Portal on OOM

The new Beijing release scalability, resiliency, and manageability are described here. These capabilities apply to the OOM/Kubernetes installation.

Installation

Follow the OOM installation instructions at


TLab specific Upgrade of Portal

Below are the notes specific for the TLab environment. If a change is made in OOM charts then the following needs to be completed to upgrade the running charts in

a. Find Rancher IP in R_Control-Plane tenant - we use "onap_dev" key and "Ubuntu" user to SSH. For example: “ssh -i onap_dev ubuntu@192.168.xx.xxx”.
   b. Login as ubuntu, then run “sudo -i” to login as root. The “oom” git repo is in the rancher vm’s root directory, under “/root/oom”.
   c. Edit portal files at /root/oom/kubernetes/portal....
   d. When complete run the following from /root/oom/kubernetes dir

   > make portal
   ....
   > make onap
   ....
   # This will do the upgrade:
   helm upgrade dev local/onap --namespace onap -f onap/resources/environments/disable-allcharts.yaml
   --set portal.enabled=true
   # When complete run the following to ensure its in the process of coming up or complete.
   kubectl get pods -n onap -o wide | grep portal

   In case, if it is a fresh install, try below command:

   helm install ./onap/ -n onap --namespace onap -f onap/resources/environments/disable-allcharts.yaml
   --set portal.enabled=true

   Stop command to stop any old namespace:

   helm del onap --purge

e. Rancher gui is at 192.168.xx.xxx:8080
   There is an interactive cli as well in the rancher gui where you can run kubectl commands.
   f. Below command will show the list of portal services.

   kubectl get services --namespace=onap | grep portal

Accessing the ONAP Portal using OOM and a Kubernetes Cluster

If everything is successful, then to access Portal - http://onap.readthedocs.io/en/latest/submodules/oom.git/docs/onap_user_guide.html#accessing-the-onap-portal-using-oom-and-a-kubernetes-cluster

Overview of the running system
kubectl get services --namespace=onap | grep portal

<table>
<thead>
<tr>
<th>Service</th>
<th>Type</th>
<th>IP Address</th>
<th>Protocol</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>portal-app</td>
<td>LoadBalancer</td>
<td>10.43.141.57</td>
<td>10.0.0.8:8989, 10.0.0.8:30215, 8010:30214</td>
<td>5d</td>
</tr>
<tr>
<td>portal-cassandra</td>
<td>ClusterIP</td>
<td>10.43.158.145</td>
<td>9160/TCP, 7000/TCP, 7001/TCP, 7199/TCP, 9042</td>
<td>5d</td>
</tr>
<tr>
<td>portal-db</td>
<td>ClusterIP</td>
<td>10.43.192.65</td>
<td>&lt;none&gt;</td>
<td>5d</td>
</tr>
<tr>
<td>portal-sdk</td>
<td>ClusterIP</td>
<td>10.43.24.82</td>
<td>&lt;none&gt;</td>
<td>5d</td>
</tr>
<tr>
<td>portal-widget</td>
<td>ClusterIP</td>
<td>10.43.101.233</td>
<td>&lt;none&gt;</td>
<td>5d</td>
</tr>
<tr>
<td>portal-zookeeper</td>
<td>ClusterIP</td>
<td>10.43.0.82</td>
<td>&lt;none&gt;</td>
<td>5d</td>
</tr>
</tbody>
</table>

kubectl get pods --all-namespaces | grep portal

<table>
<thead>
<tr>
<th>Pod</th>
<th>Status</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev-portal-app-b8c6668d8-56b3f</td>
<td>2/2 Running</td>
<td>2m</td>
</tr>
<tr>
<td>dev-portal-app-b8c6668d8-g6whb</td>
<td>2/2 Running</td>
<td>2m</td>
</tr>
<tr>
<td>dev-portal-app-b8c6668d8-xshwg</td>
<td>2/2 Running</td>
<td>2m</td>
</tr>
<tr>
<td>dev-portal-cassandra-5dd3b059d-fc62f</td>
<td>1/1 Running</td>
<td>2m</td>
</tr>
<tr>
<td>dev-portal-db-6e7658648-sp9sf</td>
<td>0/1 Running</td>
<td>2m</td>
</tr>
<tr>
<td>dev-portal-sdk-8696596d47-55jxx</td>
<td>0/2 Init:0/1</td>
<td>2m</td>
</tr>
<tr>
<td>dev-portal-widget-694c45175f-nqdt</td>
<td>0/1 Init:0/1</td>
<td>2m</td>
</tr>
<tr>
<td>dev-portal-zookeeper-db466fc-kgsaw</td>
<td>1/1 Running</td>
<td>2m</td>
</tr>
</tbody>
</table>

**Healthchecks**

Verify that the portal healthcheck passes by the robot framework:


<table>
<thead>
<tr>
<th>Basic Portal Health Check</th>
<th>Result</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>yes</td>
<td>PASS</td>
</tr>
</tbody>
</table>

**portal-app Scaling**

To scale a new portal-app, set the replica count appropriately.

