
CERN

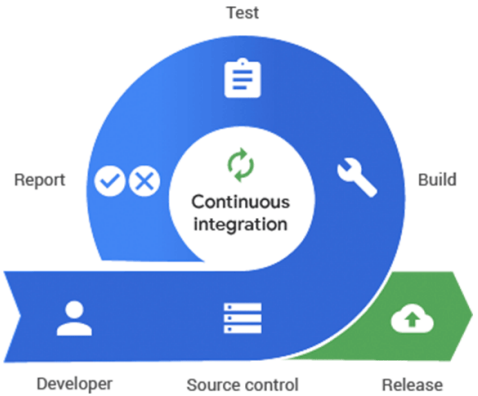
Full OS image automation

Continuous Integration for CERN's distros

2021-10-25

Daniel Juárez González
On behalf of CERN's Linux Support team

HEPiX Autumn 2021



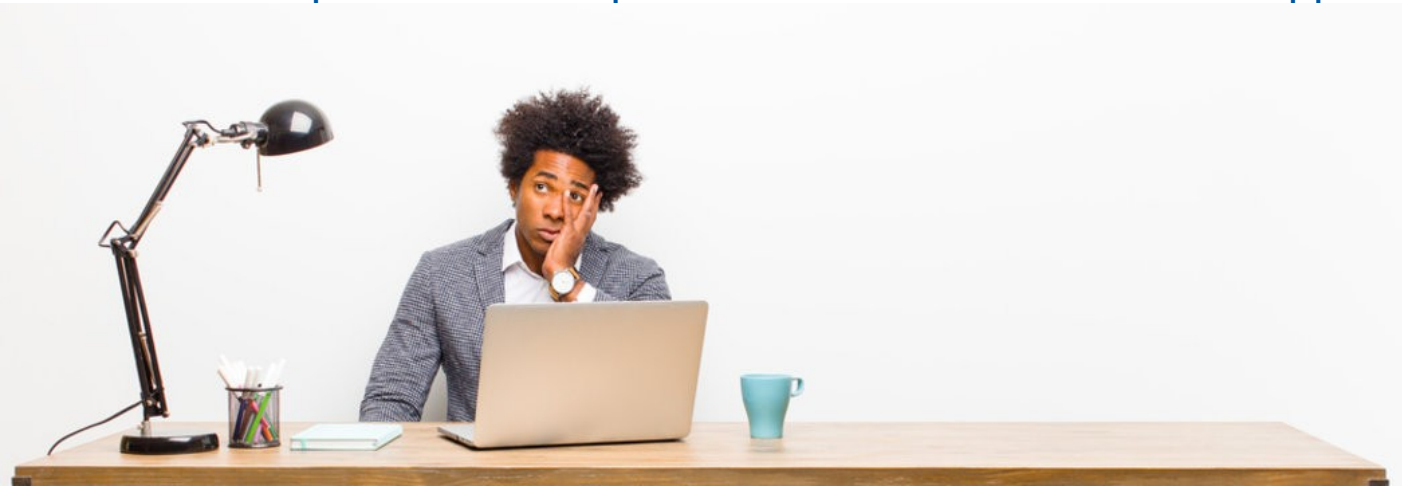
What is this talk about?

- We will focus on the OS image automation
  - Cloud images
  - Docker images
- Automated testing and publishing
  - For managed machines. They use Puppet for configuration
  - For unmanaged machines

- For OS release management @ CERN please refer to Navigating the Stream: automating CentOS releases at CERN

Background

- The year is 2019
- Linux Support has a weekly rotaer that takes care of running manual scripts for package updates and image building
- No image testing is performed on a regular basis
- Tasks are repetitive and require to be done for each of our supported distros



Not scalable: a recipe for trouble

- Problems for new versions are not easily detected until we start receiving tickets
- Support windows for CERN distros overlap and given the increase of them these tasks are more and more time consuming

So what did we do then?

