Red Hat Enterprise Linux 8

Technical Overview

Patrick Ladd

Technical Account Manager

pladd@redhat.com

https://people.redhat.com/pladd

What's New in RHEL 8?

RHEL 8 Basics	In Place Upgrades	Cockpit
Release Cycle	Ansible Sys Roles	OCI / UBI
App Streams	Image Builder	VDO
yum v4	Insights	

RHEL 8

At a glance:

The latest updates from upstream communities, combined with continuity of expected tools

Kernel Version	4.18+
System Compiler	GCC 8.2, LLVM 6.0
Hardware Arhictectures	Intel/AMD 64-bit, IBM Power LE, IBM z Systems, ARM 64-bit
Default File System	XFS
Package Management	Yum v4
Time Synchronization	Chrony
Networking	NetworkManager

Red Hat

Releases and Packaging

The Conflict

"

I want the latest and greatest

versions of tools

—

Developers

"

I want to know everything is

stable and supported

—

Operations

Predictable updates

3 years
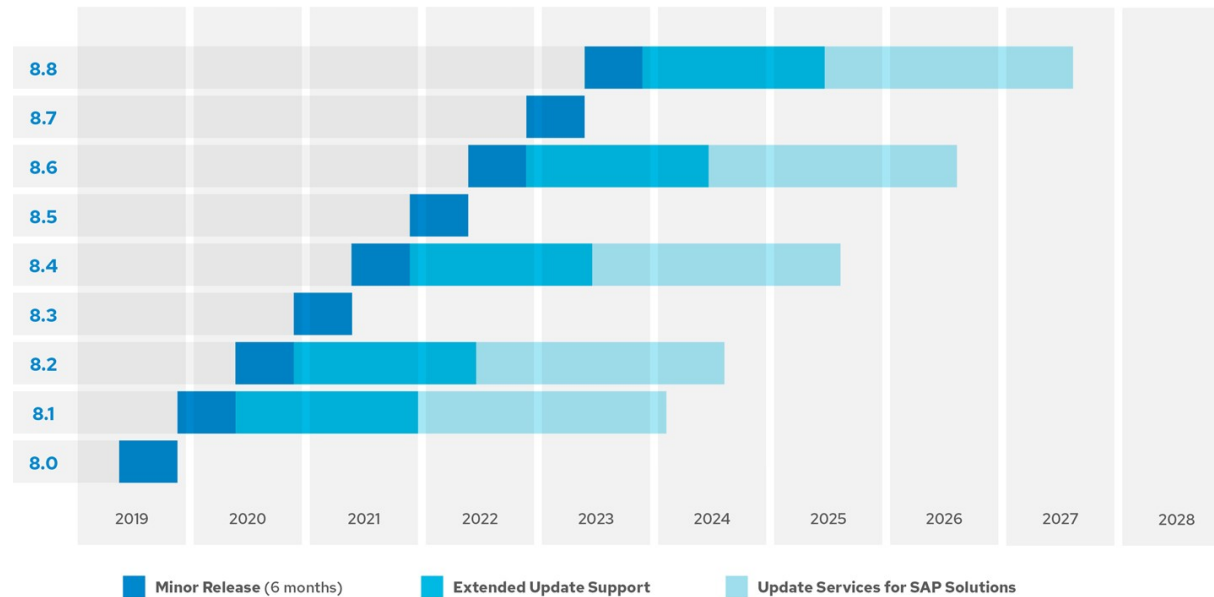Major releases

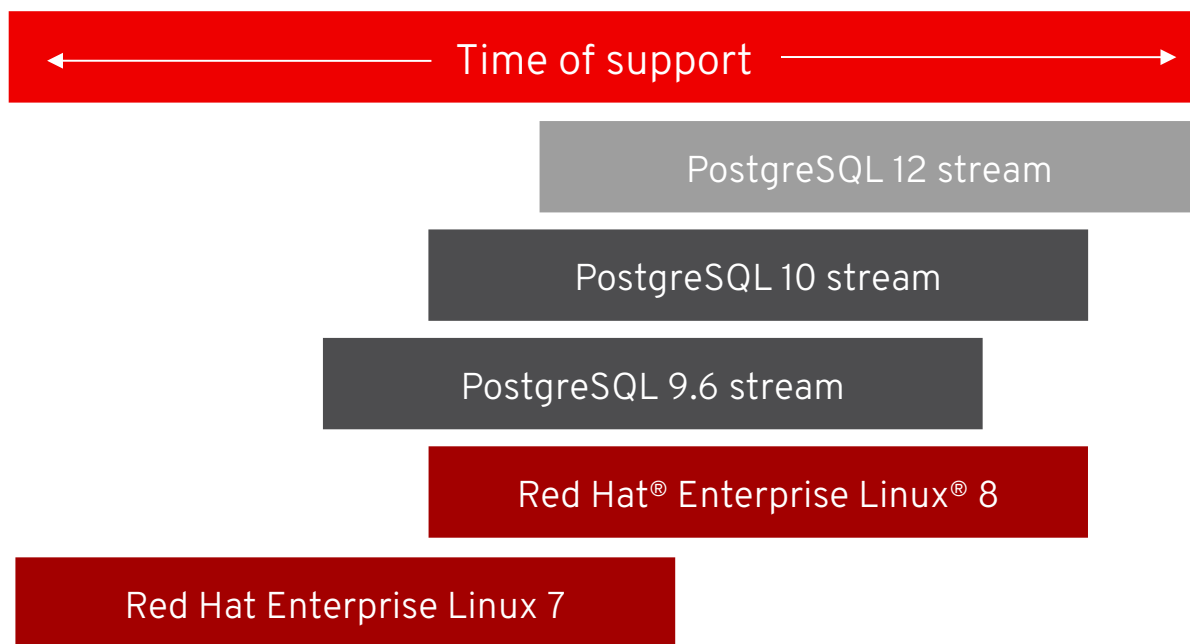## 6 months
Minor updates

2 phases
Support life cycle

Red Hat

RHEL 8 Support Cycles

Application streams

Time of support

PostgreSQL 12 stream

PostgreSQL 10 stream

PostgreSQL 9.6 stream

Red Hat® Enterprise Linux® 8

Red Hat Enterprise Linux 7

More choice
Offers versions of the open source tools and frameworks developers need

