Application Management with IBM Cloud Pak for Multicloud Management

IBM

Contents

Introduction

To meet the unique needs of your business and remain competitive in today's fast-moving environment, you may find yourself adopting infrastructure and solutions from a wide range of cloud vendors.

A hybrid, multicloud world is quickly becoming the new normal. But managing your cloud-based services and data across multiple providers can feel overwhelming. With each set of cloud services coming with its own tools, you're likely facing increased complexity and cost. New management solutions and delivery methods can help optimize performance, control costs, provide quick cloud access and secure your mix of applications, environments, and data, whether they are inside your data center or in the cloud.

IBM Cloud Pak for Multicloud Management can manage Kubernetes clusters that are deployed on any target infrastructure - either in your own data center or in a public cloud. IBM Cloud Pak for Multicloud Management includes IBM Cloud App Management to simplify monitoring your applications across any cloud environment.

IBM Cloud Pak for Multicloud Management helps companies make the transition from traditional monitoring systems to cloud-based ones more easily. It effectively monitors all kinds of IT resources in a hybrid environment. It helps Operation teams manage hybrid environments without hiring new personnel to support each new technology that is being used by developers.

Cloud Pak for Multicloud Management provides consistent visibility, automation, and governance across a range of multicloud management capabilities such as cost and asset management, infrastructure management, application management, multi-cluster management, edge management, and integration with existing tools and processes. Customers can leverage Cloud Pak for Multicloud Management to simplify their IT and application ops management, while increasing flexibility and cost savings with intelligent data analysis driven by predictive signals.

In this tutorial, you will explore the following key capabilities:

- Understand Cloud Pak for Multicloud Management
- Deploy an application using Channels and Subscriptions
- Move the application between clusters using Placement Policies
- Visualize the application health using Grafana dashboards

For more information about IBM Cloud Pak for Multicloud Management, visit:
https://www.ibm.com/cloud/cloud-pak-for-management

Business Scenario

As a member of the Cloud Operation team, you are having problems to manage your multicloud hybrid world. Operate your cloud-based services and data across multiple providers is overwhelming your team. Your company is deploying multiple Kubernetes clusters to address their specific needs. Some Dev teams are deploying clusters across public and private clouds, and some are deploying clusters across regions, and some are deploying clusters to support the development and test needs.

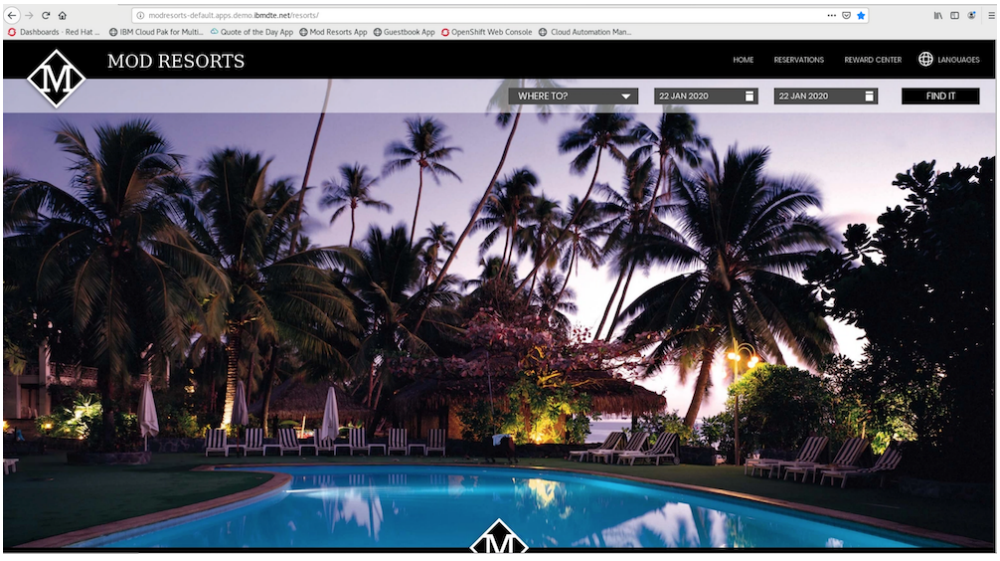
As different teams deploy more clusters, new challenges are introduced:

- Where are my services running?
- How can I monitor applications across clusters and clouds?
- How can I manage clusters as if they were one environment?
- How do I monitor usage across clouds?
- Where are the failed components?
- How do I deploy applications across these environments?
- How do I move workloads across environments?
- How do I set consistent security policies across environments?
- Which clusters are compliant?
- How can I place workloads based on capacity, policy?

Because of that, you want to explore how IBM Cloud Pak for Multicloud Management, provides consistent visibility, governance and automation of your complex environment.

IBM Cloud Pak for Multicloud Management provides enhanced application management capabilities through an improved application model and deployment options. The concept helps simplify and streamline application life cycle management across clusters.

In this tutorial, you will be using a sample Modresort application to demonstrate how to deploy an application in multiple clusters using channels and subscriptions. The Modresort is a WebSphere Liberty Java application available in Dockerhub.



In this tutorial, you use two Red Hat OpenShift clusters.

- **Local-cluster** is the Hub cluster that includes management console, federated monitoring, and all the controllers. In this Lab, you identify the local-cluster with the label **Dev** for environment.
- **Managed-cluster** is a single node Openshift cluster managed by the Hub cluster. In this Lab, you identify the managed cluster with the label **QA** for environment

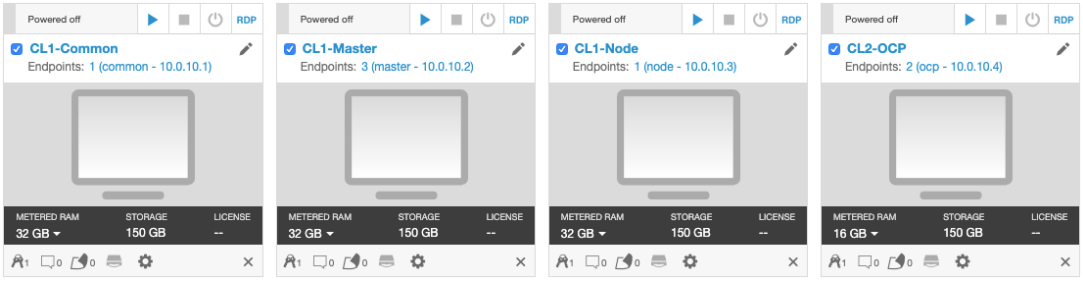
You complete the following tasks:

- Deploy Modresort application using Channels and Subscriptions
- Move the application between clusters using Placement Policies
- Visualize application health by using Grafana dashboard

Tutorial Environment

Environment Overview

Four virtual machines have been provided for this tutorial.

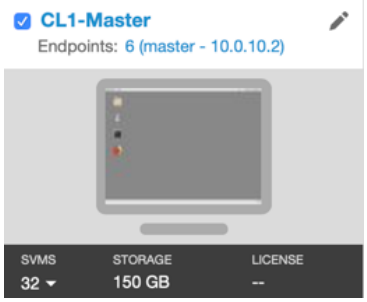


- CL1 is the Hub Cluster. It is an OpenShift Container Platform, **OCP.** It is installed in three VMs, the **CL1-Master** VM with master node, **CL1-Common** VM with all common services, and one compute node VM: **CL1-Node**. As part of this tutorial, you use CL1-Master as workstation too.
- CL2 is the Managed Cluster. It is an All-in-One OpenShift cluster. Here you have **CL2-OCP** VM.

Starting your Environment

__1. Follow the instructions on the right to provision the lab environment.
__2. If the environment is not started automatically, click the play button in the upper right to start all four virtual machines. This takes approximately 15 minutes.

__3. Click **CL1-Master** to access the desktop of the server



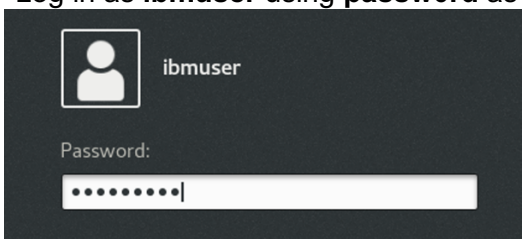
__4. A Linux desktop appears in your browser tab. Feel free to resize the window for a better view.

Lab - IBM Cloud Pak for Multicloud Management Cluster Management

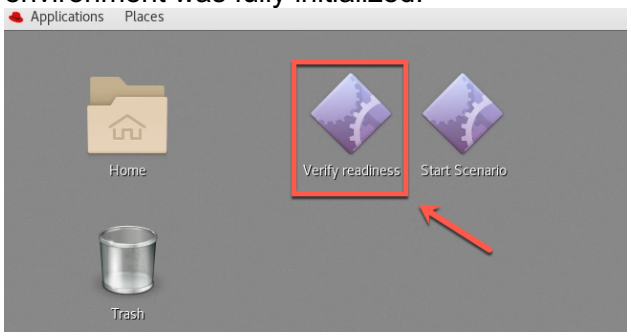
4.1 Accessing the IBM Cloud Pak for Multicloud Management

In this section, you explore your IBM Cloud Pak for Multicloud Management environment.

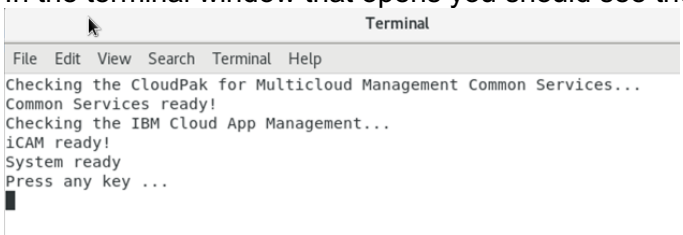
__1. Log in as **ibmuser** using **passw0rd** as a password.



__2. Verify that the environment was fully initialized. On the desktop, you should see an icon named **Verify readiness**. Double-click on the icon to run the verification script that checks if the environment was fully initialized.

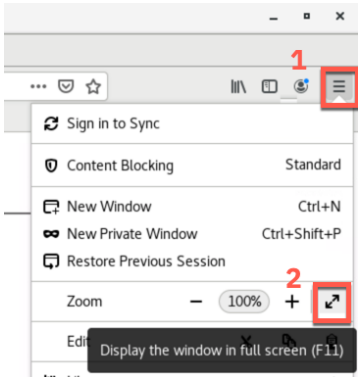


__3. In the terminal window that opens you should see the following text:

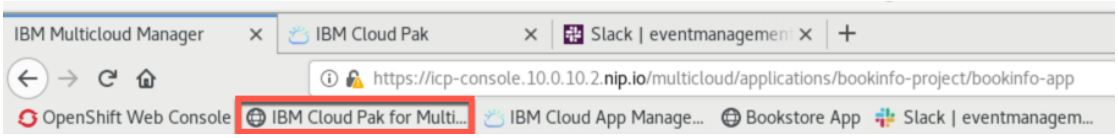


If the environment is not ready, wait until the "System ready" message is displayed.

__4. Start the **Firefox** browser (link is on the desktop). For better viewing, switch the browser in the virtual machine into the "Fullscreen" mode.

