Project title

AUTOMATED DEPLOYMENT PIPELINE

INTRODUCTION:

The project aims to automate the deployment process of web applications using Jenkins, a popular continuous integration and continuous deployment (CI/CD) tool, and Nginx, a high-performance web server. This documentation outlines the steps required to set up and execute an automated deployment pipeline using Jenkins to deploy on an Nginx server.

Prerequisites:

Before setting up the Automated Deployment Pipeline with jenkins and nginx ensure the following prerequisites are met

Tools:

1.Ec2 instance

2. Jenkins

3.Github

4. Cloud Watch

What is CI/CD?

Continuous Integration/Continuous Deployment (CI/CD) is a software development practice aimed at automating the process of integrating code changes into a shared repository (CI) and rapidly deploying code to production environments (CD). In a single automated deployment pipeline, CI involves automatically triggering builds, running tests, and validating code changes whenever developers push code to a version control system. CD extends CI by automating the deployment process, enabling frequent and reliable delivery of applications to production environments. By automating tasks such as testing, building, and deployment, CI/CD pipelines streamline development workflows, improve code quality, and accelerate time-to-market, fostering collaboration and agility within development teams

There are Few steps to follow to create a CI/CD Pipeline.

1.Launch Ec2 Instance.

2.Install Jenkins

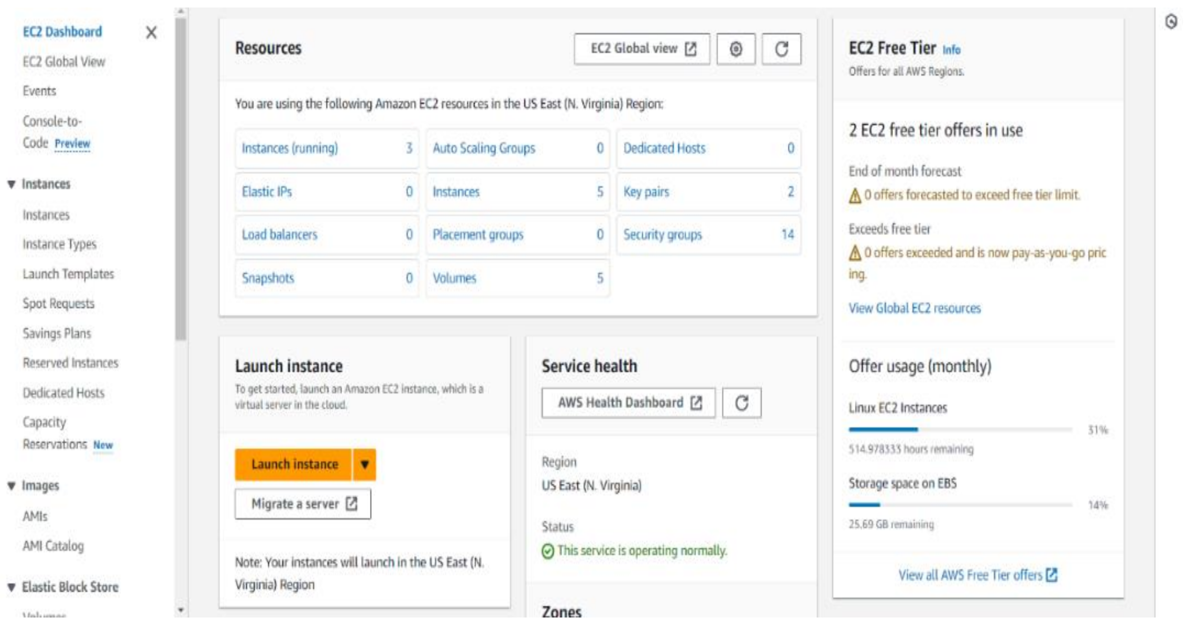
3.Deploying Application.

4.Monitoring Capabilities(cloud Watch)

1.Launch an EC2 Instance:

For creating an EC2 instance.Initially We have to sign in to Aws console .Then

Click on EC2 service After Click on Launch Instance.



Resources

EC2 Global view

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	3	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	5	Key pairs	2
Load balancers	0	Placement groups	0	Security groups	14
Snapshots	0	Volumes	5		

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance

Migrate a server

Note: Your instances will launch in the US East (N. Virginia) Region

Service health

AWS Health Dashboard

Region
US East (N. Virginia)

Status
This service is operating normally.

Zones

EC2 Free Tier Info
Offers for all AWS Regions.

2 EC2 free tier offers in use

End of month forecast
⚠ 0 offers forecasted to exceed free tier limit.

Exceeds free tier
⚠ 0 offers exceeded and is now pay-as-you-go pricing.

View Global EC2 resources

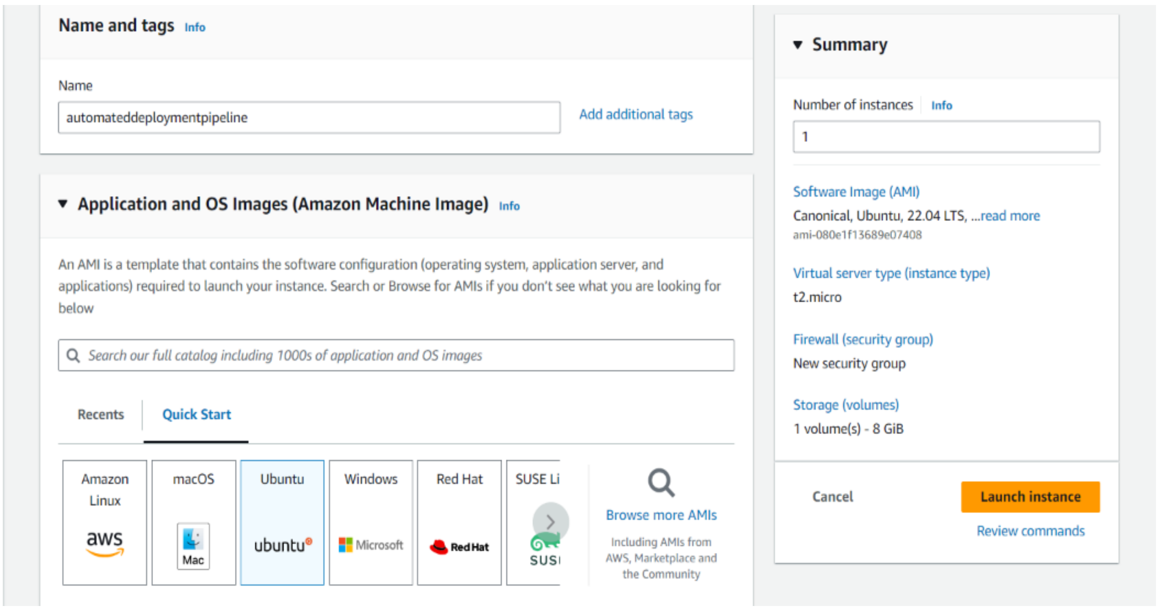
Offer usage (monthly)

