**Lab: Managing Docker Containers  
Name**: Jyotsna Santosh Mali

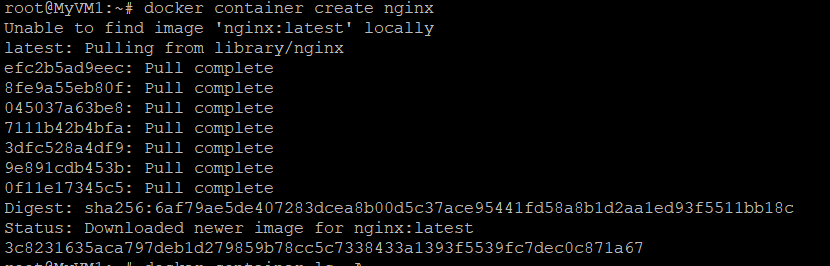
**1.1** Let’s list the containers if any by executing the below command.

*# docker container ls -a*

**Output:  
  
**

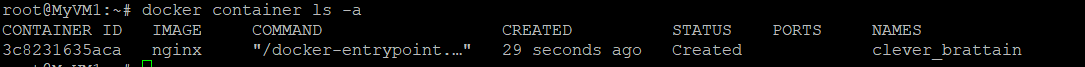
**1.2** Let’s create a nginx container by executing the below command.

*# docker container create nginx*

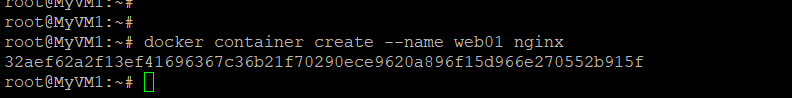
**Output:  
**

**1.3 Executing the below command to list container created from the above step.***# docker container ls -a*

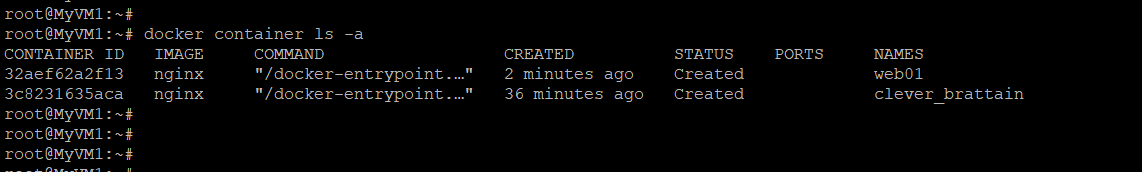
**Output:**

****

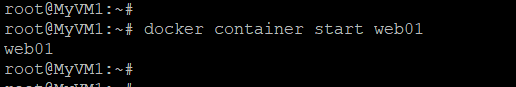
**1.4** Let’s create a container and name it as web01 by executing the below command**.***# docker container create --name web01 nginx*

**Output:  
**

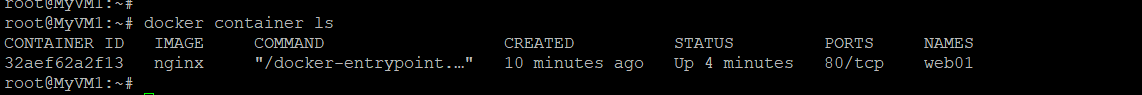
**1.5** Let’s list the container created by executing the below command.   
  
*# docker container ls -a*

**Output:  
**

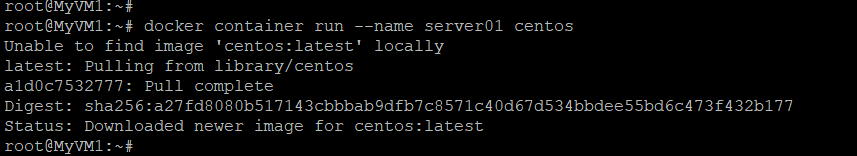
**1.6** Let’s start the container which is created by executing the below command.*# docker container start web01*

**Output:  
**

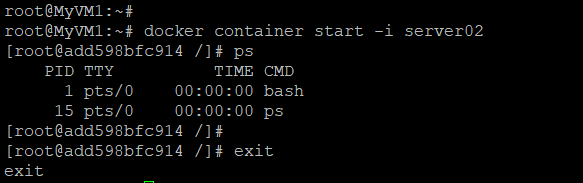
**1.7** Let’s list the container created by executing the below command.*# docker container ls -a*

