CONFIDENTIAL Designator

OpenShift 4.x Architecture Workshop

Securing Containers
Control, Defend, Extend

July 2019

Red Hat

CONTAINERS CHANGE HOW WE DEVELOP, DEPLOY AND MANAGE APPLICATIONS

INFRASTRUCTURE

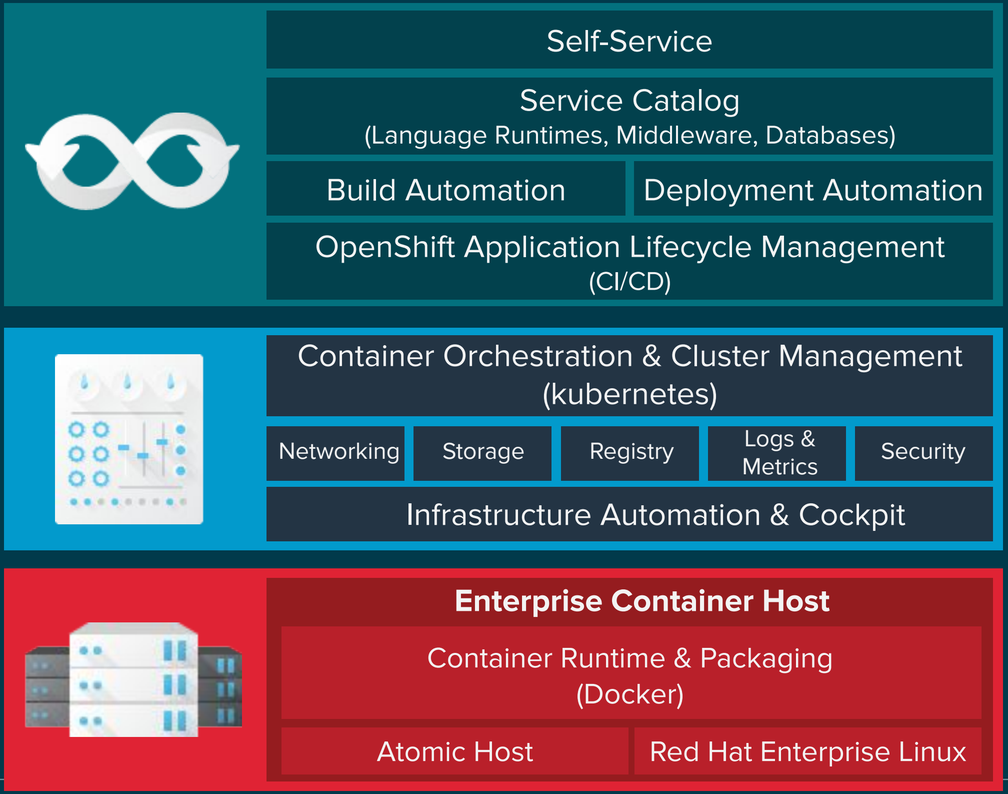
- Sandboxed application processes on a shared Linux OS kernel
- Simpler, lighter, and denser than virtual machines
- Portable across different environments

APPLICATIONS

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

THEY ALSO CHANGE HOW WE SECURE OUR WORKLOADS

ELEMENTS OF AN ENTERPRISE CONTAINER SOLUTION



Self-Service

Service Catalog
(Language Runtimes, Middleware, Databases)

Build Automation

Deployment Automation

OpenShift Application Lifecycle Management
(CI/CD)

Container Orchestration & Cluster Management
(kubernetes)

Networking

Storage

Registry

Logs & Metrics

Security

Infrastructure Automation & Cockpit

Enterprise Container Host

Container Runtime & Packaging
(Docker)

Atomic Host

Red Hat Enterprise Linux

AUTOMATED & INTEGRATED SECURITY

CONTROL

Application Security

- Container Content
- CI/CD Pipeline
- Container Registry
- Deployment Policies

DEFEND

Infrastructure

- Container Platform
- Container Host Multi-tenancy
- Network Isolation
- Storage
- Audit & Logging
- API Management

EXTEND

- Security Ecosystem

CONTROL

Secure the Pipeline & the Applications

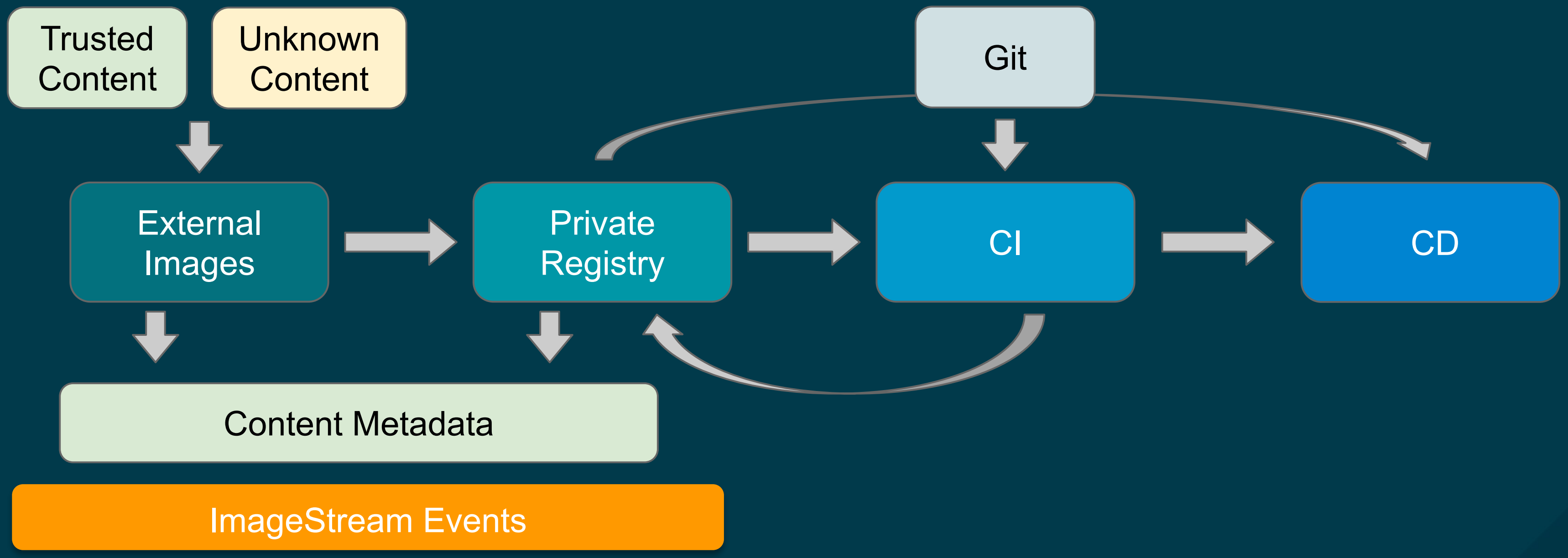
Container Content

CI/CD Pipeline

Container Registry

Deployment Policies

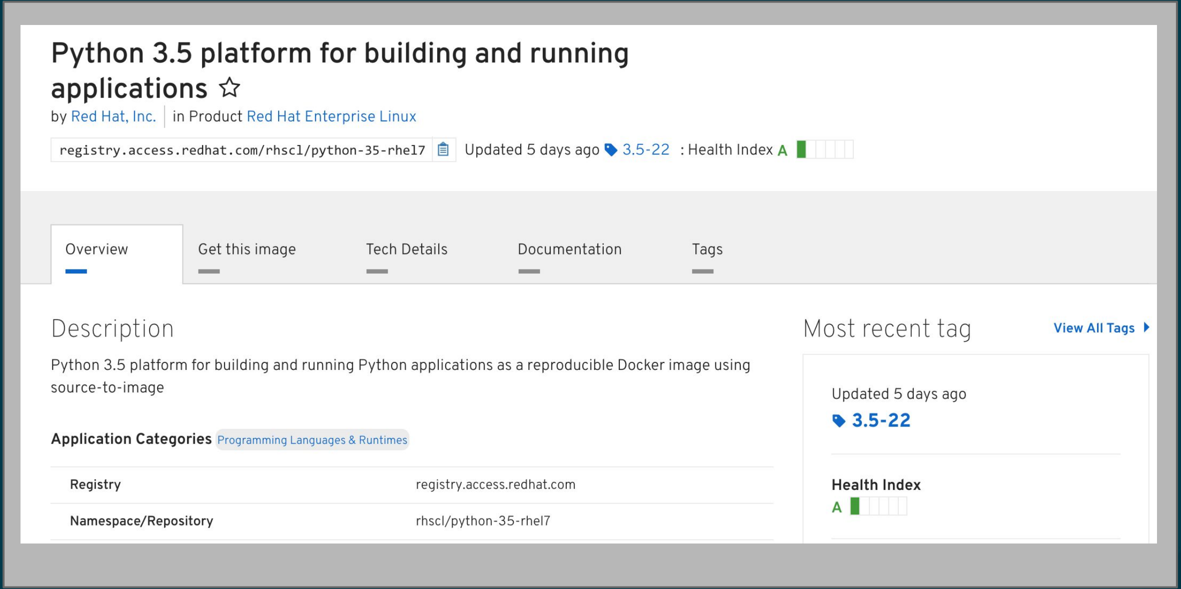
SECURE THE CONTAINER LIFECYCLE



External Images

CONTENT: USE TRUSTED SOURCES

- Are the container images signed?
- Are the runtime and OS layers up to date?
- How frequently will the container be updated and how will I know when it's updated?



Python 3.5 platform for building and running applications ☆

by Red Hat, Inc. | in Product Red Hat Enterprise Linux

`registry.access.redhat.com/rhscl/python-35-rhel7` Updated 5 days ago 3.5-22 : Health Index A

Description

Python 3.5 platform for building and running Python applications as a reproducible Docker image using source-to-image

Application Categories Programming Languages & Runtimes

Registry	registry.access.redhat.com
Namespace/Repository	rhscl/python-35-rhel7

Most recent tag

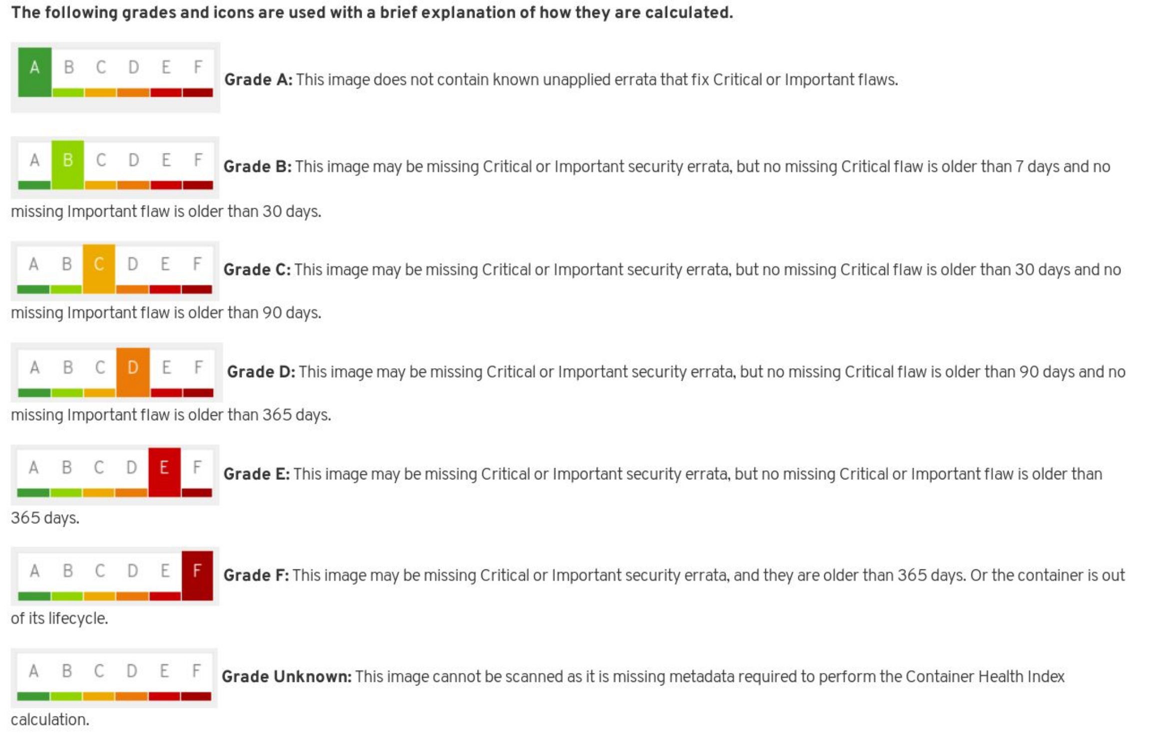
Updated 5 days ago
3.5-22

Health Index
A

Red Hat rebuilds container images when security fixes are released

External Images

CONTENT: CONTAINER HEALTH INDEX



The following grades and icons are used with a brief explanation of how they are calculated.

Grade A: This image does not contain known unapplied errata that fix Critical or Important flaws.

Grade B: This image may be missing Critical or Important security errata, but no missing Critical flaw is older than 7 days and no missing Important flaw is older than 30 days.

Grade C: This image may be missing Critical or Important security errata, but no missing Critical flaw is older than 30 days and no missing Important flaw is older than 90 days.

Grade D: This image may be missing Critical or Important security errata, but no missing Critical flaw is older than 90 days and no missing Important flaw is older than 365 days.

Grade E: This image may be missing Critical or Important security errata, but no missing Critical or Important flaw is older than 365 days.

Grade F: This image may be missing Critical or Important security errata, and they are older than 365 days. Or the container is out of its lifecycle.

