# Why RHEL Image Mode is revolutionary, what it is and how it works.

## The significant benefits it brings to organizations and your ability to close deals.

**Brian Dumont**
Lt Principal Solution Architect



Red Hat

# What we'll discuss today

- RHEL Image Mode introduction
- RHEL Image Mode in action
- Q&A







# Introducing Image mode for RHEL





# Infrastructure & organizational complexity

is still a problem...




## Common challenges that involve the OS

- Different platforms require different tools, teams and expertise
- Testing and validation are time consuming
- Application support matrix
- No one budgets for maintenance and upgrades
- Negotiating between stakeholders
- Drift between images, instances, and runtime
- Immutable aspirations vs. mutable realities
- Image inventory, versioning, and pruning
- Let’s not forget security!

# Outcomes

What does image mode fix today?



## Less risk

Reduce the risk associated with updates with atomic transactions and rollbacks



## Better builds

Improve the composability and repeatability of standard builds through layering



## Move faster

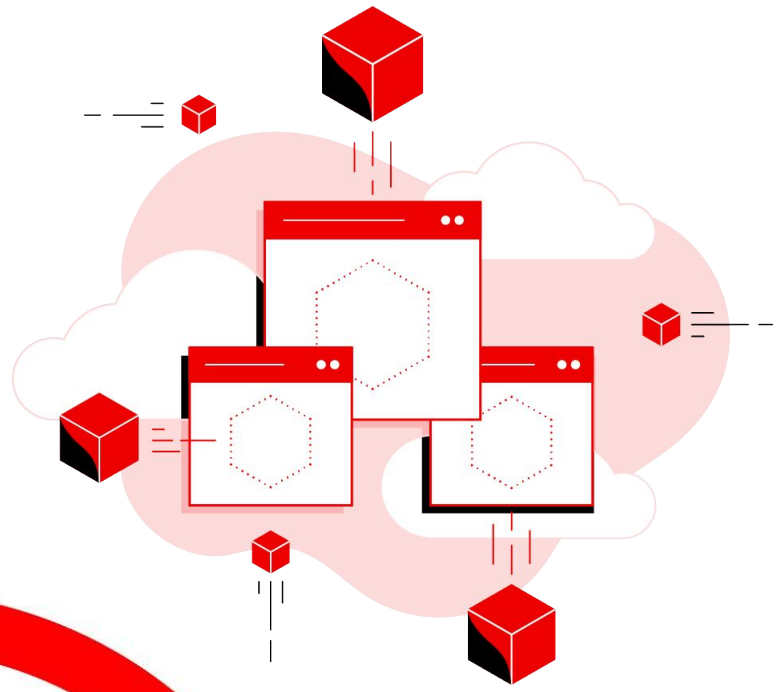
Increase the speed of experimentation



## Streamline process

Simplify end to end management with a single process for OS and applications

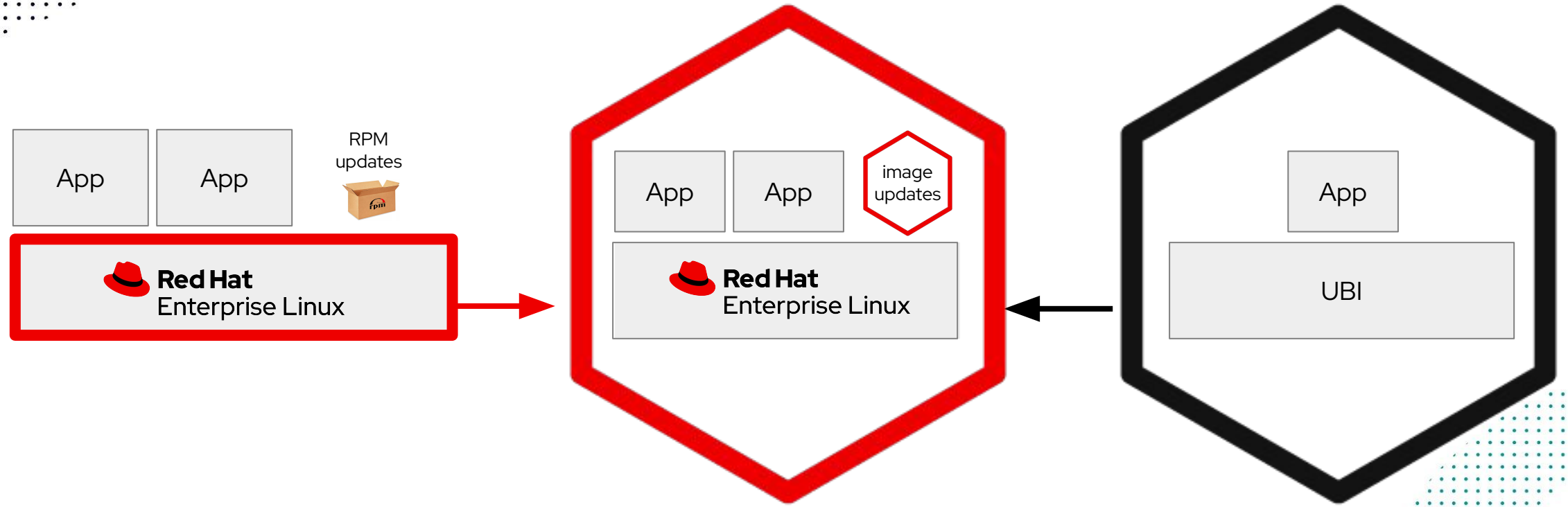
# Image mode creates opportunities to think differently



- **All RHEL users** will benefit from standardization, simplicity and portability across all of their environments that span the hybrid clouds
- **DevOps teams** can easily plug RHEL into their CI/CD & GitOps workflows, easing the friction that exists between the platform and the application.
- **Security teams** can apply container security tools, from scanning and validation to cryptography and attestation to the base elements of the operating system, making their jobs far less complex.
- **Solution providers** will love how easy it is to build and distribute their offerings on the trusted RHEL platform