**Output:  
**

**1.8** Let’s create a new container using run option by executing the below command.  
  
*# docker container run --name server01 centos*

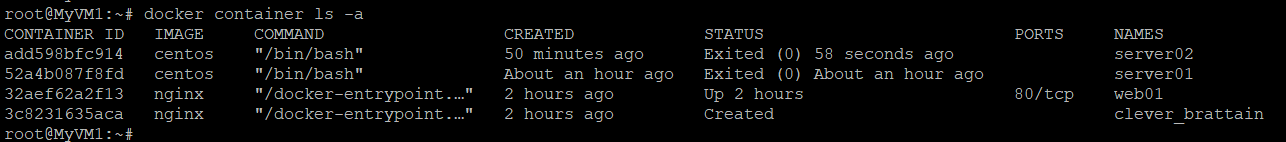
**Output:  
**

**1.10** Let’s create a new container by supplying (-i and -t) option to get interactive terminal by executing the below command.   
  
*# docker container run --name server02 -i -t centos***Output:**

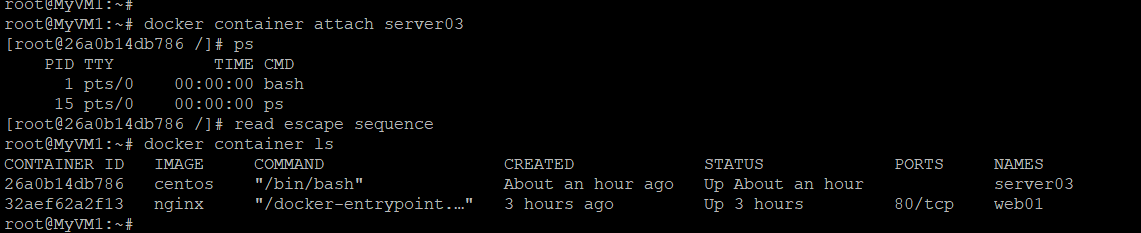


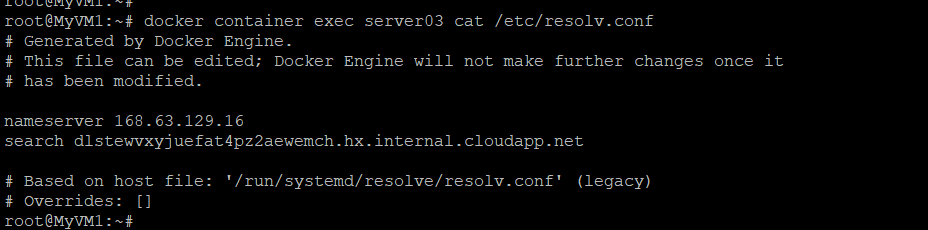
**1.11** Let’s list the container by executing the below command.

*# docker container ls -a*  
**Output:**

****

**1.12** Let’s create a new container with detached interactive terminal (-d -i -t) option, by executing the below command.

*# docker container run --name server03 -dit centos*Let’s attach the container server03 by executing the below command*.  
  
# docker container attach server03***Output:****

Let’s run a command inside the container without attaching to the container by using **exec** option   
  
*# docker container exec server03 cat /etc/resolv.conf***Output:**  


Let’s **rename** the existing container by executing the below command.

*# docker container rename server03 server003*

Let’s **list** the container by executing the below command.   
  
*# docker container ls*Let’s **pause** the existing container by executing the below command. *# docker container pause server003*

Let’s **list** the container by executing the below command.