Grade Unknown: This image cannot be scanned as it is missing metadata required to perform the Container Health Index calculation.

https://access.redhat.com/articles/2803031

External Images

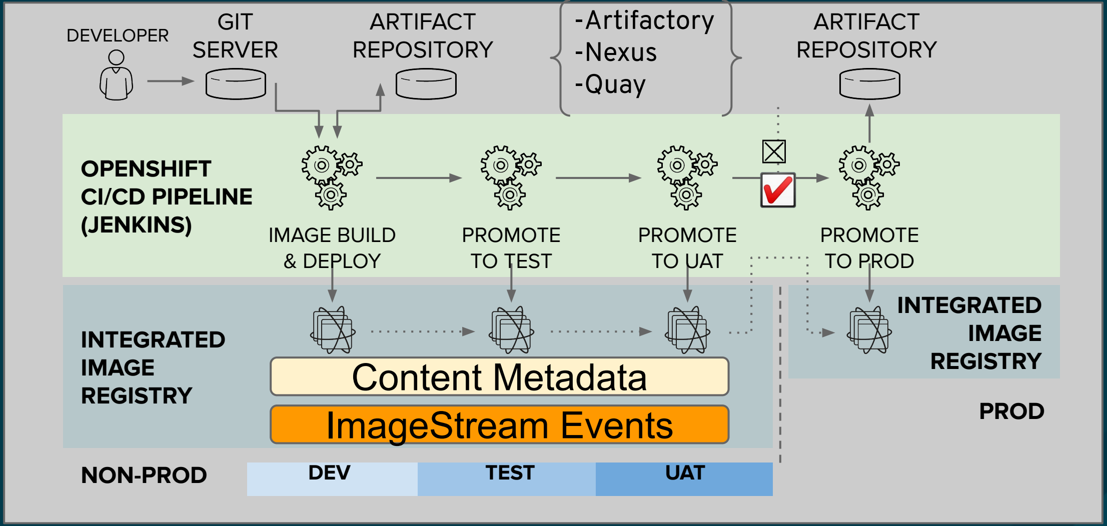
CONTENT: SIGNED IMAGES FROM RED HAT

- Cryptographically verifying that images have come from Red Hat
  - Assure provenance and integrity
  - Enable non-repudiation
  - Red Hat images are signed using Hardware Security Modules (HSMs)

Private Registry

PRIVATE REGISTRIES: SECURE ACCESS TO IMAGES

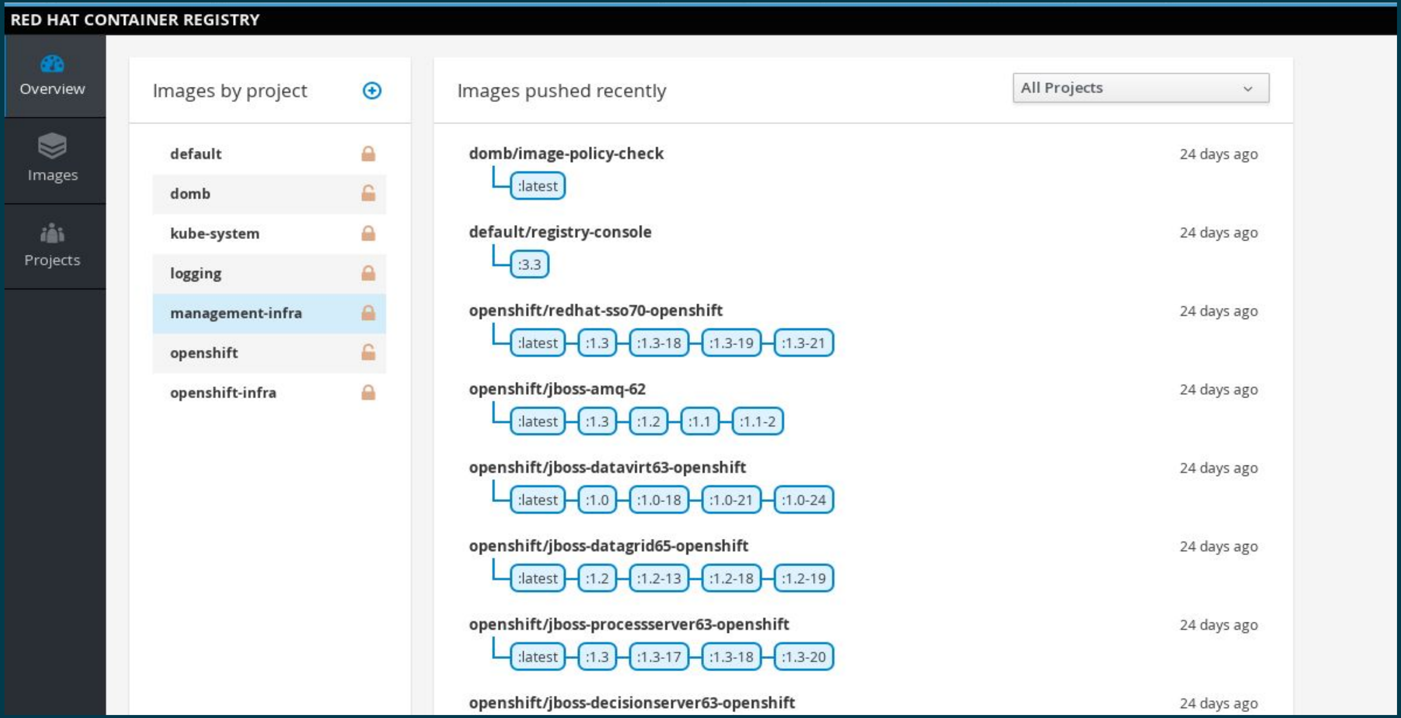
- Manage access to and promotion of images
- Metadata to automate policies for approved use (e.g. dev, test, UAT, production)
- Monitor changes to external sources
- Manage image signatures for your custom containers



Red Hat

Private Registry

RED HAT CONTAINER REGISTRY LOCAL AND SECURE WITH RBAC



RED HAT CONTAINER REGISTRY

Images by project

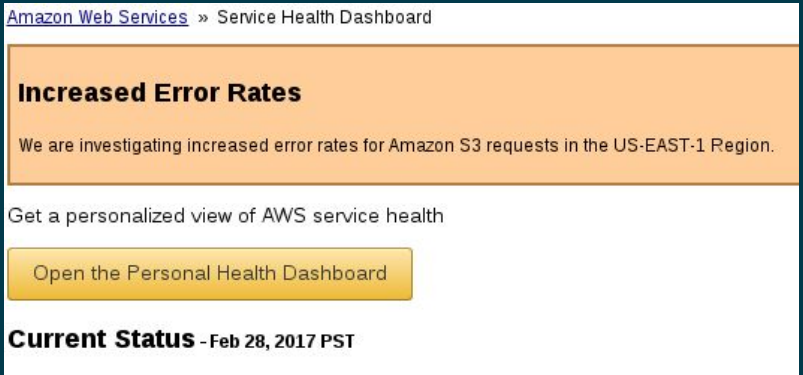
- default
- domb
- kube-system
- logging
- management-infra
- openshift
- openshift-infra

Images pushed recently

domb/image-policy-check :latest	24 days ago
default/registry-console :3.3	24 days ago
openshift/redhat-sso70-openshift :latest :1.3 :1.3-18 :1.3-19 :1.3-21	24 days ago
openshift/jboss-amq-62 :latest :1.3 :1.2 :1.1 :1.1-2	24 days ago
openshift/jboss-datavirt63-openshift :latest :1.0 :1.0-18 :1.0-21 :1.0-24	24 days ago
openshift/jboss-datagrid65-openshift :latest :1.2 :1.2-13 :1.2-18 :1.2-19	24 days ago
openshift/jboss-processserver63-openshift :latest :1.3 :1.3-17 :1.3-18 :1.3-20	24 days ago
openshift/jboss-decisionserver63-openshift	24 days ago

Private Registry

IS YOUR REGISTRY SECURE & AVAILABLE?



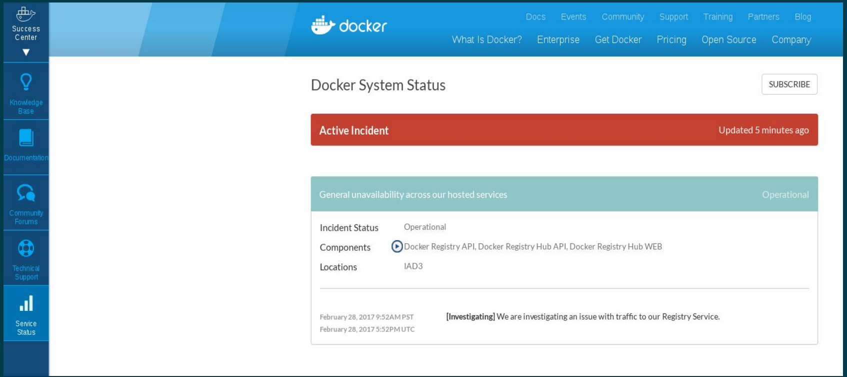
Amazon Web Services » Service Health Dashboard

Increased Error Rates

We are investigating increased error rates for Amazon S3 requests in the US-EAST-1 Region.

Get a personalized view of AWS service health

Current Status - **Feb 28, 2017 PST**



Docker System Status

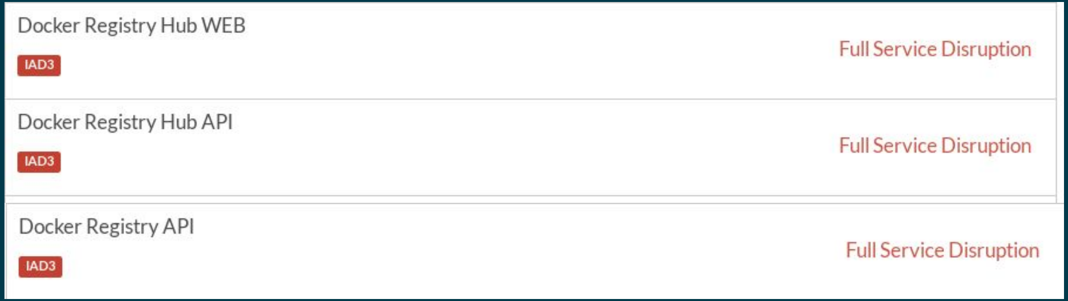
Active Incident Updated 5 minutes ago

General unavailability across our hosted services — Operational

Incident Status	Operational
Components	Docker Registry API, Docker Registry Hub API, Docker Registry Hub WEB
Locations	IAD3

February 28, 2017 9:52AM PST
February 28, 2017 5:52PM UTC

[Investigating] We are investigating an issue with traffic to our Registry Service.



Docker Registry Hub WEB IAD3	Full Service Disruption
Docker Registry Hub API IAD3	Full Service Disruption
Docker Registry API IAD3	Full Service Disruption

Private Registry

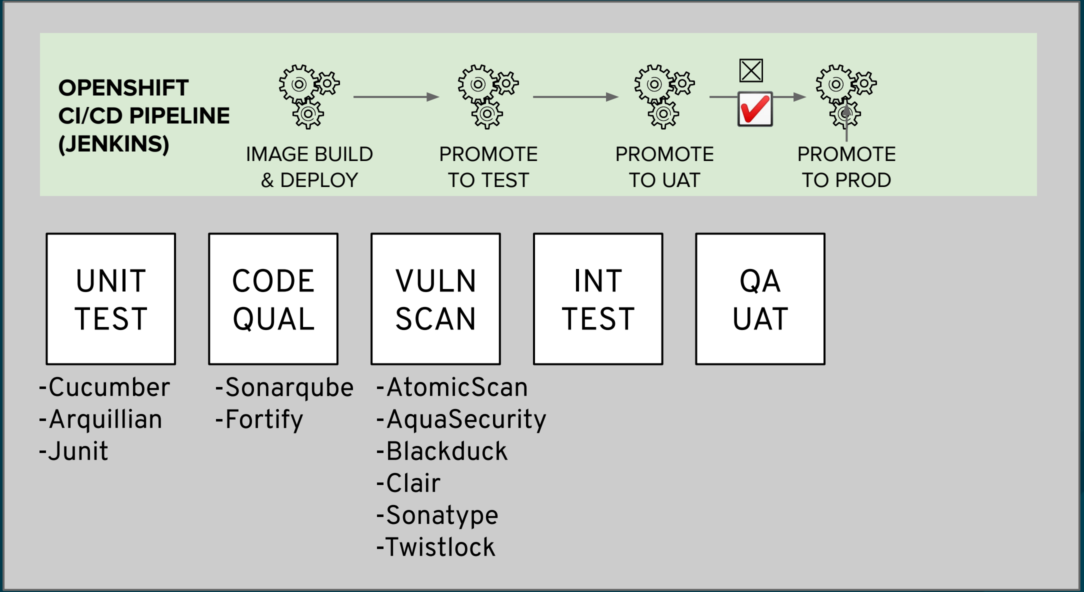
RESTRICT WHERE YOUR CONTAINERS COME FROM

```
- name: allow-images-from-internal-registry
  onResources:
  - resource: pods
  - resource: builds
  matchIntegratedRegistry: false
- name: allow-images-from-dockerhub
  onResources:
  - resource: pods
  - resource: builds
  matchRegistries:
  - docker.io
```

CI

CONTINUOUS INTEGRATION MUST INCLUDE SECURITY GATES

- Integrate security testing into your build / CI process
- Use automated policies to flag builds with issues
- Trigger automated rebuilds
- Sign your custom container images
- Design for separation of concerns



OPENSHIFT CI/CD PIPELINE (JENKINS)

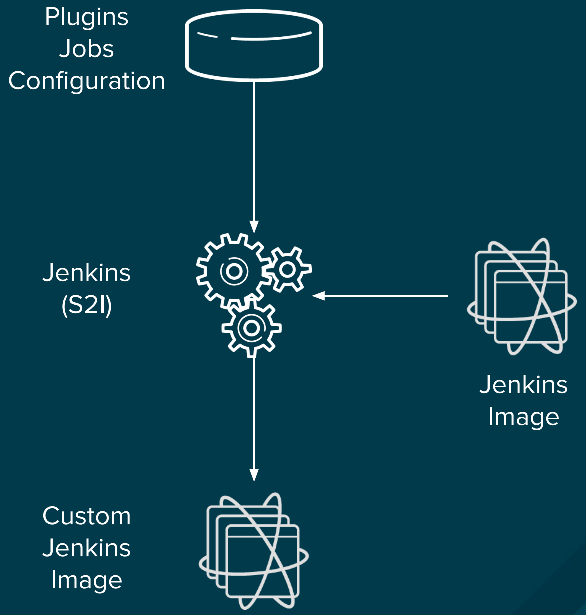
IMAGE BUILD & DEPLOY → PROMOTE TO TEST → PROMOTE TO UAT ☒ ☑ → PROMOTE TO PROD

UNIT TEST	CODE QUAL	VULN SCAN	INT TEST	QA UAT
-Cucumber -Arquillian -Junit	-Sonarqube -Fortify	-AtomicScan -AquaSecurity -Blackduck -Clair -Sonatype -Twistlock		

CI

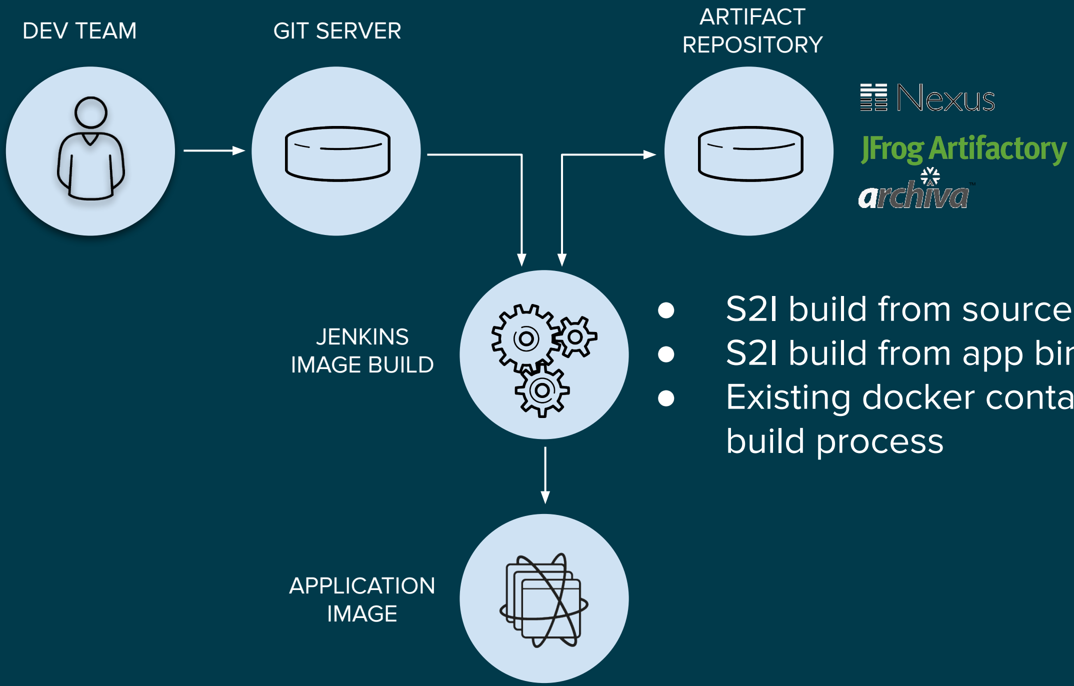
JENKINS-AS-A-SERVICE ON OPENSHIFT

- Certified Jenkins images with pre-configured plugins
  - Provided out-of-the-box
  - Follows Jenkins 1.x and 2.x LTS versions
- Jenkins S2I Builder for customizing the image
  - Install Plugins
  - Configure Jenkins
  - Configure Build Jobs
- OpenShift plugins to integrate authentication with OpenShift and also CI/CD pipelines
- Dynamically deploys Jenkins slave containers



