**Task 14: Creating Containers, Pull & Push Images and Hosting Applications using Docker Commands**

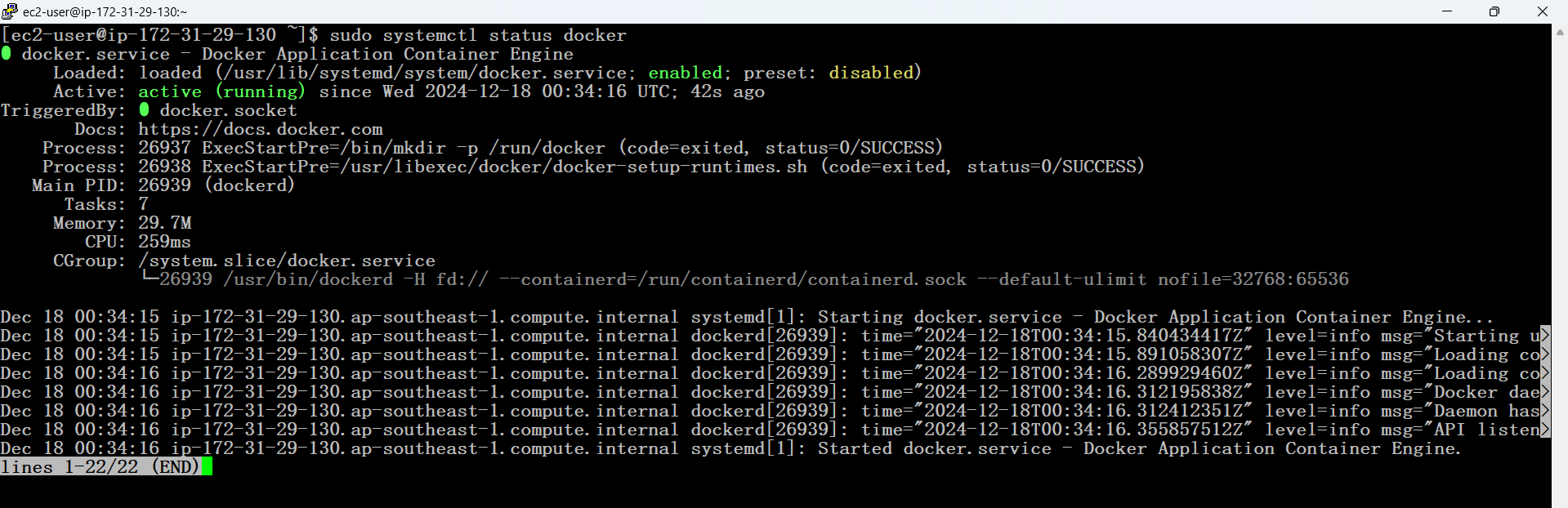
**Docker** is an open-source platform for developing, shipping, and running containerized applications. It provides tools and services to create and manage containers, making it one of the most popular platforms for containerization.

**Containerization:**

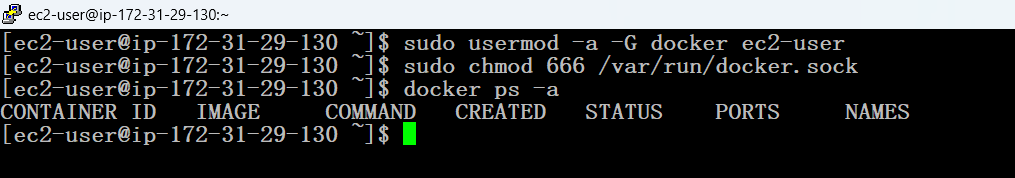
**Containerization** is a lightweight form of virtualization that allows applications to run in isolated environments, called containers. A container includes everything the application needs to run—such as the code, runtime, libraries, and system tools—without requiring an entire virtual machine (VM). Containers share the host operating system's kernel but remain independent of one another, ensuring portability and consistency across development, testing, and production environments.

**Docker Basic Commands:**

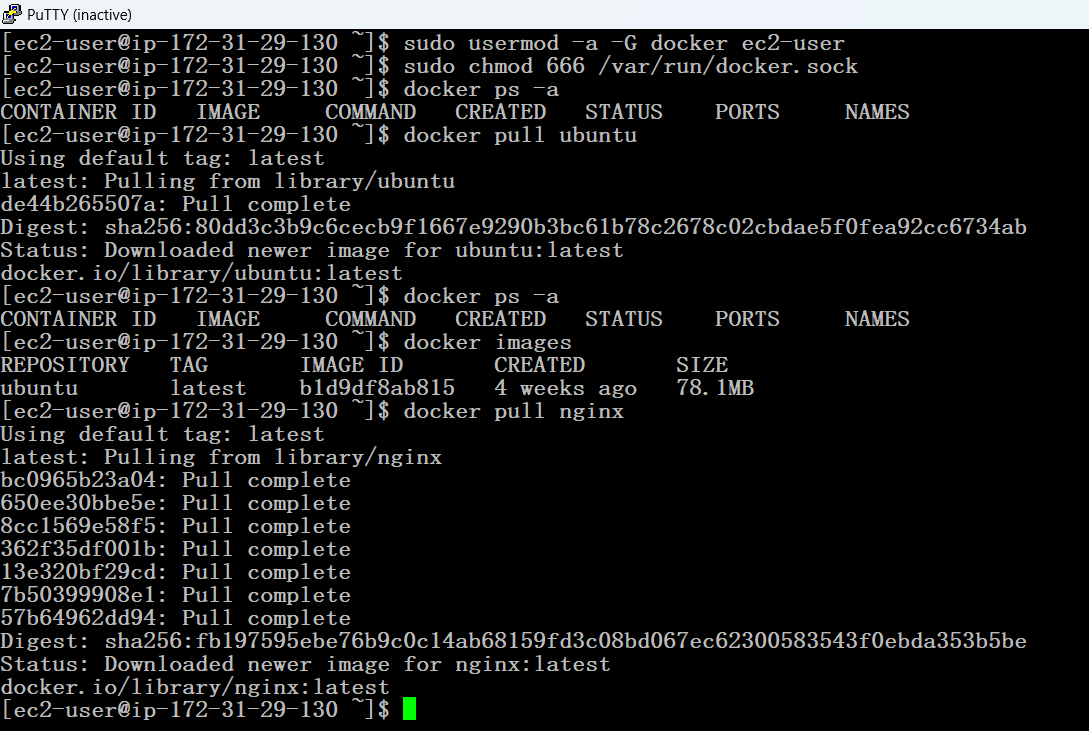
1. Launched an Ec2 Instance, Installed Docker and Service started on the EC2



