Central DevOps with SLATE & PRP

Rob Gardner
University of Chicago

H.O.W. 2019
March 21, 2019

Goal

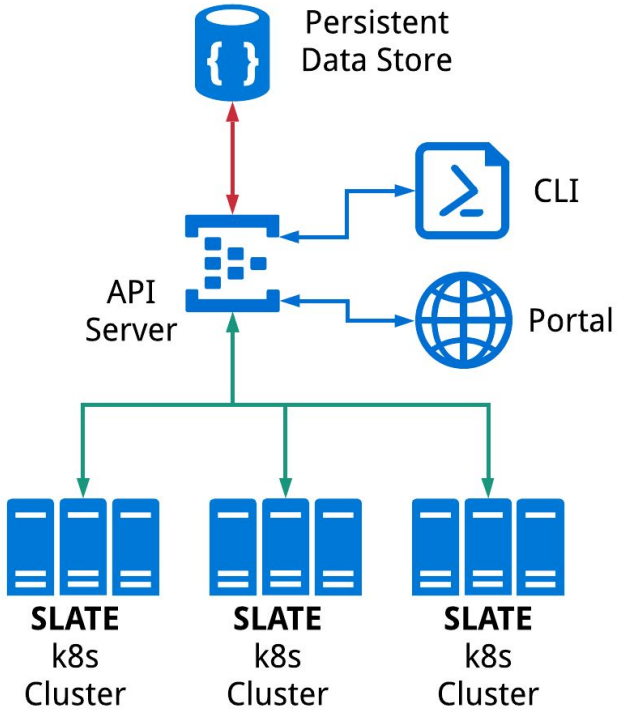
- Remotely manage edge services at sites (e.g. squid, xcache, ...) with central expert teams
- Deploy updates more quickly
- Introduce new services more easily
- Save time and effort for the local admins
- SLATE and PRP (Pacific Research Platform) are two activities with cache deployment as early use case

Create a federation of edge clusters

- **SLATE: S**ervices **L**ayer **A**t **T**he **E**dge
- Distributed service orchestration platform
- Kubernetes-based
- Start with a single server and scale as needed
- Loosely federated, share projects/users/applications across institutions
- Good for any site but "**lightweight**" sites might find it particularly useful

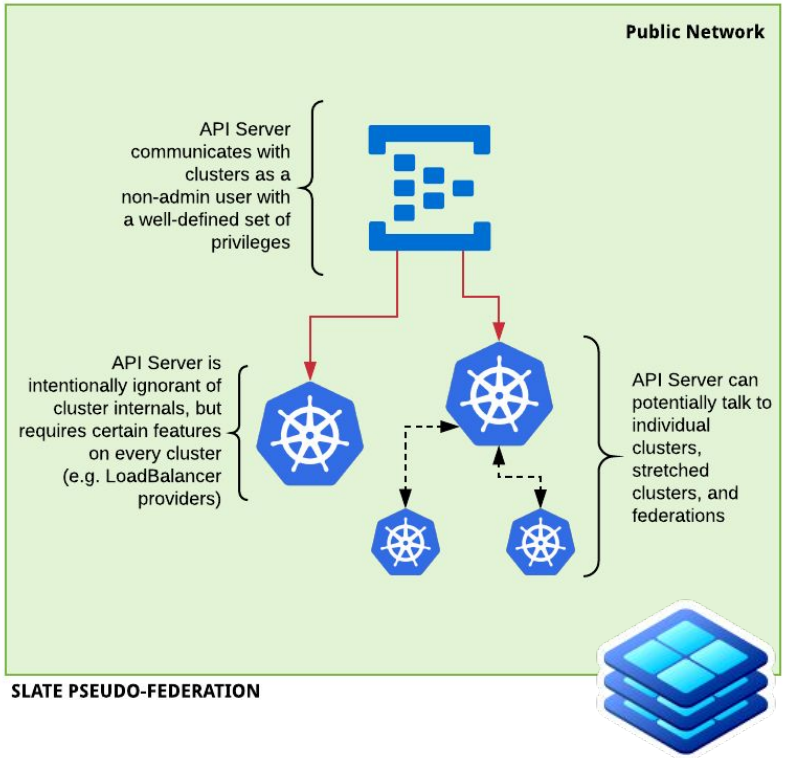
Basic SLATE Architecture

- Lightweight federation and application catalog layer on top of Kubernetes
  - Security-conscious, site autonomous
  - Sites retain administrative control
- Single entrypoint using institutional identity
- Simple UNIX-like permissions model (Users + Groups)
- **Application catalog** provides natural boundary between configuration knobs users actually want to change and complex Kubernetes configurations
- SLATE is an infrastructure **and software**