Linux EC2 Instances 31%
514.978333 hours remaining

Storage space on EBS 14%
25.69 GB remaining

View all AWS Free Tier offers

Give the Name for instance Automated Deployment pipeline and select the Ubuntu os



Name and tags Info

Name
automateddeploymentpipeline

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Summary

Number of instances Info
1

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-080e1f13689e07408

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

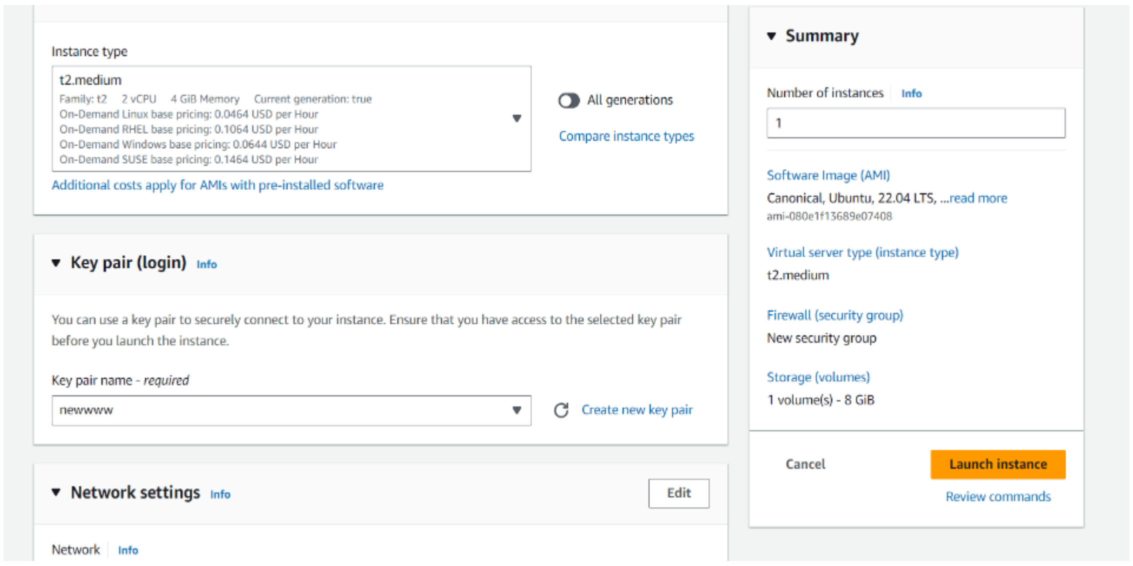
Storage (volumes)
1 volume(s) - 8 GiB

Cancel

Launch instance

Review commands

And Select the instance type As Per Requirement and Select a key pair new or Existing


Instance type

t2.medium
Family: t2 2 vCPU 4 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0464 USD per Hour
On-Demand RHEL base pricing: 0.1064 USD per Hour
On-Demand Windows base pricing: 0.0644 USD per Hour
On-Demand SUSE base pricing: 0.1464 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

newwww

Create new key pair

Network settings Info

Edit

Network Info

Summary

Number of instances Info

1

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-080e1f13689e07408

Virtual server type (instance type)
t2.medium

Firewall (security group)
New security group

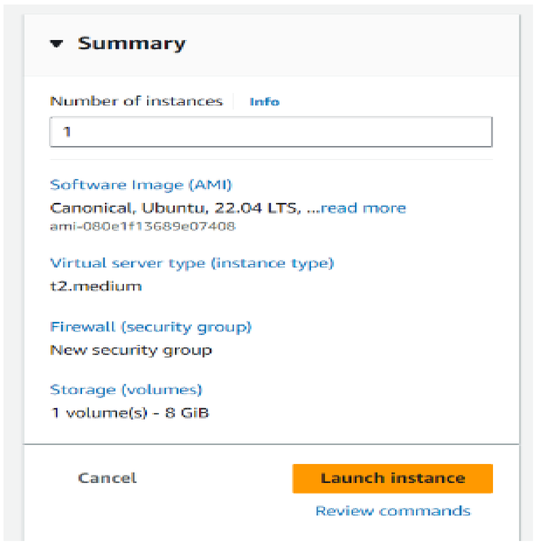
Storage (volumes)
1 volume(s) - 8 GiB

Cancel

Launch instance

Review commands

Click Launch instnace.


Summary

Number of instances Info

1

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-080e1f13689e07408

Virtual server type (instance type)
t2.medium

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

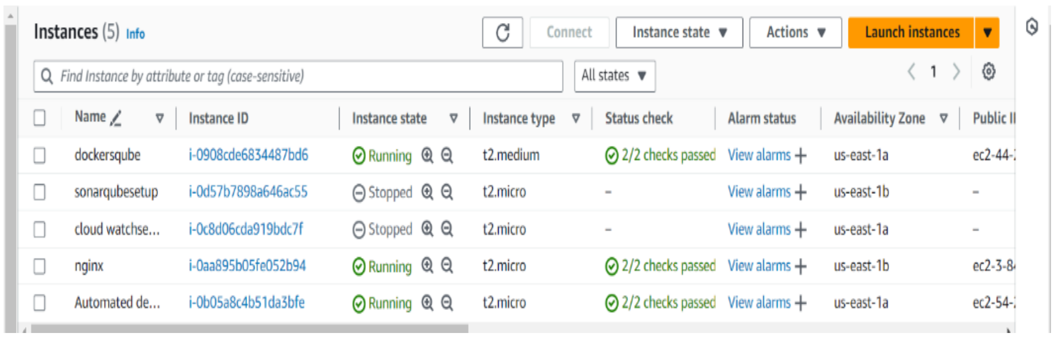
Cancel

Launch instance

Review commands

Sucessfully we created the instnce. Wait for instance state to be as running

State as shown below.

