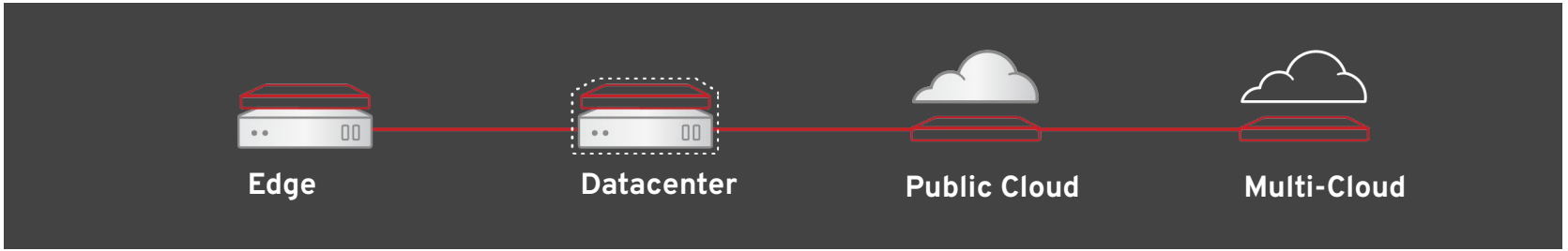
Red Hat Enterprise Linux CoreOS
Security Capabilities

Alfred Bach
Principal Solution Architect
Red Hat EMEA

Red Hat

What makes an effective hybrid cloud platform?

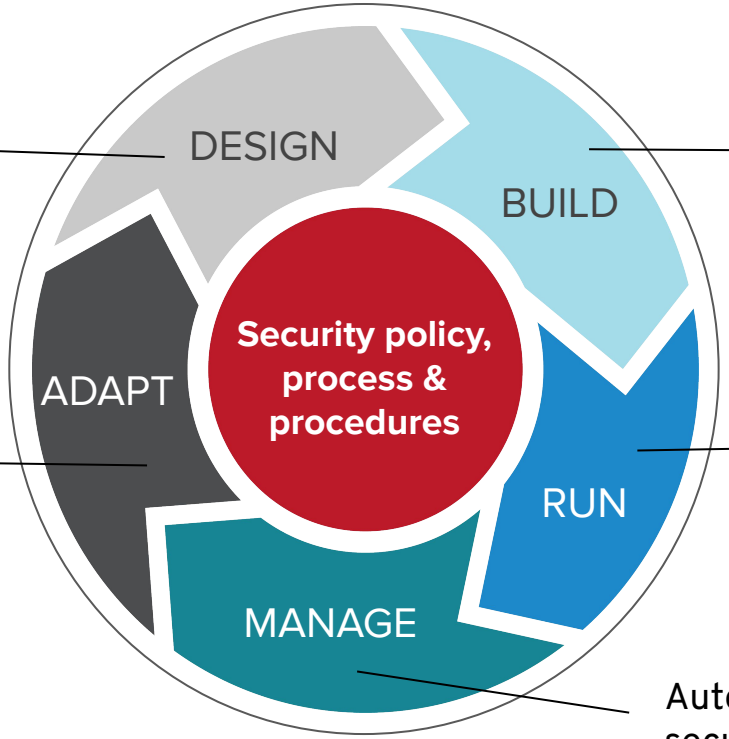
BROAD ECOSYSTEM	BROADEST APPLICATION SUPPORT	DEVELOPER EXPERIENCE & ON-DEMAND
AUTOMATED OPERATIONS	STANDARDS, PORTABILITY & INTEROPERABILITY	SECURITY & COMPLIANCE



Red Hat

Security must be continuous

And integrated throughout the stack and the IT lifecycle



Identify security requirements & governance models

Built-in from the start; not bolted-on

Revise, update, remediate as the landscape changes

Deploy to trusted platforms with enhanced security capabilities

Automate systems for security & compliance

Red Hat

OpenShift 4

Considerations for Securing Containers and Kubernetes

NIST 800-190

"Use container-specific host OSs instead of general-purpose ones to reduce attack surfaces."

NIST Special Publication 800-190 Application Container Security Guide

CNCF Cloud Native Security Whitepaper https://github.com/cncf/sig-security/blob/master/security-whitepaper/CNCF_cloud-native-security-whitepaper-Nov2020.pdf

CNCF Kube Security Audit

"...the underlying hosts, components, and environment of a Kubernetes cluster must be configured and managed. This management has a direct impact on the capabilities of the cluster..."

"...configuration and deployment of Kubernetes [is] non-trivial, with certain components hav[ing] confusing default settings, missing operational controls, and implicitly defined security controls."

Kubernetes Security Whitepaper, Trail of Bits, May 31, 2019

Cloud Workload Protection

"The best way to secure these rapidly changing and short-lived workloads is to start their protection proactively in the development phase ..."

"Replace antivirus (AV)-centric strategies with a "zero-trust execution"/default deny/application control approach to workload protection where possible...." [1]

Gartner: Market Guide for Cloud Workload Protection Platforms, ID G00356240, April 8, 2019

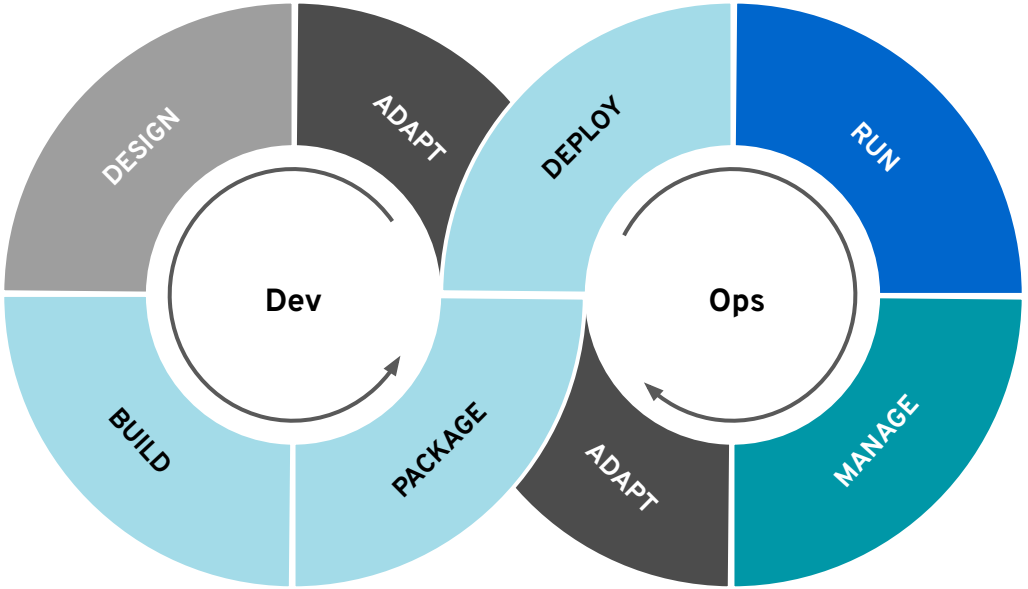
Red Hat

Continuous Security and DEVOPS

Control

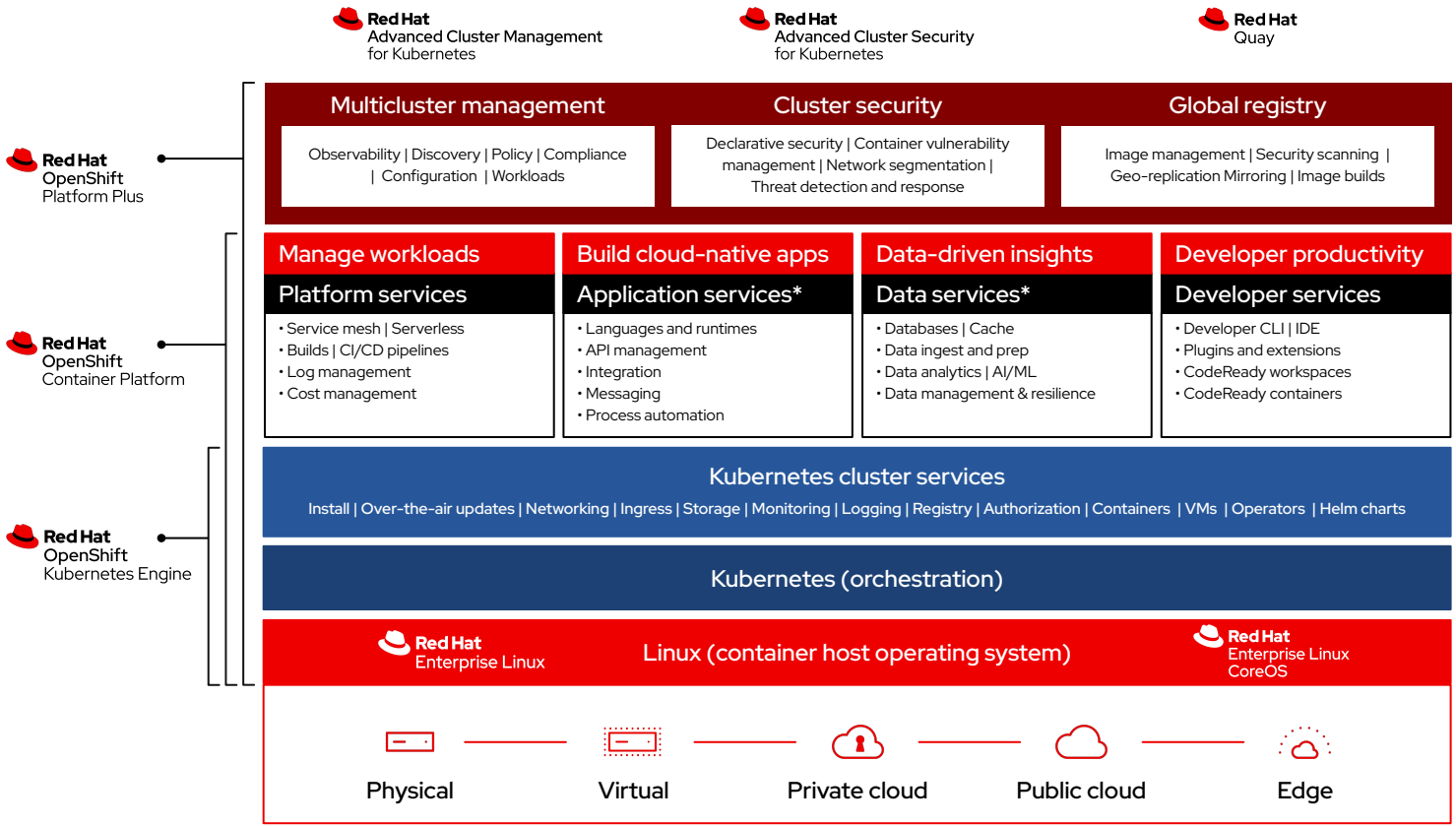
Protect

Detect & Respond



Red Hat

OpenShift Platform Plus

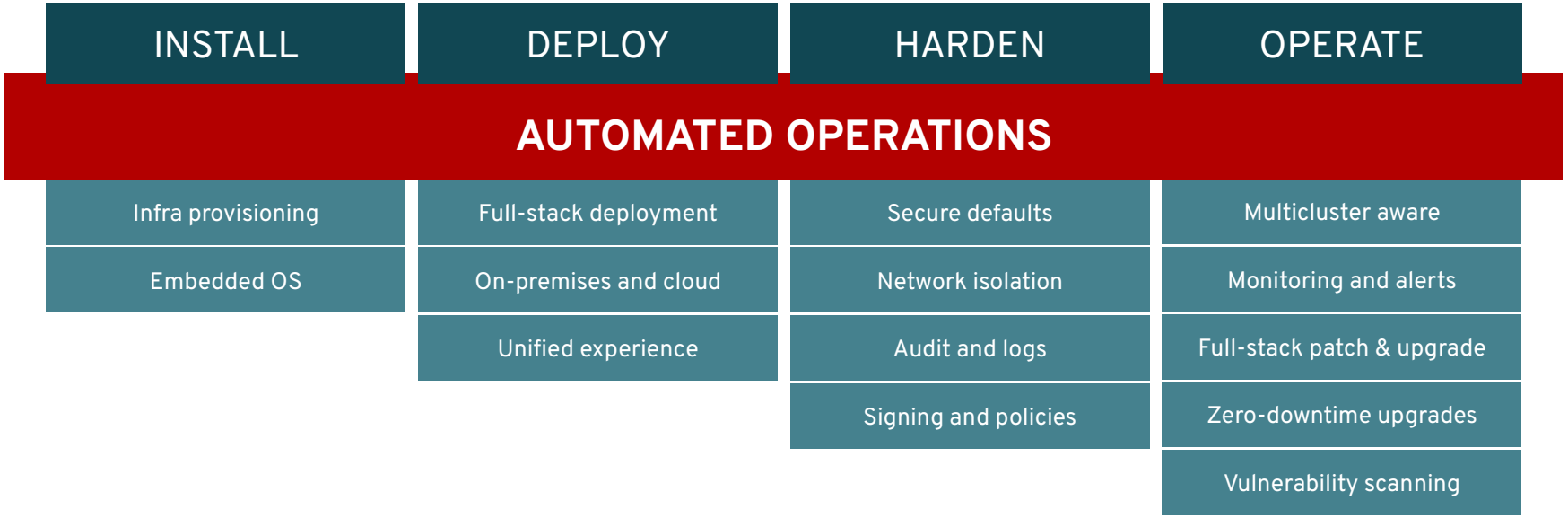


* Red Hat OpenShift® includes supported runtimes for popular languages/frameworks/databases. Additional capabilities listed are from the Red Hat Application and Data Services portfolio.

