KubeCon | CloudNativeCon

Europe 2018

kubeadm deep dive

Luke Marsden - Dotmesh
Tim St. Clair - Heptio
Alexander Kanevskiy - Intel

Agenda

1. Luke: kubeadm intro & history
2. Tim: Upgrades, self-hosting & HA
3. Alexander: Using kubeadm in enterprise environment

KubeCon | CloudNativeCon

Europe 2018

kubeadm intro & history

Luke Marsden - dotmesh

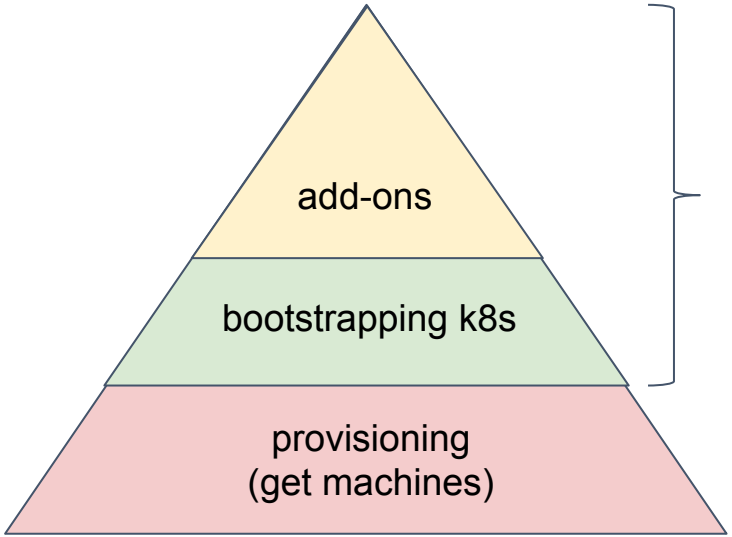
Formation of SIG-cluster-lifecycle

It's September 2016…

- "Kubernetes is too hard to install!"
  - Kubernetes the hard way!
- Community felt pressure to deliver a standard, *simple* way to install Kubernetes
- Many projects to install Kubernetes forming… clearly there was a need
- Spun out SIG-cluster-lifecycle from SIG-cluster-ops
  - Goal to build tools and a toolkit to make Kube easier to install

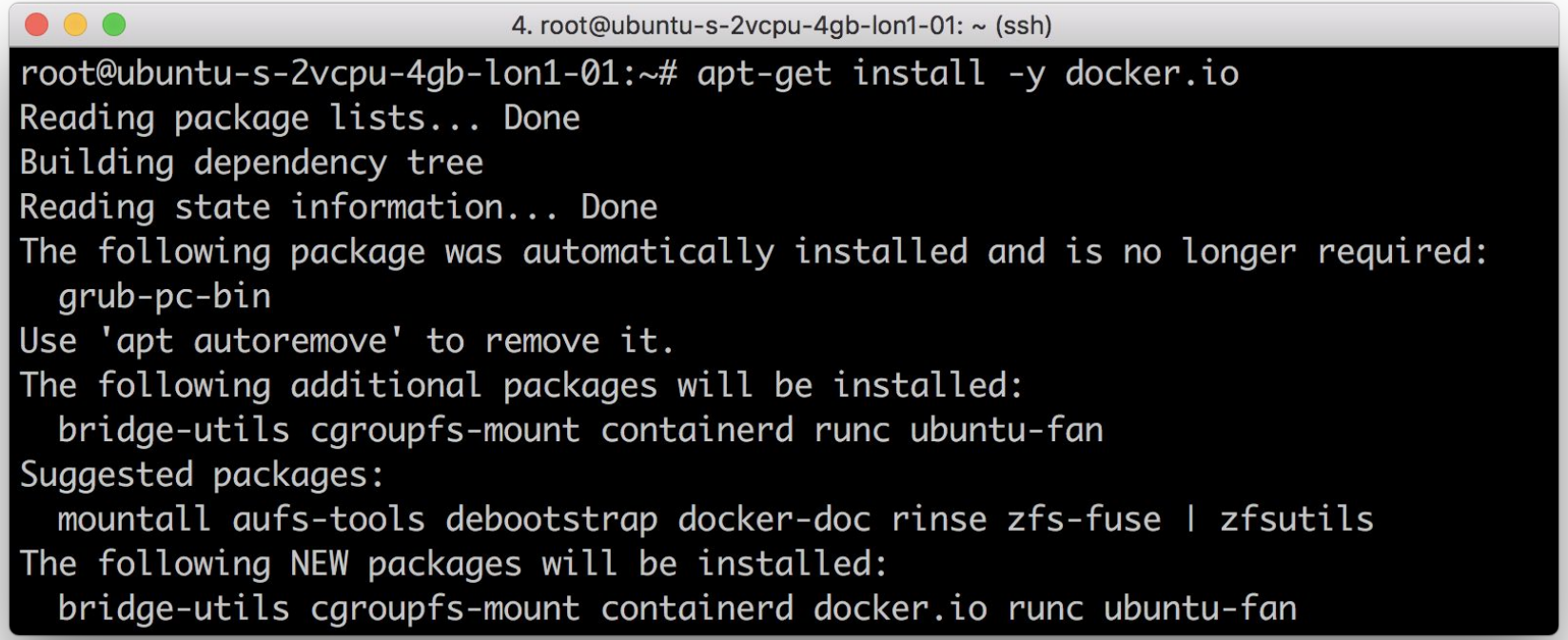
kubeadm goals

- Make Kubernetes insanely easy to install
- 3 phases to installation
- Variety in how people provision machines
- Decided to focus on bootstrapping, not provisioning!
- Two commands, plus add-ons

