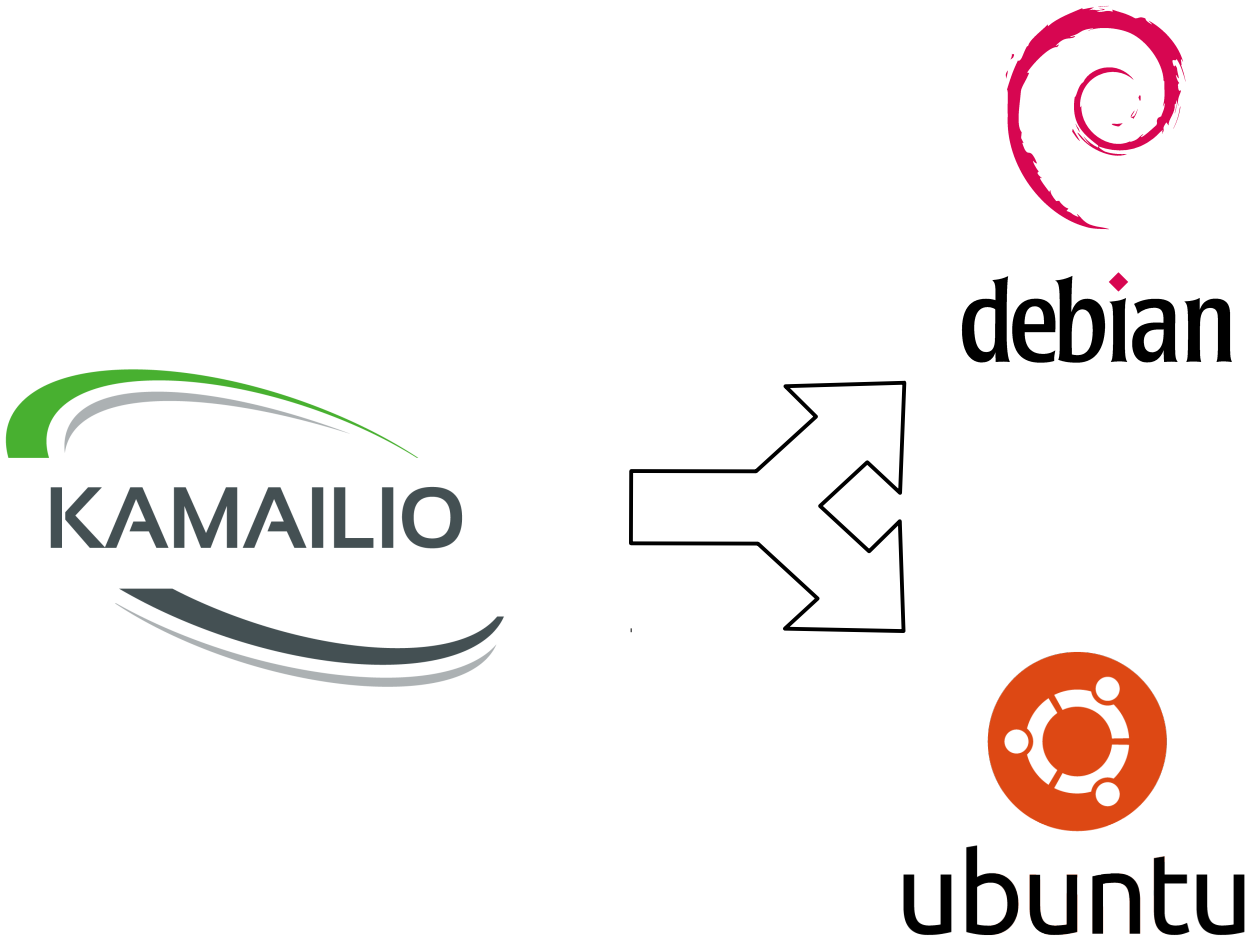
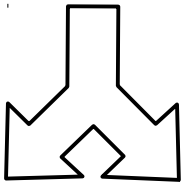
The new Kamailio Build System



Andreas Granig
<agranig@sipwise.com>

deb.kamailio.org

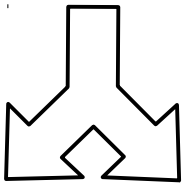


stable

4.1, 4.0, 3.3

nightly

master, 4.1, 4.0, 3.3

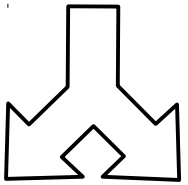


debian

squeeze, wheezy, jessie

ubuntu

Precise



amd64

i386

How did it work in the past?

