redhat.

OPENSHIFT TECHNICAL OVERVIEW

The application behaves differently in production than it did in test.

I don't have enough environments to perform testing.

It takes weeks for an environment to get delivered.

We have no idea what is the current state of machines.

Deployments are manual, painful, and infrequent.

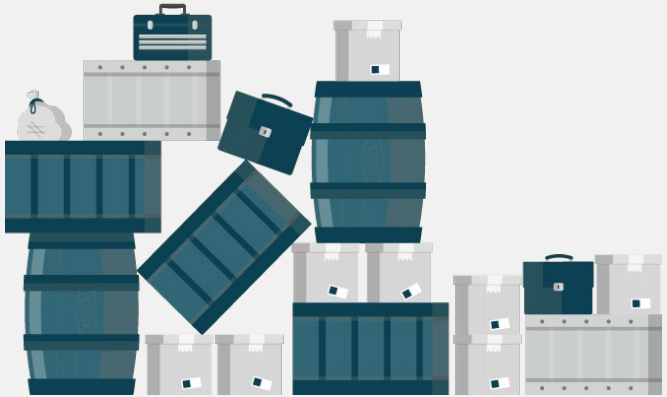
Each environment has a different set of configurations to manage.

We have a proliferation of technologies we have to manage.

THE PROBLEM

Applications require complicated installation and integration every time they are deployed leading to

- Slow service delivery
- Reduced service quality
- Frequent down times



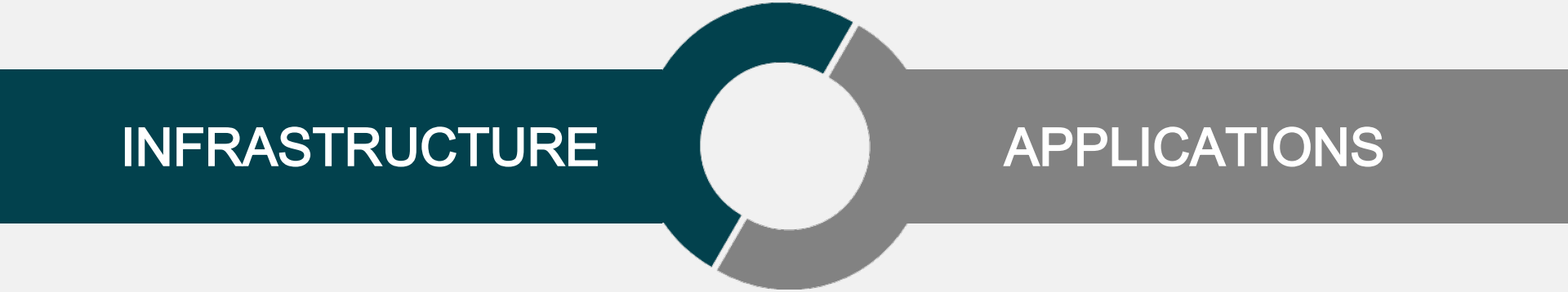
THE SOLUTION

Adopting a container strategy will allow applications to be easily shared and deployed

- Consistent environment and tools
- Predictable building blocks
- Faster deployment

WHAT ARE CONTAINERS?

It Depends Who You Ask



- Sandboxed application processes on a shared Linux OS kernel **(multi-tenancy)**
- Simpler, lighter, and denser than virtual machines **(density)**
- Portable across different environments **(portability)**

- Package my application and all of its dependencies **(encapsulation)**
- Deploy to any environment in seconds and enable CI/CD **(ephemerality)**
- Easily access and share containerized components **(standardization)**

WE NEED MORE THAN JUST CONTAINERS

Scheduling
Decide where to deploy containers

Lifecycle and health
Keep containers running despite failures

Discovery
Find other containers on the network

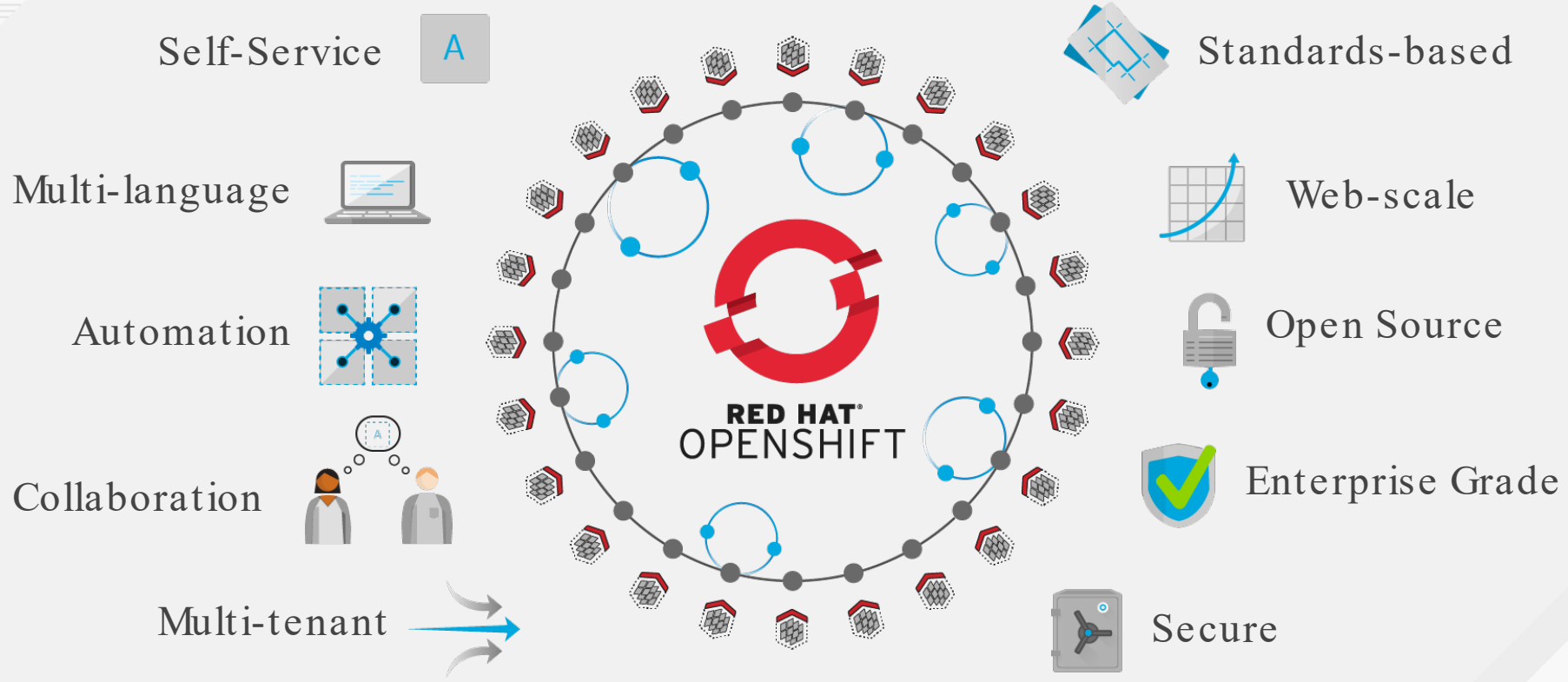
Monitoring
Visibility into running containers

Security
Control who can do what

Scaling
Scale containers up and down

Persistence
Survive data beyond container lifecycle

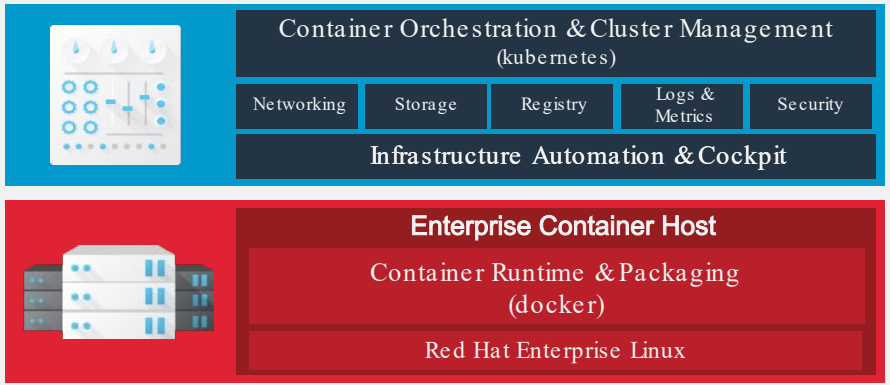
Aggregation
Compose apps from multiple containers


Self-Service
Multi-language
Automation
Collaboration
Multi-tenant
RED HAT®
OPENSHIFT
Standards-based
Web-scale
Open Source
Enterprise Grade
Secure

OPENSHIFT CONTAINER PLATFORM

Trusted by Fortune Global 500 companies

OPENSHIFT CONTAINER PLATFORM



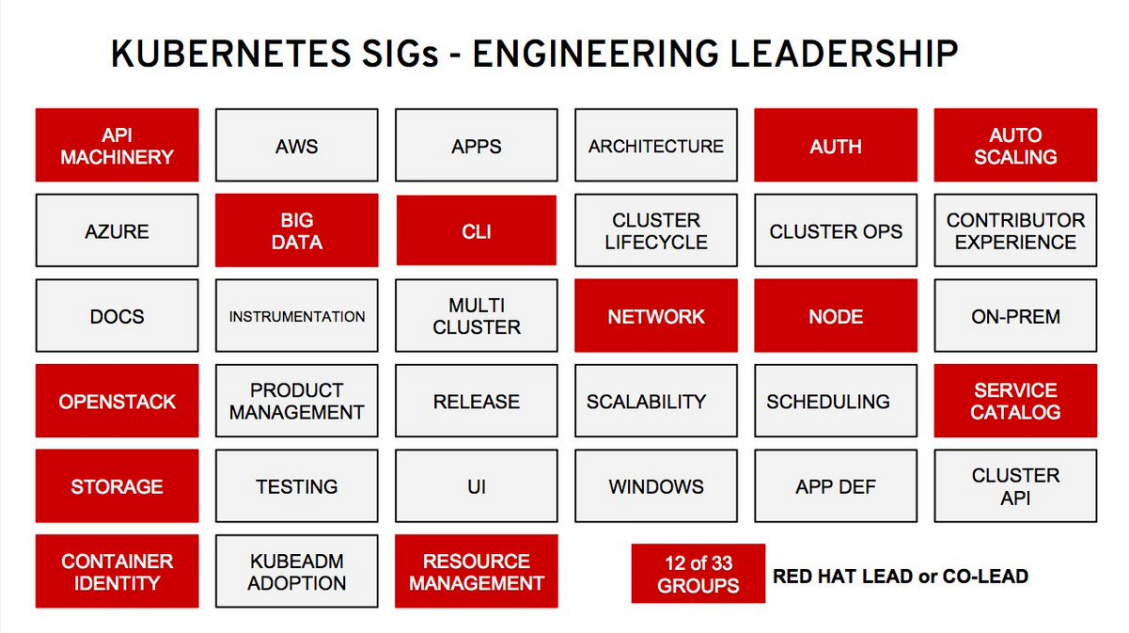
Enterprise Kubernetes++ container orchestration

Trusted by Fortune Global 500 companies

Kubernetes is an open-source system for automating deployment, operations, and scaling of containerized applications across multiple hosts

