

REDDY PRAKASH

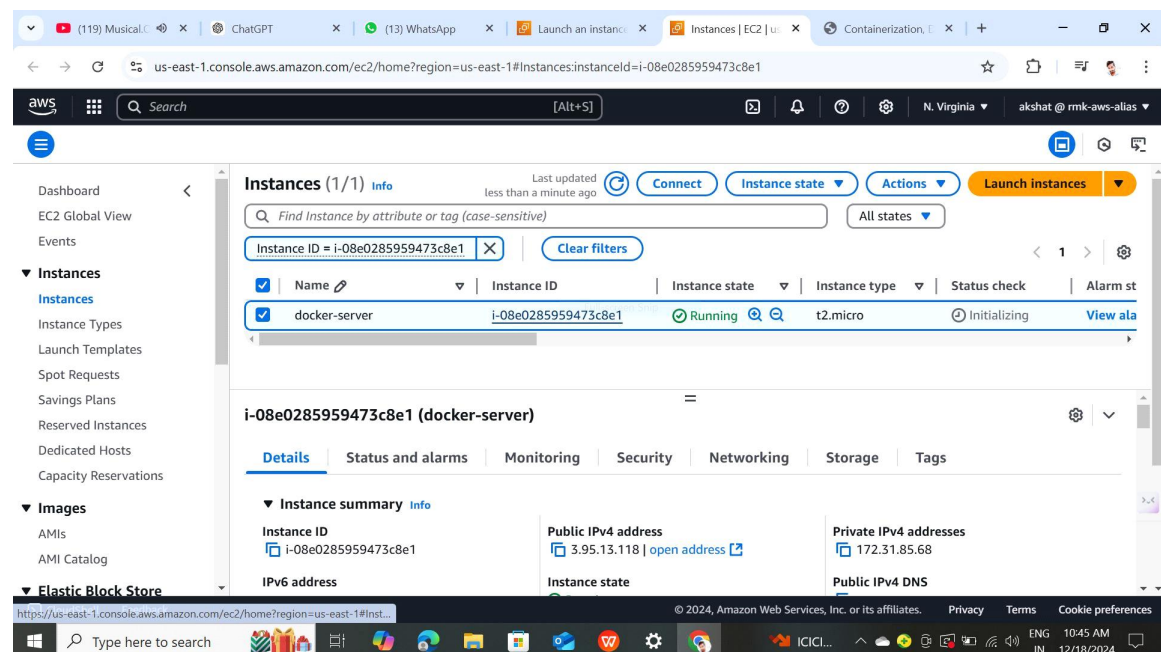
Assignments Containerization, Docker, and Docker Hub

