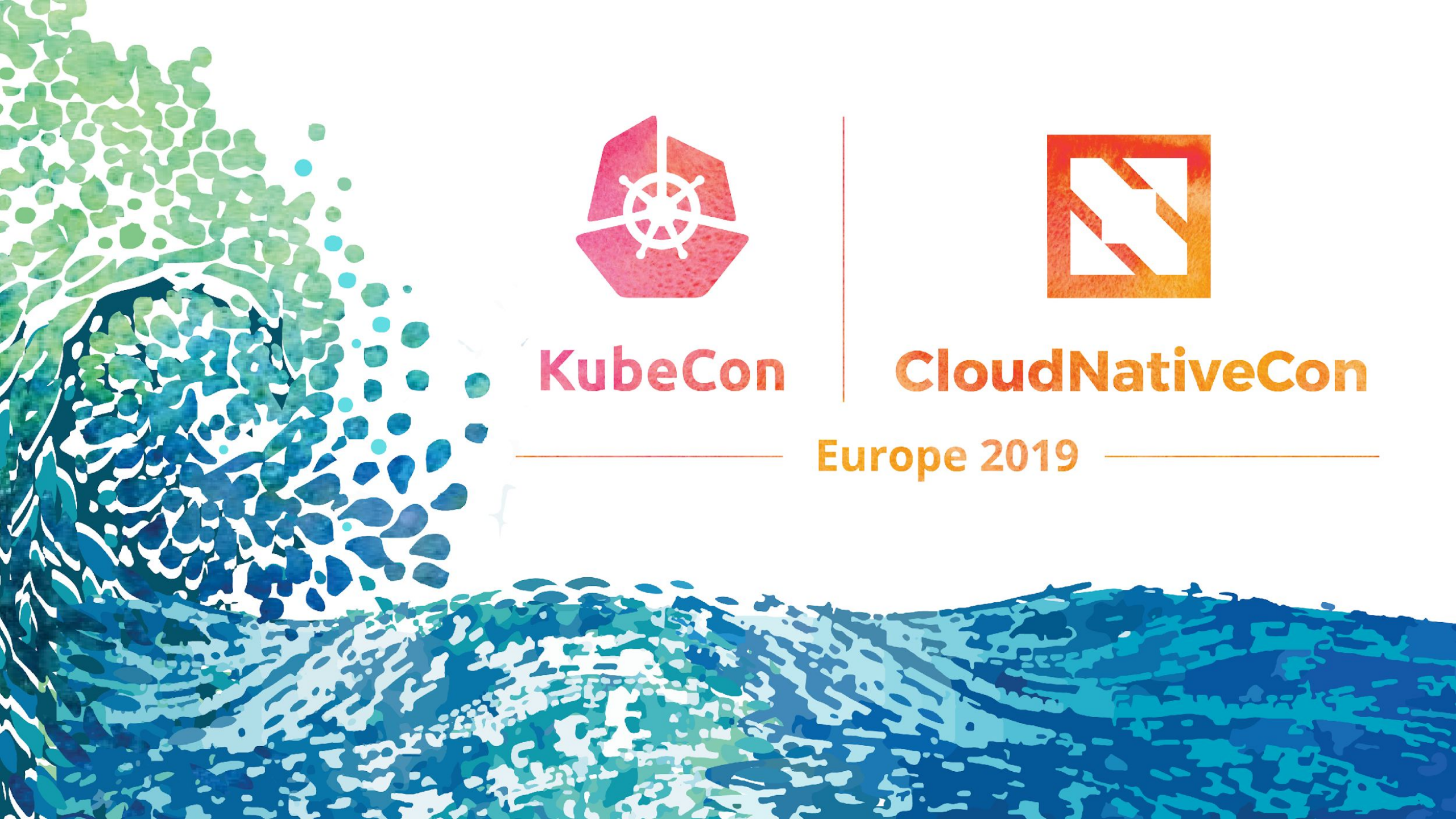

KubeCon
CloudNativeCon
Europe 2019

Kubeadm Deep Dive

SIG Cluster Lifecycle

Who are we?

Lubomir I. Ivanov

SIG Cluster Lifecycle Contributor

Open Source Engineer @VMware

@neolit123

Fabrizio Pandini

SIG Cluster Lifecycle Contributor

Enterprise Architect @UniCredit*

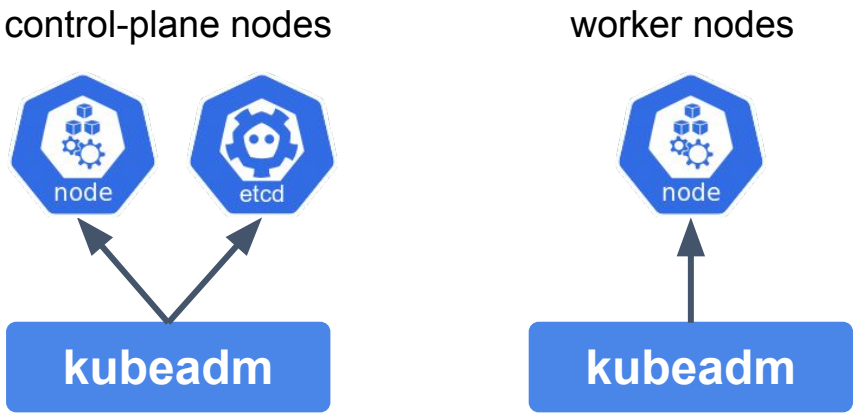
@fabriziopandini

** "The information, estimates and evaluations communicated by the speaker during the event and contained in this document (hereinafter "Document") represent the independent opinion of the speaker/author of the Document, are therefore expressed in a personal capacity and they are in no way attributable and / or referable to the corporate role played in UniCredit Group by the speaker/author of the Document nor to UniCredit itself"*

What is kubeadm

Kubeadm is a tool built to provide best-practice "fast paths" for creating Kubernetes clusters.
It performs the actions necessary to get a minimum viable, secure cluster up and running in a user friendly way.