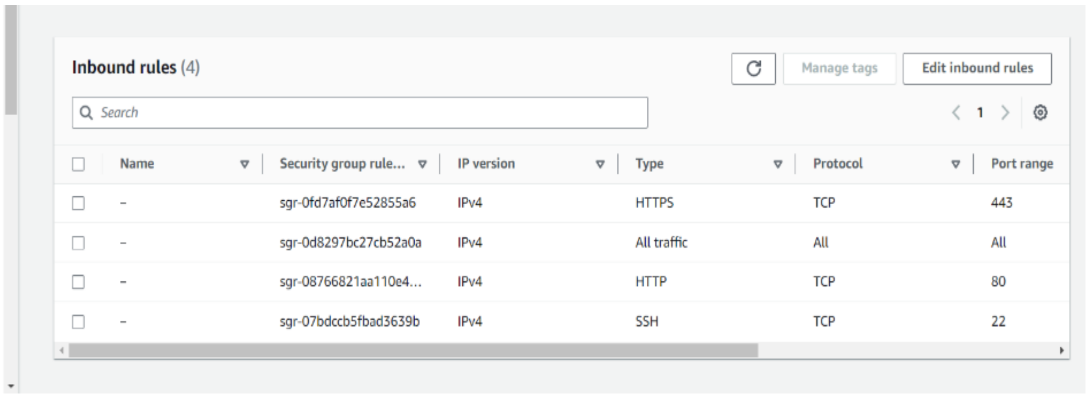

Instances (5) Info

Find Instance by attribute or tag (case-sensitive)

All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public I
☐	dockersqube	i-0908cde6834487bd6	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1a	ec2-44-
☐	sonarqubesetup	i-0d57b7898a646ac55	Stopped	t2.micro	-	View alarms +	us-east-1b	-
☐	cloud watchse...	i-0c8d06cda919bdc7f	Stopped	t2.micro	-	View alarms +	us-east-1a	-
☐	nginx	i-0aa895b05fe052b94	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-3-8
☐	Automated de...	i-0b05a8c4b51da3bfe	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-54-

Now open the Instance and click on security and then security group edit the inbound rules.



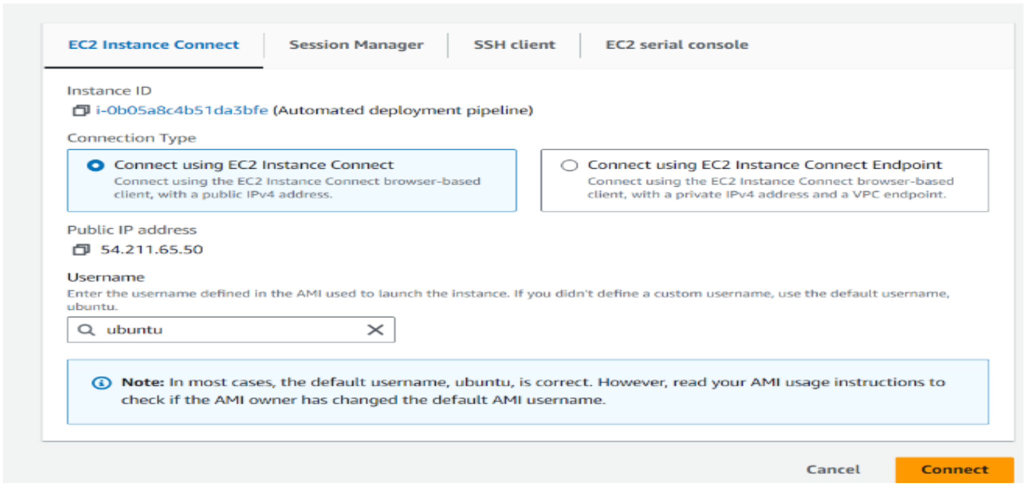
Inbound rules (4)

	Name	Security group rule...	IP version	Type	Protocol	Port range
☐	-	sgr-0fd7af0f7e52855a6	IPv4	HTTPS	TCP	443
☐	-	sgr-0d8297bc27cb52a0a	IPv4	All traffic	All	All
☐	-	sgr-08766821aa110e4...	IPv4	HTTP	TCP	80
☐	-	sgr-07bdccb5fbad3639b	IPv4	SSH	TCP	22

2.Installing Jenkins In EC2:

Connect through ssh or ec2 connect .

After Connecting suucessfully.first we need to login as root user by using command



Instance ID
i-0b05a8c4b51da3bfe (Automated deployment pipeline)

Connection Type
[✓ Connect using EC2 Instance Connect] Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
Connect using EC2 Instance Connect Endpoint — Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address
54.211.65.50

Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.
ubuntu

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

sudo-i .after we need to install java . for installation of java by following the below command.

```
 sudo apt update
sudo apt install openjdk-11-jre
```

After we need to install jenkins follow the below commands.

```
 curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
```

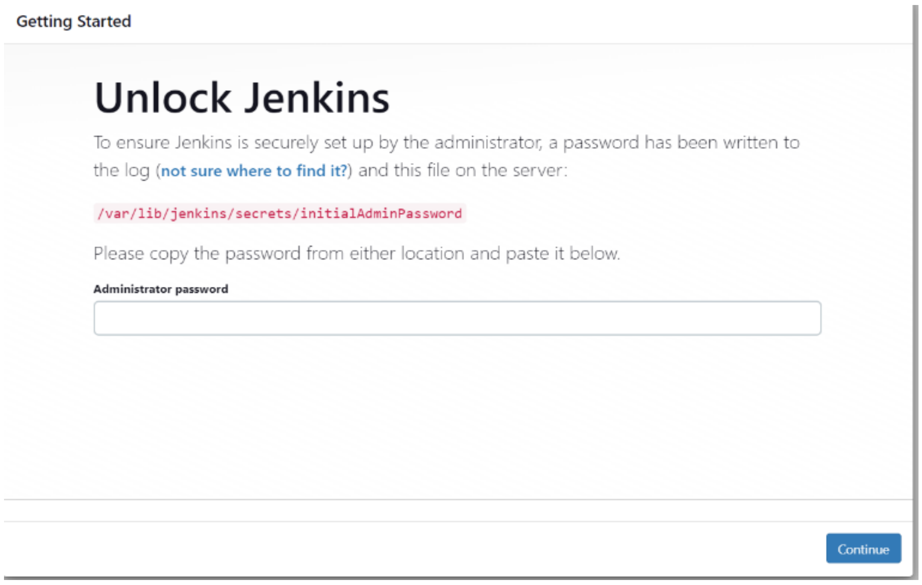
After sucesfully installed the jenkins.