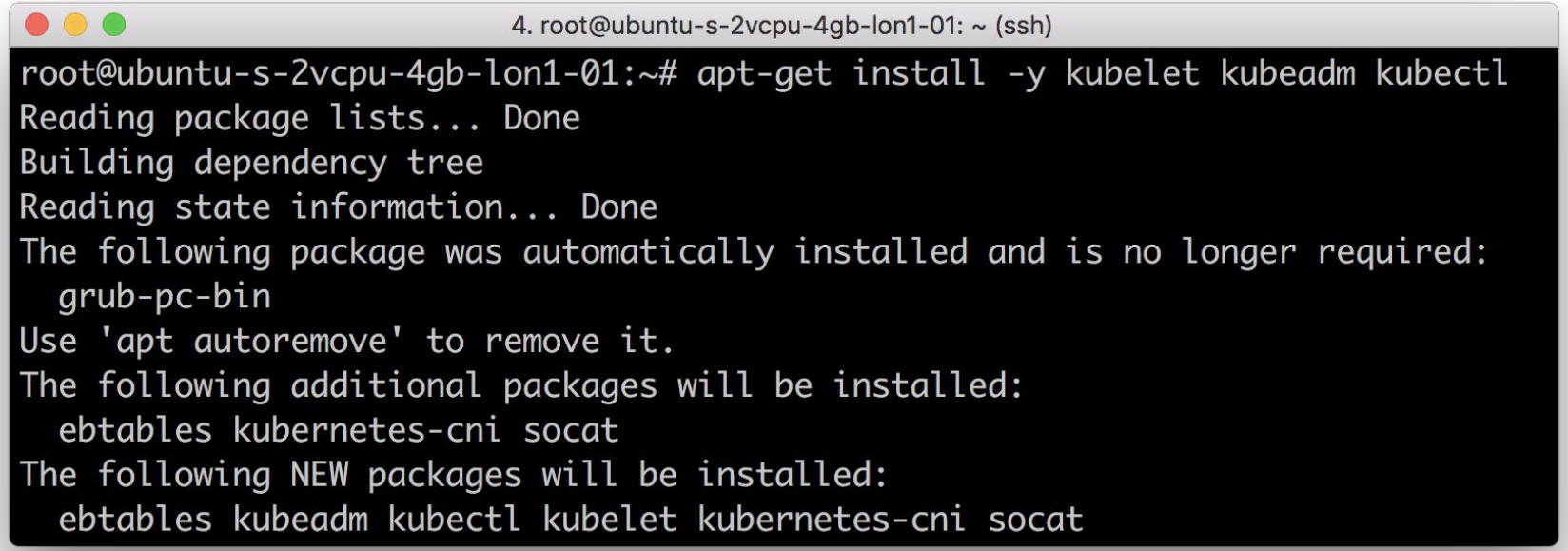


kubeadm example

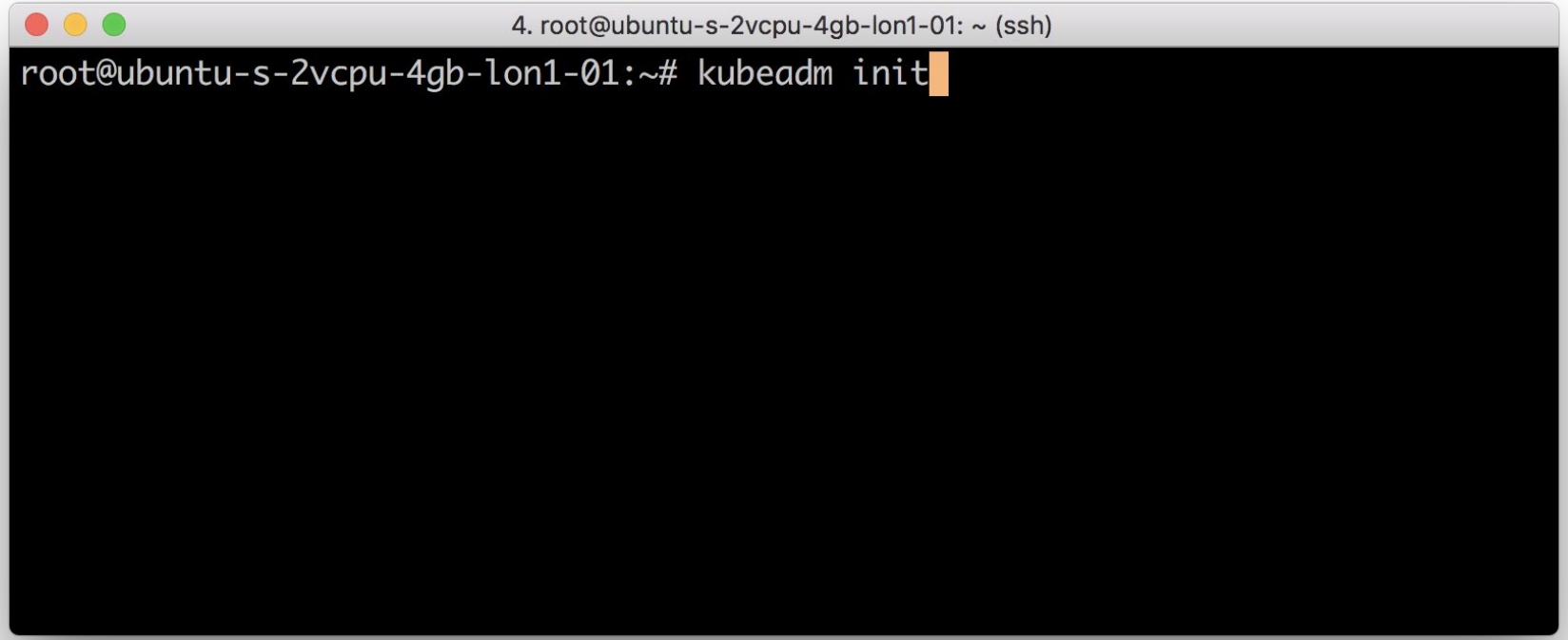
- First release in Kubernetes 1.4
- Step 1: get some computers running Linux
- Then...