Red Hat

Automated Container Operations

FULLY AUTOMATED DAY-1 AND DAY-2 OPERATIONS



Red Hat

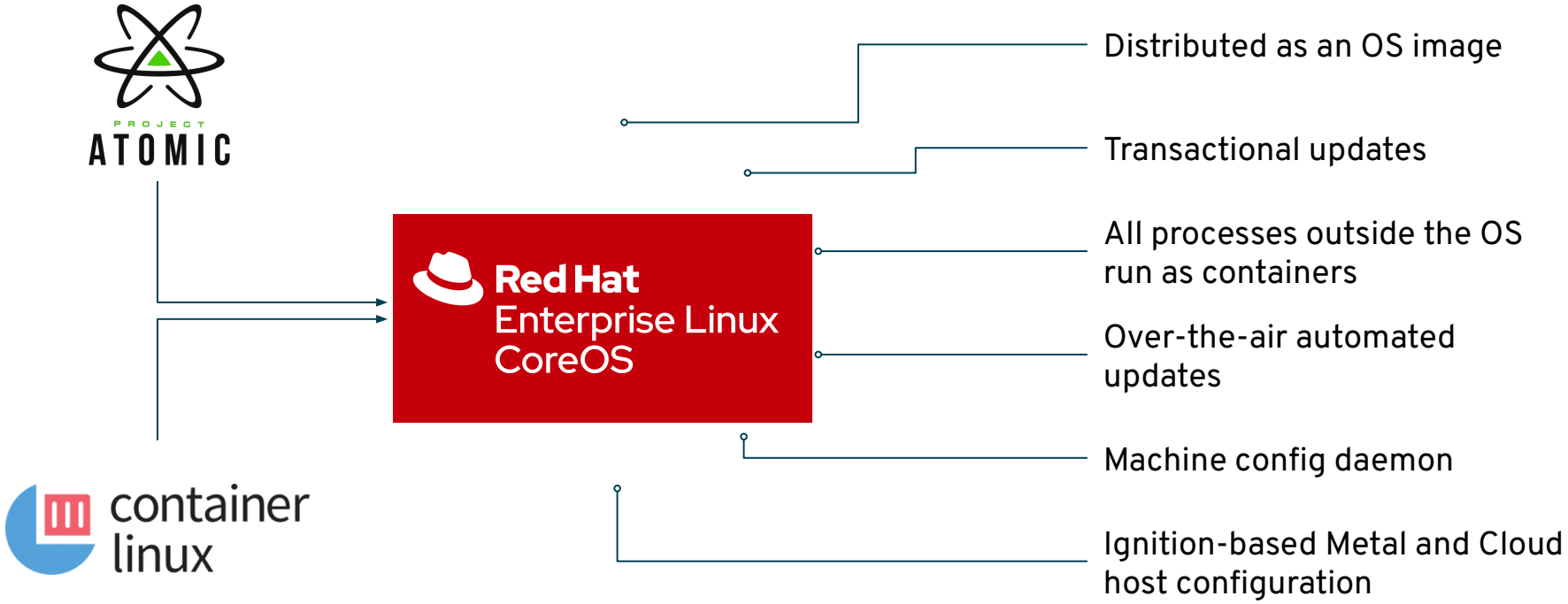
AGENDA

- About RHEL CoreOS
- Security of the code
- System Management
- Crypto roots
- Securing Workloads
- Data protection
- Audit and Compliance
- Security and Hardware

AGENDA

- **About RHEL CoreOS**
- Security of the code
- System Management
- Crypto roots
- Securing Workloads
- Data protection
- Audit and Compliance
- Security and Hardware

RHEL CoreOS is the best of both....



Red Hat

Immutable Operating System

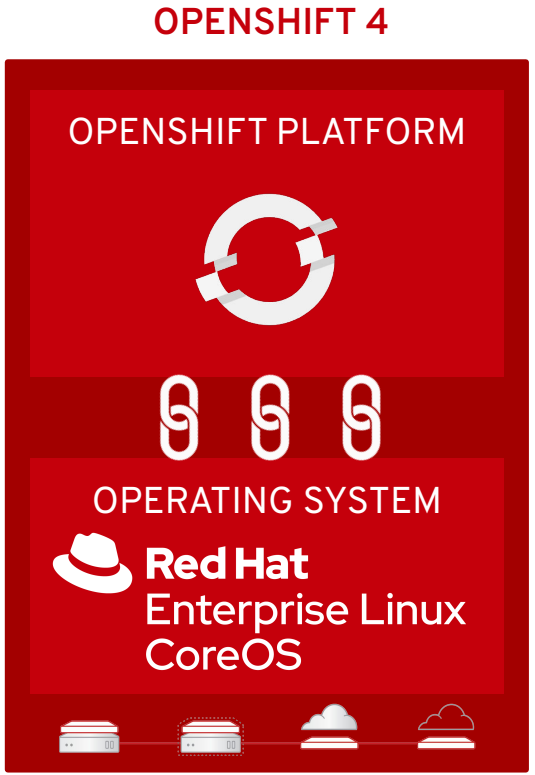
Red Hat Enterprise Linux CoreOS is versioned with OpenShift

RHEL CoreOS is tested and shipped in conjunction with the platform. Red Hat runs thousands of tests against these configurations.

Red Hat Enterprise Linux CoreOS is managed by the cluster

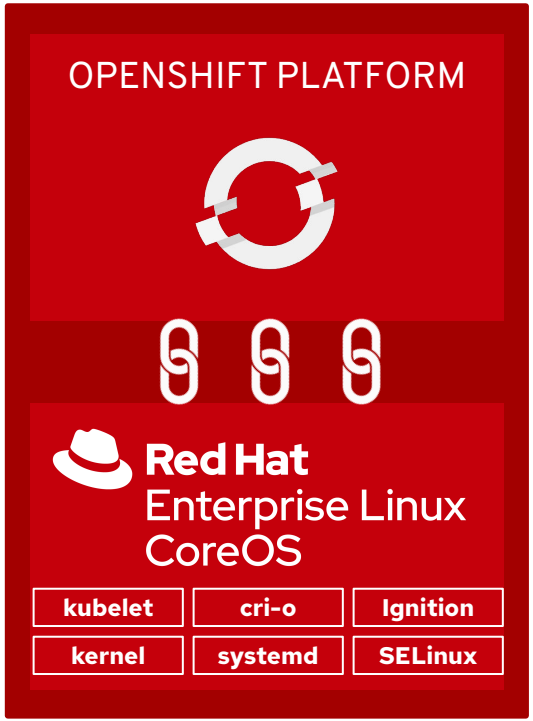
The Operating system is operated as part of the cluster, with the config for components managed by Machine Config Operator:

- CRI-O config
- Kubelet config
- Authorized registries
- SSH config



Red Hat

Value of controlled immutability



For Day 2 management, the cluster needs full control over the nodes.

Immutability = repeatability

Immutability = auditability

Immutability ≠ static clusters

Immutability ≠ no config changes

RED HAT ENTERPRISE LINUX

	RED HAT® ENTERPRISE LINUX®	RED HAT® ENTERPRISE LINUX CoreOS
	General Purpose OS	**Immutable container host**
BENEFITS	• 10+ year enterprise life cycle • Industry standard security • High performance on any infrastructure • Customizable and compatible with wide ecosystem of partner solutions	• Self-managing, over-the-air updates • Immutable and tightly integrated with OpenShift • Host isolation is enforced via Containers • Optimized performance on popular infrastructure
WHEN TO USE	When customization and integration with additional solutions is required	When cloud-native, hands-free operations are a top priority

Key characteristics of RHEL CoreOS

- **Transactional updates** - RHCOS is distributed as an image and each operating system update is versioned and distributed as containers. Major releases (and some z stream releases) provide new boot images. The OS always boots into a known-good version; this is similar in principle to how container images are managed and deployed.
- **Immutable management** - RHCOS is built to be managed in an immutable fashion by the Machine Config Operator and Kubernetes API. While certain parts of the OS are truly immutable, others are not. Immutable management enables us to spawn new nodes and ensure that the cluster is the single source of truth for provisioning configurations, OS versions, and run-time configuration. Apart from consistency, this also enables elastic clusters to spawn and destroy nodes.

Key characteristics of RHEL CoreOS (cont'd)

- **Applications need to run in containers** - Installing RPMs on RHCOS is not supported. The OS is built to run all processes outside the OS as a container. This allows us to guarantee successful upgrades and automation beyond what a traditional operating system can deliver.
- **rpm-ostree** - This is the technology used to assemble the operating system. RHEL RPMs are used to create the OS images, and versions can easily be queried using the rpm command.
  - **/usr** is where the operating system binaries and libraries are stored and is read-only.
  - **/etc**, **/boot**, **/var** are writable on the system but only intended to be altered by the Machine Config Operator.
  - **/var/lib/containers** is the graph storage location for storing container images.

RHCOS & anti-virus scanners

- Can anti-virus scanners be installed on RHCOS?
  - Solutions, such as anti-virus scanners, can be deployed as daemonsets or container images. However, few, if any anti-virus vendors are delivering their software in this form.
  - We recommend talking with your anti-virus vendor and asking what solutions they have available for a container-optimized OS.
  - However, in a 2019 paper from Gartner on Cloud Workload Protection Platforms, recommends that clients "Replace antivirus (AV)-centric strategies with a "zero-trust execution"/default deny/application control approach to workload protection where possible, even if used only in detection mode." [1]

1. Gartner: Market Guide for Cloud Workload Protection Platforms, ID G00356240, April 8, 2019

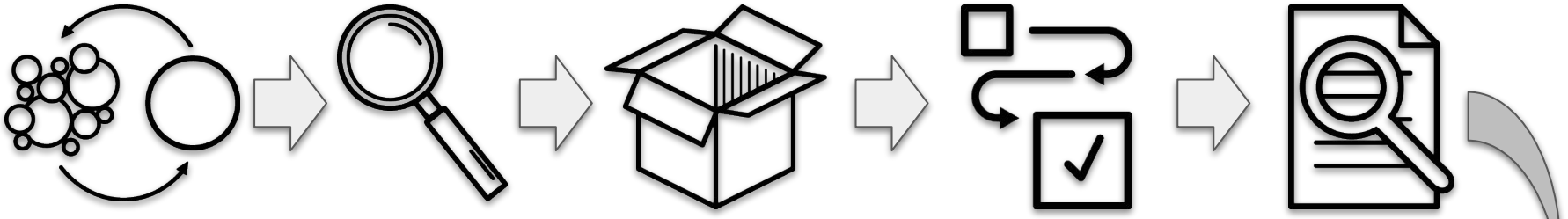
It’s still Red Hat Enterprise Linux

- To see the **packages installed**, run the same commands as on a RHEL system: ṙpm -qa |sort`
  -
- To see which **ports are in use**: ṡs -tulpṅ
-

AGENDA

- About RHEL CoreOS
- **Security of the code**
- System Management
- Crypto roots
- Securing Workloads
- Data protection
- Audit and Compliance
- Security and Hardware

Supply Chain Security



UPSTREAM FIRST!
Community Leadership

Red Hat Package
Review

Track packages for
release in Fedora

Some packages are
selected for RHEL /
RHEL CoreOS

Static Code Analysis

Compiler Flags set for
hardening and security

Extensive QE testing
per release

OS Image created
and digitally
signed

Secure
Distribution

Continuous
security updates
via OpenShift z
stream releases

Red Hat

Common Vulnerabilities and Exposures (CVE)

Patches are applied in a rolling fashion across the OpenShift cluster

Across all Red Hat products, and for all issue severities, we fixed 1,313 vulnerabilities by releasing more than 968 security advisories in 2019.

If you look solely at Linux vendors, the same CVE can have different effects, depending on how the product is compiled or deployed.

Manage your risks. Don't let your risks manage you. RHEL CoreOS can help.

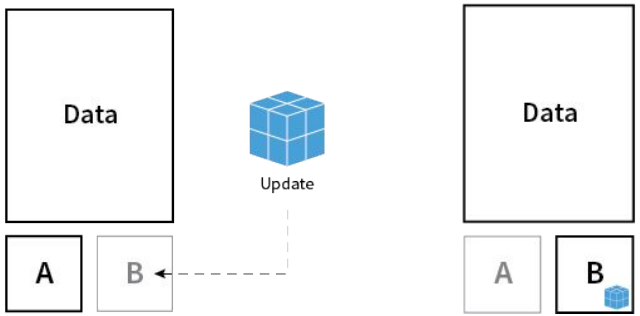
Red Hat Product Security Risk Report 2019

Transactional Updates via rpm-ostree

Versioning and Simplifying OS Updates

Transactional updates ensure that RHEL 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.
- Updates encapsulated in container images
- file system and package layering is available for hotfixes and debugging
- This is how CVE fixes as well as other types of fixes are delivered



Red Hat

AGENDA

- About RHEL CoreOS
- Security of the code
- **System Management**
- Crypto roots
- Securing Workloads
- Data protection
- Audit and Compliance
- Security and Hardware

One Touch provisioning via Ignition

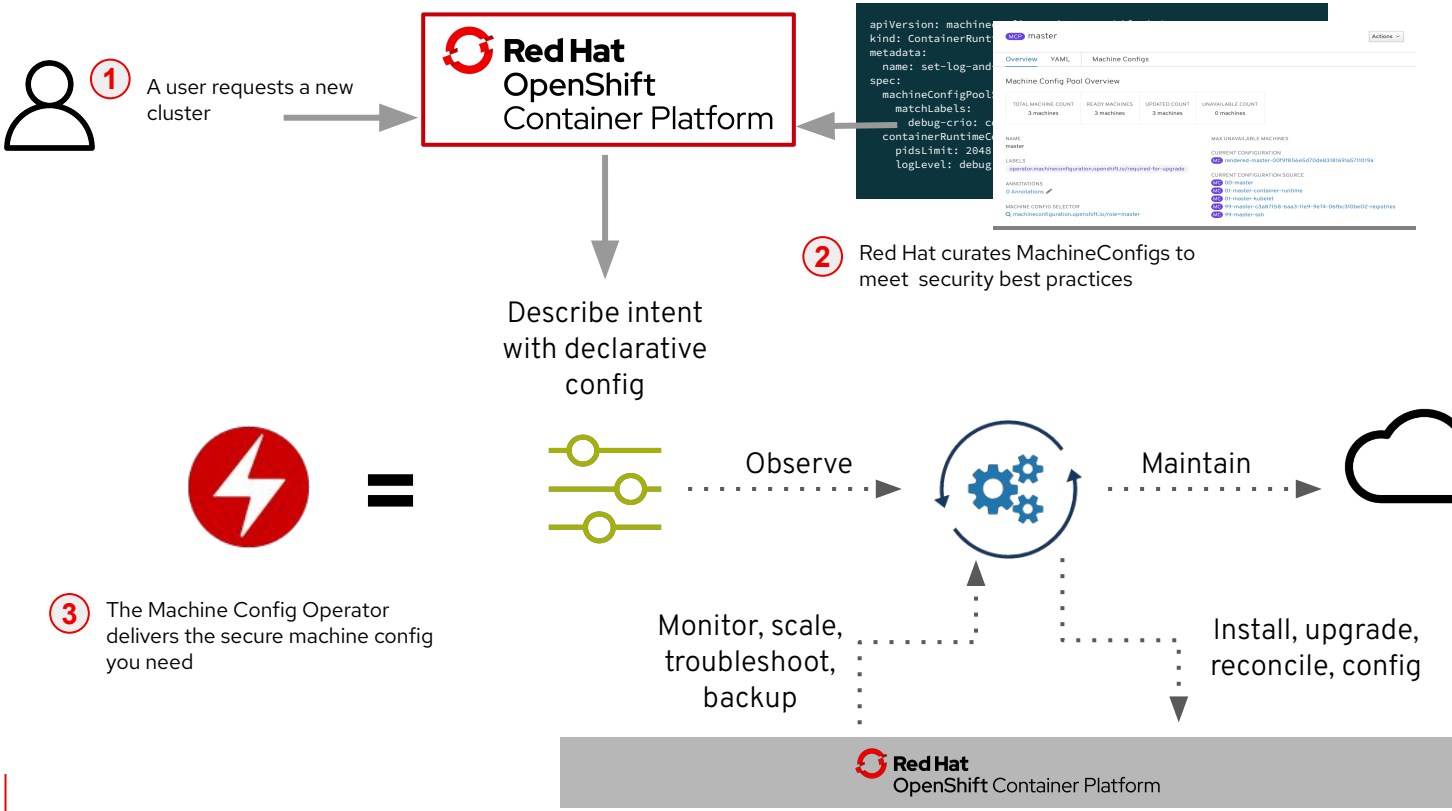
Machine generated; machine validated

Ignition applies a declarative node configuration early in the boot process. Ignition unifies kickstart and cloud-init.

