



KubeCon



CloudNativeCon

Europe 2018

kubeadm deep dive

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



Agenda



KubeCon



CloudNativeCon

Europe 2018

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



KubeCon



CloudNativeCon

Europe 2018

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



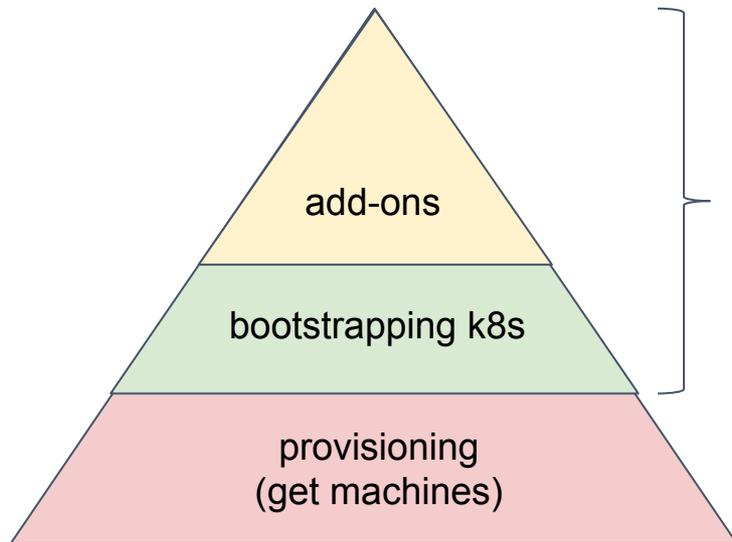
KubeCon



CloudNativeCon

Europe 2018

- 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



KubeCon



CloudNativeCon

Europe 2018

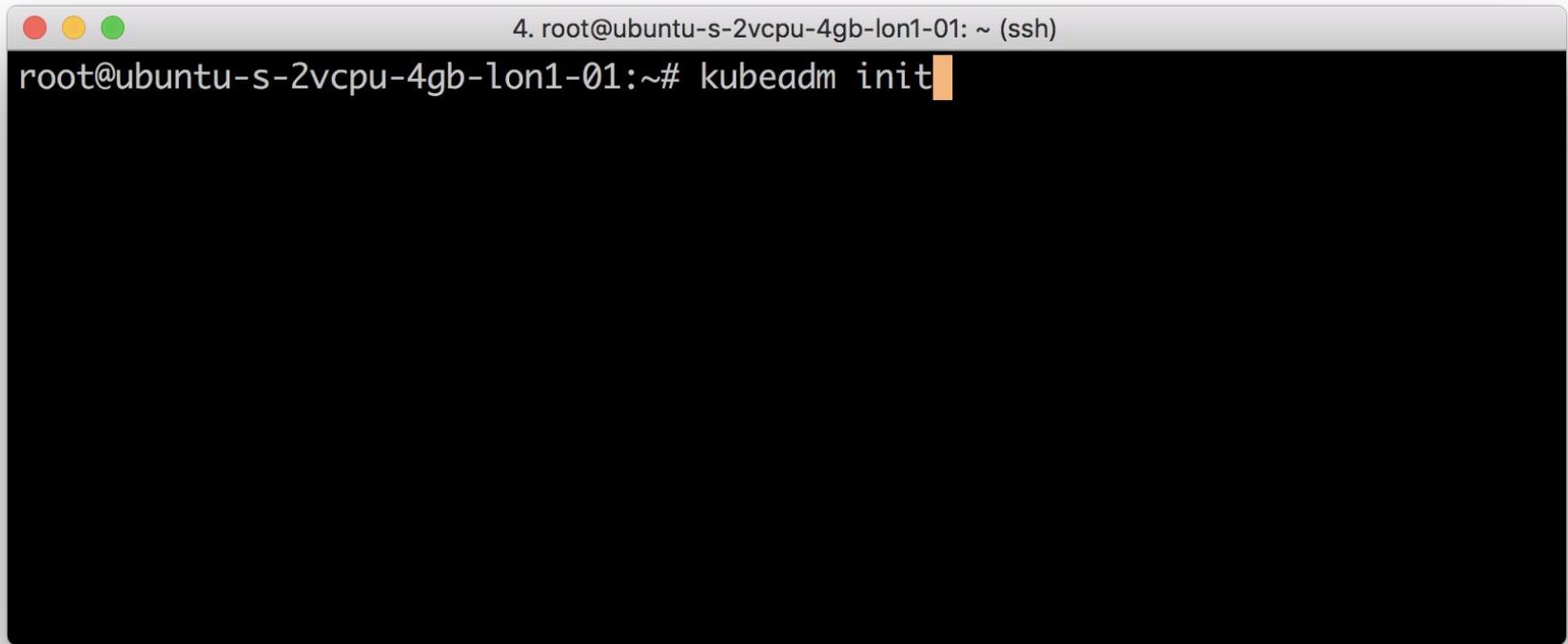
- First release in Kubernetes 1.4
- Step 1: get some computers running Linux
- Then...

```
4. root@ubuntu-s-2vcpu-4gb-lon1-01: ~ (ssh)
root@ubuntu-s-2vcpu-4gb-lon1-01:~# apt-get install -y docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'apt autoremove' to remove it.
The following additional packages will be installed:
  bridge-utils cgroupfs-mount containerd runc ubuntu-fan
Suggested packages:
  mountall aufs-tools debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils cgroupfs-mount containerd docker.io runc ubuntu-fan
```

