What’s New in Red Hat
OpenShift Container Platform 3.11

Product Management Team
October 2018
OpenShift Commons Briefing
https://commons.openshift.org
First release of OCP with Admin-focused Console. Including visibility into Nodes, RBAC/access control, cluster-wide event feeds and more.

Operator Framework driven application services. Helm operator for expanded ecosystem content. APB integration with Ansible Galaxy.

Prometheus with default alerts for the cluster and Grafana dashboards for more detail about important metrics.
<table>
<thead>
<tr>
<th>CONTAINER</th>
<th>CONTAINER</th>
<th>CONTAINER</th>
<th>CONTAINER</th>
<th>CONTAINER</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELF-SERVICE</td>
<td>SERVICE CATALOG (LANGUAGE RUNTIMES, MIDDLEWARE, DATABASES, …)</td>
<td>BUILD AUTOMATION</td>
<td>DEPLOYMENT AUTOMATION</td>
<td></td>
</tr>
<tr>
<td>APPLICATION LIFECYCLE MANAGEMENT (CI / CD)</td>
<td>CONTAINER ORCHESTRATION &amp; CLUSTER MANAGEMENT (KUBERNETES)</td>
<td>NETWORKING</td>
<td>STORAGE</td>
<td>REGISTRY</td>
</tr>
<tr>
<td>INFRASTRUCTURE AUTOMATION &amp; COCKPIT</td>
<td>OCI CONTAINER RUNTIME &amp; PACKAGING</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOMIC HOST / RED HAT ENTERPRISE LINUX</td>
<td></td>
<td>Physical</td>
<td>Virtual</td>
<td>Private</td>
</tr>
</tbody>
</table>
Users have a choice of experience based on their role or technical abilities

- Admin/CaaS experience with heavy exposure to Kubernetes
- AppDev/PaaS experience with standard OpenShift UX
- Sessions are not shared across the Consoles but credentials are
- Both hosted on cluster, in openshift-console and openshift-webconsole namespaces
Expanded ability to manage and troubleshoot cluster Nodes

- Node status events are extremely helpful in diagnosing resource pressure and other failures
- Runs node-exporter as DaemonSet on all nodes, with a default set of scraped metrics from kube-state-metrics project
- Protected via RBAC so metrics aren’t world read-able
- Cluster-reader and above can view metrics.
CONTAINERS AS A SERVICE

View, edit and delete the full range of Kubernetes objects

- Networking:
  - Routes and Ingress
- Storage:
  - PVs and PVCs
  - Storage Classes
- Admin:
  - Projects and Namespaces
  - Nodes
  - Roles and RoleBindings
  - CRDs
Visual management of the cluster’s RBAC Roles and RoleBindings

- Track down users and service accounts with a specific Role
- View cluster-wide or namespaced bindings
- Visually audit a Role’s verbs and objects

Project admins can self-manage roles and bindings scoped to their namespace
Cluster-wide event stream helps you debug really quickly

- **All namespaces** accessible by anyone who can list NS and Events
- **Per-namespace** accessible for all project viewers
- Optionally filter by category and object type
OPERATOR SDK

Tools to get started quickly embedding application business logic into an Operator.

- Saves you from doing the dirty work to set up scaffolding
- Help run end to end tests of your logic on a local or remote cluster
- Used by Couchbase, MongoDB, Redis and more

There will be a “getting started” guide published at launch

Embed unique operational knowledge

Package and install on OCP clusters

Couchbase

 Operators

Product Manager: Rob Szumski
Ships outside of OpenShift
Use existing Helm charts as an Operator without writing code

- No tiller running
- Operator calls Helm internal code a library
- Builds on existing cluster RBAC
- Chart is re-applied when an object is modified
- Built with the Operator SDK

Read the blog post

```bash
$ docker build --build-arg HELM_CHART=https://path-to-chart/tomcat-0.1.0.tgz --build-arg API_VERSION=apache.org/v1alpha1 --build-arg KIND=Tomcat -t quay.io/<namespace>/tomcat-operator:v0.0.1 .
$ docker push quay.io/<namespace>/tomcat-operator:v0.0.1
$ oc create -f operator.yaml
$ oc create -f tomcat.yaml
apiVersion: apache.org/v1alpha1
dergkin: Tomcat
metadata:
  name: example-app
spec:
  replicaCount: 2

$ oc get Tomcats --all-namespaces
```

