OpenStack Troubleshooting Tool Box
Walking the Great Wall of Containers

https://tinyurl.com/y5w749j9

Keith Berger
Senior Software Engineer
SUSE/kberger@suse.com

Nicolas Bock
Senior Software Engineer
SUSE/nbock@suse.com

Agenda

- Introduction
- Toolbox Environments
- Useful Troubleshooting Tools
- Moving around K8s (Kubernetes)
- How OpenStack Services are Deployed on K8s
- Basic K8s Troubleshooting
- High Availability, Scaling, and Service Recovery in K8s
- Debugging OpenStack on K8s
- Q&A

Introduction

Introduction

The goal of this session is to provide tips and tricks to troubleshoot an OpenStack cloud that is deployed using OpenStack Helm and is running in a K8s environment.

We adapt. You succeed.

Toolbox Environments

Toolbox Environments

The environments for this session are using an OpenStack Helm (OSH) instance running on a Ubuntu 16.04 instance host. The container images are based on Ubuntu 16.04 and openSUSE Leap 15.0. This deployment is only meant for demo purposes.

Each student will be given an IP address for the environment they will be using for this workshop. The login user and password are:

```
workshop/oshTS2019!
```

Useful Troubleshooting Tools

Useful Troubleshooting Tools

- **docker (runtime container engine)**

  https://docs.docker.com/engine/reference/commandline/cli/
- **kubeadm (tool to deploy k8s cluster)**

  https://kubernetes.io/docs/reference/setup-tools/kubeadm/kubeadm/
- **kubectl (tool to run commands on a k8s cluster)**

  https://kubernetes.io/docs/reference/kubectl/
- **helm (k8s package manager)**

  https://helm.sh/docs/helm/#helm
- **netstat (linux command used to examine ports and connections)**

  http://manpages.ubuntu.com/manpages/xenial/man8/netstat.8.html

Moving around K8s

Moving around K8s

kubeadm config view – Provide a description of the k8s cluster

kubeadm config images list – List the images used to deploy the k8s cluster

kubectl get namespaces – List the namespaces used in the k8s cluster

Namespaces are used as "selectors" for the following commands

kubectl get pods – List the pods deployed in the k8s cluster

kubectl get nodes – List the pods deployed in the k8s cluster

kubectl describe pods – Print the configuration of a pod in the k8s cluster

kubectl describe nodes – Print the configuration of a pod in the k8s cluster

kubectl exec – Execute a command on a specified pod in the k8s cluster

kubectl logs – Print the logs for a specified pod in the k8s cluster

Exercise 1

1. **Using** *kubeadm config view*, **what is the value of podSubnet?**
2. **Using** *kubeadm config images list,* **what version of etcd is used?**
3. **Using** *kubectl get namespaces*, **how many namespaces exist?**
4. **Using** *kubectl get pods,* **how many neutron pods are deployed?**
5. **Using,** *kubectl describe pods,* **what is the IP of the neutron-dhcp-agent pod?**
6. **Using** *kubectl logs*, **view the nova-compute logs?**
7. **Using** *kubectl exec –ti,* **find the rabbitmq cluster status?**

Exercise 1- Solution

What is the podSubnet of the k8s cluster?

```
kubeadm config view
…
podSubnet: 192.168.0.0/16
```

What version of etcd is used in the k8s cluster?

```
kubeadm config images list
…
k8s.gcr.io/etcd:3.2.24
```

How many namespaces are deployed in the k8s cluster?

```
kubectl get namespaces
…
ceph, default, kube-public, kube-system, nfs, openstack
```

Exercise 1- Solution

How many neutron pods are deployed in the k8s cluster?

```
kubectl get pods --all-namespaces
…
neutron-db-init-snkkb, neutron-db-sync-mfd4s, neutron-dhcp-
agent-default-bqcjl, neutron-ks-endpoints-5wgd7, neutron-ks-
service-fd2q2, neutron-ks-user-tnqkv, neutron-l3-agent-default-
b2gvb, neutron-metadata-agent-default-lw9d4, neutron-ovs-agent-
default-84stb, neutron-rabbit-init-mbb8p, neutron-server-
5f97476b6d-hf7l7
```

What is the IP of the neutron-dhcp-agent pod?

```
kubectl describe pods neutron-dhcp-agent-default-bqcjl --namespace
openstack
…
IP:                    172.17.0.1
```

Exercise 1- Solution

View the nova-compute logs

```
kubectl logs nova-compute-default-thmnk --namespace openstack
```

Find the rabbitmq cluster status

```
kubectl exec -ti  rabbitmq-rabbitmq-0 --namespace openstack bash
rabbitmq@rabbitmq-rabbitmq-0:/$ rabbitmqctl cluster_status
Cluster status of node rabbit@rabbitmq-rabbitmq-
0.rabbitmq.openstack.svc.cluster.local ...
[{nodes,
     [{disc,
          ['rabbit@rabbitmq-rabbitmq-0.rabbitmq.openstack.svc.cluster.local',
           'rabbit@rabbitmq-rabbitmq-1.rabbitmq.openstack.svc.cluster.local']}]},
 {running_nodes,
….
….
```

How OpenStack services are deployed on K8s

How OpenStack services are deployed on K8s

OpenStack-Helm provides a collection of Helm charts that simply, resiliently, and flexibly deploy OpenStack and related services on Kubernetes.

The charts for OpenStack and the dependent services are located in two repos:

https://github.com/openstack/openstack-helm

https://github.com/openstack/openstack-helm-infra

How OpenStack services are deployed on K8s

Each chart has a "node_selector_key" that is checked against a node "Label" to determine if that chart can be deployed on that node. In addition, all of the service specific parameters are defined in the chart.

kubectl get nodes – List k8s cluster nodes

kubectl describe nodes – Print the configuration of a k8s cluster node

helm ls – List deployed charts in the k8s cluster

helm status – Print resource information for a chart in the k8s cluster

helm search – Searched for a chart in the k8s cluster

helm inspect – Print configuration details for a chart in the k8s cluster

helm delete – Delete a chart in the k8s cluster

Exercise 2

1. Using *kubectl describe nodes,* what are the roles and labels of the node?

   Hint for #2 and #3: Preface the helm chart with *local/*

2. Using helm inspect, what are the node_selector_keys for the neutron chart?
3. Using *helm inspect*, what is the nova database user and password?
4. Using *kubectl exec –ti, c*onnect to the mariadb pod and run mysql as the nova user. Verify you can access the database.