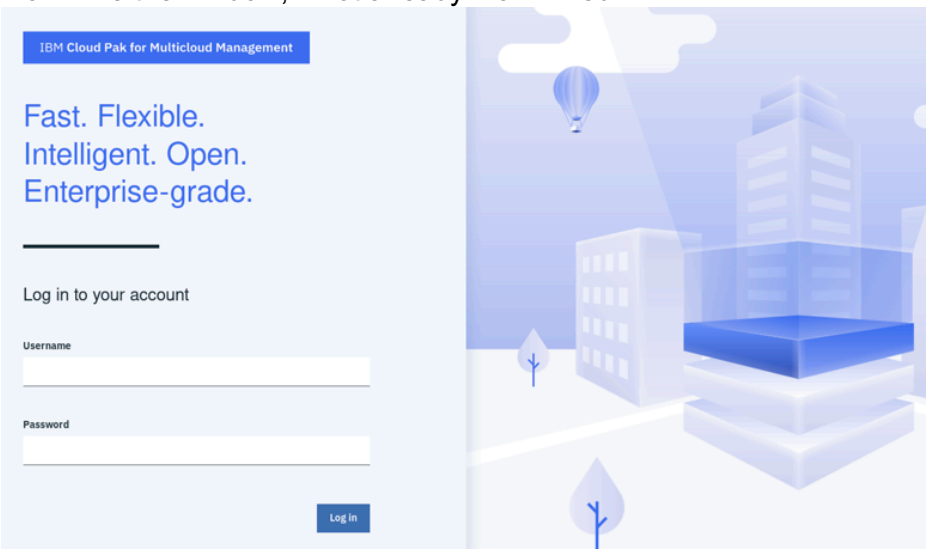


__5. The link to the IBM Cloud Pak for Multicloud Management is added to the Bookmark toolbar. Click the bookmark to open the UI.

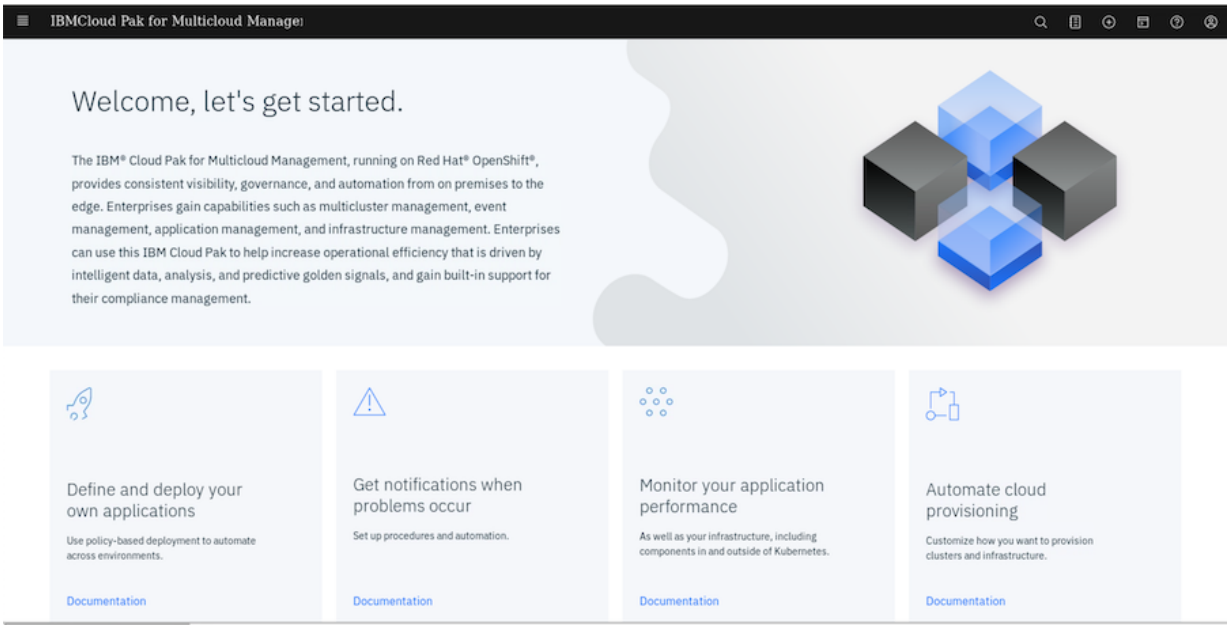


HINT: At any point during the lab, if you are lost navigating in the Cloud Pak UI you can click the link again to return to the main product screen.

__6. If not already logged in, log in as **admin** with a password of **Passw0rd!** Maximize the window, if not already maximized.



__7. After you log in, you are presented with a Welcome screen. If you see a different screen (Authentication), click again on the IBM Cloud Pak for Multicloud Management link on the bookmark toolbar.

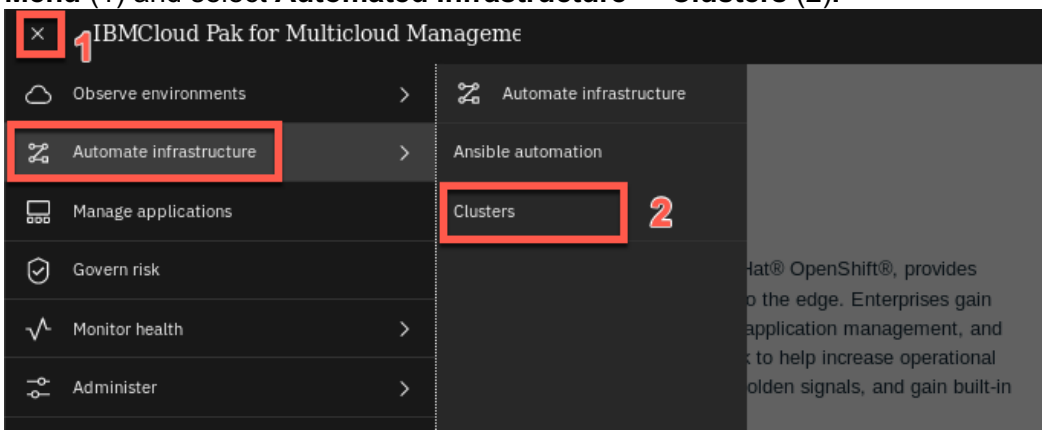


4.2 Add a Managed Cluster

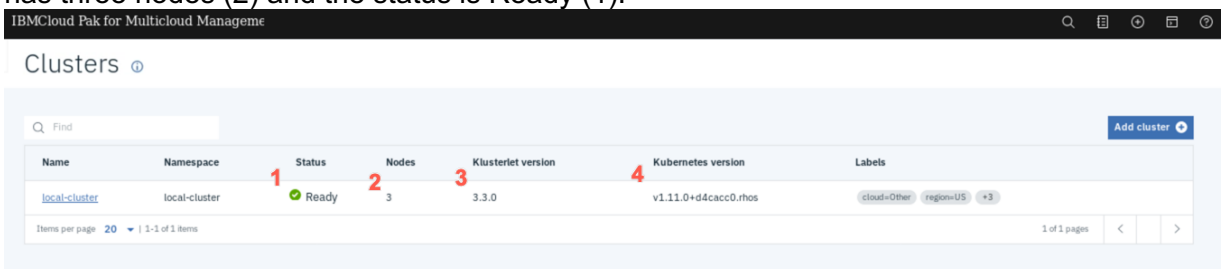
IBM Cloud Pak for Multicloud Management provides consistent visibility, governance and automation from on premises to the edge. Enterprises gain capabilities such as multicluster management, event management, application management and infrastructure management. In this section, you add a new managed cluster in your Control Panel.

Note: If you are using the same environment of the Cluster Management tutorial, you should already have the managed cluster, so you can skip this section and move to the next one (Define application channel).

__1. To view the cluster information available in your Management environment, click the hamburger **Menu** (1) and select **Automated Infrastructure -> Clusters** (2).

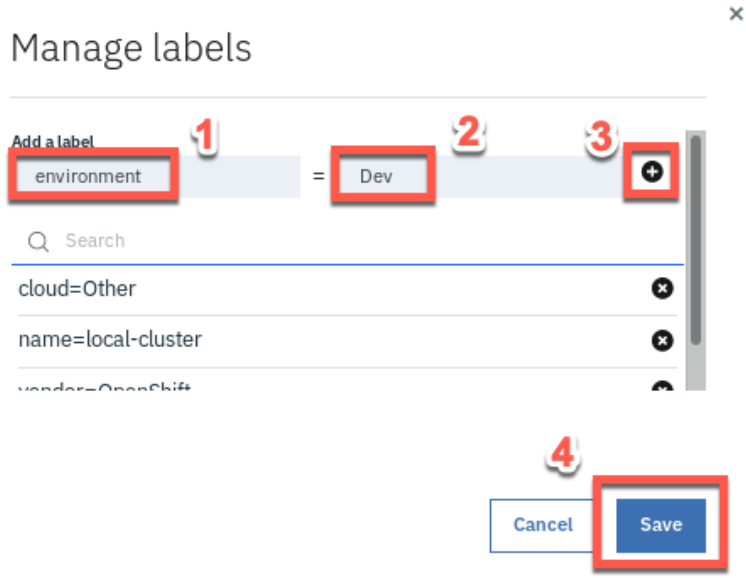


__2. Initially, only one cluster is shown in the list, which is the local-cluster (hub cluster). The cluster has three nodes (2) and the status is Ready (1).

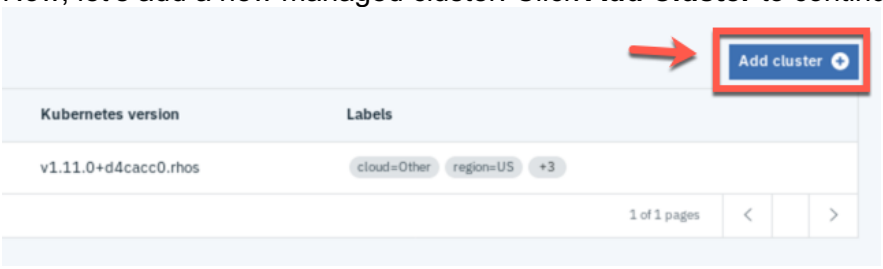


__3. You need to add labels to identify your cluster. On the local-cluster row, click on the three dots icon (1) and select **Edit labels** (2).

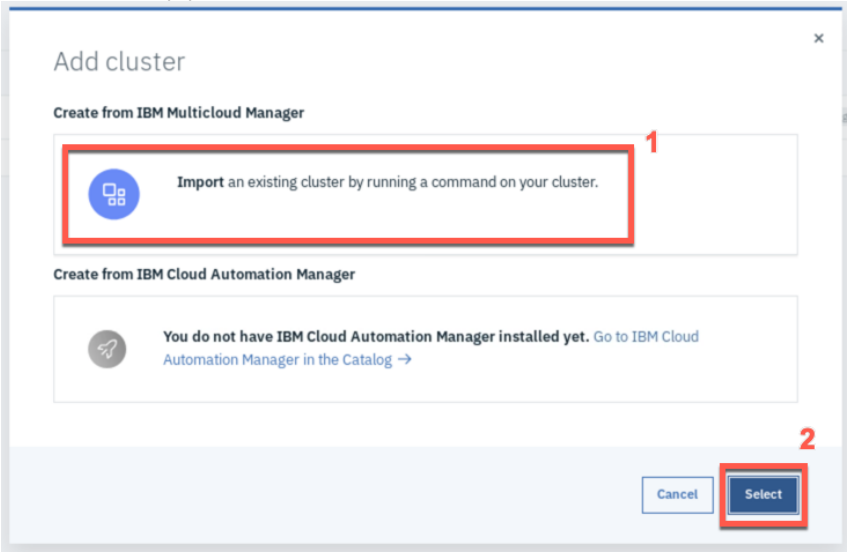
__4. Add a label named **environment** (1) with a value of **Dev** (2). Click **+** (3) and click **Save (4)**.