Newer versions
Provides access to newer versions as they stabilize

Simpler access
Maintains standard locations for tools and libraries
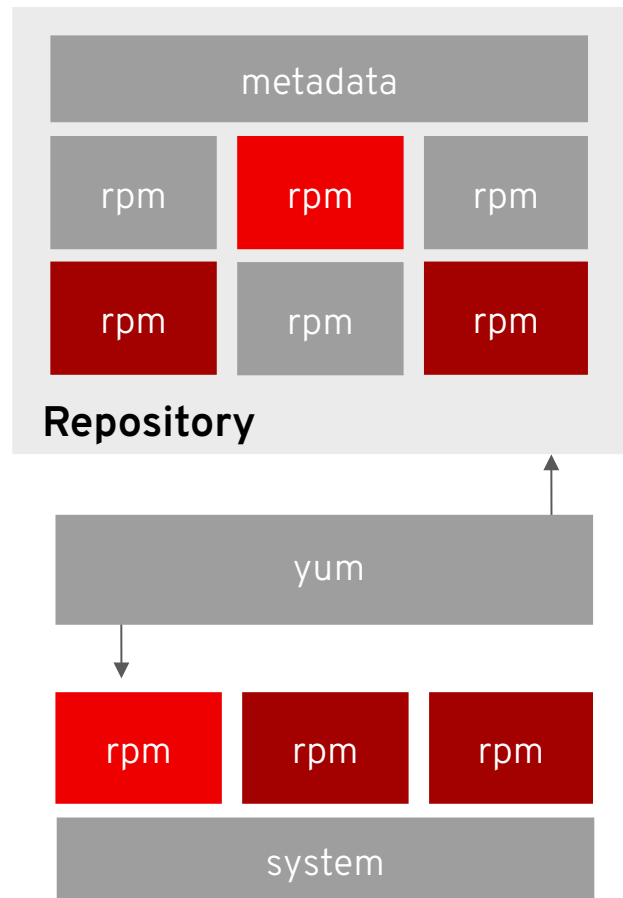
Simplified access to software

Red Hat Enterprise Linux 7 repositories

extras	rhscl	dotnet	devtools	supplementary	optional

server

Red Hat Enterprise Linux 8 repositories

appstream	supplementary	codeready-builder

baseos

Red Hat

The newest yum package manager: version 4

New technology
- RPM 4.14
- YUM v4 (based on DNF)

Better dependency management
- YUM v3 had custom dependency resolution code
- YUM v4 uses libsolv library for dependency resolution

Stable API
Provides new application programming interface (API) for extending yum that will progress into the future
- command line compatible with RHEL7

yum4 Modules

- Modularity features are available through the yum module command.
- Modularity is implemented in a way that is compatible with existing YUM v3 workflows.
- Using modularity features is optional, not mandatory.
- Implementation details:
  - Module streams are treated as virtual repositories within the AppStream repository.
  - Module streams have dependencies that allow to auto-enable dependent streams.
  - Module profiles are treated as package (comps) groups.

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/installing_manag ing_and_removing_user_space_components/installing-rhel-8-content_using-appstream

Modules vs Software Collections

Modules are the next generations SCLs

Advantages vs SCLs

- Modules install in the same locations as upstream versions
- All application tooling works out of the box
- Client connectors work without the need for wrappers and links
- Modules don't need to be specifically enabled in a shell

Disadvantages vs SCLs

- Only one version of a module may be installed on a system at a time
- Modules may have cross dependencies that prevent some combinations of modules from working on the same system with other modules

yum4 Rich and weak dependencies

Weak dependencies

- Introduce four new dependency types
  - Recommends
  - Supplements
  - Suggests
  - Enhances
- Ability to create reverse dependencies on 3rd party packages.
- Ability to create dependencies on packages in repositories that are not enabled.
- Ability to install with minimal footprint.

More information:
https://fedoraproject.org/wiki/Packaging:WeakDependencies

Rich (boolean) dependencies

- Dependencies can contain conditions and operators
- Ability to *Require* one package from a set of packages without creating virtual *Provides*.
- Ability to create conditional dependencies, for example to install a language pack for a package if a language is installed.

More information:
https://rpm.org/user_doc/boolean_dependencies.html

Demo

- `yum module list`
- `yum module list postgresql`
- `yum module info postgresql:9.6`
- `yum module info postgresql:9.6 --profile`
- `yum module install postgresql:9.6/client`
- `yum module install postgresql`
- `yum module enable postgresql:10`
- `yum distro-sync`
- `yum module reset postgresql`
- `yum module list postgresql`

Installs and Upgrades

Create images for all your environments with image builder

Blueprint

Bare metal

Hypervisors

Public clouds
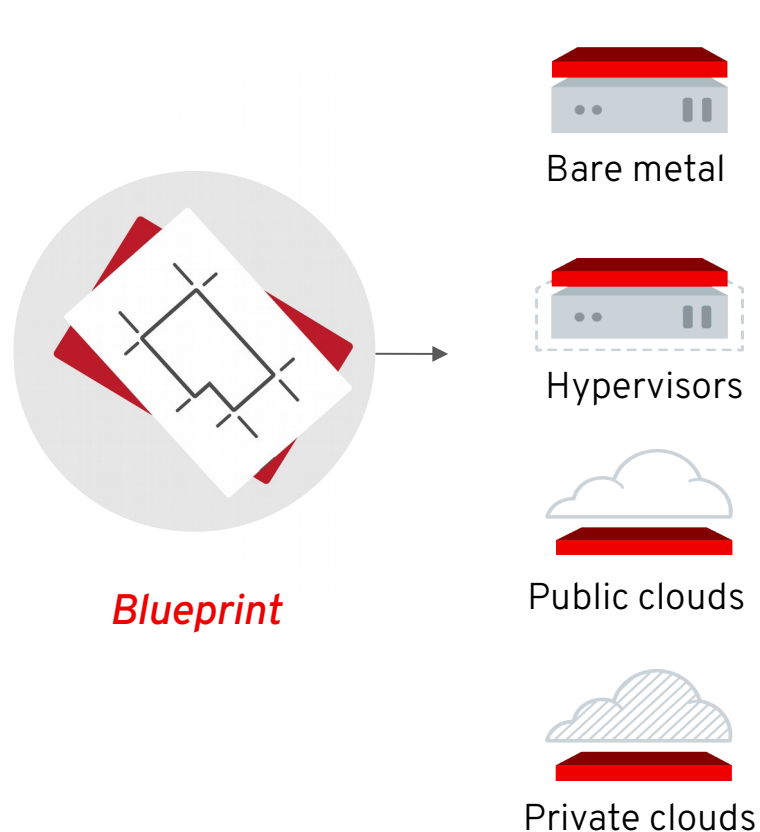
Private clouds

Single source
Lets you create gold images for any environment from the same blueprint increasing stability and consistency

Any footprint
Supports public cloud, private cloud, enterprise hypervisors, and bare metal

Simple interface
Provides web-based view within the web console for selecting packages and creating blueprints

Image Builder

Image Formats
- Raw disk (.img)
- Live ISO (.iso)
- File system (.img)
- Tarball (.tar.xz)
- VMDK (VMware® vSphere® Hypervisor)
- AMI (Amazon Web Services®)
- VHD (Microsoft® Azure®)
- QCOW2 for KVM/RHV/Satellite/CloudForms
- QCOW2 for OpenStack

Composer Interfaces
- Command Line
- Cockpit Plugin

Installation
- `yum install lorax lorax-composer composer-cli cockpit-composer`
- `systemctl enable --now lorax-composer.socket`
- `systemctl restart cockpit.service`

