OSiRIS Site Deployment
Leveraging puppet and foreman to build a distributed ceph cluster
What is OSiRIS?

OSiRIS combines a **multi-site Ceph cluster** with **SDN** and **AAA infrastructure** enabling scientific researchers to efficiently access data with federated institution credentials.

The current OSiRIS deployment spans Michigan State University, University of Michigan, and Wayne State University. Indiana University is also a part of OSiRIS working on SDN network management tools.
The OSiRIS project goal is enable scientists to collaborate on data easily and without building their own infrastructure.

We have a wide-range of science stakeholders who have data collaboration and data analysis challenges to address within, between and beyond our campuses.

High-energy physics, High-Resolution Ocean Modeling, Degenerative Diseases, Biostatics and Bioinformatics, Population Studies, Genomics, Statistical Genetics and Aquatic Bio-Geochemistry
Our first site required manual steps to bring up VM host, and Foreman/Puppet installation. The rest, including Ceph components, is automated from there.
How we deploy

"Seed" OSiRIS Site

- Foreman
- Provisioning Proxy - TFTP/DHCP
- Libvirt Plugin

Additional OSiRIS Site

- Provisioning Proxy - TFTP/DHCP
- VM
- Bare Metal
- Libvirt Plugin

OSiRIS - Supercomputing 2016
How we manage

Puppet Master

Puppet Environments (Git Branches)

References Puppetfile to pull external modules into deployed environments

R10K

Github

Branch for new work

Collaborators make pull requests to get work into production branch

OSiRIS Collaborator

Query Available Environments

Foreman is puppet ENC (External Node Classifier)

Which environment is this node in?

Compile and serve manifest for environment specified by Foreman

OSiRIS Node
How we organize

site = um, msu, etc  
role = stor, virt, omd, etc

Generally we don’t directly include classes - instead we include ‘profiles’ that include classes
Deploying a new site

Step 1: Define site specific information in site/sitename.yaml (hiera)

- Network information for provisioning (subnet info, dhcp ranges, etc)
- Ceph CRUSH location
- NTP, DNS, etc
Deploying a new site

yaml file matching site from site-role.osris.org hostname

Site specific info such as dhcp for provisioning, ns, default osd crush location
Deploying a new site

Step 2:
- Create a new host in Foreman for the site virtualization host
- Export bootable image
- Install virtualization host, puppet
  configures necessary packages/services
- Register compute resource in Foreman
After build we can define as a compute resource in Foreman

Define host network interface, build by exporting boot image from Foreman
Deploying a new site

Step 3:
- Download VM template for provisioning proxy
- Run VM, configure network

- run puppet to complete configuration and register with master Foreman instance
Deploying a new site

Smart Proxies

<table>
<thead>
<tr>
<th>Name</th>
<th>URL</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>msu-prov-be.osris.org</td>
<td><a href="https://msu-prov-be.osris.org:8443">https://msu-prov-be.osris.org:8443</a></td>
<td>Templates, TFTP, and DHCP</td>
</tr>
<tr>
<td>sc-prov.osris.org</td>
<td><a href="https://sc-prov.osris.org:8443">https://sc-prov.osris.org:8443</a></td>
<td>Templates, TFTP, and DHCP</td>
</tr>
<tr>
<td>um-puppet.osris.org</td>
<td><a href="https://um-puppet.osris.org:8443">https://um-puppet.osris.org:8443</a></td>
<td>TFTP, Puppet, Puppet CA, and DHCP</td>
</tr>
<tr>
<td>wsu-prov-be.osris.org</td>
<td><a href="https://wsu-prov-be.osris.org:8443">https://wsu-prov-be.osris.org:8443</a></td>
<td>Templates, TFTP, and DHCP</td>
</tr>
</tbody>
</table>

Puppet triggers provisioning host to register itself as a ‘smart proxy’ in foreman (auth info propagated in configuration)

Smart proxy can provide kickstart templates, tftp, dhcp to local network at site
Deploying a new site OSD

In hiera:
- Define the OSD devices used for storage block(s)
- Define the network interfaces to collect stats to Influx/Grafana (collectd-ethstat)
- Define OSD id to collect stats (collectd-ceph)
Deploying a new site OSD

Interfaces and collectd-ceph daemons in yaml matching hostname

Most of our storage nodes identical, define ceph osd devices at role level (for now)
Deploying a new site

From this point we’re ready to build new storage blocks, monitor, mds, grafana, omd, etc.

All of the above automated with puppet, and with Foreman groups defining appropriate partitions or data volumes
Dynamic and Scalable

While OSD are initializing and coming online we have a client data transfer ongoing. You can see the impact on the transfer and the progress of the OSD addition on our monitoring dashboard.
OSD Count climbing as puppet agent uses ceph-disk to init new

Cluster moving data replicas to new OSD
Ongoing during our talk is a demo of live data movement leveraging the Data Logistics Toolkit created at Indiana University.

This demo showcases the movement of USGS earthsat data from capture to storage not only in the main OSiRIS Ceph cluster but also in a dynamic OSiRIS Ceph cluster deployment built at Cloudlab.

Activity can be seen on the Periscope dashboard
http://dev.crest.iu.edu/map/
Questions or comments?