

[illegible]

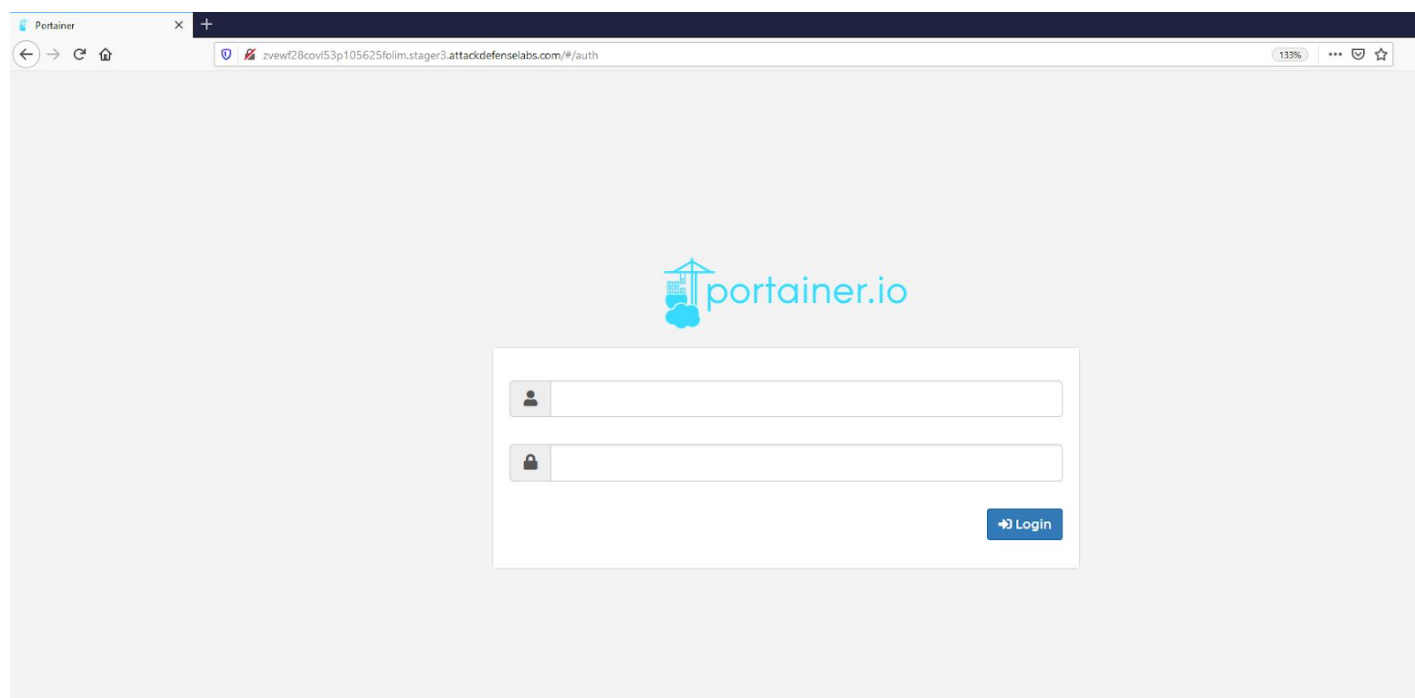
<b>Name</b>	Weakest Link
<b>URL</b>	<a href="https://attackdefense.com/challengedetails?cid=1415">https://attackdefense.com/challengedetails?cid=1415</a>
<b>Type</b>	DevSecOps : Docker Breakouts