install docker

```
4. root@ubuntu-s-2vcpu-4gb-lon1-01: ~ (ssh)
root@ubuntu-s-2vcpu-4gb-lon1-01:~# apt-get install -y kubelet kubeadm kubectl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'apt autoremove' to remove it.
The following additional packages will be installed:
  ebtables kubernetes-cni socat
The following NEW packages will be installed:
  ebtables kubeadm kubectl kubelet kubernetes-cni socat
```

install kubelet, kubeadm, kubectl

A terminal window with a grey title bar containing three colored window control buttons (red, yellow, green) on the left and the text '4. root@ubuntu-s-2vcpu-4gb-lon1-01: ~ (ssh)' on the right. The main area of the terminal is black with white text. The prompt 'root@ubuntu-s-2vcpu-4gb-lon1-01:~#' is followed by the command 'kubeadm init' and a small orange cursor block at the end of the line.

```
4. root@ubuntu-s-2vcpu-4gb-lon1-01: ~ (ssh)
root@ubuntu-s-2vcpu-4gb-lon1-01:~# kubeadm init
```

kubeadm init

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.krz4aicqumhjh1r --discovery-token-ca-cert-hash sha256:3b901ad6e5b293cc21c6e64b8d950f11571dc8e5a47b303be0dd12e5926ccd7d
```

```
root@ubuntu-s-2vcpu-4gb-lon1-01:~#
```

```
4. root@ubuntu-s-2vcpu-4gb-lon1-02: ~ (ssh)
root@ubuntu-s-2vcpu-4gb-lon1-02:~# kubectl join 167.99.200.30:6443 --token dr3ept
.krz4aicqumhjh1r --discovery-token-ca-cert-hash sha256:3b901ad6e5b293cc21c6e64b8
d950f11571dc8e5a47b303be0dd12e5926ccd7d
```

kubectl 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:~# █

```
4. root@ubuntu-s-2vcpu-4gb-lon1-01: ~ (ssh)
root@ubuntu-s-2vcpu-4gb-lon1-01:~# kubectl get nodes
NAME                                STATUS    ROLES    AGE     VERSION
ubuntu-s-2vcpu-4gb-lon1-01        NotReady master   7m      v1.10.2
ubuntu-s-2vcpu-4gb-lon1-02        NotReady <none>   4m      v1.10.2
ubuntu-s-2vcpu-4gb-lon1-03        NotReady <none>   1m      v1.10.2
root@ubuntu-s-2vcpu-4gb-lon1-01:~#
```

bare cluster ready for networking

