



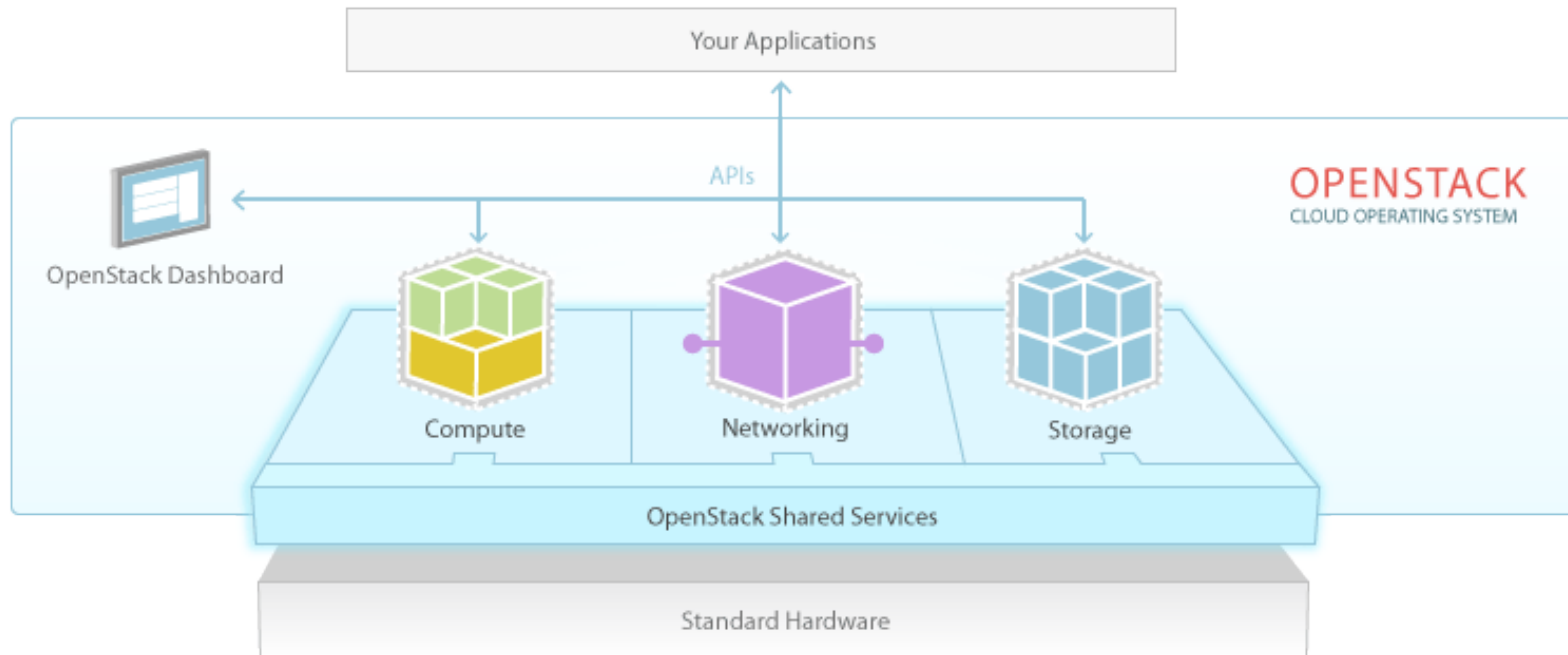
# Infrastructure-as-Code and CI Infrastructure at Open Stack

A look at one of the largest CI systems and system administration

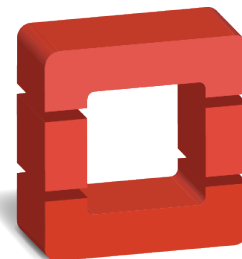
Andreas Jaeger  
Product Manager  
SUSE [aj@suse.com](mailto:aj@suse.com)