<table>
<thead>
<tr>
<th>NAMESPACE</th>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
<th>RESTARTS</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>production</td>
<td>prod-1.2.7</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4d</td>
</tr>
<tr>
<td>staging</td>
<td>staging-v1.2.8</td>
<td>1/1</td>
<td>Running</td>
<td>1</td>
<td>2h</td>
</tr>
</tbody>
</table>
Golang SDK for building advanced Operators

- Full power of the tools curated by the Kubernetes upstream dev community
- Suited towards ISVs and development teams (vs ops teams)
- Most advanced Operators that exist are using Golang

```
$ oc get Tomcats --all-namespaces
NAMESPACE     NAME             READY     STATUS    RESTARTS   AGE
production    prod-1.2.7       1/1       Running   0          4d
staging       staging-v1.2.8   1/1       Running   1          2h
```

**Tomcat Object**
```
apiVersion: apache.org/v1
kind: Tomcat
metadata:
  name: prod-1.2.7
  Namespace: production
spec:
  replicaCount: 10
  maxActiveSessions: 1000
maxActiveSessions: 500
```
Run, manage and upgrade many Operators on top of the cluster

- Catalog of curated Operators
- Handles rolling updates of all Operators to new versions
- Access control (RBAC) for certain teams to use certain Operators

**Video Walkthrough**

$ openshift-ansible -i inventory playbooks/olm/config.yaml
$ oc create -f operator.csv.yaml
Catalog of Operators for customers to try out in dev preview.

- ISVs
  - Couchbase
  - Dynatrace
  - MongoDB
- Red Hat
  - AMQ Streams
  - etcd
  - Prometheus

Customers can add in their own Operators or from other ISVs post-install.
Support discovering/running APB sources published to [Ansible Galaxy](https://galaxy.ansible.com/docs/contributing/creating_apb.html) from the OpenShift Automation Broker.

**Documentation:** [https://galaxy.ansible.com/docs/contributing/creating_apb.html](https://galaxy.ansible.com/docs/contributing/creating_apb.html)

**How it works:**

- APB’s can be now be created right from `mazer` command line tool using the `init` command with the `--type apb` flag and then pushed to Ansible Galaxy.

- The APB registry must be configured to connect to Ansible Galaxy:

  ```yaml
  registry:
  - type: galaxy
    name: galaxy
    org: ansibleplaybookbundle
    url: https://galaxy.ansible.com
  ```

  - Configuration documentation: [https://github.com/openshift/ansible-service-broker/blob/master/docs/config.md#registry-configuration](https://github.com/openshift/ansible-service-broker/blob/master/docs/config.md#registry-configuration)

- Broker should now be able to discover and provision APB-based services published to Ansible Galaxy and also make them available in the service catalog.
What’s new for 3.11:

1. **Broker support for authenticated registries** *(registry.redhat.io)*
   a. Uses cluster-wide settings for registry auth credentials as default for broker
   b. To configure credentials, oreg_auth_user and oreg_auth_password need to be defined in the inventory file

2. **Service Catalog access control so system level APB’s are only shown in specific namespaces**
   a. Broker now confirms to the Open Service Broker specification where they can be registered with the service catalog as either a cluster-scoped ClusterServiceBroker, or a namespace-scoped ServiceBroker kind.
   b. Depending on the broker's scope, its services and plans will be available to the entire cluster, or scoped to a specific namespace.
   c. When installing the broker, the broker_kind argument can either be set as ServiceBroker (namespace specific) or ClusterServiceBroker (cluster-wide)
BUILDS & JENKINS

- Jenkins Plugin: Stability improvements with upgrades to Java client libraries

- Jenkins: updated to 2.121.3

- Builds: Use ConfigMap for input
  `oc new-build --build-config-map=mysecrets`

- Builds: Ability to set image change triggers, without immediately triggering builds
CDK 3.6