- Someone or something should provide the machines
- **kubeadm creates a Kubernetes node on the machine**



- Someone or something should install the CNI plugin

Kubeadm: key design takeaways

- The user experience should be *simple*
- The cluster reasonably *secure*
- kubeadm's **scope is intentionally limited:**
  - Only ever deals with the local filesystem and the Kubernetes API
  - Agnostic to how exactly the kubelet is run
  - Setting up or favoring a specific CNI network is out of scope

Recent changes in kubeadm

KubeCon CloudNativeCon Europe 2019

Photo by Taylor Grote on Unsplash

Kubeadm is GA!

What does GA really mean?

Stable command-line UX

Command or flag that exists in a GA version must be kept for at least 12 months after deprecation

Stable underlying implementation

The control plane is run as a set of static Pods, ComponentConfig is used for configuring installed components (as of today only kubelet, kube-proxy) and BootstrapTokens are used for the kubeadm join flow

Upgrades between minor versions

kubeadm configuration file

You can now tune almost every part of the cluster declaratively

```
apiVersion: kubeadm.k8s.io/v1beta1
kind: ClusterConfiguration
kubernetesVersion: "v1.12.2"
networking:
  serviceSubnet: "10.96.0.0/12"
  dnsDomain: "cluster.local"
etcd:
 ...
apiServer:
 extraArgs:
    ...
 extraVolumes:
    ...
```

You can tune also the properties of the node where kubeadm is executed

```
apiVersion: kubeadm.k8s.io/v1beta1
kind: InitConfiguration
localAPIEndpoint:
   advertiseAddress: "10.100.0.1"
   bindPort: 6443
nodeRegistration:
  criSocket: "/var/run/crio/crio.sock"
  kubeletExtraArgs:
    cgroupDriver: "cgroupfs"
```

```
apiVersion: kubeadm.k8s.io/v1beta1
kind: JoinConfiguration
...
```

kubeadm phases

The “toolbox” interface of kubeadm — Also known as **phases**.
If you don’t want to perform all kubeadm init tasks, you can instead apply more fine-grained actions using the kubeadm init phase command

v.13 **kubeadm init phase**

```
preflight
kubelet-start
certs
  /...
kubeconfig
  /...
control-plane
  /...
etcd
upload-config
  /..
```

v.14 **upload-certs [EXPERIMENTAL]**

```
mark-control-plane
bootstrap-token
addon
  /...
```

v.14 **kubeadm join phase**

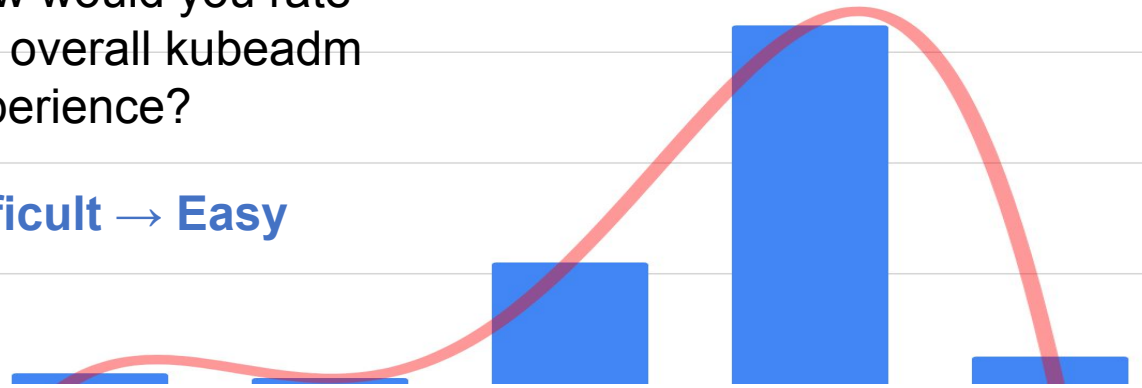
```
preflight
control-plane-prepare
  /download-certs [EXPERIMENTAL]
  /certs
  /kubeconfig
  /control-plane
kubelet-start
control-plane-join
  /etcd
  /update-status
  /mark-control-plane
```

Kubeadm survey

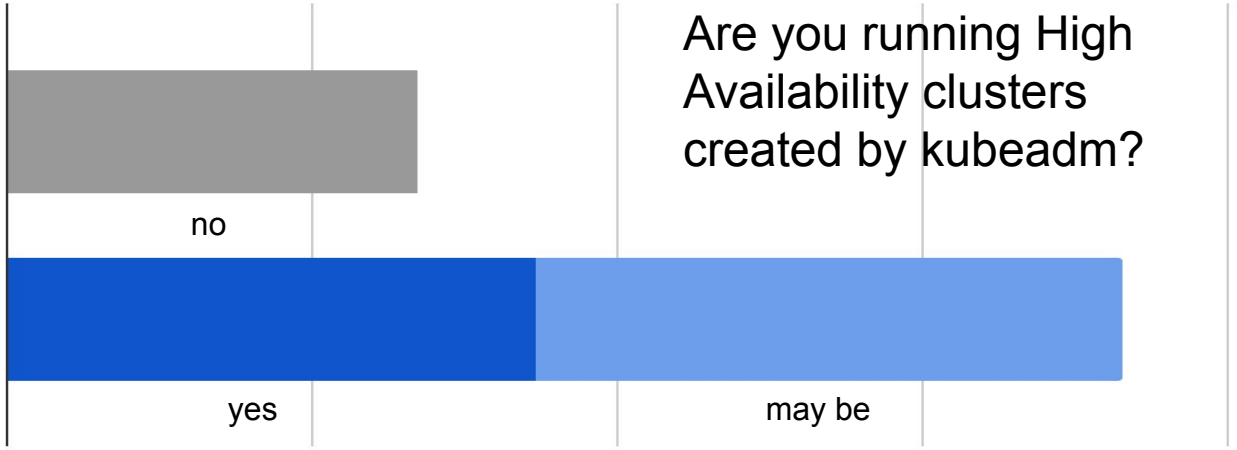
thank you

How would you rate the overall kubeadm experience?

Difficult → Easy



Are you running High Availability clusters created by kubeadm?