# OpenStack

Open source software for building private and public clouds



# Project History



openstack™

- Provide components for Infrastructure-as-a-Service
- Started by Rackspace and NASA in July 2010
- **Today:** > 200 companies involved in ecosystem
- Releases every 6 months:
  - So far: Austin, Bexar, Cactus, Diablo, Essex, Folsom, Grizzly, Havana, Icehouse, Juno, Kilo, Liberty, Mitaka, Newton
  - Next ones: Ocata, Pike, Queens

# OpenStack Newton

> **5,900,000** Lines of code

> **196,000** Reviews

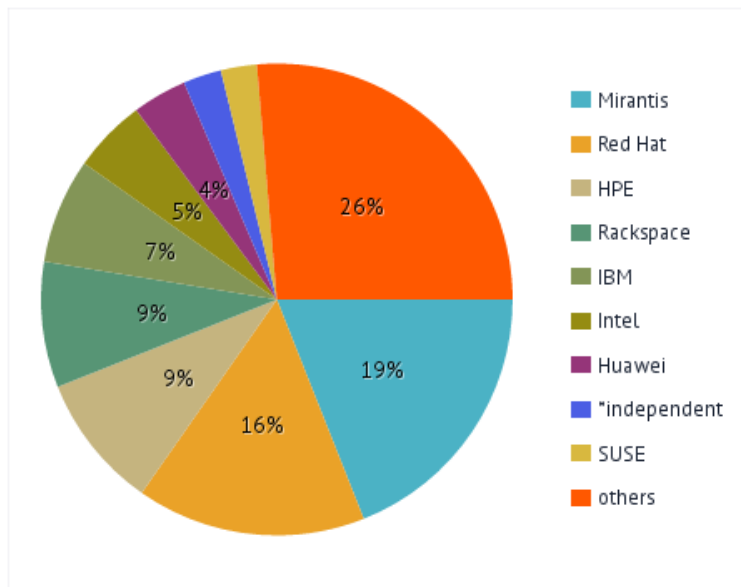
> **40,000** Commits

**2680** Contributors

**183** Organizations

# Contributors

Contribution by companies



Source: <http://stackalytics.com/?metric=marks&release=newton>

- 
- Unaffiliated individuals
  - Commercial entities
  - Nonprofit organizations
  - National and local governments
  - Number, quality, and area of contributions can change daily

**“The most insane CI infrastructure  
I've ever been part of”**

**– Alex Gaynor on IRC #openstack-infra**

# OpenStack Projects include:

## Servers:

- Compute (nova)
- Object Storage (swift)
- Image Service (glance)
- Identity (keystone)
- Dashboard (horizon)
- Networking (neutron)
- Block Storage (cinder)
- Telemetry (ceilometer)
- Orchestration (heat)
- Database Service (trove)
- ...

## Client libraries:

- python-novaclient
- python-swiftclient
- python-glanceclient
- python-keystoneclient
- python-neutronclient
- python-cinderclient
- python-heatclient
- python-ceilometerclient
- python-openstackclient
- Python-troveclient
- ...

# Programs/Horizontal Efforts

- Documentation
- Infrastructure
- Oslo
  - Common libraries
- Quality Assurance
  - Integration Testing
  - Upgrade Testing
- Release Management
- Internationalization/Translation
- Vulnerability Management



# Release Management

- Time Based Releases
- Six Month Cadence
- Design summits each cycle
- Continuously Open Trunk
- Develop directly on master
- Intra-Cycle Milestone Releases
- Post-Release Stable Branches