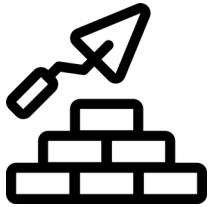
- Slowly moved to CI/CD tasks
- Added automated testing, both upstream and CERN specific
- We adapted our CI so it is as much distro independent as possible to ease adding future ones

Some of the tools we are using

Given we will reference these tools from now on, it is worth spending a minute on them:

- Koji: https://docs.pagure.org/koji/
  - RPM and image building system
- Openstack Nova (VMs), Glance (cloud images), Magnum (container orchestration), Ironic (bare metal provisioning)
- GitLab, GitLab runners, GitLab CI/CD capabilities
- A healthy dose of Python and Bash scripts sticking everything together

Let’s start with the testing



All our automation is based on the same sets of tests:

- https://gitlab.cern.ch/linuxsupport/testing/centos_functional_tests
  - Mirrored from upstream CentOS Automated QA process
  - Basic functional testing
- https://gitlab.cern.ch/linuxsupport/testing/cern_centos_functional_tests/
  - Tests specific to CERN. We welcome contributions
  - So far AFS, NTP, IPv6, Kerberos and partitioning tests among others
- https://gitlab.cern.ch/ai-config-team/cern_puppet_centos_functional_tests
  - Tests only relevant for Puppet-managed machines
  - Cloud-init bootstrapping, Puppet runs, IPv6, Auth setup among others

Foundations of our image validation, now let’s move on

Image life cycle: Docker images

The current selection of built images:

- CentOS Linux 8 x86_64 and aarch64
- CentOS Stream 8 x86_64 and aarch64
- CERN CentOS 7 x86_64

We have both x86_64 and aarch64 **GitLab runners** for our testing

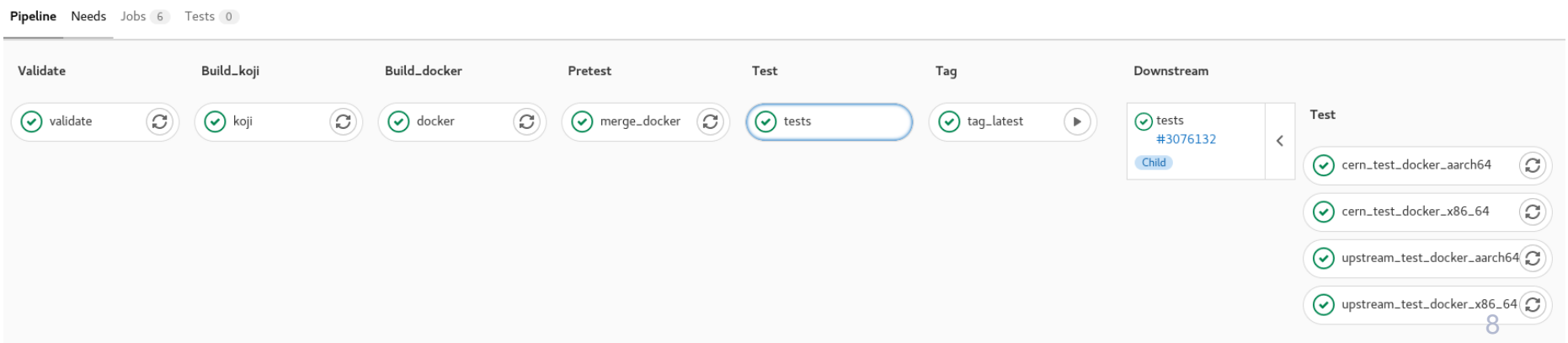


Image life cycle: Docker images

Once a month:

- Validate the Kickstart file to use
- We build each image using Koji image building capabilities and create Docker images
- We then take the resulting tarball and push it to our internal registry as a test image
- We run then our set of tests, if they pass it is autopromoted to the *:latest* tag and pushed to our Dockerhub organisation

Easy peasy!