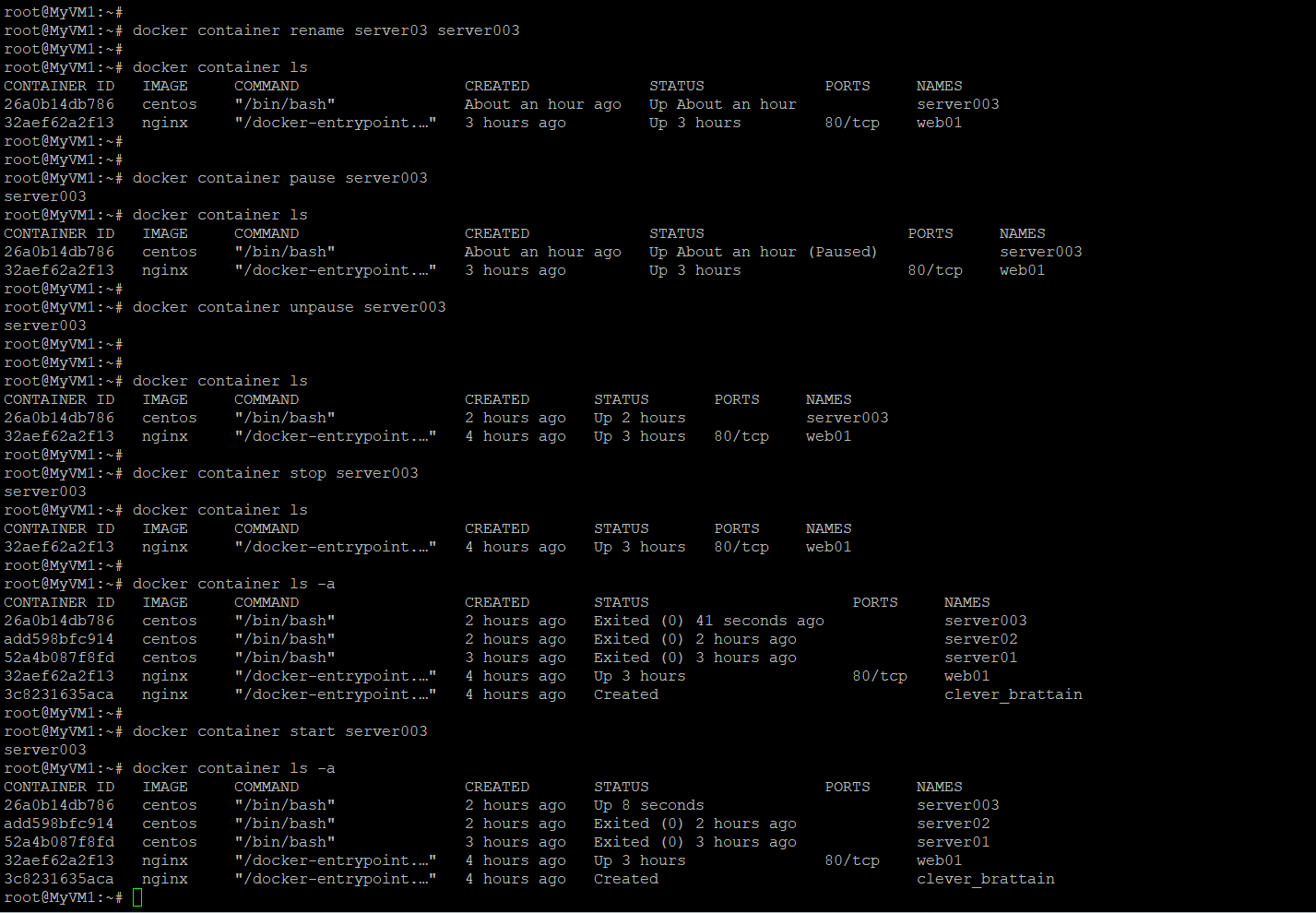
*# docker container ls*Let’s **unpause** the existing container by executing the below command.  
  
*# docker container unpause server003*  
  
Let’s **list** the container by executing the below command.

