Kubernetes

This page is a placeholder for information and processes surrounding Kubernetes (K8).

A quick-start guide can be found here: https://github.com/kubernetes/kubernetes/blob/release-1.1/docs/getting-started-guides/docker.md/

Installing kubectl

The following set of commands can be used to install kubectl on your machine. You may need to change the version number below:

```
mkdir /home/core/kubectl
cd /home/core/kubectl
curl -L "https://storage.googleapis.com/kubernetes-release/release/v1.1.3/bin/linux/amd64/kubectl" > .
export PATH=$PATH:`pwd`
chmod +x kubectl
```

Single-Node (Developer)

- A single CoreOS node running all components of Kubernetes and NDS Labs, suitable for developing and testing new features for the platform.

Setup

1. Create a single CoreOS machine on Nebula with an assigned SSH key.
2. Assign a floating IP to this machine.
3. Add the new machine to the "remote SSH", "remote HTTP", and "Kubernetes NodePort" security groups.
4. SSH into the new machine using the assigned key.
5. Execute the following command to obtain the NDS Labs startup scripts:

   ```
   git clone https://github.com/craig-willis/ndslabs-startup.git
   cd ndslabs-startup/
   ```

6. Execute kube-up.sh to start Kubernetes on your single-node setup:

   ```
   ./kube-up.sh
   ```

7. Modify **ndslabs/apiserver.yaml** and **ndslabs/gui.yaml** so that it reflects the addresses in your cluster.
8. Once Kubernetes is up and running, you can run ./ndslabs-up.sh to bring up the GUI and API server:

   ```
   ./ndslabs-up.sh
   ```

Multi-Node (Production)

Setup

Run the Ansible deployment playbook(s) to provision all necessary machines and volume(s).

Troubleshooting

Node status is NotReady

Sometimes, worker nodes will enter a "NotReady" state, which will prevent Kubernetes from scheduling pods on the node.

```
core@lambert-master1 ~ $ kubectl get nodes
NAME              STATUS     AGE
lambert-gfs1      Ready      2d
lambert-loadbal   Ready      2d
lambert-node1     NotReady   2d
```

To discover the reason behind the error, you can use **kubectl describe**:

```
core@lambert-master1 ~ $ kubectl describe node lambert-node1
Name:                   lambert-node1
Labels:                 kubernetes.io/hostname=lambert-node1,ndslabs-node-role=compute
CreationTimestamp:      Fri, 27 May 2016 20:31:53 +0000
Phase:
Conditions:
  Type          Status          LastHeartbeatTime                       LastTransitionTime
Reason                  Message
  ----          ------          -----------------                       ------------------
------                  -------
  OutOfDisk     Unknown         Sat, 28 May 2016 12:56:01 +0000         Sat, 28 May 2016 12:56:41 +0000
NodeStatusUnknown       Kubelet stopped posting node status.
  Ready         Unknown         Sat, 28 May 2016 12:56:01 +0000         Sat, 28 May 2016 12:56:41 +0000
NodeStatusUnknown       Kubelet stopped posting node status.
Addresses:      192.168.100.253,192.168.100.253
Capacity:
 cpu:           2
 memory:        4051772Ki
 pods:          110
System Info:
 Machine ID:                    ef45bc8b101a4ac987d295ac7886e358
 System UUID:                   EF45BC8B-101A-4AC9-87D2-95AC7886E358
 Boot ID:                       6a95eea9-99c2-42f8-abe1-0f57545eee7a
 Kernel Version:                4.3.6-coreos
 OS Image:                      CoreOS 899.11.0
 Container Runtime Version:     docker://1.9.1
 Kubelet Version:               v1.2.4
 Kube-Proxy Version:            v1.2.4
ExternalID:                     lambert-node1
Non-terminated Pods:            (1 in total)
  Namespace                     Name                            CPU Requests    CPU Limits      Memory Requests
Memory Limits
  ---------                     ----                            ------------    ----------      ---------------
-------------
  default                       glusterfs-client-8ets6          0 (0%)          0 (0%)          0 (0%)
0 (0%)
Allocated resources:
  (Total limits may be over 100%, i.e., overcommitted. More info: http://releases.k8s.io/HEAD/docs/user-guide
/compute-resources.md)
  CPU Requests  CPU Limits      Memory Requests Memory Limits
  ------------  ----------      --------------- -------------
  0 (0%)        0 (0%)          0 (0%)          0 (0%)
Events:
  FirstSeen     LastSeen        Count   From                    SubobjectPath   Type            Reason
Message
  ---------     --------        -----   ----                    -------------   --------        ------
-------
  2d            7s              19526   {controllermanager }                    Normal          DeletingAllPods
Node lambert-node1 event: Deleting all Pods from Node lambert-node1.
```

We can see from the verbose output above (under "Conditions") that the node seems to have run out of disk space.