CI

CONTINUOUS DELIVERY PIPELINE

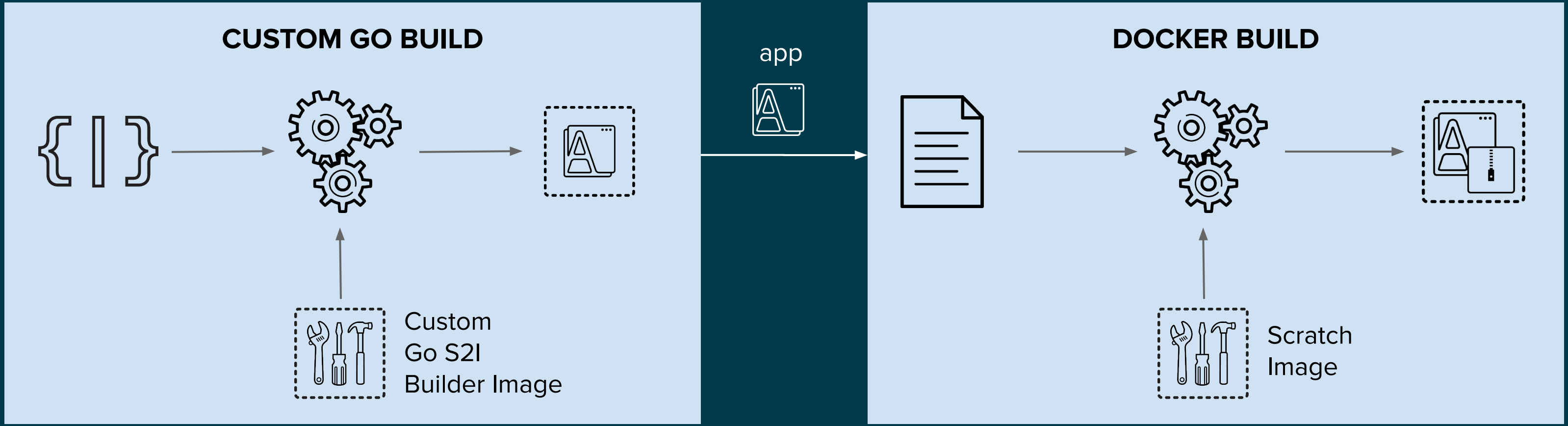


- S2I build from source code
- S2I build from app binary
- Existing docker container image build process

CI

EXAMPLE: SMALL LEAN RUNTIMES

Build the app binary and deploy on small scratch images

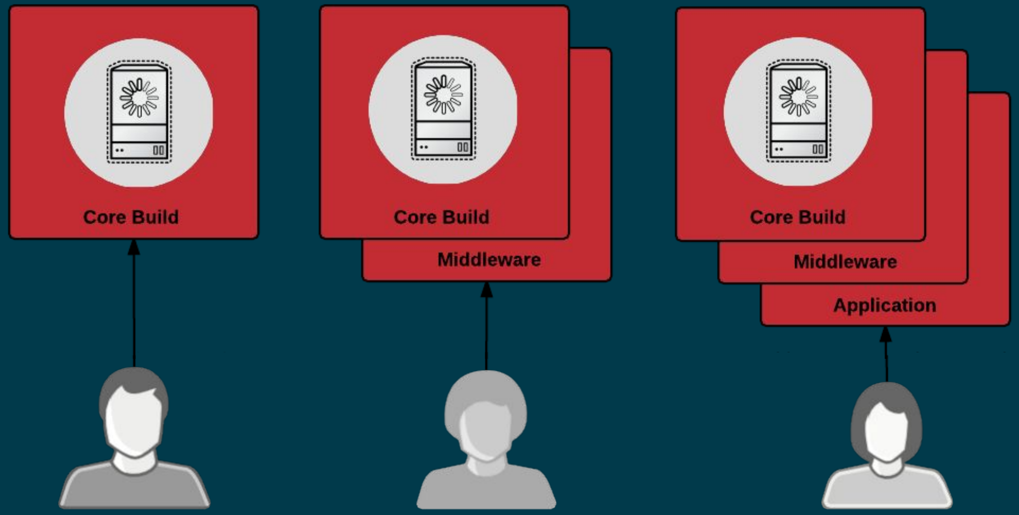


read more on https://blog.openshift.com/chaining-builds/

How to use a non-builderimage for the final application image

CI

OR BRING YOUR OWN CI & DESIGN FOR SEPARATION OF CONCERNS



Operations

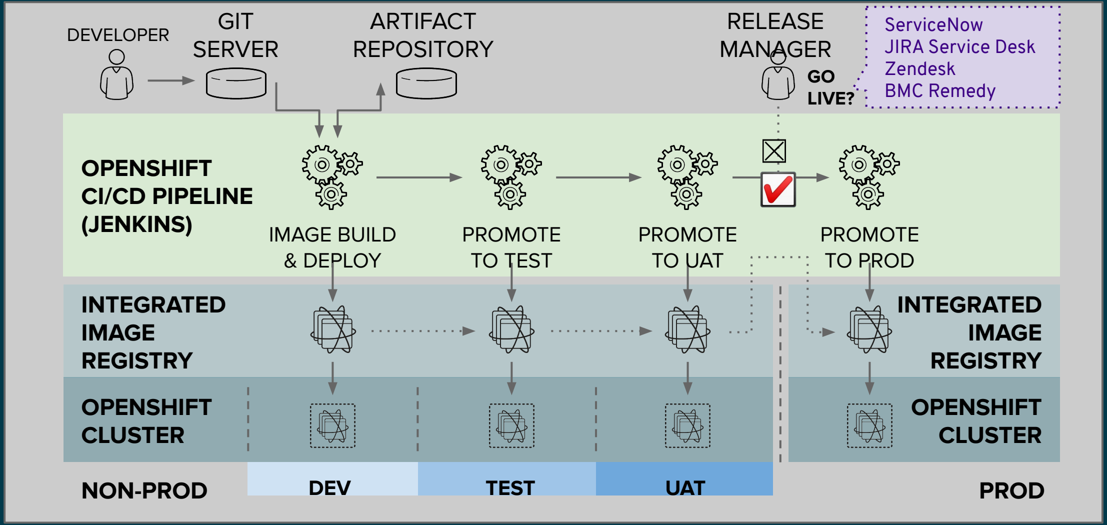
Architects

Application developers

CD

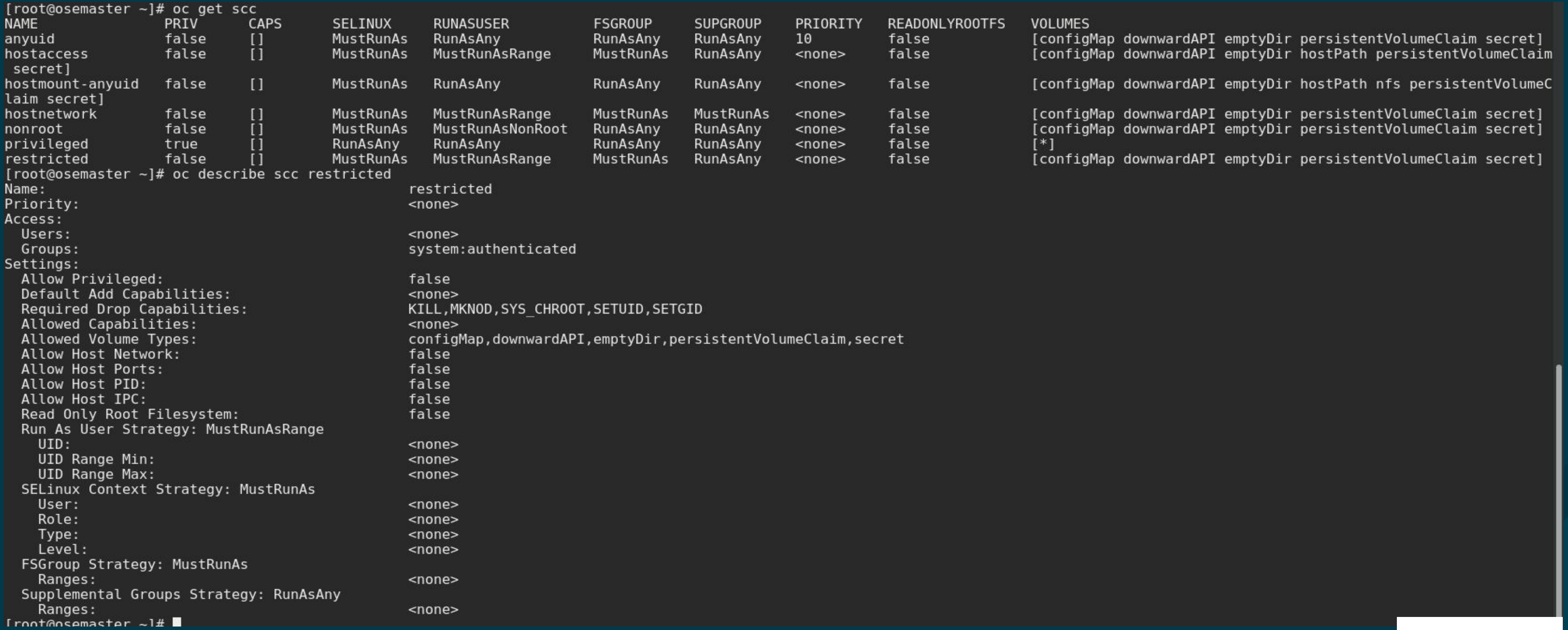
MANAGING CONTAINER DEPLOYMENT

- Monitor image registry to automatically replace affected images
- Enforce signatures at node level via signing trust policy
- Use policies to gate what can be deployed: e.g. if a container requires root access, prevent deployment
- Trust is temporal; rebuild & redeploy as needed



CD

CONTAINER DEPLOYMENT PERMISSIONS: Security Context Constraints



```
[root@osemaster ~]# oc get scc
NAME               PRIV      CAPS      SELINUX     RUNASUSER          FSGROUP     SUPGROUP    PRIORITY   READONLYROOTFS   VOLUMES
anyuid             false     []        MustRunAs   RunAsAny           RunAsAny    RunAsAny    10         false            [configMap downwardAPI emptyDir persistentVolumeClaim secret]
hostaccess         false     []        MustRunAs   MustRunAsRange     MustRunAs   RunAsAny    <none>     false            [configMap downwardAPI emptyDir hostPath persistentVolumeClaim
 secret]
hostmount-anyuid   false     []        MustRunAs   RunAsAny           RunAsAny    RunAsAny    <none>     false            [configMap downwardAPI emptyDir hostPath nfs persistentVolumeC
laim secret]
hostnetwork        false     []        MustRunAs   MustRunAsRange     MustRunAs   MustRunAs   <none>     false            [configMap downwardAPI emptyDir persistentVolumeClaim secret]
nonroot            false     []        MustRunAs   MustRunAsNonRoot   RunAsAny    RunAsAny    <none>     false            [configMap downwardAPI emptyDir persistentVolumeClaim secret]
privileged         true      []        RunAsAny    RunAsAny           RunAsAny    RunAsAny    <none>     false            [*]
restricted         false     []        MustRunAs   MustRunAsRange     MustRunAs   RunAsAny    <none>     false            [configMap downwardAPI emptyDir persistentVolumeClaim secret]
[root@osemaster ~]# oc describe scc restricted
Name:                                           restricted
Priority:                                       <none>
Access:
  Users:                                        <none>
  Groups:                                       system:authenticated
Settings:
  Allow Privileged:                             false
  Default Add Capabilities:                     <none>
  Required Drop Capabilities:                   KILL,MKNOD,SYS_CHROOT,SETUID,SETGID
  Allowed Capabilities:                         <none>
  Allowed Volume Types:                         configMap,downwardAPI,emptyDir,persistentVolumeClaim,secret
  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>
[root@osemaster ~]#
```

CD

REST ENDPOINT FOR SIGNATURES

Content Metadata

READ:

PUT
/extensions/v2/{namespace}/{name}/signatures/{digest}

```
$ curl
http://<user>:<token>@<registry-endpoint>:5000/extensions/v2/<namespace>/<name>/signatures/sha256:<digest>
```

JSON:

```
{
  "version": 2,
  "type":    "atomic",
  "name":
"sha256:4028782c08eae4a8c9a28bf661c0a8d1c2fc8e19dbaae2b018b21011197e1484@cddeb7006d914716e2728000746a0b23",
  "content": "<base64 encoded signature>",
}
```

WRITE:

GET
/extensions/v2/{namespace}/{name}/signatures/{digest}

```
$ curl
http://<user>:<token>@<registry-endpoint>:5000/extensions/v2/<namespace>/<name>/signatures/sha256:<digest>
```

JSON:

```
{
  "signatures": [
  {
    "version": 2,
    "type":    "atomic",
    "name":
"sha256:4028782c08eae4a8c9a28bf661c0a8d1c2fc8e19dbaae2b018b21011197e1484@cddeb7006d914716e2728000746a0b23",
    "content": "<base64 encoded signature>",
  }
  ]
}
```

CD

VALIDATE IMAGE SIGNATURES

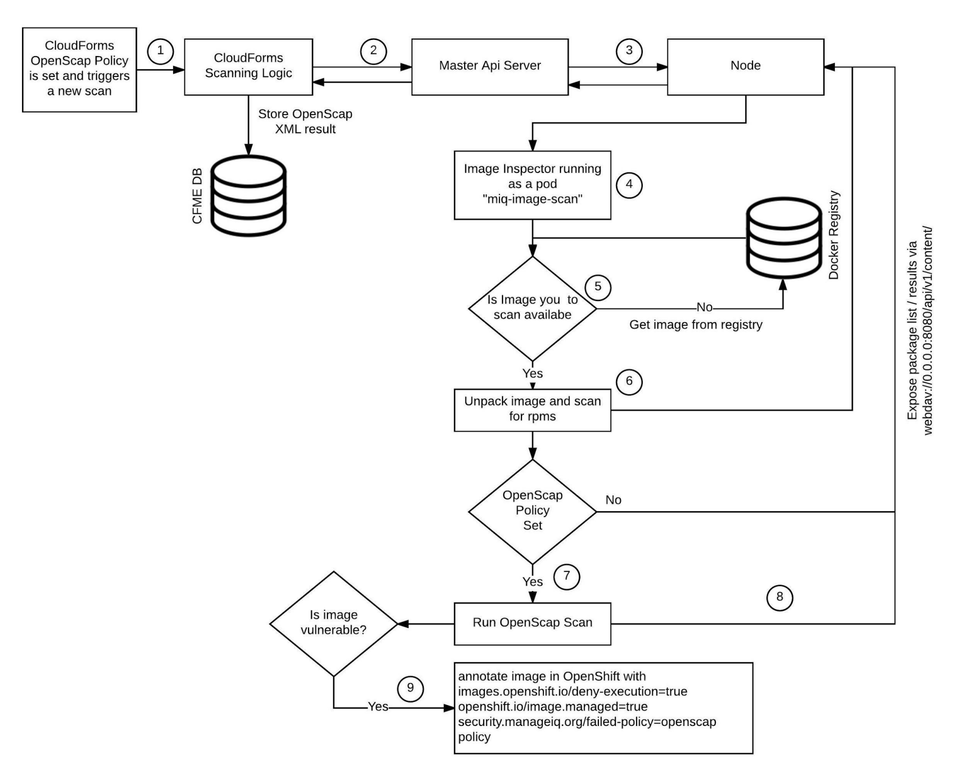
Content Metadata

```
# Verify the image and save the result back to image stream
$ oadm verify-image-signature
sha256:c13060b74c0348577cbe07dedcdb698f7d893ea6f74847154e5ef3c8c
9369b2c \
  --expected-identity=172.30.204.70:5000/test/origin-pod:latest --save
--as=system:admin
sha256:c13060b74c0348577cbe07dedcdb698f7d893ea6f74847154e5ef3c8c
9369b2c signature 0 is verified (signed by key: "172B61E538AAC0EE")
```

CD

VULNERABLE? CLOUDFORMS TAKES ACTION!

