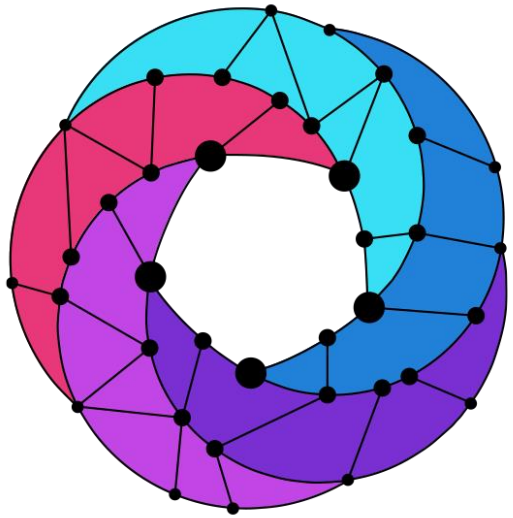


Open Source MANO

OSM RELEASE 1 DEVOPS STATUS AND PLANS

October 2016

Jeremy Mordkoff, RIFT.io

ETSI

TEAM

Open Source MANO

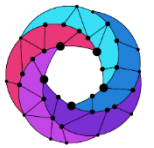
Core

- Carmine Rimi (Mirantis)
- Gerardo García (Telefónica)
- Jeremy Mordkoff (RIFT.io)

With Help from

- Alfonso Tierno Sepulveda (Telefónica)
- Silvia Almagia, Thomas Meredith, Mahdi Ben Alaya (ETSI)
- Adam Israel (Canonical)

SCOPE



Open Source MANO

- Automate the build and test process
  - Use pipelines where possible
- Define deployment options
  - Build from source, install from packages
- Define integration and system test goals
- Define and implement/select test harness
  - Includes configuring all components
  - Does not include implementing the tests themselves

ACCOMPLISHMENTS AND PLANS

Open Source MANO

- R1 accomplishments
  - Jenkins based build system
  - Container management
  - Single entry point for all MDGs for compile and install
  - Gerrit integration
- Planned Post R1 Release Tasks
  - Artifact (deb packages) generation and management
  - Integration test automation
  - System test automation
  - JuJu Charm Installer
- R2
  - In Planning Stage

COMPONENTS

Open Source MANO

- Git – source control
  - one repository per MDG plus devops and descriptors
- Gerrit – code review
- Jenkins – build and test management and tracking
  - Slave processes do the real work, run in VMs
  - Builds and tests are run inside containers inside these VMs
- Deb repo -- build artifacts
  - Multiple instances with various levels of accessibility/stability
  - Latest -> stable -> release
- HIVE – access to external labs (ETSI, TEF, etc)

