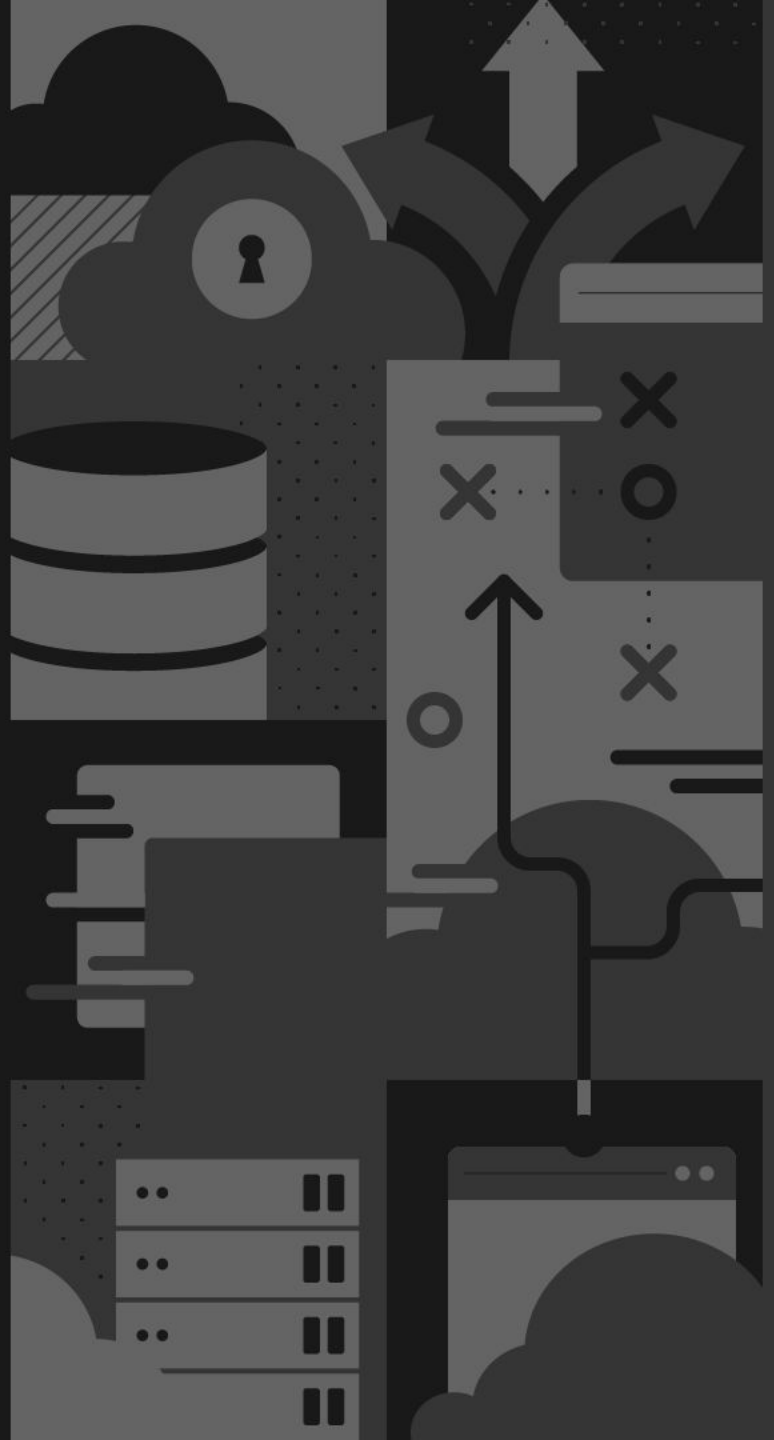


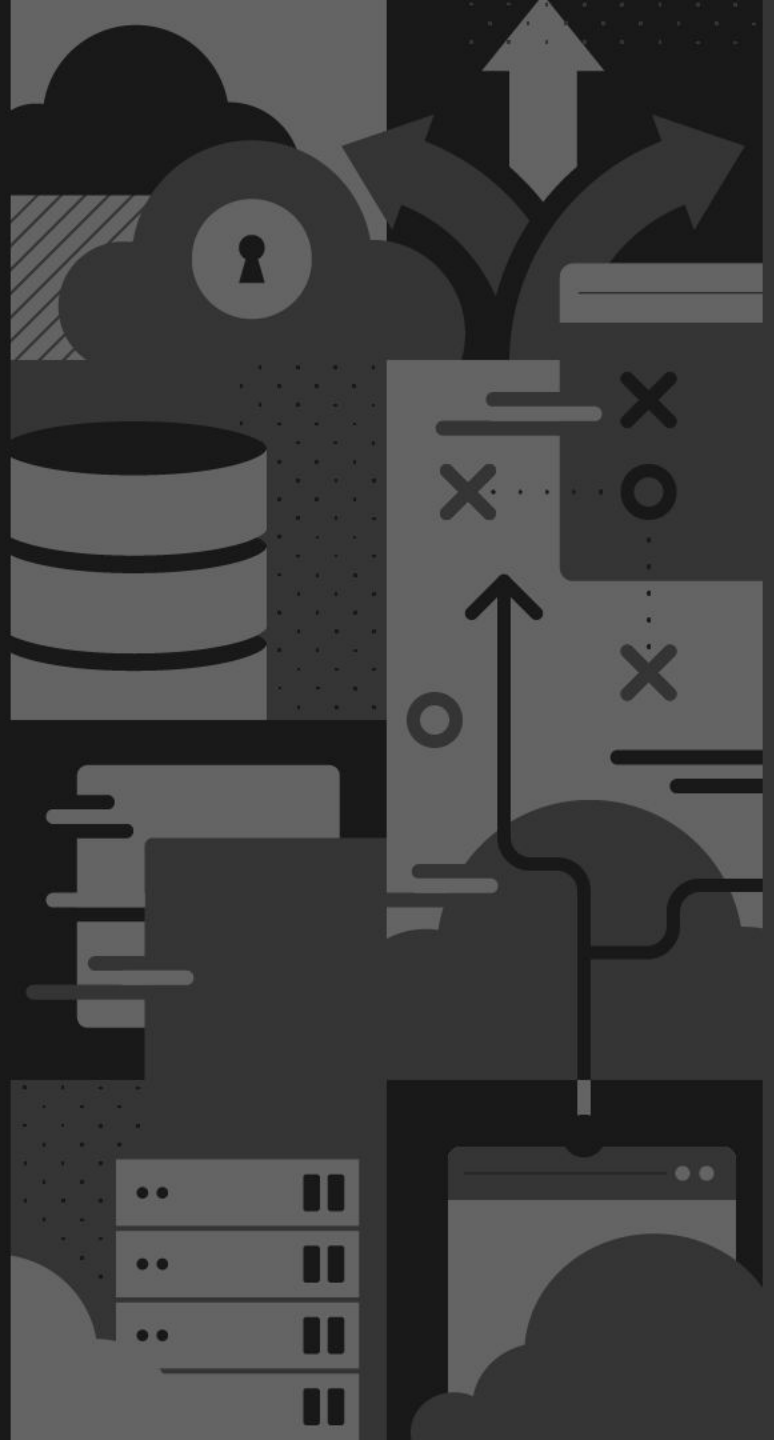
Infrastructure and Release Engineering

What goes into building Fedora Workstation?

Akashdeep Dhar
Community Linux Engineering
Fedora Council

Samyak Jain
Community Linux Engineering
Red Hat

Red Hat



People

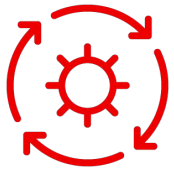
मानव संसाधन

उपखण्ड

Subclasses

Build system

Monitoring service deployment
Handling community infrastructure
Managing lifecycle of services



Ecosystem support

Collaborating on community projects
Developing solutions for infrastructure
Creating project-related decisions

Multiple utilizations

Performing release gating checks
Developing testcases with the community
Providing feedbacks on artifacts

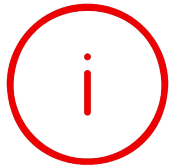
Source:
https://docs.fedoraproject.org/en-US/infra/
https://docs.fedoraproject.org/en-US/infra/day_to_day_fedora/
https://docs.fedoraproject.org/en-US/infra/overview/

सेवाएं

Services

Maintained by us, deployed by us

AAA services like Noggin and FASJSON
Infra tools like Poddlers and Mirror From Pagure
Packaging apps like Bodhi and RPMAutoSpec



Maintained by others, deployed by us

AAA services like IPA and Ipsilon
Community tools like Mailman3 and Wiki
Packaging apps like OSBS and Koji

