

Automate RHEL bootc / Image Mode with Ansible on OpenShift Virtualization

Speaker



Wolfgang Marx

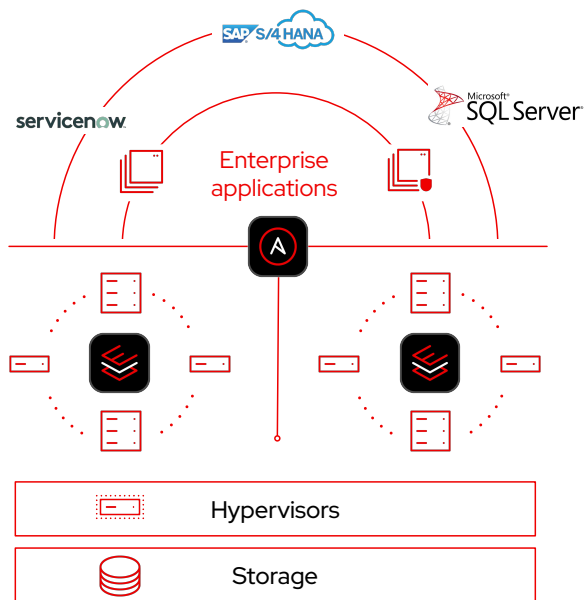
Senior Specialist Adoption Architect



**Future Ready
Platforms**



What is AAP2 for >>> Red Hat Enterprise Linux?



Ansible for Red Hat Enterprise Linux Automation is our content domain focused on this OS running in the data center, across clouds and at the edge. Ansible can also help automate related infrastructure including networks, storage, toolchain solutions, app deployment and more. The goal is to help teams simplifying tasks and processes.

Ansible infrastructure Certified Content Collections are designed to ensure configuration and management is consistent as well as reduce human error and eliminate manual processes.



Ansible Content Collections for Red Hat Enterprise Linux automation

Red Hat Enterprise Linux System Roles

Manages Red Hat Enterprise Linux system components.

Convert2RHEL

Runs the Convert2 RHEL tool for migrating from CentOS to Red Hat Enterprise Linux

Red Hat Lightspeed: Insights* and Insights EDA

Two content collections to configure and deploy the Insights client on hosts; and to enable automated remediation from Insights alerts

Leapp

Runs the Leapp tool for Red Hat Enterprise Linux upgrades, e.g. 8 to 9, 9 to 10 etc.

Red Hat Satellite

Installs Red Hat Satellite management as well as modules to interact with the Satellite API

Red Hat Single Sign On (RH-SSO) and Identity Management

Installs Red Hat Single Sign-On and keycloak; and deploys FreeIPA server with user/group topology