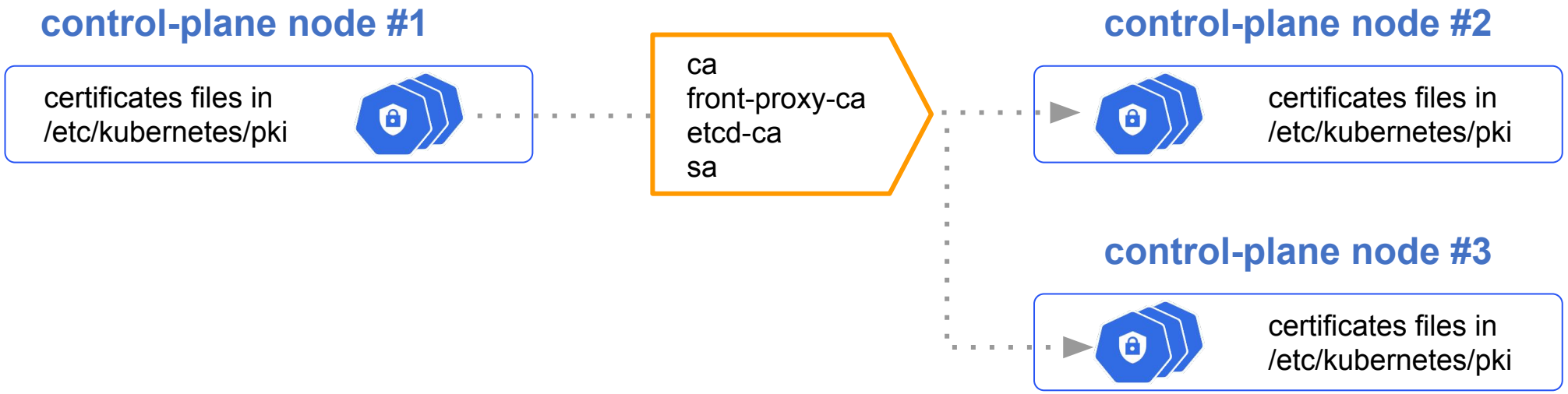
Deep dive: HA in kubeadm - part 1

KubeCon CloudNativeCon Europe 2019

Photo by Dan Gold on Unsplash

Certificates copy in a nutshell

When creating a K8s HA cluster, **certificate authorities and service account signing key must be shared across all the control-plane nodes** in order to make the cluster work



Why should you care?

kubeadm implements support for automating certificate copy across control-plane nodes

Why should you care about kubeadm automatic certificates copy?

- It simplify K8s administrators life when creating HA clusters
  (no more ssh, scp, scripts for copying certificates)
- It is really important to understand how critical parts of the K8s PKI are managed

How it works @ init time

Pass the **--experimental-upload-certs** flag to instruct kubeadm to prepare for certificate copy

certificates files are created in /etc/kubernetes/pki folder (as usual)

certificates files that must be shared across control-plane nodes are **encrypted** and uploaded into the **kubeadm-certs** Secret

the kubeadm output provide instruction for joining another control-plane node and a **certificate key** for getting access to the uploaded certificates

```
kubeadm init --experimental-upload-certs                    1
...
[certs] Using certificateDir folder "/etc/kubernetes/pki"   2
[certs] Generating "ca" certificate and key
[certs] Generating "sa" key and public key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "etcd/ca" certificate and key
...
[upload-certs] storing the certificates in secret            3
               "kubeadm-certs" in the "kube-system"
               Namespace
...
Your Kubernetes control-plane has initialized successfully!
...
You can now join any number of the control-plane node running
   the following command on each as root:

  kubeadm join 172.17.0.4:6443 --token abcdef...\
    --discovery-token-ca-cert-hash sha256:... \
    --experimental-control-plane \                           4
    --certificate-key 0123456789012345678901234567....
```

How it works @ join time

Pass the **--certificate-key** to trigger automatic copy of certificates when joining

kubeadm join reads the **kubeadm-certs** secret, decrypt it using the **certificate key**, and saves all the shared certs in the /etc/kubernetes/pki folder

```
kubeadm join 172.17.0.4:6443 --token abcdef...\
  --discovery-token-ca-cert-hash sha256:... \
  --experimental-control-plane \
  --certificate-key 0123456789012345678901234....
...
[preflight] Reading configuration from the cluster
...
[download-certs] Downloading the certificates in Secret
                 "kubeadm-certs" in the "kube-system"
                 Namespace
...
[certs] Using certificateDir folder "/etc/kubernetes/pki"
...

This node has joined the cluster and a new control plane
instance was created!
```

1

2

Key takeaways!

At init time, certificates to be shared are encrypted and uploaded into the **kubeadm-certs** secret
At join time, certificates are downloaded and decrypted using the **certificate key**

The certificate key must be kept safe!
If someone gets the certificate key and gets access to the kubeadm-certs secret, someone can destroy your cluster!

As a risk mitigation strategy, **the kubeadm-certs secret gets automatically deleted after two hours**. You can upload again certificates and generate a new certificate key any time by using `kubeadm init phase upload-certs`

In case you are using an external etcd cluster, etcd certificates should be provided by you on the first control-plane node only

In case you are providing an externally generated CA (without providing keys), you can't use automatic copy certificate function; you must provide CA, certificates and kubeconfig files on all nodes by other means

