**Lesson 03 Demo 01**

**Deploying MySQL Database as Docker Container**

**Objective:** To demonstrate how to deploy a MySQL database as a Docker container on an AWS EC2 instance with Ubuntu

**Tools required:** AWS Management Console

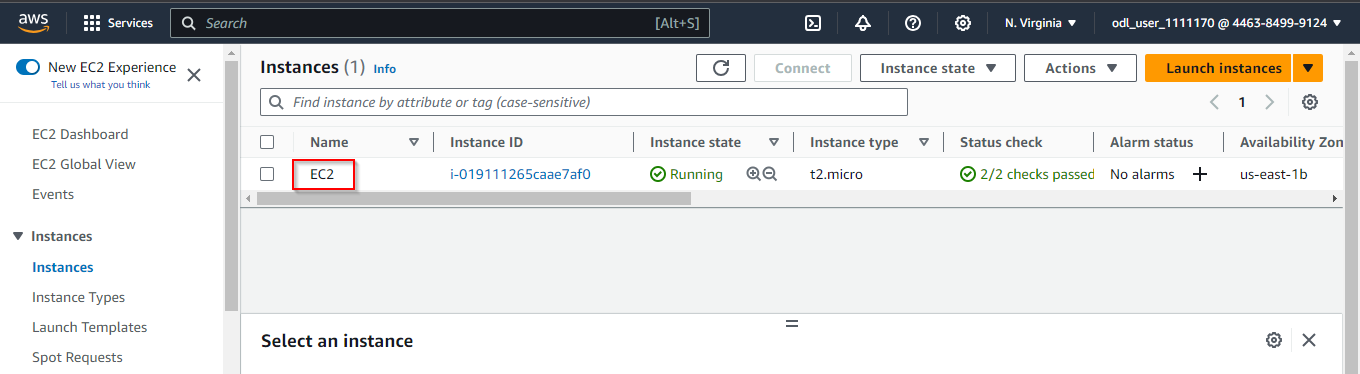
**Prerequisites:** An AWS EC2 instance with Ubuntu as the operating system

Steps to be followed:

1. Set up an EC2 instance
2. Install the Docker on Ubuntu
3. Run the MySQL container
4. Connect to MySQL server within container

