Namespace labs

**Lab Setup: Understanding Docker Namespaces**

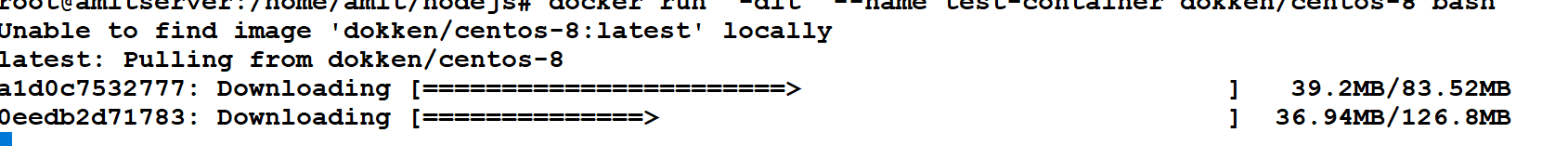
**Prerequisites**

* A Linux machine with **Docker** installed
* Basic knowledge of Linux commands

**Step 1: Run a New Docker Container**

Run a new Ubuntu container in detached mode:

docker run -dit --name test-container dokken/centos-8 bash



This starts a new **Centos container** with a separate namespace.

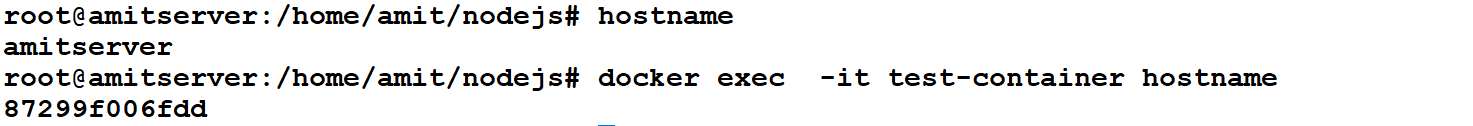
**Step 2: Check Host and Container Hostname (UTS Namespace)**

On the **host machine**, check the hostname:

hostname

Now, **inside the container**, check the hostname:

docker exec -it test-container hostname



**Step 3: Check Network Namespace Isolation**

On the **host machine**, check the IP:

ip a|grep ens

Now, **inside the container**, check the IP:

docker exec -it test-container ip addr show

**Step 4: Check Process Namespace (PID Namespace)**

Run this command to find the process ID of the container's bash:

docker exec -it test-container ps aux

You will see that the **PID inside the container is different from the host**, proving that **PID Namespace** is isolated.

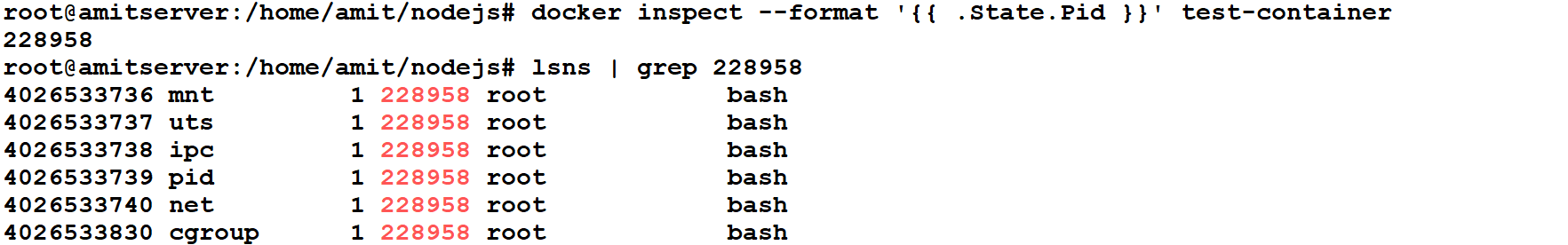
**Step 5: View Namespaces of a Process (lsns Command)**

On the host machine, find the **PID of the running container**:

docker inspect --format '{{ .State.Pid }}' test-container

Use the lsns command to see all namespaces associated with that process:

lsns | grep <PID>

****

**Kill the Process and Observe Container Removal**

Now, find the **main PID** of the container process and kill it:

kill <PID>

Check if the container is running:

docker ps