   Hint: Run *mysql --user=<user> --password=<password>*

Exercise 2 - Solution

What are the roles and labels of the k8s node?

```
kubectl describe nodes nbock-osh
Name:               nbock-osh
Roles:              master
Labels:             beta.kubernetes.io/arch=amd64
                          beta.kubernetes.io/os=linux
                          ceph-mds=enabled
…
                          openstack-compute-node=enabled
                          openstack-control-plane=enabled
…
```

Exercise 2 - Solution

What are the node_selector_keys for the neutron chart?

```
helm inspect local/neutron
…
labels:
  agent:
    dhcp:
      node_selector_key: openstack-control-plane
      node_selector_value: enabled
    l3:
      node_selector_key: openstack-control-plane
      node_selector_value: enabled
    metadata:
      node_selector_key: openstack-control-plane
      node_selector_value: enabled
…
```

Exercise 2 - Solution

What is the DB password for the nova user in the nova chart?

```
helm inspect local/nova
…
 oslo_db_api:
    auth:
      admin:
        username: root
        password: password
      nova:
        username: nova
        password: password
…
```

Exercise 2 - Solution

Connect to the mariadb pod and run mysql as the nova user. Verify you can access the database.

```
kubectl exec -ti  mariadb-server-0  --namespace openstack bash
mysql@mariadb-server-0:/$ mysql --user=nova --password=password

Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4633
Server version: 10.2.18-MariaDB-1:10.2.18+maria~bionic mariadb.org binary
distribution
…
MariaDB [(none)]> show databases;
…
| nova                     |
| nova_api                 |
| nova_cell0               |
…
```

How OpenStack services are deployed on K8s

kubectl api-resources – Provides a complete list of supported resources in the k8s cluster

kubectl explain – Provides a description of the specified resource in the k8s cluster

kubectl get svc – List the network configuration of a pod in the k8s cluster

kubectl get configmap – List configuration maps in the k8s cluster

kubectl describe configmap – Provides details information of a configuration map in the k8s cluster

kubectl get pv – List persistent volumes in the k8s cluster

kubectl get pvc – List persistent volume claim in the k8s cluster

Exercise 3

1. Using *kubectl get svc*, find the port and IP address glance-registry is configured to use?
2. Using *kubectl get configmap, w*hat is the name of the configmap used by rabbitmq?
3. Using *kubectl describe configmap* what plugins are enabled for rabbitmq in the configmap?
4. Using *kubectl get pvc*, find the capacity of the persistent volume used to store glance images?
5. What is the host path of the glance images store?

Hints: Use the data from question 4.
The OSH instance uses mounted nfs paths to provide volumes.

Exercise 3 - Solution

What port and IP is glance-registry configured to use?

```
kubectl get svc --namespace openstack | grep glance
glance                        ClusterIP   10.104.162.229   <none>        80/TCP,443/TCP                 7d20h
glance-api                    ClusterIP   10.106.157.68    <none>        9292/TCP                       7d20h
glance-reg                    ClusterIP   10.98.159.119    <none>        80/TCP,443/TCP                 7d20h
glance-registry               ClusterIP   10.110.195.72    <none>        9191/TCP                       7d20h
```

Exercise 3 - Solution

What is the name of the configmap used by rabbitmq?

```
kubectl get configmap --namespace openstack | grep rabbit

rabbitmq-rabbitmq-bin                    7       7d16h
rabbitmq-rabbitmq-etc                    2       7d16h
```

Exercise 3 - Solution

Using “describe” what plugins are enabled for rabbitmq in the configmap?

```
kubectl describe configmap rabbitmq-rabbitmq-etc   --namespace openstack

Name:           rabbitmq-rabbitmq-etc
Namespace:      openstack
Labels:         <none>
Annotations:    <none>

Data
====
enabled_plugins:
----

[rabbitmq_management,rabbitmq_peer_discovery_k8s].
…
```

Exercise 3 - Solution

What is the capacity of the persistent volume used to store glance images?

```
kubectl get pvc --namespace openstack

…
glance-images  Bound pvc-ae8d13e4-d377-11e9-b275-000c2982db1f  2Gi RWO general
7d16h
…
```

Exercise 3 - Solution

What is the host path of the glance images store?

```
mount | grep pvc-ae8d13e4-d377-11e9-b275-000c2982db1f

10.111.157.23:/export/pvc-ae8d13e4-d377-11e9-b275-000c2982db1f on /var/lib/kubelet/pods/aebf8fcf-d377-11e9-
b275-000c2982db1f/volumes/kubernetes.io~nfs/pvc-ae8d13e4-d377-11e9-b275-000c2982db1f type nfs4
(rw,relatime,vers=4.1,rsize=1048576,wsize=1048576,namlen=255,hard,proto=tcp,port=0,timeo=600,retrans=2,sec=sys
,clientaddr=192.168.66.129,local_lock=none,addr=10.111.157.23)

sudo ls -al /var/lib/kubelet/pods/aebf8fcf-d377-11e9-b275-000c2982db1f/volumes/kubernetes.io~nfs/pvc-ae8d13e4-
d377-11e9-b275-000c2982db1f
…
-rw-r----- 1 42424 42424 13267968 Sep  9 20:05 bc8babd1-bad0-4da1-8214-b4cc45ae96f7
…
openstack image list
+--------------------------------------+---------------------+--------+
| ID                                   | Name                | Status |
+--------------------------------------+---------------------+--------+
| bc8babd1-bad0-4da1-8214-b4cc45ae96f7 | Cirros 0.3.5 64-bit | active |
```

Basic K8s troubleshooting

Basic K8s troubleshooting

kubectl get replicasets – List the active replicas and desired state in the k8s cluster

kubectl get deployments – List the deployments in the k8s cluster

kubectl descibe deployments – Provdides details of a selected deployment

kubectl get secrets – List secrets used in the k8s cluster

To decode a secret you retrieve the secret with the –o yaml option and then pipe it to \`base 64 –decode\`

https://kubernetes.io/docs/concepts/configuration/secret/#decoding-a-secret

Basic K8s troubleshooting

- Check the replicasets to ensure the correct number of pods are running.
- Check the deployments to verify dependencies, secrets, and volumes
- Using the paths found in question 2, use the “exec” command to do a *ls* to verify the paths configured for the deployment.
- Verify etcd

Exercise 4

1. Using *kubectl get replicasets,* count the number of replicas of nova-api.
2. Using *kubectl get deployments* and *kubectl describe deployments*, what services is neutron-server dependent on and what is their status?
3. Using *kubectl describe deployments,* find the name of the secret used by the glance-api deployment.
4. Using *kubectl get secrets* and the secret from question 3, what is the contents of glance-api.conf?
   Hint: *kubectl get secret <secret> -o yaml*
   echo <Hash> | base64 –decode
5. Using *kubectl describe deployments,* find the volumes and volume types used by keystone-api .