# Introducing image mode for Red Hat Enterprise Linux

Combining the power of RHEL with the benefits of containers



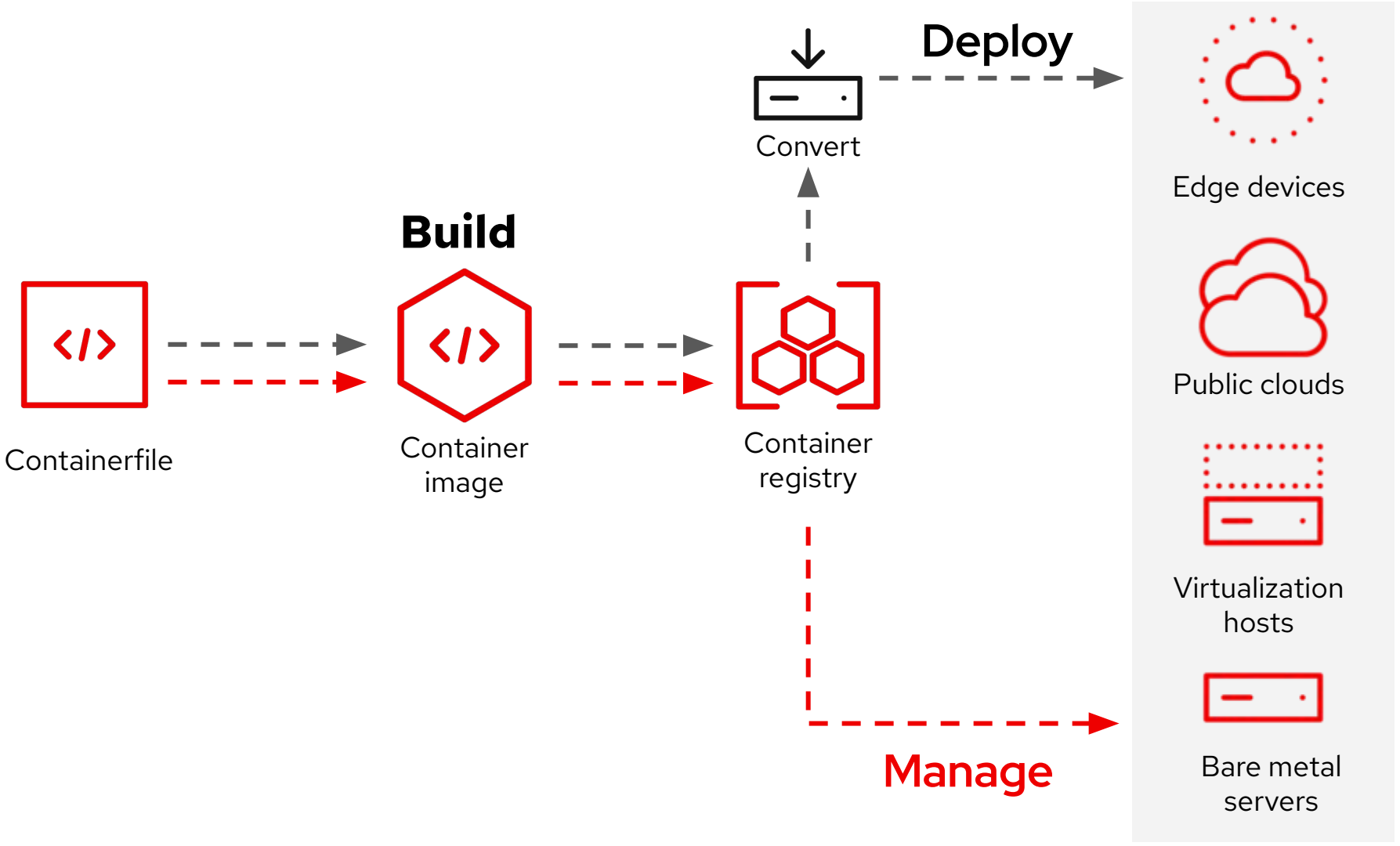

Package mode

Image mode

Application container

# Image mode for Red Hat Enterprise Linux

## Simple. Consistent. Anywhere.



# Image mode for RHEL

A container-native workflow for the life cycle of a system

```
FROM rhel9/rhel-bootc:latest

RUN dnf install -y [software]
[dependencies] && dnf clean all

ADD [application]
ADD [configuration files]

RUN [config scripts]
```

## Build

A *bootc* base image & container file is all that's needed to describe a system, applications, and dependencies. Use your existing container tools or pipelines to quickly create and test images.

## Deploy

Easily convert to a VM/cloud image or deploy on bare metal using RHEL's installer. The container image includes full hardware drivers, but not cloud agents by default.

