

Deploy the application

Step1: Create a EC2 Instance

I. login to aws account

II. Search for EC2 in search bar

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with options like EC2 Global View, Events, Tags, Limits, Instances (selected), Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, and Images. The main area displays EC2 resources: Instances (running) 0, Dedicated Hosts 0, Elastic IPs 0, Instances 0, Key pairs 0, Load balancers 0, Placement groups 0, Security groups 1, Snapshots 0, and Volumes 0. Below this, a callout box suggests using the AWS Launch Wizard for Microsoft SQL Server Always On availability groups. To the right, there's an 'Account attributes' panel with sections for Supported platforms (VPC), Default VPC (vpc-036e0f2e88e302ecc), Settings (EBS encryption, Zones, EC2 Serial Console, Default credit specification, Console experiments), and an 'Explore AWS' section featuring Amazon GuardDuty Malware Protection and a 40% better price performance offer.

Step2: Click on instances in the side menu

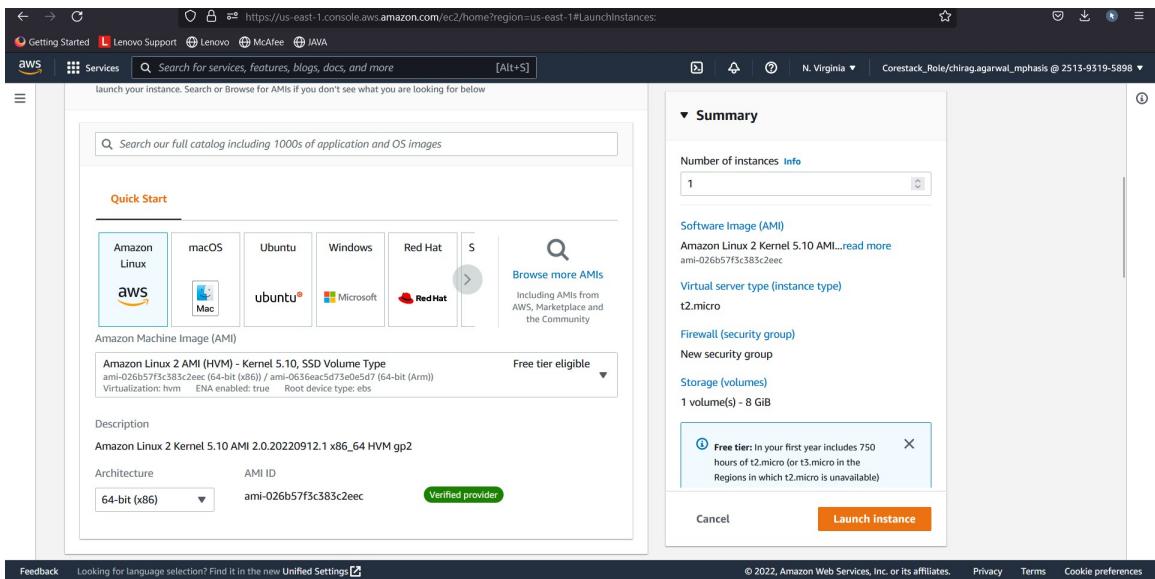
Click on launch instances and create a new instance.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with sections like 'EC2 Dashboard', 'Events', 'Tags', 'Limits', 'Instances' (selected), 'Images', and 'Feedback'. The main content area has a heading 'Instances Info' with a search bar and a table header for 'Name', 'Instance ID', 'Instance state', 'Instance type', 'Status check', 'Alarm status', 'Availability Zone', and 'Public IPv4 DNS'. Below the table, it says 'No instances' and 'You do not have any instances in this region'. A 'Launch instances' button is at the bottom. A modal window titled 'Select an instance' is open.

