

# Volume Support with Heketi

In-progress notes from exploratory work on NDS-784. So far I haven't been able to get Heketi to deploy on OpenStack.

## Kargo+Heketi

Use the default [Kargo-cli container](#), but it lacks the shade libraries required for OpenStack;

```
$ docker run -v `pwd`/kargoconf:/etc/kargo quay.io/smana/k8s-kargocli bash

$ easy_install -U pip
$ pip install "ansible==2.2.1.0" --force-reinstall
$ pip install shade
$ apt-get update -y
$ apt-get install -y vim
```

kargo.yml

```
kargo_git_repo: "https://github.com/kubespray/kargo.git"
loglevel: "info"

# OpenStack options
# ---
os_auth_url: "http://nebula.ncsa.illinois.edu:5000/v2.0"
os_username: "you"
os_password: "yourpassword"
os_project_name: "NDSLabsDev"
masters_flavor: "m1.medium"
nodes_flavor: "m1.medium"
etcds_flavor: "m1.medium"
image: "ubuntu-yakkety-cloud"
network: "NDSLabsDev"
sshkey: "yourkey"
```

Deploy a 3-node cluster:

```
$ kargo openstack --nodes 3
```

Heketi requires a few dependencies that aren't in the default ubuntu image, so ssh into each node and install manually (we can add this to kargo or another playbook later)

```
sudo apt-get install -y python-minimal docker.io glusterfs-client
sudo systemctl start docker
sudo modprobe dm_thin_pool
```

Heketi requires that each node have a volume. Via Horizon or openstack CLI, create one volume per instance and attach but do not format. (We can automate this later)

Deploy the kargo cluster:

```
$ kargo deploy -k your.pem -u ubuntu
```

ssh into the master node with kubectl (the first created instance) and install heketi

```
$ ssh -i your.pem ubuntu@192.168.x.x
$ wget https://github.com/heketi/heketi/releases/download/v4.0.0/heketi-client-v4.0.0.linux.amd64.tar.gz
$ tar xvfz heketi-client-v4.0.0.linux.amd64.tar.gz
$ export PATH=~/.heketi-client/bin/:$PATH
$ git clone https://github.com/gluster/gluster-kubernetes.git
$ cd gluster-kubernetes/deploy
```

Edit the topology.json file, there the node name should be the same as output from `kubectl get nodes`, the storage host should be the internal IP (192.168.x.x.).

Add an exit statement before the first call to `heketi-cli` in `gk-deploy` (this seems to be broken):

Run `gk-deploy`

```
$ gk-deploy -g
$ kubectl get ep
$ heketi-cli -s http://<ip>:8080 topology load --json=topology.json
```

At this point, things fail. The first node/volume is added, but then hangs when adding the second. Looking at the `heketi` deployment logs, I'm seeing "gluster peer probe failures". The symptoms are reported in <https://github.com/heketi/heketi/issues/625>.

## Vagrant+Heketi

Apparently, no one uses Kargo to deploy this arrangement. Most of the issues in `heketi` git refer to a vagrant-based deploy, which I'll try next.

## Other notes:

- kargo creates ports and security groups that need to be deleted manually via `neutron port-list/port-delete` and `openstack secgroup-list/secgroup-delete`