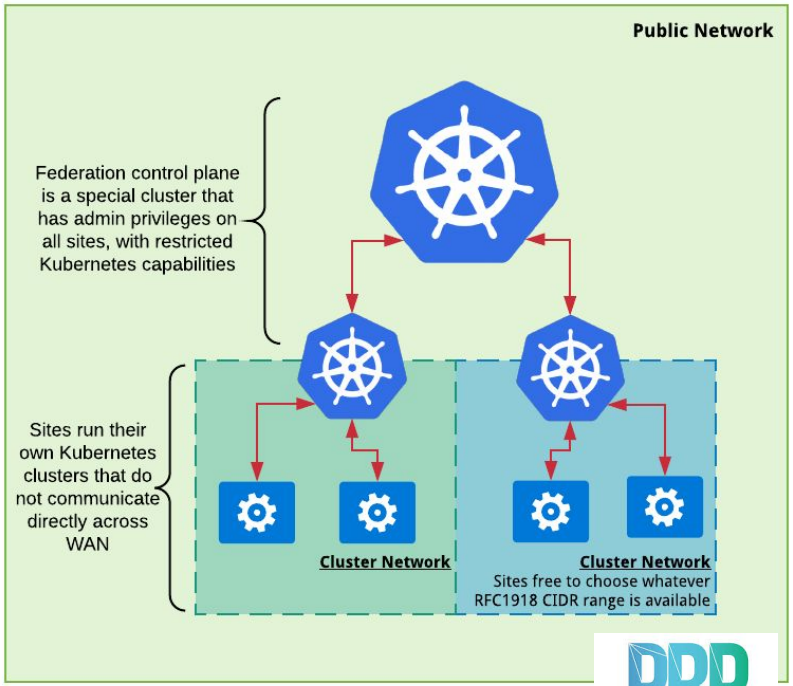


Create & manage your own federation over independently managed Kubernetes clusters

SLATE and PRP federation approaches



SLATE PSEUDO-FEDERATION

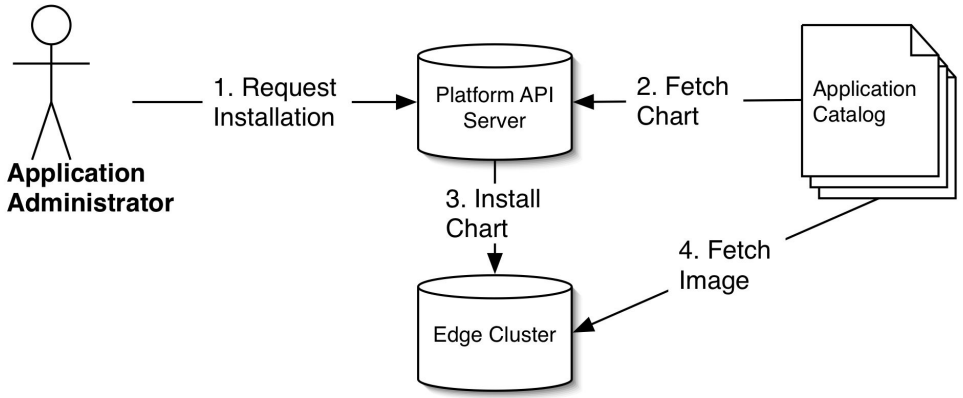


FEDERATION CONTROL PLANE

PACIFIC RESEARCH PLATFORM

Deploying Services ("Applications" in k8s)

- A "central" service expert deploys & operates many sites
- Helm charts and Docker images
- Command line or web interface (in dev)