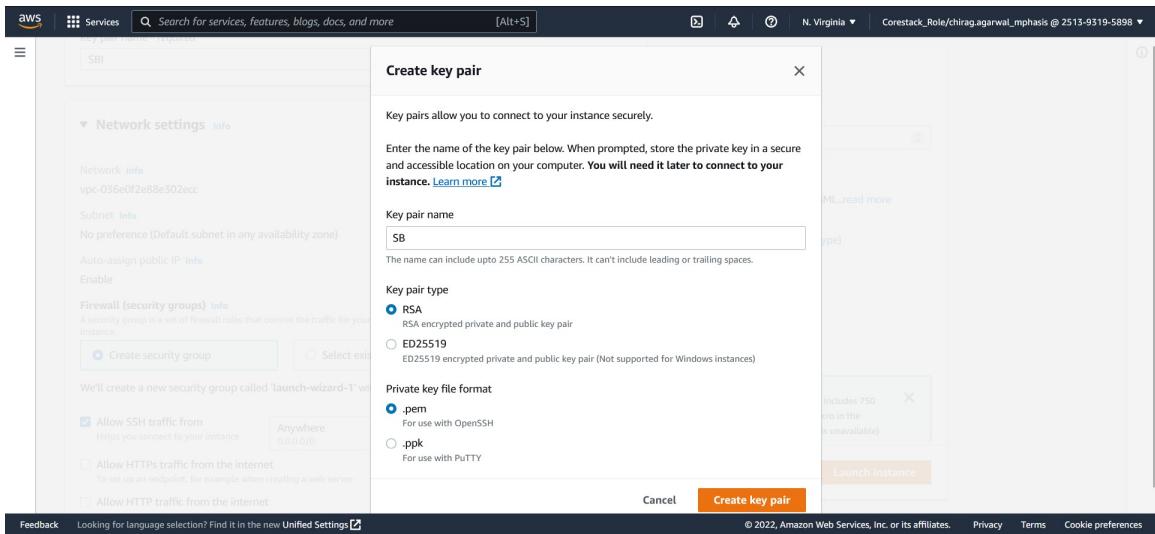
Step3: Enter the name of your instance which must be unique

The screenshot shows the 'Launch an instance' page. It has a summary section with 'Number of instances' set to 1, 'Software Image (AMI)' set to 'Amazon Linux 2 Kernel 5.10 AMI...', 'Virtual server type (instance type)' set to 't2.micro', and 'Storage (volumes)' set to '1 volume(s) - 8 GiB'. A note about the free tier is displayed. The 'Name and tags' section shows 'Name' as 'SpringBootInstance'. The 'Application and OS Images (Amazon Machine Image)' section has a search bar. A message at the top says: 'You've been opted into the new launch experience. You can return to the previous version, but next time you log in, you'll automatically be opted into the new experience.' with a link to 'Opt out to the old experience'.

