A Hands-on Intro to Containers

- Welcome!

- Login credentials are at
- https://www.dropbox.com/s/8fkxf4lm1ppjr9y/codemash2.log?dl=0
- https://bit.ly/2TBhpzb

- Please `mkdir inuse` to make sure we didn't double up on numbers

- Then SSH into the workstation as a test
- The first Docker command is: `docker run hello-world`

coveros
Agility. Security. Delivered.

A Hands-on Intro to Containers

Gene Gotimer

About Coveros

- Coveros helps companies accelerate the delivery of secure, reliable software using agile methods
- Services
  - Agile Transformations & Coaching
  - Agile Software Development
  - Agile Testing & Automation
  - DevOps and DevSecOps Implementations
  - Software Security Assurance & Testing
- Agile, DevOps, Test Automation, Security Training
- Open Source Products
  - SecureCI –Secure DevOps toolchain
  - Selenified – Agile test framework

Areas of Expertise

Selected Clients

Akamai FASTER FORWARD

vibrent

HUD

BNP PARIBAS

UNOS DONATE LIFE

NATIONAL CANCER INSTITUTE

DEPARTMENT OF THE AIR FORCE · UNITED STATES OF AMERICA · MCMXLVII

Symantec.

UnitedHealth Group

IFC International Finance Corporation World Bank Group

ADVENT®

DELTA DENTAL

EMC²

Liberty Mutual. INSURANCE

Aol.

ABIOMED

EMPLOYERS

UNITED STATES ARMY

Fannie Mae

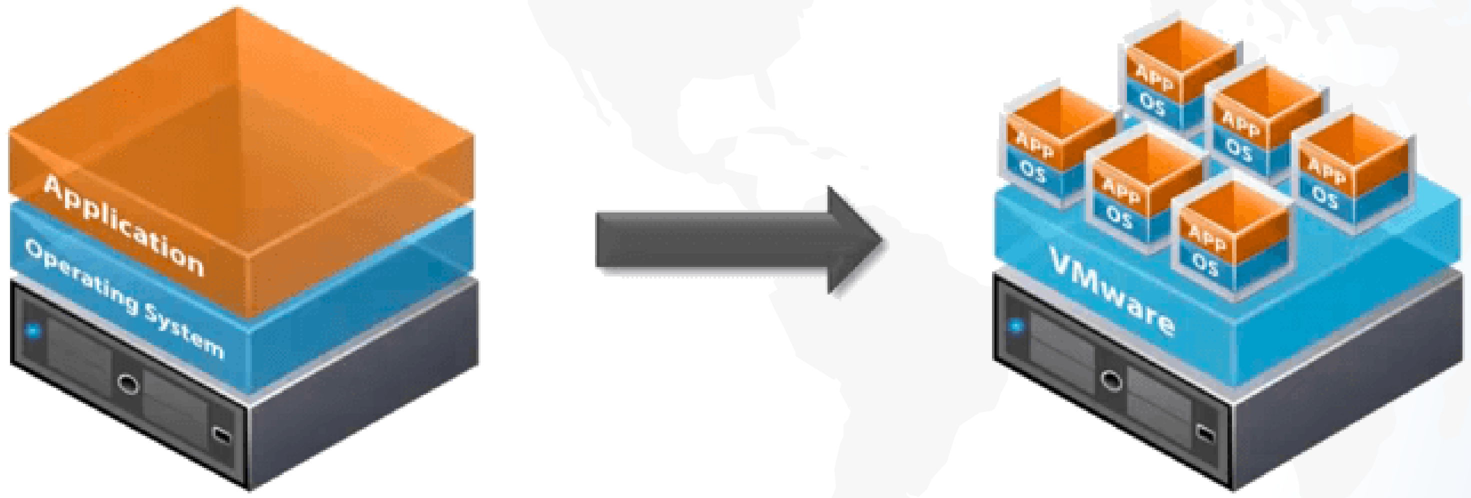
DEPARTMENT OF HEALTH & HUMAN SERVICES · USA

U.S. DEPARTMENT OF HOMELAND SECURITY

What are Containers?

Virtualization

- One physical system simulating multiple virtual systems
- Shared:
  - CPU
  - Memory
  - Networking
  - Hard drive
  - Peripherals