Login to the Jenkins server:

To login to the jenkins server.through port 8080

http://ip-address:8080(You can get the ec2-instance-public-ip-address from your AWS EC2 console page)

Now we will be acess to the jenkins page



Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

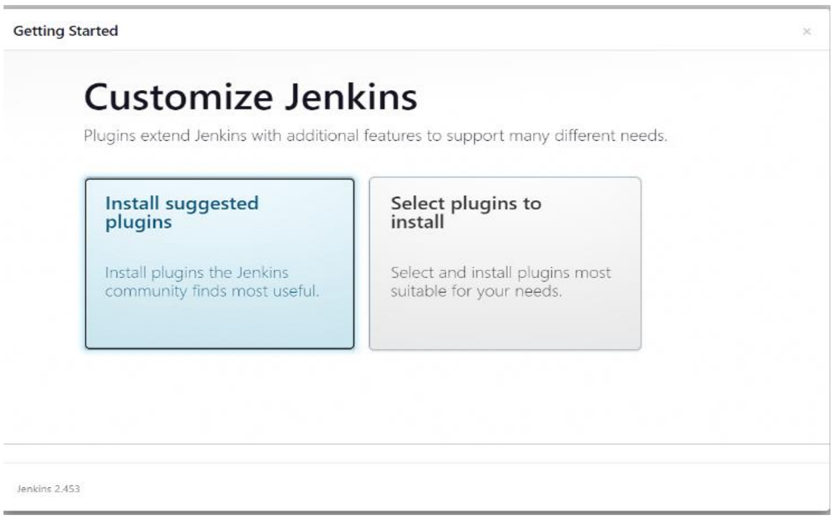
Please copy the password from either location and paste it below.

Administrator password

initially We need to enter the adminstrator password to login.for adminstrator password

sudo cat `/var/lib/jenkins/secrets/initialAdminPassword`

after enter the adminstrator password and install all suggested plugins



Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

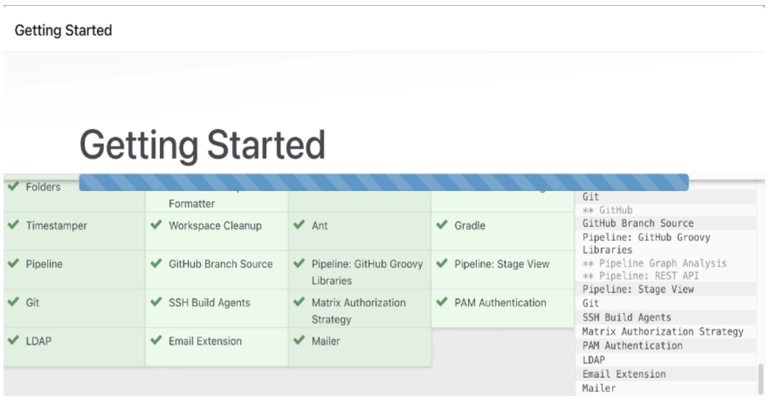
Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.453



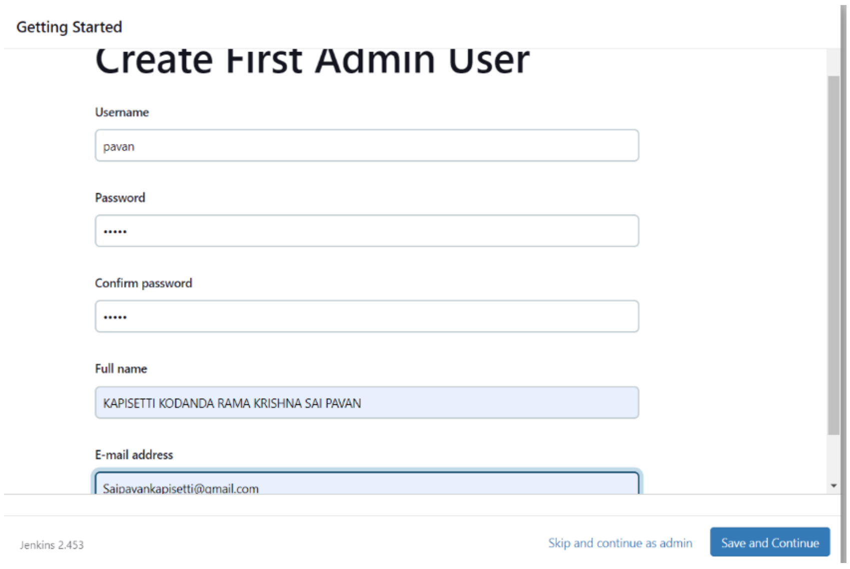
Getting Started

Getting Started

✓ Folders	Formatter		
✓ Timestamper	✓ Workspace Cleanup	✓ Ant	✓ Gradle
✓ Pipeline	✓ GitHub Branch Source	✓ Pipeline: GitHub Groovy Libraries	✓ Pipeline: Stage View
✓ Git	✓ SSH Build Agents	✓ Matrix Authorization Strategy	✓ PAM Authentication
✓ LDAP	✓ Email Extension	✓ Mailer	

```
Git
** GitHub
GitHub Branch Source
Pipeline: GitHub Groovy
Libraries
** Pipeline Graph Analysis
** Pipeline: REST API
Pipeline: Stage View
Git
SSH Build Agents
Matrix Authorization Strategy
PAM Authentication
LDAP
Email Extension
Mailer
```

