Breakthroughs in Firefly deployment with Docker and Kubernetes

Overview

- Docker has created given us a lot of flexibility in our deployment issues
- Docker was fairly easy to learn
- Docker is becoming our primary way to deploy firefly
- Kubernetes has enable to handle deployments more abstractly
- Kubernetes has help us detach deployments from a machine or VM

The Problems

3 Big Issues

- We needed a better testing environment.
  - How do I start a server running a new Firefly bug fix or feature?
  - i.e. Run 10 instances of Firefly from 10 different branches
- A better sharing environment
  - How do I download and start firefly running?
- A better way to do production management
  - I am having trouble keeping up with all the firefly servers
  - What is I want to go from two to three?
  - What if I want to have several small ones.
  - What if I want to experiment with cloud deployment.

Past Answers

- ~~Install Java (what version?)~~
- ~~Install Tomcat~~
- ~~Edit config files~~
- ~~Understand the Tomcat start/stop process~~
- ~~Don't start more then on Tomcat on a server~~

This is why
we need Docker

Docker in my own Words

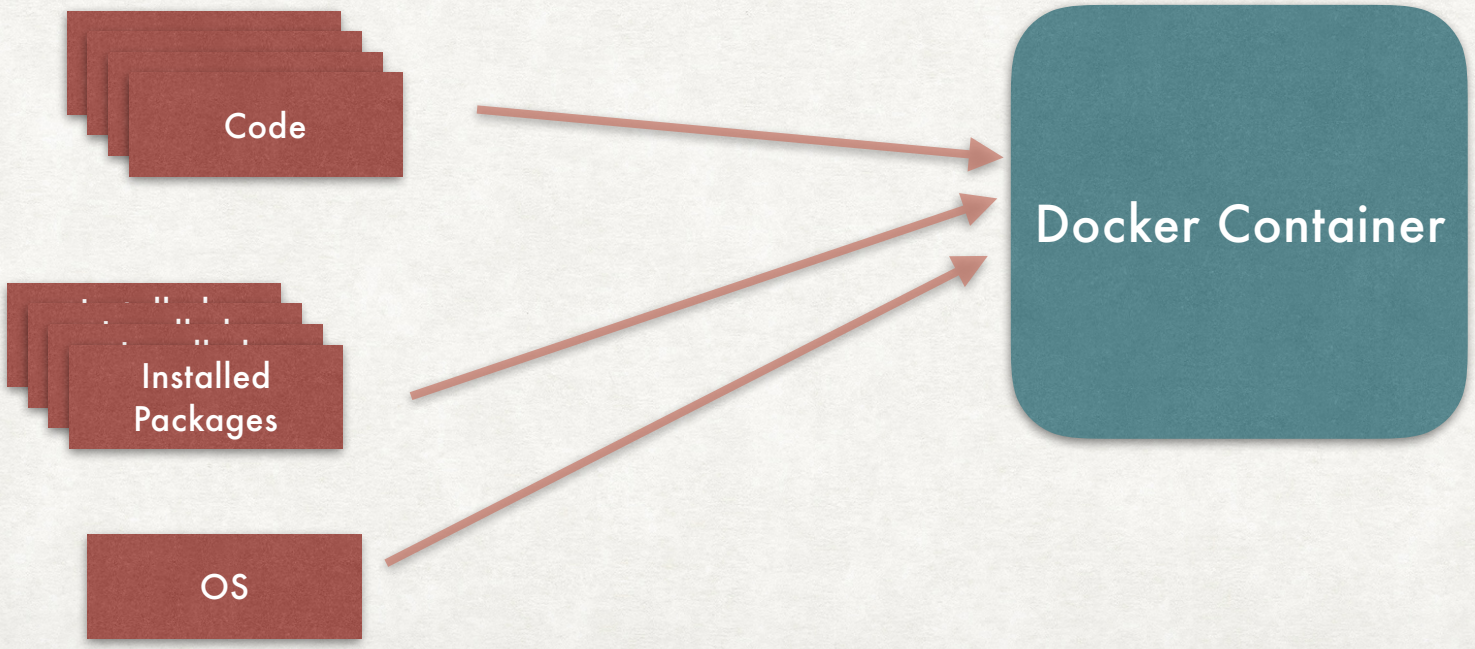
Docker is....

- A way to package and run applications
- Used to distribute an application
- Isolates the application from the OS
- Allows an application to run on various operating systems
- Not a Virtual Machine, but it might feel like one at first
- What makes it awesome: Just one command away…

Package and Run an Application

Or What is a Container?

