



# Red Hat Enterprise Linux 8

## Technical overview

---

Alfred Bach  
Partner Enablement Manager  
EMEA

# AGENDA

- Value of Red Hat Subscriptions
  - Competition
  - RHEL 8 Overview
  - Red Hat Insights
- Installation
    - Composer
    - Upgrade
  - Server Management with Cockpit
  - Storage / Network / Virtualization
  - CoreOS
  - Universal Base Image
  - Security and Compliance
    - Session Recording
  - Upcoming Trainings

# Installation Upgrading and Composer

# One RHEL 8

## With a selectable purpose

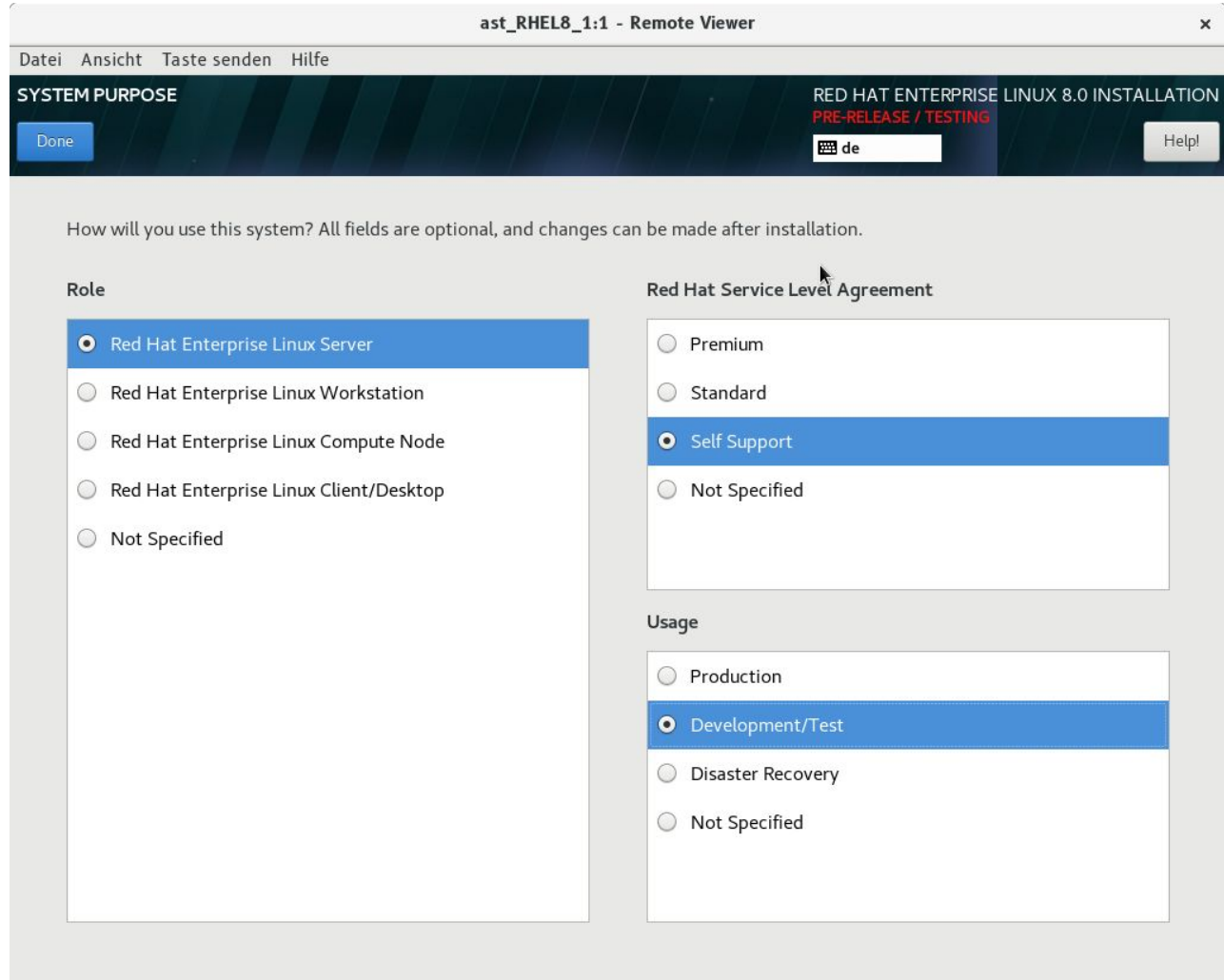
No more separate  
Releases/ISO for

- Desktop
- Server
- Compute Node

Updated Installer

- NVDIMM Support
- TPM Booting
- Modules via kickstart

*Auto Subscription & anaconda modules  
will follow after initial release*



The screenshot shows the 'ast\_RHEL8\_1:1 - Remote Viewer' window displaying the 'SYSTEM PURPOSE' selection screen for 'RED HAT ENTERPRISE LINUX 8.0 INSTALLATION'. The interface includes a 'Done' button, a language selector set to 'de', and a 'Help!' button. A message states: 'How will you use this system? All fields are optional, and changes can be made after installation.'

**Role**

- ☒ Red Hat Enterprise Linux Server
- ☐ Red Hat Enterprise Linux Workstation
- ☐ Red Hat Enterprise Linux Compute Node
- ☐ Red Hat Enterprise Linux Client/Desktop
- ☐ Not Specified

**Red Hat Service Level Agreement**

- ☐ Premium
- ☐ Standard
- ☒ Self Support
- ☐ Not Specified

**Usage**

- ☐ Production
- ☒ Development/Test
- ☐ Disaster Recovery
- ☐ Not Specified

# COMPOSER

## Components

Based on the Upstream Tools

- **Lorax**

Toolset to create bootable System Images for Anaconda-based Linux-Distributions

<https://github.com/weldr/lorax>

- **Weldr**

Integration of the Lorax Toolset into Cockpit

<https://github.com/weldr/welder-web>

# Composer

## Cockpit

RED HAT ENTERPRISE LINUX root

[Back to Blueprints](#) > [demo1](#) > Edit Blueprint

[Commit](#) [Discard Changes](#) [Create Image](#)

### demo1

#### Available Components

python

Name: python X Clear All Filters

1 - 50 of 319 < 1 of 7 >

- Python 3 bindings for openscap
- openwsman-python3**  
Python bindings for openwsman client API Add latest version (3.6.8)
- platform-python**  
Internal interpreter of the Python programming language +
- platform-python-coverage**  
Code coverage testing module for Python 3 +
- platform-python-debug**  
Debug version of the Python runtime +

#### Blueprint Components

Name Filter by Name... ↓ ↺ ↻

**Selected Components** 7 **Dependencies** 171

- ☐ **bind-utils**  
Utilities for querying DNS name servers  
Version 9.11.4 Release 16.P2.el8
- ☐ **mc**  
User-friendly text console file manager and visual shell  
Version 4.8.19 Release 9.el8
- ☐ **net-tools**  
Basic networking tools  
Version 2.0 Release 0.51.20160912git.el8
- ☐ **nginx-all-mo...**  
A meta package that installs all available Nginx  
Version 1.14.1 Release 8.module+el8+2505+fe936cef

# UPGRADING SERVERS TO RED HAT ENTERPRISE LINUX 8



LeApp is an extendable framework designed to assist administrators with application modernization. It supports Red Hat Enterprise Linux, CentOS, and Fedora, and is the preferred method for in-place upgrades from RHEL 7 to RHEL 8.

## LeApp Modernization Framework

Enables users to modernize existing workloads without disrupting them.

Three methods: upgrade in place, migrate to new place, or containerize.

Designed modular architecture to replace the preupgrade-assitant tool.

Various migration strategies and application-specific logic are kept in independent modules or plugins.

LeApp is message-driven, for passing data between actors. The execution of actors is dependent on the data produced by other actors running before them.

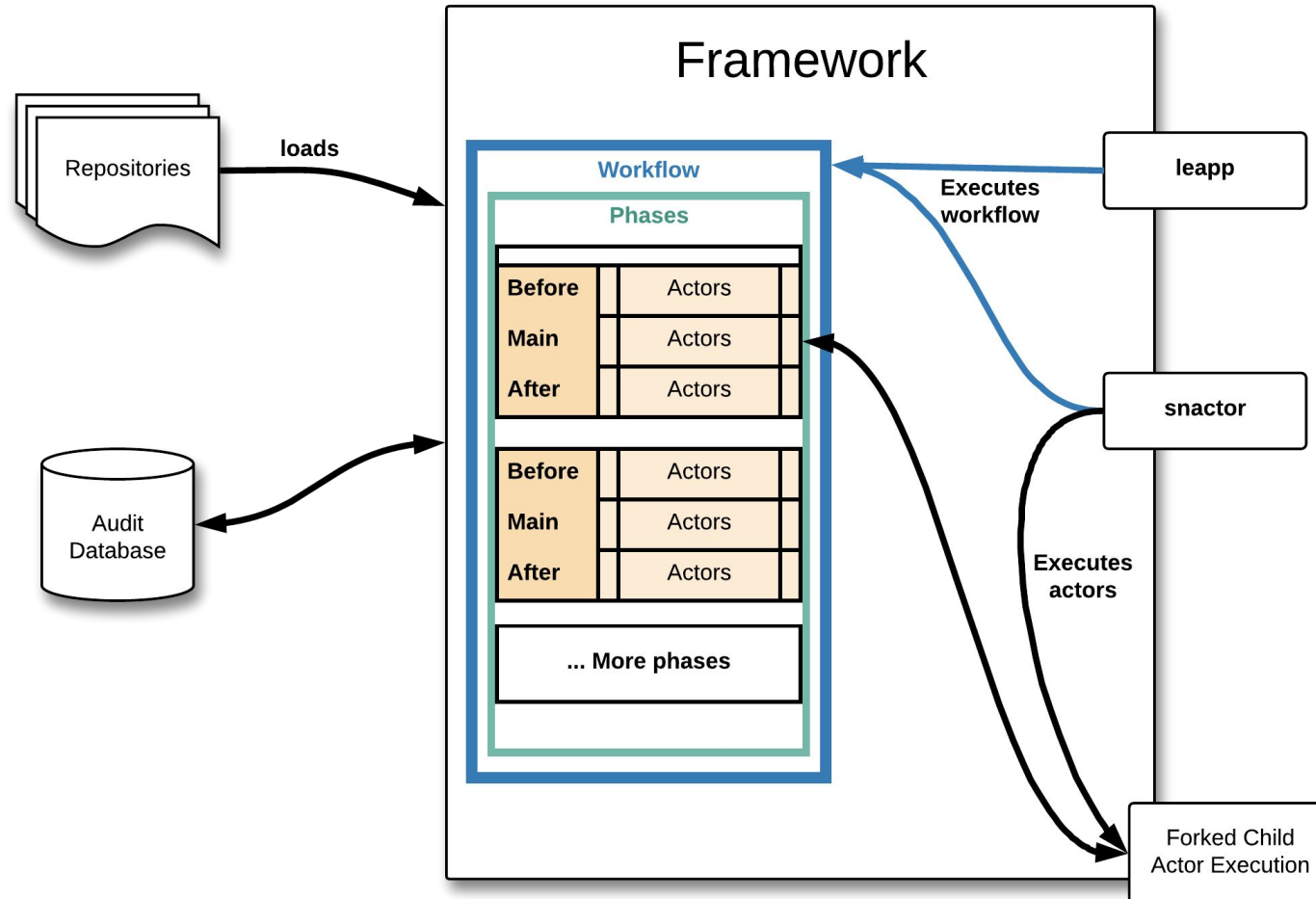
# LeApp Components



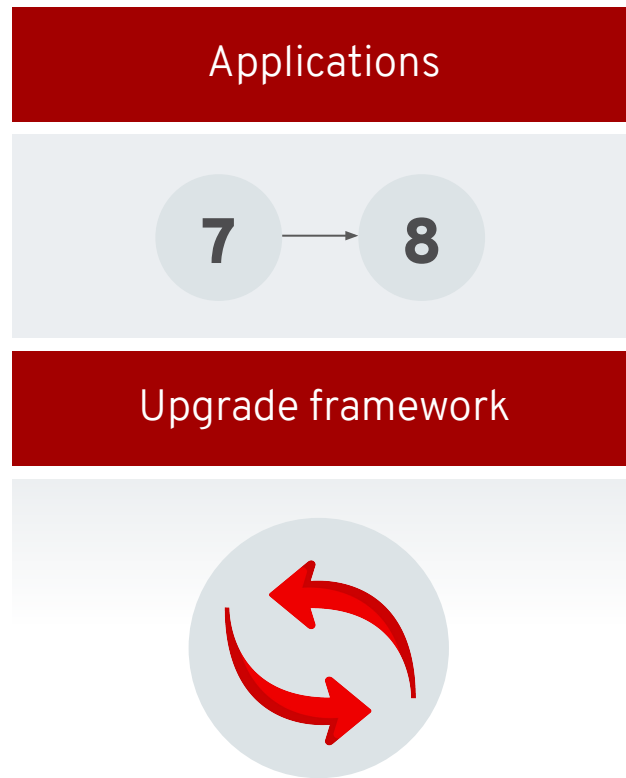
- Workflow: describes an entire upgrade process.
- Phase: a section of the workflow dedicated to a specific part of the upgrade.
- Stage: phases are broken into stages, Before, Main, and After.
- Actor: a step in the workflow that performs a task. Actors can be tagged which allows them to be included in a workflow automatically by the framework.
- Tag: allows the framework to locate and execute actors.
- Message: used to transfer information between actors.
- Model: defines the structure of the data sent in messages. Models use Topics to group the data.
- Topic: defines the subject area for Model data.



