CentOS Stream

- preview of RHEL
- solid base for CentOS SIGs

Adam Samalik
Principal Software Engineer, Red Hat
CentOS Stream Engineering Lead

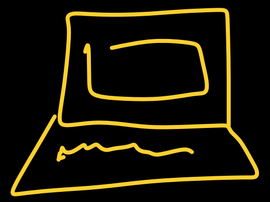
Fedora

Fedora Server

Fedora Silverblue

Fedora ELN

Fedora CoreOS



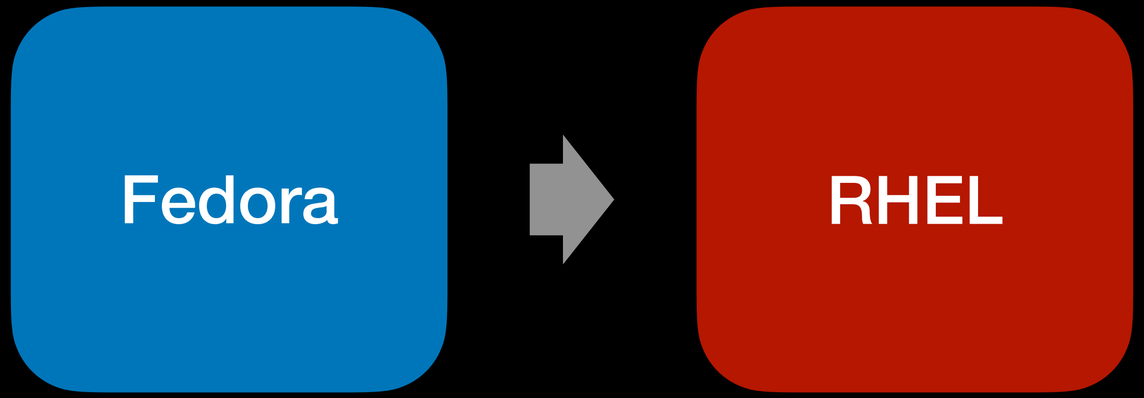
Fedora

Fedora Kinoite

Fedora Workstation

Fedora IoT

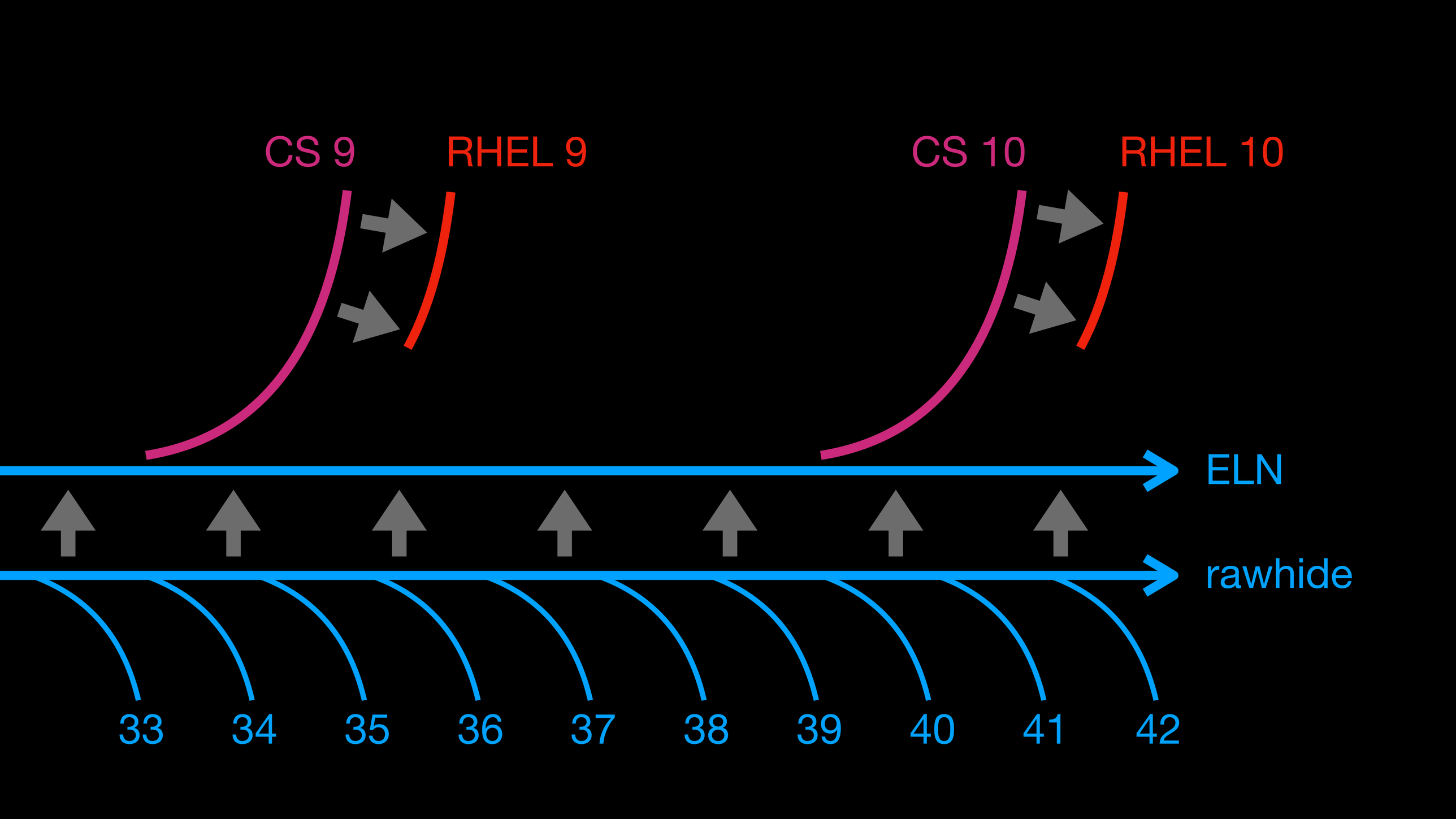
Fedora KDE

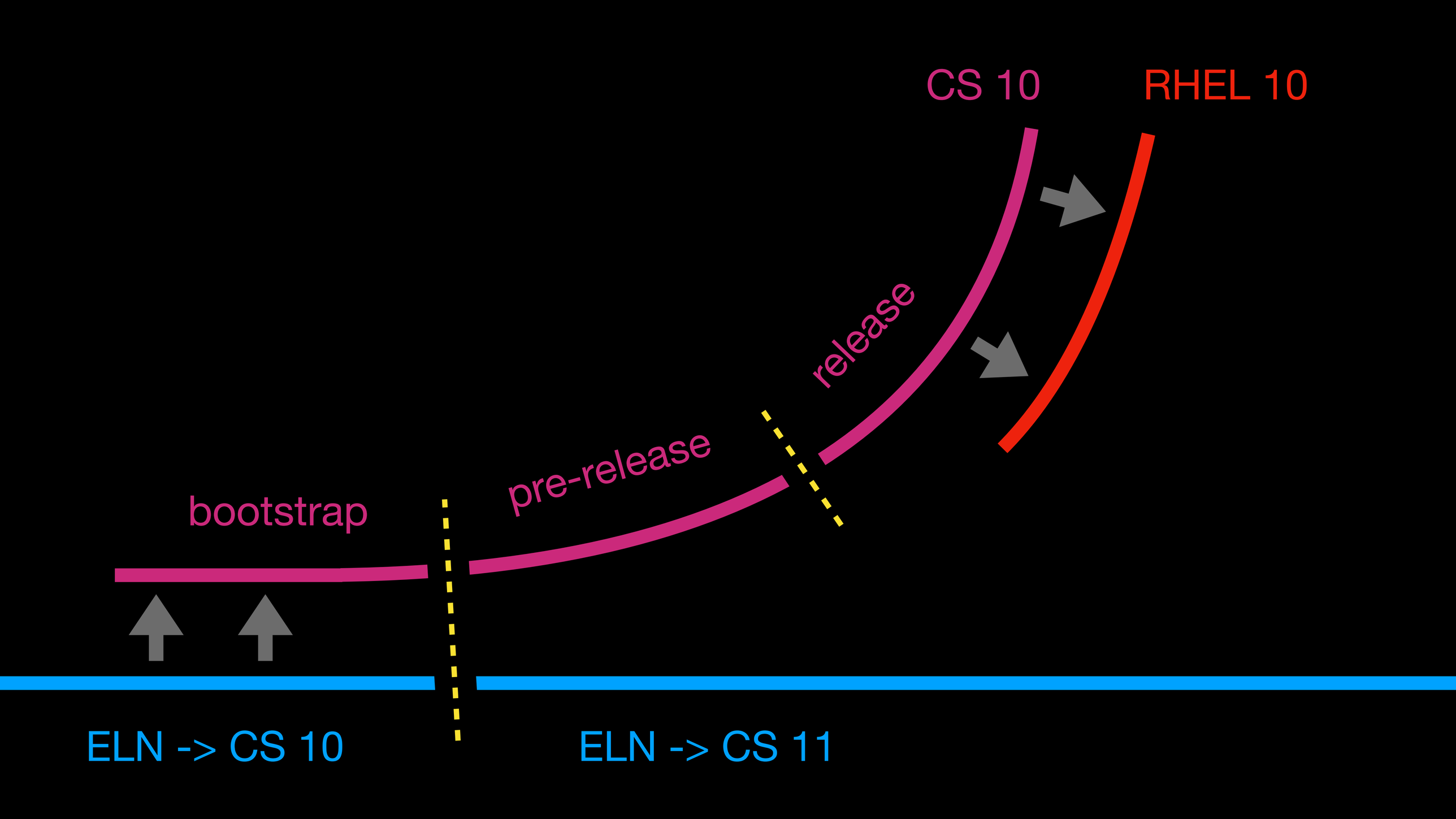

Fedora
RHEL

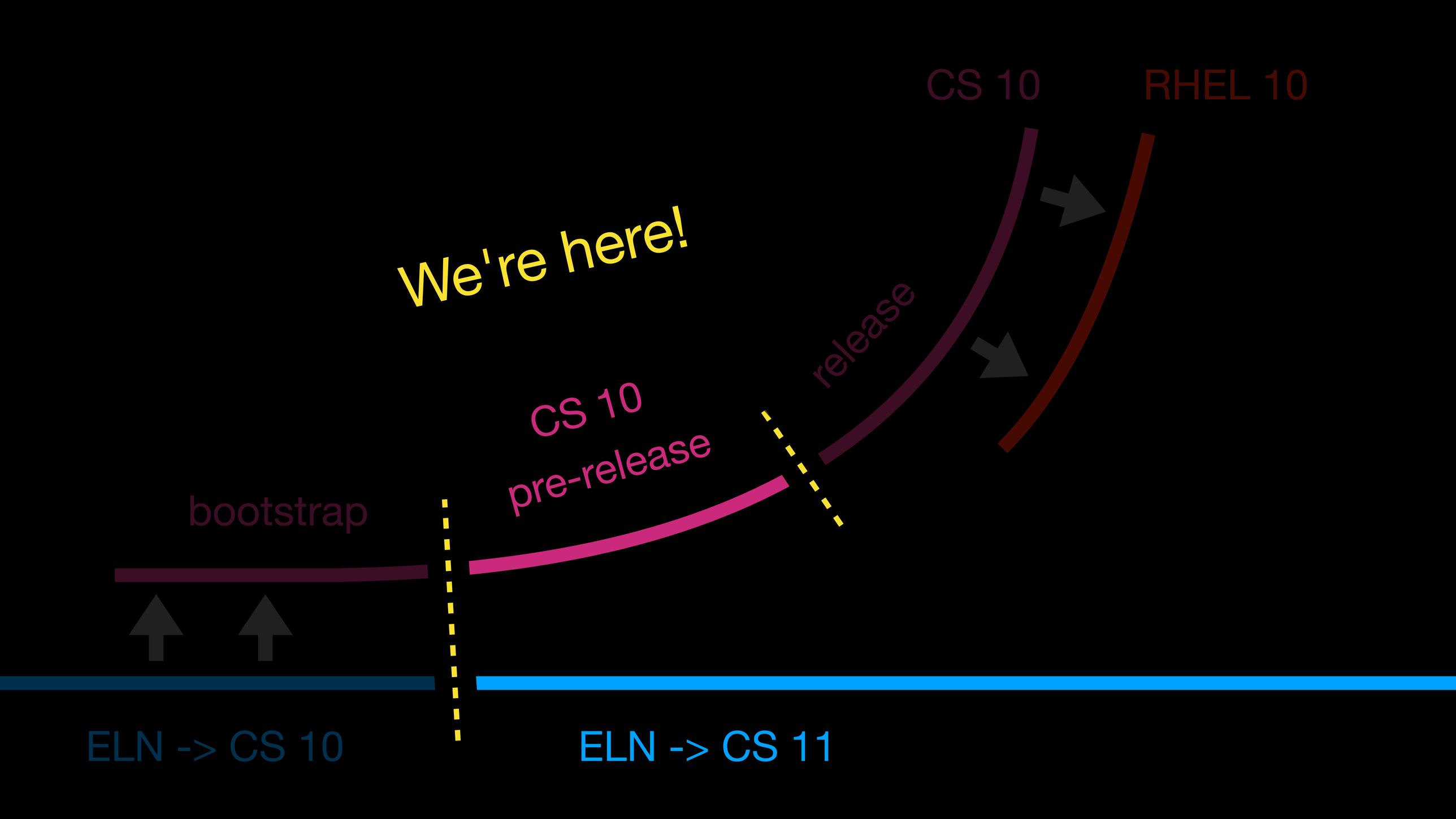

Fedora
CentOS
Stream
RHEL

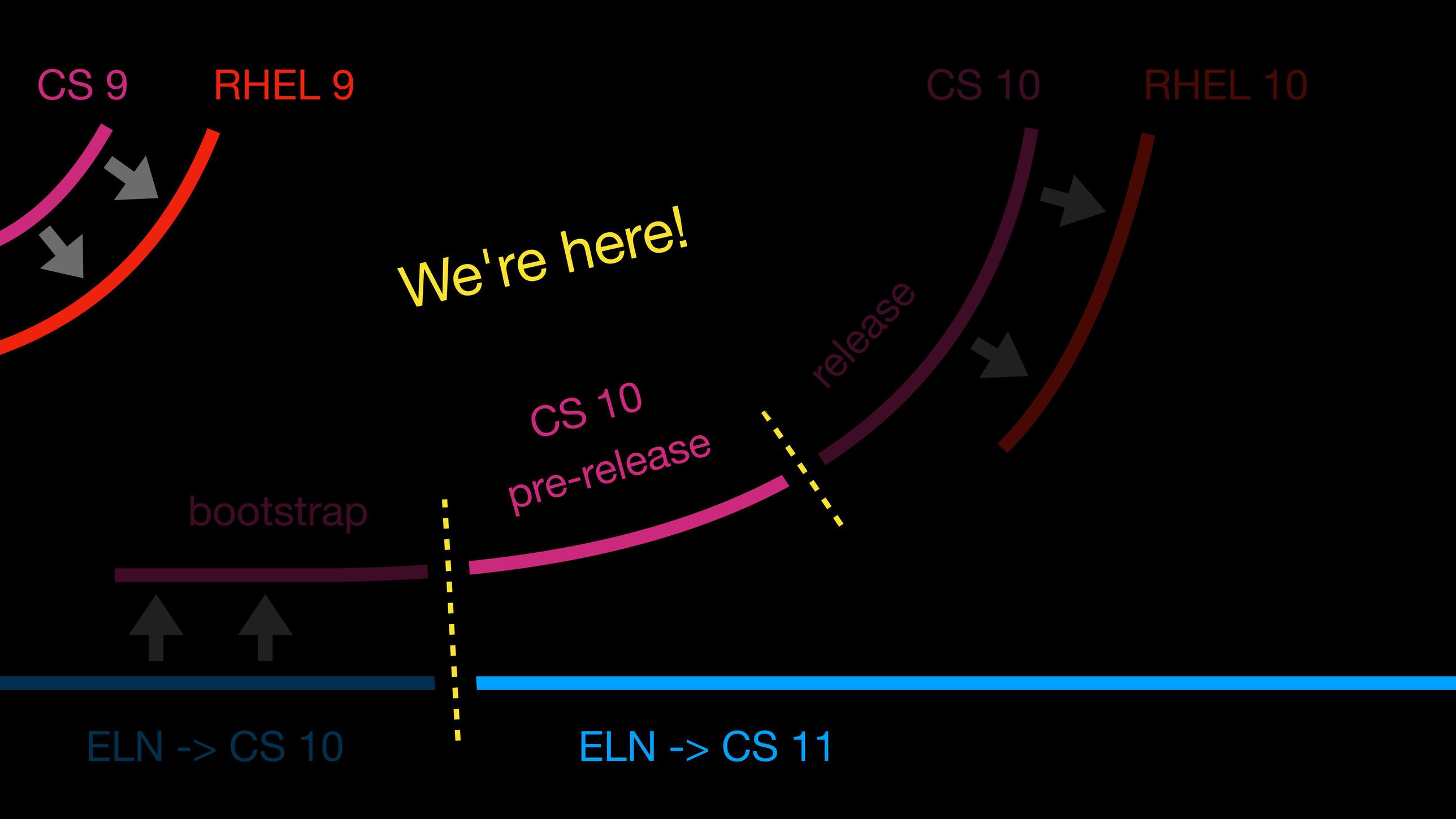
CentOS Stream

a Linux OS maintained by RHEL engineers

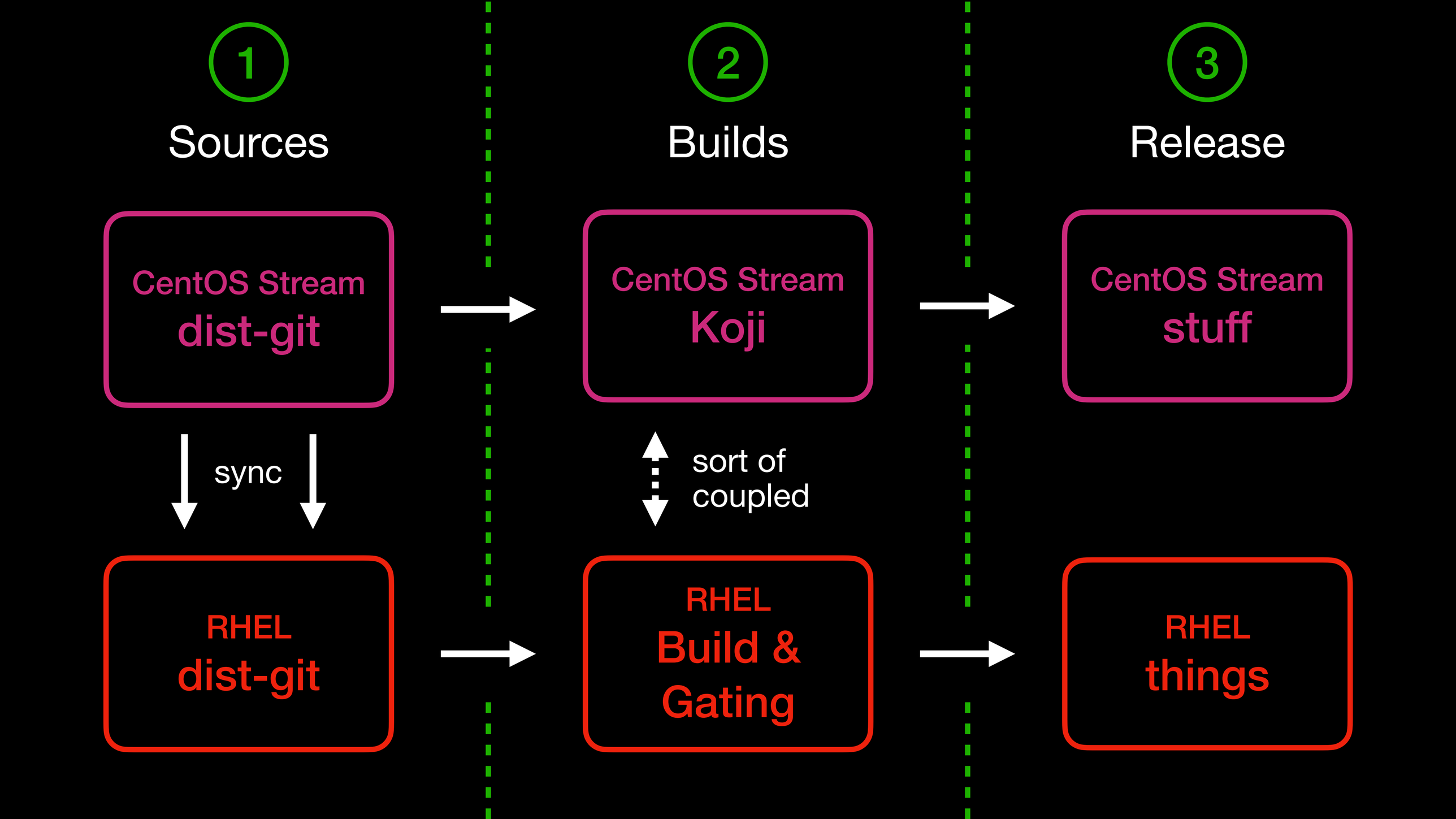

CS 9
RHEL 9
CS 10
RHEL 10
ELN
rawhide
33
34
35
36
37
38
39
40
41
42


CS 10
RHEL 10
release
pre-release
bootstrap
ELN -> CS 10
ELN -> CS 11


CS 10
RHEL 10
We're here!
release
CS 10
pre-release
bootstrap
ELN -> CS 10
ELN -> CS 11