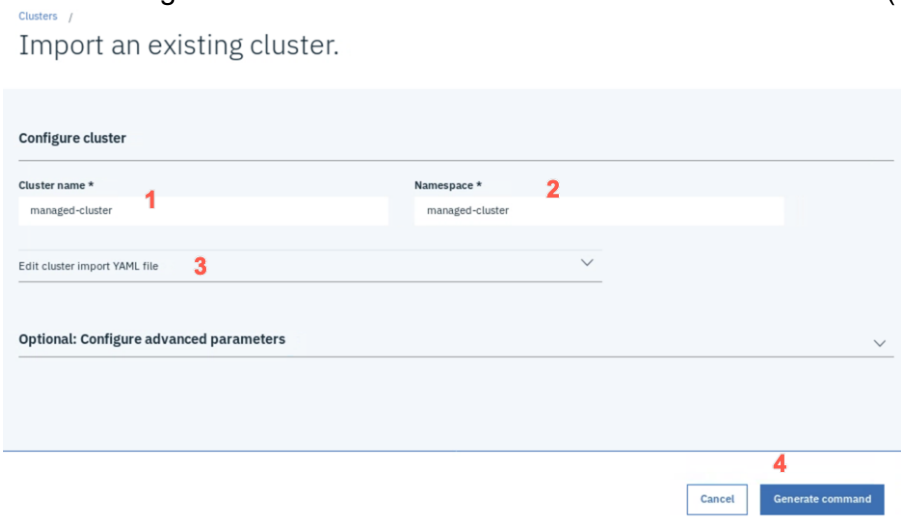
__5. Now, let's add a new managed cluster. Click **Add Cluster** to continue.



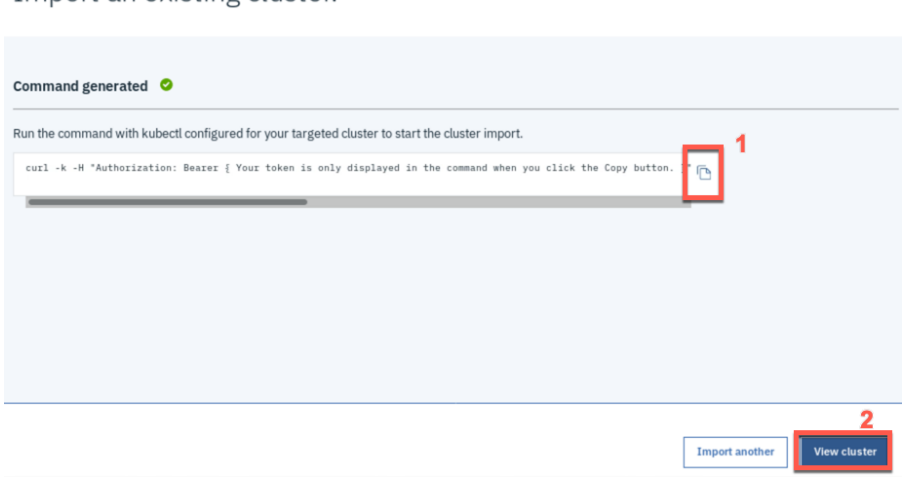
__6. You can add a cluster by Importing an existing cluster or provisioning a new cluster that uses Cloud Automation Manager. We use the first option. Select **Import an Existing cluster** (1) and click **Select** (2).



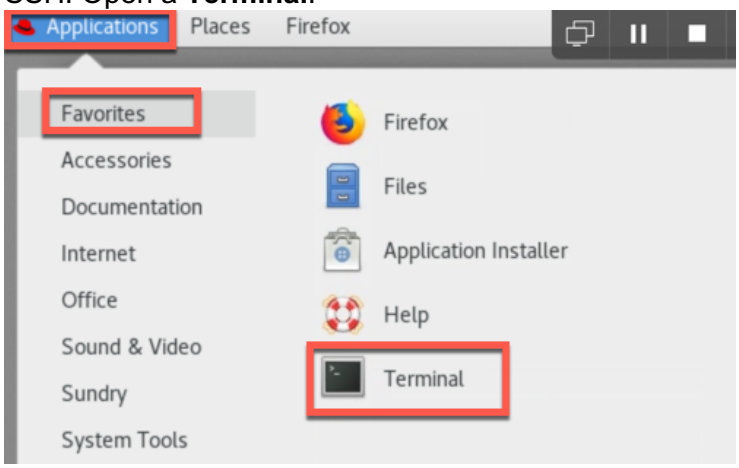
__7. Enter **managed-cluster** for cluster name (1) and **managed-cluster** for namespace (2). You can view the yaml file and change the settings as needed (3). To import an OpenShift cluster no further changes are needed. Click **Generate command** to continue (4).



__8. A curl command is generated that you will use to add the new cluster. Click **Copy command** button (1) and click **View cluster** (2) to see the new managed-cluster details page.



__9. You need to access your managed-cluster to execute the import command. Let's do it using SSH. Open a **Terminal**.



__10. Your managed server (CL2-OCP VM) has a DNS name: OCP. Issue these commands bellow to access the managed server and connect as root user.

ssh ocp

oc login -u admin -p passw0rd https://ocp.ibm.demo:8443

```
[ibmuser@master ~]$ ssh ocp
Last login: Thu Apr  9 14:39:32 2020 from master.ibm.demo
[ibmuser@ocp ~]$ oc login -u admin -p passw0rd https://ocp.ibm.demo:8443
Login successful.

You have access to the following projects and can switch between them with 'oc
project <projectname>':

    default
    kube-public
    kube-service-catalog
```

__11. **Paste** the generated command that you previously copied in the clipboard (Edit -> Paste or Shift+Ctrl+V). When you run the command, several Kubernetes objects are created in the multicluster-endpoint namespace.

```
1/clusters/managed-cluster/managed-cluster/import.yaml | kubectl apply -f -
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  6270    0  6270    0     0   4185      0 --:--:--  0:00:01 --:--:--  4185
customresourcedefinition.apiextensions.k8s.io/endpoints.multicloud.ibm.com creat
ed
namespace/multicluster-endpoint created
secret/klusterlet-bootstrap created
secret/multicluster-endpoint-operator-pull-secret created
serviceaccount/ibm-multicluster-endpoint-operator created
endpoint.multicloud.ibm.com/endpoint created
```

If you receive the following error, run the command again:
error: unable to recognize "STDIN": no matches for kind "Endpoint" in version "multicloud.ibm.com/v1beta1"

__12. You can view the progress by entering the command:

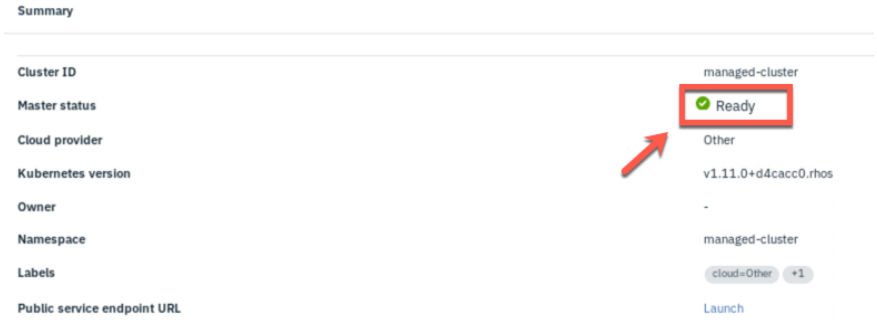
oc get pods -n multicluster-endpoint

Make sure all the pods are in the running state.

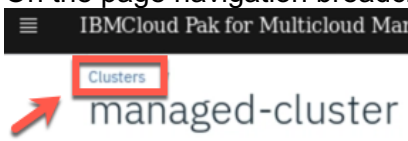
```
[root@ocp ~]# oc get pods -n multicluster-endpoint
NAME                                                READY     STATUS    RESTARTS   AGE
endpoint-appmgr-564d67bc5-mmlsg                     1/1       Running   0          4m
endpoint-certmgr-55fc77fcdf-c7dwl                   1/1       Running   0          4m
endpoint-component-operator-b889dc78-prprx          1/1       Running   0          4m
endpoint-connmgr-57fdbc94cf-mxkh9                   1/1       Running   0          4m
endpoint-policyctrl-55bdf4b5d9-4ctxp                2/2       Running   0          4m
endpoint-search-558c88bd6c-ghc6c                    1/1       Running   0          4m
endpoint-svcreg-b5765895c-z9ftb                     1/1       Running   0          4m
endpoint-svcreg-coredns-547f764d9c-lvj7x            1/1       Running   0          4m
endpoint-tiller-6568dccbbd-fhqpl                    1/1       Running   0          4m
endpoint-topology-weave-collector-564dd856b-zjbpx   1/1       Running   0          4m
endpoint-topology-weave-scope-app-54f776489c-68rrz  2/2       Running   0          4m
endpoint-topology-weave-scope-xrgsm                 1/1       Running   0          3m
endpoint-workmgr-7994bb4698-5mm89                   1/1       Running   0          4m
ibm-multicluster-endpoint-operator-598559cf69-fgfm9 1/1       Running   0          4m
```

__13. The cluster end point is ready when all the Pods are in Running state. **Close** the terminal window.

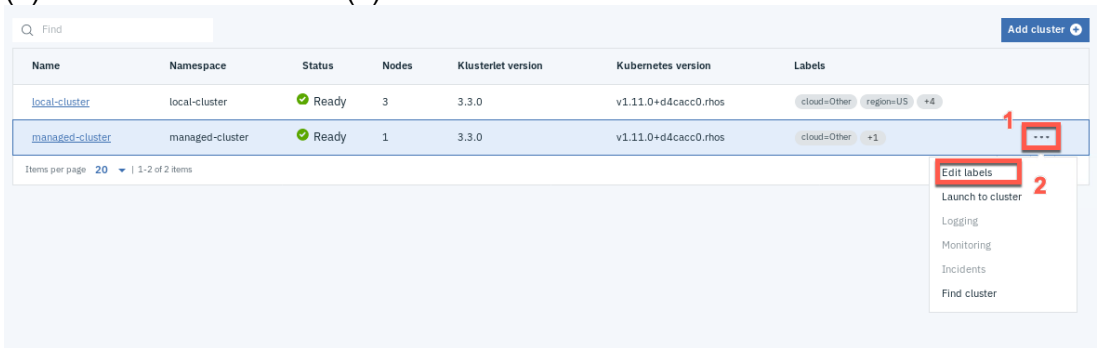
__14. Back to the Firefox windows, make sure that the cluster status is **Ready** now (if necessary, refresh the managed-cluster details page).



