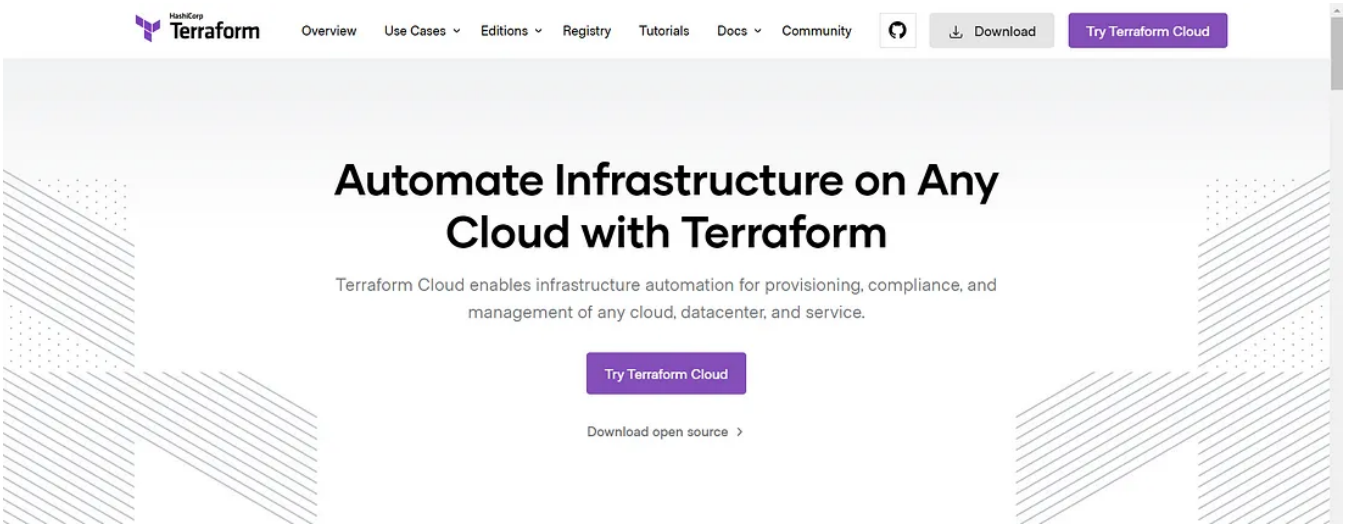
Gauri Shirkande · Follow
4 min read · Mar 6

Terraform is an open-source infrastructure as a code tool that lets you build, change, and version cloud and on-prem resources safely and efficiently. It helps us manage and provision the infrastructure using a simple language which is Hashicorp Configuration Language. Terraform being cloud-agnostic can be used to automate the provisioning and managing of resources.

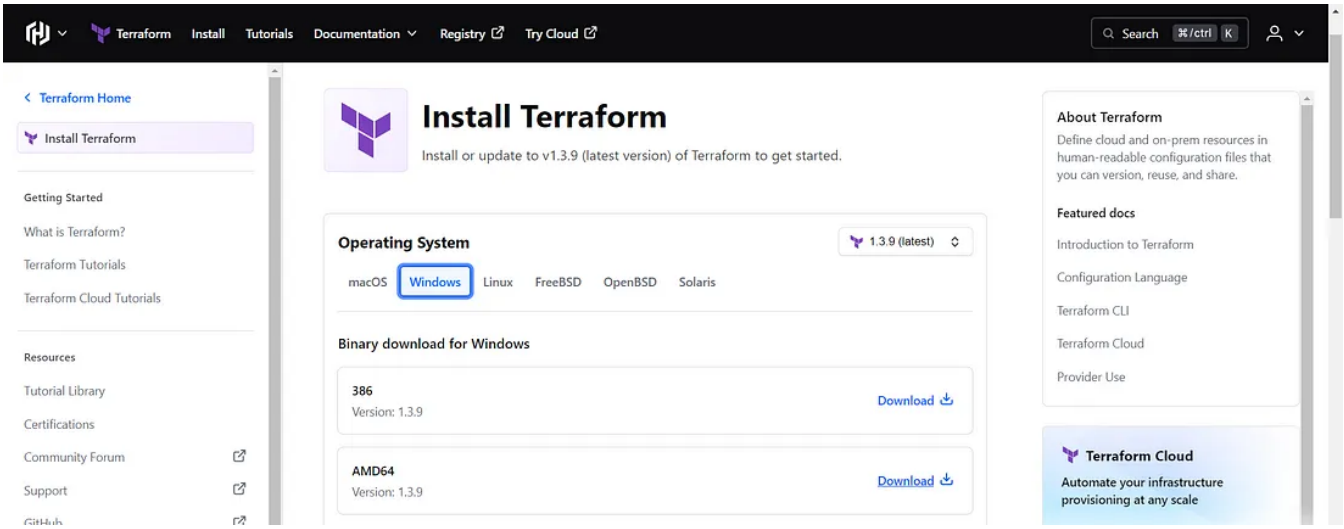
In this blog, we'll see how to create an AWS EC2 Instance using Terraform. So let's begin.

1. Download Terraform on a Windows machine

Go to the official website of terraform: https://www.terraform.io/.



