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

We now have Docker images ready for use for XCache and the OSG worker node.

These are quality, stand-alone Docker images - but the first thing everyone does is run them in k8s.

Right now, the SLATE team is producing Helm charts; where to put the layer between “OSG” and “other” is unclear.

See talk from Brian B:
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:

\texttt{docker run \texttt{imagename}}

Run a container with an interactive shell:

\texttt{docker run -ti \texttt{imagename} /bin/sh}

Remove a running container:

\texttt{docker rm \texttt{imagename}}

See running containers:

\texttt{docker ps}

See downloaded images

\texttt{docker images}

Remove a downloaded image

\texttt{docker rmi \texttt{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:

```bash
kubectl get nodes
```

List running pods:

```bash
kubectl get pods
```

Run a simple Nginx service:

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

Delete deployment:

```bash
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](https://github.com/slateci/slatelite) (Docker-in-Docker based)

● If you do already have a Kubernetes cluster
    ■ (RPMs also available)
  ○ Get a API token from the Portal ([https://portal.slateci.io](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

```bash
$ 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

● 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
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?