```
4. root@ubuntu-s-2vcpu-4gb-lon1-01: ~ (ssh)
root@ubuntu-s-2vcpu-4gb-lon1-01:~# export kubever=$(kubectl version | base64 | tr -d '\n')
root@ubuntu-s-2vcpu-4gb-lon1-01:~# kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$kubever"
serviceaccount "weave-net" created
clusterrole.rbac.authorization.k8s.io "weave-net" created
clusterrolebinding.rbac.authorization.k8s.io "weave-net" created
role.rbac.authorization.k8s.io "weave-net" created
rolebinding.rbac.authorization.k8s.io "weave-net" created
daemonset.extensions "weave-net" created
root@ubuntu-s-2vcpu-4gb-lon1-01:~# █
```

install networking (example)

```
4. root@ubuntu-s-2vcpu-4gb-lon1-01: ~ (ssh)
root@ubuntu-s-2vcpu-4gb-lon1-01:~# kubectl apply -f https://get.dotmesh.io/yaml/dotmesh-k8s-1.8.yaml
serviceaccount "dotmesh" created
serviceaccount "dotmesh-operator" created
clusterrole.rbac.authorization.k8s.io "dotmesh" created
clusterrolebinding.rbac.authorization.k8s.io "dotmesh" created
clusterrolebinding.rbac.authorization.k8s.io "dotmesh-operator" created
service "dotmesh" created
deployment.apps "dotmesh-operator" created
serviceaccount "dotmesh-provisioner" created
clusterrole.rbac.authorization.k8s.io "dotmesh-provisioner-runner" created