1. L1 - Create Docker file and build the docker

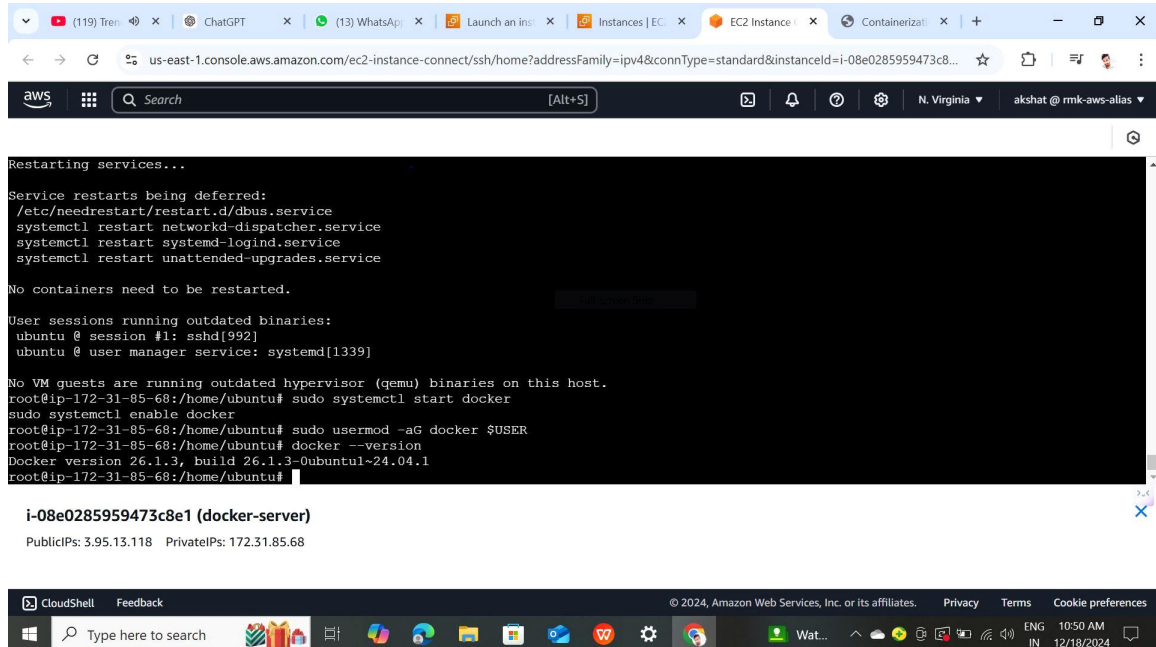
container Application Image for the application

build in Jenkins Module

Successfully create a instance with name docker-service



Successfully update packages and install docker



The screenshot shows a terminal window within the AWS CloudShell interface. The terminal output is as follows:

```
Restarting services...
Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service

No containers need to be restarted.

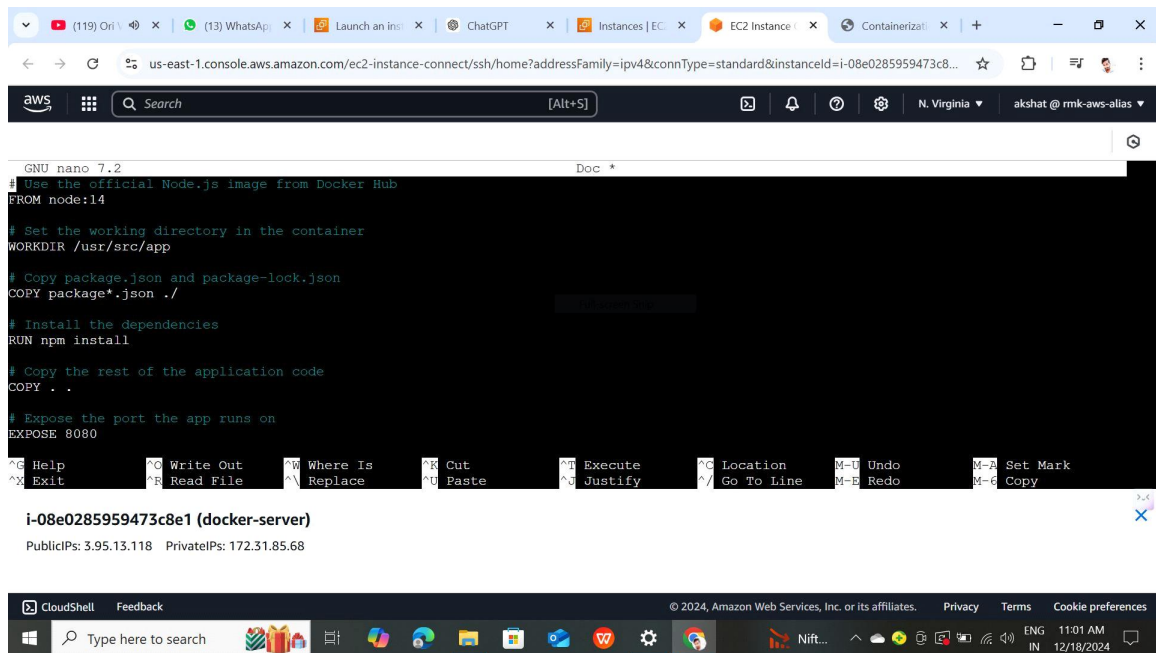
User sessions running outdated binaries:
ubuntu @ session #1: sshd[992]
ubuntu @ user manager service: systemd[1339]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-85-68:/home/ubuntu# sudo systemctl start docker
sudo systemctl enable docker
root@ip-172-31-85-68:/home/ubuntu# sudo usermod -aG docker $USER
root@ip-172-31-85-68:/home/ubuntu# docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
root@ip-172-31-85-68:/home/ubuntu#
```