Maintained by others, deployed by others

Packaging apps like COPR and OpenQA

Source:
https://docs.fedoraproject.org/en-US/infra/
https://docs.fedoraproject.org/en-US/infra/day_to_day_fedora/
https://docs.fedoraproject.org/en-US/infra/overview/

विविधता

Variety

Collaboration

Docs, Magazine and Wiki
Bugzilla, Calendar, Elections, Meetbot, Discussions etc.
Transtats and Weblate

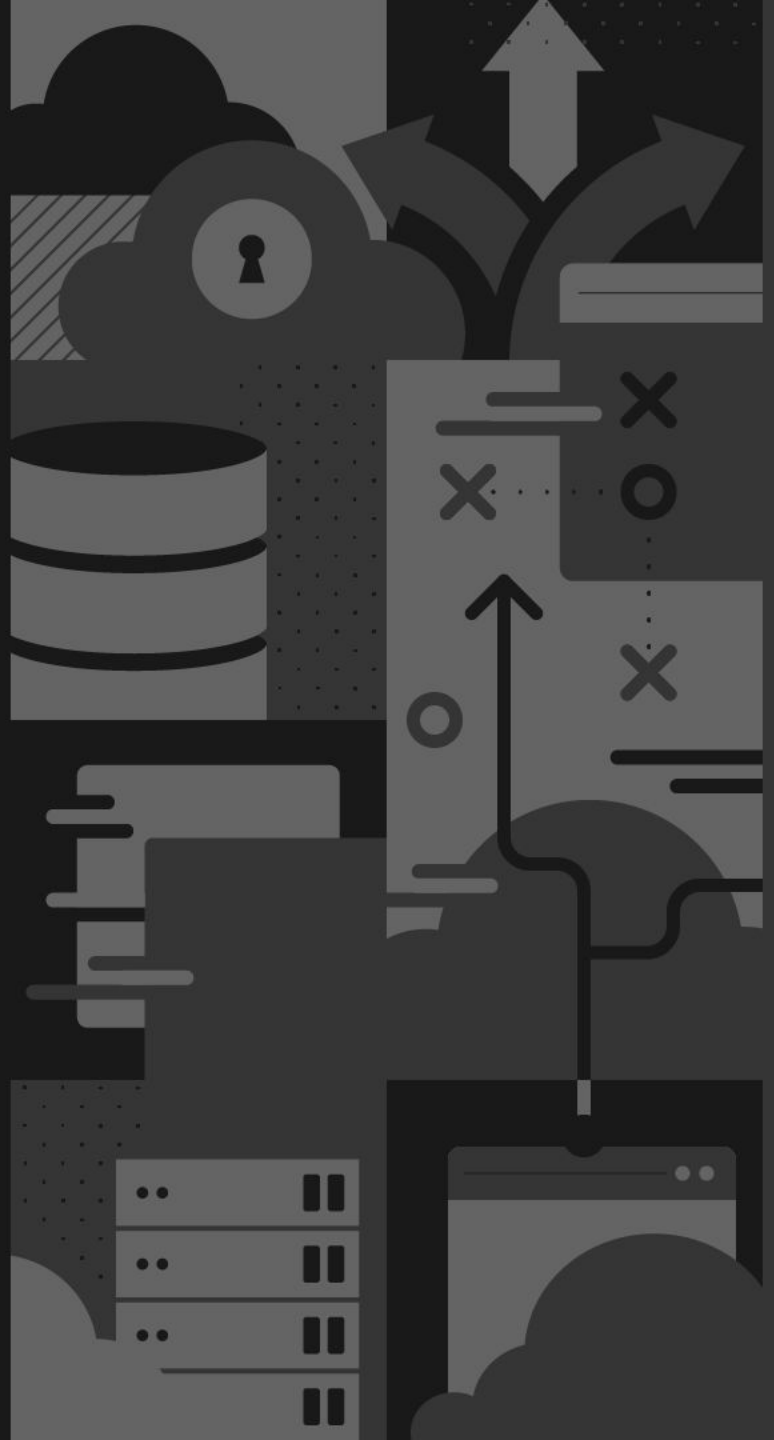
Artifacts

Bodhi, COPR, Koji, DistGit, Packages and MBS
Release Monitoring, Fedora CI and Mirror Manager
Retrace, Koschei, Blocker Bugs and Kernel Test