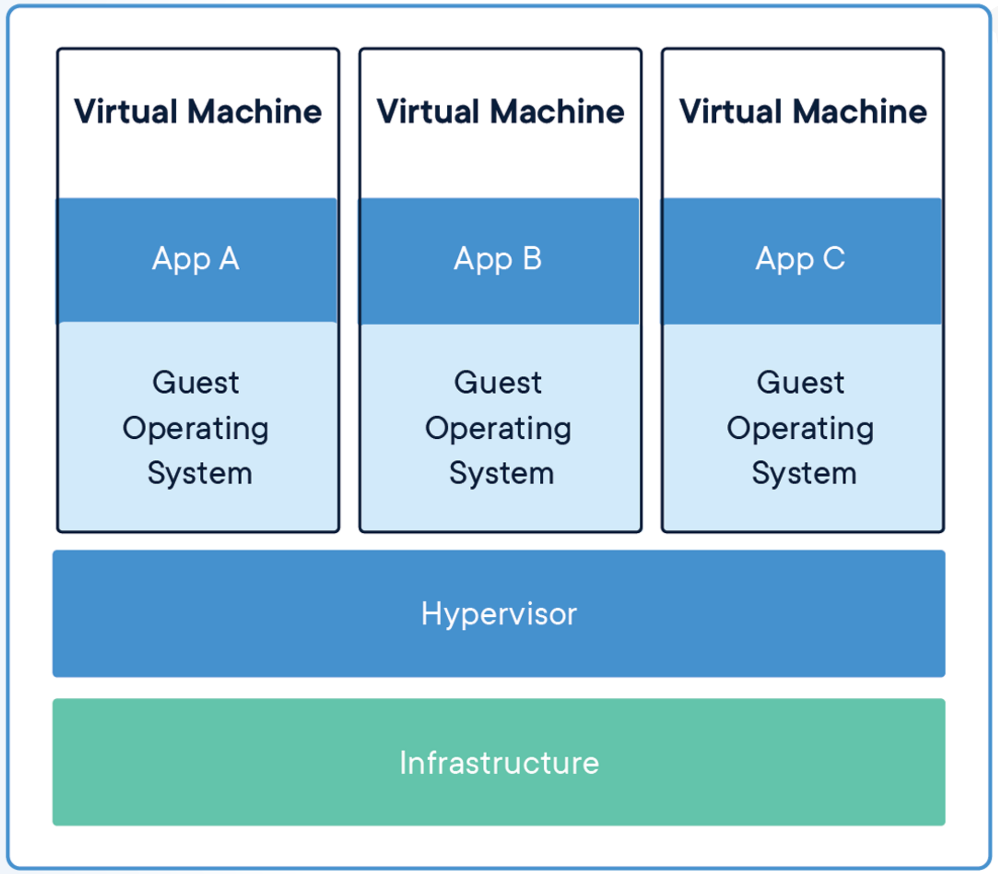
Traditional Architecture

- Single operating system
- Single application

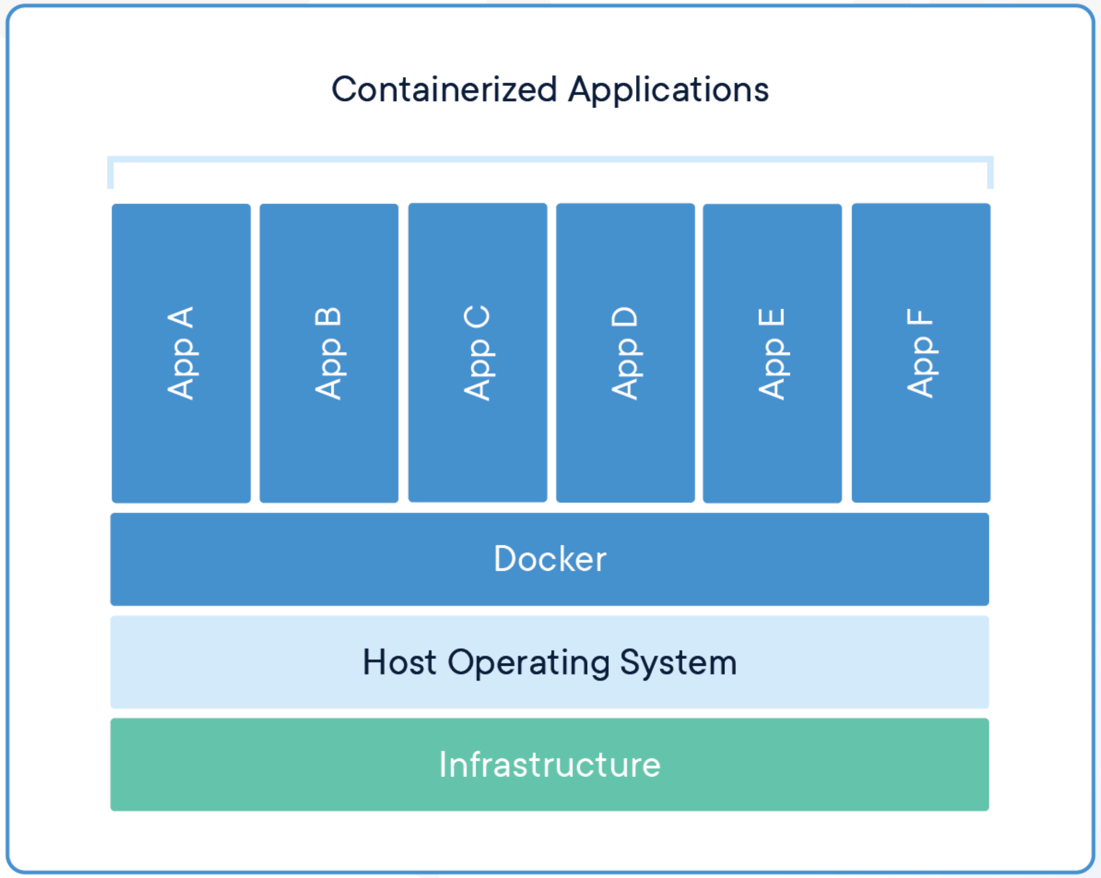
Virtual Architecture

- Virtualize many VMs using VMware Hypervisor

Containerization



Virtual Machines



Containers

Where should we use Containers?

