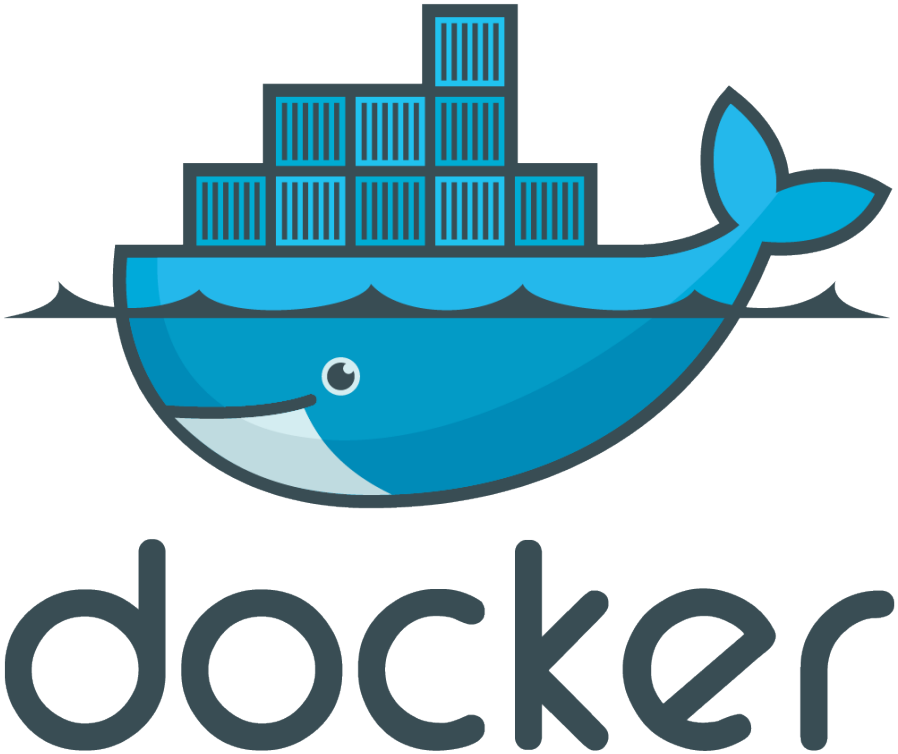
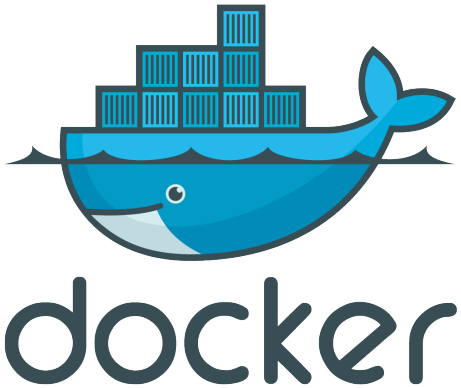

docker

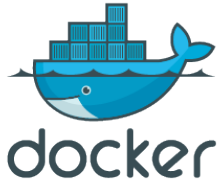


Why Docker is Awesome

Sam Clements

BCS Show and Tell - February 2015

What Docker promises

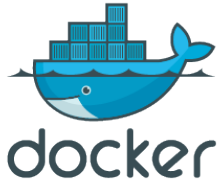


Docker is an open platform for developing, shipping, and running applications.

It can be used to package and run applications on any Unix system*.

*Currently Linux, Solaris and FreeBSD.

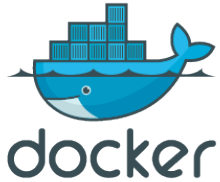
What Docker does



Docker is a “lightweight container virtualization platform”.

- ➔ “lightweight” - Smaller that other virtualization tools.
- ➔ “container” - Runs processes on the same OS/kernel.
- ➔ “virtualization” - Runs processes in a sandbox.
- ➔ “platform” - More than just the `docker` tool.

Why Docker is used for

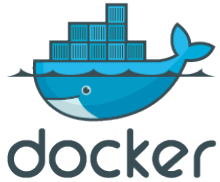


Developers can package applications and run them anywhere.