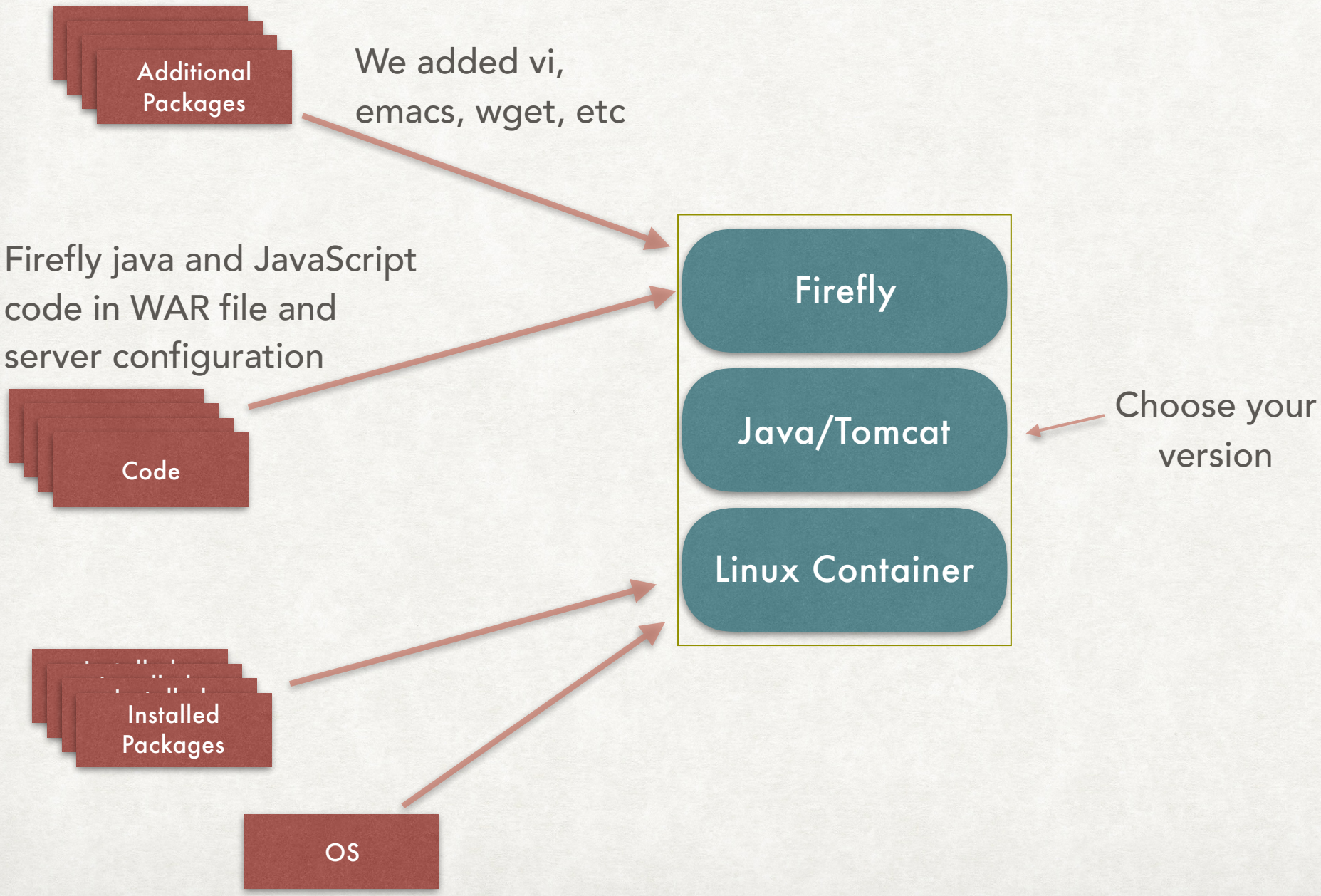
- A Base environment
- All your code
- All the libraries, binaries, scripts, or dependencies