Deep dive: HA in kubeadm - part 2

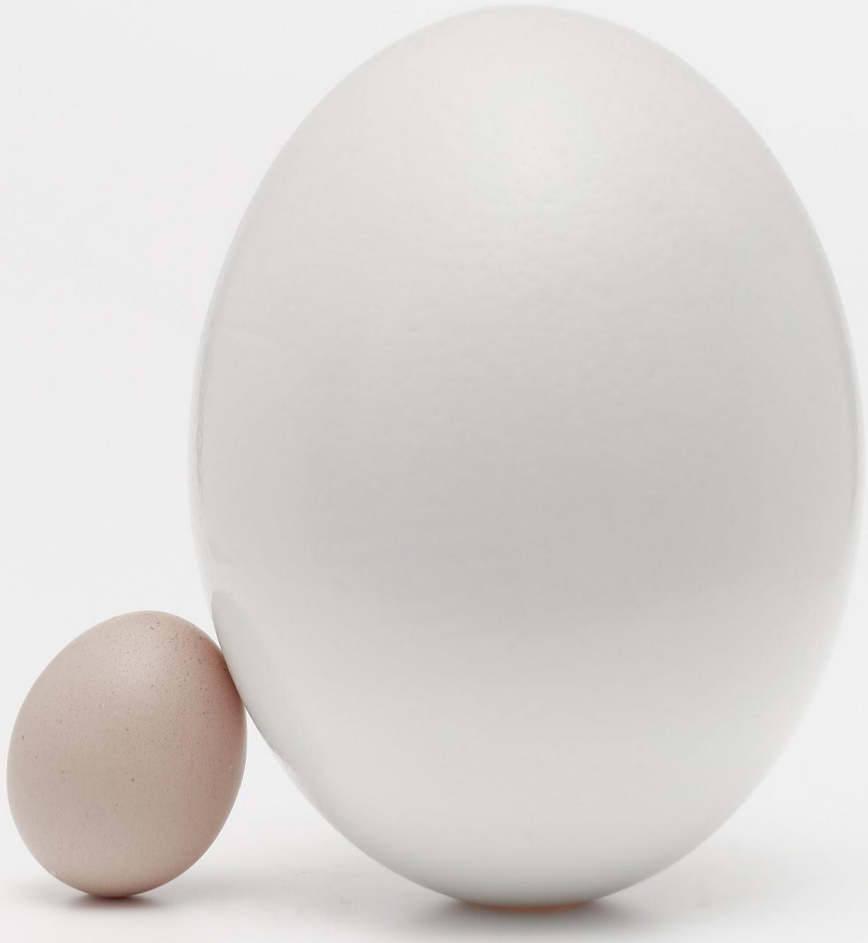
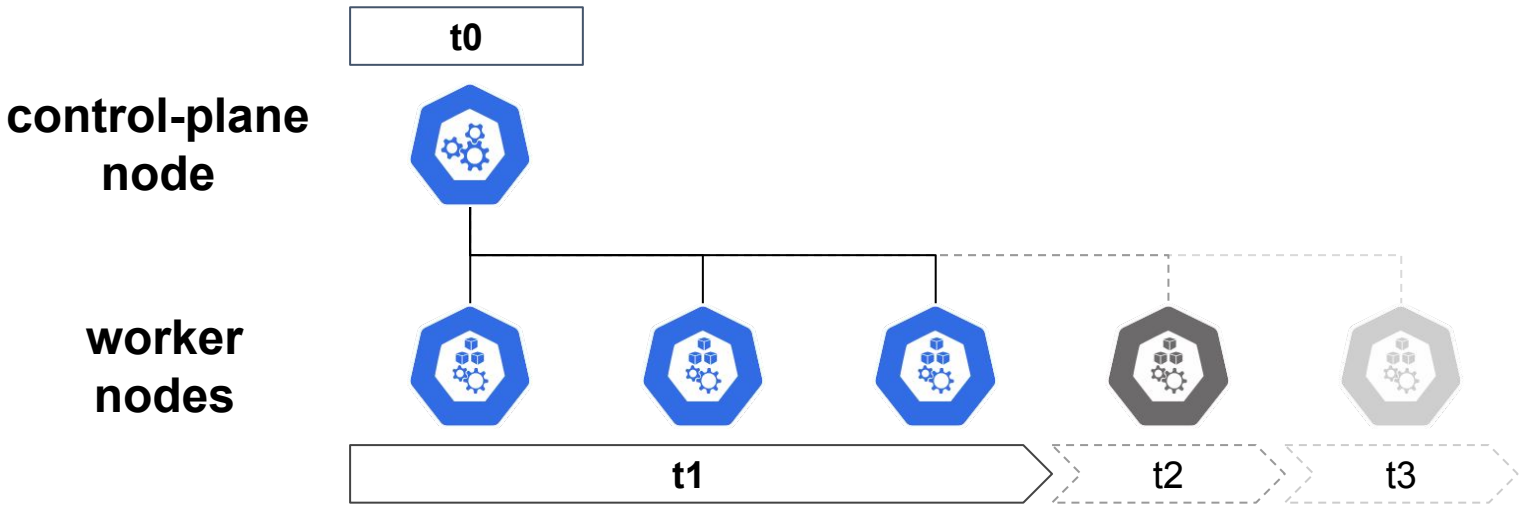


Photo by Daniele Levis Pelusi on Unsplash

Dynamic workflow in a nutshell

The kubeadm distinctive init-join workflow allows you to **dynamically grow** your cluster



initialize your cluster:
kubeadm init

dynamically grow worker nodes:
kubeadm join

Why should you care?

Why should you care about kubeadm dynamic workflow (again)?

- Because HA in kubeadm is implemented by dynamically growing the control-plane nodes (using the same UX pattern already used for worker nodes)
- Because dynamically growing control-plane nodes requires some additional considerations

