



Navigating the Stream

Automating CentOS releases at CERN

2021-10-25

Alex Iribarren

On behalf of CERN's Linux Support Team

HEPiX Autumn 2021



The Problem

- We need a stable Linux OS
- Many systems don't have external connectivity
- Lack of control is scary
- Some customizations and CERN-specific tools would be nice

How can we solve it?

- Mirror upstream distributions
- “Curate” the upstream distribution
 - Keep out changes we don’t want
 - Control how often changes are made
- Add our tools and customizations

In short, create our own releases

How did we do it in the past?

- Scientific Linux, in collaboration with Fermilab
 - Rebuild of Red Hat Enterprise Linux
 - Until SLC6
- CERN CentOS 7
 - Upstream CentOS 7 with some customizations
 - No longer rebuilding (for most things)

What did it look like?

- Local mirrors of upstream repositories
- Every day, move new packages to testing repositories
 - http://linuxsoft.cern.ch/cern/centos/7/updates-testing/x86_64/
- On Thursdays, move contents of testing repositories to production ones
 - http://linuxsoft.cern.ch/cern/centos/7/updates/x86_64/
- Send emails to our users, publish web page announcing the changes
- Most machines use only the production repositories
- Some machines (QA) also use the testing repositories
 - If somebody notices an issue on a QA machine, they have to yell before Thursday

What's so bad about that?

- Testing was very limited
 - Nothing scripted, relies on somebody noticing an issue by chance
 - Some packages had less time to be tested than others
- Everybody forced on the same schedule
- Manual release process: person on support rota had to go through a detailed procedure *every day*
 - 9 scripts all told
 - Run one script, check output, run next script based on previous output, etc.

CentOS 8 was good opportunity to make changes



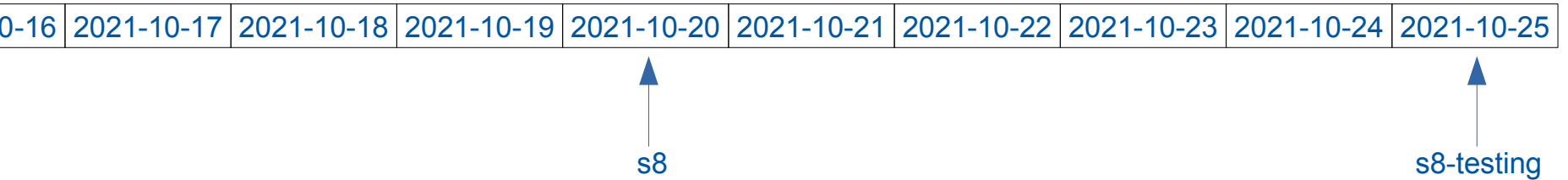
Work smarter, not harder

- Automate the process
 - Robots only intervene when absolutely necessary
- Improve the testing
 - Automate some tests
 - Make sure packages get more time in testing
- Let users move at their own pace
 - But maintain a reasonable “hands-off” default



Basic Idea

- Daily snapshots
- Testing and production symlinks



What's the new process?

- Local mirrors of upstream repositories
- Create daily snapshots
 - http://linuxsoft.cern.ch/cern/centos/s8-snapshots/20211025/BaseOS/x86_64/
- Modify the repositories (add/remove packages), if needed
- Add our own repositories
 - http://linuxsoft.cern.ch/cern/centos/s8-snapshots/20211025/CERN/x86_64/
- Check if that day's snapshot includes anything we have to rebuild
 - New kernel? Build kmod-openafs automatically against it
 - Some packages still have to be built by humans
- Run some basic validation checks
 - Are the repos well formed? Can you install some basic packages?

What's the new process?