CS 9
RHEL 9
CS 10
RHEL 10
We're here!
CS 10
pre-release
release
bootstrap
ELN -> CS 10
ELN -> CS 11

CentOS Stream
Pipeline


1
Sources
2
Builds
3
Release
CentOS Stream
dist-git
CentOS Stream
Koji
CentOS Stream
stuff
sync
sort of
coupled
RHEL
dist-git
RHEL
Build &
Gating
RHEL
things

Red Hat

Jira

issues.redhat.com

the RHEL project

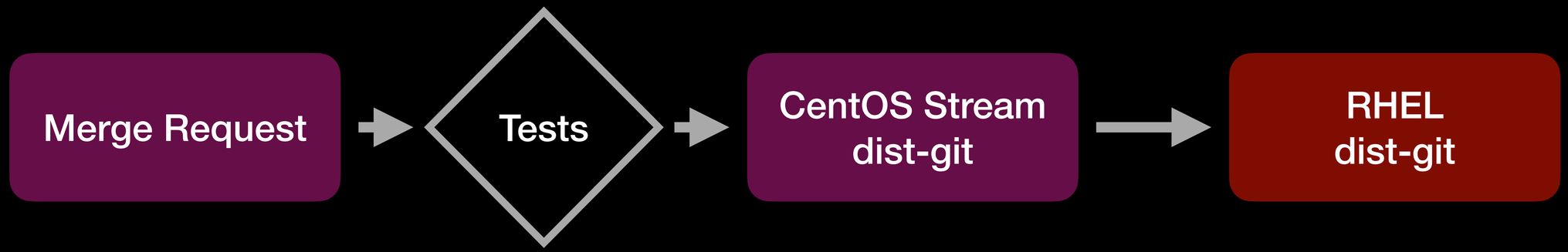
bug / change tracking

communication with RHEL maintainers

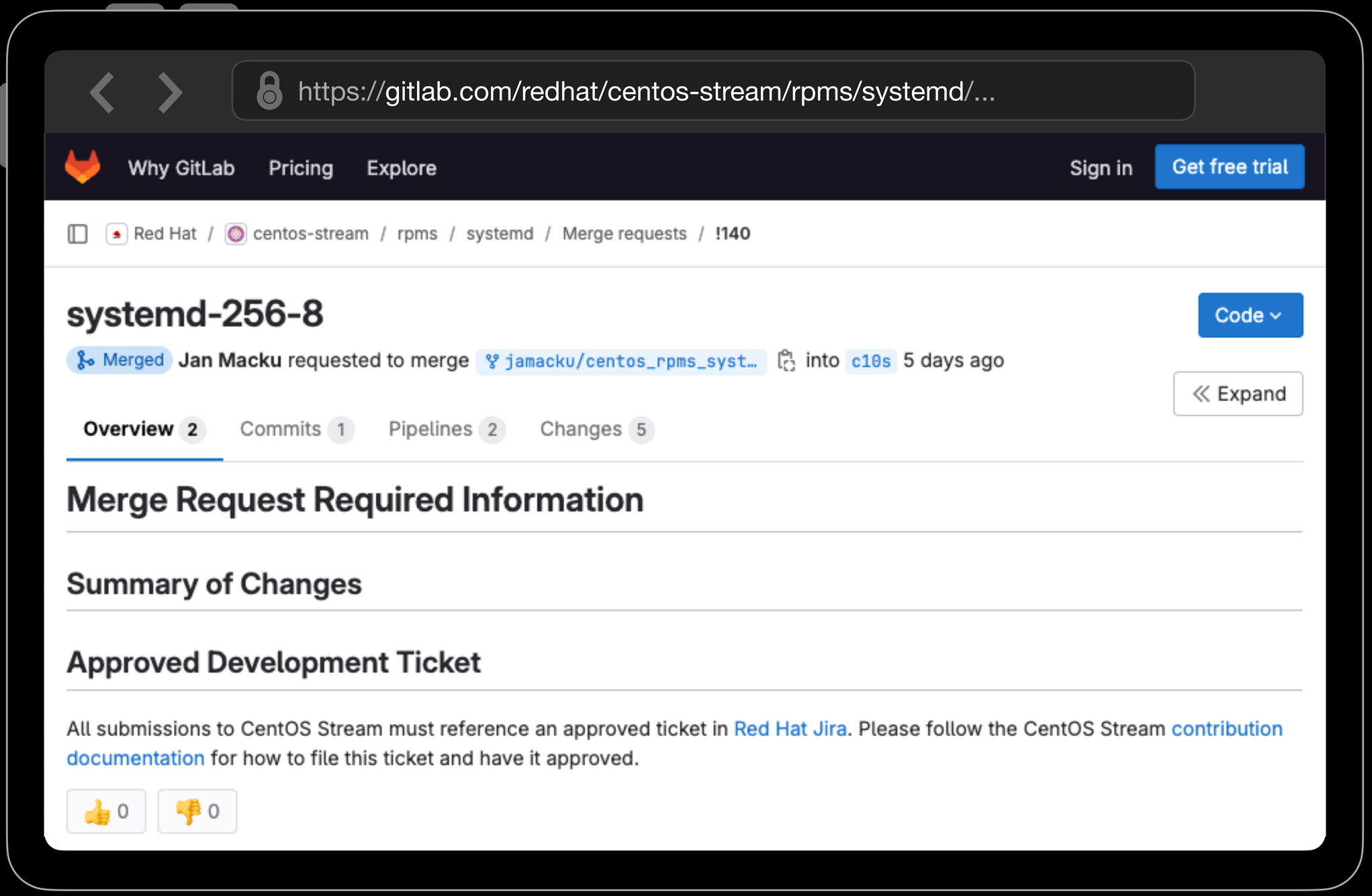
CentOS Stream

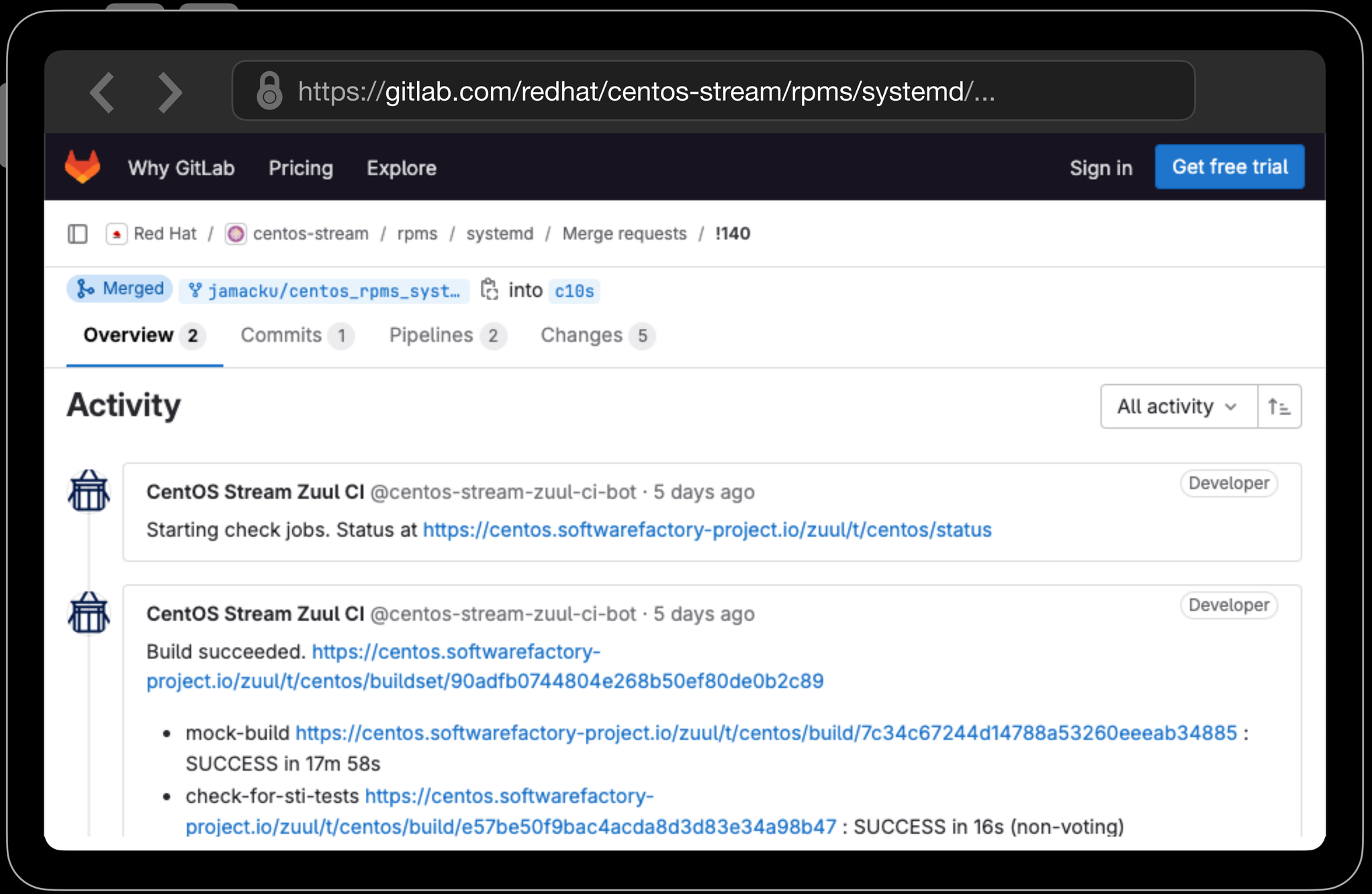
Sources

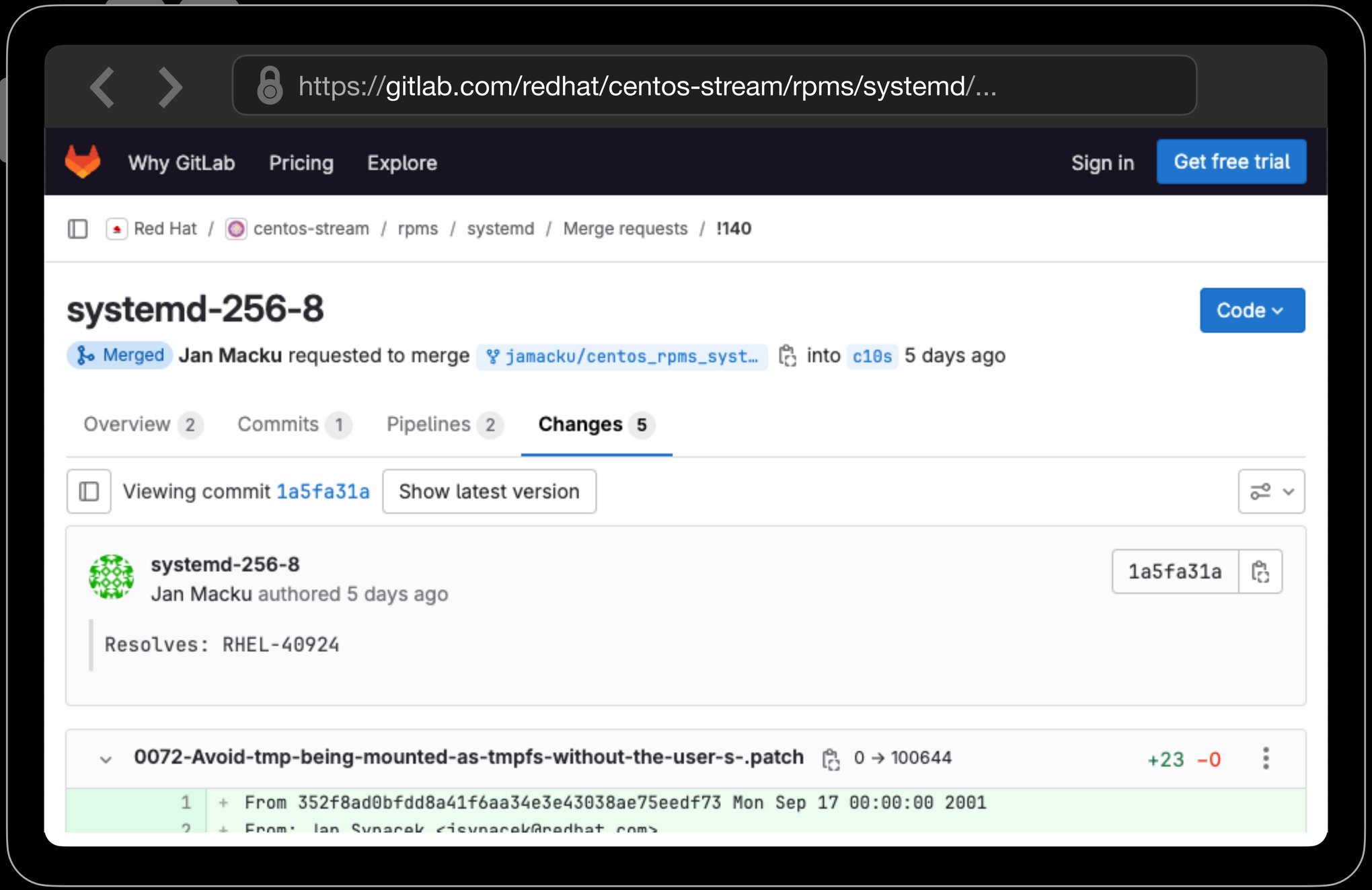
CentOS Stream Sources

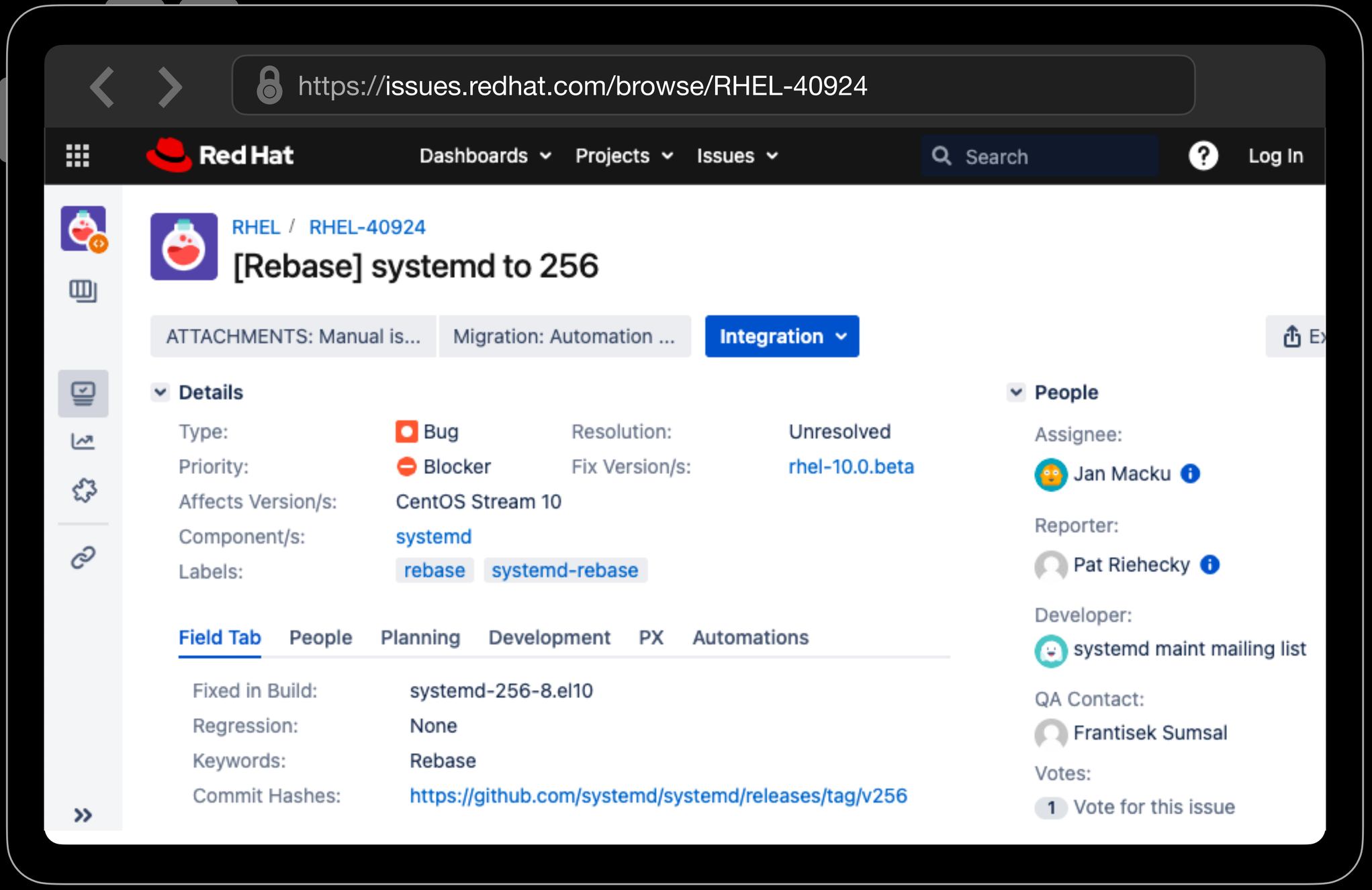


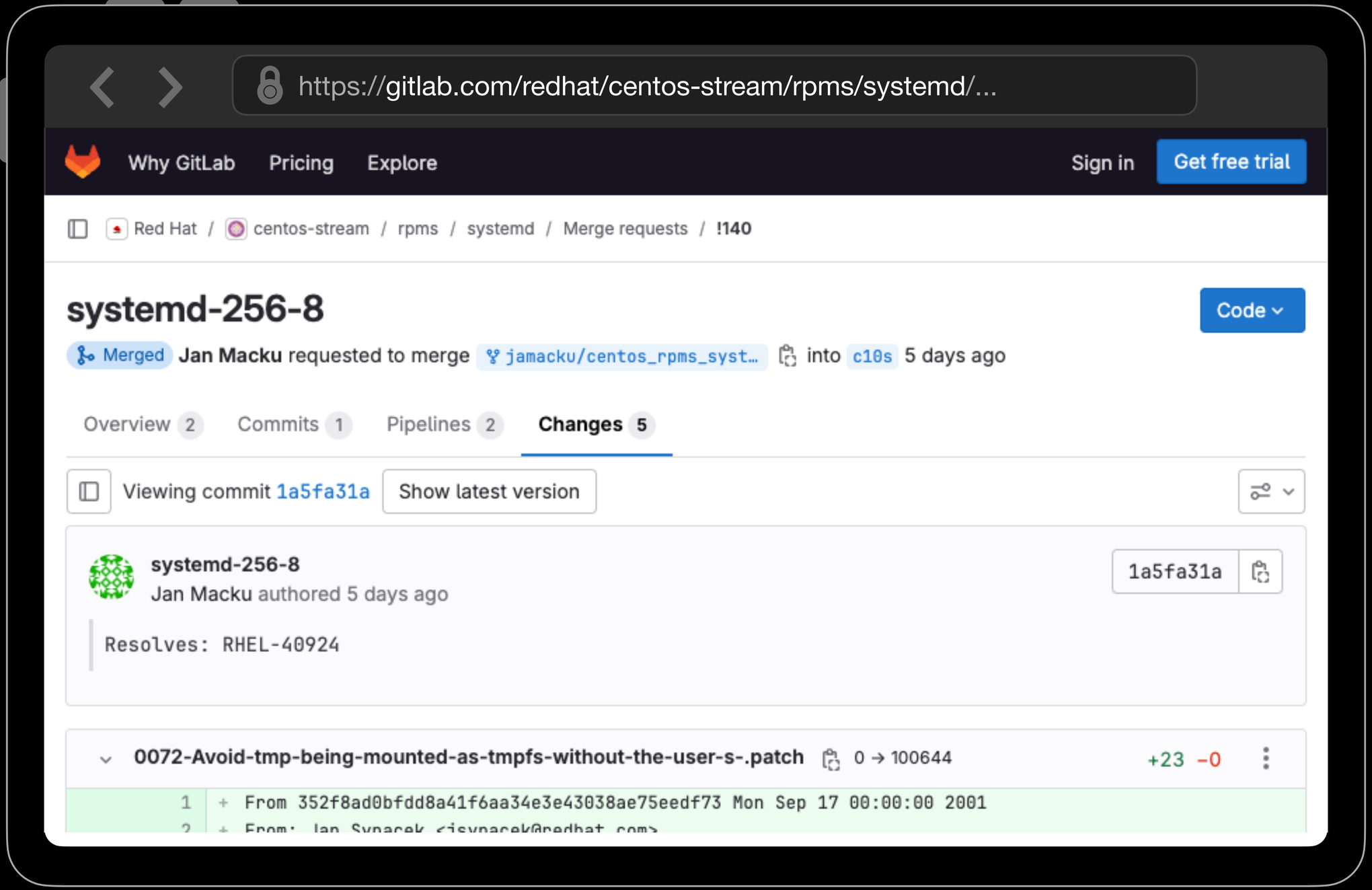
gitlab.com / redhat / centos-stream / rpms

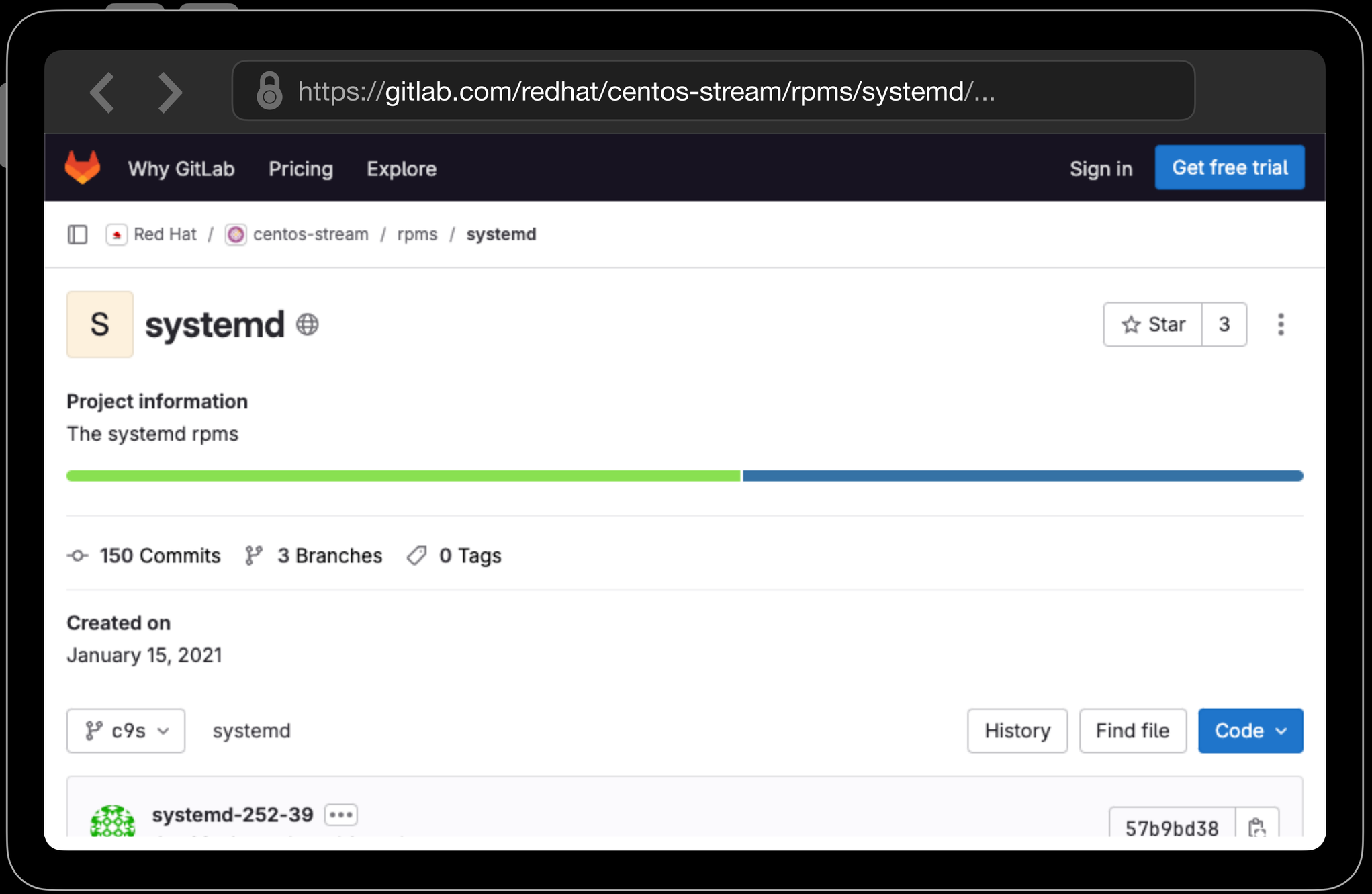

https://gitlab.com/redhat/centos-stream/rpms/systemd/...
Why GitLab
Pricing
Explore
Sign in
Get free trial
Red Hat / centos-stream / rpms / systemd / Merge requests / !140
systemd-256-8
Code
Merged Jan Macku requested to merge jamacku/centos_rpms_syst… into c10s 5 days ago
Expand
Overview 2
Commits 1
Pipelines 2
Changes 5
Merge Request Required Information
Summary of Changes
Approved Development Ticket
All submissions to CentOS Stream must reference an approved ticket in Red Hat Jira. Please follow the CentOS Stream contribution documentation for how to file this ticket and have it approved.
0
0

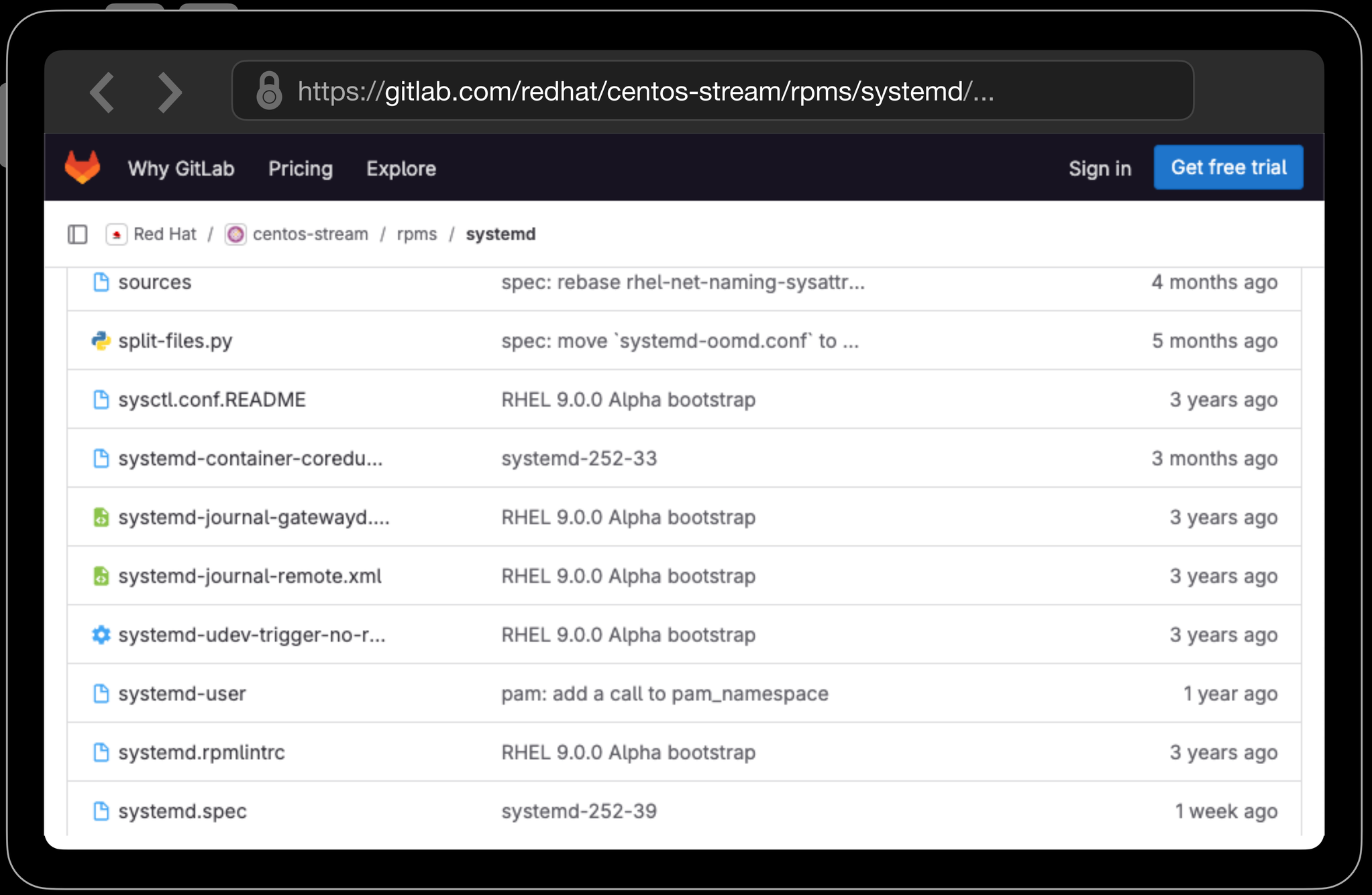

https://gitlab.com/redhat/centos-stream/rpms/systemd/...
Why GitLab
Pricing
Explore
Sign in
Get free trial
Red Hat / centos-stream / rpms / systemd / Merge requests / !140
Merged
jamacku/centos_rpms_syst…
into
c10s
Overview 2
Commits 1
Pipelines 2
Changes 5
Activity
All activity
CentOS Stream Zuul CI @centos-stream-zuul-ci-bot · 5 days ago
Developer
Starting check jobs. Status at https://centos.softwarefactory-project.io/zuul/t/centos/status
CentOS Stream Zuul CI @centos-stream-zuul-ci-bot · 5 days ago
Developer
Build succeeded. https://centos.softwarefactory-project.io/zuul/t/centos/buildset/90adfb0744804e268b50ef80de0b2c89
mock-build https://centos.softwarefactory-project.io/zuul/t/centos/build/7c34c67244d14788a53260eeeab34885 : SUCCESS in 17m 58s
check-for-sti-tests https://centos.softwarefactory-project.io/zuul/t/centos/build/e57be50f9bac4acda8d3d83e34a98b47 : SUCCESS in 16s (non-voting)


https://gitlab.com/redhat/centos-stream/rpms/systemd/...
Why GitLab
Pricing
Explore
Sign in
Get free trial
Red Hat / centos-stream / rpms / systemd / Merge requests / !140
systemd-256-8
Code
Merged Jan Macku requested to merge jamacku/centos_rpms_syst… into c10s 5 days ago
Overview 2
Commits 1
Pipelines 2
Changes 5