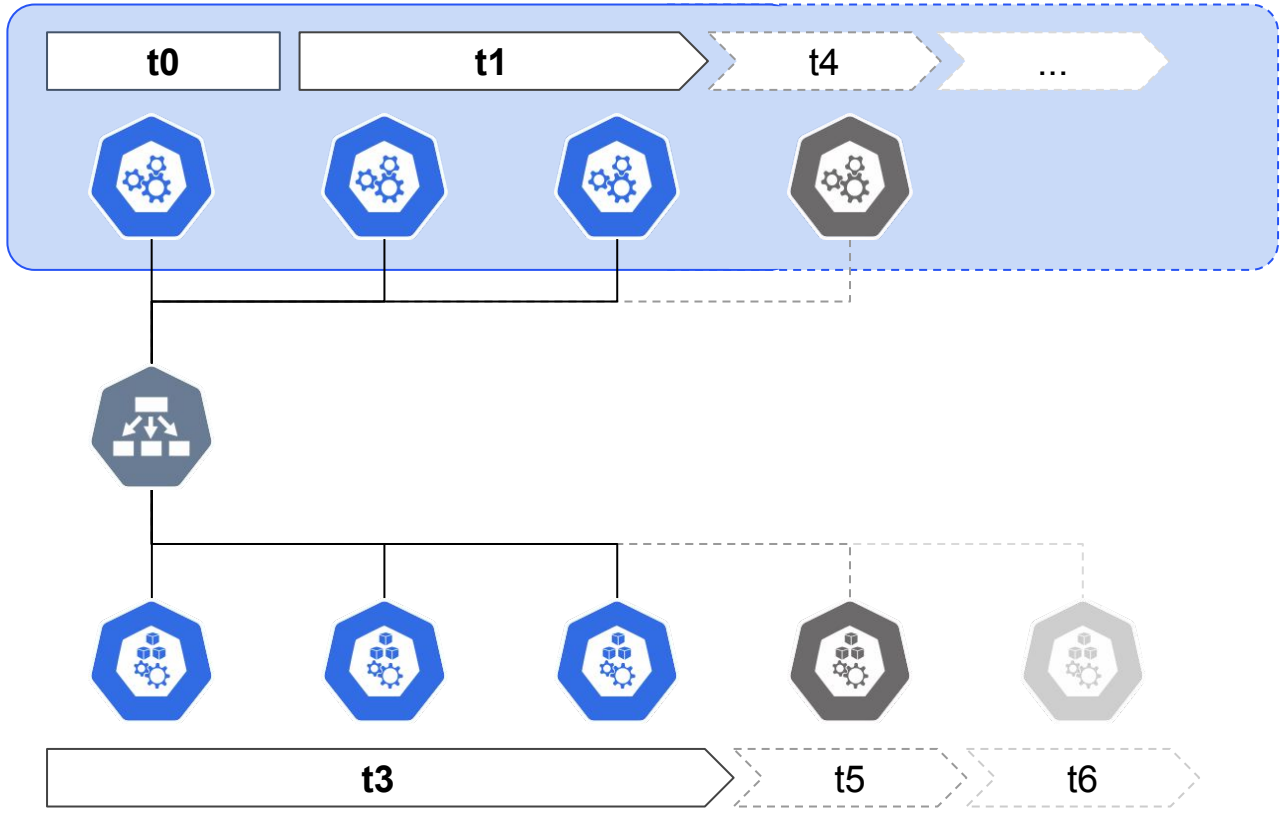
The external load balancer

In order to dynamically grow the control-plane nodes you need an **external load balancer**

HA control-plane

external load balancer

worker nodes



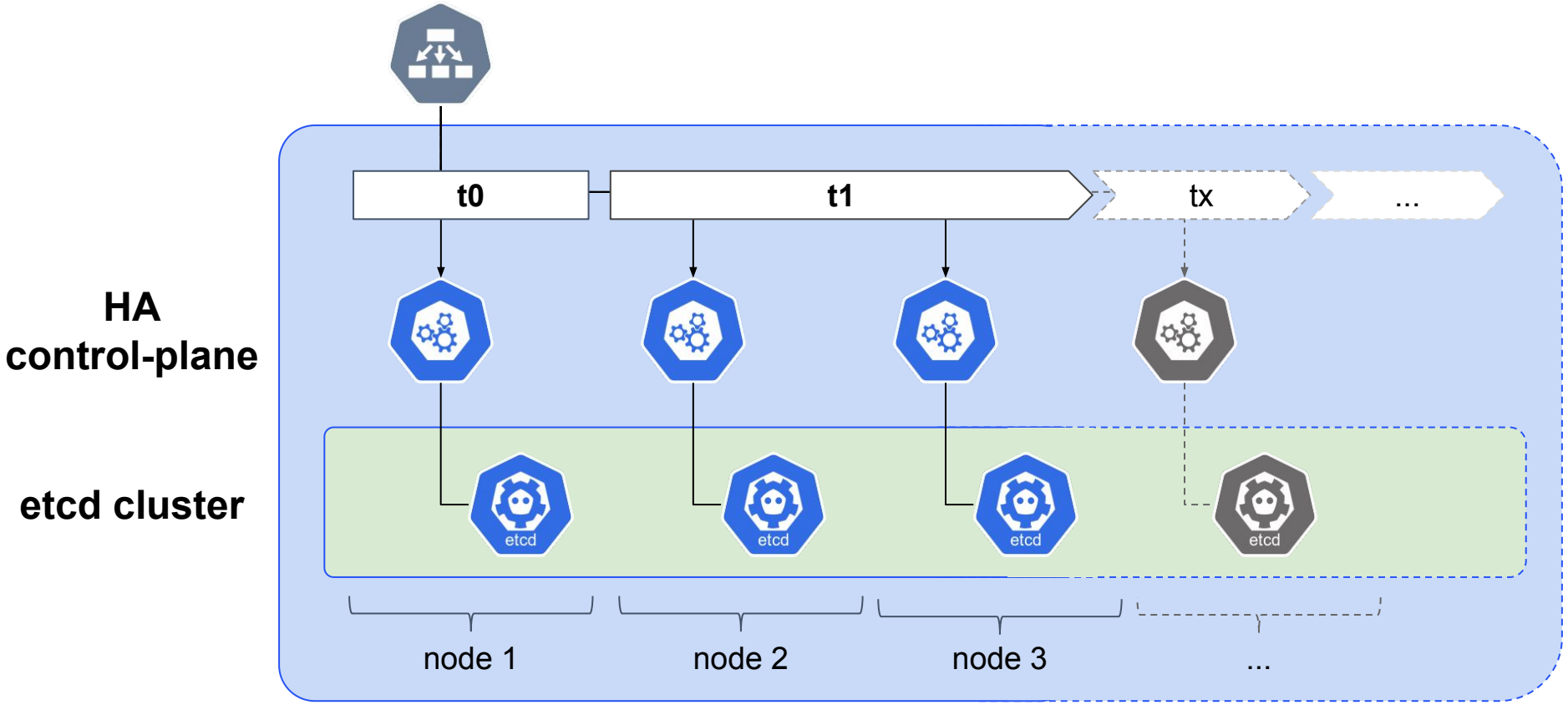
initialize your cluster:
kubeadm init

dynamically grow control-plane nodes:
kubeadm join
--experimental-control-plane

dynamically grow worker nodes:
kubeadm join

Stacked etcd

In case you are not providing an external etcd cluster, **kubeadm creates an etcd node stacked on the same node where the control-plane exist**. Also the stacked etcd cluster dynamically grows



kubeadm join
--experimental-control-plane
dynamically grows the stacked etcd cluster

Key takeaways!