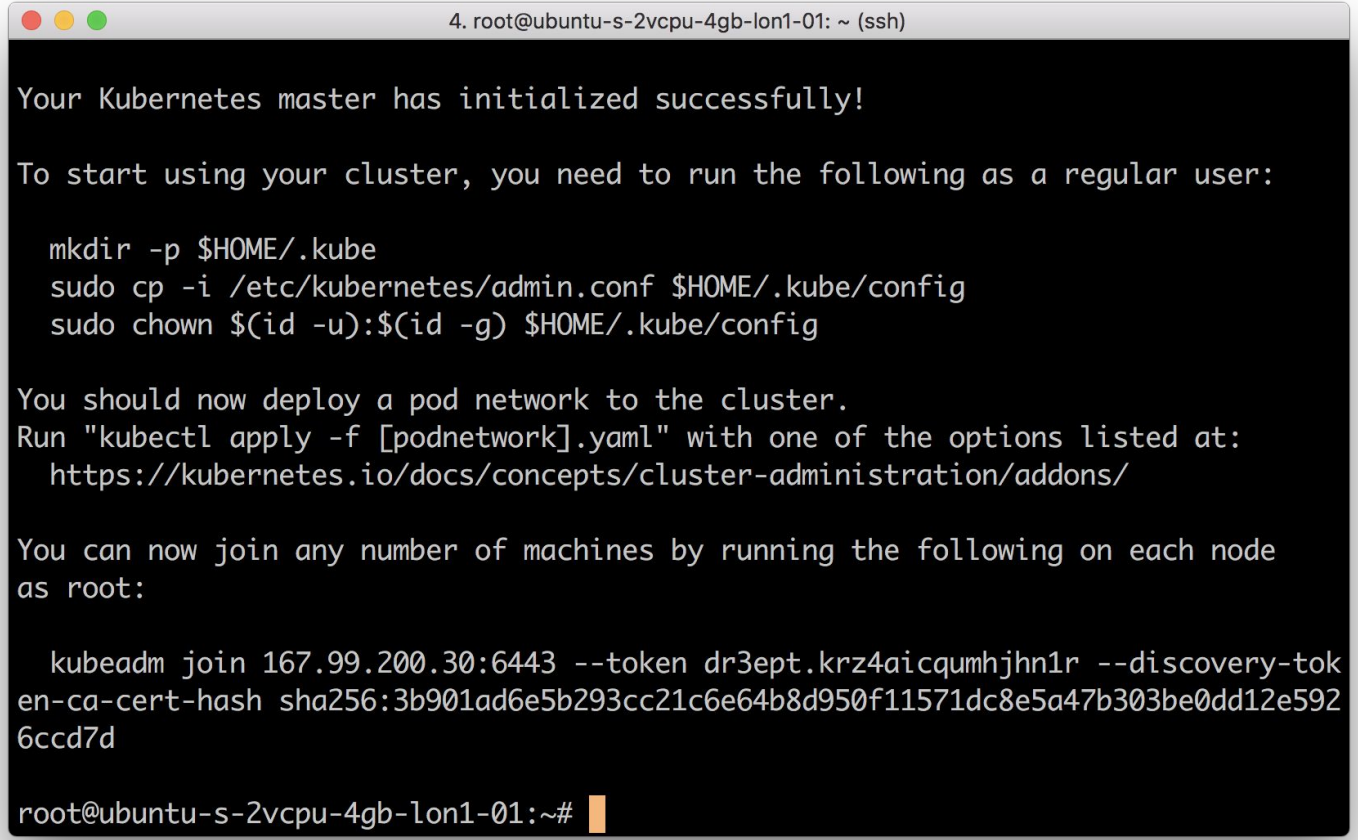
install docker

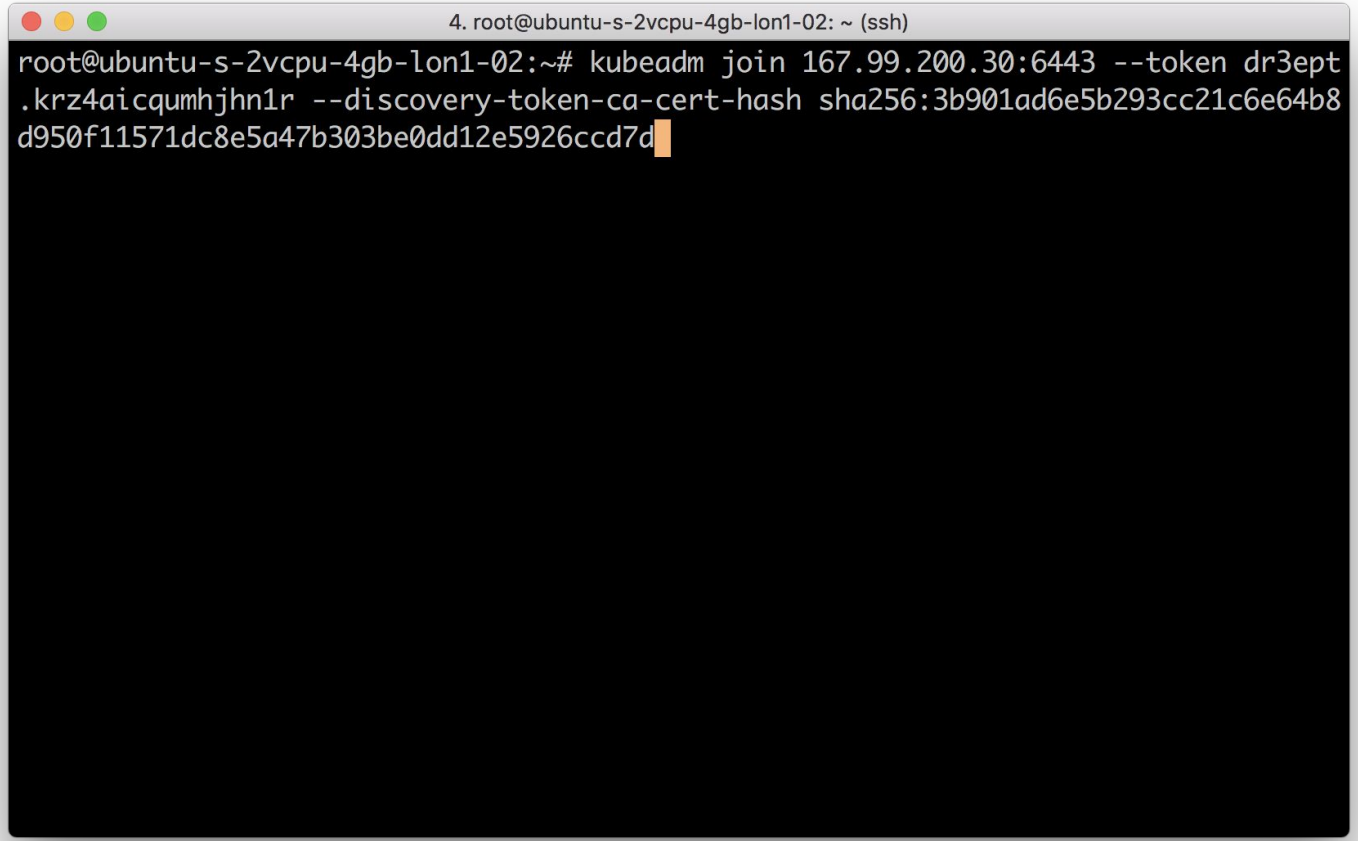


install kubelet, kubeadm, kubectl

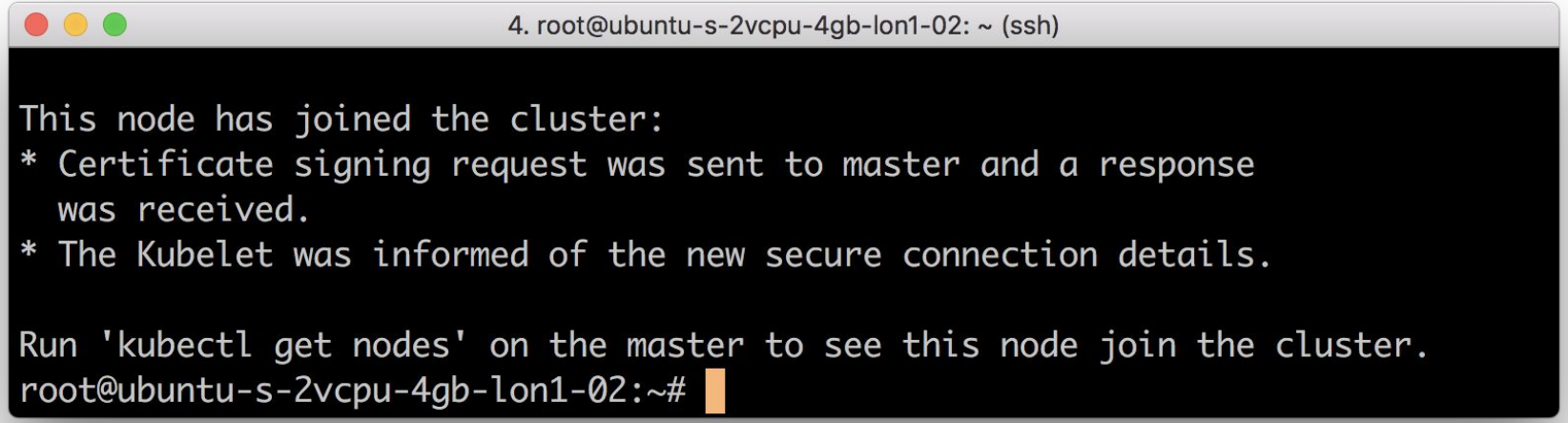


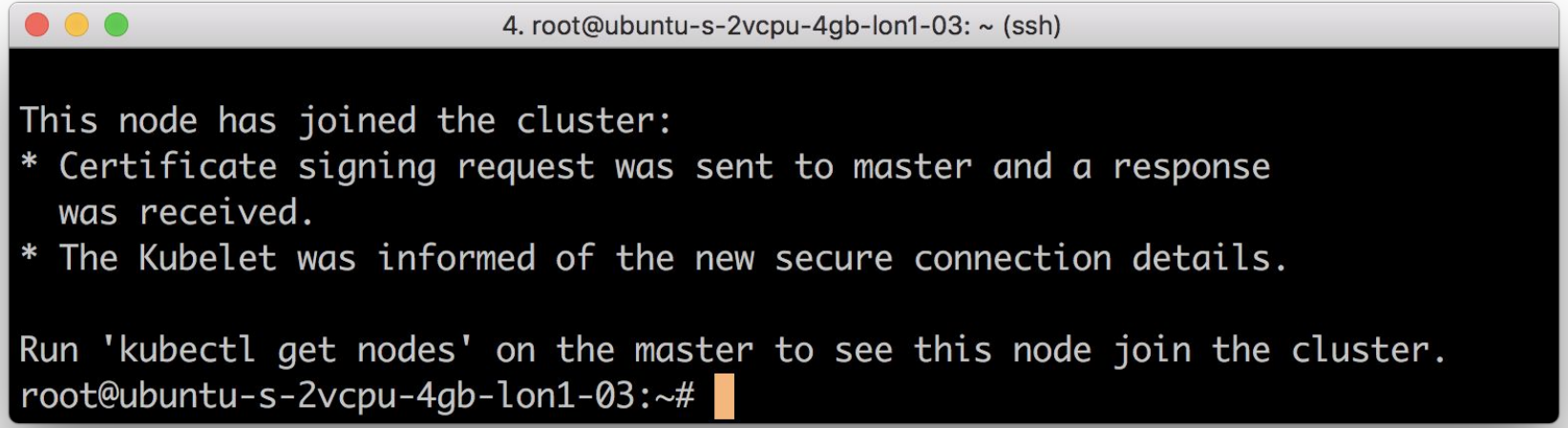
kubeadm init

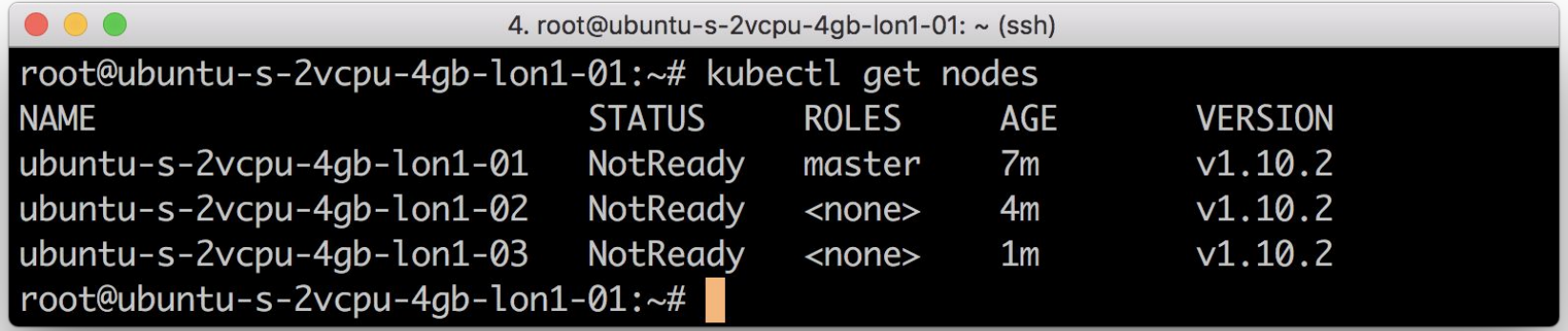

4. root@ubuntu-s-2vcpu-4gb-lon1-01: ~ (ssh)
Your Kubernetes master has initialized successfully!