# LeApp Components



# In-place upgrades for your systems



## Reduced migrations

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

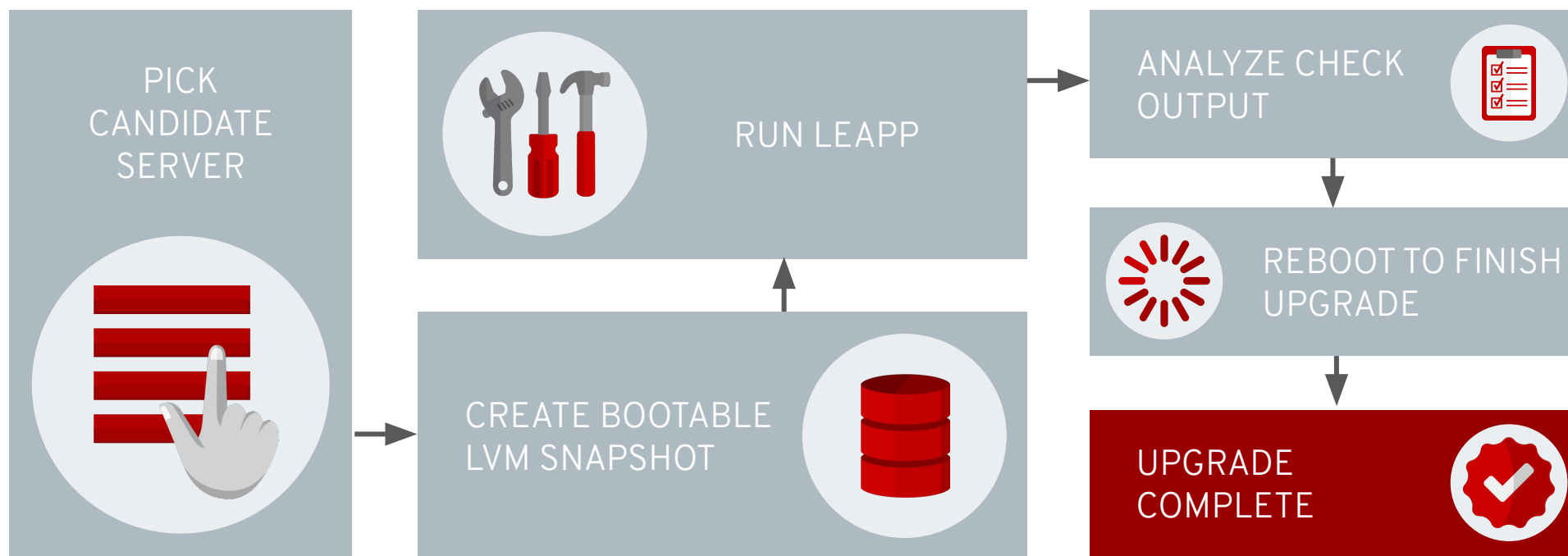
## Easy rollback options

Combine with bootable LVM snapshots for safety

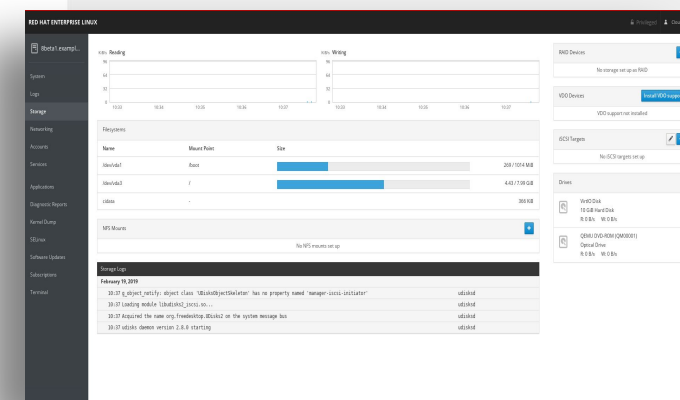
## Improved framework

Get better analysis and a simplified process with a more extensible framework

## Can I upgrade this host?



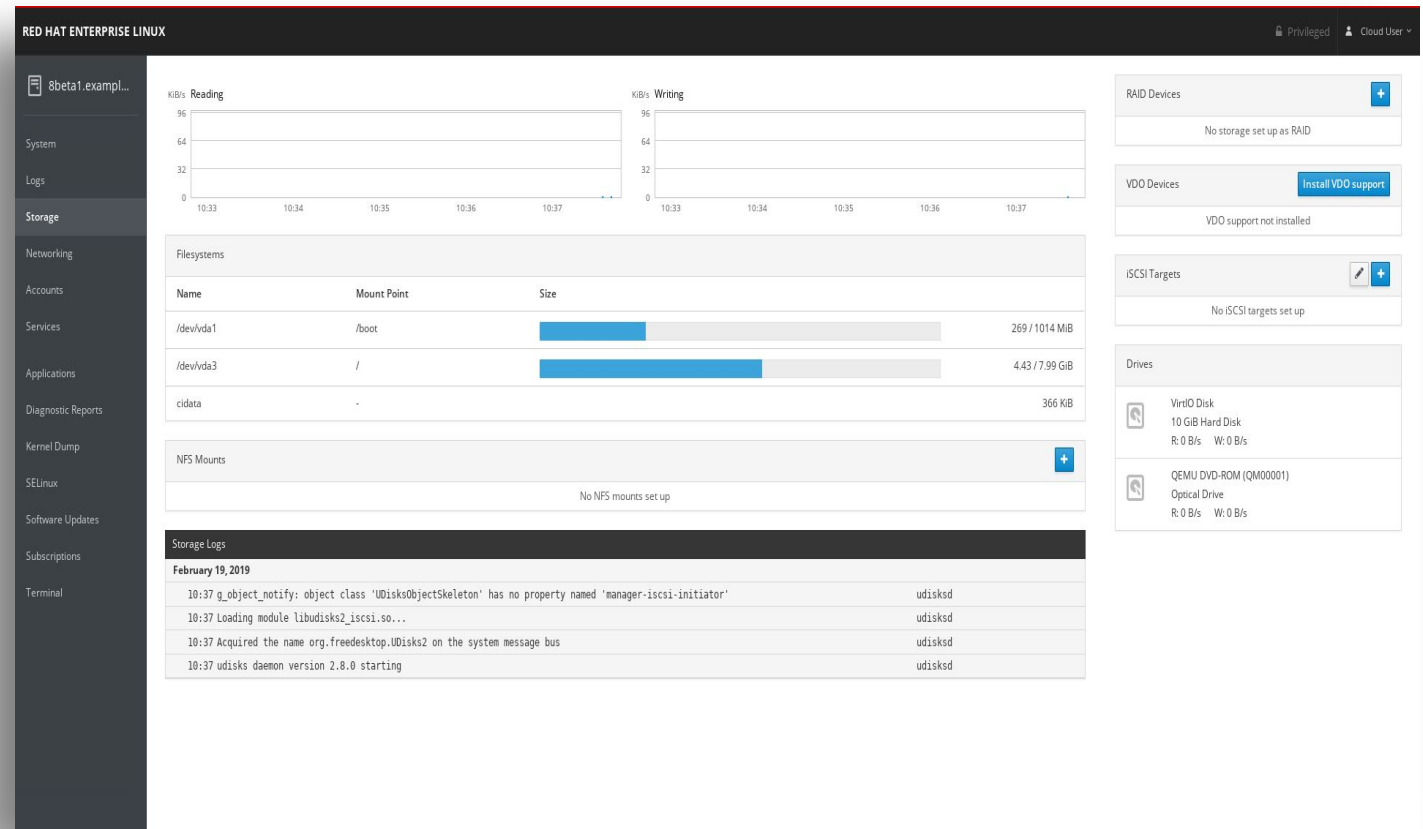
# Server Management



# Red Hat Web Console

## Cockpit

- › Feature rich interface
- › More accessible for non-Linux experts
- › Integration with Network Bound Disk Encryption, Single Sign-on and more
- › Added support for virtual machines, session recording, and more

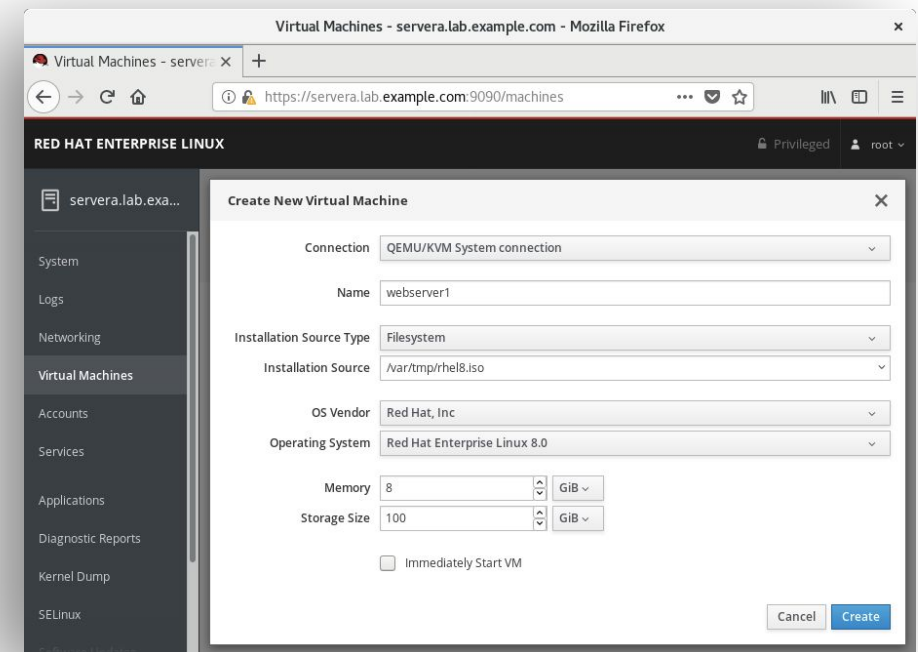


# Cockpit

## Virtual machine management

Cockpit replaces virt-manager, which is deprecated. Red Hat recommends Cockpit for virtual machine management. virt-manager will be removed in a subsequent release.

- Cockpit in RHEL 8 offers basic functionality for administering virtual machines. Install the cockpit-machines package to enable the Cockpit component for managing virtual machines.
- For more advanced configurations, use the libvirt tools such as virsh or virt-install.
- The virt-manager graphical interface is still available but deprecated, and will be removed in a subsequent release.



# RHEL Virtualization Available as a Module

For convenience, you can install the virtualization software using the new yum module command.

The virt Yum module has one stream called rhel and a single default profile.

A separate virt stream will be shipped via another repository for layered products such as Red Hat Virtualization (RHV), which will be able to receive major updates on a different cadence than RHEL major releases usually allow.