After installing all the plugins then create your login credentials

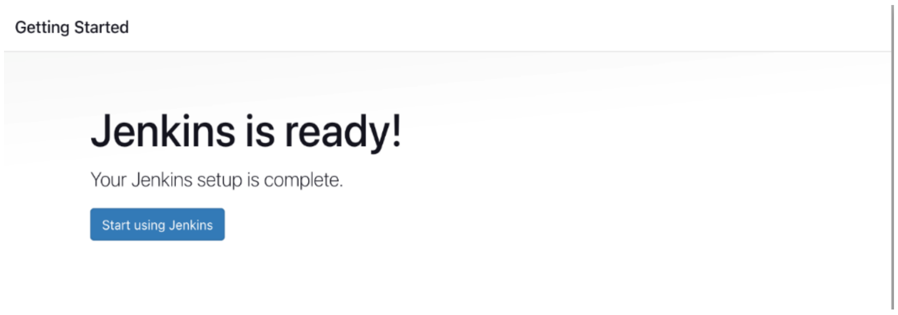


Getting Started

Create First Admin User

Username	pavan
Password	•••••
Confirm password	•••••
Full name	KAPISETTI KODANDA RAMA KRISHNA SAI PAVAN
E-mail address	Saipavankapisetti@gmail.com

Jenkins 2.453

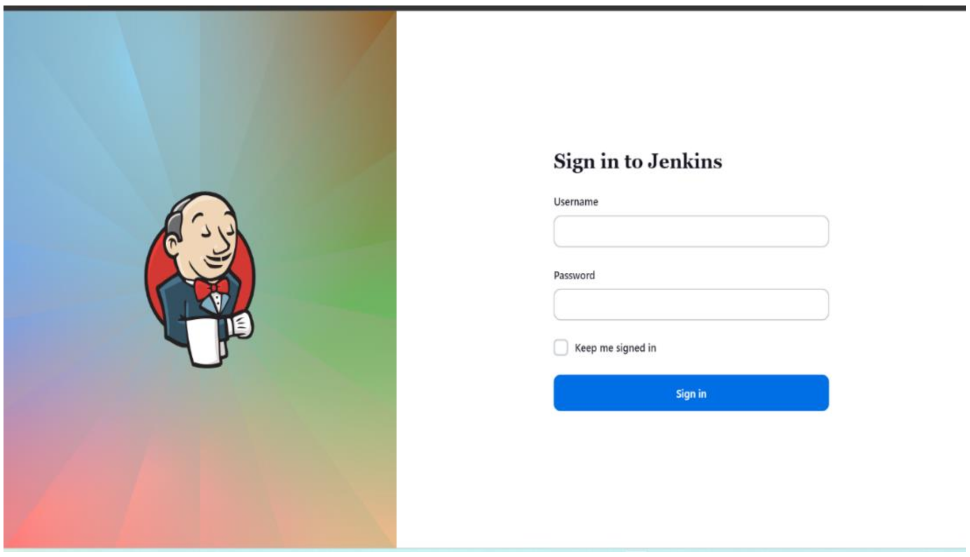


Getting Started

Jenkins is ready!

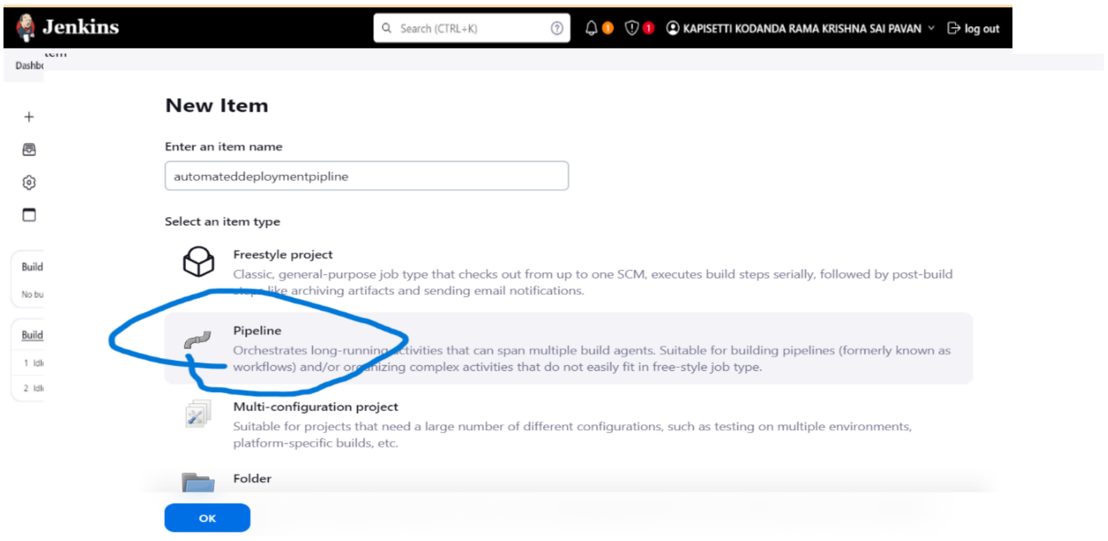
Your Jenkins setup is complete.

Click on start using Jenkins.Then login to your jenkins page



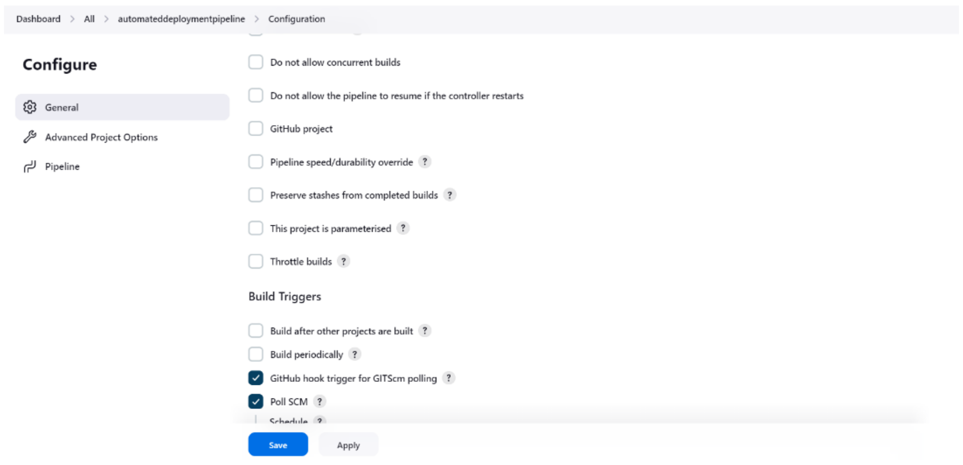
3.Deploying Application:

After login sucessfully .click on new item


Jenkins
Search (CTRL+K)
KAPISETTI KODANDA RAMA KRISHNA SAI PAVAN
log out
Dashb
New Item
Enter an item name
automateddeploymentpipline
Select an item type
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
OK

And then click on pipeline project and give the project name.