Below the terminal output, the instance details are shown: **i-08e0285959473c8e1 (docker-server)**, with PublicIPs: 3.95.13.118 and PrivateIPs: 172.31.85.68.

- we say the docker version 26.1.3

Successfully create docker file for json application



- Expose the port 8080 through internet

Build the docker image successfully for jenkins application model

```
14 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities

----> Removed intermediate container ae5a5bceba20
----> 3f54d597a95a
Step 5/7 : COPY . .
----> d1f24b949082
Step 6/7 : EXPOSE 8080
----> Running in b602b057dcf8
----> Removed intermediate container b602b057dcf8
----> 75107e7570d1
Step 7/7 : CMD ["node", "index.js"]
----> Running in 57ca22595f16
----> Removed intermediate container 57ca22595f16
----> 42b1d1541cde
Successfully built 42b1d1541cde
Successfully tagged my-app-image:latest
root@ip-172-31-85-68:/home/ubuntu/my-app#
```

i-08e0285959473c8e1 (docker-server)
PublicIPs: 3.95.13.118 PrivateIPs: 172.31.85.68

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search

ICICI...

ENG 11:09 AM
IN 12/18/2024

we successfully create a container by using docker image

```
----> Removed intermediate container ae5a5bceba20
----> 3f54d597a95a
Step 5/7 : COPY . .
----> d1f24b949082
Step 6/7 : EXPOSE 8080
----> Running in b602b057dcf8
----> Removed intermediate container b602b057dcf8
----> 75107e7570d1
Step 7/7 : CMD ["node", "index.js"]
----> Running in 57ca22595f16
----> Removed intermediate container 57ca22595f16
----> 42b1d1541cde
Successfully built 42b1d1541cde
Successfully tagged my-app-image:latest
root@ip-172-31-85-68:/home/ubuntu/my-app# docker run -d -p 8080:8080 --name my-app-container my-app-image
32de9d34911d3868ae7b3c84819eb0850756587225846dc511a747cdf12c0bf3
root@ip-172-31-85-68:/home/ubuntu/my-app# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
32de9d34911d   my-app-image   "docker-entrypoint.s..." 26 seconds ago Up 25 seconds 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp my-app-container
root@ip-172-31-85-68:/home/ubuntu/my-app#
```

i-08e0285959473c8e1 (docker-server)
PublicIPs: 3.95.13.118 PrivateIPs: 172.31.85.68