In order to setup an HA control plane, **Before** you need an **external load balancer**
Then init the cluster
After join control-plane nodes and worker nodes **in any order/at any time**

In case you are not providing an external etcd, **a stacked etcd cluster is automatically generated** and it spans across all the control-plane nodes

Api-server certificate, etcd server/peer and other **certificates are node specific**.
You cannot copy them around.

Each api server instance is connected **only to the local etcd member**.
if an etcd member fails on a node, the entire control-plane on that node fails.

If you want to preserve the dynamic workflow feature, **don't override "addresses" fields for kube-apiserver or etcd** (e.g. --advertise-address or listen-client-urls) **.**

Bonus pack

KubeCon CloudNativeCon Europe 2019

Photo by Math on Unsplash

The starting point

Creating a single control-plane node with kubeadm => create local artifacts + in-cluster artifacts

certificates files in
/etc/kubernetes/pki

kubeconfig files in
/etc/kubernetes

static pod manifests in
/etc/kubernetes/manifest

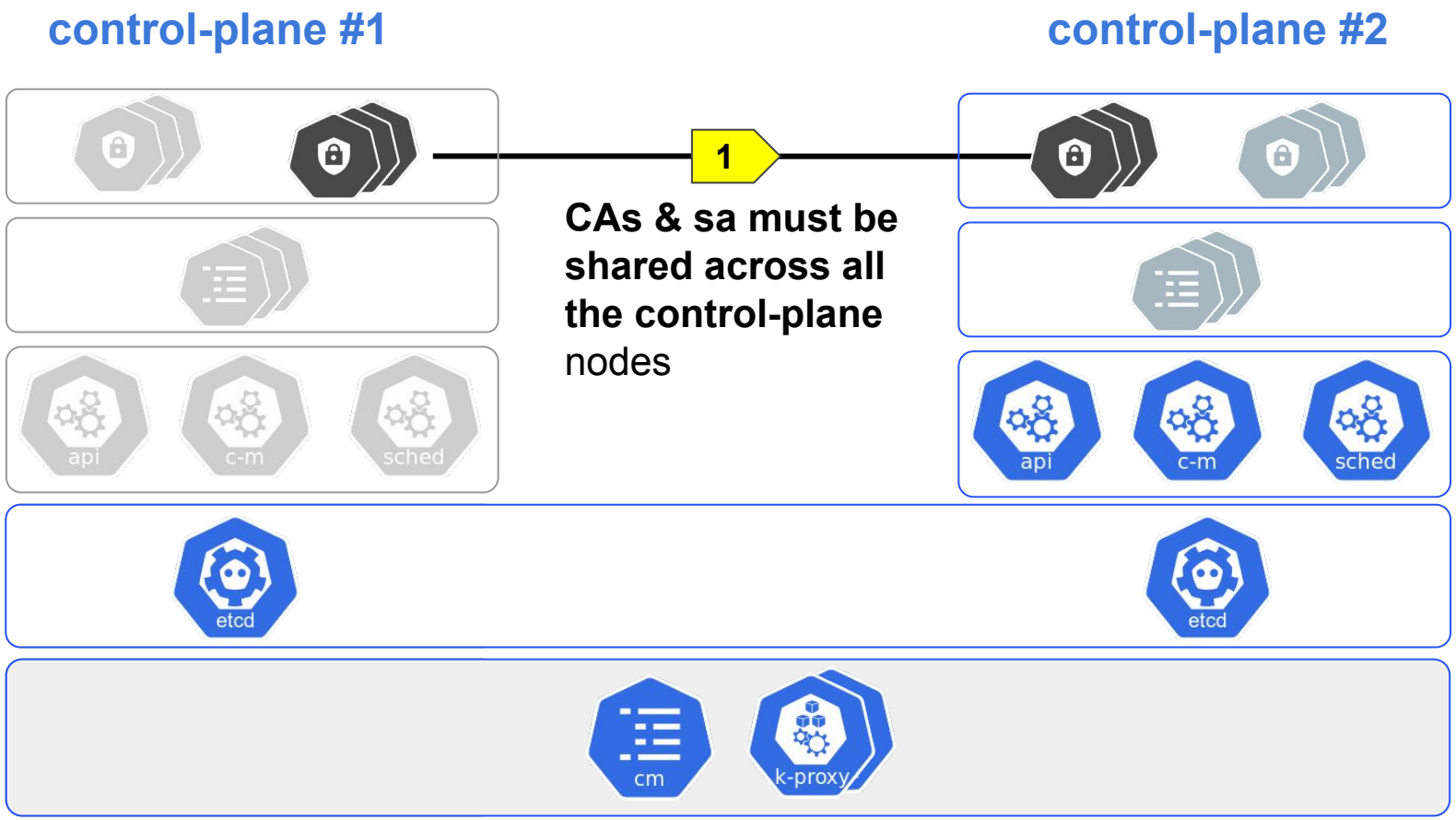
kubeadm ConfigMap
+ core addons + RBAC rules,
bootstrap-tokens
are deployed in the
K8s cluster

```
$ kubeadm init
...
[certs] Using certificateDir folder
  "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
...
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
...
[control-plane] Using manifest folder
  "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for
  "kube-apiserver"
...
[etcd] Creating static Pod manifest for local etcd in
  "/etc/kubernetes/manifests"
...
[addons] Applied essential addon: CoreDNS
...
Your Kubernetes control-plane has initialized
  successfully!
```

1 2 3 4

The grand theory of HA in kubeadm

Adding a second control-plane, requires again to create certificates, kubeconfig, manifests, but….



