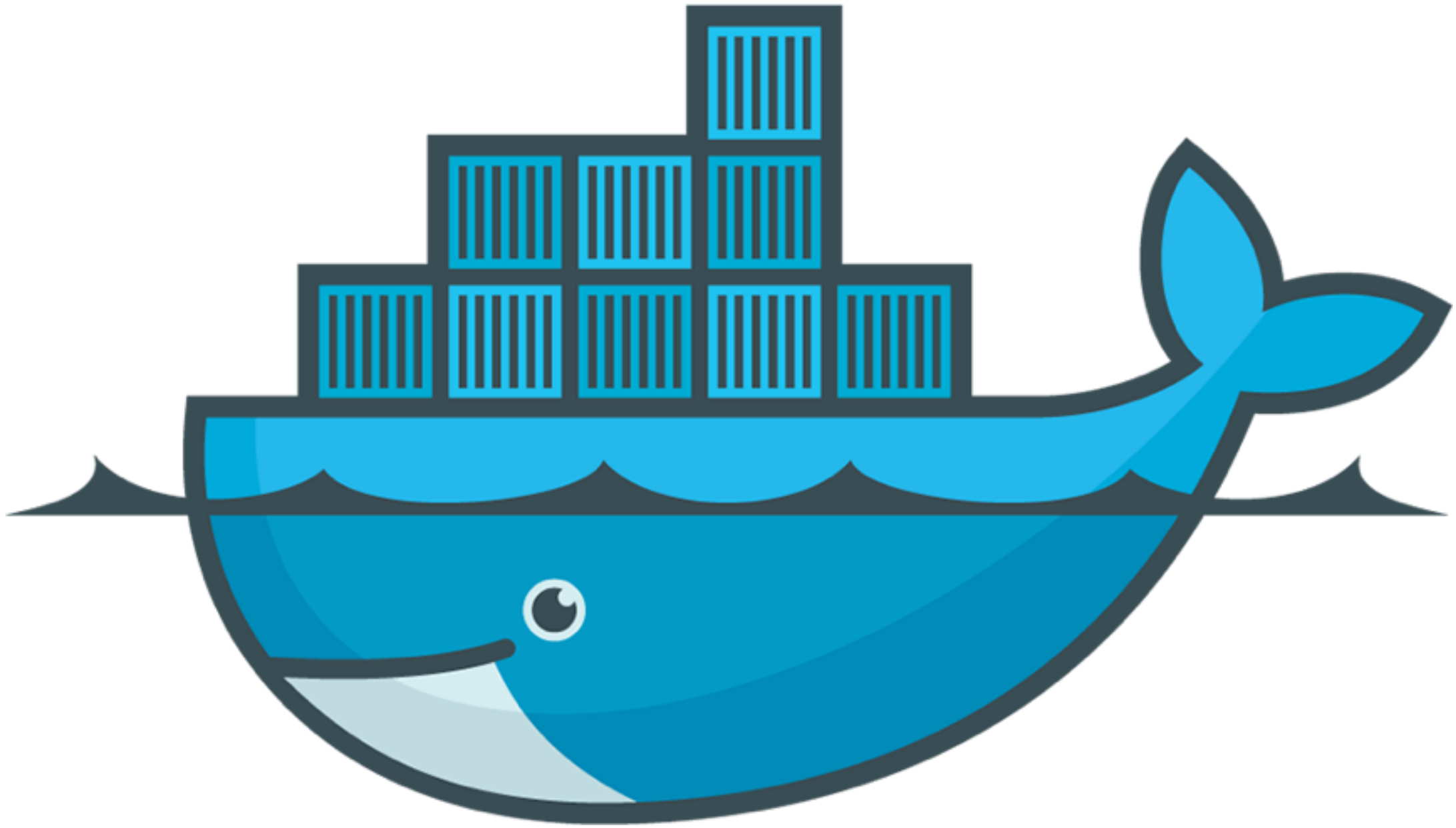
Docker

An introduction for the VM initiate

Docker



VMs are heavy

- Lots of disk space
- Lots of duplicate OS processes to run
- Crude resource management between guest and master
- Unnecessary battery drain
- Still useful enough to be worth it

Docker is light

- Linux cgroups provides cpu/memory/io isolation and identifier namespaces
- Runs only the processes you tell it to - no services.
- Better resource management than VMs - one kernel handles all processes interactively
- Images stored in layered union filesystems to save space.
- Cheap to create
- Can do everything VMs can, if you want a Linux guest

It all sounds so easy…

- It is, but it's very new
- There are new ways of doing things to learn
  - start a shell in a running container
  - connect up containers
  - expose containers to the outside world

Why not use it?

- It's new
- Significant learning required to get benefits
- It only runs Linux programs
- It's command line only (pretty much)

Automation, composition and orchestration

- Fig
- Geard
- Decking
- Centurion
- Octohost
- Panamax

Getting docker

- Linux
  - Kernel 3.8 or later, 3.10 advised.
- Windows/Mac - boot2docker
  - Runs a small Linux VM that containers run in
- http://docker.com/ - really good documentation

Docker containers

- Are running process groups, started a single command
- Have their own isolated writable storage
- Are based on common images
- Have isolated networking stacks
- Can have persistent data volumes
- Are very cheap to create and destroy

A toy example

Starting a shell in Ubuntu Linux:
`docker run --rm -it ubuntu bash`

Or Centos:
`docker run --rm -it centos bash`

Starting a database server:
`docker run -d --name postgres postgres`

Building an image

First we need a Dockerfile - this tells docker how to build the image.

