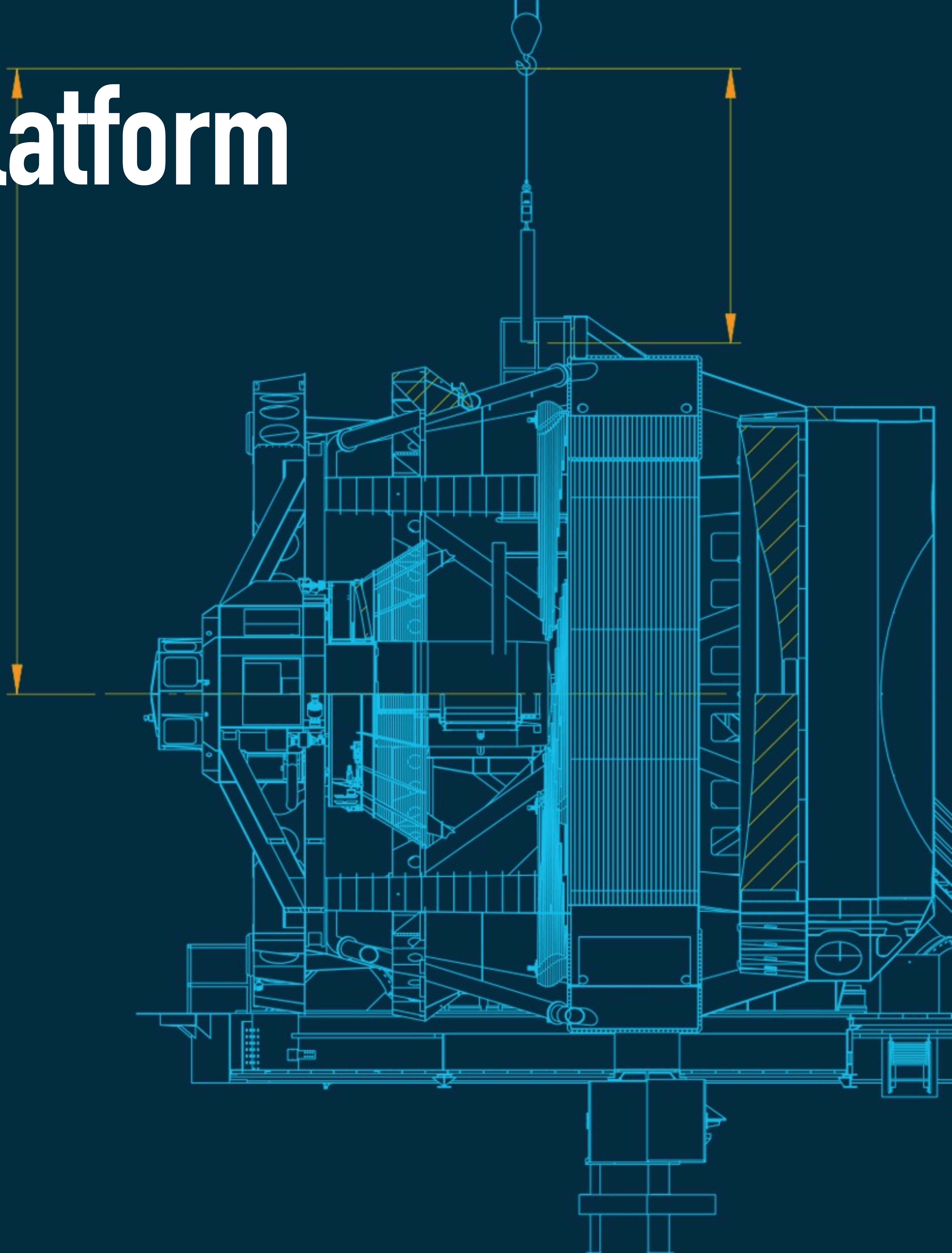
Why is the LSST Science Platform Built on Kubernetes?

Christine Banek @ LSST
cbanek@lsst.org

ADASS 2019
Groningen - October 2019

Large Synoptic Survey Telescope



Why is the LSST Science Platform Built on Kubernetes?

Agenda

1. The key unstated requirement for any scientific software
2. A (mostly) complete history of software deployment in 5 minutes
3. Introduction to Kubernetes and Helm
4. Installing the LSST Science Platform
5. Sharing services and tools next to the data

The Key Unstated Requirement of Scientific Software is…

Reproducibility

All scientific software needs reproducible…

- Creation (source control, compilation)
- Installation, deployment, and configuration
- Behavior, testing, and results

If you can't trust your software, you can't trust your science.

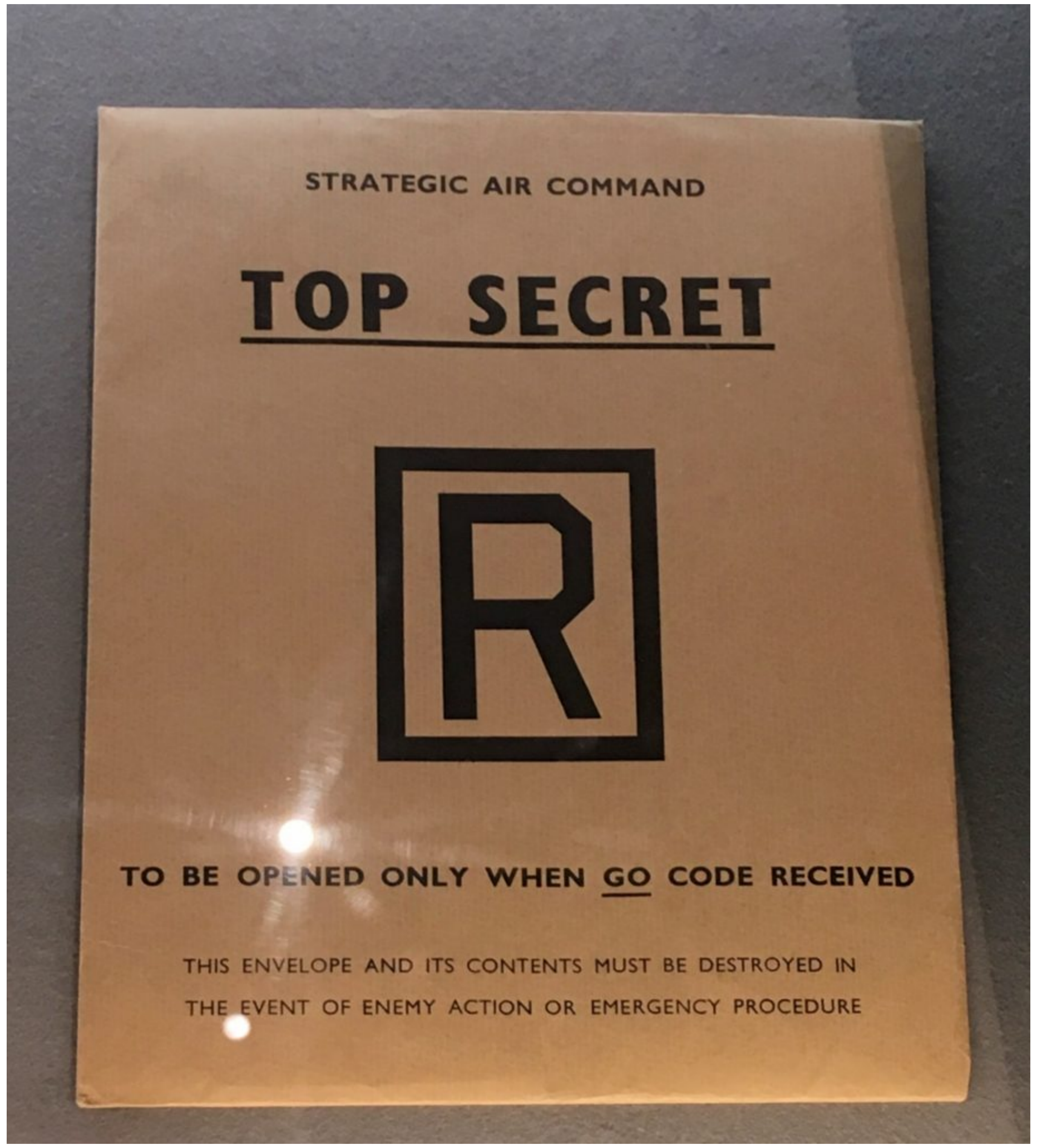
The Key Unstated Requirement of Scientific Software is…

Reproducibility

Plan R:

Always be ready to re-build and re-deploy your software from scratch.

- If you have user data, be able to backup and restore it (and test this procedure regularly)
- Best if this can be done by one person of a technical nature, not just the owner/creator



STRATEGIC AIR COMMAND

TOP SECRET

R

TO BE OPENED ONLY WHEN GO CODE RECEIVED

THIS ENVELOPE AND ITS CONTENTS MUST BE DESTROYED IN THE EVENT OF ENEMY ACTION OR EMERGENCY PROCEDURE

A (Mostly) Complete History of Software Deployment in 5 Minutes

The Four Great (and not so great) Ages of Software Deployment

1. Sysadmin
2. Virtualization
3. Containerization
4. Container Orchestration

A (Mostly) Complete History of Software Deployment in 5 Minutes

Your Local (Benevolent?) Sysadmin

1. You call your local sysadmin and tell them what you need installed or fixed
2. Sysadmin finds a machine for it to run on
3. Sysadmin installs (builds?) and configures the software for you on that machine
4. If things break, patiently wait for the sysadmin to fix it

Do not anger the sysadmin, for he has root, and you likely do not.

A (Mostly) Complete History of Software Deployment in 5 Minutes

Virtualization

1. Call your sysadmin and tell them what resources you need
2. Sysadmin creates a VM, and emails the root password to you
3. You install the software on the machine and are responsible for it
4. VMs generally live forever, or until no longer needed
5. VMs are isolated from each other and the host

Developers self-manage their software

Different virtual machines can run otherwise conflicting versions of software

A (Mostly) Complete History of Software Deployment in 5 Minutes

Containerization

1. Physical hosts or VMs run Docker and host containers
2. Containers created once at build time with all dependencies included
3. Building and installation steps not run at deployment time
4. Hosts not altered by running containers

Each application runs in isolation, no cruft from previous versions

Containers run the same anywhere

A (Mostly) Complete History of Software Deployment in 5 Minutes

Container Orchestration

1. Schedule containers on a pool of hosts
2. Containers on unhealthy hosts are restarted automatically on healthy hosts
3. Orchestrator manages DNS and routing to the right containers
4. Automatically start and stop containers (and hosts) to elastically scale to load

Benefits of containers across multiple hosts

Automatically recover from host failures

Introduction to Kubernetes and Helm

Kubernetes

- An open-source container orchestrator started by Google
- The current de facto platform for container orchestration
- Native support by cloud providers such as Amazon and Google
- Resources defined and configured by YAML documents control behavior
- Resources define the desired state, Kubernetes makes it happen

Introduction to Kubernetes and Helm

Kubernetes Resource Types

- Pod - one or more containers on the same host
- Deployment - groups of identical pods with a desired replication factor
- Service - grouping of pods that can be discovered by DNS name
- Volumes - persistent storage that pods can mount
- Configmap - configuration files that pods can mount
- Secret - configuration such as passwords or certificates that pods can mount
- Ingress - routing external network requests to a service

Introduction to Kubernetes and Helm

Helm

- A package manager for Kubernetes
- Group a set of Kubernetes resources into a Helm chart that represents an application to run in Kubernetes
- Kubernetes Resource YAML documents are first run through a templating engine that injects user values and configuration into Kubernetes resources
- Helm charts can be easily published and shared between organizations
- Helm has a set of stable charts that contains a lot of great open-source software

Installing the LSST Science Platform

LSST Science Platform Helm Charts

1. Landing page - simple web page to help users navigate to the different aspects of the Science Platform
2. Nublado - customized JupyterHub system to allow users to run notebooks next to the data
3. Firefly - IPAC portal for in-browser data visualization
4. CADC's TAP service - IVOA TAP service for QServ
5. Fileserver - NFS fileserver for persistent storage of notebooks and user data

Installing the LSST Science Platform

Pre-requisites

1. SSL certificate (letsencrypt.org)
2. DNS for the name on the certificate (Route 53)
3. Kubernetes cluster (Google Cloud)
4. GitHub OAuth client id and secret

Installing the LSST Science Platform

Installation Helper Scripts

1. git clone https://github.com/lsst-sqre/lsp-deploy.git && cd lsp-deploy/gke-develop
2. Edit nublado-values.yaml to add your GitHub OAuth info
3. ./install_tiller.sh - installs the Kubernetes side of Helm
4. ./install_ingress.sh - takes your SSL certificate as an argument, configures nginx to proxy backend services
5. ./install_lsp.sh - installs all Helm charts for the LSP
6. Set DNS to the IP address returned by ./public_ip.sh
7. Celebrate! Now you too can install the LSST Science Platform!

Installing the LSST Science Platform

Helm + Kubernetes: A Great Way to Install the LSST Science Platform!

1. Easy for internal and external users to install in minutes
2. Easy to create new Kubernetes clusters on the cloud for testing when needed
3. Helm charts manage creating and deleting Kubernetes resources
4. Configurable variables are abstracted out in a values.yaml file
5. Kubernetes namespaces prevent naming conflicts, other software can be installed alongside the LSP with no changes on either side

Sharing Services and Tools Next to the Data

Science Platform à la carte

1. Create containers with your software installed and publish them
2. Create helm charts for your services or software tools and publish them
3. Use helm to install a set of charts in a Kubernetes cluster next to your data to create a science platform with services and tools you want
4. Mix and match - use the LSP charts, non-astronomy open source charts, your own charts, or any combination!
5. More charts to come from LSST!

Why is the LSST Science Platform Built on Kubernetes?

My Favorite Reasons

1. Incredibly reproducible deployments - same on any cloud or any datacenter
2. Great for deploying services in a fault tolerant way on a cluster
3. Allows for scaling up and down individual services independently
4. Everyone with a credit card has access to a Kubernetes cluster in the cloud!
5. Great ecosystem and community developing new tools around it
6. Helm simplifies sharing services running on Kubernetes

Thank You!

https://github.com/lsst-sqre/lsp-deploy

https://github.com/lsst-sqre/charts

Additional Resources Follow

Repositories

Find us on GitHub

LSP Deployment Scripts: https://github.com/lsst-sqre/lsp-deploy

LSP Helm Charts: https://github.com/lsst-sqre/charts

Nublado: https://github.com/lsst-sqre/nublado

Firefly/SUIT: https://github.com/lsst/suit https://github.com/Caltech-IPAC/firefly

LSST fork of CADC TAP: https://github.com/lsst-sqre/lsst-tap-service

CADC on GitHub: https://github.com/opencadc

Further reading

https://docker-curriculum.com

https://docs.docker.com/get-started/

https://kubernetes.io/docs/tutorials/

https://helm.sh

https://cloud.google.com/kubernetes-engine/docs/how-to/creating-a-cluster

https://helm.sh/docs/chart_repository/

https://docs.bitnami.com/kubernetes/how-to/create-your-first-helm-chart/