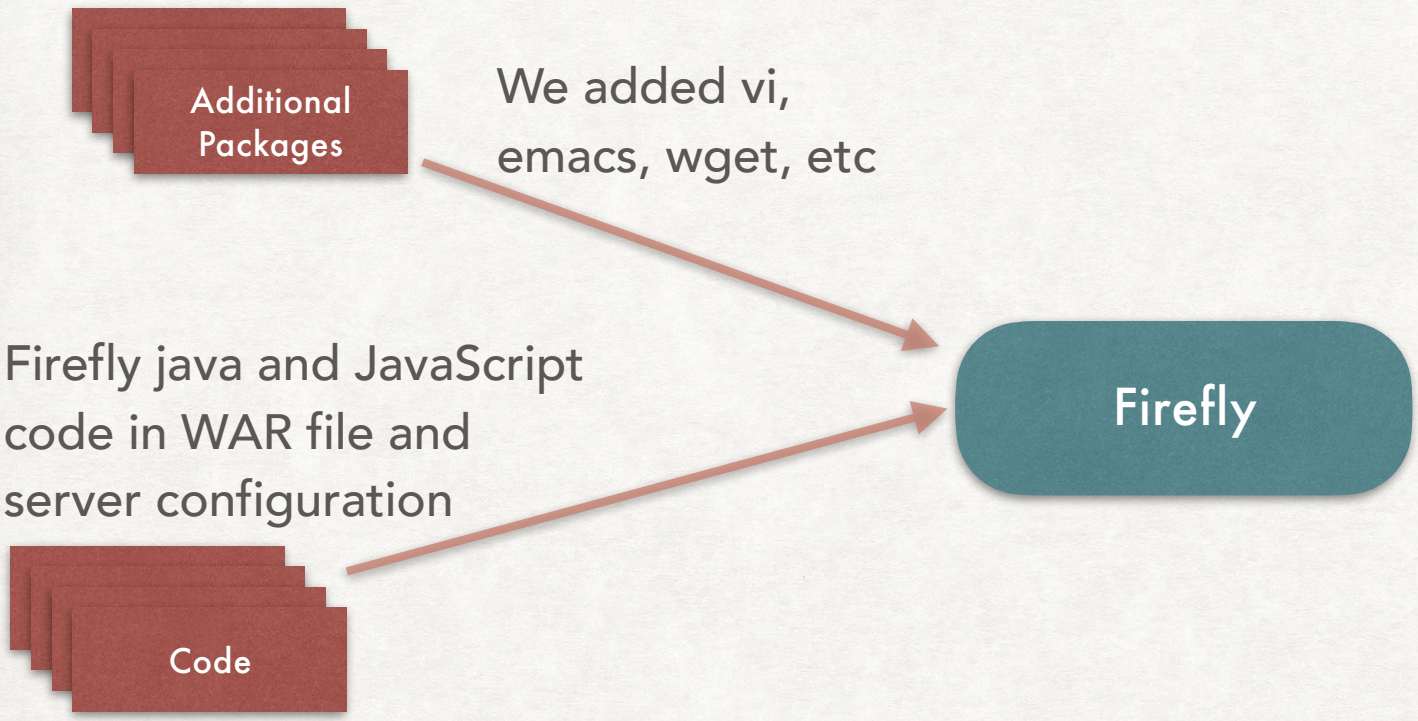
Experimenting with Docker

- Never start from scratch
- Docker is made to layer on (layer on layer)
- Library may put out containers that you can build on