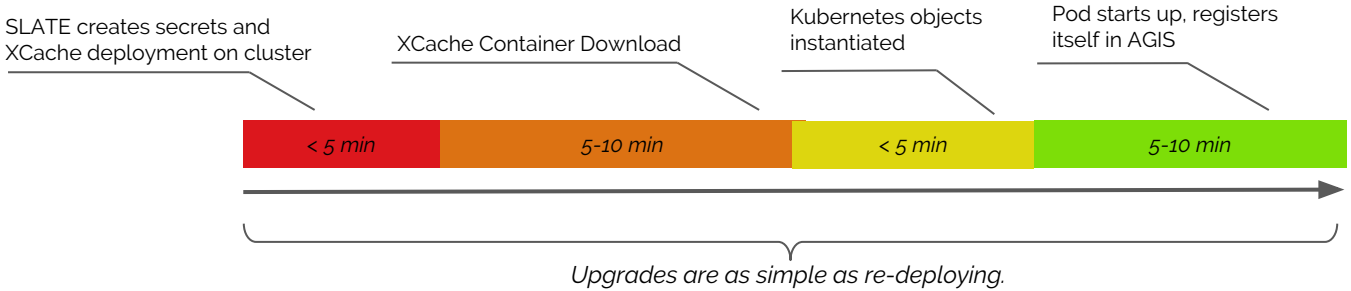
Deployment experience in ATLAS

- Goal: build an XCache-based caching network as part of the DOMA activity
- SLATE-based deployment will simplify operations and allow for rapid development and debugging
- SLATE services already operational at MWT2, AGLT2, LRZ
- XCache application already in SLATE catalog
  - Ilija is developing & testing daily

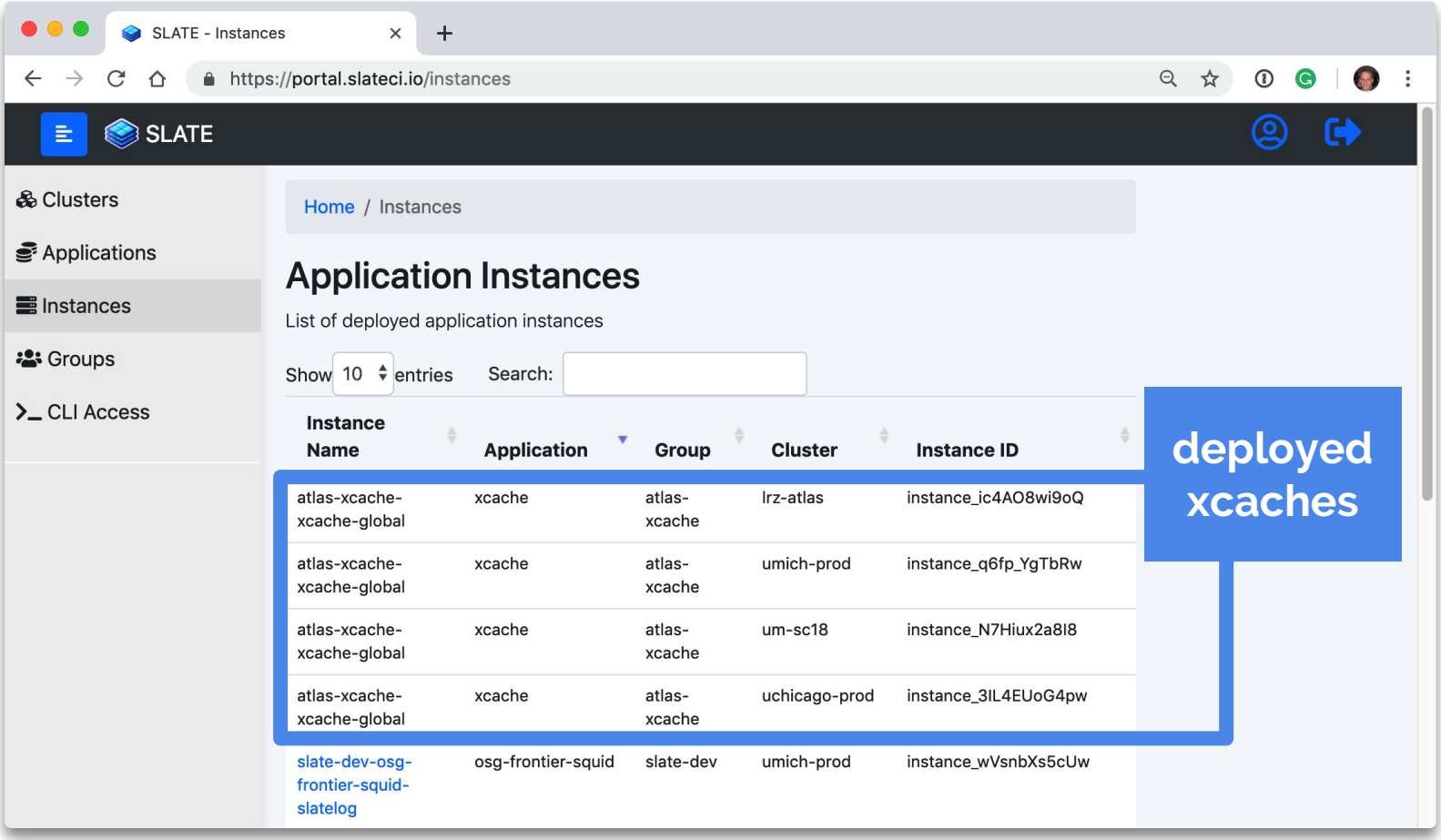
XCache deployment process

- Register a cluster with SLATE and allow the `atlas-xcache` group
- Apply a few special extra steps for XCache:
  - Node labeled in Kubernetes (`xcache-capable=true`)
  - One or more storage volumes mounted (e.g. `/xcache`) & communicated to Ilija
  - Endpoint protocol registered in AGIS
- Test suite containerized
  - Launch a very realistic stress test from Google Compute Engine in minutes

XCache Deployment & Upgrade Cycle:



XCache fully deployed in less than 20 minutes.