To start using your cluster, you need to run the following as a regular user:
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/
You can now join any number of machines by running the following on each node
as root:
kubeadm join 167.99.200.30:6443 --token dr3ept.krz4aicqumhjhn1r --discovery-tok
en-ca-cert-hash sha256:3b901ad6e5b293cc21c6e64b8d950f11571dc8e5a47b303be0dd12e592
6ccd7d
root@ubuntu-s-2vcpu-4gb-lon1-01:~#



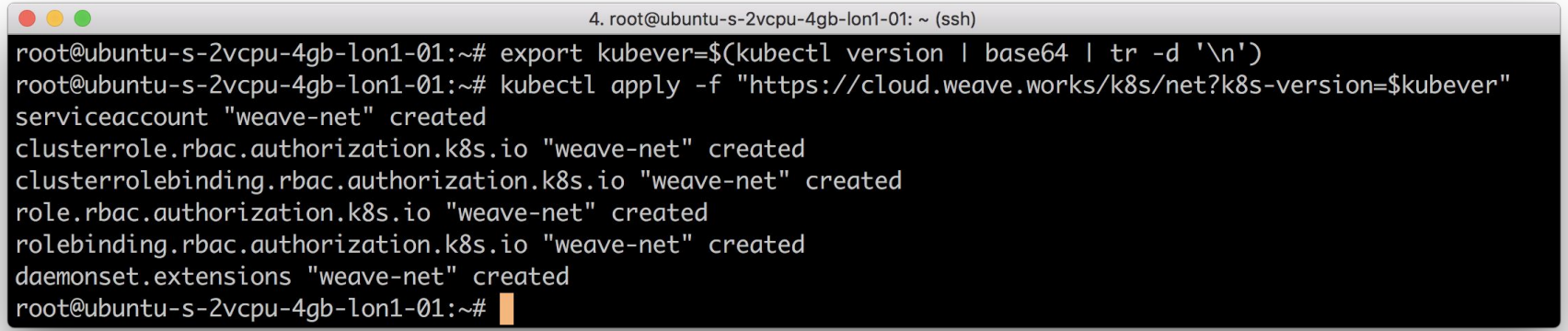
kubeadm join


4. root@ubuntu-s-2vcpu-4gb-lon1-02: ~ (ssh)
This node has joined the cluster:
* Certificate signing request was sent to master and a response
was received.
* The Kubelet was informed of the new secure connection details.
Run 'kubectl get nodes' on the master to see this node join the cluster.
root@ubuntu-s-2vcpu-4gb-lon1-02:~#

