

Name: Abdurrahman Qureshi

Roll No: 242466

Practical No: 9

Date Of Performance: 10/09/2025

Aim: To understand containerization by deploying a lightweight Nginx web server using Docker on an EC2 instance, demonstrating the practical advantages of containers over traditional virtual machines.

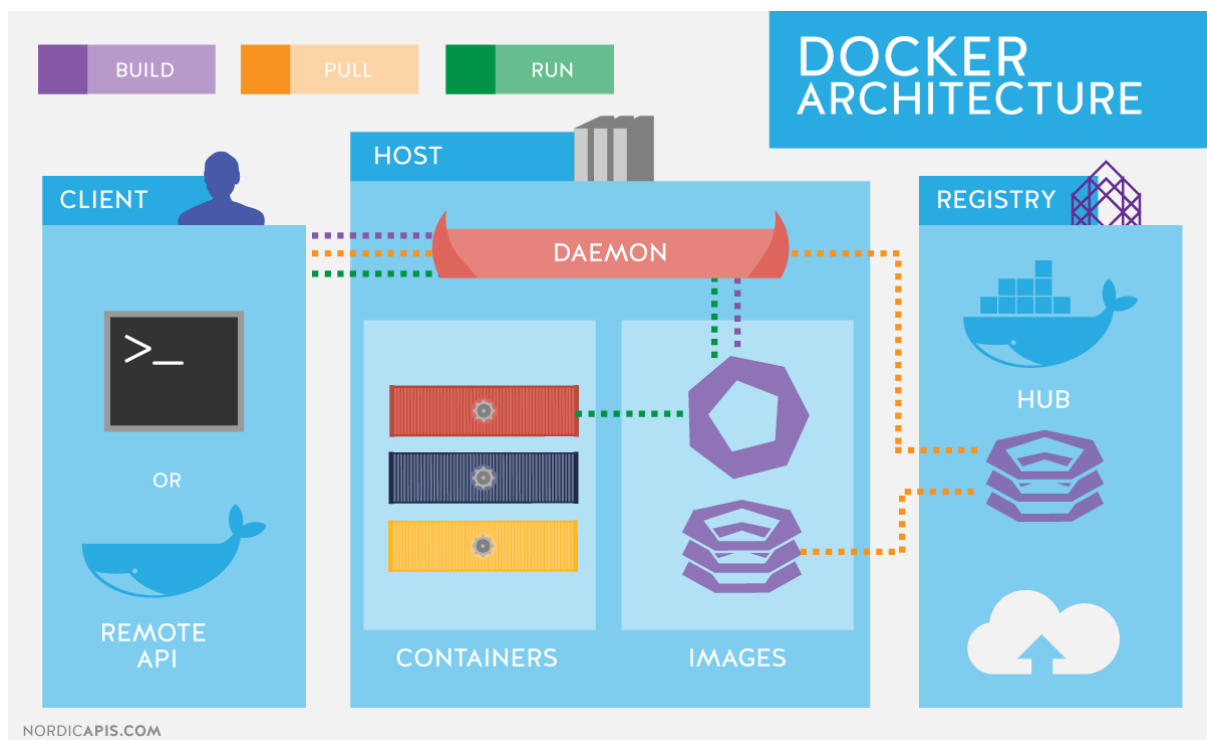
1. What is Containerization / Docker? Explain Docker Architecture with the help of diagram
2. Compare Containers vs VMs
3. Why are Containers lightweight?
4. Deploy a containerized web Application on AWS EC2 Linux. [install Docker, pull nginx image and run it]. Pull python images and run the command to list all the locally stored docker images.

[Terminate the resources after performing the practical-terminate environment and application both]

ANS.1:

Containerization is a lightweight form of virtualization that packages an application and its dependencies (libraries, config files, etc.) into an isolated, portable unit called a **container**.

Docker is the most popular platform that enables developers to build, ship, and run containers consistently across different environments.