## Manage

Designed for modern GitOps & CI/CD driven environments. Systems will auto-update from the container registry by default. More advanced control and automation is available via custom rollouts (e.g. Ansible). Intelligence via Insights and on-prem content curation via Satellite are planned for the future.

# Image mode for RHEL

Encapsulate differences in a sequence of builds

```
# Derive Standard Operating Environment
FROM rhel9/rhel-bootc:latest

RUN dnf install -y [system agents]
[dependencies] && dnf clean all

COPY [unpackaged application]
COPY [configuration files]

RUN [config scripts]
```

```
# Derive database server from SOE
FROM corp-repo/corp-soe:latest

RUN dnf install -y [database]
[dependencies] && dnf clean all

COPY [configuration files]

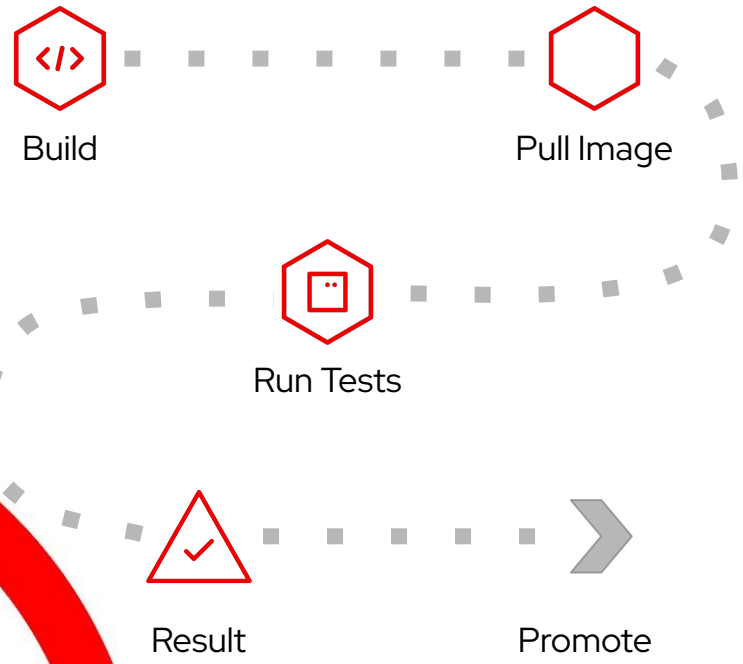
RUN [config scripts]
```