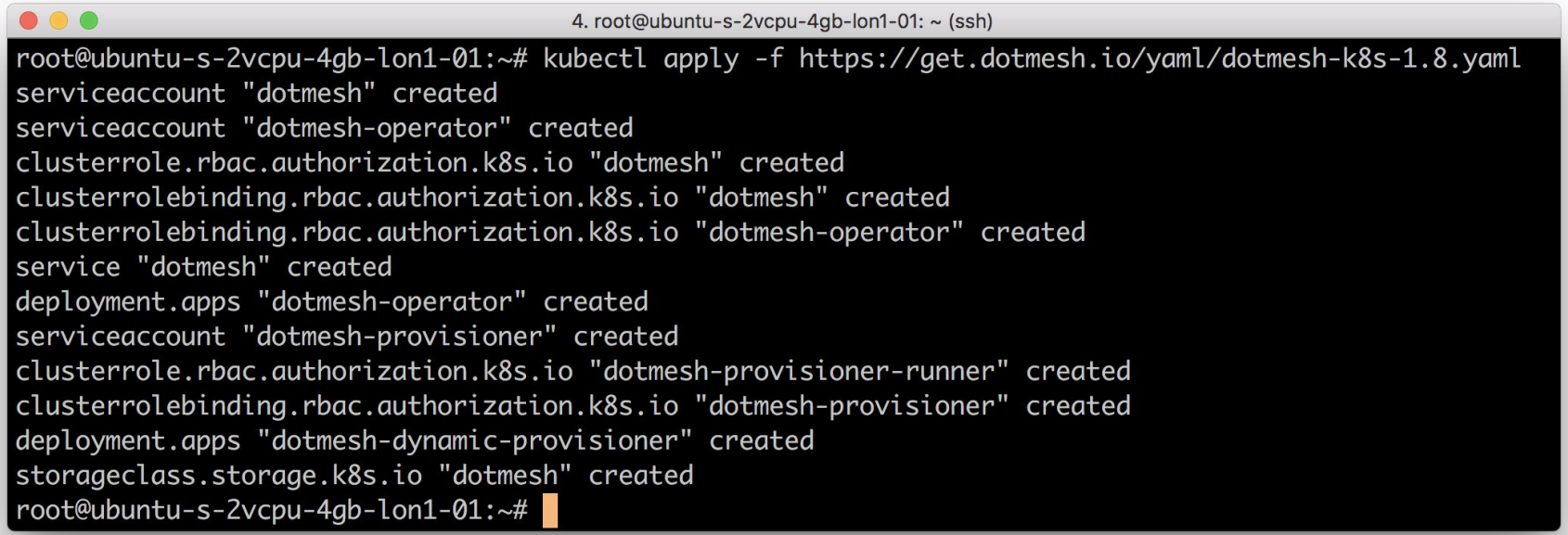

4. root@ubuntu-s-2vcpu-4gb-lon1-03: ~ (ssh)
This node has joined the cluster:
* Certificate signing request was sent to master and a response
was received.
* The Kubelet was informed of the new secure connection details.
Run 'kubectl get nodes' on the master to see this node join the cluster.
root@ubuntu-s-2vcpu-4gb-lon1-03:~#



bare cluster ready for networking



install networking (example)



install storage (example)

cluster ready for your apps!

kubeadm example

- Limitations: initially, kubeadm clusters were:
  - hard to upgrade
  - there was no "easy path" to setting up High Availability
- Over to Tim...

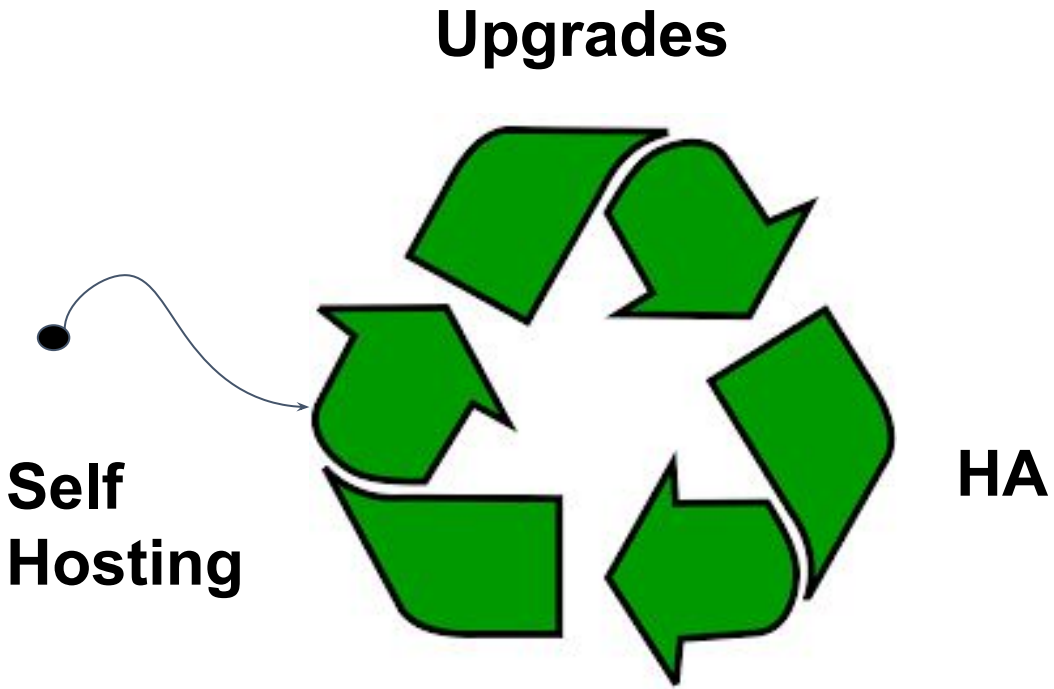
KubeCon | CloudNativeCon

Europe 2018

Self-Hosting, Upgrades, HA

Timothy St. Clair - Heptio

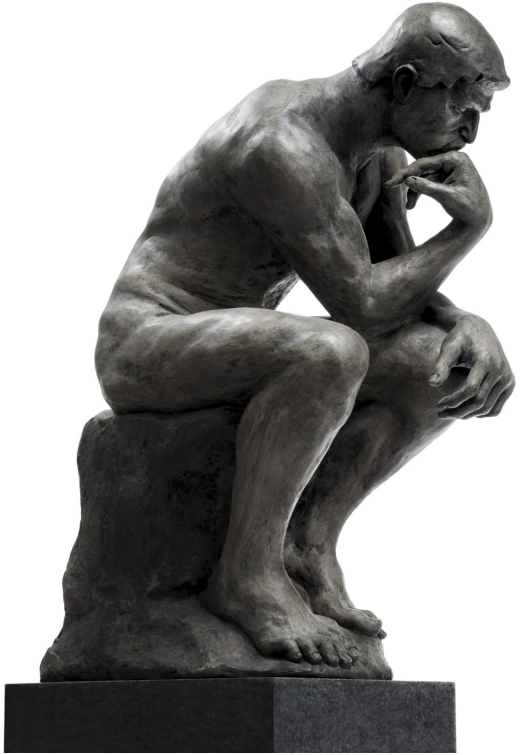
Understanding the Feature Circle



What is Self Hosting?