Miscellaneous

Accounts, Fedora People and Notifications
Datagrepper, MDAPI and Infrastructure Status
AskNot, Badges and Planet

Source:
https://docs.fedoraproject.org/en-US/infra/
https://docs.fedoraproject.org/en-US/infra/day_to_day_fedora/
https://docs.fedoraproject.org/en-US/infra/overview/



Processes

प्रक्रिया

fedora Red Hat

Planning and Scheduling

preparation ensures that every phase balances deadlines with quality

Rebuilding

All Fedora Linux packages are rebuilt with the latest system changes

Branching

A new branch for the release is created from Rawhide

Freezing

Stabilization phases for beta and final release candidates

Source:
https://docs.fedoraproject.org/en-US/infra/release_guide/

Fedora Linux aims to stay as close to the upstream **GNOME** packages as possible and hence the *biannual release cycle* of **Fedora Linux** plays well with the *biannual release cycle* of **GNOME**.

Mass Rebuild

Play well with recent system changes

Processes

- ***Stability***
  Detect issues with compatibility among system wide changes for addressing before release progresses further

- ***Optimization***
  Incorporate performance and consistency after packages are rebased upon updated compilers and tooling systems

Mass Rebuild

all artifacts are rebuilt with the latest system changes

#1 Initiation

Rebuild is triggered for artifacts in the Rawhide branch using Koji

#2 Addressal

Build logs help decide if failed packages are to be fixed or retired

#3 Completion

Successful builds are tagged appropriately for the upcoming branch

Source:
https://docs.fedoraproject.org/en-US/infra/release_guide/mass_rebuild/

मुख्य साधन

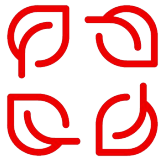
Koji

Build system

A free and open source artifact build service written in Python 3, Koji creates artifacts for the Fedora Linux using Mock to create chroot software environments.

Ecosystem support

There are interactions between ecosystem applications like Fedpkg and Pungi with Koji to facilitate both build jobs from individual packagers and release engineers.



Multiple utilizations

Koji is used globally in communities including but not limited to CERN, Ergo Project, NetNix Estonia, ACOSS France, Caltech, Amazon, XCP-ng, NIWA and Katello.

Source:
https://docs.pagure.org/koji/
https://koji.fedoraproject.org/koji/
https://pagure.io/koji/

Like every other *RPM packages* in **Fedora Linux**, a curated set of **GNOME** *RPM packages* for **Fedora Workstation** are built and tested with the use of side tags to ensure quality experience.

Mass Branching

Transitioning from Rawhide to Release

Why though?

- ***Development isolation***
  To separate the next release from ongoing development, thereby ensuring stability for the upcoming release

- ***Parallel workflows***
  To allow the work progress on the features planned for ongoing development while stabilizing the next release version

Mass Branching - Part 1

segregate the development version and release version for various purposes

#1
Preparation

Date is decided and tools are tested for smooth branch execution

#2
Creation

Fedpkg and Dist-Git help with the automating the created branch

#3
Tagging

Updated Koji tags reflect the created branch for the built packages

Source:
https://docs.fedoraproject.org/en-US/infra/release_guide/sop_mass_branching/

Mass Branching - Part 2

segregate the development version and release version for various purposes

#4
Configuration

Release keys and project repos are updated while configuring buildroots

#5
Communication

Maintainers and contributors are invited to work on the branch

#6
Testing

Superficial testing is done to catch residual issues on the branch

Source:
https://docs.fedoraproject.org/en-US/infra/release_guide/sop_mass_branching/

मुख्य साधन

Pagure

Home grown

A free and open source software git-centered forge written in Python 3 and based on PyGit 2, Pagure was built for Fedora Project folks and by Fedora Project folks.

Package sources

A specialized version of Pagure is maintained for managing package sources including codebases and specfiles that define the packages installable on Fedora Linux.

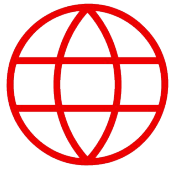
Managed history

Separate git repository is maintained for issue tickets, pull requests and project documentation apart for the one that is displayed in the main section for source code.

Source:
https://docs.pagure.org/pagure/usage/index.html
https://pagure.io/pagure/blob/master/f/README.rst
https://pagure.io/about/

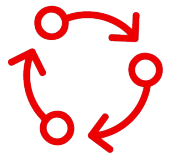
मुख्य साधन

Fedora Messaging



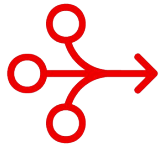
Working together

A free and open source software exchanging messages from services, Fedora Messaging allows for publishing and consuming messages across applications.