# Validating OS updates has never been easier

## CI pipelines used for apps now work with the OS




### Test/validate as a container

Bootc images can run as bare metal, VMs, **and containers**. This enables faster and lighter weight testing/validation of each build's userspace.

### Easy pipeline integration

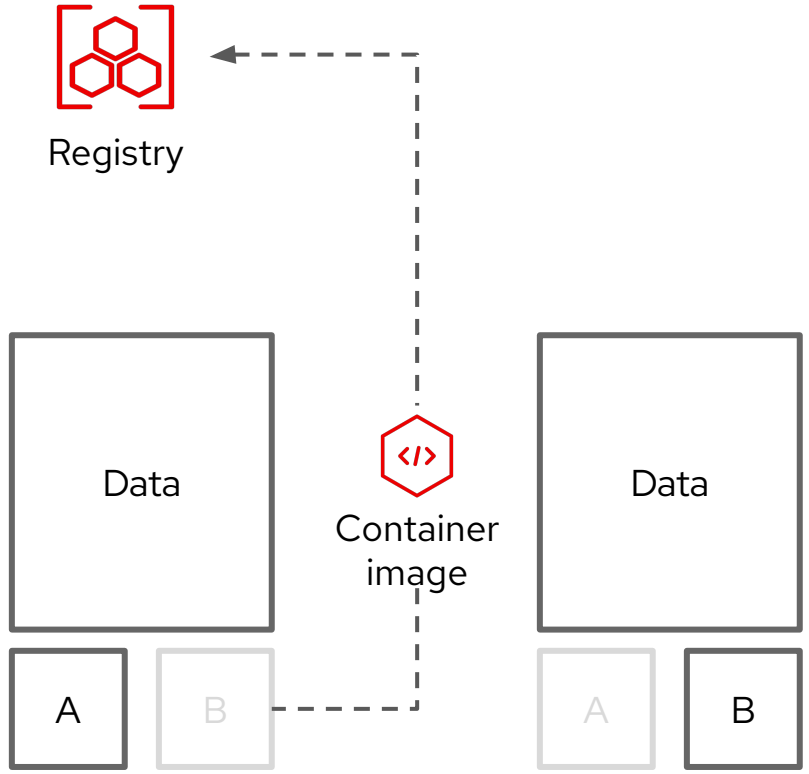
Containers have broad support across Github, Gitlab, Gitea, Circle CI, Jenkins, etc for the common container related tasks and testing. Use any system you like..

### Simple promotion through registry tagging

Tags are a powerful tool to identify dev → test → prod promotions.

# Bootc: Image-based updates perfected

Immutable by default - secure by design




### Transactional updates (A → B model)

Bootc uses composefs and ostree to convert the container image into the root filesystem on the host.

