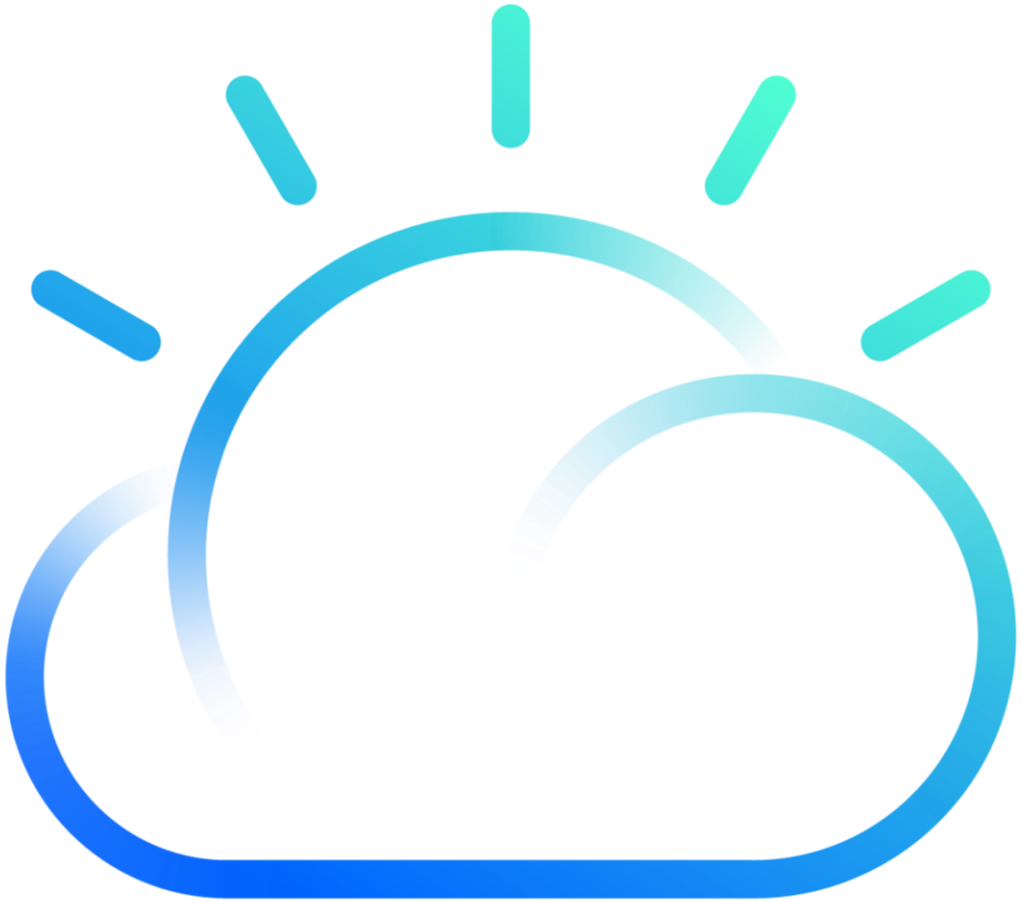
IBM

IBM Webinar:

OpenShift Deployment Review

Getting ready for Day 2 operations



Eric Kleinsorgen
Hendrik van Run
Colin Henderson

IBM Cloud Integration Expert Labs

24 September 2020

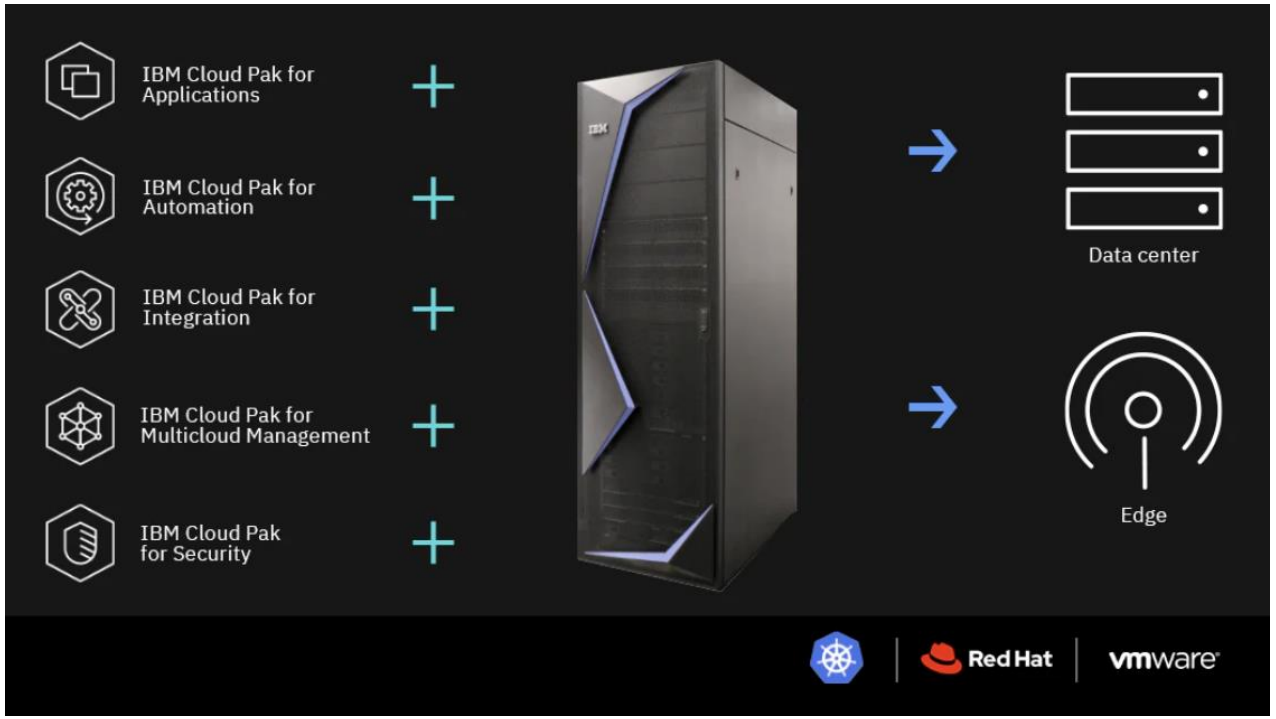
IBM Cloud

OpenShift Cluster Deployment

Day 1

IBM Cloud Pak System

- On-premises ***converged infrastructure appliance*** with built-in compute, network and storage resources
- Includes ***entitlement*** for Red Hat Enterprise Linux, IBM Spectrum Scale and VMware vSphere
- ***Automates*** deployment of IBM Cloud Paks, Red Hat OpenShift Container Platform and IBM Edge using IBM Cloud Pak System ***accelerators***