- Bootstrap target chroots once
- For each target, manually do:
  - `pkg/kamailio/deb/$target` → `debian/`
  - `dpkg-buildpackage -S`
  - source packages → chroot
  - `dpkg-buildpackage`
  - `reprepro`
- For each new release, resulting in
  - Non-reproducable builds
  - Potentially broken dependencies
  - Still, better than nothing!

Continuous Integration to the rescue

- Jenkins CI
- Jenkins-debian-glue

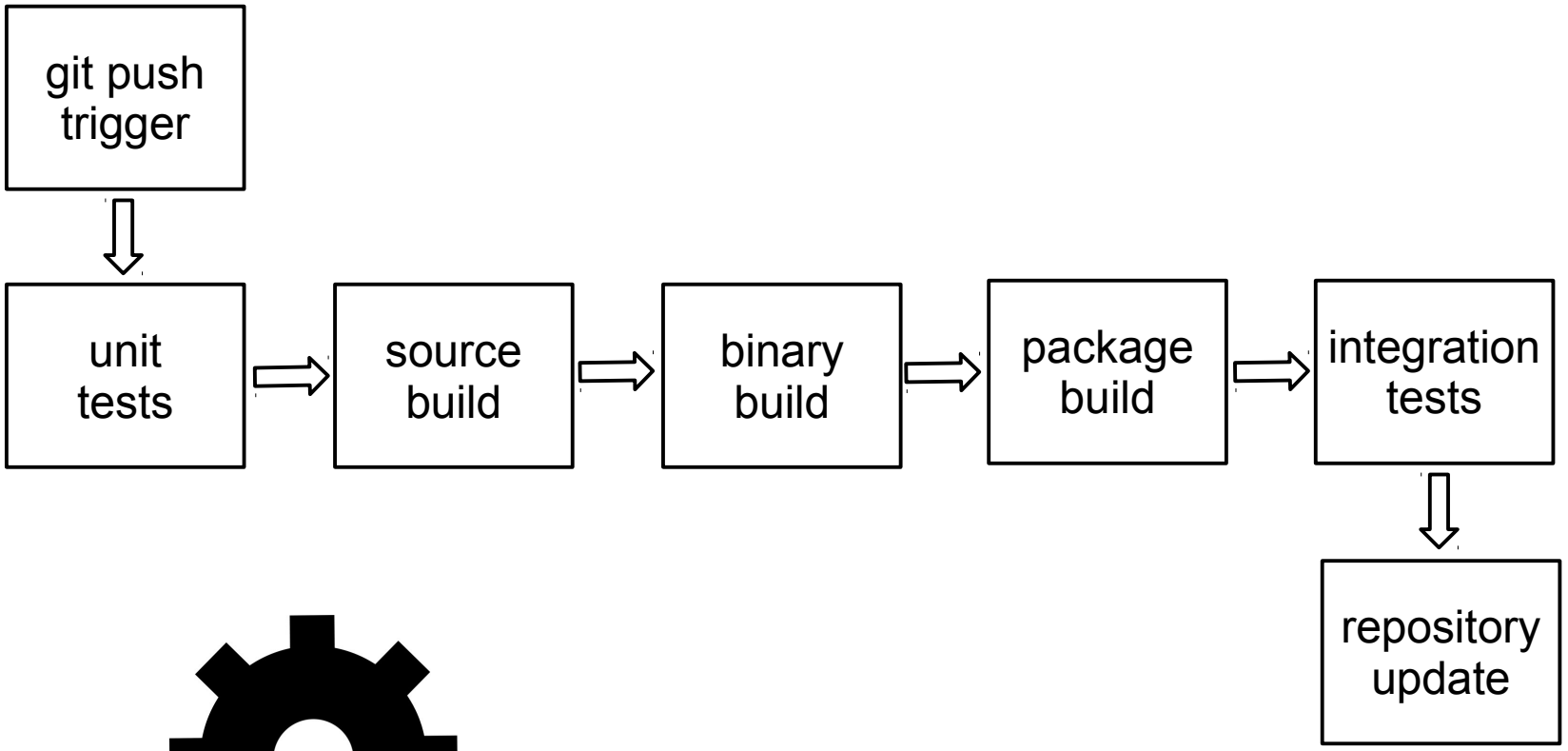
Pardon, what?

