Utilizing Containers to Simplify your Oracle DevOps

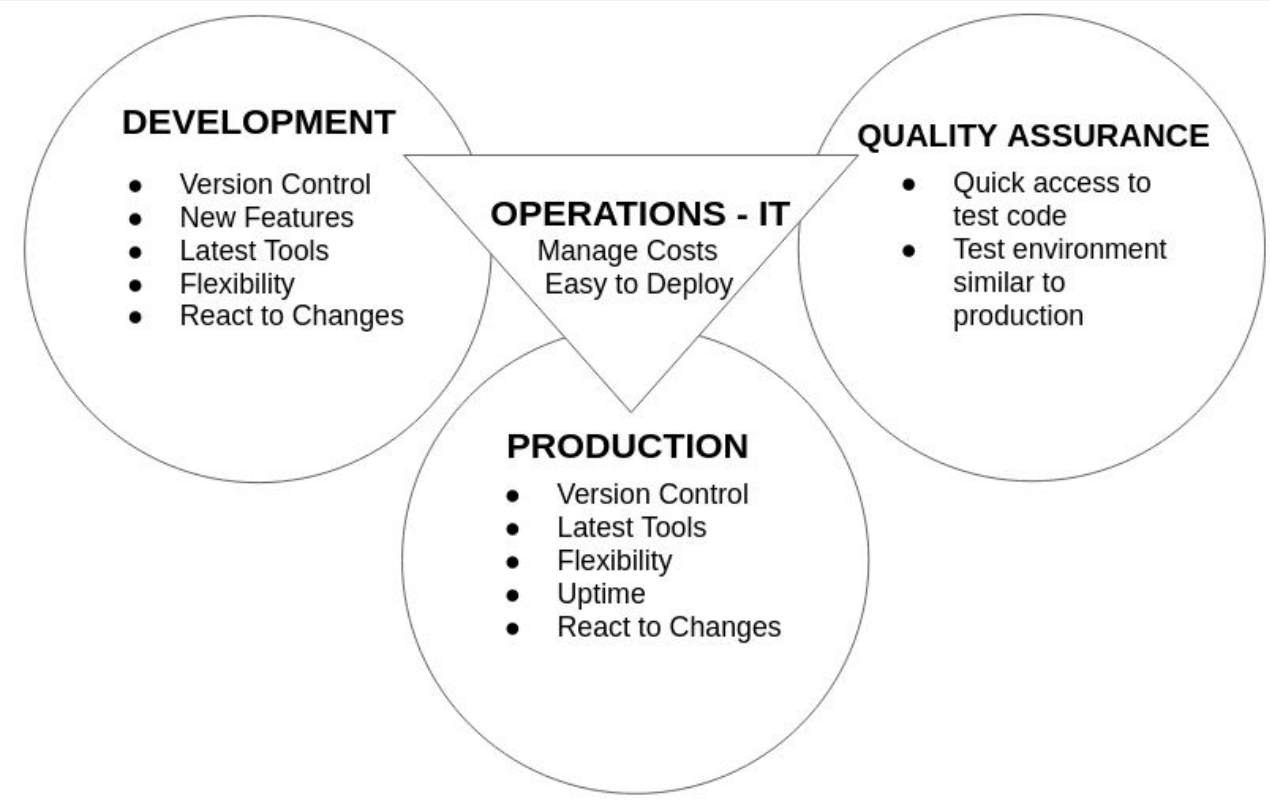
Roger Lopez
Sanjay Rao
Oct 1st, 2017

Agenda

- Underlying Problem
- Understanding DevOps
- Containers
- Using containers to simplify DevOps