### Roll forward or backwards

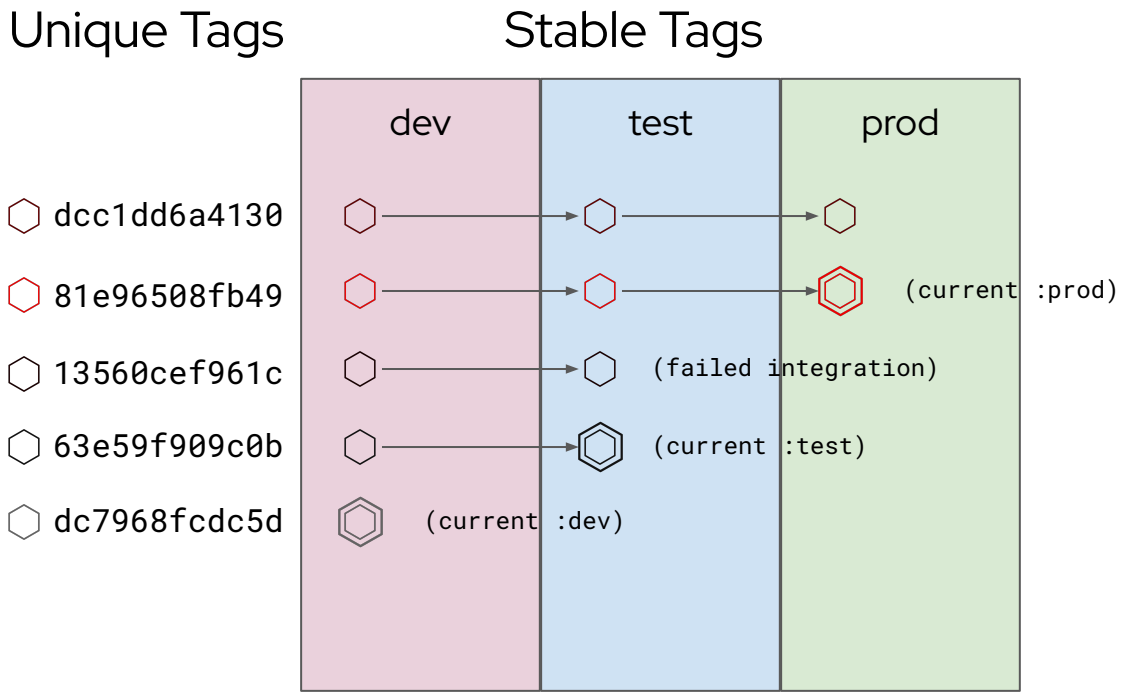
Updates are staged in the background and applied when the system reboots. The transactional model enables rollbacks for additional assurance

### Upgrades have never been easier

While there are some limits, bootc enables moving between minor releases of RHEL (9.4 → 9.5), as well as major releases (9.4 → 10.0)

# OS Updates via Container Registries

Tagging is powerful to version and promote updates




### Tags offer simple versioning and visibility

Tags are simple to automate and use for promotions. Bootc will default to updating from a `repository:tag`.

### Control updates via tagging

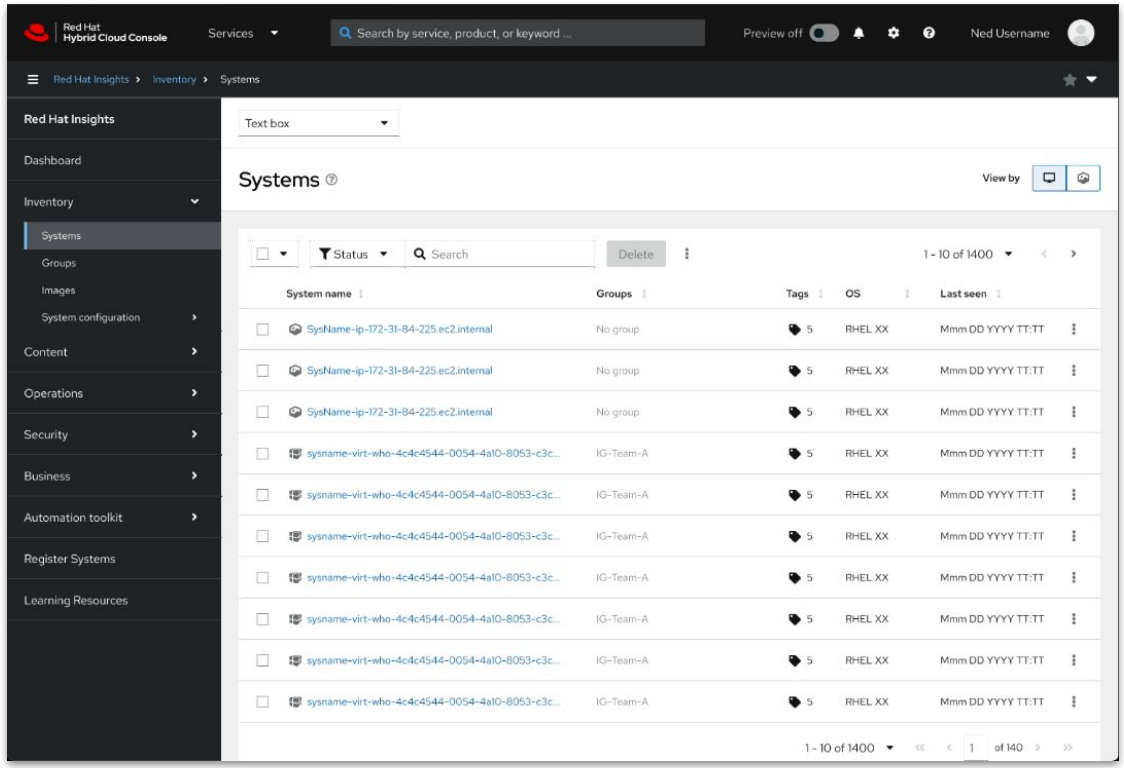
Combine tagging with the optional automatic updates to control fleets of systems via registry tags.