Content Metadata



CloudForms OpenScap Policy is set and triggers a new scan

1

CloudForms Scanning Logic

2

Master Api Server

3

Node

Store OpenScap XML result

CFME DB

Image Inspector running as a pod "miq-image-scan"

4

Docker Registry

Expose package list / results via webdav://0.0.0.0:8080/api/v1/content/

5

Is Image you to scan availabe

No

Get image from registry

Yes

6

Unpack image and scan for rpms

OpenScap Policy Set

No

Yes

7

Run OpenScap Scan

8

Is image vulnerable?

9

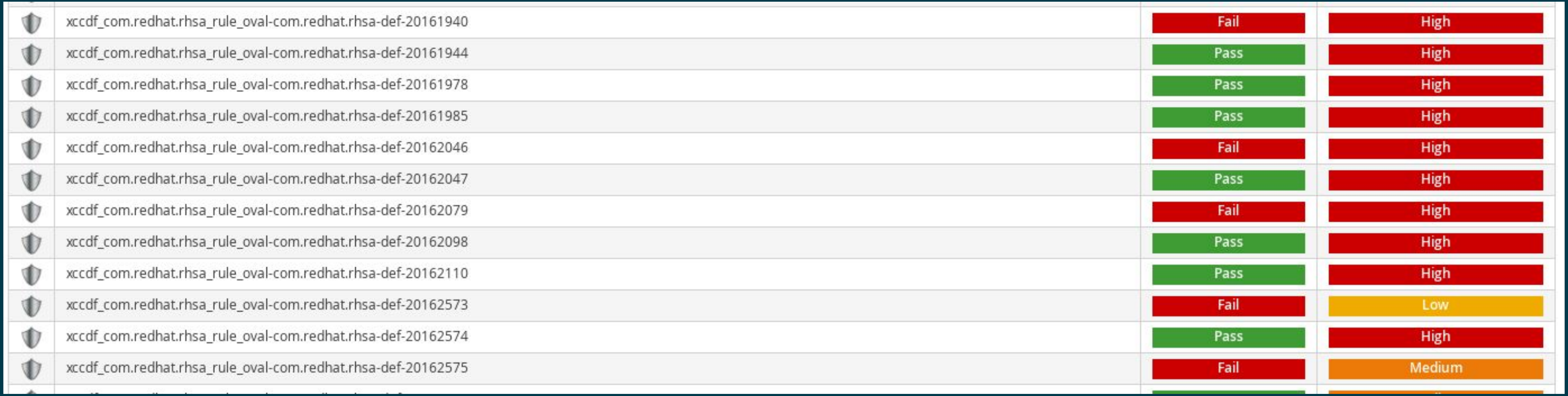
Yes

annotate image in OpenShift with
images.openshift.io/deny-execution=true
openshift.io/image.managed=true
security.manageiq.org/failed-policy=openscap policy

CD

VULNERABLE? CLOUDFORMS TAKES ACTION!

Content Metadata

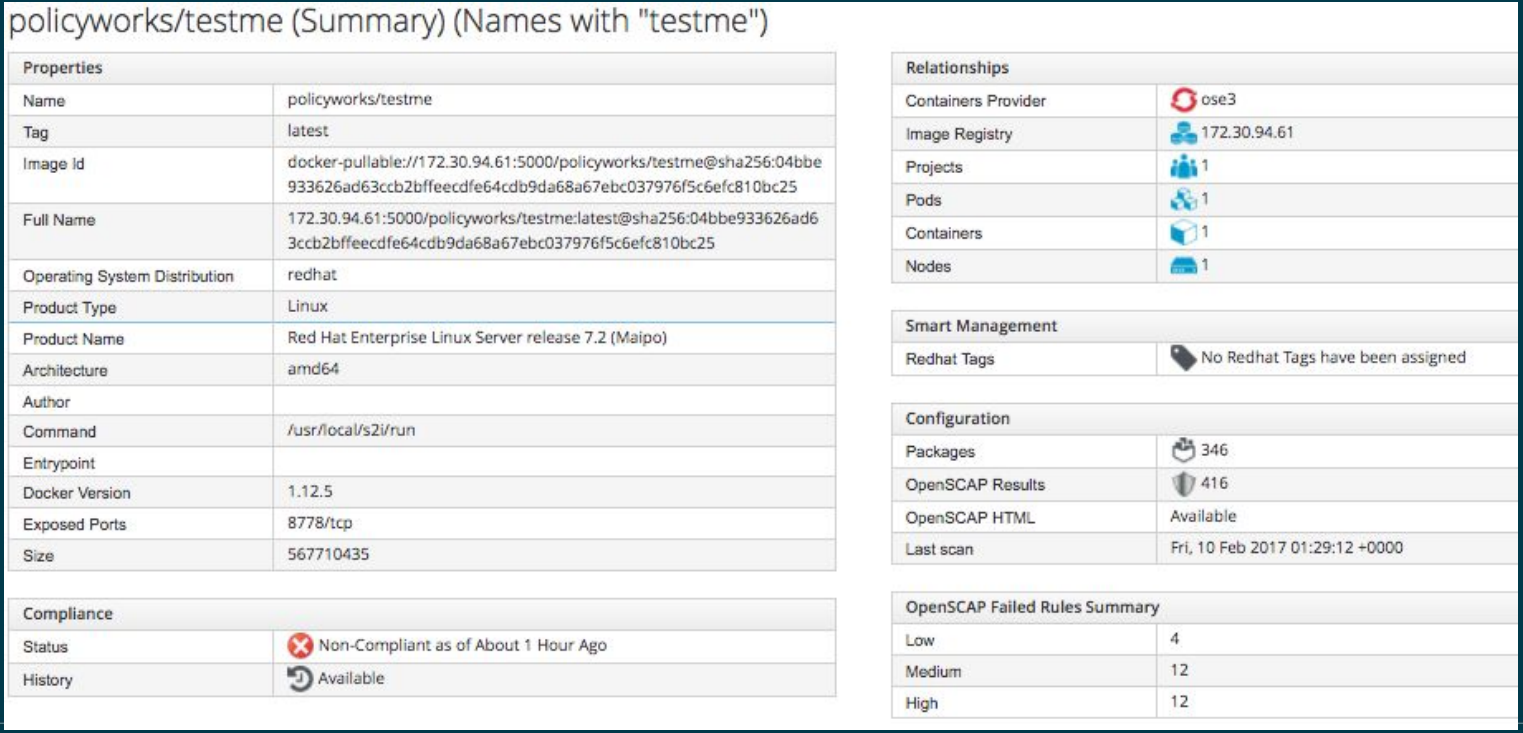


xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20161940	Fail	High
xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20161944	Pass	High
xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20161978	Pass	High
xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20161985	Pass	High
xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20162046	Fail	High
xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20162047	Pass	High
xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20162079	Fail	High
xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20162098	Pass	High
xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20162110	Pass	High
xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20162573	Fail	Low
xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20162574	Pass	High
xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20162575	Fail	Medium

CD

VULNERABLE? CLOUDFORMS TAKES ACTION!

Content Metadata



policyworks/testme (Summary) (Names with "testme")

Properties	
Name	policyworks/testme
Tag	latest
Image Id	docker-pullable://172.30.94.61:5000/policyworks/testme@sha256:04bbe933626ad63ccb2bffeecdfe64cdb9da68a67ebc037976f5c6efc810bc25
Full Name	172.30.94.61:5000/policyworks/testme:latest@sha256:04bbe933626ad63ccb2bffeecdfe64cdb9da68a67ebc037976f5c6efc810bc25
Operating System Distribution	redhat
Product Type	Linux
Product Name	Red Hat Enterprise Linux Server release 7.2 (Maipo)
Architecture	amd64
Author	
Command	/usr/local/s2i/run
Entrypoint	
Docker Version	1.12.5
Exposed Ports	8778/tcp
Size	567710435

Compliance	
Status	☒ Non-Compliant as of About 1 Hour Ago
History	Available

Relationships	
Containers Provider	ose3
Image Registry	172.30.94.61
Projects	1
Pods	1
Containers	1
Nodes	1

Smart Management	
Redhat Tags	No Redhat Tags have been assigned

Configuration	
Packages	346
OpenSCAP Results	416
OpenSCAP HTML	Available
Last scan	Fri, 10 Feb 2017 01:29:12 +0000

OpenSCAP Failed Rules Summary	
Low	4
Medium	12
High	12

CD

VULNERABLE? CLOUDFORMS TAKES ACTION!

Content Metadata

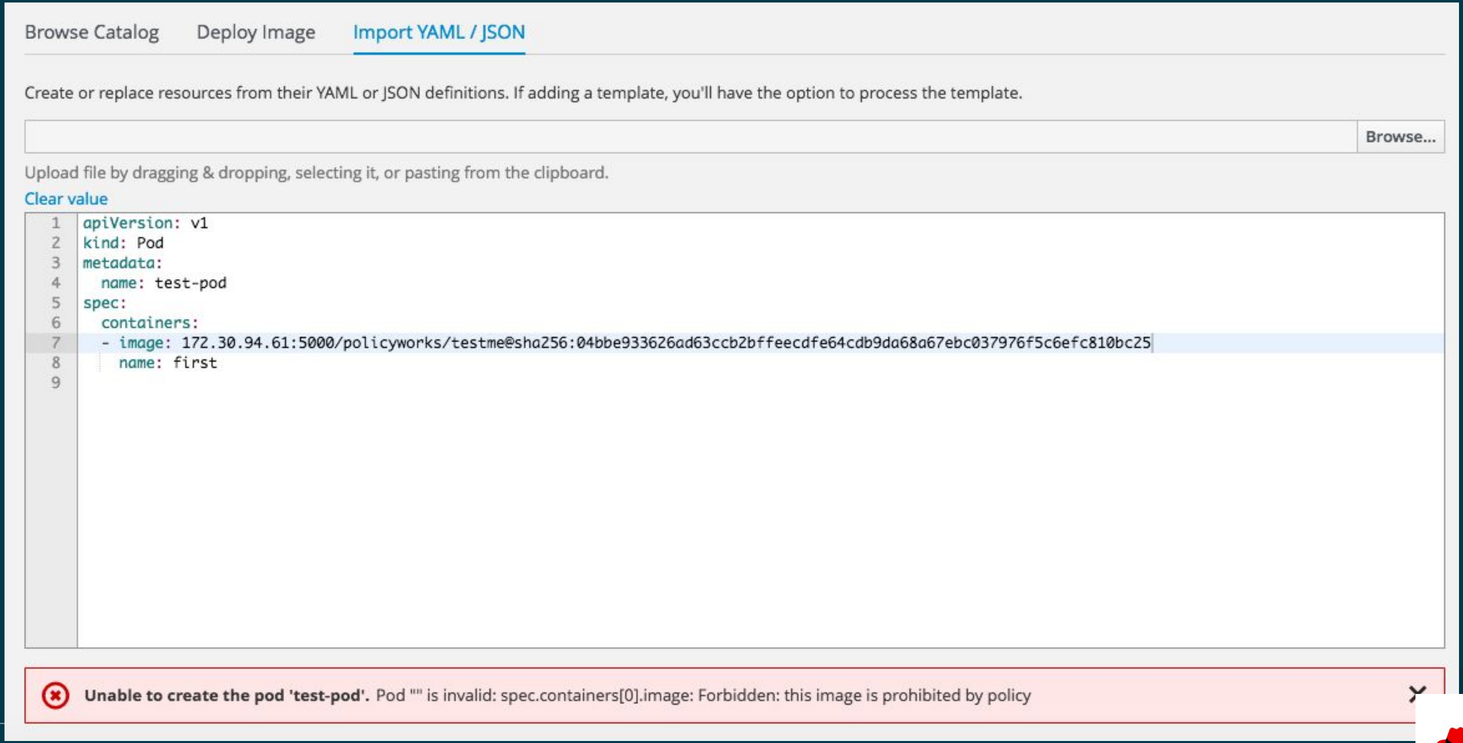
CloudForms sets the following annotations to prevent the image from running.

image.openshift.io/deny-execution: true
openshift.io/image.managed: true
security.manageiq.org/failed-policy: openscap policy

CD

VULNERABLE? CLOUDFORMS TAKES ACTION!

Content Metadata



Create or replace resources from their YAML or JSON definitions. If adding a template, you'll have the option to process the template.

Upload file by dragging & dropping, selecting it, or pasting from the clipboard.