- Move the `s8-testing` symlink so it points to today's snapshot
 - `s8-testing -> s8-snapshots/20211025`
- Wednesday only: move `s8` symlink (production) to the previous Wednesday's snapshot
 - `s8 -> s8-snapshots/20211013`
- Upload new PXE images to AIMS (our network installation server)
- Assemble the list of changes and update our website
 - https://linux.web.cern.ch/updates/cs8/prod/latest_updates/
- Email a summary of the changes
- Remove old snapshots (> 13 months)

Manual control options

- Stop symlink updates
 - Only testing, only prod or everything
- Force updating of production symlink
 - Move it forward in case of emergency security fixes
- Add/remove packages from upstream repos
 - And regenerate metadata as needed
- Build our own packages with higher epoch
 - Supersede upstream RPMs

Multiple schedules

- Daily QA testing:
 - <https://linuxsoft.cern.ch/cern/centos/s8-testing/>
- Weekly production:
 - <https://linuxsoft.cern.ch/cern/centos/s8/>
- Quarterly releases:
 - <https://linuxsoft.cern.ch/cern/centos/s8.2021.4/>, etc.
- Custom snapshot date:
 - <https://linuxsoft.cern.ch/cern/centos/s8-snapshots/20210201/>

No updates on weekends of Christmas holidays

How is this done?

- Several Bash scripts
 - https://gitlab.cern.ch/linuxsupport/cronjobs/stream8_snapshots/-/tree/master/stream8_snapshots
- Encapsulated in a Docker container with all needed dependencies
- Run by a **Nomad** cluster
- Changes deployed by Gitlab CI
- Logs in Elastic Search
- Monitored by CollectD and admin emails

The screenshot displays the Nomad web interface for a job named 'prod_stream8_snapshots'. The interface is divided into several sections:

- Header:** Shows the Nomad logo and a search bar.
- Navigation:** Includes tabs for 'Overview', 'Definition', 'Versions', 'Allocations', and 'Evaluations'. The 'Overview' tab is selected.
- Job Information:** Displays the job name 'prod_stream8_snapshots' with status indicators: 'RUNNING' (green), 'PERIODIC', and 'Force Launch' (orange). There are 'Exec' and 'Stop' buttons.
- Configuration:** Shows 'Version: 0 | Priority: 50 | Cron: 0 8 * * *'.
- Children Status:** A bar chart shows the status of children: 0 Pending, 1 Running, and 24 Dead. A 'collapse' button is present.
- Job Launches:** A table lists recent job launches.

Name	Submitted At	Status	Type	Priority	Groups	Summary
prod_stream8_snapshots/periodic-1635141600	Oct 25 08:00:00 +0200	RUNNING	batch	50	1	

Per page: 50

New OS? No* problem!

- Mirror new upstream repositories
- Fork a new Nomad cronjob
 - https://gitlab.cern.ch/linuxsupport/cronjobs/stream9_snapshots/
- Adapt it as necessary
- Start building packages
 - Easier thanks to [rpmci](#)
- Profit!

* Some problems, this isn't magic.

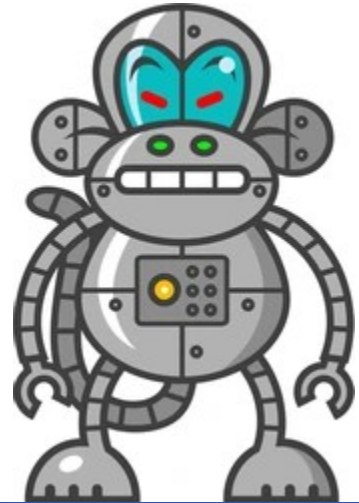
Future work

- Possibility to regenerate boot images (initrd) for all arches
- Closer integration with Image CI
 - See “Full OS image automation: Continuous Integration for CERN’s distros”
- Bring more of the improvements back to the CC7 workflow
- Continue working hard to work as little as possible

Questions?

Code available at:

- https://gitlab.cern.ch/linuxsupport/cronjobs/centos8_snapshots/
- https://gitlab.cern.ch/linuxsupport/cronjobs/stream8_snapshots/
- https://gitlab.cern.ch/linuxsupport/cronjobs/stream9_snapshots/





www.cern.ch