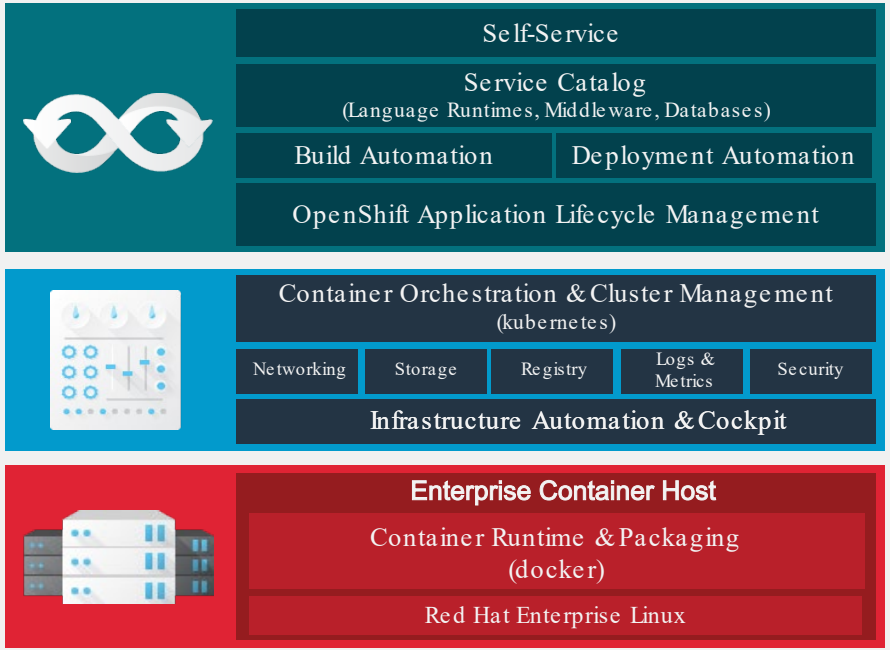
kubernetes

RED HAT LEADERSHIP IN KUBERNETES



RED HAT LEADERSHIP IN KUBERNETES

OPENSHIFT CONTAINER PLATFORM



Developer Experience

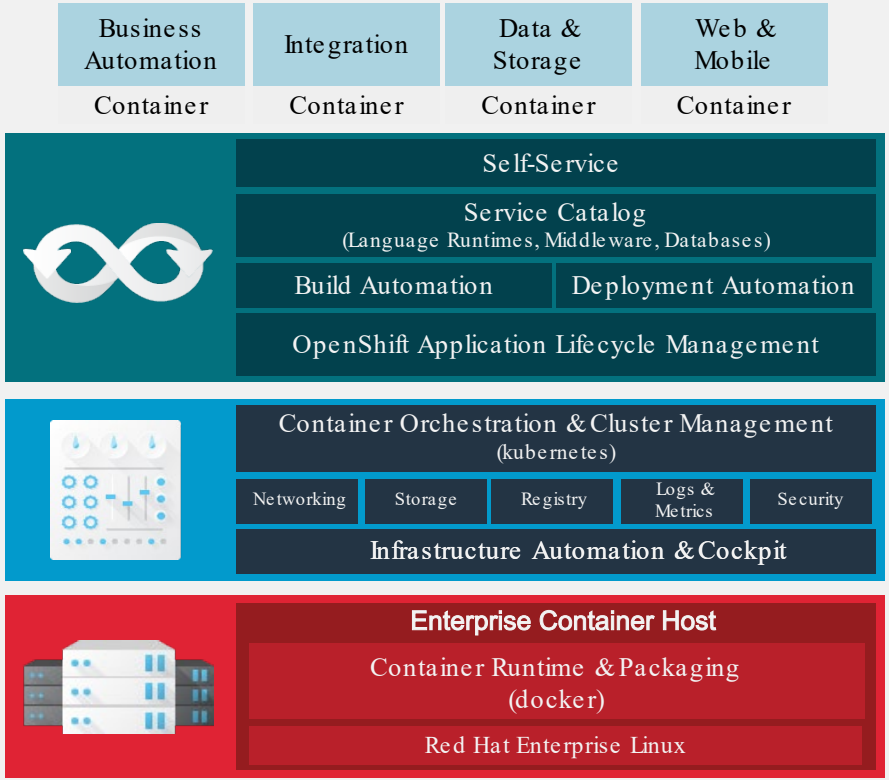
Enterprise Kubernetes++ container orchestration

Trusted by Fortune Global 500 companies

WHAT DOES OPENSHIFT PROVIDE OVER KUBERNETES?

FEATURE	KUBERNETES	OPENSHIFT CONTAINER PLATFORM
Multi-host container scheduling	✔	✔
Self-service provisioning	✔	✔
Service-discovery	✔	✔
Persistent storage	✔	✔
Multi-tenancy	✖	✔
Collaboration	✖	✔
Networking	✖	✔
Image registry	✖	✔
Monitoring	✖	✔
Log aggregation	✖	✔
CI/CD and DevOps	✖	✔
Certified application services (databases, runtimes, ...)	✖	✔
Certified middleware services	✖	✔
Built-in operational management	✖	✔

OPENSHIFT CONTAINER PLATFORM

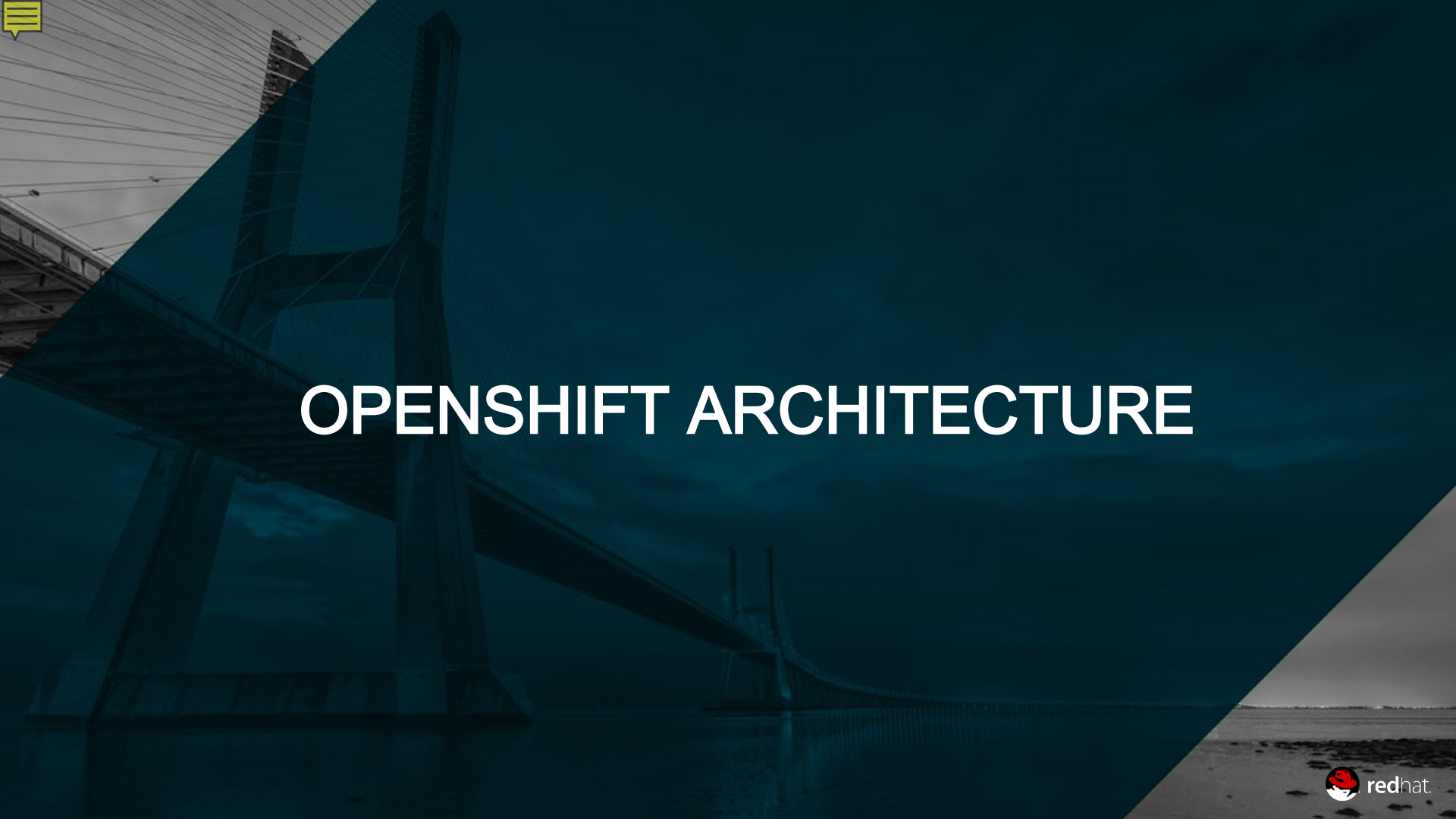


Traditional, stateful, and cloud-native apps

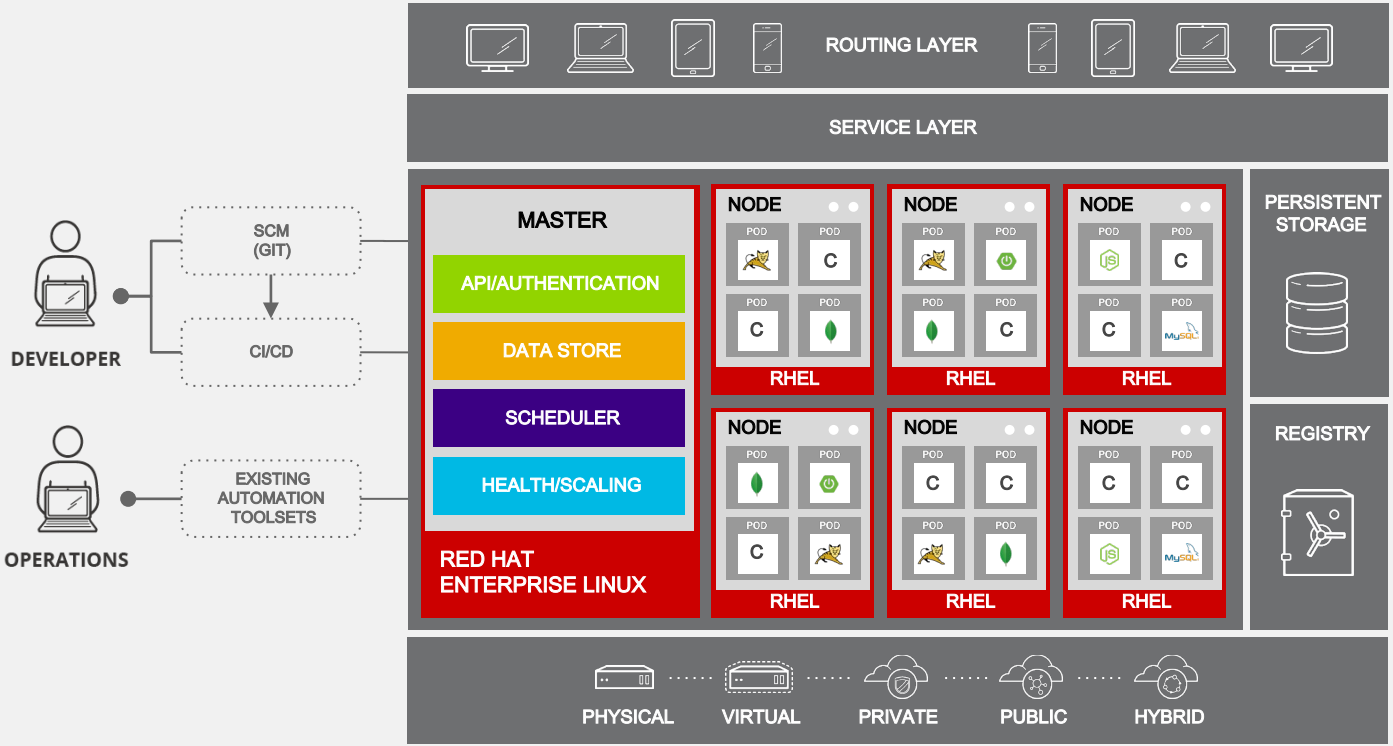
Developer Experience

Enterprise Kubernetes++ container orchestration

Trusted by Fortune Global 500 companies


OPENSHIFT ARCHITECTURE
redhat.