LVM

Creates and rolls back Logical Volume Manager snapshots

SAP

Deploys SAP on Red Hat Enterprise Linux and automates Day 2 SAP operations

Example: automated RHEL image build CI/CD

Manage the full lifecycle of Red Hat Enterprise Linux image mode

Create

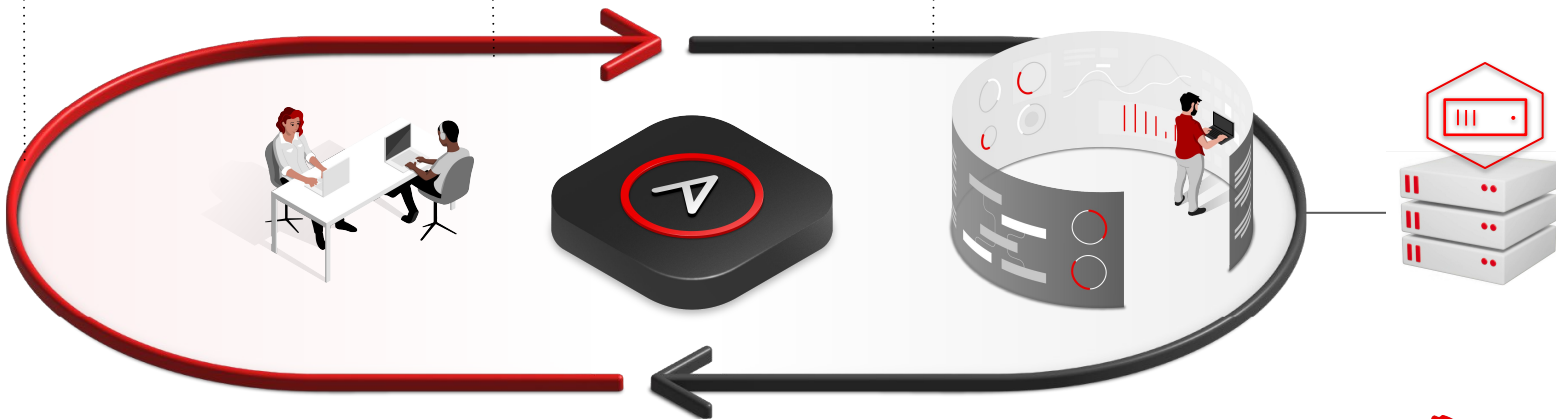
- Plan components
- Apply system roles
- Call Ansible as needed
- Push components to registry
- Optional full CI/CD management
- CMDB updates and management

Test

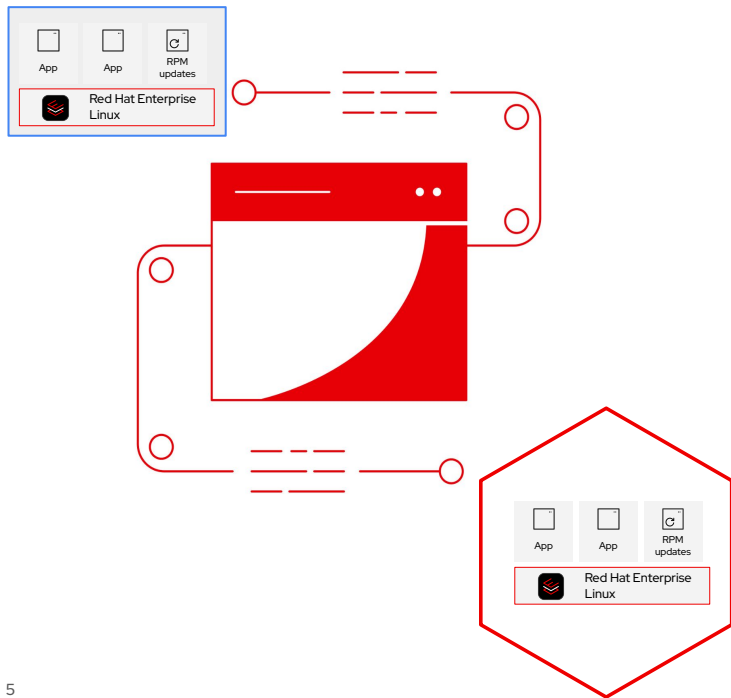
- Automate test infrastructure provisioning
- Automate unit, integration, hardening tests
- Validate libraries, apps, configs and version of all assets

Deploy / Upgrade

- Push completed image to registry for complex builds and orchestration
- Employ automated workflows
- Upgrade to latest images, including implementing rolling updates
- Complete reporting (compliance etc.)
- Use system roles as desired
- Orchestrate related infra / apps



Automated provisioning with RHEL Image Mode



Package mode option

- Call Kickstart from Ansible Playbook or Rulebook
- Use a Source of Truth for configuration (Satellite or other)
- Build desired automated workflow to manage related infrastructure and apps (network, certificates, monitoring, etc.).