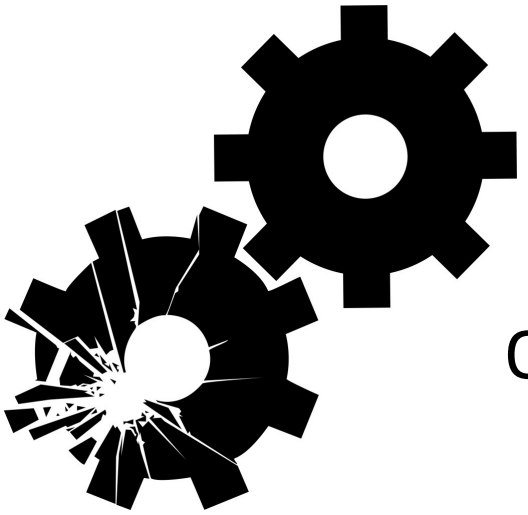
A Primer to Continuous Integration

A brief intro to Jenkins CI

- An application that monitors executions of repeated jobs
- Specifically for
  - Building and testing of software projects
  - Monitoring executions of external jobs

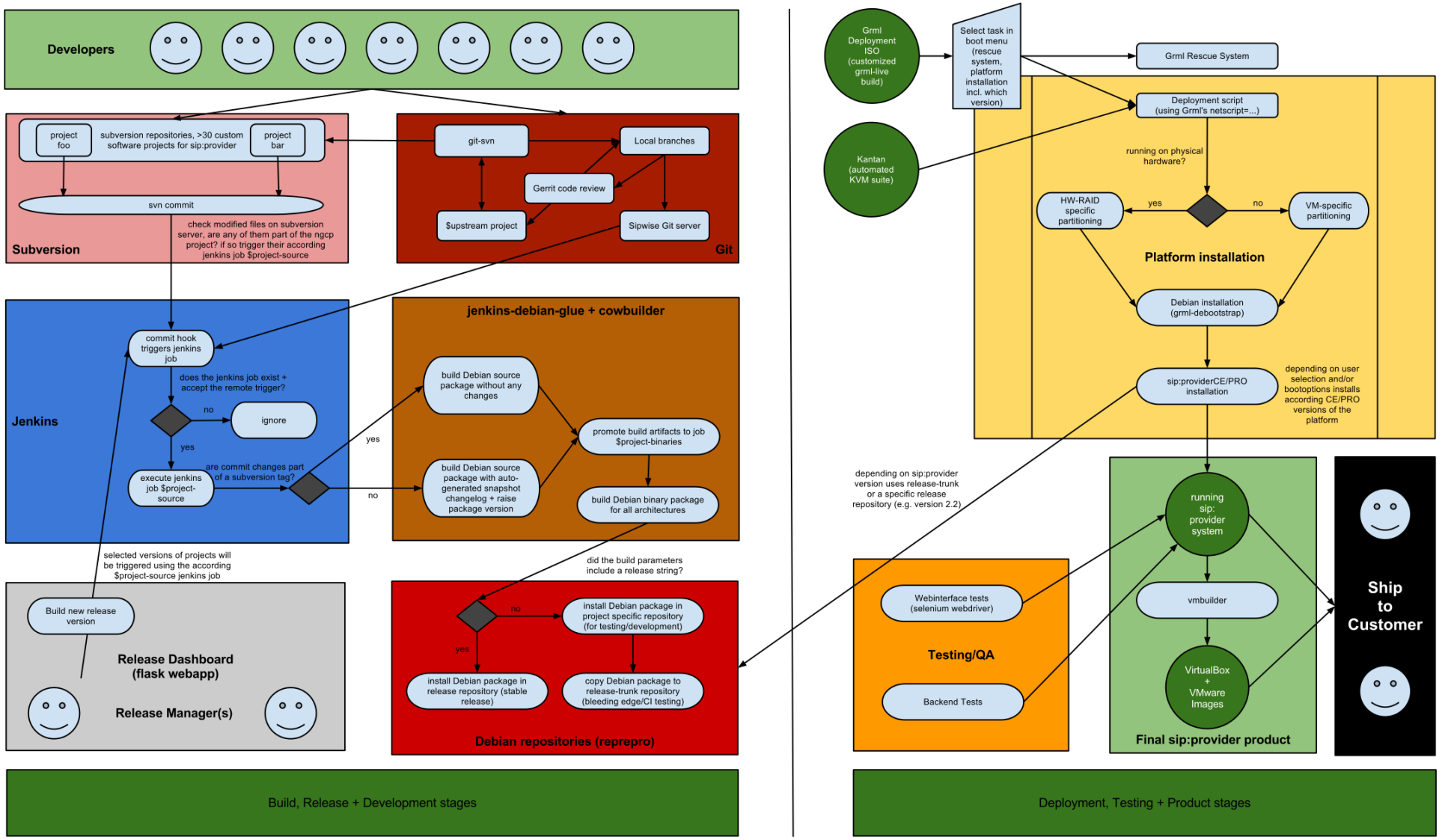
How Jenkins CI works (in theory)





Catch errors early and often!

Deployment Pipeline at Sipwise.com



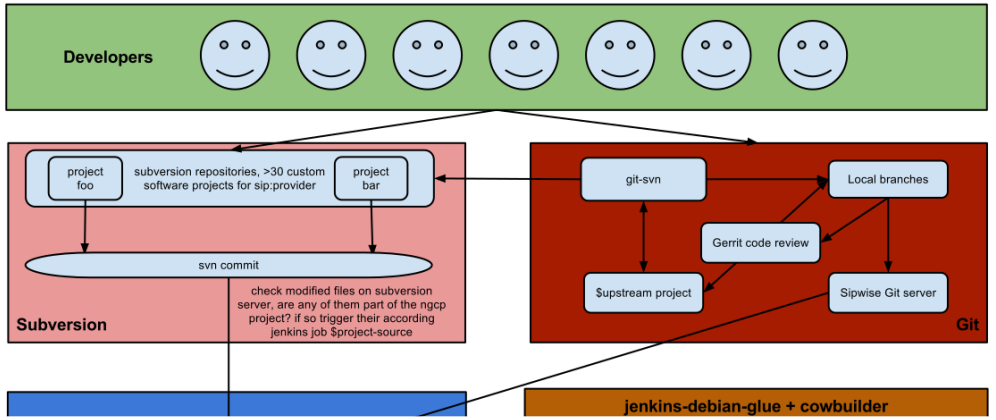
Legend:

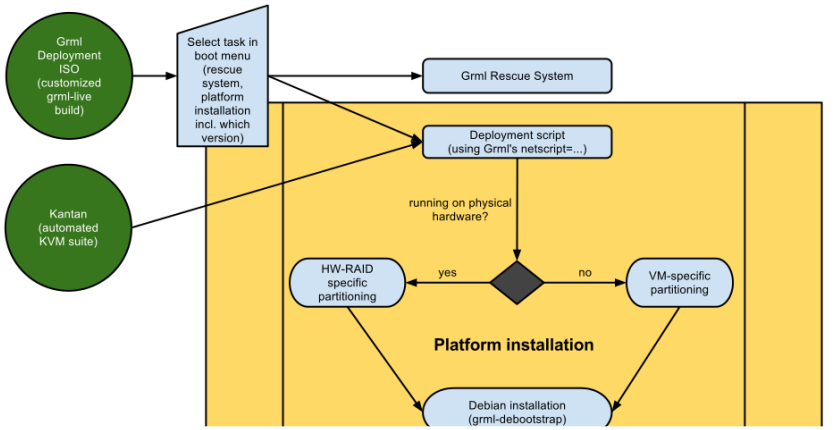
Decision | Action | Software Product | People | Major specific part solution in Development pipeline | User selection | Project

Software:

Cowbuilder: http://packages.debian.org/sid/cowbuilder
Debian: http://debian.org
Flask: http://flask.pocoo.org
Git: http://git-scm.com
Grml: http://grml.org
Jenkins: http://jenkins-ci.org
Jenkins-debian-glue: https://github.com/mika/jenkins-debian-glue
Kantan: https://github.com/mika/kantan
Reprepro: http://mirrorer.alioth.debian.org
Selenium: http://seleniumhq.org
Selenium-webdriver + rspec: https://github.com/mika/sip-provider-selenium-webdriver-tests
Subversion: http://subversion.tigris.org

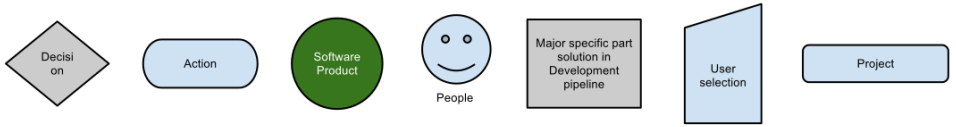
Deployment Pipeline at Sipwise.com





Woah, hold on for a second!

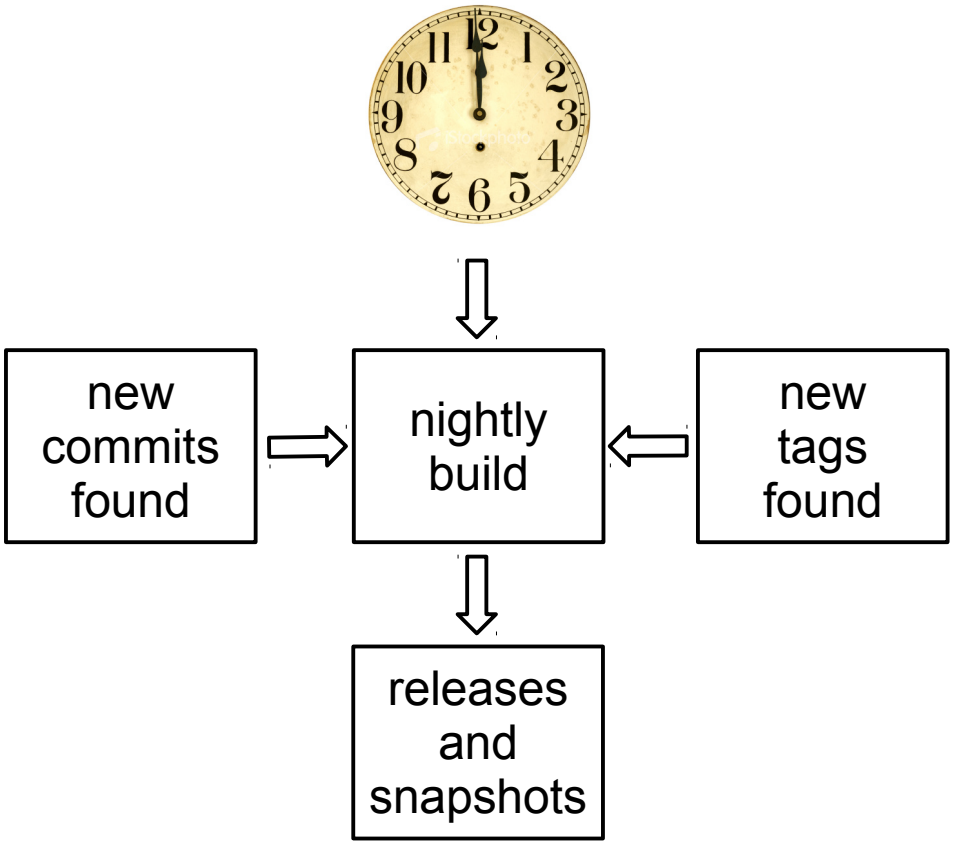
Legend:



Software:

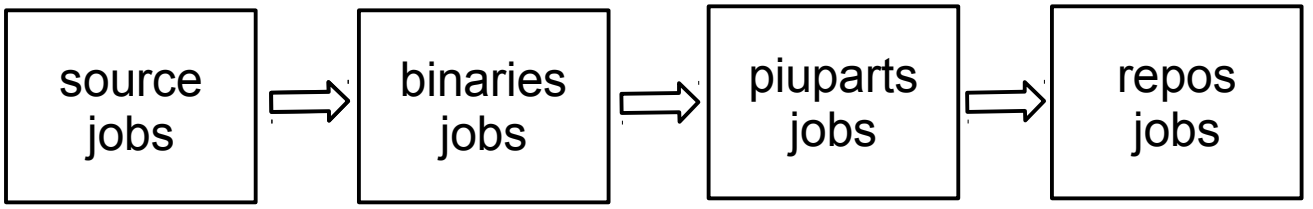
Cowbuilder: http://packages.debian.org/sid/cowbuilder
Debian: http://debian.org
Flask: http://flask.pocoo.org
Git: http://git-scm.com
Grml: http://grml.org
Jenkins: http://jenkins-ci.org
Jenkins-debian-glue: https://github.com/mika/jenkins-debian-glue
Kantan: https://github.com/mika/kantan
Reprepro: http://mirrorer.alioth.debian.org
Selenium: http://seleniumhq.org
Selenium-webdriver + rspec: https://github.com/mika/sip-provider-selenium-webdriver-tests
Subversion: http://subversion.tigris.org