**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:**

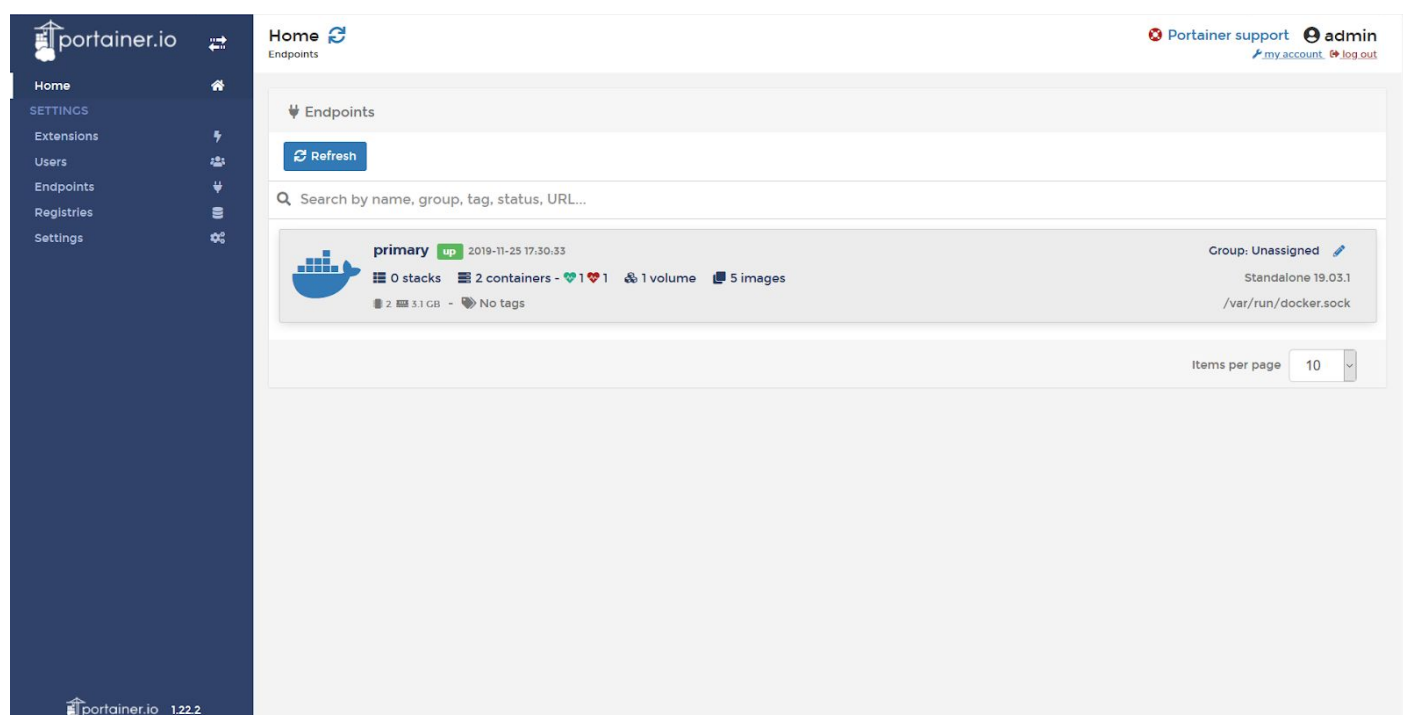
**Login page:**



**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.

Home

PRIMARY

Dashboard

App Templates

Stacks

Containers

Images

Networks

Volumes

Events

Host

SETTINGS

Extensions

Users

Endpoints

Registries

Settings

Dashboard

Endpoint summary

Endpoint info

Endpoint	primary 2 3.1 GB - Standalone 19.03.1
URL	/var/run/docker.sock
Tags	-

0 Stacks

5 Images 1 GB

3 Networks

2 Containers 1 running 1 stopped

1 Volume

Portainer support

admin

my account

log out

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

Home

PRIMARY

Dashboard

App Templates

Stacks

Containers

Images

Networks

Volumes

Events

Host

SETTINGS

Extensions

Users

Endpoints

Registries

Settings

Image list

Images

Pull image

Image

e.g. myImage:myTag

Registry

DockerHub

Image name is required.

Note: if you don't specify the tag in the image name, latest will be used.

Pull the image

Images

Settings

Remove

+ Build a new image

Import

Export

Search...

