

Docker Health Check

1. What is a health check?

Health checks are exactly what they sound like - a way of checking the health of some resource. In the case of Docker, a health check is a command used to determine the health of a running container.

When a health check command is specified, it tells Docker how to test the container to see if it's working. With no health check specified, Docker has no way of knowing whether or not the services running within your container are actually up or not.

Take an example of python with flask framework.

<https://github.com/ybmadhu/pythonapp>

Let's start with the **requirements.txt**:

```
Flask==0.12.2
```

And the Dockerfile:

```
FROM python:3.6-alpine
```

```
COPY . /app
```

```
WORKDIR /app
```

```
RUN pip install -r requirements.txt
```

```
CMD ["python", "app.py"]
```

And finally, app.py:

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route('/')
```

```
def hello_world():
```

```
    return 'Hello world'
```

```
if __name__ == '__main__':
```

```
    app.run(host='0.0.0.0')
```

Now let's build the container:

```
docker build -t docker-flask .
```

This should build pretty quickly. Then we'll run the container.

```
docker run --rm --name docker-flask -p 5000:5000 docker-flask
```

Now test by opening up your browser to **localhost:5000**. You should see "Hello world".

Add a health check to the Dockerfile

Since the goal of our container is to serve traffic on port 5000, our health check should make sure that is happening.

A health check is configured in the Dockerfile using the HEALTHCHECK instruction. There are two ways to use the HEALTHCHECK instruction:

HEALTHCHECK [OPTIONS] CMD command

or if you want to disable a health check from a parent image:

HEALTHCHECK NONE

So we're obviously going to use the first. So let's add the HEALTHCHECK instruction, and we'll use curl to ensure that our app is serving traffic on port 5000.

So add this line to the Dockerfile right before the last line (CMD).

```
HEALTHCHECK CMD curl --fail http://localhost:5000/ || exit 1
```

In this case, we are using the default options, which are **interval 30s, timeout 30s, start-period 0s, and retries 3**. Read the health check instruction reference for more information on the options.

See the health status

Let's rebuild and run our container.

```
docker build -t docker-flask .
```

```
docker run --rm --name docker-flask -p 5000:5000 docker-flask
```

Now let's take a look at the health status. Notice we have the --name option to the above command so we can easily inspect the container.

```
docker inspect --format='{{json .State.Health}}' docker-flask
```

If you run this immediately after the container starts, you'll see Status is starting.

```
{"Status":"starting","FailingStreak":0,"Log":[]}
```

And after the health check runs (after the default interval of 30s):

```
{"Status":"healthy","FailingStreak":0,"Log":[{"Start":"2017-07-21T06:10:51.809087707Z","End":"2017-07-21T06:10:51.868940223Z","ExitCode":0,"Output":"Hello world"}]}
```

We have a little more information here. We see the Status is healthy as well as some details about the health check.

We can also see the health status by running docker ps.

docker ps

You'll see the following:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
9f89662fc56a	howchoo/docker-flask	"python app.py"	2 minutes ago	Up 2 minutes	
(healthy) 0.0.0.0:5555->5000/tcp		docker-flask			

Notice under STATUS, the status is Up with (healthy) next to it. The health status appears only when a health check is configured.