Running Kubernetes on Kubernetes

- Only the k8s control plane and
    - Not Etcd
    - Not Kubelet
- Ease of upgrades, use kubernetes primitives



Why does this take so long

- There are deep tensions between feature enablement and simple and clean UX. (per lukes earlier points)
  - “Config knobs”
- Distillation of best practices and lessons learned
- Desire to be supportable
- Legacy of long-tail untenable support configurations
  - Breaking feature changes.
- Pending on other feature enablements
  - Component Configuration
  - To checkpoint or not to checkpoint
    - Understanding the security and other dependent implications

What is the status of self hosting?

- Still alpha
- Rethinking the problem to avoid the checkpoint dilemma
  - Current implementations (boot-kube) force checkpointing on pod, secrets, configmaps …
  - You *only* need an api-server to come back online
  - Write-up a KEP on what we are calling the Sentinel, or “Pilot Light”

Rethinking of Self Hosting (Edge)

1. Kubeadm deploys a single static manifest (kubeadm-sentinel)
2. Sentinel checks is api-server is running
   a. checks on well defined sentinel file (/var/run/kubernetes)
3. If not, deploys static manifest using host volume mounts for certs
4. Waits for local kubelet to checkin and restart it's bound pods
5. Self hosted sentinel+apiserver pod restarts and drops sentinel marker
6. Static sentinel shuts down api-server and enters wait-loop

Upgrades

- Distillation of Best Practices
- Ensuring we only rely on Beta+ Features
- Need to create better test jiggery
  - Want a canonical provisioning tool “cluster-api”

High Availability - understanding

There is often a conflation between HA of the control plane and HA of your workloads.

- Focus more on your workloads
- The control plane can recover from a prolonged outage
- Provide alerting on your master nodes
- Understand your tolerances
  - MTTF
  - MTTR
  - Flux rate, or gradient, of your cluster “How much churn”

High Availability - status

(consensus) etcd

- Can be done today using docs today
- Better documentation coming in 1.11phases

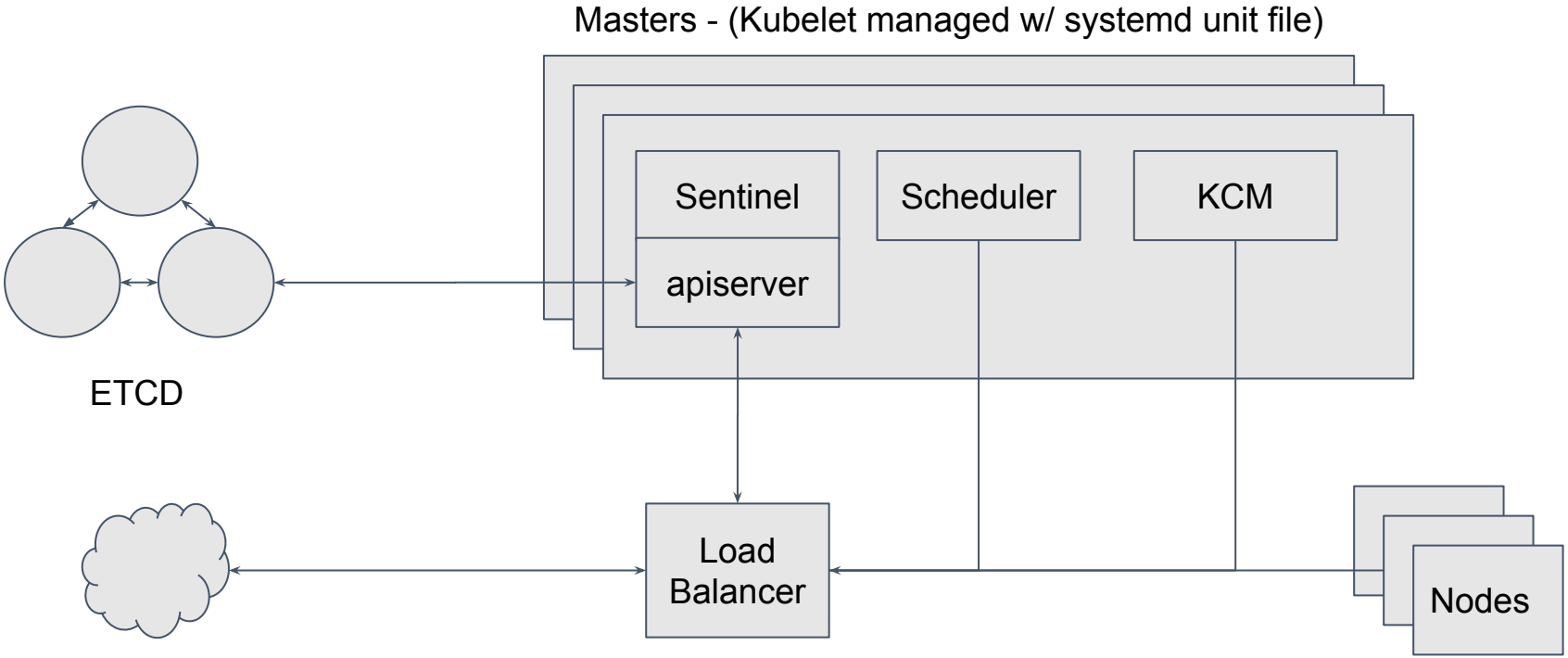
(active-active) api-server

- Requires configuration changes, and is documented

(active-passive) controller manager, scheduler

- Needs shift to component config + config map locking
- Also can be done today

Example Deployment



Deployment Constraints

- Planning
  - Ingress and Egress
  - air-gapping
  - LB’ers
- … Over to Alexander.