You can also install the virtualization packages individually using traditional yum commands.

# Storage





# STRATIS STORAGE MANAGER

## MANAGING LAYERED STORAGE WITH STRATIS

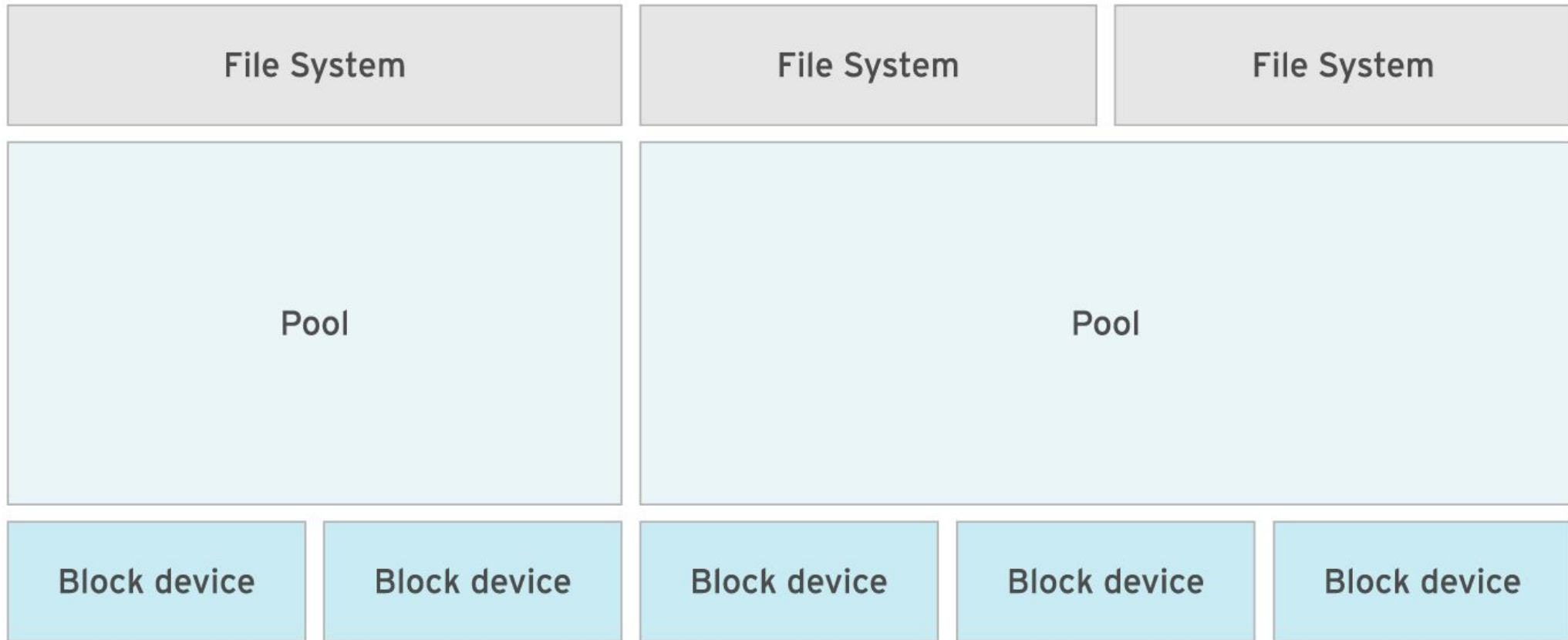
Red Hat Enterprise Linux 8 includes the Stratis storage manager, which supports management of collections of block devices to create flexible file systems.

The combined file system and volume management functionality first learned in ZFS or Btrfs is now available in Stratis.

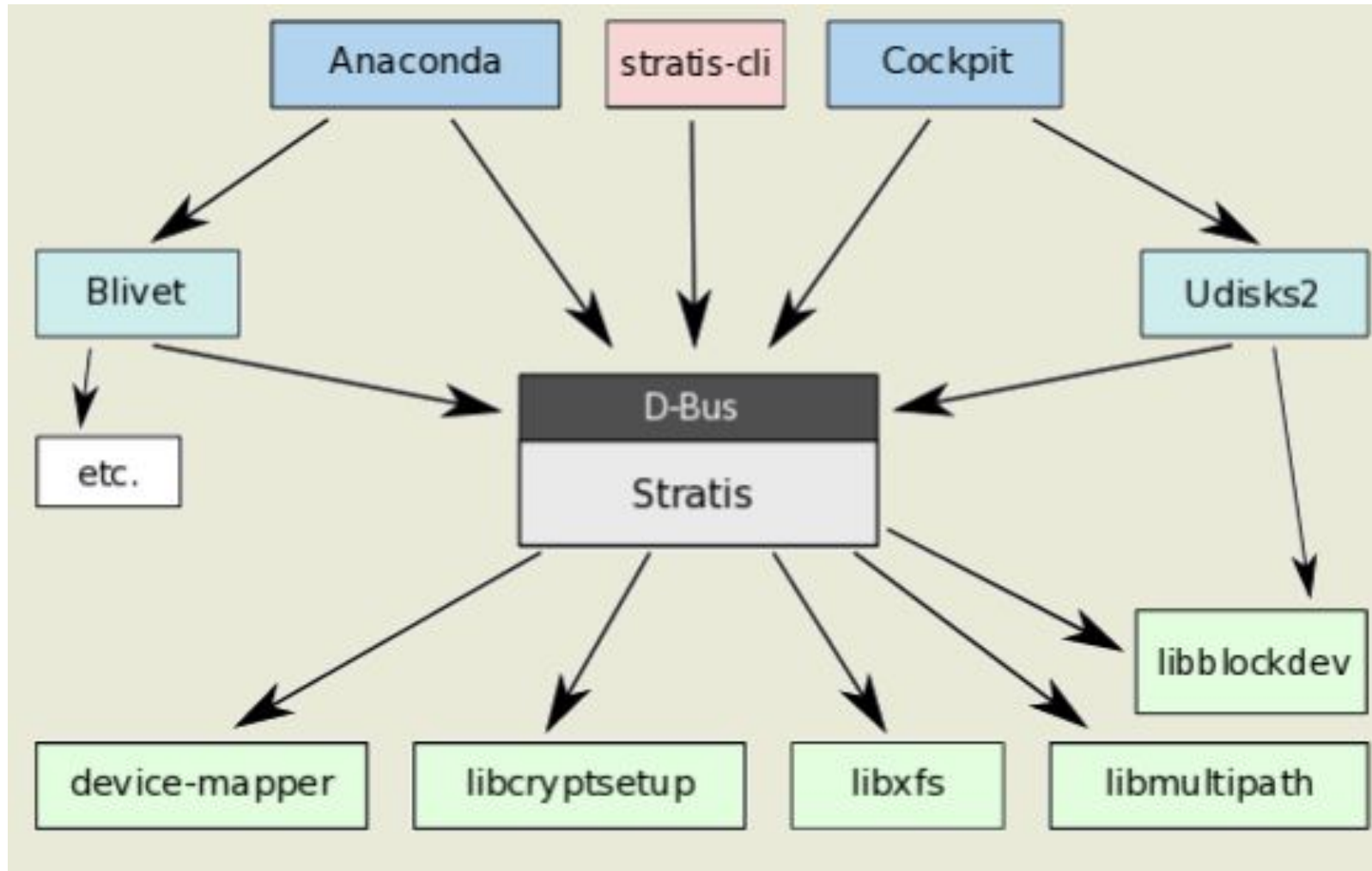
**Both the Btrfs and the ZFS file systems are unsupported and no longer available in Red Hat Enterprise Linux 8.**

Stratis also provides advanced storage features like thin provisioning, snapshotting, and monitoring.

# STRATIS STORAGE MANAGER



# STRATIS STORAGE MANAGER



# STRATIS STORAGE MANAGER

Stratis is a volume managing file system (VMF).

Volume managing file systems integrate the file system in the volume itself, in contrast with LVM where the volume requires a file system on top of it.

Stratis' design assumes SSD as the default storage type, or at least as a cache tier, so the focus of the design is on improving flexibility and reliability.

BtrFS and ZFS are no longer supported or available in Red Hat Enterprise Linux 8.

Stratis provides advanced features like thin provisioning, snapshotting, and monitoring.

# STRATIS STORAGE MANAGER

```

[root@rhel8beta ~]# lsblk

```

Device	Size	Used	Filesystem	Mount	
vda	252:0	0	156	0	disk
└─vda1	252:1	0	16	0	part /boot
└─vda2	252:2	0	146	0	part
└─rhel-root	253:0	0	12.5G	0	lvm /
└─rhel-swap	253:1	0	1.5G	0	lvm [SWAP]
vdb	252:16	0	10G	0	disk
vdc	252:32	0	10G	0	disk
vdd	252:48	0	10G	0	disk

```

[root@rhel8beta ~]# stratis pool create test-pool /dev/vdb /dev/vdc
[root@rhel8beta ~]# stratis pool add-data test-pool /dev/vdd
[root@rhel8beta ~]# stratis blockdev list

```

Pool Name	Device Node	Physical Size	State	Tier
test-pool	/dev/vdb	10 GiB	In-use	Data
test-pool	/dev/vdc	10 GiB	Not-in-use	Data
test-pool	/dev/vdd	10 GiB	Not-in-use	Data

```

[root@rhel8beta ~]# stratis pool list

```

Name	Total Physical Size	Total Physical Used
test-pool	30 GiB	60 MiB

```

[root@rhel8beta ~]# stratis filesystem create test-pool test-filesystem1
[root@rhel8beta ~]# stratis filesystem create test-pool test-filesystem2
[root@rhel8beta ~]# stratis filesystem list

```

Pool Name	Name	Used	Created	Device	UUID
test-pool	test-filesystem1	546 MiB	Jan 11 2019 16:27	/stratis/test-pool/test-filesystem1	6b24bdd3c501408d8385bdebe00759c2
test-pool	test-filesystem2	546 MiB	Jan 11 2019 16:28	/stratis/test-pool/test-filesystem2	e15cd49331944fe799c0a5ebcbff23f2

```

[root@rhel8beta ~]# mkdir /test-filesystem1
[root@rhel8beta ~]# mkdir /test-filesystem2
[root@rhel8beta ~]# mount /stratis/test-pool/test-filesystem1 /test-filesystem1
[root@rhel8beta ~]# mount /stratis/test-pool/test-filesystem2 /test-filesystem2
[root@rhel8beta ~]#

```

# VIRTUAL DATA OPTIMIZER

## Reducing Data Footprint with VDO

- Virtual data optimizer (VDO) is a Linux device mapper driver to reduces disk space usage on block devices, and minimize replication bandwidth.
- VDO includes two kernel modules, the kvdo module to control data compression, and the uds module for deduplication.
- VDO supports both inline data deduplication and compression.

## VIRTUAL DATA OPTIMIZER

- VDO reduces the data footprint on storage in three phases: zero-block elimination, deduplication of redundant blocks, and data compression.
- VDO removes blocks which only include zeros, and keeps their metadata.
- The universal deduplication service (UDS) kernel module reviews the available VDO metadata to detect duplicated blocks. If a duplicated block is found, this block points to the metadata of the already available block.
- When done with zero-block elimination and deduplication, the kvdo kernel module compresses blocks using LZ4 compression and groups them on 4FKB blocks.

# NFS Enhancements

The default NFS version in Red Hat Enterprise Linux 8 is 4.2, the NFSv4 and NFSv3 major versions are supported, NFSv2 is no longer supported.

The NFS configuration file is `/etc/nfs.conf`. The `/etc/sysconfig/nfs` configuration file is deprecated.

Introduction of the new `nfsconf` tool to manage NFS configuration files.

The `autofs` service uses `/etc/auto.master` (master map) as its default configuration file.

The `amd` format is recognized as well.

Red Hat Enterprise Linux 8 removes the `nfsnobody` user, and changes the UID and GID of the `nobody` user and group to 65534. Any UID or GID not being available maps to the `nobody` user and group.

NFSv4 no longer requires `rpcbind` service to operate, eliminating the need of UDP connections.



## NFS Enhancements

Server-side copy, enables the NFS client to efficiently copy data inside NFS file systems, without wasting network resources.