The Underlying Problem

Typical Ecosystem

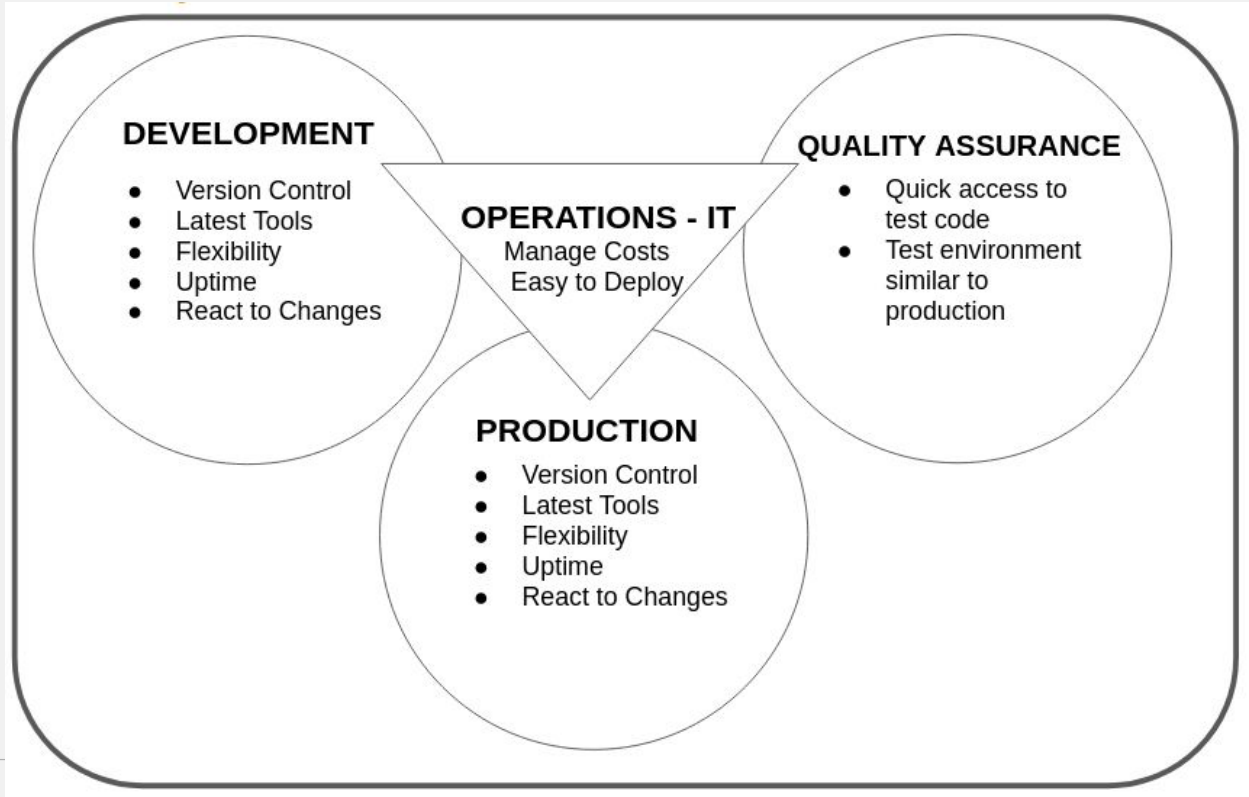


Understanding DevOps

DevOps

- It is a concept, practice
- DevOps Movement - CAMS
- Better collaboration between Developers and Operations (IT) to facilitate
- Software version management
- Easy transition between the various stage of Software Lifecycle
- Software delivery and Infrastructure changes
- Delegate control down to each Business Unit