- OpenShift Container Platform v3.10.45 (and instructions for 3.11)
- Based on Minishift 1.24

Minishift 1.24

- Configuration used to start a profile is not saved
- Provide a way to modify the kube-apiserver config same as openshift-apiserver.
- Do not apply templates in xpaas addon one by one
- Local proxy server to handle proxy issues. (technology preview)

kubectl

- We always shipped kubectl for Linux on the master’s file system, but now we will offer it in the oc client downloads
OCI Distribution Spec: Catalog API

- Part of [OCI distribution spec](#)
- OCP internal registry now fully supports the `/v2/_catalog` endpoint primarily to list repositories / imageStream names
- Required by 3rd party integrations (Twistlock, Aquasec, SAP)
- Pagination implemented (excluding pagination support for tags list)

**Note:** Requires that the authenticated user has 'list' rights on 'imagestreams' in the whole cluster!

Bulk ImageStream Tag Mirroring

- imageStream tags can be mirrored to multiple other registries (`oc image mirror`)

**Transferring Images**

To move images from one Docker registry to another, use the `oc image mirror` command. The images are streamed from registry to registry without being stored locally.

For example, to copy an image from Docker Hub to the integrated registry use the following command:

```
$ oc image mirror docker.io/library/busybox:latest 172.30.0.0/16/myproject/toybox:latest
```

It is possible to copy an image to several locations at once. To do this, you must specify several destinations:

```
$ oc image mirror 172.30.0.0/16/myproject/busybox:latest docker.io/myrepository/busybox:stable docker.io/myrepository/toybox:dev
```

`oc image mirror` runs locally, not inside the OKD cluster. Therefore, `oc image mirror` must have access to the source and destination registry.

Product Manager: Dirk Herrmann
New Registry Metrics

- Exposed to Prometheus
- Shown in OpenShift console
- Sample Metrics
  - HTTP metrics (request/response size)
  - Registry (BlobStore Create, StorageDriver Error counts, Digest Cache Requests, etc.)

Less verbose default logging

- Default has been “debug” level
- Now: “info” level

Quay Core Specification Draft (Replacement)

- While Red Hat Quay will remain a standalone product a light-weight version of Quay (“Quay Core”) will replace the OCP internal registry
- Goal is to provide feature parity as of today
- This will exclude several Quay features (geo-replication, Builds, notifications,...)
- Targeted for second half of CY2019

Going to be deprecated in 3.11 and eventually removed in future versions:

- Standalone Registry Installer
- Registry Console (Cockpit)
- `oc adm registry` command
**Quay v3** (targeted for e/o Nov’18)

- OCI distribution spec v2_2
  - Manifest list support
  - Required for multi-arch (including for Windows Images)
  - Required to store arbitrary mime-types
- Quay v3 release
  - Rebranded (Logo, UI)
  - RHEL base image

**Clair v3** (planned to be released with Quay v3)

- v3 gRPC API and HTTP/JSON via gRPC GW
- A new immutable data model
  - Images can now be scanned by multiple namespaces (OS + programming language)
  - Support for programming languages
  - Support to run Clair offline / disconnected
- Supported k8s deployment (via Helm) for Clair

**SKUs, Collateral & Docs**

- **New SKUs** available for direct & channel
- Quay datasheet & whitepaper, Docs moved to Cust Portal

*Note: Red Hat Quay releases are not tied to OpenShift releases (standalone product) | ** Clair has independent release cycles, too

**Product Manager: Dirk Herrmann**

Generally Available
Clustered Container Infrastructure

Applications Run Across Multiple Containers & Hosts

CONTAINER ORCHESTRATION & CLUSTER MANAGEMENT
(KUBERNETES)

NETWORKING | STORAGE | REGISTRY | LOGS & METRICS | SECURITY

OCI CONTAINER RUNTIME & PACKAGING

ATOMIC HOST / RED HAT ENTERPRISE LINUX

Physical | Virtual | Private | Public

Product Manager: Mike Barrett
Enable elasticity for your application workload with AWS auto-scaler