Image mode option

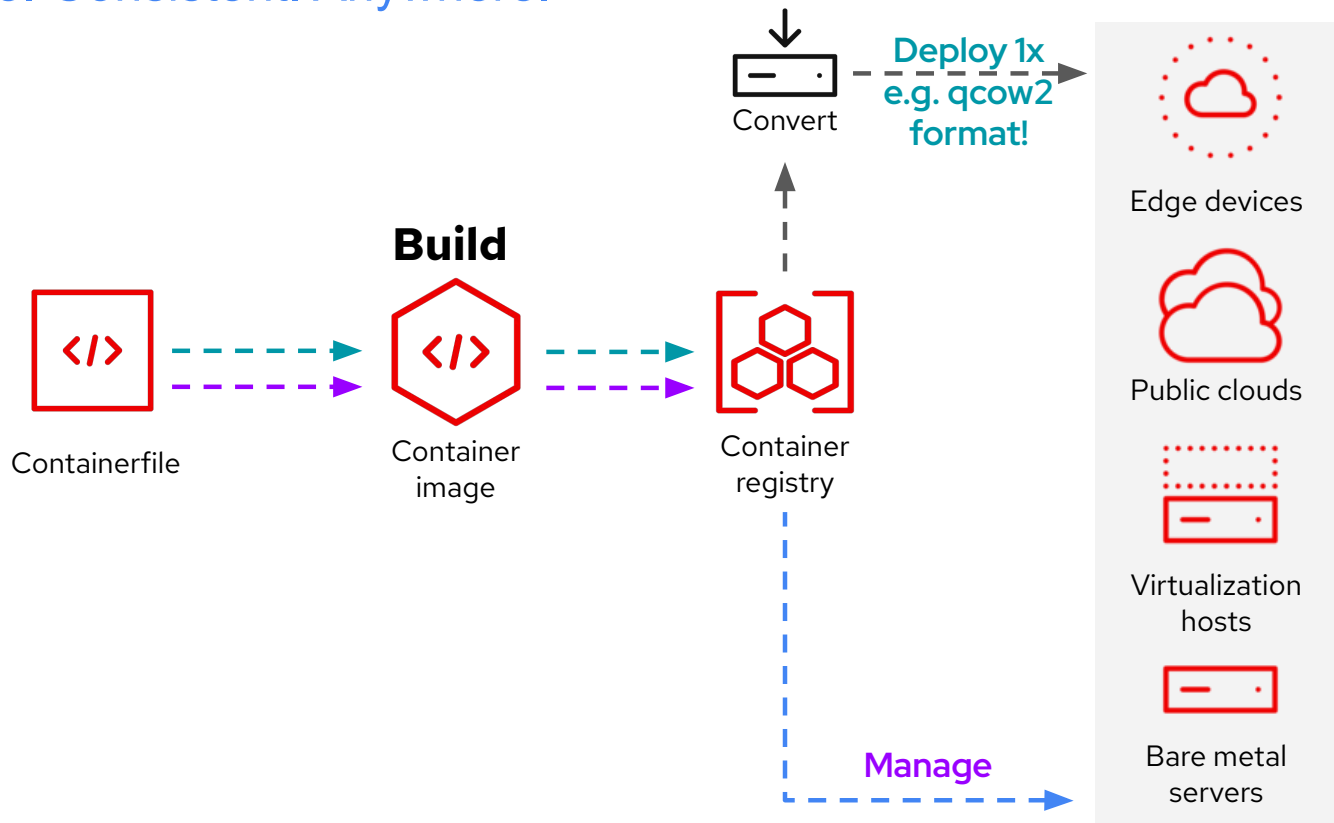
- Build containerized images in an automated CI/CD workflow
- Call images in workflow to create new instances and automate customizations in specific and allowed RHEL directories
- Use workflows for installing and managing related infrastructure and applications.
- Apply runtime configuration and changes.

