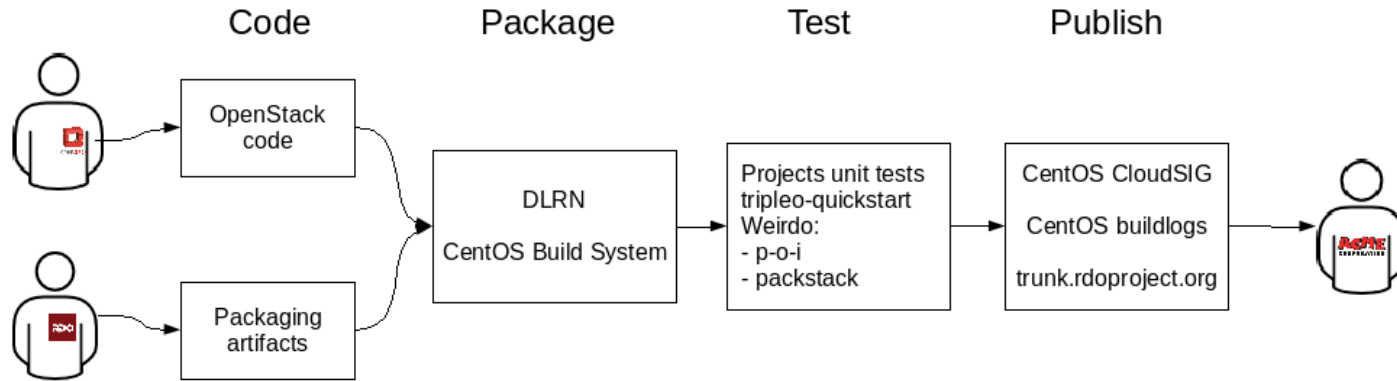




# CI in RDO: What do we test?

Alfredo Moralejo  
amoralej in #rdo  
25-Oct-2016

# The RDO workflow



# Testing while packaging

## While building packages in DLRN or for stable repositories:

- Run project unit tests in building environment via %check
  - Test dependencies provided with RPMs in the same repository or release dependencies one.
- Install just created packages using the same repository (+ release dependencies)

## What value provides:

- Help identify required changes in distgits, typically new dependencies added to test and run the project.
- Special case for master-head DLRN worker:
  - Identify issues in projects with latest commits of libraries and clients (not tested in upstream projects until added to a point release and included in upper-constraints.
- Used as gate job to test changes in packaging in <https://review.rdoproject.org>

# Testing after packaging

## Expected value

- The goal is to validate quality of repos and provide known-good content to users
- Identify issues in deployment tools with new commits in deployed projects.
- Identify run-time issues introduced by recent commits that passed gate devstack-based tests
- For the master-head case, we test projects with newest libraries versions ahead of upstream gate jobs

# Testing after packaging (II)

## Promotion pipelines

- Executed in <http://ci.centos.org/view/rdo> for the different repositories managed in RDO project
  - Test repositories content by testing different RDO deployment scenarios
  - Generate TripleO images.
  - TripleO quickstart (<https://github.com/openstack/tripleo-quickstart>)
    - Minimal deployment
    - Minimal pacemaker deployment
  - Weirdo jobs (<http://weirdo.readthedocs.io/en/latest/how.html>)
    - Three puppet-openstack-integration scenarios
    - Three packstack scenarios
  - Publish created images (on success)

# Testing after packaging (III)

## Promotion pipelines

- For RDO Trunk repos of supported releases (currently liberty, mitaka, newton and master):
  - Jobs rdo-delorean-promote-<release>
    - When jobs pass, the tested repo is promoted to current-passed-ci and images are copied to stable folder.
- For CentOS CloudSIG repos (currently only used in mitaka and newton)
  - Jobs rdo-promote-cloudsig-testing-<release>
    - Execute deployment scenarios on CBS builds in testing stage.
    - Available in [http://buildlogs.centos.org/centos/7/cloud/x86\\_64/openstack-<release>](http://buildlogs.centos.org/centos/7/cloud/x86_64/openstack-<release>)
    - When job passes content is promoted to release stage and synchronized to official RDO CloudSIG repos.
    - Working to automate jobs execution and promotion (still manual).
- Deployment tools specific coverage pipelines for master DLRN:
  - Puppet only pipeline in master (weirdo-promote-puppet) – puppet-passed-ci
  - TripleO only pipeline (rdo-delorean-promote-master-current-tripleo) – current-tripleo-rdo