Demo

- Install Image Builder
- Create blueprint
- Customize blueprint
- Create image

In-place upgrades for your systems

Applications

7 → 8
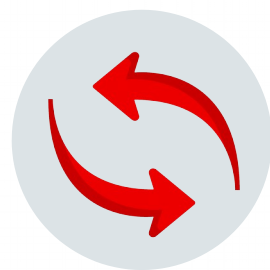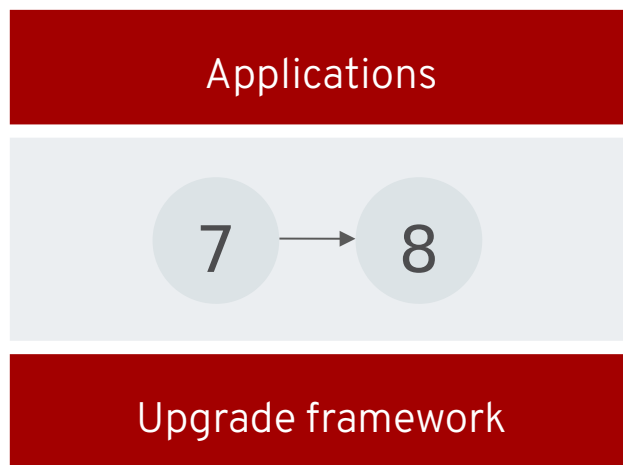
Upgrade framework

Reduced migrations
Analyze systems to determine if upgrading in place can avoid a costly migration

Easy rollback options
Combine with bootable LVM snapshots for safety

Improved framework
Get better analysis and a simplified process with a more extensible framework
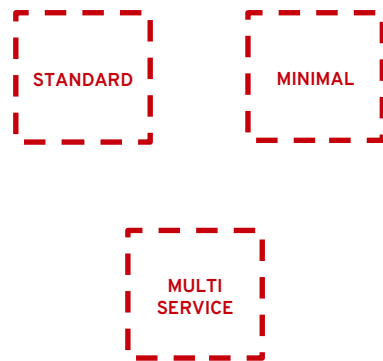
Can I upgrade this host?

PICK CANDIDATE SERVER

RUN LEAPP

ANALYZE CHECK OUTPUT

CREATE BOOTABLE LVM SNAPSHOT

REBOOT TO FINISH UPGRADE

UPGRADE COMPLETE
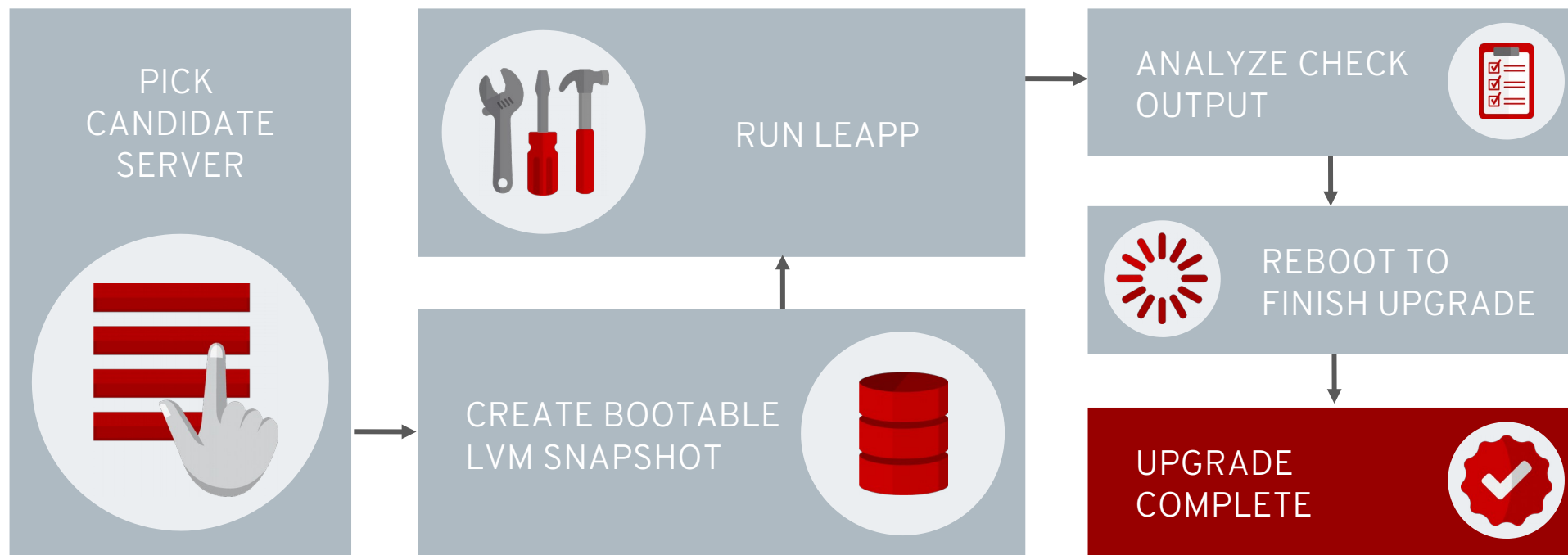
Demo

- Create LVM bootable snapshot
- Run LEAPP
- Analyze output
- Reboot to finish

Containers are Linux

Red Hat Universal Base Image (UBI)

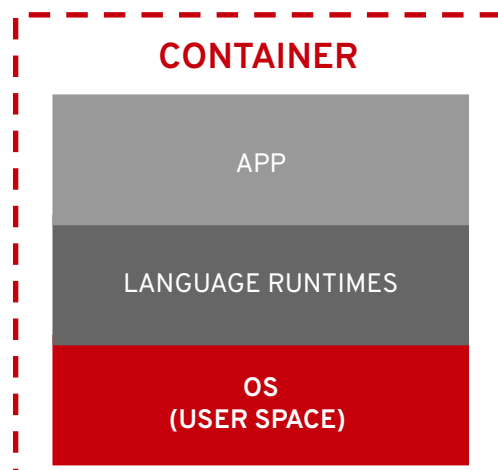*"To be the highest quality and most flexible base container image available"*

STANDARD

MINIMAL

MULTI SERVICE

Node.js

ALL RED HAT ENTERPRISE LINUX PACKAGES

UNIVERSAL BASE IMAGE PACKAGES
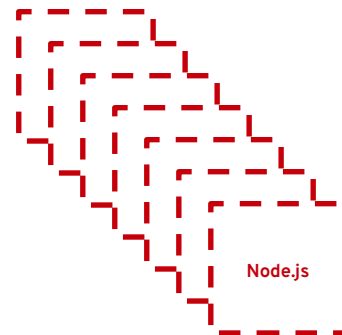
Base
Images

Pre-Built
Language
Images

Package
Subset

Red Hat Universal Base Image (UBI) Licensing

Run anywhere at no charge

CONTAINER

APP

LANGUAGE RUNTIMES

OS
(USER SPACE)

The Red Hat Universal Base Image is based on RHEL and made available at no charge by a new end user license agreement.

