Creating new Kubernetes clusters

By employing OpenStack Magnum you can easily create Kubernetes clusters over OpenStack, using either the Cleura Cloud Management Panel or the OpenStack CLI. Let us demonstrate the creation of a Kubernetes cluster following both approaches.

Prerequisites

First and foremost, you need an account in Cleura Cloud. If you prefer to work with the OpenStack CLI, go ahead and enable it first. Then, in addition to the Python openstackclient module, make sure you also install the corresponding plugin module for Magnum. Use either the package manager of your operating system or pip:

Debian/Ubuntu Mac OS X with Homebrew Python Package

apt install python3magnumclient

This Python module is unavailable via

Creating a Kubernetes cluster

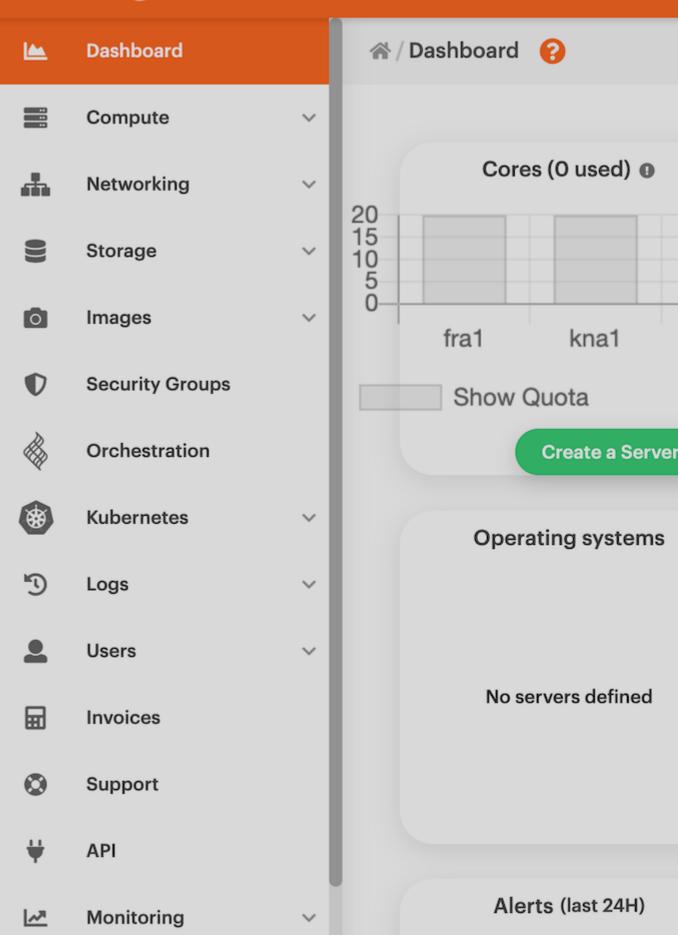
pip create a Kubernetes cluster from the Cleura Cloud Management Panel, fire up

b browser, navigate to the Cleura Cloud Management Panel start

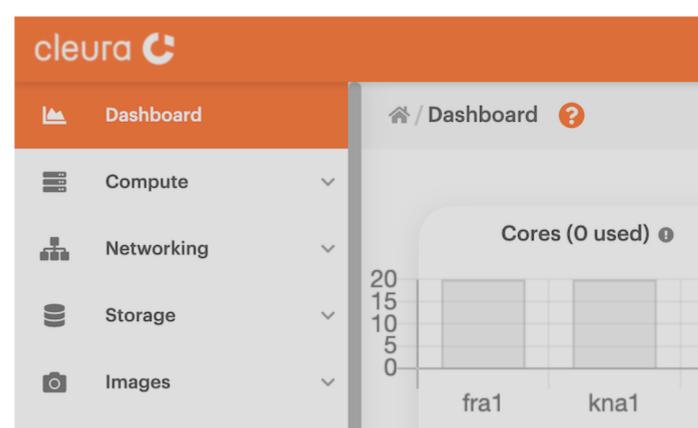
pip install
pythonmagnumclient to your Cleura Cloud account. On the other hand, if you prefer to
penStack CLI, please do not forget to source the RC file first.

On the top right-hand side of the Cleura Cloud Management Panel, click the *Create* button. A new

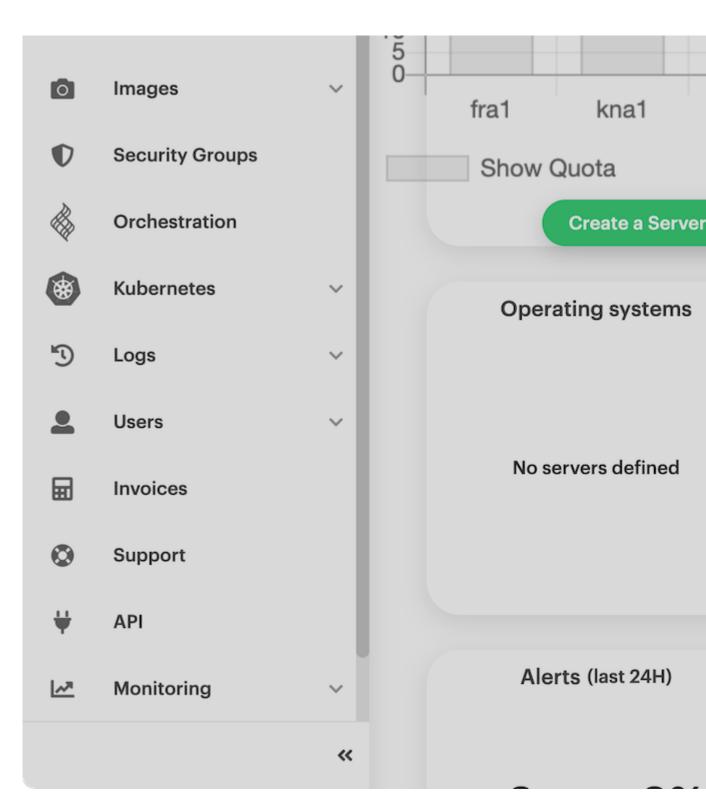
cleura 📞



You will notice several rounded boxes on that pane. Go ahead and click the one labeled $Magnum\ C$



Then, select one of the available templates to base the new cluster on. In the example below, we have about to be deployed, the characteristics of the cluster nodes, the operating system they run, and the characteristics of the cluster nodes.



For now, you may skip the *Advanced Option* section. Click the green *Create* button, and Magnum v A simple, general command for creating a new Kubernetes cluster with Magnum looks like this:

```
openstack coe cluster create \
--cluster-template $CLUSTER_TMPL \
--keypair $KEYPAIR \
$CLUSTER_NAME
```

Let us list all available templates in the region:

openstack coe cluster template list

Select the template you want by setting the corresponding uuid value to the CLUSTER_TMPL variable

```
CLUSTER_TMPL="f9e1a2ea-b1ff-43e7-8d1e-6dd5861b82cf" # just an example
```

Then, list all available keypairs...

```
openstack keypair list
```

```
+-----+
| Name | Fingerprint | Type |
+-----+
| husavik | 34:3b:58:ba:ec:95:f5:17:17:df:04:38:11:89:e6:3d | ssh |
+-----+
```

...and set the KEYPAIR variable to the name of the keypair you want:

```
KEYPAIR="husavik" # again, this is just an example
```

Finally, decide on a name for your new Kubernetes cluster:

```
CLUSTER_NAME="bangor"
```

With everything in place, go ahead and create your new Kubernetes cluster:

```
openstack coe cluster create \
--cluster-template $CLUSTER_TMPL \
--keypair husavik
bangor
```

If everything went well with your request for a new cluster, on your terminal, you would see a mes

```
Request to create cluster e0df8c62-c6f6-4c7d-b67e-33e3606e9ab6 accepted
```

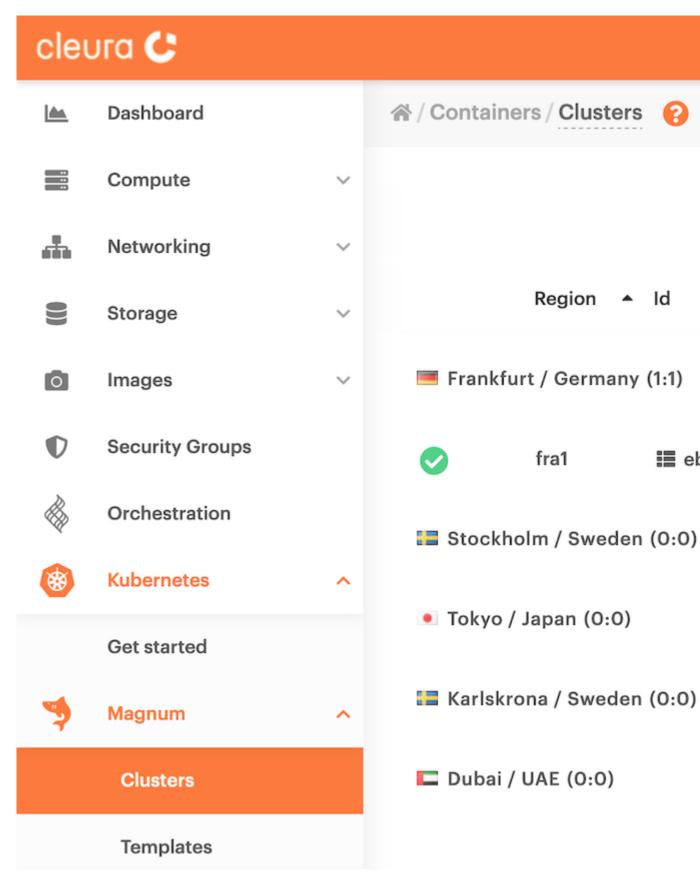
The cluster creation process takes some time to complete, and while you are waiting, you can check

If everything is going well, the message you will get will be ${\tt CREATE_IN_PROGRESS}$. When Magnum

Viewing the Kubernetes cluster

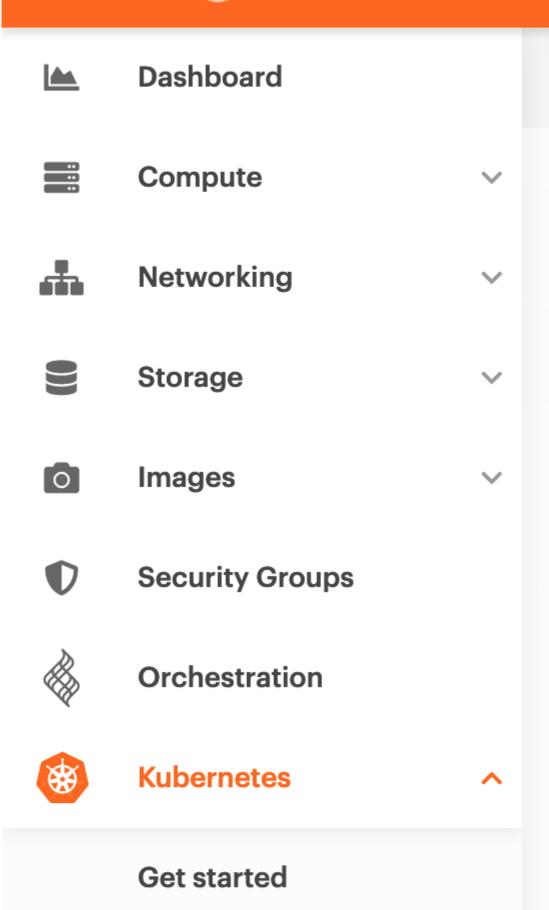
After the Kubernetes cluster is ready, you may at any time view it and get detailed information about it.

Claura Cloud Management Panel OpenStack CLI
In the left vertical pane of the Cleura Cloud Management Panel, click through Kubernetes, Magnus





cleura Ċ



☆/ Con

👅 Fra

②

N

M

D

To list all available Kubernetes clusters, just type:

For many more details on a specific cluster, note its name and run a command like this:

```
openstack coe cluster show bangor
```

```
| Value
              | CREATE COMPLETE
status
| health status
              | HEALTHY
|\ cluster\_template\_id\ |\ f9e1a2ea-b1ff-43e7-8d1e-6dd5861b82cf
| node addresses
                 ['185.52.156.105']
uuid
             | e0df8c62-c6f6-4c7d-b67e-33e3606e9ab6
| stack_id
              | e3725aed-f665-4e8d-9409-85f5ee5e2f4a
status reason
                None
              | 2022-11-14T07:32:02+00:00
created at
              2022-11-14T07:37:26+00:00
| updated_at
                | v1.18.6
coe version
labels
              | {'kube tag': 'v1.18.6', 'heat container agent tag': 'train-stable'}
| labels_overridden | {}
| labels skipped | {}
labels added
                 1 { }
| fixed_network
                 None
| fixed subnet
                 | None
| floating ip enabled | True
| faults
               | husavik
keypair
               | https://89.46.80.136:6443
api address
| master_addresses | ['89.46.80.136']
| master lb enabled | False
create timeout
                 | 60
| node count
                 | 1
discovery url
                 https://discovery.etcd.io/23af721dc3ee773d2674db4881ff70cb
docker volume size | 50
master count | 1
| container_version | 1.12.6
name
              | bangor
| master_flavor_id | 2C-4GB-20GB
              | 2C-4GB-20GB
| flavor id
| health status reason | {'bangor-id6nijycp2wy-master-0.Ready': 'True', 'bangor-id6nijycp2wy-
            | node-0.Ready': 'True', 'api': 'ok'}
```

Accessing the Kubernetes cluster with kubectl

You may install the Kubernetes command line tool, kubectl, on your local computer, and run commands against your cluster. To install kubectl, use the package manager of your operating system.

Debian/Ubuntu **Mac OS X with Homebrew**

apt install kubectl

brew install

ining commands against a specific cluster, you must have the kubectl ding config file on your computer.

Cleura Cloud Management Panel OpenStack CLI

Downloading

a config file

from the

Cleura

Cloud

Management

Panel is

currently not

supported.

You can still

fetch the

config file of

your newly

created

Kubernetes

cluster using

the

OpenStack

CLI.

To download

the config

file for your

Kubernetes

cluster, type

the

following:

Then, you can use kubectl to run commands against your cluster. See, for instance, all cluster nodes...

```
NAME STATUS ROLES AGE VERSION
bangor-id6nijycp2wy-master-0 Ready master 113m v1.18.6
bangor-id6nijycp2wy-node-0 Ready <none> 111m v1.18.6
```

...or all running pods in every namespace:

kubectl get pods --all-namespaces

| NAMESPACE | NAME | READY STATUS RESTARTS AGE |
|-------------|----------------------------------|---------------------------------|
| kube-system | coredns-786ffb7797-tw2hg | 1/1 Running 0 167m |
| kube-system | coredns-786ffb7797-vbqwn | 1/1 Running 0 167m |
| kube-system | csi-cinder-controllerplugin-0 | 5/5 Running 0 167m |
| kube-system | csi-cinder-nodeplugin-4nr69 | 2/2 Running 0 166m |
| kube-system | csi-cinder-nodeplugin-vtwqf | 2/2 Running 0 167m |
| kube-system | dashboard-metrics-scraper-6b488 | 84c9d5-4mlrg 1/1 Running 0 167m |
| kube-system | k8s-keystone-auth-wk5v2 | 1/1 Running 0 167m |
| kube-system | kube-dns-autoscaler-75859754fd- | -2wsd9 1/1 Running 0 167m |
| kube-system | kube-flannel-ds-7z9dp | 1/1 Running 0 167m |
| kube-system | kube-flannel-ds-dmvk6 | 1/1 Running 0 166m |
| kube-system | kubernetes-dashboard-c98496485 | 5-stn42 1/1 Running 0 167m |
| kube-system | magnum-metrics-server-79556d6 | 6999-xdlpm 1/1 Running 0 167m |
| kube-system | npd-5p6gk | 1/1 Running 0 165m |
| kube-system | openstack-cloud-controller-manag | ger-44rz9 1/1 Running 0 167m |

Last update: 2022-11-29 Created: 2022-11-14

Authors: Christos Varelas, Florian Haas