OpenShift Container Platform Accelerator

Capabilities and Versions

IBM Cloud Pak System Firmware Version	OpenShift Accelerator Version	OpenShift Version	Automation for Vertical Scaling	Automation for Horizontal Scaling	Automation for OpenShift Container Storage	Comments
2.3.1.0 *Dec 2019*	**1.0.1.0** *Dec 2019*	**3.11**	Yes	No	No	*Deprecated in IBM CPS 2.3.3.0*
2.3.2.0 *Mar 2020*	**1.0.3.0** *Mar 2020*	**4.2.18**	Yes	No	No	-
2.3.2.0 *Mar 2020*	**4.3.1.0** *May 2020*	**4.3.1**	Yes	No	No	-
2.3.3.0 *Sep 2020*	**4.4.0.0** *Sep 2020*	**4.4.6**	Yes	Yes	Yes	-

OpenShift Container Platform Accelerator
Pre-requisites

- Integration with **Red Hat Satellite Server** (RHSS) 6
  - New RHSS deployed on IBM Cloud Pak System
  - Existing RHSS reachable over the network from IBM Cloud Pak System

- Integration with **Docker Private Registry**
  - New Docker Private Registry on IBM Cloud Pak System
  - Existing Docker Private Registry reachable over the network from IBM Cloud Pak System

- Red Hat OpenShift Container Platform **BYOL binaries** for IBM Cloud Pak System
  - Available from IBM Fix Central and to be imported into IBM Cloud Pak System

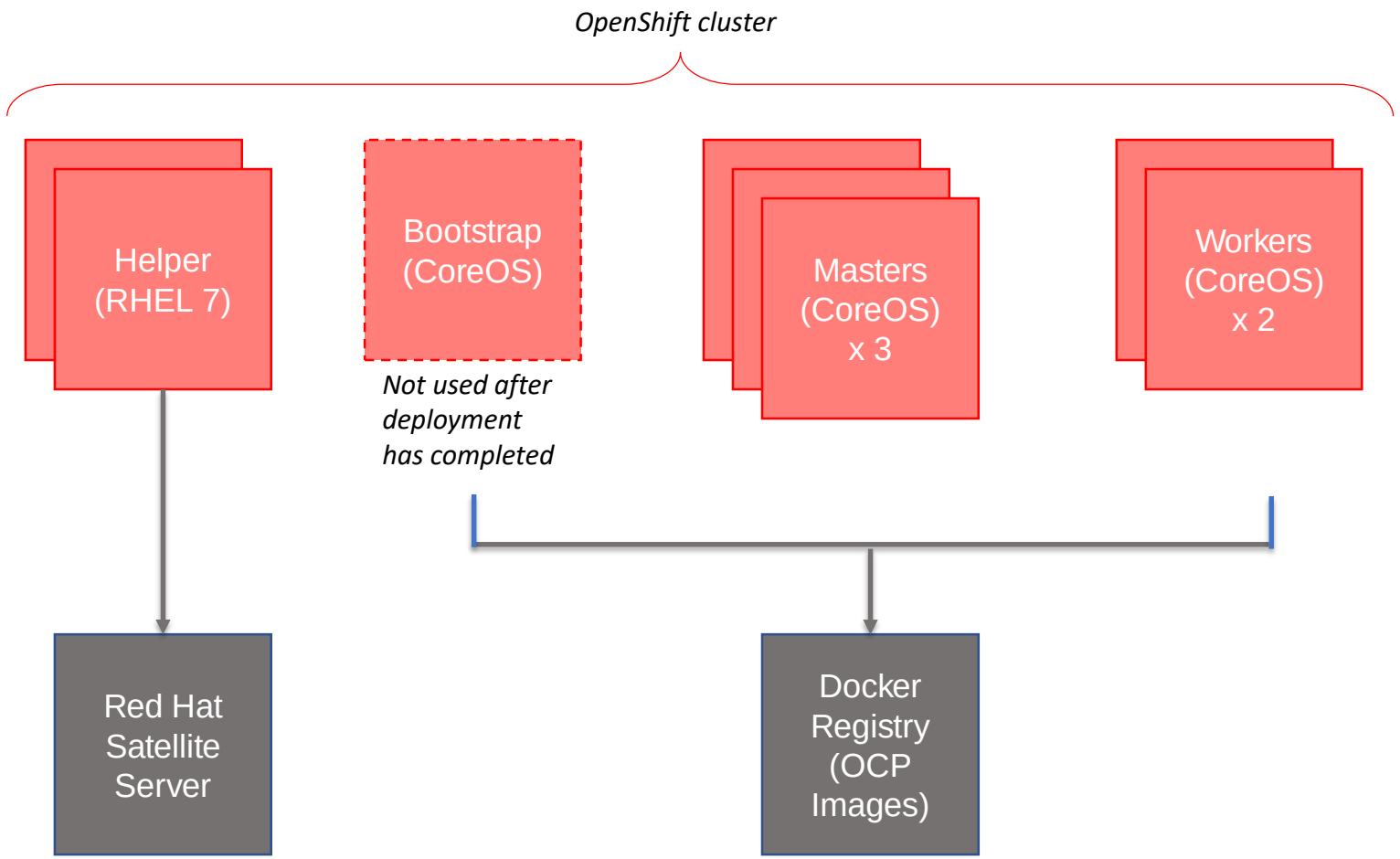
- Red Hat OpenShift **licenses/entitlement**
  - Required - Red Hat OpenShift Container Platform
  - Optional - Red Hat OpenShift Container Storage