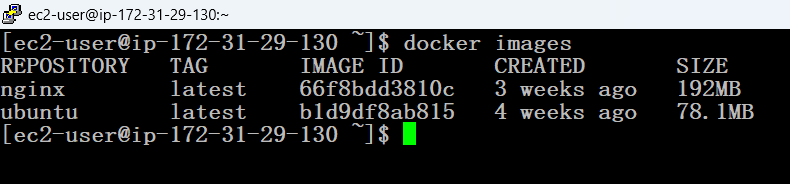
1. Added Sudo user in the Docker User Group & Provide Permission



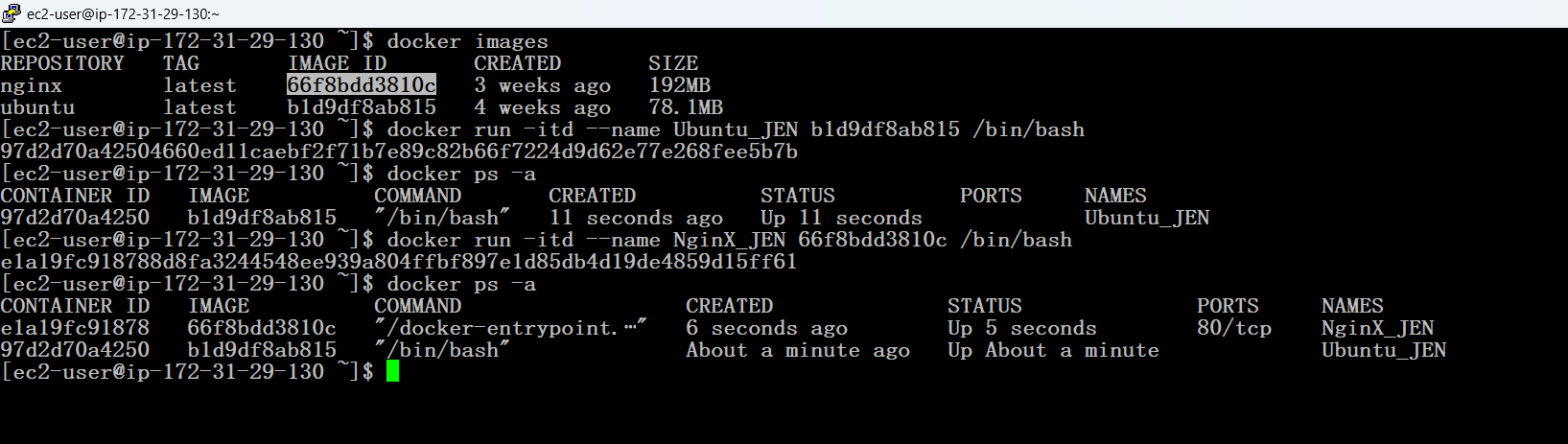
1. Ubuntu & Nginx Application Images Pulled form Docker Hub