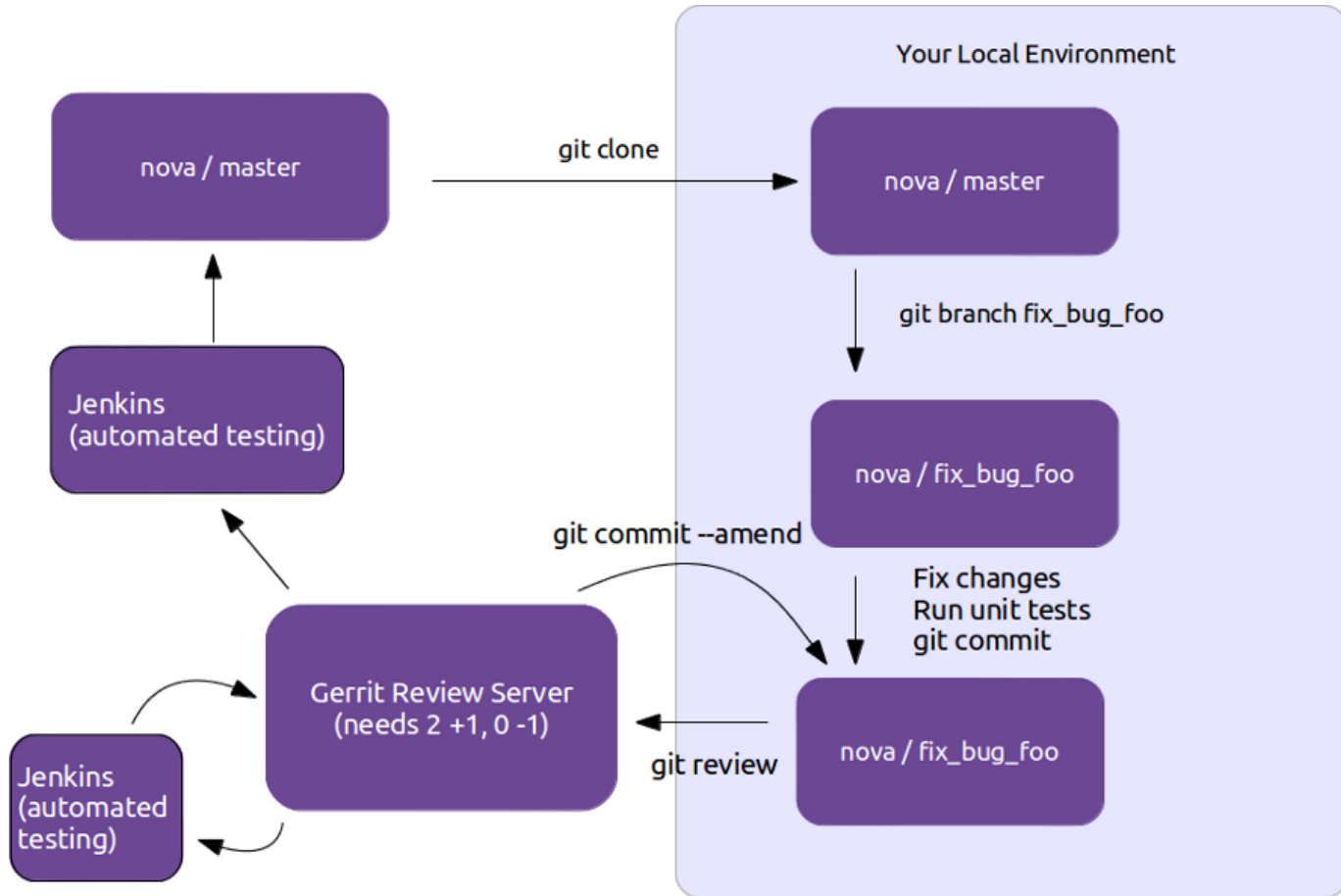
# CI Challenges

- Lots of individual projects (> 1500 git repos)
- Projects must work together
- Changes should never break the master
- Code should be syntactically clean
- Completely automated testing

# Consistent CI infrastructure

- All official OpenStack projects
- Unofficial projects
- Documentation
- Systems Administration – incl. CI