- Supported as RHEL when running on RHEL
- Same Performance, Security & Life cycle as RHEL
- Can attach RHEL support subscriptions as RHEL

https://www.redhat.com/en/about/red-hat-end-user-license-agreements#UBI
http://crunchtools.com/ubi-licensing/

Red Hat Universal Base Image (UBI)

Standard Image Types

MINIMAL

APP

microdnf + coreutils

glibc (En locale)
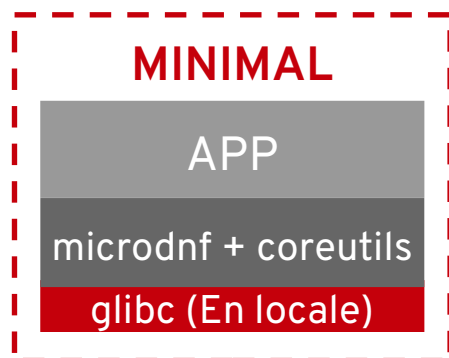
ubi8/ubi-minimal

Designed for applications that contain all dependencies (Golang, dotnet, etc)

- Minimized content set
- No suid binaries
- Minimal package manager (install, update, remove)

PLATFORM

APP

YUM + @base

glibc (full locales)
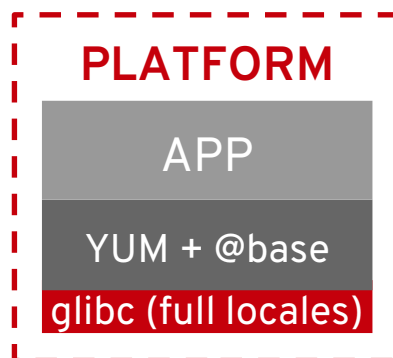
ubi8/ubi

For any application that runs on RHEL

- Unified, openssl crypto stack
- Full YUM stack
- Includes useful basic OS tools (tar, gzip, vi, etc)

MULTI-SERVICE

APP0   APP1

/usr/sbin/init

glibc (full locales)
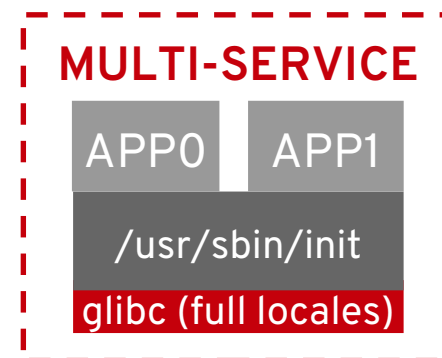
ubi8/ubi-init

Eases running multiple services in a single container

- Configured to run systemd on start
- Simply enable the services at build time

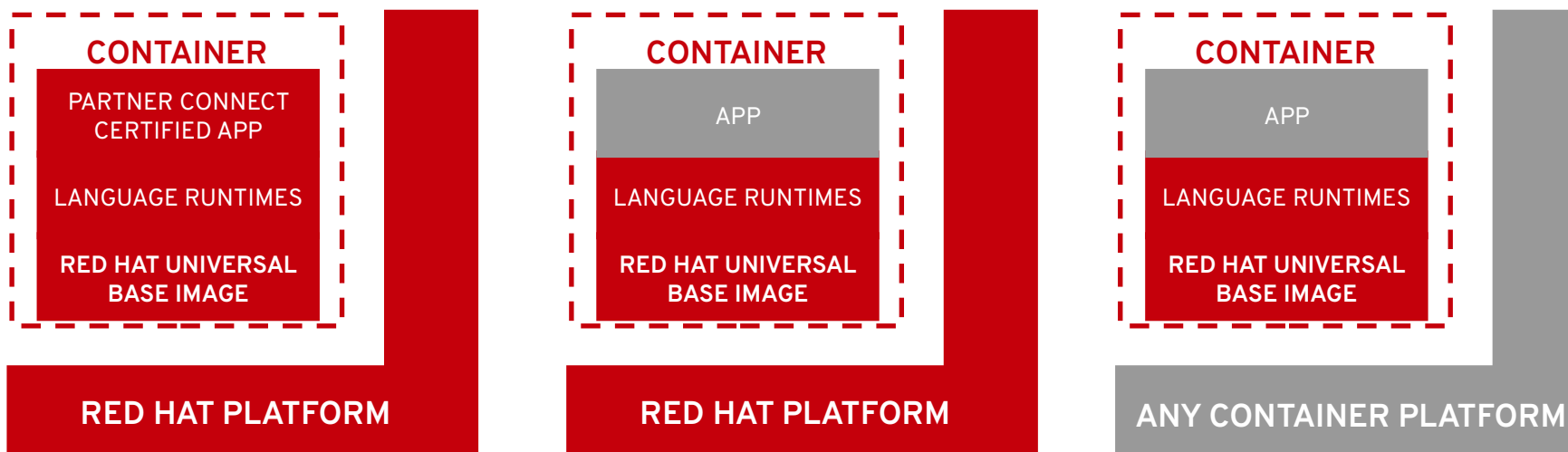# Red Hat Universal Base Image (UBI)

Standard Runtimes

- DotNet
- Perl
- PHP
- NodeJS
- Python
- Ruby
- s2i

Detailed list at <u>registry.redhat.io</u>

Red Hat Universal Base Image (UBI)

Supportability

CONTAINER		CONTAINER		CONTAINER
PARTNER CONNECT CERTIFIED APP		APP		APP
LANGUAGE RUNTIMES		LANGUAGE RUNTIMES		LANGUAGE RUNTIMES
RED HAT UNIVERSAL BASE IMAGE		RED HAT UNIVERSAL BASE IMAGE		RED HAT UNIVERSAL BASE IMAGE

RED HAT PLATFORM   **RED HAT PLATFORM**   **ANY CONTAINER PLATFORM**

Certification provides the highest level of support

Enterprise support when run on Red Hat platforms

Trusted base for any environment

Red Hat Universal Base Image (UBI)

Updates

CONTAINER

APP

LANGUAGE RUNTIMES

RED HAT UNIVERSAL BASE IMAGE

Base Image Updates

Associated RPM Updates
(separate CDN)