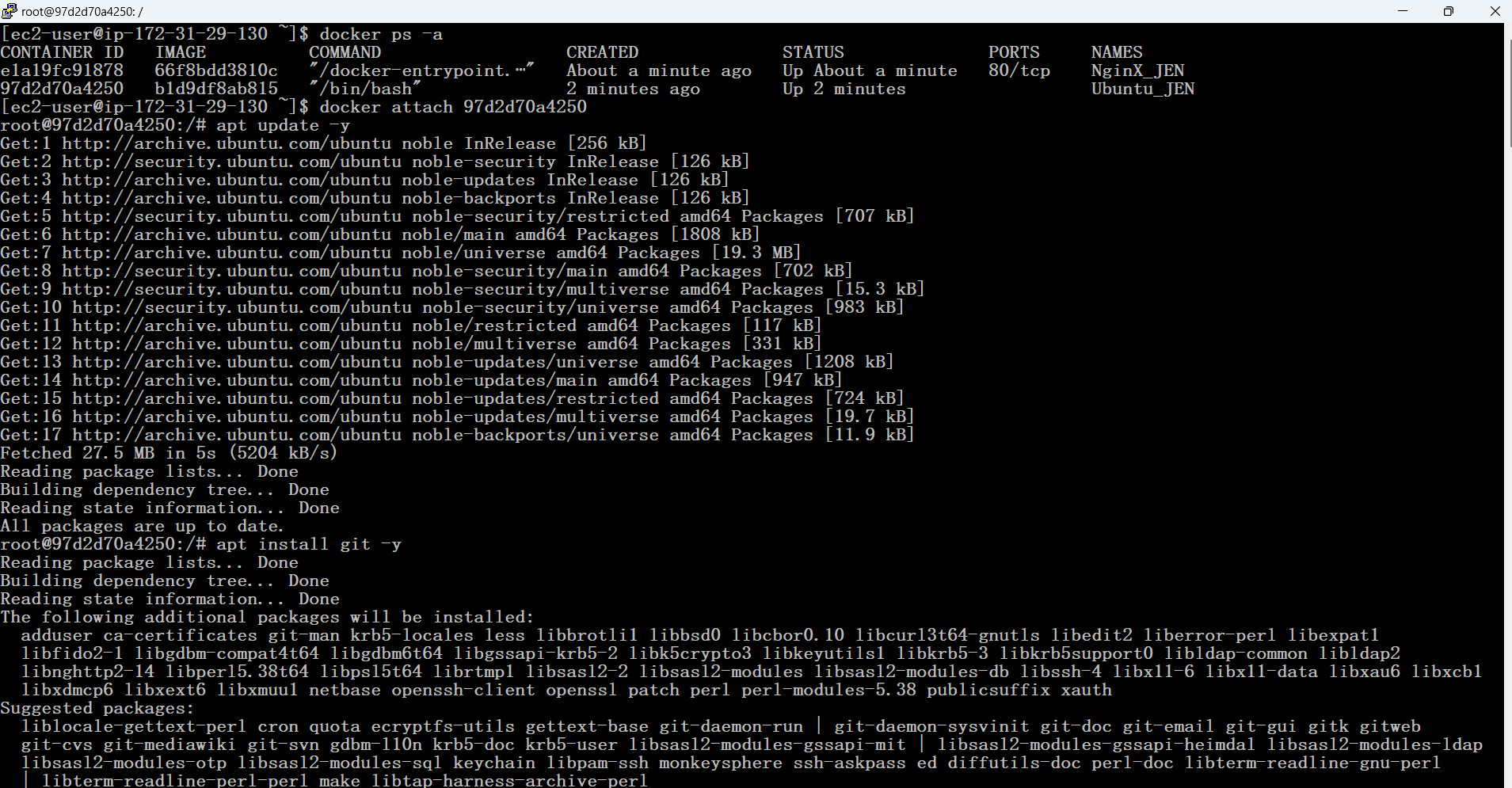
1. Listed out the Docker Images



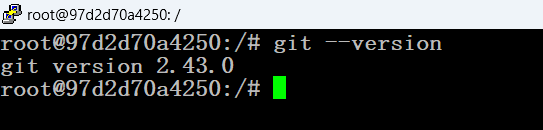
1. Created 2 Containers Based on the Pulled Images



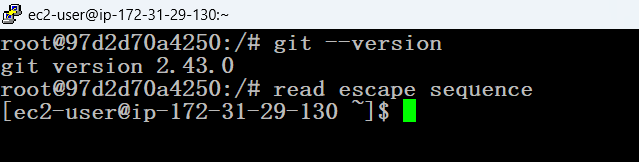
1. Getting into the container & Git installed in the container