One RHEL, two modes

	Package mode	Image mode
Smallest Unit	RPM package	OCI image
Updates	Package update (dnf update)	Image update (bootc upgrade)
Package installation	dnf install	Build OCI image with new package installed; then update (bootc upgrade)
System installation	Anaconda with %packages	Anaconda with ostreecontainer --url
Build raw, qcow2, AMI, GCP, VMware, Azure image	Image Builder	Bootc Image Builder

Image mode for Red Hat Enterprise Linux

Simple. Consistent. Anywhere.



registry.redhat.io/rhel10/rhel-bootc:10.2

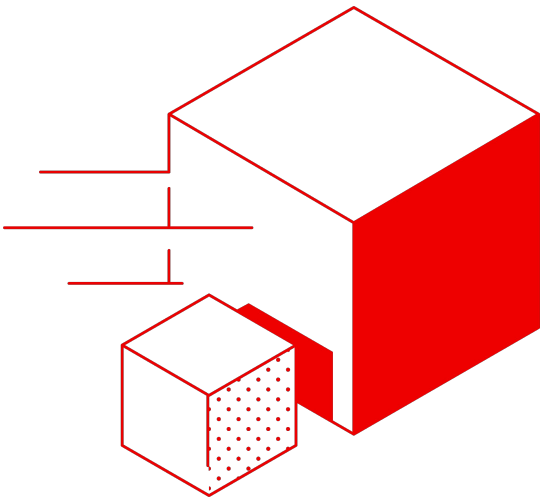
The RHEL bootc image is available here!

Image Specs:

- ~400 rpms
- ~750M compressed
- ~2.0G on disk

Primary contents:

- systemd, kernel, bootc
- rpm-ostree
- linux-firmware
- NetworkManager
- podman
- python
- Misc CLI tools: jq, sos

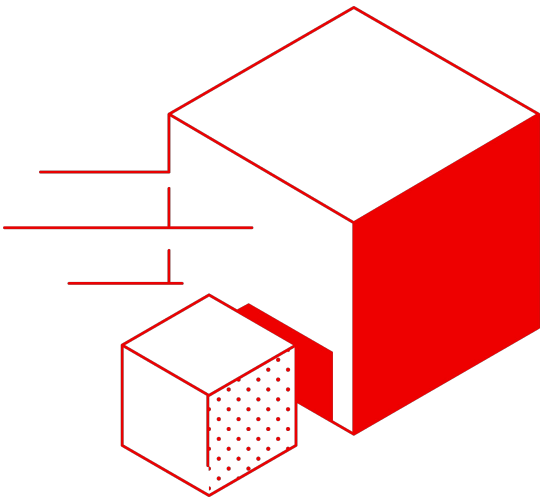


The Features / Differences!

Feature / Property	Classic RHEL (Package Mode)	RHEL Image Mode
<u>Update Process</u>	Step-by-step package updates via dnf update on the running system.	Atomic replacement of the entire OS image via bootc update.
<u>Immutability</u>	File system is writable; configuration drift over time is typical.	/usr /boots /lib is read-only. The system remains exactly as defined in the image.
<u>Rollback Behavior</u>	Difficult; requires VM snapshots or complex package rollbacks.	A/B Booting: If errors occur, the system simply boots the previous image on the next restart.
<u>Tooling</u>	Kickstart, Ansible, Puppet, DNF/YUM.	Containerfile/Dockerfile, Podman, Container Registries (e.g., Quay).
<u>Development Approach</u>	"Day 2" configuration directly on the server.	"Shift Left": Server configurations are already defined during the build process in CI/CD.

Typical Use Cases!

Where Do I need this??



What is suitable?

- Edge Infrastructure /Remote Locations
- AI/ML / GPU-Workloads
- Appliance Style
- Developer-to-production OS pipelines
- Cloud Images