Cross compatible

Fedora Messaging has schemas for services like but not limited to Anitya, Bodhi, COPR, Fedocal, Elections, Planet, Ansible, Koji, Accounts, MDAPI and Pagure.



Deserving sequel

This Pika-backed library was developed as a replacement for the PyZMQ-back FedMsg library that was used before with multiple significant improvements.

Source:
https://github.com/fedora-infra/fedora-messaging
https://docs.fedoraproject.org/en-US/infra/developer_guide/messaging/
https://fedora-messaging.readthedocs.io/en/stable/

Quality Assurance

Tried, tested and ergo trusted

Why though?

- ***High standards***
  Testing ensures that releases meet high standards of quality and reliability that users expect from Fedora Linux

- ***Triaging defects***
  Community members and Fedora QA team prioritize and resolve issues logged in Bugzilla related to the oncoming release

मुख्य साधन

OpenQA

Automation testing

A free and open source software platform for automated testing of operating systems, OpenQA allows for running tests across kernel, bootloaders and GUI.

Properly contained

OpenQA performs tests inside virtual machines and closely monitors the state of the running instance in a predictable manner to ensure consistent testing results.

Almost everywhere

OpenQA is maintained primarily by the friends at OpenSUSE but alongside Fedora Project, it is used by and contributed to by a lot of community folks.

Source:
https://open.qa/
https://openqa.fedoraproject.org/
https://open.qa/documentation/

मुख्य साधन

Blocker Bugs

Objective validation

A free and open source software platform for proposing and track bugs, Blocker Bugs focusses on defects that block releases and are freeze exceptions.

Properly segregated

Defects are classified across various categories like zero day blockers, previous release blockers, freeze exceptions, prioritized blockers, prioritized bugs etc.

Almost everyone

While primarily used by the Fedora QA team, this can be participated into by every contributor participating in the Fedora Linux release engineering process.

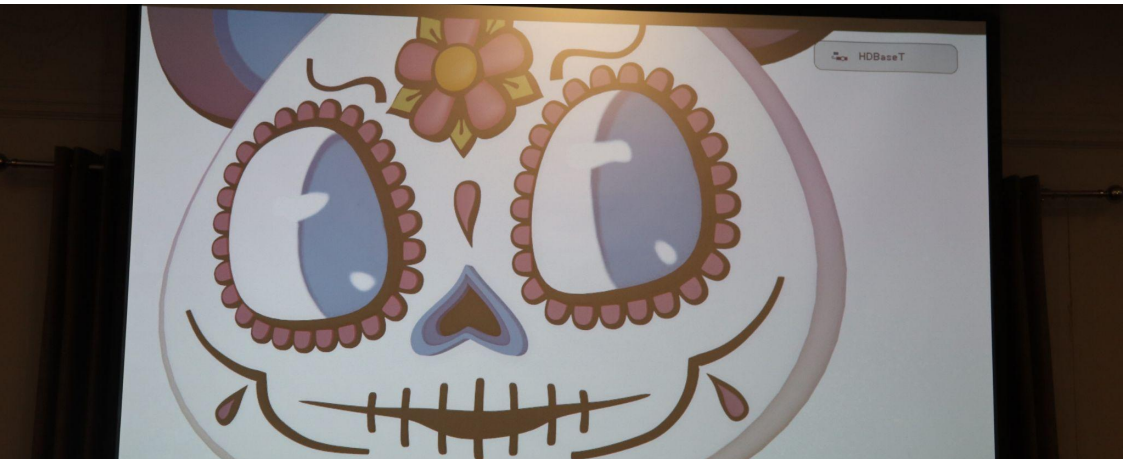
Source:
https://fedoraproject.org/wiki/Blocker_Bug_FAQ
https://fedoraproject.org/wiki/QA:SOP_blocker_bug_process
https://qa.fedoraproject.org/blockerbugs/

It is advised to report *software defects* or *feature requests* for **GNOME applications** to the upstream **GNOME community** and focus only on the distribution specific details in the quality assurance.

Fedora Linux has a dedicated set of testcases for the *Workstation group of RPM packages* having a curated set of **GNOME** applications to ensure that the users get the best **GNOME** experience.