1. Git Version Checking

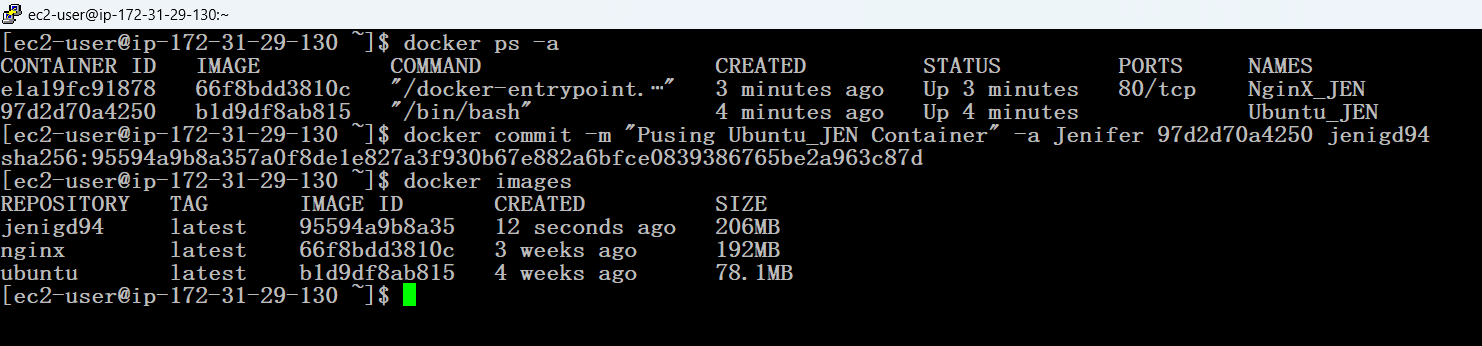


1. Exit from the Current Running Container

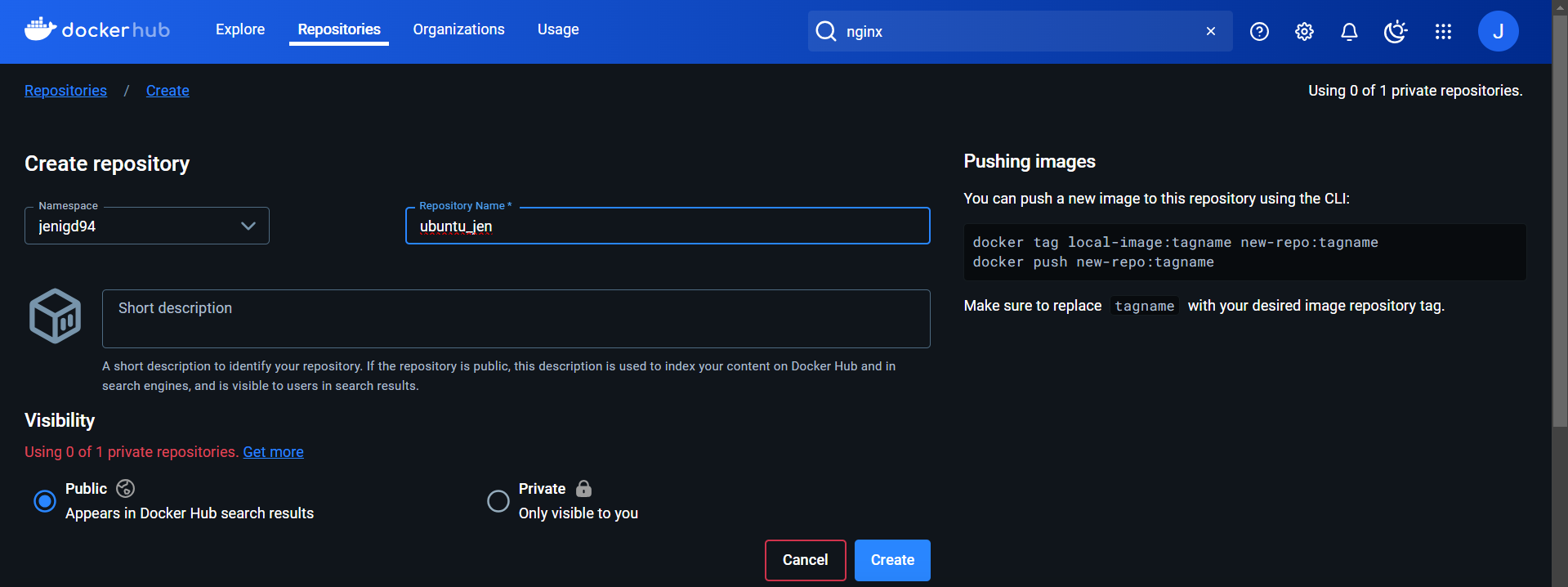


**Pull & Push Images:**

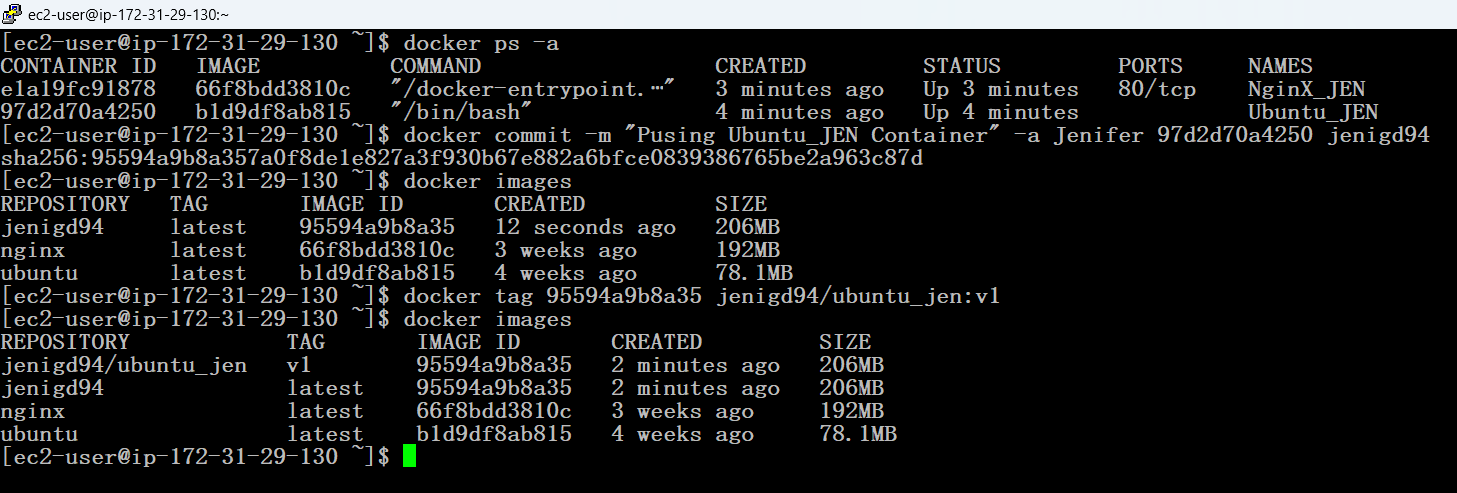
1. Creating an Image from the Container and Pushing it into the Docker Hub



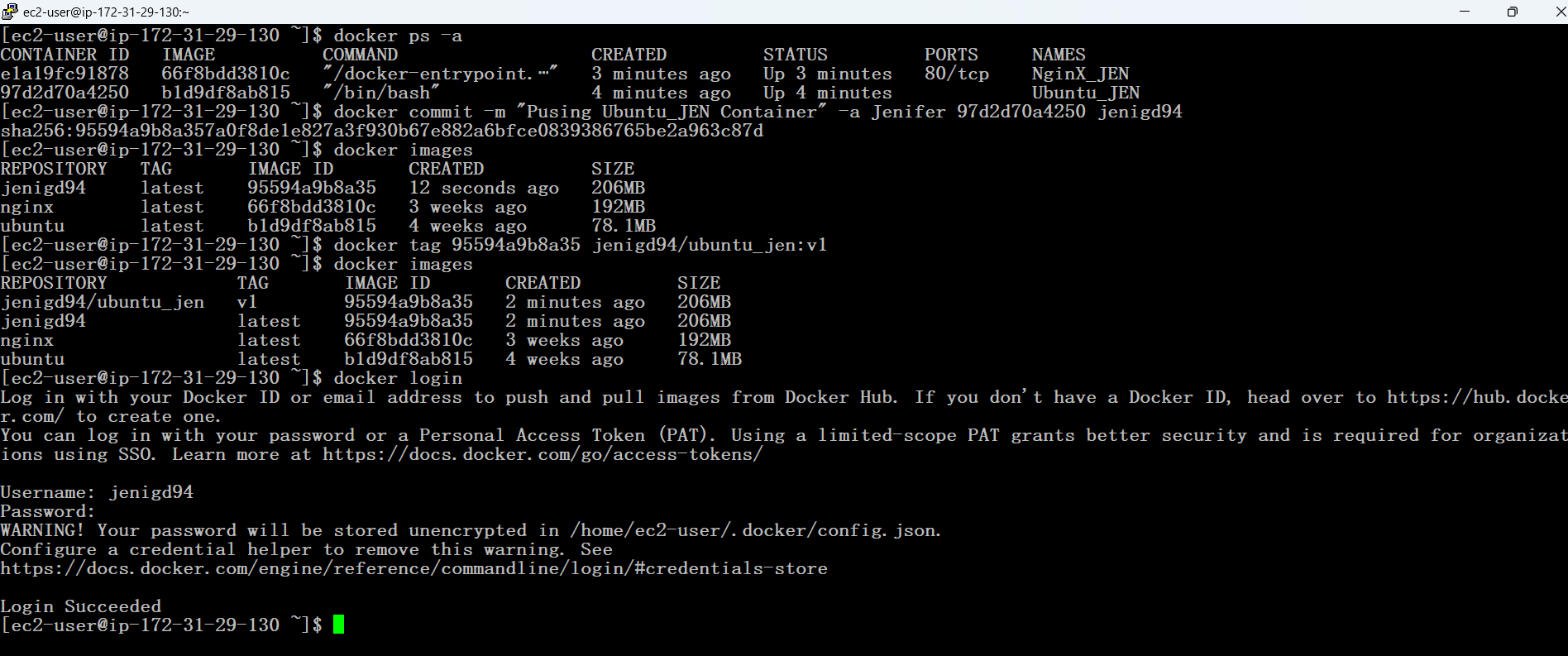
1. Created a new repository in Docker Hub



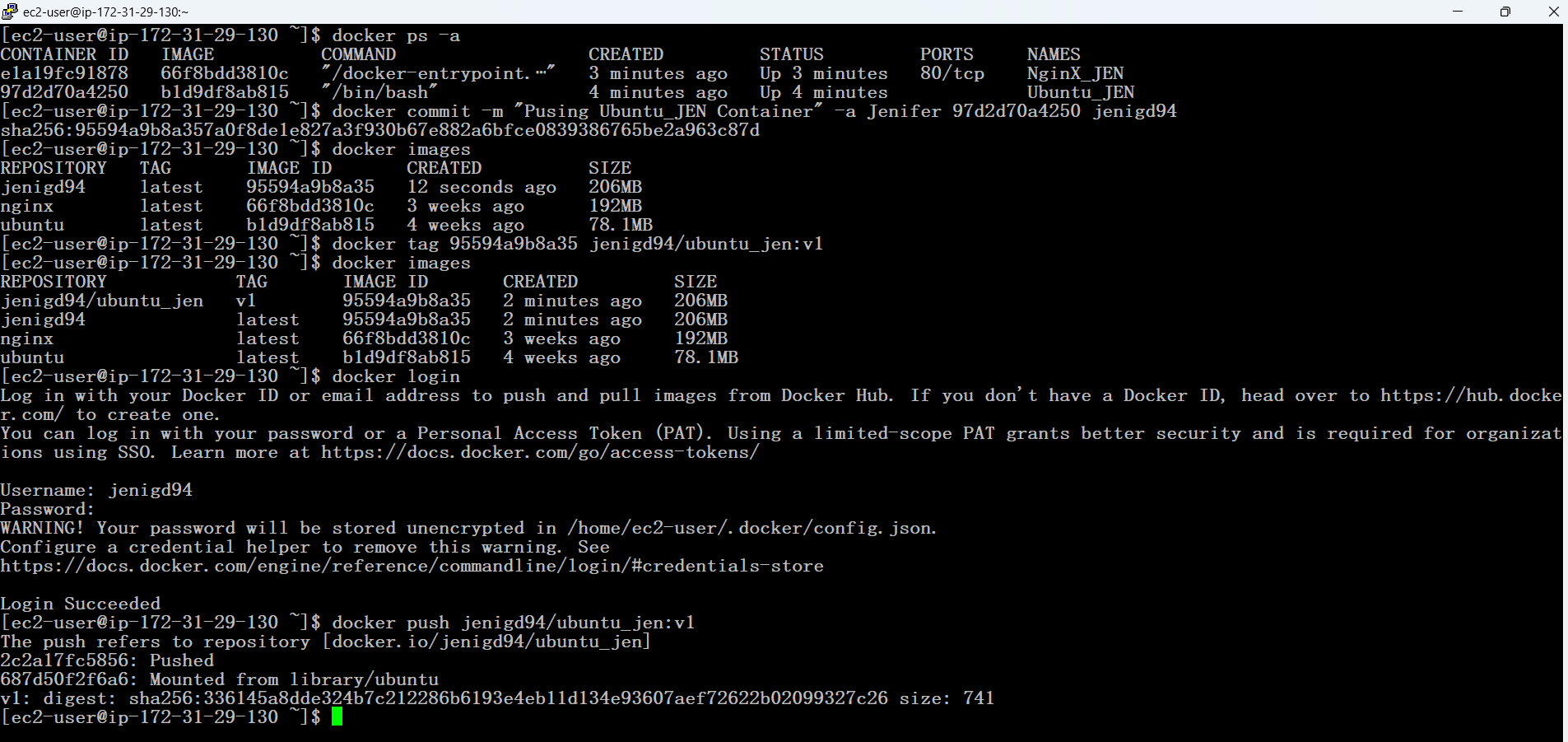
1. Created a new image



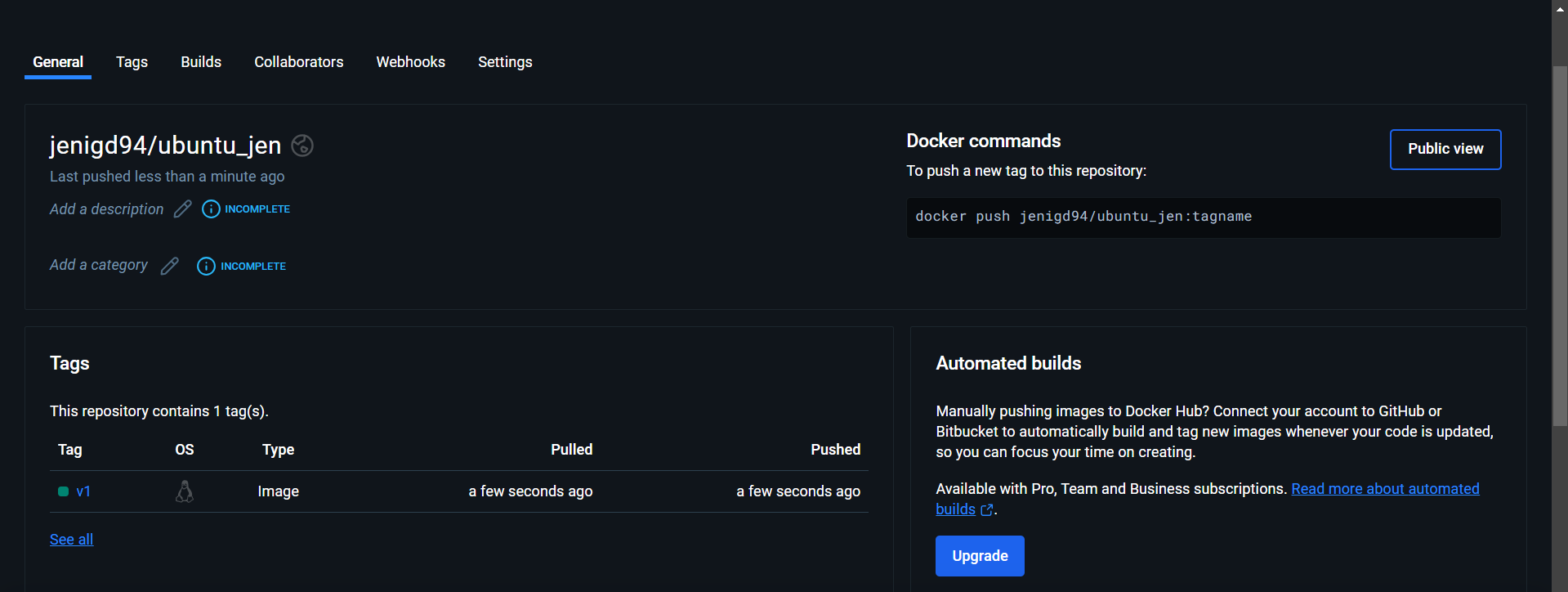
1. Logged into the Docker Hub by providing Username & Password



1. Image Successfully Pushed into the Docker Hub



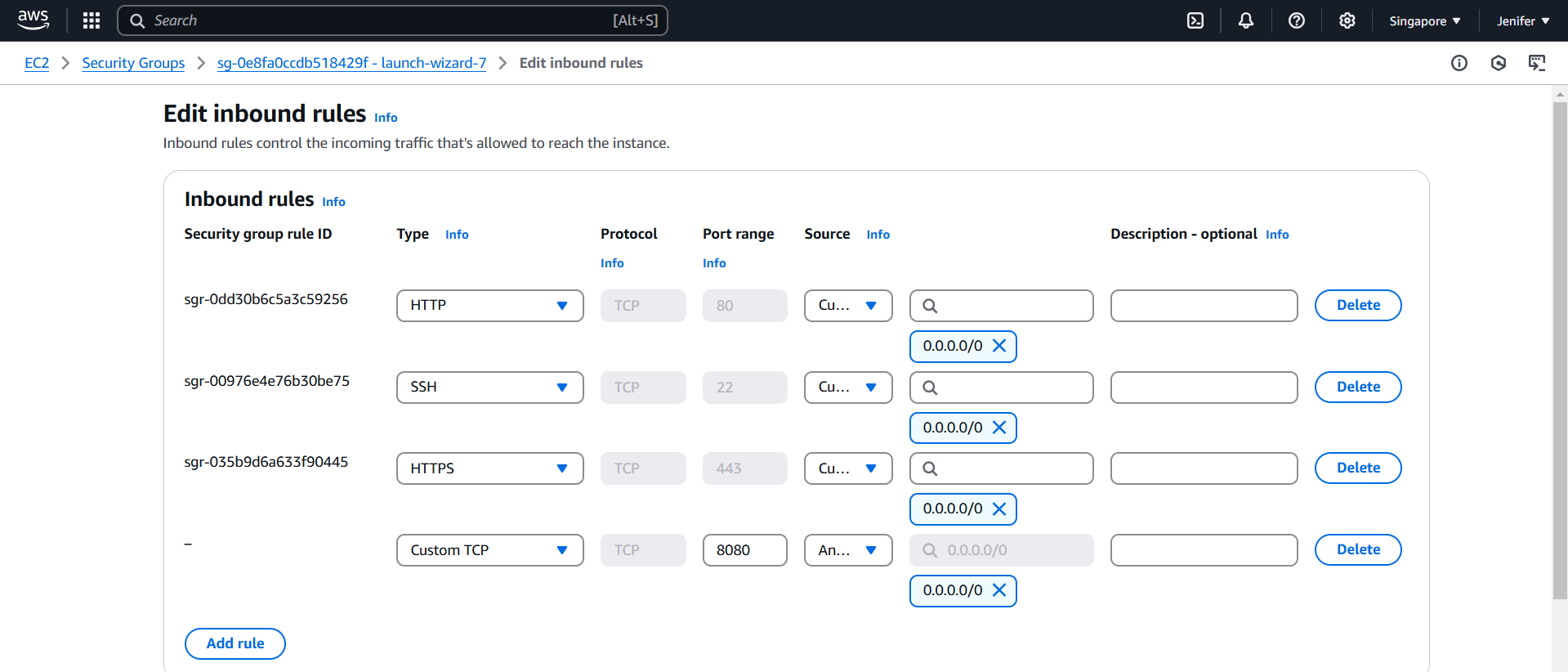
1. The Pushed Image successfully reflected in the Docker Hub

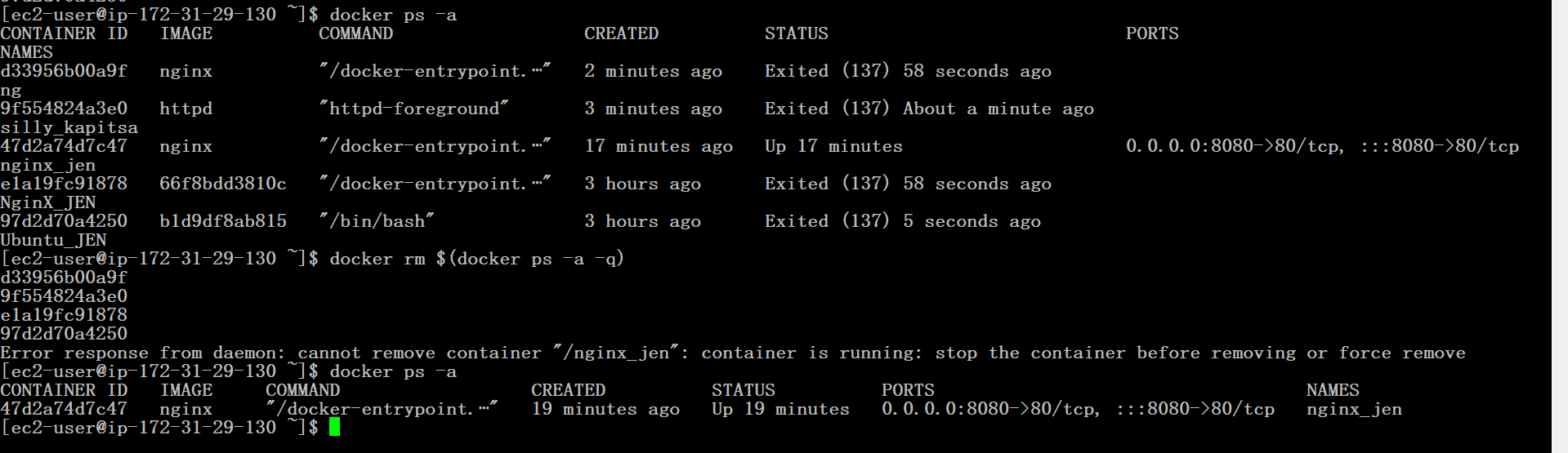


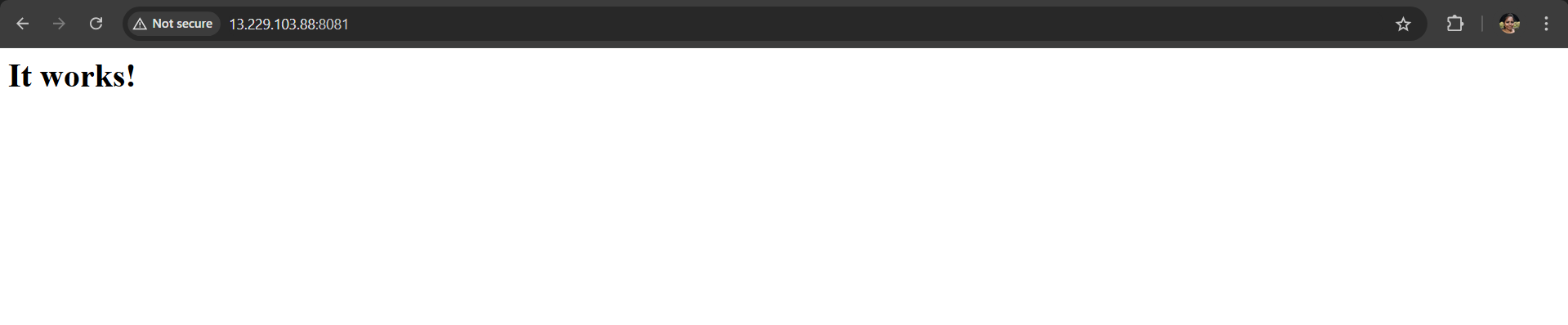
**Hosting Application:**

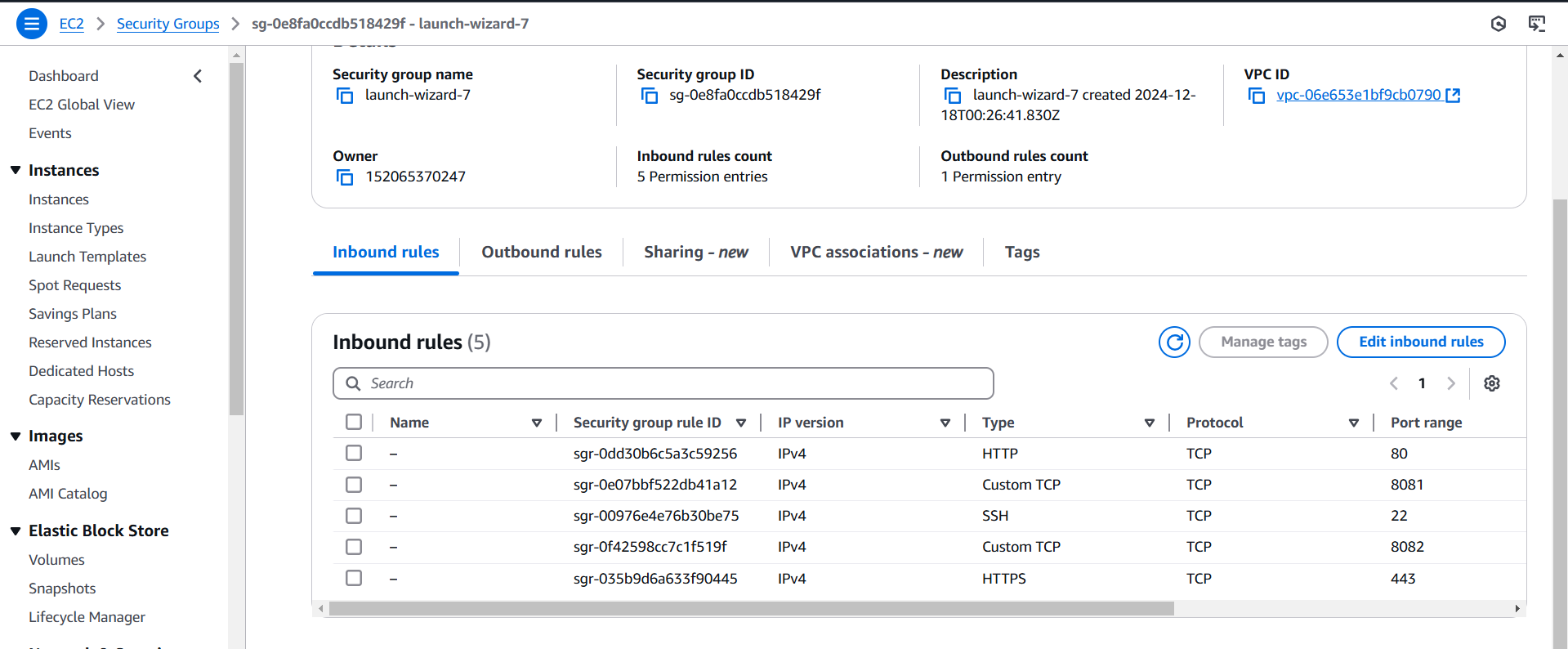
**Port Forwarding in Docker** is the process of mapping a port on the host machine to a port inside a Docker container. This allows external systems to communicate with the services running inside the container. It's essential when running containerized applications that need to be accessible to users or other systems, such as web servers, databases, or APIs.

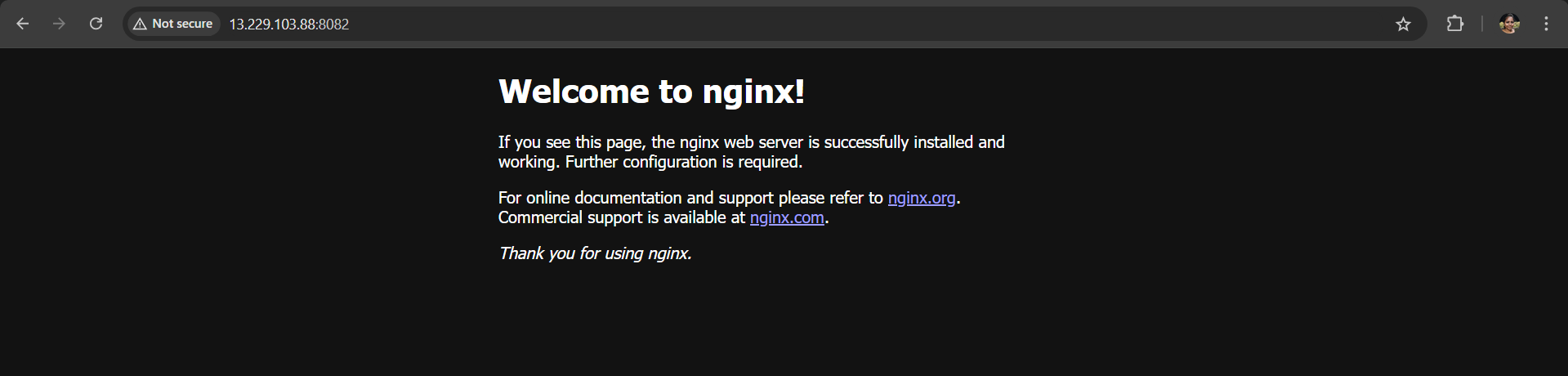
1. Updated the SG for external system communication



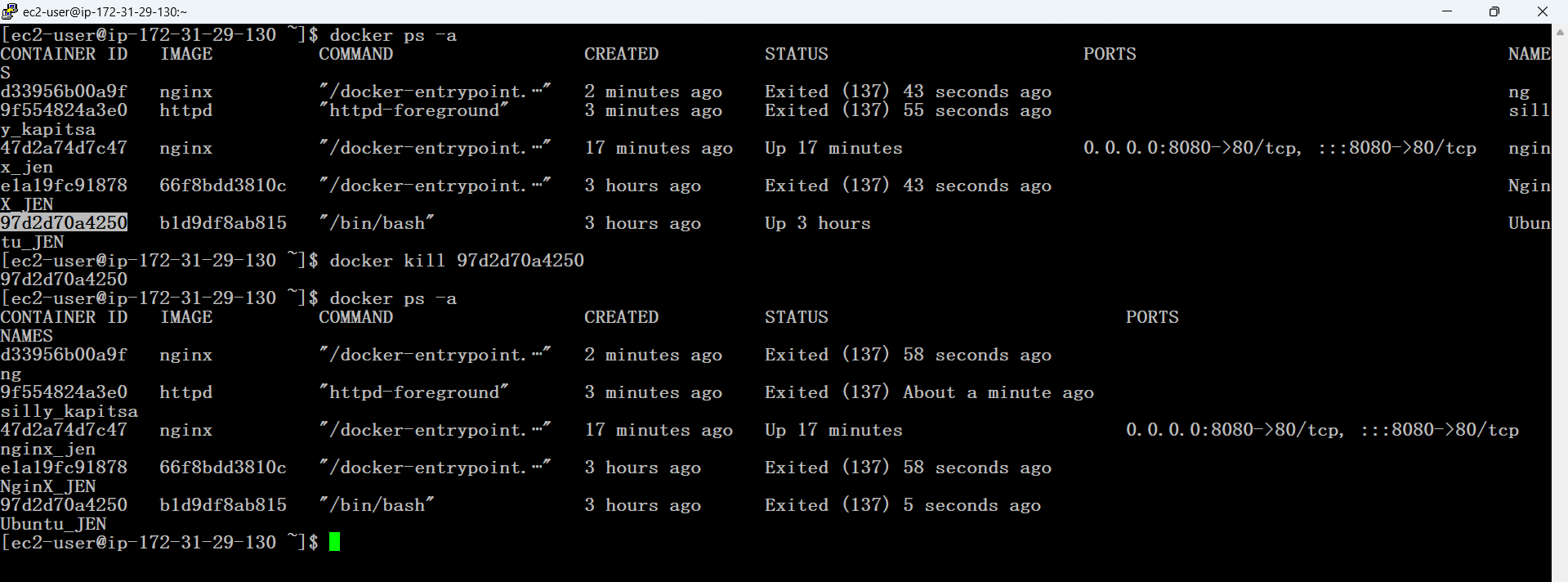


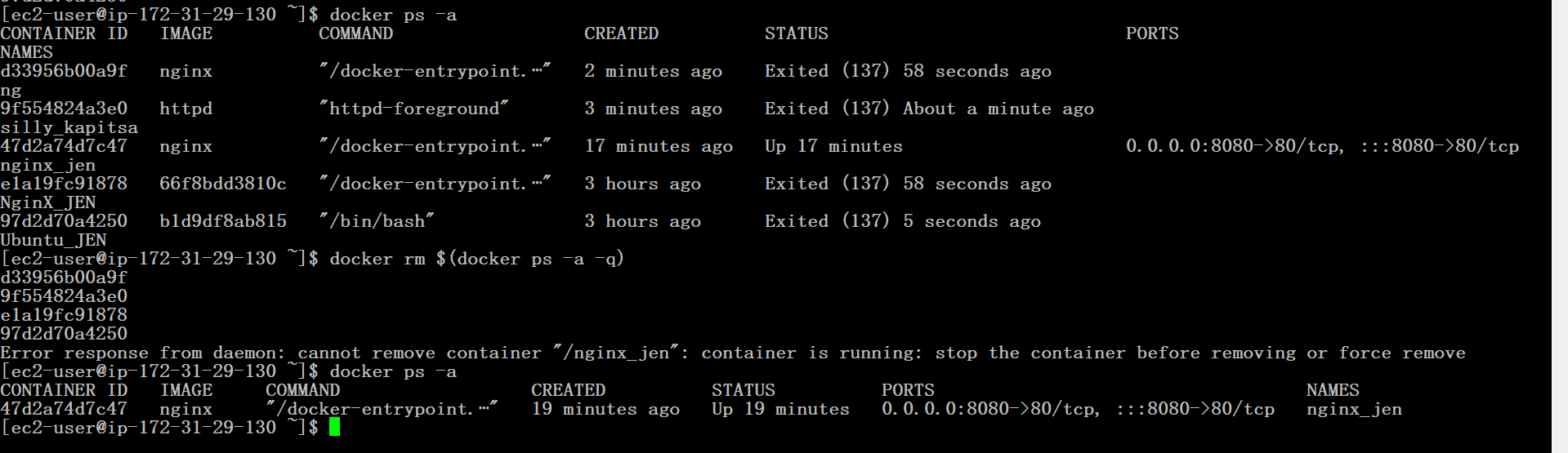






1. Exiting all the Containers and removing them





1. Removing all the Images