DevOps Environment



Containers

Linux Containers

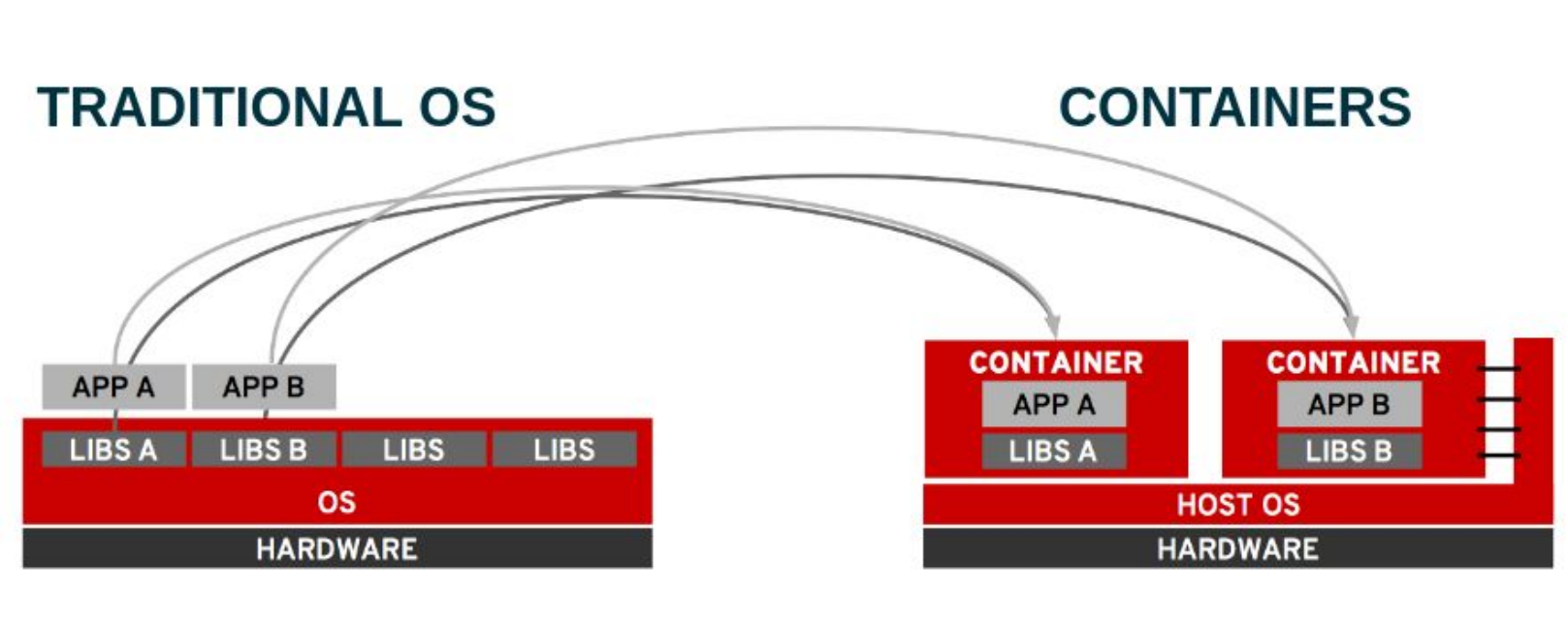
- What are Linux Containers (LXC)?
  - An OS level virtualization method that allows running multiple isolated environments for applications (containers) on the same host.
- How is this accomplished?
  - By using the Linux kernel’s Cgroups and isolated namespaces

Linux Containers

Benefits:

- Complete isolation from other processes, applications and containers running on the same host
- Resource friendly
- Platform independence
- Portability
- Resource control

Containers are an OS Technology



Containers vs Virtual Machines

- User space libraries
- More Density
- Performance on par with platform
- No live Migration

- Entire OS
- Some performance loss due to abstraction of certain functions
- Live Migration

Considerations when moving Legacy Applications to Containers

- Look for ways to split up your legacy application
- Understand the different application configurations
- Logging