OPENSHIFT ARCHITECTURE

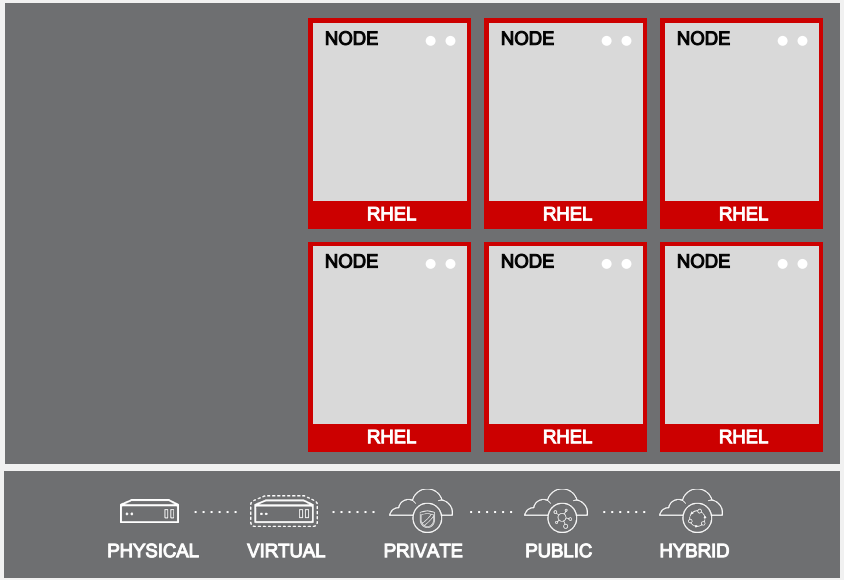


redhat.

YOUR CHOICE OF INFRASTRUCTURE

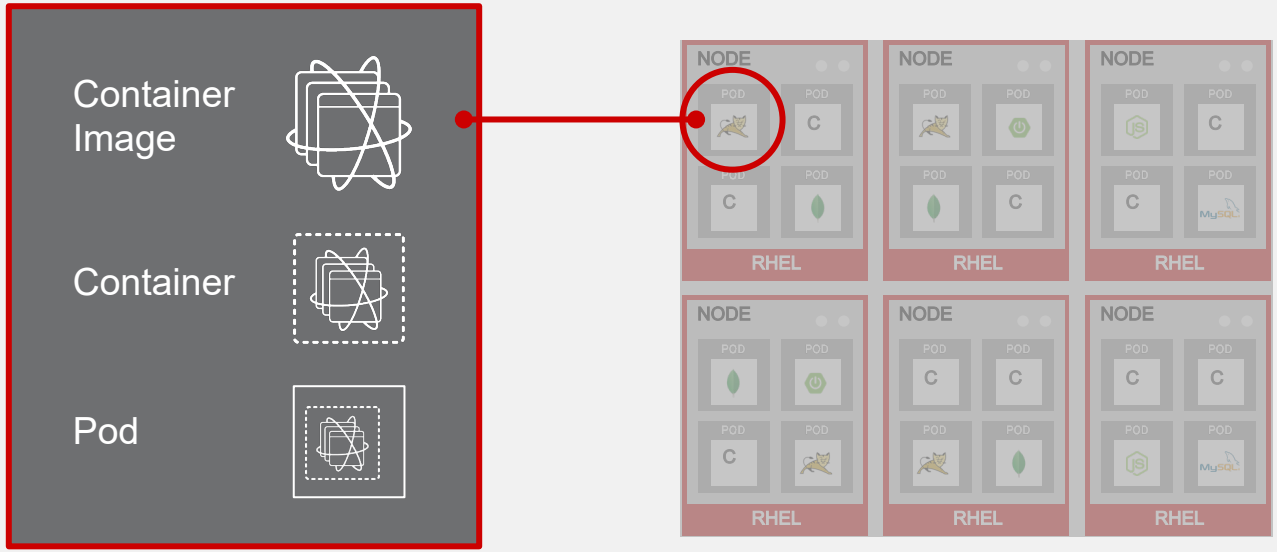
redhat.

NODES RHEL INSTANCES WHERE APPS RUN

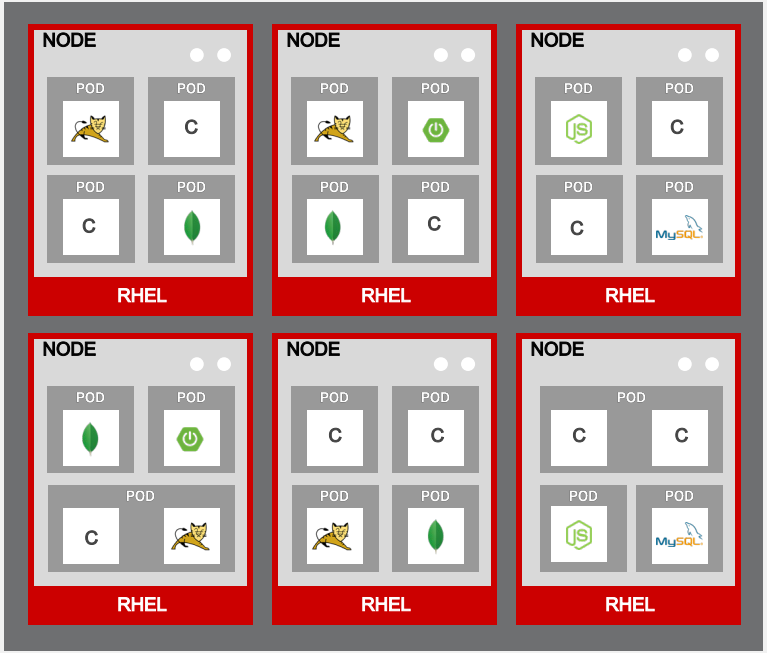


redhat.

APPS RUN IN CONTAINERS

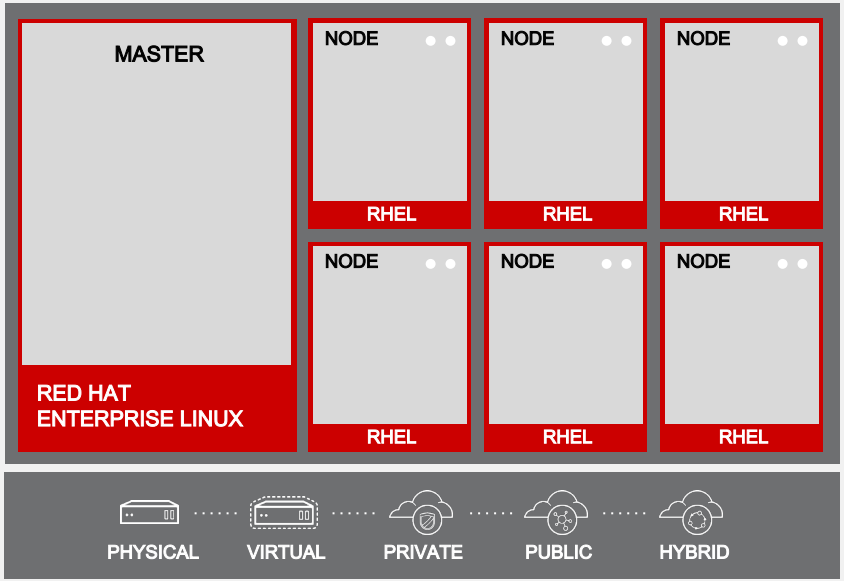


PODS ARE THE UNIT OF ORCHESTRATION



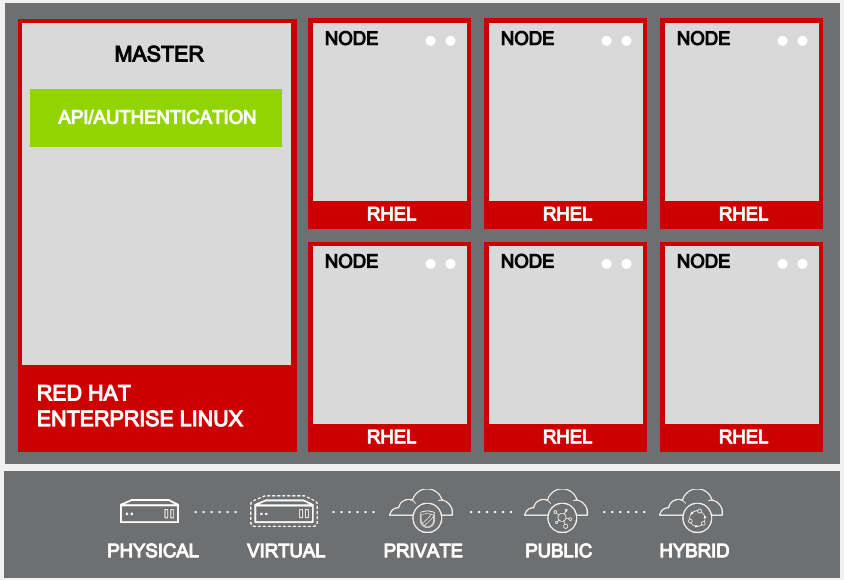
redhat.

MASTERS ARE THE CONTROL PLANE



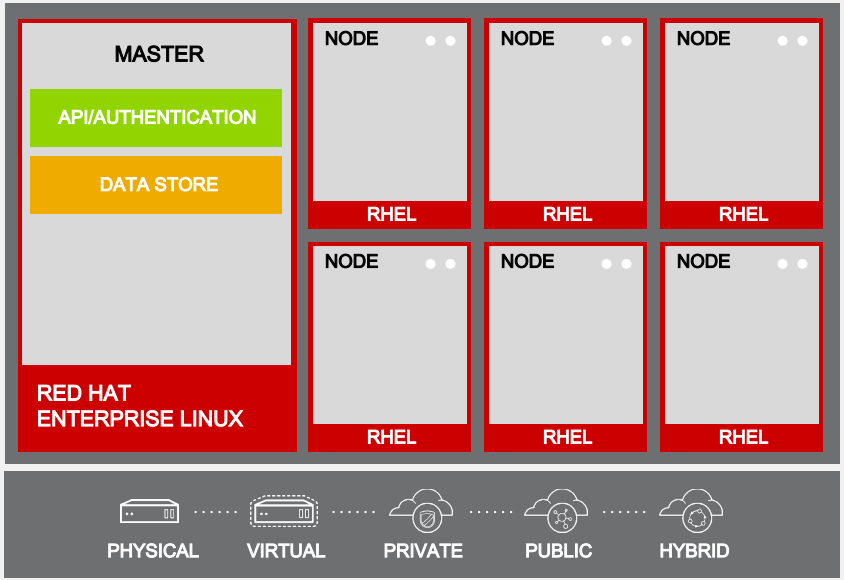
redhat.