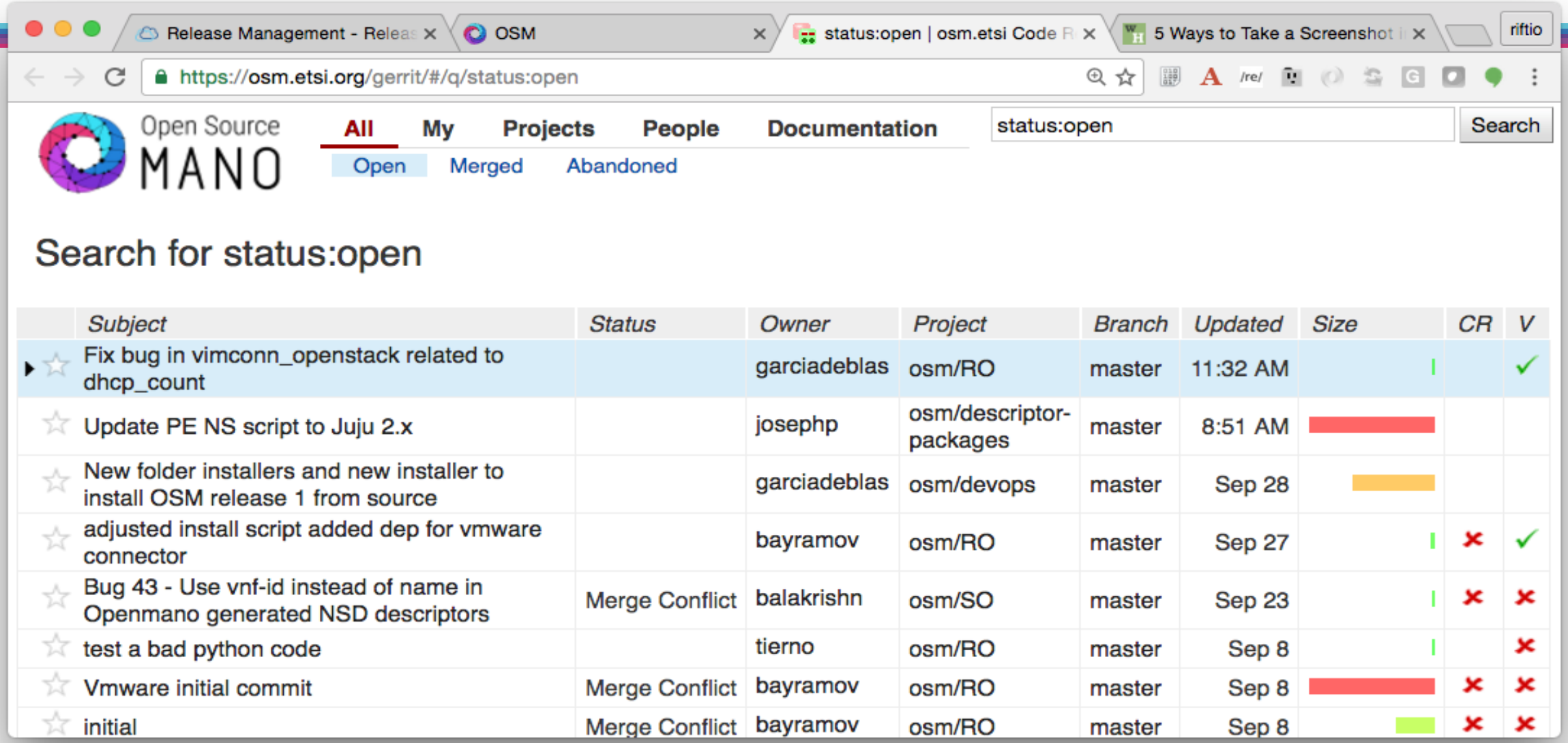
PROCESS FLOWS

Open Source MANO

- Git push to refs/for/master
  - Initiates code review
  - Starts a jenkins job
    - merges this change into the master
    - builds (and someday tests) this code
    - Reports back to Gerrit
- Once approved by MDG lead, code arrives in master
  - Becomes baseline for the next code change
- Integration / System tests (future)
  - Separate Jenkins job
  - Focuses on interactions between components and VIMs

GERRIT

Open Source MANO



Subject	Status	Owner	Project	Branch	Updated	Size	CR	V
Fix bug in vimconn_openstack related to dhcp_count		garciadeblas	osm/RO	master	11:32 AM			✓
Update PE NS script to Juju 2.x		josephp	osm/descriptor-packages	master	8:51 AM			
New folder installers and new installer to install OSM release 1 from source		garciadeblas	osm/devops	master	Sep 28			
adjusted install script added dep for vmware connector		bayramov	osm/RO	master	Sep 27		✗	✓
Bug 43 - Use vnf-id instead of name in Openmano generated NSD descriptors	Merge Conflict	balakrishn	osm/SO	master	Sep 23		✗	✗
test a bad python code		tierno	osm/RO	master	Sep 8			✗
Vmware initial commit	Merge Conflict	bayramov	osm/RO	master	Sep 8		✗	✗
initial	Merge Conflict	bayramov	osm/RO	master	Sep 8		✗	✗