OpenShift Container Platform Accelerator

Infrastructure Topology

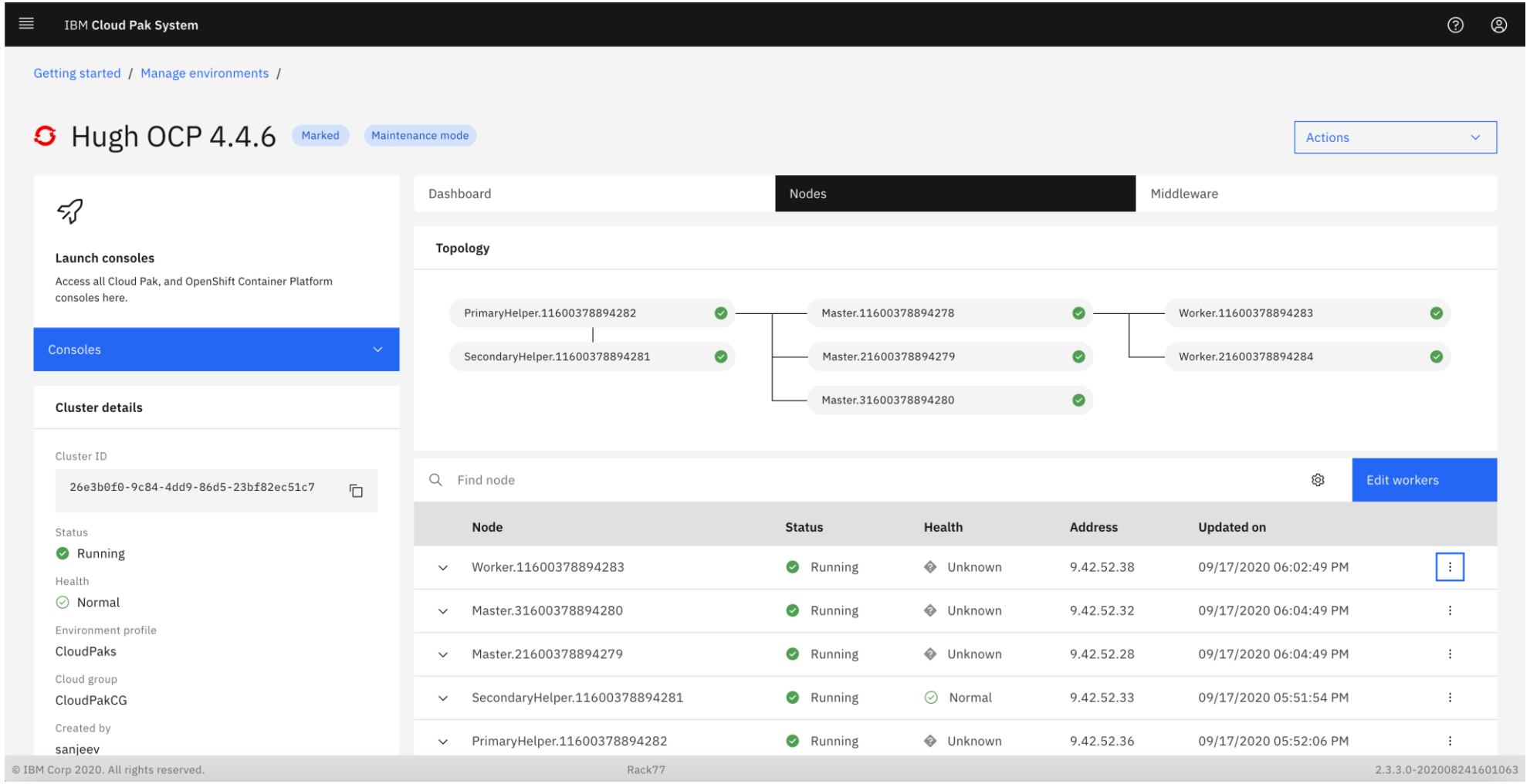
Helper Node

- Supports the automated installation and deployment of OCP
- Has a primary and secondary helper for High Availability
- Supports runtime of OCP with an integrated load balancer



OpenShift Container Platform Accelerator

Deployed OpenShift Cluster



IBM Cloud Pak System and Red Hat OpenShift
References to get started

- IBM Cloud Pak System blog
  https://ibmcloudpaksystem.home.blog/

- IBM Cloud Pak System blog list
  https://www.ibm.com/support/pages/node/6239266

- Deploying Red Hat OpenShift Container Platform 4.4 on IBM Cloud Pak System
  https://developer.ibm.com/recipes/tutorials/deploying-red-hat-openshift-container-platform-4-4-on-ibm-cloud-pak-system/

- Deploying RedHat OpenShift 4.3 on IBM Cloud Pak System - Step by step tutorial
  https://developer.ibm.com/recipes/tutorials/deploying-redhat-openshift-4-3-on-ibm-cloud-pak-system/

- Deploying Red Hat OpenShift 4 on IBM Cloud Pak System 2.3.2.0 - Step by step tutorial
  https://developer.ibm.com/recipes/tutorials/deploying-red-hat-openshift-4-on-ibm-cloud-pak-system-2-3-2-0/

- Accelerate your Red Hat OpenShift Container Platform deployment with IBM Cloud Pak System - How to easily implement and configure on-premises Kubernetes platforms
  https://developer.ibm.com/tutorials/accelerate-your-red-hat-openshift-container-platform-deployment-with-ibm-cloud-pak-system/

OpenShift Cluster Configuration

Day 2

- Connect OpenShift to LDAP
- Remove self-provisioner role
- LDAP Group Sync
- Create OpenShift groups
  - Cluster Admin Group
  - Dev groups
    - <namespace>-admins
    - <namespace>-developer
- Service Accounts
- Environment Review

Connect OpenShift to LDAP

- Fairly straight forward as described in [Red Hat OpenShift Documentation]
- Navigate to Administration → Cluster Settings → Global Configuration → OAuth
- Add LDAP identity provider
- If not using TLS, enter YAML tab and set **insecure** option to **true**
- Save configuration

Setting	**Value**	**Required**
Name	openldap	Yes
URL	ldap://ldap.acme.com:389/O=ACME.COM?uid?sub?(objectClass=*)	Yes
BindDN	uid=OpenShiftAdm,CN=Users,CN=Admin,O=ACME.COM	Yes
Bind Password	<provided by ACME>	Yes
Attribute	**Value**	
ID	uid	Yes
Preferred Username	uid	
Name	cn	
Email		

