



# Adopting Red Hat Satellite 6 for Lifecycle Management

Rennie Scott [rennie@fnal.gov](mailto:rennie@fnal.gov)

& Patrick Riehecky [riehecky@fnal.gov](mailto:riehecky@fnal.gov)

HEPiX Workshop Fall 2016

20 October 2016

# Introduction

Spring 2015 Scientific Linux Architecture Management (SLAM) started a new project to update the group's system management infrastructure. We decided on using Red Hat Satellite 6.

This presentation will describe the SLAM group, describe some of the reasons for choosing Satellite 6, and our experiences with the implementation.

# Why?

The current system designed to meet workstation environment management.

- We were still running Puppet 2.
- Grew to scale meet new group areas of responsibility.
- “Near misses” were become more prevalent:
  - Change management had to be done for the smallest workstation change (not always sure what Puppet will do).
- We wanted to leverage commercial resources to help underpin our small resources.

# SLAM Services

- Staff of 3 technical FTEs and 1 Manager/Architect.
- SLAM manages 454 systems, for 27 separate organizations, 146 different base configurations.
- Service Area: Scientific Linux Systems Engineering:
  - Scientific Linux Distribution: Global distribution infrastructure of SL.
  - Scientific Linux Engineering: Highest escalation support for SL, packaging, update distribution, Fermi site SL system inventory and auditing.
  - Managed Scientific Workstations: Support SL Workstations across 11 organizations.

# Online Engineering Lifecycle Services

- **Scientific Test Stand Engineering:** First level of Standard Operating Environment (SOE) design. Support both component level and DAQ testing environments.
- **Control Room System Management:** Implementing multi-monitor, scalable SOE for detector monitoring and controls.
- **Online System Engineering and Lifecycle Management:** 24x7 Production Level operations support of active data taking experiments. Engineered SOE and services focused on high uptime, high data rates, continuity of operations, and risk assessment and mitigation.
- **DAQ Infrastructure Operations Engineering:** Holistic System Engineering service to design and implement online computing infrastructure to meet experiment requirements.

# Requirements

- Completely isolate experiment environments.
  - Schedule package and configuration updates based on experiment groups.
- Reduced learning curve for new employees
- Unified provisioning approach across disparate configurations
- Rollback: Return to EXACT runtime system states
- Phased SOE promotion (Dev, Test/Integration, Prod)

# Project constraints

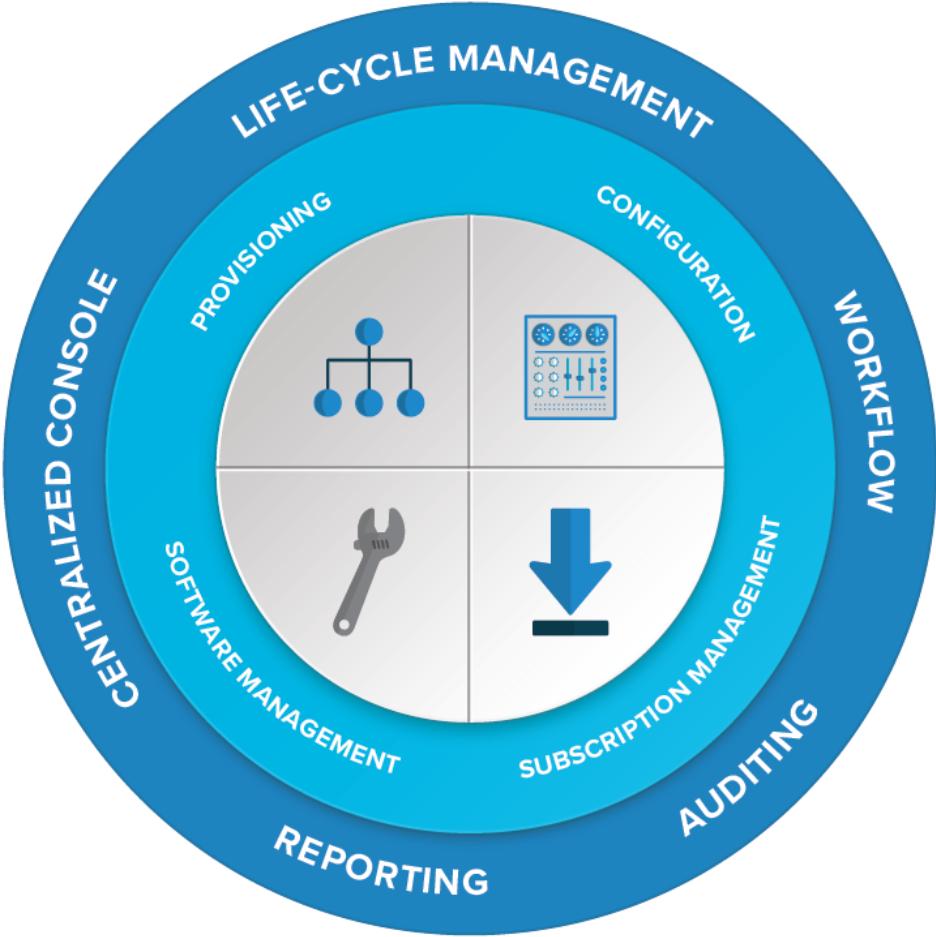
- Had to be completed by June (beam shutdown/experiment maintenance window).
- Limited senior level engineers not engaged with other high priority projects.
- Limited Project Management resources due to allocation to complete The Scientific Service Management Onboarding Project.