[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)

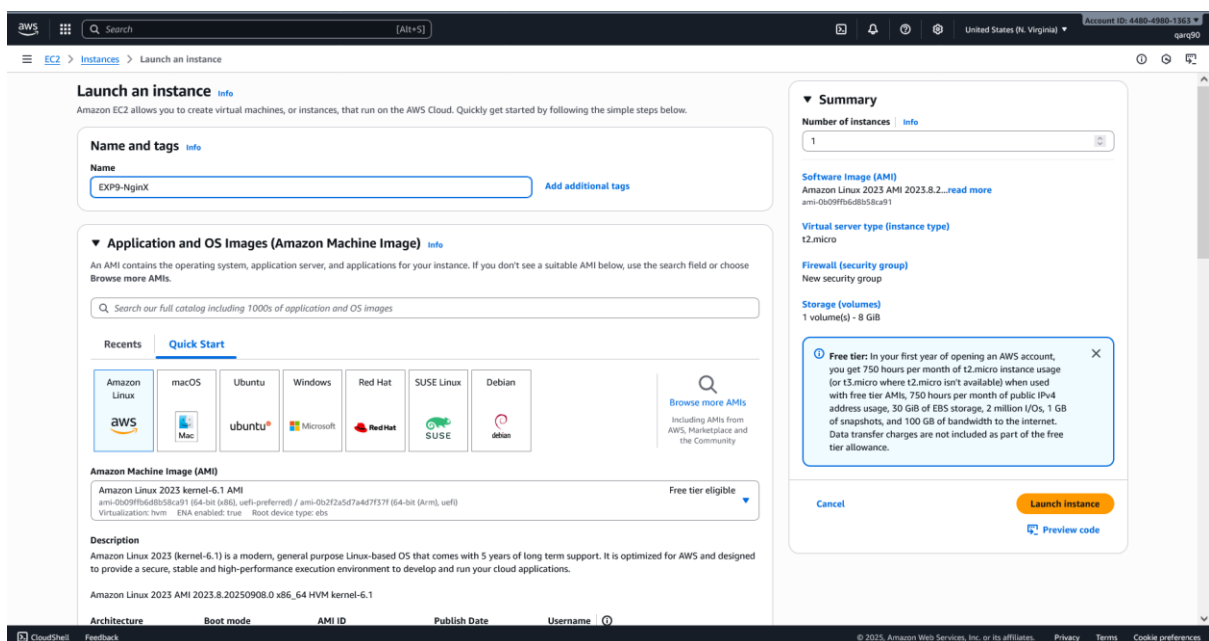
ANS.2:

Containers	Virtual Machines (VMs)
Share the host OS kernel	Each VM has its own full OS
Lightweight, fast boot-up	Heavy, slower boot-up
Less resource overhead	High resource overhead
Portable and consistent	Less portable due to size
Isolated at process level	Fully isolated hardware virtualization