After creating sucessfully open a project .click on configrue and Enable Github hook trigger for GITSCM polling.



Dashboard > All > automateddeploymentpipeline > Configuration

Configure

- General
- Advanced Project Options
- Pipeline

- [ ] Do not allow concurrent builds
- [ ] Do not allow the pipeline to resume if the controller restarts
- [ ] GitHub project
- [ ] Pipeline speed/durability override
- [ ] Preserve stashes from completed builds
- [ ] This project is parameterised
- [ ] Throttle builds

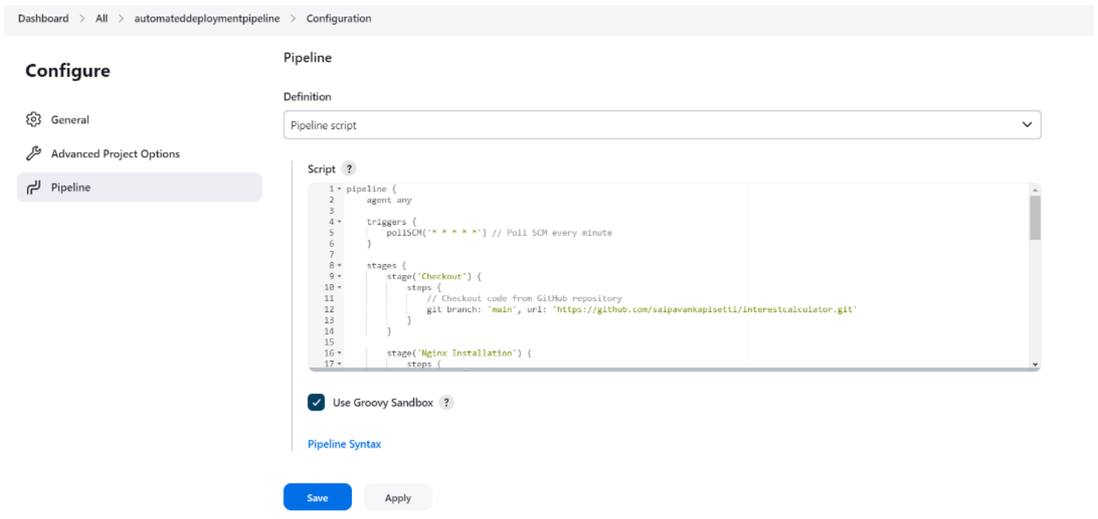
Build Triggers

- [ ] Build after other projects are built
- [ ] Build periodically
- [x] GitHub hook trigger for GITScm polling
- [x] Poll SCM

Schedule

Save Apply

And scroll down to and add the pipeline script.



Dashboard > All > automateddeploymentpipeline > Configuration

Configure

- General
- Advanced Project Options
- Pipeline

Pipeline

Definition

Pipeline script

Script

```
pipeline {
    agent any

    triggers {
        pollSCM('* * * * *') // Poll SCM every minute
    }

    stages {
        stage('Checkout') {
            steps {
                // Checkout code from GitHub repository
                git branch: 'main', url: 'https://github.com/saipavankapisetti/interestcalculator.git'
            }
        }

        stage('Nginx Installation') {
            steps {
```

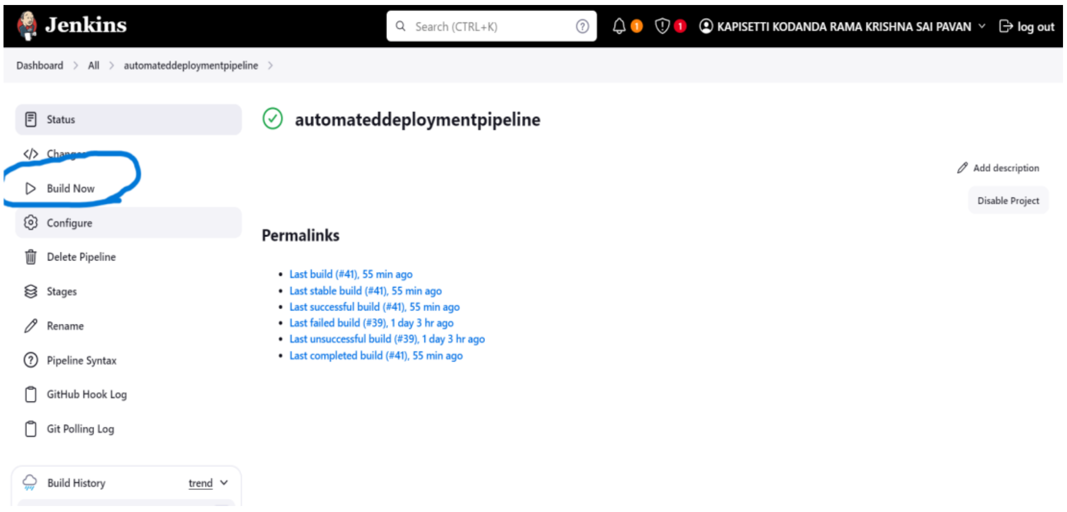
- [x] Use Groovy Sandbox

Pipeline Syntax

Save Apply

after writing a pipeline script and click on apply and save it .