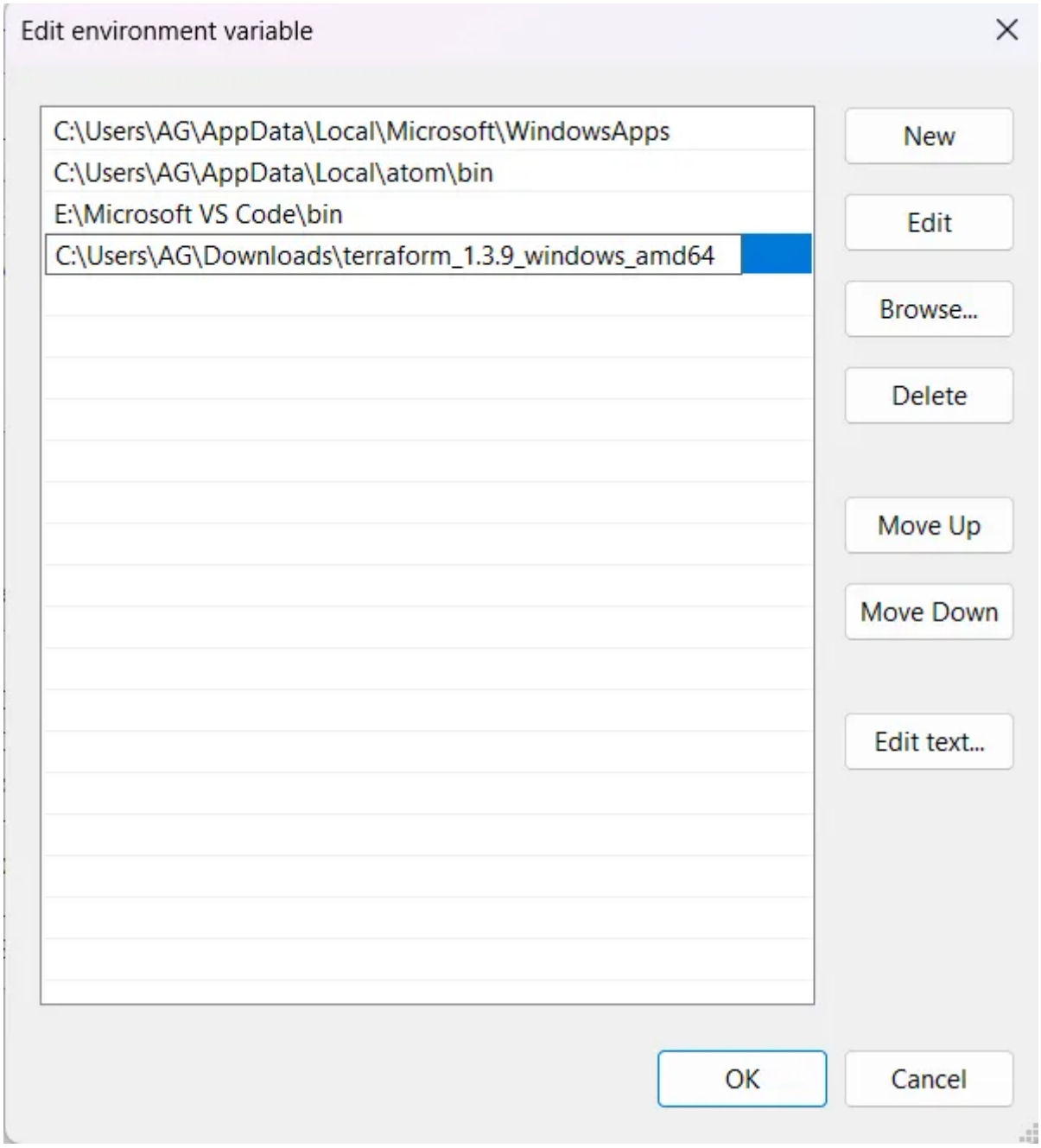
Click on **Download** and then select the **Windows** tab since we are downloading for a windows machine.



And then select the AMD64 binary for downloading terraform.

Once you click on Download, a zip folder for terraform gets downloaded on the local windows machine.

Unzip the folder and then add the path of your binary to the Environment Variables.



Once done, open the command prompt and check the version of installed terraform by using the command below:

```
terraform version
```

2. Download and Install AWS-CLI on windows

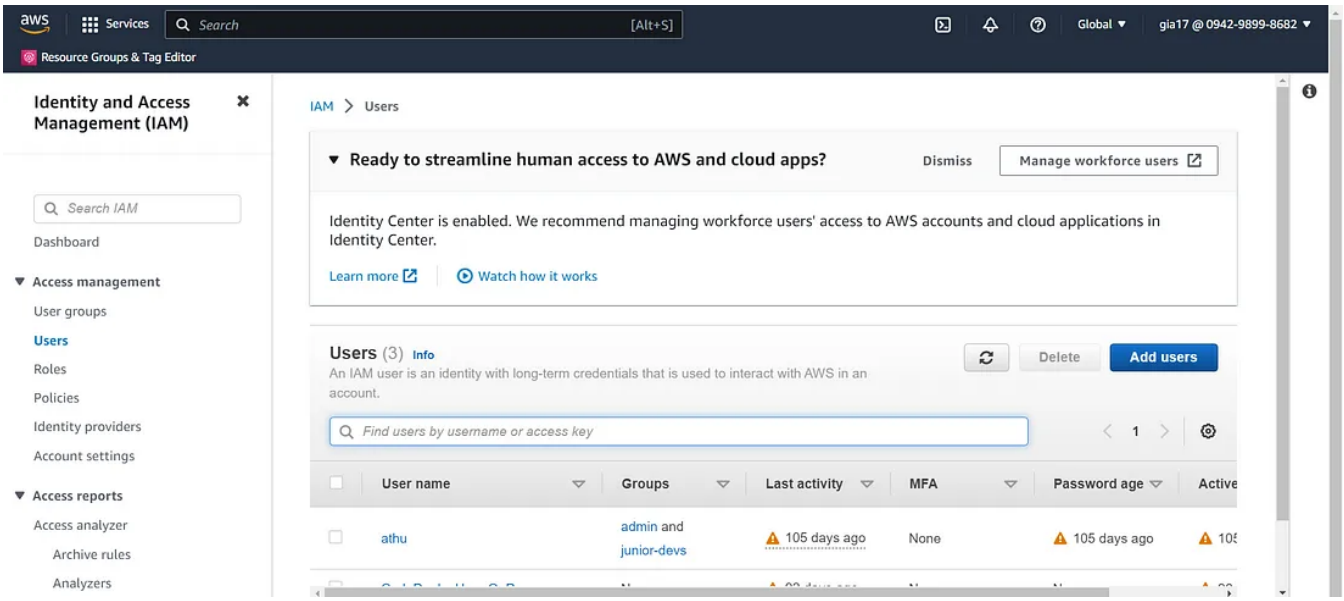
Download and run the AWS CLI MSI installer for Windows:
https://awscli.amazonaws.com/AWSCLIV2.msi

Once done, open the command prompt and check the version of the installed AWS CLI by using the command below:

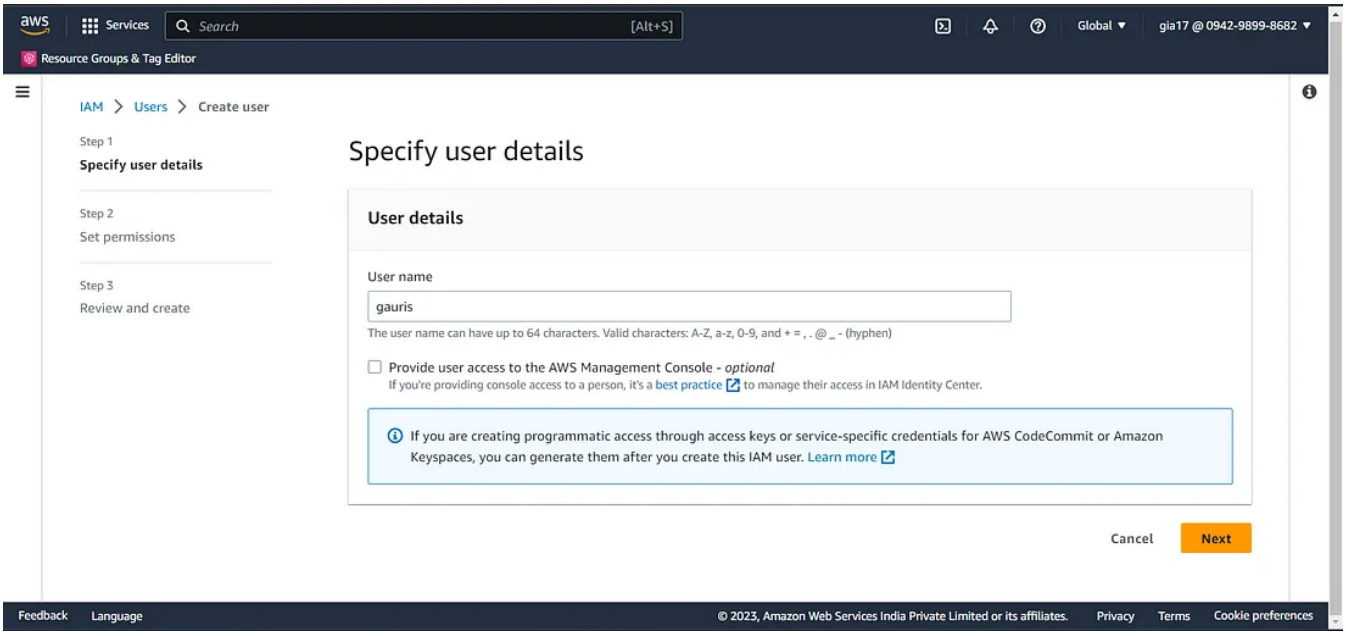
```
aws --version
```

3. Create an AWS User with the required permission

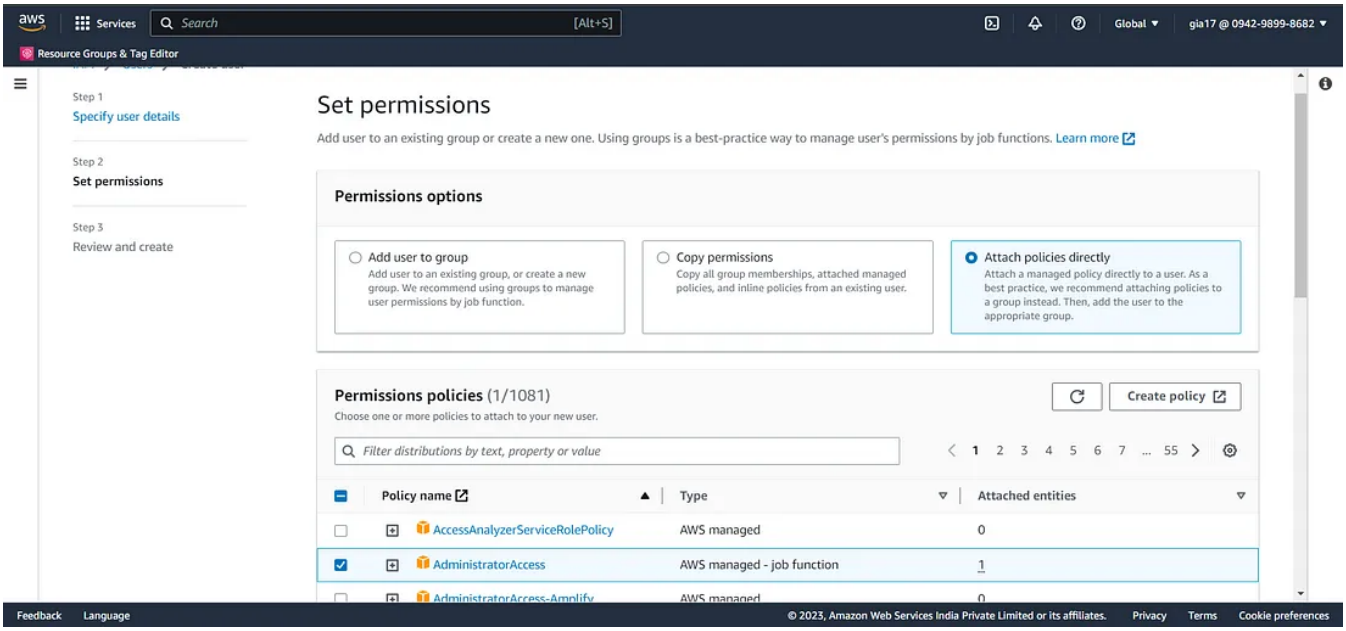
Go to your AWS Management Console and in services search for **IAM.** In the left side panel, select **Users.** In the **Users** section, click on **Add Users.**



