Virtualization/Containerization of the PNNL High Energy Physics Computing Infrastructure

Kevin Fox, David Cowley, Malachi Schram, Evan Felix, James Czebotar, Smith Gary
Grid Services Deployed

- **DIRAC**
  - Distributed Data Management System
  - Gatekeeper Services
  - Many development and testing services
- **Condor CE's**
  - DIRAC SiteDirector
  - HTCondor cluster
  - Squid Cache
- **Leadership Class Facility CE's**
  - DIRAC SiteDirector
  - HPC Cluster
- **SE's**
  - BestMan2
  - Gridftp
  - Backed by Lustre
- **Belle2DB**
  - REST Service
  - UI Service
  - Payload Service
  - Squid Cache
  - Postgresql Relational Database
- **FTS3**
- **CVMFS Stratum**
  - Zero
  - One
- **Authorization**
  - Gums
  - VOMS Server with multiple VO's
Note to the Sysadmins

- New methodology for system administration.
- Cloud Native focuses around what the user cares about most, not what we Sysadmins are used to caring about.
- Users care about services. Users do not care about machines providing service.
- Pets vs Cattle analogy.
- We must unlearn what we have learned.
- Try and separate pets and cattle to different pools of resource.
Our Infrastructure Journey

- Individual machines
- Automated provisioning
- Virtual machines
- OpenStack Cloud
- Repo Mirrors
- Containers
- Kubernetes
## Infrastructure Deployed

<table>
<thead>
<tr>
<th>Kubernetes + Docker Engine</th>
<th>Prometheus</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenStack + KVM</td>
<td>Grafana</td>
</tr>
<tr>
<td>Ceph</td>
<td>CheckMK</td>
</tr>
<tr>
<td>GitLab</td>
<td>ElasticSearch</td>
</tr>
<tr>
<td>Lustre</td>
<td>389-DS</td>
</tr>
<tr>
<td>LoadBalancing/HA</td>
<td>Cobbler</td>
</tr>
<tr>
<td>PerfSonar</td>
<td>NFS</td>
</tr>
</tbody>
</table>
Metric/Log gathering is very important for system problem analysis

Current tool stack includes

- CheckMK
- Grafana/Prometheus
- Kibana/ElasticSearch/LogShippers
- Kubernetes
Load Balancers

- Give users a load balancer to talk to.
- Back it with multiple instances of the software making up of the service whenever possible.
- When not possible, make it very quick to redeploy.
Separate Build and Deploy steps.

Kubernetes/Docker example:

#Build
> docker build . -t pnnlhep/condor-compute:2017-09-01
... 
> docker push pnnlhep/condor-compute:2017-09-01
...

#Deploy
> helm install --name ce0-compute condor-compute \ 
  --set version=2017-09-01
...
> helm upgrade ce0-compute condor-compute \ 
  --set version=2017-09-02
...
#Kubernetes object description

kind: Deployment
spec:
  replicas: 3
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
      MinReadySeconds: 60

#Kubernetes Commands:

> kubectl rollout pause deployment <deployment>
> kubectl rollout resume deployment <deployment>
> kubectl rollout undo deployment <deployment>

Canary Deployments
Software
Ceph - Software Defined Storage

- Fault tolerant, tiered, and replicated storage.
- Uses cheap nodes.
- Replication is over nodes.
- Performance is ok.
- Rock solid.
Kubernetes

- **Service oriented container orchestration by Google.**

- **Supports**
  - Container Scheduling
  - Checking & Healing
  - Load Balancing
  - Storage Provisioning
  - VM's and BareMetal
  - Autoscaling
  - Helm Package Manager
Logging

- Log Shipping
  - Fluent-Bit
  - Fluentd
  - Logstash

- ElasticSearch
  - Storage
  - Indexing
  - Query

- UI
  - Kibana
Looking to the future, we would like to share our Helm packages to deploy HEP services on top of Kubernetes as well as other stuff we've done.

Is HSF the right forum for this?

If not, if anyone interesting in contributing to such a project, please don't hesitate to contact me.
Questions?