Documentation: [https://docs.openshift.com/container-platform/3.11/admin_guide/cluster-autoscaler.html](https://docs.openshift.com/container-platform/3.11/admin_guide/cluster-autoscaler.html) (not live yet)

How it Works:

- OCP auto-scaler repeatedly checks to see how many pods are pending node allocation and ensures that enough nodes are active to run your pods and that the number of active nodes is proportional to current demand.
- Requires the following assets to be previously configured:
  1. **AWS Auto Scaling Group (ASG):** Logical representation of a set of machines
     - Configured with minimum, maximum and desired number of machines per group. *(Configured from AWS CLI)*
  2. **AWS Launch Configurations (LC):** A template that an ASG uses to launch instances
     - You specify information such as the AMI ID, instance type (e.g. *m4.large*), key pair, one or more security groups, or subnets. *(Configured from AWS CLI)*
  3. **Primed (Golden) Image:** Image that runs when a new instance is provisioned
     - Bootstrapped node image that is capable of automatically joining the cluster (without any manual intervention) once provisioned by the ASG’s LC
     - ‘build_amr.yml’ playbook can be used to produce a primed image, which can be utilized by the auto-scaler.
- Instance running the CA must have permissions to scale up/down
- Auto-scaler must manually be deployed as described in the OpenShift documentation
**SUPPORT FOR ANSIBLE 2.6**

*openShift-ansible* now supports Ansible 2.6 for both installation OCP 3.11 and upgrading from OCP 3.10

**Documentation:** [https://docs.openshift.com/container-platform/3.11/getting_started/install_openshift.html#set-up-repositories](https://docs.openshift.com/container-platform/3.11/getting_started/install_openshift.html#set-up-repositories) *(not yet live)*

**How it Works:**

- Minimum version of Ansible required for OCP 3.11 will now be 2.6.x to run playbooks
- On both master and node, use `subscription-manager` to enable the repositories that are necessary in order to install OpenShift Container Platform using Ansible 2.6. For example:

```
$ subscription-manager repos --enable="rhel-7-server-rpms" \
    --enable="rhel-7-server-extras-rpms" \
    --enable="rhel-7-server-ose-3.11-rpms" \
    --enable="rhel-7-server-ansible-2.6-rpms"
```
Ansible configuration has been updated to ensure OpenShift installations are logged by default

RFE: https://bugzilla.redhat.com/show_bug.cgi?id=1458018

How it Works:

- Ansible configuration parameter `log_path` is now defined
- Requires users to be in the `/usr/share/ansible/openshift-ansible` directory prior to running any playbooks

```bash
# cd /usr/share/ansible/openshift-ansible
# pwd
/usr/share/ansible/openshift-ansible
# grep -C1 log_path ansible.cfg
[defaults]
# Set the log_path
log_path = ~/openshift-ansible.log
# ansible-playbook -i /tmp/inventory-host-file playbooks/openshift-checks/pre-install.yml
# ls -l ~/openshift-ansible.log
-rw-r--r--. 1 root root 75873 Aug  3 04:27 /root/openshift-ansible.log
```
**Release:** ocpsupplemental-3.11 (4-6 weeks after 3.11 GA)

Since 3.10, Reference Architecture Implementation guides are now part of the OpenShift product documentation ([https://docs.openshift.com](https://docs.openshift.com)).

Documentation for deploying OCP 3.11 on: *(not live yet)*

- [OpenShift 3.11 on Red Hat OpenStack Platform (RHOSP)]
- [OpenShift 3.11 on Amazon Web Services (AWS)]
- [OpenShift 3.11 on Microsoft Azure]
- [OpenShift 3.11 on VMware vSphere]
- [OpenShift 3.11 on Google Cloud Platform (GCP)]
- [OpenShift 3.9 on Red Hat Virtualization 4 (RHV)] *(update in progress)*
# ROUTER (HAproxy) ENHANCEMENTS