2 Generate only **node specific certificates**

3 Generate kubeconfig files, **except kubelet.conf** (TLS bootstrap)

4 Generate the static pod manifests

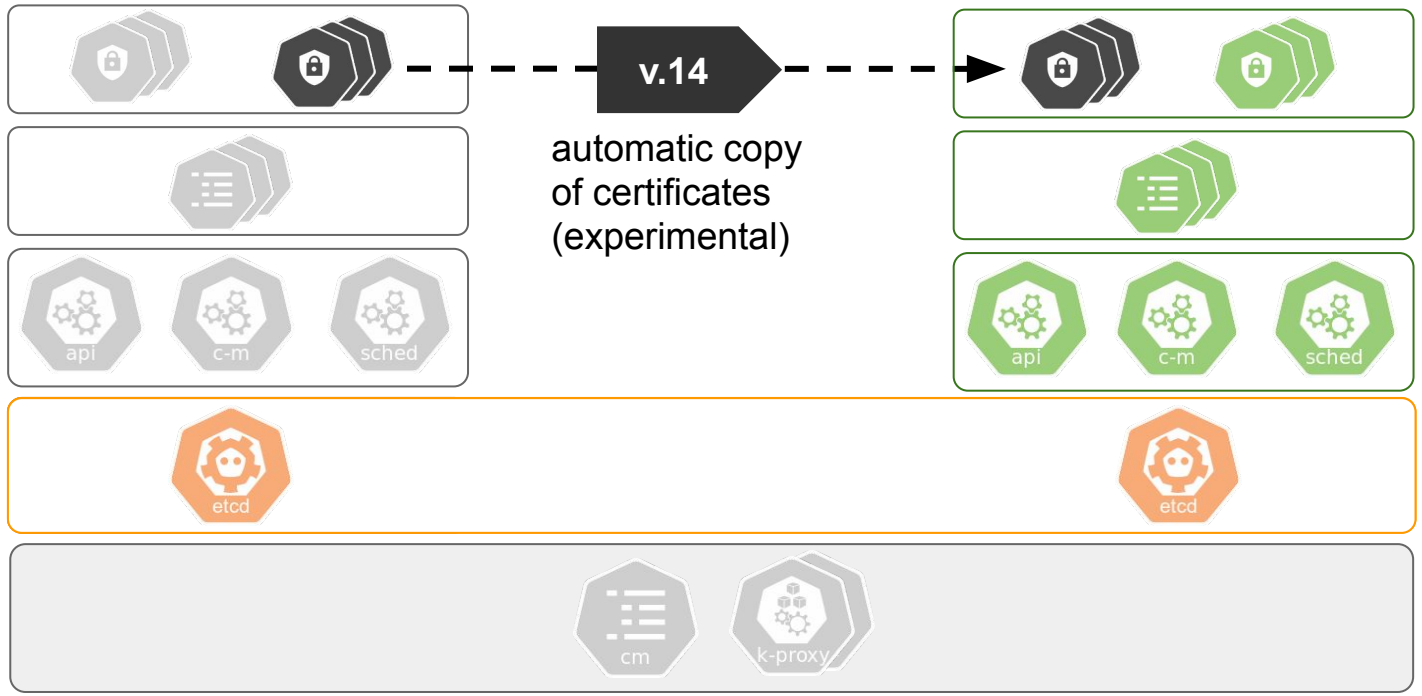
5 **Add the new etcd member to the existing etcd cluster**

Resources deployed in the K8s cluster at init time are already shared across all control plane nodes!

History of HA in kubeadm

Implementing HA took some time and an incremental approach…but finally we are at the end of it !

v1.11 Split cluster-wide configuration from node configurations



v1.12 Join control plane (with manual copy of certificates and only for cluster with external etcd)

v1.13 Join control plane with stacked etcd (with manual copy of certificates)

v1.15 **Test, tests, tests + (hopefully) craduate to beta!**

Coming soon… 2019 roadmap

Photo by Samuel Zeller on Unsplash

The kubeadm roadmap

- HA support in kubeadm to Beta!
- kubeadm config v1beta2 (small improvements)
- (Bring back) support for Windows nodes in kubeadm
- Consolidate story about certs management (external CA, renewal, cert location)
- Improve our CI signal, mainly for HA and upgrades
- Cleanup how K8s artifacts are built and installed
- **Evaluate usage of Kustomize for allowing advanced customization**
- …