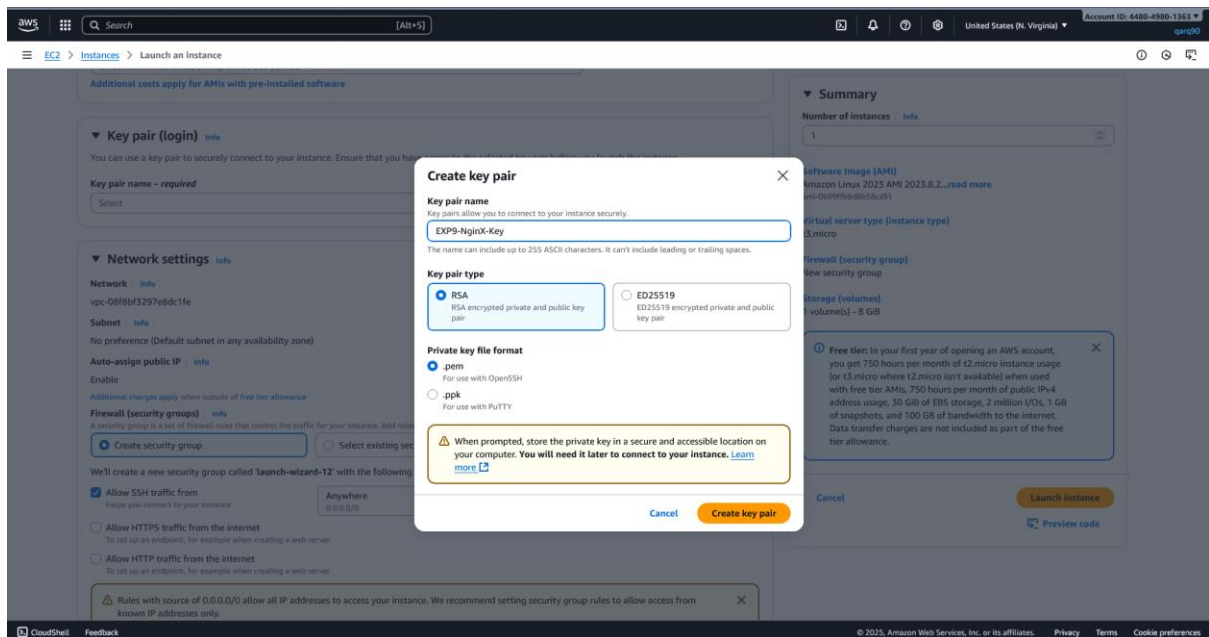
ANS.3:

Containers are lightweight because they **share the host operating system's kernel** and do not require a full operating system for each instance. Only the application, its dependencies, and a minimal runtime are packaged, eliminating the overhead of multiple guest OSes.

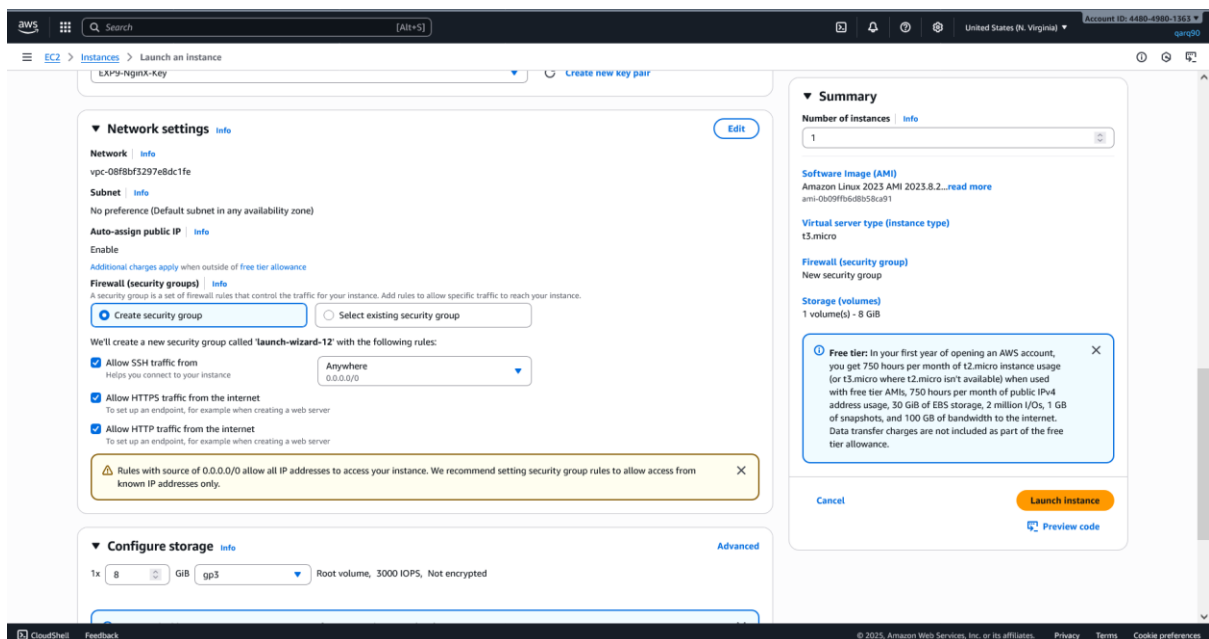
ANS.4:



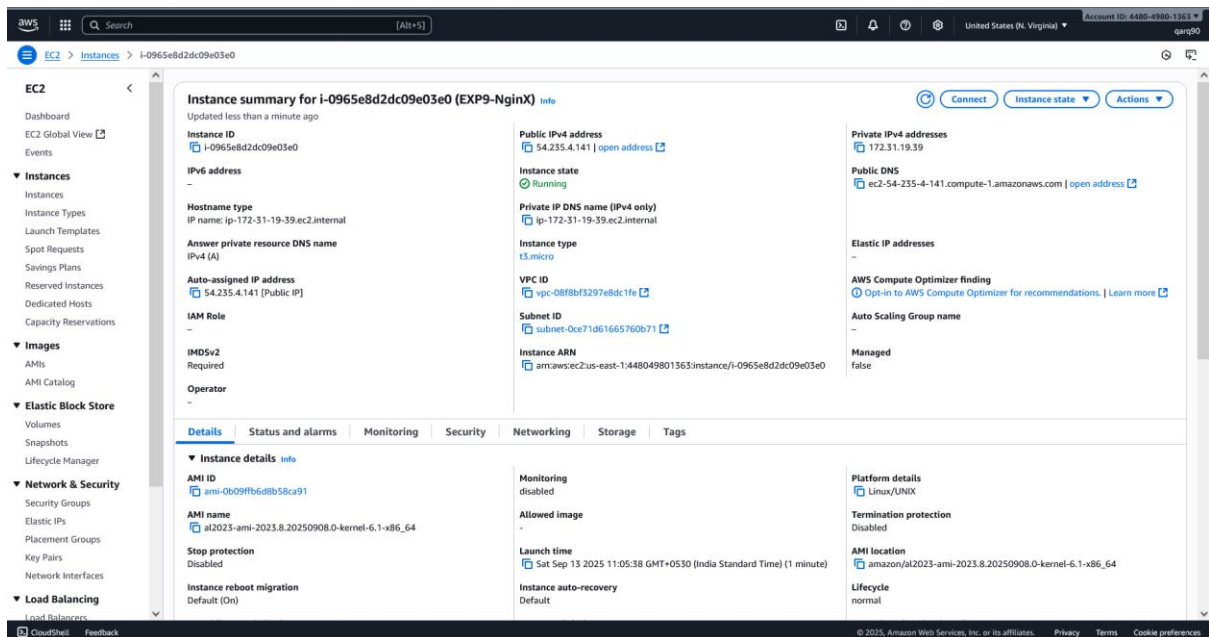
Creating a new Instance



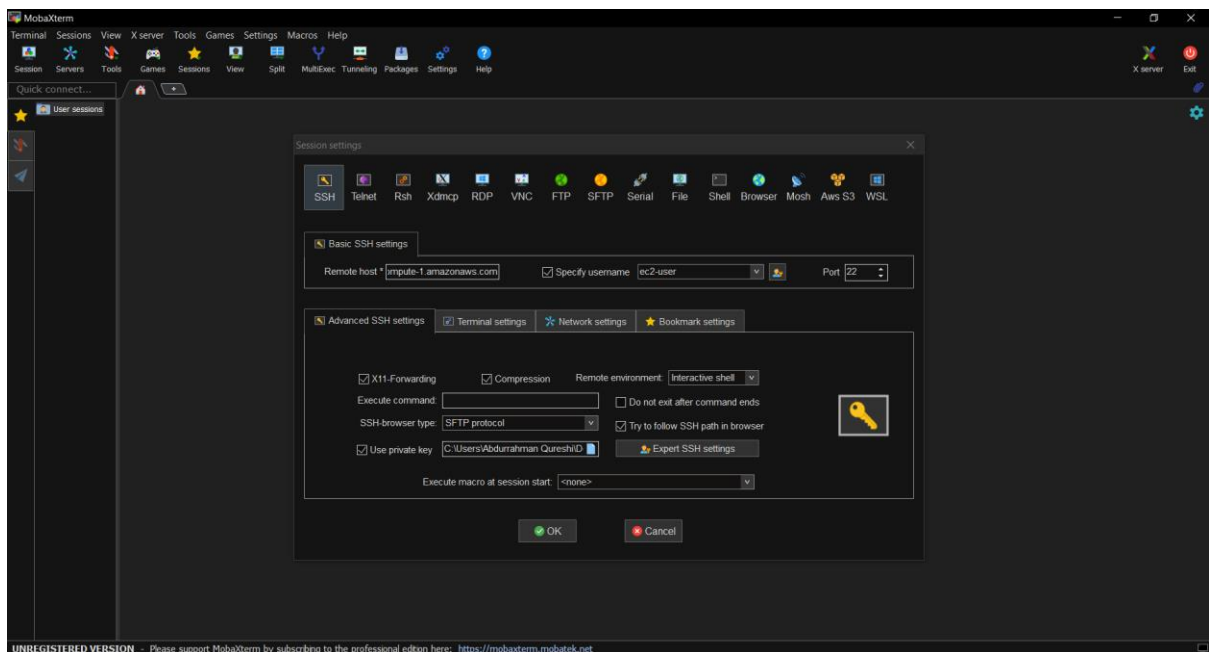
Creating Key Pair



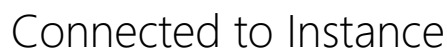
Configuring Network Settings