# Red Hat Satellite 6 Overview

- Completely new product from Red Hat Satellite 5 (SpaceWalk)
- “Red Hat’s easy-to-use system management product that allows keeping the infrastructure running efficiently, properly secured, and compliant.”
- A single centralized management tool
- Secure connection policies for remote administration
- Standardize machine configurations
- Digitally signed content
- From: [Red Hat Satellite 6 website](#)



# Satellite Overview



Complete life-cycle management in 1 console

# Satellite Architecture

Open source upstream is Katello. It aggregates various open source products into a single collected workflow.

Includes:

- Puppet 3 (configuration management)
- Pulp (repository management)
- Foreman (External Node Classifier)
- Open SCAP (Auditing and compliance)
- Candlepin (subscription management)
- IPMI Web console
- System Administration Job scheduler

# Patrick Riehecky took on the project

- Had to act as his own PM
- Tasks:
  - Design a architecture that would be the central core of all our operations and services
    - Implement in a high available environment.
    - Designed to scale to the foreseeable future.
    - Ability to meet changing customer needs.
  - Test and deployment environments.
  - Rebuild and redesign all core Puppet modules.
  - Document and train the rest of the group members and include them on design issues.

# Issues encountered

- New product (very immature product) 6.1
  - Limited documentation
  - Limited deployment and design best practices experience at Red Hat
  - 20 Product Defects filed
  - Over 50 RFEs filed for workflow issues
    - Sat 6.2 Feature frozen time frame

# Workflows

- Sample workflow - not worthy of a change ticket:
  - A ticket comes in "Please add a user to my hosts".
  - Authorized Admin locates relevant puppet class.
  - Authorized Admin adds user to class parameters.
  - Next puppet run adds user.

# Workflows (cont.)

- A sample workflow - with change ticket:
  - A ticket comes in "Please update my system with all pending errata".
  - Pilot system designated (via pre-existing process).
  - Change Ticket approved for build and test.
  - New content view is created.
  - Content view is promoted to TESTING.
  - Puppet is run and packages are updated.
  - User approves Pilot system behavior.
  - Change is approved for Go Live.
  - Content view is promoted.

# Differences from current infrastructure

- Less debug logging than existing infrastructure
- No out-of-the-box method to reference actions with tickets
- Completely new paradigm for crusty old sysadmins

# Improvements seen so far

- Granular level of system attributes and characteristics
  - Detailed YAML configurations
- A unambiguous workflow
- Exact point in time replication on bare metal
- Direct paths to virtualization, cloud, and container(future proof)
- Its not weird or homegrown
- Vendor supported
- Service Now Integration



# Future activities

- Rolling out OpenSCAP (FY17 approved project)
- Upgrade to Sat 6.2 or 6,3
  - Remote exec
  - Performance improvements
  - 400 bug fixes
- Start to dig into automation, orchestration, and service integration.
- Customer and management reporting.
- Much more integration with Service Now with CMDB, management orchestration, and reporting.

# Summary takeaways

- Don't a release below Satellite 6.2
- There is a paradigm shift and learning curve from traditional system administration.
- There is a heavy upfront cost in design.
  - Either trial and error or very careful scenario planning.
- Just works.
- Reduced duplication and truly self-documenting.

# Supplemental Info

- Satellite 6 management scripts published at github:
- \* <https://github.com/RedHatSatellite>
- \* Community driven tools using the API for mass management
- \* Supervised by Red Hat's Satellite team

# Questions

Contacts:

Rennie Scott [rennie@fnal.gov](mailto:rennie@fnal.gov)

Patrick Riehecky [riehecky@fnal.gov](mailto:riehecky@fnal.gov)