# Git Workflow



# Peer Review

- Anybody can review and leave comments and +1/-1
- Core reviewers can leave also +2/-2
- Approval by core reviewer after at least two +2s
- Automatic review by tools
- Testing on supported scenarios:
  - Different hypervisors
  - Different storage backends
  - Different databases
  - Different OSes

# Gating?

- To ensure code quality
- To protect developer – they always start from working code
- To protect tree from bad code
- Same process for everybody:
  - Transparent
  - Automated
- Testing of patch against current state of all other projects

# Gating: Sequential Merge

- Assume three changes A, B, C for testing
- Zuul will test:
  - Test A: Merge A; test
  - Test B: Merge A, B; test
  - Test C: Merge A, B, C; test
- If B fails:
  - Retest C: Merge A, C; test
- “Depends-On” across repositories
- After merge: merge-check of open reviews

# Jobs

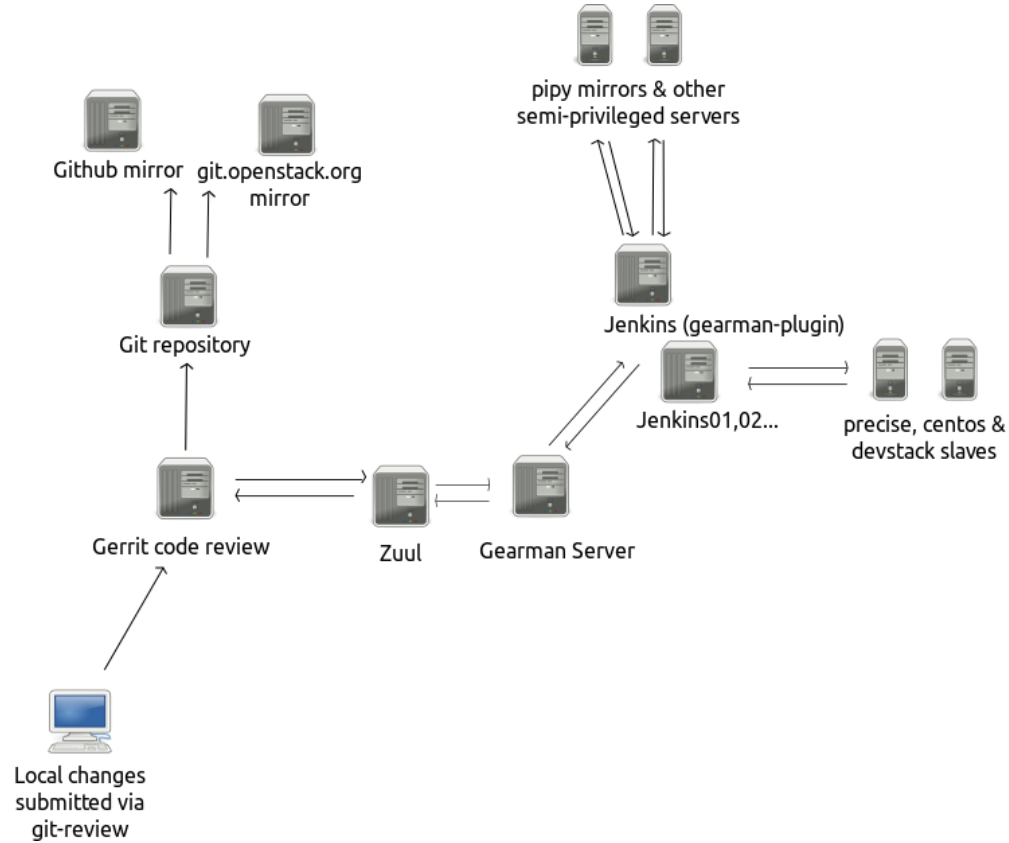
- Jobs are started on new fresh VM
- Run in clouds by Rackspace, OSIC, OVH, Bluebox, Internap, and own Infracloud
- Currently over 1500 VMs available