# Testing RDO tooling

RDO develops its own tools and artifacts:

- DLRN
- rdoinfo
- rdopkg
- weirdo
- distgits

Gating CI with OpenStack upstream tooling

- In review.rdoproject.org
- Based on gerrit + zuul + nodepool + jenkins
  - Using SoftwareFactory (<https://github.com/redhat-cip/software-factory> )

# Improvements on RDO testing

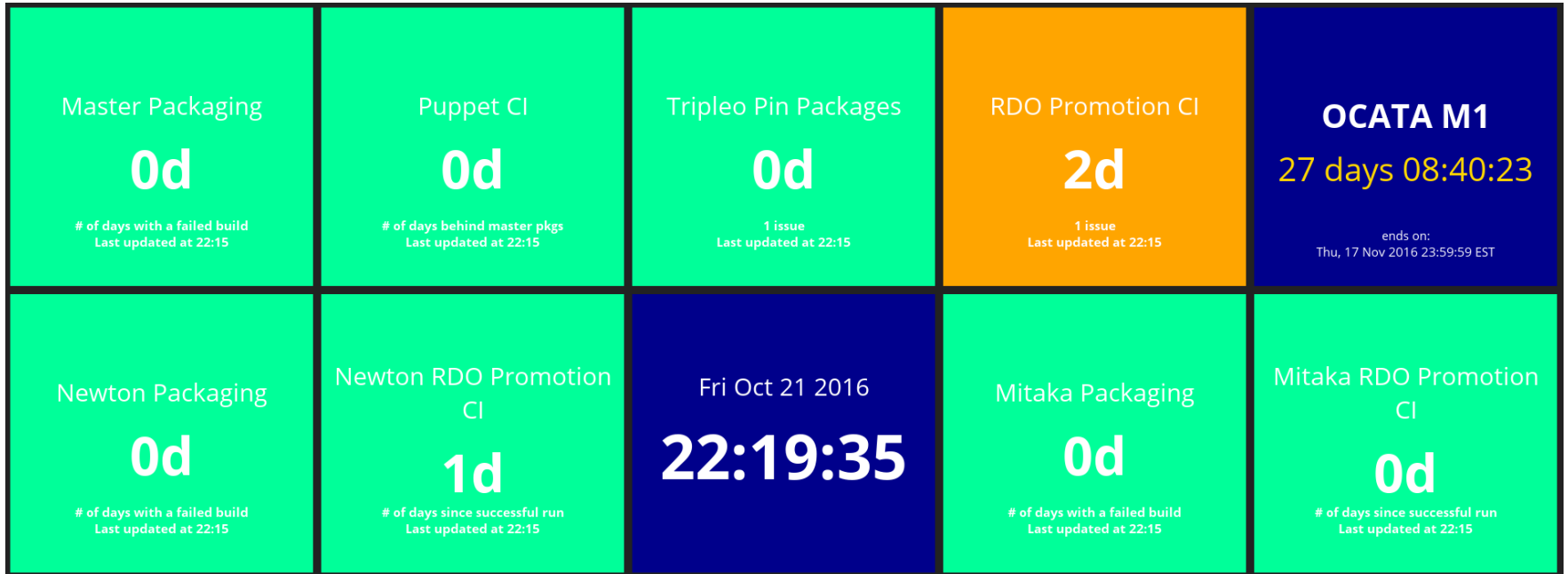
## Upstream gating

- Detecting new requirements or changes in packaging as soon as possible would improve proactivity.
- There are plans to implement Third Party CI on upstream projects to execute dlrn builds on each changeset
  - Non-voting jobs
  - We could introduce modifications in distgits before getting actual “Fail To Build From Source” errors



# Interested in the status of RDO CI?

<https://dashboards.rdoproject.org/rdo-dev>



**THANK YOU !!!**

**BACKUP SLIDES**

# DLRN master-head Vs master

