neueda

futureproof your workforce

CLUSTER AND MACHINE MANAGEMENT

NEUEDA.COM

Objectives

- Backup and Restore Operations
- Cluster Shutdown and Restart
- Machines and MachineSets
- Cluster Autoscaling

neueda
futureproof your workforce

Backup and Restore Operations

- There are two main data pools you may / should backup
    - The Cluster key-value store (etcd), which persists the state of all resource objects.
    - Application resources and Persistent Volumes

ETCD Backup

- This should be done regularly and stored in a secure location.
- You *must* restore you cluster from a backup taken in the same patch release version (4.y.z)
- There is a backup script in the control plane
  - Use `oc debug node/<control plane node>` to access a control plane node
  - Then run the backup script

ETCD Backup Example

```
$ oc debug node/$(oc get nodes | grep master | awk -F' ' '{ print $1 }'| head -n 1)
Starting pod/ip-10-0-132-51eu-west-2computeinternal-debug ...
To use host binaries, run `chroot /host`
Pod IP: 10.0.132.51
If you don't see a command prompt, try pressing enter.
sh-4.4# chroot /host
sh-4.4# /usr/local/bin/cluster-backup.sh /home/core/assets/backup
found latest kube-apiserver: /etc/kubernetes/static-pod-resources/kube-apiserver-pod-11
>--SNIP --<
Snapshot saved at /home/core/assets/backup/snapshot_2022-05-25_170931.db
Deprecated: Use `etcdutl snapshot status` instead.

{"hash":1546201531,"revision":1419436,"totalKey":25275,"totalSize":237514752}
snapshot db and kube resources are successfully saved to /home/core/assets/backup
sh-4.4# exit
exit
sh-4.4# exit
exit

Removing debug pod ...
```

In the command line

In the control node

Control Plane Restore

- There are two reasons you might need to use an etcd backup:
  - An unhealthy etcd member (machine not running, node not ready, pod crashlooping)
  - Disaster recovery
    - Something has happened that requires the cluster to be restored to a previous state.
- This is an involved process, and beyond the scope of this course.

Application Backup

- OpenShift provides the OpenShift API for Data Protection (OADP) for backing up and restoring applications.
- By default this uses Velero
  - Integrates with cloud providers to back up and restore resources

OADP Features

- Backup
  - All resources on your cluster, or filter by type, namespace or label.
  - Save k8s objects and internal images by saving them as an archive file on object storage (like AWS S3).
- Restore
  - Restore all objects or filter restored objects by namespace, PV or label

OADP Features

- Schedule
  - You can schedule backups at specified intervals
- Hooks
  - These are commands to run in a container on a pod, like `fsfreeze` to freeze the file system.
  - Hooks can be configured to run before or after a backup or restore - e.g. `init container` in the application container.

OADP Installation and Use

- This is dependent on the storage provider chosen.
- See the OpenShift documentation for more info

Cluster Shutdown

- You might want to shut a cluster down for maintenance, or to save on resource costs.
- It is important to do this “gracefully”, so the cluster can be restarted at a later date.
- Notes:
  - You can only expect a cluster to restart gracefully up to a year from shutdown - after that the cluster certificates expire.
  - You must have taken and etcd backup before shutdown.

Cluster Shutdown Process

- Open a debug shell in each node
- Run the linux shutdown -h 1 command
  - -h 1 indicates how long, in minutes, the process lasts before the control-plane nodes are shut down.
  - For large-scale clusters set to 10 minutes or longer to make sure all the compute nodes have time to shut down first.

```
for node in $(oc get nodes -o
jsonpath='{.items[*].metadata.name}'); do oc debug node/${node} --
chroot /host shutdown -h 1 ; done
```

Cluster Restart

- The cluster should restart fine by simply turning all the machines back on.
- You can check if the control nodes are ready via:

```
oc get nodes -l node-role.kubernetes.io/master
```

- You can check if the worker nodes are ready via:

```
oc get nodes -l node-role.kubernetes.io/worker
```

- If any nodes are not ready it may be there are pending certificate signing requests to be approved

neueda
futureproof your workforce

Cluster Restart: Operators

- You should also check that none of your cluster operators are degraded:

```
$ oc get clusteroperators
NAME                                        VERSION   AVAILABLE   PROGRESSING   DEGRADED   SINCE
MESSAGE
authentication                              4.10.8    True        False         False      70m
baremetal                                   4.10.8    True        False         False      2d11h
cloud-controller-manager                    4.10.8    True        False         False      2d11h
cloud-credential                            4.10.8    True        False         False      2d11h
cluster-autoscaler                          4.10.8    True        False         False      2d11h
config-operator                             4.10.8    True        False         False      2d11h
console                                     4.10.8    True        False         False      33h
csi-snapshot-controller                     4.10.8    True        False         False      2d11h
dns                                         4.10.8    True        False         False      2d11h
```

Machine Management

- Machine Management can be used to work flexibly with underlying infrastructure from a variety of cloud providers.
  - Control the cluster
  - Perform cluster auto-scaling based on specific workload policies

Machine API Operator Resources

Machine	Fundamental unit that describes the host for a Node.
MachineSet	Groups of machines (only worker nodes)
Machine Autoscaler	Automatically scales machines in a cloud.
Cluster Autoscaler	Cluster-wide resource limits workload priorities
Machine Health Checks	Checks when a machine is unhealthy and replaces it

neueda
futureproof your workforce

Manual MachineSet Scaling Demo

- This demo just follows the procedure of [Manually scaling a machine set | Machine management | OpenShift Container Platform 4.10] on the cluster.
  - Note that cordoning and draining a node is only possible if there are no pods with persistent data there.

Deleting a Machine

- A machine can be deleted just using

```
$ oc delete machine <machine> -n openshift-machine-api
```

- The machine controller tries to drain the node first
- If the machine is in a MachineSet a new machine is immediately created to match the declarative requirement.

Cluster Autoscaling

- Only works in clusters where the machine API is working.
- Provides infrastructure management no reliant on a specific cloud provider.
- Has cluster scope (not in a namespace or project)
- Cluster size is increased whenever there are pods that fail to schedule, up to your specified limits.
- Cluster size is decreased if a node has utilization less than a *node utilization threshold* for the cluster.

HPA and Cluster Autoscaling

- The Horizontal Pod Autoscaler (HPA) creates new replicas based on service load, regardless of cluster resources.
- The cluster autoscaler adds resources so the HPA created pods can run.
- If the load decreases, the HPA stops some replicas, which drops node utilization so the cluster autoscaler also scales back.

Deploying Cluster Autoscaling

1. Create the resource definition YAML file
2. Deploy the ClusterAutoscaler resource:

```
$ oc create -f cluster-autoscaler-example.yaml
```

3. Configure at least on Machine autoscaler
    1. Create the resource definition YAML file
    2. Deploy the Machine autoscaler resource:

```
$ oc create -f machine-autoscaler-example.yaml
```

neueda

futureproof your workforce

Summary

- Backup and Restore Operations
- Cluster Shutdown and Restart
- Machines and MachineSets
- Cluster Autoscaling

neueda
futureproof your workforce

Questions and Comments?

neveda futureproof your workforce