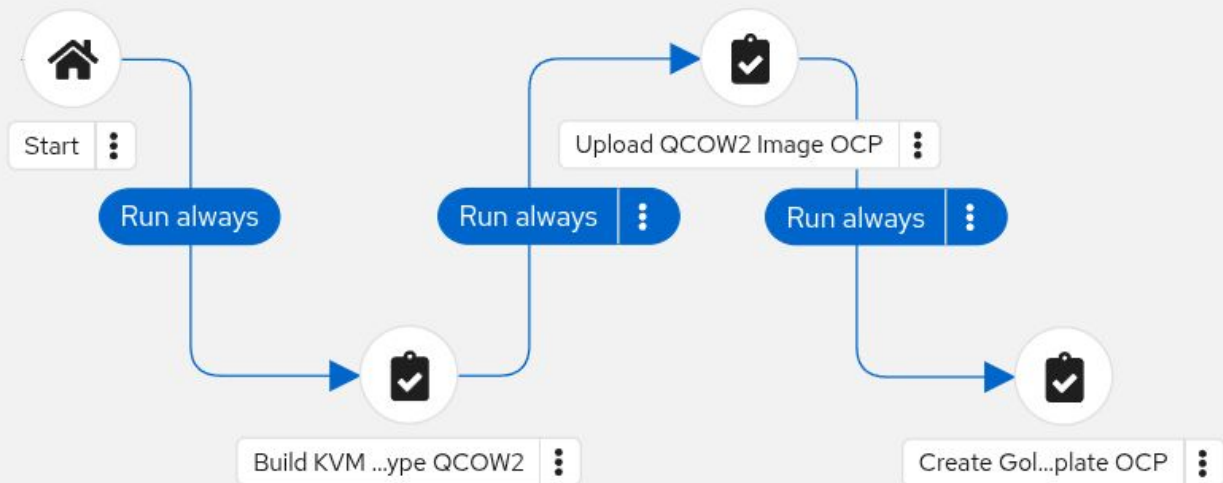
What is NOT suitable?

- maintained pet servers
- snowflake systems
- Servers with lots of ad-hoc package changes