Containers are Linux

Run	Build	
		Application binaries
		Application dependencies
		CONTAINERS

SELinux

Seccomp

Cgroups

Namespaces

CPU	Memory	Storage	Network
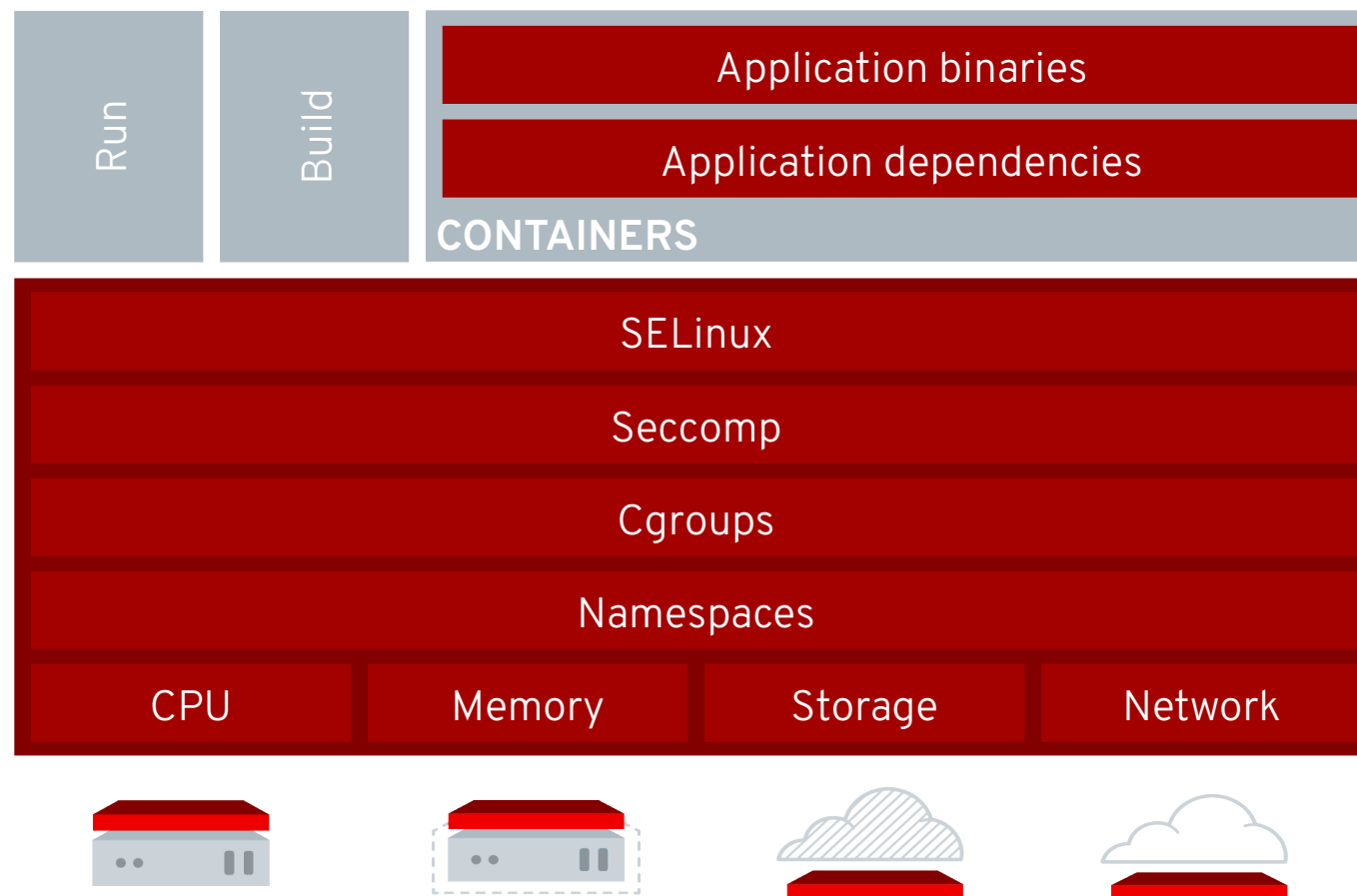
Powering the adoption of containerized workloads

TRADITIONAL DEVELOPMENT					CLOUD-NATIVE	
Find	Run	Build		Share	Integrate	Deploy
Red Hat Enterprise Linux (Podman/Buildah/Skopeo)				Quay	Red Hat OpenShift® (Kubernetes)	
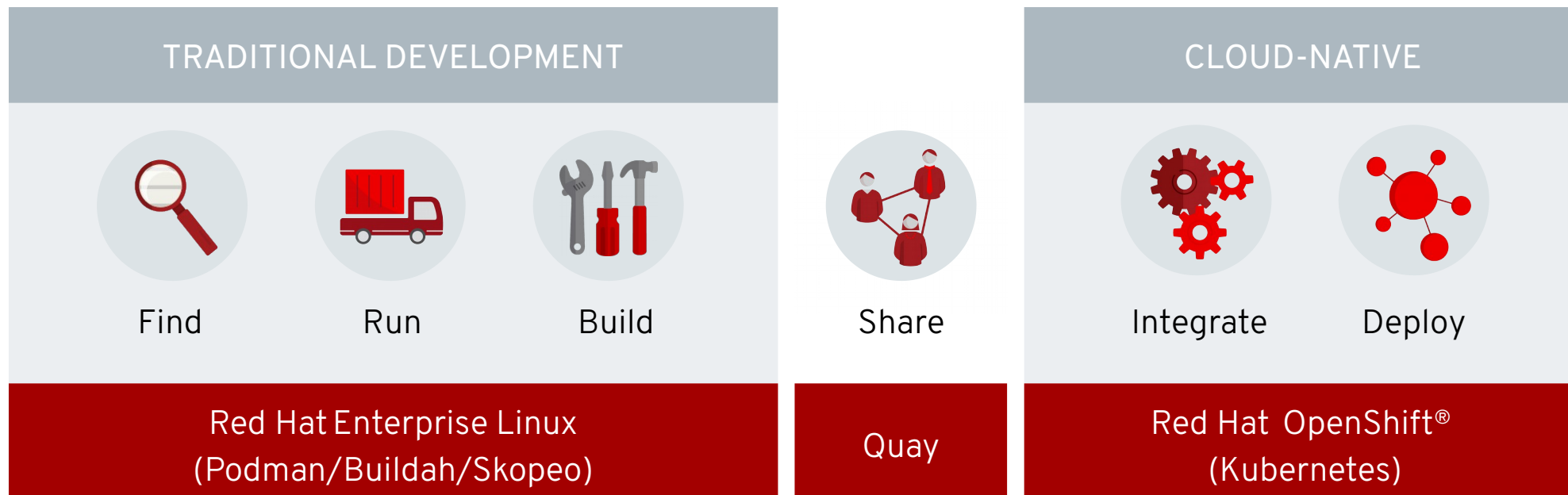
Manage containers with Podman

Kernel

RunC

Podman

Images
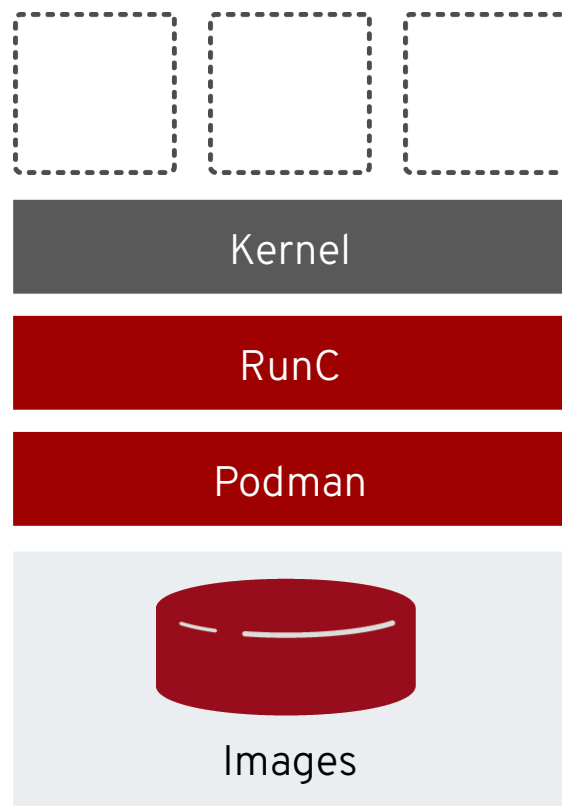
Fast and lightweight
No daemons required

Advanced namespace isolation
Rootless operations for container run and build

Open standards compliant
Creates and maintains any standard Open Containers Initiative (OCI) - compliant containers and pods