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

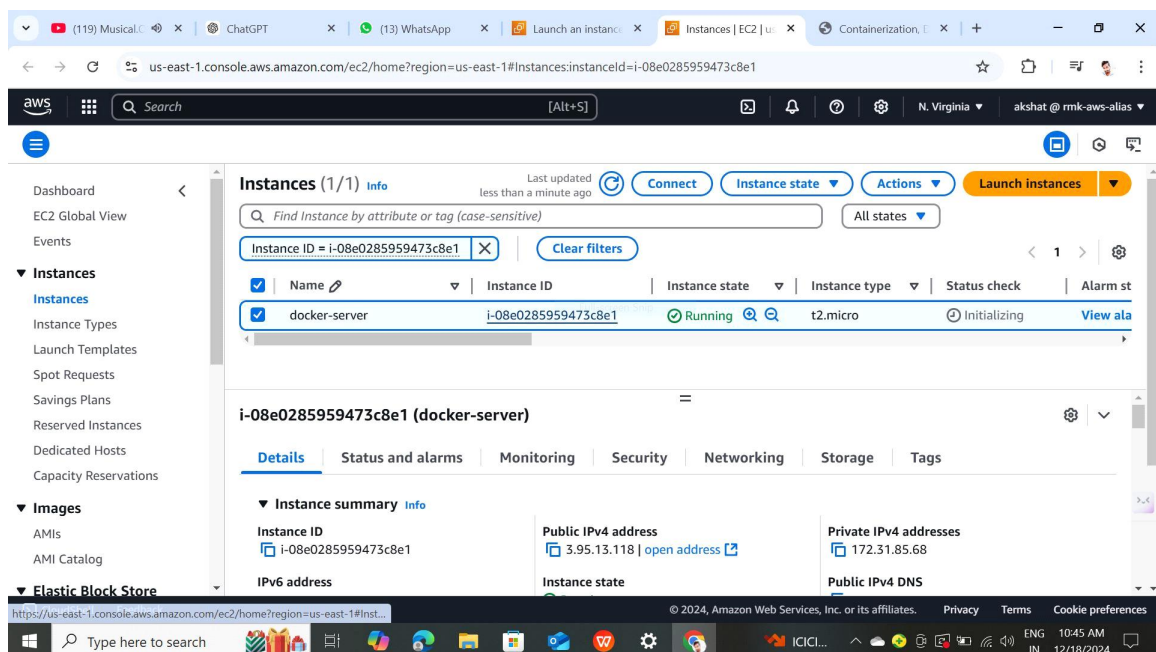
Type here to search

SBIN

ENG 11:10 AM
IN 12/18/2024

2. L2 - Create the Container using the same Application Image and run the application in a Web Browser using container port mapping

Successfully launch the instance with name docker-server



install docker and update package

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-08e0285959473c8...

Restarting services...

Service restarts being deferred:

```
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service
```

No containers need to be restarted.

User sessions running outdated binaries:

```
ubuntu @ session #1: sshd[992]
ubuntu @ user manager service: systemd[1339]
```

No VM guests are running outdated hypervisor (qemu) binaries on this host.

```
root@ip-172-31-85-68:/home/ubuntu# sudo systemctl start docker
sudo systemctl enable docker
root@ip-172-31-85-68:/home/ubuntu# sudo usermod -aG docker $USER
root@ip-172-31-85-68:/home/ubuntu# docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
root@ip-172-31-85-68:/home/ubuntu#
```

i-08e0285959473c8e1 (docker-server)

PublicIPs: 3.95.13.118 PrivateIPs: 172.31.85.68

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search

Wat...

ENG 10:50 AM
IN 12/18/2024

create image and build the container with expose port 8080

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-08e0285959473c8...

```
----> Removed intermediate container ae5a5bceba20
----> 3f54d597a95a
Step 5/7 : COPY . .
----> d1f24b949082
Step 6/7 : EXPOSE 8080
----> Running in b602b057dcf8
----> Removed intermediate container b602b057dcf8
----> 75107e7570d1
Step 7/7 : CMD ["node", "index.js"]
----> Running in 57ca22595f16
----> Removed intermediate container 57ca22595f16
----> 42b1d1541cde
Successfully built 42b1d1541cde
Successfully tagged my-app-image:latest
root@ip-172-31-85-68:/home/ubuntu/my-app# docker run -d -p 8080:8080 --name my-app-container my-app-image
32de9d34911d3868ae7b3c84819eb0850756587225846dc511a747cdf12c0bf3
root@ip-172-31-85-68:/home/ubuntu/my-app# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
32de9d34911d	my-app-image	"docker-entrypoint.s..."	26 seconds ago	Up 25 seconds	0.0.0.0:8080->8080/tcp, :::8080->8080/tcp	my-app