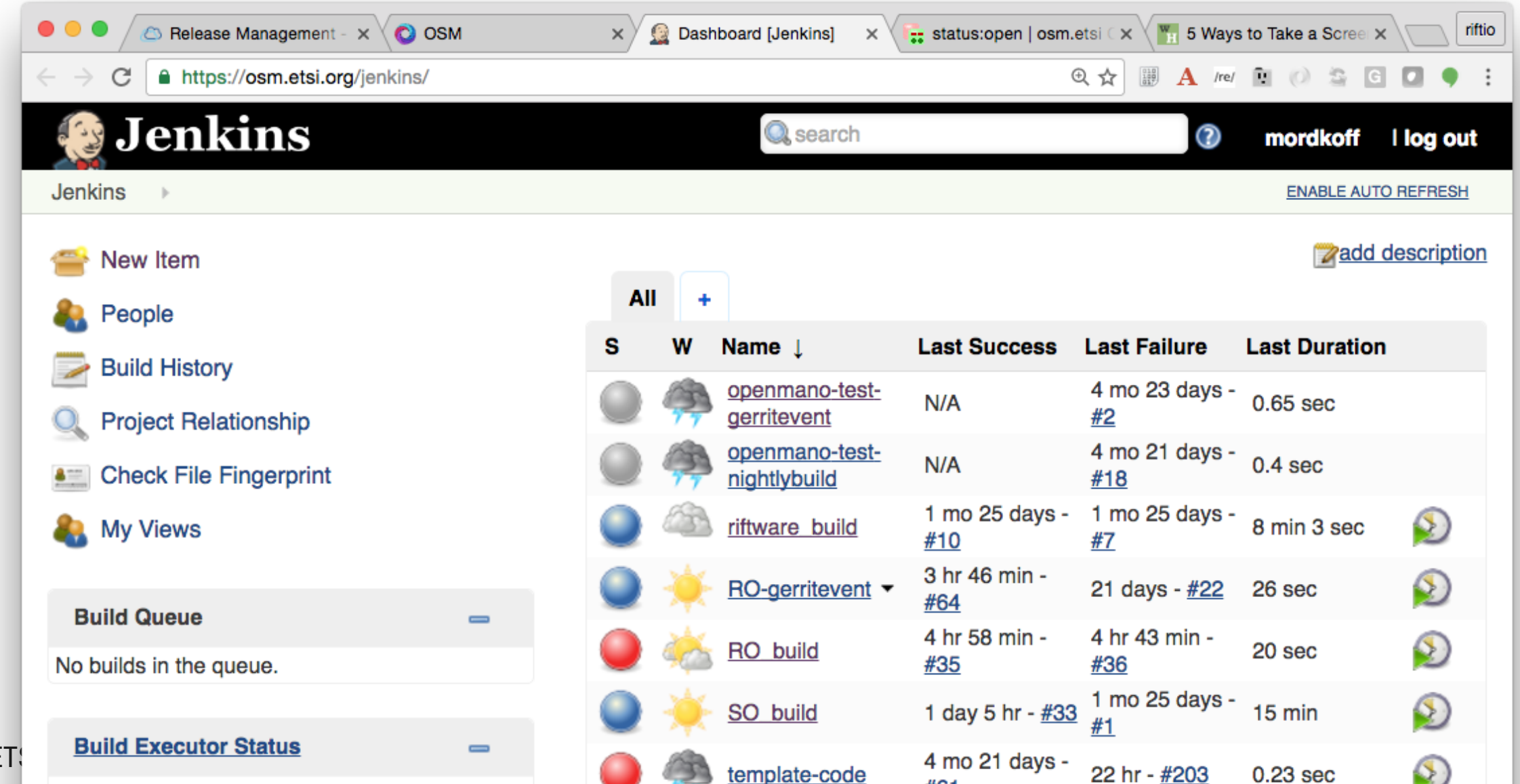
JENKINS BUILDS

Open Source MANO

- Triggered manually, by Gerrit or by a commit
- Create (or update) build container using LXC inside a VM
- Gerrit builds just report success or failure
  - Used as an input to the code review process
  - Build and unittest
- Commits to master cause a full build
  - Build, unittest, and package
  - Artifacts pushed to ”latest” debian repo