Viewing commit 1a5fa31a
Show latest version
systemd-256-8
Jan Macku authored 5 days ago
1a5fa31a
Resolves: RHEL-40924
0072-Avoid-tmp-being-mounted-as-tmpfs-without-the-user-s-.patch
0 → 100644
+23 -0
1 + From 352f8ad0bfdd8a41f6aa34e3e43038ae75eedf73 Mon Sep 17 00:00:00 2001


https://issues.redhat.com/browse/RHEL-40924
Red Hat
Dashboards
Projects
Issues
Search
Log In
RHEL / RHEL-40924
[Rebase] systemd to 256
ATTACHMENTS: Manual is...
Migration: Automation ...
Integration
Details
Type: Bug
Priority: Blocker
Affects Version/s: CentOS Stream 10
Component/s: systemd
Labels: rebase systemd-rebase
Resolution: Unresolved
Fix Version/s: rhel-10.0.beta
Field Tab
People
Planning
Development
PX
Automations
Fixed in Build: systemd-256-8.el10
Regression: None
Keywords: Rebase
Commit Hashes: https://github.com/systemd/systemd/releases/tag/v256
People
Assignee: Jan Macku
Reporter: Pat Riehecky
Developer: systemd maint mailing list
QA Contact: Frantisek Sumsal
Votes: 1 Vote for this issue


https://gitlab.com/redhat/centos-stream/rpms/systemd/...
Why GitLab
Pricing
Explore
Sign in
Get free trial
Red Hat / centos-stream / rpms / systemd / Merge requests / !140
systemd-256-8
Code
Merged Jan Macku requested to merge jamacku/centos_rpms_syst… into c10s 5 days ago
Overview 2
Commits 1
Pipelines 2
Changes 5
Viewing commit 1a5fa31a
Show latest version
systemd-256-8
Jan Macku authored 5 days ago
1a5fa31a
Resolves: RHEL-40924
0072-Avoid-tmp-being-mounted-as-tmpfs-without-the-user-s-.patch
0 → 100644
+23 -0
1 + From 352f8ad0bfdd8a41f6aa34e3e43038ae75eedf73 Mon Sep 17 00:00:00 2001


https://gitlab.com/redhat/centos-stream/rpms/systemd/...
Why GitLab
Pricing
Explore
Sign in
Get free trial
Red Hat / centos-stream / rpms / systemd
S
systemd
Star 3
Project information
The systemd rpms
150 Commits
3 Branches
0 Tags
Created on
January 15, 2021
c9s
systemd
History
Find file
Code
systemd-252-39
57b9bd38


https://gitlab.com/redhat/centos-stream/rpms/systemd/...
Why GitLab
Pricing
Explore
Sign in
Get free trial
Red Hat / centos-stream / rpms / systemd
sources
spec: rebase rhel-net-naming-sysattr...
4 months ago
split-files.py
spec: move `systemd-oomd.conf` to ...
5 months ago
sysctl.conf.README
RHEL 9.0.0 Alpha bootstrap
3 years ago
systemd-container-coredu...
systemd-252-33
3 months ago
systemd-journal-gatewayd....
RHEL 9.0.0 Alpha bootstrap
3 years ago
systemd-journal-remote.xml
RHEL 9.0.0 Alpha bootstrap
3 years ago
systemd-udev-trigger-no-r...
RHEL 9.0.0 Alpha bootstrap
3 years ago
systemd-user
pam: add a call to pam_namespace
1 year ago
systemd.rpmlintrc
RHEL 9.0.0 Alpha bootstrap
3 years ago
systemd.spec
systemd-252-39
1 week ago