Create images with Buildah

Application layer

Java™ runtime layer

OS update layer

Base Red Hat
Enterprise Linux

Java runtime and
dependencies, and
application

From base, multilayer   From scratch, single layer
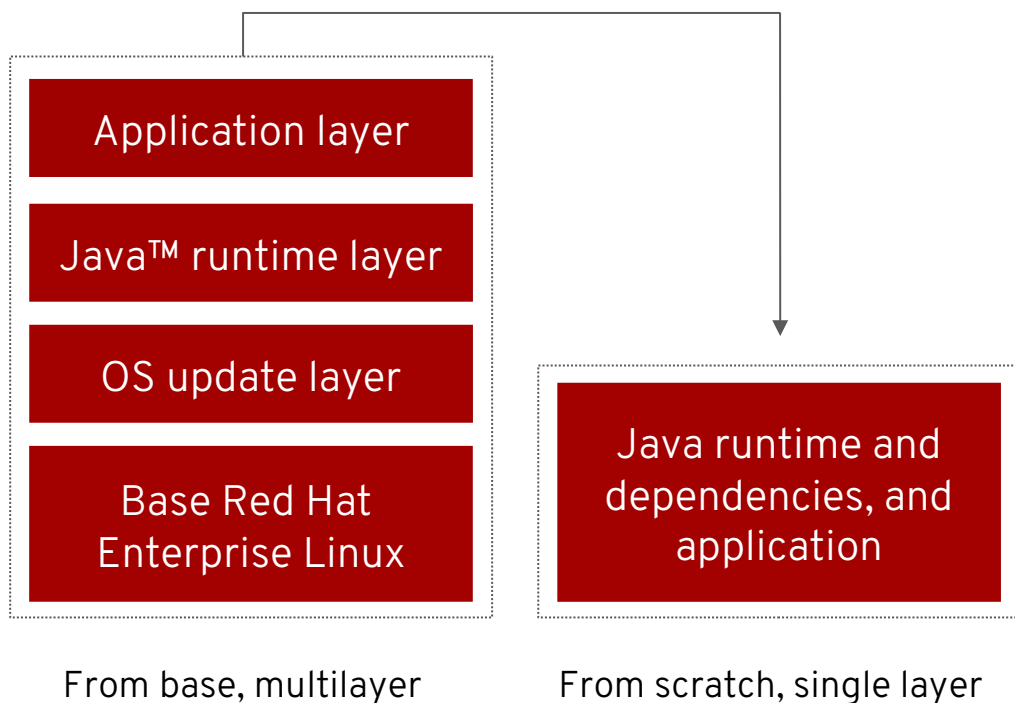
More control
Scriptable tooling for fine-grained
image control, and maximum control
starting from base or scratch images

Minimization of images
Elimination of unneeded dependencies
by using host-based tools

Red Hat

Inspect and transport images with Skopeo

Red Hat OpenShift Container Platform

QUAY

docker

Image registry	Image registry

SKOPEO

Metadata signature	Image repository
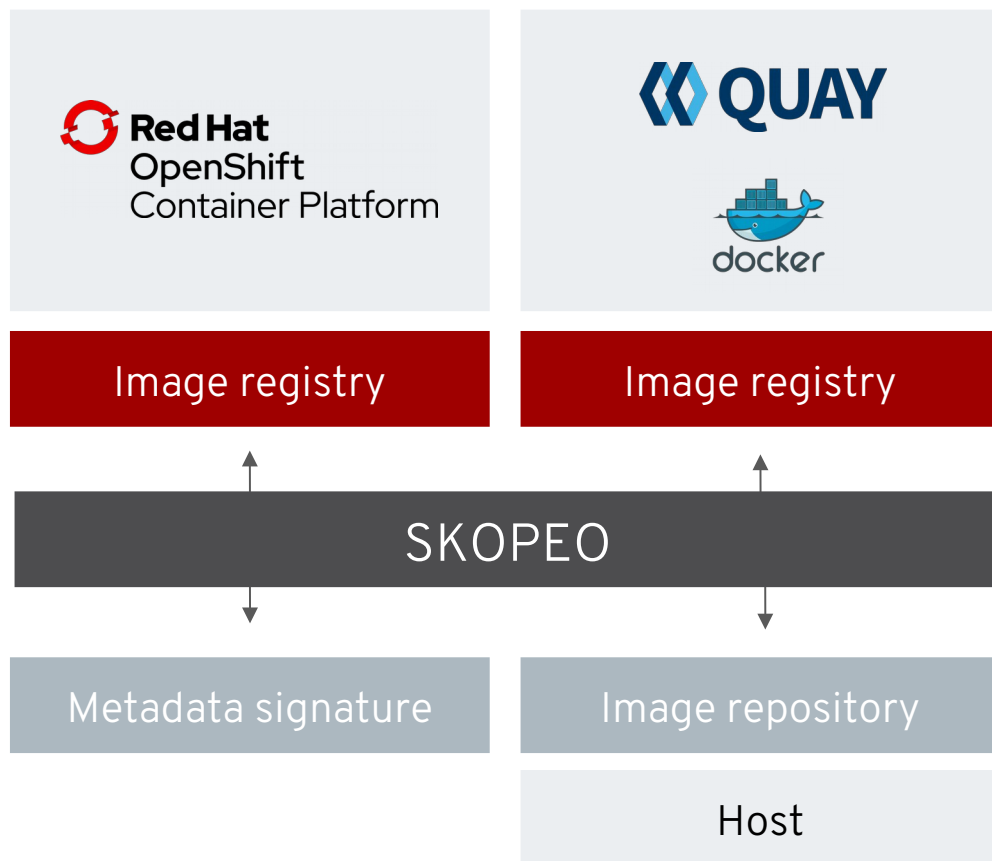
Host

Inspect images remotely
Examine image metadata without needing to download

Publish and transfer images
Copy images from registries to hosts or directly between registries