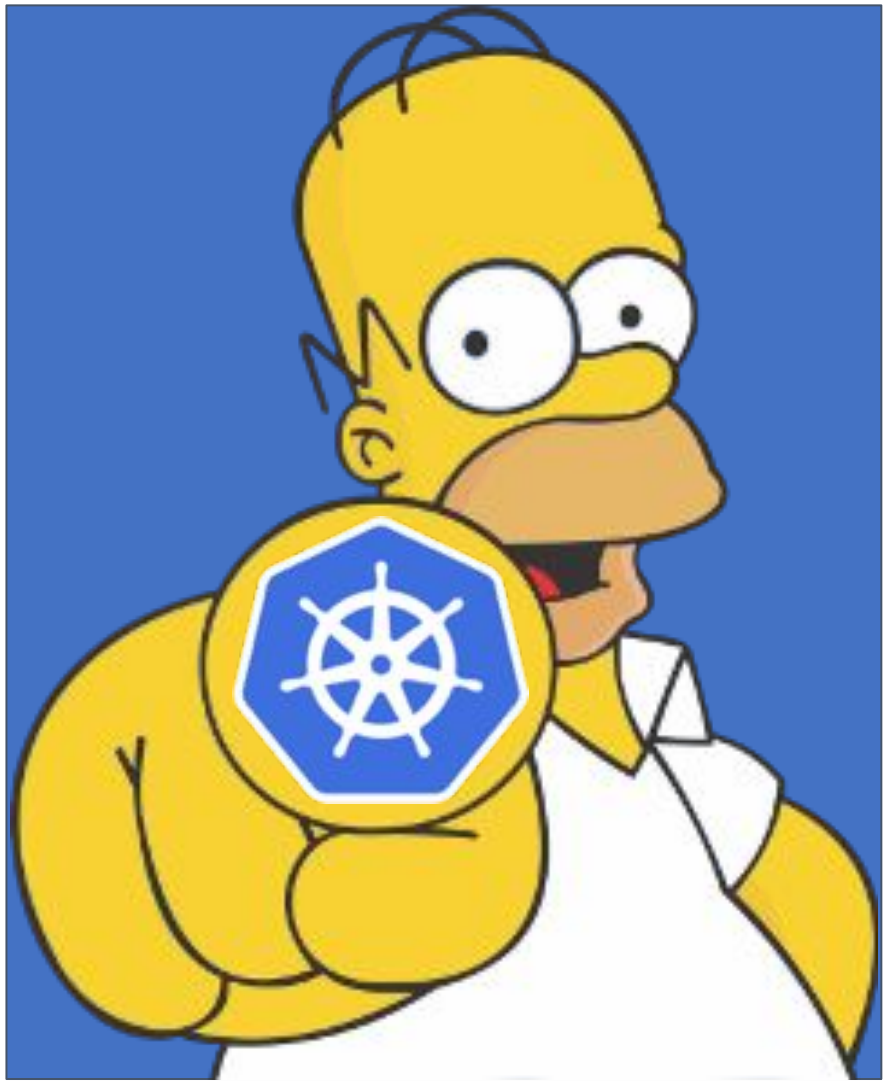
How can you Contribute

- Smaller core group of active maintainers
  - Tim, Lubomir, Ross, Jason, Liz, Chuck (VMWare)
  - Marek, Rafael (SUSE)
  - Alex, Ed (Intel)
  - Luxas, Fabrizio, Yago (Other/Independent)
- EU timezone friendly!
- Take a look at the SIG Cluster Lifecycle New Contributor Onboarding video
- Look for “good first issue”, “help wanted” and “sig/cluster-lifecycle” labeled issues in our repositories (in k/k or in various project repository)
- Join us on slack #kubeadm #sig-cluster-lifecycle
- We have “Kubeadm Office Hours” every week
- Contributing to SIG Cluster Lifecycle documentation

Logistics

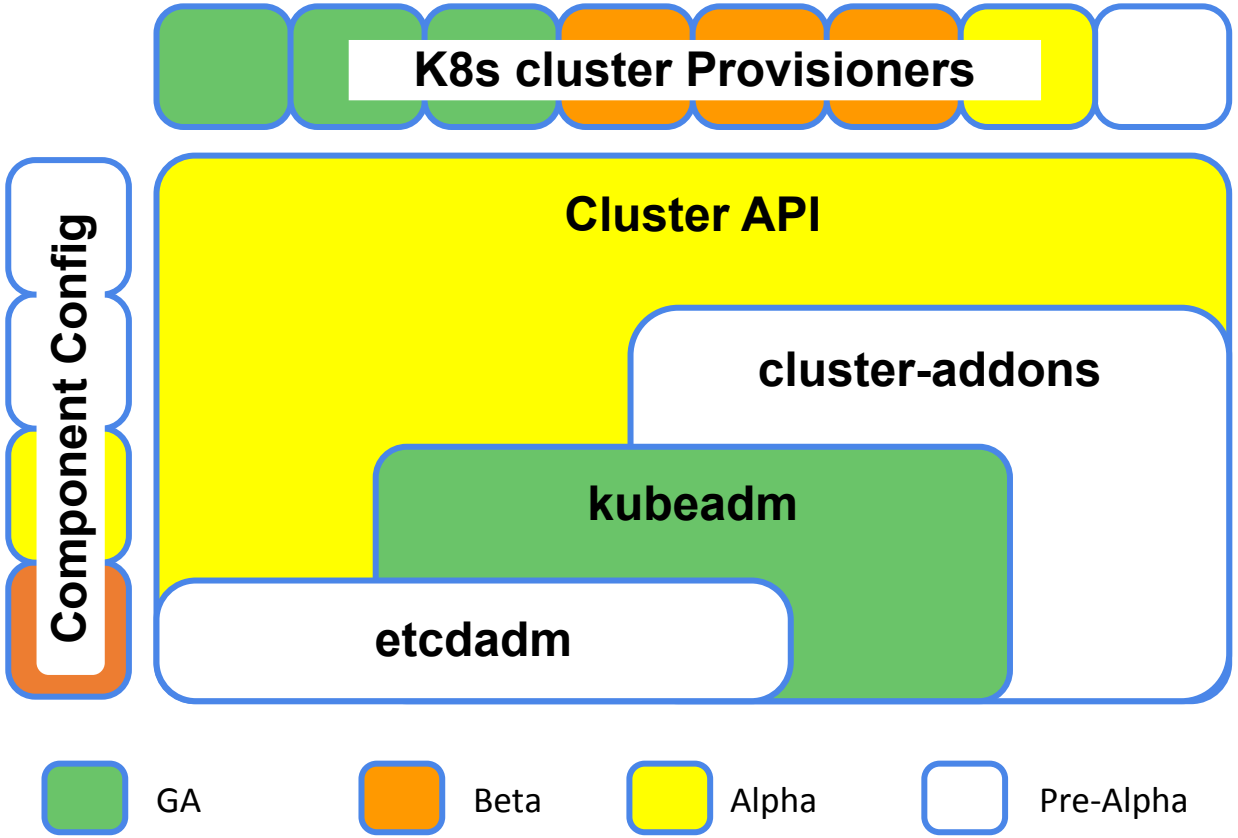
- Follow the SIG Cluster Lifecycle YouTube playlist
- Check out the meeting notes for our weekly office hours meetings
- Join #sig-cluster-lifecycle, #kubeadm channels
- Check out the kubeadm setup guide, reference doc and design doc
- Read how you can get involved and improve kubeadm!

SIG cluster lifecycle roadmap



We need your help!

There is still a lot of work to do in onder to get the full puzzle in place!



Questions and Answers

KubeCon CloudNativeCon Europe 2019

Photo by Nik Shuliahin on Unsplash