SLATE - Instances
https://portal.slateci.io/instances
SLATE
Clusters
Applications
Instances
Groups
CLI Access
Home / Instances
Application Instances
List of deployed application instances
Show 10 entries
Search:
Instance Name
Application
Group
Cluster
Instance ID
atlas-xcache-xcache-global
xcache
atlas-xcache
lrz-atlas
instance_ic4AO8wi9oQ
atlas-xcache-xcache-global
xcache
atlas-xcache
umich-prod
instance_q6fp_YgTbRw
atlas-xcache-xcache-global
xcache
atlas-xcache
um-sc18
instance_N7Hiux2a8I8
atlas-xcache-xcache-global
xcache
atlas-xcache
uchicago-prod
instance_3IL4EUoG4pw
slate-dev-osg-frontier-squid-slatelog
osg-frontier-squid
slate-dev
umich-prod
instance_wVsnbXs5cUw
deployed xcaches

XCache updates

- Even simpler
- Completely transparent to site admin.

```
$ slate instance list
$ slate instance delete <instance name>
$ slate app install --group atlas-xcache --cluster uchicago-prod --conf MWT2.yaml xcache
```

Additional benefits:

- Automatic core dump collection
- Containerized environment makes it easier to debug

StashCache deployment on PRP

- Leveraging an already existing PRP Kubernetes federation infrastructure maintained by SDSC.
- A shift in traditional grid deployment. The hardware and software responsibilities are split. **All software (including the cache) run on docker containers (k8 pods) and it is maintained centrally.** The local admins take care of hardware issues (Ex: disk).
- On every node there is a **perfsonar pod.** This helps deliver a quality of service since now network responsables (PRP) have full access.
- Stashcache Containers can be found at https://github.com/efajardo/prp-stashcache

Endpoint Caches

Network backbone Caches

In Collaboration with Internet 2. A pilot model on locating caches in the PoPS of Internet 2.

We are now in talks to place a similar one in GEANT at London.