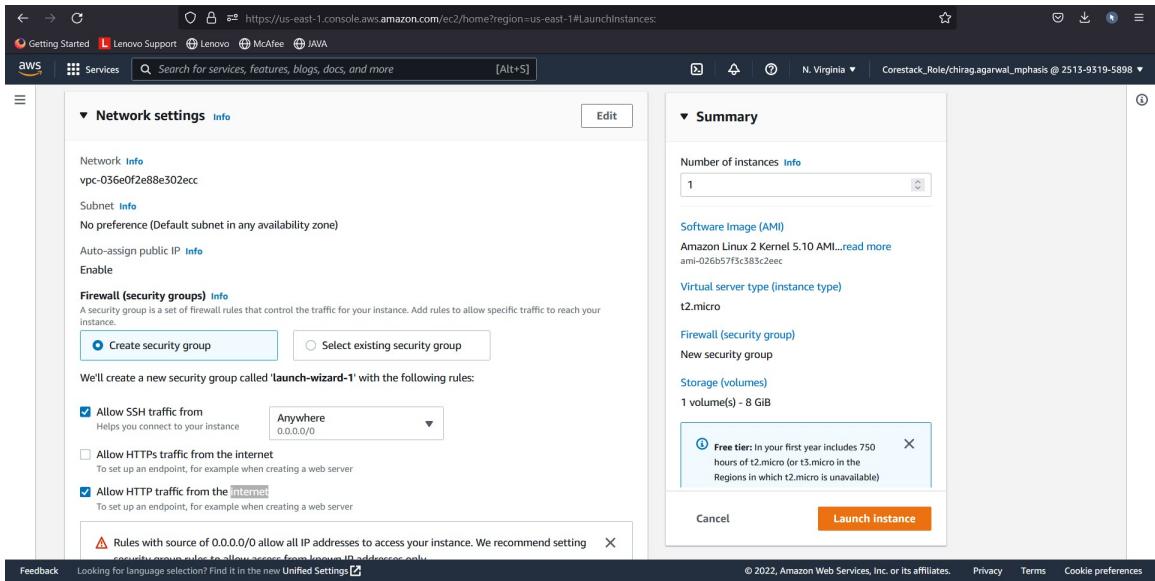
Step4: Select Os image as amazon Linux



Step5: Create a new key pair and download the .pem file



Step6: In network setting check on allow ssh from anywhere and allow http traffic from internet



Step7: In advance setting - user data enter the script to install the java 17

```
#!/bin/bash

//for installing java 17

sudo yum install java-17-amazon-corretto-headless

//for making directory app

mkdir app

//for going to director app

cd app

and click on launch instance and wait for few min come
instance in running state
```

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, and AMI Catalog. The main content area displays a table titled 'Instances (1) Info' with one row. The row details a SpringBoot instance with ID i-006a666badeb45e24, which is currently 'Running' on an 't2.micro' instance type, 'Initializing' status, and located in 'us-east-1c' availability zone with a public IPv4 DNS of 'ec2-3-84-55-57.con'. Below the table is a modal window titled 'Select an instance'.

Step8: Create a role in IAM
and search IAM in search bar
and click on create User

The screenshot shows the AWS IAM Roles page. The left sidebar includes options for Identity and Access Management (IAM), Access management (User groups, Users, Roles, Policies, Identity providers, Account settings), Access reports (Archive rules, Analyzers, Settings, Credential report, Organization activity), and a feedback link. The main content area shows a table titled 'Roles (14) Info' with 14 entries. Each entry includes the role name, trusted entity, and last activity date. The roles listed are: AWSServiceRoleForAmazonElasticFileSystem, AWSServiceRoleForAutoScaling, AWSServiceRoleForCloudTrail, AWSServiceRoleForConfig, AWSServiceRoleForDAX, AWSServiceRoleForECS, AWSServiceRoleForElasticLoadBalancing, and AWSServiceRoleForGlobalAccelerator.

Step9: Select Trusted entity as AWS service

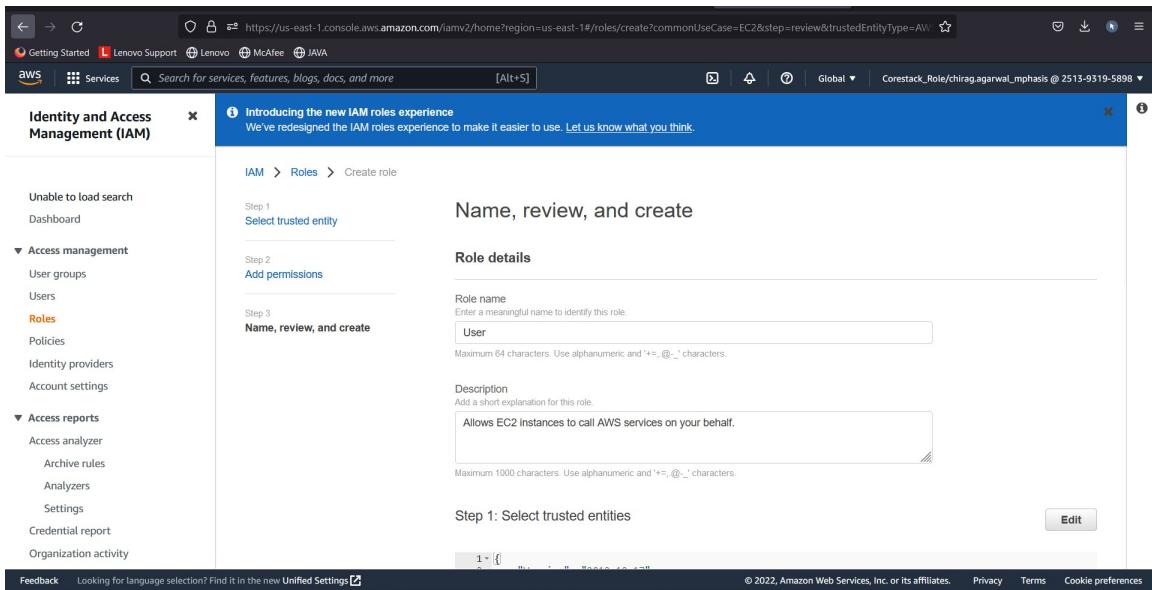
select use case as EC2 under common use cases