KubeCon | CloudNativeCon

Europe 2018

kubeadm in enterprise environment

Alexander Kanevskiy - Intel

Kubeadm in enterprise: The problem statement

Problem: many users are experiencing issues of using kubeadm in not-so-ideal environments

- Installation and upgrades
- Fine-tuning startup parameters
- “Calling home” and offline Kubernetes cluster installs
- Network and proxies

Kubeadm in enterprise: Install and Upgrade

How users are getting kubeadm

- Supported distributions
  - Container Linux
  - DEBs and RPMs:
    Ubuntu, Debian, Hypriot, RHEL, CentOS, Fedora
- There are other distros
  - OpenSuSE, ArchLinux, …

What is actually needed on the node

- kubeadm
- kubelet
- kubectl
- CNI plugins

For unsupported distros you can use Container Linux section for manually install binaries
https://kubernetes.io/docs/setup/independent/install-kubeadm/

Kubeadm in enterprise: Fine-tuning kubelet systemd unit

SystemD units shipped with kubeadm:

- /etc/systemd/system/kubelet.service
  - https://raw.githubusercontent.com/kubernetes/kubernetes/${RELEASE}/build/debs/kubelet.service
- /etc/systemd/system/kubelet.service.d/10-kubeadm.conf
  - https://raw.githubusercontent.com/kubernetes/kubernetes/${RELEASE}/build/debs/10-kubeadm.conf

Local configuration:

- /etc/systemd/system/kubelet.service.d/90-local.conf

```
[Service]
Environment="KUBELET_CGROUP_ARGS=--cgroup-driver=cgroupfs"
Environment="KUBELET_EXTRA_ARGS=--fail-swap-on=false"
```

Kubeadm in enterprise: Offline installation

- “Calling home”
  - --kubernetes-version
    - stable, stable-1, stable-1.9, ...
    - latest, latest-1, latest-1.10, ...
    - ci/latest-1.11
  - upgrade plan

- Images from k8s.gcr.io
  - pause
  - etcd
  - kube-apiserver
  - kube-controller-manager
  - kube-scheduler
  - kube-proxy

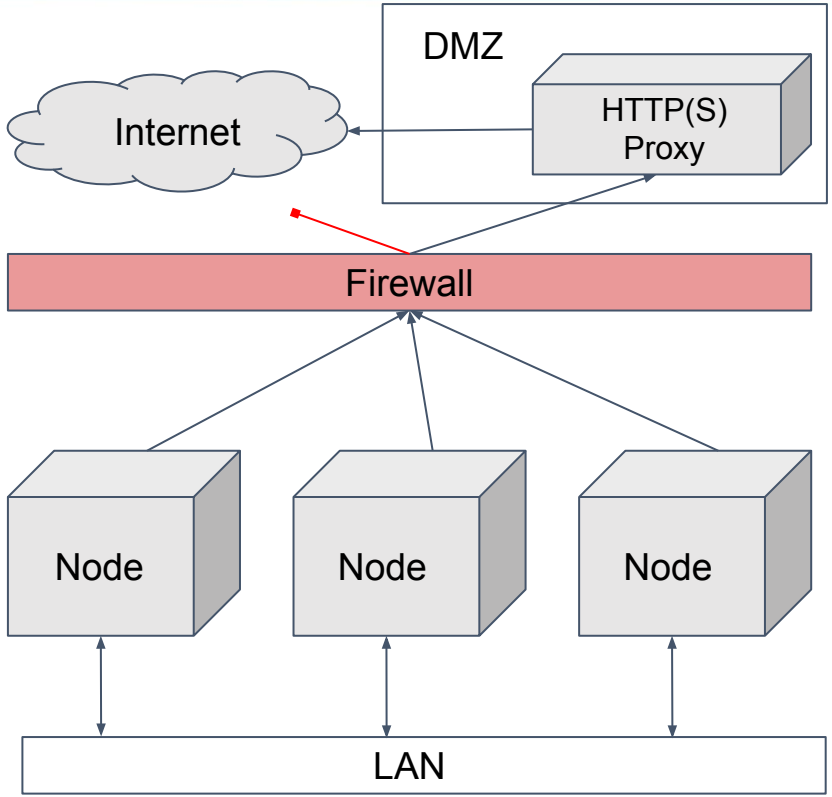
https://dl.k8s.io/release/${RELEASE}/bin/linux/amd64

```
$ gsutil ls -l gs://kubernetes-release/release/v1.10.2/bin/linux/amd64/
```

Kubeadm in enterprise: Network and Proxies

- Proxy for CRI
  - HTTP_PROXY
  - HTTPS_PROXY
  - NO_PROXY
    - Attention: local registries
- Proxy for kubeadm
  - HTTP_PROXY
  - HTTPS_PROXY
  - NO_PROXY
    - Node IPs range
    - Service IPs range
    - POD IPs range

```
NO_PROXY=example.com,192.168.0.0/16,10.0.0.0/8
```



Thank you!

- Questions?
- Further reading
  - https://kubernetes.io/blog/2016/09/how-we-made-kubernetes-easy-to-install
  - https://kubernetes.io/docs/setup/independent/install-kubeadm/
  - https://docs.dotmesh.com/install-setup/kubernetes/
  - https://kubernetes.io/docs/setup/independent/high-availability/
  - https://github.com/kubernetes/kubeadm for issues