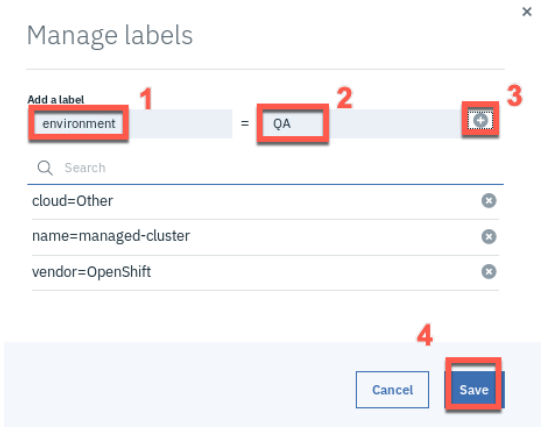
__15. On the page navigation breadcrumb, click on **Clusters** link.



__16. You notice that a second cluster with the name managed-cluster is added to the list. You can add labels to identify your new cluster. On the managed-cluster row, click on the three dots icon (1) and select **Edit labels** (2).



__17. Add a label named **environment** (1) with a value of **QA** (2). Click **+** (3) and click **Save (4)**.



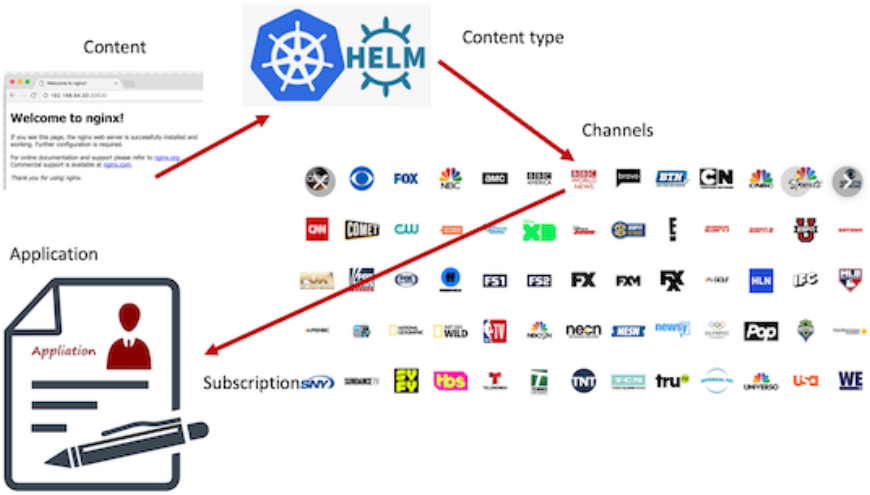
You have successfully added a cluster. Using Multicloud Management you will be able to manage both the clusters from a single pane of glass. Let's check it in the next section.

4.3 Define Application Channel

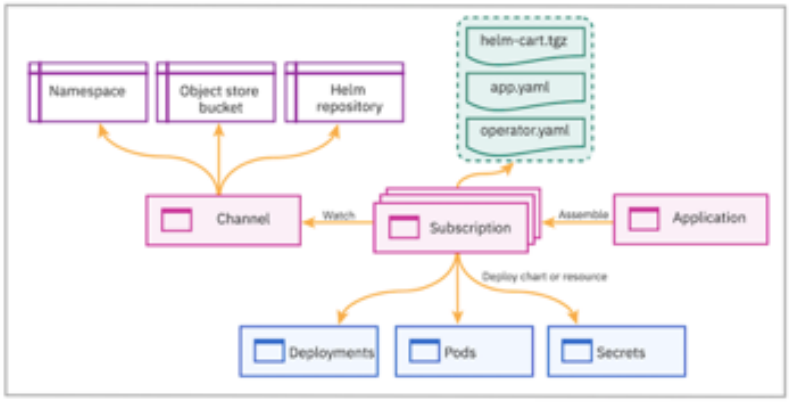
IBM Cloud Pak for Multicloud Management provides enhanced application management capabilities through an improved application model and new deployment options. The new model and deployment options are designed to unify and simplify the deployment experience for creating and managing your applications across clusters.

The new application management capabilities use Channels and Subscriptions to gain improved continuous and automated delivery of deployables to target managed clusters.

The concept is similar to subscription model of TV channels. In this model, all the applications, which are packaged as helm charts, will be hosted in one or more repositories. The repositories, which contain the application packages, are defined as channels that broadcast across the clusters. If you want to deploy an application, then define a subscription to the channel with the name of the application (one or more) you want to deploy.



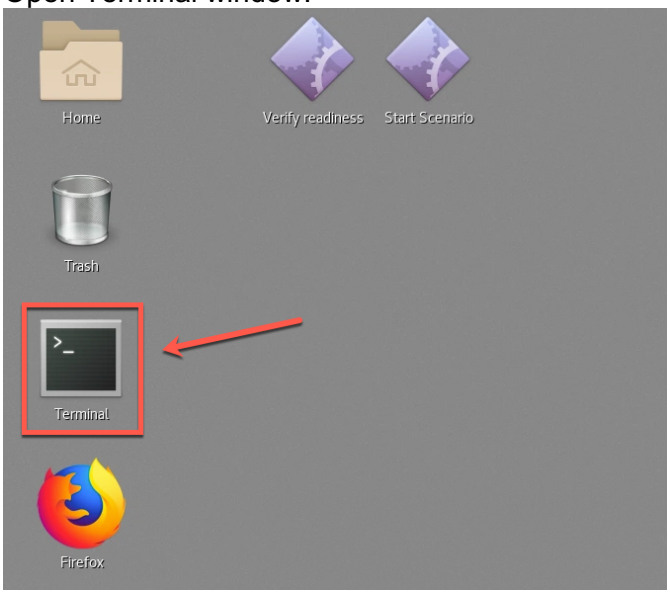
Channels (Channel.app.ibm.com) define a namespace within the hub cluster and point to a physical place where resources are stored for deployment; such as an object store, Kubernetes namespace, or Helm repository.



In this tutorial, the channel is of type namespace, meaning that the yaml you create will be deployed and stored in OpenShift namespaces rather than in a Helm chart or Object store.

In this section, you define Application and Channel resources to deploy the Modresort application. The modresort application is a simple application with only one component.

__1. Open Terminal window.



__2. Let's create a new directory, and then change the directory to the new directory, by executing the command bellow:

git clone https://github.com/rafosorio/appmgmtlab.git

cd appmgmtlab

```
[ibmuser@master ~]$ git clone https://github.com/rafosorio/appmgmtlab.git
Cloning into 'appmgmtlab'...
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 29 (delta 8), reused 13 (delta 5), pack-reused 0
Unpacking objects: 100% (29/29), done.
[ibmuser@master ~]$ cd appmgmtlab
[ibmuser@master appmgmtlab]$
```

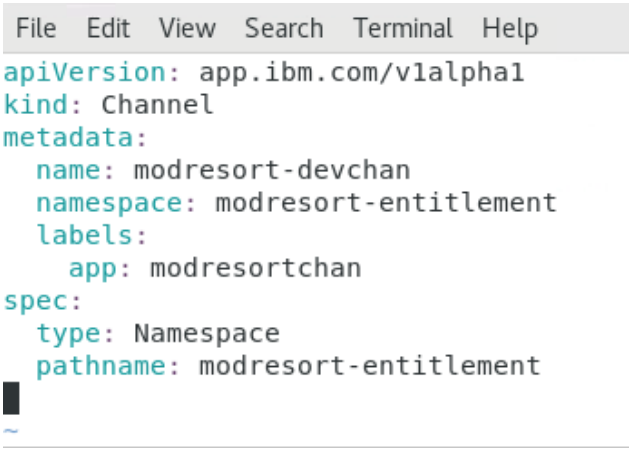
__3. Now you have two folders (modresortchan and modresortapp). Let's start exploring the modresortchan and use any editor to see the content of the **channel.yaml** file.

cd modresortchan

vi channel.yaml

__4. Our channel.yaml has the content below.

```
apiVersion: app.ibm.com/v1alpha1
kind: Channel
metadata:
  name: modresort-devchan
  namespace: modresort-entitlement
  labels:
    app: modresortchan
spec:
  type: Namespace
  pathname: modresort-entitlement
```



It is a really simple file. The spec collection defines the type of the channel. In this lab, the channel is of type namespace, meaning that the yaml you create will be deployed and stored in OpenShift namespaces rather than in a Helm chart or Object store. This file is ready, and you don't need to edit it, go ahead and **close** the channel.yaml file.

__5. The modresort application component consists of a **deployment resource definition** and a **service resource definition**.

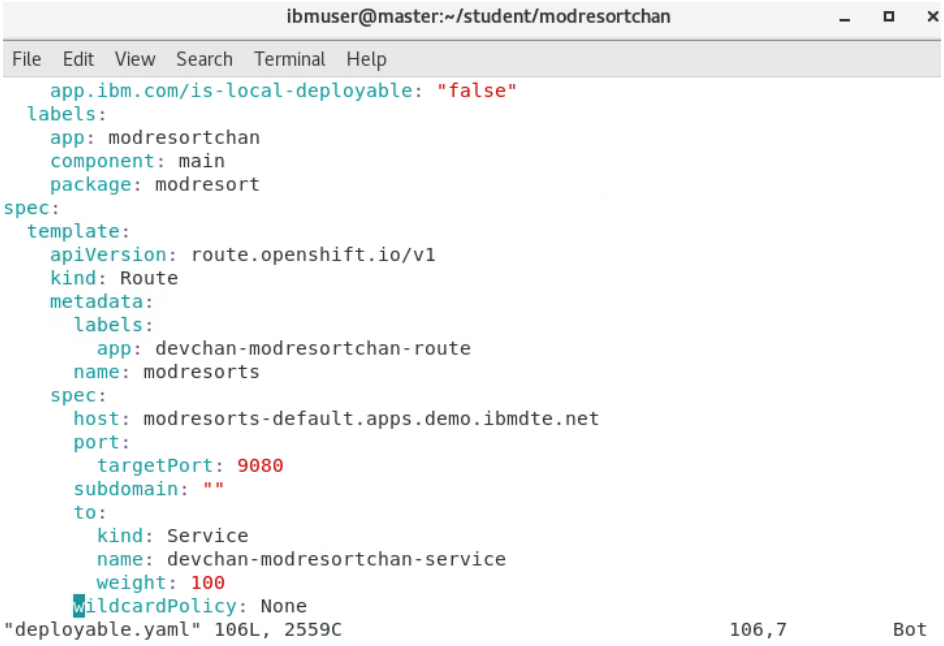
To enable these components to be used by the channel subscription, each of the resources need to be wrapped by a new custom resource definition (CRD) called **Deployable**.