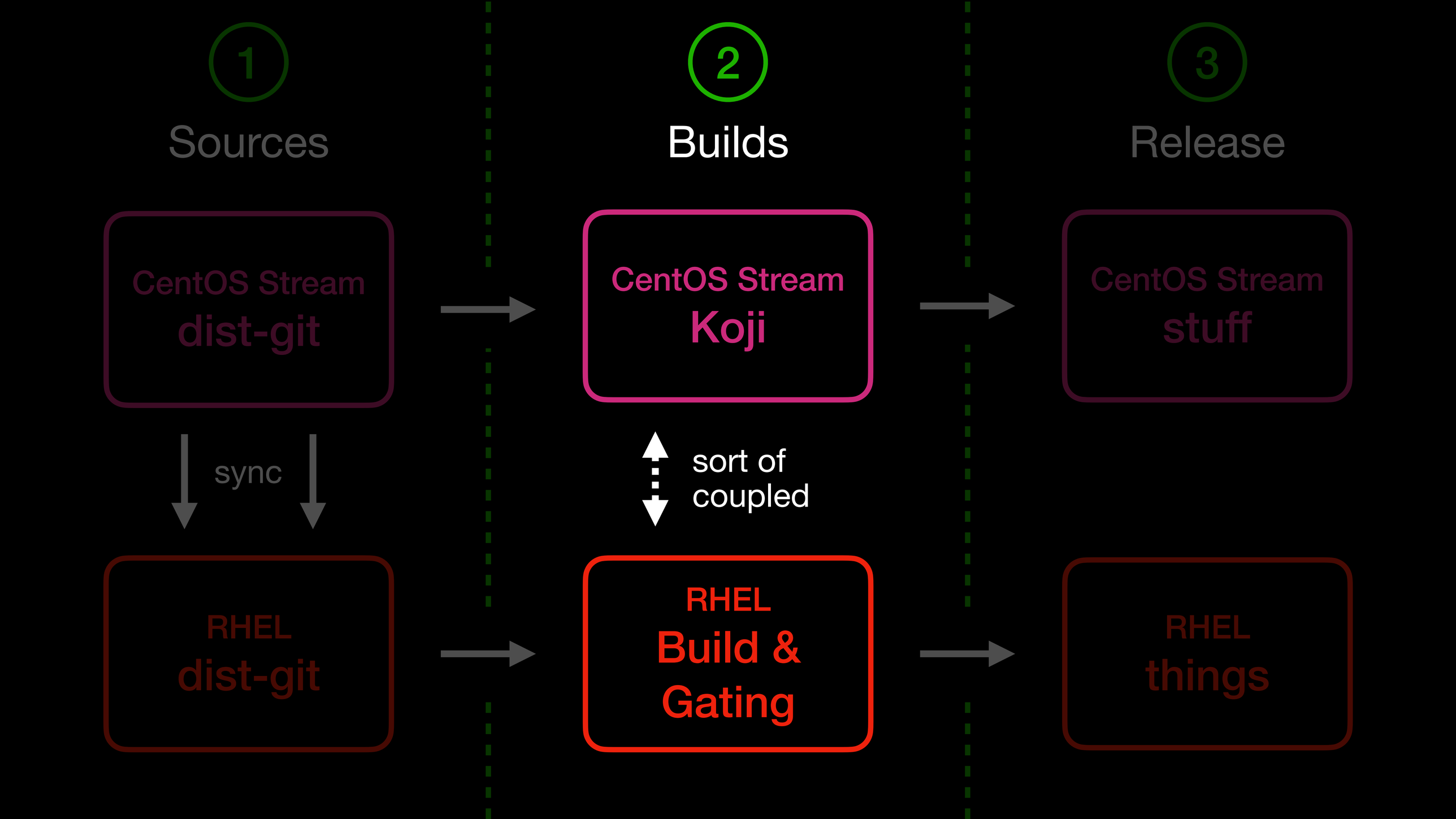

$ podman run --rm -it fedora:40
[root@199d0cc90e4a /]# dnf install centpkg

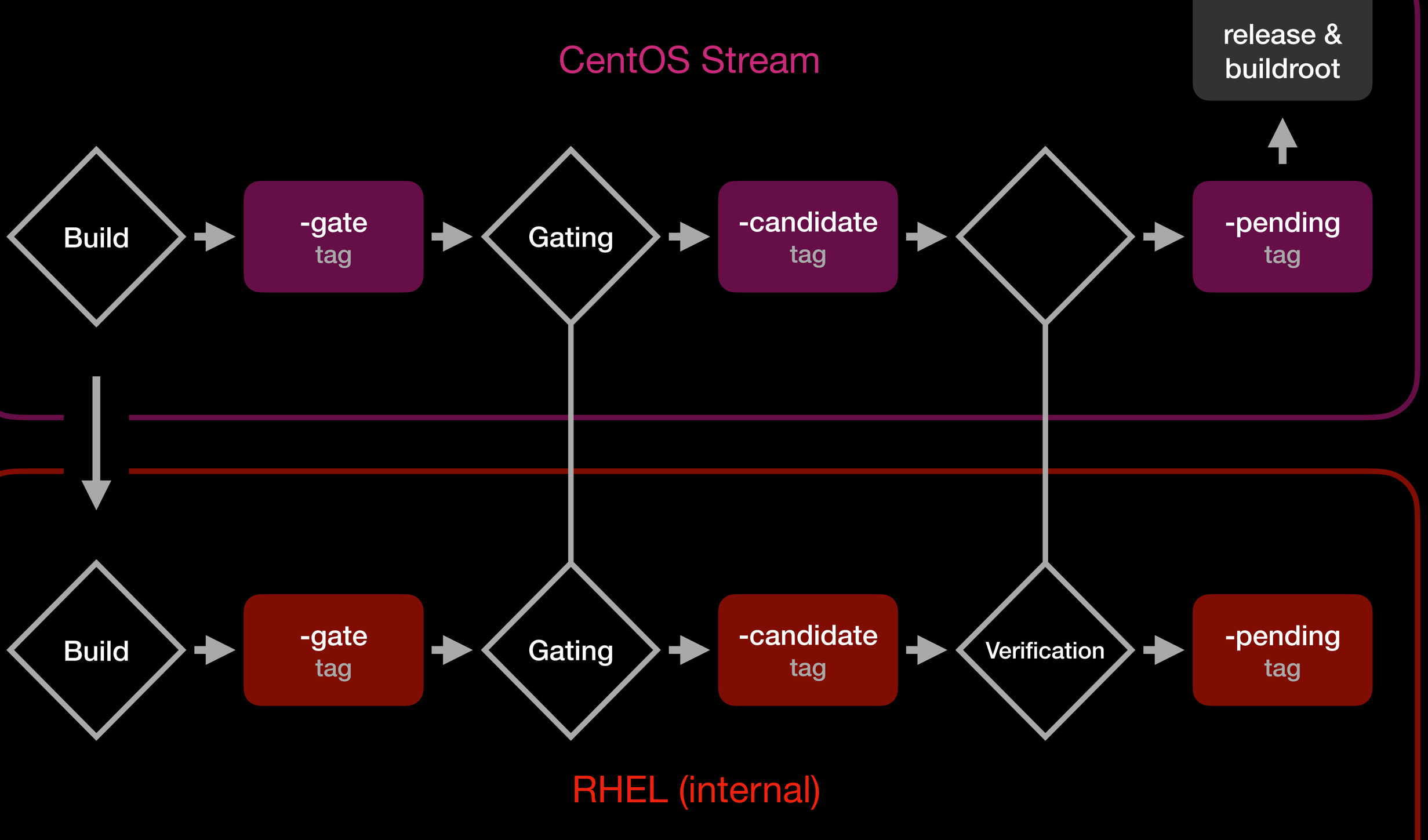
```
  rpmlint-fedora-license-data-1.51-1.fc40.noarch
  rust-srpm-macros-26.3-1.fc40.noarch
  sdbus-cpp-1.4.0-2.fc40.x86_64
  shadow-utils-subid-2:4.15.1-3.fc40.x86_64
  shared-mime-info-2.3-5.fc40.x86_64
  systemd-255.10-1.fc40.x86_64
  systemd-container-255.10-1.fc40.x86_64
  systemd-networkd-255.10-1.fc40.x86_64
  systemd-pam-255.10-1.fc40.x86_64
  systemd-resolved-255.10-1.fc40.x86_64
  systemd-rpm-macros-255.10-1.fc40.noarch
  systemd-udev-255.10-1.fc40.x86_64
  unzip-6.0-63.fc40.x86_64
  usermode-1.114-9.fc40.x86_64
  util-linux-2.40.1-1.fc40.x86_64
  xkeyboard-config-2.41-1.fc40.noarch
  xxhash-libs-0.8.2-2.fc40.x86_64
  yajl-2.1.0-23.fc40.x86_64
  zig-srpm-macros-1-2.fc40.noarch
  zip-3.0-40.fc40.x86_64

Complete!
[root@199d0cc90e4a /]# mkdir directory-somewhere
[root@199d0cc90e4a /]# cd directory-somewhere/
[root@199d0cc90e4a directory-somewhere]# centpkg clone -a sys
temd
```

```
  systemd-udev-255.10-1.fc40.x86_64
  unzip-6.0-63.fc40.x86_64
  usermode-1.114-9.fc40.x86_64
  util-linux-2.40.1-1.fc40.x86_64
  xkeyboard-config-2.41-1.fc40.noarch
  xxhash-libs-0.8.2-2.fc40.x86_64
  yajl-2.1.0-23.fc40.x86_64
  zig-srpm-macros-1-2.fc40.noarch
  zip-3.0-40.fc40.x86_64

Complete!
[root@199d0cc90e4a /]# mkdir directory-somewhere
[root@199d0cc90e4a /]# cd directory-somewhere/
[root@199d0cc90e4a directory-somewhere]# centpkg clone -a sys
temd
Cloning into 'systemd'...
remote: Enumerating objects: 9305, done.
remote: Counting objects: 100% (9169/9169), done.
remote: Compressing objects: 100% (5746/5746), done.
remote: Total 9305 (delta 3410), reused 9161 (delta 3402), pa
ck-reused 136 (from 1)
Receiving objects: 100% (9305/9305), 7.75 MiB | 65.05 MiB/s,
done.
Resolving deltas: 100% (3470/3470), done.
[root@199d0cc90e4a directory-somewhere]# cd systemd/
[root@199d0cc90e4a systemd]# centpkg srpm
```

CentOS Stream

Builds


1
Sources
2
Builds
3
Release
CentOS Stream
dist-git
CentOS Stream
Koji
CentOS Stream
stuff
sync
sort of
coupled
RHEL
dist-git
RHEL
Build &
Gating
RHEL
things


CentOS Stream
release &
buildroot
Build
-gate
tag
Gating
-candidate
tag
-pending
tag
Build
-gate
tag
Gating
-candidate
tag
Verification
-pending
tag
RHEL (internal)

https://kojihub.stream.centos.org/koji/buildinfo?buildID=66056

CentOS Stream Build Service

Mon, 29 Jul 2024 09:23:18 UTC | login

Packages SEARCH

Summary Packages Builds Tasks Tags Build Targets Users Hosts Reports Search API

Information for build systemd-256-8.el10

ID	66056
Package Name	systemd
Version	256
Release	8.el10
Epoch	
Draft	False
Source	git+https://gitlab.com/redhat/centos-stream/rpms/systemd.git#1a5fa31a19c0769033bcfcb284bbeca6e93ef861
Summary	System and Service Manager
Description	systemd is a system and service manager that runs as PID 1 and starts the rest of the system. It provides aggressive paralle capabilities, uses socket and D-Bus activation for starting services, offers on-demand starting of daemons, keeps track of Linux control groups, maintains mount and automount points, and implements an elaborate transactional dependency-bas control logic. systemd supports SysV and LSB init scripts and works as a replacement for sysvinit. Other parts of this packa logging daemon, utilities to control basic system configuration like the hostname, date, locale, maintain a list of logged-in

https://kojihub.stream.centos.org/koji/buildinfo?buildID=66056

control logic. systemd supports SysV and LSB init scripts and works as a replacement for sysvinit. Other parts of this packa
logging daemon, utilities to control basic system configuration like the hostname, date, locale, maintain a list of logged-in
accounts, runtime directories and settings, and a logging daemons.

Built by jamacku

State complete

Volume koji02

Started Tue, 23 Jul 2024 13:09:26 UTC

Completed Tue, 23 Jul 2024 13:14:30 UTC

Task build (c10s-candidate, /redhat/centos-stream/rpms/systemd.git:1a5fa31a19c0769033bcfcb284bbeca6e93ef861)

Extra {'custom_user_metadata': {'rhel-target': 'latest'}, 'source': {'original_url': 'git+https://gitlab.com/redhat/centos-stream/rpms/systemd.git#1a5fa31a19c0769033bcfcb284bbeca6e93ef861'}}

Tags c10s-candidate
c10s-gate
c10s-pending
c10s-pending-signed

RPMs

src

systemd-256-8.el10.src.rpm (info) (download)

aarch64

systemd-256-8.el10.aarch64.rpm (info) (download)

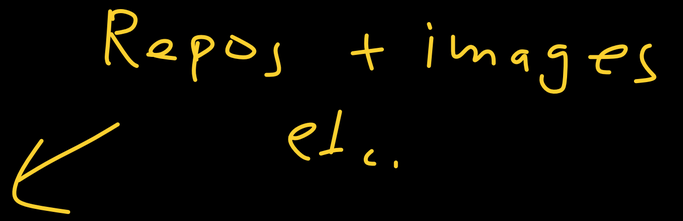
CentOS Stream

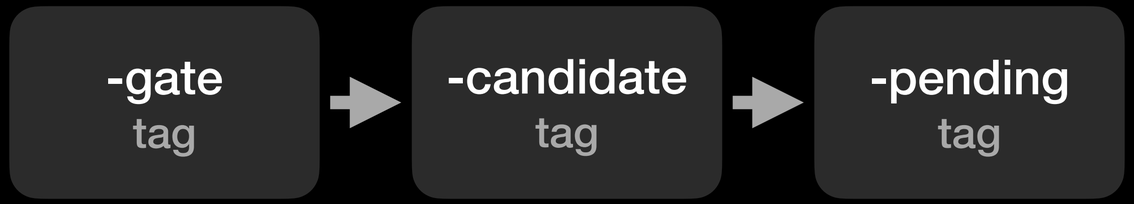
Release

1) Composes

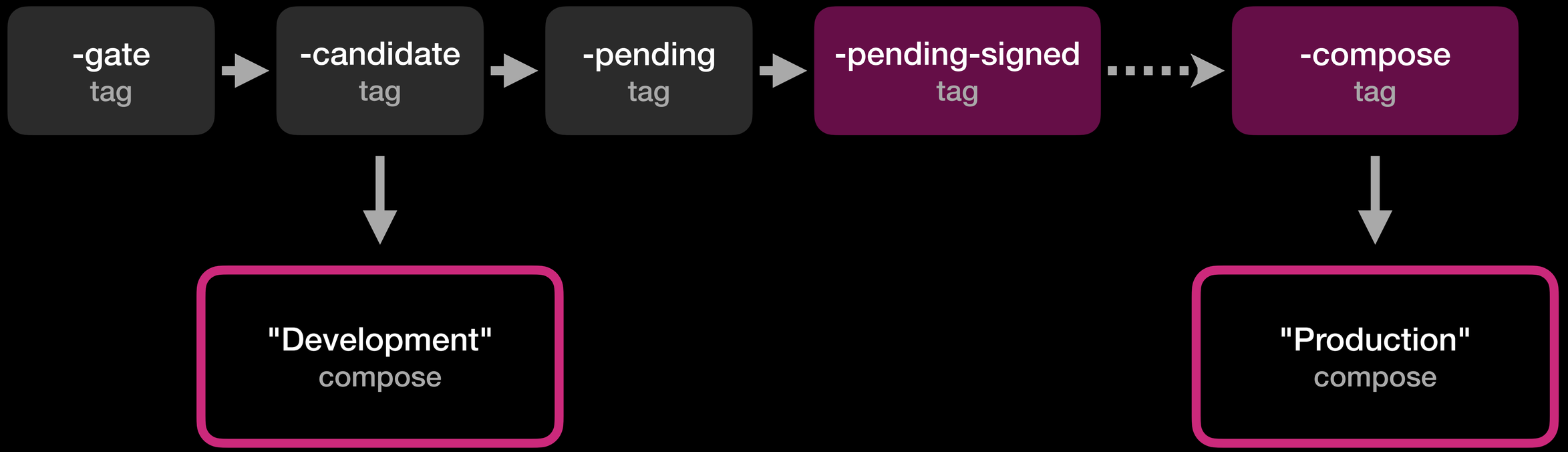
2) Testing

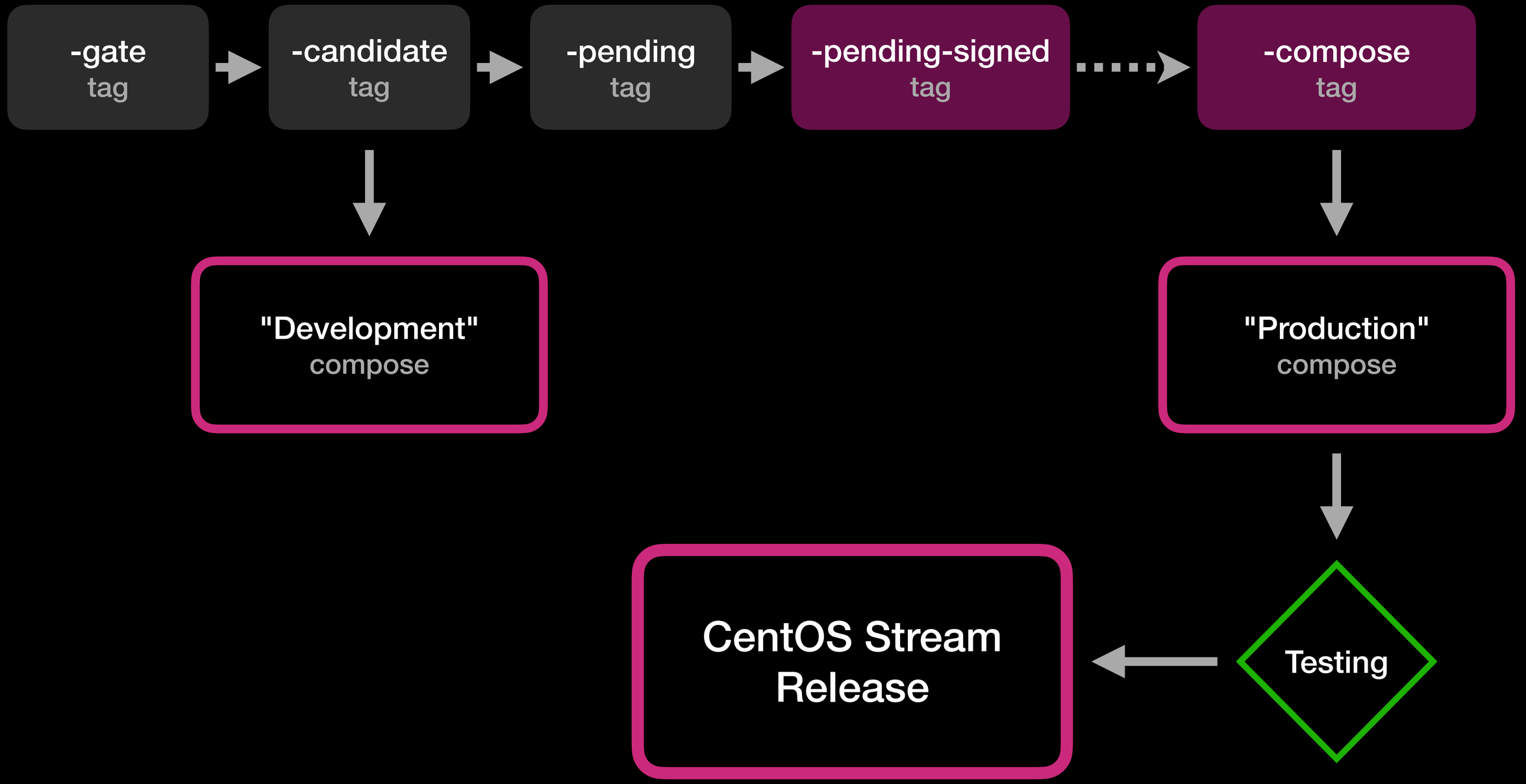
3) Release to mirrors etc.




-gate
tag
-candidate
tag
-pending
tag


-gate
tag
-candidate
tag
-pending
tag
-pending-signed
tag
-compose
tag


-gate
tag
-candidate
tag
-pending
tag
-pending-signed
tag
-compose
tag
"Development"
compose
"Production"
compose


-gate
tag
-candidate
tag
-pending
tag
-pending-signed
tag
-compose
tag
"Development"
compose
"Production"
compose
Testing
CentOS Stream
Release

CentOS Stream Release

Most of the content

BaseOS repository

AppStream repository

-devel packages, etc.

CRB repository

add-ons

HighAvailability repository

NFV repository

ResilientStorage repository

RT repository

SAP repository

SAPHANA repository

DVD iso image

netinstall iso image

container image

AWS image

qcow2 image

CentOS Stream Release

Most of the content

BaseOS repository

AppStream repository

-devel packages, etc.