Enter the **User name** and click **Next**

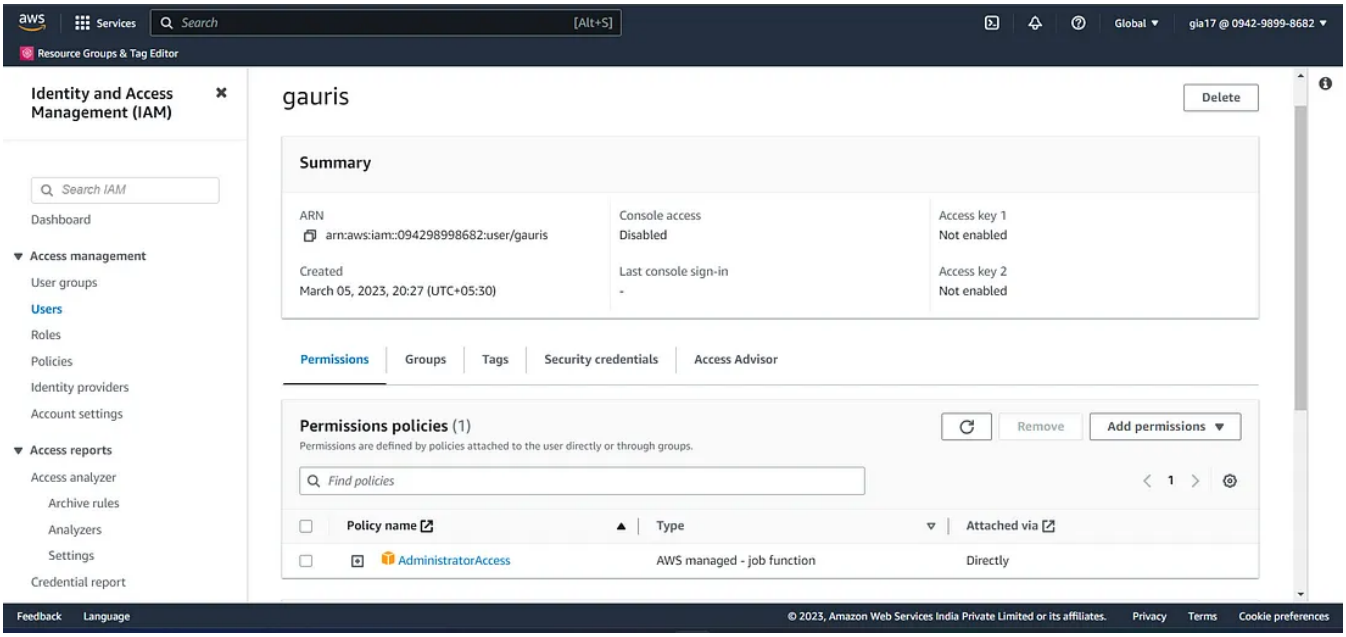


Select **Attach policies directly** option and select the **AdministratorAccess** or **AmazonEC2FullAccess** and click **Next** again.