Moving Legacy Applications to Containers

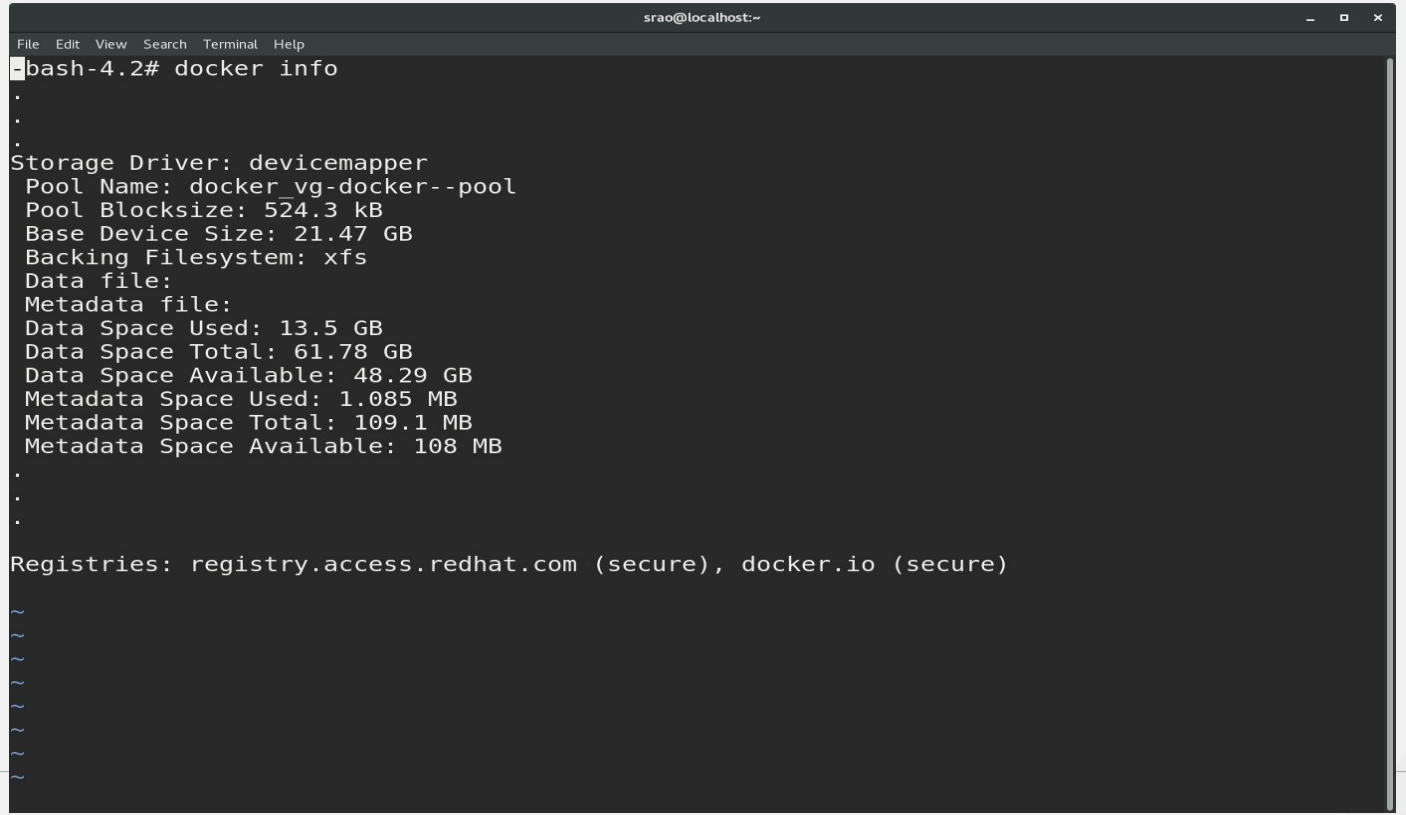
- Provide all the commands required to setup and install all the application dependencies within a Dockerfile
- Build an image based upon that Dockerfile, i.e. “gold image”
- Start your container
- Run your application as you would on bare metal environment

Building a Container Image

Building a Container Image

- Ensure that the image is validated or comes from a trusted source
- Get the required OS runtime image for the application
- Install all the runtime libraries and software that is required to be part of the container image

Trusted Registries



Build the Container Image - Dockerfile

- The Dockerfile is a text document that contains all the commands a user would run to build their application.
- An example Dockerfile to build an Apache server can be found here: http://bit.ly/2naWwez
- Let’s take a closer look!

Build the Container Image - Dockerfile

- An example of a more complex Dockerfile to build a Container can be found here: http://bit.ly/2nWxe3C
- Let's take a closer look!