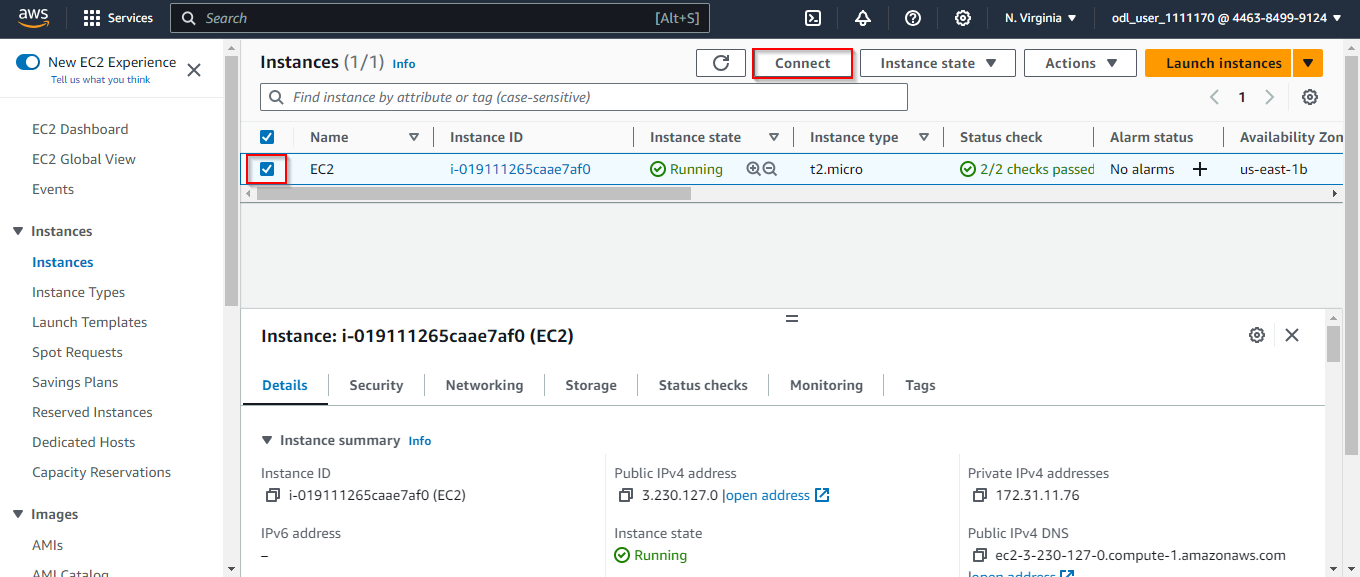
**Step 1: Set up an EC2 instance**

1. Launch a new EC2 instance with Ubuntu as the operating system. Ensure that you have the necessary security group rules to allow SSH access.



|  |
| --- |
| **Note:** Please refer to previous lesson demos on how to launch an EC2 instance. |