Build Infrastructure

- Jenkins
  - Slaves are fresh Docker images created as needed
  - Always a clean build environment
  - Easy to parallelize

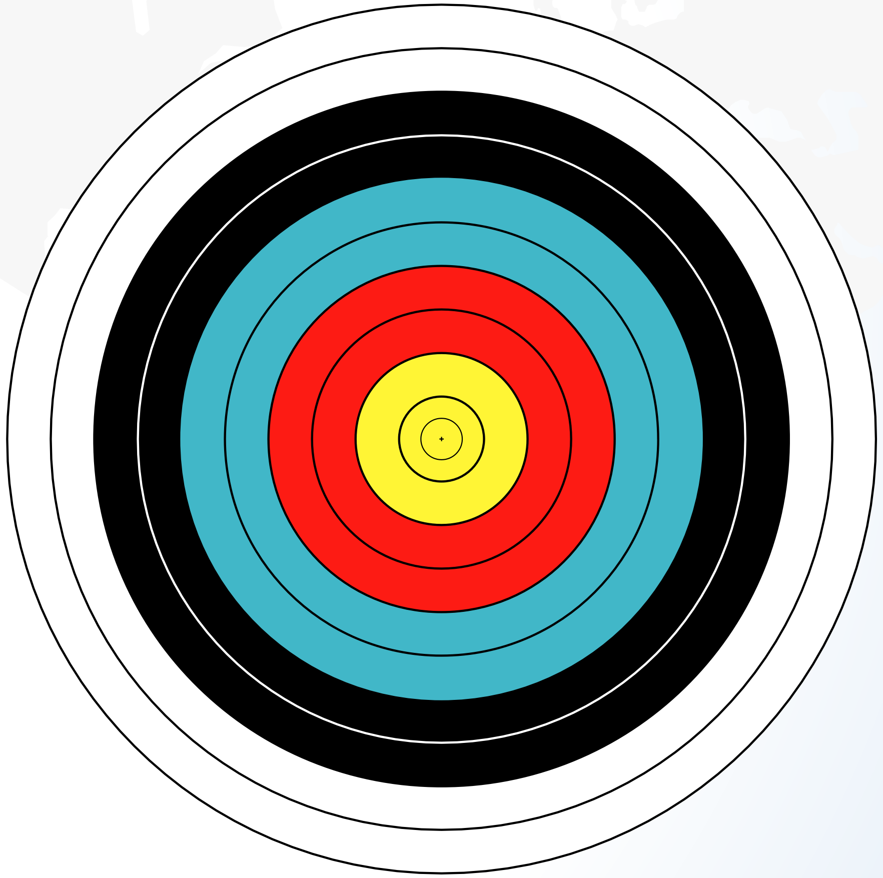
- Jenkins X
  - “opinionated way to do continuous delivery with Kubernetes”

Testing

- Selenium Grid Nodes
  - Easy to spin up nodes with different versions of different browsers
  - Makes parallel testing easy

Deployment

- System Under Test
  - Always a fresh system
  - No garbage built up from prior deploys
  - Everyone can have their own
  - Create and recreate on demand



Immutable

- Installed and configured
  - Just use as is
  - Deploy only once
  - No installation or configuration to do

- Can be passed through the SDLC as a single piece
  - Never changes, so what was tested is exactly what was deployed
  - No more “works on my machine”

Scalability

- Immutable, so they can easily be cloned
  - Spin up and shut down instances as needed
  - Elasticity

Experimentation

- Experiment with
  - New tools
  - New versions
  - New techniques
  - New configurations

Docker Basics

Docker

- Containerization software
- Free, Open-Source
- Runs on Windows, Mac, Linux
- Easy to install
- Works well on AWS, Azure, Google