Beta and Final Freezes

Spicy features - Locked and loaded



Possible variants

- ***Beta freeze***
  Recent features are locked and packaged so that their updates can be provided via Bodhi for testing among interested folks

- ***Final freeze***
  Only critical fixes are allowed to be made atop the beta freeze while preparation for the final compose is done

मुख्य साधन

Bodhi

Everyone voices

A free and open source software web platform written in Python 3 and based on Pyramid, Bodhi allows for package update testing process to be democratized.

Feedback loop

Maintainers can push updates while testers can test them in a unified platform where a feedback loop for package for various distribution versions is established.

Automated installs

Apart from that, Bodhi announces the arrival of new packages, publishes update release notes, generates repositories and displays testing results.

Source:
https://bodhi.readthedocs.io/en/stable/
https://github.com/fedora-infra/bodhi
https://bodhi.fedoraproject.org/

It is a rare occurrence but sometimes **Fedora Linux** takes the liberty to decide GNOME experience packages, for example, selecting **Ptyxis** for the default console application in **Fedora Linux 41**.

Release composing

Taming the Anaconda using Pungi

Deliverables

- ***Physical***
  ISOs usable for installing Fedora Linux on using bootable medium are generated by manipulating Anaconda in Pungi

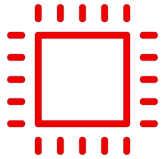
- ***Virtual***
  Images intended for cloud purposes (eg. AWS, OpenStack etc.) and for container purposes (eg. Podman etc.)

मुख्य साधन

Pungi

Unified deliverables

A free and open source software distribution composer written in Python 3, Pungi allows for the creation of release snapshots containing release deliverables.



Delegating management

Tasks are delegated to separate executables in Pungi and the proper order and location of artifacts are ensured for creating RPM repositories and installer ISOs.

Interesting origins

Pungi works with Koji to start the process using a compose configuration and as Anaconda was the software Pungi was exploring, that is where the name came from.

Source:
https://docs.pagure.org/pungi/about.html
https://en.wikipedia.org/wiki/Pungi_(software)
https://pagure.io/pungi

Backports for *critical or non-critical updates* are first tested in **Fedora Rawhide** before the updates make their way into **Bodhi** for feedback and finally coming to the **Fedora Linux** installations.

Deliverables testing

How does it all work - TOGETHER

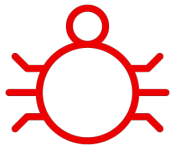
Validating

- ***Physical***
  Composes are validated for installability and functions across various hardware configuration among contributors alike

- ***Virtual***
  Deliverable checksums and GPG signatures are validated to ensure the consistency of the compose file

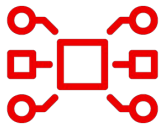
मुख्य साधन

Bugzilla



Tracking defects

A free and open source software bug tracking system written in Perl, Bugzilla allows for submitting and reviewing software defects in Red Hat distributions.



Workflow specific

Our deployment has customizations that support workflows that are more specific to the RHEL, CentOS Stream and Fedora Linux for automating bug life cycle.

Upstream first

Defects related to the software packages are focused on here and every other defects are recommended to be reported in the upstream communities.

Source:
https://bugzilla.redhat.com/page.cgi?id=faq.html
https://bugzilla.redhat.com/docs/en/html/index.html
https://bugzilla.redhat.com/

As a part of *regularly scheduled test days*, we organize testing of the additions that make it to the **Fedora Linux** release – including **GNOME applications** so that they can be polished before release.

Pushing Releases

One compose makes it to you

Processes

- ***Availability***
  Composes - including the final one - make it across to the mirrors worldwide maintained by the community members

- ***Announcement***
  Release is announced on Fedora Magazine and the resource links are updated across the Fedora Project websites

मुख्य साधन

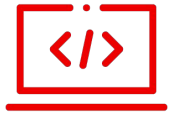
Mirror Manager

Metalink Provider

A free and open source software mirroring system handler written in Python 3, Mirror Manager tracks the list of valid mirrors and processes out download URLs.

Scheduled operation

Regularly scheduled chores allow for updates to be made based on the changes noticed from the main mirror and then the difference is conveyed via transfer.



Rudimentary frontend

A frontend serves a heavily cached collection of content as old as 12 hours to ensure that it serves only as an overview for all the existing mirrors for Fedora Project.