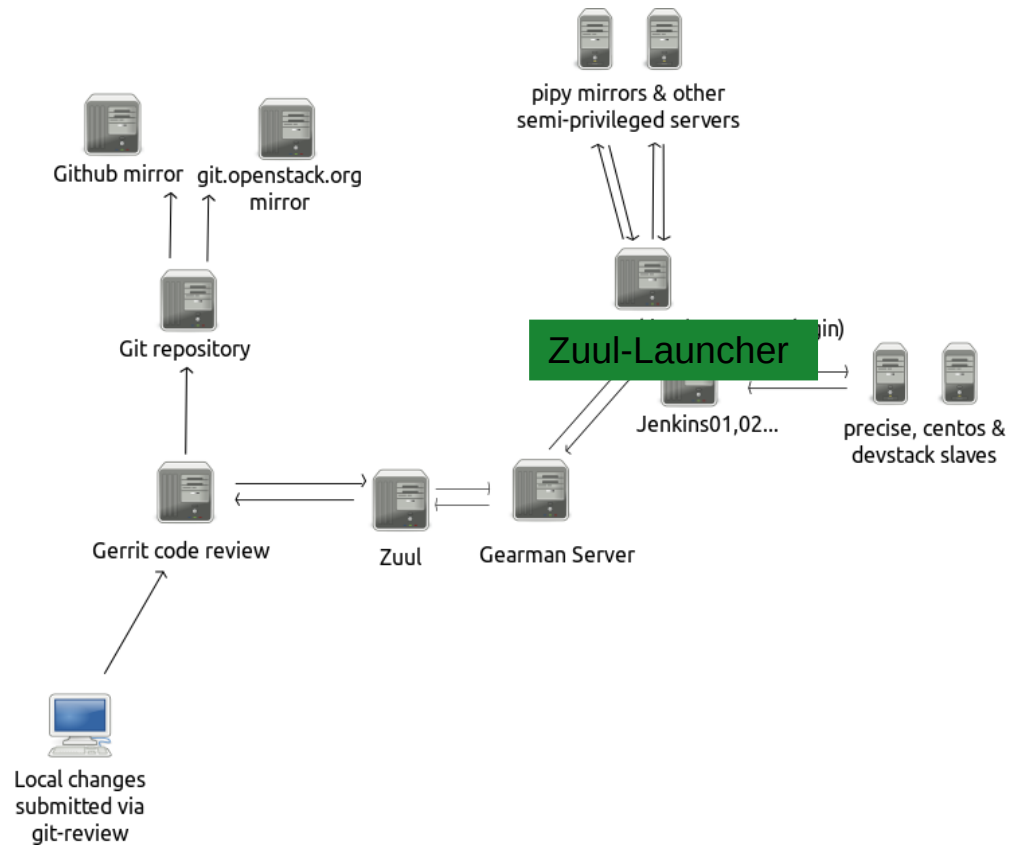
# CI Infrastructure

- Launchpad and Storyboard
- Git
- Gerrit
- Zuul
- Gearman
- Nodepool

# CI Workflow



# CI Workflow (Since 17<sup>th</sup> June)



# Jobs



Status **Zuul** Rechecks Release Reviews Bugday

## Zuul Status

Zuul is a pipeline oriented project gating and automation system. Each of the sections below is a separate pipeline configured to automate some portion of the testing or operation of the OpenStack project. For more information, please see [the Zuul reference manual](#).

Queue lengths: 3904 events, 546 results. Filters:  Expand by default:

**check** (670) **gate** (27) **post** (80)

Newly uploaded patchsets enter this pipeline to receive an initial +/-1 Verified vote from Jenkins.

Changes that have been approved by core developers are enqueued in order in this pipeline, and if they pass tests in Jenkins, will be merged.

This pipeline runs jobs that operate after each change is merged.