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HTTP/2</strong></td>
<td>Implements HAproxy router HTTP/2 support (terminating at the router).</td>
<td><code>$ oc set env dc/router ROUTER_ENABLE_HTTP2=true</code></td>
</tr>
</tbody>
</table>
| **Performance**        | Increase the number of threads that can be used by HAproxy so that we can serve more routes. | Scale down the default router and create a new router using 2 threads:  
                         |                                                                             | `$ oc scale dc/router --replicas=0`                                      |
|                        |                                                                             | `$ oc adm router myrouter --threads=2                                   |
|                        |                                                                             | `--images='openshift3/ose-haproxy-router:v3.x'`                         |
|                        |                                                                             | Set a new thread count (e.g. 7) for the HAproxy router:                 |
|                        |                                                                             | `$ oc set env dc/myrouter ROUTER_THREADS=7`                             |
| **Dynamic changes**    | Implement changes to the HAproxy router without requiring a full router reload. | `$ oc set env dc/router ROUTER_HAPROXY_CONFIG_MANAGER=true`            |
| **Client SSL/TLS cert validation** | Enable mTLS for route support of older clients/services that don’t support SNI, but where certificate verification is a requirement. | `$ oc adm router myrouter --mutual-tls-auth=optional --mutual-tls-auth-ca=/root/ca.pem --images="$image" |
| **Logs captured by aggregated logging/EFK** | Collect access logs so Operators can see them. Longer term: send to ES for users to access logs for their respective projects. | Create a router with an rsyslog container:  
                                                                 |                                                                 | `$ oc adm router myrouter --extended-logging --images='xxxx'`          |
|                        |                                                                             | Set log level:                                                         |
|                        |                                                                             | `$ oc set env dc/myrouter ROUTER_LOG_LEVEL=debug`                      |
|                        |                                                                             | Check the access logs in the rsyslog container:                        |
|                        |                                                                             | `$ oc logs -f myrouter-x-xxxxx -c syslog`                              |
In **3.10.1**, basic active/backup HA for project/namespace egress IPs was added by allowing a namespace to have multiple egress IPs hosted on different cluster nodes.

The first listed egress IP will be used by default (if available) until that node stops responding, upon which other nodes will switch to using the next listed egress IP, etc. This solution requires ≥2 IPs.

If/When the original IP eventually comes back, nodes will switch back to using the original egress IP.

More information: [Trello](#), [GitHub](#), [Docs](#)
Summary:
Projects/Namespaces are automatically allocated a single egress IP on a node in the cluster and that IP is automatically migrated from a failed node to a healthy node.

More information:
- Trello: [here](#) and [here](#)
- GitHub: [here](#) and [here](#)
Summary:
OpenShift's SDN overlay VXLAN port is now configurable (default is 4789).

Motivation:
VMware modified the VXLAN port used in their NSX SDN (≥v6.2.3) from 8472 to 4789 to adhere to RFC 7348. When running OpenShift's SDN overlay on top of VMware's NSX SDN underlay, there is a port conflict since both use the same VXLAN port (4789). With a configurable VXLAN port, customers can choose the port configuration of the two products, used in combination, for their particular environment.

How to Configure the VXLAN Port:
1. Modify the VXLAN port in master-config.yaml with the new port number (e.g. 4889 instead of 4789):
   `vxlanPort: 4889`
2. Delete clusternetwork and restart the master API and controller:
   `$ oc delete clusternetwork default`
   `$ master-restart api controller`
3. Restart all SDN pods in the openshift-sdn project:
   `$ oc delete pod -n openshift-sdn -l app=sdn`
4. On all nodes, allow the new port on the firewall:
   `# iptables -i OS_FIREWALL_ALLOW -p udp -m state --state NEW -m udp --dport 4889 -j ACCEPT`
Security Context Constraints have new options

- New option to prevent containers from gaining new privileges: `allowPrivilegeEscalation` and `DefaultAllowPrivilegeEscalation`

- Control which `sysctl` options can be defined in a pod spec: `forbiddenSysctls` and `allowedUnsafeSysctls`

Trello: [No New Privs](https://trello.com/invite/s/00000000000000000000000000000000)
Trello: [Add forbiddenSysctls and allowedUnsafeSysctls options to SCCs](https://trello.com/invite/s/00000000000000000000000000000000)
GitHub Enterprise is now an auth provider. OAuth facilitates a token exchange flow between OpenShift Container Platform and GitHub or GitHub Enterprise.