*# docker container ls*

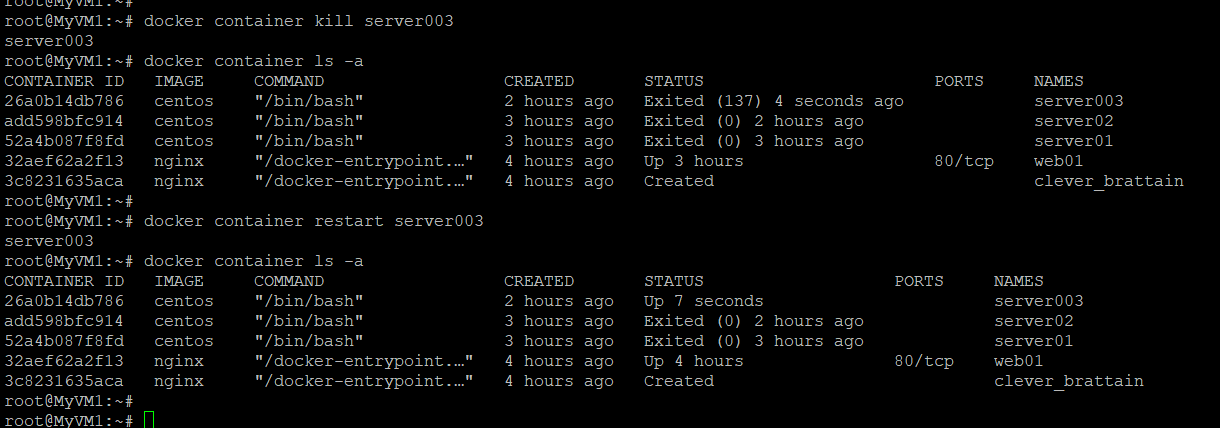
Let’s **stop** the existing container by executing the below command.  
  
*# docker container stop server003*

Let’s **list** the container by executing the below command.

*# docker container ls -a*Let’s **start** the existing container by executing the below command.  
  
*# docker container start server003*  
  
Let’s **list** the container by executing the below command.  
  
*# docker container ls -a*

**Output:**

Let’s **kill** the container by executing the below command.

*# docker container kill server003*Let’s **restart** the existing container by executing the below command.   
  
*# docker container restart server003*Let’s **list** the container by executing the below command. *# docker container ls -a***Output:** **

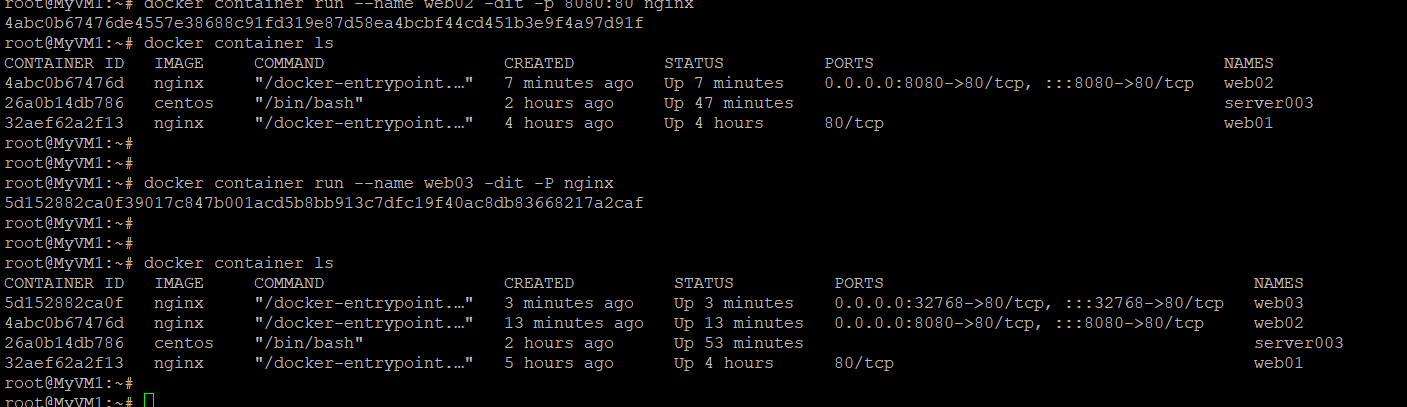
Let’s expose the **custom port** to the container by executing the below command.

*# docker container run --name web02 -dit -p 8080:80 nginx*Let’s **list** the container by executing the below command   
  
*# docker container ls*

Let’s expose the **random host port** to the container by executing the below command.