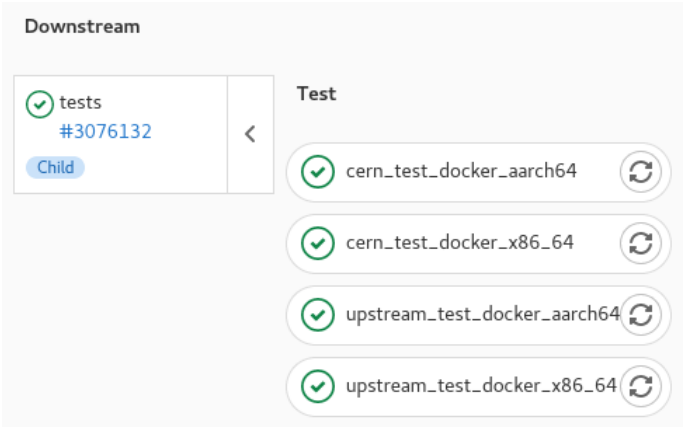


Image life cycle: Cloud images

Once a month:

- Make use of our Koji Building System to build disk images
- Each and every one of our images is built using a Kickstart file

The current selection of built images:

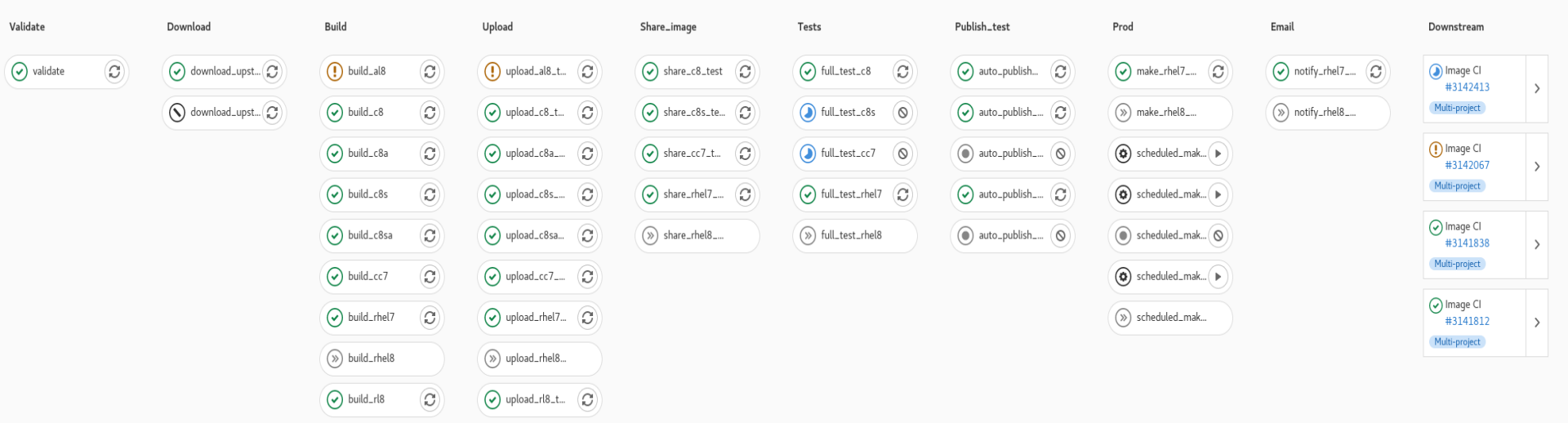
- CentOS Linux 8 x86_64 and aarch64
- CentOS Stream 8 x86_64 and aarch64
- CERN CentOS 7 x86_64
- Rocky Linux 8 x86_64
- Alma Linux 8 x86_64

} Technical previews, for internal test only

We have both x86_64 and aarch64 **Koji builders**
We only support and publish CentOS images (**plus RH7&8**)
Our images are compatible with both BIOS and UEFI boot modes