Source:
https://mirrormanager.fedoraproject.org/statistics
https://docs.fedoraproject.org/en-US/infra/sysadmin_guide/mirrormanager/
https://fedoraproject.org/wiki/Infrastructure/Mirroring

Minor versions of **GNOME** application updates come as a part of software updates through **Fedora Linux** package repositories while major versions are exclusive to **Fedora Linux** releases.

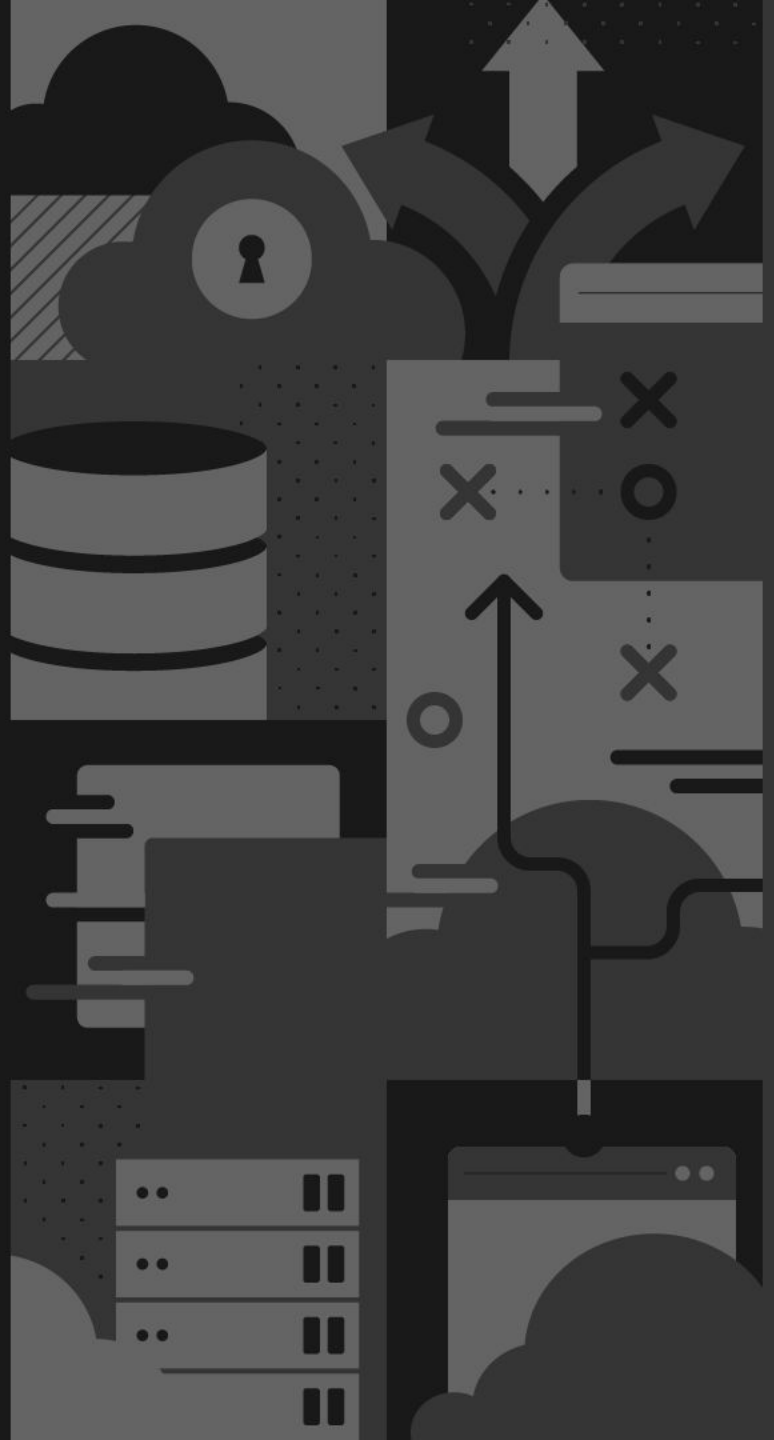
Ongoing Maintenance

Of course - it is not over yet

Processes

- ***Updates and rollouts***
  Minimal package updates are provided without needing a major update for most packages through Bodhi and DNF

- ***Back to Rawhide***
  Collected feedback is incorporated in the next cycle of Rawhide where the development resumes with same processes