- Generated via openshift-install & MCO
- Configures storage, systemd units, user, & remote configs
- Executed in the initramfs


```
{
 "ignition": {
  "config": {},
  "timeouts": {},
  "version": "2.1.0"
 },
 "passwd": {
  "users": [
   {
    "name": "core",
    "passwordHash": "$6$43y3tkl...",
    "sshAuthorizedKeys": [
     "key1"
    ]
   }
  ]
 },
 "storage": {},
 "systemd": {}
}
```

Openshift Machine Config Operator: Monitor for Configuration Drift



Red Hat

Machine Config Operator

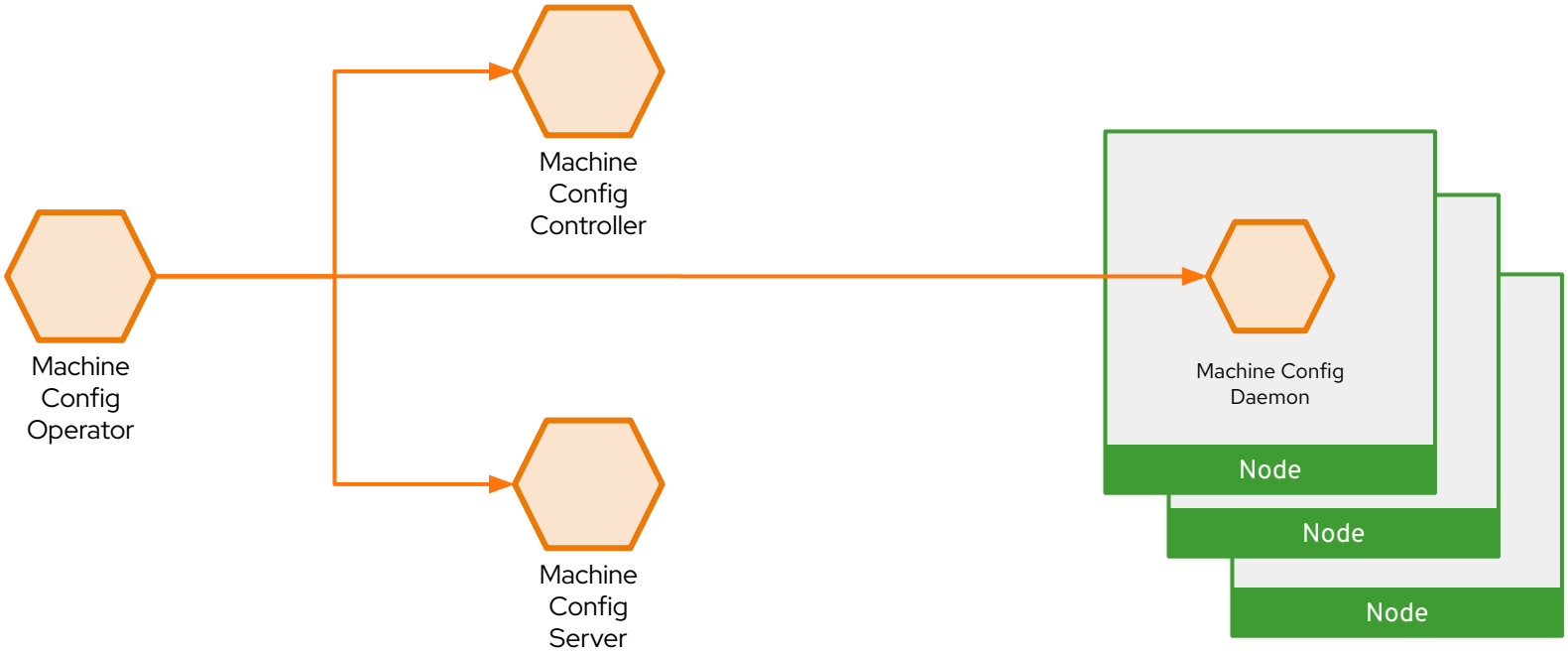
A Kube-native way to configure hosts

OS configuration is stored and applied across the cluster via the Machine Config Operator.

- Subset of ignition modules applicable post provisioning
  - SSH keys
  - Files
  - systemd units
  - kernel arguments
- Standard k8s YAML/JSON manifests
- Desired state of nodes is checked regularly
- Can be paused to suspend operations

```yaml
# test.yaml
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: test-file
spec:
  config:
    storage:
      files:
      - contents:
          source: data:,hello%20world%0A
          verification: {}
        filesystem: root
        mode: 420
        path: /etc/test
```

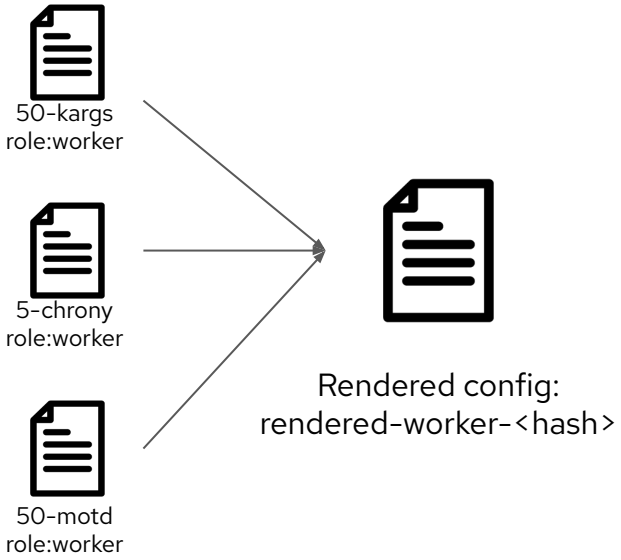
Operator/Operand Relationships



Red Hat

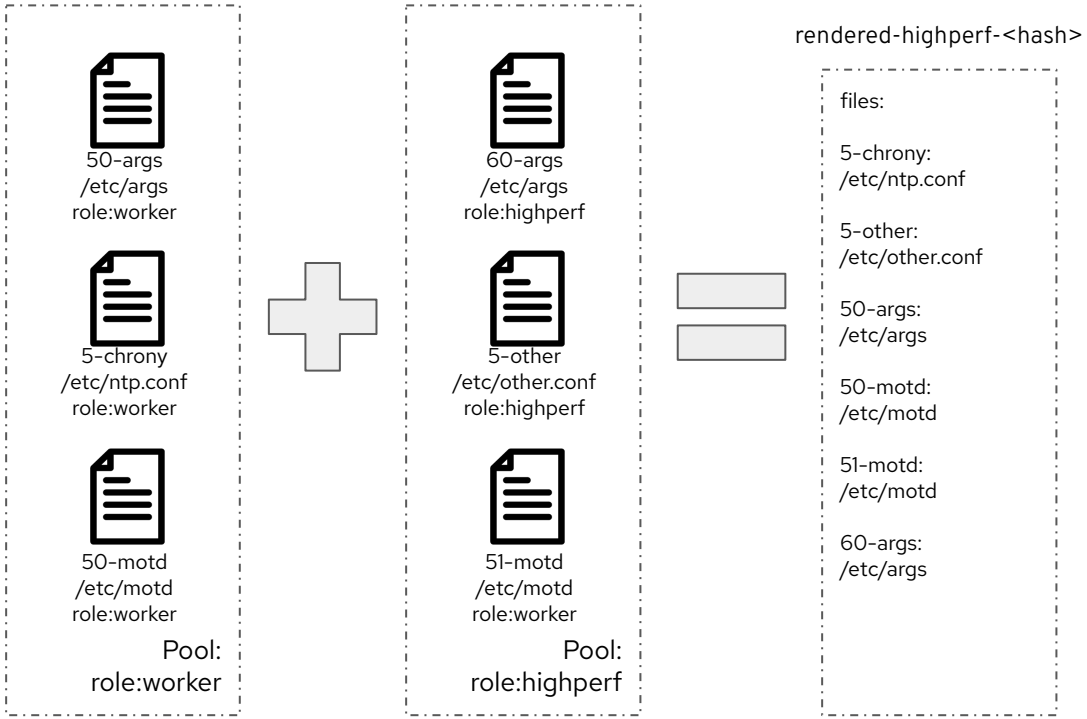
Machine Config and Machine Config Pool

Inheritance-based mapping of configuration to nodes



Custom Machine Config Pools

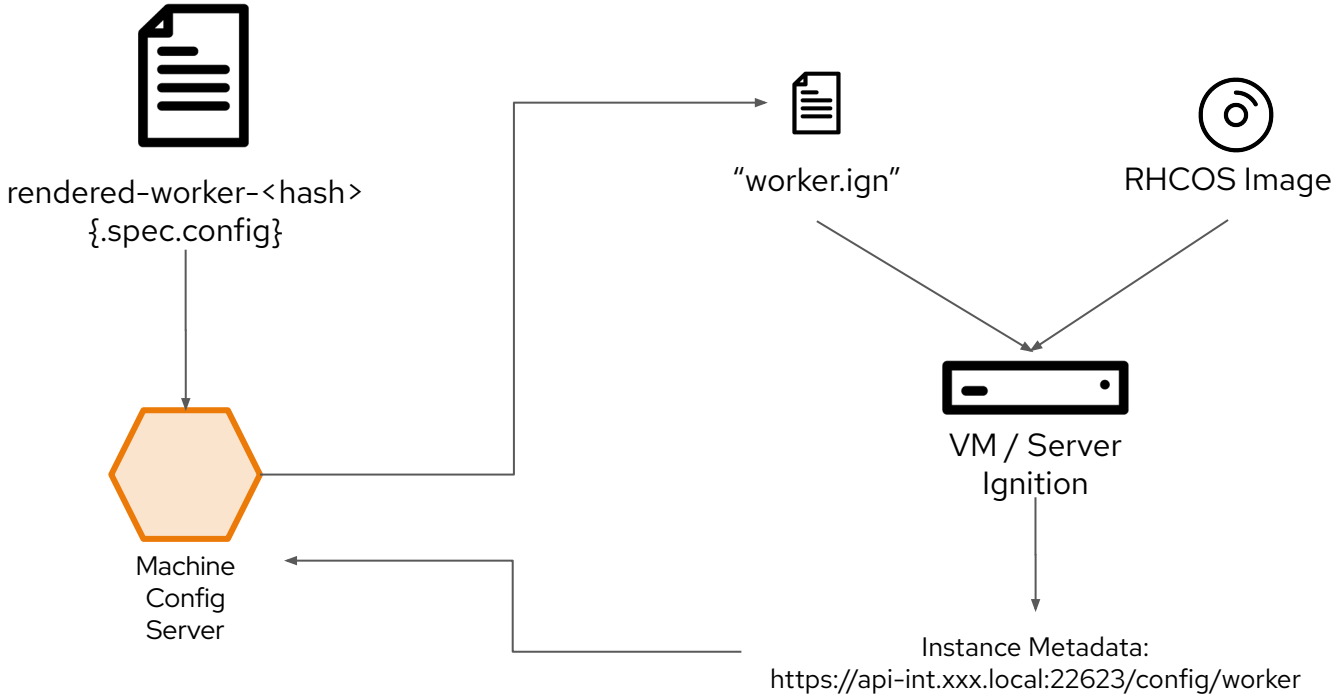
Hierarchical/layered configuration rendering



Red Hat

Machine Config Server

Providing Ignition configuration for provisioning



Red Hat

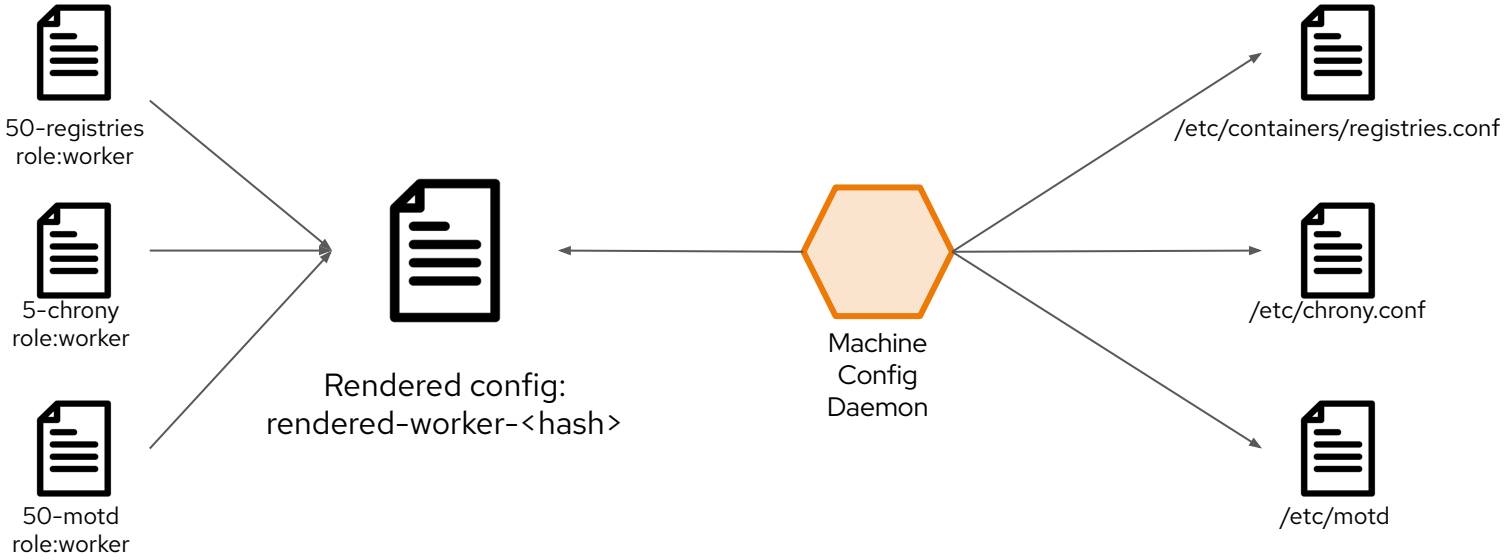
Machine Config Server

Identical nodes at massive scale

Red Hat

Machine Config Daemon

Preventing drift



Red Hat

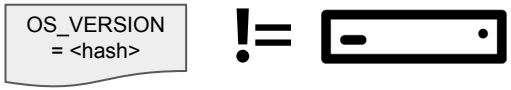
Machine Config Daemon

Acting on drift

The MCO coordinates with the MCD to perform the following actions, in a rolling manner, when OS updates and/or configuration changes are applied:

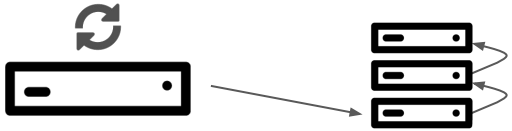
- Cordon / uncordons nodes
- Drain pods
- Stage node changes
  - OS upgrade
  - config changes
  - systemd units
- Reboot

1. Validates node state matches desired state



2. Validate cluster state & policy to apply change

3. Change is rolled across cluster

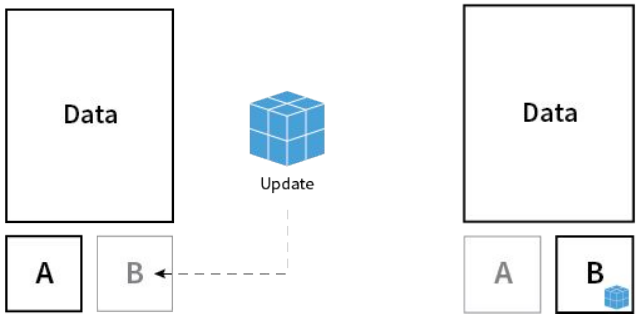


Transactional Updates via rpm-ostree

Versioning and Simplifying OS Updates

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

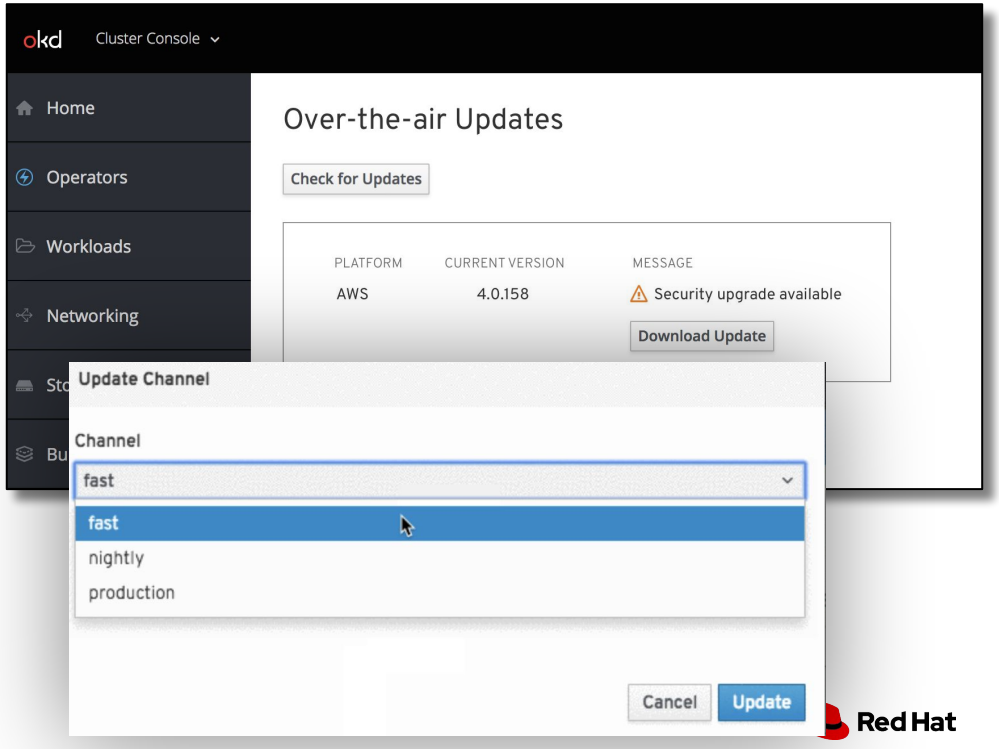
- Each OS update is versioned and tested as an complete image.
- Updates encapsulated in container images
- file system and package layering is available for hotfixes and debugging
- This is how CVE fixes as well as other types of fixes are delivered



Over-the-air Updates - Delivery Mechanism

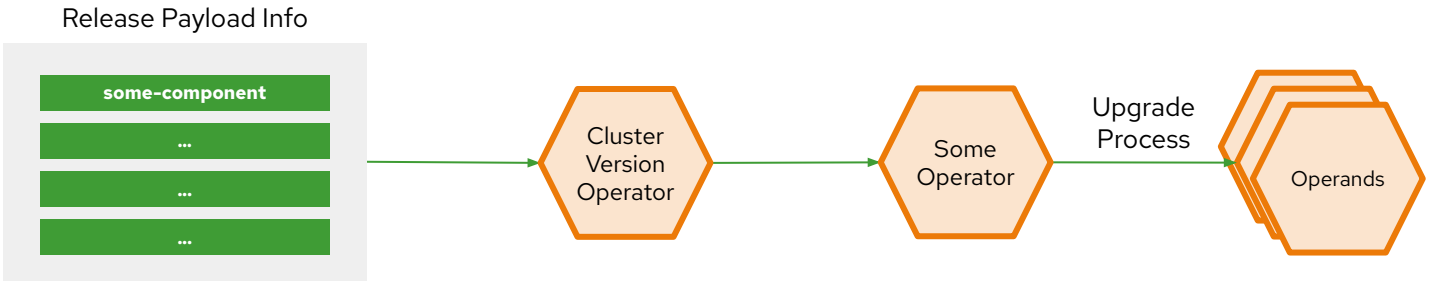
For Hosted and on-prem Deployments

- Updates can be driven from either *cloud.openshift.com* and/or the Cluster Console
- All updates are delivered via container images
- Auto-update support
- Manual updates will be supported for disconnected environments
  - Tooling to automate updates will be added in later release
  - Single source of content to mirror

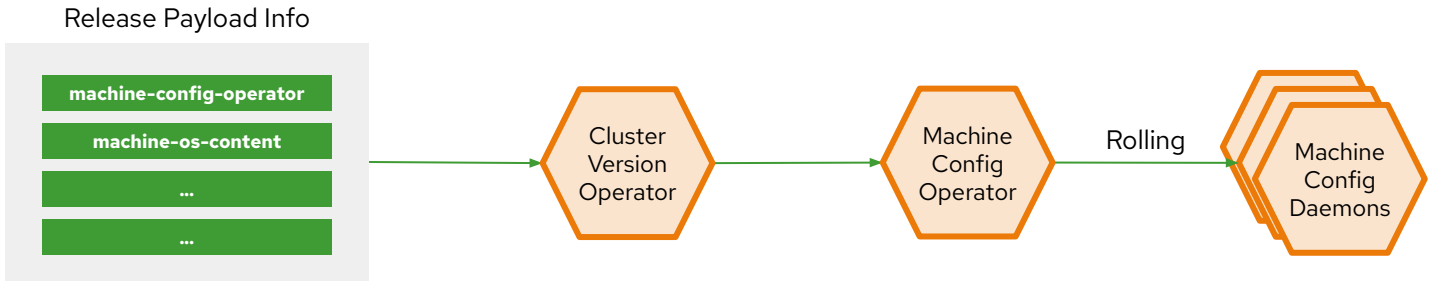


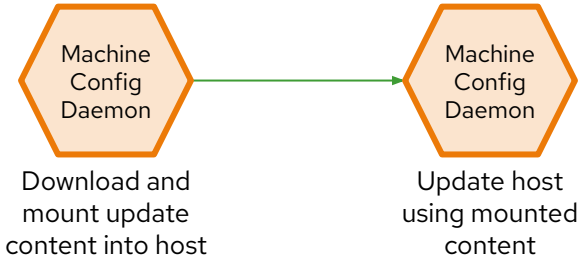
Red Hat

Over-the-air updates: Cluster Components



Over-the-air updates: Nodes





AGENDA

- About RHEL CoreOS
- Security of the code
- System Management
- **Crypto roots**
- Securing Workloads
- Data protection
- Audit and Compliance
- Security and Hardware

Crypto Libraries

Core Crypto Libraries

- OpenSSL
- GnuTLS
- NSS

Other components that implement crypto

- BIND
- OpenSSH
- Python
- Samba

OpenShift 4 and Fips 140-2

RHEL CoreOS FIPS mode

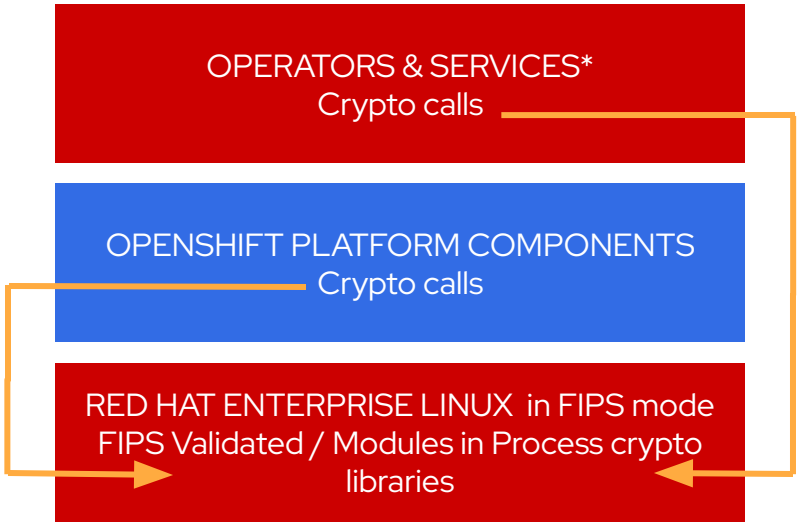
- Built with RHEL 8 binaries
- Configure at install to enforce use of FIPS Modules in Process* modules

OpenShift calls FIPS validated crypto

- When running on RHEL 7.6 in FIPS mode, OpenShift components bypass go cryptographic routines and call into a RHEL FIPS 140-2 validated cryptographic library
- This feature is specific to binaries built with the RHEL go compiler and running on RHEL

FIPS ready Services

- When built with RHEL 7 base image



*When built with RHEL base images

More about RHEL go and FIPS 140-2

https://csrc.nist.gov/Projects/cryptographic-module-validation-program/Modules-In-Process/Modules-In-Process-List

Red Hat

Federal Information Processing Standards (FIPS)

Enabling FIPS 140 mode

Red Hat Enterprise Linux 7

```
# yum install dracut-fips
# yum install dracut-fips-aesni
# dracut -v -f
[Modify boot loader configuration.]
$ df /boot
$ blkid /dev/sda1
[Edit file]
# grub2-mkconfig -o /etc/grub2.cfg
# reboot
```

Red Hat Enterprise Linux CoreOS

Set `fips: true` in the `install-config.yaml` file before you deploy your cluster.

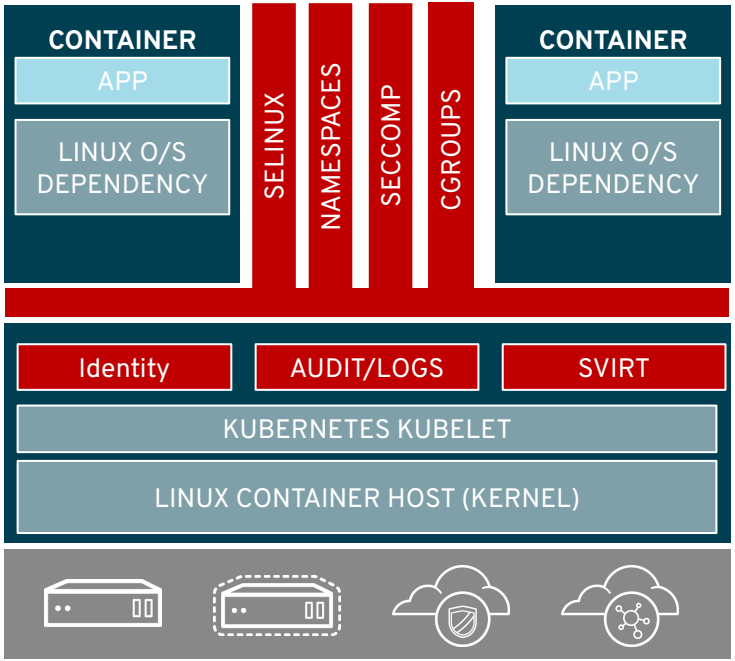
Official FIPS certification every minor release

AGENDA

- About RHEL CoreOS
- Security of the code
- System Management
- Crypto roots
- **Securing Workloads**
- Data protection
- Audit and Compliance
- Security and Hardware

Container security starts with Linux security

- Security in the RHEL host applies to the container
- RHEL enables container multitenancy
- SELinux and Kernel Namespaces are the one-two punch no one can beat
- Protects not only the host, but containers from each other
- RHEL CoreOS provides minimized attack surface



Red Hat

SELinux - Mandatory Access Control

Link to the book

Red Hat

SELinux & Other Controls

- Everything in the operating system has a label
- Policy defines the interaction between a labeled process and labeled resources
- Policy comes with the distribution but you can add your own
- Policy is enforced by Kernel
- Enforcement is turned on by default
- Systems with SELinux enabled are less susceptible (e.g. container breakouts)

- Linux capabilities - break **root** privileges into smaller groups and control them
- Libseccomp - syscall filtering mechanism
- Namespaces - isolation primitives

SELinux mitigates container runtime vulnerabilities

SELinux Mitigates container Vulnerability

January 13, 2017 | Joe Brockmeier

< Back to all posts Tags

A new CVE, (CVE-2016-9962), for the docker container runtime and runc were rec
released. Fixed packages are being prepared and shipped for RHEL as well as Fedo
CentOS. This CVE reports that if you `exec` d into a running container, the processe
the container could attack the process that just entered the container.

https://www.redhat.com/en/blog/selinux-mitigates-container-vulnerability

Latest container exploit (runc) can be blocked by SELinux

February 28, 2019 | Dan Walsh

< Back to all posts Tags: *Security*, *Containers*

A flaw in runc (CVE-2019-5736), announced last week, allows container processes to "escape" their containment and execute programs on the host operating system. The good news is that well-configured SELinux can stop it.

https://www.redhat.com/en/blog/latest-container-exploit-runc-can-be-blocked-selinux

Use OpenShift Security Context Constraints to manage these controls

Allow administrators to control permissions for pods

Restricted SCC is granted to all users

By default, no containers can run as root

Admin can grant access to privileged SCC

Custom SCCs can be created

```
$ oc describe scc restricted
Name:                                   restricted
Priority:                               <none>
Access:
  Users:                                <none> (1)
  Groups:                               system:authenticated (2)
Settings:
  Allow Privileged:                     false
  Default Add Capabilities:             <none>
  Required Drop Capabilities:           KILL,MKNOD,SYS_CHROOT,SETUID,SETGID
  Allowed Capabilities:                 <none>
  Allowed Seccomp Profiles:             <none>
  Allowed Volume Types:                 configMap,downwardAPI,emptyDir,persistentVolumeClaim,projected,
  Allow Host Network:                   false
  Allow Host Ports:                     false
  Allow Host PID:                       false
  Allow Host IPC:                       false
  Read Only Root Filesystem:            false
  Run As User Strategy: MustRunAsRange
    UID:                                <none>
    UID Range Min:                      <none>
    UID Range Max:                      <none>
  SELinux Context Strategy: MustRunAs
    User:                               <none>
    Role:                               <none>
    Type:                               <none>
    Level:                              <none>
  FSGroup Strategy: MustRunAs
    Ranges:                             <none>
  Supplemental Groups Strategy: RunAsAny
    Ranges:                             <none>
```

1. Lists which users and service accounts the SCC is applied to.
2. Lists which groups the SCC is applied to.

([Security Context Constraints])

Red Hat

AGENDA

- About RHEL CoreOS
- Security of the code
- System Management
- Crypto roots
- Securing Workloads
- **Data protection**
- Audit and Compliance
- Security and Hardware

Volume Encryption

Policy Based Disk Encryption

- Provides encryption for local storage
- Addresses disk/image theft
- Platform/cloud agnostic implementation
- TPM/vTPM (v2) and Tang endpoints for automatic decryption

RHEL CoreOS Disk Encryption

- Requires 4.3 or later
- This is a Day 1 configuration
- IPI: Only supported with TPM 2.0
- UPI bare-metal: only supported when using the CoreOS Installer
  - TPM 2 requires that the TPM chip be enabled in the BIOS
  - or
  - NBDE with a Tang server
- The key is generated randomly by RHCOS
  - There is no bring-your-own key.
  - There is no support for key-escrow or Cloud-based KMS.