### Standardized & scaleable infra

Container registries scale very well and any standard registry can be used.

# Management with Red Hat Insights

Visibility and reporting made simple




## RHEL is RHEL

Image mode systems appear in inventory like package based systems.

## Registration is simple

Activation keys can be baked into images via Containerfile, allowing auto registration to Insights at boot time.

## Insights has ... Insights

Image mode systems can be scanned for security and operational recommendations.

## Updates on your terms

Image mode systems can be updated to new versions of images or remediated based on image-specific recommendations.

# Use Cases

Where does image mode fit today?



## AI/ML Stacks

Perfectly version app dependencies from kernel, GPU & accelerator drivers, frameworks, runtimes, etc



## 1:1 App/Host

Manage the OS AND app as a single unit



## Edge appliances

Easily manage a fleet of systems with registries and auto-updates



## Standalone container hosts

Use common toolchains and pipelines to build containerized applications and the hosting OS

# Image mode Roadmap

| | Now | Near | Future |
|---|---|---|---|
| **Build** | · rhel-bootc offers a small footprint base for image mode. Available as a technology preview<br>· FIPS mode support<br>· Base images for all supported RHEL architectures. | · Full support for bootc & rhel-bootc<br>· Custom base image creation<br>· Support of security compliance hardening<br>· zstd & zstd:chunked image compression | · Improved developer tooling for OS configurations & image creation<br>· **Composefs: Image signing & sealing** |
| **Deploy** | • Support for custom Kickstart (bare metal) and native cloud/virt images via Image Builder<br>• Bootc image builder improved partitioning options<br>• **Logically bound application containers to bootc image versions** | · Azure & Hyper-V image creation<br>· Composefs - optional backing store for podman/bootc<br>· zstd:chunked support brings foundational technology for client-side payload optimizations. | · bootc & Podman unified storage<br>· Composefs: Image integrity with Fs-verity<br>· Podman fsck - image store repair.<br>· Bootc-image-builder support for installer media with logically bound images<br>· **Factory reset / post-provisioning reset** |
| **Manage** | • Red Hat Insights support for inventory, CVE reporting, and applying image updates. | • Insights support for image analysis & CVE remediations<br>• Satellite provisioning, client support, and registry distribution for image mode & FlightCTL<br>• Transient & persistent package installs with DNF | • Update payload optimizations<br>• Soft reboot support<br>• Ability to separate OS configuration via config maps |