https://www.docker.com/

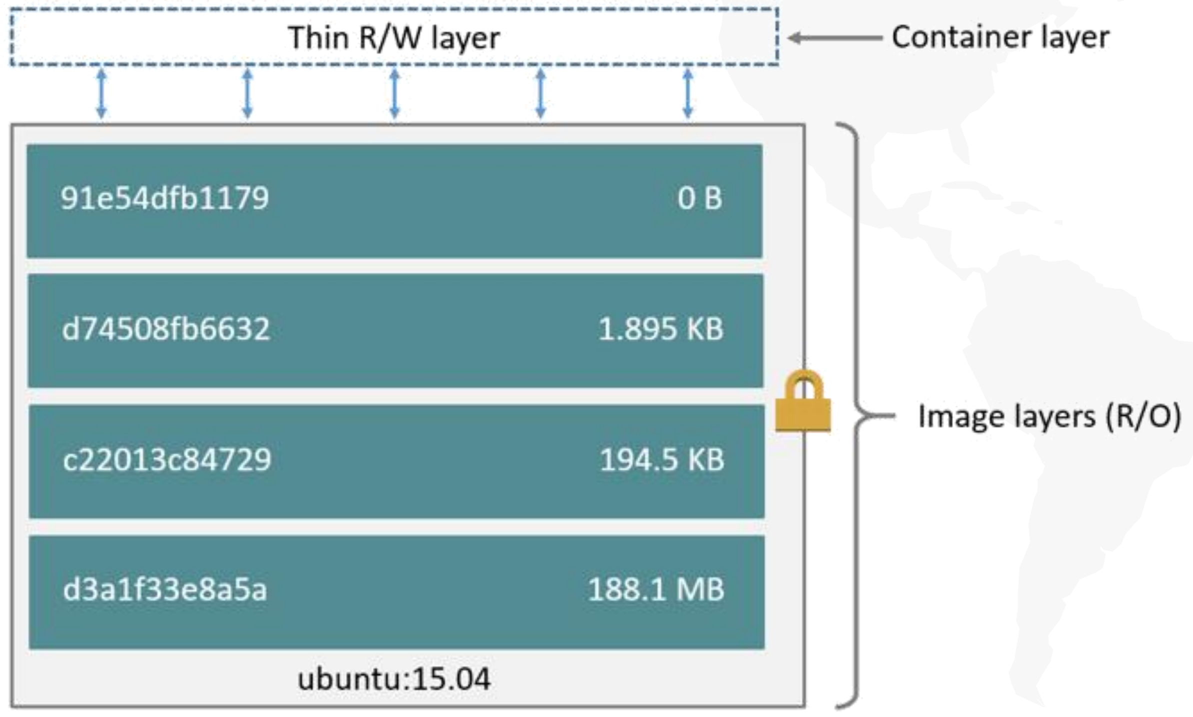
Value of Docker in Testing

- **No more “works on my machine”**
  - and its cousin, “you must have configured it wrong”
- Same binary deployed to different environments
- Simplifies deployments for testing
- Disposable test environments
- Browser versions, emulators into containers

- Stand up multiple environments => Run multiple tests in parallel

Layer

- A single instruction in the Dockerfile used to create it
  - Each layer is a set of differences from the layer below it



Container
(based on ubuntu:15.04 image)

Image and Container

- Image – software bundle that will run on Docker
- Container – running (or stopped) instance of an image
    - read-write layer overlaid

Registry

- Registry – cloud repository of Docker images
    - public or hosted
- Docker Hub – public Docker hosted registry https://hub.docker.com/
- Some alternatives:
    - Quay - https://quay.io/
    - Google GCR - https://cloud.google.com/container-registry/
    - Amazon ECR - https://aws.amazon.com/ecr/
    - Sonatype Nexus - https://www.sonatype.com/

Our First Docker Container

Our First Docker Container

```
$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world

b04784fba78d: Pull complete
Digest: sha256:f3b3b28a45160805bb16542c9531888519430e9e6d6ffc09d72261b0d26ff74f
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.
```

$ docker run

```
$ docker run hello-world
```

- downloads the `hello-world` image from Docker Hub (*if needed*)
- starts a new container with the image
- allocates a filesystem
- adds read-write layer
- allocates network interface
- sets up IP address
- executes process
- captures output
- exits container when process finishes

https://docs.docker.com/engine/reference/run/

Interacting with a Docker Container

```
$ docker run -it ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
d5c6f90da05d: Pull complete
1300883d87d5: Pull complete
c220aa3cfc1b: Pull complete
2e9398f099dc: Pull complete
dc27a084064f: Pull complete
Digest: sha256:34471448724419596ca4e890496d375801de21b0e67b81a77fd6155ce001edad
Status: Downloaded newer image for ubuntu:latest
root@7c046df68298:/# exit
exit

$ docker run -it ubuntu bash
root@7d096414d3e8:/# exit
exit
```

$ docker run -it

```
$ docker run -it ubuntu bash
```

- downloads the `ubuntu` image from Docker Hub (*if needed*)
- starts a new container with the image
- `-i` keeps STDIN open, so we can interact
- `-t` allocates a terminal
- executes `bash` process
- captures output
- exits container when process finishes

https://docs.docker.com/engine/reference/run/

Creating a Docker Image Interactively

- Docker images are layered
- Start with a base image, in `-it` mode
  - `docker run -it ubuntu bash`
- Add software, configuration, etc.

```
root@f706b83a0762:/# apt-get update
…
Reading package lists... Done
root@f706b83a0762:/# apt-get -y install fortune
…
root@f706b83a0762:/# /usr/games/fortune
You are capable of planning your future.
root@f706b83a0762:/# /usr/games/fortune
You will wish you hadn’t.
root@f706b83a0762:/#
```

Image names

- `hello-world`
- `ubuntu`
  - No repository given, so they come from the official repository
- `coveros/fortune1`
  - User name `coveros`, image `fortune1`
  - Use your own user name
- `coveros/fortune1:latest`
  - With a tag, defaults to `latest`

https://docs.docker.com/docker-hub/repos/#searching-for-images

Save the Interactive Docker Image

- In a separate window:

```
$ docker ps
CONTAINER ID  IMAGE   COMMAND  CREATED         STATUS        PORTS  NAMES
f706b83a0762  ubuntu  "bash"   2 minutes ago   Up 2 minutes         tender_franklin
$ docker commit f706b83a0762 coveros/fortune1
sha256:4550417358e8a572d2c52dc5d0a80093486d104613722260970 9e4e14cad36c9
$ docker stop tender_franklin
$
```

- Notice the interactive session exits

```
root@f706b83a0762:/# /usr/games/fortune
You will wish you hadn’t.
root@f706b83a0762:/# exit
$
```

$ docker ps

```
$ docker ps
```

- show running containers

```
$ docker ps -a
```

- show all containers, running and stopped

https://docs.docker.com/engine/reference/commandline/ps/

$ docker stop

```
$ docker stop tender_franklin
```

- stop the container with the randomly assigned tag

```
$ docker stop f706b83a0762
```

- stop the container with the container ID
  - at least 5 digits

https://docs.docker.com/engine/reference/commandline/stop/

$ docker logs

```
$ docker logs tender_franklin
```

- show output of running or stopped container

https://docs.docker.com/engine/reference/commandline/logs/

Run the Docker Image

- Run the image and the command

```
$ docker run coveros/fortune1 /usr/games/fortune
Next Friday will not be your lucky day.  As a matter of fact, you don't
have a lucky day this year.
$
```