Instance Details

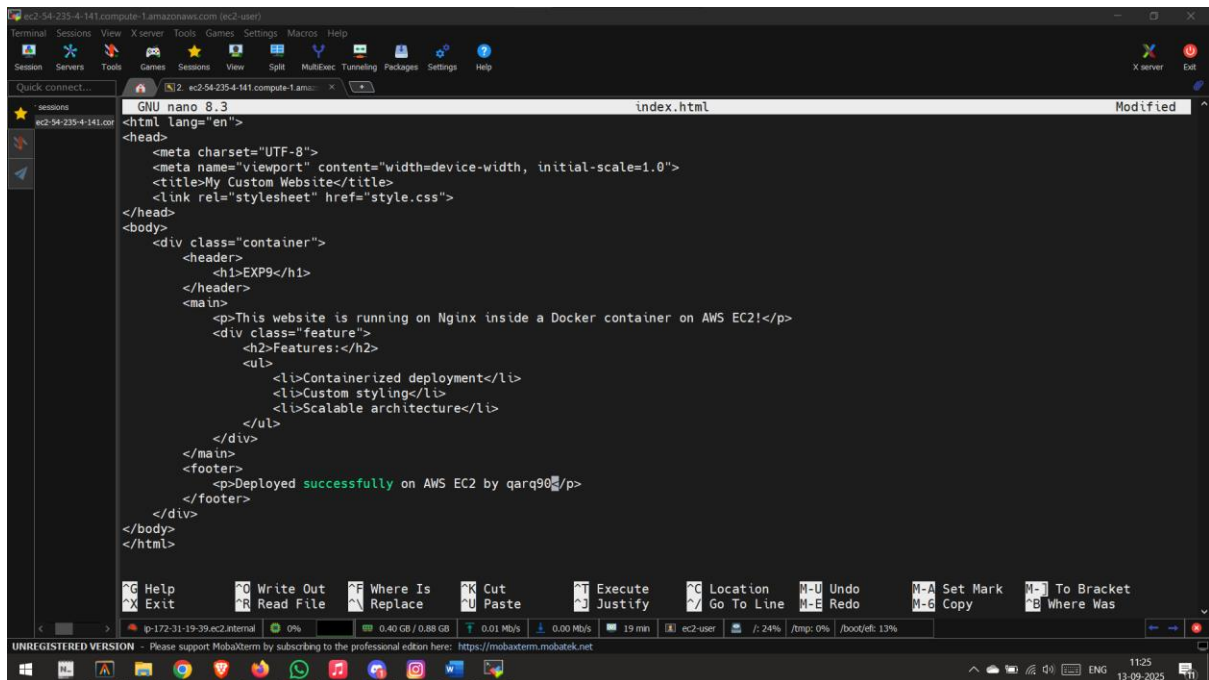


Connecting to Instance on MobaXterm

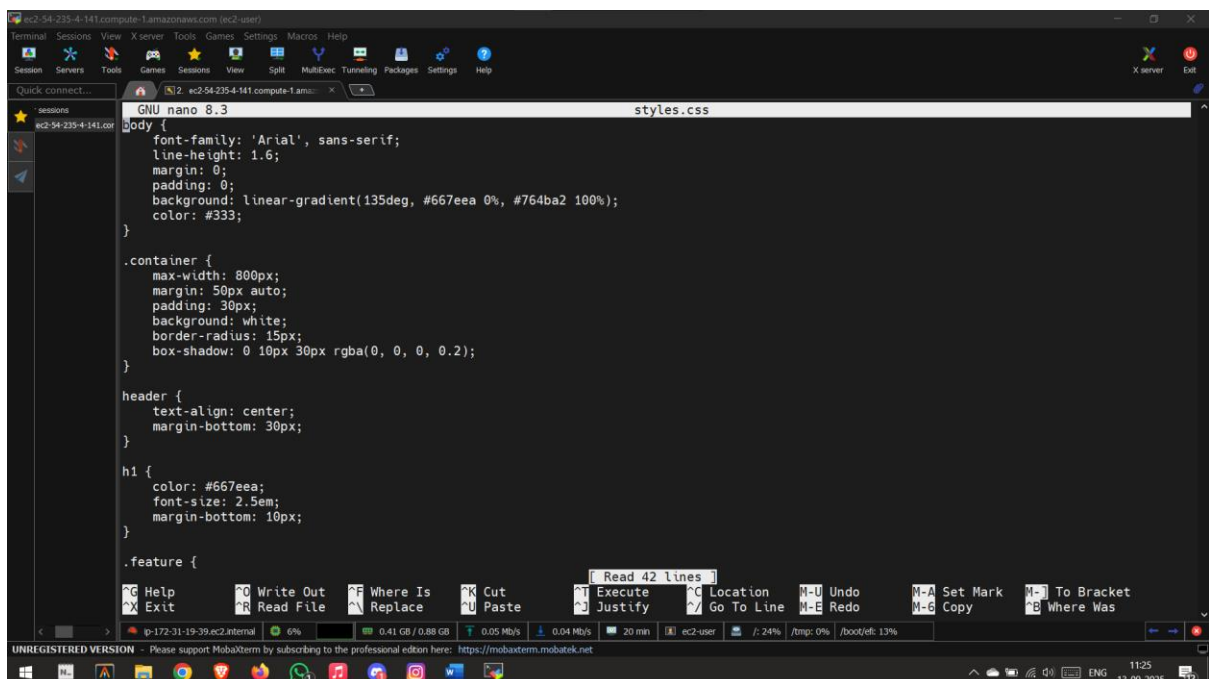


```
[ec2-user@ip-172-31-19-39 ~]$ sudo systemctl start docker
[ec2-user@ip-172-31-19-39 ~]$ sudo systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-19-39 ~]$ sudo usermod -a -G docker ec2-user
[ec2-user@ip-172-31-19-39 ~]$ docker --version
Docker version 25.0.8, build 0bab007
```