CRB repository

add-ons

HighAvailability repository

NFV repository

ResilientStorage repository

RT repository

SAP repository

SAPHANA repository

DVD iso image

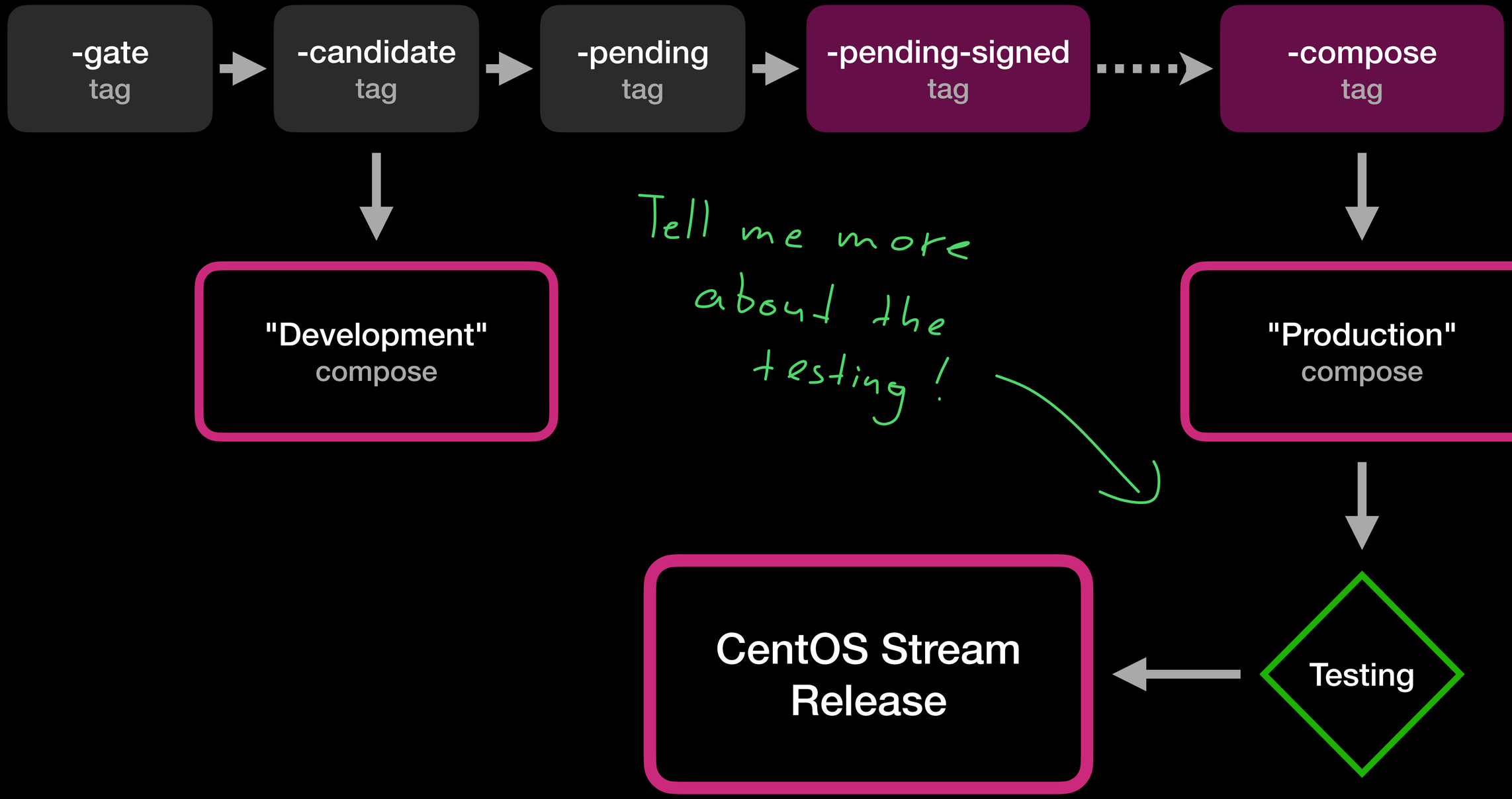
netinstall iso image

container image

AWS image

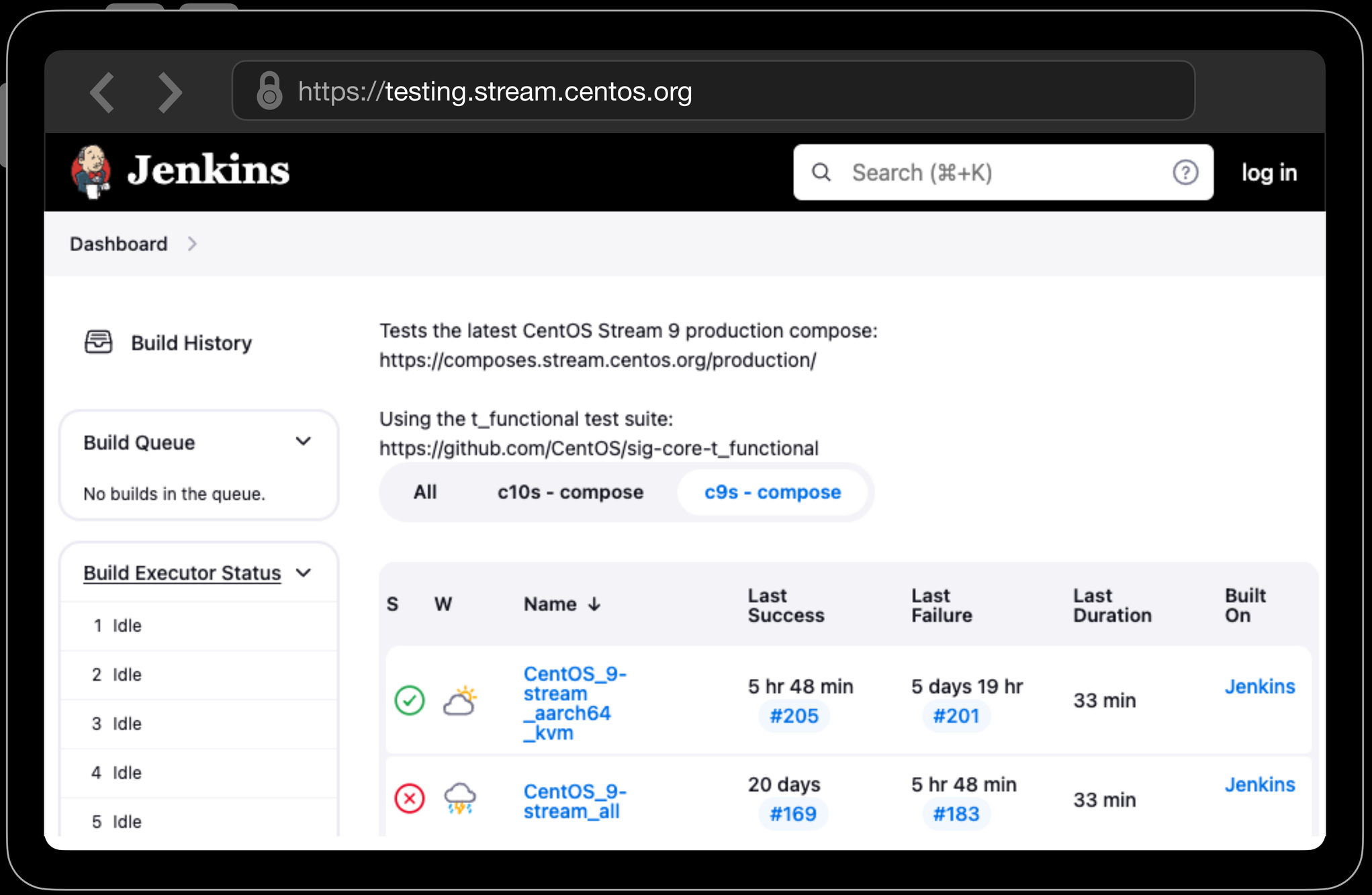
qcow2 image

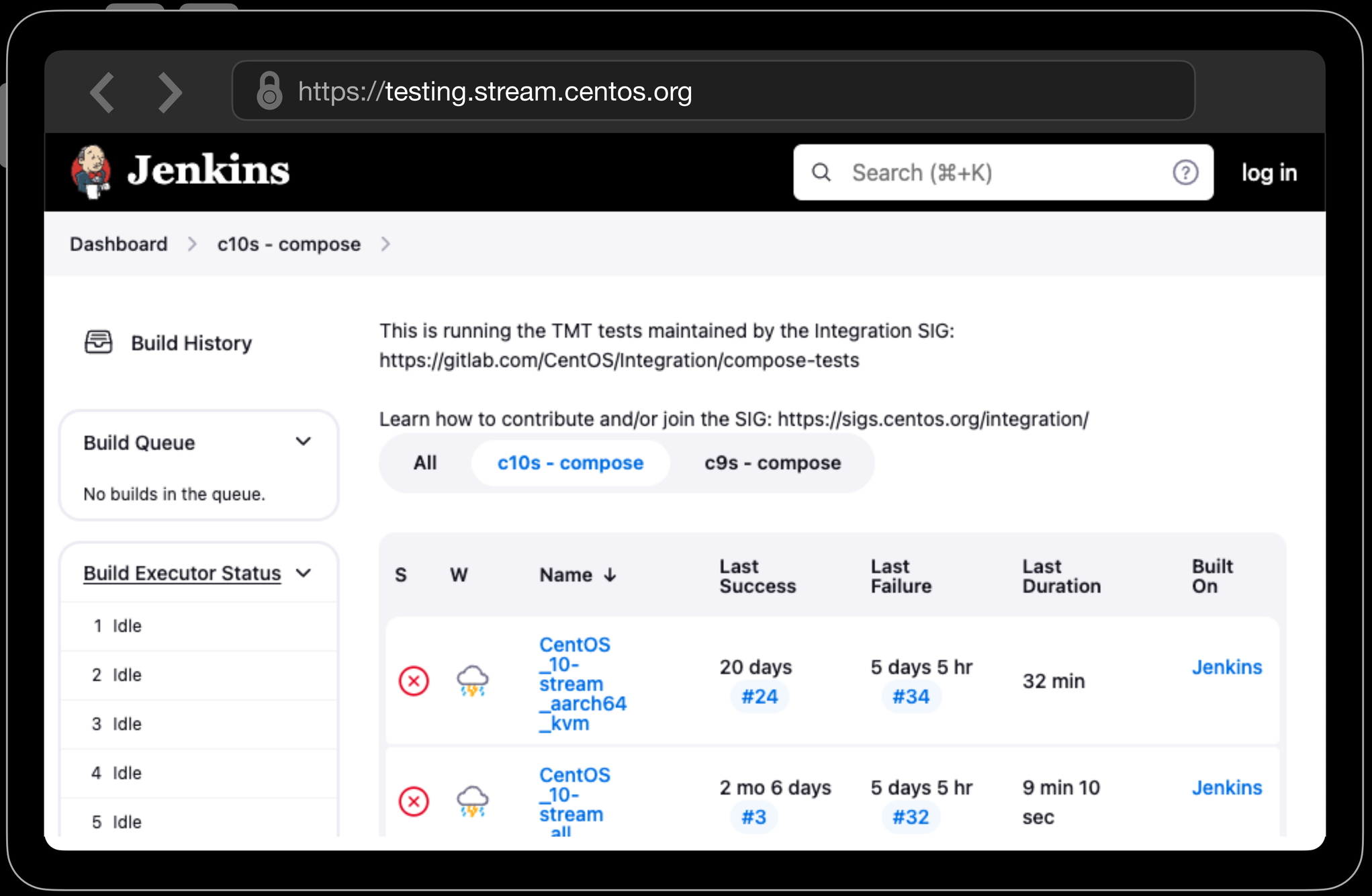
More images? -> Alternative Images SIG

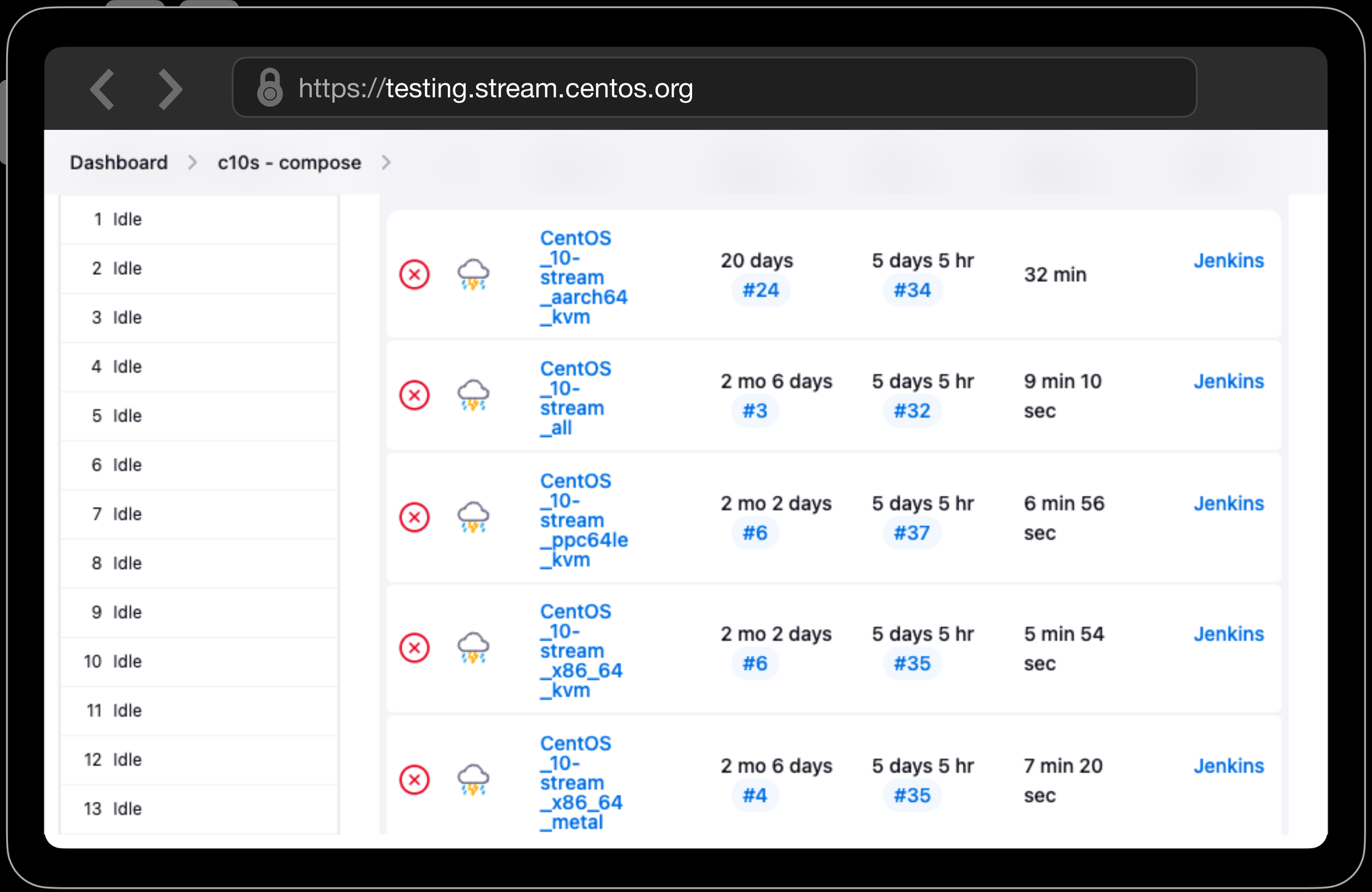

-gate
tag
-candidate
tag
-pending
tag
-pending-signed
tag
-compose
tag
"Development"
compose
Tell me more about the testing!
"Production"
compose
CentOS Stream
Release
Testing

CentOS

Integration SIG


https://testing.stream.centos.org
Jenkins
Search (⌘+K)
log in
Dashboard
Build History
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
3 Idle
4 Idle
5 Idle
Tests the latest CentOS Stream 9 production compose:
https://composes.stream.centos.org/production/
Using the t_functional test suite:
https://github.com/CentOS/sig-core-t_functional
All
c10s - compose
c9s - compose
S
W
Name ↓
Last Success
Last Failure
Last Duration
Built On
CentOS_9-stream_aarch64_kvm
5 hr 48 min
#205
5 days 19 hr
#201
33 min
Jenkins
CentOS_9-stream_all
20 days
#169
5 hr 48 min
#183
33 min
Jenkins


https://testing.stream.centos.org
Jenkins
Search (⌘+K)
log in
Dashboard
c10s - compose
Build History
This is running the TMT tests maintained by the Integration SIG:
https://gitlab.com/CentOS/Integration/compose-tests
Learn how to contribute and/or join the SIG: https://sigs.centos.org/integration/
Build Queue
No builds in the queue.
All
c10s - compose
c9s - compose
Build Executor Status
1 Idle
2 Idle
3 Idle
4 Idle
5 Idle
S
W
Name
Last Success
Last Failure
Last Duration
Built On
CentOS_10-stream_aarch64_kvm
20 days #24
5 days 5 hr #34
32 min
Jenkins
CentOS_10-stream_all
2 mo 6 days #3
5 days 5 hr #32
9 min 10 sec
Jenkins


https://testing.stream.centos.org
Dashboard
c10s - compose
1 Idle
2 Idle
3 Idle
4 Idle
5 Idle
6 Idle
7 Idle
8 Idle
9 Idle
10 Idle
11 Idle
12 Idle
13 Idle
CentOS_10-stream_aarch64_kvm
20 days #24
5 days 5 hr #34
32 min
Jenkins
CentOS_10-stream_all
2 mo 6 days #3
5 days 5 hr #32
9 min 10 sec
Jenkins
CentOS_10-stream_ppc64le_kvm
2 mo 2 days #6
5 days 5 hr #37
6 min 56 sec
Jenkins
CentOS_10-stream_x86_64_kvm
2 mo 2 days #6
5 days 5 hr #35
5 min 54 sec
Jenkins
CentOS_10-stream_x86_64_metal
2 mo 6 days #4
5 days 5 hr #35
7 min 20 sec
Jenkins

https://sig.centos.org/integration

CentOS Integration SIG

Search

integration/docs
0 2

CentOS Integration SIG
Overview
- Introduction
- Purpose of the SIG
  - Goals
  - Deliverables
  - First work items
- How to join
- Questions

Architecture

Compose Tests

Overview

Introduction

Integration is verifying that products and services built on top of RHEL or CentOS Stream will continue to work on CentOS Stream and the next release of RHEL and will not break on package updates.

As RHEL content becomes available only after the release, RHEL-based services traditionally use a *catching-up integration* pattern: people have to adjust their products and services to work on new RHEL after the update is shipped. Adjusting the services takes time, eating into the supported RHEL lifecycle period. It also reduces the options for how we can deal with breaking changes.

CentOS Stream provides a way to enable *forward-looking integration*: you can do the integration early during the development before the change is shipped to the CentOS Stream or RHEL repositories. This allows us to prevent or at least prepare better for any breaking changes, which might be shipped via CentOS Stream or RHEL updates.

Purpose of the SIG

Provide a shared space to develop and maintain tooling and knowledge base on collaborative gating and testing of CentOS Stream updates before they are published to CentOS mirrors. This includes both - package-level and compose-level integration.

GOALS

- Document existing integration workflows used by other SIGs.
- Identify common issues.

https://sig.centos.org/integration

CentOS Integration SIG

Search

integration/docs
0 2

CentOS Integration SIG

Overview

Architecture

Compose Tests

- Where to see the tests in action
- How to run the compose tests locally
- How to debug a specific test failure
- How to contribute

Compose Tests

Thanks to the work done by Carlos Rodriguez-Fernandez the compose tests have been ported from the t_functional test wrapper to the tmt (Test Management Tool).