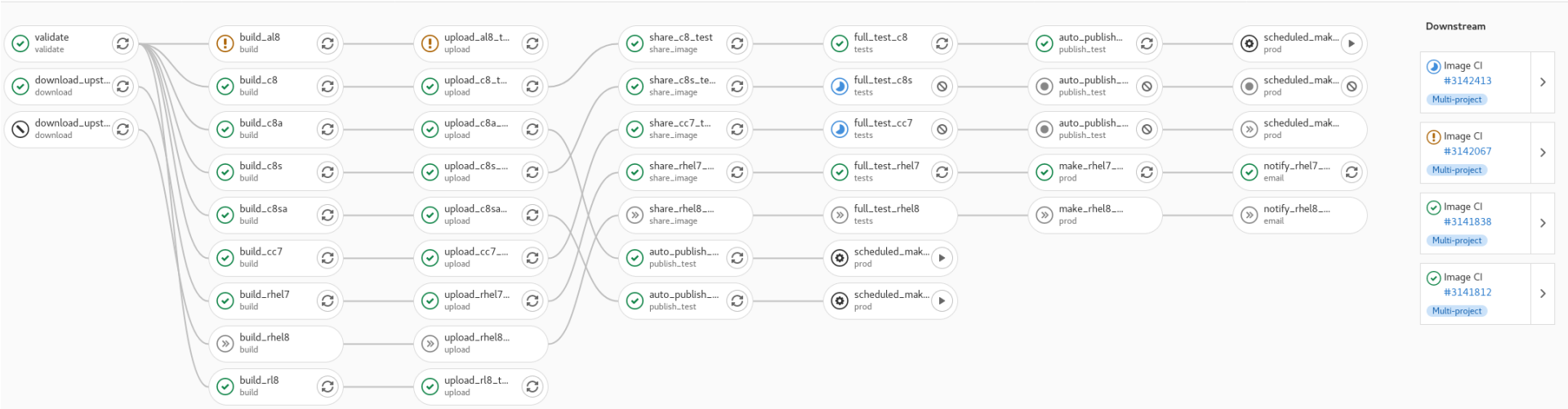
Complete cloud image pipeline

- This is where things get more complex, as bootstrapping is involved



Complete cloud image pipeline

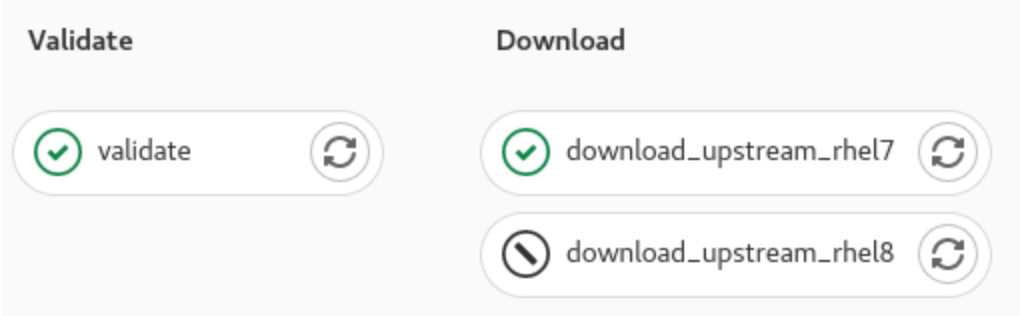
- Here there is a representation of the job dependencies