Sign and verify images
Supports GPG key signing on publish

OCI Demo

```
yum install docker
```

Management

RHEL System Roles

kdump

timesync

network

selinux

postfix*

dbserver

webserver

Red Hat
Ansible
Automation
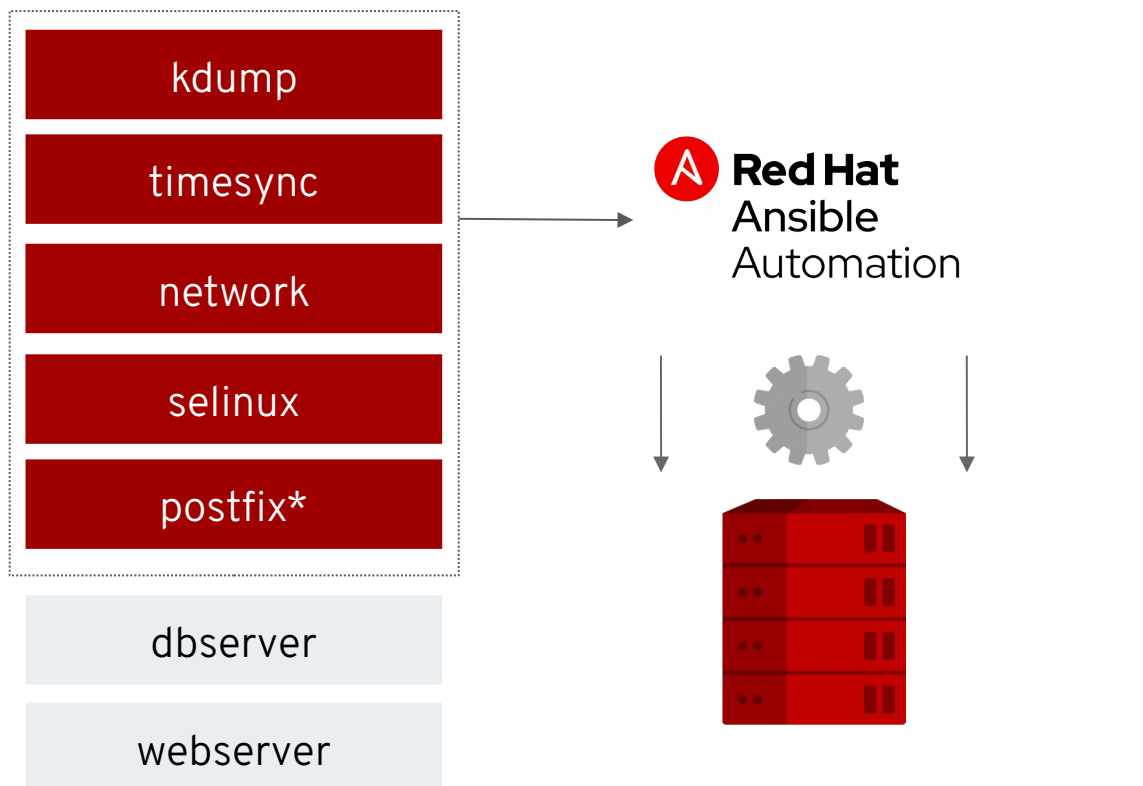
Common automation
Manage multiple versions of
Red Hat Enterprise Linux from a
single role – RHEL 6, 7, 8

https://access.redhat.com/articles/3050101

Playbooks:
`/usr/share/ansible/roles/rhel-system-roles.SUBSYSTEM/`

Documentation:
`/usr/share/doc/rhel-system-roles-<version>/SUBSYSTEM/`

* tech preview

System Role Demo

Using the Network system role

RHEL8 System Roles

Detect and fix issues with Red Hat Insights

Now Free With All RHEL Subscriptions!
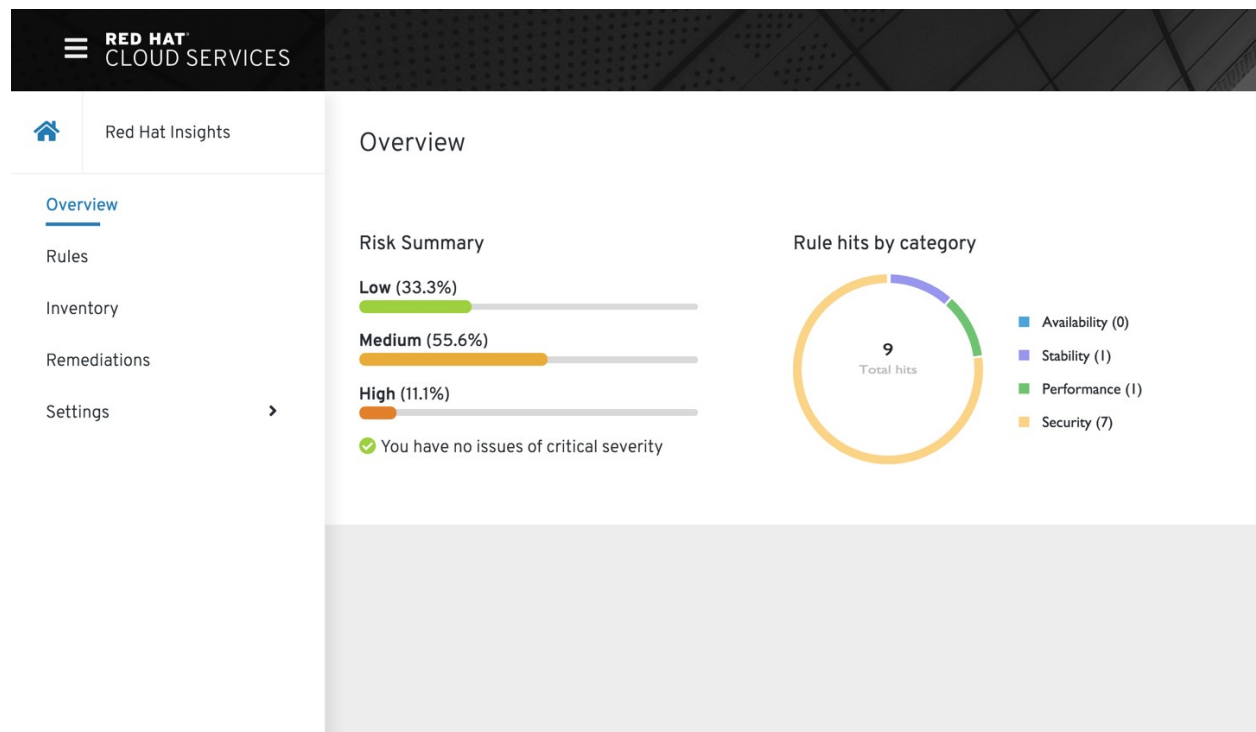
Proactive advice
Identification of issues before they become problems

Continuous assessment
Real-world results to help find new risks

Simple remediation with Ansible
Tailored results at the host level

Red Hat Insights Demo

Insights Dashboard

Insights Rules

Remediate problems with Ansible

Cockpit – the RHEL Web Console

Installed by default – needs to be enabled

```
[0] 16:28 pladd pladd-laptop.usersys ~ $ ssh pladd@rhel8
pladd@rhel8's password:
X11 forwarding request failed on channel 0
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Wed Jun 12 16:28:28 2019 from 192.168.122.1
```

```
[0] 16:30 pladd pladd-laptop.usersys ~ $ ssh pladd@rhel8
pladd@rhel8's password:
X11 forwarding request failed on channel 0
Web console: https://rhel8.local:9090/ or https://192.168.122.128:9090/

Last login: Wed Jun 12 16:28:55 2019 from 192.168.122.1

ATTENTION! Your session is being recorded!
```

Cockpit Login

Cockpit Main Screen

Red Hat Cockpit Demo

Systems

Storage

Logs

Users

Services

Terminal

Features

Recording user terminal sessions

Audit activities
Create a record of actions taken for review against security policies

Create visual guides
Build run books and training materials with demonstrations

Record and play back
Logged via standard channels with multiple playback options

Install - `yum install tlog cockpit-session-recording`
Enable & Check
Playback -
Export - `yum install systemd-journal-remote`
(whole file: `journalctl -o export | /usr/lib/systemd/systemd-journal-remote -o /tmp/example.journal -`)
Session only:
`journalctl -o verbose | grep -i \"rec\"`
`id tlog`
`journalctl -xe -o json-pretty _UID=<User ID of tlog>`
`journalctl -o export TLOG_REC=<Session ID> | /usr/lib/systemd/systemd-journal-remote -o /tmp/example.journal -`
Playback:
`tlog-play -r journal --file-path=/tmp/example.journal -M TLOG_REC=<Session ID>`
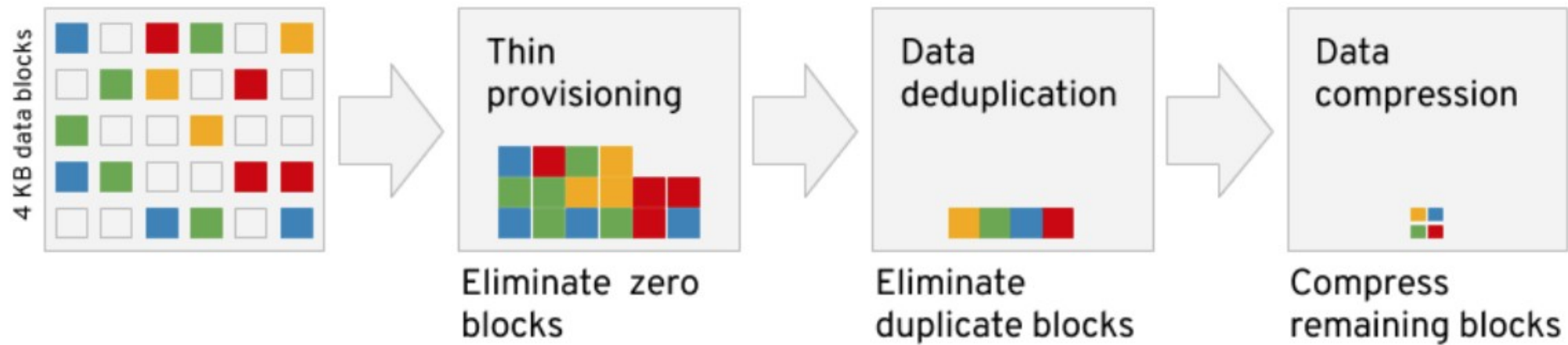
Session Recording Demo

Session Recording

- Documentation:
  - https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/recording_sessions/index

- Red Hat portal
  - https://access.redhat.com/solutions/3902881
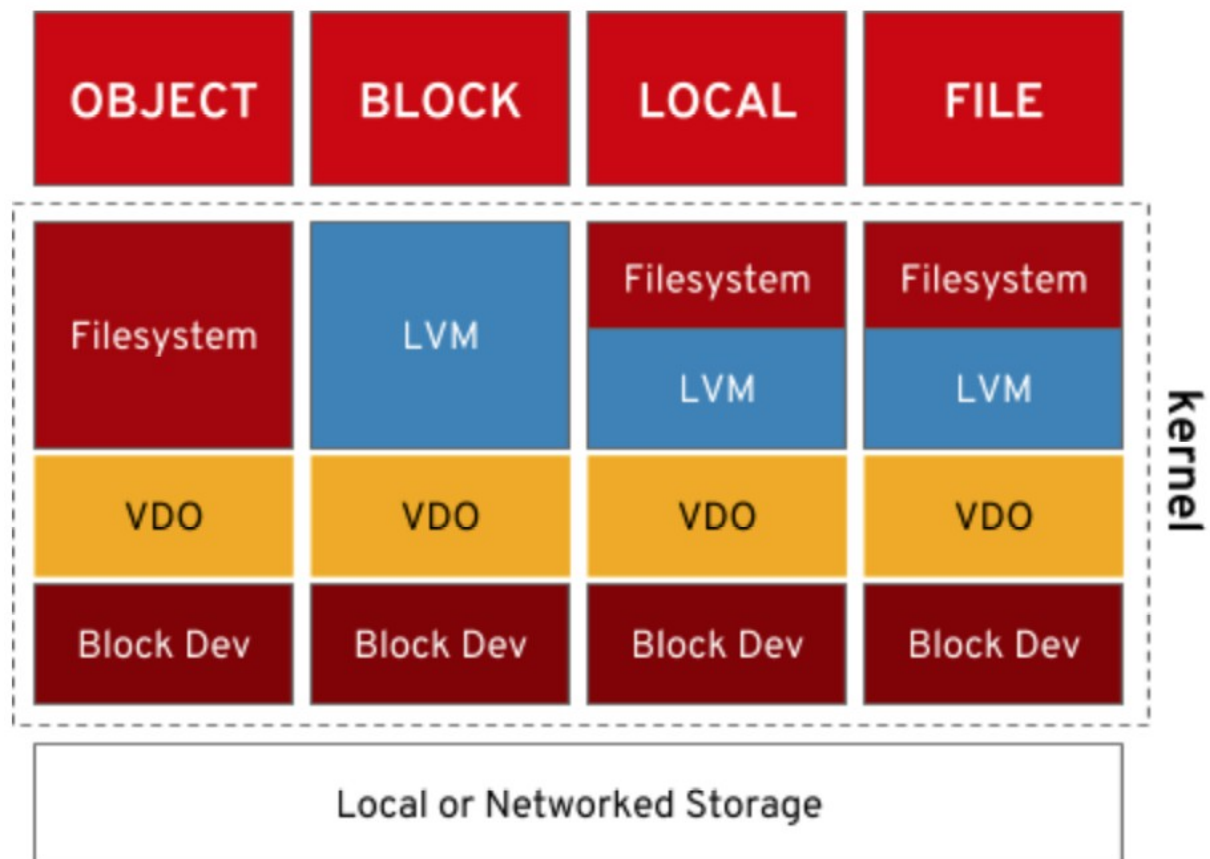 - https://access.redhat.com/solutions/4068941

Virtual Data Optimizer (VDO)

Disk De-duplication and Compression

Virtual Data Optimizer (VDO)

Where It Fits

VDO Demo

Create VDO device
Format and mount it
Copy some data
Observe usage

Virtual Data Optimizer (VDO)

Resources

- Documentation:
  - https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/deduplicating_and_compressing_storage/index
- Red Hat blog:
  - https://www.redhat.com/en/blog/look-vdo-new-linux-compression-layer
  - https://www.redhat.com/en/blog/understanding-concepts-behind-virtual-data-optimizer-vdo-rhel-75-beta
  - https://www.redhat.com/en/blog/determining-space-savings-virtual-data-optimizer-vdo-rhel-75-beta
  - https://www.redhat.com/en/blog/how-set-new-virtual-data-optimizer-device-using-cockpit-web-admin-console

Even More Stuff!

Check this out too!

- nftables
- Network bound disk encryption
- Fast file copy with XFS shared data extents
- Kernel EBPF tracing
- eBPF XDP (Xpress DataPath) and TC (Traffic Control)
- IPSec crpyto offloading
- TCP BBR for Flaky Mobile Networks
- 5 level page tables
- Stratis storage manager
- LUKS2 disk encryption

https://access.redhat.com/articles/4079441

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

in linkedin.com/company/red-hat

▶ youtube.com/user/RedHatVideos

f facebook.com/redhatinc

🐦 twitter.com/RedHat

Red Hat