



TIPS FOR USING DOCKER

Nisha Lad @nishalad95

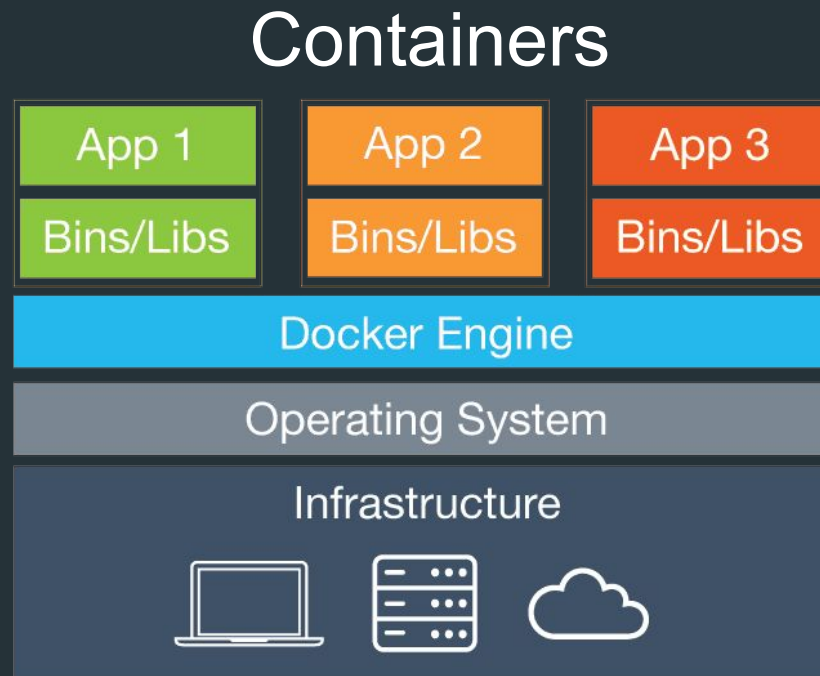
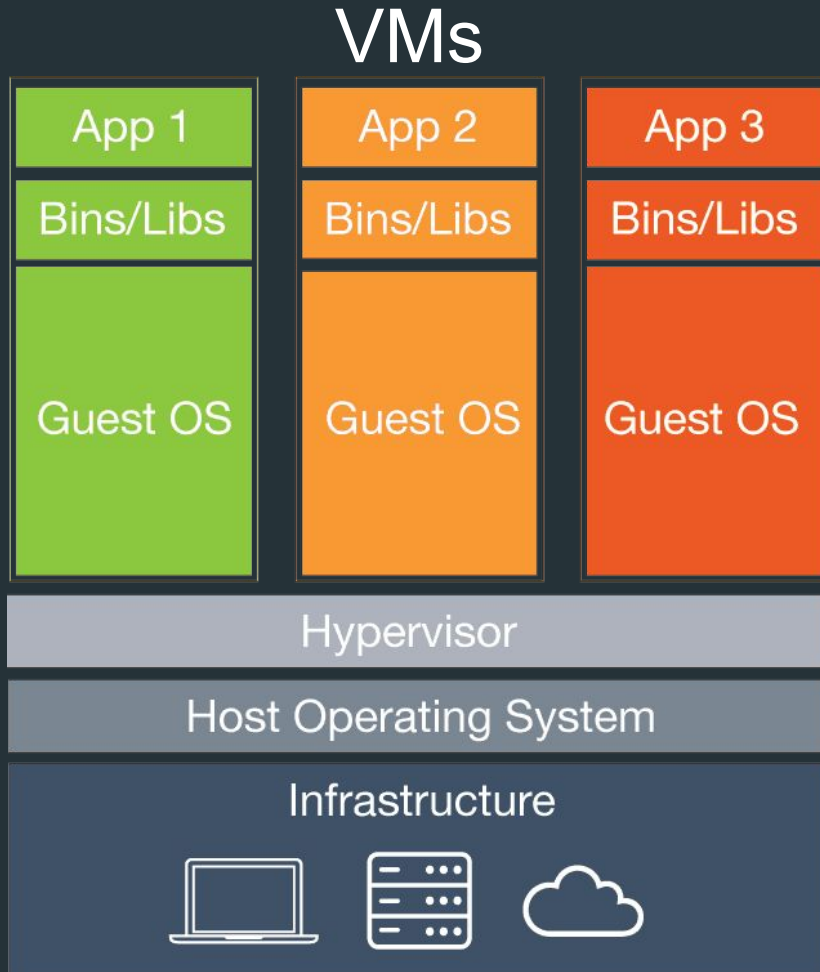
Graduate Cloud Developer Oracle

WHAT IS DOCKER?

- ▶ Open Source!
- ▶ Container-based system
- ▶ Docker wraps up the application & all of its dependencies into a software container
- ▶ Automates the deployment of applications
- ▶ Independent of underlying OS of the host system
- ▶ Simple to use!



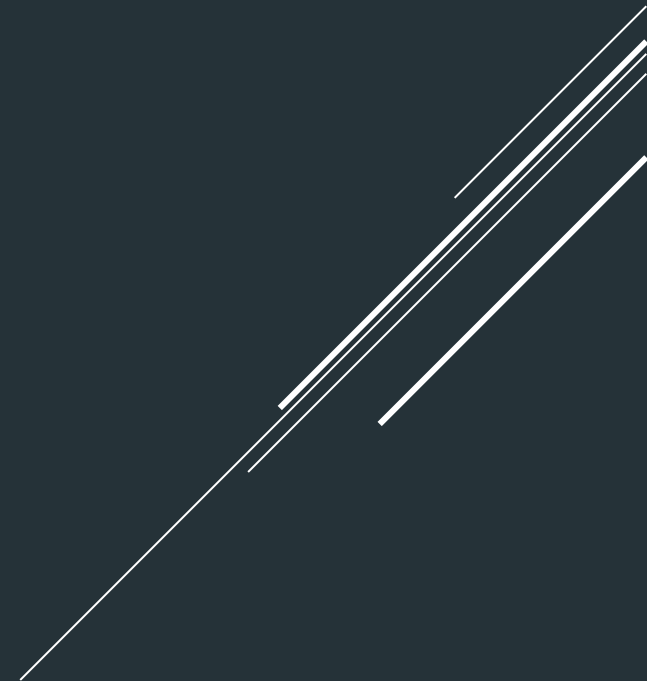
THE CONTAINER WORLD...



WHAT IS KUBERNETES?



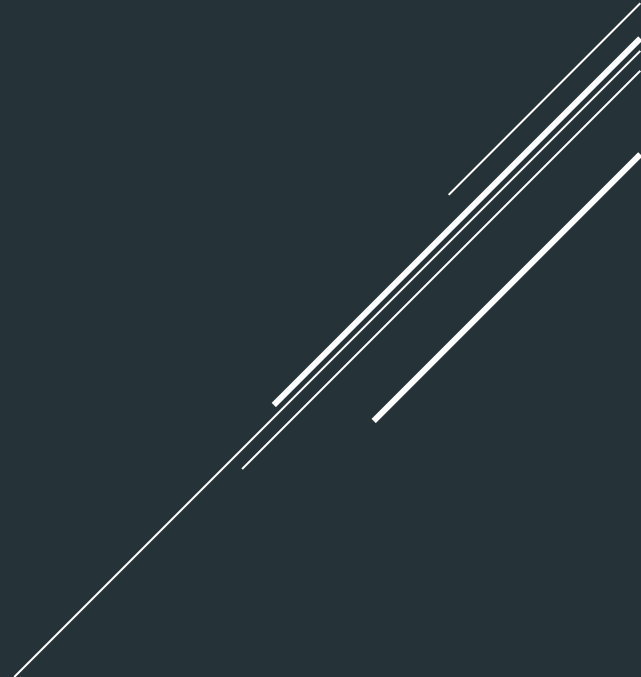
- ▶ Scale up containers
- ▶ Container Cluster Management System
- ▶ DevOps focused
- ▶ Originally designed by Google