Use your editor again and check now the deployable.yaml file.

vi deployable.yaml

```
apiVersion: app.ibm.com/v1alpha1
kind: Deployable
metadata:
  name: devchan-modresortchan-deployment
  namespace: modresort-entitlement
  annotations:
    app.ibm.com/is-local-deployable: "false"
  labels:
    app: modresortchan
    component: main
    package: modresort
spec:
  template:
    kind: Deployment
    apiVersion: apps/v1
    metadata:
      name: devchan-modresortchan-deployment
      labels:
        app: modresortchan
    spec:
      selector:
        matchLabels:
          app: modresortchan
          release: modresort-devchan
          tier: frontend
      replicas: 1
      template:
        metadata:
          labels:
            app: modresortchan
            release: modresort-devchan
            tier: frontend
        spec:
          containers:
            - name: frontend
              image: "kpostreich/modresort:1.0"
              imagePullPolicy: Always
              ports:
                - containerPort: 9080
              env:
              - name: GET_HOSTS_FROM
                value: dns
              - name: WLP_LOGGING_CONSOLE_FORMAT
                value: json
              - name: WLP_LOGGING_CONSOLE_LOGLEVEL
                value: info
              - name: WLP_LOGGING_CONSOLE_SOURCE
                value: message,trace,accessLog,ffdc
---
apiVersion: app.ibm.com/v1alpha1
kind: Deployable
metadata:
  name: devchan-modresortchan-service
  namespace: modresort-entitlement
  annotations:
    app.ibm.com/is-local-deployable: "false"
  labels:
```

```
    app: modresortchan
    component: main
    package: modresort
spec:
  template:
    kind: Service
    apiVersion: v1
    metadata:
      name: devchan-modresortchan-service
      labels:
        app: modresortchan
    spec:
      type: NodePort
      ports:
        - port: 9080
      selector:
        app: modresortchan
        release: modresort-devchan
        tier: frontend
---
apiVersion: app.ibm.com/v1alpha1
kind: Deployable
metadata:
  name: devchan-modresortchan-route
  namespace: modresort-entitlement
  annotations:
    app.ibm.com/is-local-deployable: "false"
  labels:
    app: modresortchan
    component: main
    package: modresort
spec:
  template:
    apiVersion: route.openshift.io/v1
    kind: Route
    metadata:
      labels:
        app: devchan-modresortchan-route
      name: modresorts
    spec:
      host: modresorts-default.10.0.10.2.nip.io
      port:
        targetPort: 9080
      subdomain: ""
      to:
        kind: Service
        name: devchan-modresortchan-service
        weight: 100
      wildcardPolicy: None
```



There are three Deployables defined in the yaml file that wrap the modresort Kubernetes resources

- One for the modresort deployment, refers to the location of the Docker image;
- One for the modresort service, refers to the service port and defines a NodePort;
- One for the modresort routes, refers to domain name and exposes to the external network;

Refer to the online documentation for details on the construct of the Deployable definition: https://www.ibm.com/support/knowledgecenter/en/SSFC4F_1.2.0/mcm/applications/managing_deployables.html

Next section, you learn how to create a subscription.

4.4 Create a Subscription

The subscription to a channel package contains

- Application Definition
- Placement Rules Definition
- Subscription Definition

Applications (Application.app.k8s.io) in IBM Multicloud Manager are used for grouping application components.

Placement rules (PlacementRule.app.ibm.com) define the target clusters where deployables can be deployed. You can use placement rules to help you facilitate the multi-cluster deployment of your deployables. Placement rules can be referenced by deployables and subscriptions.

Subscriptions (Subscription.app.ibm.com) are sets of definitions that identify deployables within channels by using annotations, labels, and versions.

The subscription controller can monitor the channel for new or updated deployables, such as an updated Helm release or a new Kubernetes deployable object. Then, the controller can download the Kubernetes deployable object or Helm release directly from the source location (Helm repository, object store, or namespace) to the target managed clusters.

__1. Back to the terminal window, let's explore the YAML files on modresortapp folder.

cd ../modresortapp

__2. Now, using your editor again, open the application.yaml.

vi application.yaml

Below is the content of the file:

```
apiVersion: app.k8s.io/v1beta1
kind: Application
metadata:
  name: modresort101-modresortapp
  namespace: modresort-project
  labels:
    app: modresortapp
spec:
  selector:
    matchExpressions:
    - key: release
      operator: In
      values:
      - modresort101
  componentKinds:
  - group: app.ibm.com
    kind: Subscription
```

Here we have the definition of our modresort application. Refer to the online documentation for details on configuring the Application resource.

__3. **Close** the application.yaml file.

__4. Use your editor again, to explore the placementrules.yaml file. Below is the file's content:

vi placementrules.yaml

```
apiVersion: app.ibm.com/v1alpha1
kind: PlacementRule
metadata:
  name: modresortapp101-modresortapp
  namespace: modresort-project
  labels:
    app: modresortapp
    release: modresort101
spec:
  clusterReplicas: 1
  clusterLabels:
    matchLabels:
      environment: Dev
```

Placement Rules defines where and how Helm charts and deployables are deployed. Use placement rules to help you facilitate multi-cluster deployments of your deployables. Refer to the online documentation for details on configuring the Placement Rules resource. In your case, the placementrules.yaml is defining to deploy the modresortapp in **Dev** environment only.

__5. Now, let's explore the **subscription.yaml** file.

vi subscription.yaml

```
apiVersion: app.ibm.com/v1alpha1
kind: Subscription
metadata:
  name: modresort101-modresortapp
  namespace: modresort-project
  labels:
    app: modresortapp
    release: modresort101
spec:
  channel: modresort-entitlement/modresort-devchan
  name: ""
  packageFilter:
    version: ">=1.x"
    labelSelector:
      matchLabels:
        package: modresort
        component: main
  placement:
    placementRef:
      name: modresortapp101-modresortapp
      kind: PlacementRule
      group: app.ibm.com
  overrides:
  - clusterName: "/"
    clusterOverrides:
```

```
    - path: "metadata.namespace"
      value: default
```

This contains the details of relating the placement rule definition with the application specification. Refer to the online documentation for details on configuring the Subscription resource:

__6. **Close** the subscription.yaml file. All your files are ready to deploy the application.

This completes enabling an existing application with policies so that they can be deployed to any Kubernetes managed cluster. Next section, you deploy the application using the channel and subscription created here.

4.5 Deploy the application

In this section, you will deploy the application components to their respective Kubernetes namespaces, using the yaml files that you created in the previous task.

__1. Back to the terminal window, execute the command bellow to login to local-cluster.

oc login -u admin -p Passw0rd! https://master.ibm.demo:8443

```
ibmuser@master:~/student/modresortapp
File  Edit  View  Search  Terminal  Help
[ibmuser@master modresortapp]$ oc login -u admin -p Passw0rd! https://master.ibm
.demo:8443
Login successful.

You have access to the following projects and can switch between them with 'oc p
roject <projectname>':

    bookinfo
    bookinfo-entitlement
    bookinfo-project
    bookinfo-source
```

__2. Now, let's log in to Multicloud Manager console CLI.

cloudctl login -a https://icp-console.10.0.10.2.nip.io -u admin -p Passw0rd! -n default

```
[ibmuser@master modresortapp]$ cloudctl login -a https://icp-console.10.0.10.2.n
ip.io -u admin -p Passw0rd! -n default
Authenticating...
OK

Targeted account mycluster Account

Targeted namespace default

Configuring kubectl ...
Property "clusters.mycluster" unset.
Property "users.mycluster-user" unset.
Property "contexts.mycluster-context" unset.
Cluster "mycluster" set.
User "mycluster-user" set.
Context "mycluster-context" created.
Switched to context "mycluster-context".
OK

Configuring helm: /home/ibmuser/.helm
OK
```

__3. Execute the command bellow to create two new Openshift projects.

oc new-project modresort-project
oc new-project modresort-entitlement

```
[ibmuser@master modresortapp]$ oc new-project modresort-project
Now using project "modresort-project" on server "https://master.ibm.demo:8443".

You can add applications to this project with the 'new-app' command. For example
, try:

    oc new-app centos/ruby-25-centos7~https://github.com/sclorg/ruby-ex.git

to build a new example application in Ruby.
[ibmuser@master modresortapp]$ oc new-project modresort-entitlement
Now using project "modresort-entitlement" on server "https://master.ibm.demo:844
3".

You can add applications to this project with the 'new-app' command. For example
, try:

    oc new-app centos/ruby-25-centos7~https://github.com/sclorg/ruby-ex.git

to build a new example application in Ruby.
[ibmuser@master modresortapp]$
```

__4. Now, let's deploy the modresortchan. Execute the commands below:

cd /home/ibmuser/appmgmtlab
oc apply -f modresortchan

The output of the above command will be similar to the illustration below. The output shows that Channel and Deployables are deployed.

```
[ibmuser@master modresortapp]$ cd /home/ibmuser/student
[ibmuser@master student]$ oc apply -f modresortchan
channel.app.ibm.com/modresort-devchan created
deployable.app.ibm.com/devchan-modresortchan-deployment created
deployable.app.ibm.com/devchan-modresortchan-service created
deployable.app.ibm.com/devchan-modresortchan-route created
[ibmuser@master student]$
```

__5. The subscription is deployed to a different namespace than the channel. Never deploy the subscription to the same namespace as the channel. Hence, deploy the subscription to the other namespace, modresort-project.

oc project modresort-project
oc apply -f modresortapp

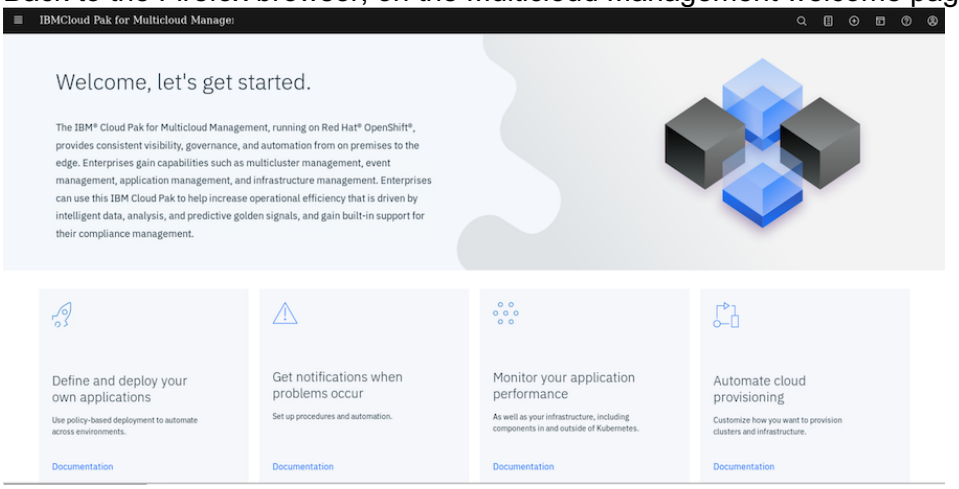
```
[ibmuser@developer student]$ oc project modresort-project
Now using project "modresort-project" on server "https://api.demo.ibmdte.net:6443".
[ibmuser@developer student]$ oc apply -f modresortapp
application.app.k8s.io/modresort101-modresortapp created
placementrule.app.ibm.com/modresortapp101-modresortapp created
subscription.app.ibm.com/modresort101-modresortapp created
```

Next section of the lab will walk you through validating the application deployment.