Sparse files, enables files to have multiple holes, which are data blocks consisting only of zeroes.

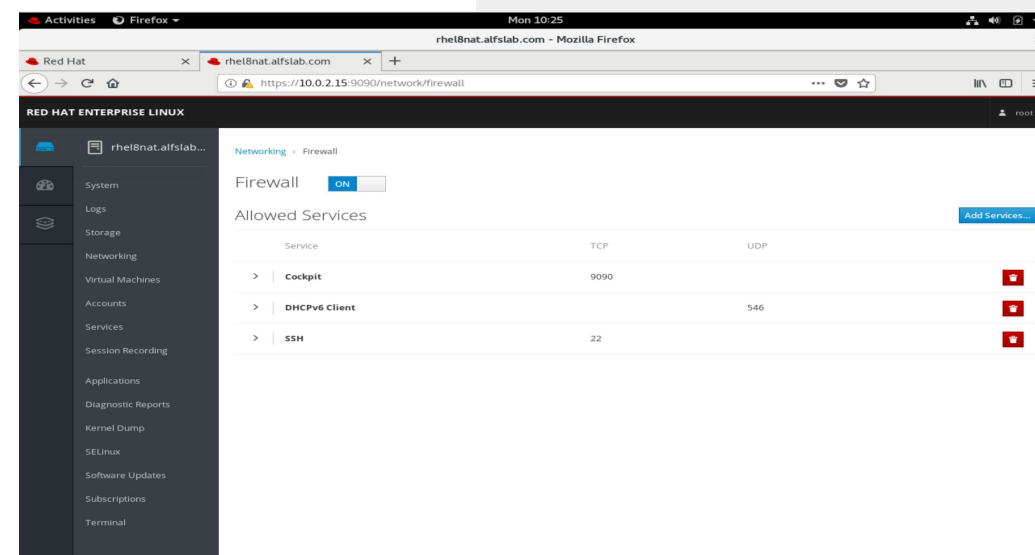
These are transferred as zeroes improving the overall speed.

Space reservation, allows to reserve free space on NFS servers, which prevents them to run out of space.

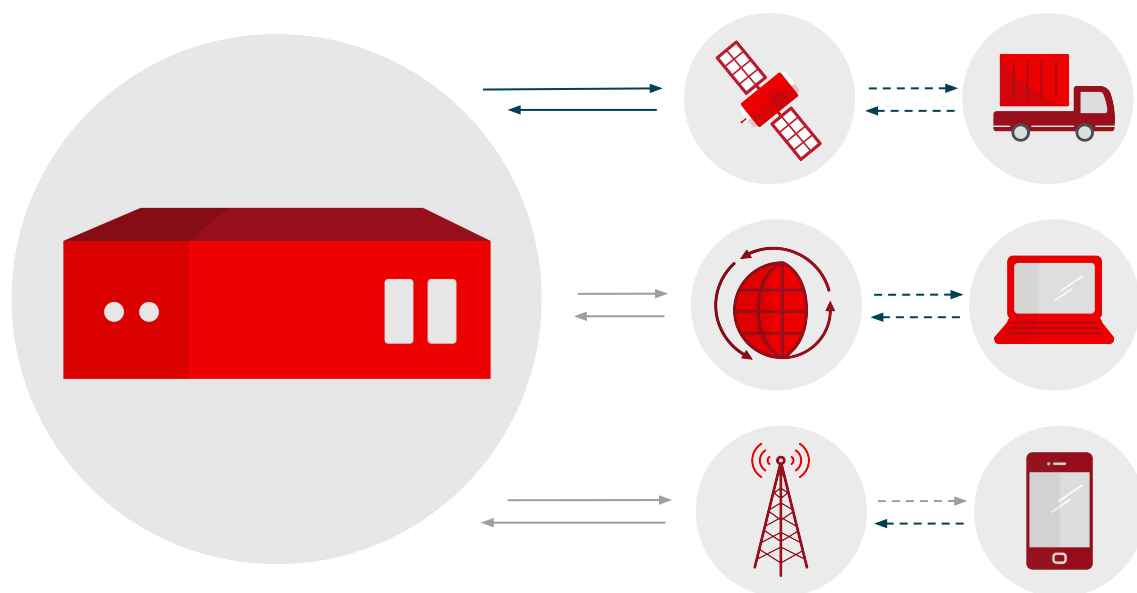
Labeled NFS, enforces data access rights and enables SELinux on NFS file systems.

Layout enhancements, which enables some Parallel NFS (pNFS) servers to collect better performance statistics.

# Network



# Improve network performance with bandwidth and round-trip propagation time congestion algorithm



## End-to-end performance

Link capacity calculation and management at server do not require client end modifications

## High-latency links

Improved performance over other algorithms on networks with high latency and congestion

## More choices

An improved network stack combined with BBR and other algorithms lets you select the highest performance combinations

# MANAGING SERVER FIREWALLS IN RHEL 8

Firewalld, the firewall management tool in RedHat Enterprise Linux, uses nftables as its new default firewall back end. In RedHat Enterprise LinuxF8, nftables replaces iptables, which is now deprecated.

The nft command now replaces the multiple iptables tools, iptables, ip6tables, arptables, and ebtables, in a unified, consistent, and easier to use command. nftables is also more efficient and can perform multiple actions in a single rule.

## Nftables

- Firewalld uses nftables as its back end.
- The nft command replaces the iptables, ip6tables, arptables, and nftables commands.
- Firewalld is the recommended way to manage the firewall, over the low-level nft command.
- The iptables commands are links to the xtables-nft-multi command which accepts the iptables syntax but creates nftables rules instead.

# MANAGING SERVER FIREWALLS IN RHEL 8

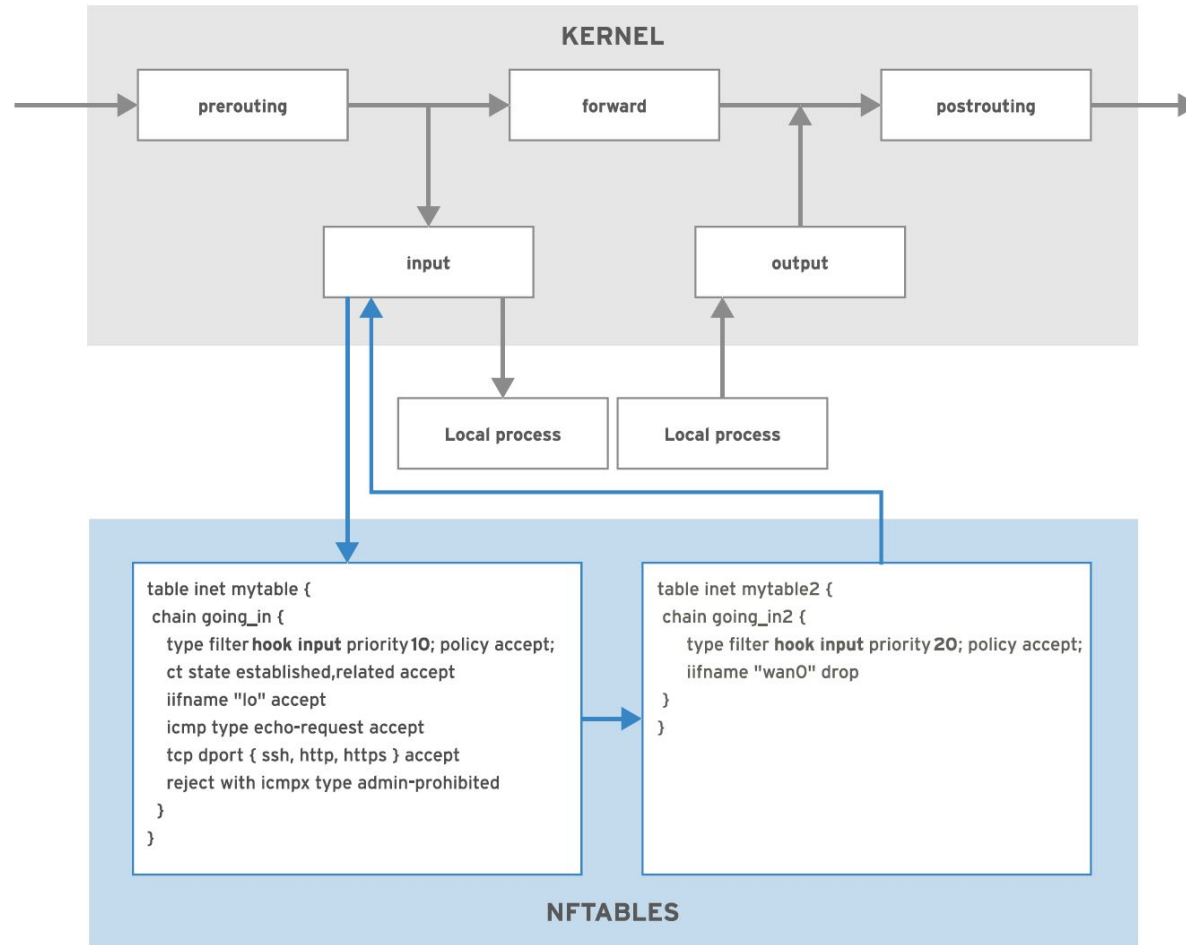


Figure 7.1: Attaching nftables chains to the input packet processing stage in the kernel

# CONFIGURING SERVER NETWORKING WITH NETWORKMANAGER

NetworkManager is the preferred network configuration tool in Red Hat Enterprise Linux 8. It can handle complex configurations and layered products, such as OpenStack, OpenShift, or Red Hat Virtualization, now relies on it.

## Configuring the Network with NetworkManager

- Red Hat recommends using NetworkManager for network configuration.
- NetworkManager can now handle complex configurations, such as Open vSwitch or SR-IOV.
- Layered products, such as OpenStack, are using NetworkManager.
- Multiple front ends are available: Cockpit, nmcli, nmtui, or the Network RHEL System Role.
- The systemd network service and the network scripts are not available anymore.
- ifup and ifdown are links to NetworkManager scripts.

# Virtualisation

# CONFIGURING VIRTUAL MACHINES

The Kernel-based Virtual Machine (KVM) kernel module and the QEMU emulator are the basis of virtualization in Red Hat Enterprise Linux 8. In this release, QEMU can now emulate the Intel Q35 motherboard chipset, which offers a better hardware platform for modern virtualized operating systems.

## New Q35 Virtual Machine Type

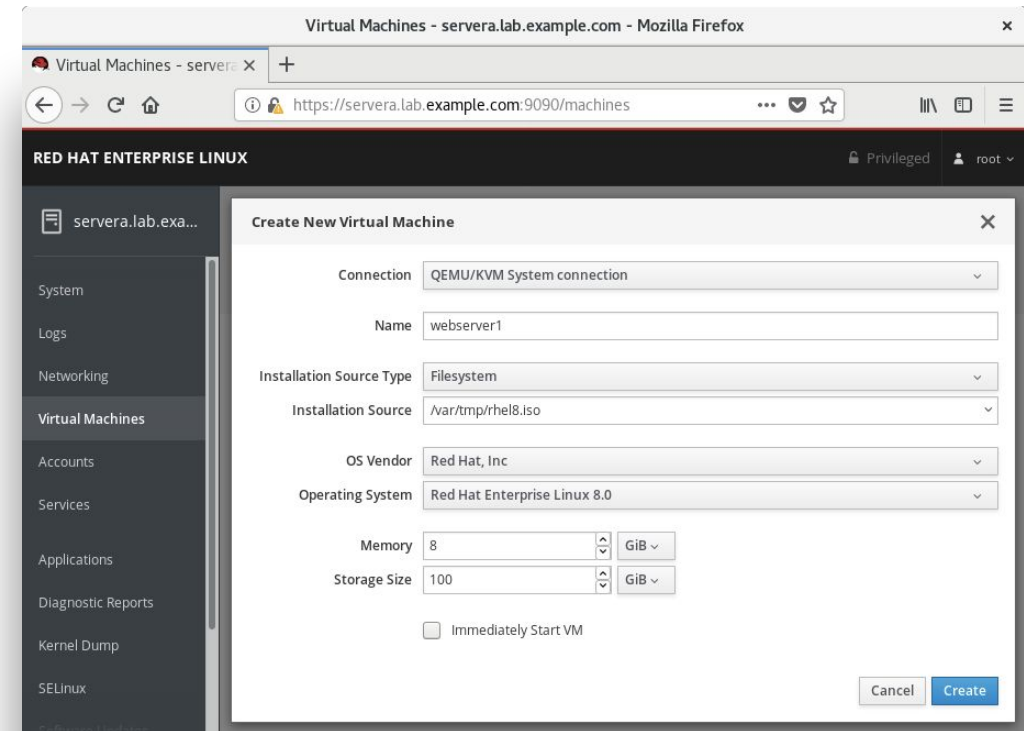
- In addition to the previous Intel 440FX machine type, QEMU now emulates the Intel Q35 chipset
- and features.
- The Q35 chipset emulation provides more current hardware devices.
- Q35 provides a PCI Express bus and supports secure boot.
- Q35 supports PCI Express pass-through and simplifies physical to virtual (p2v) migrations.



