

Zuul Interoperability

What's Zuul?

- ▶ Project gating CI system
- ▶ Focus on Git and code review workflows
- ▶ Cross-project, multi-tenant, multi-source
- ▶ Open source/design/development/community
- ▶ Host it yourself or use a hosted service
- ▶ Represented by the OSF

A Brief History

- ▶ 2012: Speculative scheduler for Jenkins (1.0)
- ▶ 2013: Mult-master Jenkins scale out (2.0)
- ▶ 2016: Experimental Ansible launcher (2.5)
- ▶ 2018: Interoperable rearchitecture (3.0)

<https://opensource.com/article/20/2/zuul>

Interoperability

- ▶ Ansible as a job description language
- ▶ Shares job definitions between repos
- ▶ Central "standard library" of building blocks
- ▶ Supports multiple source connections
- ▶ Runs jobs on many environments and multi-node
- ▶ Cross-repo/cross-source dependencies

Ansible Executor

- ▶ Shouldn't invent a domain-specific language
- ▶ Avoid "not implemented here" temptation
- ▶ Something people were already using
- ▶ Made for orchestrating across multiple hosts
- ▶ Extensible with Python modules
- ▶ Can still easily run arbitrary shell scripts

Job Sharing

- ▶ Workflow/triggers not tied to individual jobs
- ▶ Reference jobs and roles from any project
- ▶ Vary jobs or inherit from them
- ▶ Designed for secure reuse between separate
 - ▶ Branches
 - ▶ Projects
 - ▶ Tenants
 - ▶ Connections

Standard Library

- ▶ Reusable building blocks (playbooks, roles)
- ▶ Generic jobs for a variety of languages
- ▶ Thoroughly documented and self-testing
- ▶ Can consume directly through Git connection
- ▶ Can consume from a locally-hosted fork
- ▶ Advisory testing performed by other sites

<https://opendev.org/zuul/zuul-jobs>

Multi-Connection

- ▶ Connect to many code review systems at once
- ▶ Bridges different review and hosting software
 - ▶ Gerrit
 - ▶ GitHub
 - ▶ Pagure
 - ▶ Git (static hosting)
 - ▶ GitLab (experimental)
 - ▶ BitBucket (in progress)

Job Environments

- ▶ OpenStack (VMs, containers, bare metal)
- ▶ Amazon (EC2, EKS)
- ▶ Azure (in progress)
- ▶ Google Cloud (GCE, GKE)
- ▶ Kubernetes pods
- ▶ OpenShift
- ▶ Static servers and appliances
- ▶ Multi-node (homogeneous or heterogeneous)

Dependencies

- ▶ Implicit in dependent pipeline manager order
- ▶ Explicit in commit message footer
- ▶ Works between projects (repositories)
- ▶ Works across separate source connections
- ▶ Enforces merge sequencing
- ▶ Provides prepared Git repositories

Thanks and Links

Visit <https://zuul-ci.org/> for more information

<http://fungi.yuggoth.org/presentations/2020-cdf>

Copyright 2020 Jeremy Stanley <fungi@yuggoth.org>

This work is licensed under a Creative Commons Attribution 3.0 Unported License.

<http://creativecommons.org/licenses/by/3.0/legalcode>