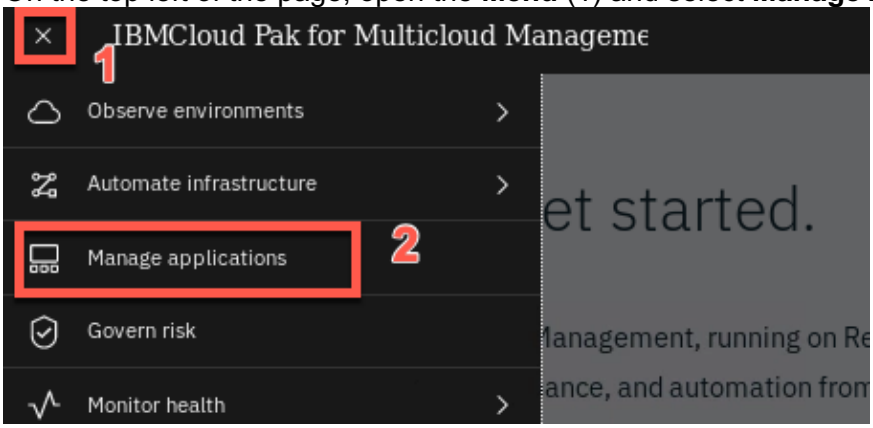
4.6 Validate the application

The Application view in Cloud Pak for Multicloud Management console provides graphical view into the applications, channels and subscriptions deployed in the clusters. Let's validate if the modresort application is deployed correctly.

__1. Back to the Firefox browser, on the Multicloud Management welcome page.

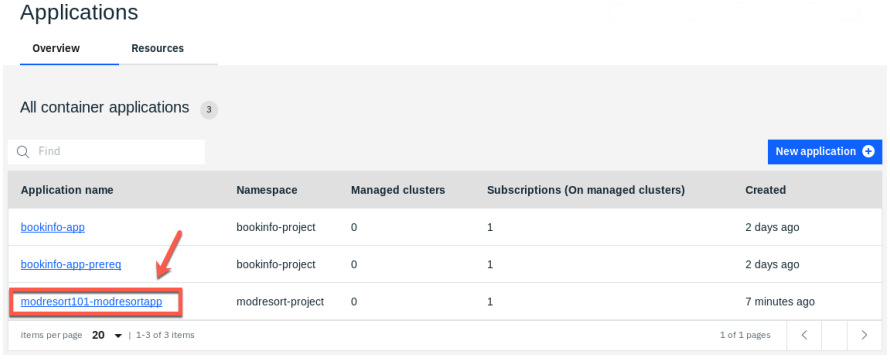


__2. On the top left of the page, open the **Menu** (1) and select **Manage Applications** (2).

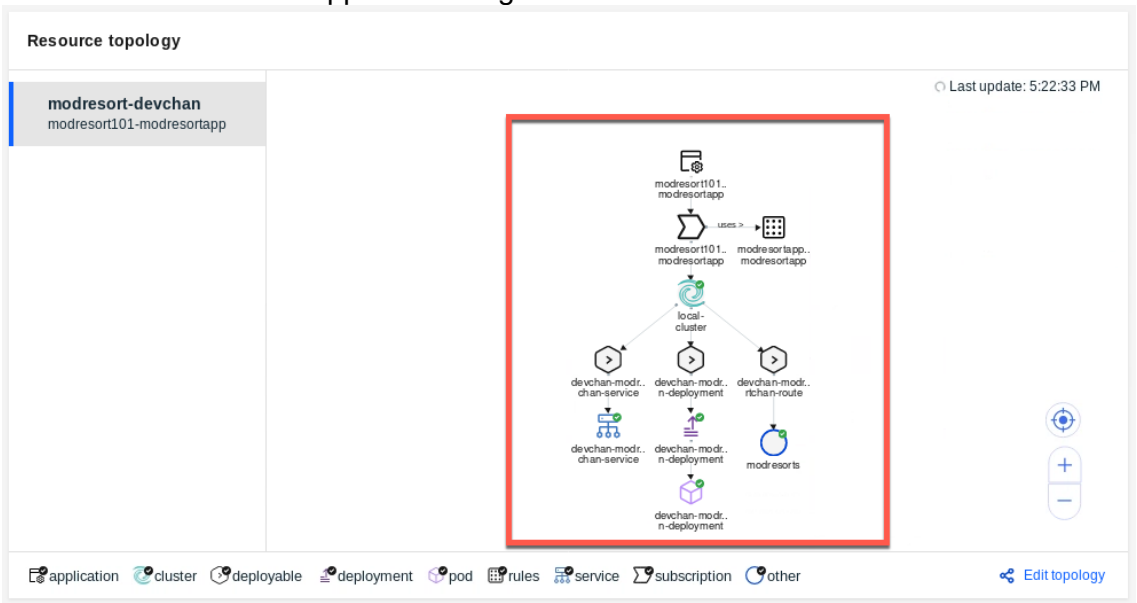


Note: It will take 2-3 minutes for the deployment to complete

__3. Click on the **modresort101-modresortapp** application link to get to the application Overview page.



__4. Scroll down to view the application diagram.



Note: It may take a couple of minutes for the application to deploy to the cluster.

When the application is successfully deployed via the subscription, the pod in the application topology view will have a green icon as illustrated below.

Note (Only if you have issues): After a couple of minutes, if your pod has a yellow icon next to the pod, indicating it is in an unknown state, you will need to troubleshoot your yaml files for correctness and redeploy the resources. Verify that your subscriptions are propagated correctly by running the command.

oc get subscription.app.ibm.com --all-namespaces

The subscription from the **mcm hub** should be in the **Propagated** state.

__5. Back to the terminal window, let's check the pods are in running state.

oc get pods -n default

You should see the devchan-modresort-deployment pod running in the default namespace

```
[ibmuser@master student]$ oc get pods -n default
NAME                                                READY     STATUS    RESTARTS   AGE
devchan-modresortchan-deployment-5647c5d446-p7jjp   1/1       Running   0          18m
docker-registry-1-xbbxt                             1/1       Running   20         45d
icam-kubernetes-resources-k8monitor-849c56696d-gl55t   2/2       Running   10         4d
ldap-5c8fb4ccdf-w4dbr                               1/1       Running   11         30d
mynode-nodejssample-nodejs-7c997dff8-jgk6m          1/1       Running   2          23h
registry-console-1-mr5g2                            1/1       Running   20         45d
router-1-kzn7k                                      1/1       Running   20         45d
[ibmuser@master student]$
```

__6. Use the command line to get the devchan-modresortchan-service NodePort.

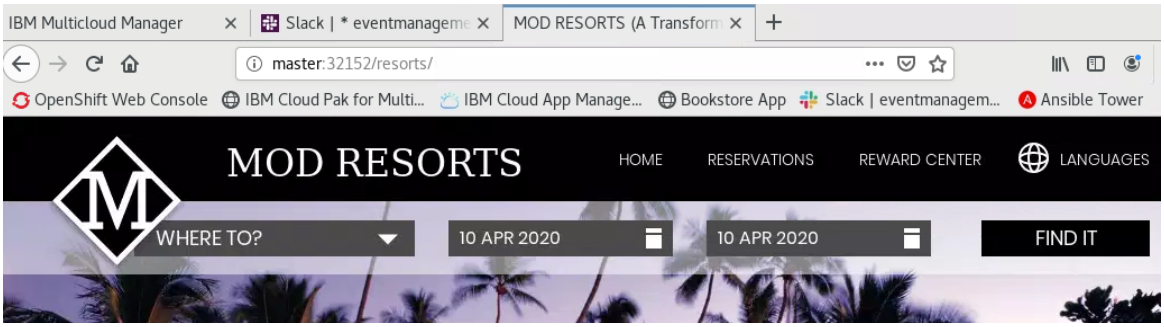
oc get services -n default

```
[ibmuser@ocp ~]$ oc get services -n default
NAME                          TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)
devchan-modresortchan-service   NodePort    172.30.224.109   <none>        9080:32414/TCP
docker-registry               ClusterIP   172.30.50.79     <none>        5000/TCP
kubernetes                    ClusterIP   172.30.0.1       <none>        443/TCP,53/UDP,53/TCP
registry-console              ClusterIP   172.30.111.52    <none>        9000/TCP
router                        ClusterIP   172.30.234.115   <none>        80/TCP,443/TCP,1936/TCP
[ibmuser@ocp ~]$
```

Take note of the devchan-modresortchan-service nodePort, you will use it to access your application.

__7. Back to Firefox browser, open a new tab and navigate to:

http://master:<NodePort>/resorts (for example: http://master:32414/resorts)



Congratulations! You have successfully deployed the modresorts application to the local OpenShift Cluster.

4.7 Move the application between clusters

Applications deployed using the Subscription model are deployed to clusters based on PlacementRules. The placement rules for deployables can be defined as a stand-alone resource and referenced by the deployable. The placement rules use cluster labels to determine where to place the applications.

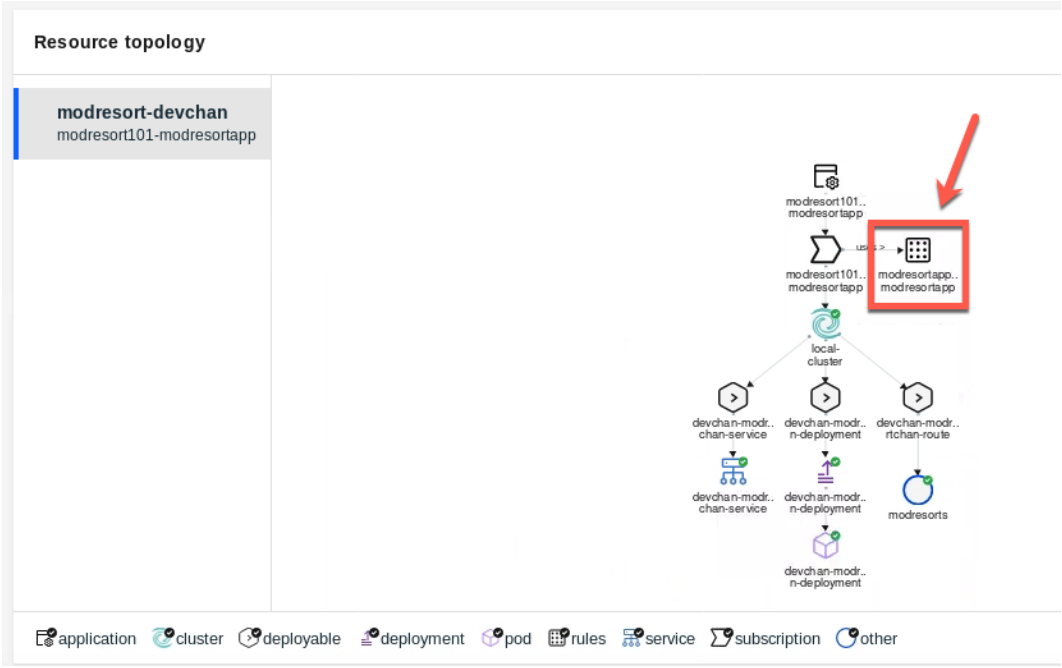
In the following steps, you will move the application from the Dev cluster to the QA cluster by changing the placement policy. You also learn how to use cluster replicas.

__1. Currently, the application deployed as running on Dev cluster because of the placement values specified in application.