Build the Container Image

- Once the Dockerfile is ready, within an empty directory that only contains the Dockerfile, build an image from a Dockerfile
- Syntax:
  docker build -t <name>:<tag> .
- Example:
  docker build -t oast:12c .
- Confirm the image is ready using
  docker images

Building and Running a Container Image

- Output of docker images

```
-bash-4.2# docker images
REPOSITORY                                    TAG                 IMAGE ID            CREATED             SIZE
oast                                          12c                 d5efd9ca1e74        About a minute ago  5.929 GB
oracle/database                               11.2.0.2-xe         8c6da4b7d1bd        23 hours ago        1.122 GB
registry.access.redhat.com/rhel7/rhel-tools   latest              264d7d025911        8 days ago          1.488 GB
registry.access.redhat.com/rhel7              latest              41a4953dbf95        13 days ago         192.5 MB
docker.io/oraclelinux                         7-slim              f005b5220b05        2 weeks ago         114.3 MB
```

- Use the docker run command to a container:
  Example:

```
docker run -it --name oast -d -v /var/oast:/perf1 oast:12c
```

- Set SELinux context for external volume:

```
chcon -Rt svirt_sandbox_file_t /var/oast
```

Building and Running a Container Image

- Check that the container is running using:
  `docker ps`
- Attach to the docker container via:
  `docker attach oast`