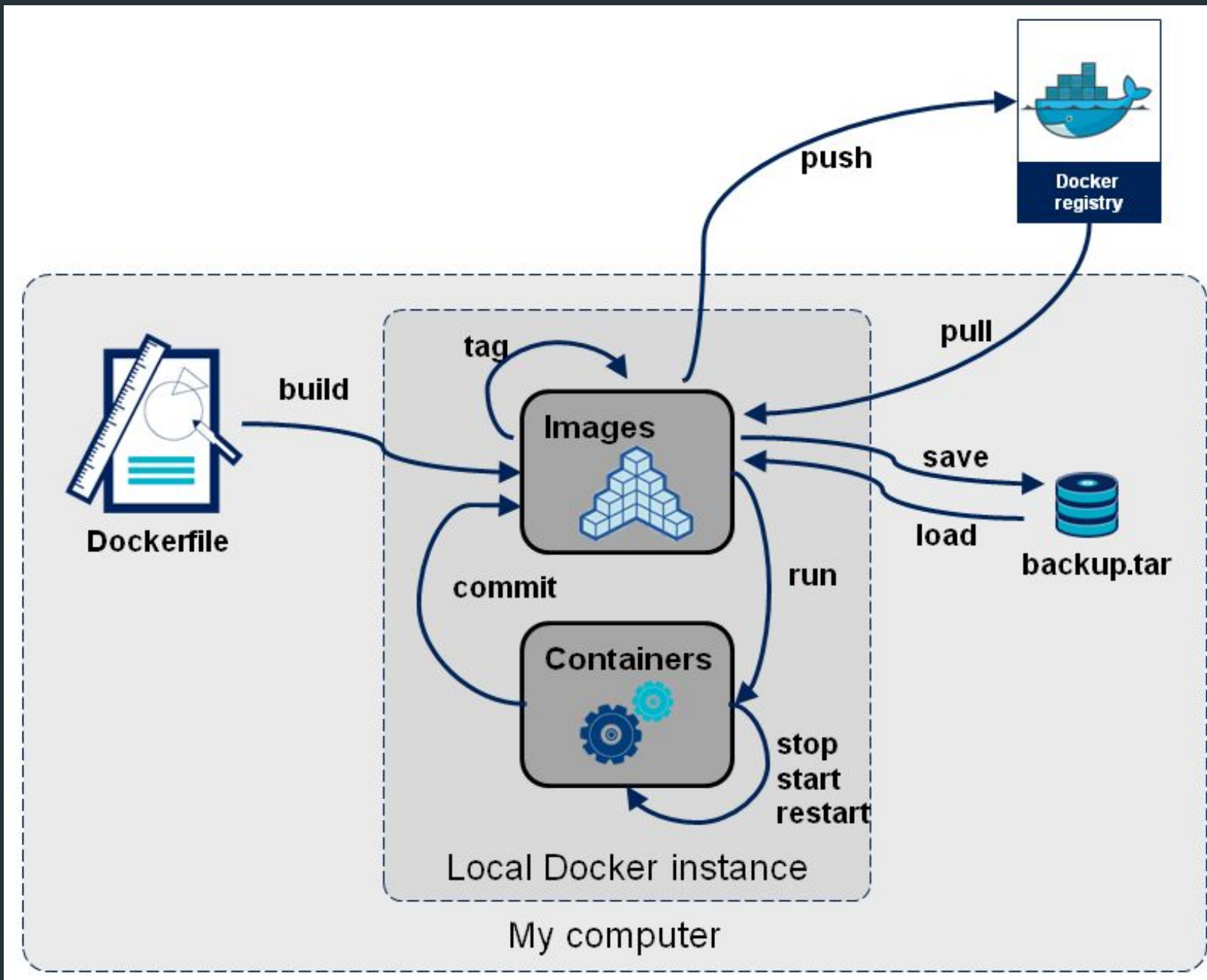


MICROSERVICES



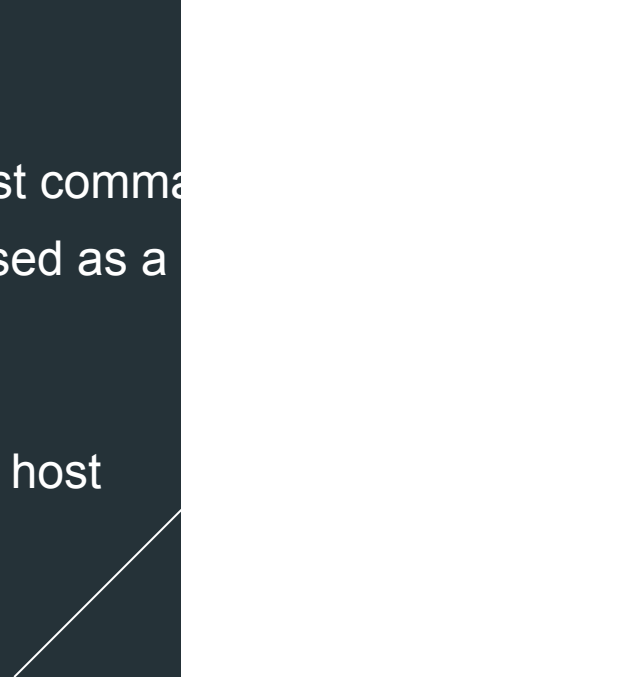
- ▶ Microservices architecture: smaller modular processes
- ▶ Docker is well suited to microservices architecture
- ▶ Agile methodology practises, efficiency for developers
- ▶ Continuous delivery, continuous integration pipelines
- ▶ Hudson & Jenkins





DOCKERFILE

Command	Description
ADD	Copies a file from the host system onto the container
CMD	The command that runs when the container starts
ENV	Sets an environment variable in the new container
EXPOSE	Opens a port for linked containers
FROM	The base image to use in the build. Mandatory and must be the first command
ONBUILD	A command that is triggered when the image in the Dockerfile is used as a base for another image
RUN	Executes a command and saves the results as a new layer
VOLUME	Creates a shared volume that can be shared among containers by host machine
WORKDIR	Set the default working directory for the container



DOCKERFILE EXAMPLE

```
# comment
```

```
INSTRUCTIONS arguments
```

```
# our base image
```

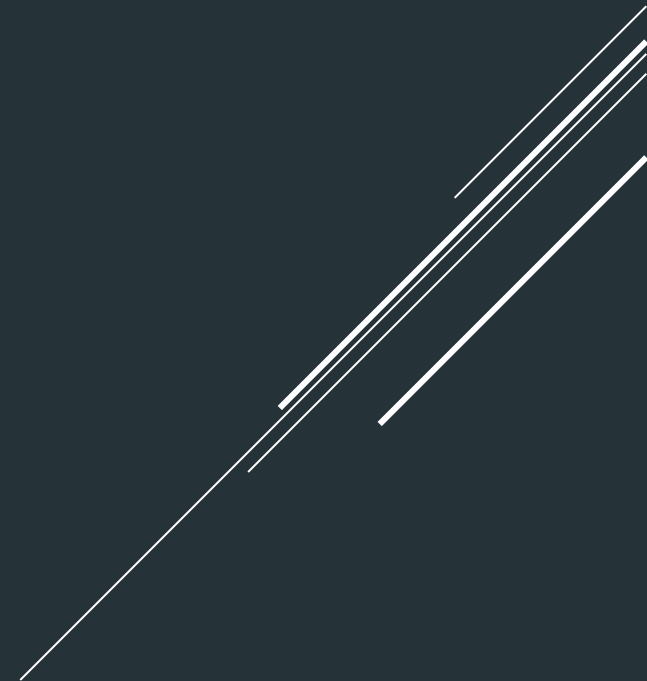
```
FROM python:3-onbuild
```

```
# specify the port number the container  
should expose
```

```
EXPOSE 5000
```

```
# run the application!
```

```
CMD ["python", "./myapp.py"]
```



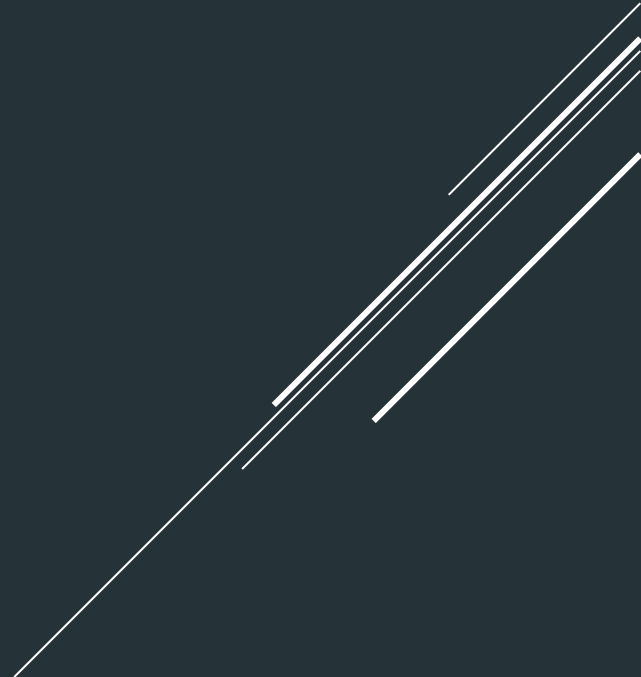
DOCKERFILE EXAMPLE

```
$ docker build -t prakhar1989/catnip
Sending build context to Docker daemon 8.704 kB
Step 1 : FROM python:3-onbuild
# Executing 3 build triggers...
Step 1 : COPY requirements.txt /usr/src/app/
---> Using cache
Step 1 : RUN pip install --no-cache-dir -r requirements.txt
---> Using cache
Step 1 : COPY . /usr/src/app
---> 1d61f639ef9e
Removing intermediate container 4de6ddf5528c
Step 2 : EXPOSE 5000
---> Running in 12cfcf6d67ee
---> f423c2f179d1
Removing intermediate container 12cfcf6d67ee
Step 3 : CMD python ./app.py
---> Running in f01401a5ace9
---> 13e87ed1fbc2
Removing intermediate container f01401a5ace9
Successfully built 13e87ed1fbc2
```