OpenShift now supports the Security Support Provider Interface (SSPI) to allow for SSO flows on Windows. (Tech Preview)

- See Configuring Authentication and User Agent for more information.
- Also Windows Authentication Technical Overview
Prometheus Cluster Monitoring

Prometheus Cluster Monitoring is now GA, providing deep insights into an OpenShift cluster with out of the box alerts and Grafana dashboards.

How it Works:

- Installed by default with every OCP 3.11+ cluster. You can opt-out by setting the Ansible variable `openshift_cluster_monitoring_operator_install` to `false`.
  - All components are installed via the Cluster Monitoring Operator in the following order: Prometheus Operator (v0.23.2), Grafana (v5.2.1), Prometheus (v2.3.2), Alertmanager (v0.15.2), Node Exporter (0.16.0), kube-state-metrics (v1.3.1).
  - Installed into the `openshift-monitoring` namespace.
  - Operator also configures out of the box alerting rules and Grafana dashboards.
- Configure the Cluster Monitoring Operator through Ansible variables inside your inventory file. Options are:
  - Enable persistent storage and configure volume size for Prometheus and Alertmanager (by default persistent storage is disabled and metrics are stored in-memory).
  - Overwrite default Alertmanager configuration to setup, for example, your own receiver such as PagerDuty.
  - Select nodes where pods should be placed.
- Prometheus Cluster Monitoring currently collects metrics from the following sources: Cluster Monitoring Operator, Prometheus itself, Prometheus Operator, Alertmanager, Kubernetes API server, kubelets including cAdvisor, kube-schedulers, kube-state-metrics, and node-exporter.
  - You can query what metrics are being collected via the Prometheus UI using `\sum({job=~"<job-name>\"})` by `(__name__)` query and replacing `\"<job-name>\"` with any of the above targets (e.g. `\"node-exporter\"`).
The stack includes three distinct UIs: Prometheus UI for querying and plotting any metrics, Alertmanager UI to manage alerts which been fired, and Grafana to browse cluster-level dashboards.

- All UIs are accessible directly via the new admin console under the “Monitoring” menu.
- You need at least view permissions to all namespaces to be able to see the UIs, or apply the `cluster_monitoring_view` cluster role.

**Important!** Currently there is no option to add your own alerting rules or dashboards. This is a future roadmap item currently under investigation.

Alerting and how it works:

- Prometheus manages alerting rules.
- If condition on one rule is met, Prometheus fires an alert that the Alertmanager catches.
- Alertmanager deduplicates and sends notifications out to the configured receivers.
- Receivers can be configured using the Ansible role `openshift_cluster_monitoring_operator` and the `openshift_cluster_monitoring_operator_alertmanager_config` variable.
Summary of What’s New

- Elasticsearch 5 and Kibana 5 (from version Elasticsearch 2 and Kibana 4)
- Kibana dashboards can be saved and shared between users
- Fluentd to Elasticsearch pipeline robustness improvements
- Bump fluentd default memory to 756M
Integration of OCS PV Expansion into OCP Web Console

• Provides ease of use for developer/tenants to expand Persistent Volume (PVC & PV) e2e
• Online expansion for OCS PV’s backed by gluster-fs (file)
• First step towards OCP + OCS UI interface integration

*PV backed by OCS Block is not supported

How it Works
• Enable feature-gates: - ExpandPersistentVolumes=true
• Storage class tag allowVolumeExpansion: True

Product Manager: Tushar Katarki/Sudhir Prasad
**POD PRIORITY AND PREEMPTION**

- **Feature** is GA in OCP 3.11
- Pods can have priority - indicates the importance relative to others
- Pods with higher priority, when not schedulable, will evict lower priority pods
- Priority affects scheduling order of pods
- Priority and QoS interactions: Priority affects eviction order of pods; Scheduler’s preemption logic does not consider QoS
- Security best practice: High Priority Pods causing Denial-of-Service; ResourceQuota can be set per user per priority level

---

**Step 1:** Admin creates Priority Classes

```
kind: PriorityClass
metadata:
  name: high-priority
value: 1000000
globalDefault: false
description: "For XYZ service pods only."
```