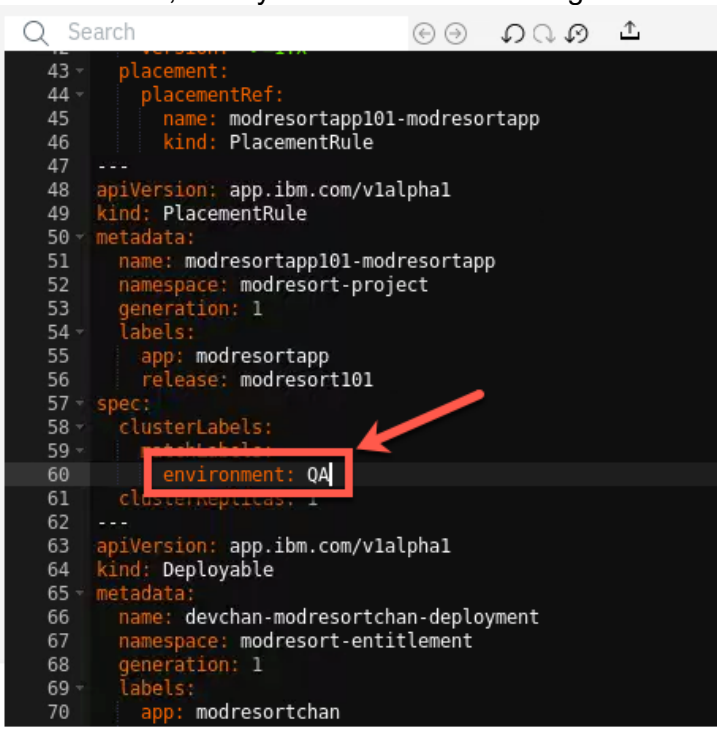
```
apiVersion: app.ibm.com/v1alpha1
kind: PlacementRule
metadata:
  name: modresortapp101-modresortapp
  namespace: modresort-project
  labels:
    app: modresortapp
    release: modresort101
spec:
  clusterReplicas: 1
  clusterLabels:
    matchLabels:
      environment: Dev
```

You can modify the placement policies from application view by selecting the placement policy and modifying the corresponding values in the yaml file. Let's do it!

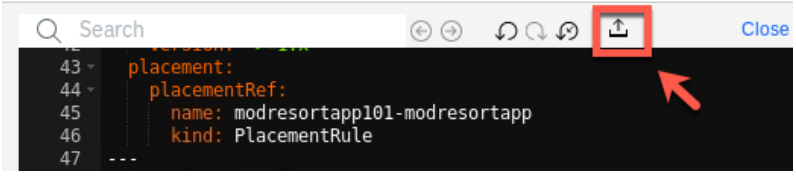
Back to the Multicloud Management page, on the Resource topology chart, click on the **modresortapp101 placement policy** icon.



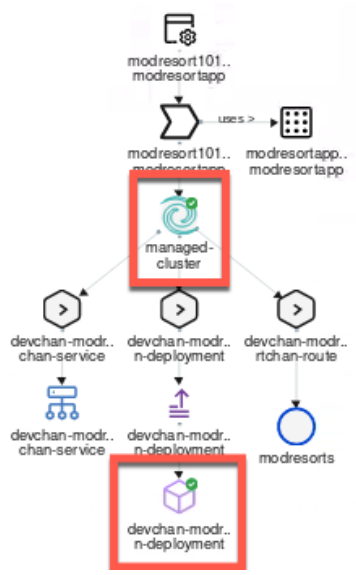
__2. On the Editor, modify the environment setting from **Dev** to **QA**.



__3. Apply changes by clicking on update icon as shown below.



__4. Within a few minutes, the cluster value changes from **local-cluster** to **managed-cluster**, and the application is now running on managed cluster instead of local-cluster.



__5. To validate that the application moved from the local-cluster to the managed-cluster, back to the Terminal window log out of the local-cluster and log in to managed-cluster (OCP):

oc logout

oc login -u admin -p passw0rd https://ocp.ibm.demo:8443 -n default

```
[ibmuser@master student]$ oc logout
Logged "admin" out on "https://master.ibm.demo:8443"
[ibmuser@master student]$ oc login -u admin -p passw0rd https://ocp.ibm.demo:8443 -n default
The server uses a certificate signed by an unknown authority.
You can bypass the certificate check, but any data you send to the server could be intercepted by others.
Use insecure connections? (y/n): y

Login successful.

You have access to the following projects and can switch between them with 'oc project <projectname>':

  * default
    kube-public
    kube-service-catalog
    kube-system
    managed-cluster
    management-infra
    multicluster-endpoint
    openshift
    openshift-ansible-service-broker
    openshift-console
    openshift-infra
    openshift-logging
    openshift-monitoring
    openshift-node
    openshift-sdn
    openshift-template-service-broker
    openshift-web-console

Using project "default".
```

__6. Check that **devchan-modresortchan-deployment** pod is **running**, executing the command below:

oc get pods

```
[ibmuser@master student]$ oc get pods
NAME                                          READY     STATUS    RESTARTS   AGE
devchan-modresortchan-deployment-5647c5d446-jgths   1/1       Running   0          17m
docker-registry-1-khdwq                       1/1       Running   13         43d
registry-console-1-m6kkh                      1/1       Running   13         43d
router-1-hhkhf                                1/1       Running   13         43d
[ibmuser@master student]$
```

__7. Execute the command below to get the service port.

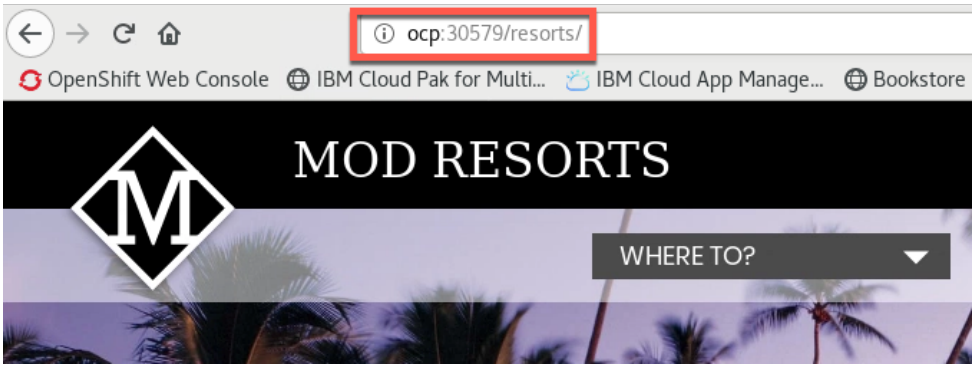
oc get svc

Make a note of the nodePort value to access the UI.

```
[ibmuser@master student]$ oc get svc
NAME                          TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                   AGE
devchan-modresortchan-service   NodePort    172.30.89.173    <none>        9080:30579/TCP            24m
docker-registry               ClusterIP   172.30.82.77     <none>        5000/TCP                  43d
kubernetes                    ClusterIP   172.30.0.1       <none>        443/TCP,53/UDP,53/TCP     43d
mynode-nodejssample-nodejs    NodePort    172.30.255.192   <none>        3000:30512/TCP            1d
registry-console              ClusterIP   172.30.195.124   <none>        9000/TCP                  43d
router                        ClusterIP   172.30.9.165     <none>        80/TCP,443/TCP,1936/TCP   43d
```

__8. Back to **Firefox** browser, open a new tab and access the application running on managed-cluster using the following link:

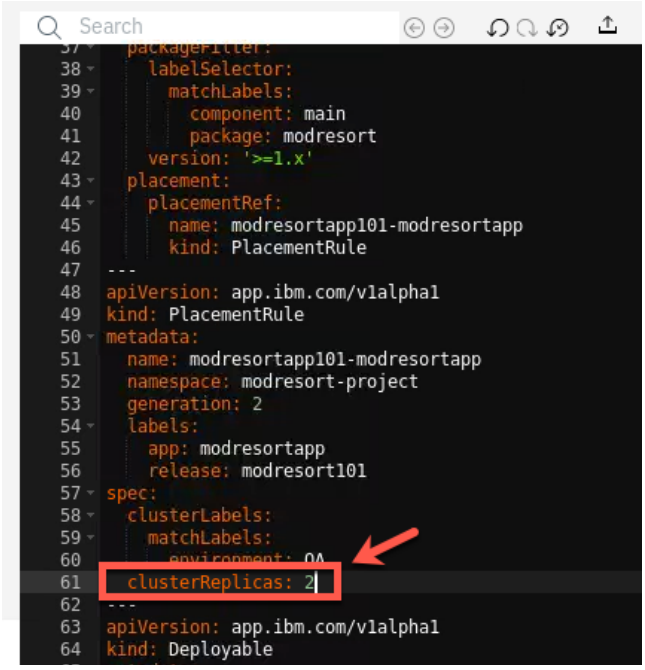
http://ocp:<nodePort>/resorts (for example: http://ocp:30579/resorts)



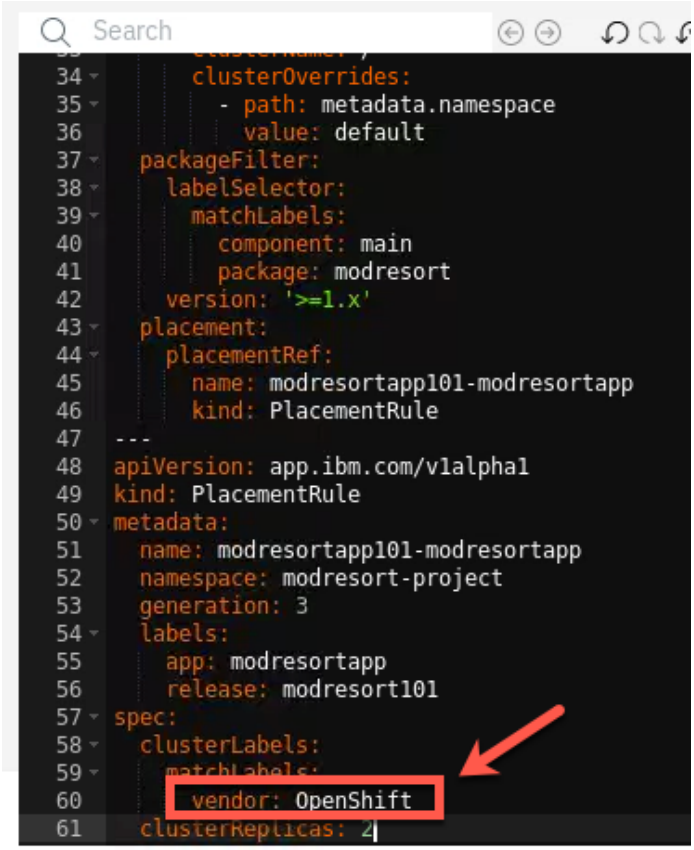
Great, you moved the application from one cluster to the other.

__9. What if you want to run the application on both the clusters? This is possible by changing clusterReplicas and matchingLabel. Let's do it!

Back to the Multicluster Management page, on the Editor, change **clusterReplicas** to **2**.

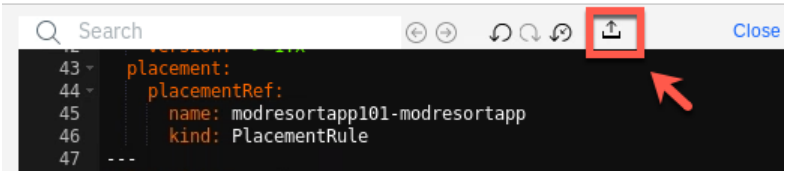


__10. On the spec **matchLabels** setting, **remove** the **environment** label, and add **vendor** label with value **Openshift**.