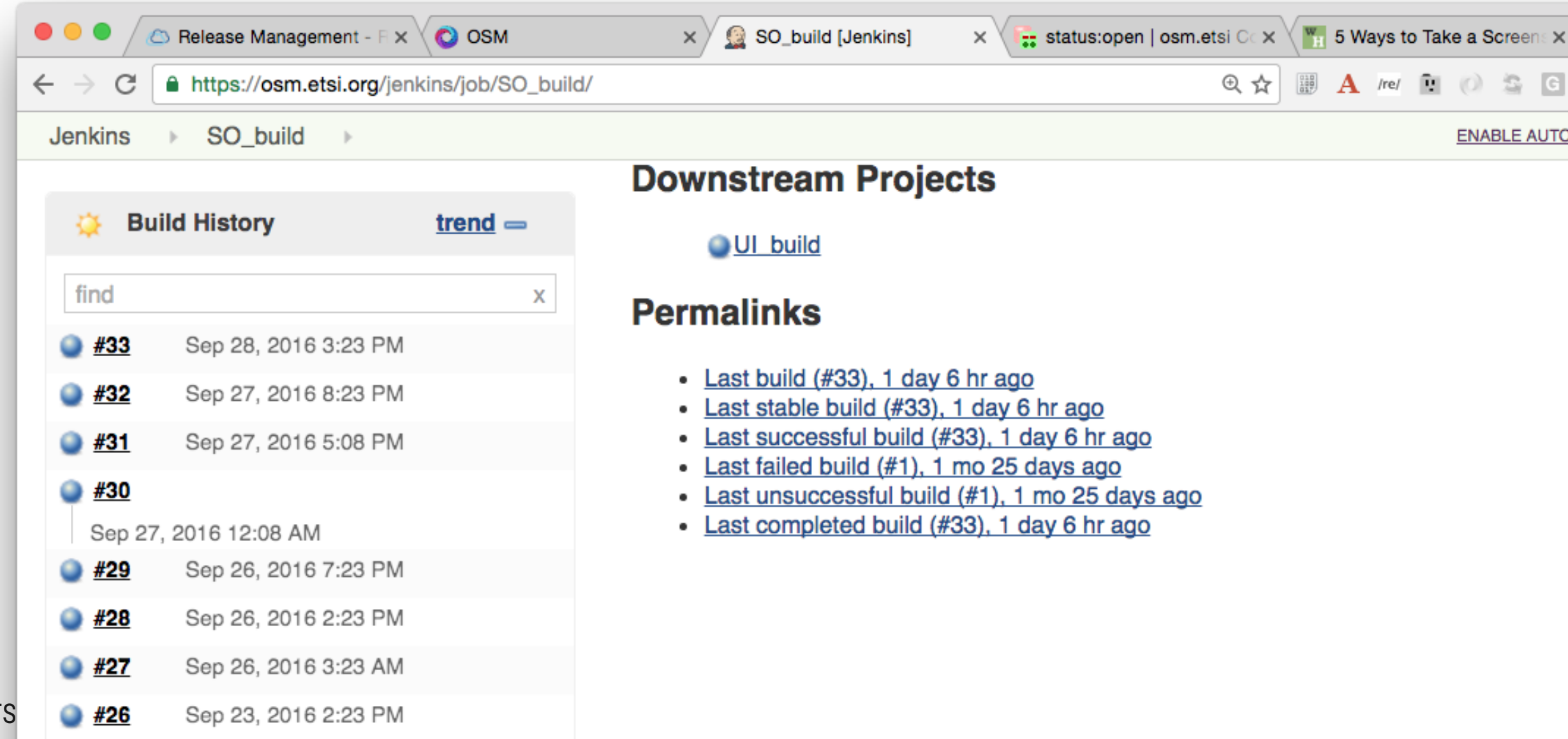
JENKINS

Open Source MANO



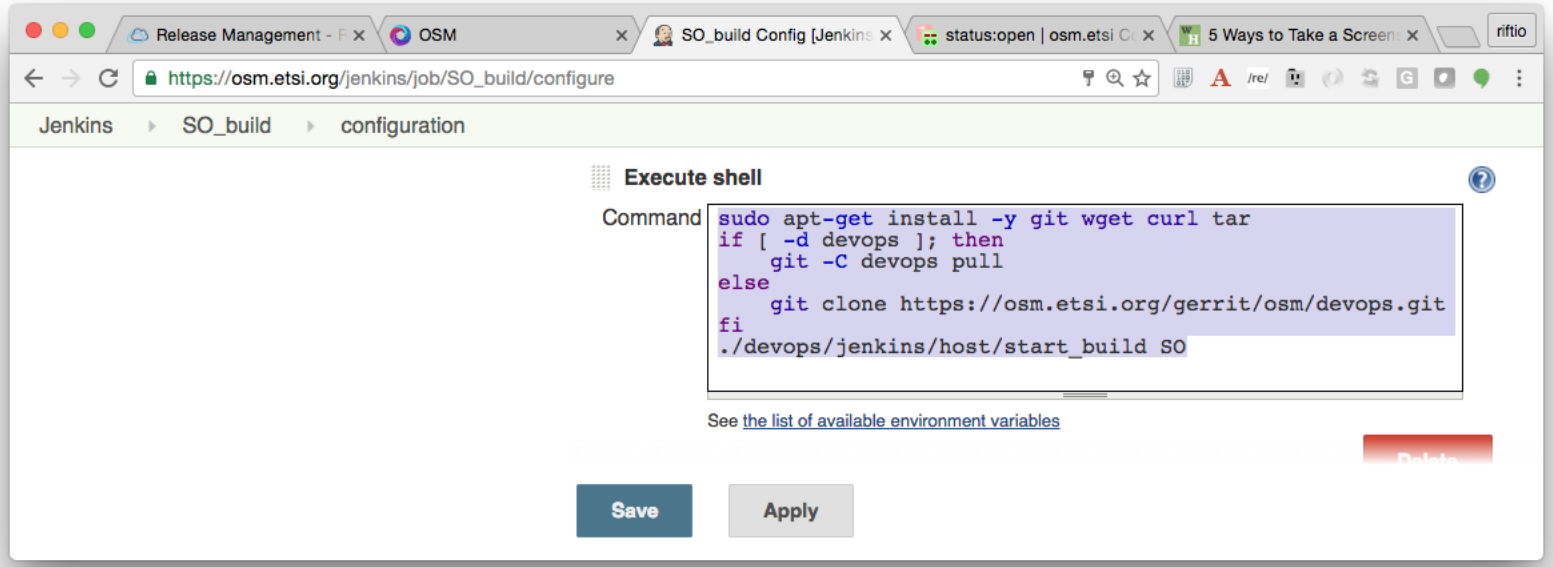
JENKINS

Open Source MANO



JENKINS

Open Source MANO



DEBIAN REPOSITORIES

Open Source MANO

- Latest
  - holds the artifacts from any build. Unit tested only.
  - Updated by jenkins builds
  - Accessible to OSM members
- Stable
  - integration testing using fake vim completed.
  - Jenkins promotes artifacts from latest to stable integration tests pass
  - Accessible to anyone but upgrade / downgrade not guaranteed (e.g.)
- Released
  - The latest public release
  - Enables deploying the solution without the need to build from source
  - Group decision when to promote from stable to released

AUTOMATED INTEGRATION TEST

Open Source MANO

- Managed by Jenkins
- Used to test that the latest change in one component does not break interoperability with all others
- Use the “stable” versions except component under test
- Install the “latest” version of the component-under-test
- If test passes, promote ”latest” to “stable”
- Exception (to be avoided)
  - If a change in one MDG cannot be made backwards compatible with all other MDGs, then the latest components can be tested and promoted together.

EXAMPLE

Open Source MANO

- Stable RO
- Stable SO
- Latest UI
- If integration tests pass, Latest UI becomes Stable UI

SYSTEM TESTS

Open Source MANO

- Focus is on interoperability with VIMs and descriptors
- Uses descriptor packages from descriptor Git repo