How it works for Kamailio today



The Build Architecture

- Currently four types of jobs

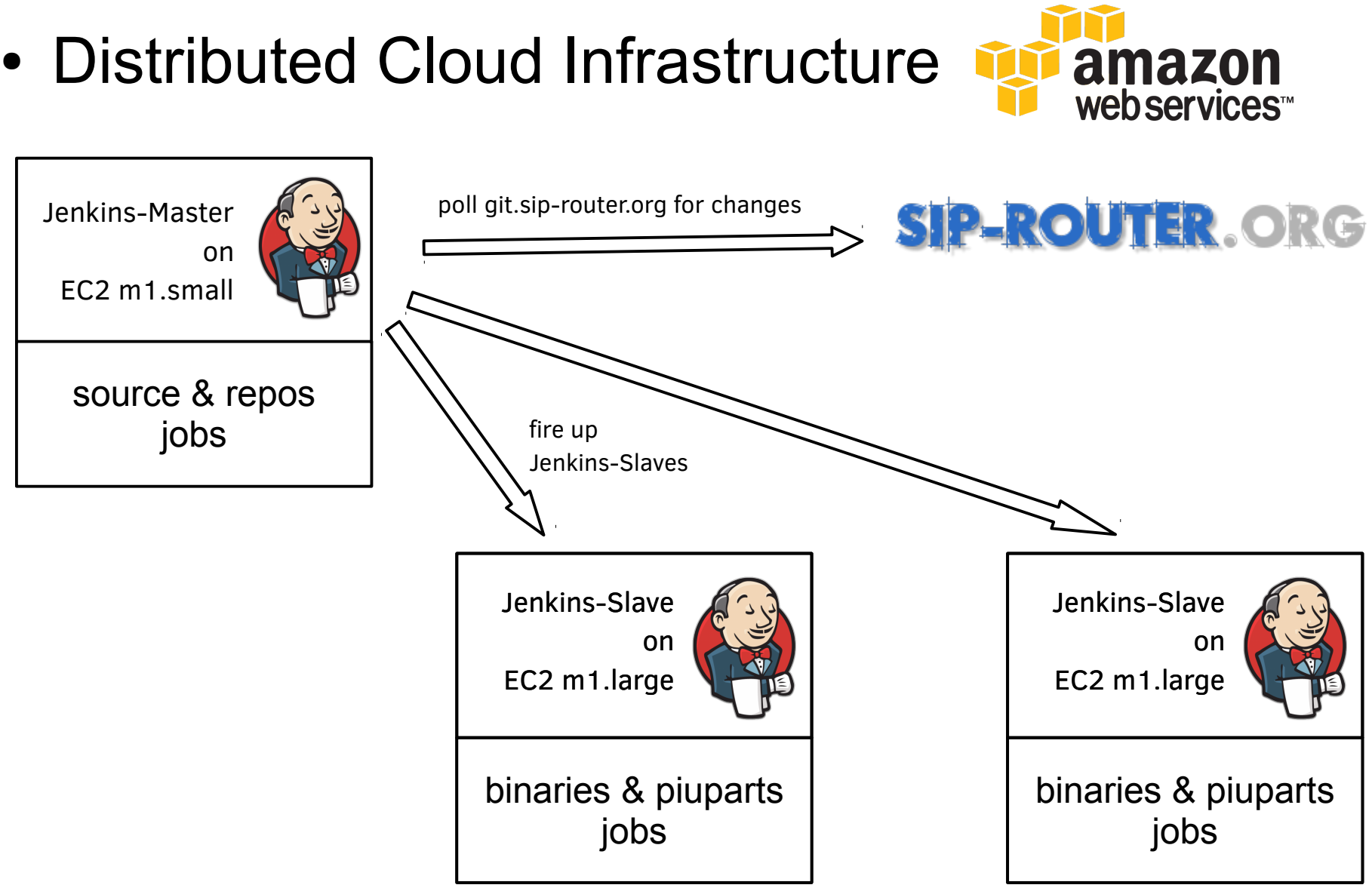


		kamailio41-nightly-binaries	3 days 7 hr - #30	1 mo 4 days - #14	2 hr 36 min
		kamailio41-nightly-piuparts	1 mo 4 days - #7	1 mo 5 days - #4	14 min
		kamailio41-nightly-repos	3 days 5 hr - #22	7 days 9 hr - #21	1 min 32 sec
		kamailio41-nightly-source	3 days 7 hr - #26	N/A	10 min

~3h overall build time per release for all targets and architectures
(i386 and amd64 for Debian squeezy, wheezy and jessie and Ubuntu Precise)

The Build Architecture

- Distributed Cloud Infrastructure



You can replicate the setup

- 100% open source
- General Information on Michael Prokop's blog

  `http://michael-prokop.at/blog/2014/03/25/building-debianubuntu-packages-on-ec2/`

- Documented in our repo at

  `https://github.com/sipwise/kamailio-deb-jenkins`

What's next?

- Build per push only makes sense with testing
- Improve overall test coverage
- Introduce lint/static tests
- Introduce long-term tests

  (mem leaks, performance degrations)
- Improve and automate system integration tests (black-box tests of module functions)

What about Code Review?

- Do we want to use Gerrit?
  - Everybody pushes branches to Gerrit
  - Jenkins signs it off if tests are ok
  - Core Devs review, iterate and approve → automatic merge
  - Feedback and code in one place

How you can help?

- Let's start a discussion about what makes sense
- How to motivate ourselves to write tests?
- Anyone with experience on anything mentioned here willing to share experience?
  - Code tests/analysis
  - Gerrit use cases

Questions?

<agranig@sipwise.com>