



SLATE Project Update

slateci.io

Shawn McKee for the **SLATE Project**
HEPiX Fall 2019 Meeting - Amsterdam
October 18, 2019



The SLATE Project

SLATE: Services Layer At The Edge

(Funded by NSF Aug 2017, Award [#1724821](#))

- An NSF DIBBs award, "SLATE and the Mobility of Capability"
- Equips the SciDMZ with a service orchestration platform, federated to create scalable, multi-campus science platforms
- Underlayment-as-a-service (UaaS) for platform builders & science gateway developers

This talk will provide a review of **SLATE** as well as new updates since the spring talk:

<https://indico.cern.ch/event/765497/contributions/3351206/>



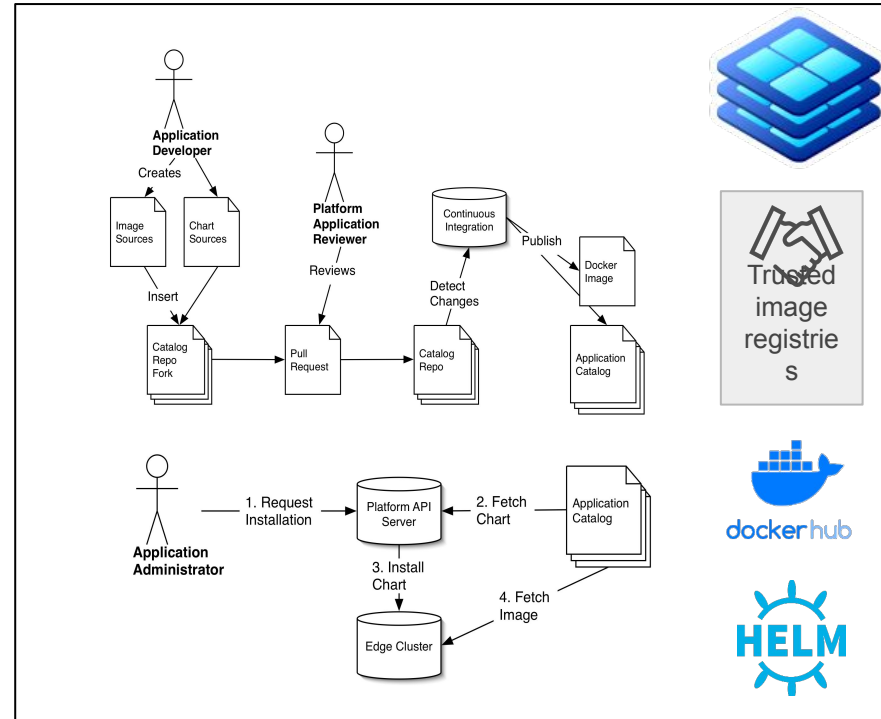
Introduction to SLATE



- **Service delivery** via **container orchestration** in **edge** networks
- The **challenges** are many
 - security and trust of container images
 - adoption of a new operations model
- The benefits are:
 - **ease ops burden on site admins**
 - **more quickly deploy updates and new services**

SLATE in a Slide

- [SLATE](#) - a value added K8s distribution
 - Support for CVMFS, ingress controller (multi-tenant, scoped privileges), Prometheus monitoring, **curated application catalog w/ Jenkins CI**
- Site security & policy conscious
 - SLATE works as an unprivileged user
 - Single entrypoint via **institutional identity**
 - Site owner controls group whitelists & service apps; **retains full control**
- With OSG, WLCG, [trustedci.org](#) & others working to establish a "CISO compliant" security posture and **new trust delegation model**



SLATE creates secrets and XCache deployment on cluster

XCache Container Download

Kubernetes objects instantiated

Pod starts up, registers itself in AGIS



Upgrades are as simple as re-deploying.

A data caching network deployed in less than 20 minutes.

Towards a NoOps Model with SLATE



NoOps - the concept that an IT environment can become so automated and abstracted from the underlying infrastructure that there is no need for a dedicated team to manage **software** in-house.

- Remotely manage edge services at sites by **expert teams** from **trusted organizations**
- Deploy updates more quickly & introduce new services more easily
- Save time and effort for local site admins via **NoOps**
- Edge federation via lightweight server/client overlay using **Kubernetes**, the industry leading container orchestration platform



- Software catalog, with push button deploy using vetted **Helm charts** (*Curated Config-knobs for Admins*)

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

The screenshot shows the SLATE Dashboard at <https://portal.slateci.io/dashboard>. The dashboard includes sections for My Instances (listing atlas-xcache-xcache-global, atlas-xcache-xcache-global, atlas-xcache-xcache-global, slate-dev-condor-ce-default, and slate-dev-condor-ce-tenthirty), News (No upcoming events), Support (Join the SLATE slack channel), Applications (Install Slate Client), Learn, and Clusters. A terminal window is overlaid on the dashboard showing the following commands:

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

Below the terminal window, there is a table of instances and a graph of Overall System Load. The table lists instances: ssi, atlas-xcache, ral-lcg2-atlas-xcache, kub-test, and lrz-atlas. The graph shows Average Load over time for various clusters: uchicago-prod, uchicago-river, uchicago-test, umich-prod, and utdsh-prod.

A New Motivation: The HL-LHC Era Challenge



arXiv:1712.06982v5 [physics.comp-ph] 19 Dec 2018

HSF-CWP-2017-01
December 15, 2017

A Roadmap for HEP Software and Computing R&D for the 2020s

HEP Software Foundation¹

ABSTRACT: Particle physics has an ambitious and broad experimental programme for the coming decades. This programme requires large investments in detector hardware, either to build new facilities and experiments, or to upgrade existing ones. Similarly, it requires commensurate investment in the R&D of software to acquire, manage, process, and analyse the shear amounts of data to be recorded. In planning for the HL-LHC in particular, it is critical that all of the collaborating stakeholders agree on the software goals and priorities, and that the efforts complement each other. In this spirit, this white paper describes the R&D activities required to prepare for this software upgrade.

"Evolutionary change towards HL-LHC is required, as the experiments will continue to use the current system. Mapping out a path for migration then requires a fuller understanding of the costs and benefits of the proposed changes. A model is needed in which the benefits of such changes can be evaluated, taking into account hardware and human costs, as well as the impact on software and workload performance that in turn leads to physics impact."

<https://arxiv.org/abs/1712.06982>

Our infrastructure needs to evolve and it seems clear it'll be in an open ecosystem - but HOW?



Alessandra Forti @ NYU, June 2019

MANCHESTER
1824

GridPF
UK Computing for Particle Physics

Not our software anymore

- EDG/EGEE/EGI/OSG/WLCG/...
 - HEP dictated the middleware development direction
 - Grid is a HEP product
- Example of a successful collaboration
CERN/Openstack
- Containers eco-system is not our sw/infrastructure anymore
 - More people use it, fix bugs etc
 - Easier to integrate with other infrastructures
 - Development not in our control, need to be part of the open source community

Mark Collier 柯理怀
@sparkycollier

Follow

CERN runs hundreds of thousands of cores of openstack. We love them. @noggin143 #fanboi



11:37 AM - 27 Oct 2016



kafka

APACHE
ARROW



Istio



Flask



kubernetes

RabbitMQ



Kubeflow

HELM



docker



We will need to
evolve our facilities
like we evolve our
software- **SLATE** has
a role to play

**NEXT
GEN
TIER2**

A fundamental **redesign** of our Tier2 complex is needed to provide **flexibility** to innovate services (CI/CD pipelines), **reproducibility** to ensure reliability, and **security** to sustain new operations models



Challenges..

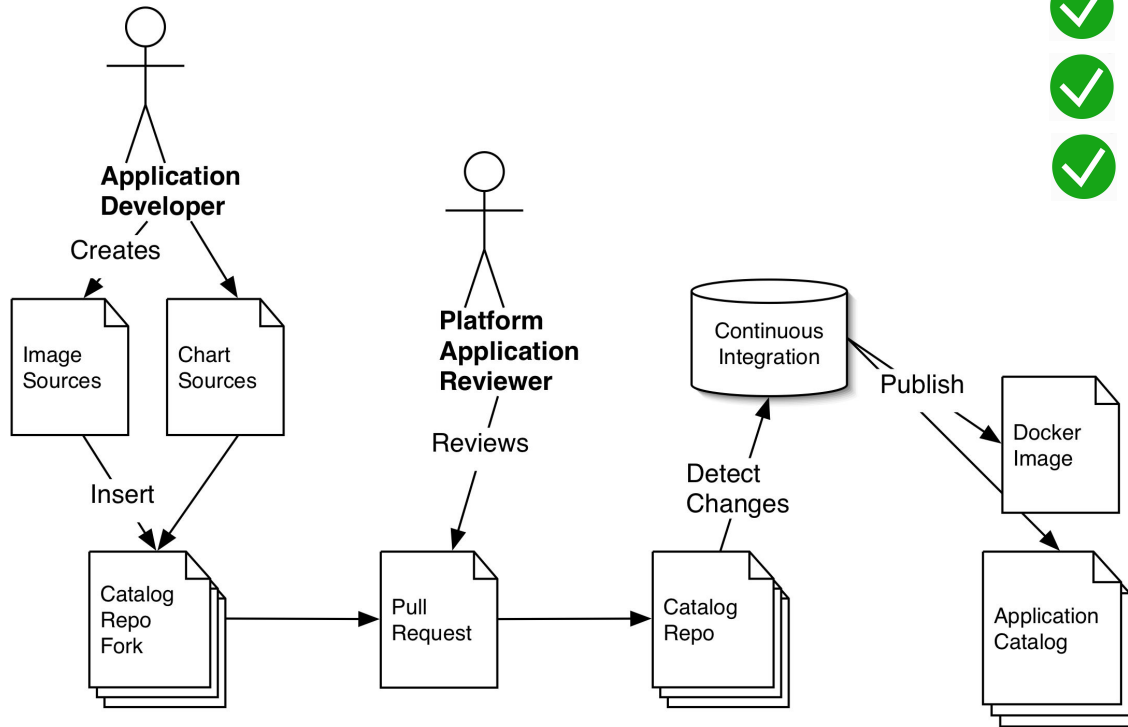
- **Site policies**
- Antiquated trust models
- Legacy services
- Legacy admin practices
- Social resistance



SLATE Requires a New Trust Model



... to gain adoption, site incentives are needed



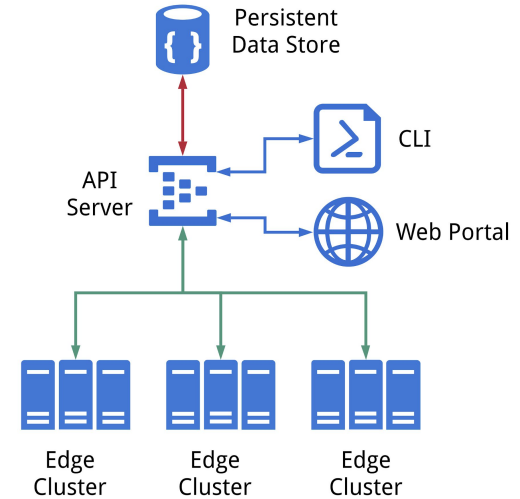
- ✓ ● [TrustedCI.org](https://www.trustedci.org) engagement
- ✓ ● OSG container security
- ✓ ● **WLCG SLATE security working group formed** ([see David Crook's presentation](#))



The 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

General SLATE News



- New applications added to the stable catalog
 - perfSONAR Testpoint
 - GridFTP
 - Globus Connect v4
 - Faucet SDN controller
- HTCondor-CE close to being ready
- **New clusters** at: **Clemson, UWashington, IU**
- Other applications, improvements here:
<https://portal.slateci.io> (have a look!)

SLATE status at US sites



- **AGLT2** (UM), **MWT2** (UC) have SLATE nodes
 - XCache up and running, being tested and managed by deployment team
 - Frontier-Squid caches
- **SWT2** in progress; **NET2** planning
- At **MWT2** is also starting to migrate existing site services to **SLATE**
- Deployed on [IRIS-HEP SSL](#) cluster at UChicago

Vulnerability patching via SLATE



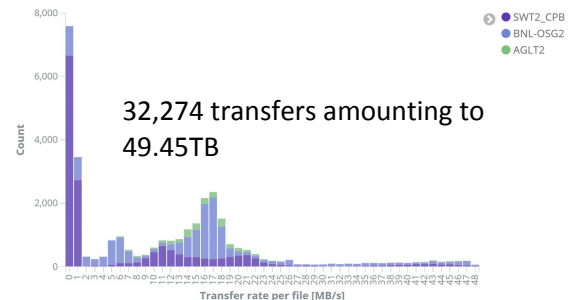
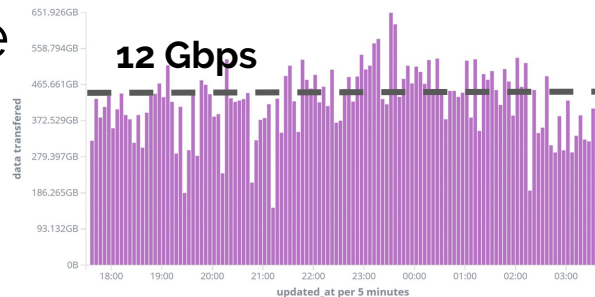
- OSG announced a vulnerability in Frontier Squid on July 26
 - **you *have* all updated right?** 😊
- SLATE instances were all updated within the hour, simply via:

```
for i in $(slate instance list | grep squid | awk '{print $4}');  
do  
    slate instance restart $i  
done
```

SLATE and XCache



- **ATLAS** is using **SLATE** to deploy an XCache-based (XRootD caching) cache network at sites
- Have been working with **ESNet** to deploy XCache *in* the network fabric, one cache up & running in production
- **Cache deployment done entirely by XCache team** - Operators in **ESNet** needed only set up the host and firewall rules for us.



SLATE Deploy XCaches



The screenshot shows the SLATE web interface. The browser address bar displays `https://portal.slateci.io/instances`. The page title is "Application Instances" and it includes a sub-header "List of deployed application instances". A search bar and a "Show 10 entries" dropdown are present. A table lists five instances, with a blue callout box highlighting the first four rows and containing the text "deployed xcaches".

Instance Name	Application	Group	Cluster	Instance ID
atlas-xcache- xcache-global	xcache	atlas- xcache	lrz-atlas	instance_ic4A08wi9oQ
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

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:

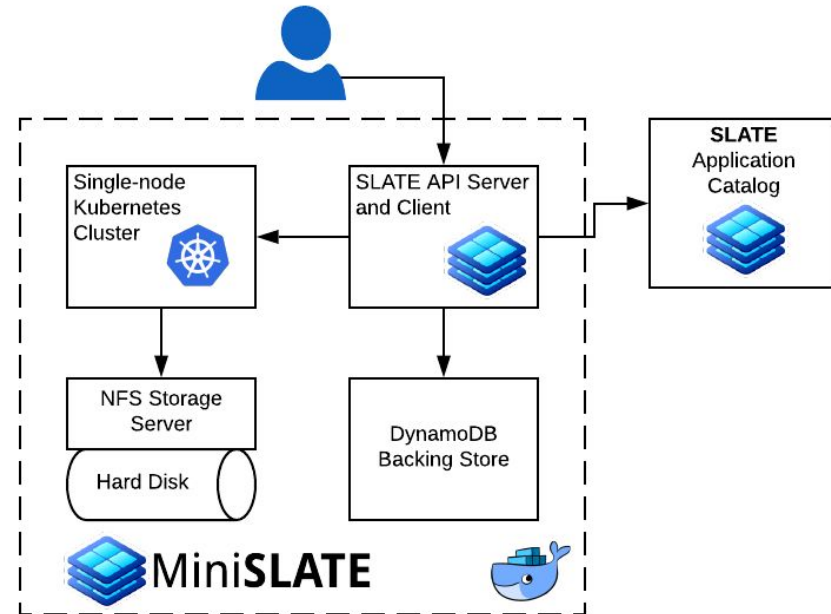
- Automatic core dump collection (part of XCache)
- Containerized environment makes it easier to debug

MiniSLATE

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



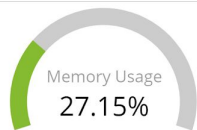
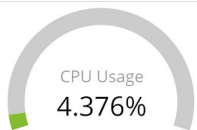
Installing MiniSLATE (<https://github.com/slateci/minislate>)



```
$ 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
```

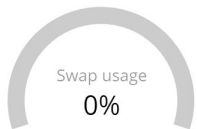
SLATE Monitoring - ES & Kibana



Inbound Traffic
227.68KB/s
Total Transferred 8.78GB

Outbound Traffic
5.89MB/s
Total Transferred 229.47GB

In Packetloss
72,997
Out Packetloss 6,720

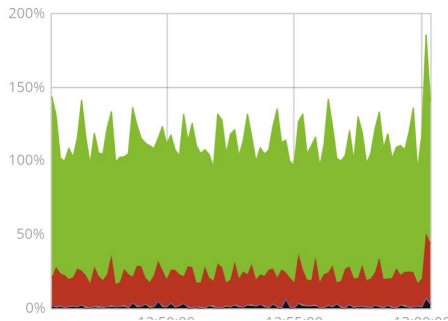


Memory usage
50.41GB
Total Memory 199.84GB

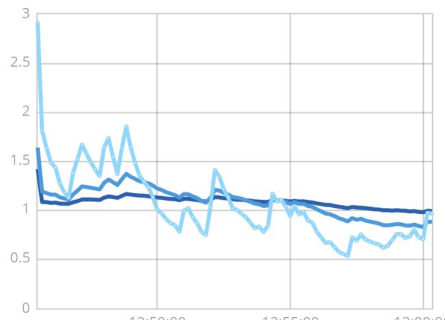
53
Processes



/etc/hosts 27.55%
/ 0%



user	96.25%
system	40.85%
nice	0.283%
irq	0%
softirq	0.617%
iowait	2.583%



1m	0.967
5m	0.883
15m	0.992

SLATE Grafana Dashboards - Beta Version

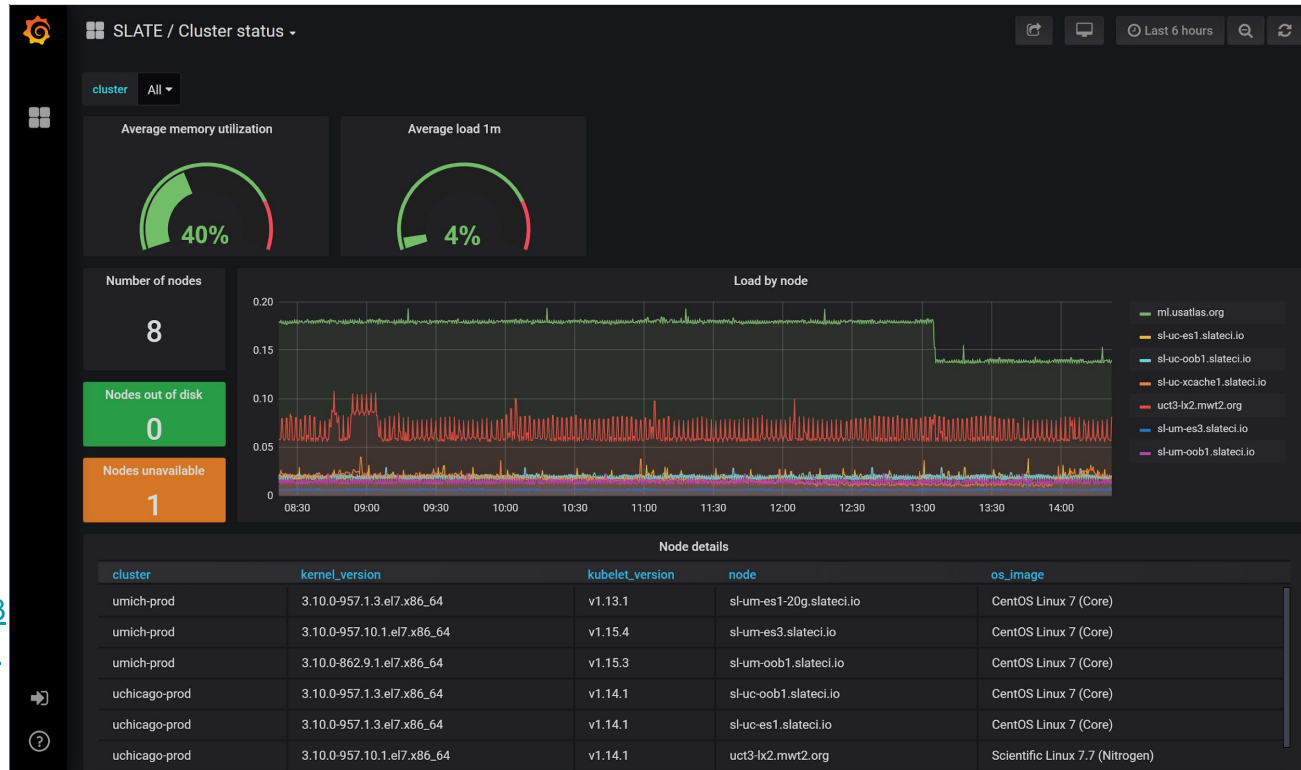


We have set up our SLATE clusters to report metrics for display by Grafana

We are collaborating with the OSiRIS project, hosting monitoring data in OSiRIS S3 buckets

Live dashboard link ->

<http://monitoring.umich-prod.slateci.net/d/3/T3FQKnZk/slate-cluster-status?orgId=1&var-cluster=All>



Interested? Get a feel for SLATE via the SLATE "Sandbox"



<https://sandbox.slateci.io/>

- Start 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

SLATE

Docs Logout (lincoln@uchicago.edu)

Welcome to the interactive SLATE sandbox!

Try "slate --help" to get started!

sandbox:~ \$

1. In this tutorial, we'll guide you through some SLATE command line client options and launch a simple edge service.

On the left pane, we have provided an interactive shell that you can use to run SLATE commands and edit configuration files. Let's start by listing all clusters available in this SLATE federation:


```
slate cluster list
```


Assuming the client has successfully contacted the SLATE central service, you should see one cluster named 'sandbox'. In a real SLATE federation, you can expect to see many clusters here.
2. At any time you can get information about the capabilities of the `slate` client by running it with the `--help` option. You can also apply this option to its subcommands (and sub-subcommands) to get more information about them. Running


```
slate cluster --help
```


will show you that the cluster subcommand has several other

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
- **Please try it out at your site!**

Conclusion



- While the **grid model** and **software** will **evolve**, certain principles will always remain in our domain:
 - **distributed trust, resource aggregation & sharing, central & local ops**
- But we need **new** ones too:
 - **declarative, reproducible infrastructure** (as code)
 - federation operations (towards "[NoOps](#)")
 - open source technology alignment and leverage
- Initial efforts are promising but will require broad community buy-in and contributed efforts
- **SLATE** is eager to contribute to **next-gen TierN design**

Acknowledgements



This material is based upon work supported by the **National Science Foundation** under Grant No. **1724821**.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Many thanks to my **SLATE colleagues** for much of the content of the presentation, especially **Rob Gardner** and **Lincoln Bryant!**

Questions?



Extra slides

Analysis Systems & DOMA SSL Blueprint



- June 21-22, 2019 @ NYU
- Engagement with AS team, SSL team and representatives from NCSA, SDSC, NYU Research Computing, Google and RedHat (26 participants)
- Key outcomes ([doc](#)):

- **Kubernetes as "common denominator"**
- **Federated "substrate" project identified**