**Step 2:** Pods pick a Priority Class

```
kind: Pod
metadata:
  name: nginx
spec:
  containers:
    - name: nginx
      <details omitted>
priorityClassName: high-priority
```

Create a globalDefault Priority Class

Pods with no Priority Class get globalDefault

Upgrades: Pre-existing pods don’t inherit Priority Class - priority is effectively Zero

Priority and Preemption or Preemption can be disabled
Prometheus Cluster Monitoring Sizing

Prometheus Cluster Monitoring PV Sizing:

- Set `openshift_cluster_monitoring_operator_prometheus_storage_capacity` in (ansible) inventory file based on above table; OOB default is 50 GiB
- For example: for 100 nodes with 3600 pods you will need 188 GB (because there are by default two Prometheus server pods each using a PV)
Trusted Container OS

The container host is the container engine

Product Manager: Ben Breard & Scott McCarty
OCP 3.11 HOST & INTERNALS HIGHLIGHTS

Hosts
- RHEL 7.4, 7.5+ (7.6 beta is out and will GA during the lifecycle of OCP 3.11)
- Atomic Host 7.5
  - Atomic Host is supported on 3.10 & 3.11
  - deprecation notice, as Red Hat CoreOS will be the future immutable host option.
  - *Red Hat CoreOS to be released with OCP 4.0 Beta*

Internals
- Docker 1.13
- CRI-O 1.11
- Docker-latest deprecation

Major Changes
- Default to CRI-O moved to OCP 4.0
- CRI-O, Podman, and Buildah all maturing
- OCI complaint containers are default

Security
- Rootless Podman/Buildah is a major focus
- User namespaces is a major focus

Performance
- eBPF enabled kernel in RHEL 7.6
- Major Podman/Buildah work on FUSE based image caching (will enable rootless builds)
- Minor bug fixes to docker 1.13

Product Manager: Ben Breard & Scott McCarty
Generally Available
CRI-O is an alternative to the docker container engine which is also OCI compliant meaning users can run their existing images without worry, while at the same time improving compatibility, security and performance with Kubernetes:

CRI-O brings:

- Bridges CRI and OCI / Docker image
- Open source, open governance
- Lean, minimal, and stable
- Excellent security, scale, performance
- Familiar operational tooling and commands

Improvements include:

- Becoming the default for work with partners
- Crictl for node debugging and troubleshooting
- Podman for image tagging & management
- Continues to mature with OpenShift online, customer, and community deployments

Generally Available

CRI-O v1.11

Product Manager: Ben Breard & Scott McCarty
Buildah 1.2 is available in RHEL 7.5 and adds support for multi-stage builds and initial user namespaces work.

Buildah is a daemon-less tool for building and modifying OCI / Docker images.

- Can build scratch containers or use existing dockerfiles
- Allows fine-grain control over image layers, the content, and commits
- Utilities on the container host can optionally be called for the build (powerful)
- Shares the underlying image and storage components with CRI-O

Generally Available

Product Manager: Ben Breard & Scott McCarty
Podman is planned to GA with RHEL 7.6.

A daemon-less CLI/API for running, managing, and debugging OCI containers and pods

- Fast and lightweight
- Leverages runC
- Provides a “docker-like” syntax for working with containers
- Remote management API via Varlink
- Provides systemd integration and advanced namespace isolation

Product Manager: Ben Breard & Scott McCarty
Container-native Virtualization enables OpenShift to host virtual machines side by side with application containers.

- Remains available only to a targeted set of customers as a developer preview (UNSUPPORTED).
- Exposes a number of new custom resources to support virtual machines (VirtualMachine, VirtualMachineInstance, DataVolume).
- Based on the upstream KubeVirt (http://kubevirt.io) project.
- Supports basic guest creation and connectivity (serial console, vnc), Host assisted cloning, basic performance options (CPU modes, hugepages, overcommit).
- Primarily CLI/API driven today, no/limited UI.