- Basically each distro follows its own workflow independently

Part 0: Preparation steps

- We validate the Kickstart files

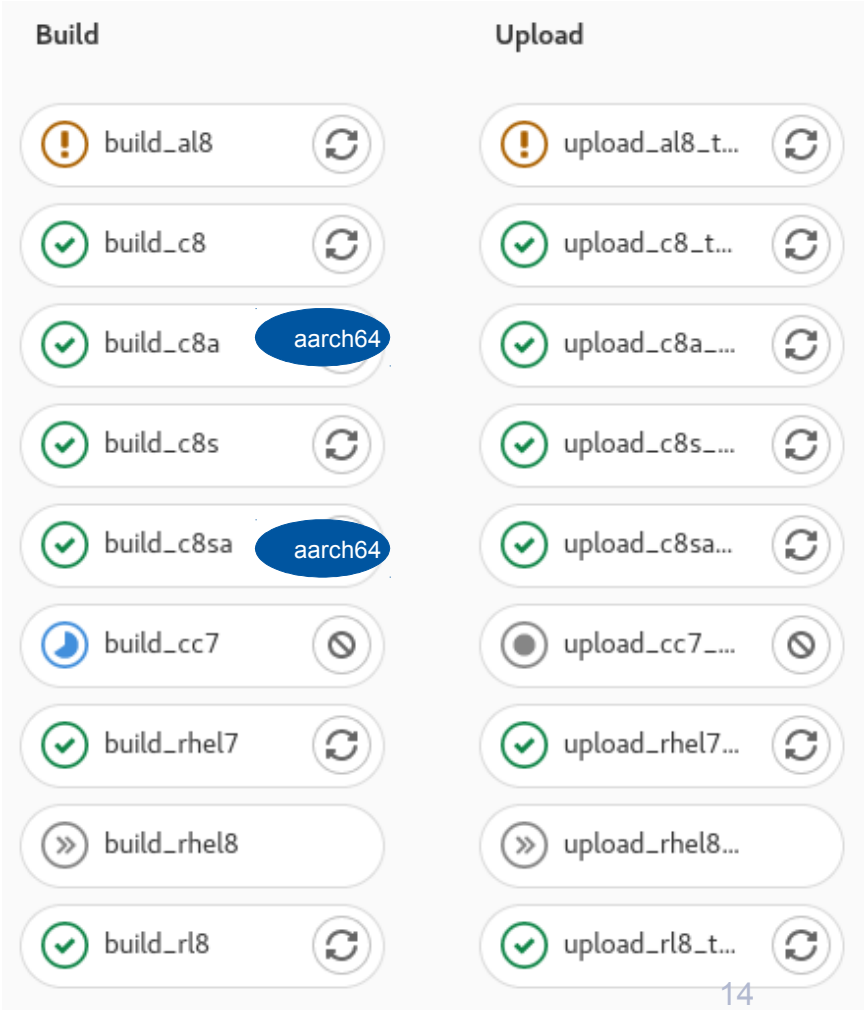


- CentOS, Rocky, Alma images are publicly downloadable, RHEL ones are not. We use our CERN subscription to download them if we detect there is a new update we have not yet published to our community