Often described as “shipping containers” for applications.

*And by anywhere, I mean any Unix system with the same CPU architecture.

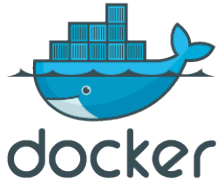
Building a Docker image



A build runs a sequence of Unix commands, one after the other, in the same container.

Each command creates a new ‘layer’ with it’s own modifications to the image (usually new files).

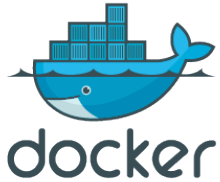
Dockerfile




```
FROM ubuntu:12.04
RUN apt-get update && apt-get install -y python python-pip curl
RUN curl -sSL https://github.com/shykes/helloflask/archive/master.tar.gz | tar -xzv
RUN cd helloflask-master && pip install -r requirements.txt
WORKDIR helloflask-master
CMD python app.py
EXPOSE 80
```

This example creates a container based on Ubuntu and installs a Flask application.

Running a docker container



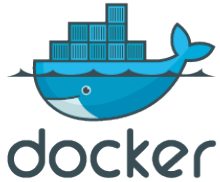

```
$ docker build -t example-container .

$ docker run example-container

* Running on http://localhost:5000/
```

Builds an image using a `Dockerfile` in the current directory, and names it. The starts a container running from that image.

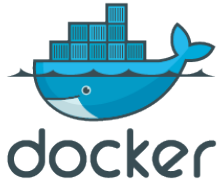
Why should I use it?



Users can run the same application on different machines, even if they are configured differently.

Developers can ensure their environment can always be recreated, even when they break it.

Resources

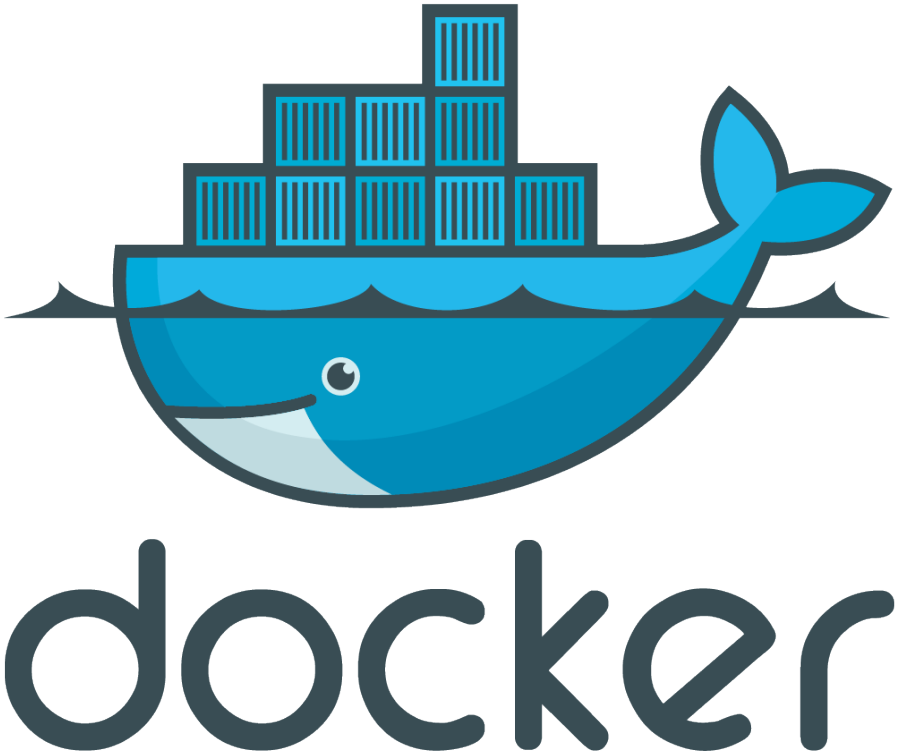


Interactive tutorial

https://www.docker.com/tryit/

Documentation

https://docs.docker.com/


docker