Firefly Container



Firefly Container



Dockerfile defined what goes into the container

Firefly Docker File

The whole thing

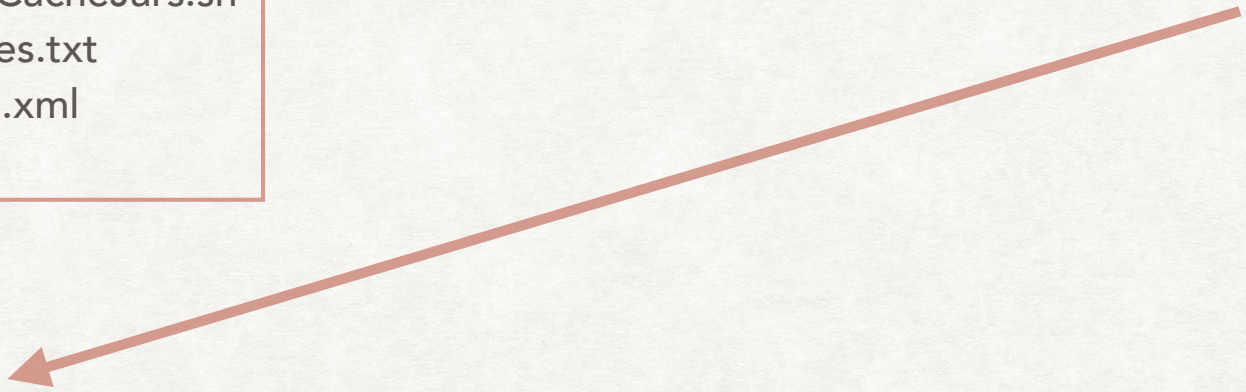
130 Lines

```
#FROM tomcat:7.0-jre8
FROM tomcat:9.0-jdk11-openjdk

#-----------------------------------------------------------------------
# To build: docker build -t ipac/firefly --build-arg IMAGE_NAME=ipac/firefly .
# For help in running: docker run --rm  ipac/firefly --help
#-----------------------------------------------------------------------

# Support single server deployments
# For multi server we need to look at multicast issues so ehcache can communicate


# add packages: vim, etc
# add any other standard apt packages here
RUN apt-get update && apt-get install -y \
        vim procps wget emacs-nox \
       && rm -rf /var/lib/apt/lists/*

# create catalina_base directory .. so tomcat can run as non-root
ENV CATALINA_HOME=/usr/local/tomcat
ENV CATALINA_BASE=/usr/local/tomcat-base
WORKDIR ${CATALINA_BASE}
RUN chmod g-s ${CATALINA_BASE} && \
    mkdir bin conf lib logs temp webapps work && \
    cp ${CATALINA_HOME}/conf/* ${CATALINA_BASE}/conf/ && \
    chmod +rw ${CATALINA_BASE}/conf/* && \
    chmod -R +rX ${CATALINA_HOME}


# These environment varibles are not really made to be overridden
# they can be but are mostly for setup
ENV JPDA_ADDRESS=5050
ENV CATALINA_PID=${CATALINA_BASE}/bin/catalina.pid

# work dir and config dir might be overridden if they were used in a mounted volume
# in the case make sure the directories exist
ENV SERVER_CONFIG_DIR=${CATALINA_BASE}/firefly-config
ENV FIREFLY_WORK_DIR=${CATALINA_BASE}/firefly-work
ENV FIREFLY_SHARED_WORK_DIR=''
ENV EXTERNAL_MOUNT_POINT=/external
ENV VISUALIZE_FITS_SEARCH_PATH=${EXTERNAL_MOUNT_POINT}

# container has access to the image name, used for help only
ARG IMAGE_NAME=''
ENV BUILD_TIME_NAME=${IMAGE_NAME}

# These are the file there are executed at startup, they start tomcat
COPY launchTomcat.sh \
     cleanup.sh \
     start-examples.txt \
     setupSharedCacheJars.sh ${CATALINA_BASE}/

# Tomcat config files, tomcat-users is for the admin username and password
# context.xml set delegate to true for we can use the classpath of tomcat
COPY tomcat-users.xml \
     context.xml  ${CATALINA_BASE}/conf/


# Make directories, make scripts executable, save old tomcat config files, remove unwanted apps
RUN chmod +x ${CATALINA_BASE}/launchTomcat.sh ${CATALINA_BASE}/cleanup.sh ${CATALINA_BASE}/setupSharedCacheJars.sh; \
    mkdir -p ${SERVER_CONFIG_DIR}; \
    mkdir -p ${FIREFLY_WORK_DIR}; \
    mkdir -p ${EXTERNAL_MOUNT_POINT}; \
    chmod 777 bin conf lib logs temp webapps work ${SERVER_CONFIG_DIR} ${FIREFLY_WORK_DIR}; \
    mv ${CATALINA_BASE}/conf/tomcat-users.xml ${CATALINA_BASE}/conf/tomcat-users.xml.in


# setenv.sh is used to defined CATALINA_OPTS and JAVA_OPTS
COPY setenv.sh ${CATALINA_BASE}/bin/

# increase max header size to avoid failing on large auth token
RUN sed -i 's/Connector port="8080"/Connector maxHttpHeaderSize="24576" port="8080"/g' ${CATALINA_BASE}/conf/server.xml

# 8080 - http
# 5050 - debug
EXPOSE 8080 5050


# ------------------------------------------------------
# ------------------------------------------------------
# Overide the following from the command line:
#          MIN_JVM_SIZE, MAX_JVM_SIZE,
#          INIT_RAM_PERCENT, MAX_RAM_PERCENT,
#          ADMIN_USER, ADMIN_PASSWORD,
#          DEBUG, jvmRoute, LOG_FILE_TO_CONSOLE, FIREFLY_OPTS,
# ------------------------------------------------------
# ------------------------------------------------------

# MIN_JVM_SIZE and MAX_JVM_SIZE can be used to set the min and max JVM side
# If MAX_JVM_SIZE is not set, the memory is autosized to the memory available to the container.
# Set the available memory on the command line with --memory="4g"
# You can change MAX_RAM_PERCENT on the command line with -e "MAX_RAM_PERCENT=80"
ENV MIN_JVM_SIZE=1G
ENV MAX_JVM_SIZE=
ENV INIT_RAM_PERCENT=10
ENV MAX_RAM_PERCENT=90
ENV JVM_CORES=0


#User name and password to use admin
ENV ADMIN_USER=admin
ENV ADMIN_PASSWORD=replaceMe
ENV DEBUG=false
ENV MANAGER=true

# if jvmRoute is not passed the hostname (the container id) is used
# such as: -e jvmRoute="myroute1"
ENV jvmRoute=''

# file to log to console, such as -e "LOG_FILE_TO_CONSOLE=firefly.log"
ENV LOG_FILE_TO_CONSOLE=''

# FIREFLY_OPTS could be used to pass any properties, setenv.sh picks it up
ENV FIREFLY_OPTS=''

# SHARE_CACHE set to TRUE when deploying multiple apps to share the VIS_SHARED_MEM cache
ENV SHARE_CACHE=FALSE


#copy all wars, typically there should only be one
COPY *.war ${CATALINA_BASE}/webapps/

RUN groupadd -g 91 tomcat && \
    useradd -r -u 91 -g tomcat tomcat

USER tomcat:tomcat
```