The screenshot shows the AWS IAM Role Creation process at Step 1: Select trusted entity. On the left sidebar, under the 'Identity and Access Management (IAM)' section, the 'Roles' option is selected. In the main content area, the heading 'Select trusted entity' is displayed. Below it, the 'Trusted entity type' section contains five options: 'AWS service' (selected), 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. Under the 'Use case' section, 'EC2' is selected. At the bottom, there are sections for 'Common use cases' (with 'EC2' selected) and 'Use cases for other AWS services'.

Step10: Choose permissions policy as AmazonS3ReadOnlyAccess

The screenshot shows the AWS IAM Role Creation process at Step 2: Add permissions. The left sidebar remains the same as the previous step. In the main content area, the heading 'Add permissions' is displayed. The 'Permissions policies' section shows one policy selected: 'AmazonS3ReadOnlyAccess'. A note below states: 'Set a permissions boundary - optional' and 'Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.' At the bottom, there are links for 'Feedback', 'Privacy', 'Terms', and 'Cookie preferences'.

Step11: Enter the role name and create click role



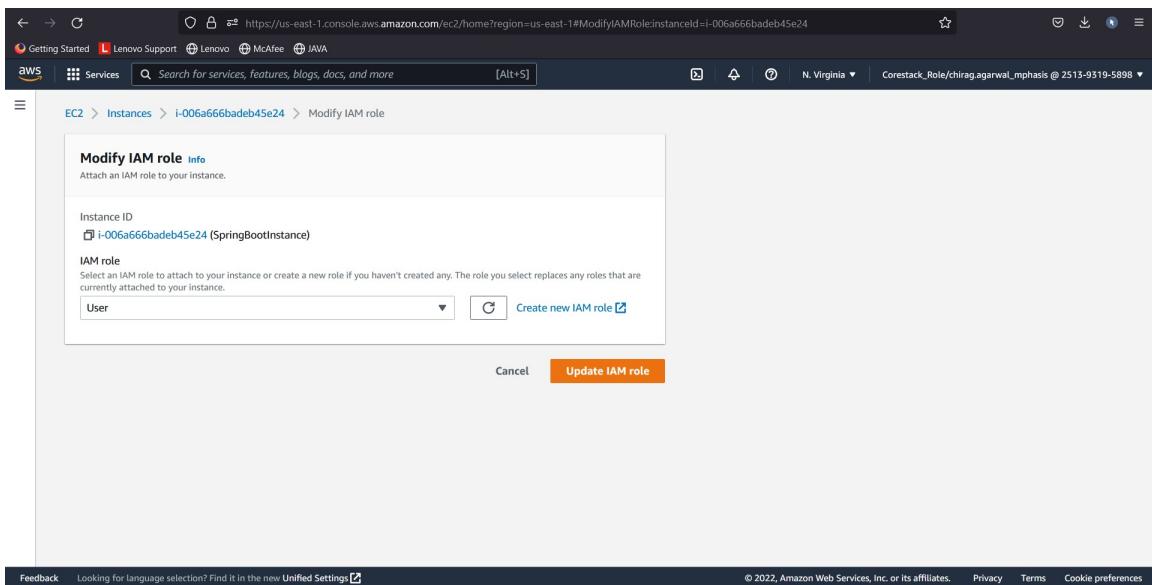
Step12: Add the IAM role to the ec2 instance

select the instance

click on actions

click on security

click on modify IAM role and choose the role we created
and click on update IAM role



Step13: Now after adding role to EC2 instance now we need to create S3 bucket

Search for S3 in search box and create bucket by entering the name and click on create bucket.

