

PLAY WITH DOCKER CLASSROOM

Docker for Beginners - Linux

training.play-with-docker.com/beginner-linux/

Apps Gmail YouTube Maps

Reading list

Play with Docker classroom

About

register for one. You will need this for later steps.

Task 1: Run some simple Docker containers

There are different ways to use containers. These include:

1. **To run a single task:** This could be a shell script or a custom app.
2. **Interactively:** This connects you to the container similar to the way you SSH into a remote server.
3. **In the background:** For long-running services like websites and databases.

In this section you'll try each of those options and see how Docker manages the workload.

Run a single task in an Alpine Linux container

In this step we're going to start a new container and tell it to run the `hostname` command. The container will start, execute the `hostname` command, then exit.

1. Run the following command in your Linux console.

```
docker container run alpine hostname
```

The output below shows that the `alpine:latest` image could not be found locally. When this happens, Docker automatically *pulls* it from Docker Hub.

After the image is pulled, the container's hostname is displayed (`888e89a3b36b`) in the example below.

Your use of Play With Docker is subject to the Docker Terms of Service which can be accessed [here](#). Site created by Tutorius

docker build automated

f t

If the commandline doesn't appear in the terminal, make sure popups are enabled or try resizing the browser window.

```
#
# The FWD team.
#####
[node1] (local) root@192.168.0.18 ~
$ git clone https://github.com/dockerexamples/linux_tweet_app
Cloning into 'linux_tweet_app'...
remote: Enumerating objects: 14, done.
remote: Total 14 (delta 0), reused 0 (delta 0), pack-reused 14
Receiving objects: 100% (14/14), 10.76 KiB | 10.76 MiB/s, done.
Resolving deltas: 100% (5/5), done.
[node1] (local) root@192.168.0.18 ~
$ docker container run alpine hostname
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
59bf1c3509f3: Pull complete
Digest: sha256:21a3deaa0d32a8057914f36584b5288d2e5ecc984380bc0118285c70fa8c9300
Status: Downloaded newer image for alpine:latest
966a7d3cbc5d
[node1] (local) root@192.168.0.18 ~
$
```

Docker for Beginners - Linux

training.play-with-docker.com/beginner-linux/

Apps Gmail YouTube Maps

Reading list

Play with Docker classroom

About

5. You can also use `docker container exec` to connect to a new shell process inside an already-running container. Executing the command below will give you an interactive shell (`sh`) inside your MySQL container.

```
docker exec -it mydb sh
```

Notice that your shell prompt has changed. This is because your shell is now connected to the `sh` process running inside of your container.

6. Let's check the version number by running the same command again, only this time from within the new shell session in the container.

```
mysql --user=root --password=$MYSQL_ROOT_PASSWORD --version
```

Notice the output is the same as before.

7. Type `exit` to leave the interactive shell session.

```
exit
```

Task 2: Package and run a custom app using Docker

In this step you'll learn how to package your own apps as Docker images using a Dockerfile.

Your use of Play With Docker is subject to the Docker Terms of Service which can be accessed [here](#). Site created by Tutorius

docker build automated

f t

If the commandline doesn't appear in the terminal, make sure popups are enabled or try resizing the browser window.

```
rbiose --help --log-bin-index=/tmp/tmp.0WN3ZMUTpd
1292          999          0:00          awk -v conf
=datadir $1 == conf {{ /^[^ \t]/ { sub(/^[^ \t]+[ \t]+/, ""); print; ex
it } }
[node1] (local) root@192.168.0.18 ~
$ docker exec -it mydb \
> mysql --user=root --password=$MYSQL_ROOT_PASSWORD --version
mysql: [Warning] Using a password on the command line interface can be
insecure.
mysql Ver 8.0.27 for Linux on x86_64 (MySQL Community Server - GPL)
[node1] (local) root@192.168.0.18 ~
$ docker exec -it mydb sh
# docker exec -it mydb sh
sh: 1: docker: not found
# mysql --user=root --password=$MYSQL_ROOT_PASSWORD --version
mysql: [Warning] Using a password on the command line interface can be
insecure.
mysql Ver 8.0.27 for Linux on x86_64 (MySQL Community Server - GPL)
# exit
[node1] (local) root@192.168.0.18 ~
$
```

Docker for Beginners - Linux

ip172-18-0-13-c7cnjdnnjsv000f8

training.play-with-docker.com/beginner-linux/

Apps Gmail YouTube Maps

Reading list

Play with Docker classroom

About

```
cp index-new.html index.html
```

2. Go to the running website and refresh the page. Notice that the site has changed.

If you are comfortable with `vi` you can use it to load the local `index.html` file and make additional changes. Those too would be reflected when you reload the webpage. If you are really adventurous, why not try using `exec` to access the running container and modify the files stored there.