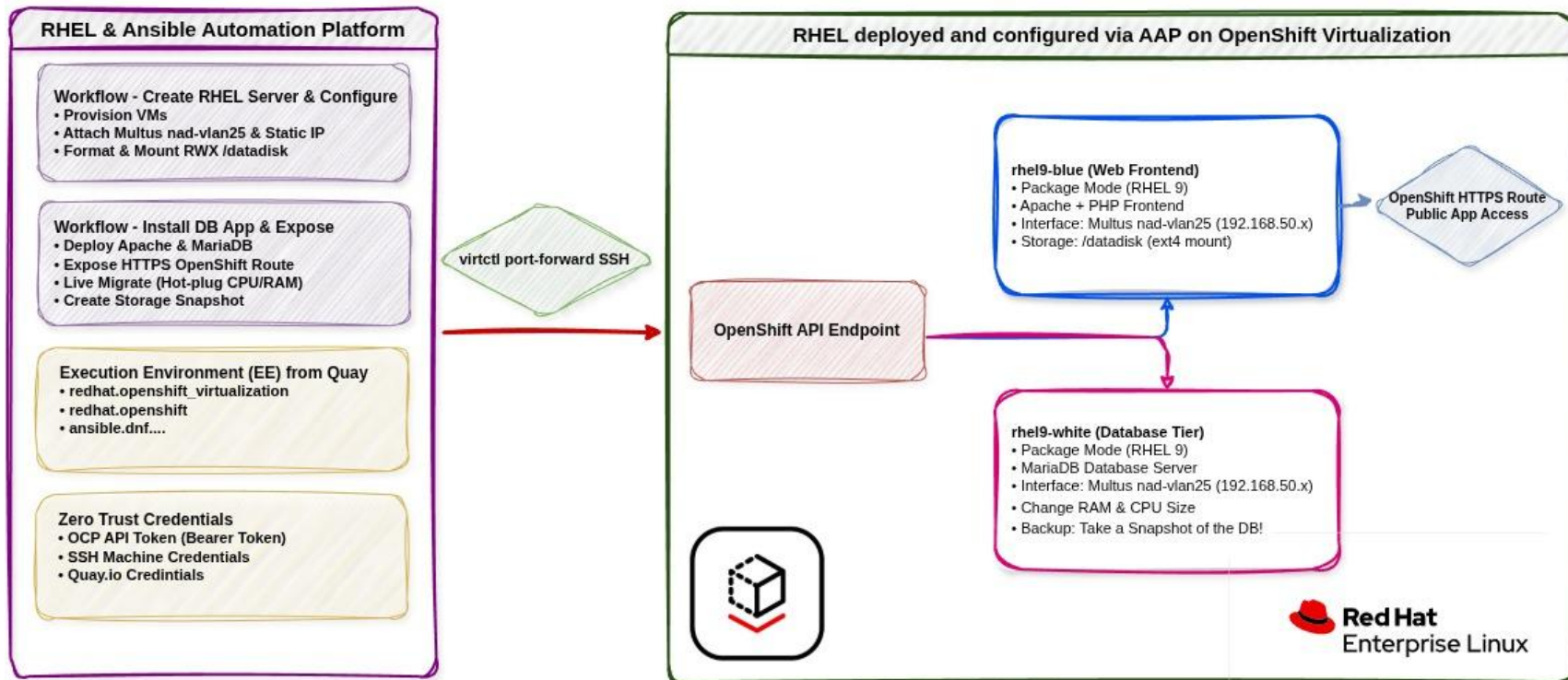
Automated workflows in action

Workflow Visualizer 020. Build bootc container - create QCOW2 Image & OCP Template

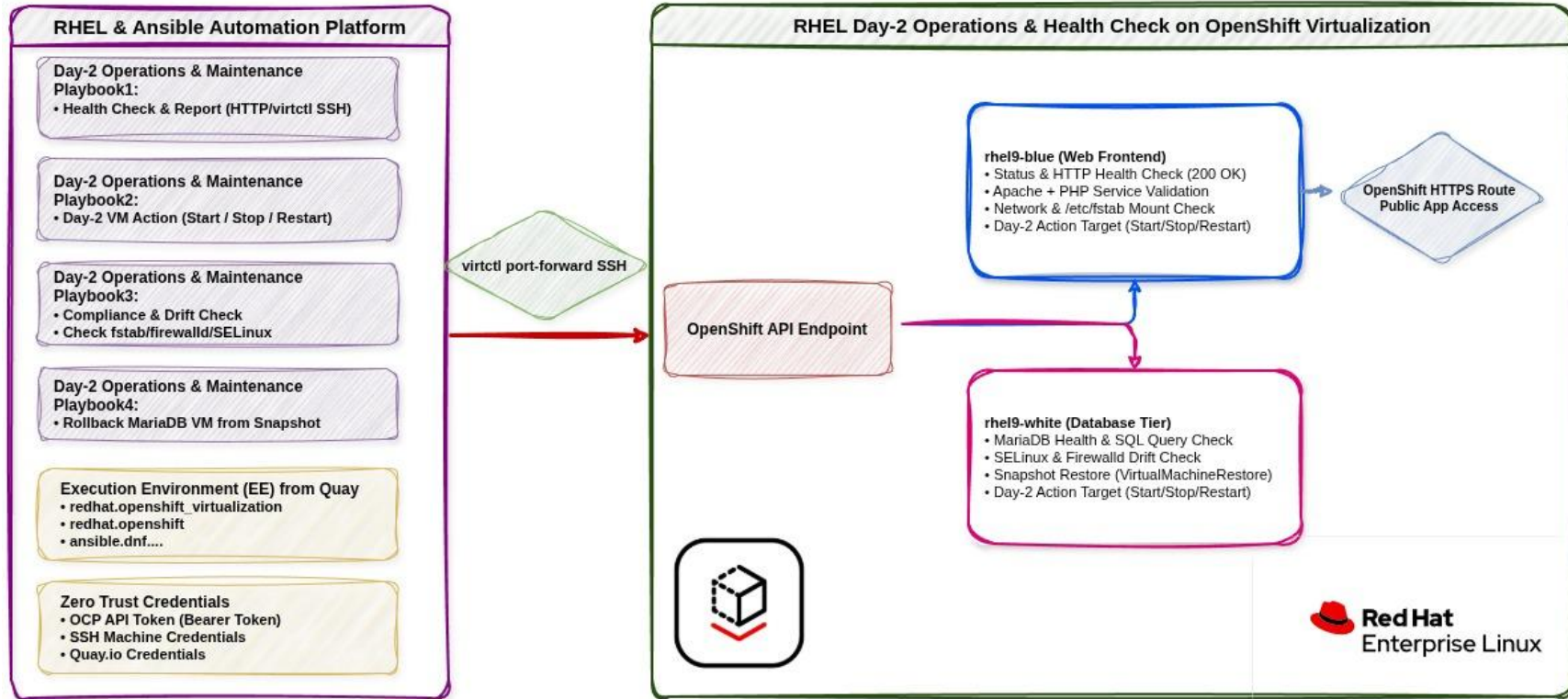
Save Add step Launch workflow ⋮ Total nodes 3