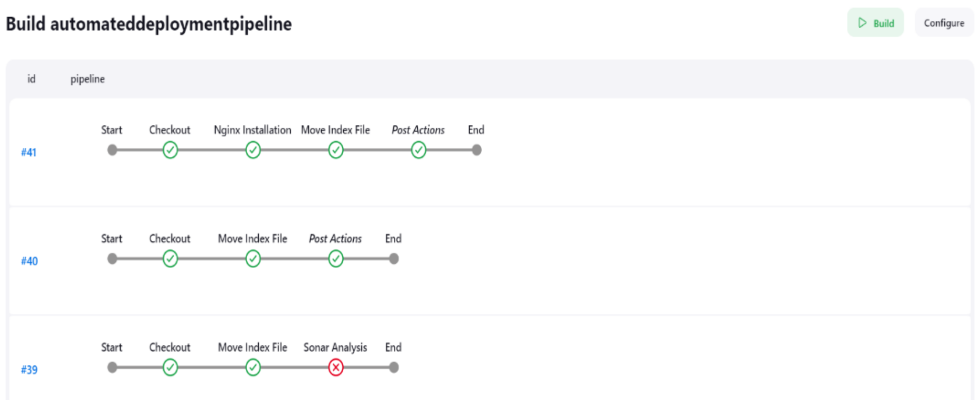
after sucessfully configrue the build steps now click on build now



Permalinks

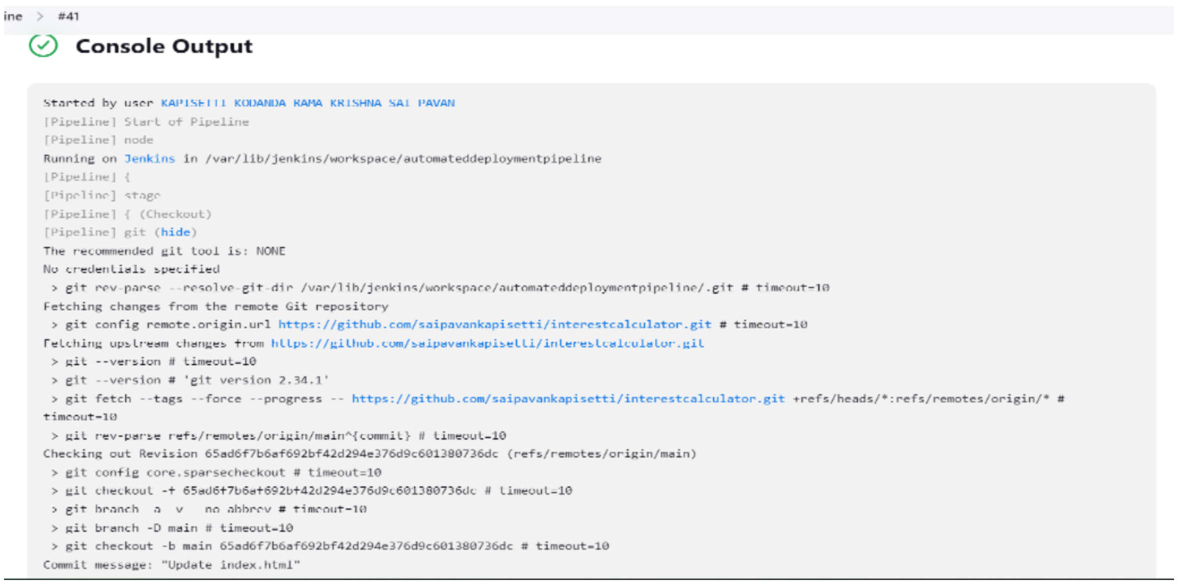
- Last build (#41), 55 min ago
- Last stable build (#41), 55 min ago
- Last successful build (#41), 55 min ago
- Last failed build (#39), 1 day 3 hr ago
- Last unsuccessful build (#39), 1 day 3 hr ago
- Last completed build (#41), 55 min ago

After build sucessfully . View stages



Sucessfully build the pipeline

if pipeline was failed you need to check errors in console output



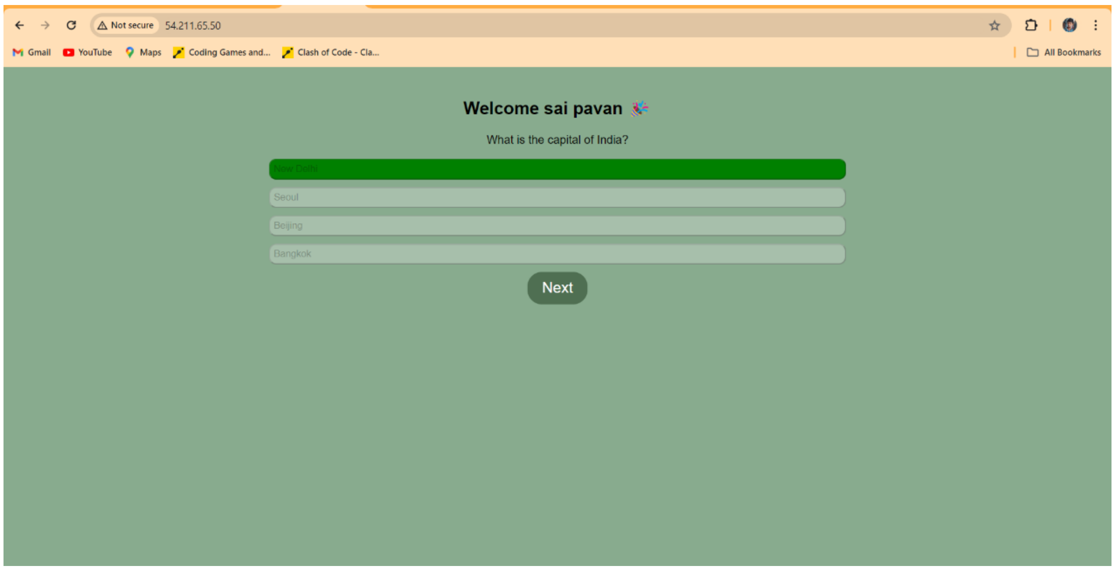
```
Started by user KAPISETTI KODANDA RAMA KRISHNA SAI PAVAN
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/automateddeploymentpipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Checkout)
[Pipeline] git (hide)
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/automateddeploymentpipeline/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/saipavankapisetti/interestcalculator.git # timeout=10
Fetching upstream changes from https://github.com/saipavankapisetti/interestcalculator.git
 > git --version # timeout=10
 > git --version # 'git version 2.34.1'
 > git fetch --tags --force --progress -- https://github.com/saipavankapisetti/interestcalculator.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 65ad6f7b6af692bf42d294e376d9c601380736dc (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 65ad6f7b6af692bf42d294e376d9c601380736dc # timeout=10
 > git branch -a -v --no-abbrev # timeout=10
 > git branch -D main # timeout=10
 > git checkout -b main 65ad6f7b6af692bf42d294e376d9c601380736dc # timeout=10
Commit message: "Update index.html"
```