API AND AUTHENTICATION

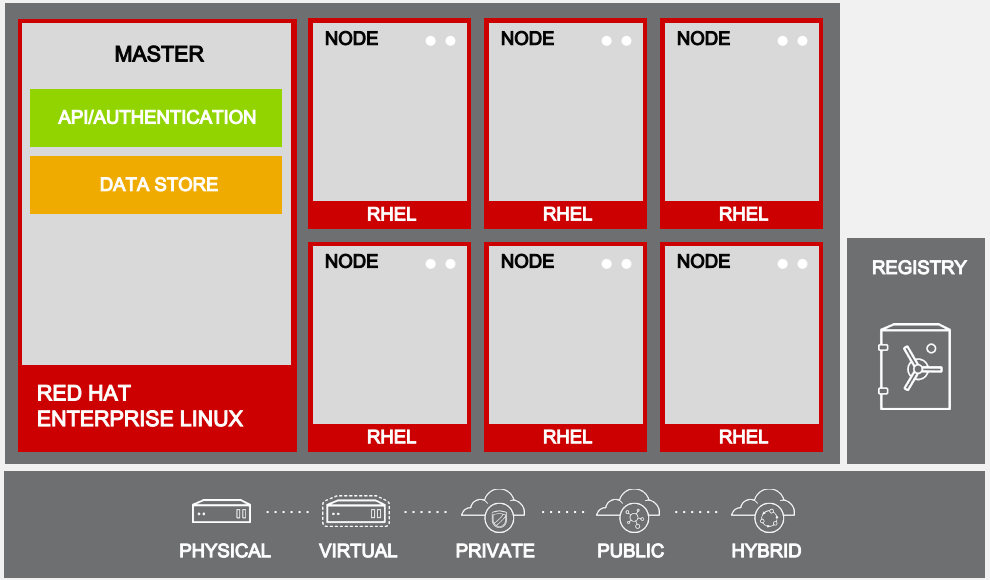


redhat.

DESIRED AND CURRENT STATE

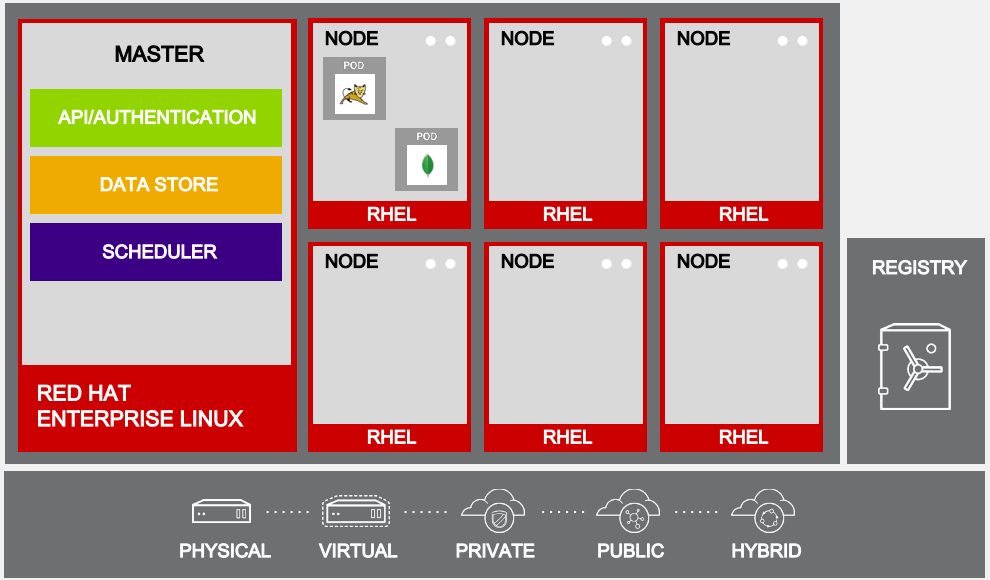


INTEGRATED CONTAINER REGISTRY



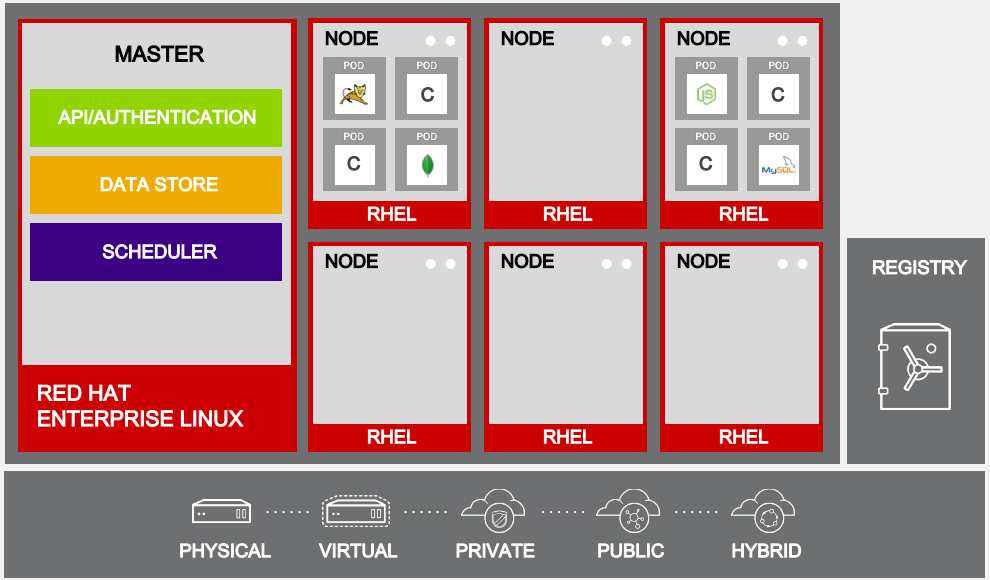
redhat.

ORCHESTRATION AND SCHEDULING



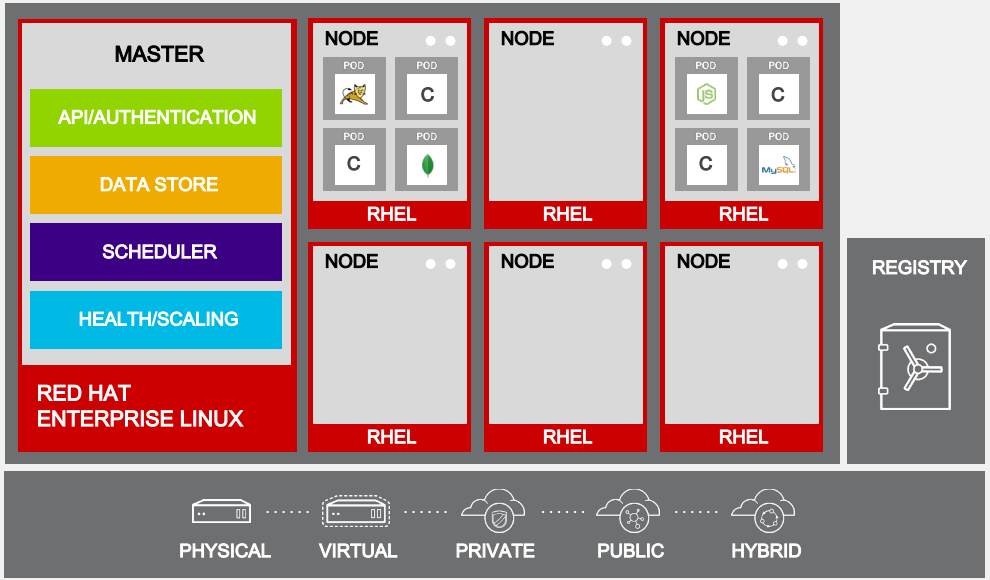
redhat.

PLACEMENT BY POLICY



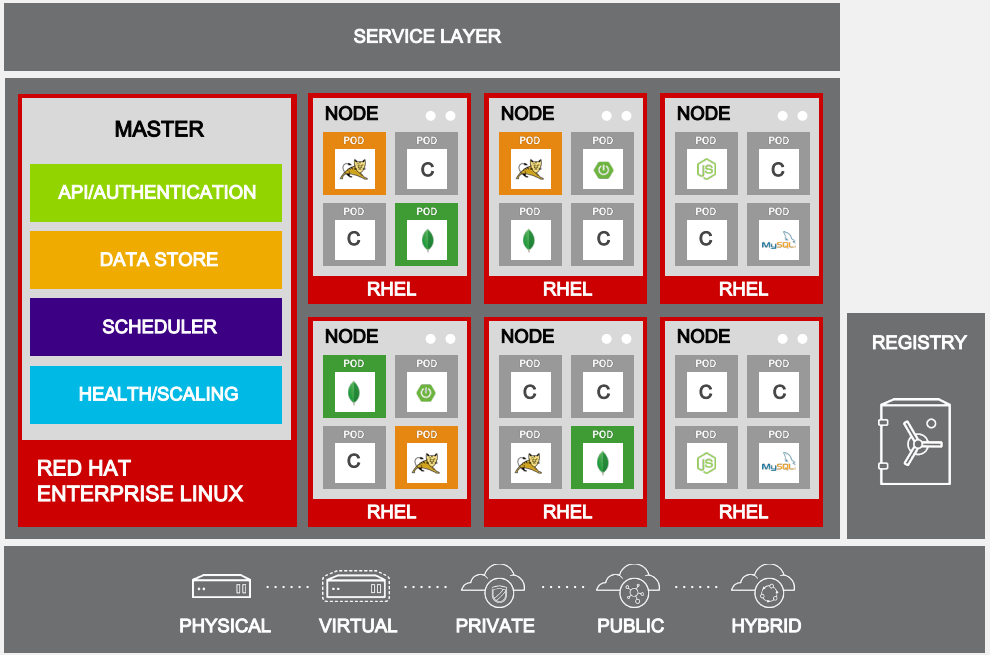
redhat.

AUTOSCALING PODS

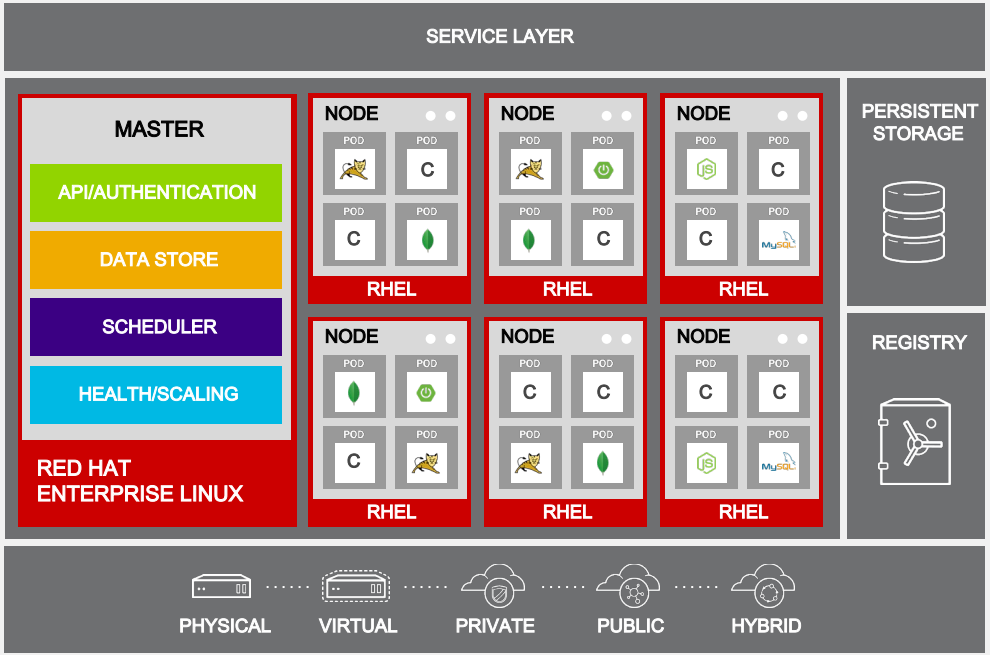


redhat.