Even though we've modified the `index.html` local filesystem and seen it reflected in the running container, we've not actually changed the Docker image that the container was started from.

To show this, stop the current container and re-run the `1.0` image without a bind mount.

1. Stop and remove the currently running container.

```
docker rm --force linux_tweet_app
```

2. Rerun the current version without a bind mount.

```
docker container run \
--detach \
--publish 80:80 \
$DOCKERID/linux_tweet_app:1.0
```

If the commandline doesn't appear in the terminal, make sure popups are enabled or try resizing the browser window.

```
> --publish 80:80 \
> --name linux_tweet_app \
> $DOCKERID/linux_tweet_app:1.0
docker: invalid reference format.
See 'docker run --help'.
[node1] (local) root@192.168.0.18 ~/linux_tweet_app
$ docker container rm --force linux_tweet_app
Error: No such container: linux_tweet_app
[node1] (local) root@192.168.0.18 ~/linux_tweet_app
$ docker container run \
--detach \
--publish 80:80 \
--name linux_tweet_app \
--mount type=bind,source="$(pwd)",target=/usr/share/nginx/html \
$DOCKERID/linux_tweet_app:1.0
docker: invalid reference format.
See 'docker run --help'.
[node1] (local) root@192.168.0.18 ~/linux_tweet_app
$ cp index-new.html index.html
[node1] (local) root@192.168.0.18 ~/linux_tweet_app
$
```

Your use of Play With Docker is subject to the Docker Terms of Service which can be accessed [here](#). Site created by Tutorius

docker build automated

f t

Docker for Beginners - Linux

ip172-18-0-13-c7cnjdnnjsv000f8

training.play-with-docker.com/beginner-linux/

Apps Gmail YouTube Maps

Reading list

Play with Docker classroom

About

2. Go to the running website and refresh the page. Notice that the site has changed.

If you are comfortable with `vi` you can use it to load the local `index.html` file and make additional changes. Those too would be reflected when you reload the webpage. If you are really adventurous, why not try using `exec` to access the running container and modify the files stored there.

Even though we've modified the `index.html` local filesystem and seen it reflected in the running container, we've not actually changed the Docker image that the container was started from.

To show this, stop the current container and re-run the `1.0` image without a bind mount.

1. Stop and remove the currently running container.

```
docker rm --force linux_tweet_app
```

2. Rerun the current version without a bind mount.

```
docker container run \
--detach \
--publish 80:80 \
--name linux_tweet_app \
$DOCKERID/linux_tweet_app:1.0
```

If the commandline doesn't appear in the terminal, make sure popups are enabled or try resizing the browser window.

```
> --publish 80:80 \
> --name linux_tweet_app \
> --mount type=bind,source="$(pwd)",target=/usr/share/nginx/html \
> $DOCKERID/linux_tweet_app:1.0
docker: invalid reference format.
See 'docker run --help'.
[node1] (local) root@192.168.0.18 ~/linux_tweet_app
$ cp index-new.html index.html
[node1] (local) root@192.168.0.18 ~/linux_tweet_app
$ docker rm --force linux_tweet_app
Error: No such container: linux_tweet_app
[node1] (local) root@192.168.0.18 ~/linux_tweet_app
$ docker container run \
--detach \
--publish 80:80 \
--name linux_tweet_app \
$DOCKERID/linux_tweet_app:1.0
docker: invalid reference format.
See 'docker run --help'.
[node1] (local) root@192.168.0.18 ~/linux_tweet_app
$
```