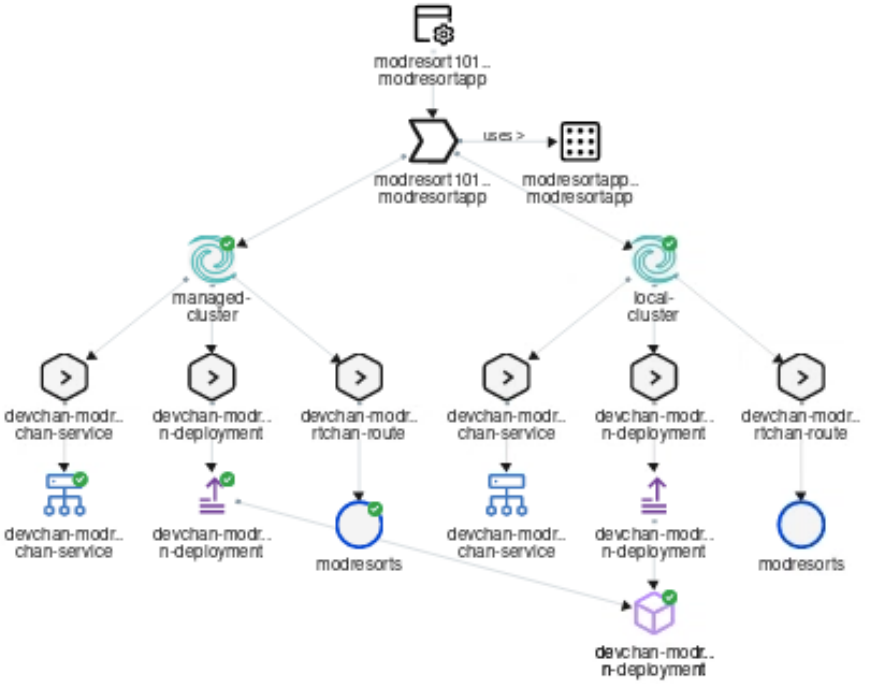


The label, vendor, exists on both the clusters and the value matches OpenShift.

__11. Apply the changes by clicking update icon.



__12. Within a few minutes, the resource overview refreshes, indicating that the application is running on both clusters. You can validate by checking the status of pods on both the clusters.

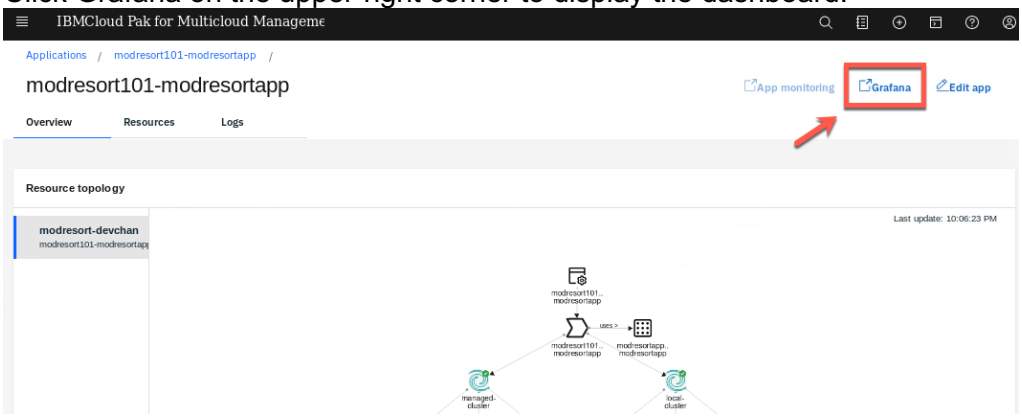


Great, now you have your application on both clusters. Next section, you explore Grafana dashboard to visualize the application health across clusters.

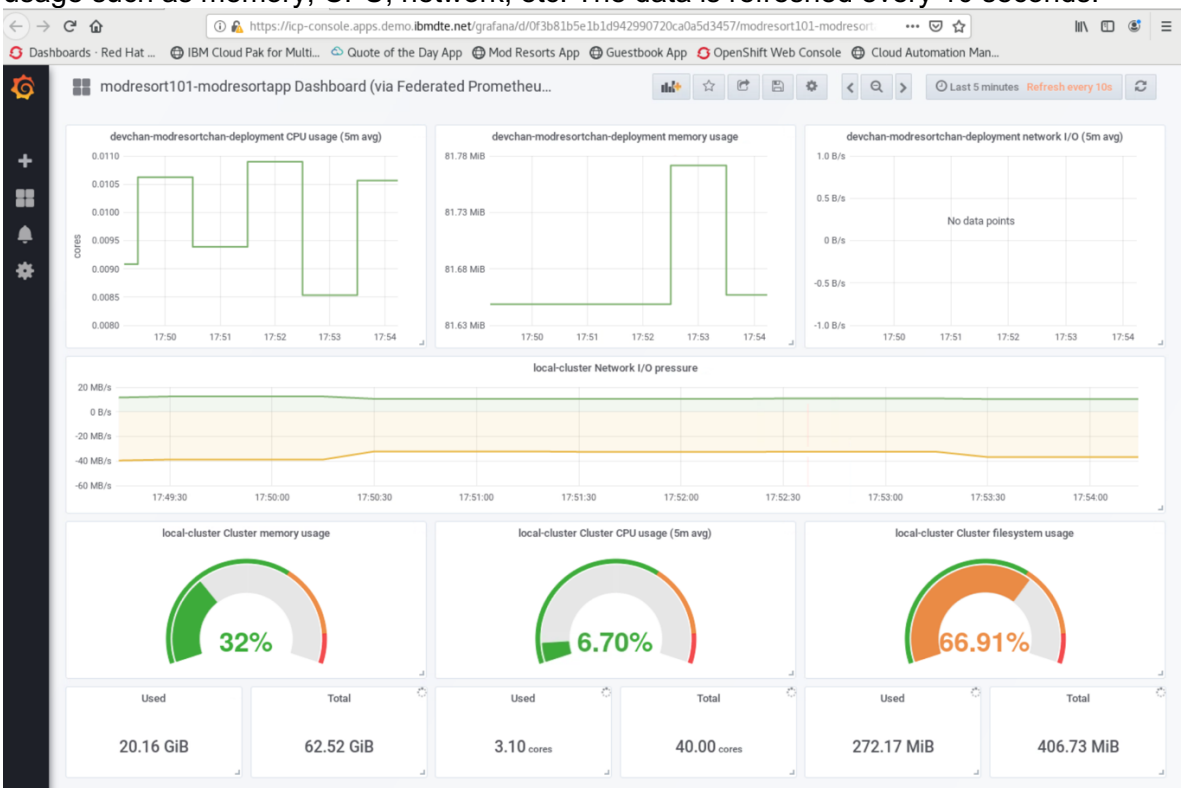
4.8 Visualize Application Health using Grafana

Cloud Pak for Multicloud Management provides Grafana dashboard via Federated Prometheus to visualize the application health across clusters.

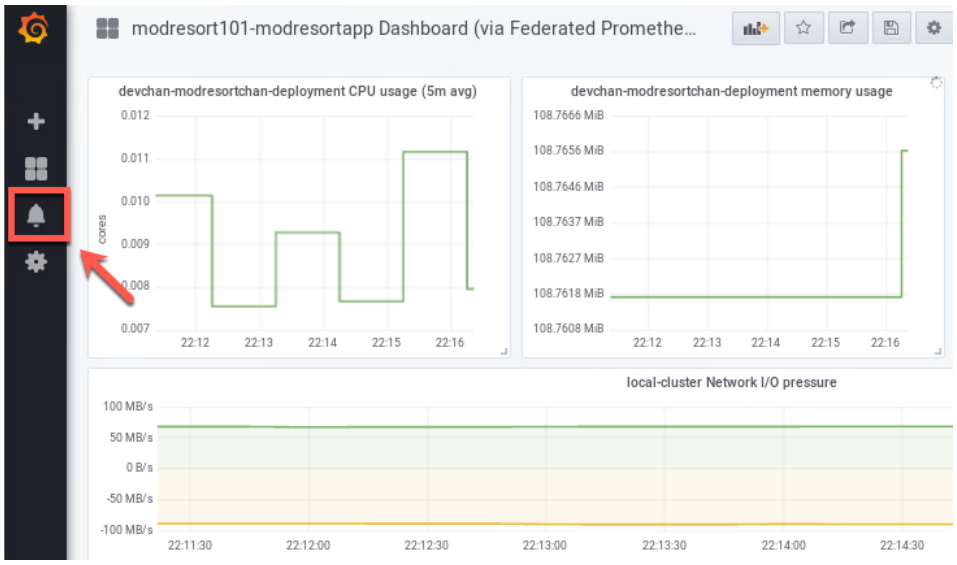
__1. Click Grafana on the upper-right corner to display the dashboard.



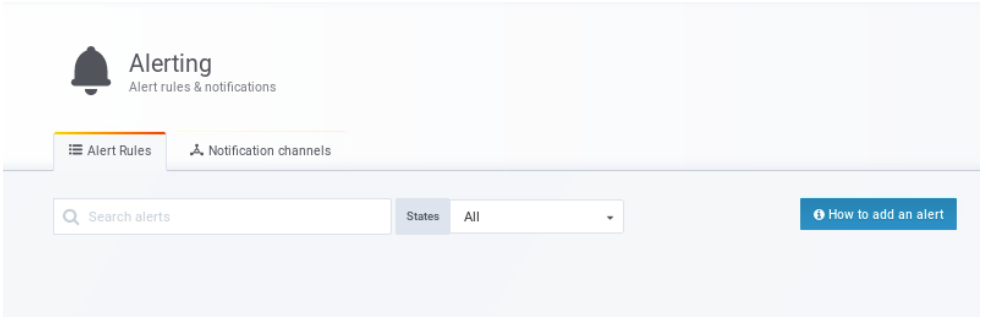
__2. From the Federated Prometheus Dashboard, you can see details on your application health and usage such as memory, CPU, network, etc. The data is refreshed every 10 seconds.



__3. Click on the **Alert** button.



__4. On the Alert page, you are able to create Alert based on Alert Rules and define Notification channels.



Congratulations! You have successfully completed the lab “Application Management with IBM Cloud Pak for Multicloud Management”.

Summary

You completed the Cloud Pak for Multicloud Management tutorial: Multi-cluster Management. Throughout the tutorial, you explored the key takeaways:

- Understand Cloud Pak for Multicloud Management
- Define an application with Channels and Subscriptions
- Deploy the application chart from the catalog
- Modify Placement Policies to move application resources across clusters
- Check the application health by using Federated Prometheus Dashboard

If you would like to learn more about Cloud Pak for Multicloud Management, please refer:

- Cloud Pak for Multicloud Management home page
- Cloud Pak for Multicloud Management Demos

Appendix A. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. All references to fictitious companies or individuals are used for illustration purposes only.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Appendix B. Trademarks and copyrights

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	AIX	CICS	ClearCase	ClearQuest	Cloudscape
Cube Views	DB2	developerWorks	DRDA	IMS	IMS/ESA
Informix	Lotus	Lotus Workflow	MQSeries	OmniFind	
Rational	Redbooks	Red Brick	RequisitePro	StrongLoop	System i
System z	*Tivoli*	*WebSphere*	*Workplace*	*System p*	

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

NOTES

NOTES

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

Please Recycle