SERVICE DISCOVERY

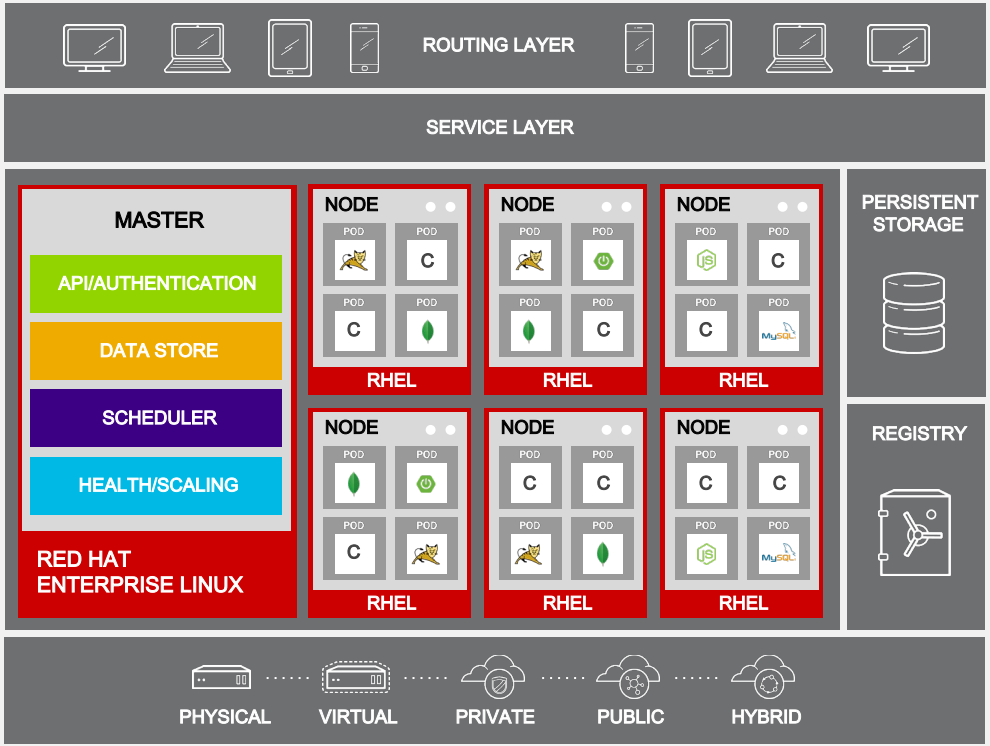


PERSISTENT DATA IN CONTAINERS



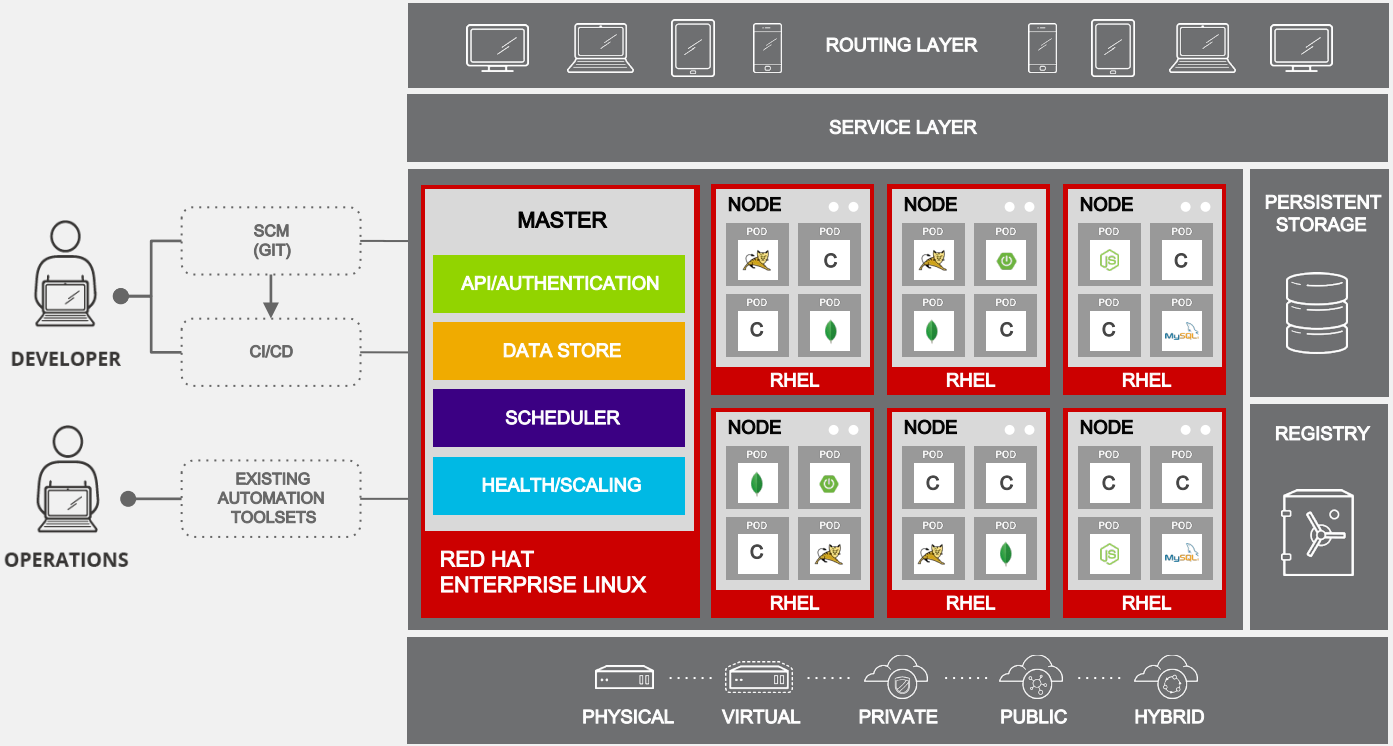
redhat.

ROUTING AND LOAD-BALANCING



redhat.

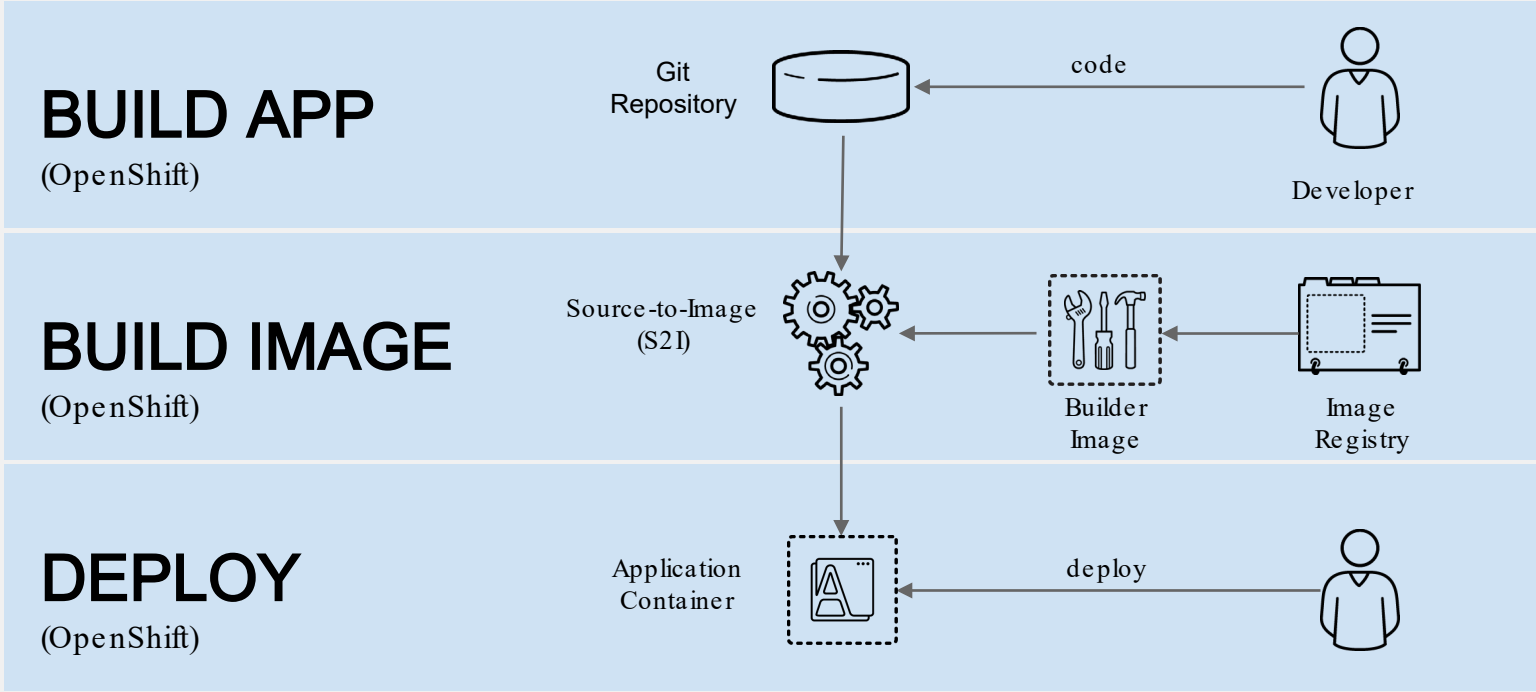
ACCESS VIA WEB, CLI, IDE AND API



redhat.

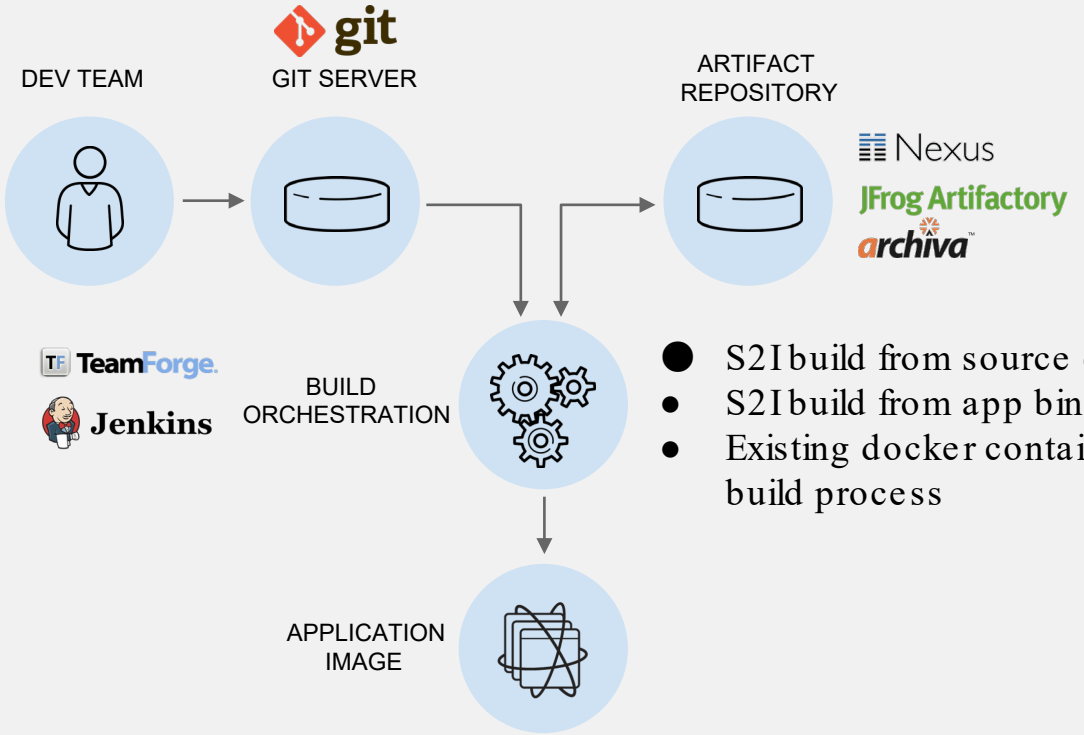
BUILD AND DEPLOY APPLICATIONS

DEPLOY SOURCE CODE WITH SOURCE-TO-IMAGE (S2I)



redhat.