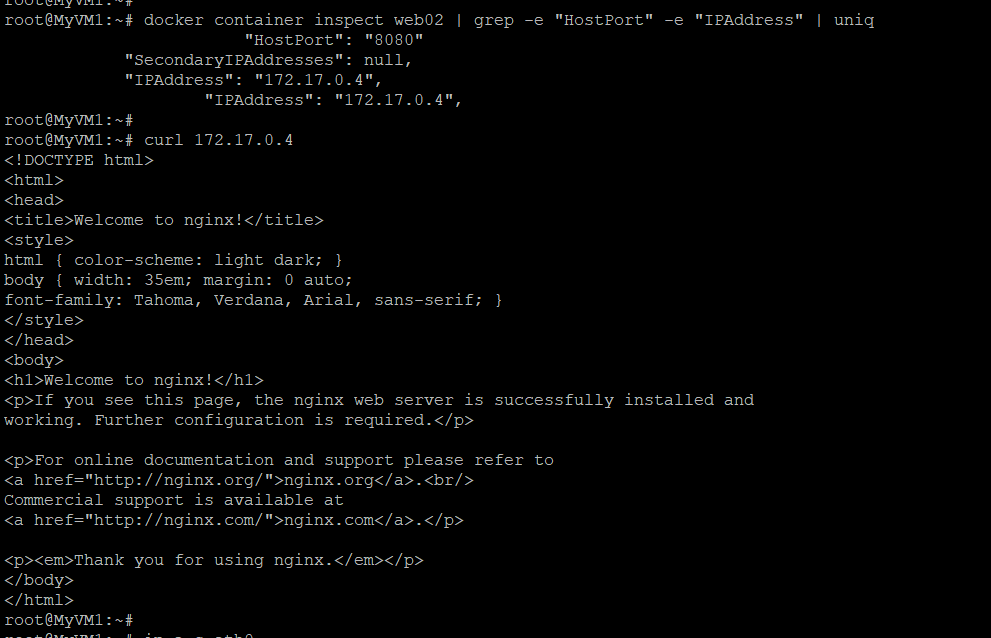
*# docker container run --name web03 -dit -P nginx*

Let’s **list** the container by executing the below command.   
  
*# docker container ls*

**Output:**

Let’s **inspect** the container by executing the below command.

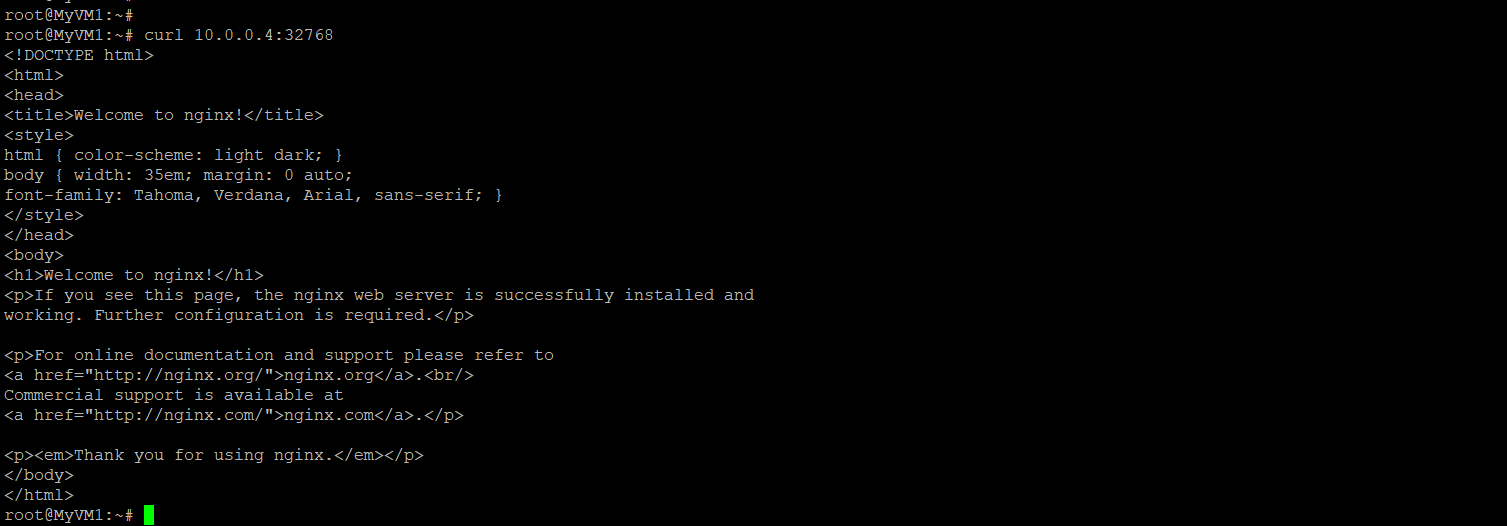
*# docker container inspect web02 | grep -e "HostPort" -e "IPAddress" | uniq*Let’s **access** the webserver by using the container ip by executing the below command.   
  