# CONFIGURING VIRTUAL MACHINES

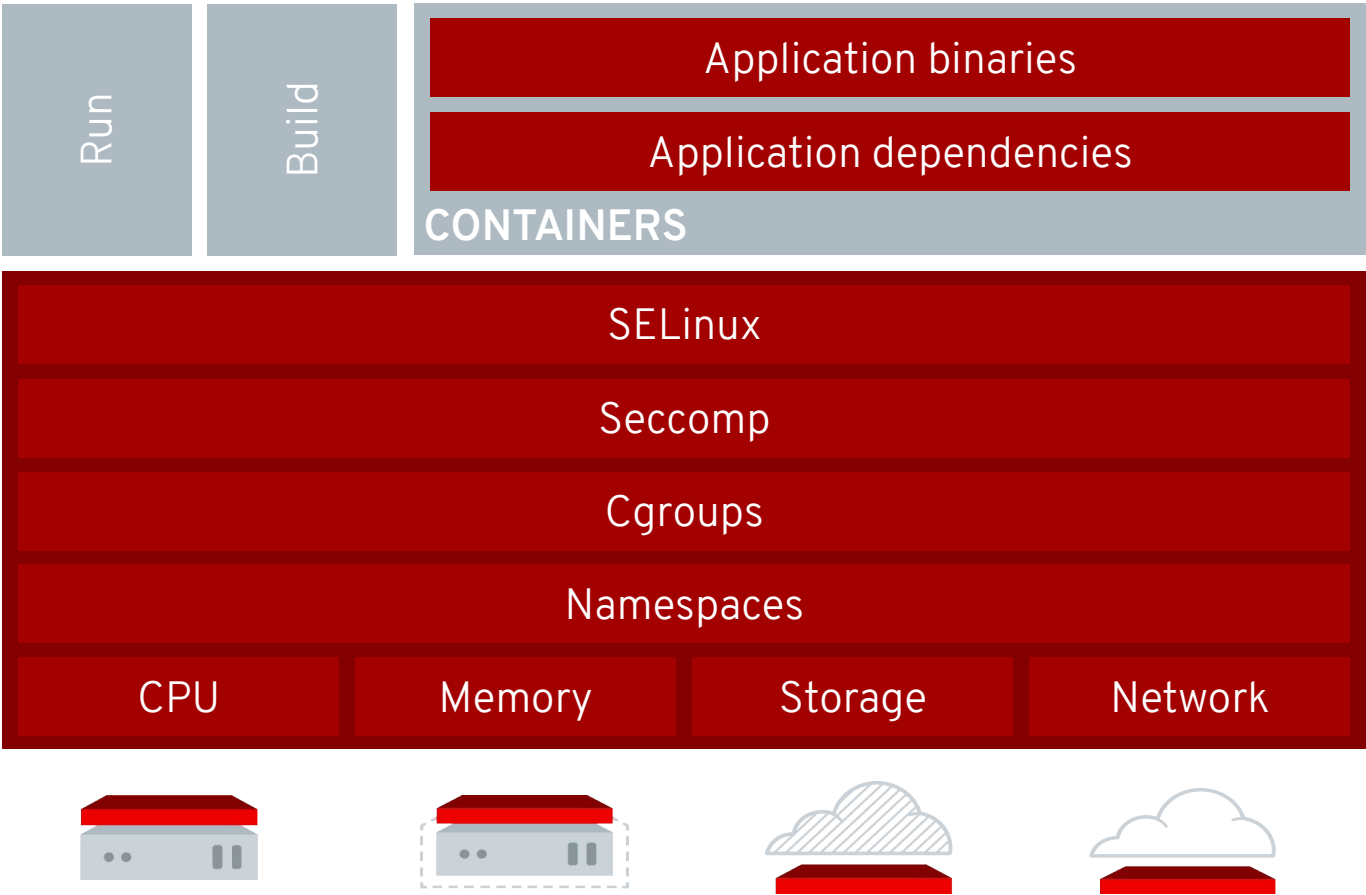
## NOTE

Cockpit replaces virt-manager, which is deprecated. Red Hat recommends Cockpit for virtual machine management. virt-manager will be removed in a subsequent release.

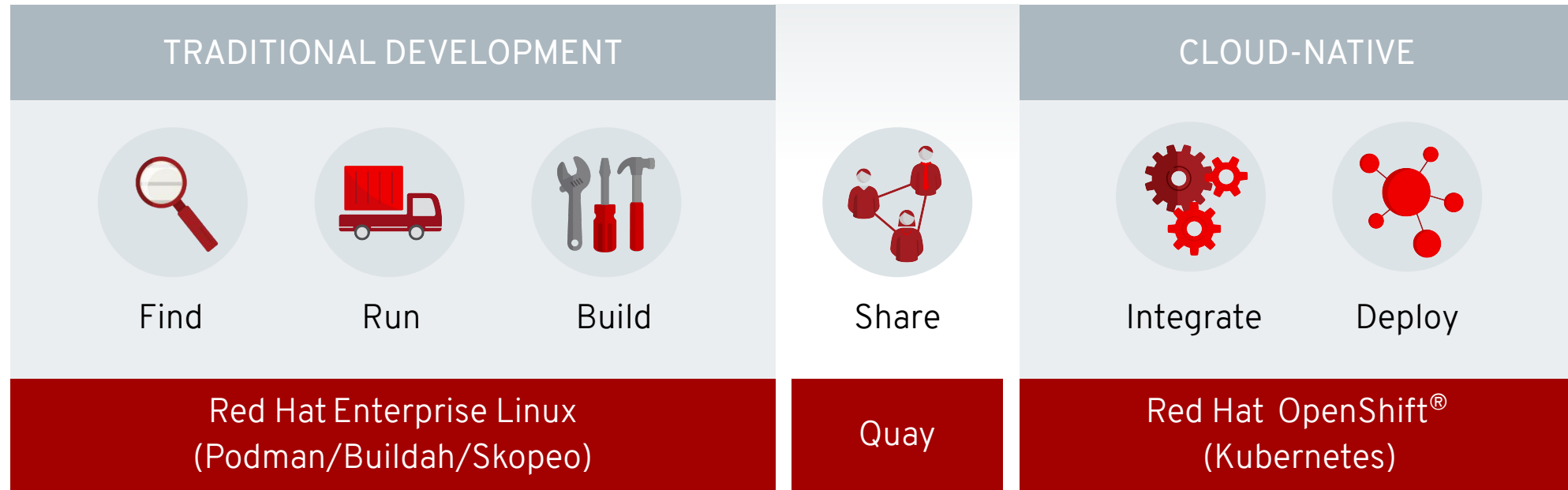


# Container

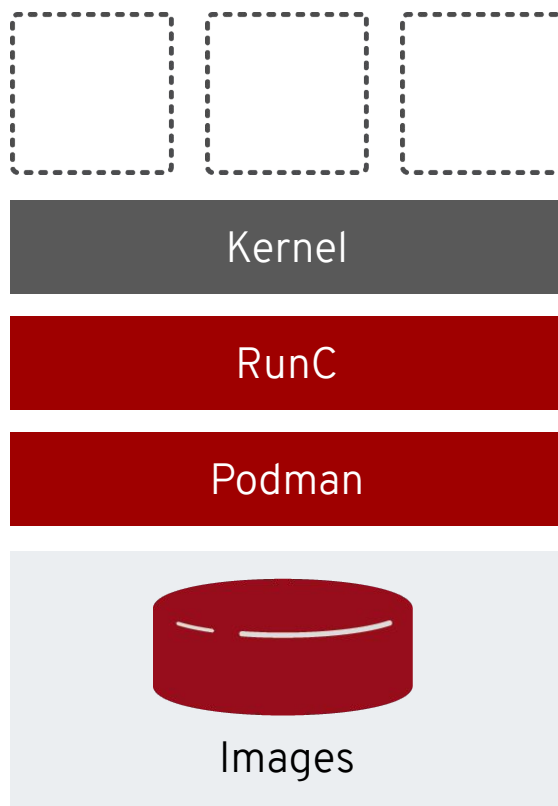
# Containers are Linux



# Powering the adoption of containerized workloads



# Manage containers with Podman



## **Fast and lightweight**

No daemons required

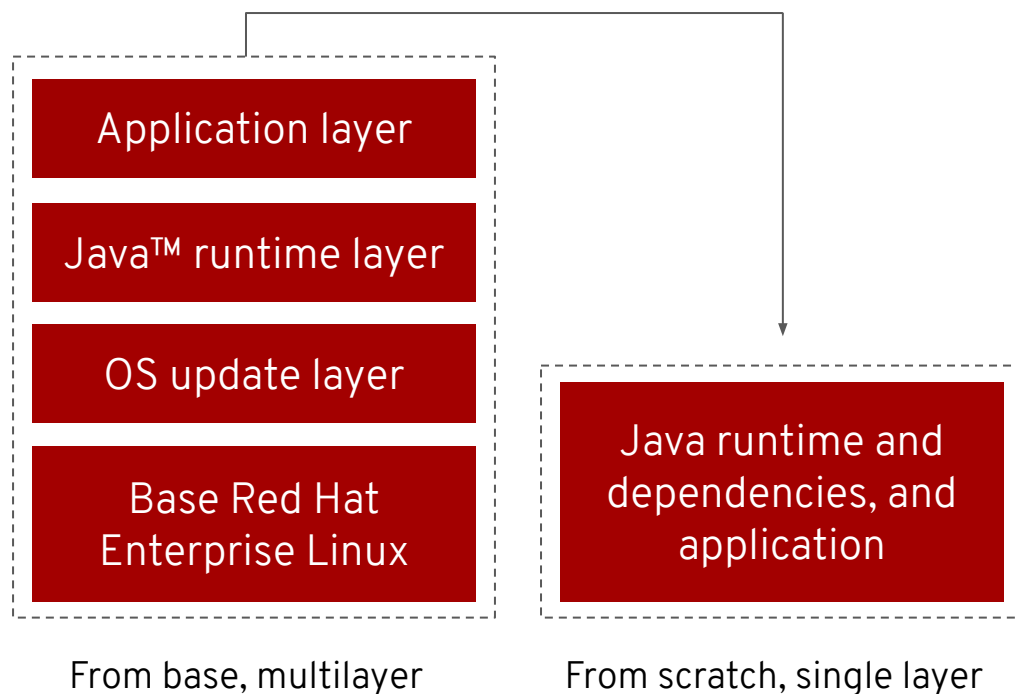
## **Advanced namespace isolation**

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



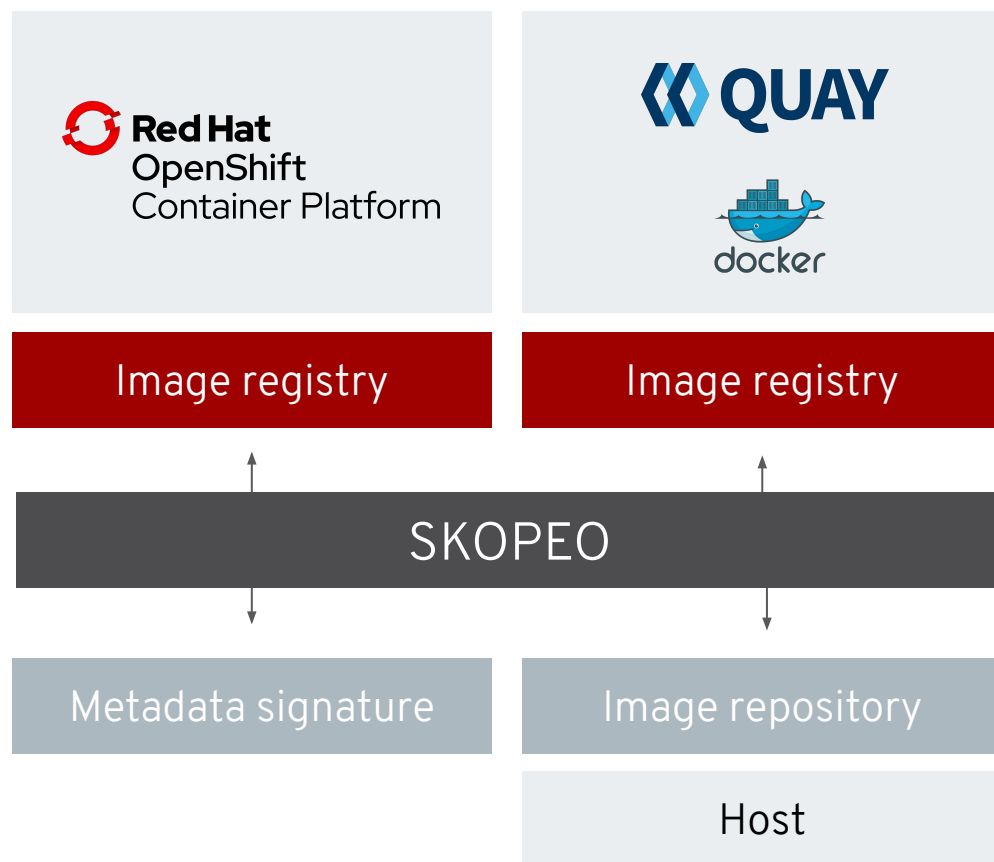
## More control

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

# Inspect and transport images with Skopeo



## 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

# CoreOS





## Red Hat CoreOS

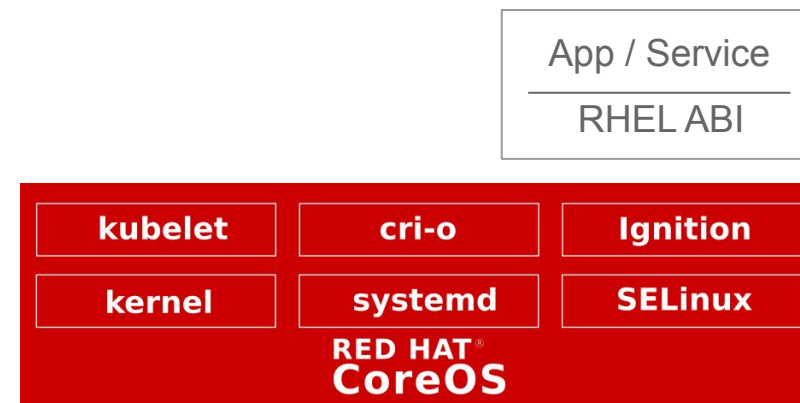
**Combining the innovations of Container Linux and Atomic with the stability and ecosystem of RHEL**

**Fully integrated and delivered via OpenShift.**

- **Small footprint, derived from RHEL**
  - **~400 packages**
- **Fast provisioning: clusters deploy in minutes**
- **Simplified, cluster-centric updates and upgrades**
- **Managed and automated via operators**

# Red Hat CoreOS

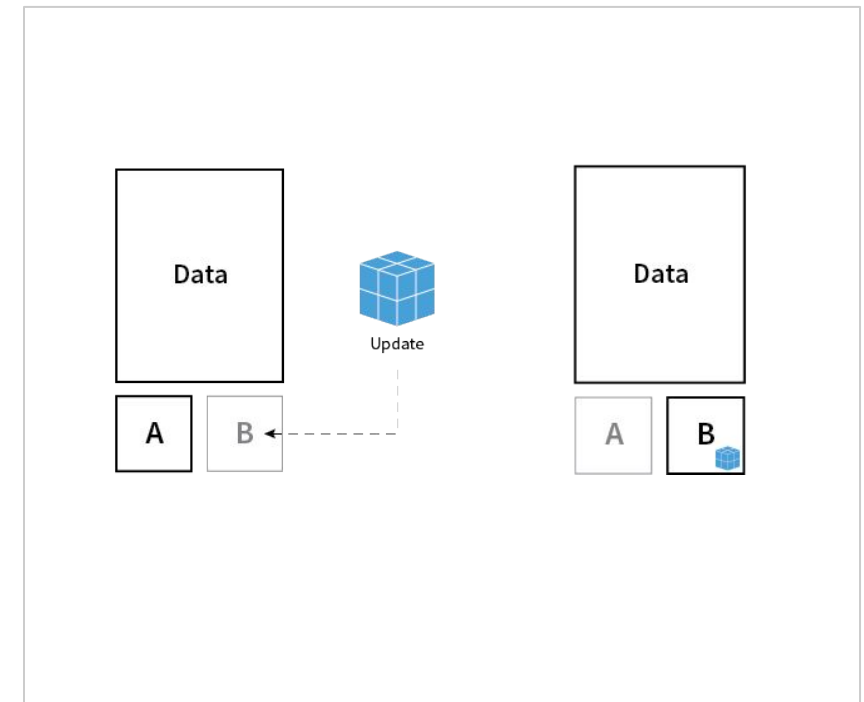
- Full support for the RHEL ABI & container ecosystem
- An immutable host, delivered and managed via OpenShift
  - Aligned lifecycle and release cadence
  - Updates & upgrades deployed via operators
- UX inspired by Container Linux
  - Read-only OS binaries in /usr
  - Integrated container & kubernetes stack
  - One-touch provisioning with Ignition



## Transactional Updates via rpm-ostree

Transactional updates ensure that the Red Hat CoreOS is never altered during runtime. Rather it is booted directly into an always “known good” version.

- Each OS update is versioned and tested as an complete image.
- OS binaries (/usr) are read-only
- Updates encapsulated in container images
- file system and package layering available for hotfixes and debugging



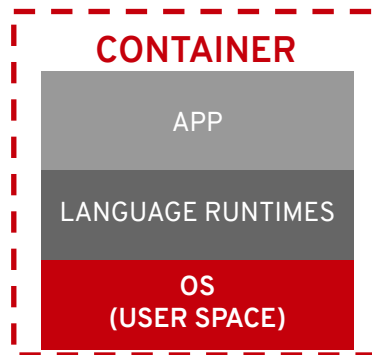
# Universal Base Image (UBI)

# THE RED HAT UNIVERSAL BASE IMAGE



“Build once, deploy anywhere”

# THE BASE IMAGE FOR ALL OF YOUR NEEDS



The Red Hat Universal Base Image is based on RHEL and made available by a new end user license agreement .

