Upgrading Kubernetes

Overview

This page captures notes and requirements for NDS-413. We're talking about upgrading Kubernetes from 1.2 to 1.4 for beta release. This page captures the risks/benefits of upgrading each component. What versoin should we be running and why? What's changed?

From the whiteboard:

- CoreOS
- Docker
    - MTU
- etcd2
- Kubernetes
    - network (calico)
    - storage drivers (dynamic provisioning)
    - Federation/ubernetes
    - upgrade paths
    - performance/bugs
    - log rotation
- development environment (how to spin up a dev cluster)
- ansible
- Qualys vulnerabilities (Curl).

CoreOS

- We are currently running stable channel 1122(docker=1.10.3, etcd2= 2.3.2).
- The latest stable release is 1185.5 (docker=1.11.2, etcd2=2.3.7)
    - https://coreos.com/releases/

Conclusion:

- There is no reason to change to beta channel
- Docker 1.11.2 should not be a problem, but we do need to test it
- 1185.5 does not fix the curl vulnerabilities

Docker

So far, our main concern with Docker has been MTU support. Looking through the docker github issues, the MTU behavior in 1.10 is by design, fixing a problem in 1.9, and is unchanged in 1.11, 1.12

- https://github.com/docker/docker/pull/18108 (closed) "don't try to use default route MTU as container MTU"
    - "Trying to use the default route's MTU as the container (bridge) MTU is a bad idea:..."
- https://github.com/docker/docker/issues/22028 (closed) "docker 1.10, 1.11 do not infer MTU from eth0; docker 1.9 does"
    - "this was a deliberate change, and specifying the MTU is the solution for this"
- https://github.com/docker/docker/issues/22297 (closed) "containers in docker 1.11 does not get same MTU as host"
    - "In another PR (#18108) it was decided to drop the behavior where the mtu for the containers running on the default `bridge` network would be inherited from the host (there were valid concerns with that approach and @phemmer IIRC lead the discussion)"

Conclusion

- Not much we care about in the engine, but there is more stability in 1.11.
- Docker 1.11.2 should not be a problem, but we need to test it.
- Need to resolve DinD API version problem if we upgrade

etcd2

- Our current etcd2 version is 2.3.2. Upgrading to CoreOS 1185 will change to etcd2 2.3.7. Latest release is 3.1
- Change from 2.3.2 - 2.3.7 is only minor bugfixes

Ansible

- contrib/ansible has been updated to support 1.4.5
    - PR https://github.com/kubernetes/contrib/pull/2049
    - Issue https://github.com/kubernetes/contrib/issues/1953
        - Upgrade Ansible installation to Kubernetes 1.4

Conclusion

- We should be able to test the new ansible with Kubernetes 1.4.5 without changing much of our beta deployment process, which is a good thing.
- David:

- To get SDSC up we need newer ansible, we need to move our ansible forward, if we're using contrib we might need to forward that as well.
- Pulled new kubernetes contrib ansible and it seems forward compatible
- Other options: (most of what we need is to provision openstack)
  - kubeadm
  - ket: https://github.com/apprenda/kismatic

Kubernetes

1.3 http://blog.kubernetes.io/2016/07/kubernetes-1.3-bridging-cloud-native-and-enterprise-workloads.html

- Easier autoscaling
- Federated, cross-cluster services
- Petsets (stateful applications)
- Minikube for local development
- Additional container support (rkt, etc)

1.4 http://blog.kubernetes.io/2016/09/kubernetes-1.4-making-it-easy-to-run-on-kuberentes-anywhere.html

- kubeadm for bootstrapping
- Standard installations for ubuntu/redhat
- Batch jobs
- Dynamic persistent volume claims (beta)
- Federation in beta (secrets, events, namespaces, ingress - alpha)
- Improved container security
- 1.4.5 appears to support Docker 1.11.2

1.5 http://blog.kubernetes.io/2016/12/kubernetes-1.5-supporting-production-workloads.html

- 12/13/2016
- HA masters
- kubefed for federation
- Windows containers

Conclusion:

- Most of the new features in 1.3 - 1.5 are things we will not immediately take advantage of, but are on the roadmap or may affect open issues. This includes network security, new storage architecture options, federation support, etc.
- Upgrade paths unfortunately exist only for Ubuntu and RedHat, not CoreOs.
- 1.4.x has log rotation, which is a minor need for us.

Dev environment

- Kubernetes now supports two options for spinning up a cluster: minikube and kubeadm
- Kubeadm is only supported on RedHat/Ubuntu
- Minikube has been problematic in the past on our Nebula VMs

Conclusion

- We should easily be able to test 1.4 using kubeadm on a non-CoreOs system

- ☑ **NDS-708** - Update ndslabs-startup to test running Kubernetes 1.5 on a single-node instance `RESOLVED`