In this case, I cleared out several large Docker images with **docker rmi** to clear up several GB of disk space.

After freeing up some disk space, we can SSH into the worker node and restart the kubelet there to being it back into the cluster:

```
core@lambert-deploy-tools ~ $ sudo ssh -i ~/private/lambert-test.pem core@192.168.100.253
core@lambert-node1 ~ $ sudo systemctl restart kubelet
core@lambert-node1 ~ $ exit
```

Back on the master, you should see shortly that the node has righted itself:

```
core@lambert-master1 ~ $ kubectl describe node lambert-node1
Name:                   lambert-node1
Labels:                 kubernetes.io/hostname=lambert-node1,ndslabs-node-role=compute
CreationTimestamp:      Fri, 27 May 2016 20:31:53 +0000
Phase:
Conditions:
  Type          Status  LastHeartbeatTime                       LastTransitionTime
Reason                          Message
  ----          ------  -----------------                       ------------------
------                          -------
  OutOfDisk     False   Mon, 30 May 2016 19:20:45 +0000         Mon, 30 May 2016 19:20:35 +0000
KubeletHasSufficientDisk        kubelet has sufficient disk space available
  Ready         True    Mon, 30 May 2016 19:20:45 +0000         Mon, 30 May 2016 19:20:45 +0000
KubeletReady                    kubelet is posting ready status
Addresses:      192.168.100.253,192.168.100.253
Capacity:
 cpu:           2
 memory:        4051772Ki
 pods:          110
System Info:
 Machine ID:                    ef45bc8b101a4ac987d295ac7886e358
 System UUID:                   EF45BC8B-101A-4AC9-87D2-95AC7886E358
 Boot ID:                       6a95eea9-99c2-42f8-abe1-0f57545eee7a
 Kernel Version:                4.3.6-coreos
 OS Image:                      CoreOS 899.11.0
 Container Runtime Version:     docker://1.9.1
 Kubelet Version:               v1.2.4
 Kube-Proxy Version:            v1.2.4
ExternalID:                     lambert-node1
Non-terminated Pods:            (2 in total)
  Namespace                     Name                                            CPU Requests    CPU Limits
Memory Requests Memory Limits
  ---------                     ----                                            ------------    ----------
--------------- -------------
  default                       glusterfs-client-8ets6                          0 (0%)          0 (0%)
0 (0%)          0 (0%)
  kube-system                   fluentd-elasticsearch-lambert-node1             100m (5%)       0 (0%)
200Mi (5%)      200Mi (5%)
Allocated resources:
  (Total limits may be over 100%, i.e., overcommitted. More info: http://releases.k8s.io/HEAD/docs/user-guide
/compute-resources.md)
  CPU Requests  CPU Limits      Memory Requests Memory Limits
  ------------  ----------      --------------- -------------
  100m (5%)     0 (0%)          200Mi (5%)      200Mi (5%)
Events:
  FirstSeen     LastSeen        Count   From                    SubobjectPath   Type
Reason                  Message
  ---------     --------        -----   ----                    -------------   --------
------                  -------
  15s           15s             1       {kubelet lambert-node1}                 Normal
Starting                Starting kubelet.
  15s           14s             2       {kubelet lambert-node1}                 Normal
NodeHasSufficientDisk   Node lambert-node1 status is now: NodeHasSufficientDisk
  14s           14s             1       {kubelet lambert-node1}                 Normal
NodeNotReady            Node lambert-node1 status is now: NodeNotReady
  2d            13s             19579   {controllermanager }                    Normal
DeletingAllPods         Node lambert-node1 event: Deleting all Pods from Node lambert-node1.
  4s            4s              1       {kubelet lambert-node1}                 Normal
NodeReady               Node lambert-node1 status is now: NodeReady


core@lambert-master1 ~ $ kubectl get nodes
NAME              STATUS    AGE
lambert-gfs1      Ready     2d
lambert-loadbal   Ready     2d
lambert-node1     Ready     2d
```