Derek Weitzel, Edgar Fajardo Hernandez

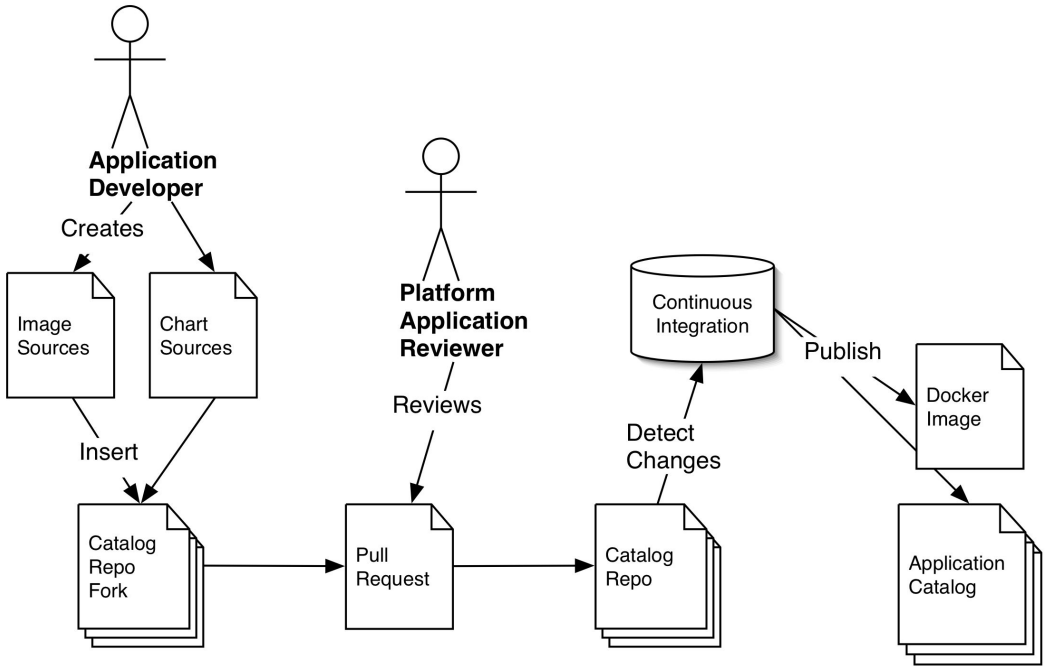
SDSC SAN DIEGO SUPERCOMPUTER CENTER

Extra slides

Evolving Security Model

Application Security for the Edge

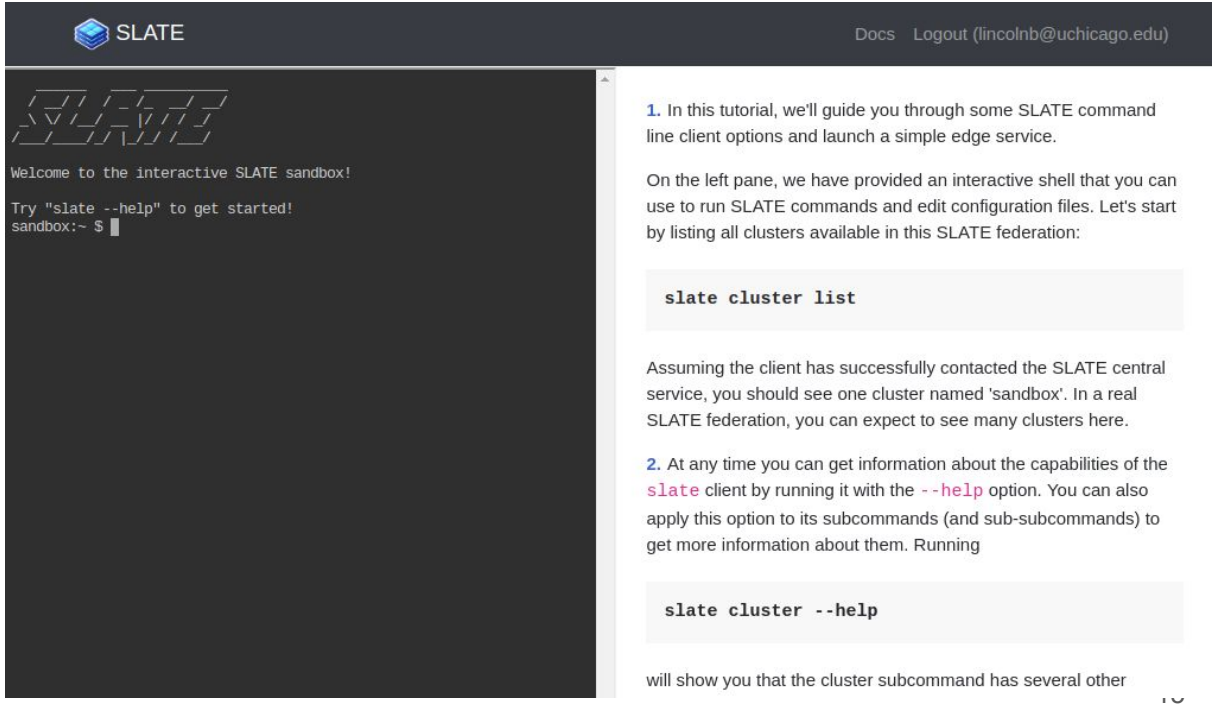
- We have considered the question of meeting sites' cybersecurity policies
- Discussions with community started: http://bit.ly/app-sec-edge
- Feel free to directly comment



Get a feel for it - SLATE "Sandbox"

- Starts a tutorial environment inside a kubernetes pod with the slate client
  - Runs an instance of the SLATE API and exposes the cluster
- Anyone can make a temporary account, try out the command line interface, and deploy a simple web server application

https://sandbox.slateci.io:5000/



SLATE provisioning options

- **SLATE**Lite (for a quick evaluation using Docker):
  - https://github.com/slateci/slatelite
- Zero to k8s+**SLATE** script on a bare edge server:
  - Installs everything necessary starting from a fresh CentOS system
    http://jenkins.slateci.io/artifacts/scripts/install-slate.sh
- "Managed" install
  - **We** will SSH to your site, set it up, and hand you the configured machine.
- Full install
  - **You** install Kubernetes, download **SLATE** client and register your cluster

Registering a cluster

```
$ slate cluster create atlas-t2-xyz \
  --group atlas-xyz-admins \
  --org "ATLAS Tier 2 XYZ"
$ slate cluster allow-group atlas-xcache
```

- Join a kubernetes cluster to a SLATE federation
  - Specifying the group which will administer it and the organization which owns the resource
- Grant users access to deploy applications on the cluster
  - In this case, just the atlas-xcache group