*# curl 172.17.0.2***Output:**

**

Let’s **access** the webserver by using the docker host ip and port exposed by executing the below command

*# curl* ***192.168.100.10****:8080  
*

Let’s **access** the webserver by using docker host ip and port exposed by executing the below command.

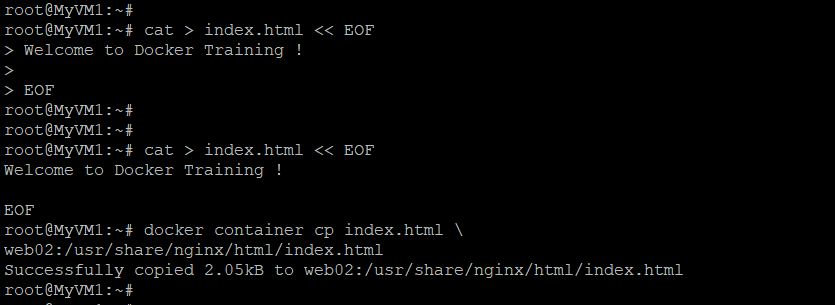
*# curl 192.168.100.10:32768  
*

Let’s **create** an **index.html** file by executing the below command.

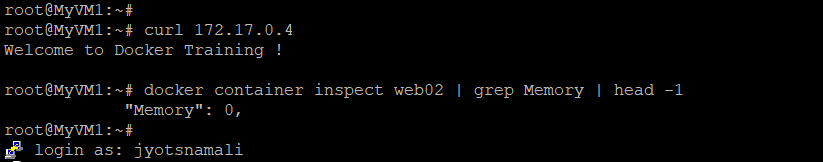
*# cat > index.html << EOF*

*Welcome to Docker Training!!  
EOF*Let’s **copy** files between local filesystem and a container by executing the below command. ex: we are copying the above created file inside a web02 container.

*# docker container cp index.html \*

*web02:/usr/share/nginx/html/index.html***Output:** **Let’s **access** the webserver by using the container IP by executing the below command.

*# curl 172.17.0.2*Let’s **update** memory configuration of the container first let us **inspect** the default memory limit set by executing the below command.

*# docker container inspect web02 | grep Memory | head -1***Output:** **

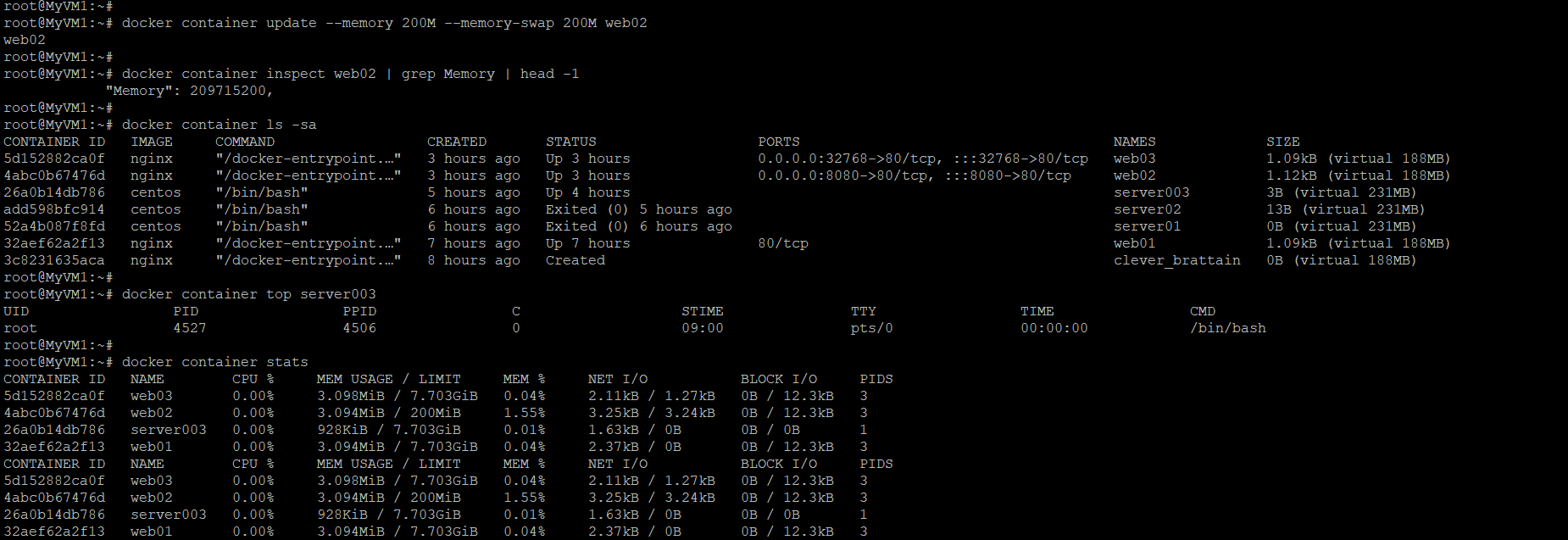
Let’s **update** the **memory** and memory-swap by executing the below command.