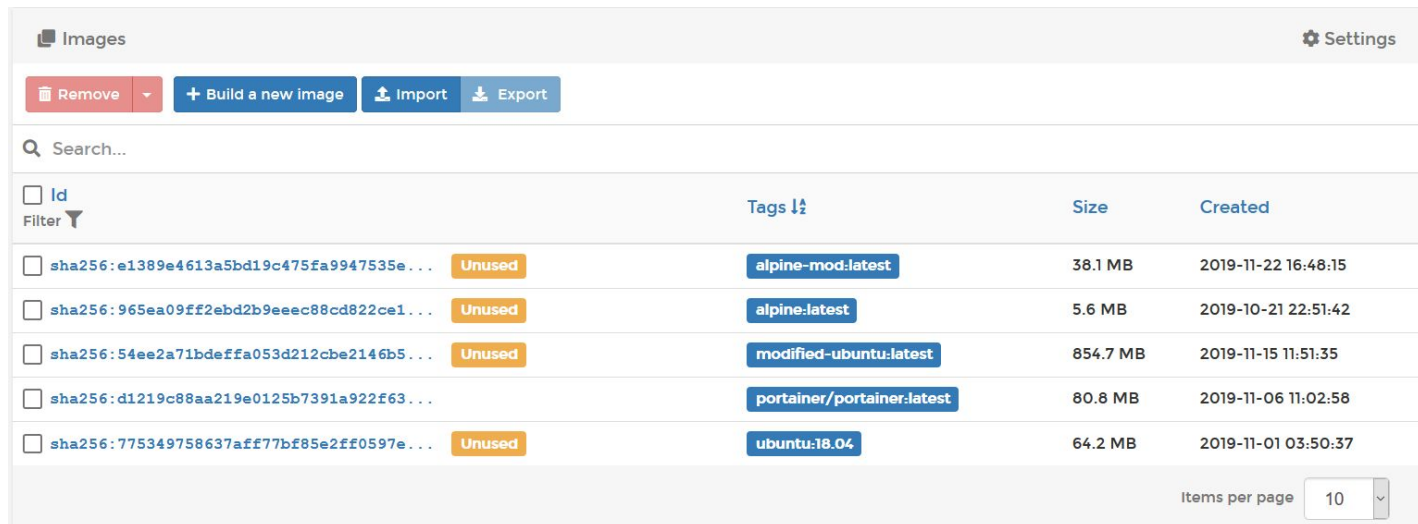
Id	Tags	Size	Created
sha256:e1389e4613a5bd19c475fa9947535e...	alpine-mod:latest	38.1 MB	2019-11-22 16:48:15
sha256:965ea09ff2ebd2b9ecec88cd822ce1...	alpine:latest	5.6 MB	2019-10-21 22:51:42
sha256:54ee2a71bdeffa053d212cbe2146b5...	modified-ubuntu:latest	854.7 MB	2019-11-15 11:51:35
sha256:d1219c88aa219e0125b7391a922f63...	portainer/portainer:latest	80.8 MB	2019-11-06 11:02:58
sha256:775349758637aff77bf85e2ff0597e...	ubuntu:18.04	64.2 MB	2019-11-01 03:50:37

Portainer support

admin

my account

log out



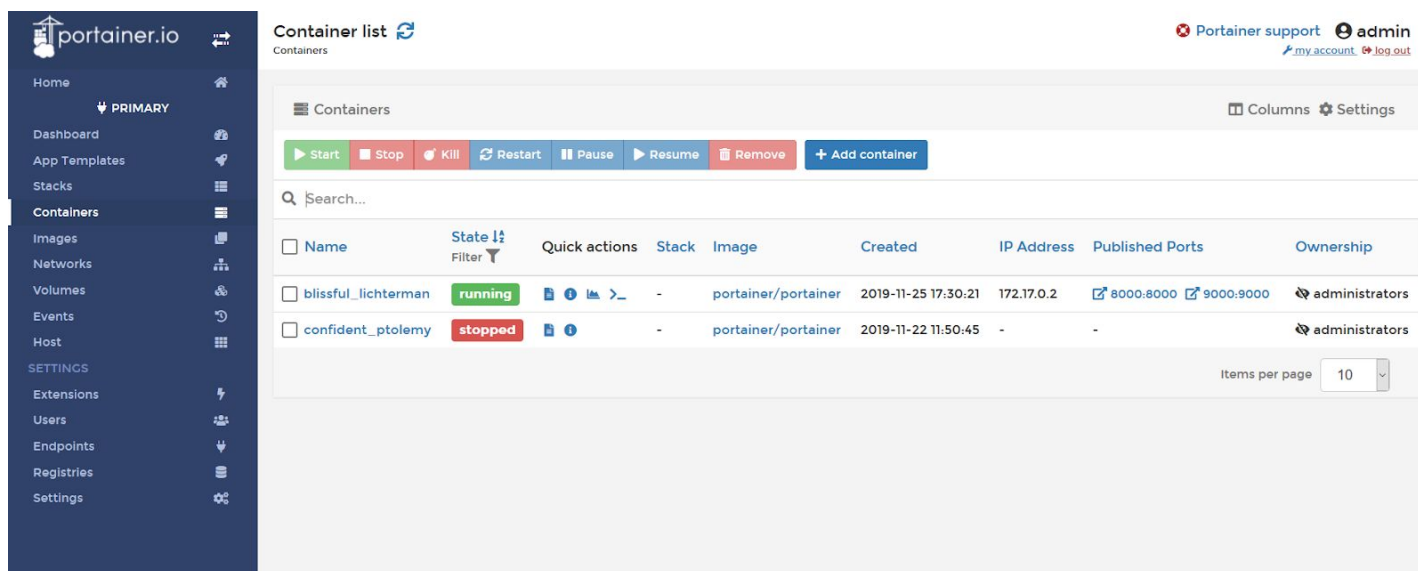
The screenshot shows the 'Images' section of the Portainer interface. At the top, there are buttons for 'Remove', 'Build a new image', 'Import', and 'Export'. Below these is a search bar. The main area contains a table of images with columns for 'Id', 'Tags', 'Size', and 'Created'. The 'Id' column has a filter icon. The 'Tags' column shows the image name and version. The 'Size' column shows the image size in MB. The 'Created' column shows the creation date and time. There are five images listed, each with a checkbox and a status label ('Unused' or 'Used').

Id	Tags	Size	Created
<input type="checkbox"/> sha256:e1389e4613a5bd19c475fa9947535e...	Unused	alpine-mod:latest	38.1 MB
<input type="checkbox"/> sha256:965ea09ff2ebd2b9ecec88cd822ce1...	Unused	alpine:latest	5.6 MB
<input type="checkbox"/> sha256:54ee2a71bdeffa053d212cbe2146b5...	Unused	modified-ubuntu:latest	854.7 MB
<input type="checkbox"/> sha256:d1219c88aa219e0125b7391a922f63...	Used	portainer/portainer:latest	80.8 MB
<input type="checkbox"/> sha256:775349758637aff77b85e2ff0597e...	Unused	ubuntu:18.04	64.2 MB

Items per page: 10

5 images are present on the machine.

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



The screenshot shows the 'Containers' section of the Portainer interface. On the left is a sidebar with the 'Containers' tab selected. The main area has a 'Container list' header with a refresh icon. Below the header are buttons for 'Start', 'Stop', 'Kill', 'Restart', 'Pause', 'Resume', 'Remove', and 'Add container'. There is a search bar. The table below has columns for 'Name', 'State', 'Quick actions', 'Stack', 'Image', 'Created', 'IP Address', 'Published Ports', and 'Ownership'. There are two containers listed: 'blissful\_lichterman' (running) and 'confident\_ptolemy' (stopped).

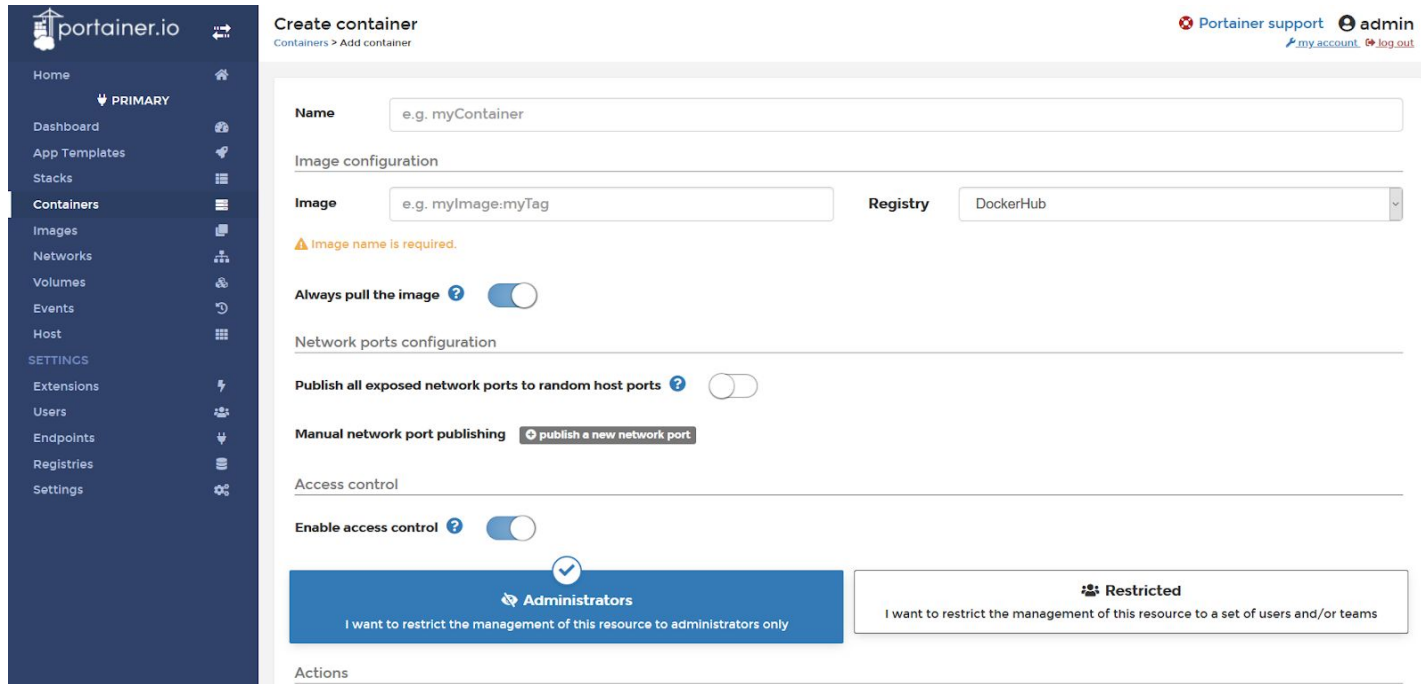
Name	State	Quick actions	Stack	Image	Created	IP Address	Published Ports	Ownership
<input type="checkbox"/> blissful_lichterman	running		-	portainer/portainer	2019-11-25 17:30:21	172.17.0.2	8000:8000 9000:9000	administrators
<input type="checkbox"/> confident_ptolemy	stopped		-	portainer/portainer	2019-11-22 11:50:45	-	-	administrators

Items per page: 10

Two containers are running on the machine.

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





portainer.io

Create container

Containers > Add container

Portainer support admin

my account log out

Home PRIMARY

Dashboard

App Templates

Stacks

Containers

Images

Networks

Volumes

Events

Host

SETTINGS

Extensions

Users

Endpoints

Registries

Settings

Name e.g. myContainer

Image configuration

Image e.g. myImage:myTag Registry DockerHub

Image name is required.

Always pull the image ? ☒

Network ports configuration

Publish all exposed network ports to random host ports ? ☐

Manual network port publishing [publish a new network port](#)

Access control

Enable access control ? ☒

Administrators

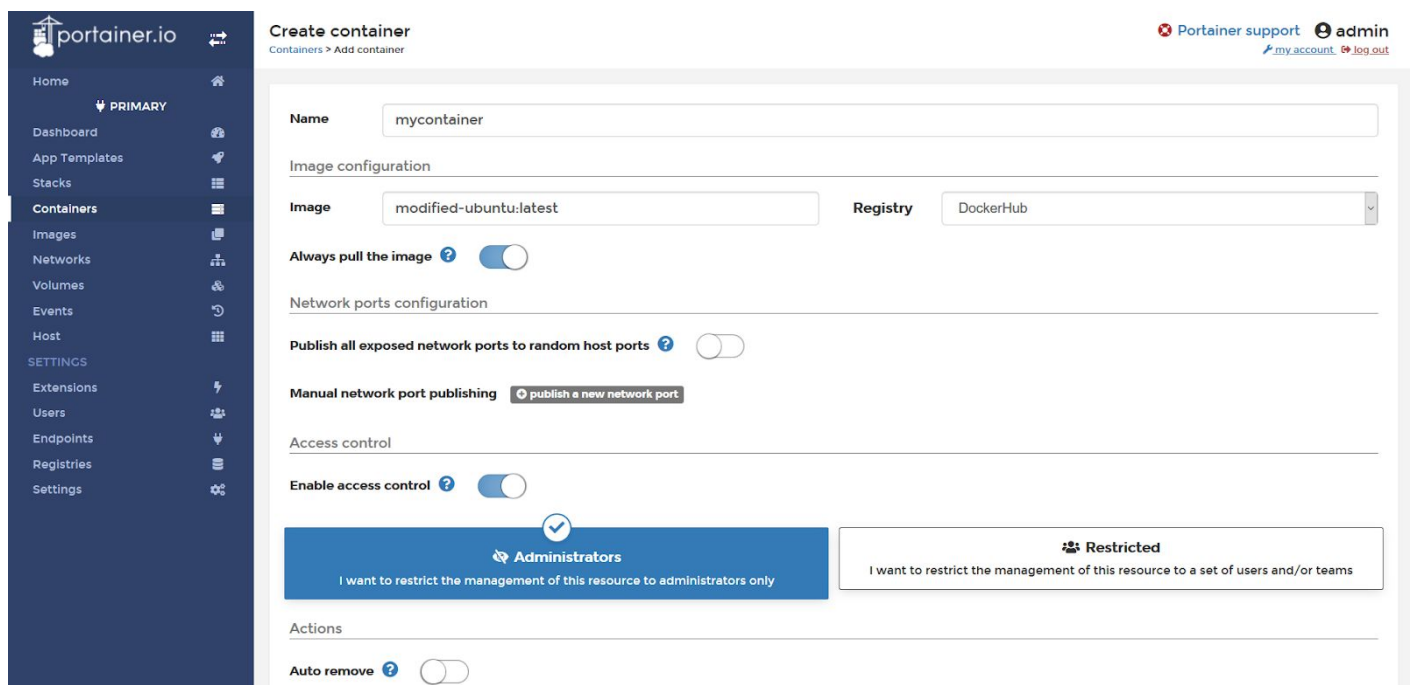
I want to restrict the management of this resource to administrators only

Restricted

I want to restrict the management of this resource to a set of users and/or teams

Actions

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



portainer.io

Create container

Containers > Add container

Portainer support admin

my account log out

Home PRIMARY

Dashboard

App Templates

Stacks

Containers

Images

Networks

Volumes

Events

Host

SETTINGS

Extensions

Users

Endpoints

Registries

Settings

Name mycontainer

Image configuration

Image modified-ubuntu:latest Registry DockerHub

Always pull the image ? ☒

Network ports configuration

Publish all exposed network ports to random host ports ? ☐

Manual network port publishing [publish a new network port](#)

Access control

Enable access control ? ☒

Administrators

I want to restrict the management of this resource to administrators only

Restricted

I want to restrict the management of this resource to a set of users and/or teams