XCache deployment process (more details)

- As XCache requires special resources this has to be communicated between Ilija and the site but is done only once:
  - Dedicated node labeled in K8s.
  - Storage should be JBODs organized.
  - Endpoint protocol registered in AGIS.
- Ilija takes over and creates secrets, server, reporting, monitoring, activates protocol in AGIS.
  - All of that is two commands and takes 30 seconds.
- Full update of all the caches in SLATE should take less than 20 min.
  - 10-15 minutes for dockerhub to rebuild image
  - 1 minute to stop running instances
  - 1 minute to start them again
  - 3 minutes to check everything worked
  - Even stress testing is containerized and Ilija can run a very realistic stress test against any xcache in matter of minutes (from Google Computing Engine)

Monitoring

Wealth of information collected even without any direct XRootD monitoring (summary or detailed stream).

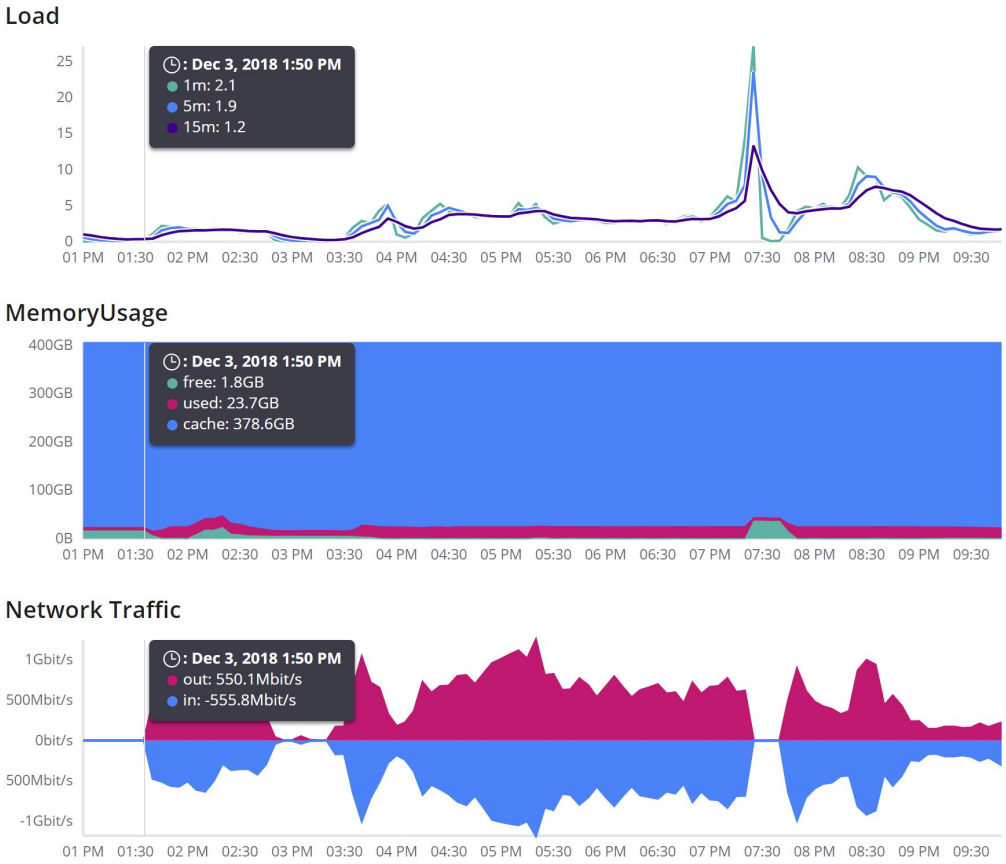
Node state (load, mem, network).

Per pod/container event and resource usage.

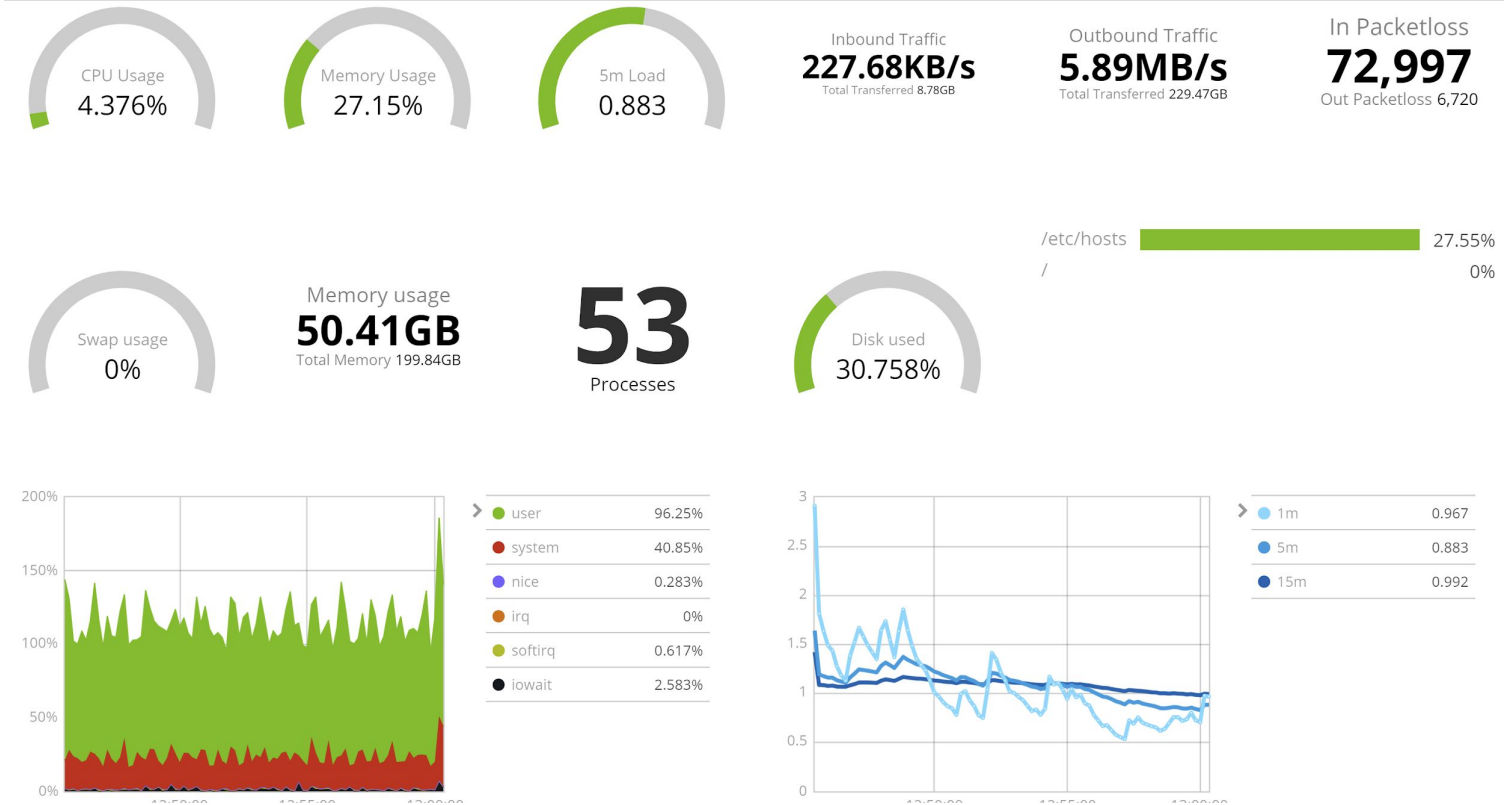
Logs. Fully searchable.

All info shipped to Elasticsearch at UChicago.

WIP - Prometheus-based monitoring



Monitoring - ES & Kibana



Monitoring continued

Really convenient logging

- No need to contact anyone
- No need to log in anywhere
- Powerful search
- All the services logs in the same place.
- Kept for 10 days.