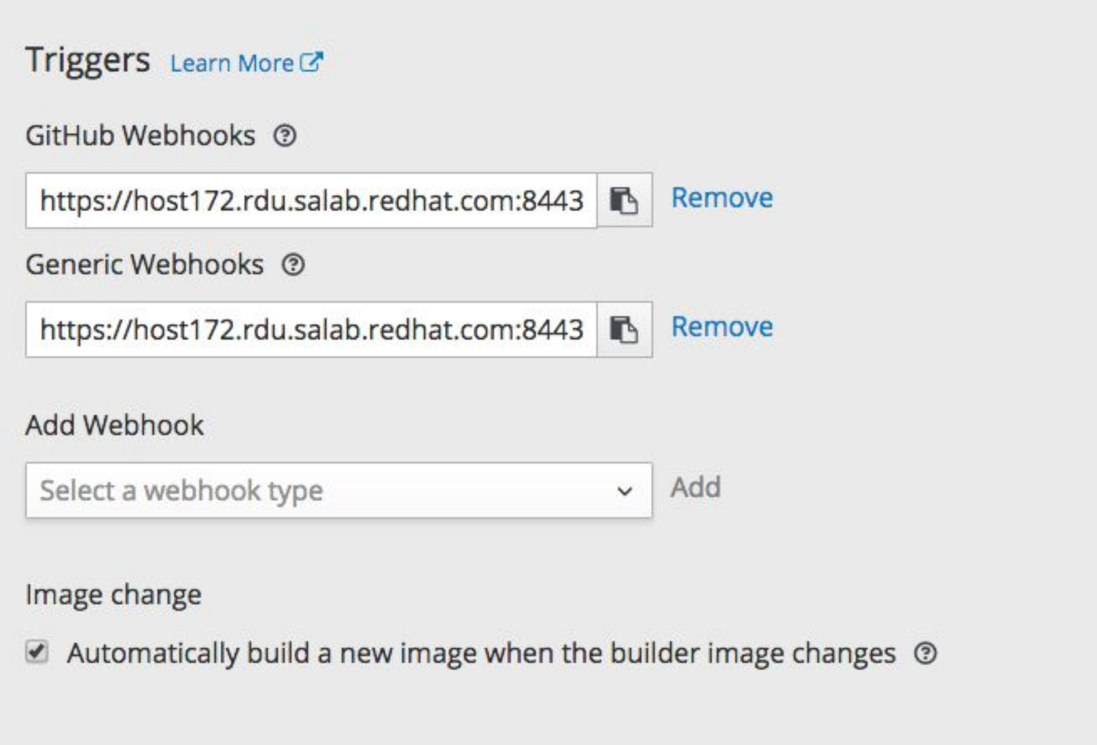
```
apiVersion: v1
kind: Pod
metadata:
  name: test-pod
spec:
  containers:
  - image: 172.30.94.61:5000/policyworks/testme@sha256:04bbe933626ad63ccb2bffeecdfe64cdb9da68a67ebc037976f5c6efc810bc25
    name: first
```

Unable to create the pod 'test-pod'. Pod "" is invalid: spec.containers[0].image: Forbidden: this image is prohibited by policy

Content Metadata

ImageStream Events

GET UPDATED IMAGE



Triggers Learn More

GitHub Webhooks

https://host172.rdu.salab.redhat.com:8443

Generic Webhooks

https://host172.rdu.salab.redhat.com:8443

Add Webhook

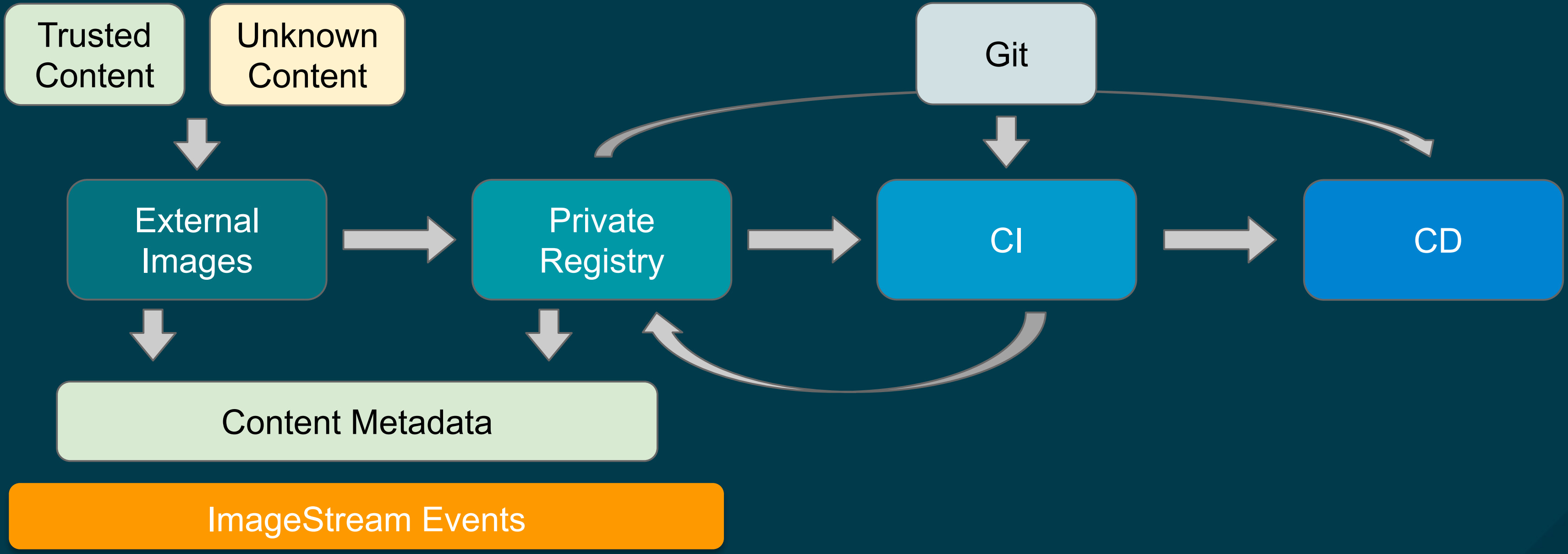
Select a webhook type

Image change

☑ Automatically build a new image when the builder image changes

AUTOMATE THE CONTENT LIFECYCLE

Trust is temporal; rebuild and redeploy as needed



DEFEND

Secure the Infrastructure

Container Platform	Container Host Multi-tenancy
Network Isolation	Storage
Audit & Logging	API Management

CONTAINER HOST & MULTI-TENANCY THE OS MATTERS

RED HAT ENTERPRISE LINUX

RED HAT ENTERPRISE LINUX ATOMIC HOST

THE FOUNDATION FOR SECURE, SCALABLE CONTAINERS

A stable, reliable host environment with built-in security features that allow you to isolate containers from other containers and from the kernel.

Minimized host environment tuned for running Linux containers while maintaining the built-in security features of Red Hat Enterprise Linux..

SELinux

Kernel namespaces

Cgroups

Capabilities

R/O Mounts

SELINUX - MAC - MCS

- SElinux is a LABELING system
- Every Process has a Label
- Every file, Directory, System object has a Label
- Policy rules control access between labeled processes and labeled objects
- The Kernel enforces the rules

SELINUX - MAC - MCS - Process

system_u:system_r:container_runtime_t:s0

```
[root@osemaster ~]# ps -efZ | grep docker-containerd-shim-current
system_u:system_r:container_runtime_t:s0 root 3035 1479  0 Feb15 ?       00:00:01
/usr/bin/docker-containerd-shim-current
4d254785cbc6ee7aae8facc48555251e2385f65d89553b319b6324b1501e4b16
/var/run/docker/libcontainerd/4d254785cbc6ee7aae8facc48555251e2385f65d89553b319b6324b1501e4b16
/usr/libexec/docker/docker-runc-current
```

The OOTB SElinux policy container.te defines what you can execute and access with the label container_runtime_t

SElinux Policy module for the container

SELINUX - MAC - MCS - Files

container_var_lib_t / svirt_sandbox_file_t

```
[root@osemaster ~]# ls -lZ
/var/lib/docker/containers/97de4217a04b6532e312cfb3e4638529aeb7dfa281a2cc067e092fcee82e6737
/
-rw-r-----. root root system_u:object_r:container_var_lib_t:s0
97de4217a04b6532e312cfb3e4638529aeb7dfa281a2cc067e092fcee82e6737-json.log
-rw-rw-rw-. root root system_u:object_r:container_var_lib_t:s0 config.v2.json
-rw-rw-rw-. root root system_u:object_r:container_var_lib_t:s0 hostconfig.json
-rw-r--r--. root root system_u:object_r:svirt_sandbox_file_t:s0 hostname
-rw-r--r--. root root system_u:object_r:svirt_sandbox_file_t:s0:c0,c1 hosts
-rw-r--r--. root root system_u:object_r:svirt_sandbox_file_t:s0 resolv.conf
-rw-r--r--. root root system_u:object_r:container_var_lib_t:s0 resolv.conf.hash
drwxr-xr-x. root root system_u:object_r:svirt_sandbox_file_t:s0:c0,c1 secrets
drwx------. root root system_u:object_r:container_var_lib_t:s0 shm
```

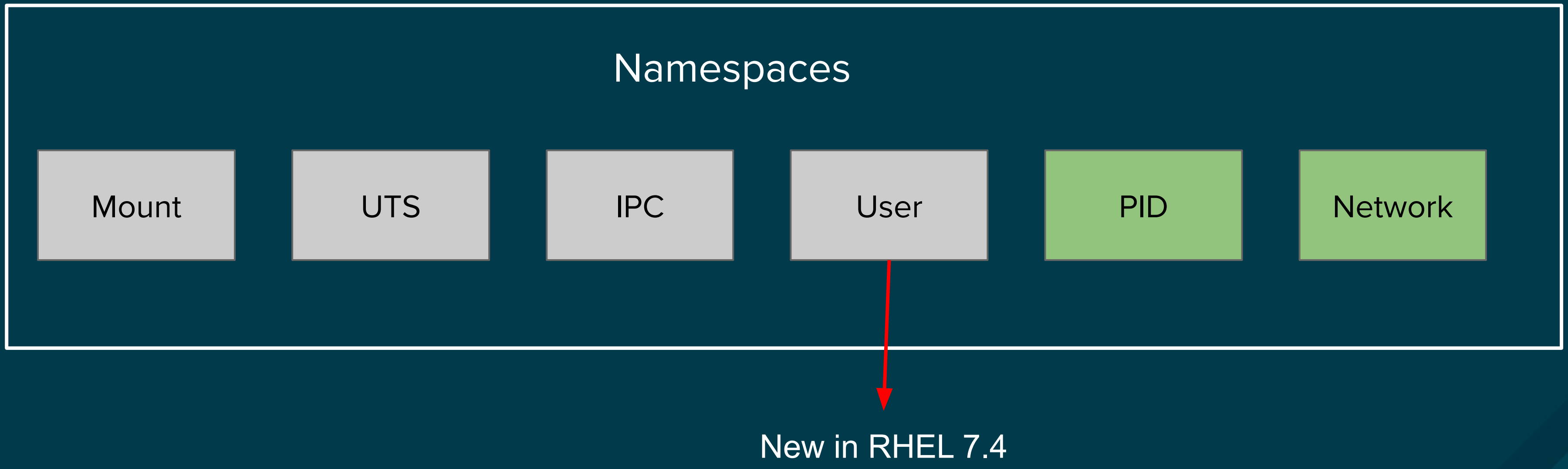
SElinux Policy module for the container

SELINUX TO THE RESCUE

On-entry container attack - CVE-2016-9962

On Red Hat systems with SELinux enabled, the dangers of even privileged containers are mitigated. SELinux prevents container processes from accessing host content even if those container processes manage to gain access to the actual file descriptors.

NAMESPACES
Process Isolation



Namespaces

Mount

UTS

IPC

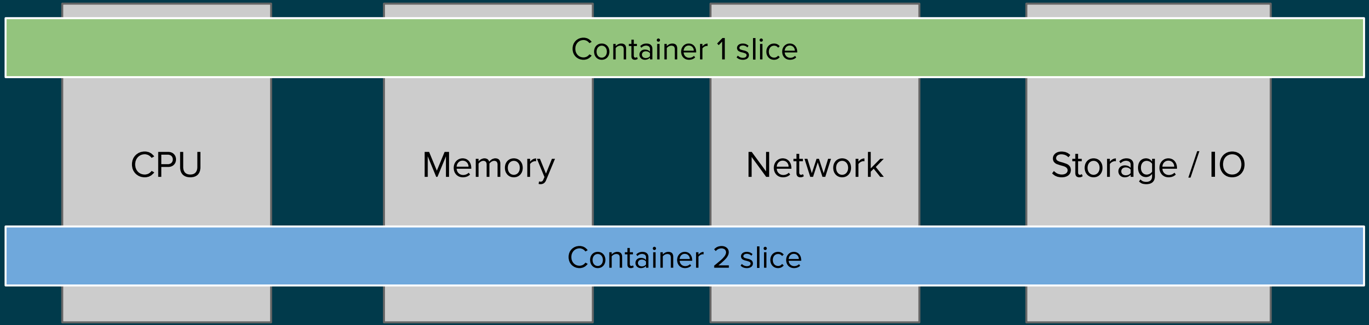
User

PID

Network

New in RHEL 7.4

CGROUPS - Resource Isolation

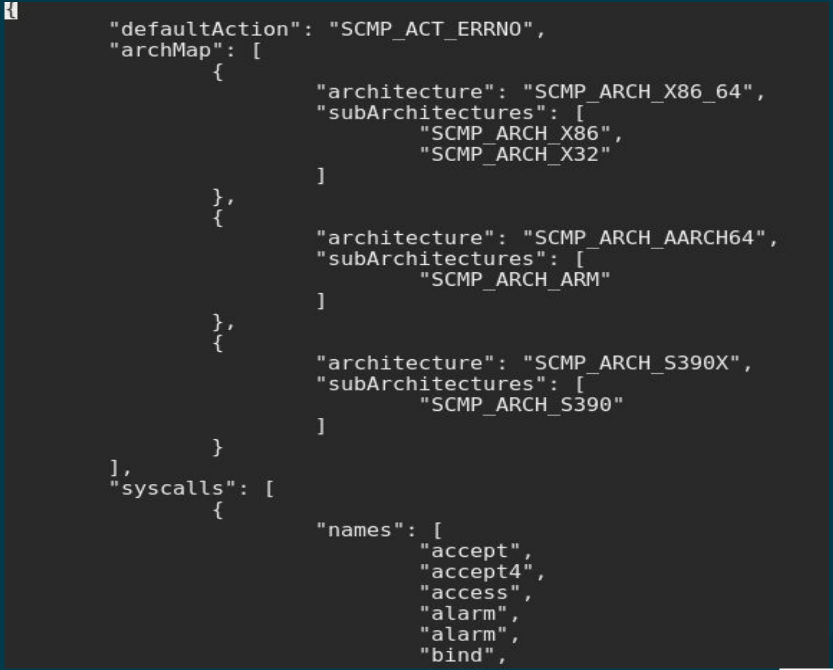


CAPABILITIES - DROPPING PRIVILEGES

CAP_SETPCAP	Modify process capabilities
CAP_SYS_MODULE	Insert/Remove kernel modules
CAP_SYS_RAWIO	Modify Kernel Memory
CAP_SYS_PACCT	Configure process accounting
CAP_SYS_NICE	Modify Priority of processes
CAP_SYS_RESOURCE	Override Resource Limits
CAP_SYS_TIME	Modify the system clock
CAP_SYS_TTY_CONFIG	Configure tty devices
CAP_AUDIT_WRITE	Write the audit log
CAP_AUDIT_CONTROL	Configure Audit Subsystem
CAP_MAC_OVERRIDE	Ignore Kernel MAC Policy
CAP_MAC_ADMIN	Configure MAC Configuration
CAP_SYSLOG	Modify Kernel printk behaviour
CAP_NET_ADMIN	Configure the network:
CAP_SYS_ADMIN	- Setting the hostname/domainname
	- mount(),unmount()
	- nfsservctl
	- ....