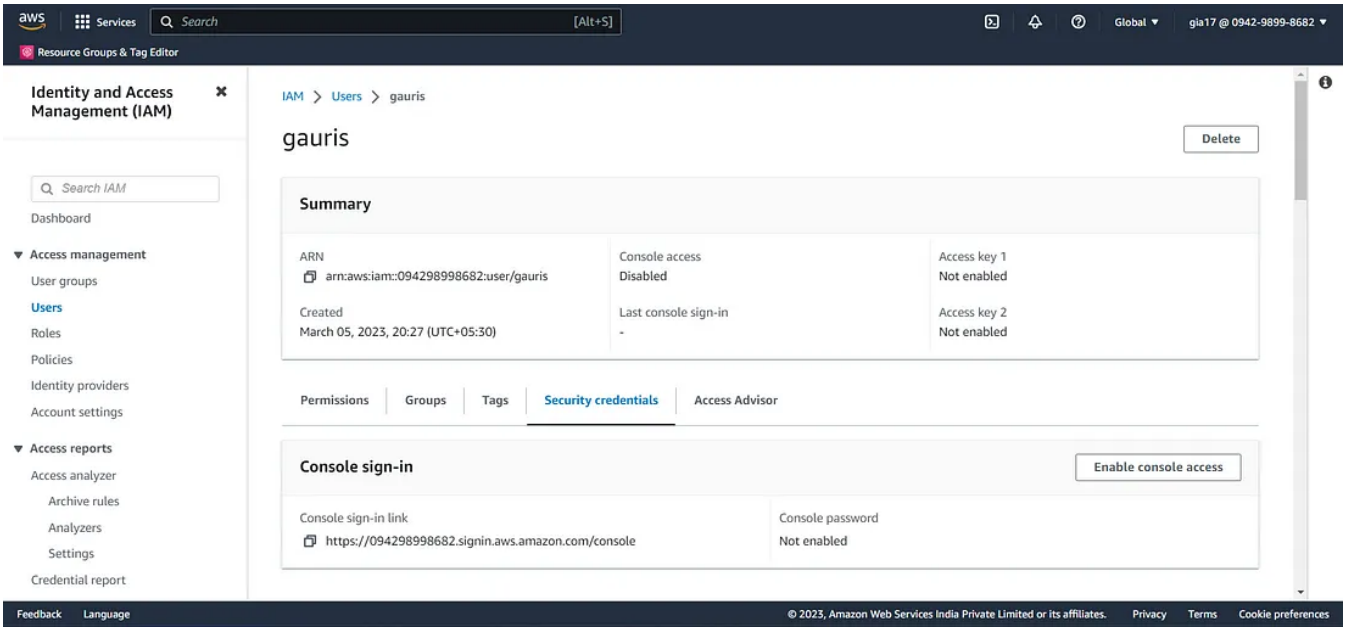


Then Review the User details and click on **Create User.**

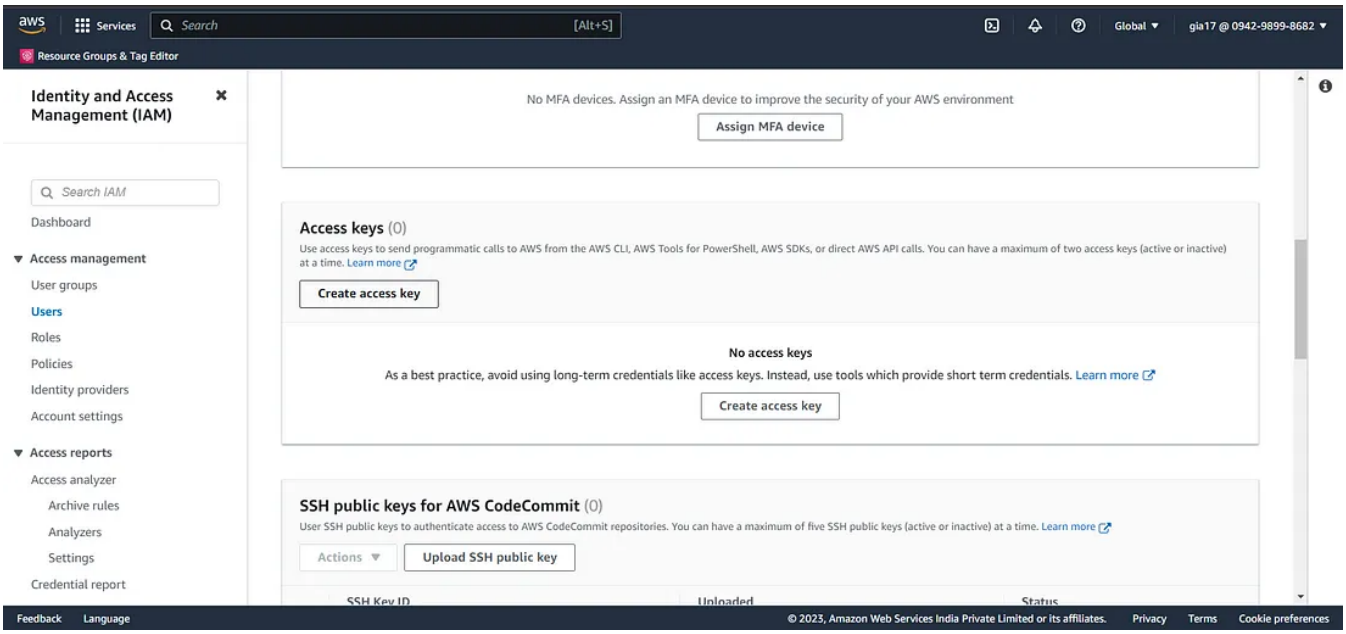
After this, you will see the user created.



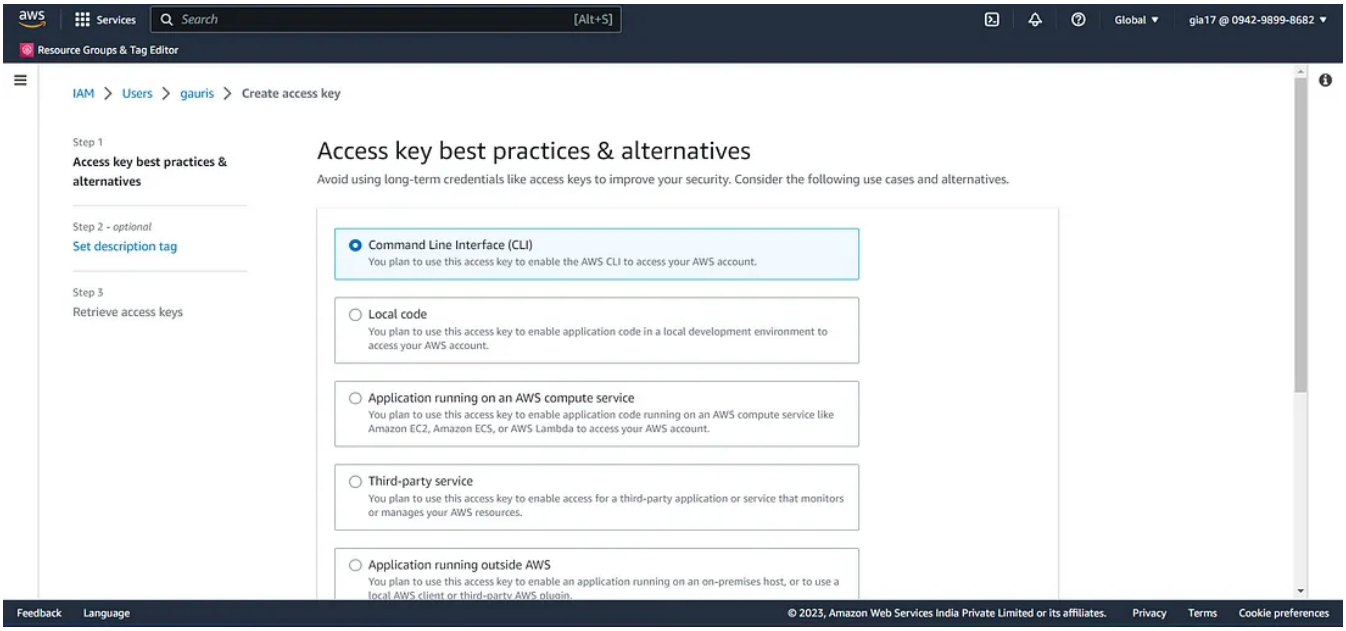
Then within the user select **Security credentials.**



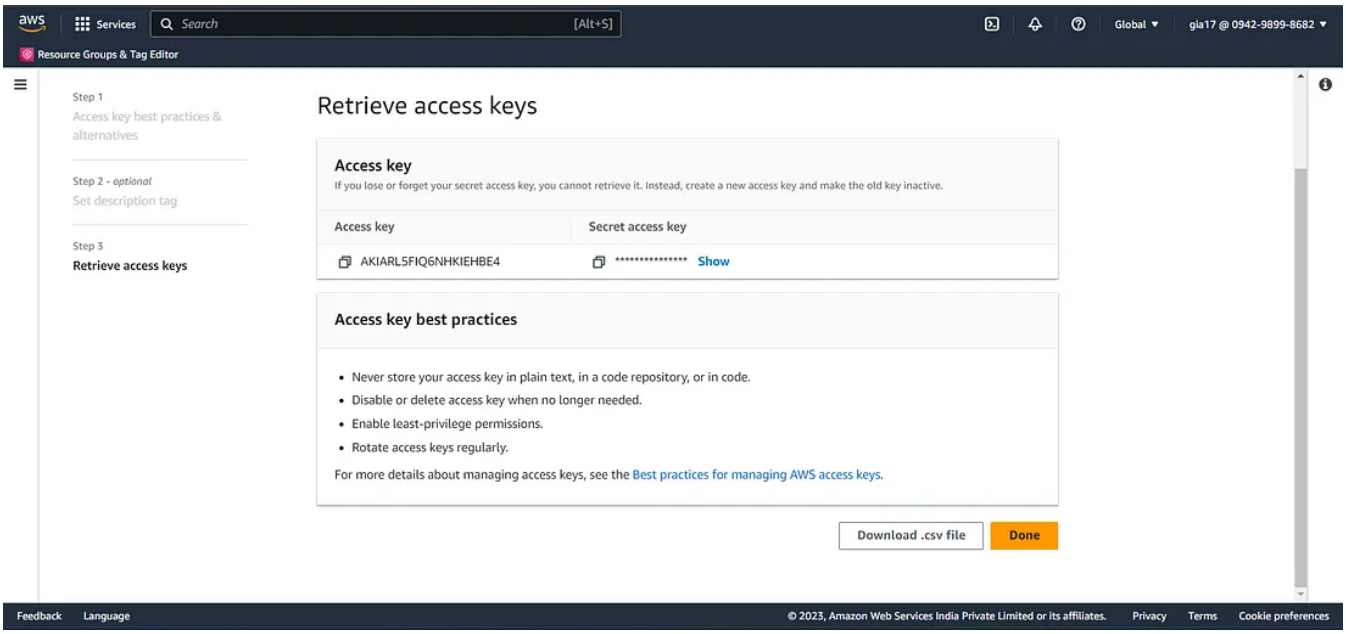
Create an Access Key for the user by selecting the **Create access key.**



