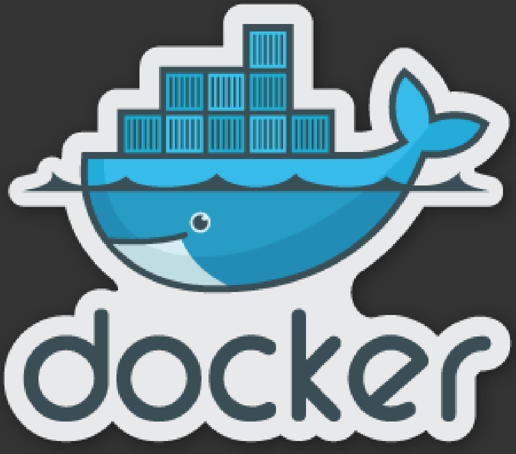
ORGANISE YOUR DOCKER CONTAINERS WITH

PERCHERON

ASH MCKENZIE

DEVOPS ENGINEER AT ZENDESK


docker

DOCKER: 101

- A toolchain around Linux Containers (LXC)
- Docker is relatively new, LXC arrived in 2008
- Fundamental pieces of Docker are:
    - The `Dockerfile`
    - images (O/S disk)
    - containers (a running O/S disk)

DOCKER: DOCKERFILE

`Dockerfile` defines build steps

```
FROM gliderlabs/alpine:latest
MAINTAINER ash@the-rebellion.net

RUN apk add --update-cache redis

EXPOSE 6379

CMD [ "/usr/bin/redis-server", "--port", "6379" ]
```

DOCKER: IMAGES

- Naming format:
    - `<owner>/<name>:<version>`
    - Name required, owner & version are optional (default `latest`)
    - `redis:latest` (official)
    - `gliderlabs/alpine:latest` (unofficial)
- `docker build` creates images from `Dockerfile`

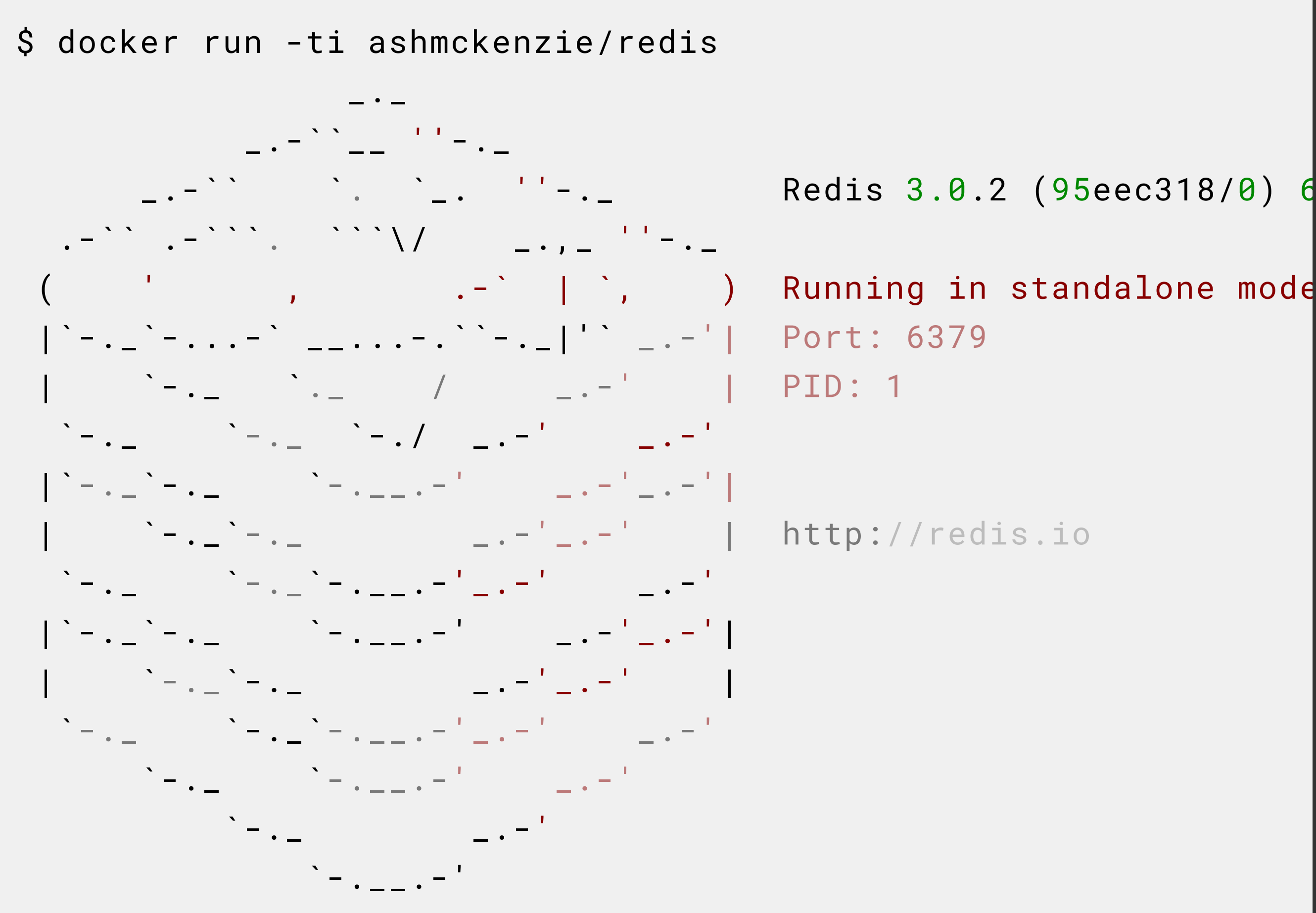
DOCKER: CONTAINERS

- Naming format is arbitrary (default is random!)
  - `backstabbing_mestorf`
  - `sleepy_turing`
- `docker run` creates containers from images

DOCKER: BUILD EXAMPLE

```
$ docker build -t ashmckenzie/redis .
Sending build context to Docker daemon 104.4 kB
Step 0 : FROM gliderlabs/alpine:latest
 ---> 5bd56d818842
Step 1 : MAINTAINER ash@the-rebellion.net
 ---> 2c21769397d4
Step 2 : RUN apk add --update-cache redis
 ---> Running in 4f9afbe971c6
(1/1) Installing redis (3.0.2-r0)
OK: 7 MiB in 16 packages
 ---> 8d5e8b1de70a
Step 3 : EXPOSE 6379
 ---> 9adebad382f8
Step 4 : CMD /usr/bin/redis-server --port 6379
 ---> 92de3334f006
Successfully built 92de3334f006
```

DOCKER: RUN EXAMPLE



```
$ docker run -ti ashmckenzie/redis
                _._
           _.-``__ ''-._
      _.-``    `.  `_.  ''-._           Redis 3.0.2 (95eec318/0) 6
  .-`` .-```.  ```\/    _.,_ ''-._
 (    '      ,       .-`  | `,    )     Running in standalone mode
 |`-._`-...-` __...-.``-._|'` _.-'|     Port: 6379
 |    `-._   `._    /     _.-'    |     PID: 1
  `-._    `-._  `-./  _.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |     http://redis.io
  `-._    `-._`-.__.-'_.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |
  `-._    `-._`-.__.-'_.-'    _.-'
      `-._    `-.__.-'    _.-'
          `-._        _.-'
              `-.__.-'
```

DOCKER: BENEFITS

- Process / application isolation (?safe/secure?)
- Reproducability - use the *same* `Dockerfile` in dev and production!
- Leverage the power of hub.docker.com wich has hundreds of official / non-official images
- Promotes 'The Twelve-Factor App' (12factor.net)

DOCKER: CHALLENGES

- How to manage / connect multiple containers
- Dependency of containers
- Version control
- How do my containers connect to each other?
- How do I manage base images?

PERCHERON

GITHUB.COM/ASHMCKENZIE/PERCHERON

PERCHERON

- Command line tool, leverages Docker API
- Extends upon the `docker build` and `docker run` fundamentals
- Similar in purpose to `docker-compose` (formerly `fig`)
- Builds Docker images & creates Docker containers
- Has a concept of Stacks (groups of containers)

ALSO.. A KICK ASS HORSE!

http://andreaschepisi.deviantart.com/art/Percheron-finished-76929325

PERCHERON: WHY?

- Make prototyping applications & services easier
- Existing tools did not have:
  - Base image management
  - Dependency management
  - Dependency graph generation
- Optimised for developer happiness!