- We do nothing if we already have the latest RHEL 7 and 8 updates

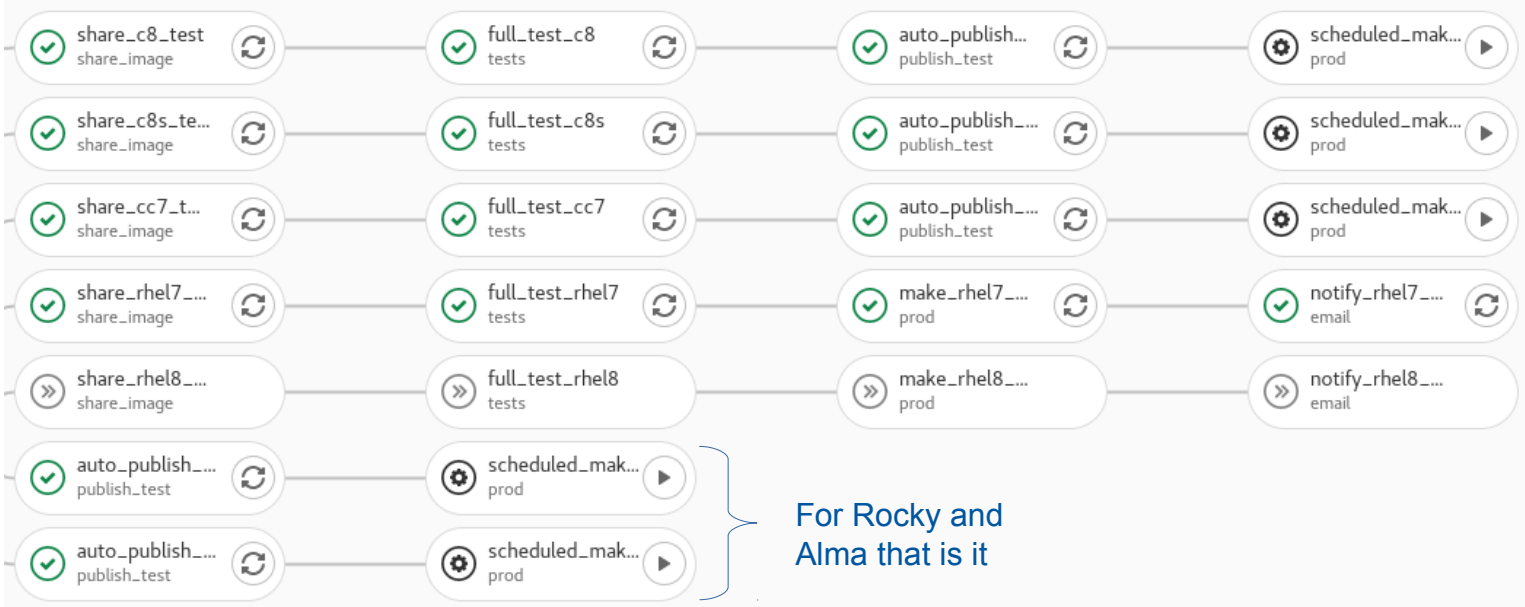
Part 1: Building

- For all CentOS (or Enterprise Linux Clones, ELC from now on) we ask Koji to build each image using our internal installation trees, i.e. our mirrors.
- For RHEL images we have to tweak what we just downloaded using libguestfs with necessary fixes for use at CERN, it is not a build per se.
- Once built, we upload it to our testing Openstack projects where we will create our test machines.



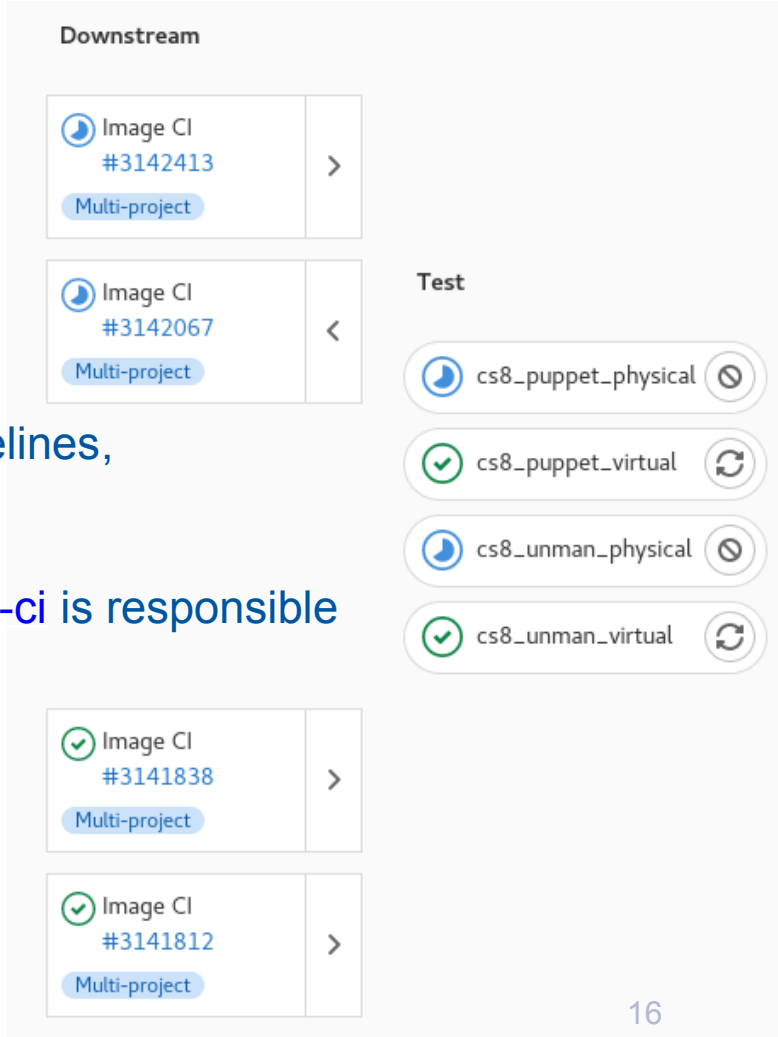
Part 2: Testing and sharing

- Once built and uploaded to Openstack we run our tests (see next slides)
- If everything goes well, we publish them for everyone as TEST images
- We then manually promote to production



