

Magnum with CAPI

This describes how to setup Magnum K8SaaS via the Vexxhost Capi Driver which is pre-bundled now with the OSISM Magnum Images.

Requirements

set `enable_magnum: yes` in your `kolla/configurations.yml` (if not already present before, reconfigure `loadbalancer` and `horizon` afterwards with `osism apply -a reconfigure loadbalancer; osism apply -a reconfigure horizon`)

K8S CAPI Cluster

You need a K8S Base Cluster to bootstrap the CAPI/CAPO Parts. You can use the OSISM K8S Cluster with it if you like, then you need to perform the following steps on your manager node:

```
osism apply kubernetes
osism apply copy-kubeconfig
```

`copy-kubeconfig` will prepare the `kubeconfig` file for magnum by copying it to the `kubeconfig` overlay file in your repository. It is important that this K8S Cluster can be reached from your `magnum-conductor` nodes and reverse the `kubernetes` cluster needs to be able to contact the `openstack apis` (`neutron`, `keystone`, `cinder`, `nova`)

Once you have your `kubernetes` cluster install `kubectl` as example on your manager node with

```
snap install kubectl
```

set your `kubeconfig` to your cluster:

```
export KUBECONFIG=/opt/configuration/environments/kolla/files/overlays/magnum/kubeconfig
```

then download `clusterctl`:

```
wget https://github.com/kubernetes-sigs/cluster-api/releases/download/v1.8.0-beta.1/clusterctl-linux-amd64
sudo mv clusterctl-linux-amd64 /usr/local/bin/clusterctl
chmod +x /usr/local/bin/clusterctl
```

Then initialize the cluster apis:

```
export EXP_CLUSTER_RESOURCE_SET=true
export EXP_KUBEADM_BOOTSTRAP_FORMAT_IGNITION=true
export CLUSTER_TOPOLOGY=true

clusterctl init --infrastructure openstack
```

Prepare Nova Policy

Nova needs a policy which allows 0 disk drives in case of some SCS flavors to allow normal users to spool up k8s clusters. Add the file

/opt/configuration/environments/kolla/files/overlays/nova/policy.yaml with the following contents:

```
os_compute_api:servers:create:zero_disk_flavor: "role:admin or role:member"
```

Reconfigure Nova:

```
sudo su - dragon
osism apply -a reconfigure nova
```

Magnum

Once you have your K8S Cluster in Place you can just roll out magnum with "osism apply magnum".

In case you run images before 2024.1 you need to remove the capi helm stackHPC driver manually in the magnum-conductor containers, you can do this by running on each controller:

```
docker exec -u root -it magnum_conductor bash
pip uninstall magnum-capi-helm
exit
docker restart magnum_conductor
```

Prepare CAPI Images

You can use the OSISM Manage Command to Download the Latest CAPI Images. Just run the following to get the latest CAPI images (you need secrets defined in environment/openstack before, please check osism docs)

```
osism manage image clusterapi --cloud admin
```

Once your images are available you can create Cluster Templates.

Prepare Templates

```
openstack coe cluster template create \  
  --image IMAGEIDOFK8sIMAGE \  
  --external-network NAMEOFOUREXTERNALNETWORK \  
  --dns-nameserver 9.9.9.9 \  
  --master-lb-enabled \  
  --master-flavor SCS-2V-4 \  
  --flavor SCS-2V-4 \  
  --network-driver cilium \  
  --docker-storage-driver overlay2 \  
  --volume-driver cinder \  
  --coe kubernetes \  
  --label kube_tag=v1.30.3 --label keystone_auth_enabled=false \  
  k8s-v1.30.3;
```

Spool Up Clusters

Afterwards you can easily spool up clusters with the openstack coe cli command or via horizon by just selecting the correct template (maybe make it public before)

Limitations

Because Magnum was before the CAPI Drivers pretty much useless alot of features in horizon (and even skyline) stil try to use heat despite another drive is available. What works in horizon ist:

- Create Clusters
- Delete Clusters
- Rolling Upgrades

In latest Skyline Magnum doesnt work at all with CAPI as it gives wrong parameters (image uuid instead of name as example). Resize doesnt work at all as the horizon implementation tries hard todo this via the old heat api calls.

You can resize nodes via the CLI by utilizing as example:

```
openstack coe cluster resize $CLUSTERID 4
```

this will make sure 4 worker nodes are in the cluster (as example if you had less then 4 before). You can only scale workers, not masters. This works both way for up and downscaling.

Debugging Issues

If something is wrong your main point to look is the magnum-conductor.log , atm there is a SQLAlchemy Bug which requires a few restarts each day or queue tuning (<https://bugs.launchpad.net/magnum/+bug/2067345>) . API initial Errors such as "could not create cluster" may appear sometimes, often a retry fixes this, CLI or API is usually better then horizon and does not show this errors.

To see your K8S Clusters connect to your K8S CAPI Cluster with kubectl (see first steps) and run:

```
oot@capi-testbed-magnum-master:/home/ubuntu# kubectl get openstackclusters -A
NAMESPACE   NAME                CLUSTER   READY NETWORK                               BASTION IP   AGE
magnum-system kube-d3qay-d79w7    kube-d3qay true  fa9f5fca-d5a2-4466-a829-6db178767f8c
30m
```

this will show you the list of existing k8s clusters spooled up via K8S and the status. You can debug this like any other normal k8s ressource. (describe, logs etc.)

Get Machine deployments:

```
root@capi-testbed-magnum-master:/home/ubuntu# kubectl get machinedeployments -A
NAMESPACE   NAME                                CLUSTER   REPLICAS  READY  UPDATED  UNAVAILABLE  PHASE
AGE  VERSION
magnum-system kube-d3qay-default-worker-x87nw    kube-d3qay  2         2        2         0           Running
31m  v1.29.6
```

Describe a Machine:

```
root@capi-testbed-magnum-master:/home/ubuntu# kubectl describe machine kube-d3qay-default-worker-x87nw -n magnum-system
Name:      kube-d3qay-default-worker-x87nw-dvmxl-4v67m
Namespace: magnum-system
Labels:    cluster.x-k8s.io/cluster-name=kube-d3qay
           cluster.x-k8s.io/deployment-name=kube-d3qay-default-worker-x87nw
           cluster.x-k8s.io/set-name=kube-d3qay-default-worker-x87nw-dvmxl
           machine-template-hash=3063136496-dvmxl
           node-role.kubernetes.io/worker=
           node.cluster.x-k8s.io/nodegroup=default-worker
           topology.cluster.x-k8s.io/deployment-name=default-worker
           topology.cluster.x-k8s.io/owned=
Annotations: <none>
.
```

.
.

It can happen that openstack sometimes may not delete cluster properly. Then you can initiate the delete directly from the capi k8s cluster:

```
kubectl delete cluster kube-d3qay -n magnum-system
```

which will remove all resources properly in openstack.

Date: 29.07.2024, tested with K8S 1.28.x - 1.30.x

Revision #1

Created 28 July 2024 21:33:55 by Enrico Kern

Updated 28 July 2024 22:03:03 by Enrico Kern