Then select the **Command Line Interface (CLI)** and click **Next.**



Then **Retrieve access keys** by downloading the access and secret keys.



4. Configure AWS CLI

Open your Command Prompt and run **aws configure**

```
aws configure
```

It will prompt you to enter the Access and Secret keys along with the region and format.

```
AWS Access Key ID [None]: <Access Key ID>
AWS Secret Access Key [None]: <Secret Key ID>
Default region name [None]: <Region>
Default output format [None]: <format>
```

5. Create your terraform configuration file

Create a terraform configuration file with .tf extension in HCL (Hashicorp Configuration Language). For creating an EC2 instance we need to mention the AWS provider in the configuration file. So we first define the provider and then define the resource that is an ec2 instance with required parameters.

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

resource "aws_instance" "myInstance" {
  ami           = "ami-006dcf34c09e50022"
  instance_type = "t2.micro"

  tags = {
    Name = "MyInstance"
  }
}
```

6. Initialize the terraform working directory

Before actually creating the infrastructure we first need to initialize the working directory. This can be done using **terraform init** command. On running this command, it downloads all the required plugins necessary for building your infrastructure.

```
terraform init
```

In order to validate the configuration file run the command below:

```
terraform validate
```

7. Preview the changes with terraform plan

Once done with the initialization and validation of the configuration file, run the command below to preview the changes before actually implementing them on AWS.

```
terraform plan
```

8. Creating the infrastructure using terraform apply

Finally, run the command **terraform apply** to actually deploy the infrastructure on the AWS cloud. This command creates the infrastructure that was previously planned. With this command, **terraform.tfstate** file gets created which keeps the track of resources created using terraform. This file is by default stored locally, but it can also be stored remotely, which works better in a team environment.

