



Containers

Containers Overview Lab Solutions

5.1. CHALLENGE: logs

- Get help on the `docker container logs` subcommand

```
$ docker container logs -h
```

```
Usage:  docker container logs [OPTIONS] CONTAINER
```

Fetch the logs of a container

Options:

<code>--details</code>	Show extra details provided to logs
<code>-f, --follow</code>	Follow log output
<code>--since string</code>	Show logs since timestamp (e.g. 2013-01-02T13:23:37Z) or relative (e.g. 42m for 42 minutes)
<code>-n, --tail string</code>	Number of lines to show from the end of the logs (default "all")
<code>-t, --timestamps</code>	Show timestamps
<code>--until string</code>	Show logs before a timestamp (e.g. 2013-01-02T13:23:37Z) or relative (e.g. 42m for 42 minutes)

```
$
```

- Display the container log data with timestamps added

```
$ docker container logs -t c1
```

```
2021-12-28T21:10:27.452306747Z hello Docker
2021-12-28T21:10:37.453477046Z hello Docker
2021-12-28T21:10:47.454366384Z hello Docker
2021-12-28T21:10:57.455326037Z hello Docker
2021-12-28T21:11:07.456228457Z hello Docker
```

```
$
```

- Use the "follow" option to continuously display the log output of the container

```
$ docker container logs -f c1

hello Docker
hello Docker
hello Docker
hello Docker
hello Docker
hello Docker

^C

$
```

6.1. CHALLENGE: top

- Run the top command with the `-t` ps switch

```
$ docker container top c2 -t
```

PID	TTY	STAT	TIME
15621	?	Ss	0:00
nginx: master process nginx -g daemon off;			
15673	?	S	0:00
nginx: worker process			
15674	?	S	0:00
nginx: worker process			

- Run the top command with the `-w` ps switch

```
$ docker container top c2 -w
```

PID	TTY	TIME	CMD
15621	?	00:00:00	nginx
15673	?	00:00:00	nginx
15674	?	00:00:00	nginx

```
$
```

- Try your own favorite ps switches with `docker container top`

Using `-ef`:

```
$ docker container top c2 -ef
```

UID	PID	PPID	C
-----	-----	------	---

```

STIME          TTY          TIME          CMD
root           15621        15599         0
21:12          ?           00:00:00      nginx: master
process nginx -g daemon off;
systemd+       15673        15621         0
21:12          ?           00:00:00      nginx: worker
process
systemd+       15674        15621         0
21:12          ?           00:00:00      nginx: worker
process

$

```

Using `-eo` and showing the namespace inode values:

```

$ docker container top c2 -eo
pid,user,ipcns,mntns,netns,pidns,utsns,usersns,cmd

PID          USER          IPCNS          MNTNS
NETNS        PIDNS          UTSNS          USERSNS
CMD
15621        root          4026532274    4026532272
4026532277    4026532275    4026532273    4026531837
nginx: master process nginx -g daemon off;
15673        systemd+     4026532274    4026532272
4026532277    4026532275    4026532273    4026531837
nginx: worker process
15674        systemd+     4026532274    4026532272
4026532277    4026532275    4026532273    4026531837
nginx: worker process

```

8. CHALLENGE: container runtime options

Run `docker container run --help` and explore available options.

You have probably noticed that Docker gives random names to containers in the pattern of `adjective_noun`. Use the appropriate options with `docker container run` to run a container with the following requirements:

- use the `nginx:1.19.0` image
- name the container `webserver`
- run the container in the background
- add an environment variable `CHALLENGE=containers`
 - exec into the container and list the environment variable to confirm it worked

```

$ docker container run --help

Usage:  docker container run [OPTIONS] IMAGE [COMMAND] [ARG...]

```

Run a command in a new container

Options:

...

<code>-d, --detach</code>	Run container in background and print container ID
---------------------------	--

...

<code>-e, --env list</code>	Set environment variables
<code>--env-file list</code>	Read in a file of environment variables

...

<code>--name string</code>	Assign a name to the container
----------------------------	--------------------------------

...

Run the container:

```
$ docker container run -d -e CHALLENGE=containers --name webserver
nginx:1.19.0

Unable to find image 'nginx:1.19.0' locally
1.19.0: Pulling from library/nginx
8559a31e96f4: Pull complete
8d69e59170f7: Pull complete
3f9f1ec1d262: Pull complete
d1f5ff4f210d: Pull complete
1e22bfa8652e: Pull complete
Digest:
sha256:21f32f6c08406306d822a0e6e8b7dc81f53f336570e852e25fbe1e3e3d0d0133
Status: Downloaded newer image for nginx:1.19.0
2f9bb12d95a7fef8ff6aebe1b2174d1ac4cd945c4121ae4bd20e9ae5560e279a

$
```

Exec into the container and list the environment variable to confirm it worked:

```
$ docker container exec webserver env

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=33ef646ef64f
CHALLENGE=containers
NGINX_VERSION=1.19.0
```

```
NJS_VERSION=0.4.1  
PKG_RELEASE=1~buster  
HOME=/root  
  
$
```

Copyright (c) 2023-2024 RX-M LLC, Cloud Native & AI Training and Consulting, all rights reserved