clusterrolebinding.rbac.authorization.k8s.io "dotmesh-provisioner" created
deployment.apps "dotmesh-dynamic-provisioner" created
storageclass.storage.k8s.io "dotmesh" created
root@ubuntu-s-2vcpu-4gb-lon1-01:~# █
```

install storage (example)

cluster ready for your apps!

kubeadm example



KubeCon



CloudNativeCon

Europe 2018

- 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



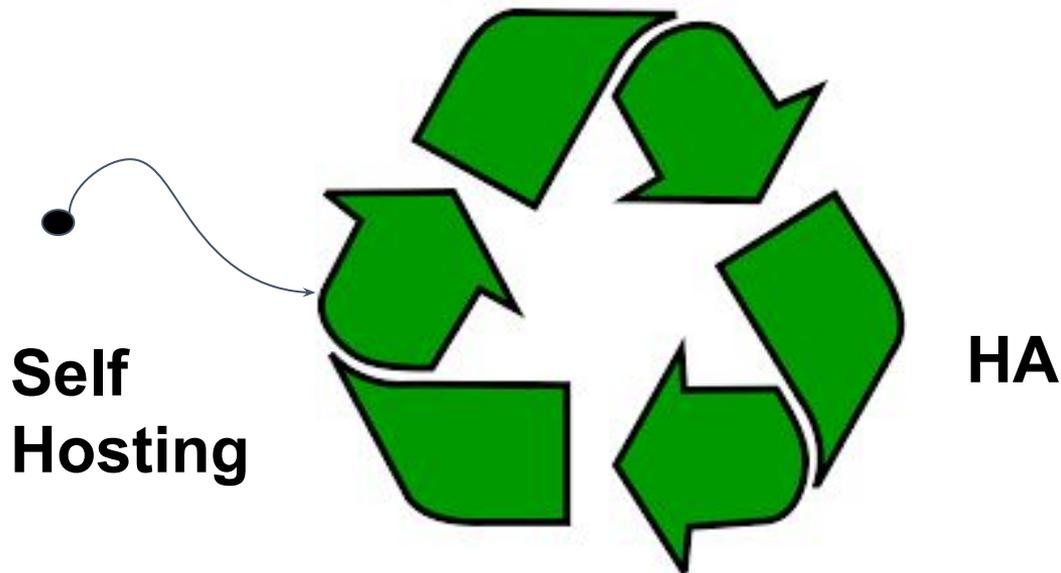
KubeCon



CloudNativeCon

Europe 2018

Upgrades



What is Self Hosting?



KubeCon

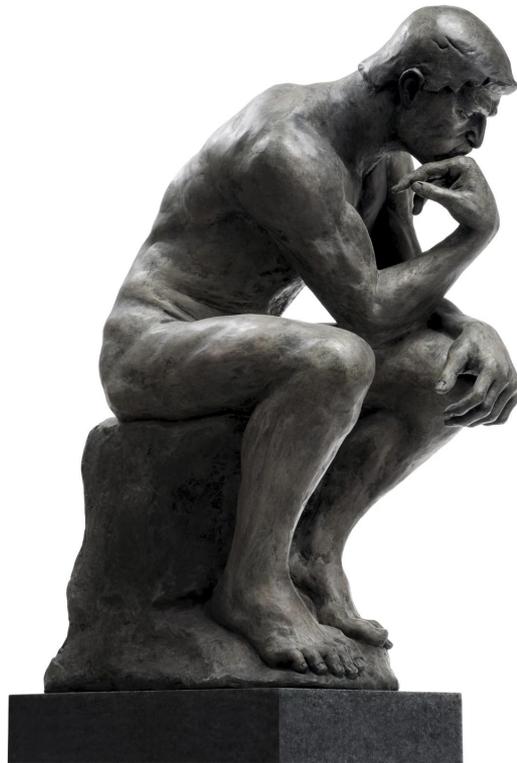


CloudNativeCon

Europe 2018

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



KubeCon



CloudNativeCon

Europe 2018

- 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?



KubeCon



CloudNativeCon

Europe 2018

- 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)



KubeCon



CloudNativeCon

Europe 2018

1. Kubeadm deploys a single static manifest (kubeadm-sentinel)
2. Sentinel checks if 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



KubeCon



CloudNativeCon

Europe 2018

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

High Availability - understanding



KubeCon



CloudNativeCon

Europe 2018

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



KubeCon



CloudNativeCon