UseCase 1a: Deploy RHEL, update, configure, Subscribe & Install an End-to-End Web Application



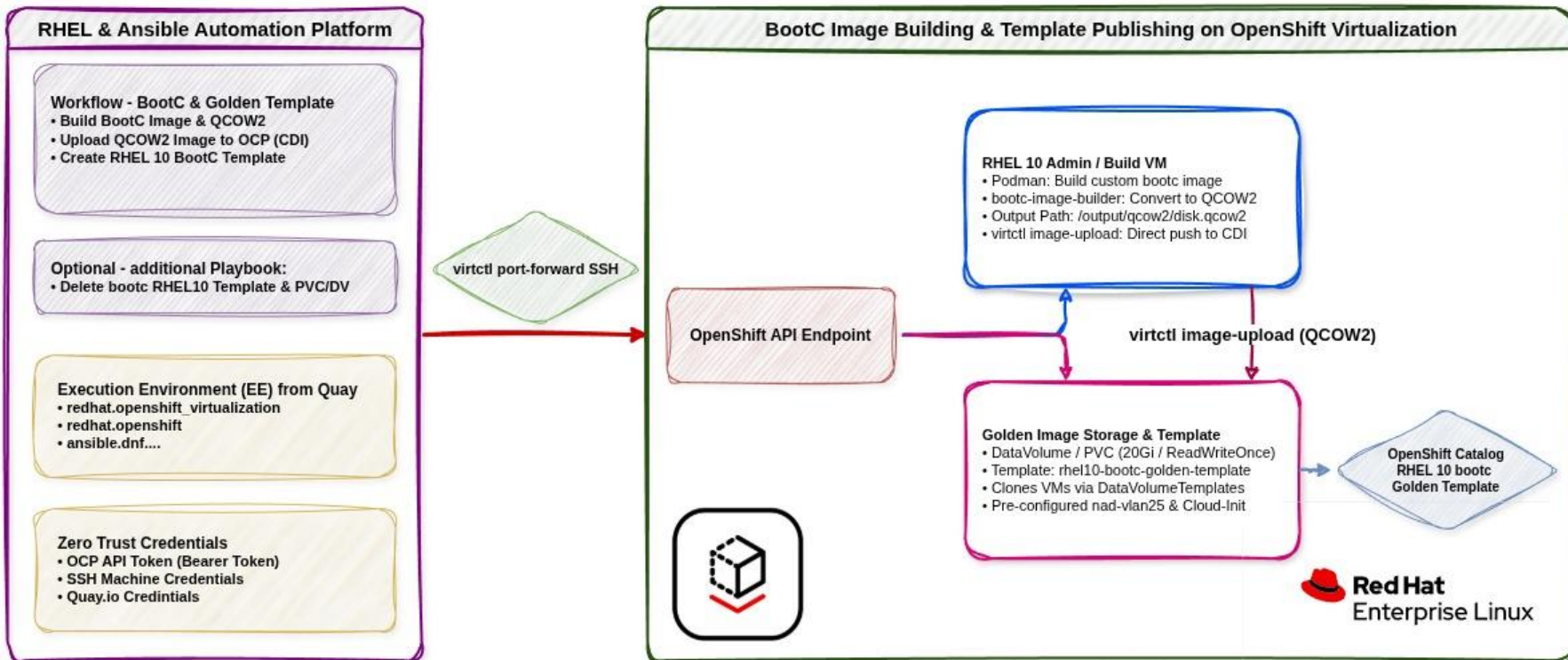
UseCase 1 b: VM Health Check, Day-2 Operations, Drift Check & Snapshot Rollback