*# docker container update --memory 200M --memory-swap 200M web02*Let’s **inspect** the container and validate if the values are updated correctly by executing the below command  
  
*# docker container inspect web02 | grep Memory | head -1*Let’s check the **size** of the container by executing the below command.

*# docker container ls -sa*Let’s verify the **top running process** inside the container by executing the below command.

*# docker container top server003*Let’s verify the **stats** of the running containers by executing the below command.

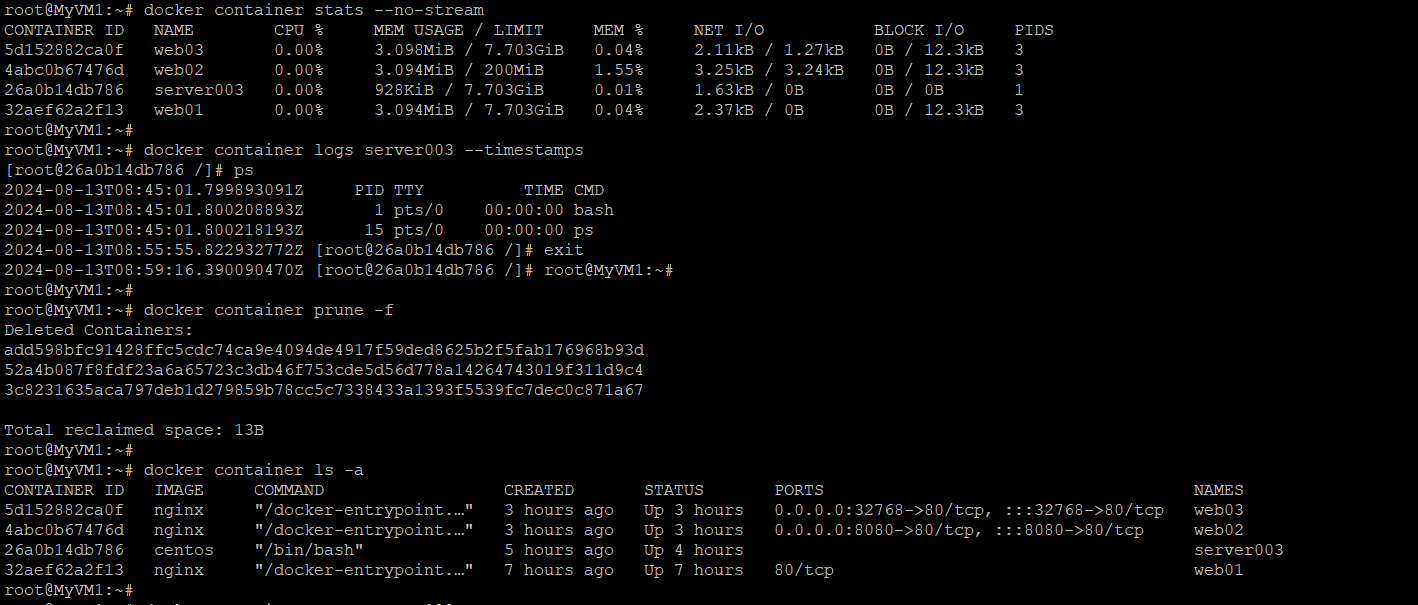
*# docker container stats***Output:**

**

*# docker container stats --no-stream*Let’s verify the **logs** of the containers by executing the below command.   
  