Part 2.1: Testing

- Each of our images triggers jobs for each of our cloud use cases
- We use extensively GitLab CI Multi-project pipelines, GitLab CI Pipeline triggers and CI templates
- https://gitlab.cern.ch/linuxsupport/testing/image-ci is responsible for storing all the CI magic
- Our *image-ci* can not only be triggered from our image building, it can also be triggered manually from any other tool that requires testing the bootstrapping process (team collab)



Benefits

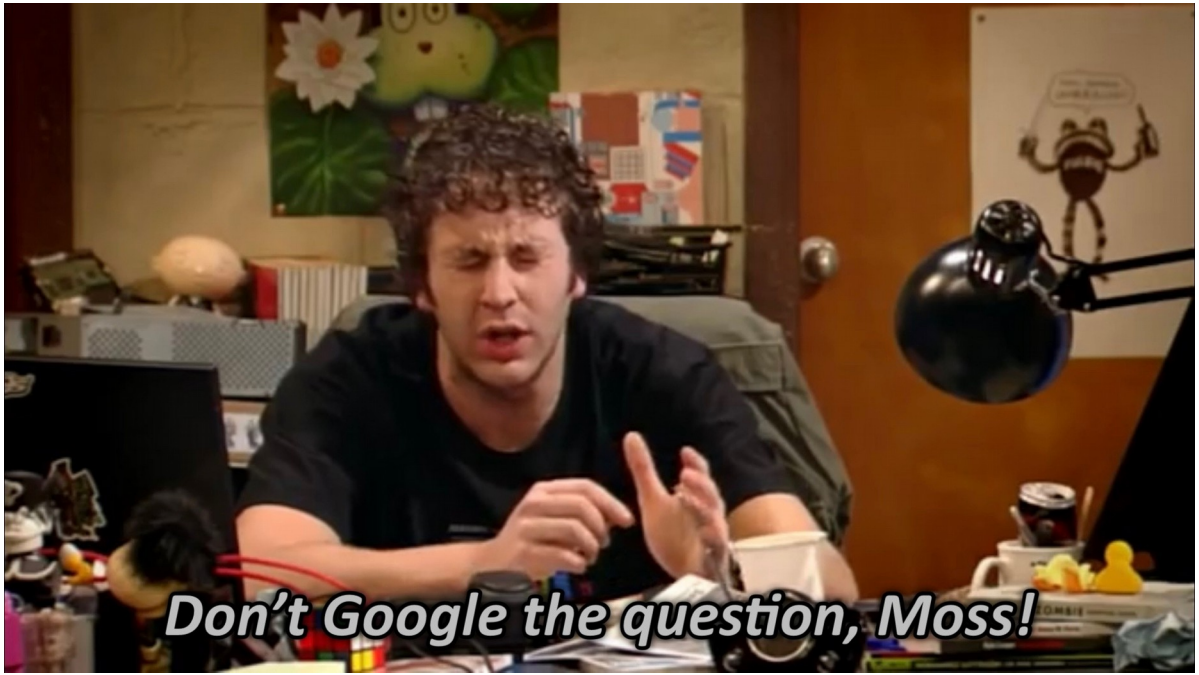
- We can add new images to our pipelines without much effort
- On the same idea, we can add support for other archs, such as aarch64
  - Still unable to test images due to the lack of hardware
- Other teams involved in the bootstrapping and configuration of machines can test their changes using our *image-ci* project
- With the extensive use of GitLab CI templates we can change all the images at once
- Control image releases with a single button
- Track past image builds

Future improvements

- PXE images are not present in this workflow
  - Used for both managed and unmanaged machines

- Machine creation shows quite some false negatives in terms of bootstrapping
  - IPv6 recurrent issues
  - Virtual machine creation failure rate
    - Not a real issue for other services, but given our creation rate this shows quite some failures from time to time that require manual investigation

Questions ?




AND REMEMBER FRIENDS
MONTHS OF CI/CD SETUP CAN
SAVE YOU YEARS OF REPETITIVE TASKS
UNTIL THE NEXT ONE!

www.cern.ch