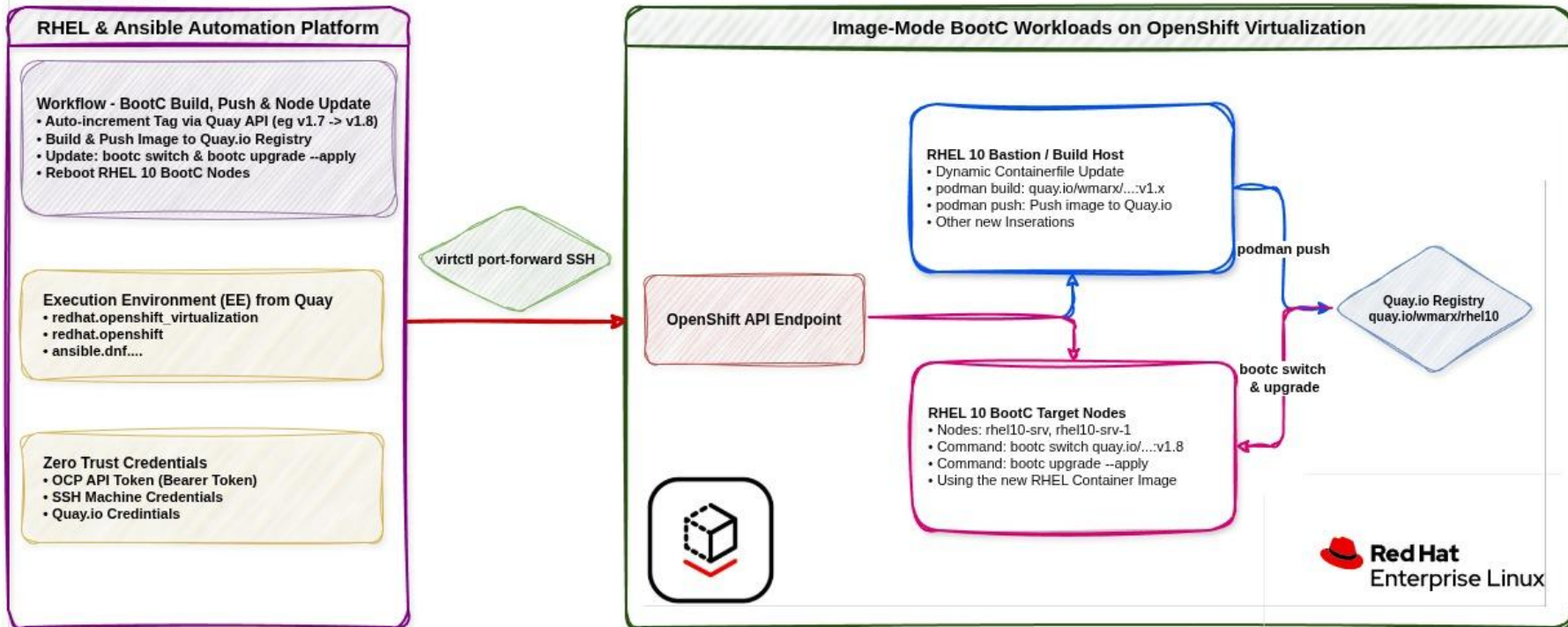
UseCase 2: Build bootc Container, Create QCOW2 Image & OCP Golden Template



Red Hat
Ansible Automation
Platform



UseCase 3: Update BootC Image, Push to Quay Registry & Atomic Node Upgrade



Special Use-Case: Policy Enforcement in AAP2.7

Speaker

Wolfgang Marx

Senior Specialist Adoption Architect



**Future Ready
Platforms**