PERCHERON: FEATURES

- Builds Docker images & creates Docker containers
- Base image generation & dependency management
- Version control
- Liquid templating of `Dockerfile`
- Support for secrets and userdata
- Dependency graph
- Written in Ruby!

PERCHERON: HOW?

- Define a `.percheron.yml`
- Similar format to `docker-compose.yml`
- Utilise `Dockerfiles` or pull down Docker images
- Fundamental pieces:
  - Units (Docker container)
  - Stacks (groups of Docker containers)

PERCHERON: CONFIG

```yaml
---
stacks:
  - name: infra-test
    description: Infrastructure units
    units:
      - name: redis
        docker_image: redis/redis:3
        version: 1.0.0
        ports:
          - 6379:6379
        start_args: [ "redis-server", "--port", "6379" ]
```

PERCHERON: COMMANDS

init	Initialise a new .percheron.yml
status	List stacks and its units
start	Start a stack
stop	Stop a stack
restart	Restart a stack
build	Build image(s) for a stack
create	Build image(s) & create units
purge	Purge a stack
shell	Shell into a unit
logs	Show logs for a unit
graph	Generate a stack graph

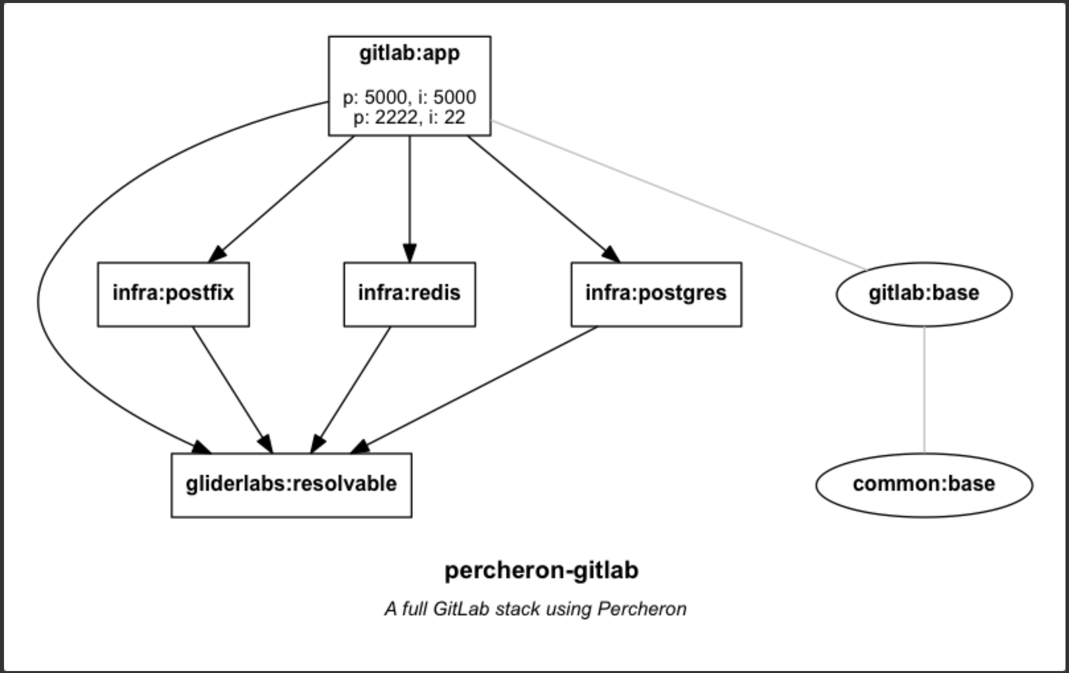
DEMO

V0.9.0-DEV

GITLAB

- Open Source Git hosting (like GitHub)
- Rails 4
- Sidekiq
- PostgresSQL
- redis
- postfix (SMTP)

GITLAB DEPENDENCY DIAGRAM



PERCHERON ROADMAP

V1.0

- Improve version control
  - Add `release` subcommand with `--patch`, `--minor` and `--major` flags
- Add `push` subcommand which pushes Docker images up to a Docker Registry
- Support inheriting ENV variables from executing shell
- Bug smash!

PERCHERON

GITHUB.COM/ASHMCKENZIE/PERCHERON