- May need to wait a few minutes for the authentication pods to restart

Connect OpenShift to LDAP

Updates/creates 2 configurations in OpenShift

```
# oc get oauth cluster -o yaml
apiVersion: config.openshift.io/v1
kind: OAuth
metadata:
  name: cluster
spec:
  identityProviders:
  - ldap:
      attributes:
        email: []
        id:
        - uid
        name:
        - cn
        preferredUsername:
        - uid
      bindDN: uid=OpenshiftAdm,CN=Users,CN=Admin,O=ACME.COM
      bindPassword:
        name: ldap-bind-password-22m6p
      insecure: true
      url: ldap://openldap.acme.com:389/O=ACME.COM?uid?sub?(objectClass=*)
    mappingMethod: claim
    name: openldap
    type: LDAP
```

```
# oc -n openshift-config get secret ldap-bind-password-22m6p -o yaml
apiVersion: v1
data:
  bindPassword: RGlkIHlvdSB0aGluayB0aGlzIHdvdWxkIGJlIGEgcmVhbCBwYXNzd29yZD8K
kind: Secret
metadata:
  name: ldap-bind-password-22m6p
  namespace: openshift-config
type: Opaque
```

OpenShift Roles

OpenShift includes a set of default cluster roles that you can assign to users and groups cluster-wide or locally.

Default Cluster Role	Description
`admin`	A project manager. If used in a local binding, an `admin` has rights to view any resource in the project and modify any resource in the project except for quota.
`basic-user`	A user that can get basic information about projects and users.
`cluster-admin`	A super-user that can perform any action in any project. When bound to a user with a local binding, they have full control over quota and every action on every resource in the project.
`cluster-status`	A user that can get basic cluster status information.
`edit`	A user that can modify most objects in a project but does not have the power to view or modify roles or bindings.
`self-provisioner`	A user that can create their own projects.
`view`	A user who cannot make any modifications, but can see most objects in a project. They cannot view or modify roles or bindings.

Remove self-provisioner role

- Once LDAP is configured users are automatically added to OpenShift when they login
- If you want to restrict the user's capability to be able to create their own projects until they are granted access by the OpenShift admin this role can be removed

```
oc adm policy remove-cluster-role-from-group self-provisioner system:authenticated:oauth
```

- To re-enable self-provisioning run the following command

```
oc adm policy add-cluster-role-to-group self-provisioner system:authenticated:oauth
```

LDAP Group Sync

- Described in detail in the [Red Hat OpenShift Documentation]
- Enables some automation to align LDAP groups with OpenShift groups
- Would need to be executed whenever LDAP groups are updated
- Not required
- Users can be added manually to groups by a cluster administrator

```
oc adm groups add-users qa-cluster-admins <user to add>
```

```
kind: LDAPSyncConfig
apiVersion: v1
url: ldap://openldap.acme.com:389
bindDN: cn=OpenshiftAdm,CN=Users,O=ACME.COM
bindPassword: ********
insecure: true
groupUIDNameMapping:
  "cn=cloudadmins,CN=Groups,O=ACME.COM": Openshift-Admins
  "cn=cloudgrp,CN=Groups,O=ACME.COM": Openshift-Users
rfc2307:
    groupsQuery:
        baseDN: "CN=Groups,O=ACME.COM"
        filter: (objectClass=groupOfNames)
        scope: sub
        derefAliases: never
        pageSize: 0
    groupUIDAttribute: dn
    groupNameAttributes: [ cn ]
    groupMembershipAttributes: [ member ]
    usersQuery:
        baseDN: "CN=Users,O=ACME.COM"
        scope: sub
        derefAliases: never
        pageSize: 0
    userUIDAttribute: dn
    userNameAttributes: [ uid ]
    tolerateMemberNotFoundErrors: false
    tolerateMemberOutOfScopeErrors: false
```

Create OpenShift groups

- Cluster Admin Group
  - Users added to this group would have **cluster admin** access rights
  - Created with the command: `oc adm groups new <group name>`
  - The group name for QA is qa-cluster-admins and for PROD is prd-cluster-admins
  - Assign **cluster-admin** role to group: `oc adm policy add-cluster-role-to-group cluster-admin qa-cluster-admins`
  - Assign user to group: `oc adm groups add-users qa-cluster-admins <user to add>`
- Dev groups
  - 2 developer group types were identified; administrators and developers
    - administrators have **admin** role at the project level
    - developers have **view** role at the project level to look at pod status and logs
    - Created with the command: `oc adm groups new <group name>`
    - Developer groups names for QA are: qa-common-admins, qa-common-developer, qa-edp-admins, qa-edp-developer
      PROD: prd-common-admins, prd-common-developer, prd-edp-admins, prd-edp-developer
    - Assign roles to groups:

```
oc project common
oc adm policy add-role-to-group admin qa-common-admins
oc adm policy add-role-to-group view qa-common-developer
oc project edp
oc adm policy add-role-to-group admin qa-edp-admins
oc adm policy add-role-to-group view qa-edp-developer
```

    - Assign user to group:

```
oc adm groups add-users qa-common-admins <user to add>
oc adm groups add-users qa-common-admins <user to add>
oc adm groups add-users qa-common-admins <user to add>
oc adm groups add-users qa-common-admins <user to add>
```

Service Accounts for CICD

- For CICD pipelines a service account with **admin** access to the project is generally sufficient
- Generate **kubeconfig** for pipelines to use for authentication:

```
oc create sa <SERVICE ACCOUNT> -n <PROJECT>
oc create rolebinding <SERVICE ACCOUNT>-admin --serviceaccount=<PROJECT>:<SERVICE ACCOUNT> --clusterrole=admin -n <PROJECT>
token_name=`oc get sa jenkins -n <PROJECT> -o template='{{ index (index .secrets 0) "name" }}'`
token=`oc get secret ${token_name} -n <PROJECT> -o template='{{ .data.token }}' | base64 -d`
oc --kubeconfig=/tmp/<SERVICE ACCOUNT>-kubeconfig.yaml \
        config set-cluster cluster --server=https://<OpenShift api URL>:6443 --insecure-skip-tls-verify=true
oc --kubeconfig=/tmp/<SERVICE ACCOUNT>-kubeconfig.yaml config set-credentials jenkins --token=${token}
oc --kubeconfig=/tmp/<SERVICE ACCOUNT>-kubeconfig.yaml \
       config set-context <SERVICE ACCOUNT>-<PROJECT> --cluster=cluster --user=<SERVICE ACCOUNT> --namespace=<PROJECT>
oc --kubeconfig=/tmp/<SERVICE ACCOUNT>-kubeconfig.yaml config use-context <SERVICE ACCOUNT>-<PROJECT>
```