Infrastructure / Logs

BETA

kubernetes.pod.name: atlas-xcache-xcache-global-bdc76dbd9-pgqc4

Stream live

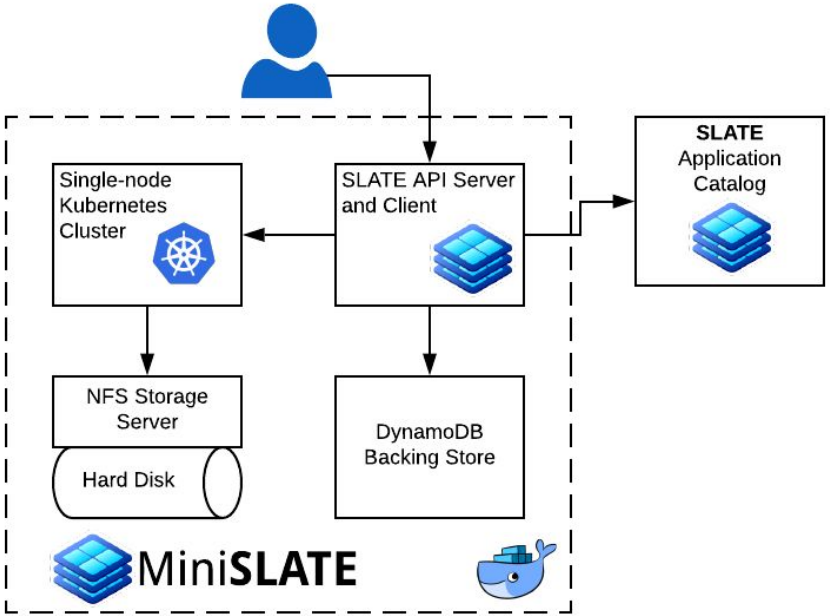
```
2019-03-02 11:02:25.908
2019-03-02 11:02:25.969   190302 10:02:25 5263 XrootdXeq: usatlas1.51
                          system.svc.cluster.local pub IPv4 login
2019-03-02 11:02:25.969   190302 10:02:25 5263 usatlas1.51852:261@192-170-227-156.prometheus-operator-kubelet.kube-system.svc.cluster.
                          ofs_open: 0-600
                          fn=/root:/xrootd.aglt2.org:1094/pnfs/aglt2.org/atlasdatadisk/rucio/mc15_13TeV/85/72/EVNT.06479607._035168.poo
2019-03-02 11:02:26.030   190302 10:02:26 5263 XrdFileCache_Manager: info Cache::Attach()
                          root://261@xrootd.aglt2.org:1094//atlas/rucio/mc15_13TeV/85/72/EVNT.06479607._035168.pool.root.1
2019-03-02 11:02:26.030   190302 10:02:26 5263 XrdFileCache_Manager: info Cache::Attach()
                          root://261@xrootd.aglt2.org:1094//atlas/rucio/mc15_13TeV/85/72/EVNT.06479607._035168.pool.root.1
2019-03-02 11:02:26.135   190302 10:02:26 5263 XrdFileCache_File: info ioActive block_map.size() = 0
                          /atlas/rucio/mc15_13TeV/85/72/EVNT.06479607._035168.pool.root.1
2019-03-02 11:02:26.135   190302 10:02:26 5263 usatlas1.51852:261@192-170-227-156.prometheus-operator-kubelet.kube-system.svc.cluster.
                          ofs_close: use=1
                          fn=/root:/xrootd.aglt2.org:1094/pnfs/aglt2.org/atlasdatadisk/rucio/mc15_13TeV/85/72/EVNT.06479607._035168.poo
2019-03-02 11:02:26.143   190302 10:02:26 5263 XrdFileCache_IO: info IOEntireFile::Detach() 0x856a9700
2019-03-02 11:02:26.144   190302 10:02:26 5263 usatlas1.51852:261@192-170-227-156.prometheus-operator-kubelet.kube-
                          system.svc.cluster.local ofs_close: use=0 fn=dummy
2019-03-02 11:02:26.145   190302 10:02:26 5263 XrootdXeq: usatlas1.51852:261@192-170-227-156.prometheus-operator-kubelet.kube-
                          system.svc.cluster.local disc 0:00:01
```

Sat 02
03 AM
06 AM
09 AM
12 PM
03 PM
06 PM
09 PM

Mini**SLATE**

A development environment for SLATE

- **Create a stand alone, miniature SLATE federation for development**
- Follows an Infrastructure as Code pattern
- Enclosed within Docker
  - Little dependency clutter
    - Python, Docker, Docker-Compose
  - Environment consistency
- Completely Destructible
  - Destroy and recreate at will
  - Mount code from host safely
- Batteries Included
  - Full development kit
  - All required software and useful tools are installed when the Docker image is built



Installing Mini**SLATE** (https://github.com/slateci/minislate)

```
$ git clone https://github.com/slateci/minislate.git
Cloning into 'minislate'...
$ cd minislate
$ ./minislate init
(...)
DONE! MiniSLATE is now initialized.
$ ./minislate slate app install nginx --group ms-group --cluster ms-c

Installing application...
...
Successfully installed application nginx as instance ms-group-nginx-default with ID
instance_tey72YzGYuw
```

If you'd like to try out SLATE

- Homepage: http://slateci.io
- Slack: http://bit.ly/slate-slack-03
- Discussion list
- Or just email robert.w.gardner@cern.ch

Acknowledgements

- SLATE team members in particular who did the work and provided input
  - Lincoln Bryant
  - Ben Kulbertis
  - Chris Weaver
  - Jason Stidd
- SLATE dashboard
  - Ilija Vukotic
- SLATE portal
  - Jeremy Van
- SLATE website
  - Shelly Johnson