A Simple Dockerfile

Dockerfile

- Contains description of the image
- Start with an empty directory
  - everything underneath will be in the Docker image

```
FROM ubuntu:latest
RUN apt-get update && apt-get install -y fortune
CMD /usr/games/fortune
```

https://docs.docker.com/engine/reference/builder/

Create the Dockerfile

- Create the text file
  - File name must be Uppercase: **D**ockerfile

```
$ cd
$ mkdir fortune2
$ cd fortune2
$ nano Dockerfile
FROM ubuntu:latest
RUN apt-get update && apt-get install -y fortune
CMD /usr/games/fortune
<ctrl-X>
Save modified buffer? Y
File Name to Write: Dockerfile <enter>
$ ls
Dockerfile
$
```

Create a Docker Image Using Dockerfile

- Build the image

```
$ docker build -t coveros/fortune2 .
Sending build context to Docker daemon 3.072 kB
Step 1/3 : FROM ubuntu:latest
 ---> ccc7a11d65b1
Step 2/3 : RUN apt-get update && apt-get install -y fortune
 ---> Running in c7f99ebca76d
…
Reading package lists...
…
 ---> 9c1f4424ee9a
Removing intermediate container c7f99ebca76d
Step 3/3 : CMD /usr/games/fortune
 ---> Running in 544ec842f7ad
 ---> 2947151ced07
Removing intermediate container 544ec842f7ad
Successfully built 2947151ced07
Successfully tagged coveros/fortune2:latest
$
```

$ docker build

```
$ docker build -t coveros/fortune2 .
```

- builds an image using the Dockerfile in this directory
- `-t` tags the image as coveros/fortune2
- `.` includes the files in and below this directory

https://docs.docker.com/engine/reference/commandline/build/

Build and Run

```
$ docker build -t coveros/fortune2 .
…
Successfully built 9cbbefdc8909
Successfully tagged coveros/fortune2:latest
$ docker run coveros/fortune2
You feel a whole lot more like you do now than you did when you used to.
$
```

List the Docker Images

- List the images you have pulled and created

```
$ docker images
REPOSITORY         TAG            IMAGE ID        CREATED              SIZE
coveros/fortune2   latest         34d514f303ed    About a minute ago   112MB
coveros/fortune1   latest         51c736c22ea7    21 minutes ago       112MB
ubuntu             latest         cd6d8154f1e1    2 days ago           85.8MB

$ docker images coveros/fortune2
REPOSITORY         TAG            IMAGE ID        CREATED              SIZE
coveros/fortune2   latest         34d514f303ed    About a minute ago   112MB
$
```

https://docs.docker.com/engine/reference/commandline/images/

Layers of the Docker Image

- Look at how the image was assembled, layer by layer

```
$ docker history coveros/fortune2
IMAGE               CREATED             CREATED BY                                      SIZE
34d514f303ed        12 minutes ago      /bin/sh -c #(nop)  CMD ["/bin/sh" "-c" "/usr…   0B
ae4054d2a21e        12 minutes ago      /bin/sh -c apt-get update && apt-get install…   26.6MB
cd6d8154f1e1        2 days ago          /bin/sh -c #(nop)  CMD ["/bin/bash"]            0B
<missing>           2 days ago          /bin/sh -c mkdir -p /run/systemd && echo 'do…   7B
<missing>           2 days ago          /bin/sh -c sed -i 's/^#\s*\(deb.*universe\)$…   2.76kB
<missing>           2 days ago          /bin/sh -c rm -rf /var/lib/apt/lists/*          0B
<missing>           2 days ago          /bin/sh -c set -xe   && echo '#!/bin/sh' > /…   745B
<missing>           2 days ago          /bin/sh -c #(nop) ADD file:3df374a69ce696c21…   85.8MB
$
```

https://docs.docker.com/engine/reference/commandline/history/

A Bigger Docker Image

Create a new Dockerfile

- New directory

```
$ cd
$ mkdir java-hello-world
$ cd java-hello-world
$ nano Dockerfile
```

- Exit and save

```
<ctrl-X>
Save modified buffer? Y
File Name to Write: Dockerfile <enter>
$ ls
Dockerfile
$
```

Dockerfile

```
FROM ubuntu:latest
RUN apt-get update && apt-get install -y --no-install-recommends \
    default-jdk-headless maven git \
    && rm -rf /var/lib/apt/lists/*
RUN git clone https://github.com/Coveros/helloworld.git
WORKDIR helloworld
RUN mvn clean package
CMD ["java", "-cp", "/helloworld/target/helloworld-1.0.jar", \
    "com.coveros.demo.helloworld.HelloWorld"]
```

Build, Build, and Run, Run

```
$ docker build -t coveros/java-hello-world .
…
Successfully built 14762192d0d4
Successfully tagged coveros/java-hello-world:latest
$ docker build -t coveros/java-hello-world .
…
Successfully built 14762192d0d4
Successfully tagged coveros/java-hello-world:latest
$ docker run coveros/java-hello-world
Hello, World!
$ docker run coveros/java-hello-world
Hello, World!
$
```

Examine the Docker Image

```
$ docker images coveros/java-hello-world
REPOSITORY                    TAG         IMAGE ID            CREATED              SIZE
coveros/java-hello-world      latest      14762192d0d4        14 seconds ago       490MB
$ docker history coveros/java-hello-world
IMAGE          CREATED          CREATED BY                                       SIZE
bb59acf0c25c   40 seconds ago   /bin/sh -c #(nop)  CMD ["java" "-cp" "/hello…    0B
6e26bb0c59b6   41 seconds ago   /bin/sh -c mvn clean package                     9.11MB
e94ce6834c5d   2 minutes ago    /bin/sh -c #(nop) WORKDIR /helloworld            0B
92d439a61bdd   2 minutes ago    /bin/sh -c git clone https://github.com/Cove…    24.4kB
2a15501e7b6a   2 minutes ago    /bin/sh -c apt-get update && apt-get install…    395MB
cd6d8154f1e1   2 days ago       /bin/sh -c #(nop)  CMD ["/bin/bash"]             0B
<missing>      2 days ago       /bin/sh -c mkdir -p /run/systemd && echo 'do…    7B
<missing>      2 days ago       /bin/sh -c sed -i 's/^#\s*\(deb.*universe\)$…    2.76kB
<missing>      2 days ago       /bin/sh -c rm -rf /var/lib/apt/lists/*           0B
<missing>      2 days ago       /bin/sh -c set -xe   && echo '#!/bin/sh' > /…    745B
<missing>      2 days ago       /bin/sh -c #(nop) ADD file:3df374a69ce696c21…    85.8MB
$
```

A Smaller Docker Image

Dockerfile

```
FROM ubuntu:latest AS development
RUN apt-get update && apt-get install -y --no-install-recommends \
    default-jdk-headless maven git \
    && rm -rf /var/lib/apt/lists/*
RUN git clone https://github.com/Coveros/helloworld.git
WORKDIR helloworld
RUN mvn clean package

FROM openjdk:alpine AS runtime
COPY --from=development /helloworld/target/helloworld-1.0.jar /

CMD ["java", "-cp", "/helloworld-1.0.jar", \
    "com.coveros.demo.helloworld.HelloWorld"]
```

Multi-stage Builds

- Reuse files from another image
  - multiple FROM statements

```
FROM ubuntu:latest AS development
…
FROM openjdk:alpine AS runtime
COPY --from=development /helloworld/target/helloworld-1.0.jar /
```

https://docs.docker.com/develop/develop-images/multistage-build/

https://docs.docker.com/develop/develop-images/dockerfile_best-practices/

Alpine Linux

- Linux distribution
- Small, ~5 MB
- Based on BusyBox, another small Linux
- Large package repository
- Security oriented

https://alpinelinux.org/

Build, Build, and Run, Run

```
$ docker build -t coveros/java-hello-world2 .
…
Removing intermediate container bf1d70dac851
 ---> 13fa2ddbbacd
Successfully built 13fa2ddbbacd
Successfully tagged coveros/java-hello-world2:latest
$ docker build -t coveros/java-hello-world2 .
…
Successfully built 13fa2ddbbacd
Successfully tagged coveros/java-hello-world2:latest
$ docker run coveros/java-hello-world2
Hello, World!
$ docker run coveros/java-hello-world2
Hello, World!
$
```

Examine the smaller Docker Image

```
$ docker images coveros/java-hello-world2
REPOSITORY                   TAG        IMAGE ID         CREATED             SIZE
coveros/java-hello-world2    latest     13fa2ddbbacd     18 seconds ago      103MB
$ docker history coveros/java-hello-world2
IMAGE        CREATED         CREATED BY                                        SIZE
13fa2ddbbacd 25 seconds ago  /bin/sh -c #(nop)  CMD ["java" "-cp" "/hello…    0B
655a744699b9 27 seconds ago  /bin/sh -c #(nop) COPY file:e740c2ab8f475954…    2.67kB
5801f7d008e5 8 weeks ago     /bin/sh -c set -x  && apk add --no-cache   o…    98.2MB
<missing>    8 weeks ago     /bin/sh -c #(nop)  ENV JAVA_ALPINE_VERSION=8…    0B
<missing>    8 weeks ago     /bin/sh -c #(nop)  ENV JAVA_VERSION=8u171        0B
<missing>    8 weeks ago     /bin/sh -c #(nop)  ENV PATH=/usr/local/sbin:…    0B
<missing>    8 weeks ago     /bin/sh -c #(nop)  ENV JAVA_HOME=/usr/lib/jv…    0B
<missing>    8 weeks ago     /bin/sh -c {   echo '#!/bin/sh';   echo 'set…    87B
<missing>    8 weeks ago     /bin/sh -c #(nop)  ENV LANG=C.UTF-8              0B
<missing>    2 months ago    /bin/sh -c #(nop)  CMD ["/bin/sh"]               0B
<missing>    2 months ago    /bin/sh -c #(nop) ADD file:25f61d70254b9807a…    4.41MB
$
```

Using pre-built images

Nginx

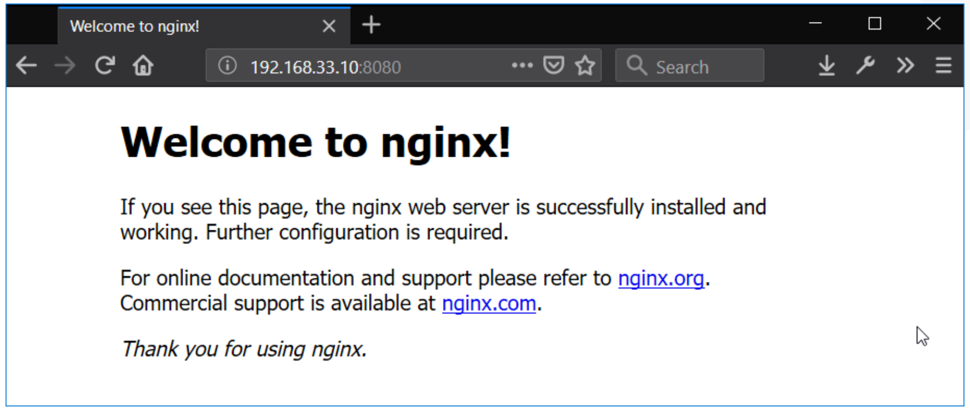
- Web server
  - also load balancer, proxy
- Fast
- Free, Open-Source

https://nginx.org/

Start nginx by hand

- Start nginx and map the port 8080 locally to port 80 on the container

```
$ docker run --name www -d -p 8080:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
802b00ed6f79: Pull complete
e9d0e0ea682b: Pull complete
d8b7092b9221: Pull complete
Digest: sha256:24a0c4b4a4c0eb97a1aabb8e29f18e917d05abfe1b7a7c07857230879ce7d3d3
Status: Downloaded newer image for nginx:latest
0e365c7ffdd178513bd4d5caaeefc085e6f1b451eb54831c3cdb12e14900ef73
```



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

$ docker run

```
$ docker run --name www -d -p 8080:80 nginx
```

- downloads the `nginx` image from Docker Hub (*if needed*)
- starts a new container with the image
- `--name www` assigns the name `www` to the container
- `-d` runs detached in the background so it stays running
- `-p 8080:80` map port `8080` on the host to port `80` on the container

https://docs.docker.com/engine/reference/run/

Stop nginx

- Stop nginx and remove the container
  - because we left it running
  - so we can reuse the name

```
$ docker stop www
www
$ docker rm www
www
$
```

$ docker rm

```
$ docker rm www
```

- remove a stopped container

```
$ docker rm --force www
```

- remove a running or stopped container

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

- remove all stopped containers

https://docs.docker.com/engine/reference/commandline/rm/

Host a static website by hand

```
$ cd
$ git clone https://github.com/Coveros/states-docker-demo.git states
Cloning into 'states'...
…
$ cd states/www
$ pwd
/home/ubuntu/states/www
$ docker run --name www \
       -v /home/ubuntu/states/www/html:/usr/share/nginx/html:ro \
       -d -p 80:80 nginx
8f2fd8847d1ced22e49e1cdf6c55a66630ad72be4202b915a3571b0be7d60fe1
$
```

$ docker run

```
$ docker run --name www \
    -v /home/ubuntu/states/www/html:/usr/share/nginx/html:ro \
    -d -p 80:80 nginx
```

- Same as before, plus
- `-v` mount local path `/home/ubuntu/states/www/html` in the container at `/usr/share/nginx/html` as read-only
- `-p 80:80` map port `80` on the host to port `80` on the container

https://docs.docker.com/engine/reference/run/

Stop nginx

- Stop nginx and remove the container

```
$ docker rm --force www
www
$
```

Assembling a Full Environment

PHP

- Server-side scripting language
  - Designed for web development
- Easy to read
- Very popular
- Free

https://php.net/

Redis

- In-memory data store
  - Key-value pairs
- Fast
- Interfaces for most languages
  - Including PHP
- Free, Open-Source

https://redis.io/

Stand up a full environment

```
$ cd ~/states
$ docker-compose up -d
Creating network "states_default" with the default driver
Building php
…
Creating states_www_1 ...
Creating states_php_1 ...
Creating states_redis_1 ... done
$
```

- We won’t type the docker-compose.yml
  - It is already in the directory
  - We will just review it

docker-compose.yml

```yaml
version: '2'

services:
  www:
    image: nginx:latest
    ports:
      - "80:80"
    volumes:
      - ./www/html:/html
      - ./www/site.conf:/etc/nginx/conf.d/default.conf
```

docker-compose.yml - Part 2

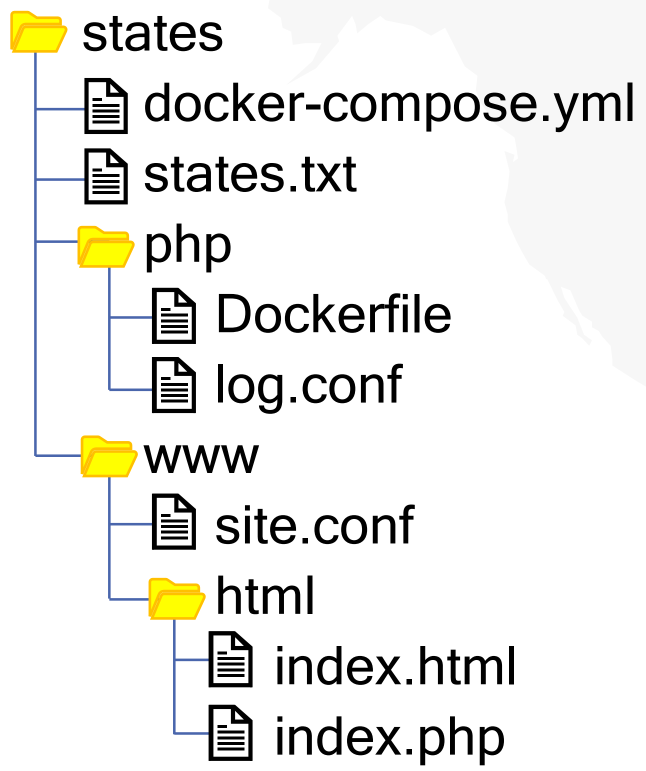
```
php:
    build: php
    volumes:
      - ./www/html:/html
      - ./php/log.conf:/usr/local/etc/php-fpm.d/zz-log.conf

  redis:
    image: redis:latest
    ports:
      - "6379:6379"
```

php/Dockerfile

- Start with a Docker Hub image
- Add an additional extension for Redis

```
FROM php:7-fpm
RUN pecl install redis \
  && docker-php-ext-enable redis
```

Docker Compose Directory Structure



```
states
├── docker-compose.yml
├── states.txt
├── php
│   ├── Dockerfile
│   └── log.conf
└── www
    ├── site.conf
    └── html
        ├── index.html
        └── index.php
```

https://docs.docker.com/compose/reference/

Load some data into Redis

- Feed a hash of states and abbreviations in bulk

```
$ cd ~/states
$ cat states.txt | redis-cli --pipe
All data transferred. Waiting for the last reply...
Last reply received from server.
errors: 0, replies: 50
$
```

Load some data into Redis

- Add items interactively

```
$ redis-cli
127.0.0.1:6379> HSET states 'PR' 'Puerto Rico'
(integer) 1
127.0.0.1:6379> SET visits 1234
OK
127.0.0.1:6379> exit
$
```

Add a Selenium Grid

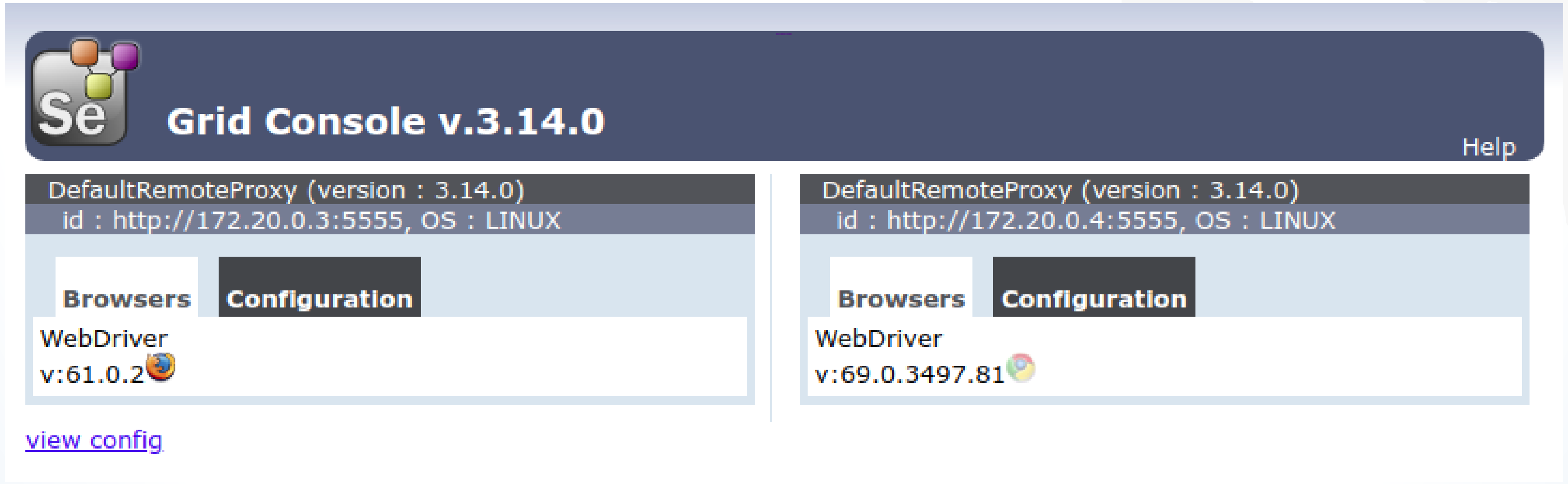
Selenium

- Web browser automation software
- Free, Open-Source
- Firefox, MSIE, Safari, Chrome, Opera
- C#, Java, JavaScript, Objective-C, PHP, Python, Ruby

https://www.seleniumhq.com/

Stand up Selenium Grid

```
$ cd ~/states/selenium
$ docker-compose up -d
Creating network "selenium_default" with the default driver
…
Creating selenium_hub_1 ...
Creating selenium_firefox_1 ...
Creating selenium_chrome_1 ... done
$
```



```
DefaultRemoteProxy (version : 3.14.0)
id : http://172.20.0.3:5555, OS : LINUX
WebDriver
v:61.0.2

DefaultRemoteProxy (version : 3.14.0)
id : http://172.20.0.4:5555, OS : LINUX
WebDriver
v:69.0.3497.81
```

Run the Java tests with Maven

- No Java or Maven installed on host
- So run it in a container that has them!

```
$ docker run -it --rm --name maven \
    -v "$(pwd)":/usr/src/maven \
    -v "$HOME/.m2":/root/.m2 \
    --workdir /usr/src/maven \
    --network selenium_default \
    maven:3.6.0-jdk-11 \
    mvn clean test -DtestUrl=http://44.55.66.77
[INFO] Scanning for projects...
…
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
…
$
```

Clean up

- Shut down the containers:

```
$ docker-compose stop
Stopping states_redis_1 ... done
Stopping states_php_1   ... done
Stopping states_www_1   ... done
$
```

- Remove the stopped containers

```
$ docker-compose rm -f
Going to remove states_redis_1, states_php_1, states_www_1
Removing states_redis_1 ... done
Removing states_php_1   ... done
Removing states_www_1   ... done
$
```

https://docs.docker.com/compose/reference/overview/

Questions?

Gene Gotimer

gene.gotimer@coveros.com

@CoverosGene