```
terraform apply
```

On executing this command it asks for the re-confirmation. Once you enter **yes,** it implements the desired changes. You can see the instance running in your AWS Management Console after the process of creation is completed.

In this way, we can create an EC2 instance using Terraform.

Hope you found this blog useful. Happy Learning !!

Follow

Written by Gauri Shirkande

14 Followers

More from Gauri Shirkande

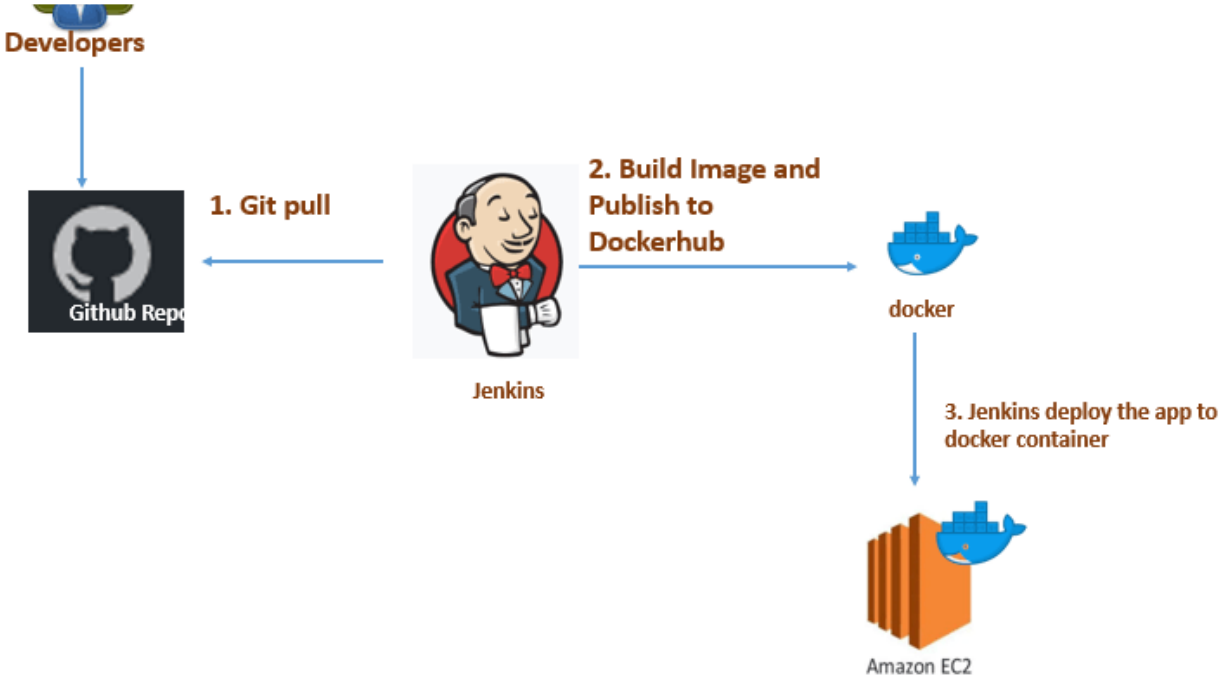
Gauri Shirkande

Understanding Kubernetes Architecture and Components, Installation and Configuration

What is Kubernetes?

6 min read · Apr 25

1



Gauri Shirkande

Create a pipeline using Jenkins, and GitHub and deploy it to AWS EC2

In this article, we are going to create a CI/CD pipeline using Jenkins for a NodeJS project.

5 min read · Mar 16

Gauri Shirkande

Terraform—Best practices

Terraform is one of the most popular IaC (Infrastructure as a Code) tools that automates the creation of infrastructure using code.

4 min read · Mar 25

1

Gauri Shirkande

Introduction to Terraform Modules

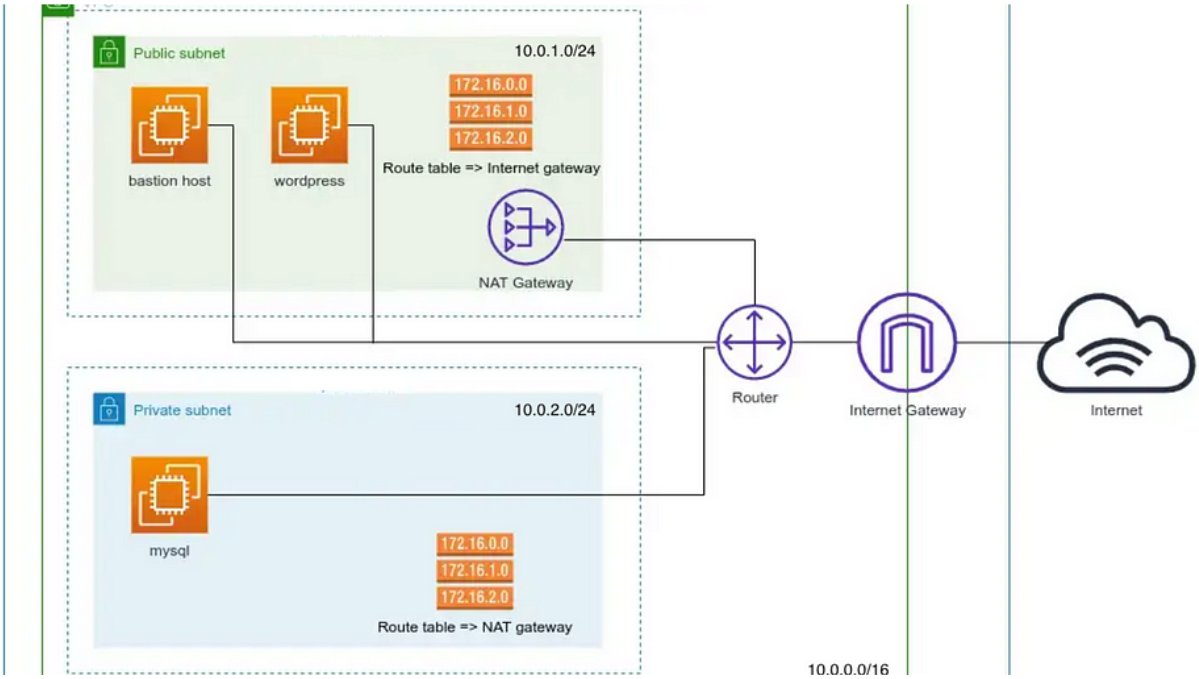
What are Terraform Modules?

3 min read · Mar 14

1

See all from Gauri Shirkande

Recommended from Medium



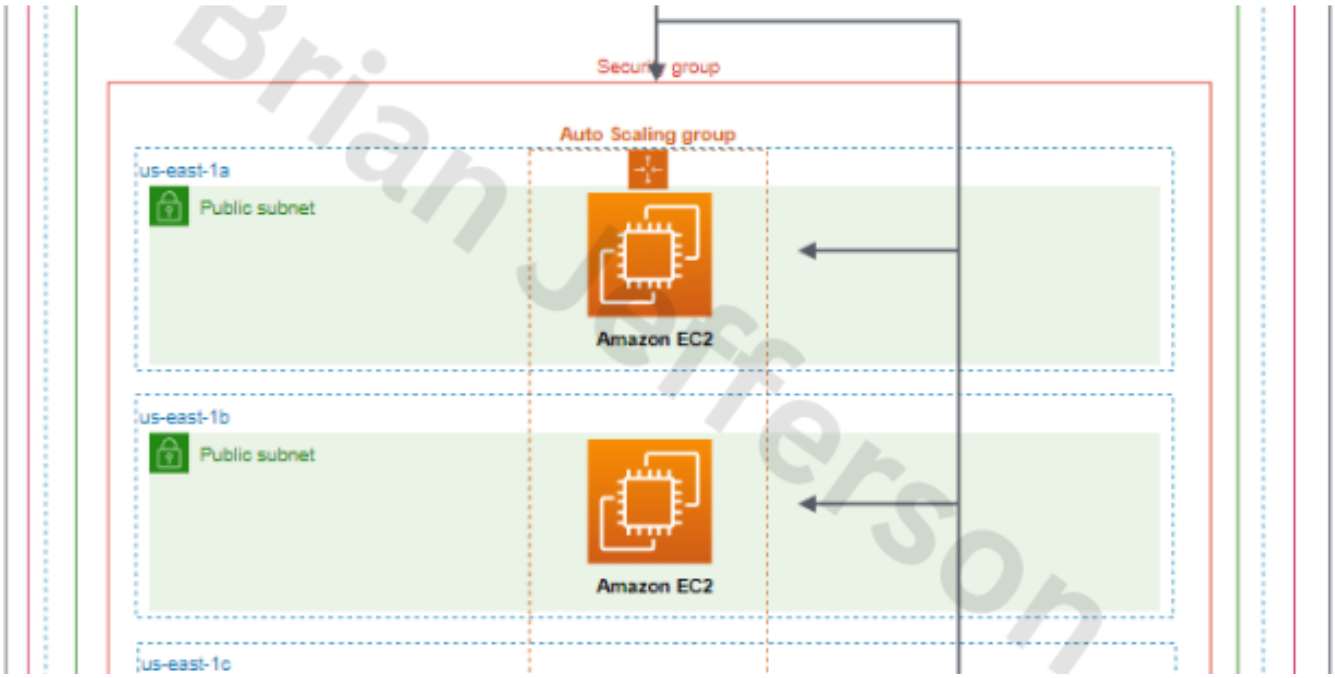
headinthecloudsin AWS Tip