- Not a drop-in VMW replacement (for this, focus on using the Infrastructure Migration Solution and RHV/OSP).
- Currently best suited to customers buying into OpenShift vision who still need to run a handful of virtual machines as part of their application.
- Requires bare-metal cluster(s), currently on-prem only.
- Formal dev preview enrollment process coming, for now please enquire with cnv-product-team@redhat.com.
# NOTICE OF DEPRECIATION

<table>
<thead>
<tr>
<th>Deprecated in OCP 3.X</th>
<th>Replaced by in OCP 4.X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hawkular, Cassandra, Heapster</td>
<td>Prometheus, Metrics-Server</td>
</tr>
<tr>
<td>Atomic Host</td>
<td>Red Hat CoreOS</td>
</tr>
<tr>
<td><code>oc adm diagnostics &amp; registry</code></td>
<td>Operator status &amp; Registry Operator</td>
</tr>
<tr>
<td>F5 Router Plugin</td>
<td>F5 Router Controller</td>
</tr>
<tr>
<td>Custom Docker Build Strategy on Builder Pods</td>
<td>Replace docker calls with podman and buildah</td>
</tr>
<tr>
<td>Cockpit and Standalone Atomic Registry</td>
<td>Red Hat Quay (separate product)</td>
</tr>
<tr>
<td>DNSmasq</td>
<td>CoreDNS</td>
</tr>
<tr>
<td>CFME OpenShift Provider &amp; Podified CFME</td>
<td>Prometheus</td>
</tr>
<tr>
<td>Install &amp; Upgrade</td>
<td>Converged Install of an Operator Enabled Platform</td>
</tr>
</tbody>
</table>
**OPENSHIFT ROADMAP**

**OpenShift Container Platform 4.0 (Jan)**
- Kubernetes 1.12 and CRI-O default
- Converged Platform
- Full Stack Automated Installer (AWS, OSP VM’s)
- Over the Air Updates
- Windows Containers Tech Preview
- UHC Portal at cloud.openshift.com for basic multi-cluster management
- Easy/Trackable Evaluations
- Red Hat CoreOS with Ignition Provisioning
- Cluster Registry
- HPA metrics from Prometheus; HPA Custom Metric

**OpenShift Dedicated, on Azure, and Online**
- Optional BYO Cloud for OpenShift Dedicated

**Q3 CY2018**

**OpenShift Container Platform 3.11 (Oct)**
- Kubernetes 1.11 and CRI-O option
- Infra monitoring, alerting with SRE intelligence, Node Problem Detector
- Operator Lifecycle Manager (Tech Preview)
- Operator Certification Program and Red Hat Fuse Operator
- Autoscaler for AWS and P-SAP features
- New web console for developers and cluster admins
- Ansible Galaxy ASB support
- CNV (Dev Preview)
- OVN (Dev Preview for Windows)
- FISMA and other Security PAGs
- Pod priority and Preemption

**OpenShift Dedicated, on Azure, and Online**
- OpenShift on Azure (Private Preview)

**Q4 CY2018**

**OpenShift Container Platform 4.1 (March)**
- Kubernetes 1.13 and CRI-O default
- Full Stack Automation (Azure)
- Istio GA
- Mobile 5.x
- Serverless (Tech Preview)
- Idling Controller
- Federated Ingress and Workload Policy
- OVN GA
- Che (Tech Preview)

**OpenShift Dedicated, on Azure, and Online**
- OpenShift on Azure GA
- OpenShift Dedicated self-service, metered billing
- OpenShift.io deploy apps to Dedicated (Tech Preview)

**Q1 CY2019**

**OpenShift Container Platform 4.0 (Jan)**
- Kubernetes 1.12 and CRI-O default
- Converged Platform
- Full Stack Automated Installer (AWS, OSP VM’s)
- Over the Air Updates
- Windows Containers Tech Preview
- UHC Portal at cloud.openshift.com for basic multi-cluster management
- Easy/Trackable Evaluations
- Red Hat CoreOS with Ignition Provisioning
- Cluster Registry
- HPA metrics from Prometheus; HPA Custom Metric

**OpenShift Dedicated, on Azure, and Online**
- Optional BYO Cloud for OpenShift Dedicated
Questions