- In the above example the **kubeconfig** file would be /tmp/<SERVICE ACCOUNT>-kubeconfig.yaml
- To grant the same service account access to another project:

```
oc create rolebinding <SERVICE ACCOUNT>-admin --serviceaccount=<PROJECT>:<SERVICE ACCOUNT> --clusterrole=admin -n <NEW PROJECT>
```

- You can create a service account and **kubeconfig** for each project to be consumed by the CICD pipelines. Or you can have a single service account and **kubeconfig** with access to many projects.
- When using a single service account for many projects you need to make sure the pipelines are configured to specify a project in any kubectl/oc commands.

Environment Review

The following items were identified to be addressed during a review of the OpenShift Container Platform environments:

- openshift-samples operator degraded
- Monitoring stack
  - Running on worker nodes
  - telemeterClient as disabled
  - Visible alarm for telemeterClient
- Dynamic software-defined storage provisioner
- Centralized logging
- Configure requests and limits for CPU and Memory for all the workloads

openshift-samples operator

- The "openshift-samples" cluster operator is degraded because of Internet connectivity and the disconnected install performed by IBM Cloud Pak System
- There is a knowledge center article that addresses the issue

Note

If the "openshift-samples" cluster operator is degraded because of internet connectivity disruption during the install process, then do the following steps:

a. Go to **Home** > **Search** > "config" > `samples.operator.openshift.io`.

b. Edit the YAML and change to `managementState: Removed`.

c. Save the file.

Also, for more information about the steps, see *step 5* of https://docs.openshift.com/container-platform/4.2/installing/install_config/installing-restricted-networks-preparations.html#installation-restricted-network-samples_installing-restricted-networks-preparations.

An OpenShift Container Platform upgrade can take approximately an hour to complete.

- Once this is resolved the cluster indicates that an update is available. It can be updated but may not be certified for IBM Cloud Pak System. See the FAQ for details.

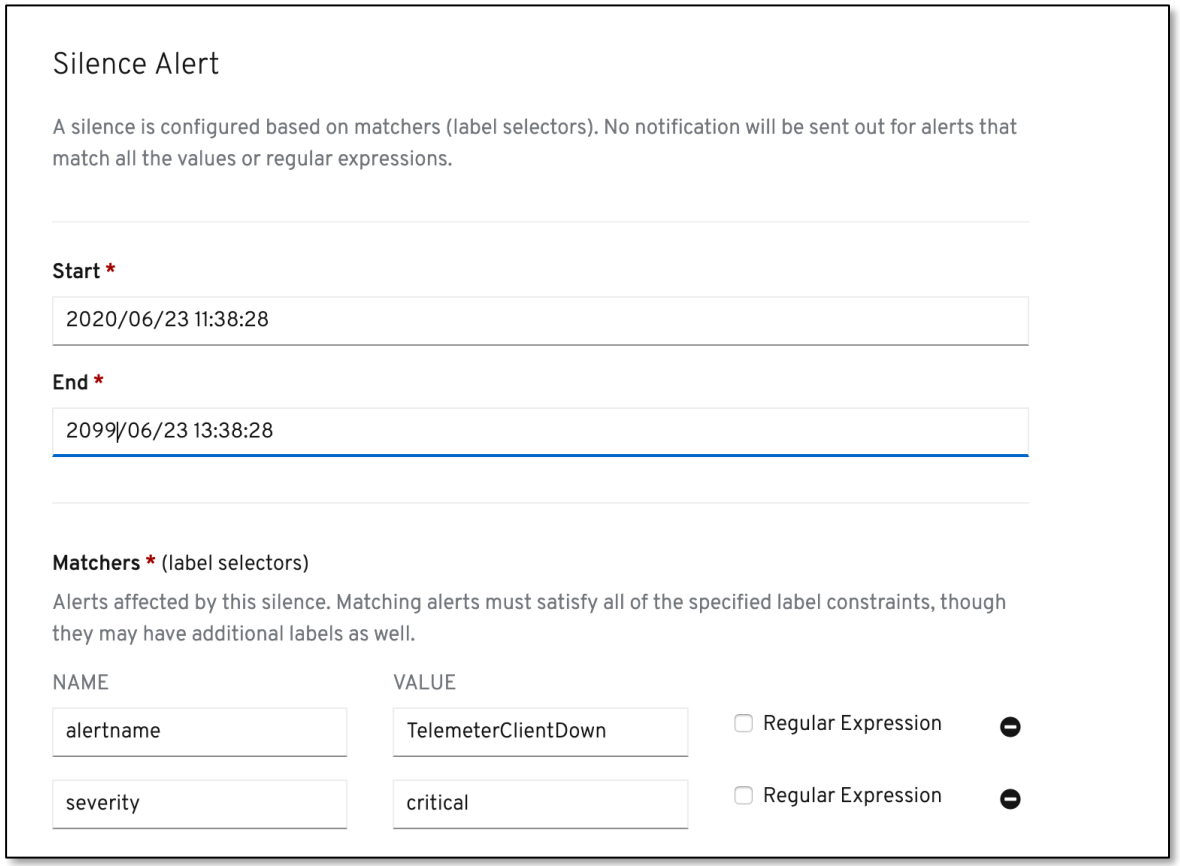
Monitoring Stack

- The monitoring stack includes Prometheus, Grafana, Alertmanager, and Telemetry.
- The IBM Cloud Pak System OpenShift installation opts out of the Telemetry service. Details about this configuration are in the Red Hat OpenShift Telemetry documentation
- The following configurations were made to the monitoring operator:
  - Disable the telemetry service
  - Silence the alarm for the telemetry service
  - Configure all the monitoring services to run on the master nodes. Can later be moved to dedicated worker nodes if needed.
- This link provides additional details for configuring the monitoring stack