Change queue: **integrated**

openstack/cinder 194929,3 unknown 5 hr 59 min

Change queue: **integrated**

openstack/cinder 189614,73 unknown 5 hr 54 min

Change queue: **integrated**

openstack/cinder 162927,13 unknown

openstack/cinder 194532,20 unknown

openstack/cinder 197393,10 unknown 5 hr 54 min

Change queue: **stackforge/os-ansible-deployment**

stackforge/os-ansible-deployment 204542,1 unknown 5 hr 52 min

Change queue: **integrated**

openstack/keystone 203572,4 unknown 4 hr 54 min

openstack/horizon 181095,58 unknown 4 hr 54 min

openstack/neutron 204544,1 unknown 4 hr 49 min

openstack/keystone 203525,1 unknown 4 hr 49 min

openstack/neutron 204546,1 unknown 4 hr 48 min

openstack/heat 204301,1 unknown 4 hr 10 min

openstack/neutron 201727,7 unknown 4 hr 10 min

Change queue: **openstack/api-site**

openstack/api-site 5310efb unknown 6 hr 10 min

Change queue: **openstack/api-site**

openstack/api-site ed4f6ae unknown 6 hr 3 min

Change queue: **stackforge/surveil**

stackforge/surveil 7e74e21 unknown 5 hr 55 min

Change queue: **stackforge/surveil**

stackforge/surveil 76194ee unknown 5 hr 54 min

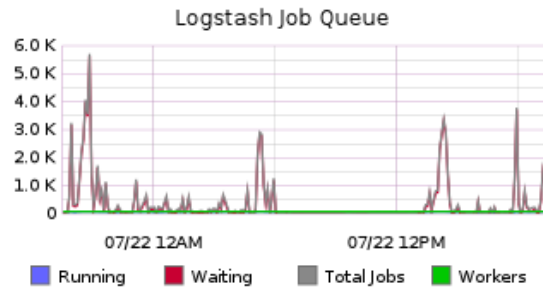
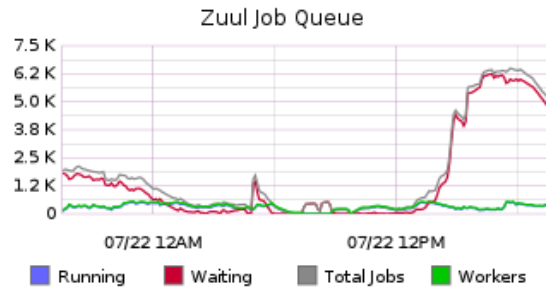
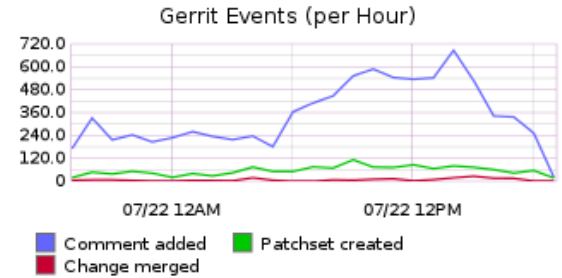
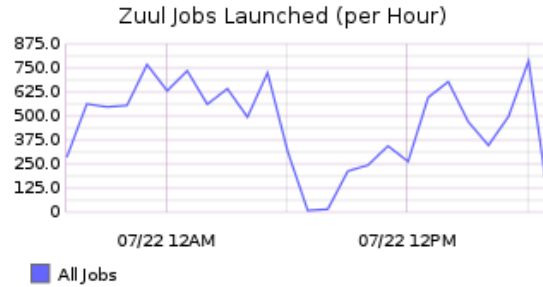
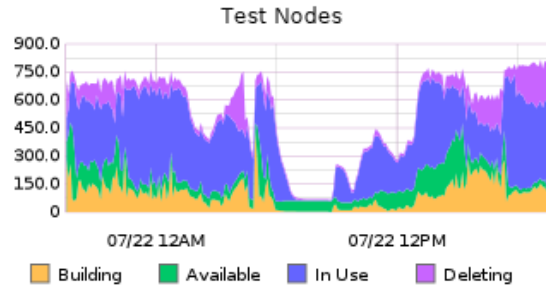
Change queue: **openstack/murano**