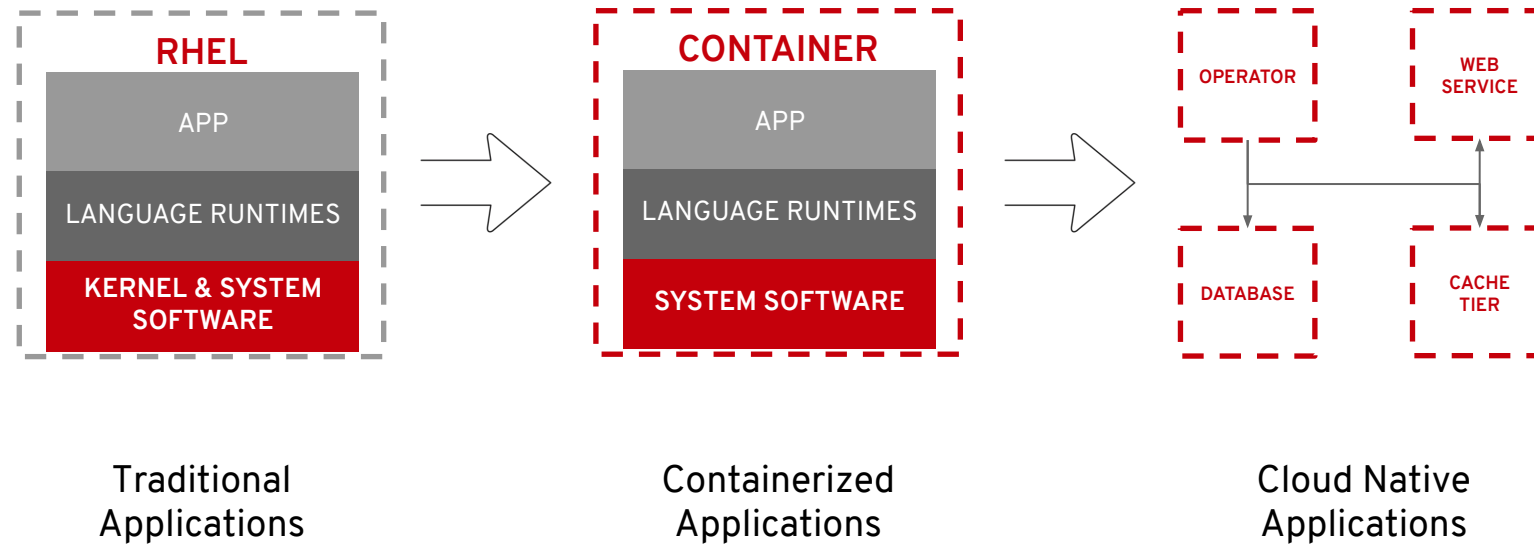
## Development

- Minimal Images
- Programming languages
- Enables a single CI/CD chain

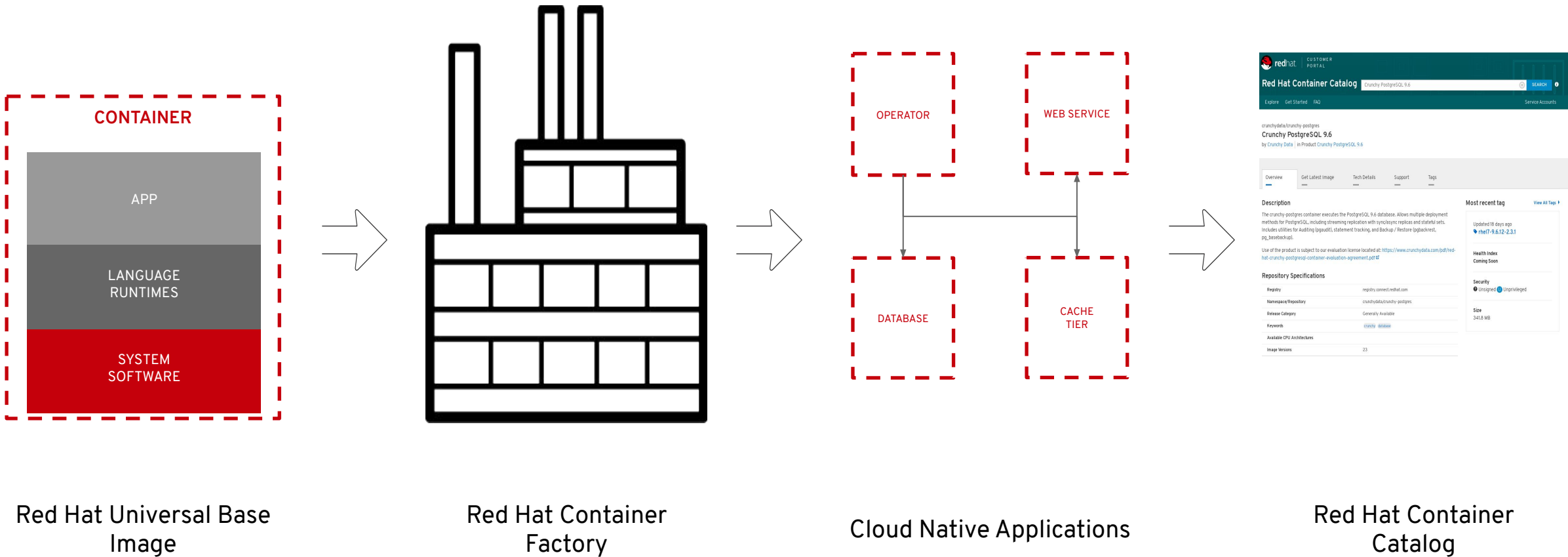
## Production

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

# THE BASE IMAGE FOR ALL OF YOUR NEEDS

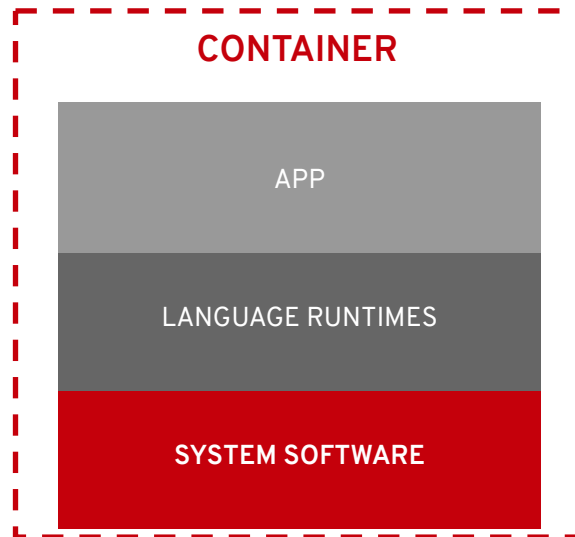


# BEHIND THE SCENES

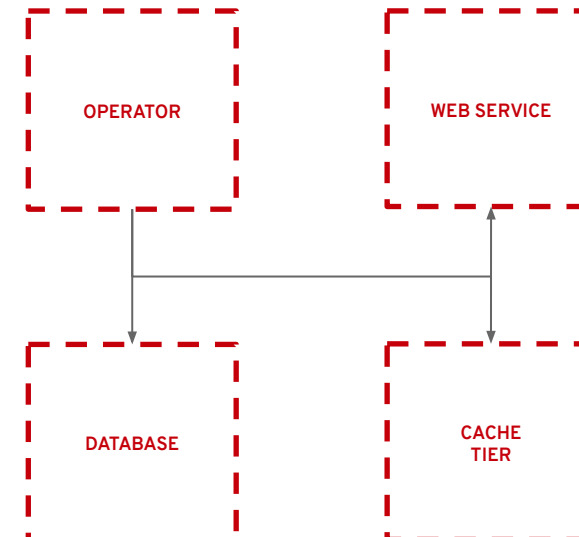




# CONTAINER CERTIFICATION

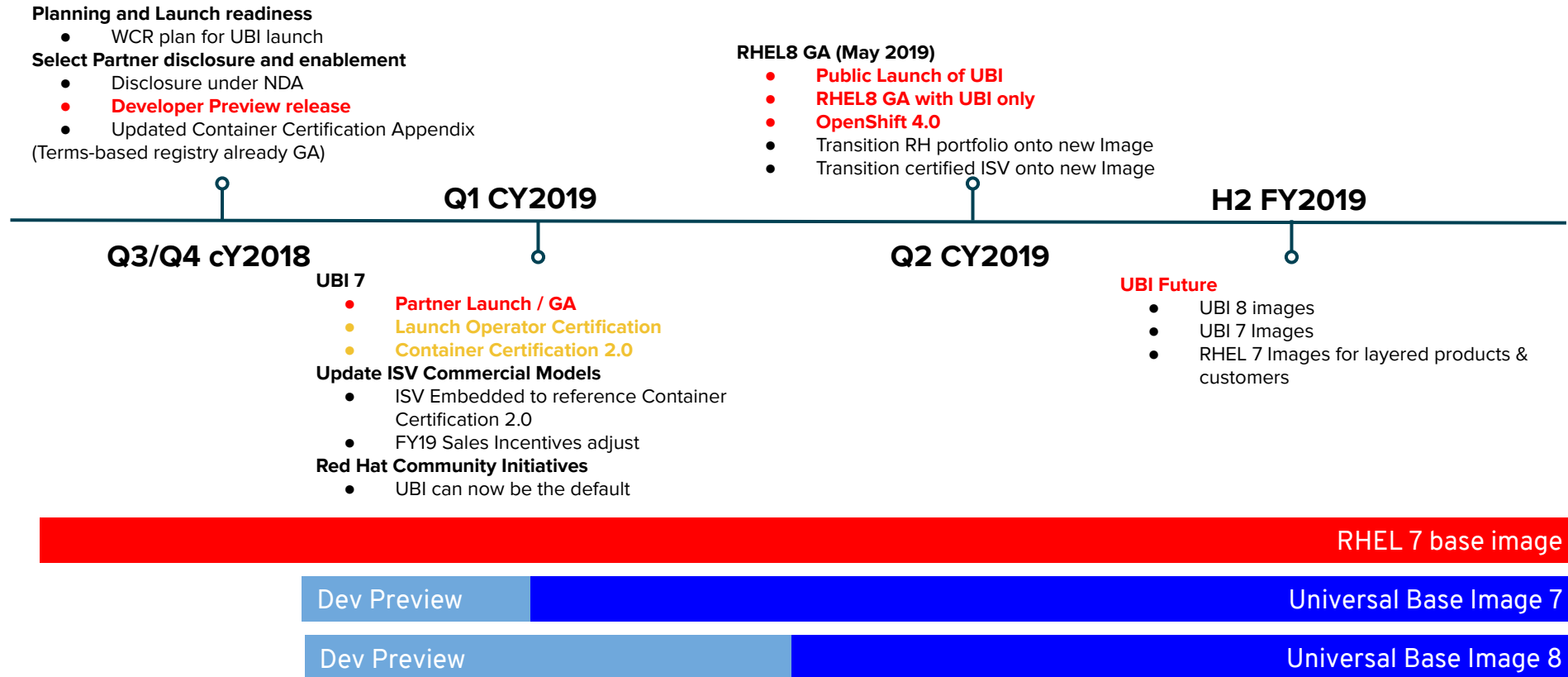


Container  
Certification



Operator  
Certification

# UNIVERSAL BASE IMAGE - TIMELINE



# Security and Compliance



# Red Hat Enterprise Linux



Common Criteria & FIPS 140-2\*

## **AUTOMATE**

regulatory compliance  
and security configuration  
remediation.

## **GAIN**

peace of mind with the  
Red Hat open source  
secure supply chain.

## **RECEIVE**

continuous vulnerability  
security updates.

## **SECURE**

and deliver resources.

# Configuring systemwide cryptographic policies



## Central configuration

Set acceptable algorithms from a single tool

## Improved consistency

Covers multiple cryptographic providers and consumers like TLS, kerberos, and Java

## Built-in policies

Including legacy systems requiring 64-bit security and FIPS allowed or approved algorithms

# USBGUARD

## FLEXIBLE

rules for device description

## WHITELIST OR BLACKLIST

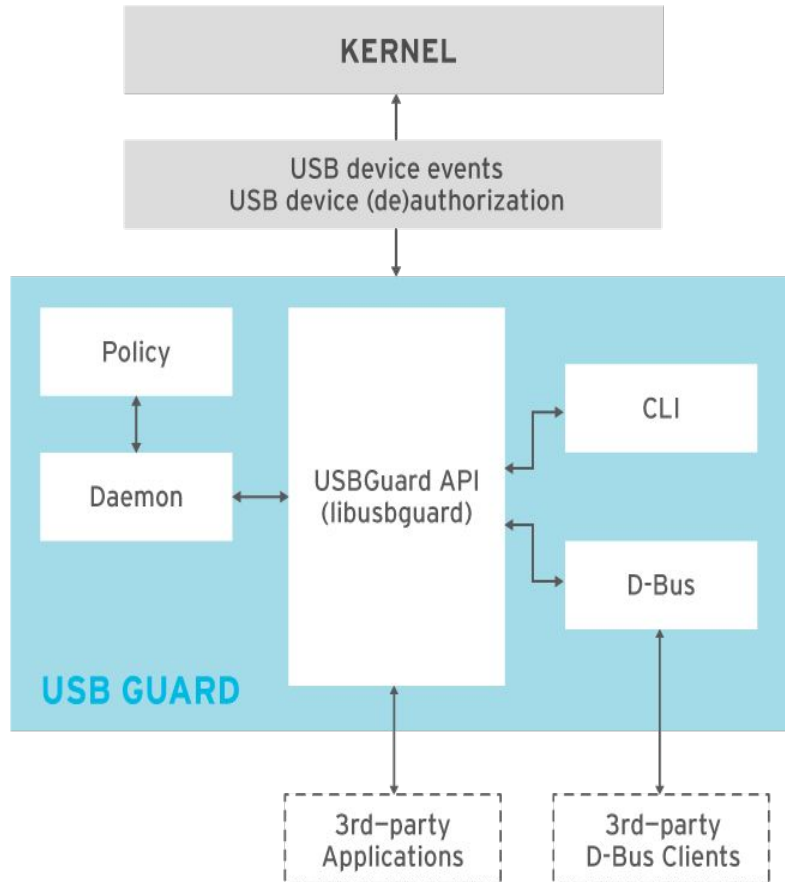
by device or class

## CHANGE DEFAULT BEHAVIOR

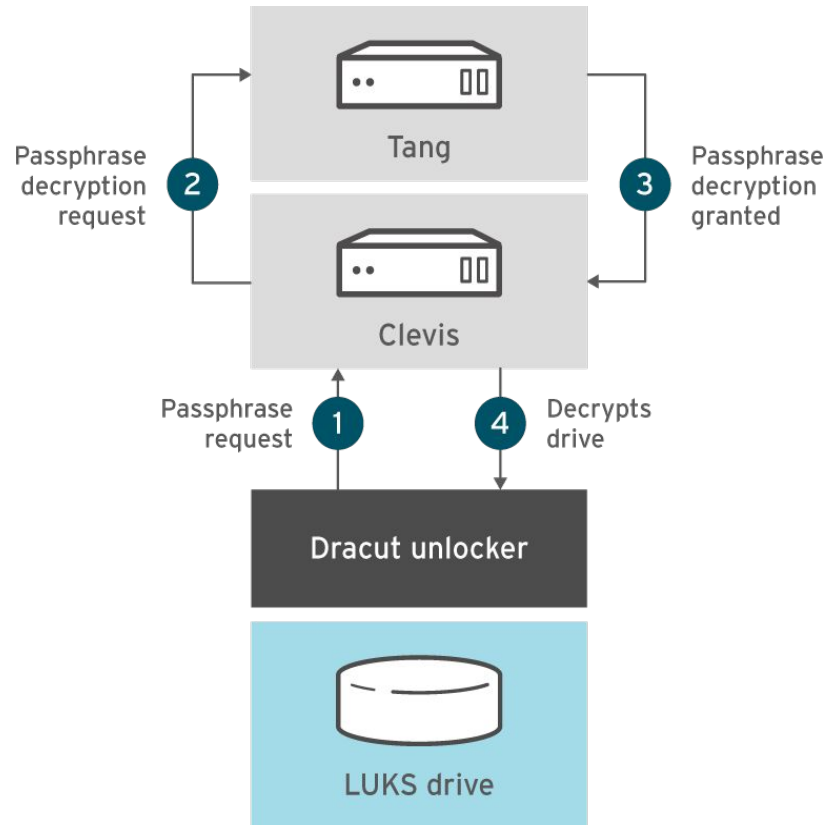
for unlisted USB devices

## UPDATE ACCESS

via command-line interface (CLI)



# NETWORK BOUND DISK ENCRYPTION



## NETWORK BOUND DISK ENCRYPTION

enables encryption and decryption of disks only on a trusted network, making data unusable if removed from the network.