Automated policy based encryption

- Root is now housed in a LUKS container whether encrypted or not
  - RHCOS 4.3 disk images use the ‘null-cipher’ (aka no encryption)
- Early in first boot, an Ignition modules gets the files
- The module looks for /etc/clevis.json:
  - If NOT found, then encryption is not applied
  - Otherwise, encryption is activated
- User-intervention is not allowed
  - The passphrase is auto-generated by RHCOS
  - For TPM2: the passphrase is escrowed in hardware
  - For Tang: the passphrase is encrypted and stored in the LUKS header

Network Bound Disk Encryption (NBDE)

- NBDE means a disk can be automatically decrypted when the host has access to a network resource.
- If system cannot access the network, then the disk is not decrypted.
- For RHCOS, this means the system will NOT boot.
- Tang is an OpenSource project that provides a zero-knowledge of the secret. Tang is supported by Red Hat

Second Boot

- After initializing and bringing up bare-minimum service, the root disk is evaluated for meta-data.
  - Remember root is housed in a LUKS container
- If NOT encrypted:
  - ‘dm-linear’ maps the rootfs to /dev/mapper/coreos-luks-root-nocrypt
  - Use of ‘dm-crypt’ for null-cipher is very slow
- If encrypted:
  - The Clevis Pin is called to decrypt the passphrase
  - Root appears on /dev/mapper/coreos-luks-root

Disaster Recovery

- Tang: Ensure that the Tang Server is up.
  - The management of Tang is outside the scope of this training.
  - See Chapter 9 Configuring Automated Unlocking of Encrypted Volumes Using Policy-based Decryption
- TPM2:
  - It should be automatic
  - If the hard disk is removed from the chassis, the disk secret is lost
- If the secret is accessible, then proceed as normal
- Re-provisioning of the node is preferred
  - That's what the Machine Config Operator is for
  - Manual activation of the disk is possible, but not supported

Recommendations

- Encrypt primary and secondary volumes
- Use NBDE to decrypt volumes unattended
- OR bind volume to hardware using TPM
  - When you have access to hardware; e.g. running on bare metal on-premises
- UPI with bare metal is the most likely use case, as
  - AWS users already get disk encryption for free.
  - GCP provides data-at-rest already for disks
    - Although GCP Shielded VM's can provide virtual TPM2 devices for the paranoid
  - Azure root disks are likely too slow for any VM-based disk encryption
  - Network-back public Cloud disks are likely too slow
  - VMware and HyperV have the ability for a virtual TPM2 device

Protection at the Network Layer

- MACSec - IEEE standard for security in wired ethernet LANs.
- TLS - in RHEL CoreOS TLS 1.3 is default across the platform
  - TLS 1.2 redesigned (4 years in the making)
  - Less clutter, faster handshake
  - Modern crypto primitives (RSA-PSS, Ed25519)
  - Performance
  - Better privacy against passive observers
  - Supported in OpenSSL 1.1.1, GnuTLS, and NSS

Note: IPSec is not currently available in RHEL CoreOS. The roadmap includes support for IPSec via OVN sometime after OVN is GA.

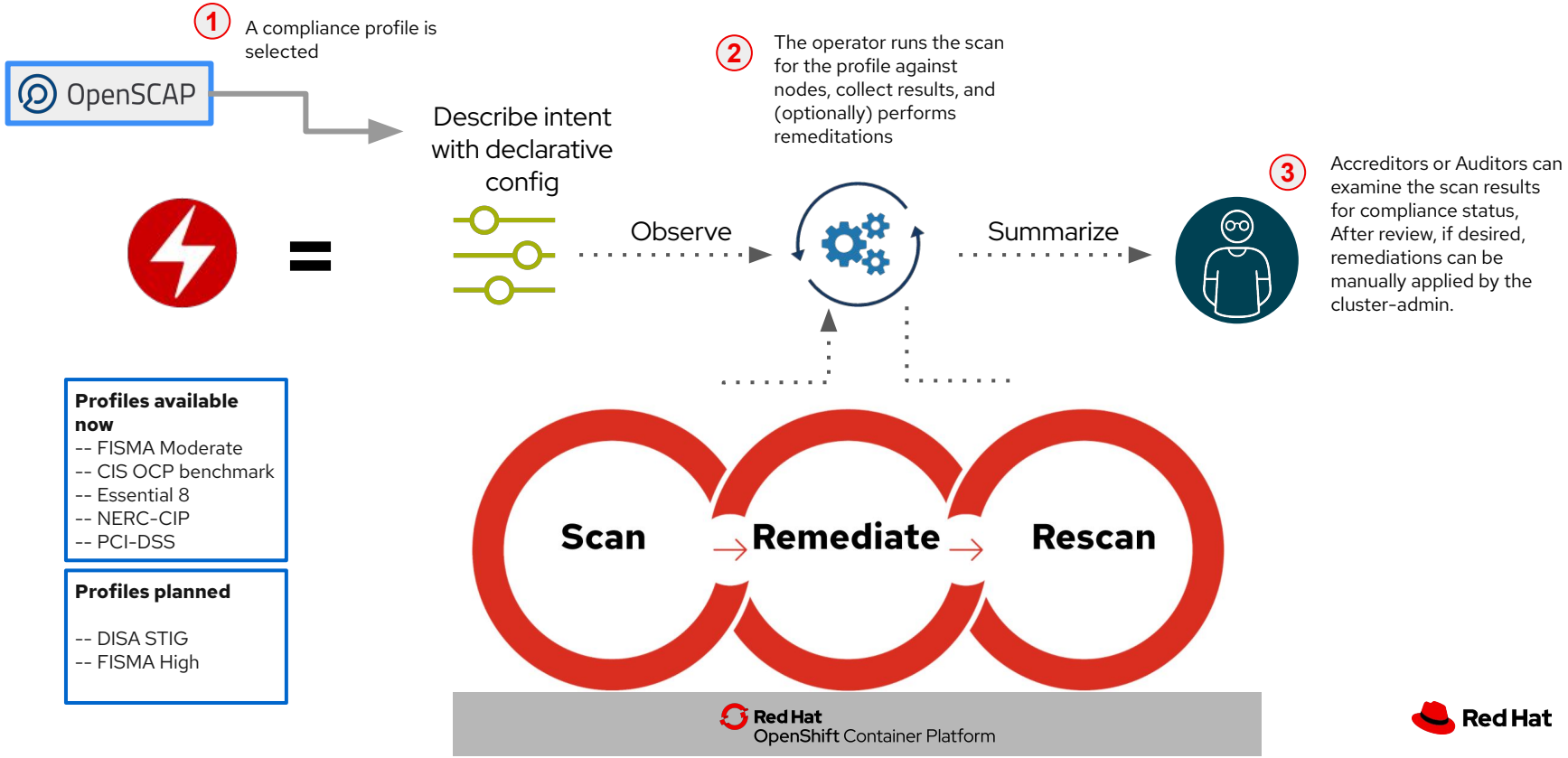
AGENDA

- About RHEL CoreOS
- Security of the code
- System Management
- Crypto roots
- Securing Workloads
- Data protection
- **Audit and Compliance**
- Security and Hardware

Auditd

- Low level system wide auditing system
- Integrated in Kernel and userspace - no security event escapes!
- Very detailed feed that meets all existing compliance standards
- Actively used by customers that need to adhere to tight security practices
- Auditd is included in RHEL CoreOS
- Host level audit logs are collected for forwarding by the OpenShift Logging Pipelines feature

Openshift Compliance Operator for Continuous Compliance



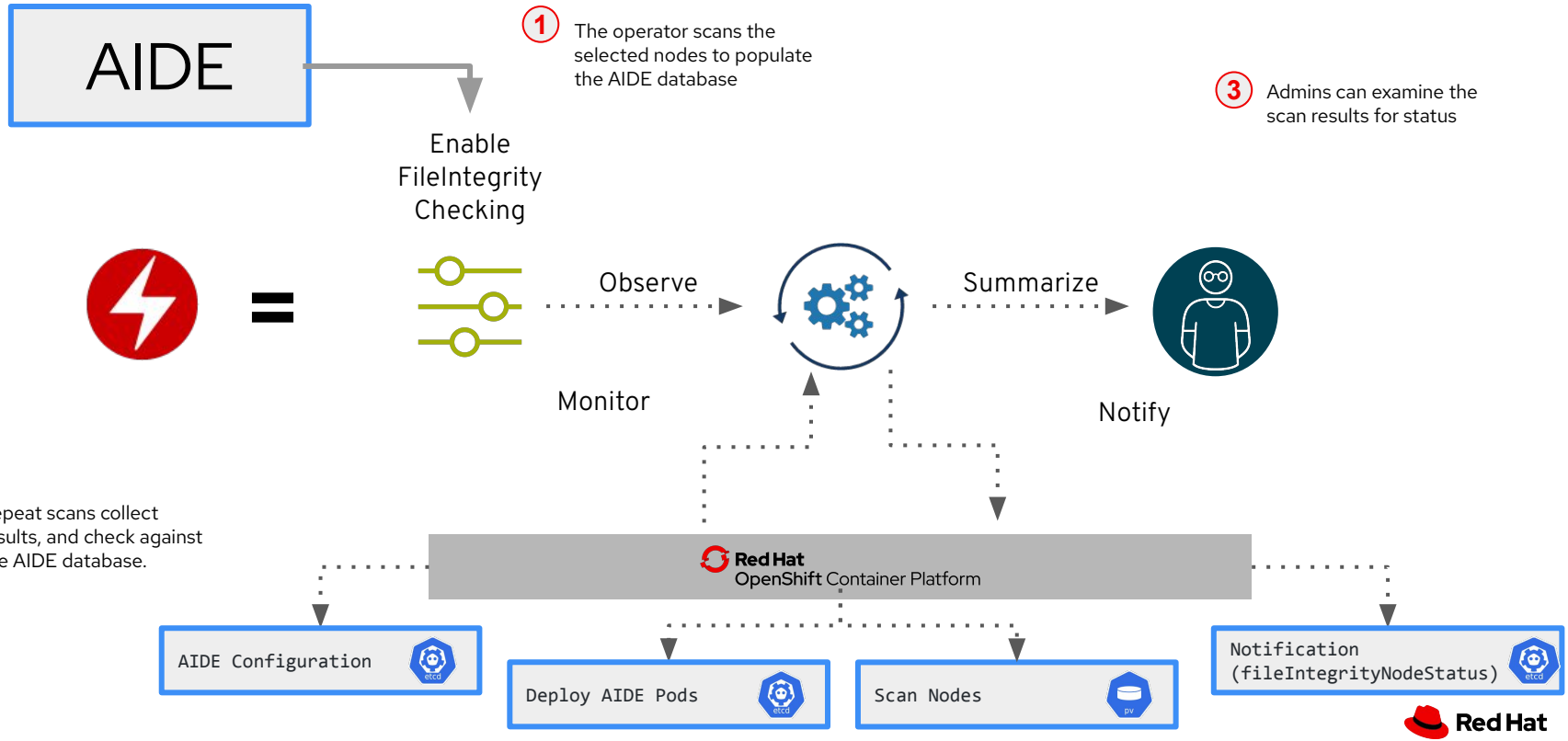
Red Hat

File Integrity

- Secure Boot - provides guarantee that a trusted, unmodified Kernel is loaded*
- File integrity monitoring
  - /usr is read only
  - Machine Config Operator marks nodes with wrongly configured files as degraded
- Roadmap: a file integrity operator using AIDE
  - Advanced Intrusion Detection Environment is a utility that creates a database of files on the system, and then uses that database to ensure file integrity and detect system intrusions

Chapter 2. Securing RHEL during installation Red Hat Enterprise Linux 8

Openshift File Integrity Operator



AGENDA

- About RHEL CoreOS
- Security of the code
- System Management
- Crypto roots
- Securing Workloads
- Data protection
- Audit and Compliance
- **Security and Hardware**

Hardware integration

- TPM 2.0 - available for use by the applications, utilized by NBDE
- Hardware enablement:
  - Crypto offload to hardware (networking)
  - HSM
  - Special hardware vendor security features (work in different stages of readiness):
    - Encrypted memory on the HW level
    - Control-flow Enforcement Technology (CET) - Intel
    - Trusted Execution Environments - (TEE)
- PKSC#11 interface to abstract access to key stores: HSMs, TPMs
- USBGuard -Roadmap. Targeted for 4.5
-

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


twitter.com/RedHat

Red Hat

Securing Kubernetes

Guidance from the CNCF Kubernetes Security Audit

" While Kubernetes facilitates high-availability workload deployments, the underlying hosts, components, and environment of a Kubernetes cluster must be configured and managed. This management has a direct impact on the capabilities of the cluster, and affects the behavior of an operator's composed objects.