Exercise 4

6. Using *kubectl describe deployments, kubectl get pods* and *kubectl exec -ti* find the mount points of the volumes in the keystone-api pod and use *ls* to list the contents.
7. Using *kubectl describe pods*, find the values for 'list-client-urls', 'advertise-client-urls' and 'trusted-ca-file' for the etcd-nbock-osh pod.
8. Using the values in question 7, and *kubectl exec --ti*, verify the etcd container status.
   exitHint: Connect to the etcd container and run *etcdctl --endpoints <xx> --cert-file <xx> --key-file <xx> --ca-file <xx> cluster-health*
   For --endpoints use the value of advertise-client-urls
   For --ca-file use the value of trusted-ca-file

Exercise 4 - Solution

How many replicas of nova-api are currently running?

```
kubectl get rs --all-namespaces | grep nova

openstack     nova-api-metadata-c84f85f86                1         1         1        7d11h
openstack     nova-api-osapi-76cbc65c8d                  1         1         1        7d11h
openstack     nova-conductor-6d98cd5794                  1         1         1        7d11h
openstack     nova-consoleauth-866476b578                1         1         1        7d11h
openstack     nova-novncproxy-7b6db69b8c                 1         1         1        7d11h
openstack     nova-placement-api-848fdfffb6              1         1         1        7d11h
openstack     nova-scheduler-56f699f9c8                  1         1         1        7d11h
```

Exercise 4 - Solution

What services is neutron-server dependent on and what is their status?

```
kubectl describe deployments neutron-server  --namespace openstack

Name:                   neutron-server
Namespace:              openstack
CreationTimestamp:      Tue, 10 Sep 2019 05:51:29 -0700
…
…
DEPENDENCY_SERVICE:
openstack:mariadb,openstack:rabbitmq,openstack:memcached,openstack:keystone-api
```

Exercise 4 - Solution

What is the name of the secret used by the glance-api deployment?

```
kubectl describe deployments glance-api --namespace openstack | grep
Secret


…
SecretName:  glance-etc
```

Exercise 4 - Solution

Using the secret from question 3, what is the contents of glance-api.conf?

```
kubectl get secrets glance-etc --namespace openstack -o yaml

apiVersion: v1
data:
…
glance-api.conf:
W0RFRkFVTFRdCmJpbmRfcG9ydCA9IDkyOTIKZW5hYmxlX3YxX2FwaSA9IHRydWUKZW5hYmxlX3YyX3JlZ2
lzdHJ5ID0gdHJ1ZQpsb2dfY29uZmlnX2FwcGVuZCA9IC9ldGMvZ2xhbmNlL2xvZ2dpbmcuY29uZgpwdWJs
aWNfZW5kcG9pbnQgPSBodHRwOi8vZ2xhbmNlLm9wZW5zdGFjay5zdmMuY2x1c3Rlci5sb2NhbDo4MC8Kcm
VnaXN0cnlfaG9zdCA9IGdsYW5jZS1yZWdpc3RyeS5vcGVuc3RhY2suc3ZjLmNsdXN
…
…
```

Continued on the next slide

Exercise 4 - Solution

Using the secret from question 3, what is the contents of glance-api.conf?

```
echo
"W0RFRkFVTFRdCmJpbmRfcG9ydCA9IDkyOTIKZW5hYmxlX3YxX2FwaSA9IHRydWUKZW5hYmxlX3YyX3JlZ
…
…
4eV9oZWFkZXJzX3BhcnNpbmcgPSB0cnVlCltwYXN0ZV9kZXBsb3ldCmZsYXZvciA9IGtleXN0b25lCg==
" | base64 -decode

[DEFAULT]
bind_port = 9292
enable_v1_api = true
enable_v2_registry = true
log_config_append = /etc/glance/logging.conf
public_endpoint = http://glance.openstack.svc.cluster.local:80/
registry_host = glance-registry.openstack.svc.cluster.local
…
…
```

Exercise 4 - Solution

What volumes and volume types are used by keystone-api?

```
kubectl describe deployment heat-api --namespace openstack

Name:                       keystone-api
Namespace:                  openstack
…
    Volumes:
   pod-tmp:
    Type:     EmptyDir (a temporary directory that shares a pod's lifetime)
    Medium:
   etckeystone:
    Type:     EmptyDir (a temporary directory that shares a pod's lifetime)
    Medium:
   wsgi-keystone:
    Type:     EmptyDir (a temporary directory that shares a pod's lifetime)
    Medium:
   logs-apache:
    Type:     EmptyDir (a temporary directory that shares a pod's lifetime)
    Medium:
```

Exercise 4 - Solution

Use the “exec” command to run *ls* on the mounts in the keystone-api pod.

```
kubectl describe deployment keystone-api --namespace openstack

Name:                        keystone-api
Namespace:                   openstack
…
Environment:  <none>
    Mounts:
      /etc/apache2/conf.d/security.conf from keystone-etc (ro)
      /etc/apache2/conf.d/wsgi-keystone.conf from keystone-etc (ro)
      /etc/apache2/mods-available/mpm_event.conf from keystone-etc (ro)
      /etc/apache2/ports.conf from keystone-etc (ro)
      /etc/keystone from etckeystone (rw)
      /etc/keystone/credential-keys/ from keystone-credential-keys (rw)
      /etc/keystone/fernet-keys/ from keystone-fernet-keys (rw)
      /etc/keystone/keystone-paste.ini from keystone-etc (ro)
      /etc/keystone/keystone.conf from keystone-etc (ro)
      /etc/keystone/logging.conf from keystone-etc (ro)
...
```