CAPABILITIES - DROPPING PRIVILEGES

A root user inside a container running in OpenShift has none of the previous capabilities available!



```
{
        "defaultAction": "SCMP_ACT_ERRNO",
        "archMap": [
                {
                        "architecture": "SCMP_ARCH_X86_64",
                        "subArchitectures": [
                                "SCMP_ARCH_X86",
                                "SCMP_ARCH_X32"
                        ]
                },
                {
                        "architecture": "SCMP_ARCH_AARCH64",
                        "subArchitectures": [
                                "SCMP_ARCH_ARM"
                        ]
                },
                {
                        "architecture": "SCMP_ARCH_S390X",
                        "subArchitectures": [
                                "SCMP_ARCH_S390"
                        ]
                }
        ],
        "syscalls": [
                {
                        "names": [
                                "accept",
                                "accept4",
                                "access",
                                "alarm",
                                "alarm",
                                "bind",
```

READ ONLY MOUNTS

/sys

/proc/sys

/proc/sysrg-trigger

/proc/irq

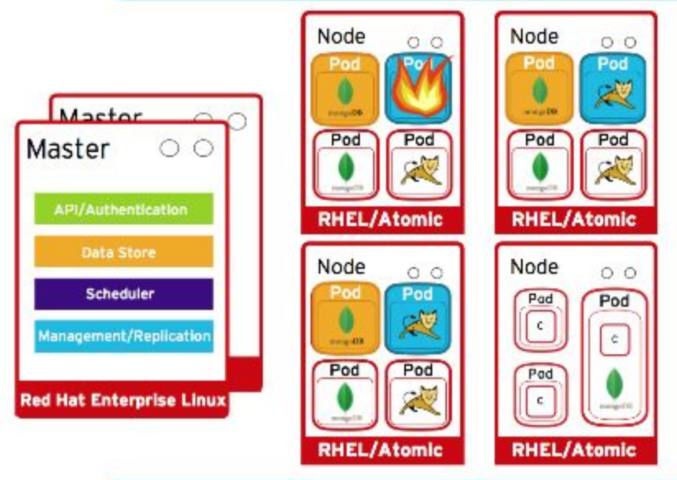
/proc/bus

R/O

SECURING THE CONTAINER PLATFORM

Use a container orchestration platform with integrated security features including

- Role-based Access Controls with LDAP and OAuth integration
- Secure communication
- Platform multitenant security
- Integrated & extensible secrets management
- Logging, Monitoring, Metrics
- Enable integration with the security ecosystem



AUTHENTICATION & AUTHORIZATION (Master)

OAUTH API AUTHENTICATION

OpenShift includes an OAuth server, which does three things:

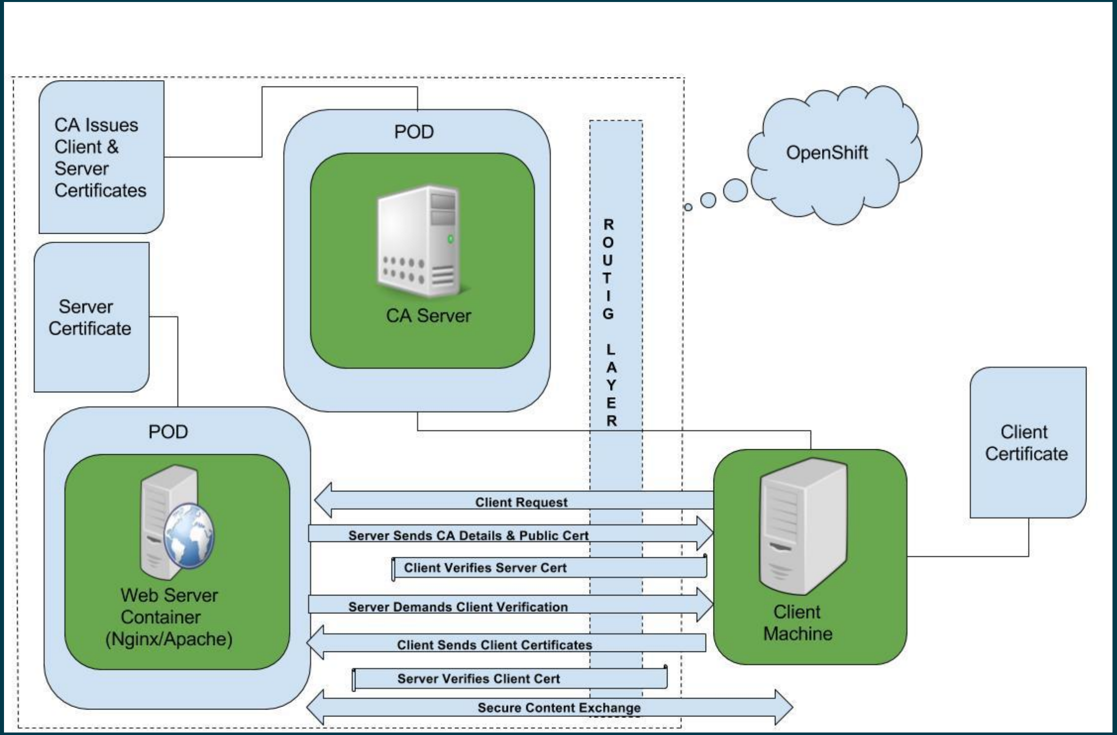
- Identifies the person requesting a token, using a configured identity provider
- Determines a mapping from that identity to an OpenShift user
  - Allows multiple identities to map to the same OpenShift user
  - Allows deconflicting between identity provider roles
- Issues an OAuth access token which authenticates that user to the API

API ROLE-BASED AUTHORIZATION

- Matches request attributes (verb,object,etc)
- If no roles match, request is denied (deny by default)
- Operator- and user-level roles are defined by default
- Custom roles are supported

SECURE COMMUNICATION

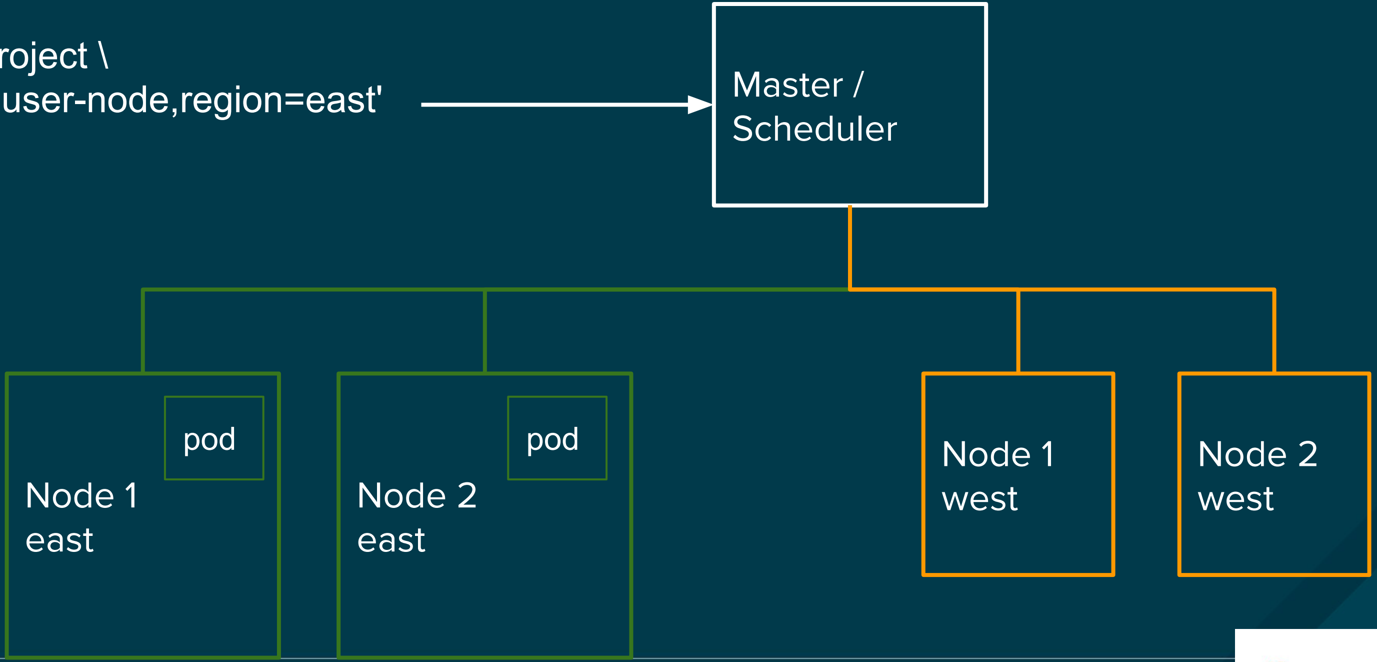
End to End
Two Way SSL
Encryption


CA Issues Client & Server Certificates
POD
CA Server
OpenShift
ROUTIG LAYER
Server Certificate
POD
Web Server Container (Nginx/Apache)
Client Certificate
Client Machine
Client Request
Server Sends CA Details & Public Cert
Client Verifies Server Cert
Server Demands Client Verification
Client Sends Client Certificates
Server Verifies Client Cert
Secure Content Exchange

https://developers.redhat.com/blog/2017/01/24/end-to-end-encryption-with-openshift-part-1-two-way-ssl/

PLATFORM MULTITENANCY
Isolate Workloads by Node

```
$ oadm new-project myproject \
    --node-selector='type=user-node,region=east'
```



SECRETS MANAGEMENT

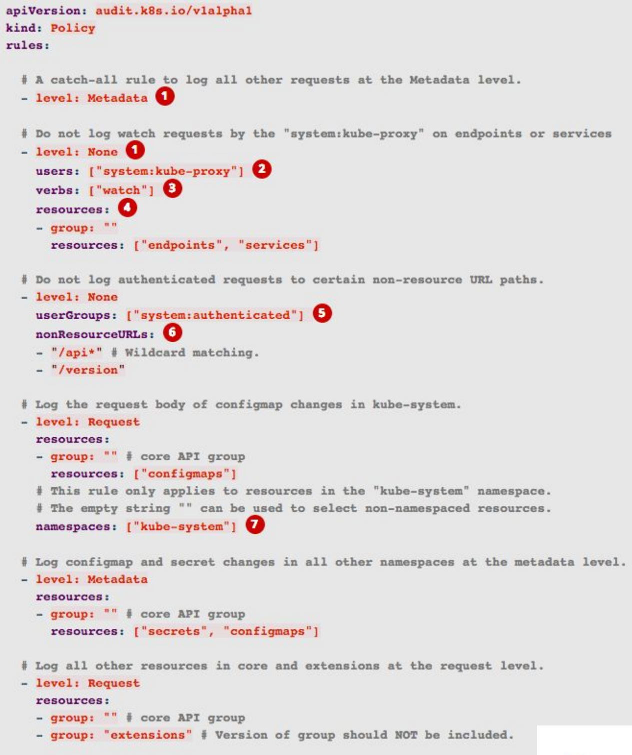
- Etcd secrets encrypted by default
- Flexvolume API supported for easier integration with 3rd party vault solutions
- Use Node Authorizer & Node Restriction Admission to prevent Pods from gaining access to secrets, configMaps, PV, PVC or API objects from other nodes

```
# oadm policy remove-cluster-role-from-group system:node system:nodes
```

LOGGING & AUDIT: EFK STACK

ElasticSearch, Fluentd, Kibana

- Event system with log aggregation
- All login, docker, Master events
- All API calls
- Use for ad hoc analytics and post mortem forensics
- Tech preview Central Audit policyFile or policyConfiguration available with 3.7

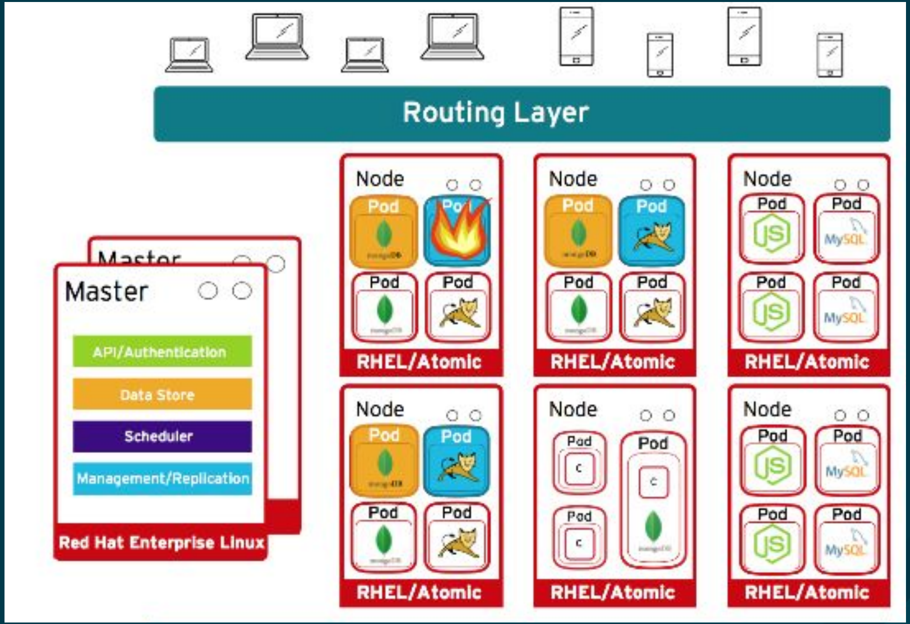


```
apiVersion: audit.k8s.io/v1alpha1
kind: Policy
rules:

  # A catch-all rule to log all other requests at the Metadata level.
  - level: Metadata ❶

  # Do not log watch requests by the "system:kube-proxy" on endpoints or services
  - level: None ❶
    users: ["system:kube-proxy"] ❷
    verbs: ["watch"] ❸
    resources: ❹
    - group: ""
      resources: ["endpoints", "services"]

  # Do not log authenticated requests to certain non-resource URL paths.
  - level: None
    userGroups: ["system:authenticated"] ❺
    nonResourceURLs: ❻
    - "/api*" # Wildcard matching.
    - "/version"

  # Log the request body of configmap changes in kube-system.
  - level: Request
    resources:
    - group: "" # core API group
      resources: ["configmaps"]
    # This rule only applies to resources in the "kube-system" namespace.
    # The empty string "" can be used to select non-namespaced resources.
    namespaces: ["kube-system"] ❼

  # Log configmap and secret changes in all other namespaces at the metadata level.
  - level: Metadata
    resources:
    - group: "" # core API group
      resources: ["secrets", "configmaps"]

  # Log all other resources in core and extensions at the request level.
  - level: Request
    resources:
    - group: "" # core API group
    - group: "extensions" # Version of group should NOT be included.
```

