

Docker for CS518

CS518

Dr. Justin F. Brunelle

Why Docker

- Quickly create and remove services
- Less overhead and setup and VM
- Less taxing on the machine
- Not suitable for permanent services
- Text file config for sharing
- Isolate processes with local monitoring
- Quazi-isolated files and management within containers

Dockerfile Setup

- Nearly identical to tutum/lamp
- Dependencies vary slightly
- Setup:
 - `docker build -t jbrunelle/oducs518f16 .`
 - `docker run`
 - `--network nginx-proxy`
 - `-e VIRTUAL_HOST=cs518.cs.odu.edu`
 - `-v /home/jbrunelle/cs518_jfb/deploymentFiles:/var/www/html`

```
FROM ubuntu:trusty
```

```
MAINTAINER Justin F Brunelle <jfbrunel@odu.edu>
```

Metadata:
Base Container (OS)
and Maintainer

```
# Install packages
```

```
ENV DEBIAN_FRONTEND noninteractive
```

```
RUN apt-get update
```

```
RUN dpkg --configure -a
```

```
RUN apt-get -y install supervisor git
```

RUN – Execute commands
for installations

```
...
```

```
RUN sed -i 's/ServerTokens OS/ServerTokens Prod/g'  
/etc/apache2/conf-enabled/security.conf
```

```
RUN sed -i 's/ServerSignature On/ServerSignature Off/g'  
/etc/apache2/conf-enabled/security.conf
```

RUN – Execute commands
to modify files

```
# Add image configuration and scripts
ADD start-apache2.sh /start-apache2.sh
ADD start-mysqld.sh /start-mysqld.sh
ADD run.sh /run.sh
ADD import_sql.sh /import_sql.sh
RUN chmod 755 /*.sh
ADD my.cnf /etc/mysql/conf.d/my.cnf
ADD supervisord-apache2.conf
/etc/supervisor/conf.d/supervisord-apache2.conf
ADD supervisord-mysqld.conf /etc/supervisor/conf.d/supervisord-
mysqld.conf
```

ADD – copy over custom
config files*

* Docker now supports
“COPY” more broadly

```
# Remove pre-installed database
```

```
RUN rm -rf /var/lib/mysql/*
```

```
# Add MySQL utils
```

```
ADD create_mysql_admin_user.sh  
    /create_mysql_admin_user.sh
```

```
RUN chmod 755 /*.sh
```

MySQL Configuration

```
# config to enable .htaccess
```

```
ADD 000-default.conf /etc/apache2/sites-available/000-  
default.conf
```

```
RUN a2enmod rewrite
```

Setting up custom server
configurations (e.g.,
enable .htaccess)

```
# Configure /app folder with sample app
```

```
ADD . /app
```

```
RUN mkdir -p /app && rm -fr /var/www/html && ln -s /app  
/var/www/html
```

Create the /var/www/html/
directory

```
#Environment variables to configure php
```

```
ENV PHP_UPLOAD_MAX_FILESIZE 10M
```

```
ENV PHP_POST_MAX_SIZE 10M
```

```
# Add volumes for MySQL
```

```
VOLUME ["/etc/mysql", "/var/lib/mysql" ]
```

Set environment variables
for PHP

Map the machine's MySQL
directory to the container's

```
EXPOSE 80 3306
```

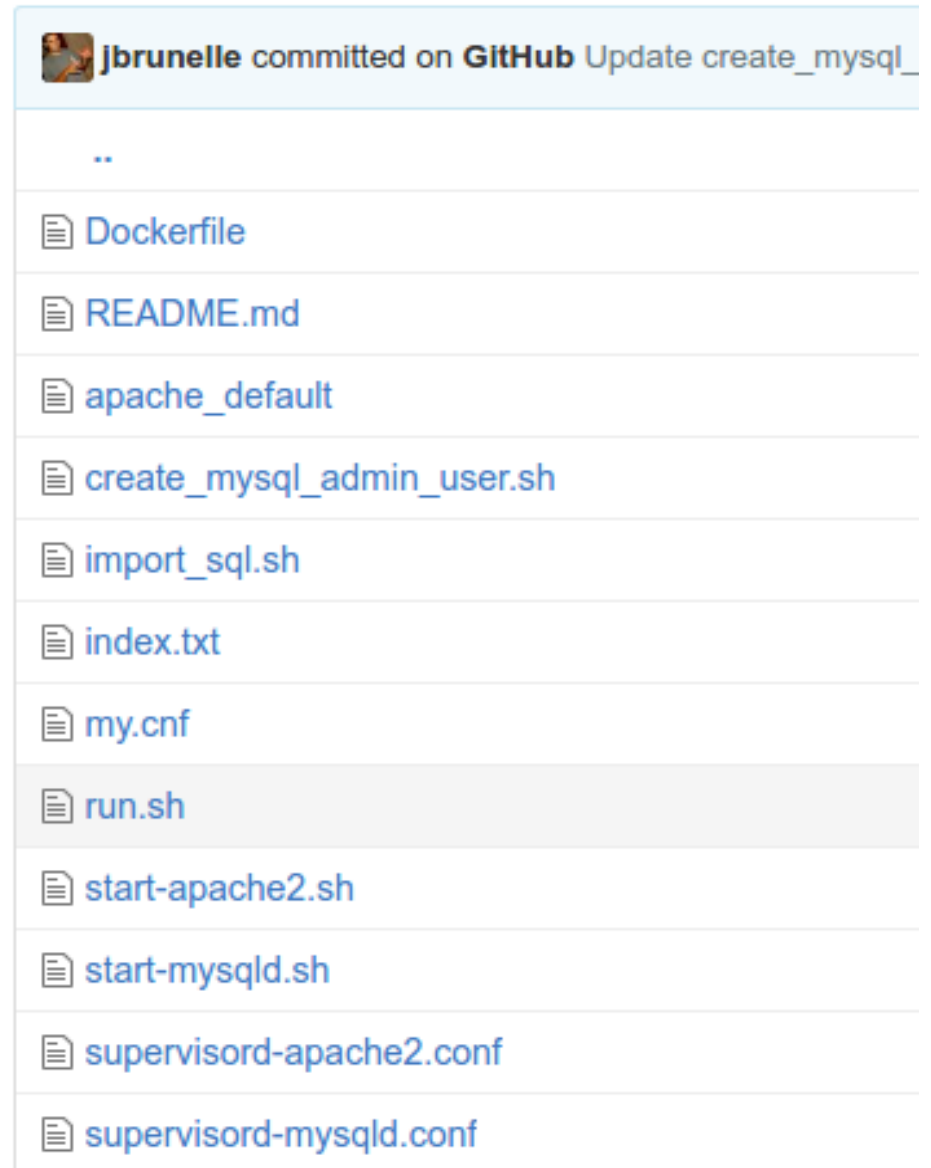
```
CMD ["/run.sh"]
```

Expose the container's
ports 80 (HTTP) and 3306
(MySQL) to the machine

Run the run.sh command

CS518 Files

- Custom and standard files
- Apache setup
 - No access to config files!
- SQL Setup
- Run.sh



000-default.conf

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/html
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/html>
        Options Indexes FollowSymLinks MultiViews
        # To make wordpress .htaccess work
        AllowOverride FileInfo
        Order allow,deny
        allow from all
    </Directory>
```

create_mysql_admin_user.sh Files

```
RET=1
```

```
while [[ RET -ne 0 ]]; do
```

```
    echo "=> Waiting for confirmation of MySQL service startup"
```

```
    sleep 5
```

```
    mysql -uroot -e "status" > /dev/null 2>&1
```

```
    RET=$?
```

```
done
```

```
#PASS=${MYSQL_PASS:-$(pwgen -s 12 1)}
```

```
PASS="M0n@rch$"
```

```
_word=$( [ ${MYSQL_PASS} ] && echo "preset" || echo "random" )
```

```
#echo "=> Creating MySQL admin user with ${_word} password"
```

```
echo "===>> Creating MySQL admin user with $PASS password"
```

```
mysql -uroot -e "CREATE USER 'admin'@'%' IDENTIFIED BY '$PASS'"
```

```
mysql -uroot -e "GRANT ALL PRIVILEGES ON *.* TO 'admin'@'%' WITH GRANT OPTION"
```

import_sql.sh

```
RET=1
```

```
while [[ RET -ne 0 ]]; do
```

```
echo "=> Waiting for MySQL to start"
```

```
sleep 5
```

```
mysql -uroot -e "status" > /dev/null 2>&1
```

```
RET=$?
```

```
done
```

```
echo "=> MySQL Started"
```

```
echo "=> Importing SQL file"
```

```
mysql -uroot --force --verbose < /app/milestone4dump.sql
```

```
echo "=> All done"
```

run.sh

```
VOLUME_HOME="/var/lib/mysql"

sed -ri -e "s/^upload_max_filesize.*/upload_max_filesize = $
{PHP_UPLOAD_MAX_FILESIZE}/" \
-e "s/^post_max_size.*/post_max_size = ${PHP_POST_MAX_SIZE}/"
/etc/php5/apache2/php.ini

if [[ ! -d $VOLUME_HOME/mysql ]]; then

echo "=> An empty or uninitialized MySQL volume is detected in
$VOLUME_HOME"

echo "=> Installing MySQL ..."

mysql_install_db > /dev/null 2>&1

echo "=> Done!"

/create_mysql_admin_user.sh

else

echo "=> Using an existing volume of MySQL"

fi
```

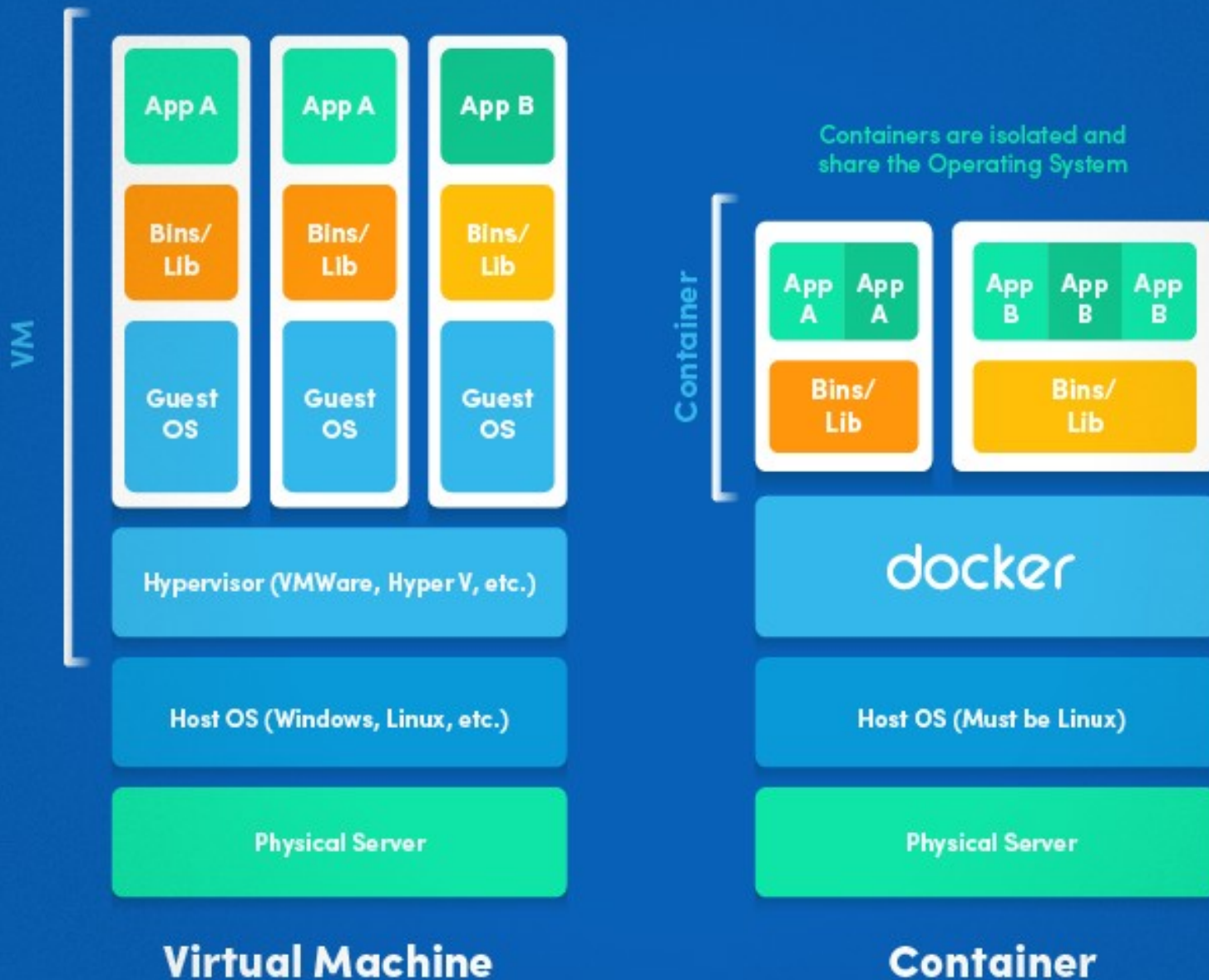
run.sh (cont)

```
mysqlFile="/app/milestone4dump.sql"
if [ -f $mysqlFile ]; then
echo "Importing MySQL data from $mysqlFile"
/import_sql.sh &
# mysql -u root -p < $mysqlFile
else
echo "I could not find $mysqlFile"
ls -l /app
fi
#ls -l /var/www/html/*
echo "justin got here"
exec supervisord -n
```

Dockerfile Commands

- `COMMAND` arguments
- `CMD`
 - Command to run when the container launches
- `docker ps`
 - See running docker containers
- `docker help`
 - usage

Containers vs. VMs



VMs vs Containers

- VMs:
 - Hypervisor
 - Close to bare metal
 - Own config, drivers, etc.
- Containers:
 - Share user space
 - Share the host OS

Potential Uses

- Dockerfile to:
 - Git clone
 - Run [installation commands]
 - Run [deployment commands]
 - CMD [automated testing script]
- Continuous development
- Rapid Testing

New Features that we won't use

- Improved isolation of containers:
 - Temporary file systems (--tempfs)
 - Better identifiers
- Swarms
 - Clusters of docker instances managed by a single machine
- Multi-container orchestration
 - Communication between multiple containers within