Exercise 4 - Solution

Use the “exec” command to run *ls* on the mounts in the keystone-api pod.

```
kubectl describe deployment keystone-api --namespace openstack

Name:                   keystone-api
Namespace:              openstack
…
Environment:  <none>
    Mounts:
      /etc/apache2/conf.d/security.conf from keystone-etc (ro)
      /etc/apache2/conf.d/wsgi-keystone.conf from keystone-etc (ro)
      /etc/apache2/mods-available/mpm_event.conf from keystone-etc (ro)
      /etc/apache2/ports.conf from keystone-etc (ro)
      /etc/keystone from etckeystone (rw)
      /etc/keystone/credential-keys/ from keystone-credential-keys (rw)
…
kubectl get pods --namespace openstack | grep keystone-api
keystone-api-5fbbf49dc4-wcbjn                  1/1     Running     18         34d

kubectl exec -ti keystone-api-5fbbf49dc4-wcbjn --namespace openstack  ls /etc/keystone /etc/apache2/conf.d
/etc/apache2/conf.d:
security.conf  wsgi-keystone.conf
/etc/keystone:
credential-keys  keystone-paste.ini  logging.conf  sso_callback_template.html
fernet-keys     keystone.conf        policy.json
```

Exercise 4 - Solution

Find the values list-client-urls and the certs used for the etcd-nbock-osh pod.

```
kubectl describe pods etcd-nbock-osh --namespace kube-system
Name:                 etcd-nbock-osh
Namespace:            kube-system
…
Command:
      etcd
      --advertise-client-urls=https://172.17.0.1:2379
      --cert-file=/etc/kubernetes/pki/etcd/server.crt
      --client-cert-auth=true
      --data-dir=/var/lib/etcd
      --initial-advertise-peer-urls=https://172.17.0.1:2380
      --initial-cluster=ubuntu=https://172.17.0.1:2380
      --key-file=/etc/kubernetes/pki/etcd/server.key
      --listen-client-urls=https://127.0.0.1:2379,https://172.17.0.1:2379
      --listen-peer-urls=https://172.17.0.1:2380
      --name=ubuntu
      --peer-cert-file=/etc/kubernetes/pki/etcd/peer.crt
      --peer-client-cert-auth=true
      --peer-key-file=/etc/kubernetes/pki/etcd/peer.key
      --peer-trusted-ca-file=/etc/kubernetes/pki/etcd/ca.crt
      --snapshot-count=10000
      --trusted-ca-file=/etc/kubernetes/pki/etcd/ca.crt
```

Exercise 4 - Solution

Using the values in question 7, verify the etcd container status.

Hint: etcdctl --endpoints <xx> --cert-file <xx> --key-file <xx> --ca-file <xx> cluster-health

For --endpoints use the value of advertise-client-urls
For --ca-file use the value of trusted-ca-file

```
…
etcd
      --advertise-client-urls=https://172.17.0.1:2379
      --cert-file=/etc/kubernetes/pki/etcd/server.crt
      --key-file=/etc/kubernetes/pki/etcd/server.key
      --trusted-ca-file=/etc/kubernetes/pki/etcd/ca.crt
…

kubectl exec -ti etcd-nbock-osh --namespace kube-system sh

etcdctl --endpoints https://172.17.0.1:2379 --cert-file /etc/kubernetes/pki/etcd/server.crt \
  --key-file /etc/kubernetes/pki/etcd/server.key --ca-file /etc/kubernetes/pki/etcd/ca.crt cluster-health

member f483d4b1e906ef01 is healthy: got healthy result from https://172.17.0.1:2379
cluster is healthy
```

We adapt. You succeed.

High Availability, Scaling, and Service Recovery in K8s

High Availability, Scaling, and service recovery in K8s

kubectl scale – Set the number of active replicas and desired state in the k8s cluster

kubectl delete – Delete the specified object in the k8s cluster

docker ps – View the status of running docker containers

docker stop – Stop a running docker container

helm delete – Delete a deployed chart and it resources.

Exercise 5

1. Using *kubectl get pods*, count the number of keystone-api pods running.
2. Using *kubectl scale*, modify keystone-api to have two replicas.
3. Using *kubectl get pods*, count the number of keystone-api pods running.
4. Using *kubectl delete pod*, remove one of the keystone-api pods.
5. Using *kubectl get deployments*, examine the keystone-api deployment.
6. Using *kubectl get pods* and *docker ps,* count the containers associated to each keystone-api pod?
7. Using *docker stop*, halt the container with name beginning *k8s_keystone-api_keystone-api-* for one of the keystone-api pods.
8. Using *kubectl get deployments and kubectl get pods,* check the pod and deployment status.

Exercise 5

9. Using *docker stop*, halt the container with name beginning *k8s_POD_keystone-api-* for one of the keystone-api pods.
10. Using *kubectl get deployments and kubectl get pods,* check the pod and deployment status.
11. Run *openstack image list*
12. Using *kubectl delete pod*, remove the mariadb-server-0 pod
13. Run *openstack image list*
14. Using *kubectl get pods,* check the mariadb-server-0 pods status. Continue to the next step once the mariadb-server-0 pod is running again.
15. Run openstack image list

Exercise 5

15. Delete the glance helm chart using ‘*helm delete --purge glance*’
16. Run openstack image list
17. Redeploy the glance helm chart by running ‘*. ~/redeploy-glance.sh*’

    Note: This will take about 10 minutes to complete.

Exercise 5 - Solution

Count the number of keystone-api pods running.

```
kubectl get pods --namespace openstack
NAME                                            READY    STATUS       RESTARTS    AGE
…
…
keystone-api-574989fff9-4k9z6                   1/1      Running      0           17d
…
```

Exercise 5 - Solution

Scale keystone-api to two replicas.

```
kubectl get deployments  --namespace openstack
NAME                          READY   UP-TO-DATE   AVAILABLE   AGE
…
keystone-api                  1/1     1            1           17d
…

kubectl scale deployment/keystone-api --replicas=2  --namespace openstack
deployment.extensions/keystone-api scaled

kubectl get deployments  --namespace openstack
NAME                          READY   UP-TO-DATE   AVAILABLE   AGE
…
keystone-api                  1/2     2            1           17d
…

kubectl get deployments  --namespace openstack
NAME                          READY   UP-TO-DATE   AVAILABLE   AGE
…
keystone-api                  2/2     2            2           17d
…
```

Exercise 5 - Solution

Count the number of keystone-api pods running.

```
kubectl get pods --namespace openstack
NAME                                          READY    STATUS      RESTARTS    AGE
…
…
keystone-api-574989fff9-4k9z6                 1/1      Running     0           17d
keystone-api-574989fff9-ngdr9                 1/1      Running     0           2m25s
…
…
```

Exercise 5 - Solution

Delete one of the keystone-api pods.

```
kubectl get pods  --namespace openstack  | grep keystone-api
keystone-api-574989fff9-4k9z6                1/1     Running     0          17d
keystone-api-574989fff9-ngdr9                1/1     Running     0          5m1s

kubectl delete pod keystone-api-574989fff9-ngdr9  --namespace openstack
pod "keystone-api-574989fff9-ngdr9" deleted
```

Check the status of the keystone-api deployment.

```
kubectl get deployments  --namespace openstack
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
…
…
keystone-api                   1/2     2            1           17d
…

kubectl get deployments  --namespace openstack
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
…
keystone-api                   2/2     2            2           17d
…
```

Exercise 5 - Solution

How many docker containers are associated for each keystone-api pod?

```
kubectl get pods   --namespace openstack  | grep keystone-api
keystone-api-574989fff9-4k9z6                       1/1     Running     0          17d
keystone-api-574989fff9-kdvxk                       1/1     Running     0          7m47s

sudo docker ps | grep keystone-api

4f4bdec8d25b        a1a23cbccd85                                                     "/tmp/keystone-api.s…"
5 minutes ago       Up 5 minutes                          k8s_keystone-api_keystone-api-574989fff9-
kdvxk_openstack_b1d9ec04-e161-11e9-b275-000c2982db1f_0

3271777bb6cd        k8s.gcr.io/pause:3.1                                             "/pause"
5 minutes ago       Up 5 minutes                          k8s_POD_keystone-api-574989fff9-
kdvxk_openstack_b1d9ec04-e161-11e9-b275-000c2982db1f_0

9b13af1ddd4e        a1a23cbccd85                                                     "/tmp/keystone-api.s…"
2 weeks ago         Up 2 weeks                            k8s_keystone-api_keystone-api-574989fff9-
4k9z6_openstack_c0258212-d376-11e9-b275-000c2982db1f_0

f369b78251be        k8s.gcr.io/pause:3.1                                             "/pause"
2 weeks ago         Up 2 weeks                            k8s_POD_keystone-api-574989fff9-
4k9z6_openstack_c0258212-d376-11e9-b275-000c2982db1f_0
```

Exercise 5 - Solution

Using docker, stop the container with name beginning *k8s_keystone-api_keystone-api-* for one of the keystone-api pods.

```
sudo docker stop k8s_keystone-api_keystone-api-574989fff9-kdvxk_openstack_b1d9ec04-e161-11e9-b275-000c2982db1f_0
```

Exercise 5 - Solution

Check the pod and deployment status.

```
kubectl get deployments  --namespace openstack
NAME                          READY   UP-TO-DATE   AVAILABLE   AGE
…
keystone-api                  1/2     2            1           17d
…
kubectl get pods --namespace openstack
NAME                                          READY   STATUS      RESTARTS   AGE
…
keystone-api-574989fff9-4k9z6                 1/1     Running     0          17d
keystone-api-574989fff9-kdvxk                 0/1     Running     1          14m
…
kubectl get deployments  --namespace openstack
NAME                          READY   UP-TO-DATE   AVAILABLE   AGE
…
keystone-api                  2/2     2            2           17d
…
kubectl get pods  --namespace openstack
NAME                                          READY   STATUS      RESTARTS   AGE
…
keystone-api-574989fff9-4k9z6                 1/1     Running     0          17d
keystone-api-574989fff9-kdvxk                 1/1     Running     1          18m
```

Exercise 5 - Solution

Using docker, stop the container with name beginning *k8s_keystone-api_keystone-api-* for one of the keystone-api pods.

```
sudo docker stop k8s_POD_keystone-api-574989fff9-kdvxk_openstack_b1d9ec04-e161-11e9-b275-000c2982db1f_0
```

Exercise 5 - Solution

Check the pod and deployment status.

```
kubectl get deployments  --namespace openstack
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
…
keystone-api                   1/2     2            1           17d
…
kubectl get pods --namespace openstack
NAME                                           READY   STATUS      RESTARTS   AGE
…
keystone-api-574989fff9-4k9z6                  1/1     Running     0          17d
keystone-api-574989fff9-kdvxk                  0/1     Running     2          25m
…
kubectl get deployments  --namespace openstack
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
…
keystone-api                   2/2     2            2           17d
…
kubectl get pods   --namespace openstack
NAME                                           READY   STATUS      RESTARTS   AGE
…
keystone-api-574989fff9-4k9z6                  1/1     Running     0          17d
keystone-api-574989fff9-kdvxk                  1/1     Running     2          28m
```

Exercise 5 - Solution

Run openstack image list

```
 openstack image list
+--------------------------------------+---------------------+--------+
| ID                                   | Name                | Status |
+--------------------------------------+---------------------+--------+
| bc8babd1-bad0-4da1-8214-b4cc45ae96f7 | Cirros 0.3.5 64-bit | active |
+--------------------------------------+---------------------+--------+
```

Delete the mariadb-server-0 pod.

```
kubectl delete pod  mariadb-server-0  --namespace openstack
pod "mariadb-server-0" deleted
```

Run openstack image list

```
openstack image list (you may see one of these errors)

Bad Gateway (HTTP 502)

An unexpected error prevented the server from fulfilling your request. (HTTP 500) (Request-ID: req-25c0a12d-
4137-4793-8d61-6b5b49cfb339)
```

Exercise 5 - Solution

Check the mariadb-server-0 pods status

```
kubectl get pods --namespace openstack | grep mariadb

mariadb-ingress-668994dc47-9wg5v                0/1     Running     1           2d18h
mariadb-ingress-668994dc47-bkk6g                0/1     Running     1           2d18h
mariadb-ingress-error-pages-56f89d4bb-djb8w     1/1     Running     1           2d18h
mariadb-server-0                                0/1     Running     0           35s

kubectl get pods --namespace openstack | grep mariadb

mariadb-ingress-668994dc47-9wg5v                1/1     Running     1           2d18h
mariadb-ingress-668994dc47-bkk6g                1/1     Running     1           2d18h
mariadb-ingress-error-pages-56f89d4bb-djb8w     1/1     Running     1           2d18h
mariadb-server-0                                1/1     Running     0           48s
```

Run openstack image list

```
 openstack image list
+--------------------------------------+---------------------+--------+
| ID                                   | Name                | Status |
+--------------------------------------+---------------------+--------+
| bc8babd1-bad0-4da1-8214-b4cc45ae96f7 | Cirros 0.3.5 64-bit | active |
+--------------------------------------+---------------------+--------+
```

Exercise 5 - Solution

Delete the glance helm chart

```
helm delete --purge glance
release "glance" deleted
```

Run openstack image list

```
openstack image list

Unable to establish connection to http://glance.openstack.svc.cluster.local:80/v2/images:
HTTPConnectionPool(host='glance.openstack.svc.cluster.local', port=80): Max retries exceeded with url:
/v2/images (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f1e34d03750>: Failed
to establish a new connection: [Errno -2] Name or service not known',))
```

Exercise 5 - Solution

Redeploy the glance helm chart.

```
. ~/redeploy-glance.sh
…
conf:
  glance:
    DEFAULT:
      enable_v1_api: true
      enable_v2_registry: true
manifests:
  deployment_registry: true
  ingress_registry: true
  pdb_registry: true
  service_ingress_registry: true
  service_registry: true
++ helm upgrade --install glance ./glance --namespace=openstack --values=/tmp/glance.yaml
Release "glance" does not exist. Installing it now.
NAME:   glance
LAST DEPLOYED: Mon Oct  7 09:10:06 2019
NAMESPACE: openstack
STATUS: DEPLOYED
…
```

Continued on the next slide

Exercise 5 - Solution

Redeploy the glance helm chart.

```
RESOURCES:
==> v1/ConfigMap
NAME        DATA  AGE
glance-bin  16    3s

==> v1/Deployment
NAME             READY  UP-TO-DATE  AVAILABLE  AGE
glance-api       0/1    1           0          2s
glance-registry  0/1    1           0          2s

==> v1/Job
NAME                  COMPLETIONS  DURATION  AGE
glance-bootstrap      0/1          2s        2s
glance-db-init        0/1          2s        2s
glance-db-sync        0/1          2s        2s
glance-ks-endpoints   0/1          2s        2s
glance-ks-service     0/1          2s        2s
glance-ks-user        0/1          2s        2s
glance-rabbit-init    0/1          2s        2s
glance-storage-init   0/1          2s        2s
…
```

Continued on the next slide

Exercise 5 - Solution

Redeploy the glance helm chart.

```
++ export OS_CLOUD=openstack_helm
++ OS_CLOUD=openstack_helm
++ openstack service list
+----------------------------------+-----------+----------------+
| ID                               | Name      | Type           |
+----------------------------------+-----------+----------------+
| 1405d27e774e4777addadc649cab093b | heat      | orchestration  |
| 27ac6722aed447e2bf57f4e8b6fe1765 | glance    | image          |
| 9f45149ec15741d4a995b96c3777841a | placement | placement      |
| b1cacd500da9431cbb65f95d9973fe8e | keystone  | identity       |
| c2bb8b061cf34989808a8072f59e1ac6 | nova      | compute        |
| cde1cbc36a0847ca910f43310714a41a | heat-cfn  | cloudformation |
| f836b5e52dd94d909cbf3b57a7e2d54f | neutron   | network        |
+----------------------------------+-----------+----------------+
++ sleep 30
++ openstack image list
+--------------------------------------+---------------------+--------+
| ID                                   | Name                | Status |
+--------------------------------------+---------------------+--------+
| bc8babd1-bad0-4da1-8214-b4cc45ae96f7 | Cirros 0.3.5 64-bit | active |
+--------------------------------------+---------------------+--------+
…
```

We adapt. You succeed.

Debugging OpenStack on K8s

Debugging OpenStack on K8s

This section will cover a complete example of debugging an OpenStack service.

- ❑ **Check Pod status**
- ❑ **Check Deployment status**
- ❑ **Check logs**
- ❑ **Enable Debug**
- ❑ **Access the service pod**

Exercise 6

1. Using *kubectl get pods*, check the status of the glance pods.
2. Using *kubectl get deployment,* check the status of the glance deployments.
3. Using *kubectl logs,* examine the log from the glance-api pod.
4. Using *kubectl describe pods*, find the mount and type for the glance-api.conf file used by the glance-api pod.
5. Using *kubectl get secrets,* decode the glance-api.conf and logging.conf secrets and save them to files with the same name.

Exercise 6

6. Edit the glance-api.conf and logging.conf files and make the following changes:

```
glance-api.conf:
    add debug: True under
        glance:
          DEFAULT:
logging.conf:
      change INFO to DEBUG in this section
        logger_glance:
             level: INFO
          handlers:
              - stdout
          qualname: glance
```

Exercise 6

7. Using *base64 –w 0*, encode the logging.conf and glance-api.conf files and save the output to logging.conf.enc and glance-api.conf.enc
8. Using *kubectl edit secrets*, replace the original encrypted text with the values in the enctrypted files from step 7.
9. Using *kubectl delete pods*, delete the glance-api pod.
10. Using *kubectl get pods*, find the name of the new glance-api pod.
11. Continue to the next step once the glance-api pod shows “1/1 Running”
12. Using *kubectl logs,* examine the log from the glance-api pod.
13. Using kubectl exec –ti, connect to the glance-api pod and examine the glance-api.conf and logging.conf located in /etc/glance.