1. Select the instance and click **Connect**



1. Click **Connect** again to proceed

A screenshot of a computer

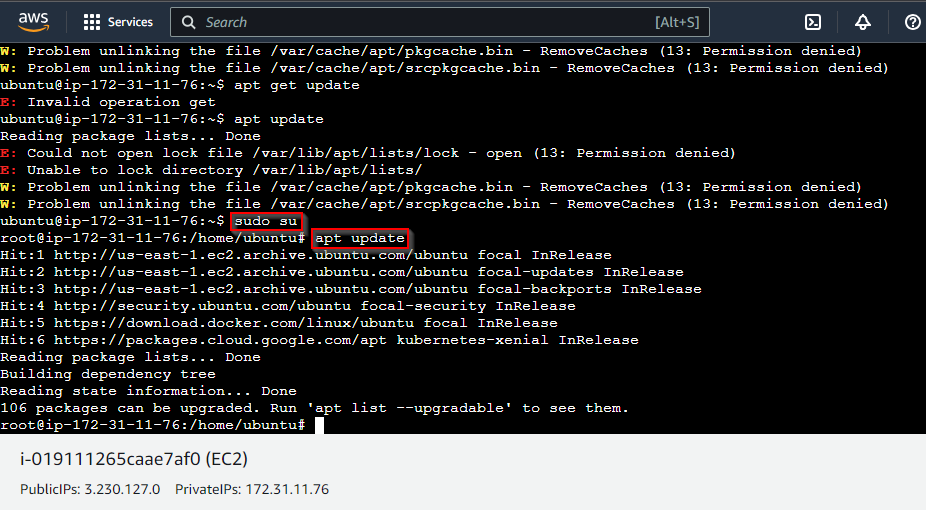
Description automatically generated

**Step 2: Install the Docker on Ubuntu**

1. Update the package list by running the following commands:

**sudo su**

**apt update**

****

1. Run the following command to install the necessary dependencies:

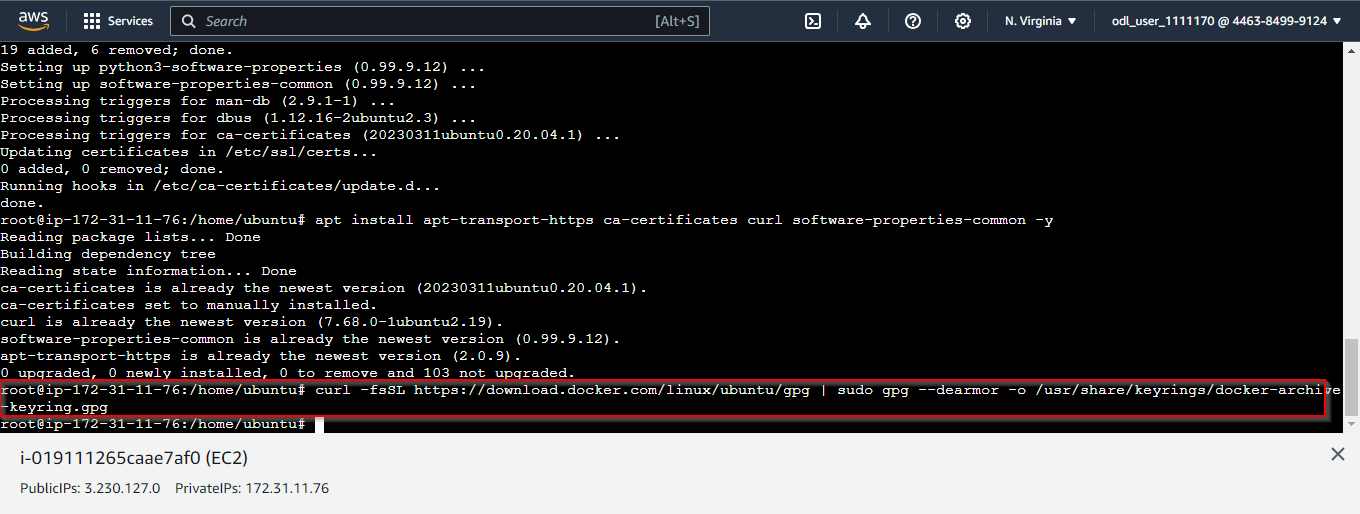
**apt install apt-transport-https ca-certificates curl software-properties-common -y**