With this in mind, the options available for configuring components of Kubernetes often fluctuate significantly in supported versions, and vary in their approach to default settings. This leads to a non-trivial amount of configuration required by an administrator to stand-up a functional cluster for a given workload.

More effort must then be spent maintaining the cluster to abide by these settings, especially when planning and executing upgrades of Kubernetes components."

Kubernetes Security Whitepaper, Trail of Bits, May 31, 2019

Securing the container host

Guidance from NIST & CNCF

Use container-specific host OSs instead of general-purpose ones to reduce attack surfaces.

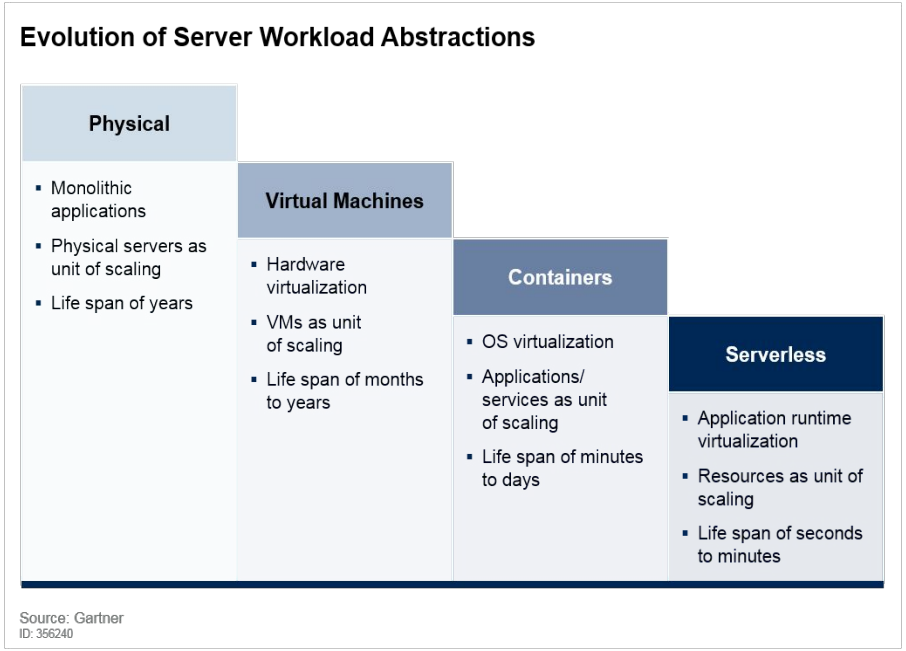
A container-specific host OS is a minimalist OS explicitly designed to only run containers, with all other services and functionality disabled, and with read-only file systems and other hardening practices employed. When using a container-specific host OS, attack surfaces are typically much smaller than they would be with a general-purpose host OS, so there are fewer opportunities to attack and compromise a container-specific host OS. Accordingly, whenever possible, organizations should use container-specific host OSs to reduce their risk.

NIST Special Publication 800-190
Application Container Security Guide

Also recommended by the CNCF Cloud Native Security Whitepaper
https://github.com/cncf/sig-security/blob/master/security-whitepaper/CNCF_cloud-native-security-whitepaper-Nov2020.pdf

Securing cloud native workloads

Guidance from Gartner



"The best way to secure these rapidly changing and short-lived workloads is to start their protection proactively in the development phase … so that when a workload is instantiated in production, it is "born" protected."

"Replace antivirus (AV)-centric strategies with a "zero-trust execution"/default deny/application control approach to workload protection where possible…." [1]

1. Gartner: Market Guide for Cloud Workload Protection Platforms, ID G00356240, April 8, 2019

Red Hat

Frequently Asked Questions

OS Certification

Q: How can we certify that RHEL CoreOS meets compliance/security/risk requirements given they cannot install tools for such?

A: Most of the challenges around security can be addressed by understanding how RHEL CoreOS works, as discussed in this deck. For example, RHCOS is delivered as an OS image and updates are generally delivered as container images. OpenShift handles patching and updating the OS. So frequently updating the platform will solve address CVE remediation.

/usr on RHCOS is read-only. And configuration changes to RHCOS are done via MachineConfigs. The Machine Config Operator will mark a node as degraded if it notes configuration drift for files it manages. The MCO also ensures that all new nodes created in the cluster are launched with the same initial configuration.

Installing applications on RHCOS

Q: How can we install additional software or applications on RHEL CoreOS?

A: RHCOS is a container-optimized operating system (as recommended in NIST 800-190). It is specifically designed to run OpenShift. If you determine that you need additional solutions at the host level, look for vendors who distribute software in a cloud-native fashion; look for vendors who distribute their product as containers that can run on OCP.

If you want to use solutions that do not follow this path, you will need to containerize such solutions yourselves and use the cluster's scheduling capabilities to ensure they run on each node (this is typically a daemonset).

Adding a vulnerability agent to RHCOS

Q: Our current policy states that all OSes have to have a vulnerability agent scan installed to scan. We understand that the nodes on OpenShift are more like "appliances" and that as the OCP software is upgraded, the nodes also get patched. However, we need to document the variance and explain why OCP should receive an exception to our policy. Is there RedHat documentation that could serve to explain why the nodes should be treated as appliances and why the "OS" does not need the agents installed?

A: In addition to these slides, please see The OpenShift Security Guide, Chapter 2, Red Hat Enterprise Linux CoreOS Security

Finally, if necessary, you can download updates before installing them by creating a mirror of the Red Hat registry (quay.io). Once downloaded the update images can be scanned by tools that can scan container images and/or OS images.

SSH vs. no-SSH

Q: How can we prevent users from logging into RHEL CoreOS?

A. At installation, a user named core is created, with your ssh key assigned to that user. This allows you to log in to the cluster with that user name and your credentials. This user is effectively root on the host. If desired, once the installation is successful, customers can disable ssh access to RHCOS. However, if you need to debug a failed installation, an ssh key is needed. See: https://docs.openshift.com/container-platform/4.3/installing/installing-gather-logs.html

There are many debugging tools provided to the cluster administration that can be used without the need to ssh into the node. For example, the must-gather tool and the oc debug command.

See:
https://docs.openshift.com/container-platform/4.3/support/gathering-cluster-data.html
https://docs.openshift.com/container-platform/4.3/metering/metering-troubleshooting-debugging.html
https://docs.openshift.com/container-platform/4.3/cli_reference/openshift_cli/developer-cli-commands.html#cli-troubleshooting-commands_cli-developer-commands

Quay & Clair / Other vulnerability scanners

Q: What vulnerability scanning tools can be used to scan RHEL and RHEL CoreOS?

A. RHEL Server and RHEL CoreOS are Container Hosts, not container images. Clair (available with Quay) only scans container images, not hosts.

Red Hat base images, as well as other container images, are available in the Red Hat Container Catalog. The Red Hat Container Catalog provides a health grade for every Red Hat image. See https://access.redhat.com/articles/2803031

However, the preferred approach for assessing security, as described by Gartner, is to scan the container images. Again, customers can download RHCOS images and updates by mirroring Quay.io. Once downloaded, they can be scanned by scanners that can process OS images and/or container images.

Finally, Red Hat creates the container images, as well as the software and packages in them. Red Hat creates Errata which include mappings to CVEs from Mitre and identifification of which CVEs are fixed in the Errata. Red Hat has a supply chain of packages that we control: scanning Red Hat images with Clair is just verification that you have done the updates.

Other vulnerability scanners

Q. What can you tell us about other scanning solutions?

A. Third party solutions mostly scan container images, but may have value add that scans hosts too. Many 3rd party scanning tools include application layer content. Some, like Aqua Security, have running sidecar containers that scan the running containers. All of these are value adds that these security companies are selling.

If you want to know how any of these third party scanners work, you would need to ask them. Red Hat provides OVAL data that does map to our CVEs for both our hosts and our images Many Red Hat partners include our OVAL data in their vulnerability databases.

Today, Clair's vulnerability data is primarily focused on Linux distributions. We are planning to expand the set of data to include more application layer data.

Other vulnerability scanners

Q. Is Qualys Container Security able to scan running RHEL CoreOS hosts on OpenShift? Are they planning to build a certified OpenShift operator?

For reference, this user guide for Qualys includes CoreOS
https://www.qualys.com/docs/qualys-container-security-user-guide.pdf

A. This Qualys user guide is not referring to Red Hat Enterprise Linux CoreOS. Nor does it mention Red Hat Enterprise Linux v8. We are checking to see whether Qualys CS is supported on OpenShift 4.

We plan to add support to scan RHCOS images stored in Quay. This scanning would be done prior to deploying the image. Given the immutability of RHCOS this should be sufficient from a vulnerability scanning perspective.

Other vulnerability scanners

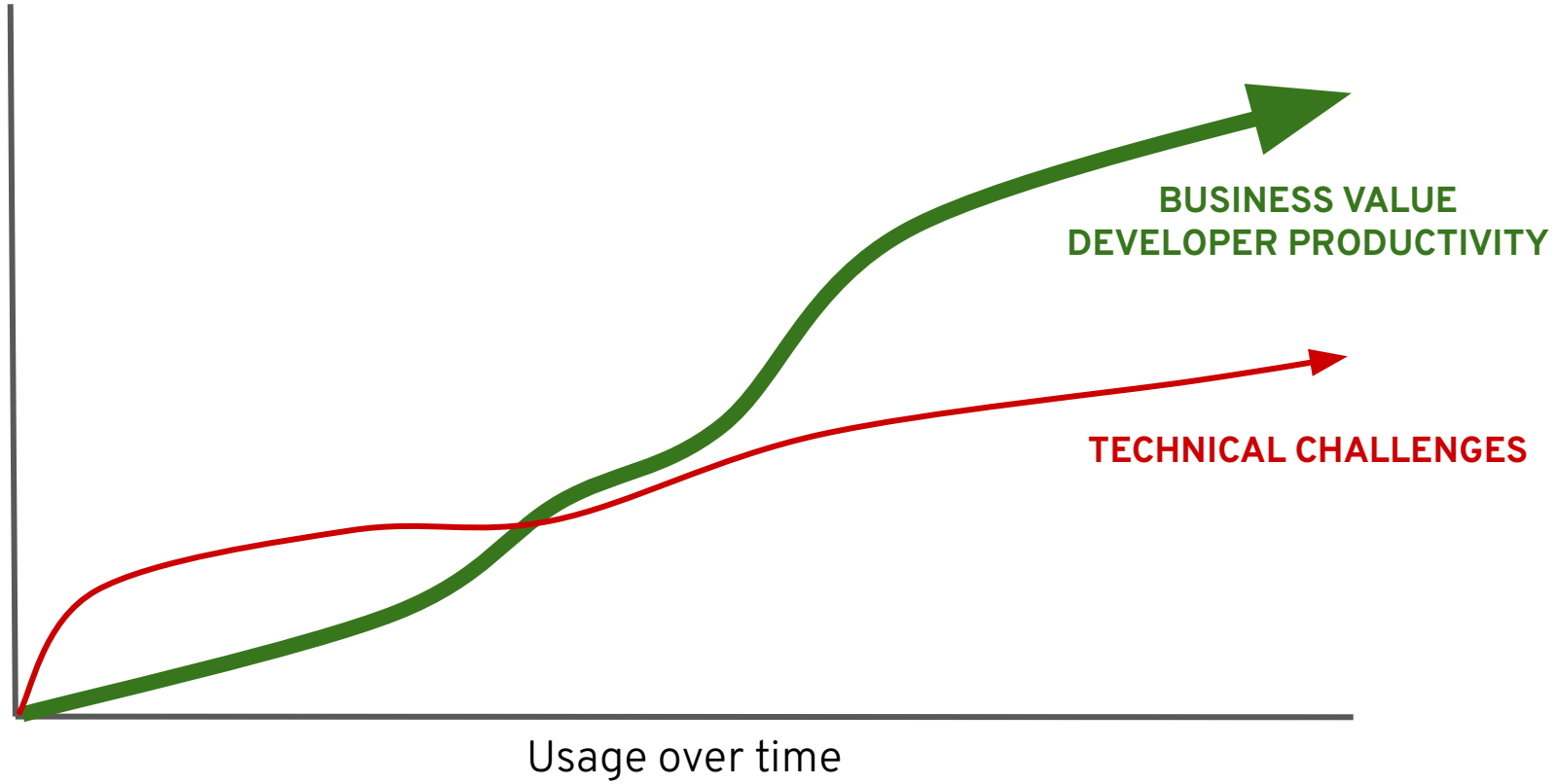
Q. Twistlock is a well known tool for container security — scanning images , hosts, registries etc. and they also have a certified OpenShift operator. Can Twistlock scan running RHEL CoreOS hosts?

https://www.twistlock.com/container-security/

A. Note that Twistlock is now known as Prisma Cloud. Twistlock does not scan running RHEL CoreOS hosts. Please check out the recording Dirk Herrmann did with the former CTO of Twistlock on NIST 800-190 where we explain what (RH) can do on the OpenShift, Quay and RHCOS side vs. what Twistlock can offer on top of that and why customers probably want to use both side-by-side:
https://blog.openshift.com/openshift-commons-briefing-container-deployment-and-security-best-practices-john-morello-twistlock-and-dirk-herrmann-red-hat/