## AUTOMATED DECRYPTION

using a client framework (CLEVIS) and modular key framework including a network key service (TANG)

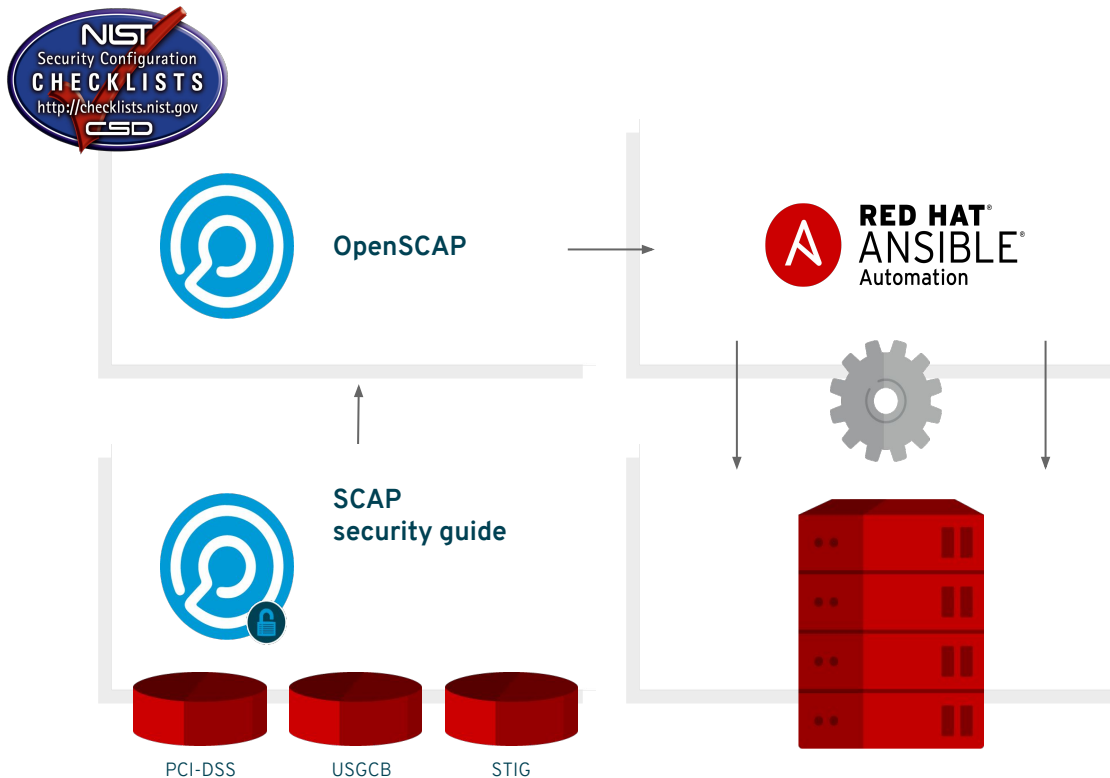
## DRACUT UNLOCKER

allows for decryption during early boot sequence

## SYSTEMD UNLOCKER

allows for decryption during system startup process

# OPENS CAP INTEGRATION WITH RED HAT ANSIBLE® AUTOMATION



## DEFINE AND TAILOR

security policies via profiles

## SCAN AND APPLY

security policies via Ansible Automation or bash

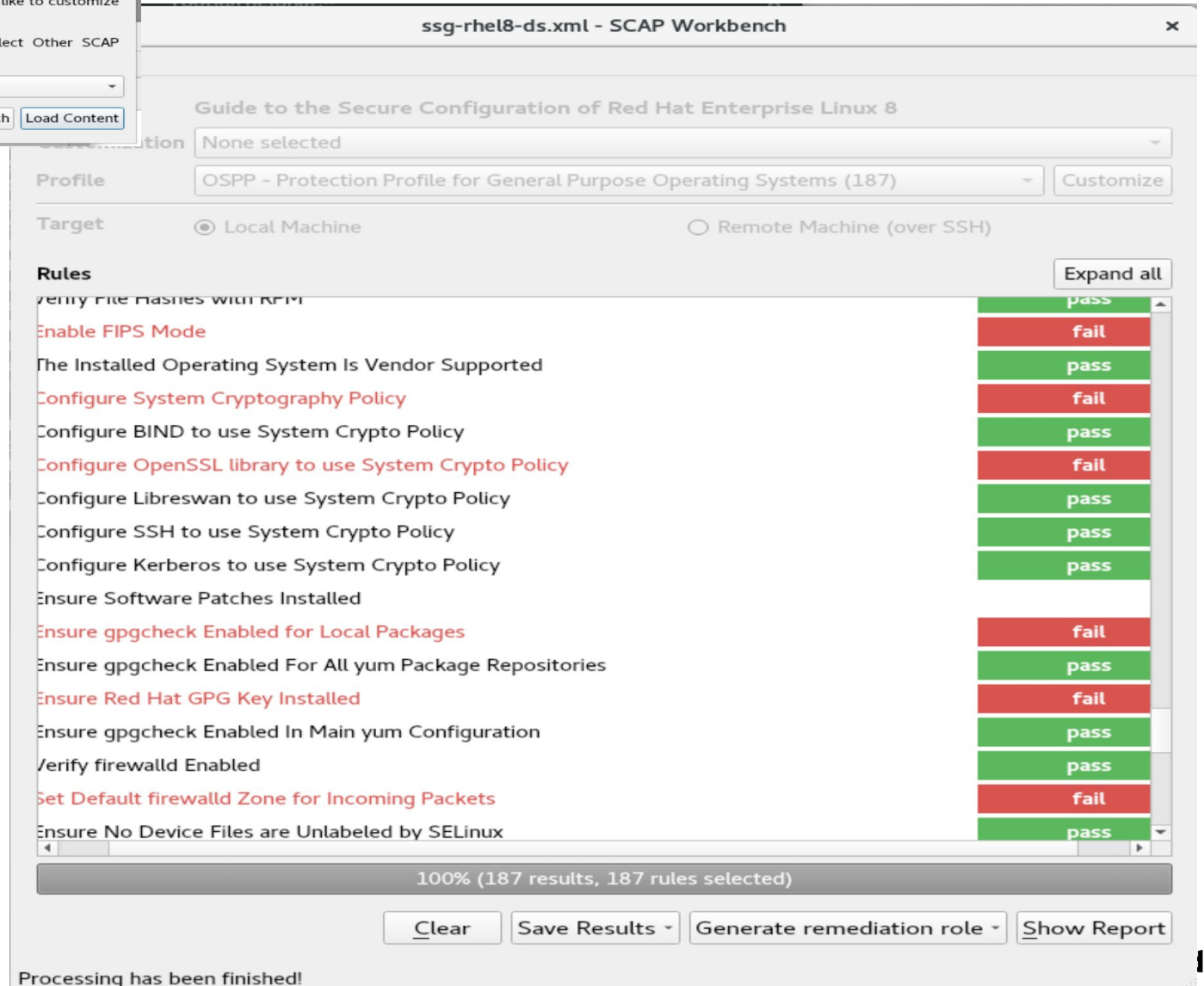
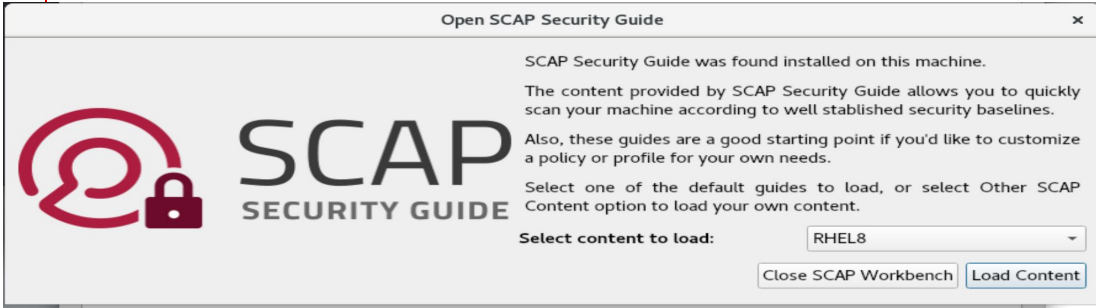
## ASSERT SECURITY POLICY

at build with Ansible Automation or Anaconda

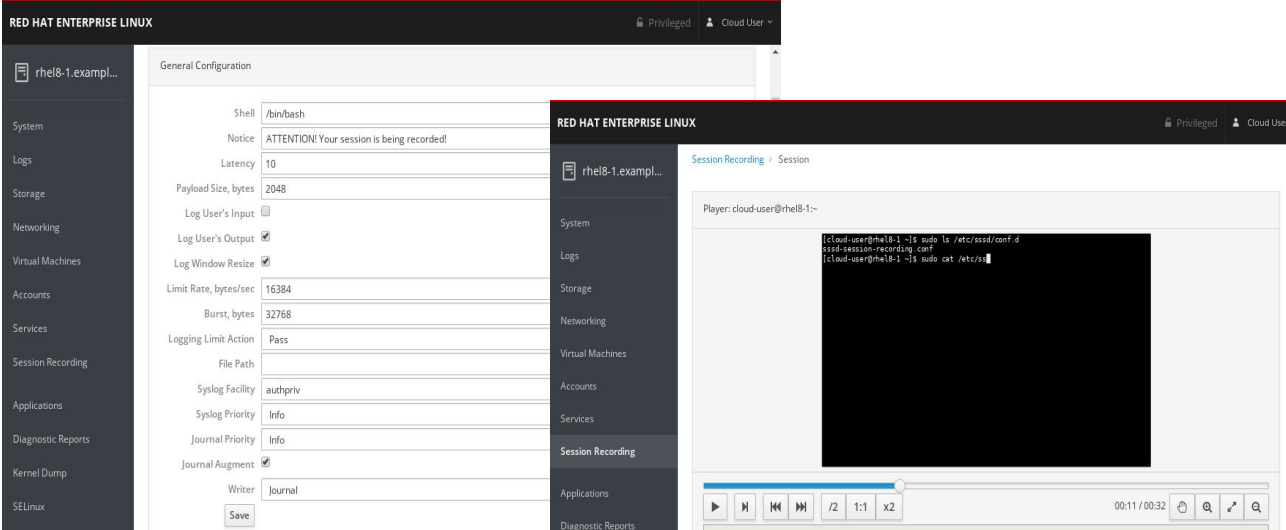
## SHIPPED NATIONAL CHECKLIST PROFILES:

DISA STIG | PCI - DSS | NIST USGCB





# Recording user terminal sessions



The image displays two screenshots from the Red Hat Enterprise Linux 8 web console. The left screenshot shows the 'General Configuration' page for a system named 'rhel8-1.example.com'. The 'Session Recording' section is highlighted in the left sidebar. The right screenshot shows the 'Session Recording' page, which includes a 'Player' view of a recorded session and a 'Recording' list at the bottom. The terminal window in the player shows a user logging in and running commands to enable session recording.

```
$ ssh cloud-user@rhel8-1.example.com
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Tue Apr  2 13:10:47 2019 from 192.168.122.1

ATTENTION! Your session is being recorded!

[cloud-user@rhel8-1 ~]$ sudo ls /etc/sss/conf.d
sssd-session-recording.conf
[cloud-user@rhel8-1 ~]$ sudo cat /etc/sss/conf.d/sssd-session-recording.conf
[session_recording]
scope=some
users=cloud-user
groups=
[cloud-user@rhel8-1 ~]$ exit
logout

Connection to rhel8-1.example.com closed.
$
```

**Recording**

ID	Hostname	Boot ID	Session ID	PID	Start	End
74e3069799604c2792af9705c363667-4ccdb523	rhel8-1.example.com	74e3069799604c2792af9705c363667	4	19661	2019-04-02 11:51:17	2019-04-02 11:51:40

## 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

# Upcoming Trainings for Partners

# RHEL 8 Enablements

## GLS-Training

RH 354 & RH 254, free for Advanced & Premier Partners !

## RHPDS

RHEL8 Workshop, after GA

## EMEA PE Offerings

This Training ;-)

## Learning Path

Future RH 200 & RH 300

## RHEL 8 Main Page

## RHEL 8 Deployment Guide

# Partner Conference 2019



25. - 27 June 2019  
PRAGUE

# 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.



[linkedin.com/company/red-hat](https://linkedin.com/company/red-hat)



[youtube.com/user/RedHatVideos](https://youtube.com/user/RedHatVideos)



[facebook.com/redhatinc](https://facebook.com/redhatinc)



[twitter.com/RedHat](https://twitter.com/RedHat)