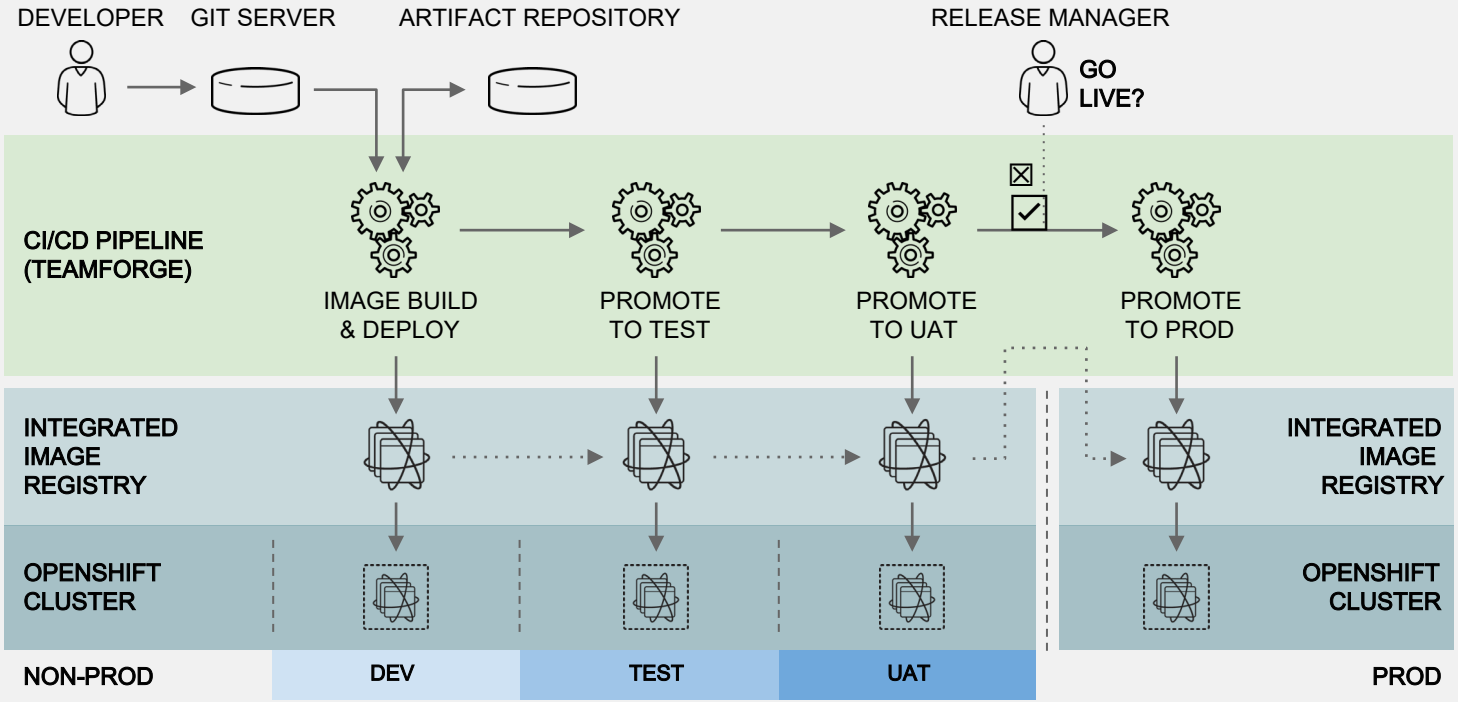
CONTINUOUS DELIVERY PIPELINE



- S2I build from source code
- S2I build from app binary
- Existing docker container image build process

redhat.

CONTINUOUS DELIVERY PIPELINE

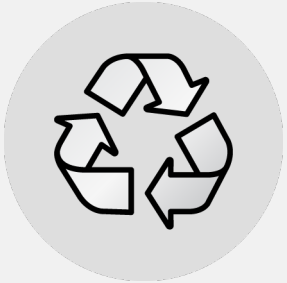


redhat.

OPERATIONS MANAGEMENT

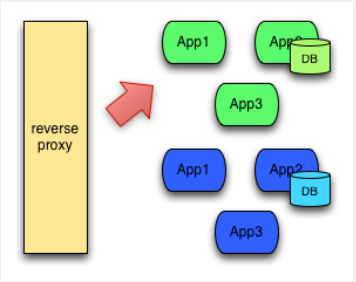
DEPLOYMENTS WITH OPENSHIFT

Painless deployments with zero/reduced downtime through automation



ROLLING DEPLOYMENTS

A rolling deployment slowly replaces instances of the previous version of an application with instances of the new version of the application.



BLUE/GREEN DEPLOYMENTS

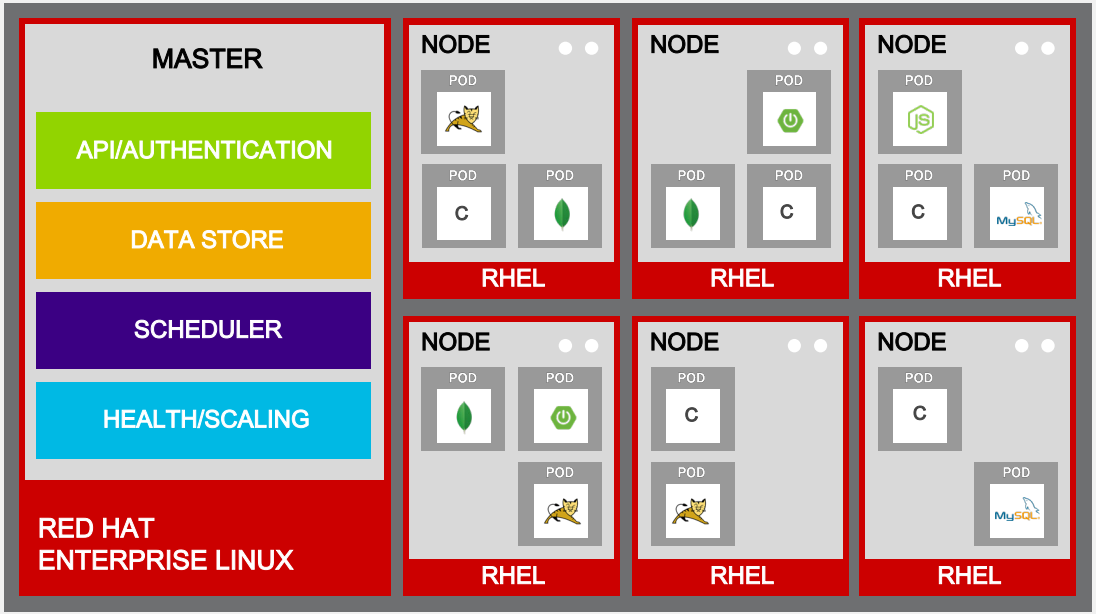
A blue/green deployment is a software deployment strategy that relies on two identical production configurations that alternate between active and inactive.

A/B DEPLOYMENTS

A/B testing (sometimes called split testing) is comparing two versions of a web page to see which one performs better.

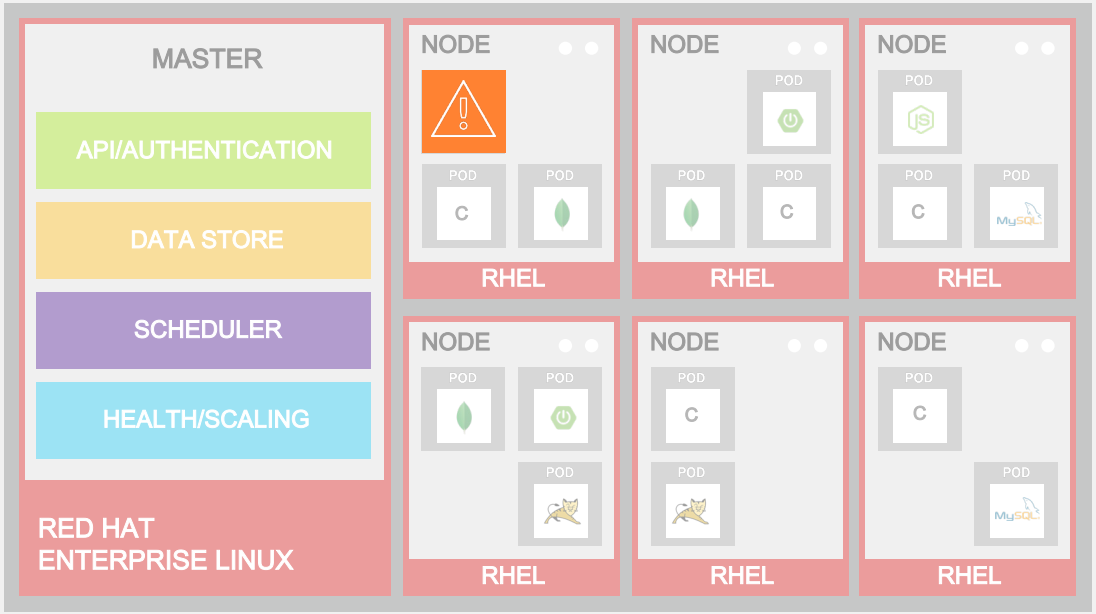
redhat.

AUTO-HEALING FAILED PODS



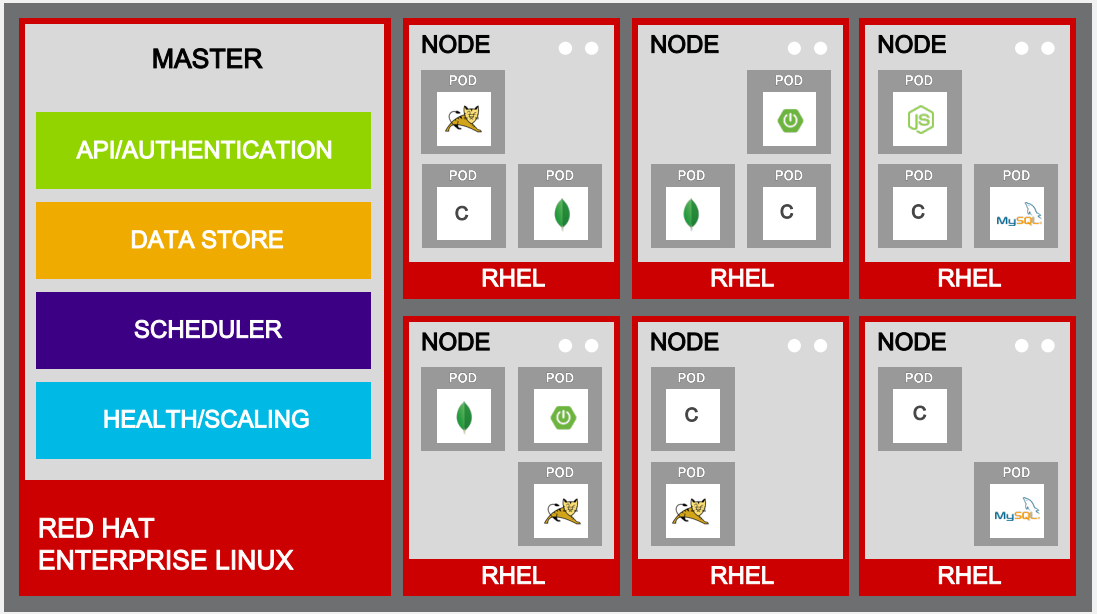
redhat.