Reimaging OpenShift: the value of OpenShift 4

The Realities of OpenShift 3



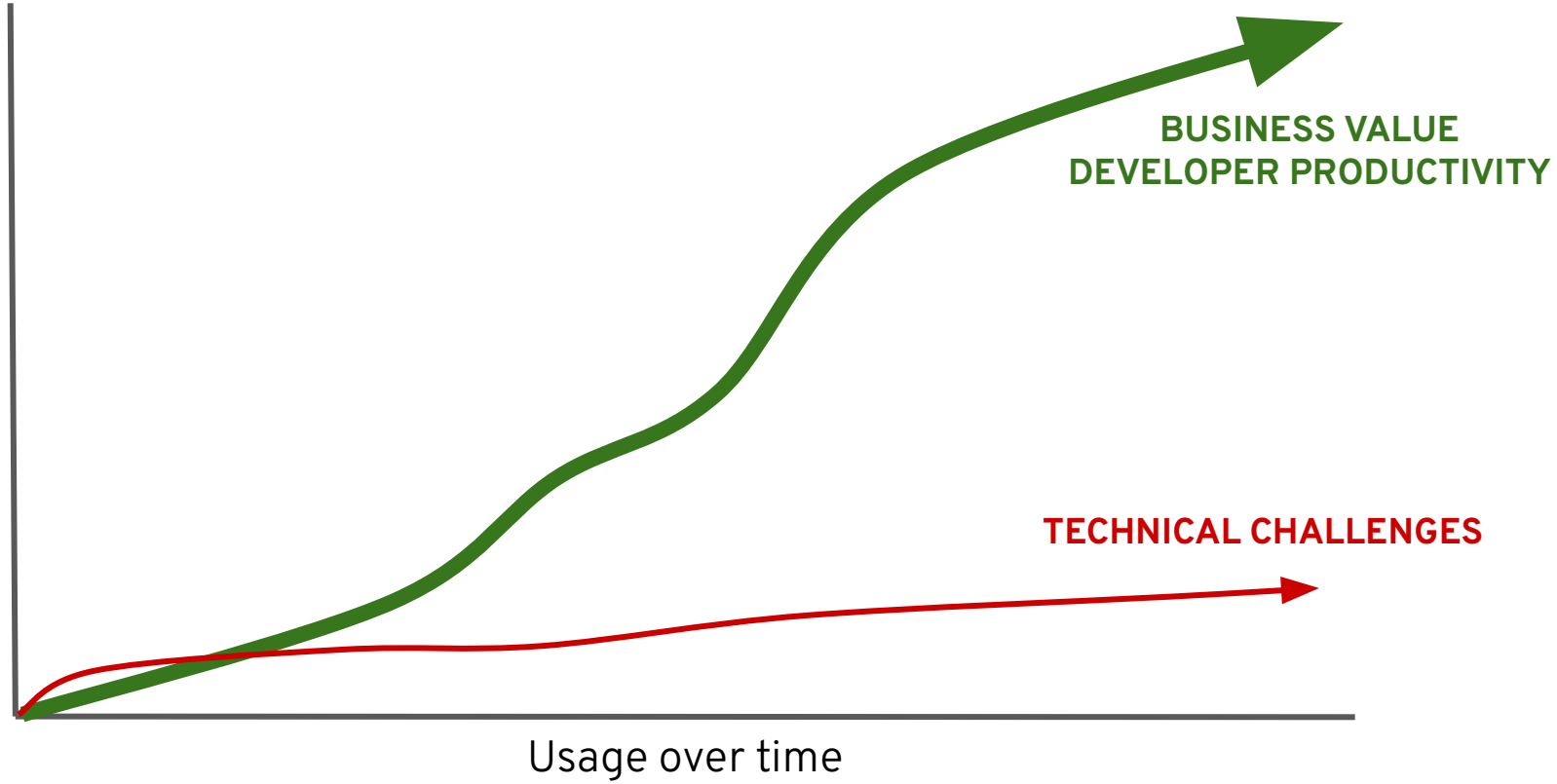
NOTE: This slide is for use with customers considering moving from OCP 3 to OCP 4

Red Hat

Reimaging OpenShift - OpenShift 4

Drive the Operator Ecosystem	Integrate Application Services	Improve Developer Services
Improve Platform Automation	Enable Telemetry	**Multi-Cloud Management**
Simplify Installations and Upgrades	**Provide Ops Greater Visibility**	**Simplify Node Management**

The Opportunity with OpenShift 4



Red Hat

Details on TPM2 configuration

TPM2 Configuration

- Provided via Machine Config
- TPM2 is the most simple: use this _exactly_.

```
"storage": {
     "files": [
          {
             "filesystem": "root",
             "path": "/etc/clevis.json",
             "contents": {
                  "source": "data:text/plain;base64,e30K"
             },
             "mode": 420
          }
       ]
}
```

Red Hat

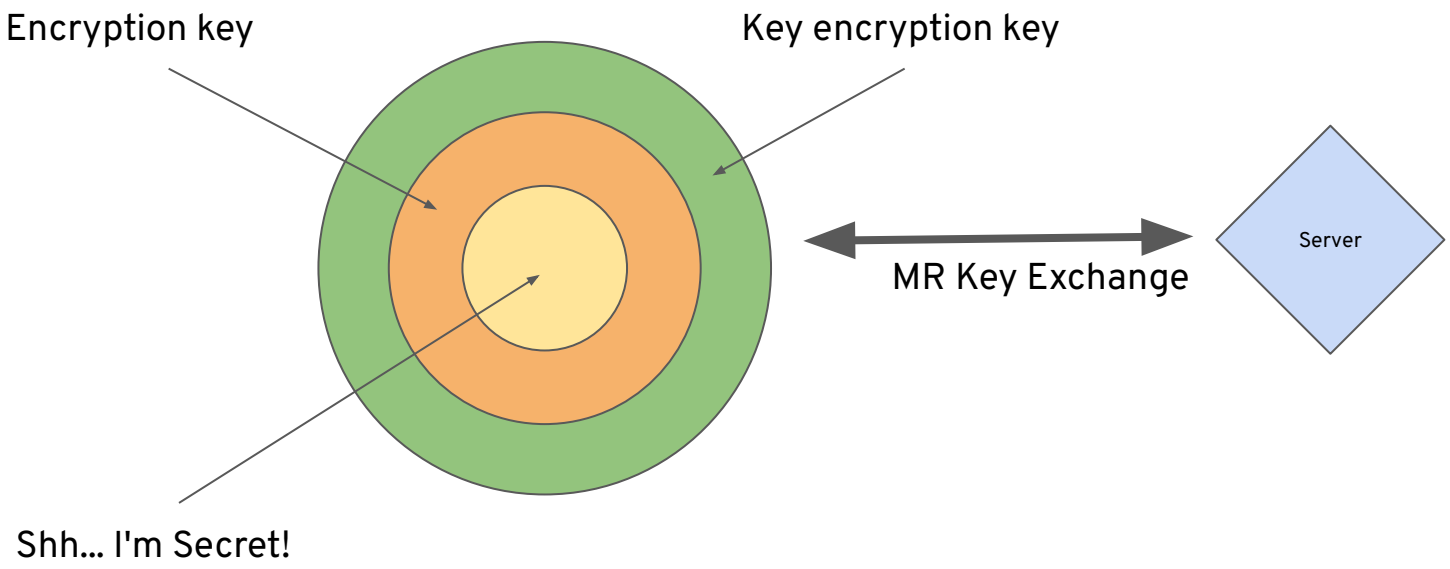
Screenshot

```
[    5.031644] coreos-cryptfs[782]: coreos-cryptfs: Fetching clevis config
[    5.072853] coreos-cryptfs[782]: coreos-cryptfs: Detected provided Clevis
config
[    5.089610] coreos-cryptfs[782]: coreos-cryptfs: detected pin=tpm2
[    5.167594] coreos-cryptfs[782]: coreos-cryptfs: Cleared token LUKS token on
/dev/vda4
[    5.169692] coreos-cryptfs[782]: coreos-cryptfs: generating new key
********************************************************************
NOTICE: Encrypting the root partition.  This can take a while depending on
    the size, speed of the disk, available entropy and CPU.

    Disk /dev/vda4 will be encrypted using AES-256 encryption and will
    configured for automatic unlock using Clevis. If encryption fails
    this node will need to be re-provisioned.

    This disk will be bound to the Clevis pin "tpm2" for
    automatic unlock.
********************************************************************
```