Participate

हिस्सा लेना

fedora Red Hat

योगदान

Contributing

Small steps

Understanding how to succeed as a contributor
Subscribing to the Fedora Infrastructure mailing list
Connecting with other fellow contributors

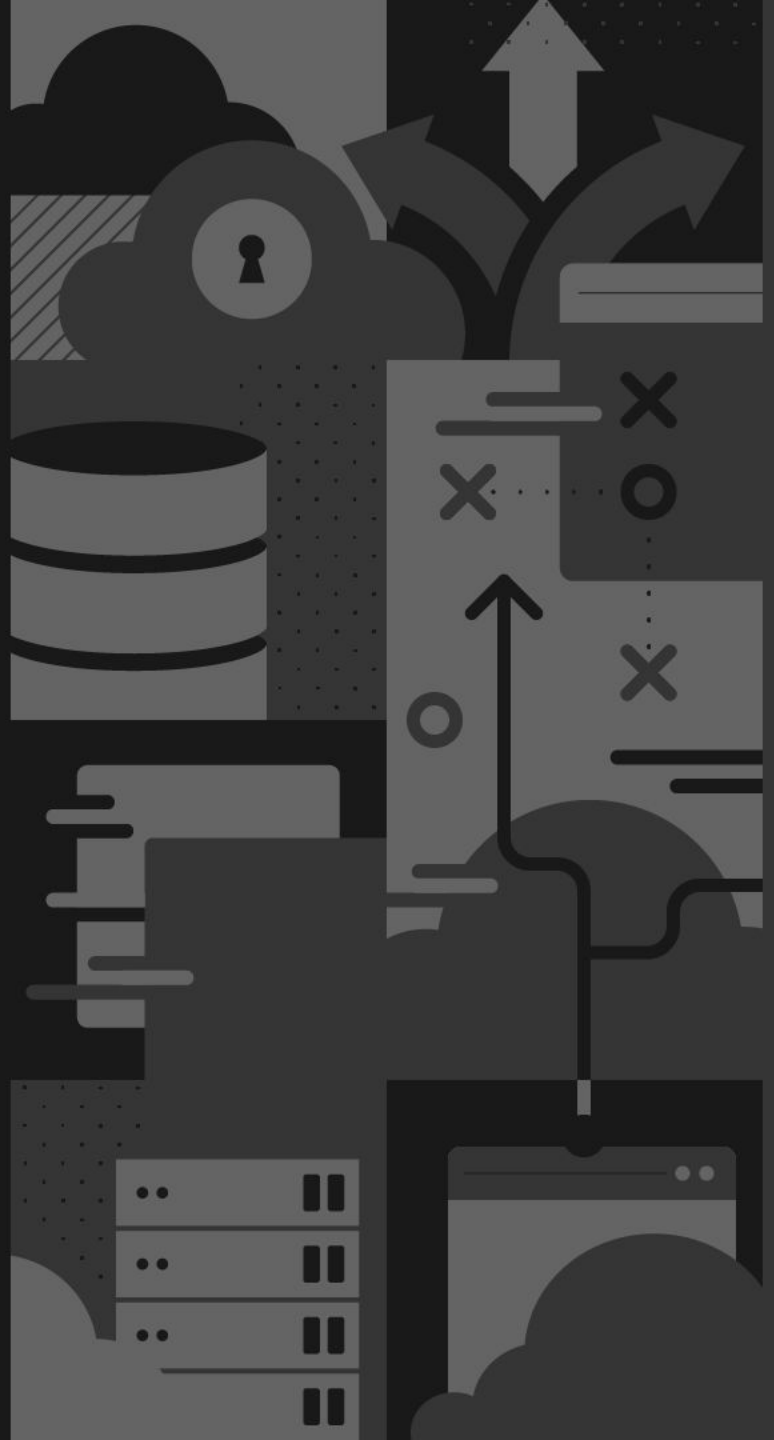
Lifting pace

Learning about the infrastructure services
Participate in the Infrastructure Apprentice group
Monitoring conversations for showing interest

Thousand miles

Maintaining services regularly in the infrastructure
Helping document for other beginner contributors
Developing solutions for the issues created

Source:
https://docs.fedoraproject.org/en-US/infra/
https://docs.fedoraproject.org/en-US/infra/day_to_day_fedora/
https://docs.fedoraproject.org/en-US/infra/overview/



Thank you

धन्यवाद

Fedora Infra - GitHub

Fedora Infra - Pagure

fedora Red Hat