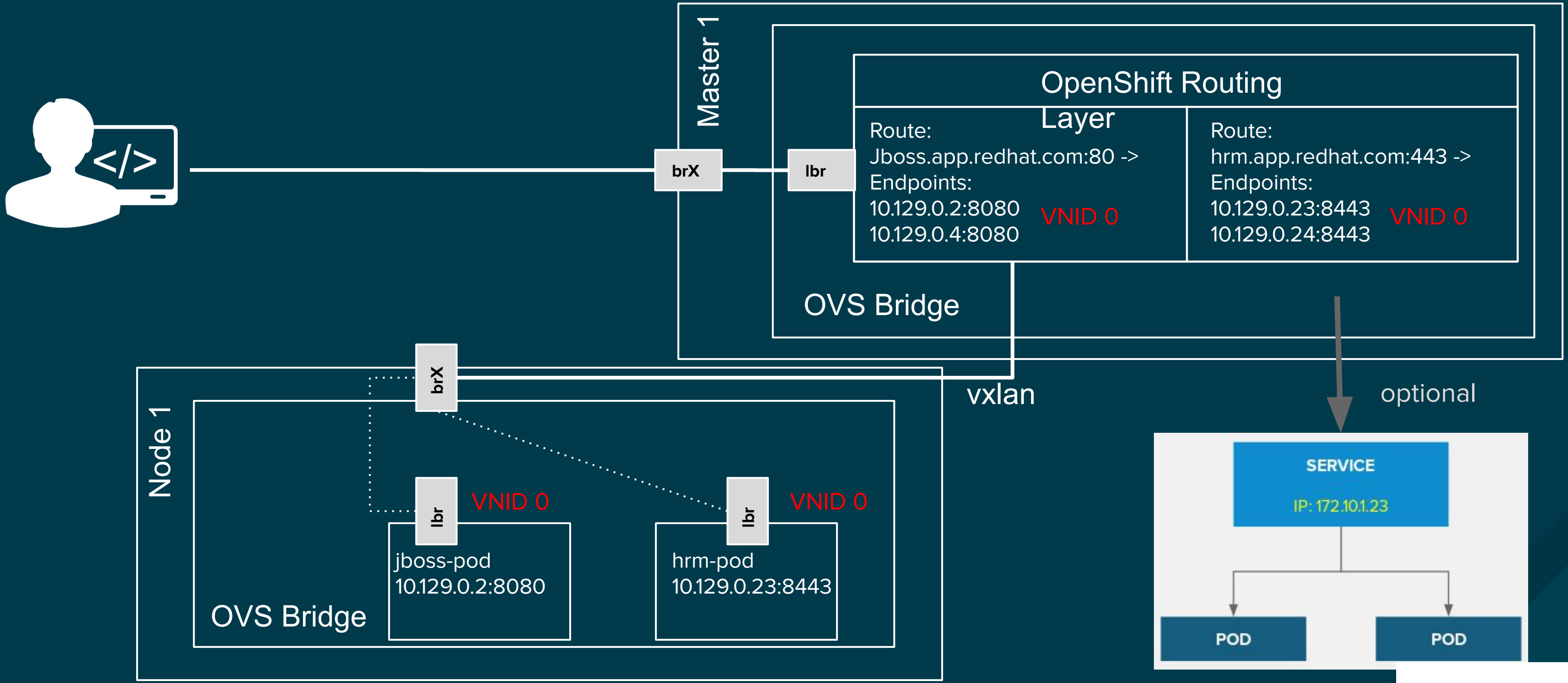
NETWORK DEFENSE

Use network namespaces to

- Isolate applications from other applications within a cluster
- Isolate environments (Dev / Test / Prod) from other environments within a cluster



OVS - SUBNET / Reverse Proxy



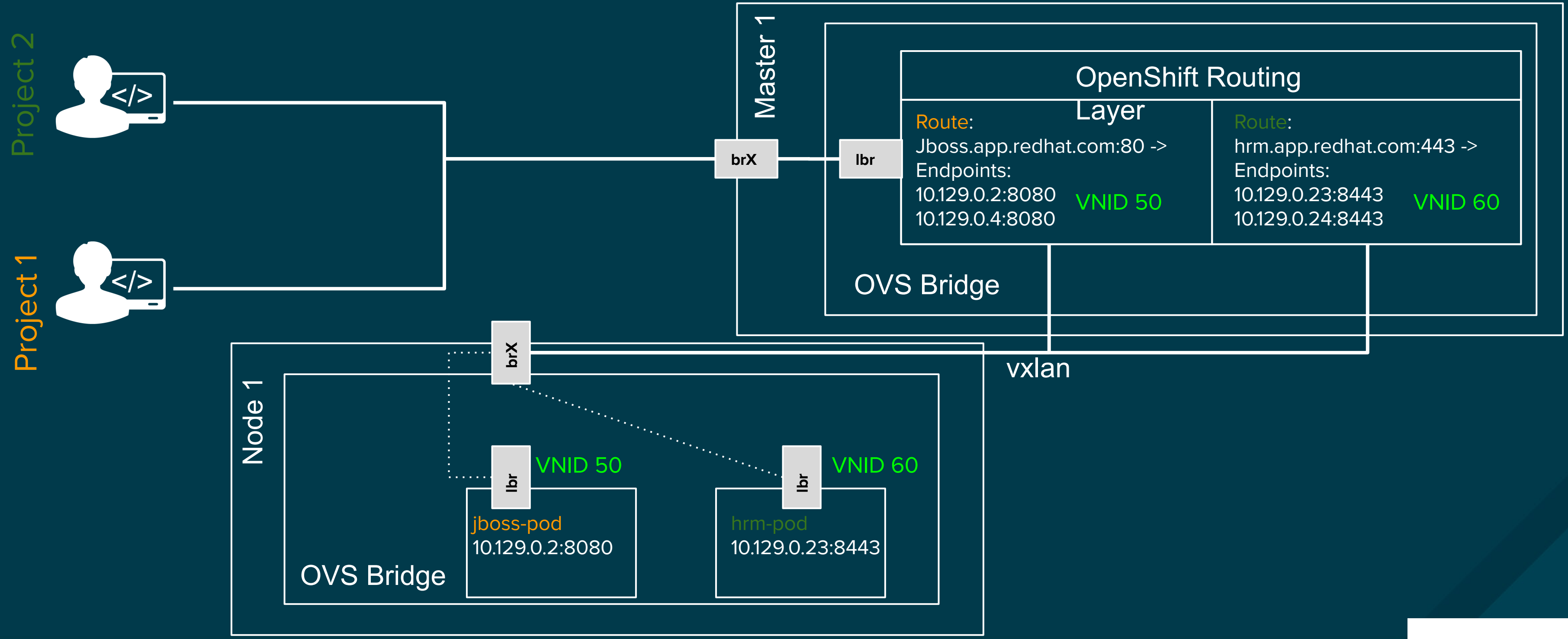
Route:
Jboss.app.redhat.com:80 ->
Endpoints:
10.129.0.2:8080
10.129.0.4:8080

Route:
hrm.app.redhat.com:443 ->
Endpoints:
10.129.0.23:8443
10.129.0.24:8443

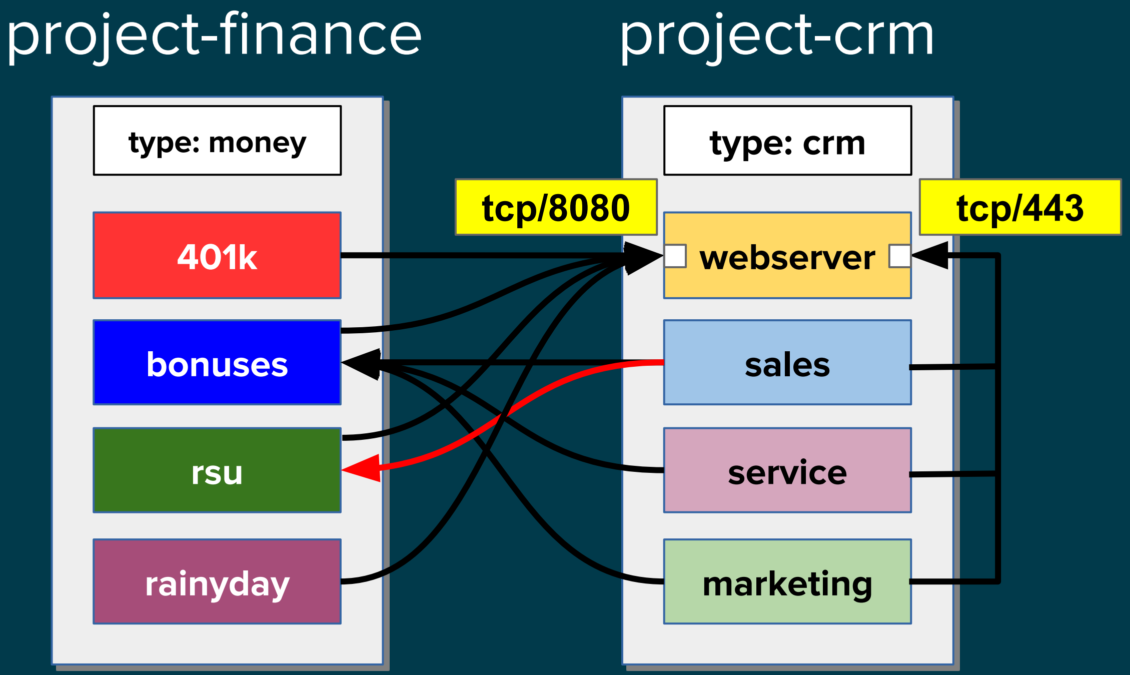
jboss-pod
10.129.0.2:8080

hrm-pod
10.129.0.23:8443

OVS - MULTITENANT



NETWORK POLICY: FINE GRAINED ISOLATION



project-finance

type: money

401k

bonuses

rsu

rainyday

project-crm

type: crm

tcp/8080

tcp/443

webserver

sales

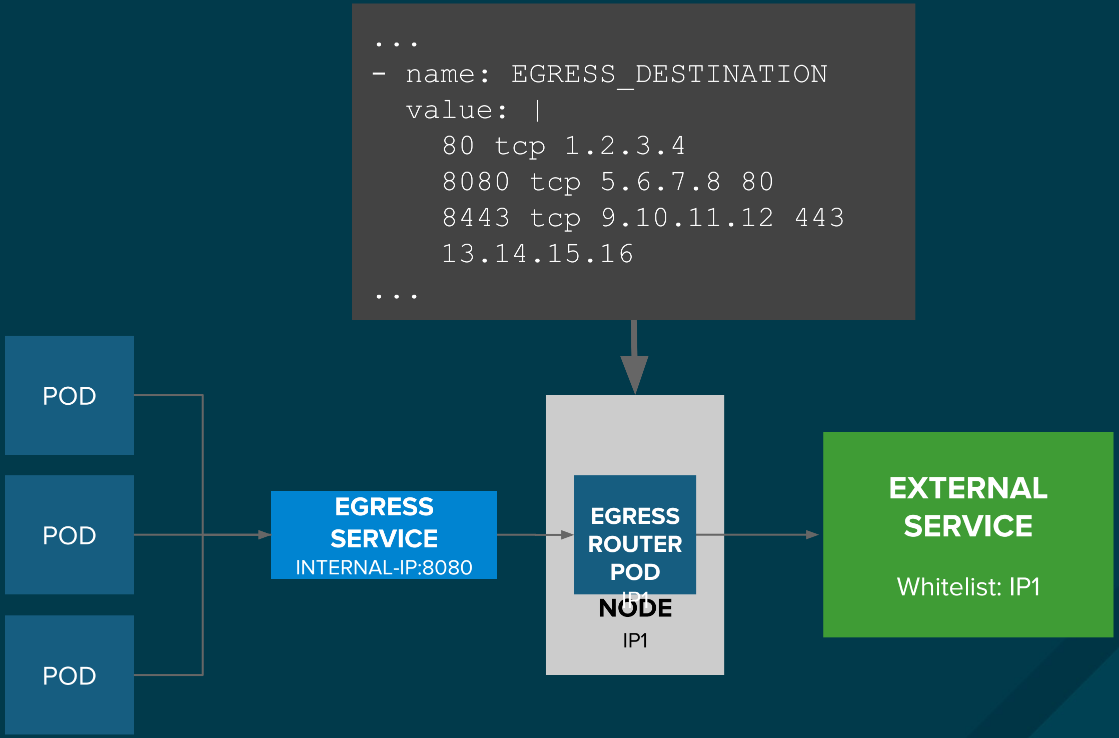
service

marketing

```
kind: NetworkPolicy
apiVersion: extensions/v1beta1
metadata:
  name: allow-to-rsu
spec:
  podSelector:
    matchLabels:
      type: rsu
  ingress:
  - {}
```

NETWORK DEFENSE: EGRESS ROUTER

The OpenShift egress router runs a service that redirects egress pod traffic to one or more specified remote servers, using a pre-defined source IP address that can be whitelisted on the remote server. The egress router can also be run as a proxy.

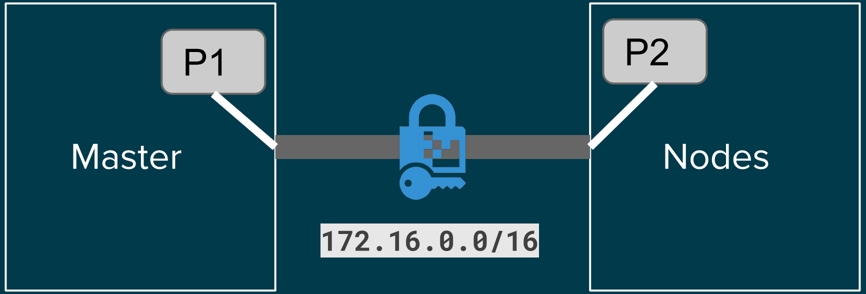


```
...
- name: EGRESS_DESTINATION
  value: |
    80 tcp 1.2.3.4
    8080 tcp 5.6.7.8 80
    8443 tcp 9.10.11.12 443
    13.14.15.16
...
```

NETWORK DEFENSE: X509 Certificates

Secures cluster communications

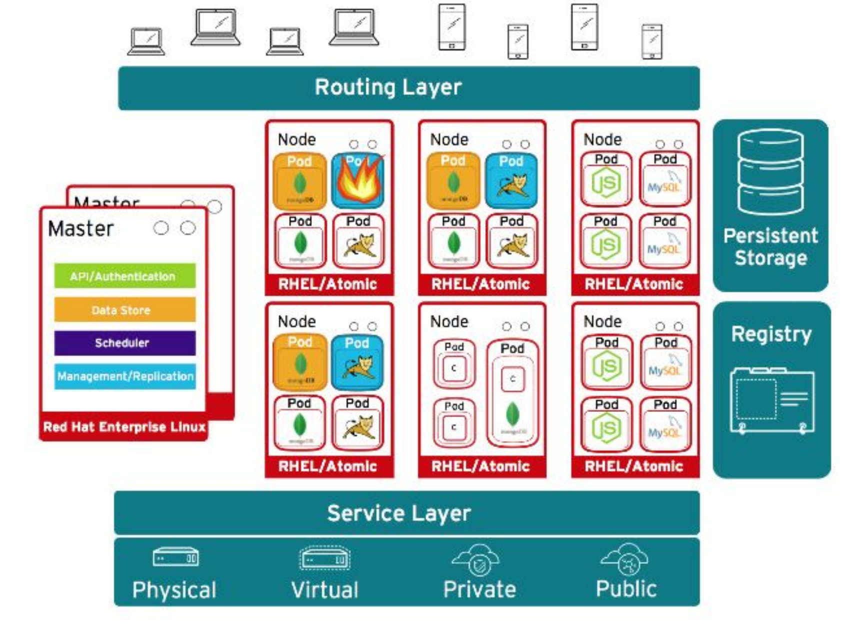
- Encryption between all Master and Node hosts (L3)
- Uses OpenShift CA and existing certificates
- Simple setup via policy defn
  - Groups (e.g. subnets)
  - Individual hosts