Bucket name: myawsbucket
AWS Region: US East (N. Virginia) us-east-1
Choose bucket

Object Ownership: Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

Step14: Wait for the S3 bucket to be created

Successfully created bucket "springbootbucket23"
To upload files and folders, or to configure additional bucket settings choose [View details](#).

Buckets (1) Info
Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date
springbootbucket23	US East (N. Virginia) us-east-1	Objects can be public	October 5, 2022, 23:54:09 (UTC+05:30)

Step15: Now click on the bucket and upload the build of spring boot app from target/*.jar

The screenshot shows the AWS S3 console interface. The user is uploading a file named 'demo-0.0.1-SNAPSHOT.jar' to a bucket called 'springbootbucket123'. The file size is 8.4 MB. The destination is specified as 's3://springbootbucket123'. The interface includes a search bar, navigation links, and standard browser controls.

Step16: Now connect to EC2 instance and create a directory and go to directory

```

Verifying : libxslt-1.1.28-6.amzn2.x86_64
Verifying : dejavu-sans-fonts-2.33-6.amzn2.noarch
Verifying : dejavu-fonts-common-headless-17.0.4+9-1.amzn2.1.x86_64
Verifying : python-lxml-3.2.1-4.amzn2.0.3.x86_64
Verifying : fontconfig-2.13.0-4.3.amzn2.x86_64
Verifying : python-javapackages-3.4.1-11.amzn2.noarch
Verifying : fontpackages-filesystem-1.44-8.amzn2.noarch
Verifying : dejavu-fonts-common-2.33-6.amzn2.noarch
Verifying : javapackages-tools-3.4.1-11.amzn2.noarch
2/10
3/10
4/10
5/10
6/10
7/10
8/10
9/10
10/10

Installed:
java-17-amazon-corretto-headless.x86_64 1:17.0.4+9-1.amzn2.1

Dependency Installed:
dejavu-fonts-common.noarch 0:2.33-6.amzn2
fontpackages-filesystem.noarch 0:1.44-8.amzn2
log4j-cve-2021-44228-hotpatch.noarch 0:1.3-7.amzn2
dejavu-sans-fonts.noarch 0:2.33-6.amzn2
javapackages-tools.noarch 0:3.4.1-11.amzn2
python-javapackages.noarch 0:3.4.1-11.amzn2
fontconfig.x86_64 0:2.13.0-4.3.amzn2
libxslt.x86_64 0:1.1.28-6.amzn2
python-lxml.x86_64 0:3.2.1-4.amzn2.0.3

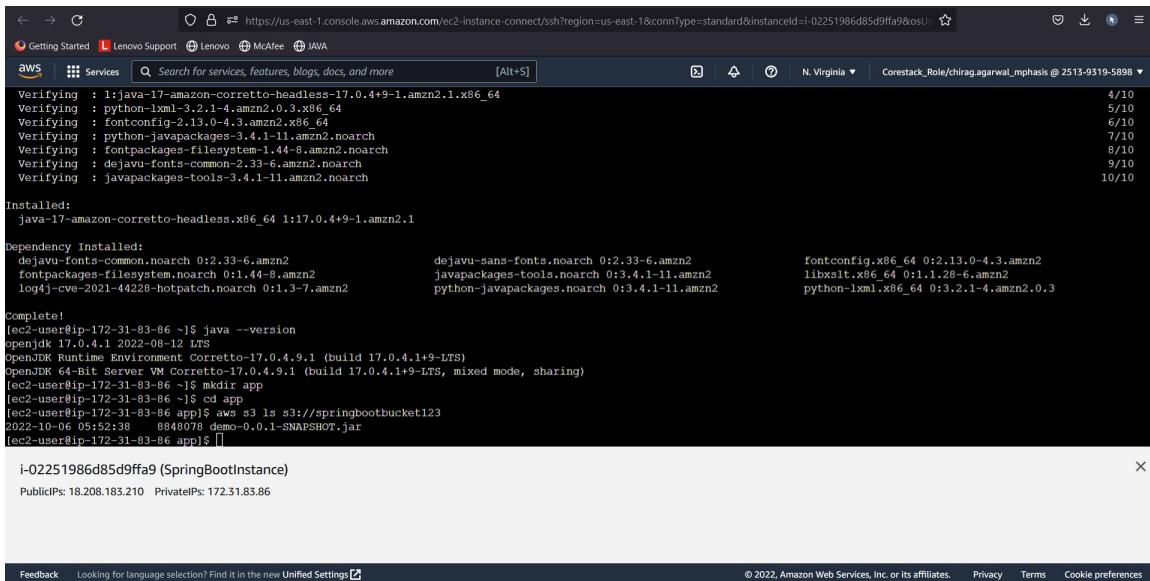
Complete!
[ec2-user@ip-172-31-83-86 ~]$ java --version
openjdk 17.0.4.1 2022-08-12 LTS
OpenJDK Runtime Environment Corretto-17.0.4.9.1 (build 17.0.4.1+9-LTS)
OpenJDK 64-Bit Server VM Corretto-17.0.4.9.1 (build 17.0.4.1+9-LTS, mixed mode, sharing)
[ec2-user@ip-172-31-83-86 ~]$ mkdir app
[ec2-user@ip-172-31-83-86 ~]$ cd app
[ec2-user@ip-172-31-83-86 app]$ 
```

i-02251986d85d9ffa9 (SpringBootInstance)
PublicIPs: 18.208.183.210 PrivateIPs: 172.31.83.86

Step17: List the files of bucket using command

aws s3 ls s3://<name of bucket>

in my case "aws s3 ls s3://springbootbucket123"



```
Verifying : 1:java-17-amazon-corretto-headless-17.0.4+9-1.amzn2.1.x86_64 4/10
Verifying : 1:python-lxml-3.2.1-4.amzn2.0.3.x86_64 5/10
Verifying : fontconfig-3.13.0-4.3.amzn2.x86_64 6/10
Verifying : python-javapackages-3.4.1-11.amzn2.noarch 7/10
Verifying : fontpackages-filesystem-1.44-8.amzn2.noarch 8/10
Verifying : dejavu-fonts-common-2.33-6.amzn2.noarch 9/10
Verifying : javapackages-tools-3.4.1-11.amzn2.noarch 10/10

Installed:
  java-17-amazon-corretto-headless.x86_64 1:17.0.4+9-1.amzn2.1

Dependency Installed:
  dejavu-fonts-common.noarch 0:2.33-6.amzn2          dejavu-sans-fonts.noarch 0:2.33-6.amzn2          fontconfig,x86_64 0:2.13.0-4.3.amzn2
  fontpackages-filesystem.noarch 0:1.44-8.amzn2      javapackages-tools.noarch 0:3.4.1-11.amzn2      libxslt.x86_64 0:1.1.28-6.amzn2
  log4j-cve-2021-44229-hotpatch.noarch 0:1.3-7.amzn2  python-javapackages.noarch 0:3.4.1-11.amzn2      python-javapackages-tools-3.4.1-11.amzn2.0.3

Complete!
[ec2-user@ip-172-31-83-86 ~]$ java --version
openjdk 17.0.4.1 2022-09-12 LTS
OpenJDK Runtime Environment Corretto-17.0.4.9.1 (build 17.0.4.1+9-LTS)
OpenJDK 64-Bit Server VM Corretto-17.0.4.9.1 (build 17.0.4.1+9-LTS, mixed mode, sharing)
[ec2-user@ip-172-31-83-86 ~]$ mkdir app
[ec2-user@ip-172-31-83-86 ~]$ cd app
[ec2-user@ip-172-31-83-86 app]$ aws s3 ls s3://springbootbucket123
2022-10-06 05:52:38    8848078 demo-0.0.1-SNAPSHOT.jar
[ec2-user@ip-172-31-83-86 app]$ [REDACTED]

i-02251986d85d9ffa9 (SpringBootInstance)
PublicIPs: 18.208.183.210 PrivateIPs: 172.31.83.86
```

Step18: Download the snapshot.jar from s3 to ec2

aws s3 cp s3://springbootbucket123/demo-0.0.1-SNAPSHOT.jar app.jar

```
[ec2-user@ip-172-31-83-86 app]$ aws s3 cp demo-0.0.1-SNAPSHOT.jar s3://springbootbucket123
The user-provided path demo-0.0.1-SNAPSHOT.jar does not exist.
[ec2-user@ip-172-31-83-86 app]$ aws s3 cp /demo-0.0.1-SNAPSHOT.jar s3://springbootbucket123
The user-provided path /demo-0.0.1-SNAPSHOT.jar does not exist.
[ec2-user@ip-172-31-83-86 app]$ aws s3 cp demo-0.0.1-SNAPSHOT.jar s3://springbootbucket123
The user-provided path demo-0.0.1-SNAPSHOT.jar does not exist.
[ec2-user@ip-172-31-83-86 app]$ aws s3 cp help
[ec2-user@ip-172-31-83-86 app]$ aws s3 cp demo-0.0.1-SNAPSHOT.jar s3://springbootbucket123^C
[ec2-user@ip-172-31-83-86 app]$ aws s3 cp demo-0.0.1-SNAPSHOT.jar s3://springbootbucket123/demo-0.0.1-SNAPSHOT.jar
The user-provided path demo-0.0.1-SNAPSHOT.jar does not exist.
[ec2-user@ip-172-31-83-86 app]$ aws s3 cp demo-0.0.1-SNAPSHOT.jar s3://springbootbucket123/demo-0.0.1-SNAPSHOT.jar
The user-provided path demo-0.0.1-SNAPSHOT.jar does not exist.
[ec2-user@ip-172-31-83-86 app]$ aws s3 ls s3://springbootbucket123
2022-10-06 05:52:38 8848078 demo-0.0.1-SNAPSHOT.jar
[ec2-user@ip-172-31-83-86 app]$ ^C
[ec2-user@ip-172-31-83-86 app]$ ^C
[ec2-user@ip-172-31-83-86 app]$ aws s3 cp s3://springbootbucket123/demo-0.0.1-SNAPSHOT.jar app.jar
download: s3://springbootbucket123/demo-0.0.1-SNAPSHOT.jar to ./app.jar
[ec2-user@ip-172-31-83-86 app]$ ^C
[ec2-user@ip-172-31-83-86 app]$ ^C
[ec2-user@ip-172-31-83-86 app]$ ^C
[ec2-user@ip-172-31-83-86 app]$ ^C
```

i-02251986d85d9ffa9 (SpringBootInstance)
PublicIPs: 18.208.183.210 PrivateIPs: 172.31.83.86

Step19: Run the jar file using java -jar app.jar

```
[ec2-user@ip-172-31-83-86 app]$ java -jar app.jar
. . .
:: Spring Boot ::

2022-10-06 06:27:08.044  INFO 12129 --- [           main] com.example.demo.DemoApplication      : Starting DemoApplication v0.0.1-SNAPSHOT using Java 17.0.4.1 on ip-172-31-83-86.ec2.internal with PID 12129 (/home/ec2-user/app.jar started by ec2-user in /home/ec2-user/app)
2022-10-06 06:27:08.051  INFO 12129 --- [           main] com.example.demo.DemoApplication      : No active profile set, falling back to 1 default profile: "default"
2022-10-06 06:27:09.443  INFO 12129 --- [           main] com.example.demo.DemoApplication      : Started DemoApplication in 2.397 seconds (JVM running for 3.314)
```

i-02251986d85d9ffa9 (SpringBootInstance)
PublicIPs: 18.208.183.210 PrivateIPs: 172.31.83.86

Step20: Go to browser and go to ipaddress:port/controllername

in my case

<http://18.208.183.210:8080/add?operand1=34&operand2=42>