AUTO-HEALING FAILED PODS

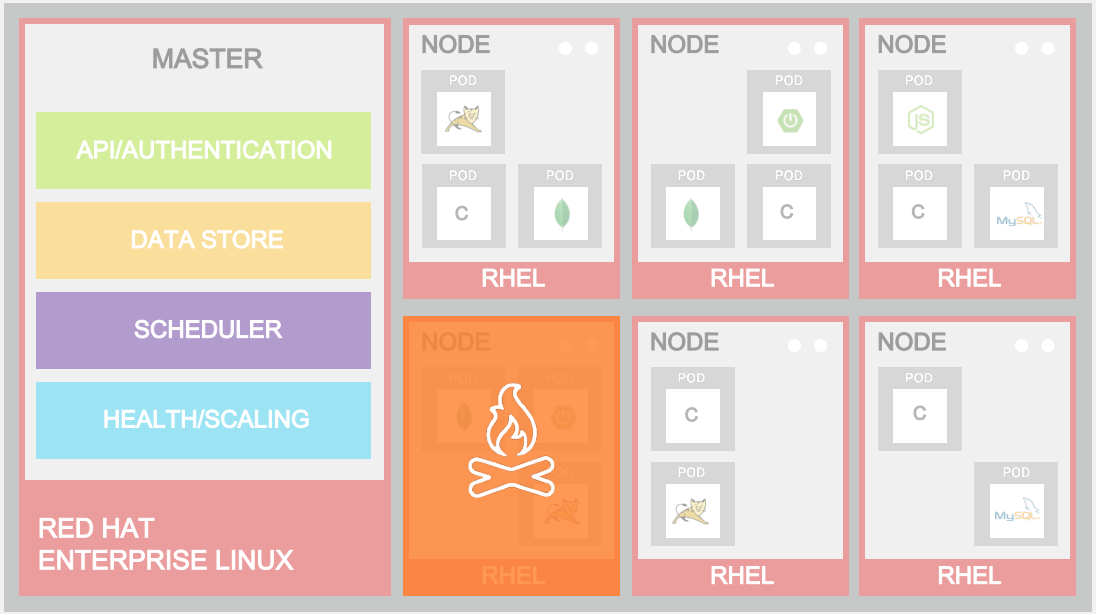


redhat.

AUTO-HEALING FAILED PODS

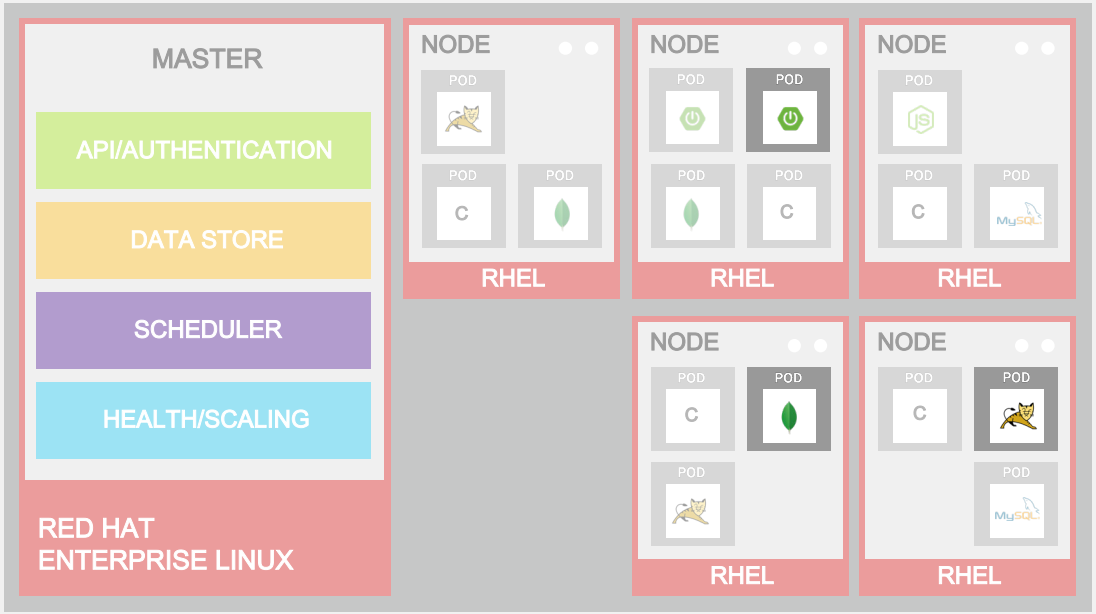


AUTO-HEALING FAILED PODS



redhat.

AUTO-HEALING FAILED PODS

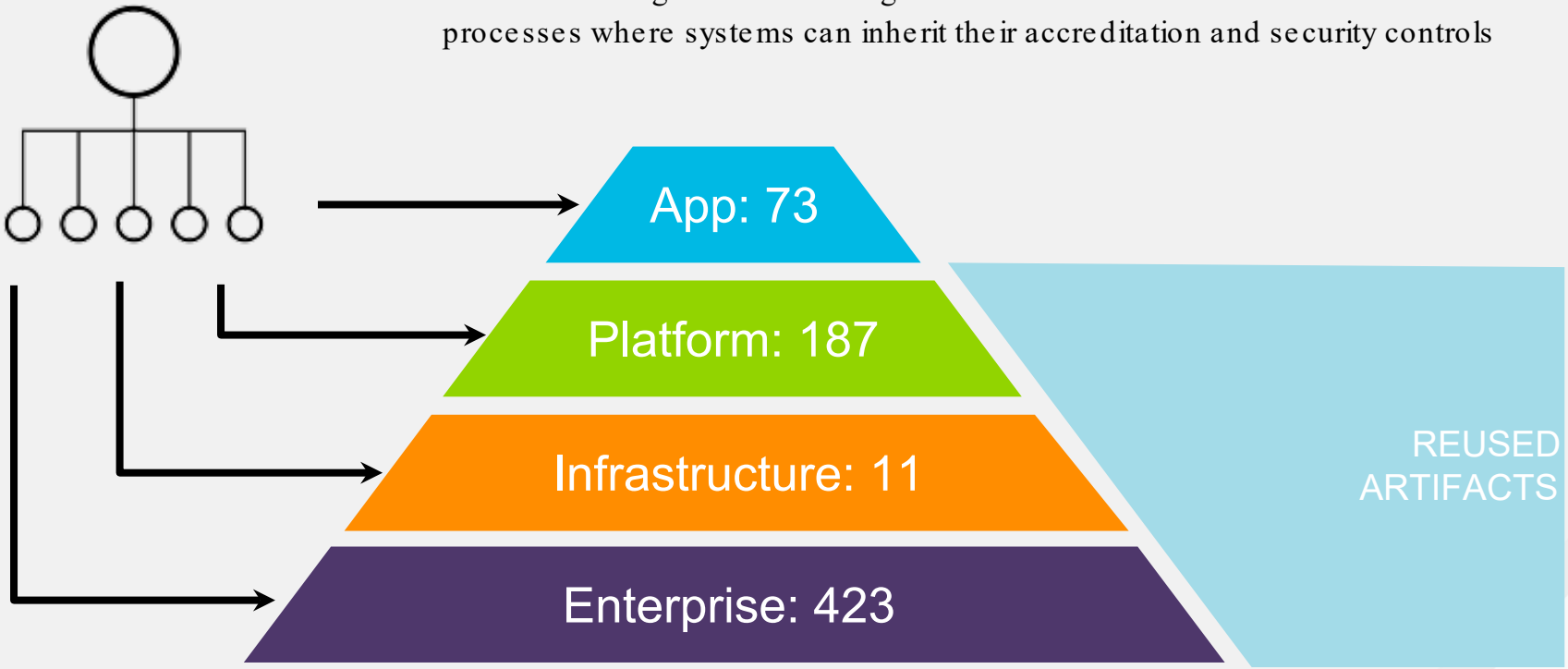


redhat.

SECURITY

SECURITY INHERITANCE MODEL

Cloud technologies are heralding in a new era of landlord-tenant accreditation processes where systems can inherit their accreditation and security controls



SECDEVOPS: APPLICATION REPO

Applications can be created from source code repo or pulled from a trusted artifact repo

SCM of choice

- GitHub
- GitLab
- Bitbucket
- Assembla

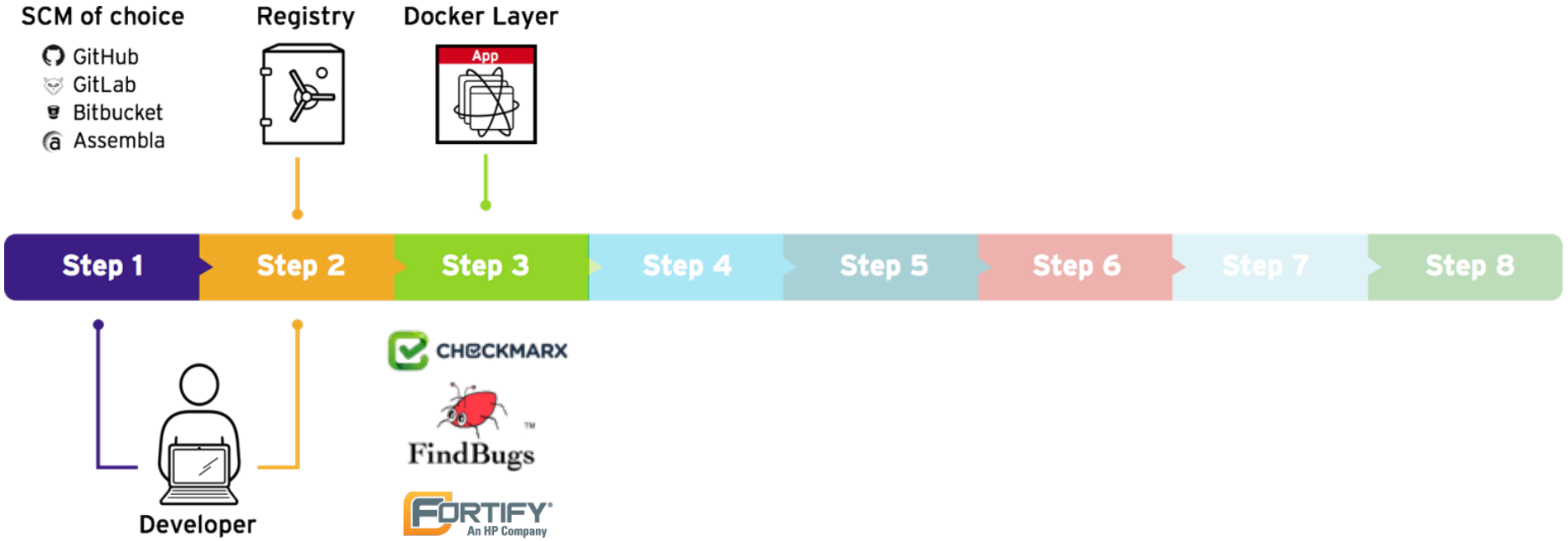
Sonatype Nexus

SECDEVOPS: CONTAINER REGISTRY

Container image is sourced from trusted registry with fine-grained access controls and image signing