Exercise 6 - Solution

Using *kubectl get pods*, check the status of the glance pods.

```
$ kubectl get pods --all-namespaces -l application=glance
NAMESPACE   NAME                                READY   STATUS      RESTARTS   AGE
openstack   glance-api-7fd996c765-lcvfk         1/1     Running     0          13m
openstack   glance-bootstrap-zkg8f              0/1     Completed   0          13m
openstack   glance-db-init-lsg72                0/1     Completed   0          13m
openstack   glance-db-sync-lpp9x                0/1     Completed   0          13m
openstack   glance-ks-endpoints-j4f75           0/3     Completed   0          13m
openstack   glance-ks-service-w7665             0/1     Completed   1          13m
openstack   glance-ks-user-jjjdr                0/1     Completed   3          13m
openstack   glance-rabbit-init-cnn9p            0/1     Completed   0          13m
openstack   glance-registry-767b4874b6-g9vkg    1/1     Running     0          13m
openstack   glance-storage-init-6ckwh           0/1     Completed   0          13m
```

Exercise 6 - Solution

Using *kubectl get deployment, check the status of the glance deployments*.

```
 kubectl get deployments  --all-namespaces -l application=glance
NAMESPACE   NAME                READY   UP-TO-DATE   AVAILABLE   AGE
openstack   glance-api          1/1     1            1           13m
openstack   glance-registry     1/1     1            1           13m
```

Exercise 6 - Solution

Using *kubectl logs, examine the log from the glance-api*.

```
 kubectl log glance-api-7fd996c765-lcvfk --namespace openstack
log is DEPRECATED and will be removed in a future version. Use logs instead.
+ COMMAND=start
+ start
+ exec glance-api --config-file /etc/glance/glance-api.conf
/var/lib/openstack/local/lib/python2.7/site-packages/paste/deploy/loadwsgi.py:22:
…
…
2019-10-28 20:17:05.541 1 WARNING glance.api.v2.images [-] Could not find schema properties file schema-
image.json. Continuing without custom properties
2019-10-28 20:17:05.541 1 WARNING glance.api.v2.images [-] Could not find schema properties file schema-
image.json. Continuing without custom properties
/var/lib/openstack/local/lib/python2.7/site-packages/paste/deploy/loadwsgi.py:22:
…
2019-10-28 20:17:06.842 1 INFO glance.common.wsgi [-] Starting 1 workers
2019-10-28 20:17:06.842 1 INFO glance.common.wsgi [-] Starting 1 workers
2019-10-28 20:17:06.847 1 INFO glance.common.wsgi [-] Started child 11
…
…
```

Exercise 6 - Solution

Using *kubectl describe pods*, find the mount for the glance-api.conf file used by the glance-api pod.

```
kubectl describe pods glance-api-7fd996c765-lcvfk   --namespace openstack
Name:               glance-api-7fd996c765-lcvfk
…
DEPENDENCY_SERVICE:          openstack:mariadb,openstack:keystone-api,openstack:rabbitmq
…
Mounts:
…
/etc/glance/glance-api.conf from glance-etc (ro)
/etc/glance/logging.conf from glance-etc (ro)
Volumes:
…
glance-etc:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  glance-etc
    Optional:    false
…
```

Exercise 6 - Solution

Using *kubectl get secrets,* decode the glance-api.conf and logging-conf files

```
kubectl describe secret glance-etc   --namespace openstack
Name:         glance-etc
Namespace:    openstack
Labels:       <none>
Annotations:  <none>

Type:  Opaque

Data
====
glance-api.conf:              1789 bytes
glance-registry-paste.ini:    1084 bytes
glance-registry.conf:         957 bytes
policy.json:                  1088 bytes
swift-store.conf:             304 bytes
api_audit_map.conf:           173 bytes
glance-api-paste.ini:         2904 bytes
logging.conf:                 972 bytes
rally_tests.yaml:             564 bytes
```

Continued on the next slide

Exercise 6 - Solution

Using *kubectl get secrets,* decode the glance-api.conf and logging-conf files.

```
kubectl get secrets glance-etc -o 'go-template={{index .data "glance-api.conf"}}'  --namespace openstack |
base64 -d > glance-api.conf

cat glance-api.conf

[DEFAULT]
bind_port = 9292
enable_v1_api = true
enable_v2_registry = true
log_config_append = /etc/glance/logging.conf
public_endpoint = http://glance.openstack.svc.cluster.local:80/
registry_host = glance-registry.openstack.svc.cluster.local
registry_port = 9191
transport_url = rabbit://glance:password@rabbitmq-rabbitmq-
0.rabbitmq.openstack.svc.cluster.local:5672,glance:password@rabbitmq-rabbitmq-
1.rabbitmq.openstack.svc.cluster.local:5672/glance
workers = 1
```

Continued on the next slide

Exercise 6 - Solution

Using *kubectl get secrets,* decode the glance-api.conf and logging.conf files.

```
kubectl get secrets glance-etc -o 'go-template={{index .data "logging.conf"}}'  --namespace openstack | base64
-d > logging.conf

cat logging.conf | more

[formatter_context]
class = oslo_log.formatters.ContextFormatter
datefmt = %Y-%m-%d %H:%M:%S
[formatter_default]
datefmt = %Y-%m-%d %H:%M:%S
format = %(message)s
[formatters]
…
[logger_glance]
handlers = stdout
level = INFO
qualname = glance
[logger_root]
handlers = stdout
level = WARNING
```

Exercise 6 - Solution

Edit the glance-api.conf and logging.conf files and make the following changes:

```
cat glance-api.conf  | more

[DEFAULT]
bind_port = 9292
# Turn debug on
debug = True
…

cat logging.conf
…
…
[logger_glance]
handlers = stdout
level = DEBUG
qualname = glance
…
…
```

Exercise 6 - Solution

Using *base64 –w 0*, encode the logging.conf and glance-api.conf files and save the output to logging.conf.enc and glance-api.conf.enc

```
cat logging.conf | base64 -w 0 | tee logging.conf.enc

W2Zvcm1hdHRlcl9jb250ZXh0XQpjbGFzcyA9IG9zbG9fbG9nLmZvcm1hdHRlcnMuQ29udGV4dEZvcm1hdHRlcgpkYXRlZm10ID0gJVktJW0tJW
QgJUg6JU06JVMKW2Zvcm1hdHRlcl9kZWZhdWx0XQpkYXRlZm10ID0gJVktJW0tJWQgJUg6JU06JVMKZm9ybWF0ID0gJShtZXNzYWdlKXMKW2Zv
cm1hdHRlcnNdCmtleXMgPSBjb250ZXh0
…
…

cat glance-api.conf  | base64 -w 0 | tee glance-api.conf.enc
W0RFRkFVTFRdCmJpbmRfcG9ydCA9IDkyOTIKIyBUdXJuIGRlYnVnIG9uCmRlYnVnID0gVHJ1ZQplbmFibGVfdjFfYXBpID0gdHJ1ZQplbmFibG
VfdjJfcmVnaXN0cnkgPSB0cnVlCmxvZ19jb25maWdfYXBwZW5kID0gL2V0Yy9nbGFuY2UvbG9nZ2luZy5jb25mCnB1YmxpY19lbmRwb2ludCA9
IGh0dHA6Ly9nbGFuY2Uub3BlbnN0YWNr
```

Exercise 6 - Solution

Using *kubectl edit secrets*, replace the original encrypted text with the values in the enctrypted files from step 7.

```
kubectl edit secret glance-etc  --namespace openstack

glance-api.conf:
W0RFRkFVTFRdCmJpbmRfcG9ydCA9IDkyOTIKIyBUdXJuIGRlYnVnIG9uCmRlYnVnID0gVHJ1ZQplbmFibGVfdjFfYXBpID0gdHJ1ZQplbmFibG
VfdjJfcmVnaXN0cnkgPSB0cnVlCmxvZ19jb25maWdfYXBwZW5kID0gL2V0Yy9nbGFuY2UvbG9nZ2luZy5jb25mCnB1YmxpY19lbmRwb2ludCA9
IGh0dHA6Ly9nbGFuY2Uub3BlbnN0YWNrLnN2Yy5jbHVzdGVyLmxvY2FsOjgwLwpyZWdpc3RyeV9ob3N0ID0gZ2xhbmNlLXJlZ2lzdHJ5Lm9wZW
5zdGFjay5zdmMuY2x1c3Rlci5sb2NhbApyZWdpc3RyeV9wb3J0ID0gOTE5MQp0cmFuc3BvcnRfdXJsID0gcmFiYml0Oi8vZ2xhbmNlOnBhc3N3
b3JkQHJhYmJpdG1xLXJhYmJpdG1xLTAucmFiYml0bXEub3BlbnN0YWNrLnN2Yy5jbHVzdGVyLmxvY2FsOjU2NzIsZ2xhbmNlOnBhc3N3b3JkQH
JhYmJpdG1xLXJhYmJpdG1xLTEucmFiYml0bXEub3BlbnN0YWNrLnN2Yy5jbHV

logging.conf:
W2Zvcm1hdHRlcl9jb250ZXh0XQpjbGFzcyA9IG9zbG9fbG9nLmZvcm1hdHRlcnMuQ29udGV4dEZvcm1hdHRlcgpkYXRlZm10ID0gJVktJW0tJW
QgJUg6JU06JVMKW2Zvcm1hdHRlcl9kZWZhdWx0XQpkYXRlZm10ID0gJVktJW0tJWQgJUg6JU06JVMKZm9ybWF0ID0gJShtZXNzYWdlKXMKW2Zv
cm1hdHRlcnNdCmtleXMgPSBjb250ZXh0LGRlZmF1bHQKW2hhbmRsZXJfbnVsbF0KYXJncyA9ICgpCmNsYXNzID0gbG9nZ2luZy5OdWxsSGFuZG
xlcgpmb3JtYXR0ZXIgPSBkZWZhdWx0CltoYW5kbGVyX3N0ZGVycl0KYXJncyA9IChzeXMuc3RkZXJyLCkKY2xhc3MgPSBTdHJlYW1IYW5kbGVy
CmZvcm1hdHRlciA9IGNvbnRleHQKW2hhbmRsZXJfc3Rkb3V0XQphcmdzID0gKHN5cy5zdGRvdXQsKQpjbGFzcyA9IFN0cmVhbUhhbmRsZXIKZm
9ybWF0dGVyID0gY29udGV4dApbaGFuZGxlcnNdCmtleXMgPSBzdGRvdXQsc3RkZX
```

Exercise 6 - Solution

Using *kubectl delete pods*, delete the glance-api pod.

```
kubectl delete  pod  glance-api-fd568bd57-24k67 --namespace openstack | grep glance
pod "glance-api-fd568bd57-24k67" deleted
```

Using *kubectl get pods*, find the name of the new glance-api pod.

```
kubectl get pods --all-namespaces -l application=glance
NAMESPACE   NAME                                 READY   STATUS      RESTARTS   AGE
openstack   glance-api-7c88fc67b-h55c9           1/1     Running     0          11m
openstack   glance-bootstrap-6fdj8               0/1     Completed   0          11m
openstack   glance-db-init-zscfv                 0/1     Completed   0          11m
openstack   glance-db-sync-459zq                 0/1     Completed   0          11m
openstack   glance-ks-endpoints-srprf            0/3     Completed   0          11m
openstack   glance-ks-service-p5tth              0/1     Completed   0          11m
openstack   glance-ks-user-6qkpn                 0/1     Completed   0          11m
openstack   glance-rabbit-init-bkn98             0/1     Completed   0          11m
openstack   glance-registry-598f988cf7-wvr7k     1/1     Running     0          11m
openstack   glance-storage-init-v8v44            0/1     Completed   0          11m
```

Exercise 6 - Solution

Using *kubectl logs, examine the log from the glance-api*.

```
kubectl logs glance-api-7c88fc67b-h55c9 --namespace openstack

+ COMMAND=start
+ start
+ exec glance-api --config-file /etc/glance/glance-api.conf
2019-10-28 20:42:41.449 1 DEBUG glance.common.config [-] Loading glance-api-keystone from /etc/glance/glance-
api-paste.ini load_paste_app /var/lib/openstack/loc
al/lib/python2.7/site-packages/glance/common/config.py:751
…
2019-10-28 20:42:42.396 1 DEBUG glance.common.config [-] debug                          = True log_opt_values
/var/lib/openstack/local/lib/python2.7/site-packag
es/oslo_config/cfg.py:2736
2019-10-28 20:42:42.396 1 DEBUG glance.common.config [-] debug                          = True log_opt_values
/var/lib/openstack/local/lib/python2.7/site-packag
…
```

Exercise 6 - Solution

Using kubectl exec –ti, connect to the glance-api pod and examine the glance-api.conf and logging.conf located in /etc/glance.

```
kubectl exec -ti glance-api-7c88fc67b-h55c9 --namespace openstack sh

grep -i debug /etc/glance/*

/etc/glance/glance-api.conf:debug = true
/etc/glance/logging.conf:level = DEBUG
```

Q&A

We adapt. You succeed.