Container Basics

Brian Lin
OSG Software
University of Wisconsin — Madison

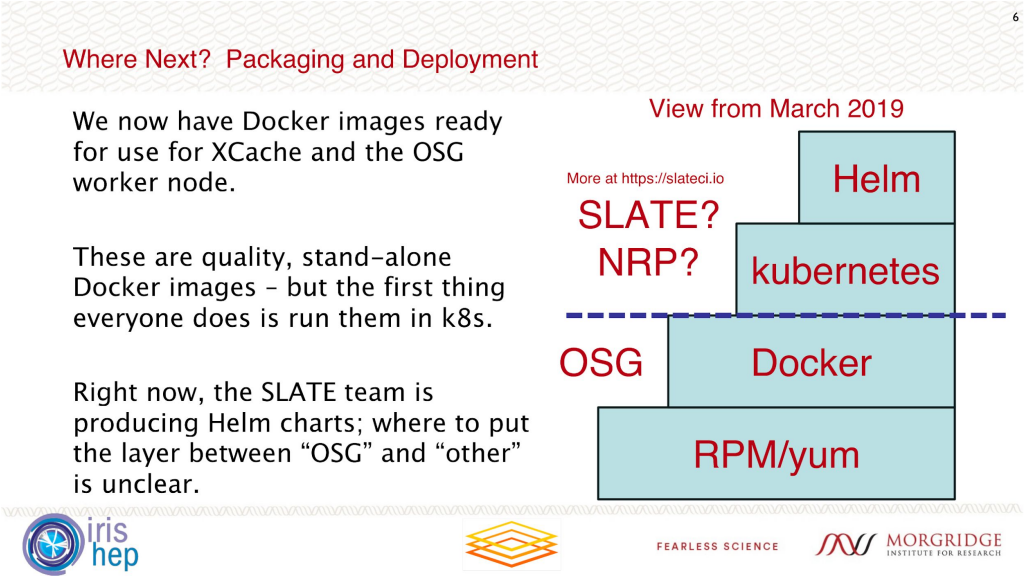
Containers

“Containers are an abstraction at the app layer that packages code and dependencies together. Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space.”

https://www.docker.com/resources/what-container

- Containers take advantage of cgroups and PID/network/mount namespaces.
- Docker vs Singularity
  - Docker runs as a service that keeps track of running containers; well-suited for running services
  - Singularity does not require a service; well-suited for running job payloads

OSG Software Support



See talk from Brian B:
https://indico.cern.ch/event/759388/contributions/3353317/attachments/1815443/2966874/OSG-AHM-2019.pdf

Docker Installation

- Docker is available via the docker repository
  https://docs.docker.com/install/linux/docker-ce/centos/#install-using-the-repository
- Install and start the docker service:
  ```
  # yum install docker-ce
  # systemctl enable --now docker
  ```
- Configure UID namespaces for better security!
  https://docs.docker.com/v17.12/engine/security/userns-remap/

Docker Basics

Run a container:

```
docker run imagename
```

Run a container with an interactive shell:

```
docker run -ti imagename /bin/sh
```

Remove a running container:

```
docker rm imagename
```

See running containers:

```
docker ps
```

See downloaded images

```
docker images
```

Remove a downloaded image

```
docker rmi imagename
```

The Future?

Kubernetes (k8s) Installation

- RESTful API server with YAML-based configuration files to instantiate various Kubernetes objects like Pods, Services, Deployments, Load Balancers, etc.
- Try it with MiniKube (VM-based, single-node Kubernetes cluster)
  - https://kubernetes.io/docs/tasks/tools/install-minikube/
- Install via kubeadm:
  - https://kubernetes.io/docs/setup/independent/install-kubeadm/
  - https://kubernetes.io/docs/setup/independent/create-cluster-kubeadm/
- Docker Compose is an alternative container orchestration tool
  https://docs.docker.com/compose/

Kubernetes Usage

Get nodes:

```
kubectl get nodes
```

List running pods:

```
kubectl get pods
```

Run a simple Nginx service:

```
kubectl create deployment --image nginx my-nginx-deployment
```

Delete deployment:

```
kubectl delete deployment my-nginx-deployment
```

Service Layer At The Edge (SLATE)

- SLATE allows the construction of lightweight federations of Kubernetes clusters with an eye toward security and local site policies
- Simple UNIX-like permissions model: Add users to groups, allow groups to access clusters.
- Application catalog simplifies service deployment while exposing configuration knobs for particular users, sites, etc.

Options for installing SLATE

- If you **don't** already have a Kubernetes cluster
  - SLATELite: https://github.com/slateci/slatelite (Docker-in-Docker based)
  - SLATE + K8S Installation script: http://jenkins.slateci.io/artifacts/scripts/install-slate.sh
- If you **do** already have a Kubernetes cluster
  - http://slateci.io/docs/quickstart/slate-client.html#installing-the-slate-client
  - Install SLATE Client: http://jenkins.slateci.io/artifacts/client/slate-linux.tar.gz
    - (RPMs also available)
  - Get a API token from the Portal (https://portal.slateci.io)

```
$ slate cluster create clustername --group defaultgroupname
--org "Default Org Name"
```

Installing a SLATE application

- Download configuration template
- Configure as necessary
- In this example the user would need to change External IP, Site, Cache directory, and location of the certificate

```
$ slate app get-conf xcache > xcache.conf

Instance: global

Service:
  Port: 1094
  ExternalIP: 192.170.227.151

SiteConfig:
  Name: MWT2
  AGISprotocolID: 433

XCacheConfig:
  CacheDirectories:
    - /scratch1
  HighWaterMark: 0.95
  LowWaterMark: 0.90
  RamSize: 16g
  BlockSize: 1M
  Prefetch: 0
  CertSecret: xcache-cert-secret
```

Expose this XRootD service on a cluster IP at port 1094

Secrets are encrypted & stored in DynamoDB

Installing a SLATE application (part 2)

- User fills out the configuration and hands it off to the 'app install' subcommand
- Specifies which cluster to install on, and under which group.
- Upon success, client returns instance ID to the user

```
slate app install xcache \
   --conf xcache.conf \
   --group atlas-xcache \
   --cluster uchicago-prod
```

Questions?