openstack/murano 3d4048d unknown 5 hr 45 min

Change queue: **openstack/magnum**

# Statistics

## Job Stats



# System Administration

# Services run by Infra Team

- CI systems
- Cacti
- Elasticsearch, Logstash and Kibana
- IRC Bots
- Etherpad
- Git
- Paste
- Planet
- Puppetboard
- Mailing Lists
- Storyboard
- Various smaller web services
- Wiki

*“We consider every command typed by a router to be a bug. We think this because:*

- we think system administration tasks should be performed by robots*
- we think inputs to the robots should go through code review”*



# Paradigms

- Infrastructure-as-code
- Collaboration
- Peer review
- Automate everything

# Everything Public

- Services available without restrictions
- Anybody can review
- Anybody can contribute
- Discussion on IRC with public archives

# Peer review

- Multiple eyes on changes prior to merging
- Good infrastructure for developing new solutions (particularly for a distributed team)
- No special process to go through for commit access
- Trains us to be collaborative by default

# Checks for CI

- flake8 (pep 8 and pyflakes)
- bashate
- puppet parser validate
- puppet-lint
- beaker-rspec
- XML
- Alphabetized project files
- Parsing of files
- ...

# Automated deployment

- Change gets checked in
  - ...Either puppet master gets updated and applies change
  - ...Or vcsrepo module in puppet pulls in latest version of project

# JJB

- Tool for configuring Jenkins jobs using YAML files
- Using templated jobs, to apply same jobs to tons of different projects:
  - job-template

```
name: 'gate-{name}-docs'  
builders:  
  - shell: 'git checkout {branch_name}'
```
- Grouping of jobs:
  - job-group:

```
name: '{name}-tests'  
jobs:  
  - '{name}-unit-tests'  
  - '{name}-perf-tests'
```

# Nodepool

- Deploy and manage a pool of images on the cloud
- Works with any OpenStack provider
- Once per day a new image is generated with cached content
- Spins instances based on desired image on demand
- Can use disk-image-builder to build images
- Communicates with Zuul using gearman-servers for getting realtime demand

# System administration

- Using puppet and ansible
  - Everything in git
- For debugging and exceptional tasks:
  - Infra-root team has ssh access everywhere, uses ansible for routine tasks
  - ssh access is granted for individuals for specific service



# Configuration Management

- All servers installed using base image and puppet.
- Setting up a new server:
  - Set up new git repository
  - Set up server locally, get it fully puppetized
    - All changes to git repository use gerrit and review
  - Set up dev server in cloud using puppet only
  - Set up production server using puppet only

# Major Manual Task

- Major updates are done collaboratively.  
Examples: Gerrit update, OS update of server
- Tested where possible on dev system
- All instructions prepared on etherpad, reviewed by team

# Limitations

- Sometimes you just need to log into a server
- More difficult for complicated migrations, upgrades
- Passwords need to be privately managed (hiera)

# Reference

# More Information

- OpenStack in general:  
<http://www.openstack.org>
- OpenStack infrastructure:  
<http://ci.openstack.org/>
- OpenStack documentation:  
<http://docs.openstack.org>

# OpenStack and (open)SUSE

- openSUSE wiki:  
<https://en.opensuse.org/Portal:OpenStack>
- Cloud packages:  
<https://build.opensuse.org/project/show?project=Cloud%3AOpenStack>
- Installation Tutorial for openSUSE and SLES at  
<http://docs.openstack.org>  
Newton version at  
<http://docs.openstack.org/newton/install-guide-obs/>
- SUSE OpenStack Cloud product:  
<https://www.suse.com/products/suse-openstack-cloud>

# Credits

- Some content copied from <http://docs.openstack.org/infra/publications>



We adapt. You succeed.