You can find new tests repository in the Integration SIG namespace on Gitlab.com:

- https://gitlab.com/CentOS/Integration/compose-tests

Where to see the tests in action

For CentOS Stream 9 these tests run in the experimental Jenkins job:

- https://testing.stream.centos.org/job/CentOS_9-stream_x86_64_kvm_Experimental%20-%20TMT%20test%20suite/

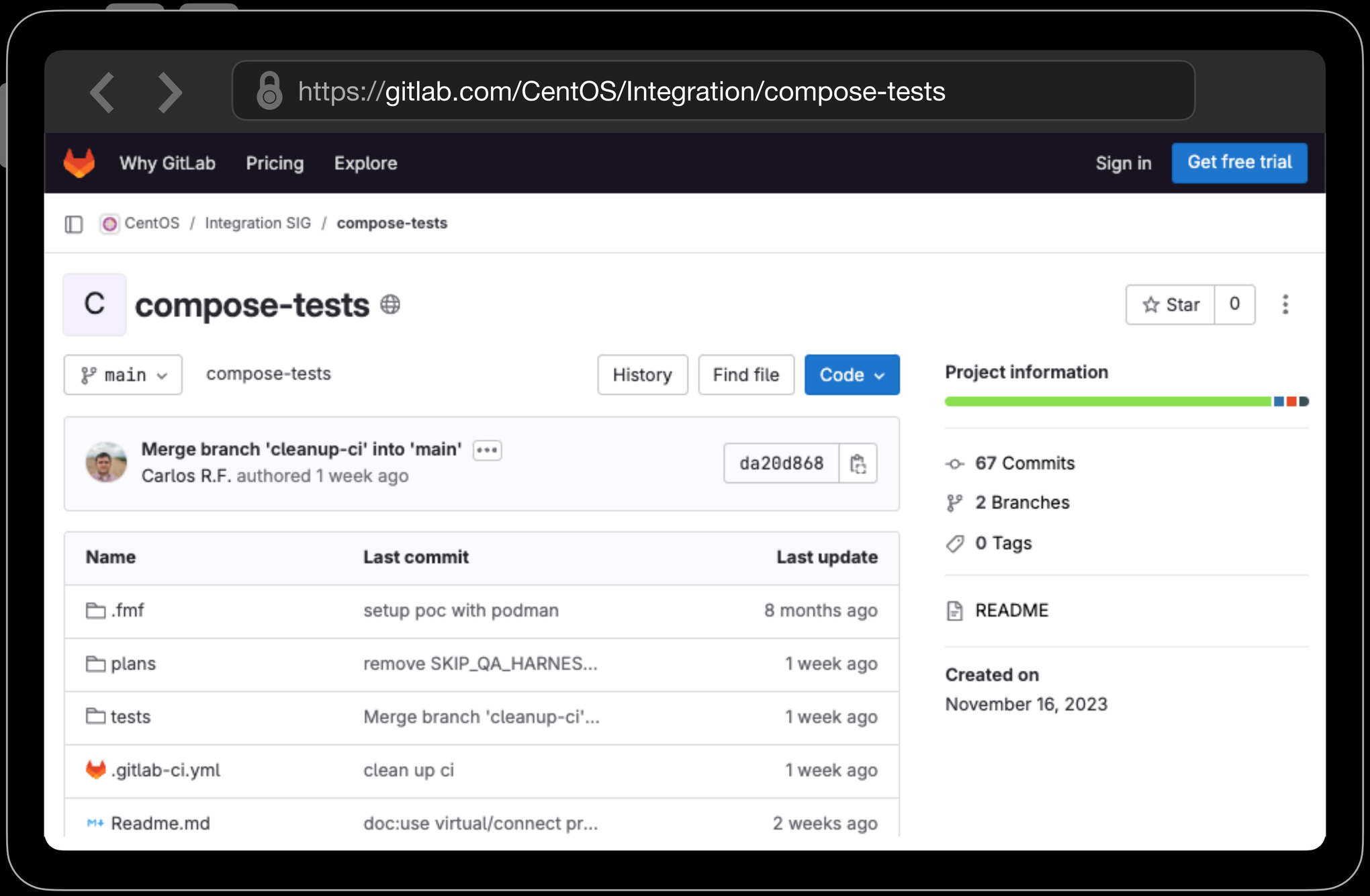
For CentOS Stream 10 we run the tests in the main Jenkins pipeline:

- https://testing.stream.centos.org/job/CentOS_10-stream_all/

How to run the compose tests locally

1. Install the tmt tool and tmt plugin with libvirt support:

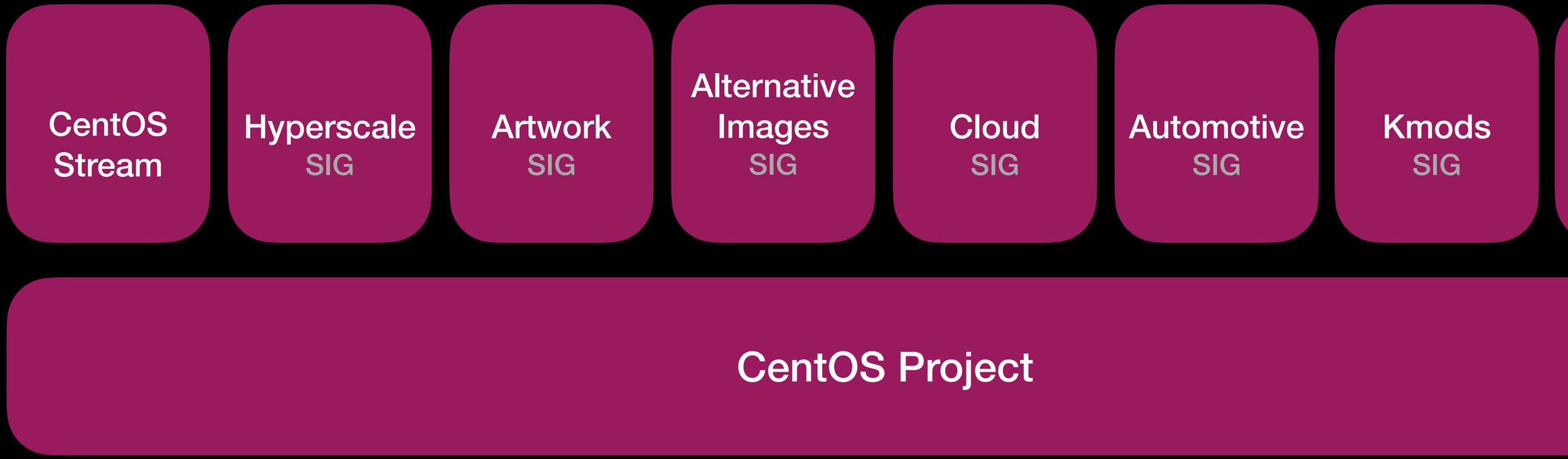
```
# dnf install tmt tmt+provision-virtual
```


https://gitlab.com/CentOS/Integration/compose-tests
Why GitLab
Pricing
Explore
Sign in
Get free trial
CentOS / Integration SIG / compose-tests
C
compose-tests
Star 0
main
compose-tests
History
Find file
Code
Merge branch 'cleanup-ci' into 'main'
Carlos R.F. authored 1 week ago
da20d868
Name
Last commit
Last update
.fmf
setup poc with podman
8 months ago
plans
remove SKIP_QA_HARNES...
1 week ago
tests
Merge branch 'cleanup-ci'...
1 week ago
.gitlab-ci.yml
clean up ci
1 week ago
Readme.md
doc:use virtual/connect pr...
2 weeks ago
Project information
67 Commits
2 Branches
0 Tags
README
Created on
November 16, 2023

CentOS

SIGs (special interest groups)

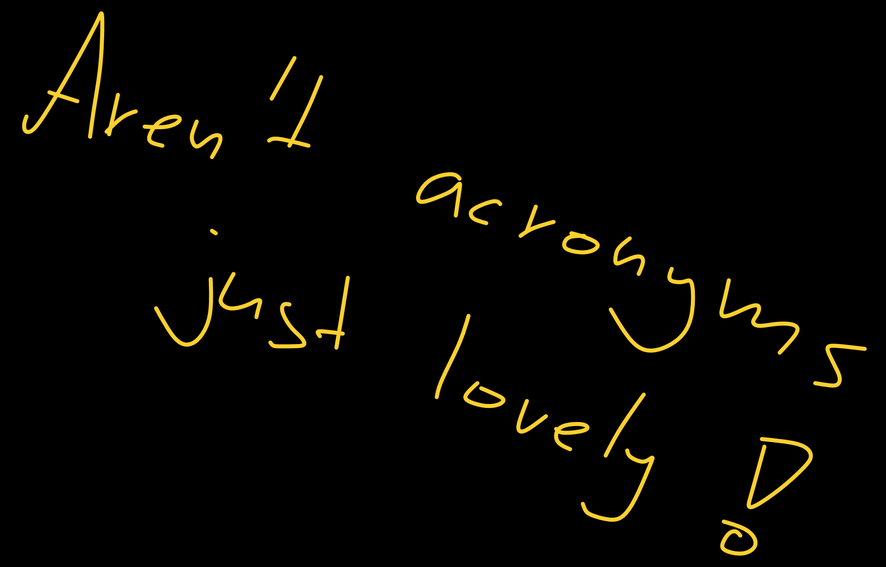
CentOS Project


CentOS Stream
Hyperscale SIG
Artwork SIG
Alternative Images SIG
Cloud SIG
Automotive SIG
Kmods SIG
CentOS Project

CentOS (the project) has two build systems:

1) CS Koji

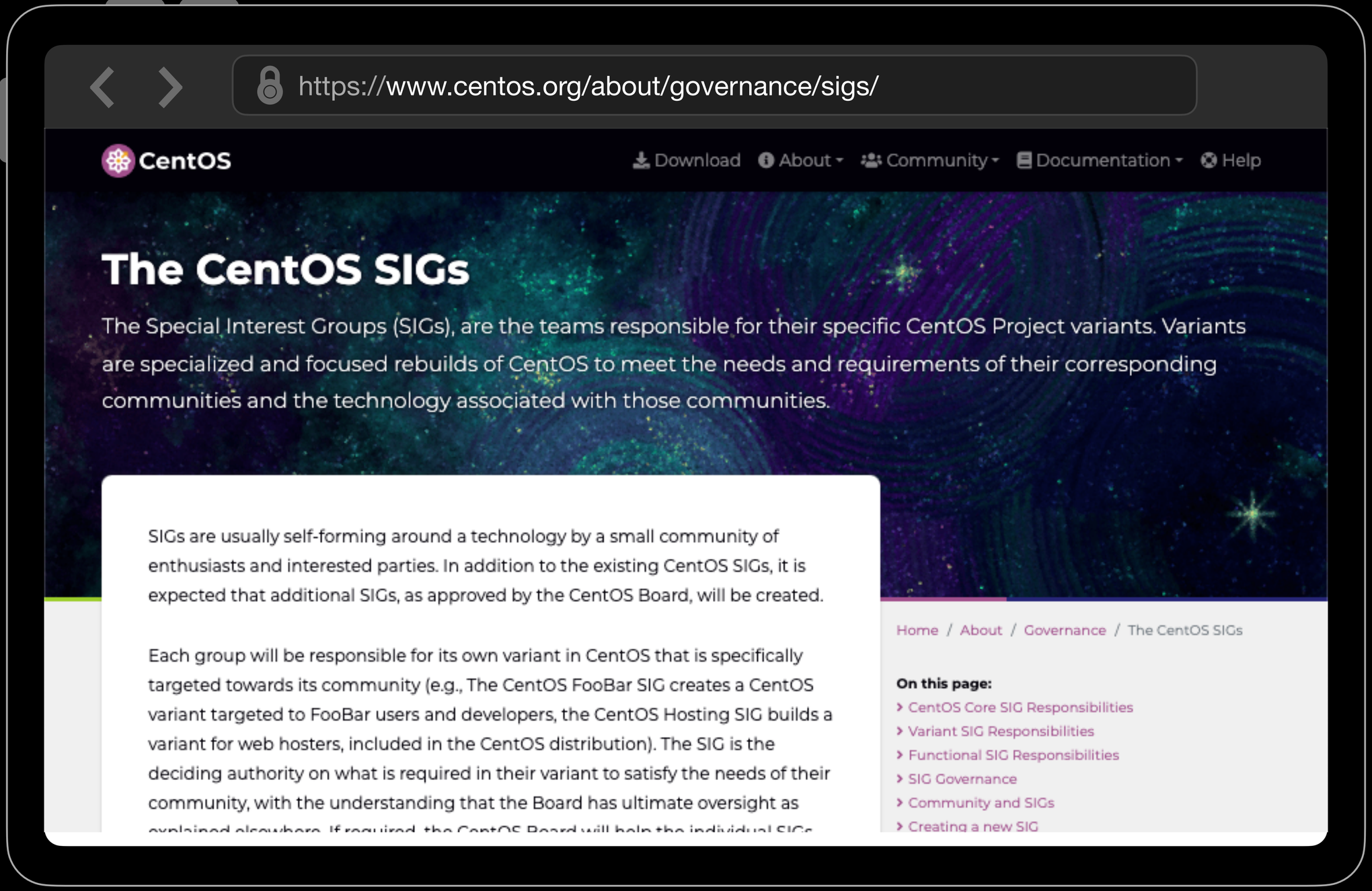
2) CBS Koji



CentOS (the project) has two build systems:

1) CentOS Stream Koji
   - read-only outside Red Hat

2) Community Build System (Koji)
   - this is where CentOS SIGs build

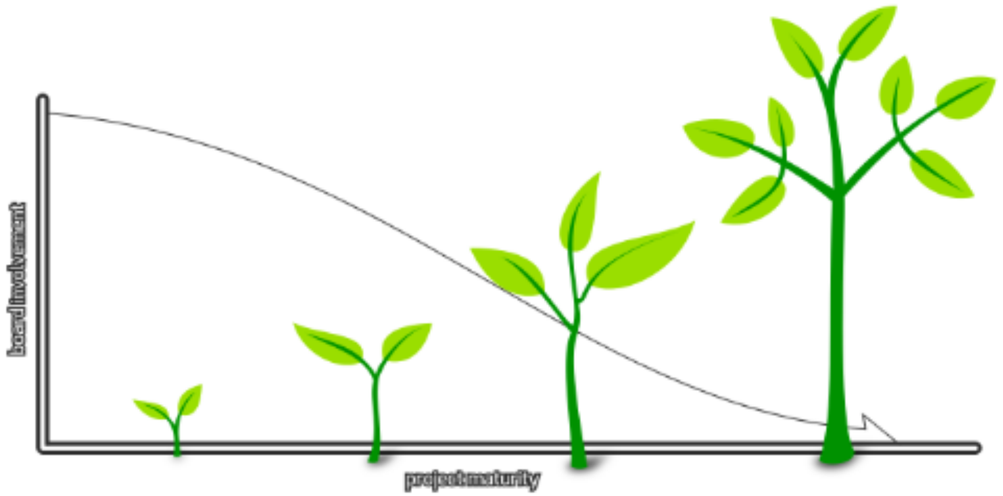

https://www.centos.org/about/governance/sigs/
CentOS
Download
About
Community
Documentation
Help
The CentOS SIGs
The Special Interest Groups (SIGs), are the teams responsible for their specific CentOS Project variants. Variants are specialized and focused rebuilds of CentOS to meet the needs and requirements of their corresponding communities and the technology associated with those communities.
SIGs are usually self-forming around a technology by a small community of enthusiasts and interested parties. In addition to the existing CentOS SIGs, it is expected that additional SIGs, as approved by the CentOS Board, will be created.
Each group will be responsible for its own variant in CentOS that is specifically targeted towards its community (e.g., The CentOS FooBar SIG creates a CentOS variant targeted to FooBar users and developers, the CentOS Hosting SIG builds a variant for web hosters, included in the CentOS distribution). The SIG is the deciding authority on what is required in their variant to satisfy the needs of their community, with the understanding that the Board has ultimate oversight as
Home / About / Governance / The CentOS SIGs
On this page:
CentOS Core SIG Responsibilities
Variant SIG Responsibilities
Functional SIG Responsibilities
SIG Governance
Community and SIGs
Creating a new SIG

https://www.centos.org/about/governance/sigs/

CentOS Download About Community Documentation Help

Home / About / Governance / The CentOS SIGs

SIG Governance



On this page:

- CentOS Core SIG Responsibilities
- Variant SIG Responsibilities
- Functional SIG Responsibilities
- SIG Governance
- Community and SIGs
- Creating a new SIG
- Retiring a SIG

The SIGs themselves also have a merit path toward autonomy and accountability for Project aspects. The determination of merit level is reflected in the amount of oversight required by the Board and the SIGs ability to self-sign and release software builds. As merit increases, Board oversight goes down, with a transition spot in the middle where the SIG naturally obtains more autonomy, usually toward the end of the "Early" phase.

https://www.centos.org/about/governance/sigs/

CentOS Download About Community Documentation Help

Creating a new SIG

The process of creating a new SIG involves two major components: community building and the administrative side.

Bring your SIG proposal first to the centos-devel mailing list to find other like-minded people who wish to start the SIG with you. Also look around outside of the CentOS project for people who may want to distribute projects on CentOS

Once you have a core group that wants to make this happen, open a ticket on the board issue tracker with your proposed SIG, and someone there will walk you through the process.

For the current list of active SIGs, refer to
http://wiki.centos.org/SpecialInterestGroup

Retiring a SIG

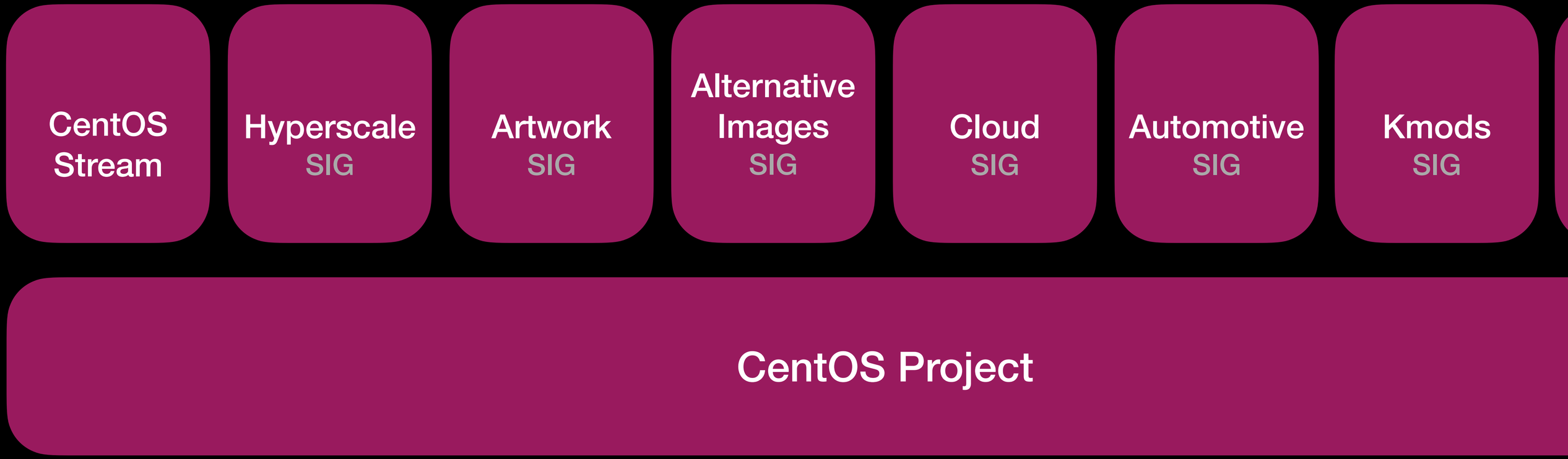
If a SIG misses two of their quarterly reports in a row: then the community manager should contact the members listed in the account system. More than a

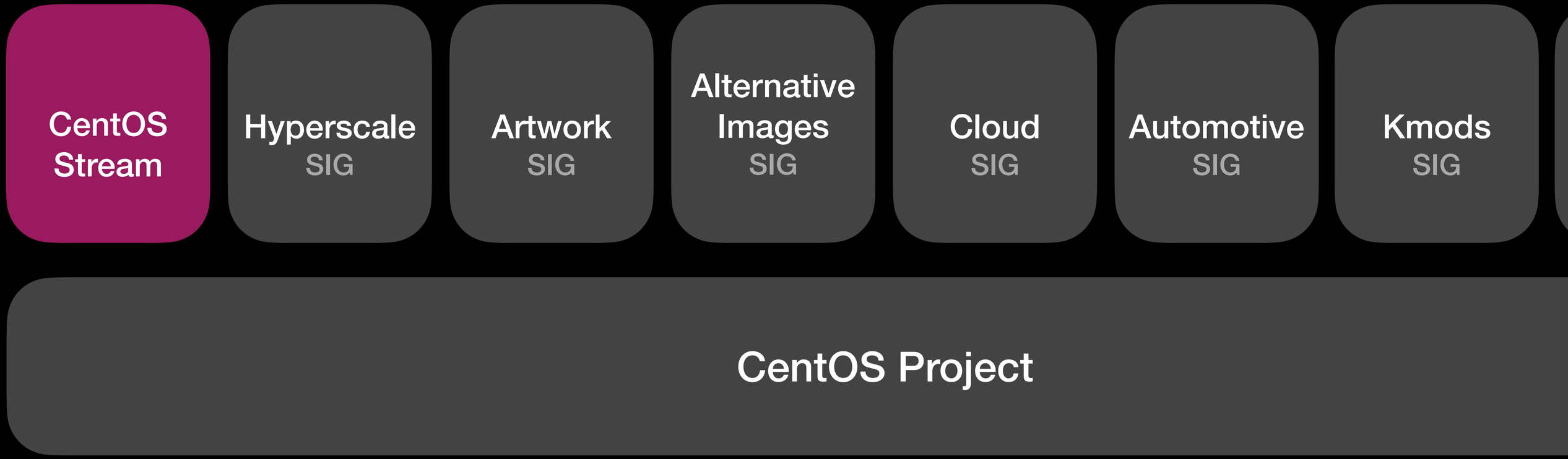
On this page:

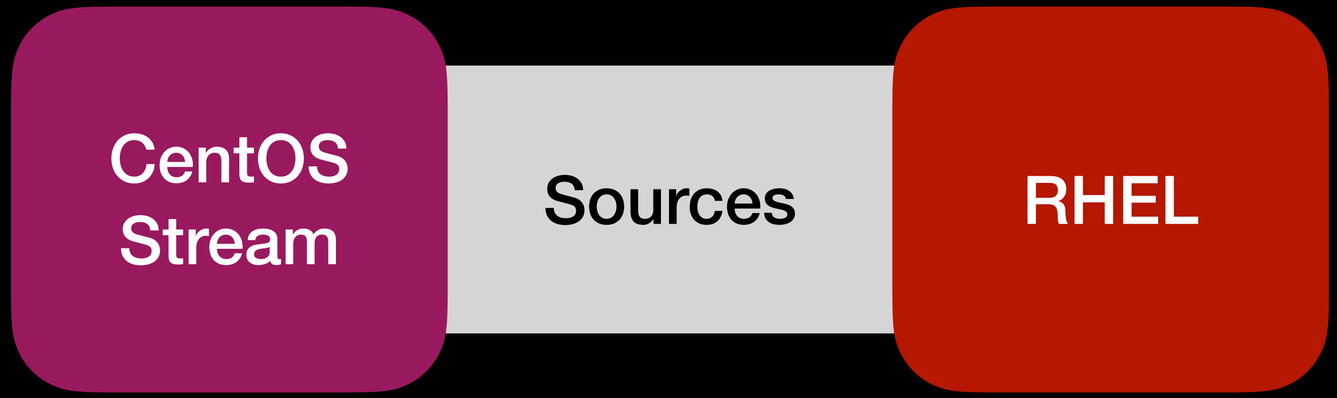
- CentOS Core SIG Responsibilities
- Variant SIG Responsibilities
- Functional SIG Responsibilities
- SIG Governance
- Community and SIGs
- Creating a new SIG
- Retiring a SIG

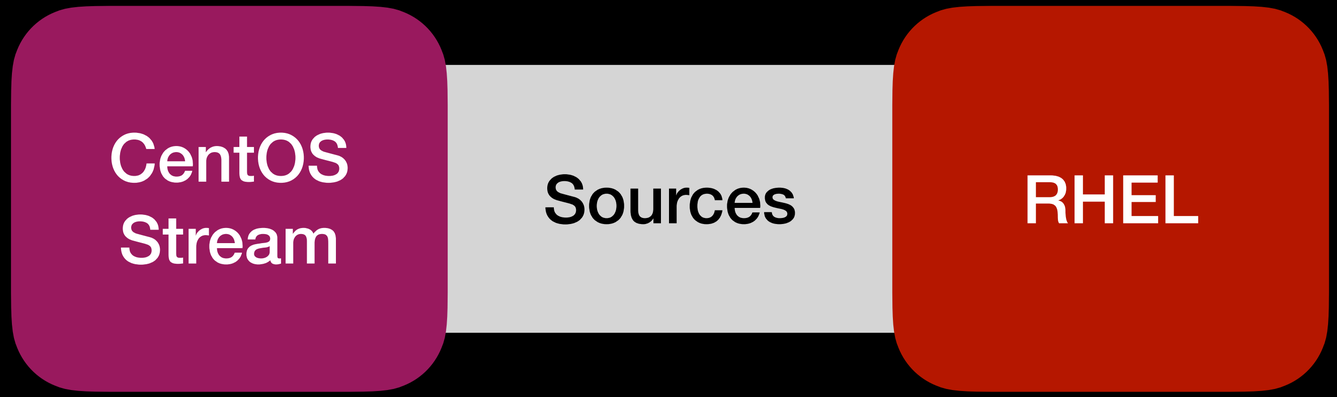
CentOS Stream

Contributions


CentOS Stream
Hyperscale SIG
Artwork SIG
Alternative Images SIG
Cloud SIG
Automotive SIG
Kmods SIG
CentOS Project


CentOS Stream
Hyperscale SIG
Artwork SIG
Alternative Images SIG
Cloud SIG
Automotive SIG
Kmods SIG
CentOS Project


CentOS
Stream
Sources
RHEL



Application Compatibility Guide (ACG)

https://access.redhat.com/articles/rhel9-abi-compatibility

Subscriptions | Downloads | Red Hat Console | Get Support

Red Hat Customer Portal

Menu

Search | English | All Red Hat | Log In

Products & Services > Knowledgebase > Red Hat Enterprise Linux 9: Application Compatibility Guide

Red Hat Enterprise Linux 9: Application Compatibility Guide

Updated May 25 2022 at 1:05 PM - English

Red Hat Enterprise Linux 9:
Application Compatibility Guide
May 2022

Note: This article discusses application compatibility for Red Hat Enterprise Linux 9. For Red hat Enterprise Linux 8, please see the Red Hat Enterprise Linux 8: Application Compatibility GUIDE.

https://access.redhat.com/articles/rhel9-abi-compatibility

Terminology

The following are basic terms used in this document:

- **Application programming interface (API)**
  An API is a public interface implemented by a software program that enables it to interact with other software, including operating system components. The API is enforced at compile time and determines source compatibility, that is, whether application source code will compile similarly across different versions of the operating environment. It covers details such as:
  - C and C++ headers for developer use
  - Language syntax, but only where it applies to a published standard
  - Public interface definitions
  - Command line interfaces, but only where it applies to documented public interfaces
- **Application binary interface (ABI)**
  An ABI is a set of runtime conventions that interact with a compiled binary representation of a program. The ABI is enforced at runtime and it describes the low-

- **ABI conformance**
  A compiler conforms to an ABI if it generates code that follows all of the specifications enumerated by that ABI. A library conforms to an ABI if it is implemented according to that ABI. An application conforms to an ABI if it is built using tools that conform to that ABI and does not contain source code that changes behavior specified by the ABI or that otherwise bypasses the ABI.

- **Binary compatibility**
  Binary compatibility means application binaries that are compiled for a specific ABI, generally for a combination of RHEL and a particular hardware architecture, will load and run similarly across different versions of RHEL. Application binaries consist of executable files, Dynamic Shared Objects (DSO), source, bytecode for interpreted just-in-time compiled languages, and their required data files.

- **Core persistent system infrastructure**
  The core persistent system infrastructure refers to interfaces and externally available data structures that represent system state or provide a means of communicating with the system (for instance, system calls and header files).

- **Compatibility in a virtualized environment**
  Virtual environments emulate bare-metal environments such that unprivileged

https://access.redhat.com/articles/rhel9-abi-compatibility

Compatibility levels

All components and packages in Red Hat Enterprise Linux are classified under one of the following four compatibility levels:

- **Compatibility level 1**
  - APIs and ABIs are stable within the lifetime of a major release and ABIs are also stable across the next two major releases; the release that introduces a new or revised ABI, and the two following major releases (n, n+1, n+2). In the case of this document, release n starts with Red Hat Enterprise Linux 9. If a change to a library causes an incompatibility with existing binaries, a separate version of the library will be provided with the older ABI to run the application without modification.
- **Compatibility level 2**
  - APIs and ABIs are stable within the lifetime of a single major release. Compatibility level 2 application interfaces will not change from minor release to minor release and can be relied upon by the application to be stable for the duration of the major release. Compatibility level 2 is the default for packages in Red Hat Enterprise Linux 9. Packages not identified as having another compatibility level may be considered compatibility level 2.

https://access.redhat.com/articles/rhel9-abi-compatibility

Compatibility exceptions

The following are exceptions to compatibility in RHEL.

SystemTap static probes

- No assurances are made at this time that integrated SystemTap static probes will continue to have the same probe name, probe location, or interpretation or number of arguments. Since the probes are primarily designed for deep analysis and debugging, the probes must be able to change as the underlying implementation changes.

Static linking with the C/C++ runtime

- Static linking with the C/C runtime is not supported. This includes linking with any files that are part of the `glibc-static` or `libstdc`-static packages. You may choose to link statically, but the resulting application binaries may fail to operate if any package in the installation is changed.

C/C++ application sanitizers

- C/C++ applications built with the compiler option `-fsanitize=[option]` cannot participate in the API or ABI guarantees provided in this document. The sanitizer

https://access.redhat.com/articles/rhel9-abi-compatibility

Guidelines for preserving binary compatibility

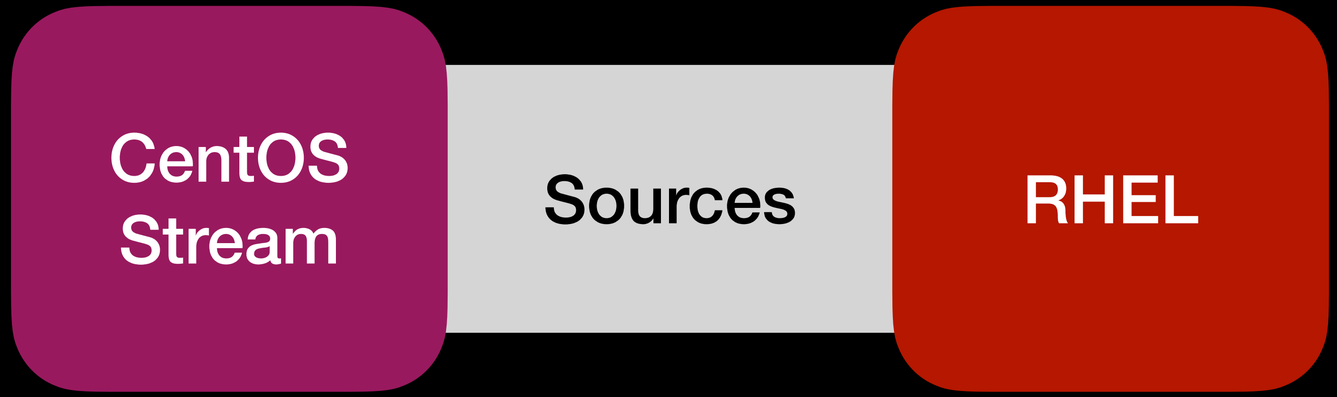
Red Hat recommends that application developers adopt the following principles in order to improve binary compatibility:

1. Use only libraries and applications listed in the compatibility level that suits your application needs.

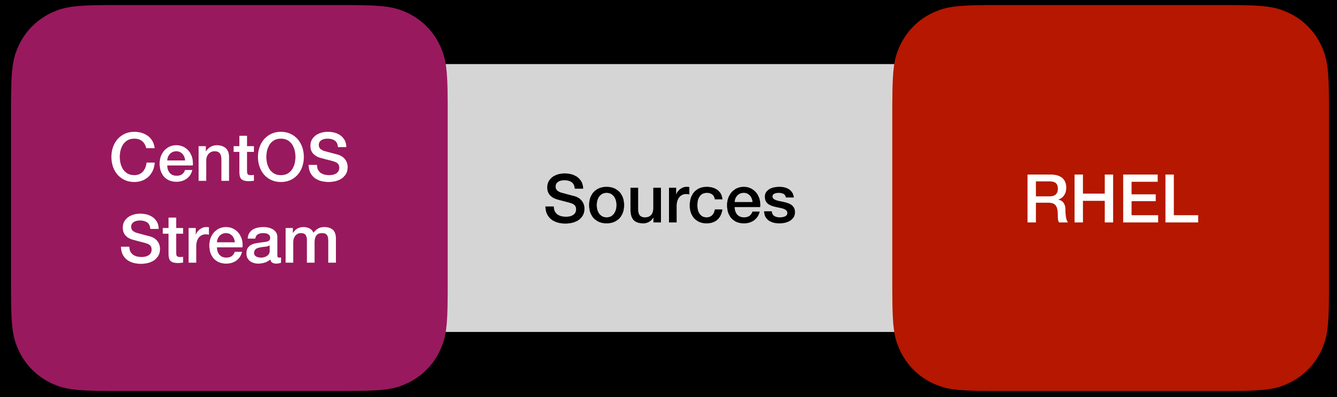
2. Build applications using the published interfaces of a library. Non-published (internal) interfaces are subject to change at any time, which can cause instability in the dependent application if relied upon.

    - If the library provides a development package, you must install and use that development package, including any provided headers, for your own development. If the library does not provide a development package or does not provide a set of headers and shared object for linking, then the API and the ABI of that library is not covered by any compatibility guarantees.

    - You must link with all libraries that are required by your application. Failing to link against all required libraries is called "underlinking." An underlinked application cannot take advantage of ABI compatibility.

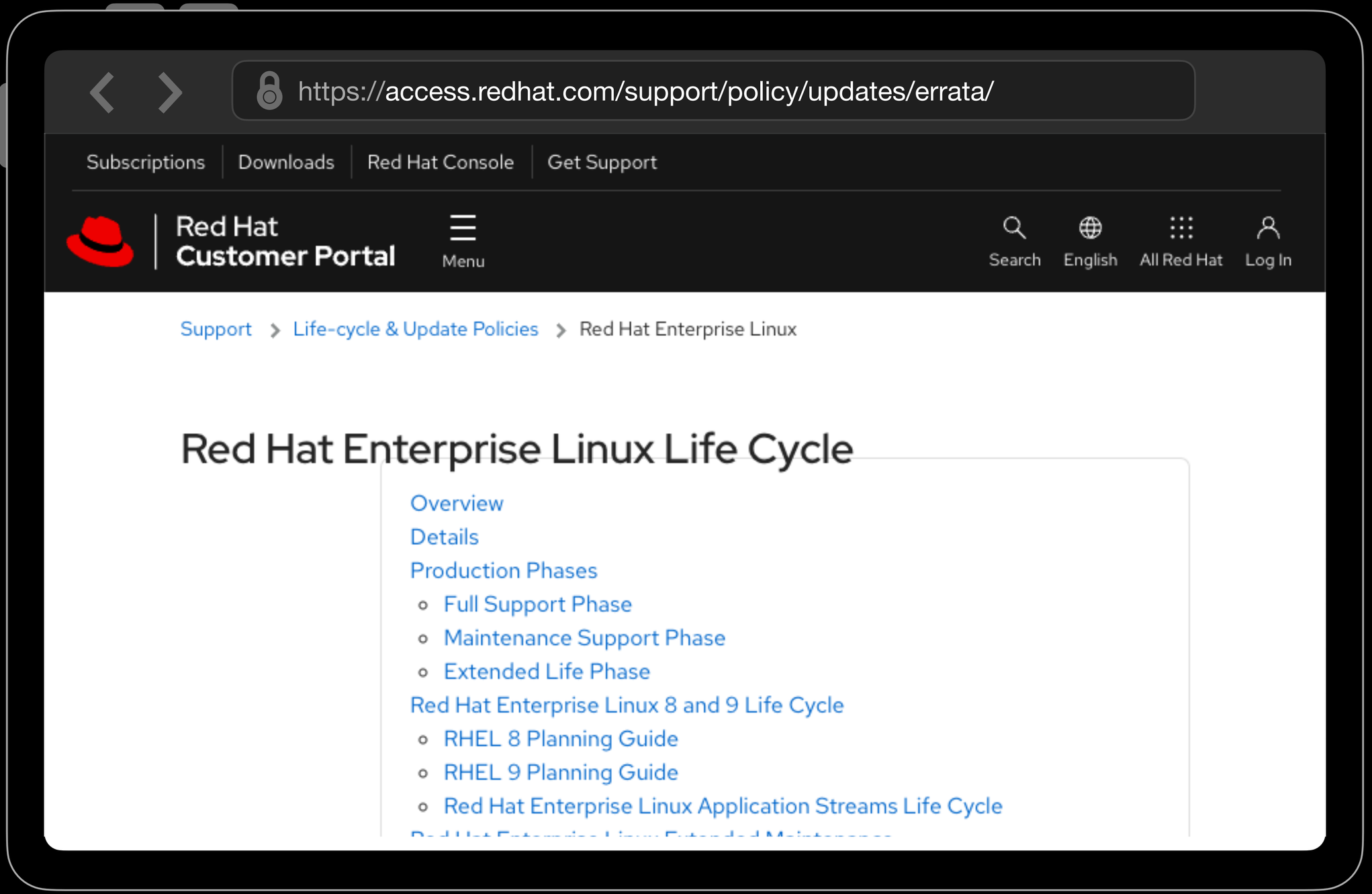


Application Compatibility Guide (ACG)



Application Compatibility Guide (ACG)

Red Hat Enterprise Linux Life Cycle


https://access.redhat.com/support/policy/updates/errata/
Subscriptions
Downloads
Red Hat Console
Get Support
Red Hat
Customer Portal
Menu
Search
English
All Red Hat
Log In
Support
Life-cycle & Update Policies
Red Hat Enterprise Linux
Red Hat Enterprise Linux Life Cycle
Overview
Details
Production Phases
Full Support Phase
Maintenance Support Phase
Extended Life Phase
Red Hat Enterprise Linux 8 and 9 Life Cycle
RHEL 8 Planning Guide
RHEL 9 Planning Guide
Red Hat Enterprise Linux Application Streams Life Cycle

https://access.redhat.com/support/policy/updates/errata/

Red Hat Enterprise Linux Version 8 and 9

Red Hat understands the importance of life cycle planning for our customers, partners, ISVs, and the Red Hat Enterprise Linux ecosystem. With the introduction of Red Hat Enterprise Linux version 8, Red Hat simplified the RHEL product phases from four to three: Full Support, Maintenance Support, and Extended Life Phase. We are also providing information on anticipated release dates and minor releases that will offer extended support.

- Red Hat Enterprise Linux Version 8 and 9 delivers a ten year life cycle in Full Support and Maintenance Support Phases followed by an Extended Life Phase. In addition, Red Hat Enterprise Linux 8 and 9 customers may purchase annual Add-on subscriptions called Extended Life-cycle Support (ELS) to extend limited subscription services beyond the Maintenance Support Phase.

Life Cycle[viii]:

https://access.redhat.com/support/policy/updates/errata/

Red Hat Enterprise Linux Life Cycle

Description	Full Support	Maintenance Support	Extended Life Phase[7]	Extended Life Cycle Support (ELS) Add-On[8]	Extended Update Support (EUS) Add-On[8] Enhanced Extended Update Support (Enhanced EUS) Add-On[8]
Access to Previously Released Content through the Red Hat Customer Portal	Yes	Yes	Yes	Yes	Yes
Self-help through the	Yes	Yes	Yes	Yes	Yes

https://access.redhat.com/support/policy/updates/errata/

Technical Support[1]	Unlimited	Unlimited	Limited[9]	Unlimited	Unlimited
Asynchronous Security Errata (RHSA)[10 11]	Yes	Yes	No	Yes[8]	Yes[8]
Asynchronous Bug Fix Errata (RHBA)[2 11]	Yes	Yes	No	Yes	Yes
Minor Releases	Yes	No	No	No	No
Refreshed Hardware Enablement[3]	Native	Using Virtualization	Using Virtualization	Using Virtualization	Using Virtualization
Software Enhancements[5]	Yes[6]	No	No	No	No
Updated Installation Images	Yes	Yes[14]	No	No	No

https://access.redhat.com/support/policy/updates/errata/

Red Hat Enterprise Linux Production Phases

Full Support Phase[iv]:

During the Full Support Phase, Red Hat defined Critical and Important Security errata advisories (RHSAs) and Urgent and Selected (at Red Hat discretion) High Priority Bug Fix errata advisories (RHBAs) will be released as they become available. Other errata advisories may be delivered as appropriate.