# RHEL Image Mode in action!



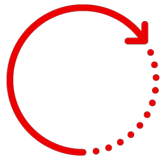




Create a Standard Operating Environment OS Image



Generate an ISO image and deploy on VM



Manage OS updates with bootc

# Use case - Create an ISO using SOE base image

Create, review and push the Containerfile to a GIT repository

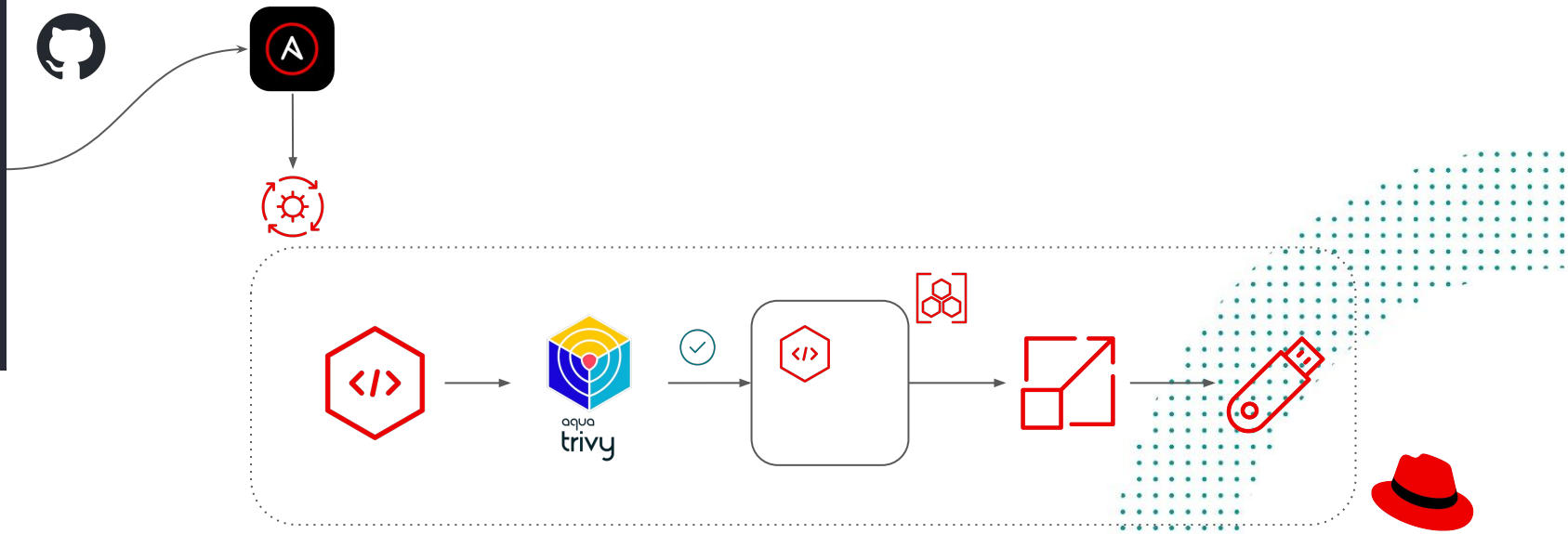
An image build is initiated on AAP via a workflow

The container image is built and scanned for vulnerabilities

Upon scan verification, the image is finally pushed in the company Image Registry and is ready

Convert the container image to an ISO image using bootc-image-builder and copy to host





