Provisioning node setup

Here's how to create a provisioning node to run deploy-tools (this came up with NDS-395)

- In Nebula, create instance, select CoreOSStable1122
- Disable updates
    - sudo systemctl stop update-engine
    - sudo systemctl stop locksmithd
- Fix the Docker BIP and MTU
- sudo su
    - mkdir -p /etc/systemd/system/docker.service.d
    - vi /etc/systemd/system/docker.service.d/10-docker0.conf

[Service]
Environment="DOCKER_OPT_BIP=--bip=172.30.0.1/16"
Environment="DOCKER_OPT_MTU=--mtu=1454"

- systemctl daemon-reload
- systemctl restart docker
- cd /var/lib/kubelet/pods && rm -rf *

Change the default docker0 bridge MTU

Multi-node deployment via Ansible currently sets this for you, but for single-node installations this is a manual step:

```
# Elevate to a superuser shell
sudo su

# Create a systemd drop-in to set the Docker MTU
mkdir -p /etc/systemd/system/docker.service.d/
bash -c "echo '[Service]
Environment=\"DOCKER_OPT_MTU=--mtu=1454\"' > /etc/systemd/system/docker.service.d/10-docker0.conf"

# Then reload / restart your Docker daemon
systemctl stop docker
systemctl daemon-reload
systemctl start docker
```

A reasonable value for docker0 MTU is the eth0 MTU minus 50:

```
ifconfig | grep eth0 | grep -i mtu
```

Disable automatic CoreOS update

To prevent CoreOS from updating and potentially changing the version of Docker out from underneath you unexpectedly, run the following commands on production instances:

```
sudo systemctl stop locksmithd update-engine
sudo systemctl disable locksmithd update-engine
sudo systemctl mask locksmithd update-engine
```

WARNING: You should still perform manual updates occasionally on your development machines to ensure that future updates will not result in catastrophic failure.

If you delete your kubelet, you MUST clear out old pods to remove stale authentication tokens

This is only necessary on single-node installations:

```
# NOTE: You would only need to do this if you explicitly delete your kubelet, like this:
docker rm -f kubelet

# Elevate to a superuser shell
sudo su

# Delete all leftover pods
cd /var/lib/kubelet/pods && rm -rf *
exit

# This will create a new kubelet, with a new auth token
./kube.sh
```

Adding an image to OpenStack

If the image doesn't exist in the current OpenStack project:

- docker run -it ndslabs/deploy-tools bash
- curl https://stable.release.core-os.net/amd64-usr/current/coreos_production_openstack_image.img.bz2 -o coreos_production_openstack_image.img.bz2
- bunzip coreos_production_openstack_image.img.bz2
- openstack image create --disk-format qcow2 --file ./coreos_production_openstack_image.img CoreOSStable1122