Europe 2018

(consensus) etcd

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

(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

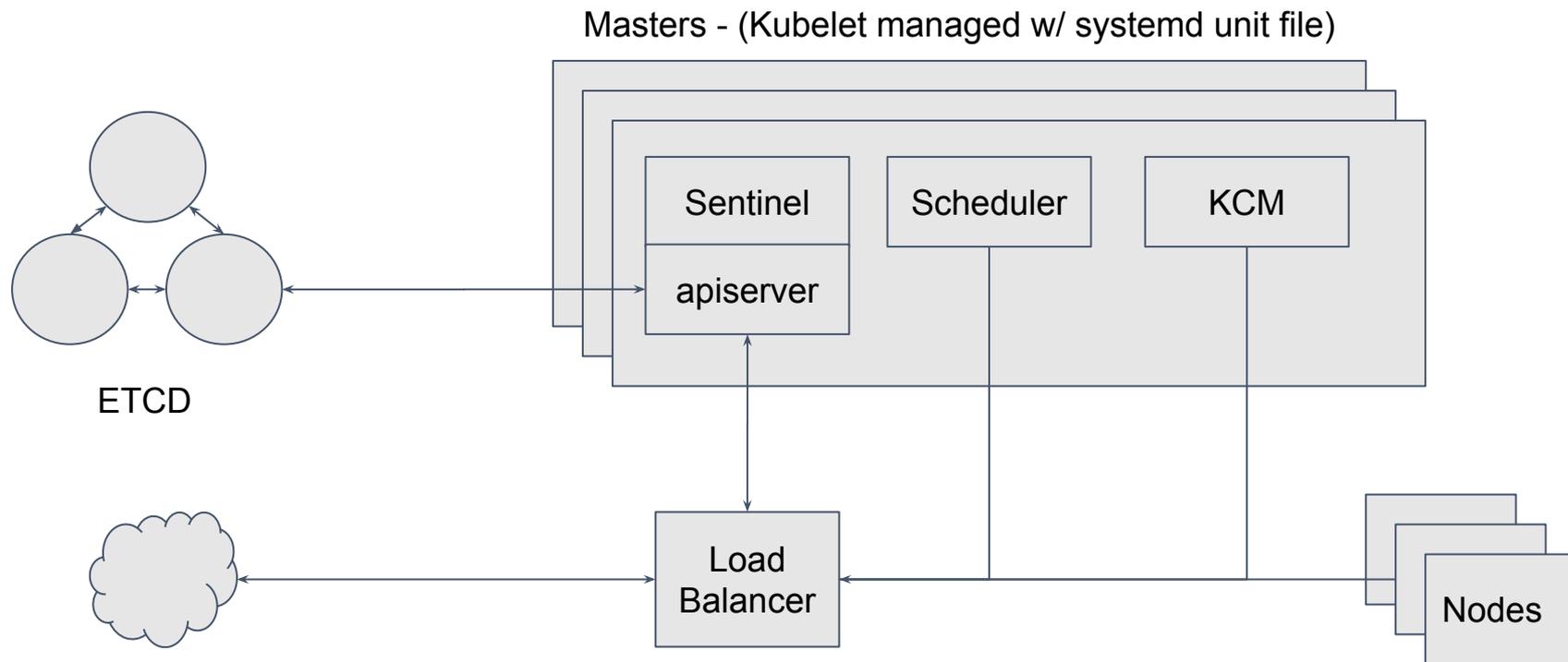


KubeCon



CloudNativeCon

Europe 2018



Deployment Constraints



KubeCon



CloudNativeCon

Europe 2018

- 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



KubeCon



CloudNativeCon

Europe 2018

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



KubeCon



CloudNativeCon

Europe 2018

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



KubeCon



CloudNativeCon

Europe 2018

SystemD units shipped with kubeadm:

- `/etc/systemd/system/kubelet.service`
 - [https://raw.githubusercontent.com/kubernetes/kubernetes/\\${RELEASE}/build/debs/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](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



KubeCon



CloudNativeCon

Europe 2018

- “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](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



KubeCon

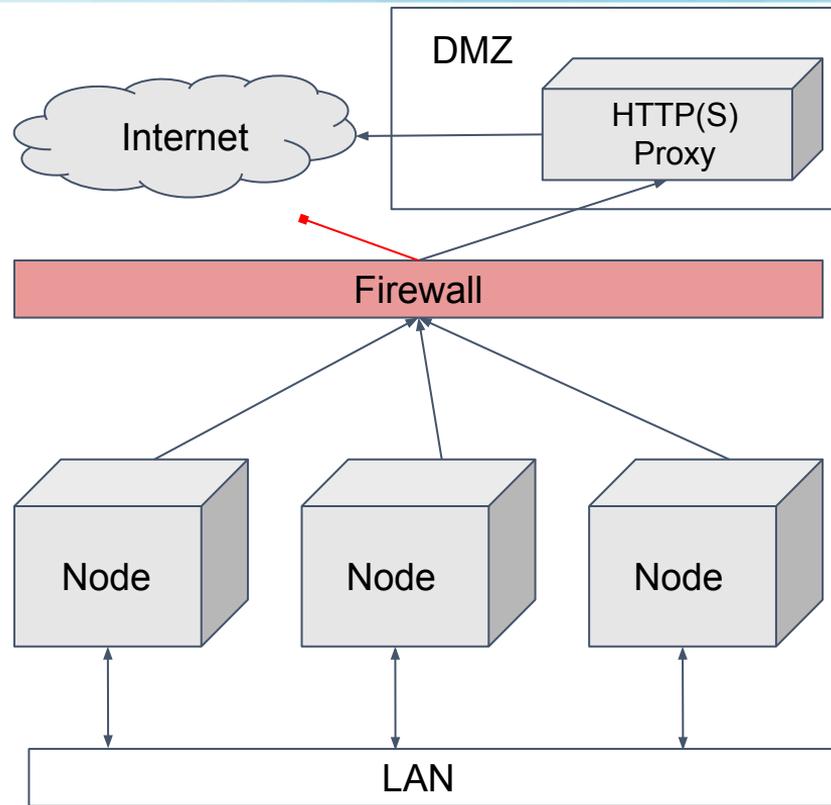


CloudNativeCon

Europe 2018

- 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!



KubeCon



CloudNativeCon

Europe 2018

- 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