Monitoring Stack Configurations

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: cluster-monitoring-config
  namespace: openshift-monitoring
data:
  config.yaml: |
    prometheusK8s:
      retention: 24h
      nodeSelector:
        node-role.kubernetes.io/master: ""
      tolerations:
      - key: node-role.kubernetes.io/master
        operator: Exists
    alertmanagerMain:
      nodeSelector:
        node-role.kubernetes.io/master: ""
      tolerations:
      - key: node-role.kubernetes.io/master
        operator: Exists
    kubeStateMetrics:
      nodeSelector:
        node-role.kubernetes.io/master: ""
      tolerations:
      - key: node-role.kubernetes.io/master
        operator: Exists
    grafana:
      nodeSelector:
        node-role.kubernetes.io/master: ""
      tolerations:
      - key: node-role.kubernetes.io/master
        operator: Exists
    telemeterClient:
      enabled: false
    k8sPrometheusAdapter:
      nodeSelector:
        node-role.kubernetes.io/master: ""
      tolerations:
      - key: node-role.kubernetes.io/master
        operator: Exists
    openshiftStateMetrics:
      nodeSelector:
        node-role.kubernetes.io/master: ""
      tolerations:
      - key: node-role.kubernetes.io/master
        operator: Exists
```

- Configures retention time for Prometheus data to 24 hours
- Configures nodeSelector for master role
- Configures a toleration to allow pods to run on master nodes
- Disables telemeterClient service
- Silence Alert for telemeterClient



Monitoring Stack Configurations

Prometheus console – Grafana

Kubernetes / Compute Resources / Cluster

Last 1 hour | 10s

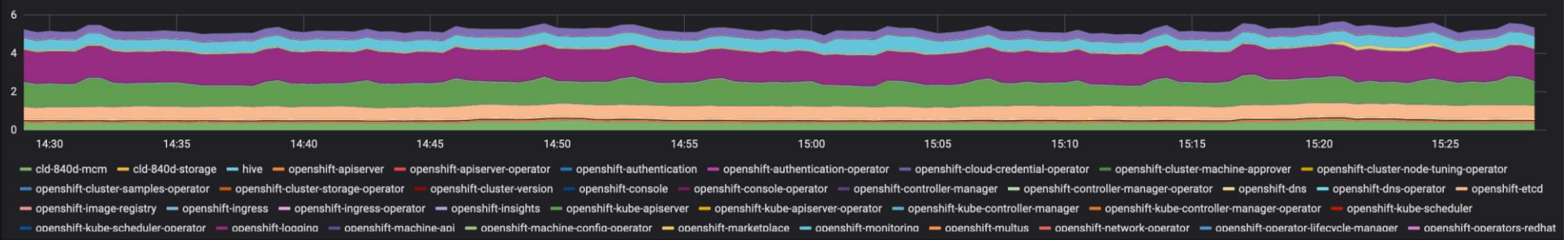
datasource: prometheus

Headlines

CPU Utilisation	CPU Requests Commitment	CPU Limits Commitment	Memory Utilisation	Memory Requests Commitment	Memory Limits Commitment
16.95%	30.16%	10.31%	19.84%	45.38%	45.79%

CPU

CPU Usage



CPU Quota

CPU Quota

Namespace	Pods	Workloads	CPU Usage	CPU Requests	CPU Requests %	CPU Limits	CPU Limits %
openshift-kube-apiserver	-	-	1.18	0.96	123.18%	-	-
openshift-kube-scheduler	-	-	0.01	0.05	20.92%	-	-

Dynamic software-defined storage

1. Starting point was manual storage provisioning on NFS
   - Manual creating Persistent Volumes and Persistent Volume Claims
2. Another option would be to leverage the NFS client provisioner
   - Uses your existing and already configured NFS server to support dynamic provisioning
3. Red Hat OpenShift Container Storage is recommended
   - Based on Red Hat Ceph® Storage
   - Provides software-defined block, file and object storage
   - Available in the Operator Hub

Centralized logging

- Cluster logging operator
- Available in the Operator Hub
- Enables EFK (Elasticsearch, Fluentd, Kibana) cluster logging components:
  - logStore – Elasticsearch: This is where the logs will be stored.
  - collection – Fluentd: Collects logs from nodes, formats them, and stores them in the logStore.
  - visualization – Kibana: UI component used to view logs, graphs, charts, and so forth.
  - curation – Curator: Trims logs by age.
- The EFK stack can be configured to forward the logs to long term storage for any customer specific retention requirements
- The curator should be configured to trim the data as needed.
  - If data is forwarded to long term storage a good rule of thumb is to remove logs older than 24 hours every 4 hours.

Centralized logging

- Elasticsearch sizing requirements
- Minimum 3 nodes with 1 CPU, 16GB RAM and 200GB persistent storage
- Persistent storage size depends on many factors

Discovery Questions

- How much raw data (GB) will you index per day?
- How many days will you retain the data?
- What is the net expansion factor of the data?

  JSON Factor * Indexing Factor * Compression Factor
- How many replica shards will you enforce?
- How much memory will you allocate per data node?
- What will be your memory:data ratio?

Constants

- Reserve +15% to stay under the disk watermarks.
- Reserve +5% for margin of error and background activities.
- Reserve the equivalent of a data node to handle failure.