- Verifies that all can be on boarded and instantiated on one or more VIMs
- Matrix may be large
  - Spare matrix testing – hit every package and every VIM at least once, but not every combination of package and VIM on each run. Vary the combinations each run so that over time all combinations are covered.
- Can also verify that new packages are compatible
  - One package against all VIMs

STATUS AND FUTURE WORK

Open Source MANO

- Jenkins Builds – done and working
- Debian Respositories -- in progress
  - interface TBD. Investigating management tools
- Integration test – in progress
  - Manual today
  - Building all components from source, should install from artifacts
  - automation next
- Installation from Artifacts -- TBD
- System test -- TBD

HOW TO HELP

Open Source MANO

- Skills needed
  - BASH, Python3
  - Debian repositories (signtool, etc)
  - Git
- Tasks
  - Tools to push artifacts to debian repo
    - BASH, signtool, etc
    - Investigate repository management tools like Aptly
  - Install using Artifacts
    - BASH, deb, apt
  - Integration and System test automation
    - JuJu and/or BASH
    - Rest/Curl/Python3

Open Source
MANO

THANKS

ETSI

Open Source
MANO

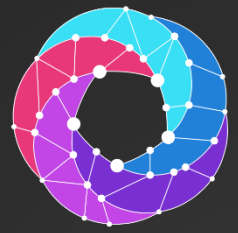
Open Source
MANO

Open Source
MANO

Open Source
MANO

Open Source
MANO

Open Source
MANO



Open Source
MANO

Open Source
MANO

Open Source
MANO

Open Source
MANO

Open Source
MANO

Open Source
MANO

Open Source
MANO