Replace with your
username on Docker
Hub and application
name

DOCKERFILE EXAMPLE

```
$ docker run -p 8888:5000 <userName>/<appName>  
* Running on http://0.0.0.0:5000/ (Press CTRL+ to quit)
```



EXAMPLE COMMANDS:

```
docker pull python:2.7
```

Pull image from DockerHub

```
docker images
```

Shows all images

```
docker rmi python:2.7
```

Remove an image

```
docker ps
```

Show only running containers

```
docker ps -a
```

Shows all containers

```
docker build -t <image_name>
```

Create your own image

```
docker logs -f <container_name>
```

Check the logs of container

```
docker load -i <CONTAINER_FILE>.tar
```

Restoring docker container

```
docker rm $(docker ps -a -q)
```

Deletes all containers but fails on the ones running

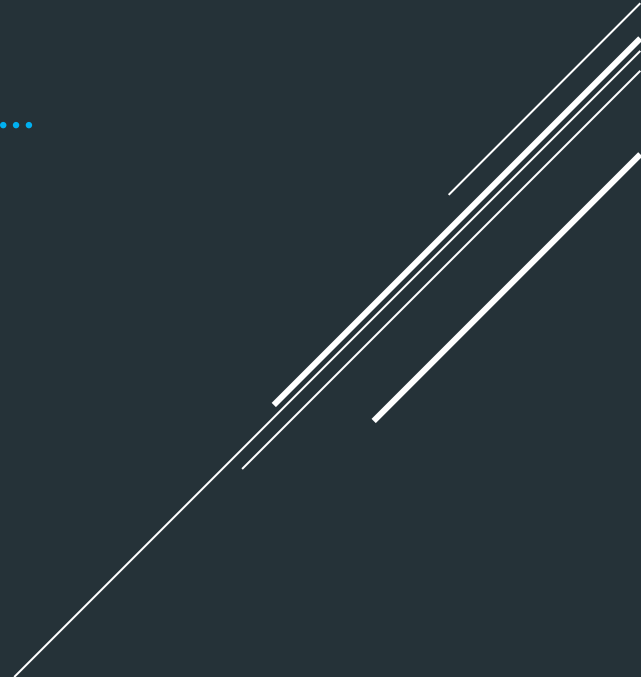
CONFIGURE DOCKER PS OUTPUT

- Default output of `docker ps`

```
$ docker ps
```

```
CONTAINER ID    IMAGE    COMMAND    .....  
247wh3jddee0   nginx   "nginx -g daemon" .....
```

- This output takes up too much room – can get annoying!



CONFIGURE DOCKER PS OUTPUT

- It can be fixed!
- Solution is to use `--format` argument

```
$ docker ps --format \
    "table {{.Names}}\t{{.Image}}\t{{.Status}}"
```

```
NAMES      IMAGE      STATUS
web        nginx     Up 20 minutes
```

- Make the output in permanent format by setting an alias in `config.json`

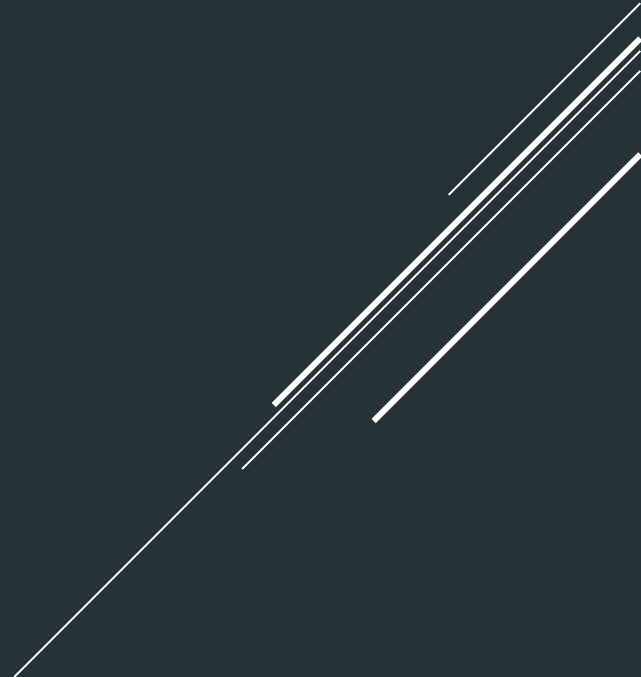
```
$ cat ~/.docker/config.json
{.....
  "psFormat": "table
  {{.Names}}\t{{.Image}}\t{{.Status}}"
```

PROCESS IDS

- Getting the ID of the last run container
- Very useful!

```
$ ID=$(docker run Ubuntu echo hello world)
hello world
$ docker commit $ID helloworld
24jkerndff9ue
```

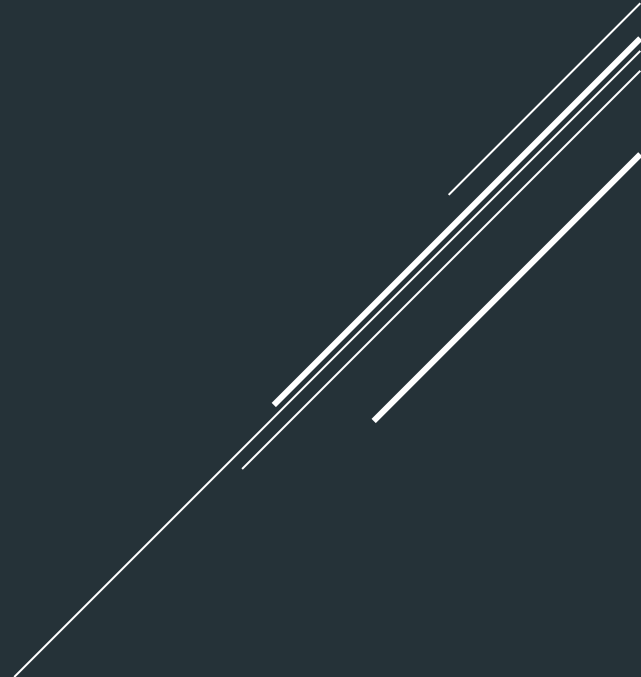
- But you have to keep assigning IDs!



PROCESS IDS

- Set an alias!

```
$ alias dl = `docker ps -l -q`  
$ docker run Ubuntu echo hello world  
hello world  
$ dl  
19733hdbwff9we  
$ docker commit `dl` helloworld  
ff09sjbew8hew3
```



USEFUL COMMANDS

- Deleting “dangling” images
- <none> images

```
$ docker image prune
```

```
WARNING! This will remove all dangling images
```

```
Are you sure you want to continue? [y/N] y
```

```
Deleted images:
```

```
deleted:
```

```
23j2j3kmcerkeddne09eea3goed232nfks..
```


```
.....
```

```
Total reclaimed space: 3GB
```


USEFUL COMMANDS

- Delete stopped containers

```
$ docker container prune
WARNING! This will remove all stopped containers
Are you sure you want to continue? [y/N] y
Deleted Containers:
deleted:
23j2j3kmcerkeddne09eea3goed232nfks..
.....
Total reclaimed space: 300MB
```



HEALTH CHECKS

- Used by docker to determine the “health” of a container

```
FROM nginx
```

```
RUN apt-get update && apt-get install -y curl
```

```
HEALTHCHECK --interval=10s
```

A decorative graphic consisting of several parallel white lines of varying lengths, slanted upwards from left to right, located in the bottom right corner of the slide.

SHRINKING A CONTAINER

- ▶ Smith CLI
 - ▶ Open Source!
 - ▶ Apache License 2.0
 - ▶ Universal Permissive License
- ▶ Command line tool for
 - ▶ Building Microcontainers
 - ▶ yum repos
 - ▶ rpm files
 - ▶ Shrinking existing containers
 - ▶ Standard container in Microcontainer out



USEFUL WEBSITES & WHERE TO LOOK

- ▶ **Docker 101 blog:**
<https://blog.docker.com/2016/05/docker-101-getting-to-know-docker/>
- ▶ **Docker Curriculum:**
<https://docker-curriculum.com/>
- ▶ **Liz Rice: Building a container from scratch in Go**
<https://www.youtube.com/watch?v=Utf-A4rODH8>
- ▶ **DigitalOcean**
- ▶ **Good defaults for node and docker – Bret Fisher**
<https://github.com/BretFisher/node-docker-good-defaults>
- ▶ **Walkthrough on Dockerfiles:**
<http://odewahn.github.io/docker-jumpstart>