**A screenshot of a computer program

Description automatically generated**

1. Run the following command to add the Docker GPG key:

**curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg**

****

1. Run the following command to add the Docker repository:

**echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null**

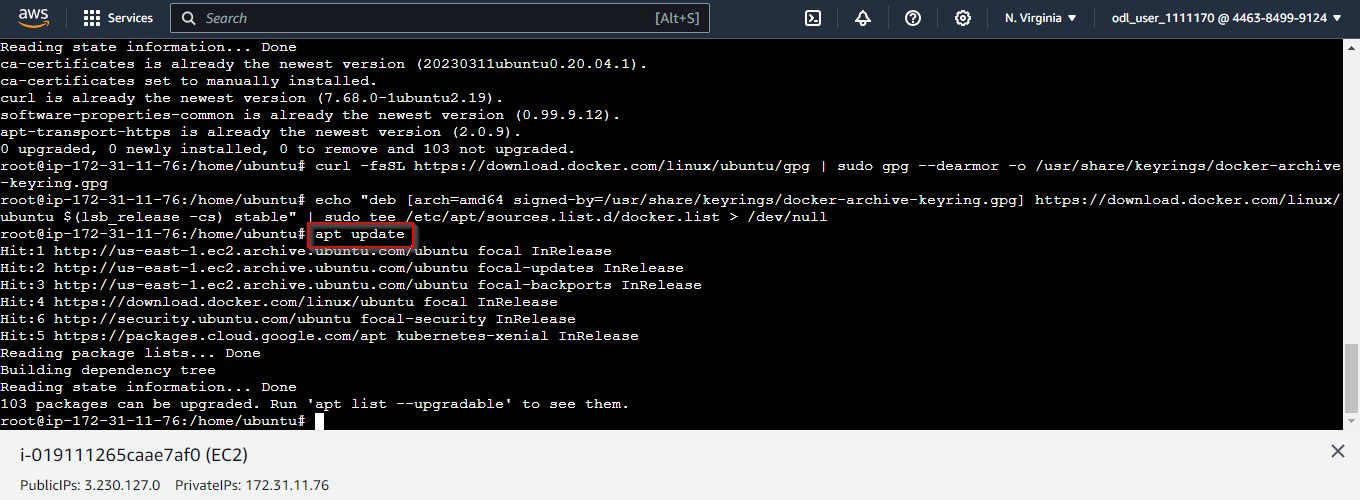
**A screenshot of a computer

Description automatically generated**

1. Run the following commands to install Docker:

**apt update**

**apt install docker-ce docker-ce-cli containerd.io**

****

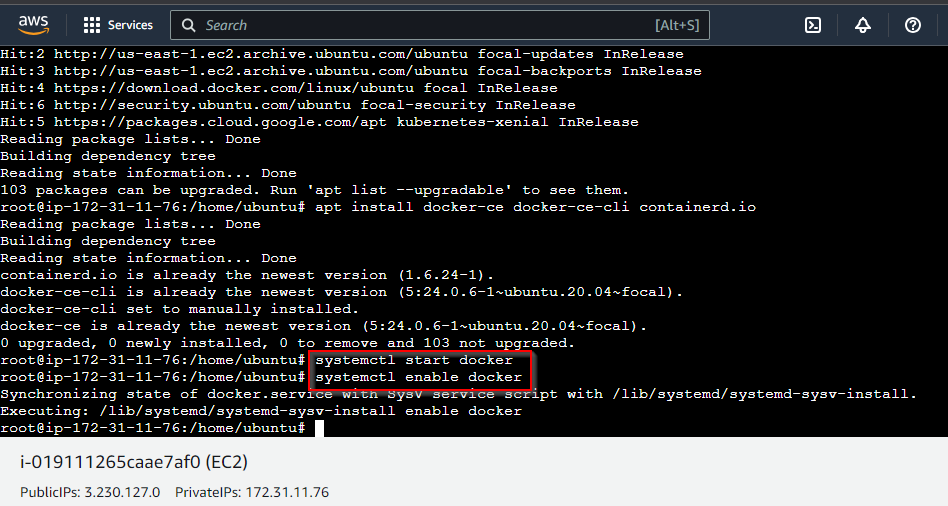
**A screenshot of a computer program

Description automatically generated**

1. Run the following commands to start and enable the Docker service:

**systemctl start docker**

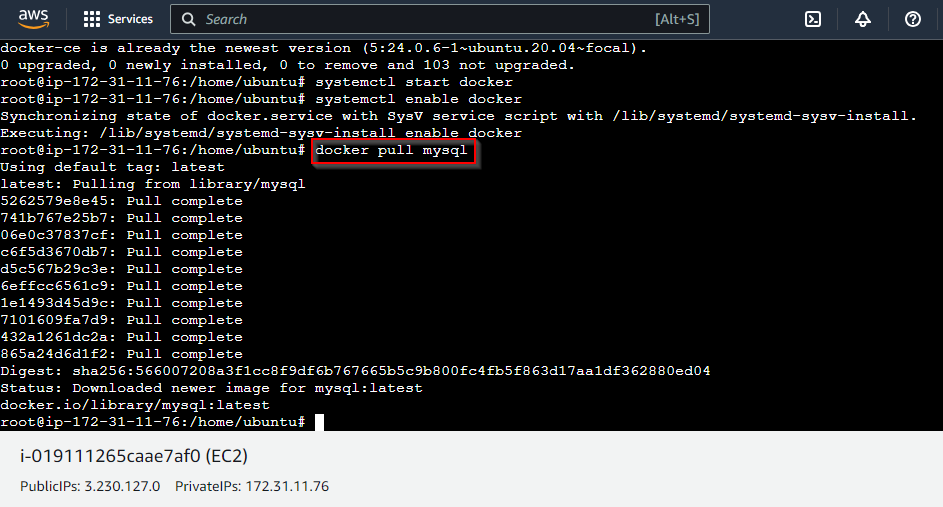
**systemctl enable docker**

****

**Step 3: Run the MySQL container**

1. Pull the latest MySQL image from Docker Hub using the following command:

**docker pull mysql**

****

1. Create a Docker volume or bind mount to store your database data persistently. For example, create a volume named **mysql-data**:

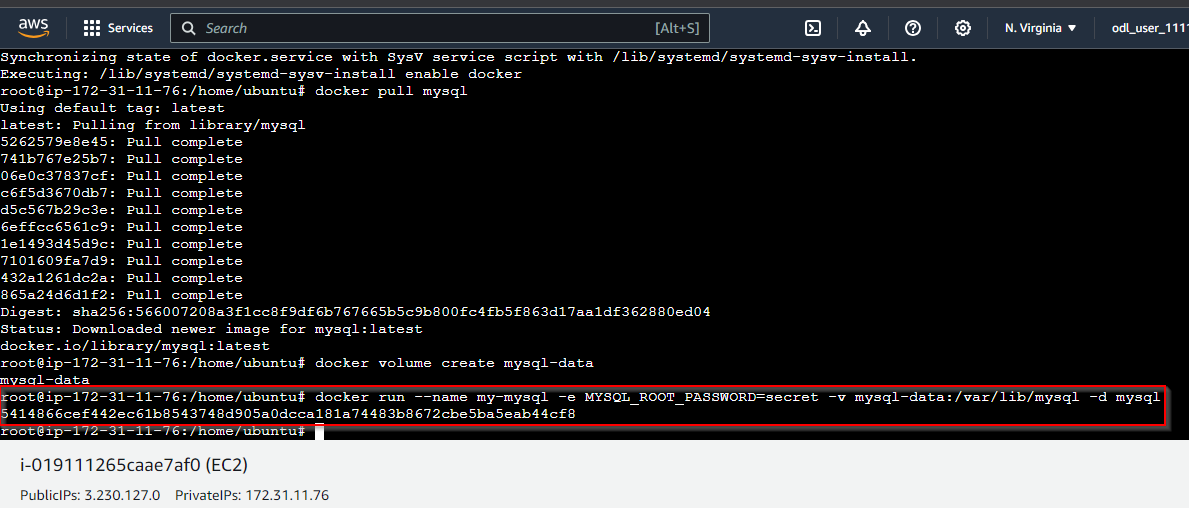
**docker volume create mysql-data**

**A screenshot of a computer

Description automatically generated**

1. Run a MySQL container and set a password for the root user using the **MYSQL\_ROOT\_PASSWORD** environment variable:

**docker run --name my-mysql -e MYSQL\_ROOT\_PASSWORD=secret -v mysql-data:/var/lib/mysql -d mysql**

****

1. Check the status of your container using the command:

**docker ps**

**A computer screen shot of text

Description automatically generated**

**Step 4: Connect to MySQL server within container**

1. Connect to your MySQL server from within the container:

**docker exec -it my-mysql mysql -p**

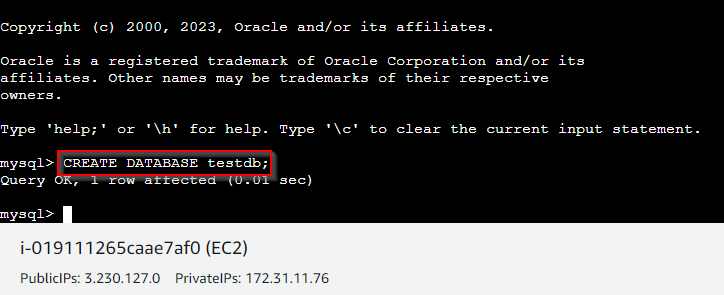
**A screenshot of a computer program

Description automatically generated**

|  |
| --- |
| **Note:** Use **secret** as the password |

1. Use the MySQL command-line interface to create databases, tables, and users and to perform queries. For example, create a database named **testdb**:

**CREATE DATABASE testdb;**

****

1. Exit the MySQL shell using the command:

**exit**

**A screenshot of a computer

Description automatically generated**

1. Expose container port 3306 to host port 33060 using the following commands:

**docker run --name my-mysql1 -e MYSQL\_ROOT\_PASSWORD=secret -v mysql-data:/var/lib/mysql -p 33060:3306 -d mysql**

**docker ps -a**

**A black screen with text

Description automatically generated**

By following these steps, you have successfully deployed a MySQL database as a Docker container on your AWS EC2 instance running Ubuntu.