PRO Tip: Detach from a running container using CTRL+P,CTRL+Q

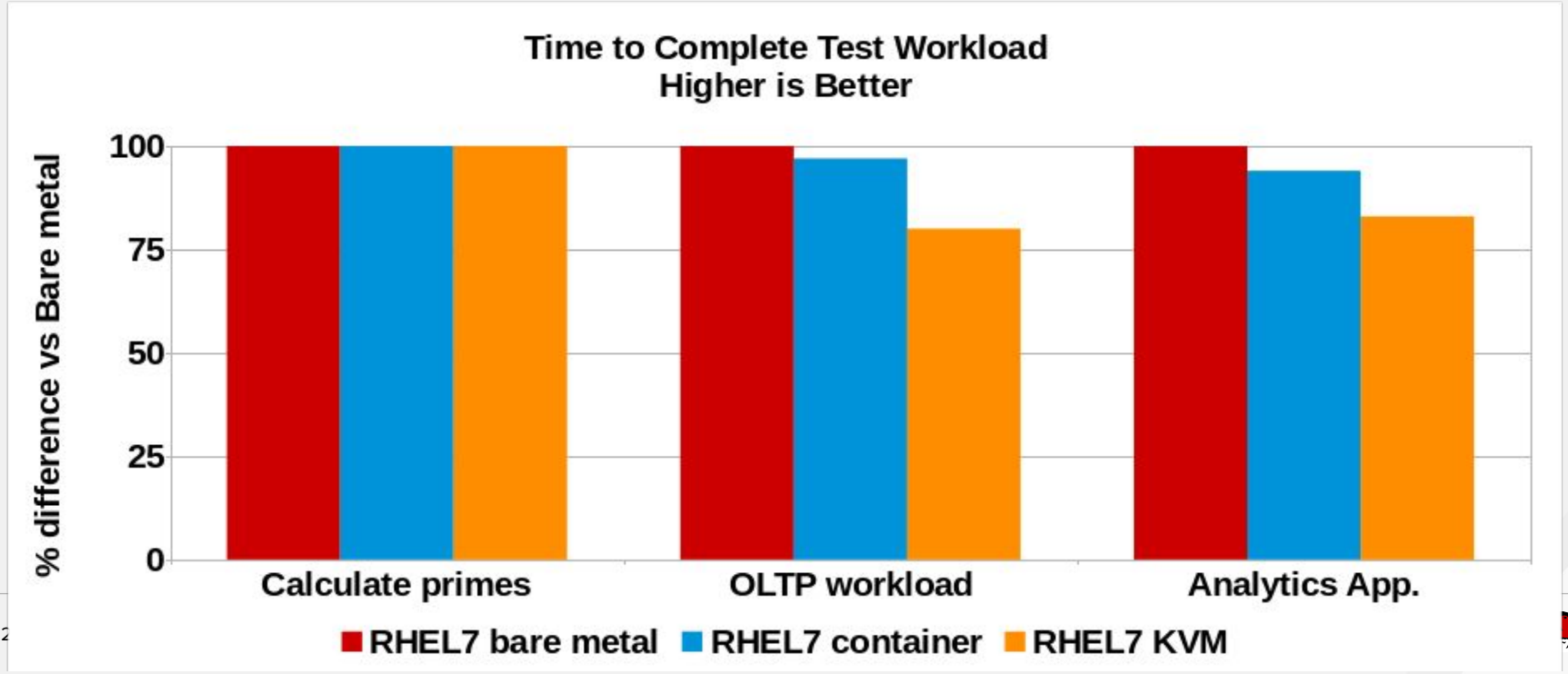
Modernizing Traditional Applications

Modernizing traditional applications

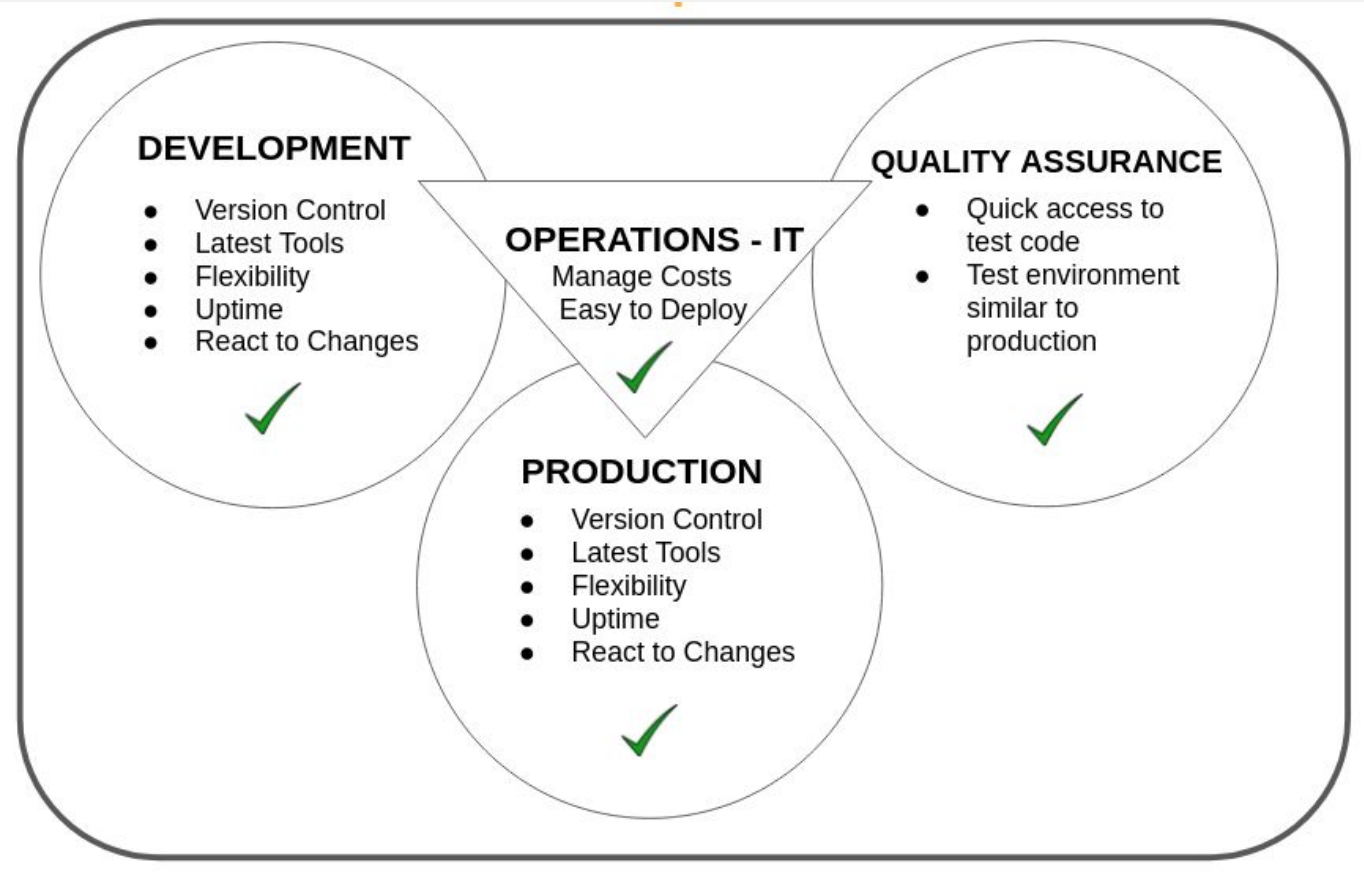
- Break down monolithic legacy applications into components
- Smaller footprint
- Better resource control
- Reduced downtime
- Scalable
- Faster Development / Testing / Deployment
- Performance