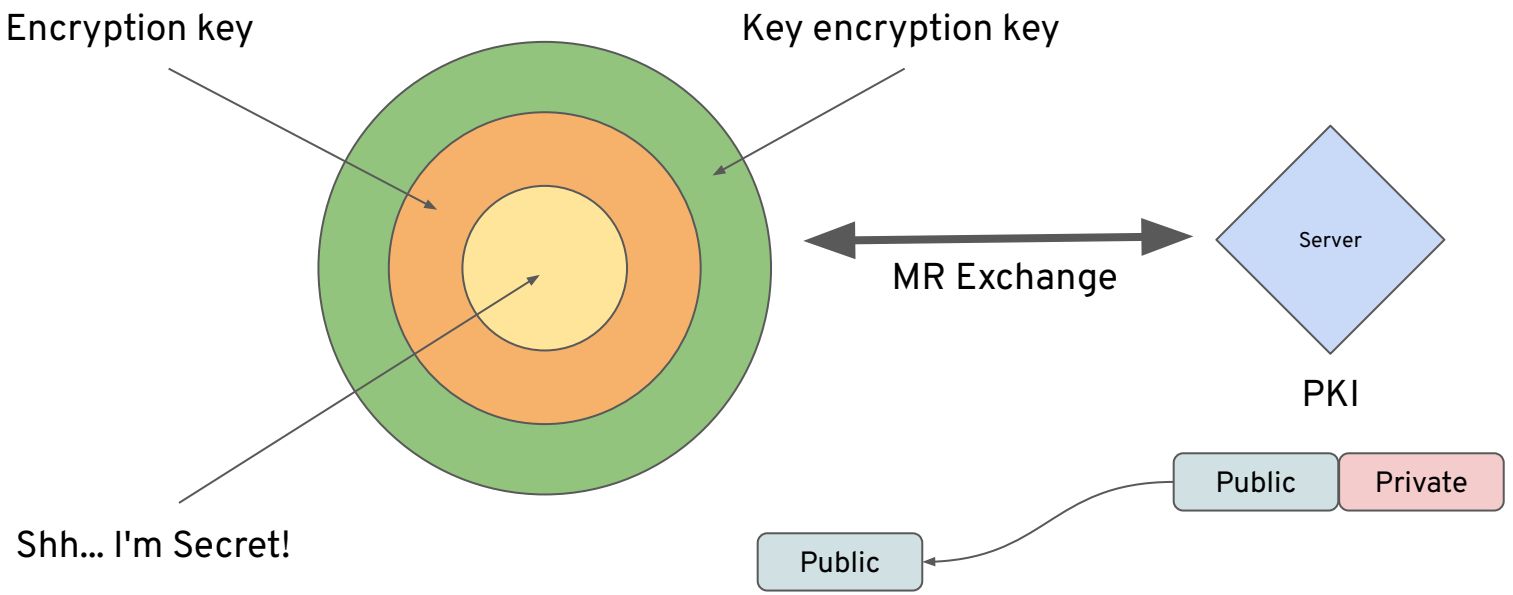
Red Hat

A graphical view of Network Bound Disk Encryption

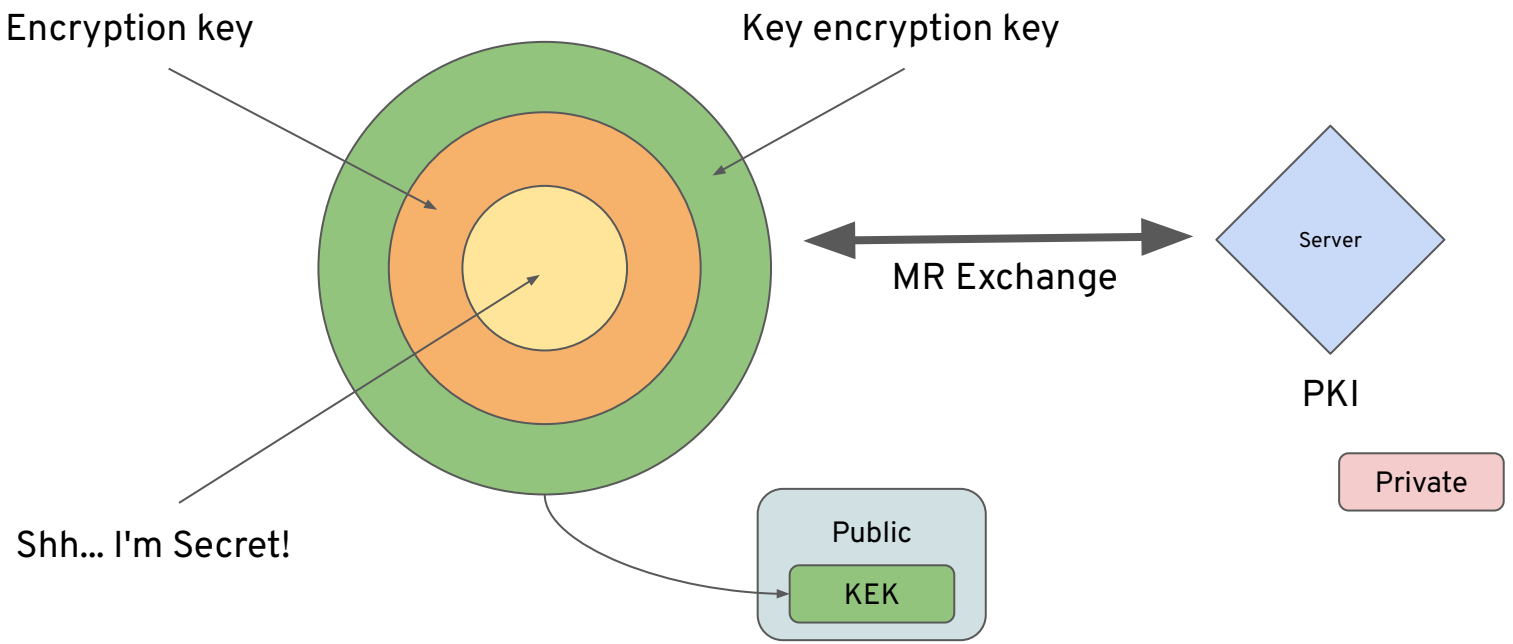
Network Bound Disk Encryption



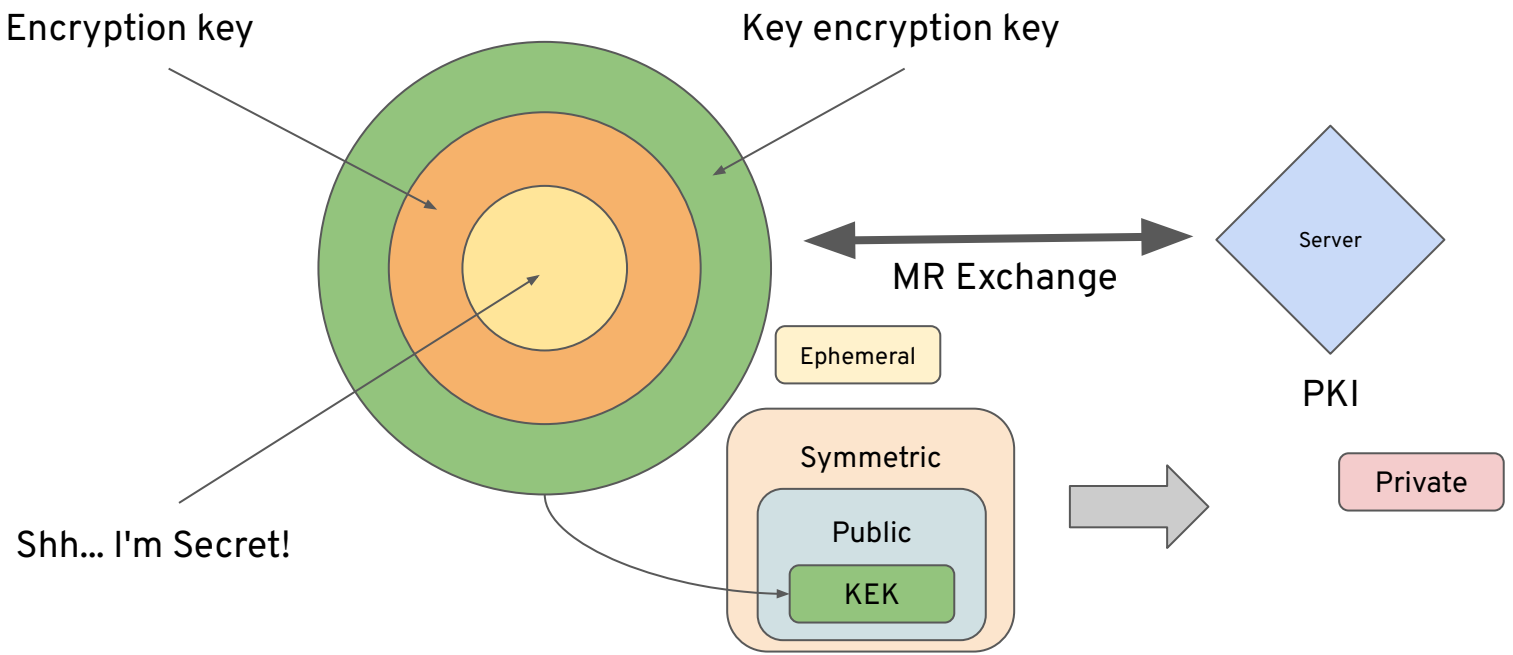
Network Bound Disk Encryption



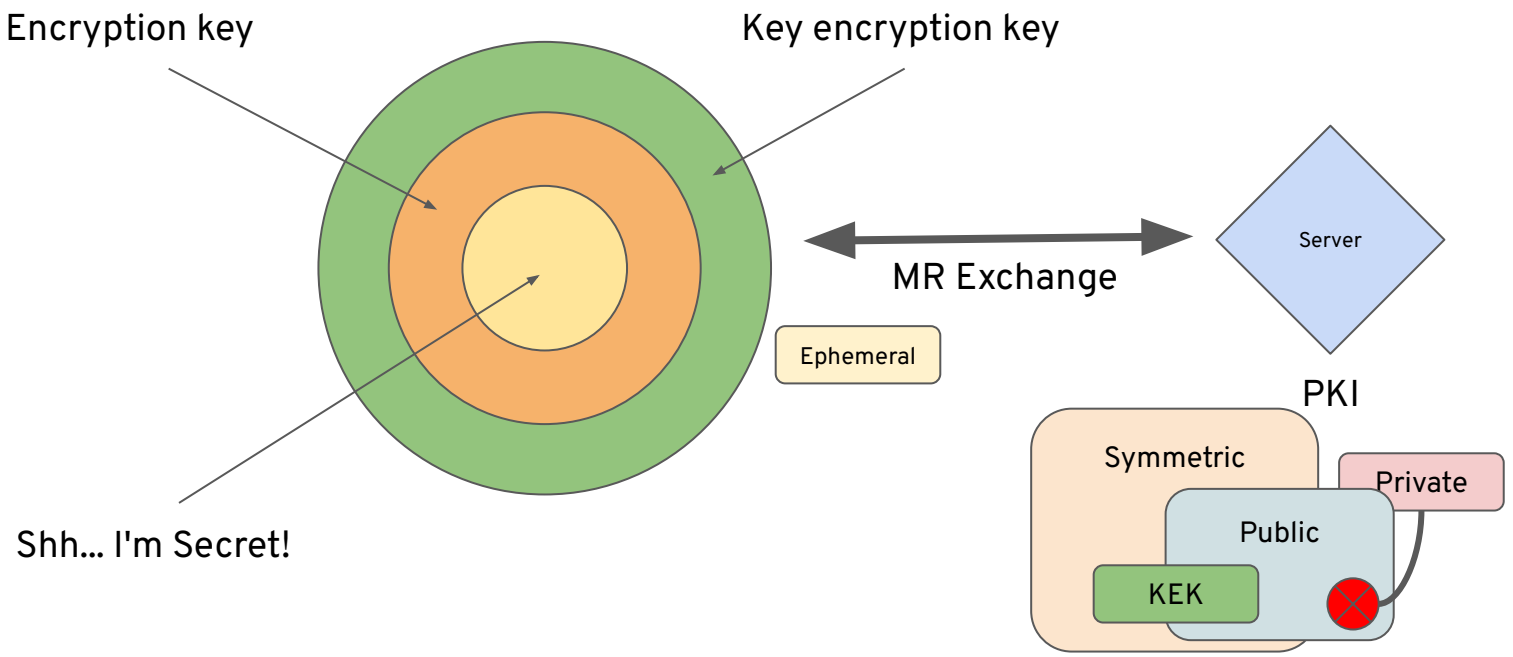
Network Bound Disk Encryption



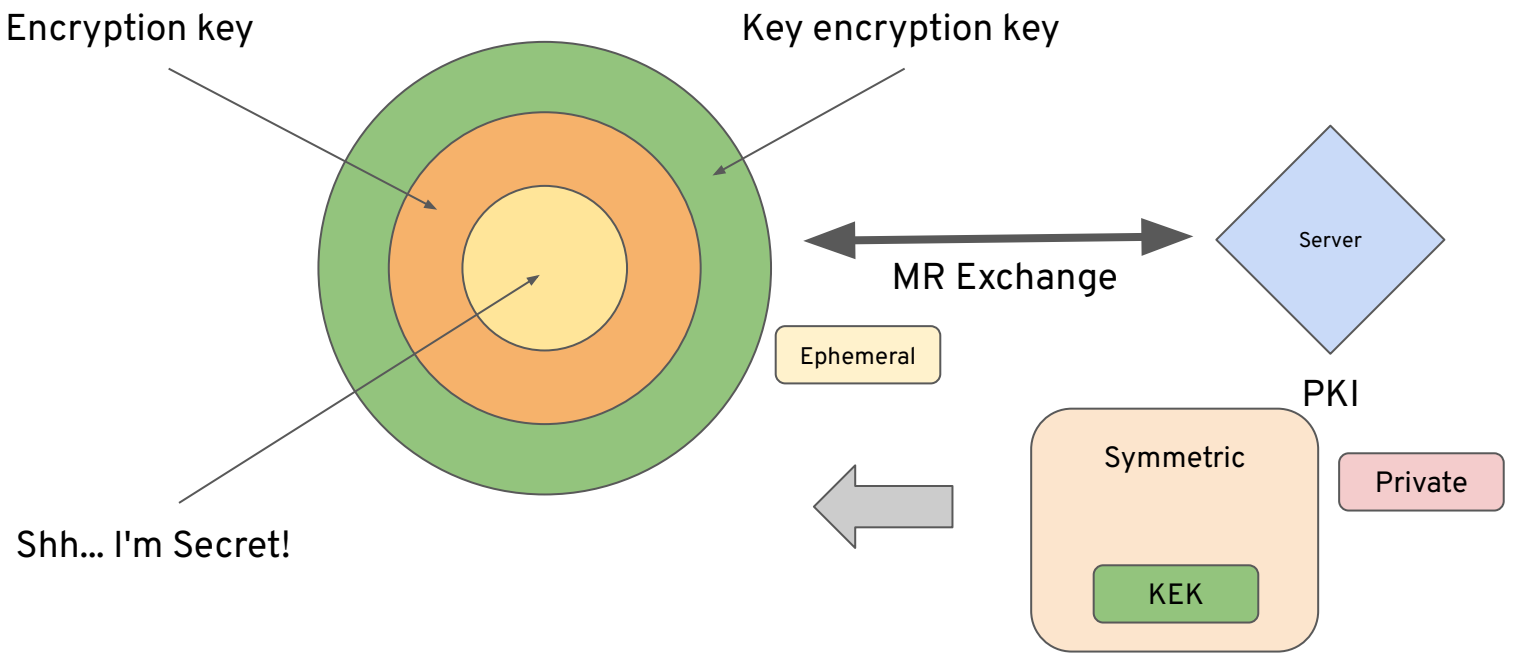
Network Bound Disk Encryption



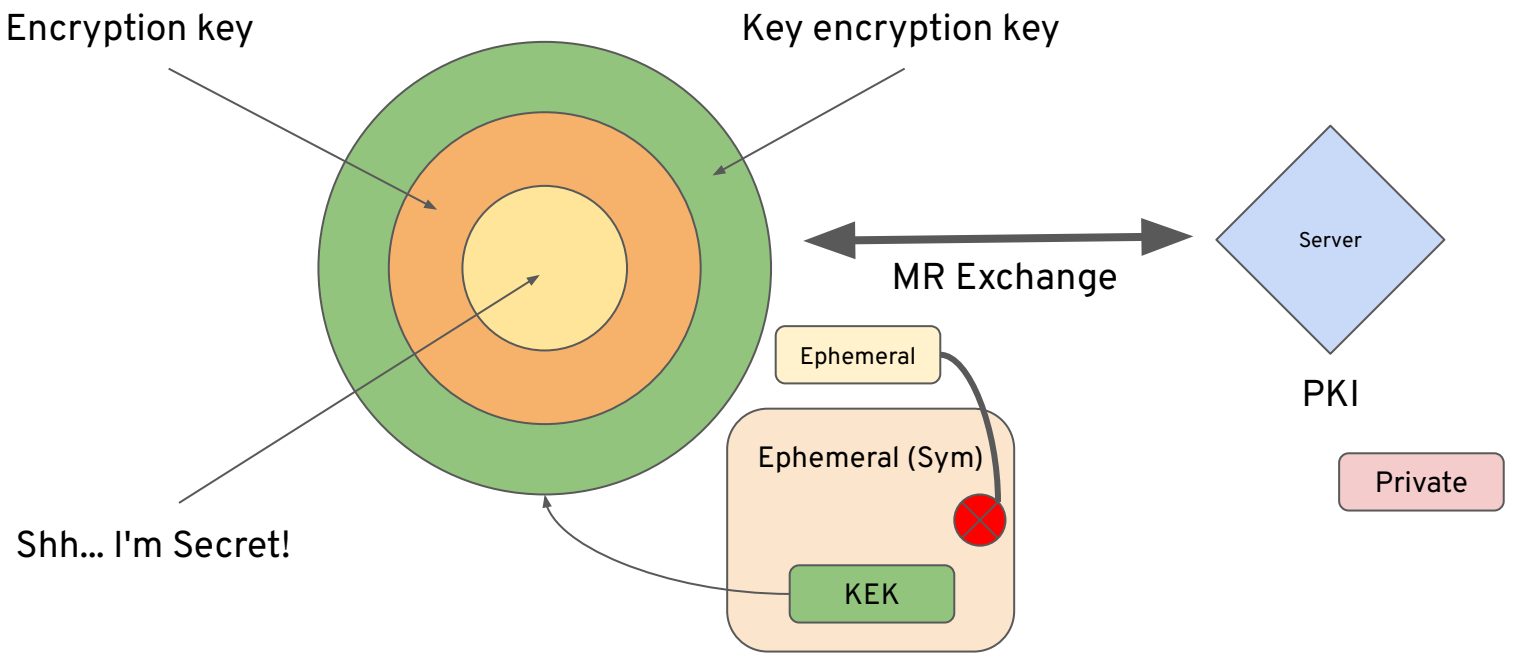
Network Bound Disk Encryption



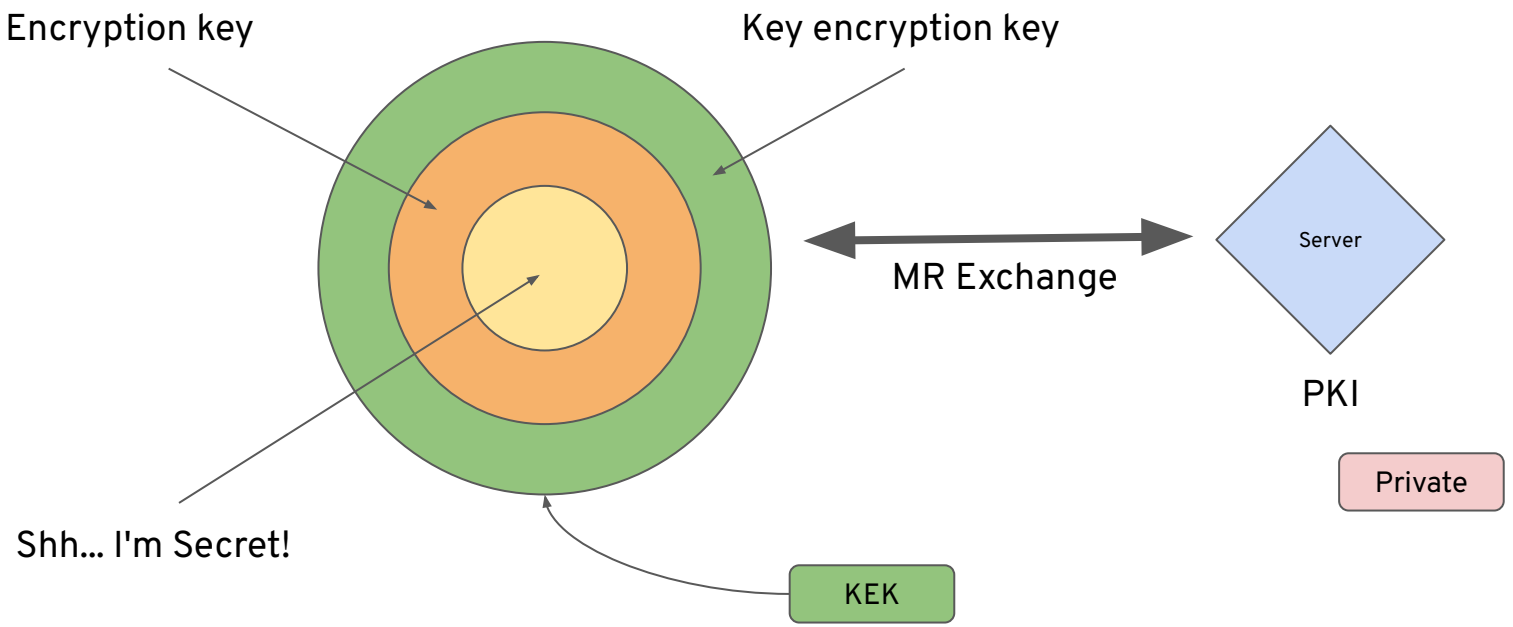
Network Bound Disk Encryption



Network Bound Disk Encryption



Network Bound Disk Encryption



Network Bound Disk Encryption

Property	Escrow	MR Key Exchange
Server presence during provisioning	Required	Optional
Server presence during recovery	Required	Required
Server knowledge of keys	Required	None
Key transfer	Required	None
Client authentication	Required	Optional
Transport encryption	Required	Optional
End-to-end Encryption	Difficult	Unneeded