ATTACHED STORAGE

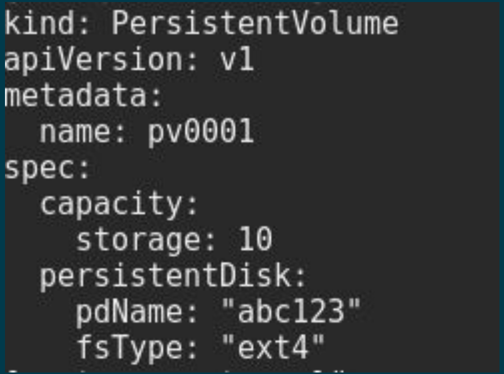
Secure storage by using

- SELinux access controls
- Secure mounts
- Supplemental group IDs for shared storage



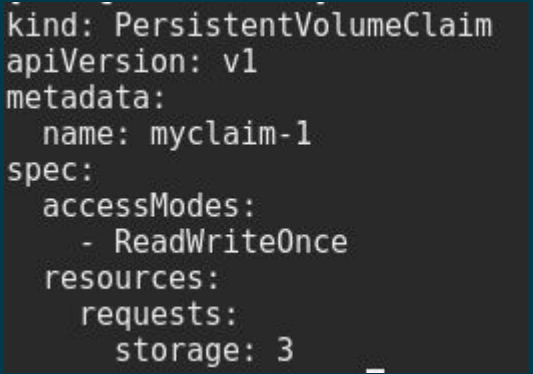
STORAGE ISOLATION

Admin provisions storage



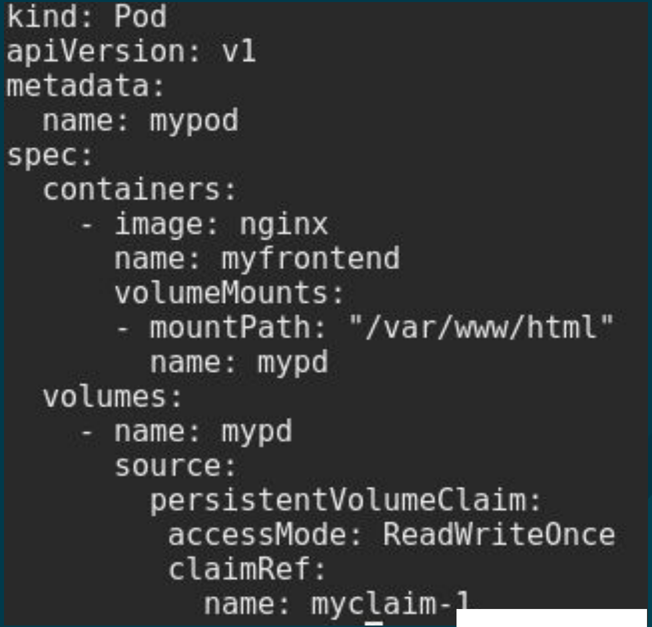
```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: pv0001
spec:
  capacity:
    storage: 10
  persistentDisk:
    pdName: "abc123"
    fsType: "ext4"
```

User requests storage



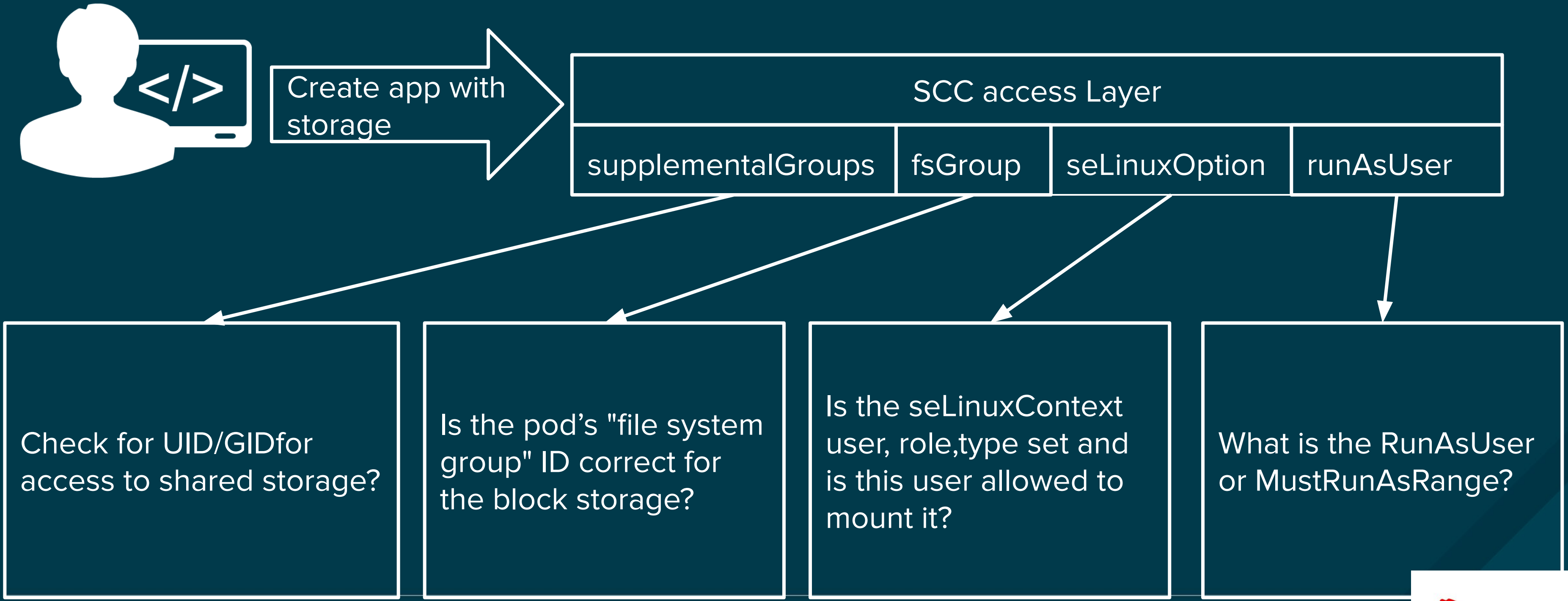
```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: myclaim-1
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 3
```

Claim usage



```
kind: Pod
apiVersion: v1
metadata:
  name: mypod
spec:
  containers:
    - image: nginx
      name: myfrontend
      volumeMounts:
      - mountPath: "/var/www/html"
        name: mypd
  volumes:
    - name: mypd
      source:
        persistentVolumeClaim:
         accessMode: ReadWriteOnce
         claimRef:
           name: myclaim-1
```

STORAGE ISOLATION



Create app with storage

SCC access Layer

supplementalGroups

fsGroup

seLinuxOption

runAsUser

Check for UID/GIDfor access to shared storage?

Is the pod’s "file system group" ID correct for the block storage?

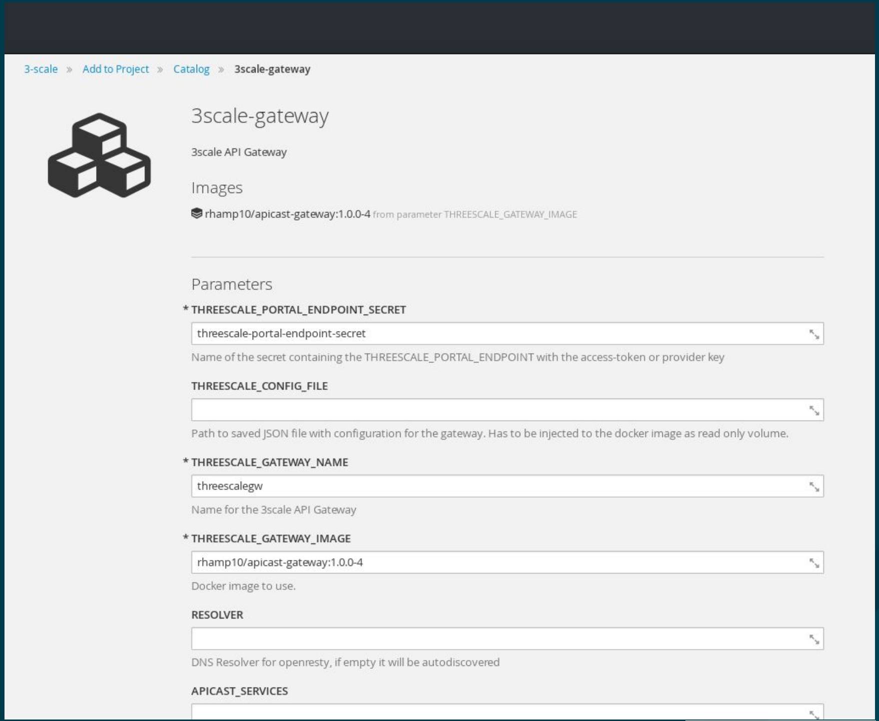
Is the seLinuxContext user, role,type set and is this user allowed to mount it?

What is the RunAsUser or MustRunAsRange?

API MANAGEMENT

Container platform & application APIs

- Authentication and authorization
- LDAP integration
- End-point access controls
- Rate limiting



3-scale » Add to Project » Catalog » 3scale-gateway

3scale-gateway

3scale API Gateway

Images

rhamp10/apicast-gateway:1.0.0-4 from parameter THREESCALE_GATEWAY_IMAGE

Parameters

* THREESCALE_PORTAL_ENDPOINT_SECRET

threescale-portal-endpoint-secret

Name of the secret containing the THREESCALE_PORTAL_ENDPOINT with the access-token or provider key

THREESCALE_CONFIG_FILE

Path to saved JSON file with configuration for the gateway. Has to be injected to the docker image as read only volume.

* THREESCALE_GATEWAY_NAME

threescalegw

Name for the 3scale API Gateway

* THREESCALE_GATEWAY_IMAGE

rhamp10/apicast-gateway:1.0.0-4

Docker image to use.

RESOLVER

DNS Resolver for openresty, if empty it will be autodiscovered

APICAST_SERVICES

EXTEND

Leverage the Ecosystem

THE SECURITY ECOSYSTEM

For enhanced security, or to meet existing policies, integrate with enterprise security tools, such as

- Identity and Access management / Privileged Access Management
- External Certificate Authorities
- External Vaults / Key Management solutions
- Container content scanners & vulnerability management
- Container runtime analysis & intrusion detection
- Security Information and Event Monitoring (SIEM)

Aporeto AquaSecurity Avi Networks big switch Black Duck Cisco Contiv Contrail dynatrace

f5 JFrog, Inc. HashiCorp NeuVector NGINX nuagenetworks Portworx

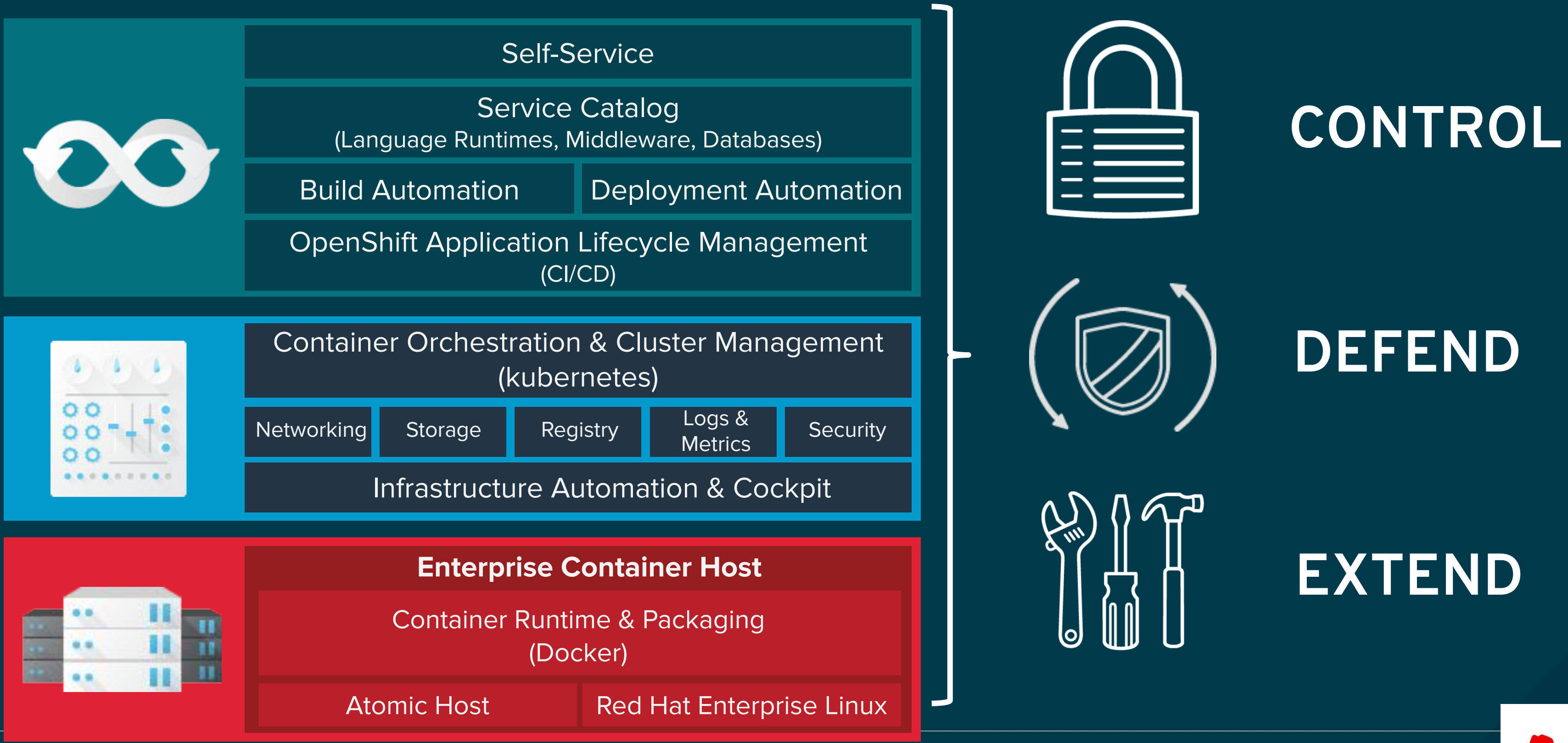
Signal Sciences Sonatype Sysdig Thales e-Security Tigera Treasure Data Tremolo

OPENSHIFT PRIMED PARTNERS

RED HAT OPENSHIFT Container Platform

BRINGING IT ALL TOGETHER

RED HAT OPENSHIFT Online



Self-Service

Service Catalog
(Language Runtimes, Middleware, Databases)

Build Automation | Deployment Automation

OpenShift Application Lifecycle Management
(CI/CD)

Container Orchestration & Cluster Management
(kubernetes)

Networking | Storage | Registry | Logs & Metrics | Security

Infrastructure Automation & Cockpit

Enterprise Container Host

Container Runtime & Packaging
(Docker)

Atomic Host | Red Hat Enterprise Linux

CONTROL

DEFEND

EXTEND

Red Hat

ADDITIONAL RESOURCES

Ten Layers of Container Security

Openshift Security Guide

Container Image Signing Integration Guide

Thank you !