Docker Directory

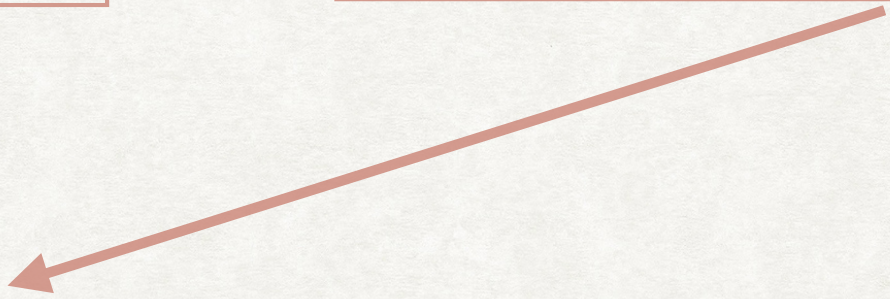
Dockerfile
cleanup.sh
context.xml
launchTomcat.sh
setenv.sh
setupSharedCacheJars.sh
start-examples.txt
tomcat-users.xml
war file

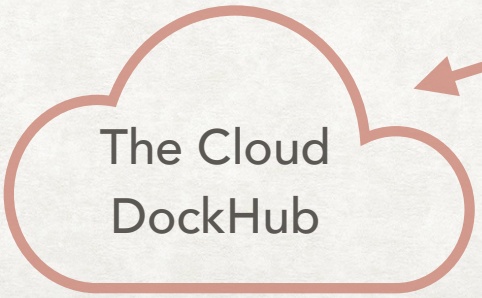
docker build -t ipac/firefly:mytag .



firefly container, name: ipac/firefly:mytag

docker push ipac/firefly:mytag



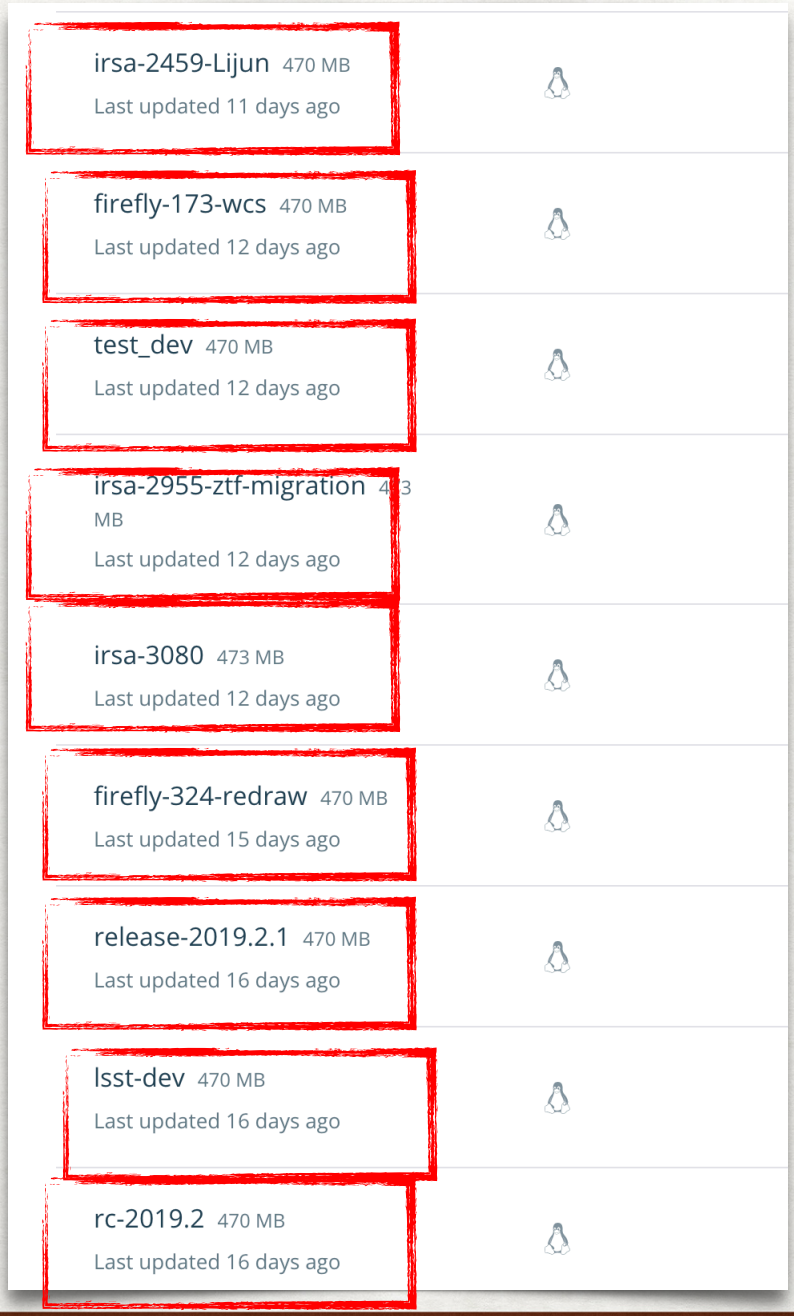


Docker Hub

Firefly Tags

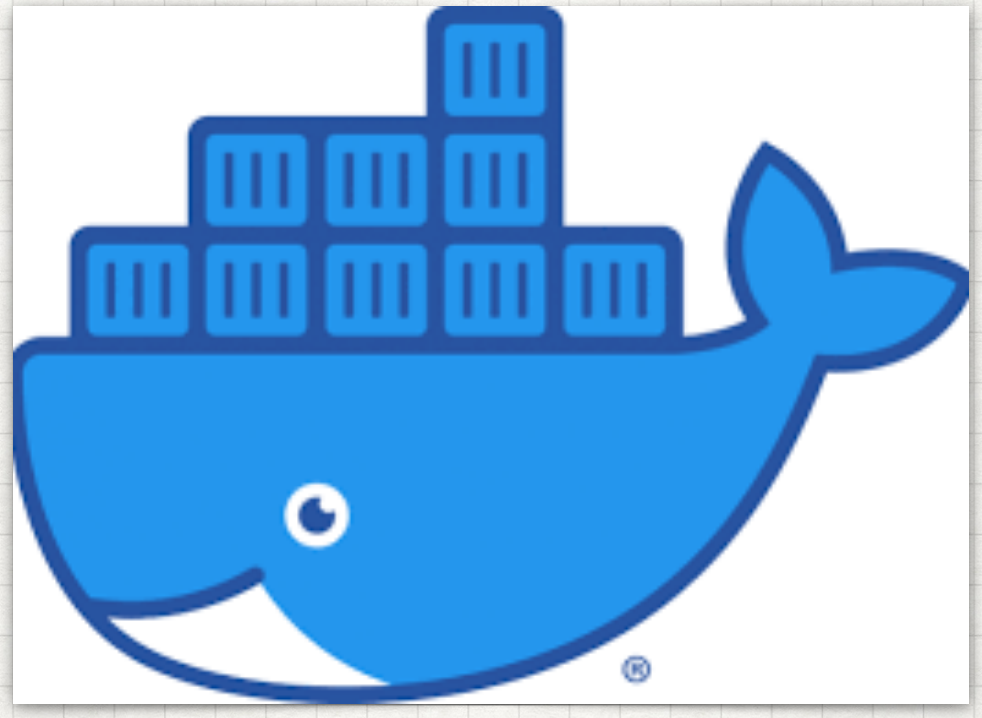
After the docker push command…

Docker Screen shot of Firefly builds



Using Docker

The Solutions



Problem solved: I need a sharing environment

Firefly docker contains live on DockerHub and are available to others

Run a release:

docker run --rm -p 8090:8080 -m 4G --name firefly ipac/firefly:release-2019.1

OR

Run a new feature or bug fix ticket:

docker run --rm -p 8090:8080 -m 4G --name firefly ipac/firefly:firefly-125

2 Problems left

- We needed a better testing environment.
- ✓ A better sharing environment
- A better way to do production management

Now we need Kubernetes!

Docker Plus Kubernetes

What is Kubernetes

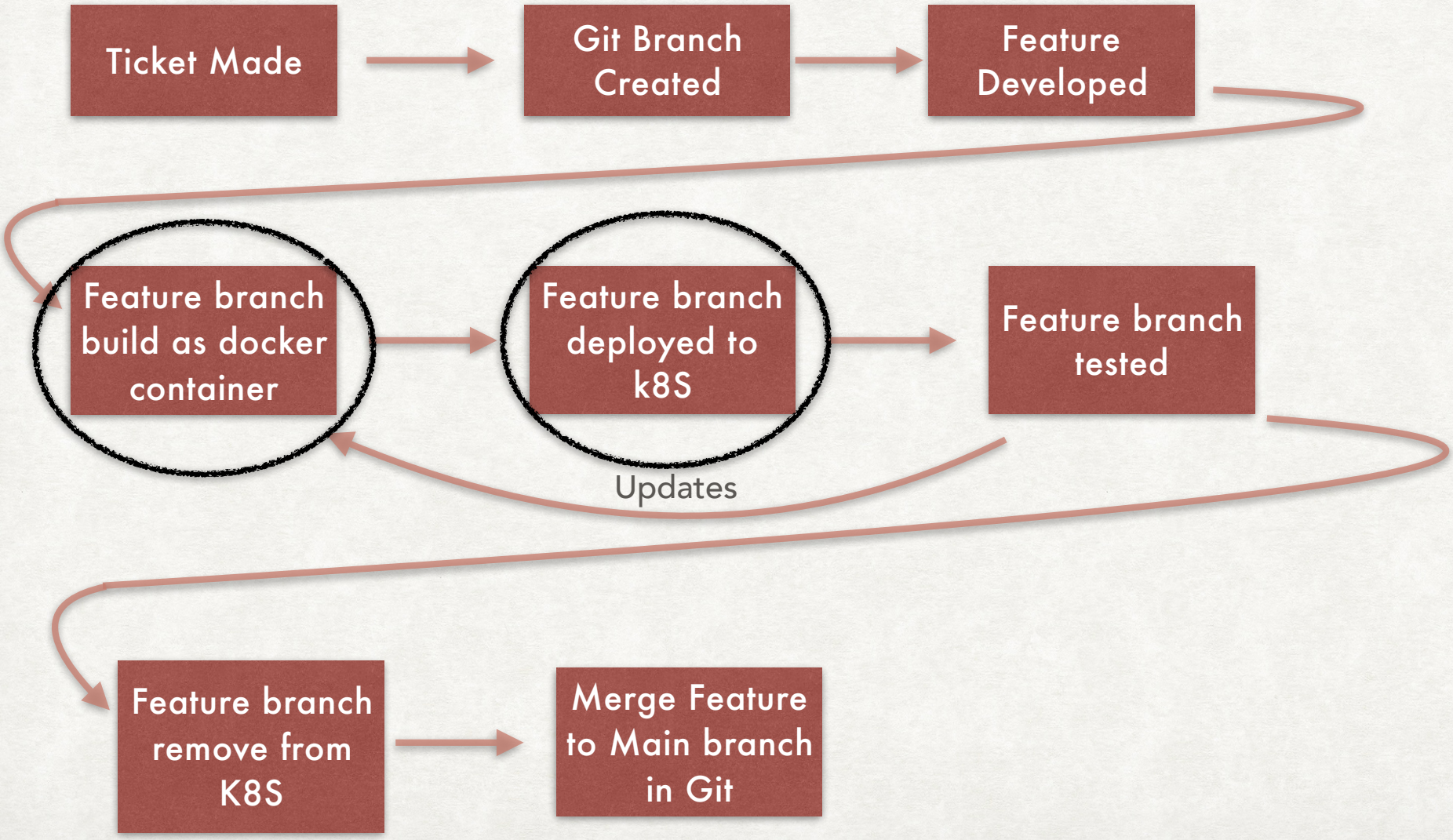
- **Kubernetes-** An open-source container-orchestration system for automating application deployment, scaling, and management.

Kubernetes in my own Words

- A way to deploy one or many Docker containers without worrying about the machine that it is running on
- A way to manage a group of machines as set of resources
  - ie. I have 96 cores and 256 GB of RAM
- A way to define what a deployment looks apart from the machines or VMs
- A deployment manager for a group of machines
- Can't spell it? **K8S**

It is the thing that gets the Docker container on a server with URL, port, memory, cores

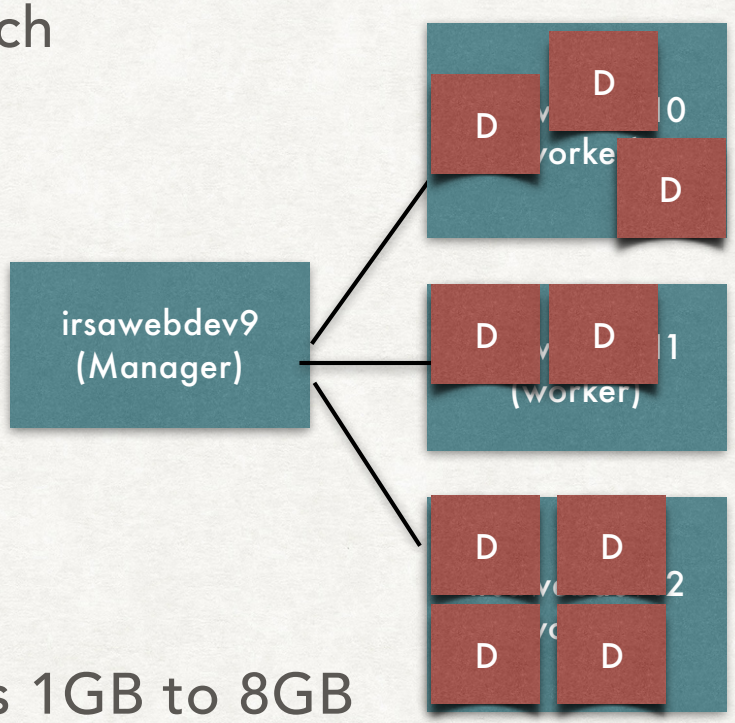
Development Flow



Firefly/IRSA Kubernetes Testing Setup

We needed a better testing environment.

- 1 manager machine irsawebdev9
- 3 worker machine 32 GB, 4 Cores each
- Allows for many testing servers



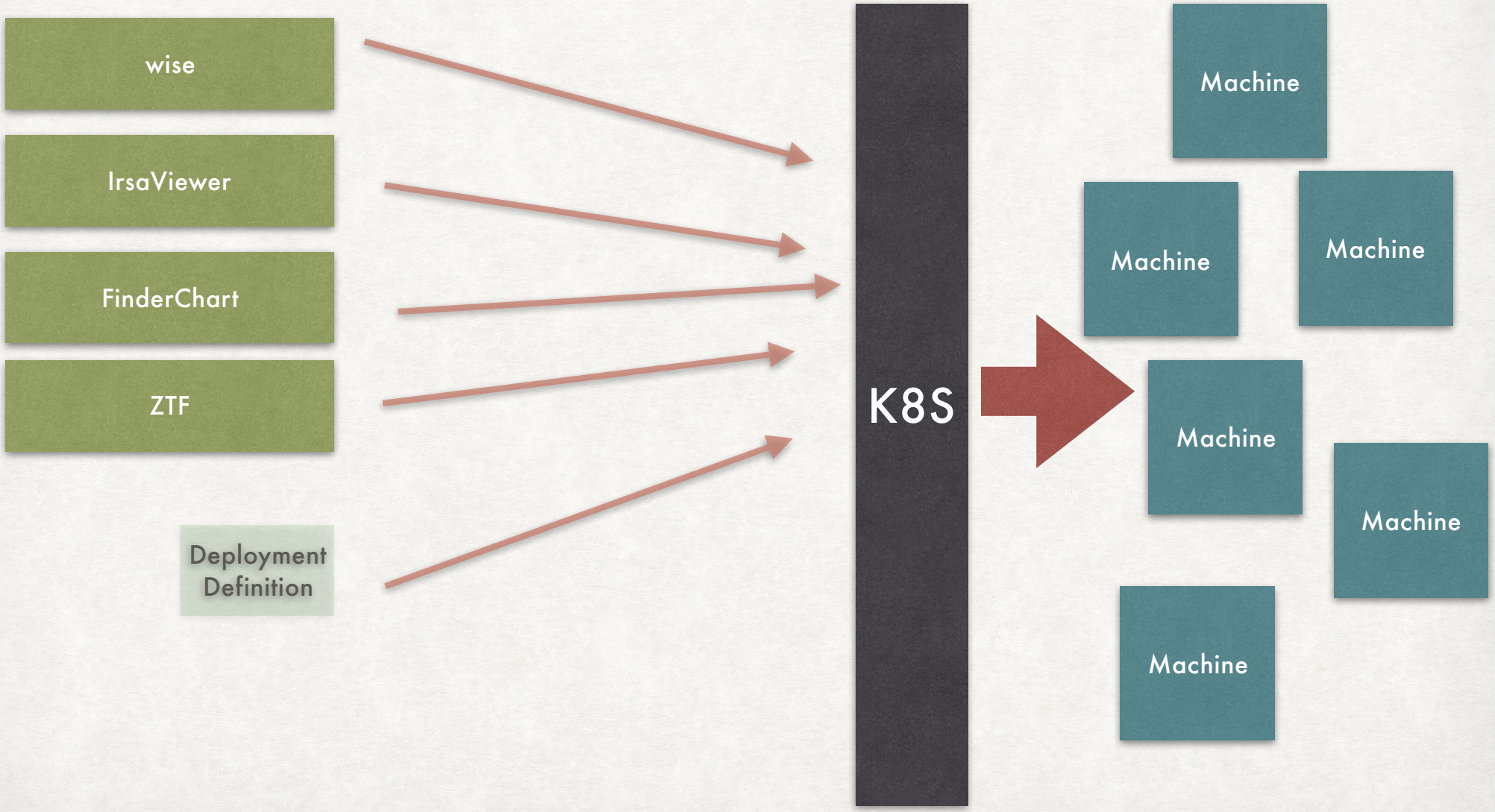
- Each Firefly docker deployment gets 1GB to 8GB
- Unique URL, for Example-
  - https://irsawebdev9.ipac.caltech.edu/irsa-1391/firefly/
  - https://irsawebdev9.ipac.caltech.edu/irsa-1234/firefly/

1 Problems left

- ✓ We needed a better testing environment.
- ✓ A better sharing environment
- ○ A better way to do production management

K8S Manages a deployment

A better way to do production management



All Problems Solved

- ✓ We needed a better testing environment.
- ✓ A better sharing environment
- ✓A better way to do production management

Summary

Wow this stuff is really Cool!

Questions? Who to Ask

- Docker Questions? Trey, Loi, Tatiana, David S.
- Kubernetes Questions? Loi, Tatiana, David S.