Docker Installed



```
GNU nano 8.3 index.html
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My Custom Website</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="container">
    <header>
      <h1>EXP9</h1>
    </header>
    <main>
      <p>This website is running on Nginx inside a Docker container on AWS EC2!</p>
      <div class="feature">
        <h2>Features:</h2>
        <ul>
          <li>Containerized deployment</li>
          <li>Custom styling</li>
          <li>Scalable architecture</li>
        </ul>
      </div>
    </main>
    <footer>
      <p>Deployed successfully on AWS EC2 by qarq90</p>
    </footer>
  </div>
</body>
</html>
```



```
GNU nano 8.3 styles.css
body {
  font-family: 'Arial', sans-serif;
  line-height: 1.6;
  margin: 0;
  padding: 0;
  background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
  color: #333;
}

.container {
  max-width: 800px;
  margin: 50px auto;
  padding: 30px;
  background: white;
  border-radius: 15px;
  box-shadow: 0 10px 30px rgba(0, 0, 0, 0.2);
}

header {
  text-align: center;
  margin-bottom: 30px;
}

h1 {
  color: #667eea;
  font-size: 2.5em;
  margin-bottom: 10px;
}

.feature {
```

Created Sample html and CSS file to deploy

The screenshot shows a terminal window with the following commands and output:

```
[ec2-user@ip-172-31-19-39 ~]$ docker run -d \
--name exp9-nginx-qarq90 \
-p 80:80 \
-v /home/ec2-user/my-website:/usr/share/nginx/html:ro \
nginx:latest
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
d107e437f729: Pull complete
cb497a329a81: Pull complete
f1c4d397f477: Pull complete
f72106e86507: Pull complete
899c83fc198b: Pull complete
a785b80f5a67: Pull complete
6c50e4e0c439: Pull complete
Digest: sha256:d5f28ef21aabddd098f3dbc21fe5b7a7d7a184720bc07da0b6c9b9820e97f25e
Status: Downloaded newer image for nginx:latest
822c2d64536dad0de270aca9c3331601329b9a51e78751211cfd597d810cc031
[ec2-user@ip-172-31-19-39 ~]$
```

The terminal window also shows a status bar at the bottom with system information: 0% CPU, 0.44 GB / 0.88 GB memory, 0.01 MB/s network, 28 min uptime, and 13-09-2025 date.

Deploying Container

The screenshot shows a terminal window with the following commands and output:

```
[ec2-user@ip-172-31-19-39 ~]$ docker stop exp9-nginx-qarq90
[ec2-user@ip-172-31-19-39 ~]$ docker rm exp9-nginx-qarq90
[ec2-user@ip-172-31-19-39 ~]$ docker run -d --name exp9-nginx-qarq90 -p 80:80 -v /home/ec2-user/qarq90:/usr/share/nginx/html:ro nginx:latest
afcf2af48e9955b76983984e4edec737dd2ceac14b21472234048503c0bc06b
[ec2-user@ip-172-31-19-39 ~]$ curl http://localhost:8080
curl: (7) Failed to connect to localhost port 8080 after 0 ms: Could not connect to server
[ec2-user@ip-172-31-19-39 ~]$ curl http://localhost
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>My Custom Website</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
<div class="container">
<header>
<h1>EXP9</h1>
</header>
<main>
<p>This website is running on Nginx inside a Docker container on AWS EC2!</p>
<div class="feature">
<h2>Features:</h2>
<ul>
<li>Containerized deployment</li>
<li>Custom styling</li>
<li>Scalable architecture</li>
</ul>
</div>
</main>
<footer>
<p>Deployed successfully on AWS EC2 by qarq90</p>
</footer>
</div>
</body>
</html>
[ec2-user@ip-172-31-19-39 ~]$
```

The terminal window also shows a status bar at the bottom with system information: 0% CPU, 0.43 GB / 0.88 GB memory, 0.01 MB/s network, 31 min uptime, and 13-09-2025 date.

Output of Deployed Container