*# docker container logs server003 –timestamps*Let’s **prune** all the stopped containers by executing the below command.

*# docker container prune -f*

Let’s **list** the container by executing the below command.

*# docker container ls -a  
*Let’s **remove** the server003 container gracefully by executing the below command.

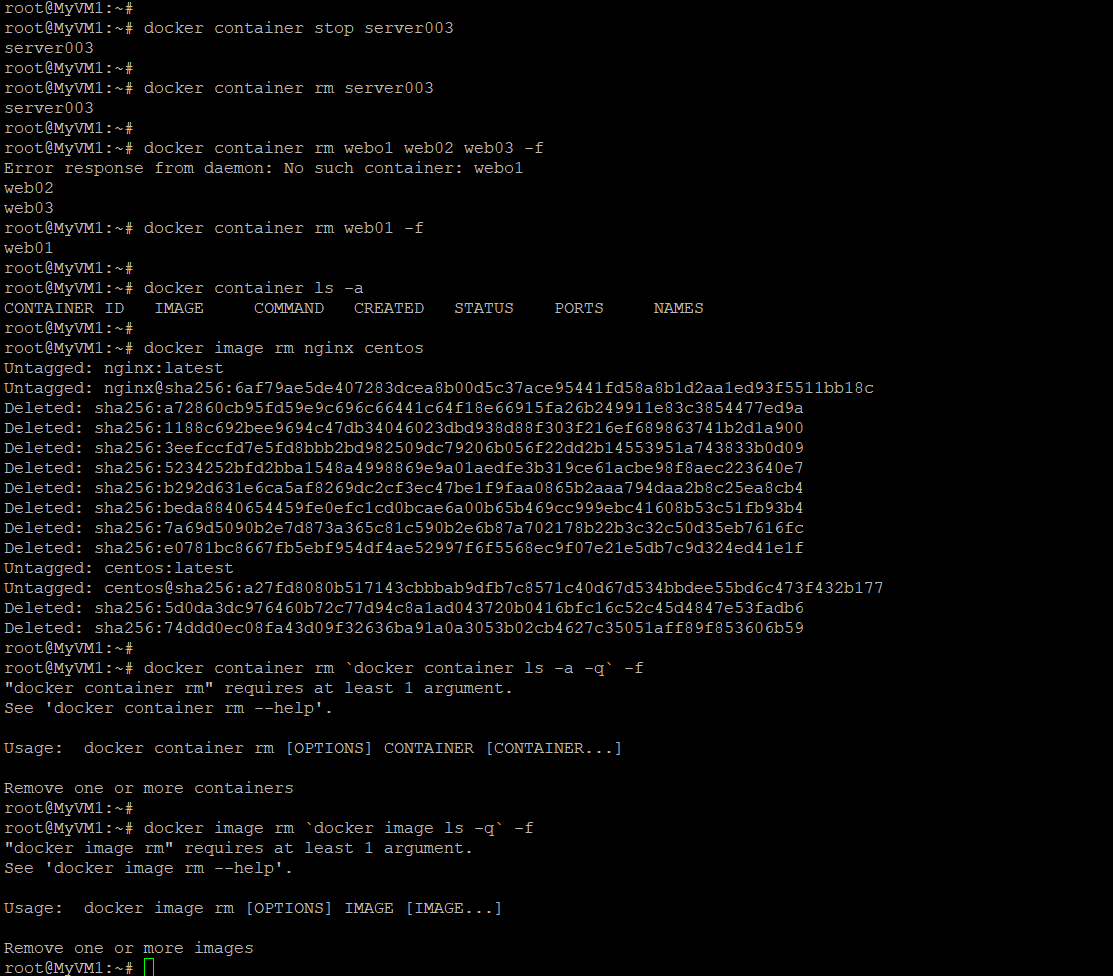
*# docker container stop server003  
  
# docker container rm server003*Let’s **remove** the running containers by force by executing the below command.  
  
*# docker container rm web01 web02 web03 -f*

Let’s **list** the container by executing the below command.

*# docker container ls -a*Let’s remove the downloaded images by executing the below command.   
  
*# docker image rm nginx centos*

Let’s **cleanup** by executing the below command.   
  
*# docker container rm `docker container ls -a -q` -f*

*# docker image rm `docker image ls -q` -f*

**