Total Data (GB)	=	Raw data (GB) per day * Number of days retained * Net expansion factor * (Number of replicas + 1)
Total Storage (GB)	=	Total Data (GB) * (1 + 0.15 Disk watermark threshold + 0.05 Margin of error)
Total Data Nodes	=	ROUNDUP(Total Storage (GB) / Memory per data node / Memory:data ratio) + 1 Data node for failover capacity

Centralized logging

Visualization - Kibana console

OPENSHIFT
CONTAINER PLATFORM

Log out

Discover
Visualize
Dashboard
Timelion
Dev Tools
Management

8,950 hits

New Save Open Share Last 1w

kubernetes.pod_name:"cluster-logging-operator-7cc469dd58-txdcc" AND kubernetes.namespace_name:"openshift-logging" AND kubernetes.container_name.raw:"cluster-logging-operato Uses lucene query syntax

Add a filter +

.operations.*

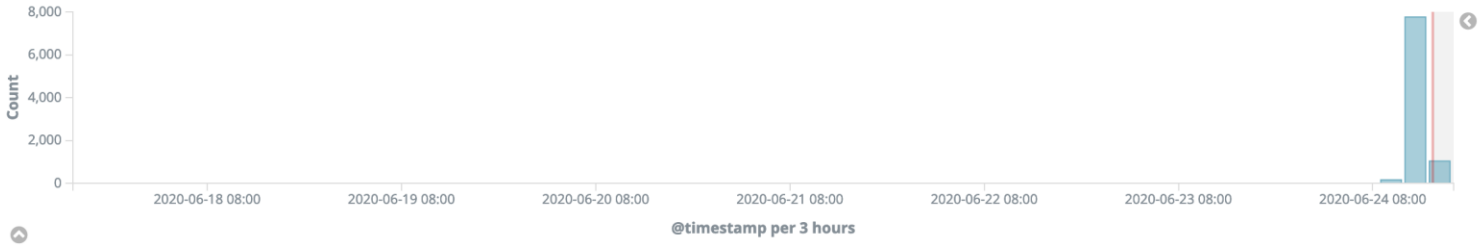
Selected Fields

t message
t kubernetes.container_n...

Available Fields

@timestamp
? _id
? _index
? _score
? _type
t docker.container_id
t hostname
? kubernetes.container_image
? kubernetes.container_imag...
t kubernetes.host
? kubernetes.labels.name
? kubernetes.labels.pod-tem...
t kubernetes.master_url

June 17th 2020, 15:25:01.389 - June 24th 2020, 15:25:01.390 — Auto

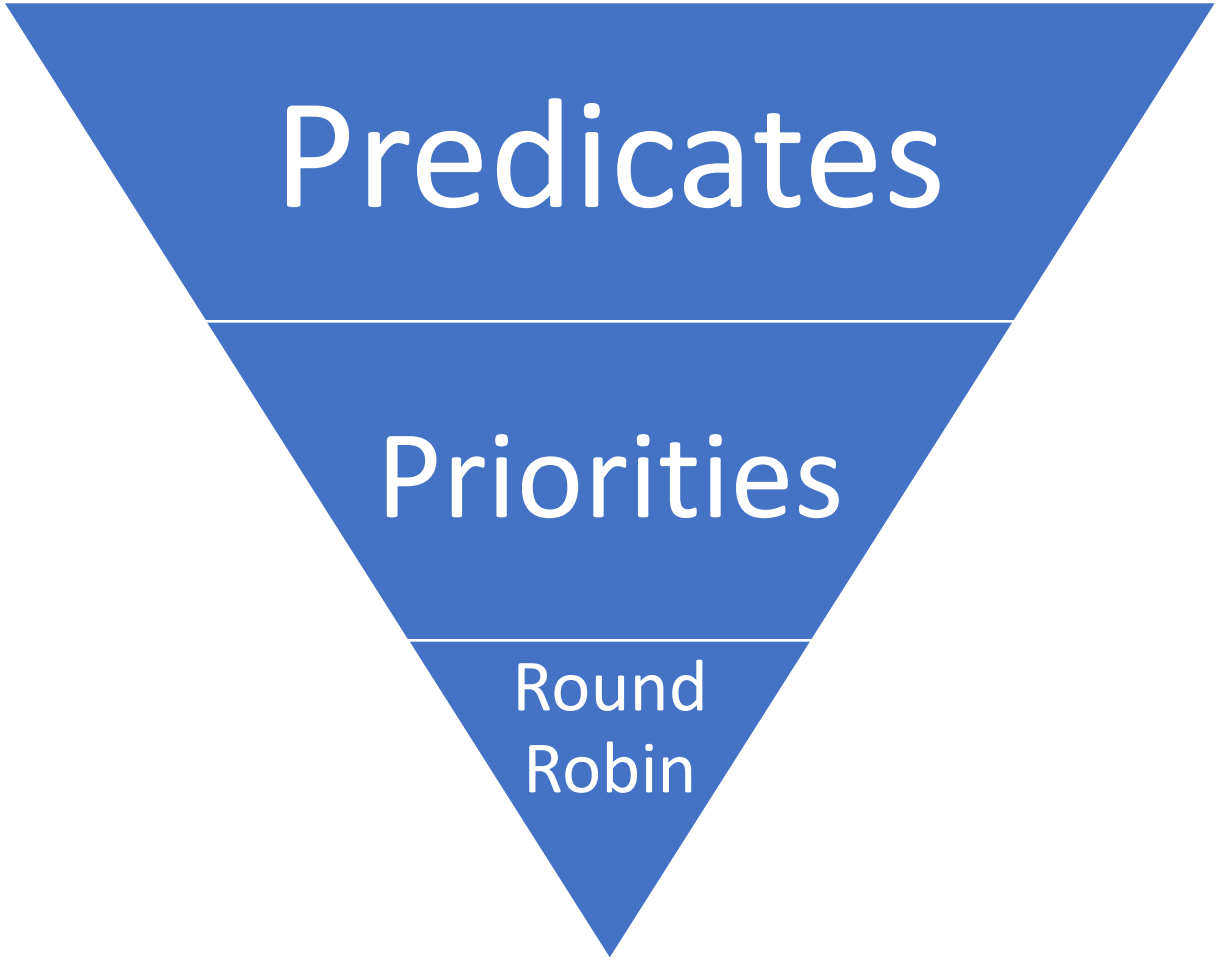


Time	kubernetes.container_name	message
June 24th 2020, 15:24:55.451	cluster-logging-operator	time="2020-06-24T19:24:55Z" level=info msg="Updating status of Curator"
June 24th 2020, 15:24:54.114	cluster-logging-operator	time="2020-06-24T19:24:54Z" level=info msg="Updating status of Curator"
June 24th 2020, 15:24:52.989	cluster-logging-operator	time="2020-06-24T19:24:52Z" level=info msg="Updating status of Curator"
June 24th 2020, 15:24:51.644	cluster-logging-operator	time="2020-06-24T19:24:51Z" level=info msg="Updating status of Curator"
June 24th 2020, 15:24:50.043	cluster-logging-operator	time="2020-06-24T19:24:50Z" level=info msg="Updating status of Curator"
June 24th 2020, 15:24:49.037	cluster-logging-operator	time="2020-06-24T19:24:49Z" level=info msg="Updating status of Curator"
June 24th 2020, 15:24:48.029	cluster-logging-operator	time="2020-06-24T19:24:48Z" level=info msg="Updating status of Curator"
June 24th 2020, 15:24:46.437	cluster-logging-operator	time="2020-06-24T19:24:46Z" level=info msg="Updating status of Curator"
June 24th 2020, 15:24:45.083	cluster-logging-operator	time="2020-06-24T19:24:45Z" level=info msg="Updating status of Curator"
June 24th 2020, 15:24:44.032	cluster-logging-operator	time="2020-06-24T19:24:44Z" level=info msg="Updating status of Curator"
June 24th 2020, 15:24:42.995	cluster-logging-operator	time="2020-06-24T19:24:42Z" level=info msg="Updating status of Curator"

Collapse

Resource Requests and Limits

Kubernetes Scheduler



Follows algorithm to select node

- Predicates are conditions that must be met to support the pod (e.g., ports, disk conflict, volume binding etc.) and returns a binary value (yes or no)
- Priorities filter the remaining nodes providing each with a score (from 1...10)
  - Affinity rules
  - Label selectors
  - Requests and limits
  - Taints and tolerations
- Round robin selection for the remaining nodes with the top score

Resource Requests and Limits

- Resource requests give the Kubernetes scheduler information to enable it to schedule pods efficiently across the available nodes.

  Try to make accurate to get more efficient pod placement

- Resource limits prevent containers from using more resources than they are allowed.
- CPU is specified in millicores, or a fraction of a CPU

  100m == 0.1 (CPU)

- Memory is specified in bytes, as an integer or as a fixed point integer with the following suffixes:

  E, P, T, G, M, K or Ei, Pi, Ti, Gi, Mi, Ki

  128974848 == 129e6 == 129M == 123Mi

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: frontend
spec:
  containers:
  - name: db
    image: mysql
    env:
    - name: MYSQL_ROOT_PASSWORD
      value: "password"
    resources:
      requests:
        memory: "64Mi"
        cpu: "250m"
      limits:
        memory: "128Mi"
        cpu: "500m"
```