Performance of Large “Expensive” Apps

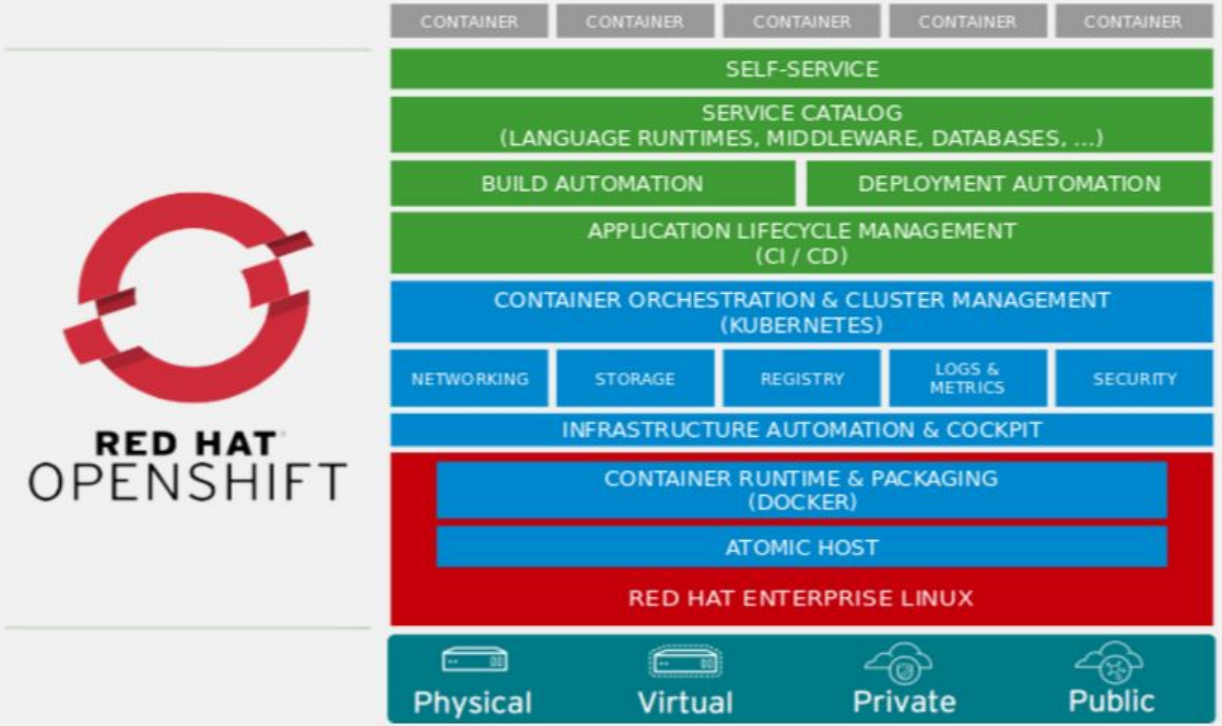
Container performance across multiple workloads



Containers for DevOps



Build, Deploy, and Manage Containerized Apps



Please visit the Red Hat Booth to see a demo on how to bring your enterprise database into the Openshift framework

Resources

Getting started with Containers (Red Hat Atomic Host)

http://red.ht/2kRHcBG

OpenShift Origin Latest Documentation

http://bit.ly/2mFiOrk

Using Oracle's pre-built Containers

https://github.com/oracle/docker-images/

Kubernetes

https://kubernetes.io/docs/

Reference Architectures

https://access.redhat.com/documentation/en/reference-architectures/

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos