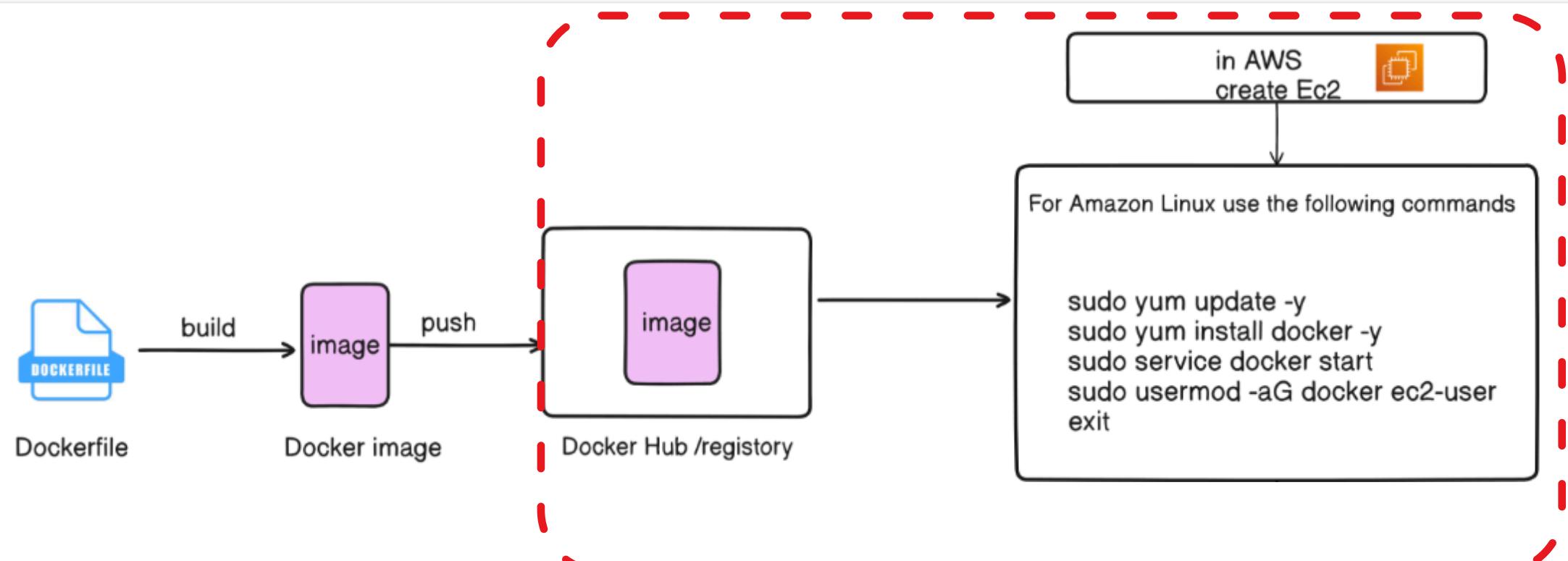




As per Architecture 1st i will do/start from Docker Hub/registry

here we are doing part 2 then will do part 1



Installing Docker:

For Amazon Linux use the following commands

Install Docker In Amazon Linux VM

Step 1

```
sudo yum update -y  
sudo yum install docker -y  
sudo service docker start  
sudo usermod -aG docker ec2-user  
exit
```

Step 1:

```
ec2-user@ip-172-31-2-140 ~]$ sudo yum update -y  
do yum install docker -y  
do service docker start  
do usermod -aG docker ec2-user  
dit  
amazon Linux 2023 Kernel Livepatch repository          130 kB/s | 14 kB   00:00
```

Step 2:

Use this command to check version of docker installed

```
docker -v
```

Step 2:

```
Admin@DESKTOP-9DFQ51N MINGW64 /g/docker-keys
$ ssh -i "docker_keys.pem" ec2-user@ec2-13-201-223-5.ap-south-1.compute.amazonaws.com
      #_
      ~\ _####_          Amazon Linux 2023
      ~~ \####\ I
      ~~ \##| I
      ~~ \#/ , __ I
      ~~ V~ , '-'> https://aws.amazon.com/linux/amazon-linux-2023
      ~~ .-. / \
      _/m/ ' / \
Last login: Wed Mar 26 14:01:17 2025 from 27.7.148.10
[ec2-user@ip-172-31-2-140 ~]$ docker -v
Docker version 25.0.8, build Obab007
[ec2-user@ip-172-31-2-140 ~]$ |
```

Step 3:

For practise pull the sample image from docker hub repository of pankaj sir academy: `docker pull psait/pankajsiracademy:latest`

For Practise pull docker official image: `docker pull hello-world`

Step 3:

```
[ec2-user@ip-172-31-2-140 ~]$ docker pull psait/pankajsiracademy:latest
latest: Pulling from psait/pankajsiracademy
38a980f2cc8a: Pulling fs layer
de849f1cfbe6: Pulling fs layer
a7203ca35e75: Pulling fs layer
1d80cce9a333: Waiting
4f4fb700ef54: Waiting
```

Step 4:

To check how many images are present in docker

docker images

Step 4:

```
ec2-user@ip-172-31-2-140 ~]$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
saif/pankajsiracademy  latest   f434d835074b  3 days ago  492MB
[ec2-user@ip-172-31-2-140 ~]$
```

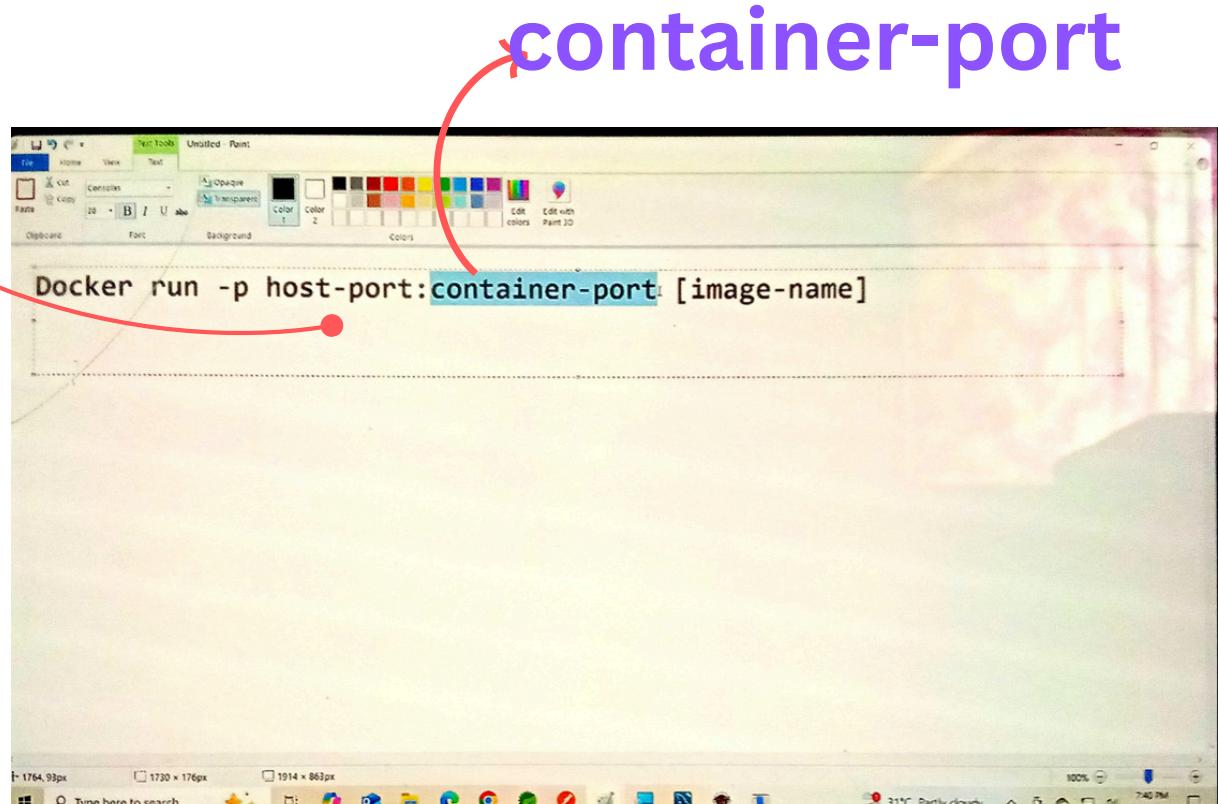
Step 5:

A screenshot of a code editor window titled "application.properties". The file contains the following configuration:

```
1 spring.application.name=demo
2
3 #Database connection
4
5 spring.datasource.url=jdbc:mysql://localhost:3306/demo_dec_db
6 spring.datasource.username=root
7 spring.datasource.password=test
8
9 spring.jpa.hibernate.ddl-auto=update
10
11 server.port=9090
```

Annotations:

- A red arrow points from the word "host-port" to the port number "9090" in line 11.
- A red arrow points from the word "container-port" to the port number "3306" in line 5.



Step 6:

```
docker run -p 9090:9090 [image-name/image-id]
```

Step 6:

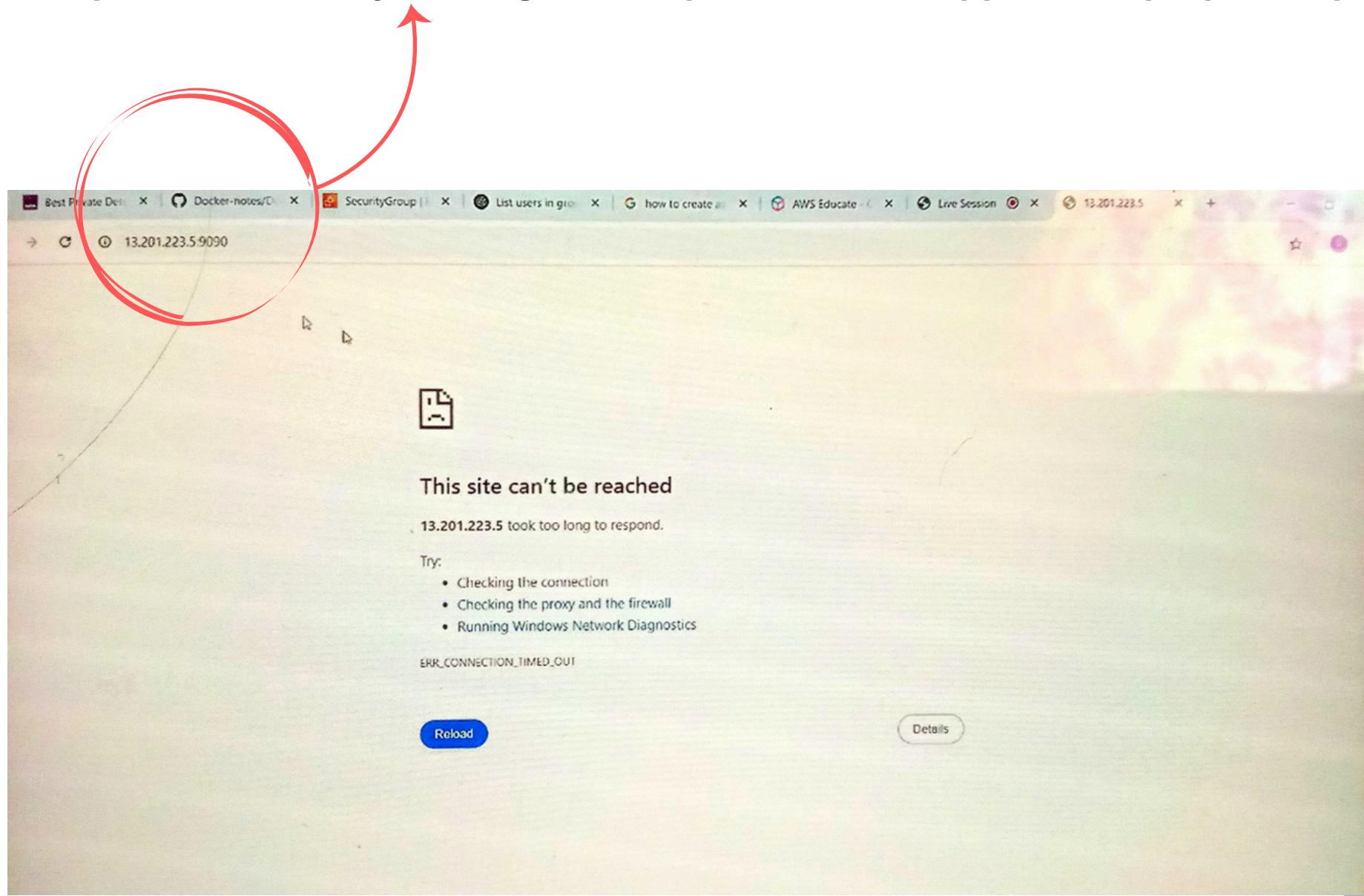
```
[ec2-user@ip-172-31-2-140 ~]$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
psait/pankajsiracademy  latest   f434d835074b  3 days ago  492MB
[ec2-user@ip-172-31-2-140 ~]$ docker run -p 9090:9090 psait/pankajsiracademy

:: Spring Boot ::           (v3.4.4)

2025-03-26T14:15:38.384Z  INFO 1 --- [demo] [main] com.demo_dockers.DemoApplica
tion      : Starting DemoApplication v0.0.1-SNAPSHOT using Java 17.0.2 with PID 1 (/usr
/app/demo-app.jar started by root in /usr/app)
2025-03-26T14:15:38.396Z  INFO 1 --- [demo] [main] com.demo_dockers.DemoApplica
tion      : No active profile set, falling back to 1 default profile: "default"
```

Step 7:

You will not see webpage because you need to configure port in security inbound rule
my host port is ex :9090(you can give host-port same as a application.properties port



Step 7:

The screenshot shows the AWS Management Console interface for managing security group rules. The URL in the browser is `ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#ModifyInboundSecurityGroupRules:securityGroupId=sg_0bdae08e8143fd491`. The page title is "Edit inbound rules" under "sg-0bdae08e8143fd491 - launch-wizard-15".

The table displays two existing rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-054dbad7eb1fc7112	SSH	TCP	22	C... 0.0.0.0/0	Info Delete
-	Custom TCP	TCP	9090	A... 0.0.0.0/0	Info Delete

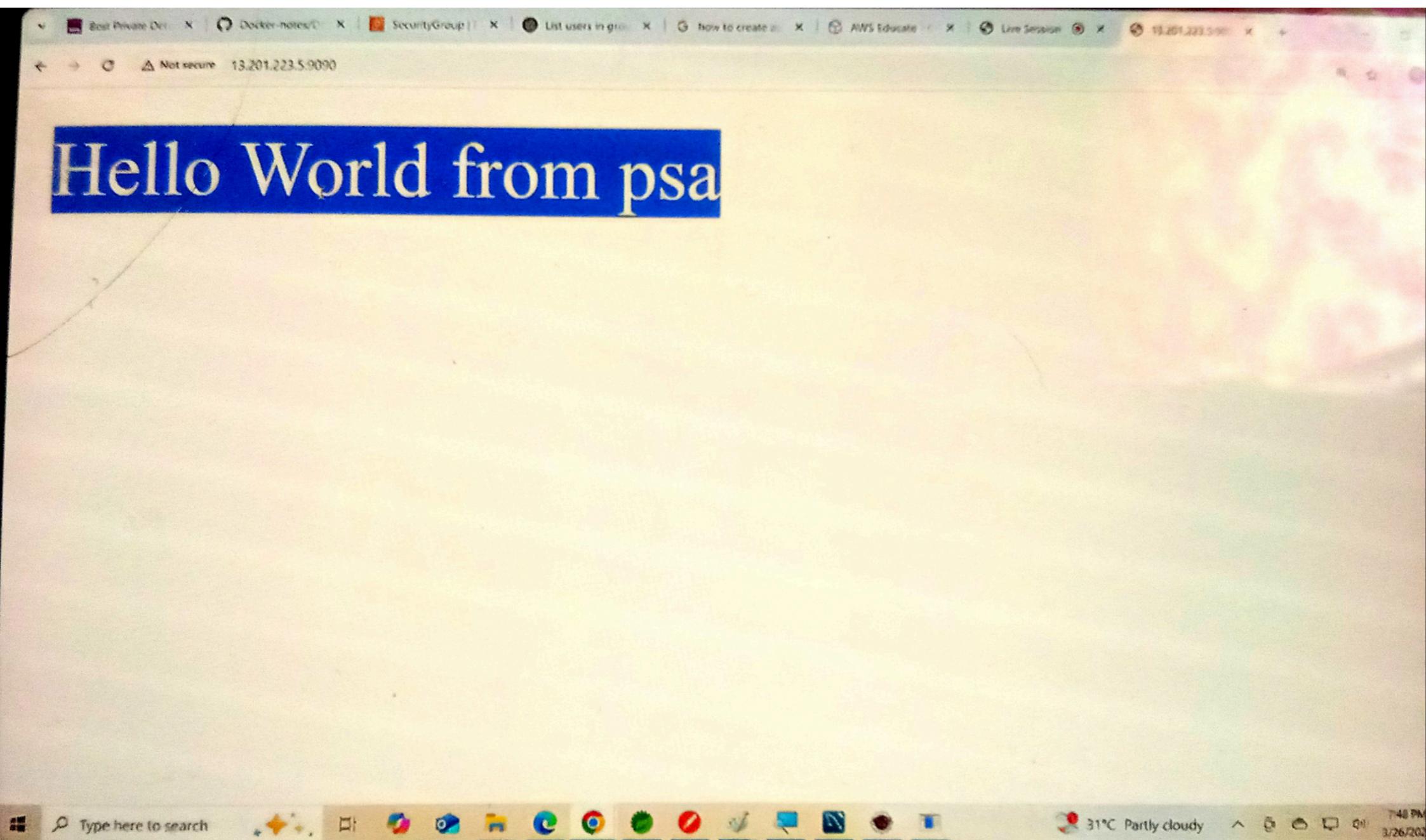
An "Add rule" button is located at the bottom left of the table area.

A warning message in a callout box states: "⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." with a close button "X".

At the bottom right, there are "Cancel", "Preview changes", and "Save rules" buttons. The "Save rules" button is highlighted in red.

The bottom navigation bar includes links for CloudShell, Feedback, and various AWS services like S3, Lambda, and CloudWatch. It also shows the date and time (7:48 PM, 3/26/2025) and weather information (31°C Partly cloudy).

after porting and inbound rule is done you will see webpage



Step 8:

- you will not see your webpage because whenever you close windows/running background
 - to avoid this you need to add -d while running your webpage
 - ex: `docker run -d -p 9090:9090 [image-name]` will run the cintainer in background

```
ec2-user@ip-172-31-2-140: ~
tion          : Starting DemoApplication v0.0.1-SNAPSHOT using Java 17.0.2 with PID 1 (/usr
/app/demo-app.jar started by root in /usr/app)
2025-03-26T14:15:38.396Z  INFO 1 --- [demo] [main] com.demo_dockers.DemoApplica
tion          : No active profile set, falling back to 1 default profile: "default"
2025-03-26T14:15:40.249Z  INFO 1 --- [demo] [main] o.s.b.w.embedded.tomcat.Tomc
atWebServer   : Tomcat initialized with port 9090 (http)
2025-03-26T14:15:40.272Z  INFO 1 --- [demo] [main] o.apache.catalina.core.Stand
ardService    : Starting service [Tomcat]
2025-03-26T14:15:40.273Z  INFO 1 --- [demo] [main] o.apache.catalina.core.Stand
ardEngine     : Starting Servlet engine: [Apache Tomcat/10.1.39]
2025-03-26T14:15:40.477Z  INFO 1 --- [demo] [main] o.a.c.c.c.[Tomcat].[localhost].[/]
          : Initializing Spring embedded WebApplicationContext
2025-03-26T14:15:40.488Z  INFO 1 --- [demo] [main] w.s.c.ServletWebServerAppli
cationContext : Root WebApplicationContext: initialization completed in 1977 ms
2025-03-26T14:15:41.656Z  INFO 1 --- [demo] [main] o.s.b.w.embedded.tomcat.Tomc
atWebServer   : Tomcat started on port 9090 (http) with context path '/'
2025-03-26T14:15:41.691Z  INFO 1 --- [demo] [main] com.demo_dockers.DemoApplica
tion          : Started DemoApplication in 4.359 seconds (process running for 5.406)
2025-03-26T14:18:15.506Z  INFO 1 --- [demo] [nio-9090-exec-1] o.a.c.c.c.[Tomcat].[localhost].[/]
          : Initializing Spring DispatcherServlet 'dispatcherServlet'
2025-03-26T14:18:15.507Z  INFO 1 --- [demo] [nio-9090-exec-1] o.s.web.servlet.DispatcherSe
rvlet        : Initializing Servlet 'dispatcherServlet'
2025-03-26T14:18:15.516Z  INFO 1 --- [demo] [nio-9090-exec-1] o.s.web.servlet.DispatcherSe
rvlet        : Completed initialization in 1 ms
#####
#####
```

Step 8: Now you can run your webpage even if you close windows/aws account by just adding below command

docker run -d -p 9090:9090 [image-name] will run the cintainer in background

```
ec2-user@ip-172-31-2-140:~$ Admin@DESKTOP-9DFQ51N MINGW64 /g/docker-keys
$ ssh -i "docker_keys.pem" ec2-user@ec2-13-201-223-5.ap-south-1.compute.amazonaws.com
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
Last login: Wed Mar 26 14:20:10 2025 from 27.7.148.10
[ec2-user@ip-172-31-2-140 ~]$ docker run -p -d 9090:9090 psait/pankajsiracademy
docker: invalid containerPort: -d.
See 'docker run --help'.
[ec2-user@ip-172-31-2-140 ~]$ docker run -d -p 9090:9090 psait/pankajsiracademy
745e59dc108435440b510a8750787becd91465da64045db377767c851e59cec9
[ec2-user@ip-172-31-2-140 ~]$
```

Best Private Dev X Docker-notes X SecurityGroup X List users in gro X how to create X AWS Educate X Live Session X 13.201.223.5:9090 X

Not secure 13.201.223.5:9090

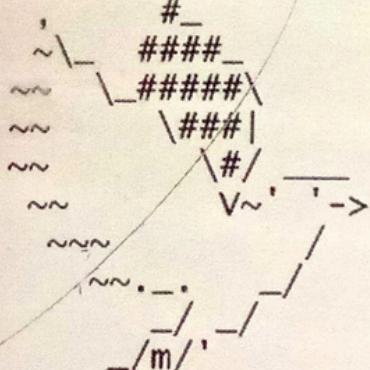
Hello World from psa

Other commands:

docker ps :To display docker containers that are running

docker ps

```
$ ssh -i "docker_keys.pem" ec2-user@ec2-13-201-223-5.ap-south-1.compute.amazonaws.com
```



Amazon Linux 2023

<https://aws.amazon.com/linux/amazon-linux-2023>

```
Last login: wed Mar 26 14:20:10 2025 from 27.7.148.10
```

```
[ec2-user@ip-172-31-2-140 ~]$ docker run -p -d 9091:9090 psait/pankajsiracademy  
docker: invalid containerPort: -d.
```

```
See 'docker run --help'.
```

```
[ec2-user@ip-172-31-2-140 ~]$ docker run -d -p 9091:9090 psait/pankajsiracademy  
745e59dc108435440b510a8750787becd914651a64045db377767c851e59cec9
```

```
[ec2-user@ip-172-31-2-140 ~]$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	P
ORTS			NAMES		
745e59dc1084	psait/pankajsiracademy	"java -jar demo-app..."	16 seconds ago	Up 16 seconds	8
080/tcp, 0.0.0.0:9091->9090/tcp, :::9091->9090/tcp			sleepy_allen		
[ec2-user@ip-172-31-2-140 ~]\$					

docker stop :To Stop docker container

docker stop [container-id]

```
Administrator:~$ docker run -p -d 9091:9090 psait/pankajsiracademy
docker: invalid containerPort: -d.
see 'docker run --help'.
[ec2-user@ip-172-31-2-140 ~]$ docker run -d -p 9091:9090 psait/pankajsiracademy
745e59dc108435440b510a8750787becd91465da4045db377767c851e59cec9
[ec2-user@ip-172-31-2-140 ~]$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
NAMES
745e59dc1084        psait/pankajsiracademy   "java -jar demo-app..."   16 seconds ago    Up 16 seconds     0.0.0.0:9091->9090/tcp, 0.0.0.0:9091->9090/tcp
[ec2-user@ip-172-31-2-140 ~]$ docker stop 745e59dc1084
[ec2-user@ip-172-31-2-140 ~]$
```

display stopped containers:

docker ps -a

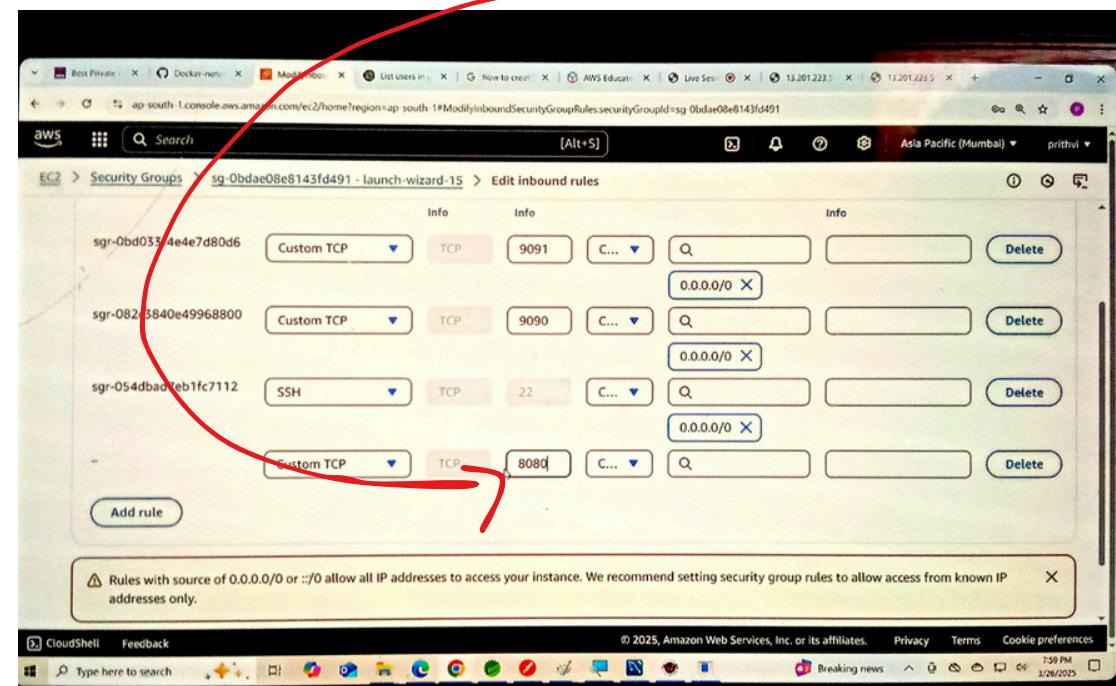


```
[ec2-user@ip-172-31-2-140 ~]$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                 NAMES
745e59dc1084        psait/pankajsiracademy   "java -jar demo-app..."   48 seconds ago    Exited (143) 11 s
econds ago
4701f3a9bc96        psait/pankajsiracademy   "java -jar demo-app..."   3 minutes ago     Exited (129) Abou
t a minute ago
5472065aeee2        psait/pankajsiracademy   "java -jar demo-app..."   3 minutes ago     Created
pedantic_lederberg
538ba4229b7d        psait/pankajsiracademy   "java -jar demo-app..."   4 minutes ago     Exited (129) Abou
t a minute ago
aeaf70a6e6cb        psait/pankajsiracademy   "java -jar demo-app..."   8 minutes ago     Exited (129) 5 mi
nutes ago
[silly_khorana]
[ec2-user@ip-172-31-2-140 ~]$
```

Install jenkins using docker image name - `docker run -d -p 8080:8080 jenkins/jenkins`

Note Enable Inbound rule in security group custom ip 8080 IPv4 anywhere

```
[ec2-user@ip-172-31-2-140 ~]$ docker run -d -p 8080:8080 jenkins/jenkins
unable to find image 'jenkins/jenkins:latest' locally
latest: Pulling from jenkins/jenkins
7cd785773db4: Pull complete
4323b613447d: Pull complete
eec1952536a9: Pull complete
b9fcf549558d: Pull complete
743c1c69ebe6: Pull complete
2b005e8acf52: Pull complete
e7e22a1da8f6: Pull complete
5daeef7ea5eb6: Pull complete
bbb418a8a466: Pull complete
ccfc3f9a95cb: Pull complete
b6a32e0eeef53: Pull complete
5faef2a0dcdc: Pull complete
digest: sha256:89b19a1fc0b079d6e4ab13951902d7e84935921a2186b0ff13534983530b1ea48
status: Downloaded newer image for jenkins/jenkins:latest
c9efae3d0c47cfef56d4707b4a2f773e5212ffbf0b98ae24e29bb377d0d58fa
[ec2-user@ip-172-31-2-140 ~]$
```



Getting Started

Continue





Installing Docker:

For Amazon Linux use the following commands

Install Docker In Amazon Linux VM

```
sudo yum update -y  
sudo yum install docker -y  
sudo service docker start  
sudo usermod -aG docker ec2-user  
exit
```

Use this command to check version of docker installed

```
docker -v
```

For practise pull the sample image from docker hub repository of pankaj sir academy: docker pull psait/pankajsiracademy:latest

For Practise pull docker official image: docker pull hello-world

Example: docker run -d -p 9090:9090 [image-name] will run the cintainer in background

Important Docker Commands

docker pull : download docker image from hub

docker pull [image-name]

docker run : run docker image - this will create container (Isolated Environment to run docker image- This is not a OSs)

docker run [image-name / image-id]

docker ps :To display docker containers that are running

docker ps

display stopped containers:

docker ps -a

docker stop :To Stop docker container

docker stop [container-id]

docker start : Start docker container

docker start [container-id]

docker rm : will remove stopped docker container

docker rm [contianer-id]

docker rmi : Will Remove docker image

docker rmi [image-name / image-id]

To remove all stopped containers and un-used docker images we can use below command

docker system prune -a

PART 2

done