If available, new or improved hardware enablement and select enhanced software functionality may be provided at the discretion of Red Hat, generally in minor releases. Hardware enablement that does not require substantial software changes may be provided independent from minor releases at Red Hat's discretion.

Minor releases will also include available and qualified errata advisories (RHSAs, RHBAs, and RHEAs). Minor releases are cumulative and include the contents of previously released updates. The focus for minor releases during this phase lies on resolving defects of medium or higher priority.

Updated installation images will be provided for minor releases during the Full Support Phase.

Maintenance Support Phase :

https://access.redhat.com/support/policy/updates/errata/

Maintenance Support Phase :

During the Maintenance Support Phase for Red Hat Enterprise Linux Version 8 & 9 Red Hat defined Critical and Important[ix] impact Security Advisories (RHSAs) and selected (at Red Hat discretion) Urgent Priority Bug Fix Advisories (RHBAs) will be released as they become available. Other errata advisories may be delivered as appropriate.

New functionality and new hardware enablement are not planned for availability in the Maintenance Support Phase.

Extended Life Phase:

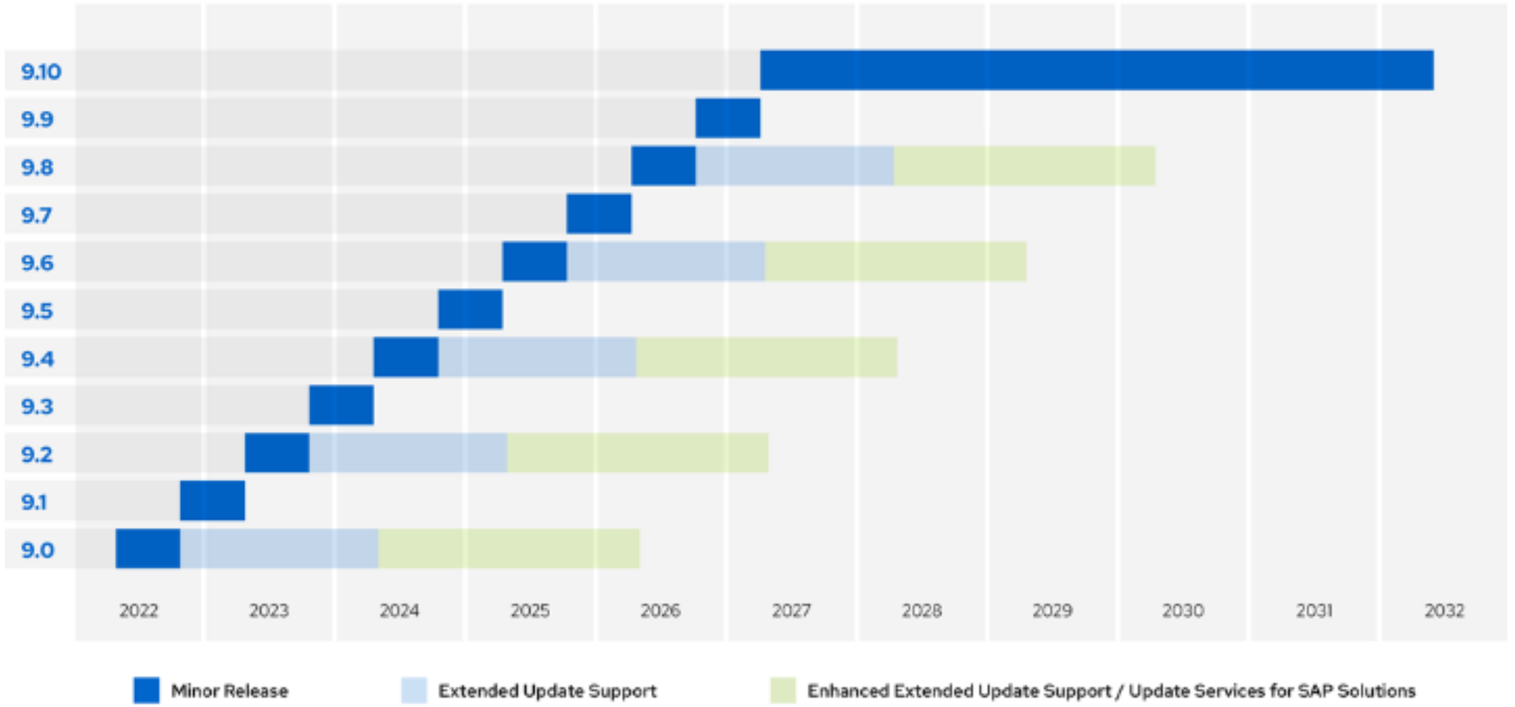
During the Extended Life Phase, a Red Hat Enterprise Linux subscription provides continued access to previously released content on the Red Hat Customer Portal, as well as other content such as documentation and the Red Hat Knowledgebase. Advice for migrating to currently supported Red Hat Enterprise Linux versions may also be provided.

For versions of products in the Extended Life Phase, Red Hat will provide limited ongoing technical support. No bug fixes, security fixes, hardware enablement or root-cause analysis will be available during this phase, and support will be provided on existing installations only.

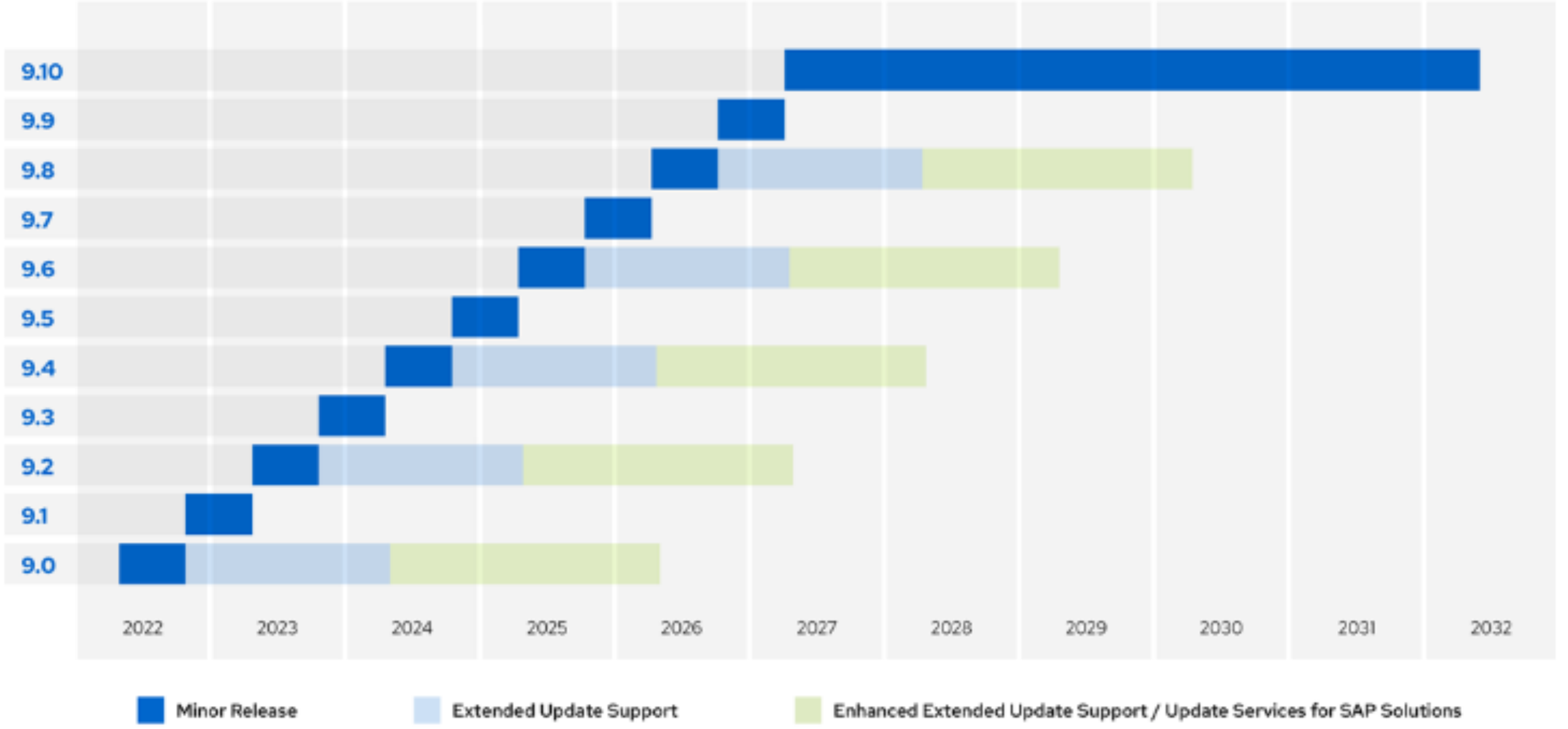
Red Hat reserves the right to terminate the ongoing support in the Extended Life Phase for a

https://access.redhat.com/support/policy/updates/errata/

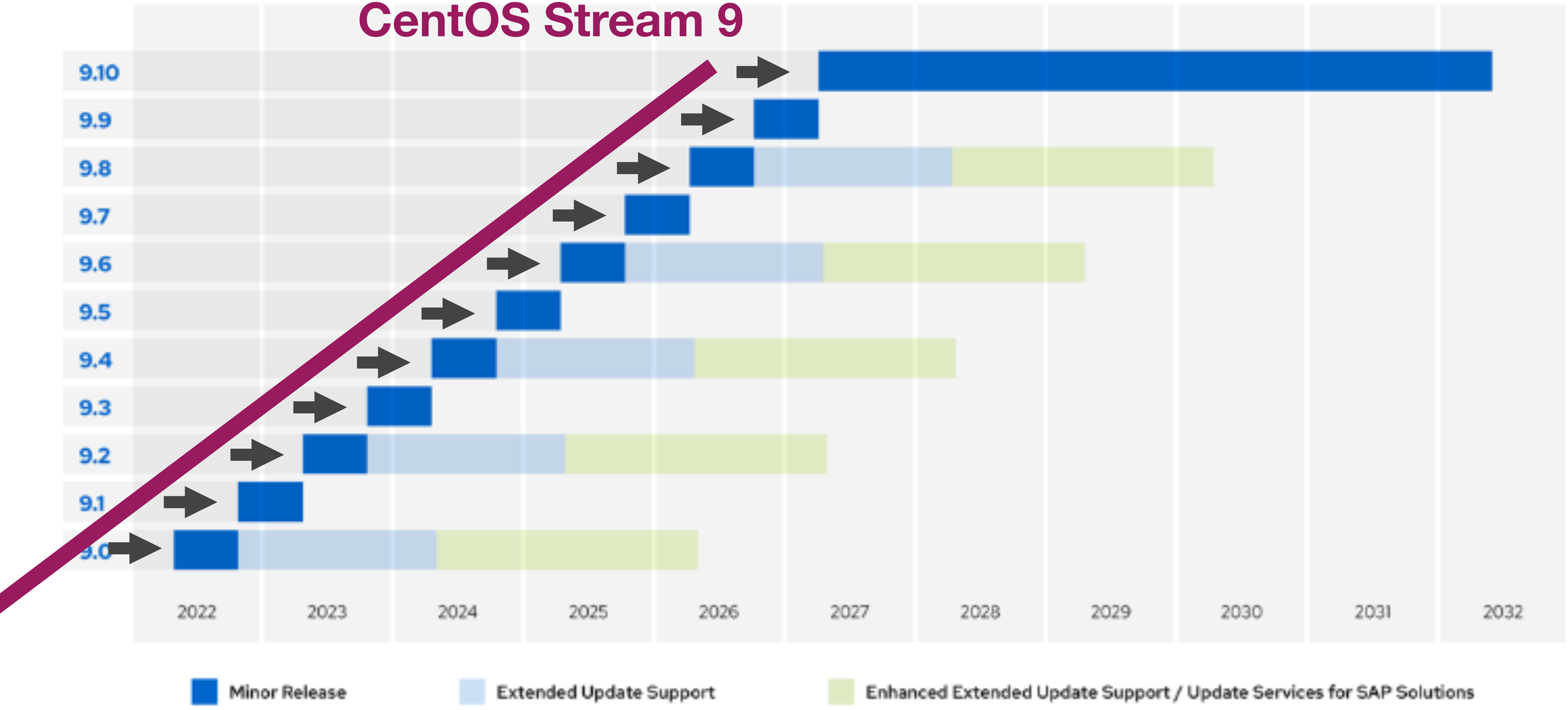
RHEL 9 Planning Guide[viii]

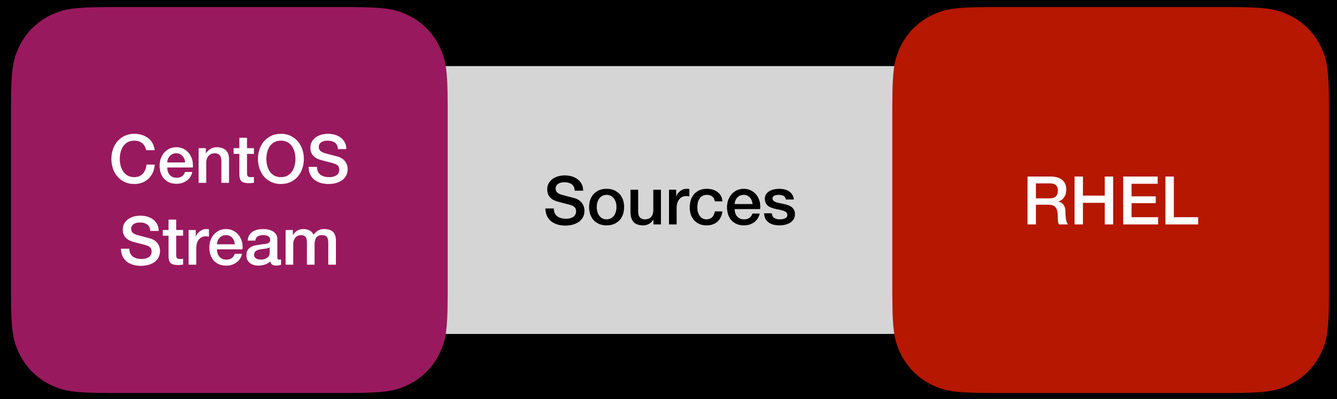


RHEL 9 Planning Guide[viii]



RHEL 9 Planning Guide[viii]





Application Compatibility Guide (ACG)

Red Hat Enterprise Linux Life Cycle

Steps:

1) Jira ticket - talk to the maintainer first
2) Merge Request
3) Maintainer decides whether
   and when to merge

bug fixes

"stable" updates
from upstream

backported features

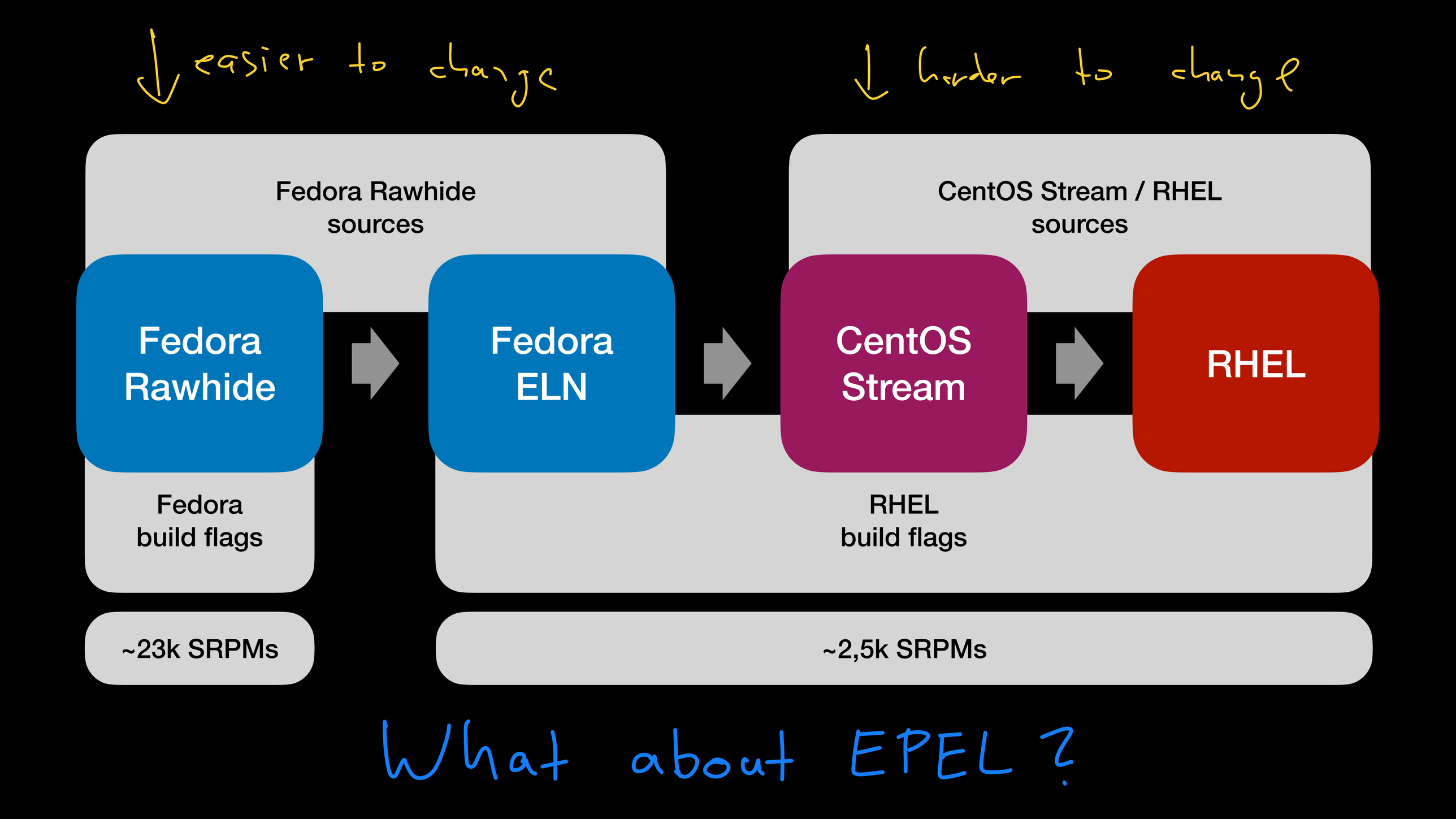
ABI-incompatible updates

docs, typos,
man pages

more disruptive
updates

It all starts in

Fedora


easier to change
harder to change
Fedora Rawhide sources
CentOS Stream / RHEL sources
Fedora Rawhide
Fedora ELN
CentOS Stream
RHEL
Fedora build flags
RHEL build flags
~23k SRPMs
~2,5k SRPMs
What about EPEL?

Fedora Project

Innovation in the OS itself

The latest & greatest

6 / 13 month lifecycle

CentOS Project

Innovation on top of the OS - SIGs!

OS maintained by RHEL engineers - CentOS Stream

3 / 5 -ish year lifecycle

CS is a preview of RHEL - use it in CI

Thank you!

CentOS Stream

- preview of RHEL
- solid base for CentOS SIGs

https://docs.centos.org/en-US/stream-contrib/quickstart/

https://issues.redhat.com ('RHEL' project)

https://gitlab.com/redhat/centos-stream/rpms

https://kojihub.stream.centos.org

https://composes.stream.centos.org

https://testing.stream.centos.org

https://mirrors.stream.centos.org

https://asamalik.fedorapeople.org/2024-flock/