Deploy WordPress and MySQL with Terraform on AWS EC2 Like a Pro

Prerequisites:

· 14 min read · Jan 3

16 2



Brian Jefferson

Ensuring High Availability: A Step-by-Step Guide to Creating an Auto-Scaling Group of EC2 Instances...

Scenario: We have been tasked to deploy a highly available web application on AWS by:

· 6 min read · May 1

9

Lists

Staff Picks

339 stories · 94 saves

Stories to Help You Level-Up at Work

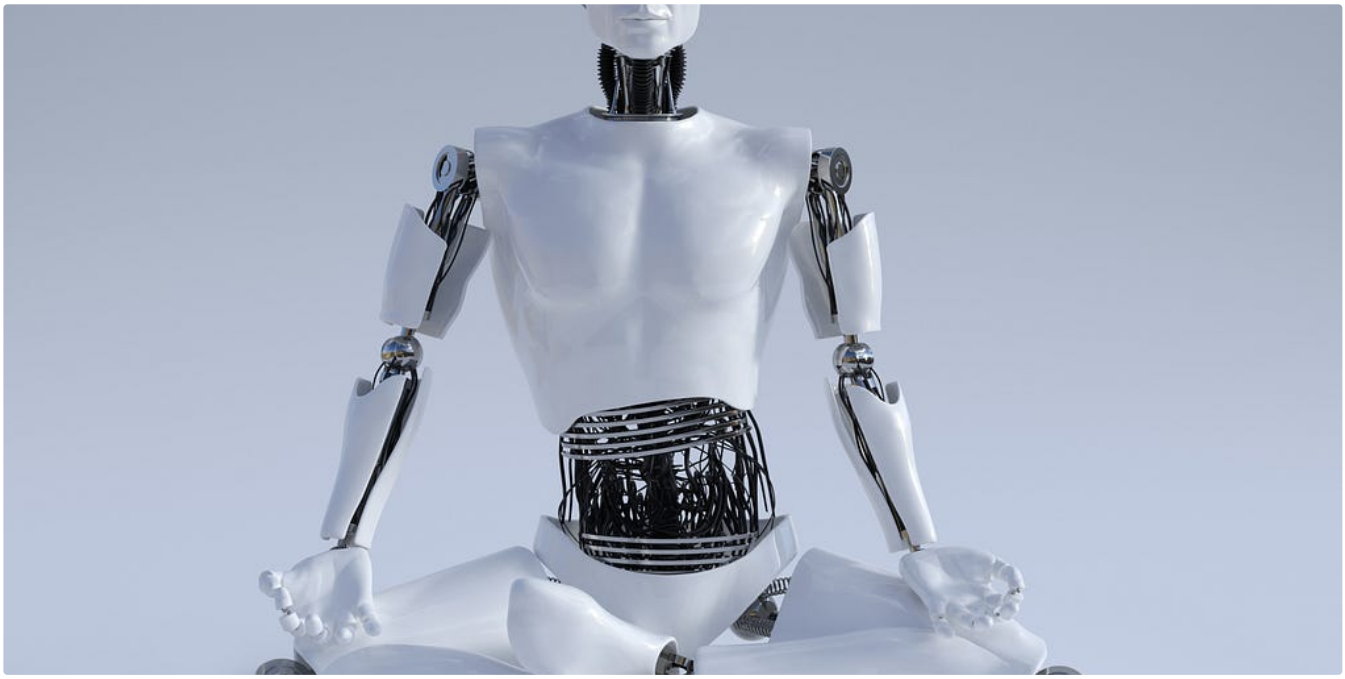
19 stories · 73 saves

Self-Improvement 101

20 stories · 130 saves

Productivity 101

20 stories · 142 saves



The PyCoach in Artificial Corner

You're Using ChatGPT Wrong! Here's How to Be Ahead of 99% of ChatGPT Users

Master ChatGPT by learning prompt engineering.

· 7 min read · Mar 17

22K 379

Jack Roper in CodeX

Terraform console command

Ever needed a quick and easy way to experiment with Terraform functions and expressions? In this short article, we will explore the...

· 4 min read · Mar 20

70

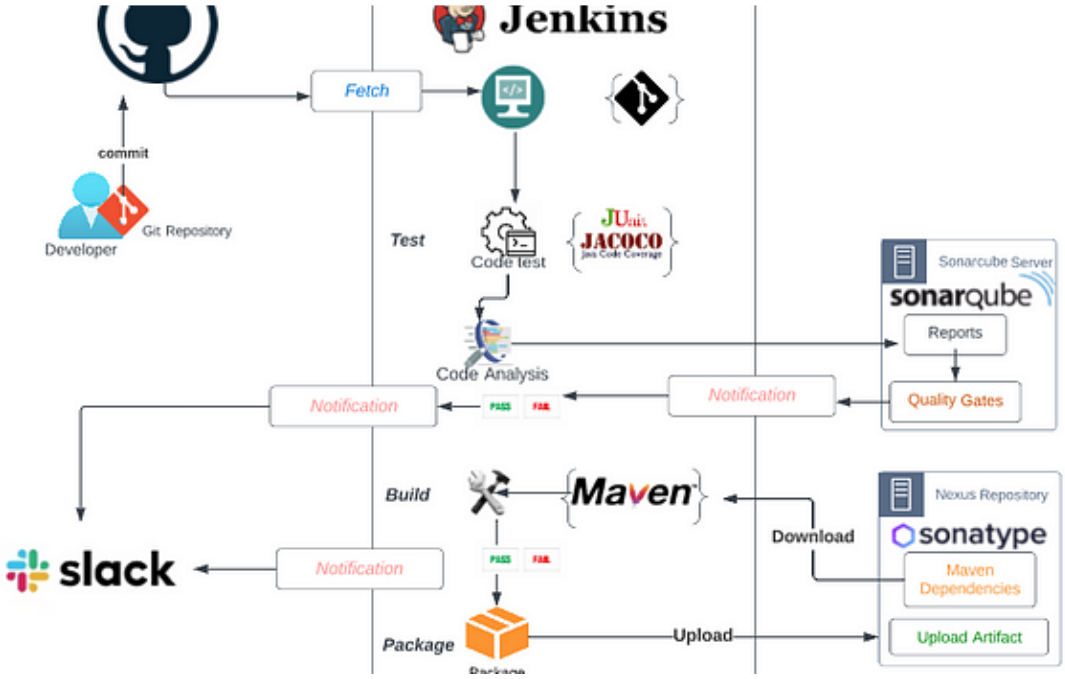
Vickie Li

Hacking LLMs with prompt injections

And ways hackers can attack GPT-based applications

✨ · 8 min read · 4 days ago

345 2



Adewopo Olalekan Amos

DevOps Project — CICD — Continous Integration of Java Web Application

Project Source: DevOps Project by Imran Teli

14 min read · May 11

12

See more recommendations