```
-container
root@ip-172-31-85-68:/home/ubuntu/my-app#
```

i-08e0285959473c8e1 (docker-server)

PublicIPs: 3.95.13.118 PrivateIPs: 172.31.85.68

CloudShell Feedback

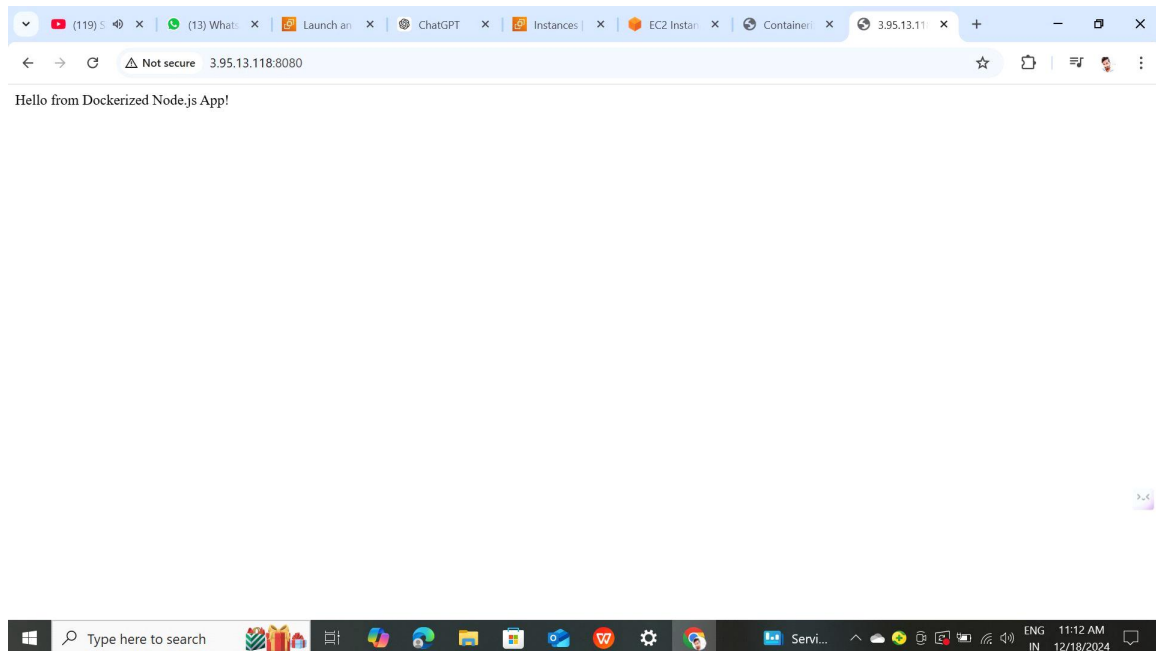
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search

SBIN

ENG 11:10 AM
IN 12/18/2024

Successfully access application through internet port 8080



3. L3 - Demonstrate Docker Compose using the

Application Image and MySQL Image to start and

stop all container services

i use same instance to install docker compose and mysql data

base and also update packages

The screenshot shows a terminal window within the AWS CloudShell interface. The terminal output displays the results of a diagnostic command, indicating that the current kernel version (6.8.0-1021-aws) does not match the expected version. It suggests a reboot to load the new kernel. Subsequently, several system services are restarted, including dbus.service, networkd-dispatcher.service, systemd-logind.service, and unattended-upgrades.service. The terminal also notes that no containers need to be restarted and that user sessions are running outdated binaries. Finally, it states that no VM guests are running outdated hypervisor (qemu) binaries. The terminal prompt is root@ip-172-31-85-68:/home/ubuntu#. Below the terminal output, the instance ID i-08e0285959473c8e1 (docker-server) and its public and private IP addresses are listed. The bottom of the screenshot shows the AWS CloudShell toolbar with various icons and the system tray of the host machine.

```
Diagnostics:
The currently running kernel version is not the expected kernel version 6.8.0-1021-aws.
Restarting the system to load the new kernel will not be handled automatically, so you should consider rebooting.
Restarting services...
Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service
No containers need to be restarted.
User sessions running outdated binaries:
ubuntu @ session #1: sshd[992]
ubuntu @ user manager service: systemd[1339]
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-85-68:/home/ubuntu#
```

i-08e0285959473c8e1 (docker-server)
PublicIPs: 3.95.13.118 PrivateIPs: 172.31.85.68

- Successfully install docker-compose

with the help of docker --version command to find version of
docker-compose


```
Restarting the system to load the new kernel will not be handled automatically, so you should consider rebooting.
Restarting services...

Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
ubuntu @ session #1: sshd[992]
ubuntu @ user manager service: systemd[1339]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-85-68:/home/ubuntu# docker-compose --version
docker-compose version 1.29.2, build unknown
root@ip-172-31-85-68:/home/ubuntu#
```

i-08e0285959473c8e1 (docker-server)
PublicIPs: 3.95.13.118 PrivateIPs: 172.31.85.68

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Successfully create docker-compose file with js application

```
GNU nano 7.2 docker-compose.yml *
version: '3.8'

services:
  app:
    image: my-app-image
    ports:
      - "8080:8080"
    depends on:
      - mysql

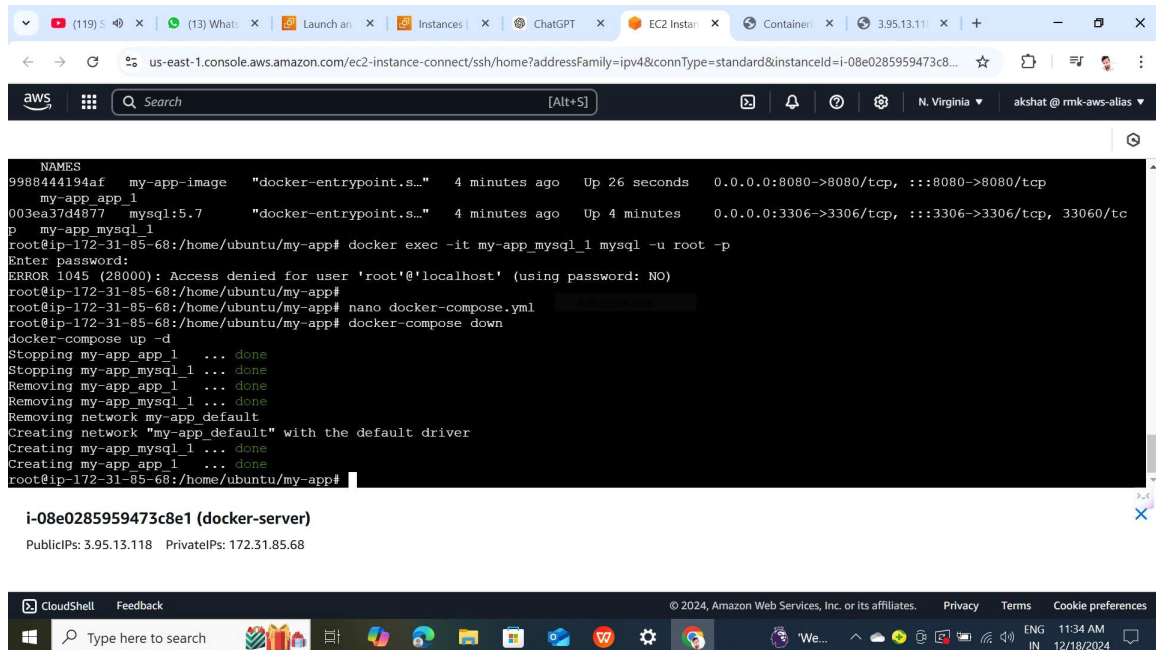
  mysql:
    image: mysql:5.7
    environment:
      MYSQL_ROOT_PASSWORD: example
      MYSQL_DATABASE: myapp_db
    ports:
      - "3306:3306"
```

i-08e0285959473c8e1 (docker-server)
PublicIPs: 3.95.13.118 PrivateIPs: 172.31.85.68

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search 26°C 11:24 AM 12/18/2024

Successfully build two container docker-compose and MySQL container



The screenshot displays the AWS Management Console interface. The top navigation bar shows the user is logged in as 'akshat@rmk-aws-alias' in the 'N. Virginia' region. The main content area shows the 'Container' tab for an EC2 instance. The logs for the 'my-app' container are visible, showing the following commands and output:

```
root@ip-172-31-85-68:/home/ubuntu/my-app# docker exec -it my-app_mysql_1 mysql -u root -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
root@ip-172-31-85-68:/home/ubuntu/my-app# nano docker-compose.yml
root@ip-172-31-85-68:/home/ubuntu/my-app# docker-compose down
docker-compose up -d
Stopping my-app_app_1 ... done
Stopping my-app_mysql_1 ... done
Removing my-app_app_1 ... done
Removing my-app_mysql_1 ... done
Removing network my-app_default
Creating network "my-app_default" with the default driver
Creating my-app_mysql_1 ... done
Creating my-app_app_1 ... done
root@ip-172-31-85-68:/home/ubuntu/my-app#
```

Below the logs, the instance details for 'i-08e0285959473c8e1 (docker-server)' are shown, including Public IPs (3.95.13.118) and Private IPs (172.31.85.68).

The bottom of the screenshot shows the CloudShell terminal window with the AWS logo and the text '© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

Above figure we saw create container and start and stop the containers

Successfully login mysql database container

The screenshot shows the AWS Management Console interface. At the top, there's a browser window with multiple tabs. The main content area displays a table of containers with columns: CONTAINER ID, IMAGE, COMMAND, CREATED, STATUS, and PORTS. Below the table, there's a terminal window showing the command `docker exec -it my-app_mysql_1 mysql -u root -p` and the subsequent MySQL login process. The terminal output includes the MySQL welcome message and the prompt `mysql>`. Below the terminal, the instance ID `i-08e0285959473c8e1` is shown, along with the label `(docker-server)` and public/private IP addresses.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
05505258a139	my-app-image	"docker-entrypoint.s..."	9 seconds ago	Up 8 seconds	0.0.0.0:8080->8080/tcp, :::8080->8080/tcp
334df4359ad5	mysql:5.7	"docker-entrypoint.s..."	9 seconds ago	Up 9 seconds	0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp

```
root@ip-172-31-85-68:/home/ubuntu/my-app# docker exec -it my-app_mysql_1 mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.44 MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

i-08e0285959473c8e1 (docker-server)
PublicIPs: 3.95.13.118 PrivateIPs: 172.31.85.68

- Successfully login the container

with help of SHOW DATABASE commd we show database history

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-08e0285959473c8...

Search [Alt+S]

N. Virginia akshat@rmk-aws-alias

```
-> docker logs my-app_mysql_1
->
-> SHOW DATABASES;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'docker logs my-app_app_1'
docker logs my-app_mysql_1
SHOW DATABASES' at line 1
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql>
```

i-08e0285959473c8e1 (docker-server)
PublicIPs: 3.95.13.118 PrivateIPs: 172.31.85.68

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search

27°C 11:51 AM 12/18/2024

Successfully access application through port 8080

New Tab (13) WhatsApp Launch an Instances ChatGPT EC2 Instance Container 3.95.13.118

Not secure 3.95.13.118:8080

Hello from Dockerized Node.js App!

Type here to search

AIRT... 11:57 AM 12/18/2024

THANK YOU