In our tests below, we are going to work with the OOM portal component in isolation. In this exercise, we scale the portal-app with 2 new replicas.

The below code needs to be added to the integration-override.yaml file.

```yaml
portal:
  portal-app:
    replicaCount: 3
```

Run below command from "root" folder to do helm upgrade

```bash
helm upgrade -i dev local/onap -f integration-override.yaml
```

**portal-app Resiliency**

A portal-app container failure can be simulated by stopping the portal-app container. The Kubernetes liveness operation will detect that the ports are down, inferring there's a problem with the service, and in turn, will restart the container.

Here is where one of the running instances was deleted. And another is starting.
Here the new instance has started and the old one is still terminating.

After the deleted instance has terminated the 3 instances are all running normally.

During this time there was no issues with the Portal Website.

TODO: In the following releases, the portal-db, portal-widget needs to be tested with similar resiliency and scaling tests. The portal-cassandra and portal-zookeeper requires coordination with MUSIC team to test the resiliency as there are some plans to provide MUSIC as a service.

Sanity Tests

Check Portal UI and perform sanity tests.

1. After 3 instances of Portal are up, edit IP in /etc/hosts file, and logon as demo user on http://portal.api.simplesdemo.onap.org:30215/ONAPPORTAL/login.htm
2. Then killed 1 instance, I am able to continue on Portal page seamlessly.

3. Another test on failover timing, when killed all 3 instances, the new Portal processes are coming up within 30 seconds.

**Troubleshoot**

In case of failures, below commands may get handy for accessing the pods/logs and troubleshooting the issue.

To find all portal pods:

```bash
> kubectl get pods --all-namespaces | grep portal
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Namespace</th>
<th>Image</th>
<th>Status</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>onap dev-portal-app-b8c6668d8-f6b6b</td>
<td>onap</td>
<td>dev-portal-app-b8c6668d8-f6b6b</td>
<td>Running</td>
<td>2m</td>
</tr>
<tr>
<td>onap dev-portal-app-b8c6668d8-q69bb</td>
<td>onap</td>
<td>dev-portal-app-b8c6668d8-q69bb</td>
<td>Running</td>
<td>2m</td>
</tr>
<tr>
<td>onap dev-portal-app-b8c6668d8-x69wg</td>
<td>onap</td>
<td>dev-portal-app-b8c6668d8-x69wg</td>
<td>Running</td>
<td>2m</td>
</tr>
<tr>
<td>onap dev-portal-cassandra-5d9b5f9f-c68rp</td>
<td>onap</td>
<td>dev-portal-cassandra-5d9b5f9f-c68rp</td>
<td>Running</td>
<td>2m</td>
</tr>
<tr>
<td>onap dev-portal-db-6d7f58648-sp9sf</td>
<td>onap</td>
<td>dev-portal-db-6d7f58648-sp9sf</td>
<td>Running</td>
<td>2m</td>
</tr>
<tr>
<td>onap dev-portal-sdk-868f869cd7-8b7jk</td>
<td>onap</td>
<td>dev-portal-sdk-868f869cd7-8b7jk</td>
<td>Init:0/1</td>
<td>2m</td>
</tr>
<tr>
<td>onap dev-portal-widget-694c46575f-n9qdt</td>
<td>onap</td>
<td>dev-portal-widget-694c46575f-n9qdt</td>
<td>Init:0/1</td>
<td>2m</td>
</tr>
<tr>
<td>onap dev-portal-zookeeper-8b4667b-kggsw</td>
<td>onap</td>
<td>dev-portal-zookeeper-8b4667b-kggsw</td>
<td>Running</td>
<td>2m</td>
</tr>
</tbody>
</table>

From above list, now to check DB logs:

```bash
> kubectl logs --namespace=onap dev-portal-db-6d7f58648-sp9sf -c portal-db
```

2018-05-30 19:49:16 139875765802880 [Note] mysqld (mysqld 10.2.15-MariaDB-10.2.15+maria~jessie) starting as process 1 ...
2018-05-30 19:49:16 139875765802880 [Note] InnoDB: Compressed tables use zlib 1.2.8
2018-05-30 19:49:16 139875765802880 [Note] InnoDB: Number of pools: 1

To check portal-app logs:

```bash
> kubectl logs --namespace=onap dev-portal-app-b8c6668d8-56b6b -c portal-app
```

/start-apache-tomcat.sh: option -i value is
/start-apache-tomcat.sh: option -n value is
/start-apache-tomcat.sh: values for IP (-i) and/or name (-n) are empty or short
/start-apache-tomcat.sh: Starting server from /opt/apache-tomcat-8.0.37

To get inside the portal-app docker and access the application logs:

References:

1. K8S / helm basic commands for ONAP integration