This will specify:

- a base image
- the changes to make to the base
- the command that the image will run

Hello, world!

```
FROM ubuntu

MAINTAINER Ronan Klyne <docker@rklyne.net>

ENV NAME world

CMD echo "Hello, ${NAME}!"

$ docker build -t hello-world .
$ docker run --rm -it hello-world
Hello, world!

$ docker run --rm -it -e NAME="geeks" hello-world
Hello, geeks!
```

Bigger things

```
FROM ubuntu

MAINTAINER Ronan Klyne <docker@rklyne.net>

RUN apt-get update -y && \
    apt-get install -y python

CMD python -c "print 'Hello, world!'"


$ docker build -t hello-world .
...
$ docker run --rm -it hello-world
```

A webapp!

```
FROM ubuntu

MAINTAINER Ronan Klyne <docker@rklyne.net>

RUN apt-get update -y && \
    apt-get install -y python

EXPOSE 80

CMD python -c "from wsgiref import simple_server;
simple_server.make_server('0.0.0.0', 80,
simple_server.demo_app).serve_forever()"

$ docker build -t hello-world .
...
$ docker run --rm -it -p 8001:80 hello-world
```

Linked containers

Run database server in container

```
docker run -d --restart=always --name postgres
postgres
```

Run mediawiki in container, linked to the database

```
docker run -d --restart=always --link
postgres:db synctree/mediawiki
```

Working on containers

Run database server in container

```
docker run -d --restart=always --name postgres
postgres
```

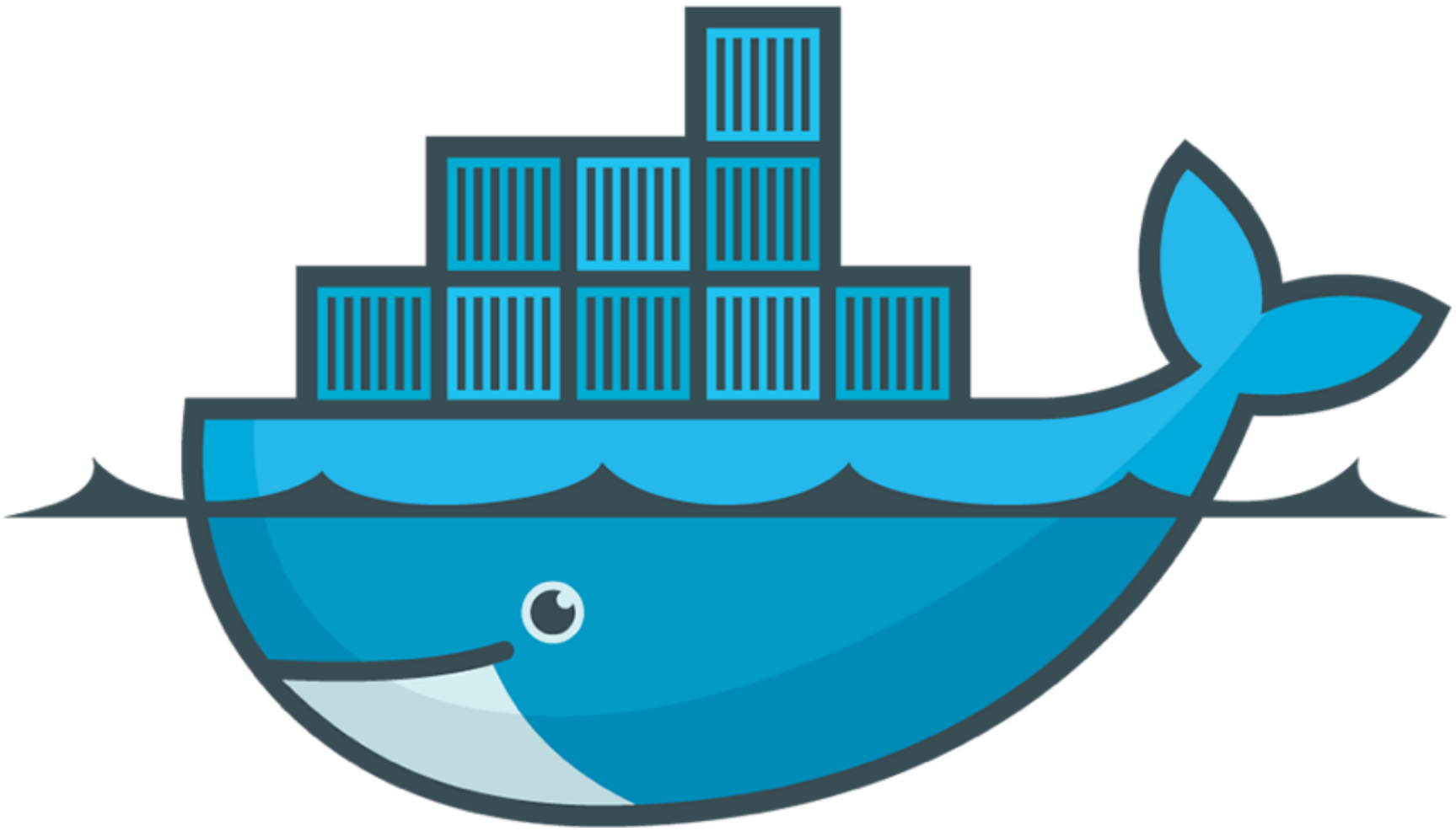
Run database shell

```
docker run --rm -it --link postgres:db postgres
psql -h db -U postgres

docker exec -it postgres psql -U postgres

docker exec -it postgres bash
```

Questions?



Ronan Klyne

@ronanklyne