64 MiB

250 millicores = ¼ core

```yaml
apiVersion: v1
kind: LimitRange
metadata:
  name: mem-limit-range
spec:
  limits:
  - default:
      memory: 512Mi
    defaultRequest:
```

Nodes and Resource Scheduling

- If there are insufficient resources available for the scheduler to fit a Pod with resource requests, it will fail to schedule, and stay in Pending state. Using *oc describe pod xxx* will show the reason:

  FailedScheduling Failed for reason PodExceedsFreeCPU and possibly others

- You can check node capacities and amounts **allocatable** with the *oc describe nodes* command:

```
kubectl describe nodes e2e-test-minion-group-4lw4
```

```
Name:            e2e-test-minion-group-4lw4
[ ... lines removed for clarity ...]
Capacity:
 cpu:                               2
 memory:                            7679792Ki
 pods:                              110
Allocatable:
 cpu:                               1800m
 memory:                            7474992Ki
 pods:                              110
[ ... lines removed for clarity ...]
Non-terminated Pods:        (5 in total)
  Namespace    Name                                  CPU Requests  CPU Limits  Memory Requests  Memory Limits
  ---------    ----                                  ------------  ----------  ---------------  -------------
  kube-system  fluentd-gcp-v1.38-28bv1               100m (5%)     0 (0%)      200Mi (2%)       200Mi (2%)
  kube-system  kube-dns-3297075139-61lj3             260m (13%)    0 (0%)      100Mi (1%)       170Mi (2%)
  kube-system  kube-proxy-e2e-test-...               100m (5%)     0 (0%)      0 (0%)           0 (0%)
  kube-system  monitoring-influxdb-grafana-v4-z1m12  200m (10%)    200m (10%)  600Mi (8%)       600Mi (8%)
  kube-system  node-problem-detector-v0.1-fj7m3      20m (1%)      200m (10%)  20Mi (0%)        100Mi (1%)
Allocated resources:
  (Total limits may be over 100 percent, i.e., overcommitted.)
  CPU Requests    CPU Limits    Memory Requests    Memory Limits
  ------------    ----------    ---------------    -------------
  680m (34%)      400m (20%)    920Mi (12%)        1070Mi (14%)
```

Total capacity of the node

Amount allocatable for pods (after some is reserved for the OS and kubelet etc

Requests/limits for each pod on that node

Total requests/limits for the node

Hitting resource limits

- If a container attempts to use more than the CPU resource limit, its CPU usage will be capped

```
kubectl describe pod cpu-demo -n cpu-example
Name:                cpu-demo
…
 Limits:
      cpu:  1
    Requests:
      cpu:        500m
-----------------------------------------------------
kubectl top pods -n cpu-example
NAME        CPU(cores)    MEMORY(bytes)
cpu-demo    999m          2Mi
```

- If a container attempts to use more than the memory resource limit it will be OOMKilled

```
kubectl get pods -n mem-example
NAME            READY     STATUS             RESTARTS   AGE
memory-demo-2   0/1       CrashLoopBackOff   1          19s
kubectl get pods -n mem-example
NAME            READY     STATUS      RESTARTS   AGE
memory-demo-2   0/1       OOMKilled   2          25s
-------------------------------------------------------------
kubectl describe pod memory-demo -n mem-example
Name:               memory-demo-2
…
State:          Waiting
      Reason:       CrashLoopBackOff
    Last State:     Terminated
      Reason:       OOMKilled
```

Why IBM Cloud Integration Education?

Invest in the skills of your workforce with 3 options:

Cost effective, digital-based technical training	→	Individual and Enterprise Digital Learning Subscriptions
Understanding skill gaps and the paths to close those gaps	→	Skill Assessments & Augmentation with skill roadmap recommendations
Develop professional acumen	→	Validate your skills with IBM Cloud Certification and Digital Credentials

Motivated employees + **Trained employees** = **Increased employee productivity** + **Faster ROI**

To learn more, contact us: **CloudIntegrationEdu@uk.ibm.com**

Why IBM Cloud Integration Expert Labs?

Our purpose	How we drive success	How you benefit
Expedite the successful deployment of IBM's Cloud Integration Solutions	Through our deep technical expertise, methodology, repeatable patterns, learning services and mentoring	Faster time to value from your IBM Cloud Integration Solutions

Contact us through email: **labs@us.ibm.com**