Packaging and Using Services in Kubernetes

Second K8s-HEP Meetup

Brian Lin
University of Wisconsin–Madison

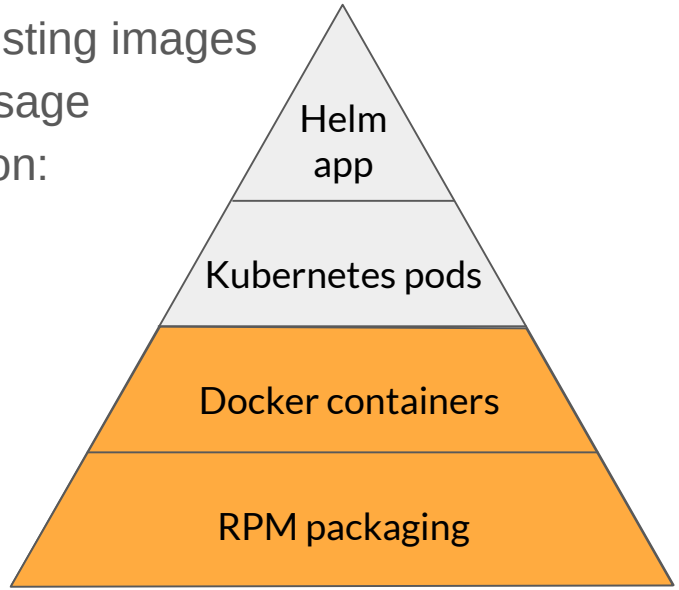
CENTER FOR
HIGH THROUGHPUT
COMPUTING

OSG Software Team

- 4 FTEs based out of UW–Madison
- We curate and distribute grid services for OSG sites
- Most of our development effort focuses on:
  - Packaging upstream grid services
  - Integrating our grid services, i.e. “glue” tooling
  - Maintaining abandoned software and transitioning to replacement services
  - Testing the integrated software stack

Containerized Grid Services

- Service container images based off the same RPM packaging used across sites
- Images re-built weekly to ensure up-to-date OS software
- Two different tags to differentiate production vs testing images
- Images use Supervisor to allow for stand-alone usage
- Service containers currently deployed in production:
  - Frontier Squid
  - Hosted CE
  - OSG VO worker node
  - XCache
- Available through Docker Hub: https://hub.docker.com/u/opensciencegrid



Container Release Process Take 1

- Current policy: https://opensciencegrid.org/technology/policy/container-release/
  - `fresh` images rebuilt weekly based on pre-release RPM packaging
  - Images are also tagged with a timestamp for easy rollback
  - Functioning images verified by users are then manually tagged as `stable`
  - Guarantees that production images are exactly the same as tested images
- Lessons learned
  - Requires regular coordination and testing processes between users and developers
  - Tagging `stable` tags risks including untested/lightly tested changes from `fresh`
  - `stable` tags may contain stale OS packaging if new `fresh` images aren't tested and released regularly
  - Manual release process with plenty of room for error

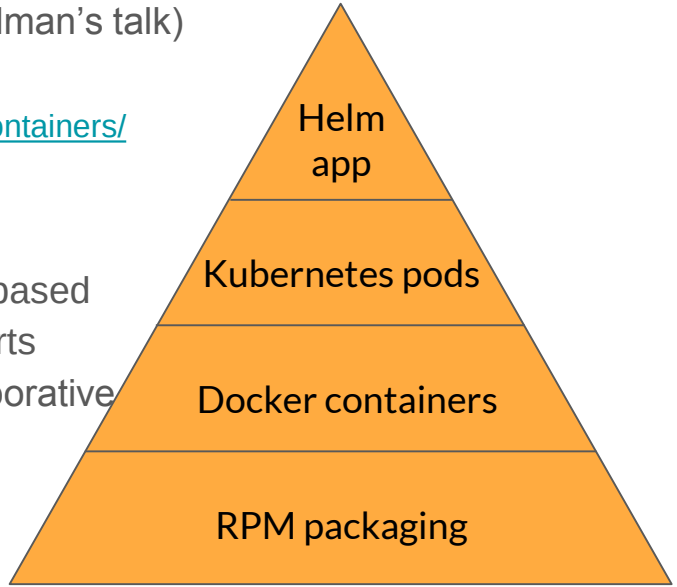
Container Release Process Take 2

Rough policy tentatively targeted for the OSG 3.6 (Q1 2021)

- Images still rebuilt at least weekly with two independent tag streams:
  - `release` based on the OSG release Yum repos, i.e. production-ready
  - `testing` based on the OSG testing Yum repos, i.e. passed automated integration tests
- Timestamp tags across both streams for easy rollback
- Generate “release candidate” `release` and `testing` images for proposed Dockerfile modifications:
  - Avoid rolling up multiple Dockerfile changes into prod by testing discrete changes
  - Removes manual steps in the release process
  - Cons: risk of differences between the tested image and the newly built images once proposed changes are merged

Orchestrating OSG Container Images

- Kubernetes pods
  - No native system for distributing Kubernetes manifests
  - Experimenting with GitOps internally (see Brian Bockelman's talk)
  - Still, opportunities for "distributing" example manifests: https://opensciencegrid.org/docs/resource-sharing/os-backfill-containers/
- SLATE (Helm charts)
  - Helm registry distribution model is familiar to our team
  - OSG Frontier Squid and Hosted CE images originally based on images developed by the SLATE team for their charts
  - Today's Hosted CE SLATE app is the result of a collaborative effort between the OSG Software and SLATE teams, including in-person hackathons



SLATE Hosted CEs

- Hosted CE: Centrally operated HTCondor-CE submitting jobs to remote sites over SSH
- Previous Hosted CE iterations installed on Chicago-based VMs
  - Manual installation requiring specialized grid knowledge
  - Installs prone to losing track of hot patches, resulting in staleness
- Currently 18 OSG Hosted CEs are deployed via SLATE
  - Automates installation, providing simpler interface to Operators
  - Intended state much easier to track through version control (including hot patches)
  - Theoretically allows for easy transition of services between data centers
  - Helm chart has allowed for rapid development of the entire WMS stack: we recently demonstrated end-to-end job submission to an Ubuntu test cluster
- SLATE features that we're looking forward to:
  - Improved certificate management (https://cert-manager.io/?)
  - State retention within a SLATE cluster
  - Support for GitOps-style deployments

Questions?