The pipeline Script used in my project:

```
 pipeline {
agent any
triggers {
    pollSCM('* * * * *')
}

stages {
    stage('Checkout') {
        steps {
            git branch: 'main', url:
'https://github.com/saipavankapisetti/interestcalculator.git'
        }
    }

    stage('Nginx Installation') {
        steps {
            script {
                try {
                    sh 'sudo apt update && sudo apt install nginx -y'
                    sh 'sudo systemctl start nginx'
                    sh 'sudo systemctl status nginx'
                } catch (Exception e) {
                    currentBuild.result = 'FAILED'
                    error "Failed to install and start Nginx: ${e.message}"
                }
            }
            }
        }

        stage('Move Index File') {
            steps {
                script {
                    try {
                        sh 'sudo mv index.html /var/www/html/index.nginx-debian.html'
                        currentBuild.result = 'SUCCESS'
                    } catch (Exception e) {
                        echo "Failed to move index.html: ${e.message}"
                        currentBuild.result = 'FAILURE'
                        error "Failed to move index.html"
                    }
                }
            }
        }
    }

    post {
        success {
            echo "Pipeline completed successfully!"
        }
        failure {
            echo "Pipeline failed!"
        }
    }

}
```

Our application was sucessfully deployed. For acess the application

Go to the instance and copy the public ipv4 address and open in a browser to verify the output.



Welcome sai pavan 🎉

What is the capital of India?

New Delhi

Seoul

Beijing

Bangkok

Sucessfully run the application.

4.Monitoring Capabilities:

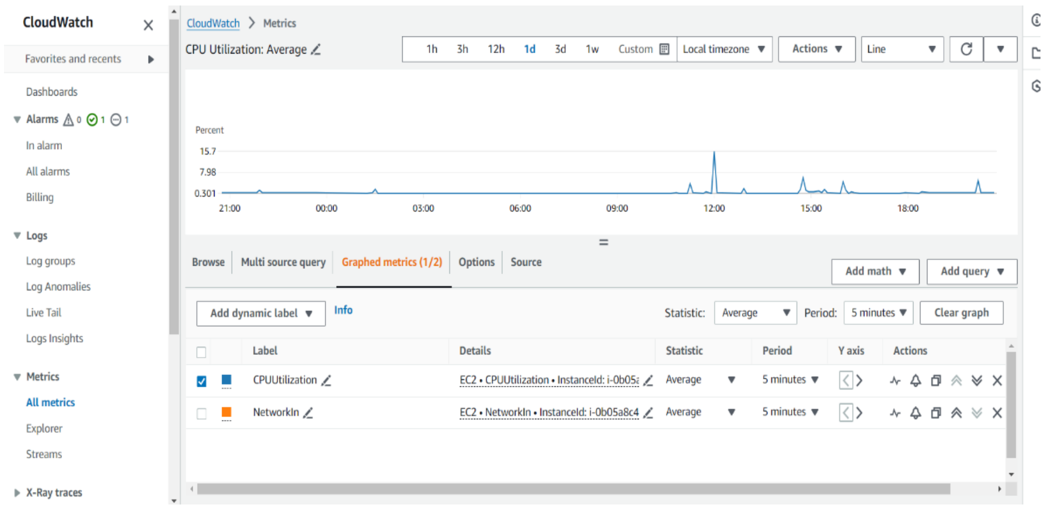
For monitoring cpu utilization and RAM utilization using cloud watch

Open the cloud watch and click on metrics section and monitor the graphs about cpu utilization ,RAM utilization and network in etc.

Create a Alarm for required

In my case I create an alarm for when cpu utilization is threshold value is greater than 200 percent.and then alarm will be activated.

Cpu utilization Graphs are shown below



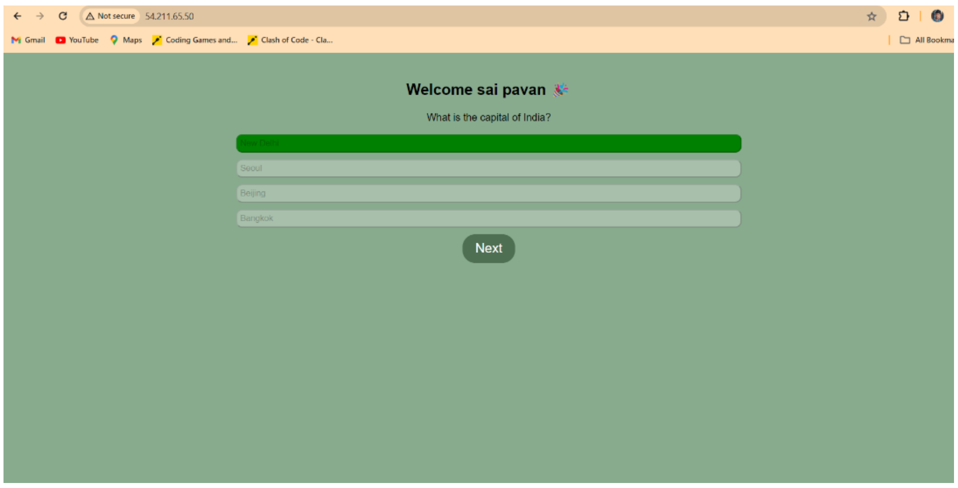
CPU Utilization: Average

Statistic: Average Period: 5 minutes

		Label	Details	Statistic	Period	Y axis	Actions
✓	🟦	CPUUtilization	EC2 • CPUUtilization • InstanceId: i-0b05a	Average	5 minutes		
	🟧	NetworkIn	EC2 • NetworkIn • InstanceId: i-0b05a8c4	Average	5 minutes		

OUTPUT:

Here is the final output.



Welcome sai pavan 🎉

What is the capital of India?

New Delhi

Seoul

Beijing

Bangkok

The application deployed on nginx server using Ec2 instance ,it contains

Html,Css,Javascript