Your use of Play With Docker is subject to the Docker Terms of Service which can be accessed [here](#). Site created by Tutorius

docker build automated

f t

Application Containerization and x ip172-18-0-13-c7cnjdngjsv000f8 x +

training.play-with-docker.com/microservice-orchestration/

Apps Gmail YouTube Maps Reading list

Play with Docker classroom

About

- Step 5: Redis Service for Caching
- Step 6: Swap Python API Service with Ruby
- Conclusions

Stage Setup

Let's get started by first cloning the demo code repository, changing the working directory, and checking the `demo` branch out.

```
git clone https://github.com/ibnesayeed/linkextractor.git
cd linkextractor
git checkout demo
```

Step 0: Basic Link Extractor Script

Checkout the `step0` branch and list files in it.

```
git checkout step0
tree
```

```

.
├── README.md
```

Your use of Play With Docker is subject to the Docker Terms of Service which can be accessed [here](#). Site created by Tutorius

docker build automated

f t

If the commandline doesn't appear in the terminal, make sure popups are enabled or try resizing the browser window.

```
# completely the user's responsibilities.
#
# The PWD team.
#####
[node1] (local) root@192.168.0.13 ~
$ git clone https://github.com/ibnesayeed/linkextractor.git
Cloning into 'linkextractor'...
remote: Enumerating objects: 144, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 144 (delta 0), reused 1 (delta 0), pack-reused 140
Receiving objects: 100% (144/144), 44.55 KiB | 8.91 MiB/s, done.
Resolving deltas: 100% (43/43), done.
[node1] (local) root@192.168.0.13 ~
$ cd linkextractor
[node1] (local) root@192.168.0.13 ~/linkextractor
$ git checkout demo
Branch 'demo' set up to track remote branch 'demo' from 'origin'.
Switched to a new branch 'demo'
[node1] (local) root@192.168.0.13 ~/linkextractor
$
```

Type here to search

17:20 08-01-2022 23

Application Containerization and x ip172-18-0-13-c7cnjdngjsv000f8 x +

training.play-with-docker.com/microservice-orchestration/

Apps Gmail YouTube Maps Reading list

Play with Docker classroom

About

This is a simple Python script that imports three packages: `sys` from the standard library and two popular third-party packages `requests` and `bs4`. User-supplied command line argument (which is expected to be a URL to an HTML page) is used to fetch the page using the `requests` package, then parsed using the `BeautifulSoup`. The parsed object is then iterated over to find all the anchor elements (i.e., `<a>` tags) and print the value of their `href` attribute that contains the hyperlink.

However, this seemingly simple script might not be the easiest one to run on a machine that does not meet its requirements. The `README.md` file suggests how to run it, so let's give it a try:

```
./linkextractor.py http://example.com/
```

```
bash: ./linkextractor.py: Permission denied
```

When we tried to execute it as a script, we got the `Permission denied` error. Let's check the current permissions on this file:

```
ls -l linkextractor.py
```

```
-rw-r--r-- 1 root root 220 Sep 23 16:26 linkextractor.py
```

This current permission `-rw-r--r--` indicates that the script is not set to be executable. We can either change it by running `chmod a+x linkextractor.py` or run it as a Python program instead of a self-executing script as illustrated below:

If the commandline doesn't appear in the terminal, make sure popups are enabled or try resizing the browser window.

```
#####
[node1] (local) root@192.168.0.13 ~
$ git clone https://github.com/ibnesayeed/linkextractor.git
Cloning into 'linkextractor'...
remote: Enumerating objects: 144, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 144 (delta 0), reused 1 (delta 0), pack-reused 140
Receiving objects: 100% (144/144), 44.55 KiB | 8.91 MiB/s, done.
Resolving deltas: 100% (43/43), done.
[node1] (local) root@192.168.0.13 ~
$ cd linkextractor
[node1] (local) root@192.168.0.13 ~/linkextractor
$ git checkout demo
Branch 'demo' set up to track remote branch 'demo' from 'origin'.
Switched to a new branch 'demo'
[node1] (local) root@192.168.0.13 ~/linkextractor
$ ls -l linkextractor.py
ls: linkextractor.py: No such file or directory
[node1] (local) root@192.168.0.13 ~/linkextractor
$
```

Type here to search

17:20 08-01-2022 23