Actions

Auto remove ? ☐

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

The screenshot shows the Portainer.io interface. On the left is a dark sidebar with a menu including Home, PRIMARY, Dashboard, App Templates, Stacks, Containers, Images, Networks, Volumes, Events, Host, SETTINGS, Extensions, Users, Endpoints, Registries, and Settings. The main panel has a top bar with a 'Deploy the container' button. Below it is the 'Advanced container settings' section with tabs for Command & logging, Volumes, Network, Env, Labels, Restart policy, Runtime & Resources (selected), and Capabilities. The 'Runtime & Resources' tab contains sections for Runtime (Privileged mode toggle is on, Runtime dropdown is set to Default), Devices (with an 'add device' button), and Resources (Memory reservation and Memory limit, both set to unlimited with sliders and input fields for 3144 and 0). The 'Memory soft limit (MB)' and 'Memory limit (MB)' labels are visible next to the input fields.

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

The screenshot shows the Portainer.io container configuration page. The sidebar is the same as in the previous screenshot. The main panel has a top bar with a 'Deploy the container' button. Below it is the 'Advanced container settings' section with tabs for Command & logging, Volumes, Network, Env, Labels, Restart policy, Runtime & Resources, and Capabilities. The 'Runtime & Resources' tab contains sections for Runtime (Privileged mode toggle is on, Runtime dropdown is set to Default), Devices (with an 'add device' button), and Resources (Memory reservation and Memory limit, both set to unlimited with sliders and input fields for 3144 and 0). The 'Memory soft limit (MB)' and 'Memory limit (MB)' labels are visible next to the input fields. Below the Resources section is the 'Access control' section with an 'Enable access control' toggle (on) and two buttons: 'Administrators' (selected) and 'Restricted'. The 'Administrators' button has a checkmark icon and the text 'I want to restrict the management of this resource to administrators only'. The 'Restricted' button has a lock icon and the text 'I want to restrict the management of this resource to a set of users and/or teams'. Below the Access control section is the 'Actions' section with an 'Auto remove' toggle (on) and a 'Deployment in progress...' button.

Container list

Containers

Start Stop Kill Restart Pause Resume Remove Add container

Search...

Name	State	Quick actions	Stack	Image	Created	IP Address	Published Ports	Ownership
mycontainer	running	[Icons]	-	modified-ubuntu:latest	2019-11-25 17:36:56	172.17.0.3	-	administrators
blissful_lichterman	running	[Icons]	-	portainer/portainer	2019-11-25 17:30:21	172.17.0.2	8000:8000 9000:9000	administrators
confident_ptolemy	stopped	[Icons]	-	portainer/portainer	2019-11-22 11:50:45	-	-	administrators

Items per page 10

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 list

Containers

Start Stop Kill Restart Pause Resume Remove Add container

Search...

Name	State	Quick actions	Stack	Image	Created	IP Address	Published Ports	Ownership
mycontainer	running	[Icons]	-	modified-ubuntu:latest	2019-11-25 17:36:56	172.17.0.3	-	administrators
blissful_lichterman	running	[Icons]	-	portainer/portainer	2019-11-25 17:30:21	172.17.0.2	8000:8000 9000:9000	administrators
confident_ptolemy	stopped	[Icons]	-	portainer/portainer	2019-11-22 11:50:45	-	-	administrators

Items per page 10



## Container Console:

Container console


Containers > mycontainer > Console

Portainer support admin

[my account](#) [log out](#)

>\_ Execute

Command



/bin/bash

Use custom command

☐

User ?

root

Connect

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

**Command:** id

Container console

Containers > mycontainer > Console

Portainer support admin

[my account](#) [log out](#)

>\_ Execute

Exec into container as default user using command bash

Disconnect

```
root@f9ebe4fdc6ca:~# id
uid=0(root) gid=0(root) groups=0(root)
root@f9ebe4fdc6ca:~#
root@f9ebe4fdc6ca:~#
root@f9ebe4fdc6ca:~#
```

**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/>)