# Use case - Update a system using bootc

Update and review the Containerfile for the bundled application

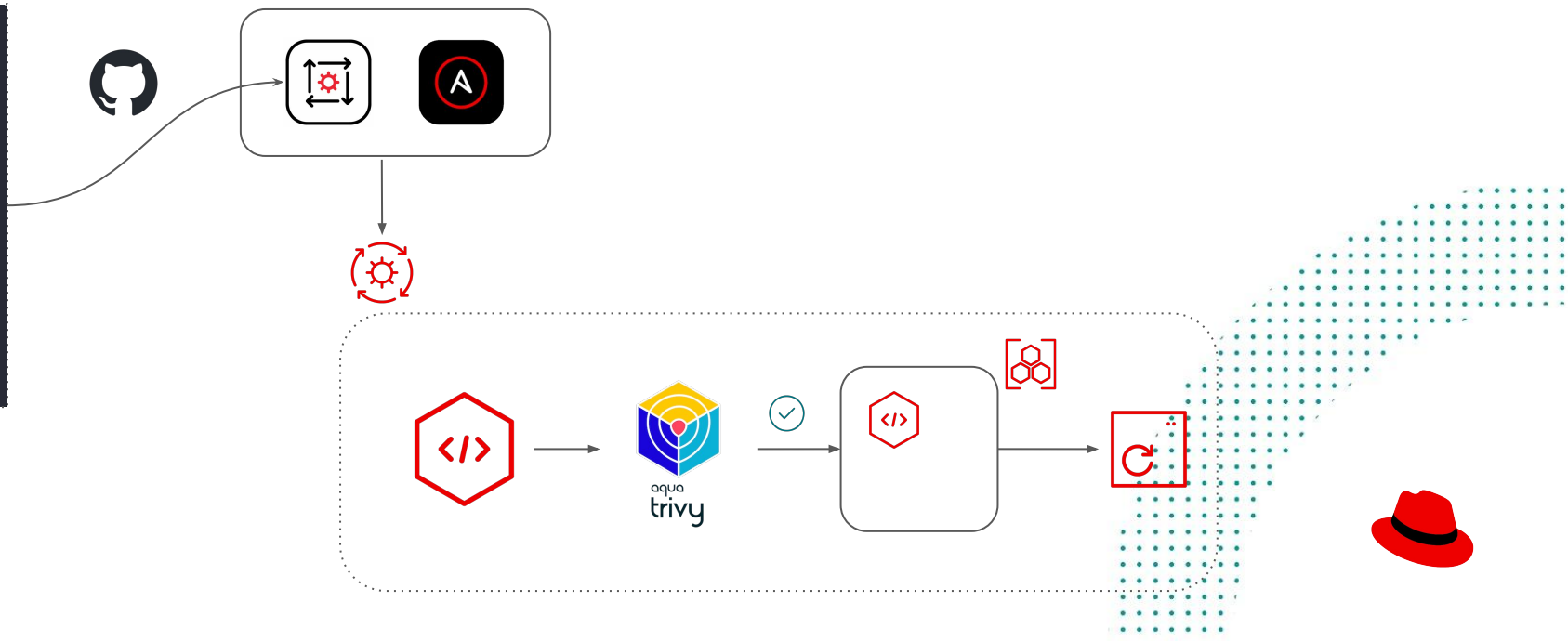
Push the modifications to git and create a tag on to trigger an event for EDA

A workflow is initiated and the container image is built and scanned for vulnerabilities

Upon scan verification, the image is finally pushed in the company Image Registry

**Run the update job on the target systems using bootc**







# Demo





# RHEL Image mode can make an impact

- RHEL Image mode can help and enable multiple personas to ease OS deployment and management
- Using familiar tools like containers enable developers and devops teams to
- A huge step ahead standardization of the OS across the organization
- Security is a first class citizen
- Automation-friendly, can be integrated with existing DevSecOps processes

# Demo content available on Github!

https://red.ht/rh1-image-mode
https://redhat-cop.github.io/redhat-image-mode-demo/

# Want to share more feedback with the user experience team?



red.ht/field-feedback



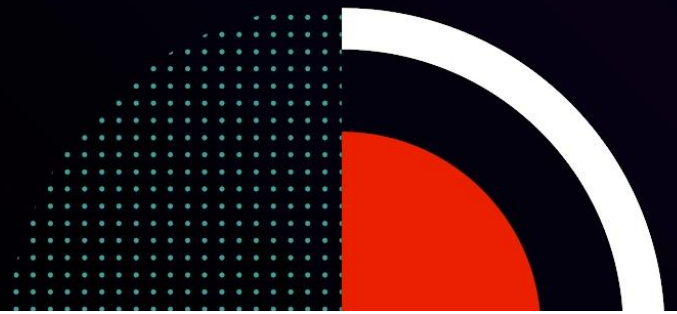
# Q&A





# 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

youtube.com/user/RedHatVideos

facebook.com/redhatinc

x.com/RedHat



Red Hat