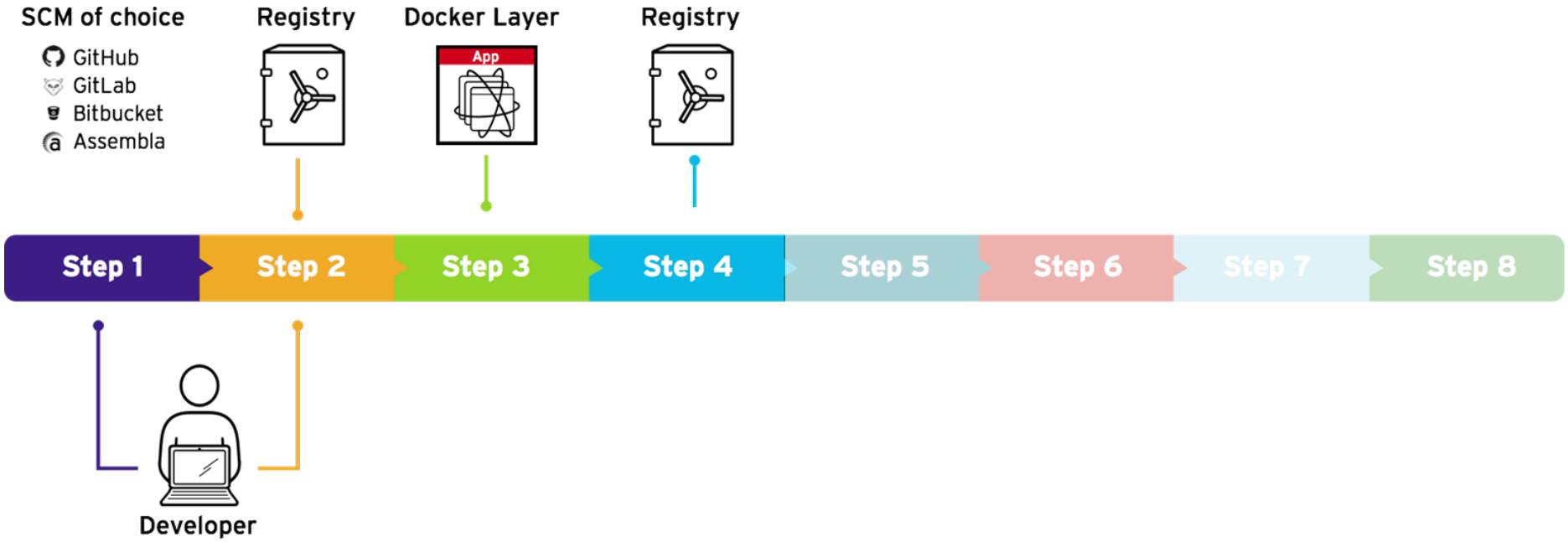
SECDEVOPS: LAYERING

OpenShift builds container using code or artifact + verified baseline in layered paradigm



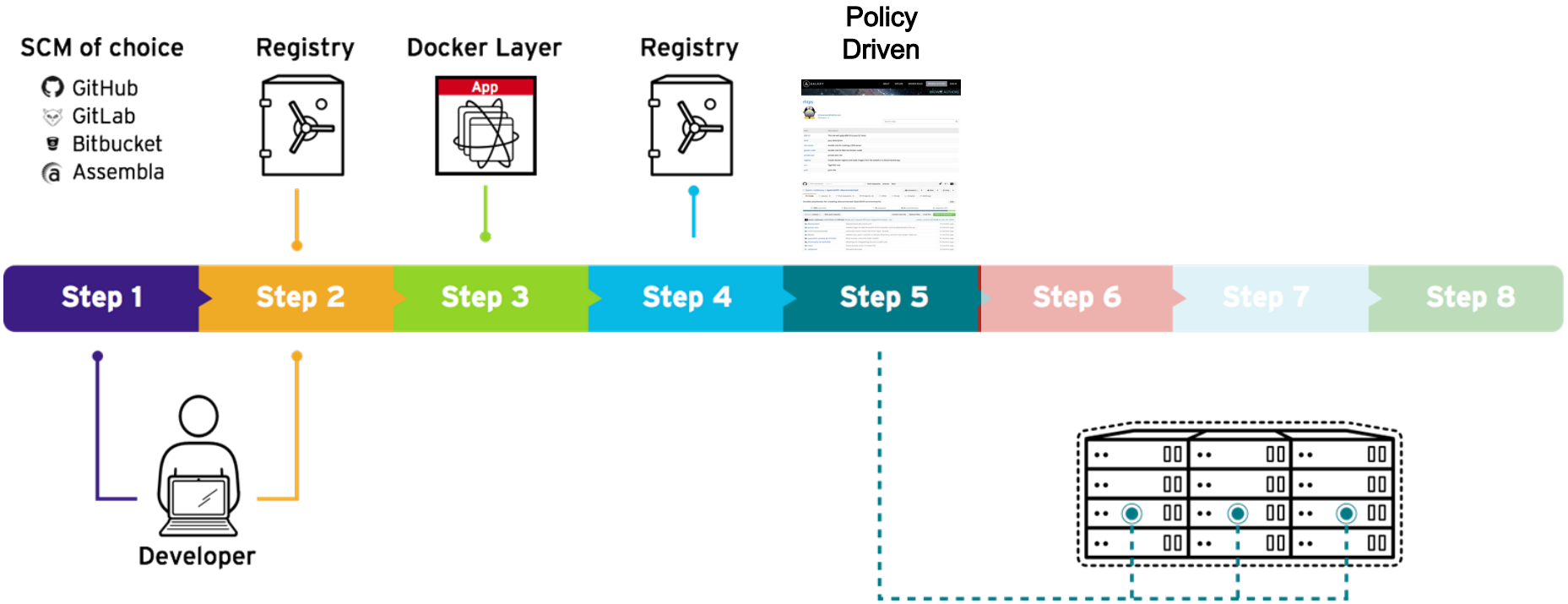
SECDEVOPS: GOLDEN IMAGE

Resulting "golden image" is signed, version, and checked into the trusted registry



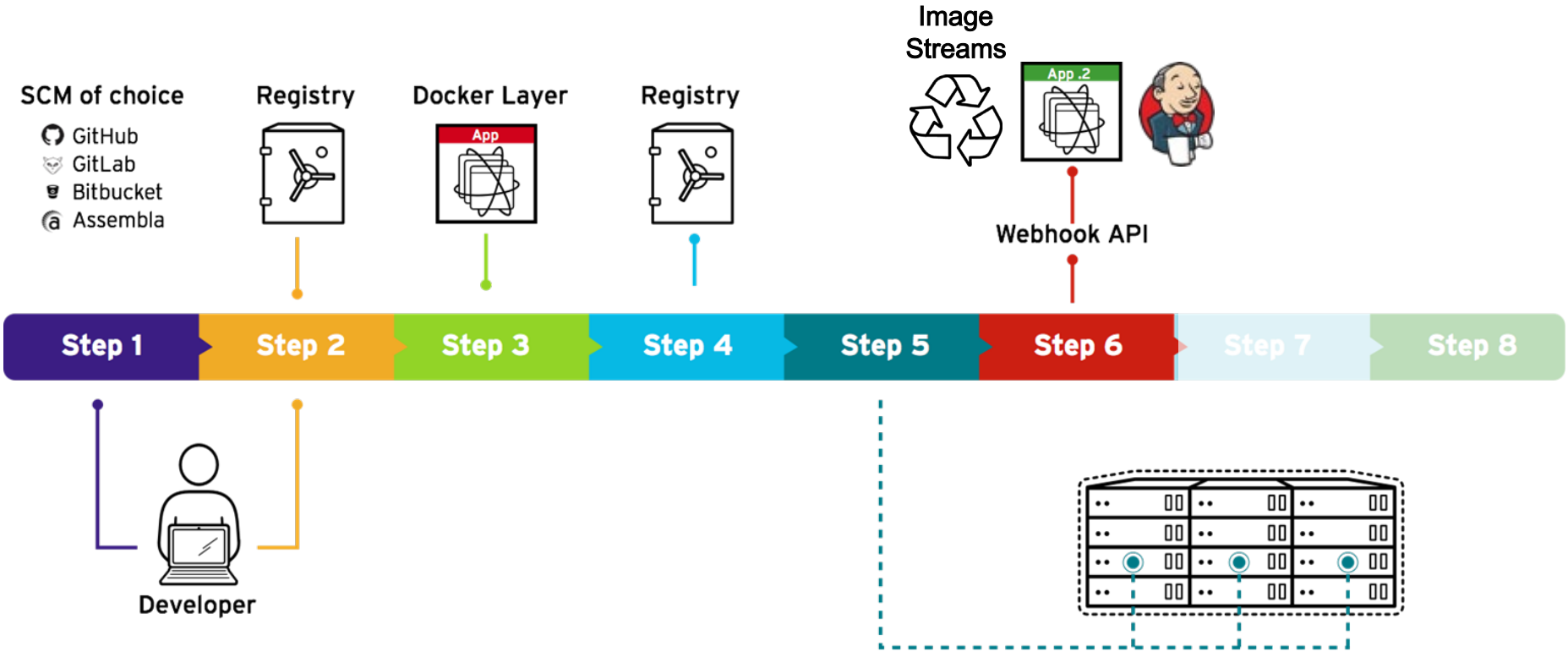
SECDEVOPS: SCHEDULING

Application image(s) are scheduled and deployed using policy-based approach



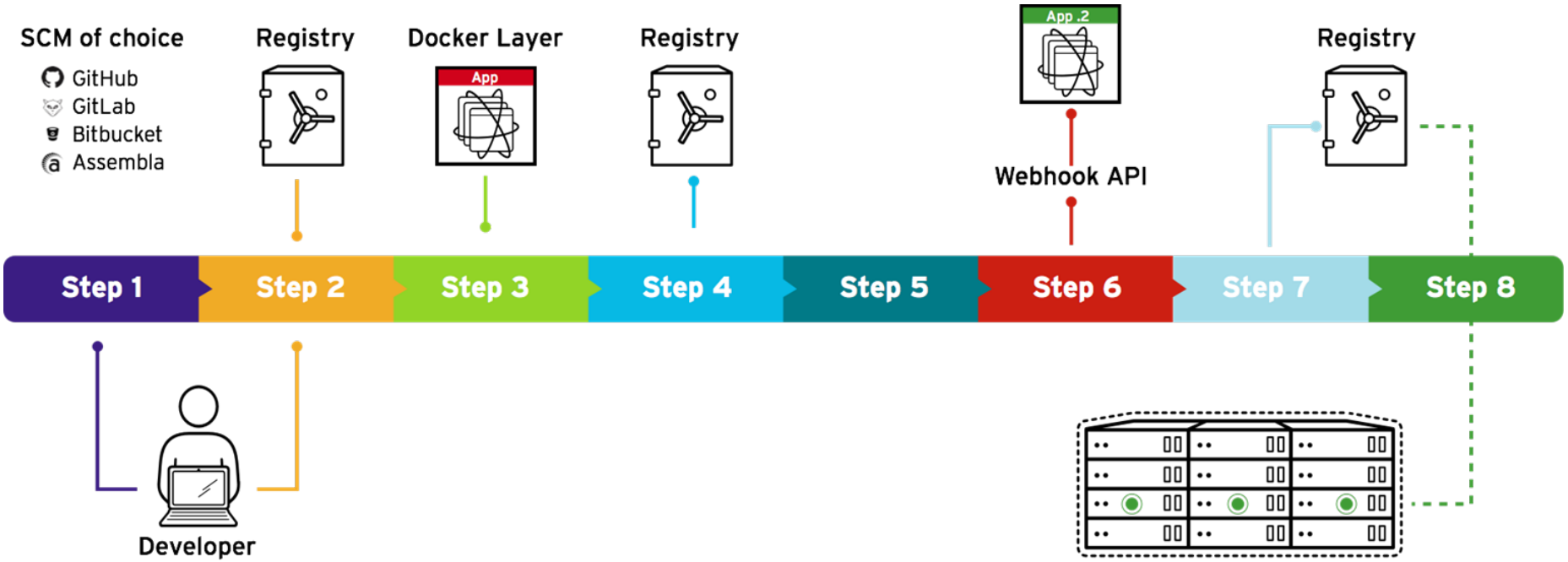
SECDEVOPS: AUTOMATED CHANGES

CI/CD integration, webhooks, or image streams can be used to automate this process



SECDEVOPS: ROLLING UPDATES

New images are signed, versioned, and checked into trusted registry then deployed as rolling updates



redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos