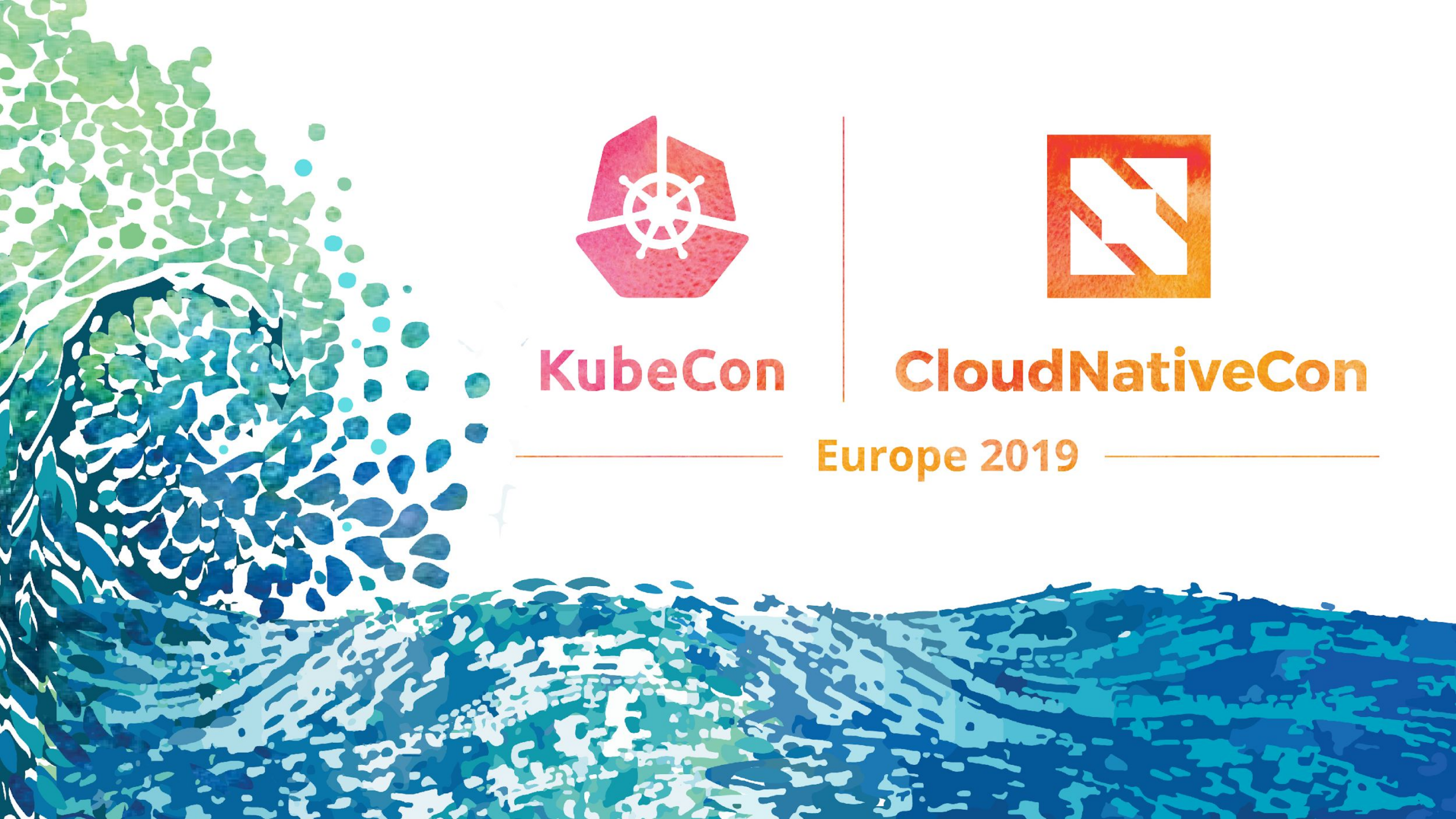


KubeCon | CloudNativeCon

Europe 2019

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"*

SIG Cluster Lifecycle

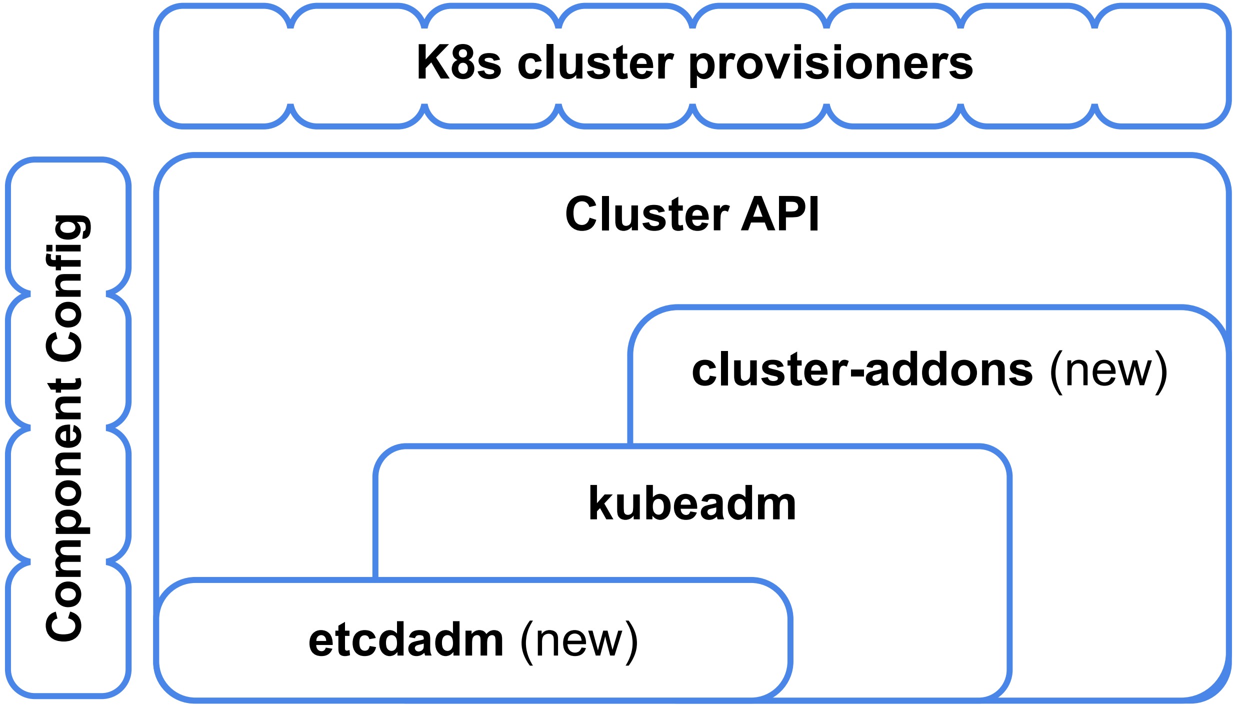
The kubeadm project is developed and maintained by **SIG Cluster Lifecycle.**

SIG Cluster Lifecycle’s objective is to simplify creation, configuration, upgrade, downgrade, and teardown of Kubernetes clusters and their components.

-- the SIG Cluster Lifecycle Charter

SCL Overview

SCL is one of the biggest kubernetes SIGs, with 100s of contributors across several companies actively contributing to 17 subprojects and several workgroups



K8s cluster provisioners:

- minikube
- kops
- kubespray
- kind (SIG Testing)
- kubeadm-dind-cluster
- Cluster-api-provider-<name>
- ...

Kubeadm: Key Design Takeways

- kubeadm’s task is to set up a **best-practice cluster**
- 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
- Composable architecture with everything divided into phases
- Versioned configuration file

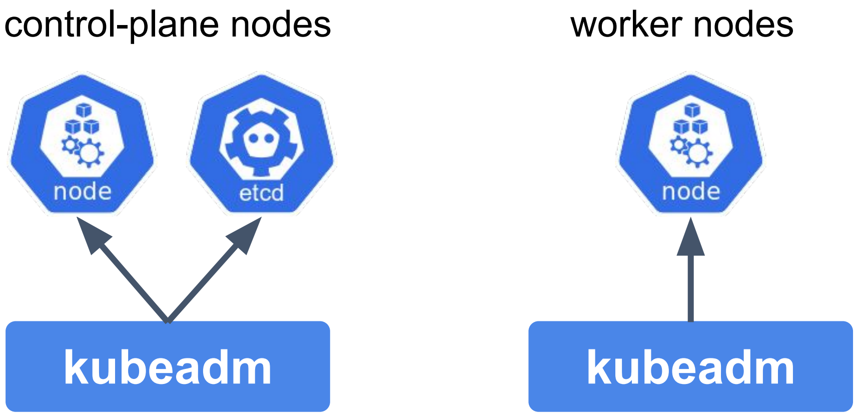
I HOPE THAT CLARIFIES OUR STRATEGY.
QUESTIONS?
Dilbert.com DilbertCartoonist@gmail.com

The Unix Philosophy & kubeadm

- Someone or something should provide the machines

- **kubeadm creates a Kubernetes node on the machine**

“Make each program do one thing well”



- Someone or something should install the CNI plugin

Recent Changes in kubeadm

Photo by Taylor Grote on Unsplash

Kubeadm is GA!

What Does really mean GA?

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`**

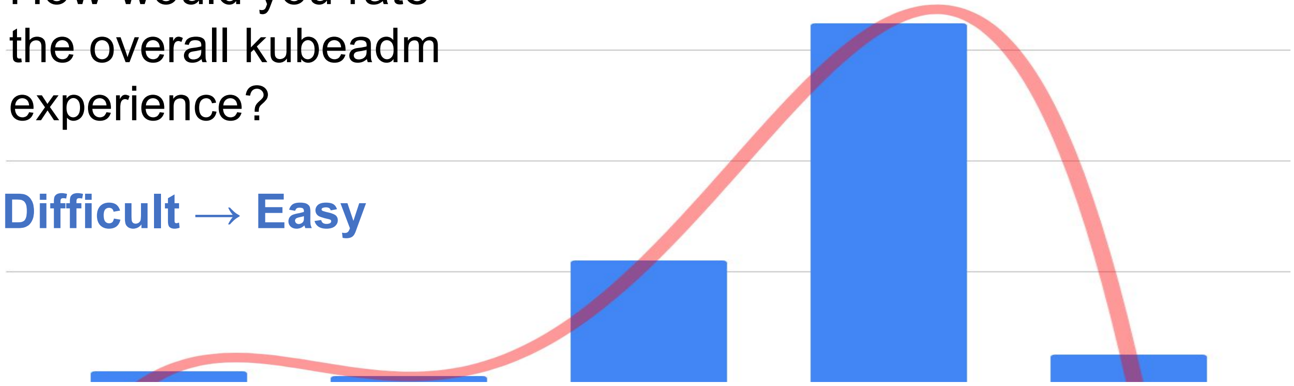
```
      preflight
      kubelet-start
      certs
        /...
      kubeconfig
        /...
      control-plane
        /...
      etcd
      upload-config
        /..
v.14  upload-certs [EXPERIMENTAL]
      mark-control-plane
      bootstrap-token
      addon
        /...
```

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
```

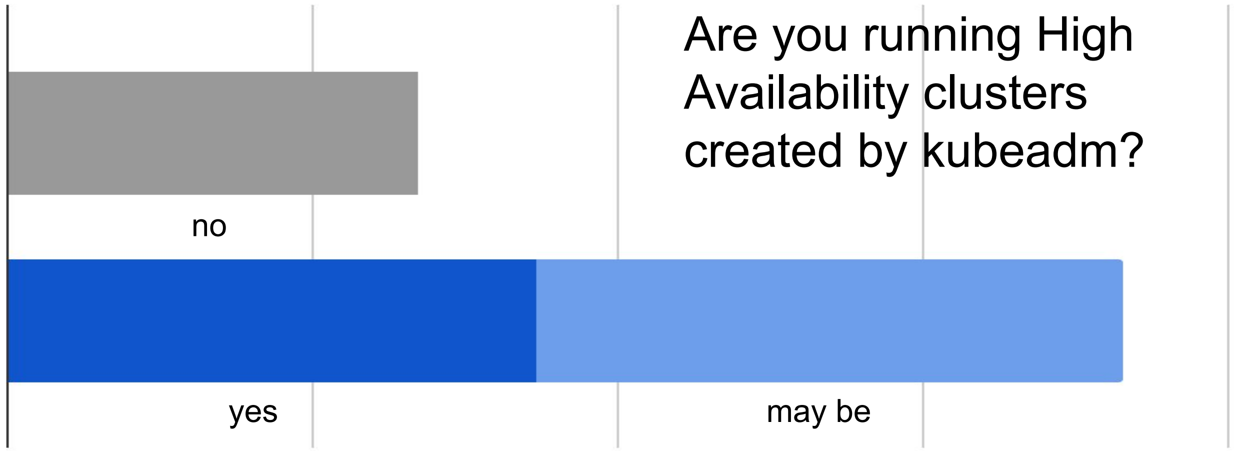
Kubeadm Survey

thank you



How would you rate the overall kubeadm experience?

Difficult → Easy



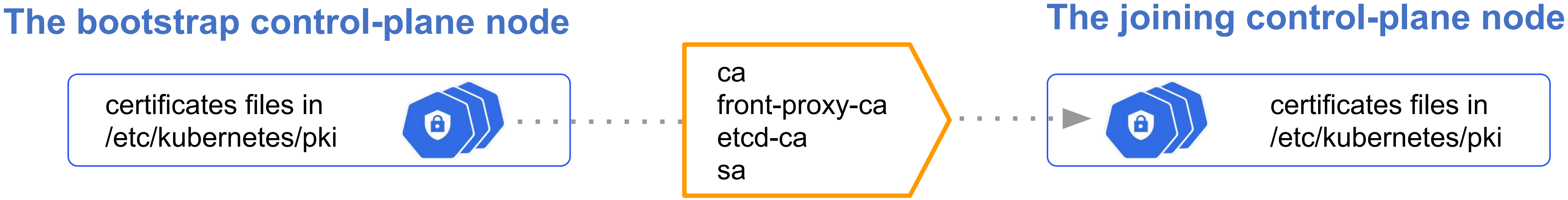
Are you running High Availability clusters created by kubeadm?

Focus: Automatic Certificates Copy

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



The bootstrap control-plane node

certificates files in
/etc/kubernetes/pki

ca
front-proxy-ca
etcd-ca
sa

The joining control-plane node

certificates files in
/etc/kubernetes/pki

Why you should care about kubeadm Automatic Certificate Copy?

- It simplify administrators life (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

At init time, pass **--experimental-upload-certs** to instruct kubeadm to prepare for certificate copy

certificates files are created in /etc/kubernetes/pki (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

1

```
kubeadm init --experimental-upload-certs
```

2

```
...
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[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
...
```

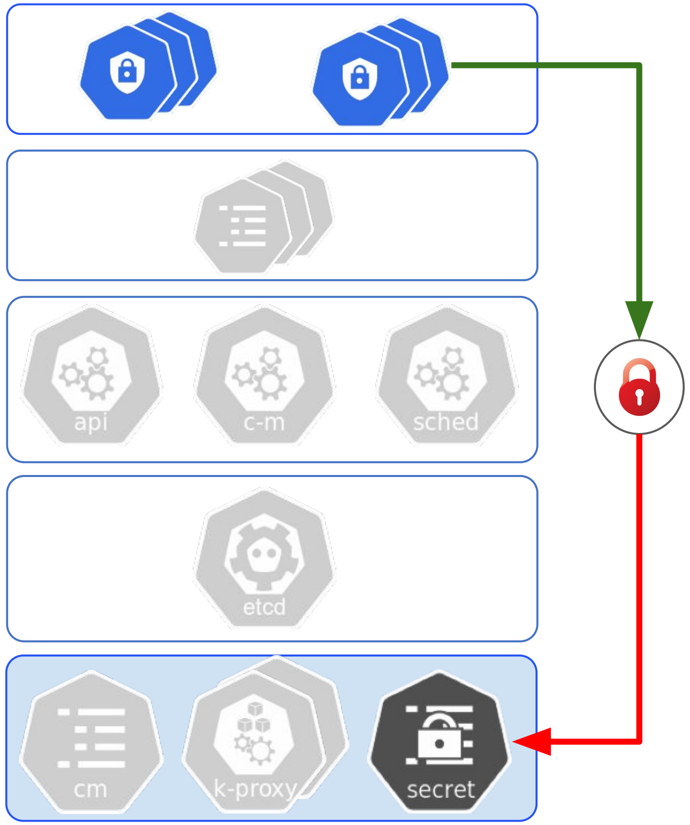
3

```
[upload-certs] storing the certificates in ConfigMap
               "kubeadm-certs" in the "kube-system"
               Namespace
...
...
Your Kubernetes control-plane has initialized successfully!
...
```

4

```
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 \
    --certificate-key 01234567890123456789012345....
```



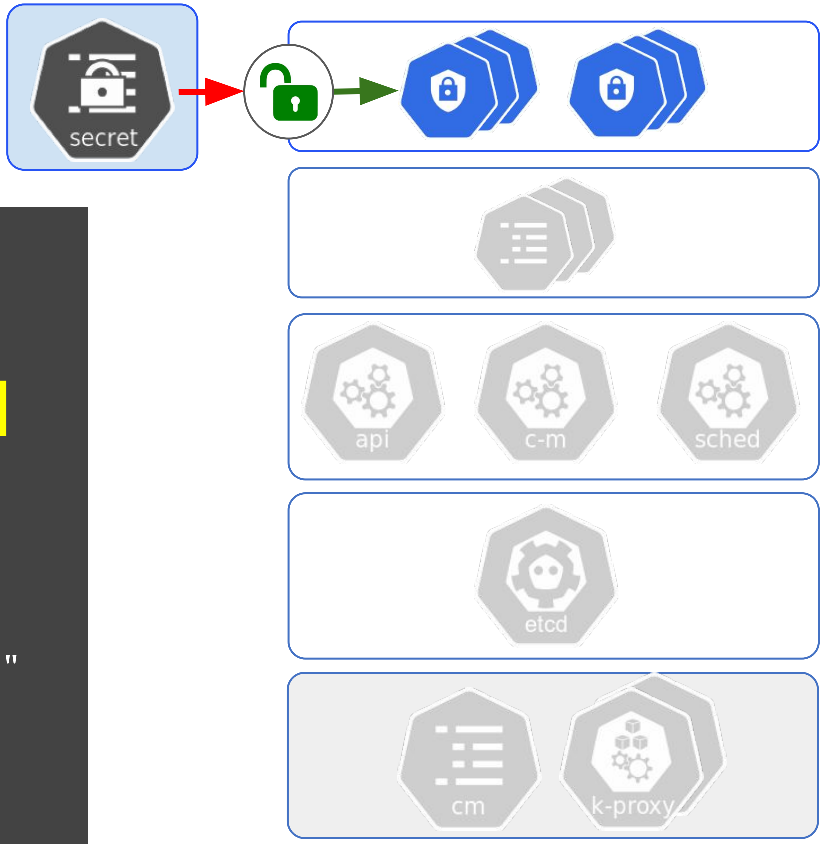
How it works @ join time

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

```
 kubeadm join 172.17.0.4:6443 --token abcdef...\
   --discovery-token-ca-cert-hash sha256:... \
   --experimental-control-plane \
   --certificate-key 01234567890123456789012345....   1
...
[preflight] Reading configuration from the cluster
...
2 [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!
```



Key takeaways!

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

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

ℹ 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`

ℹ *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

Focus: The Dynamic Workflow

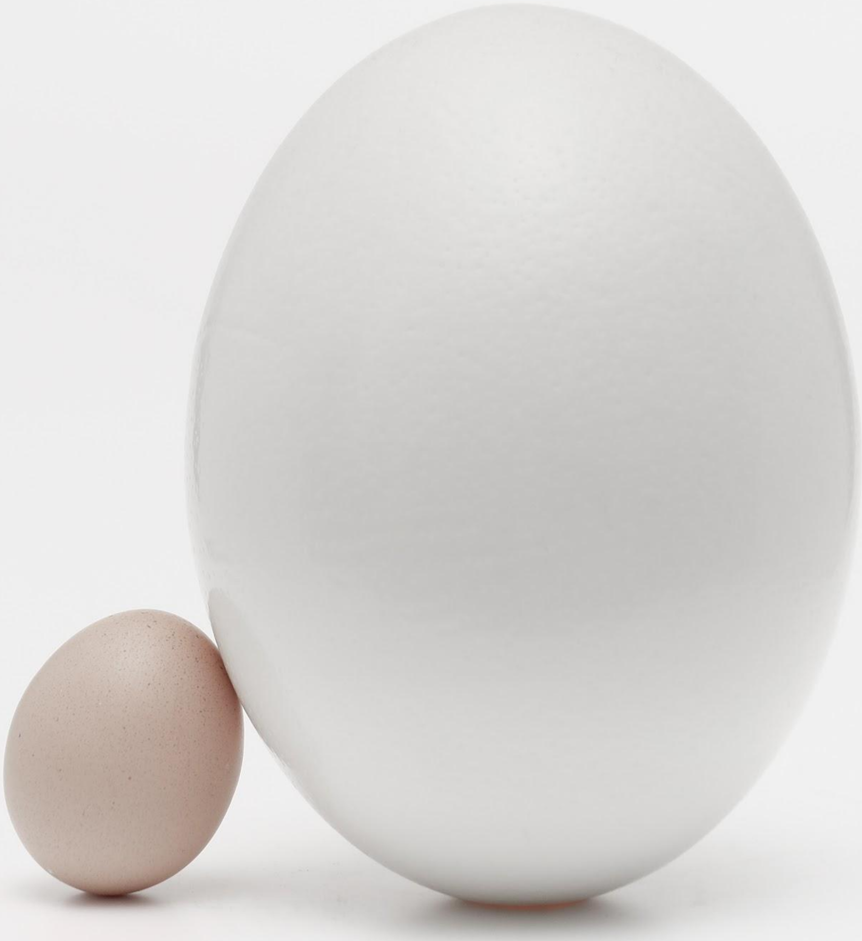
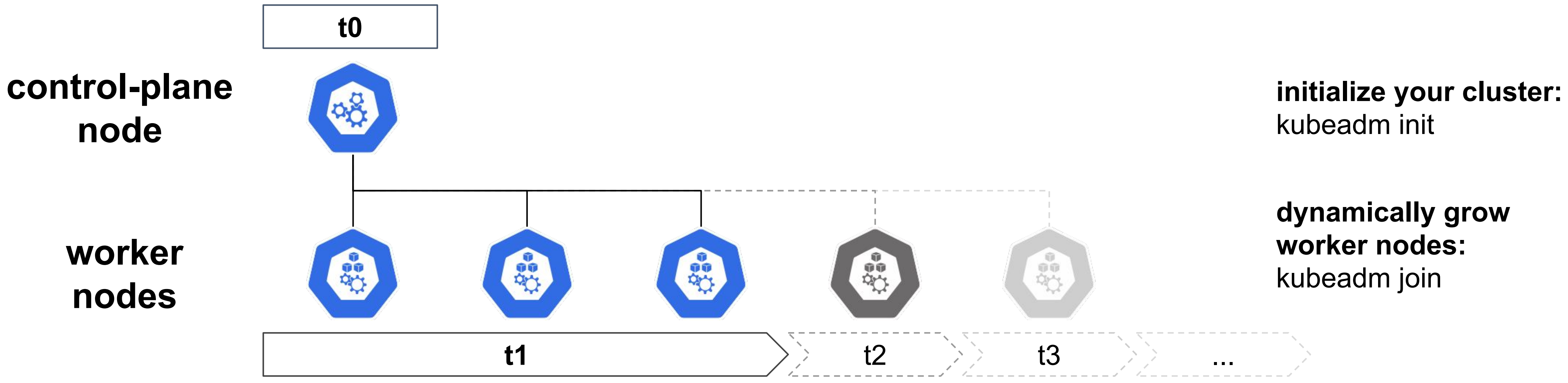


Photo by Daniele Levis Pelusi on Unsplash

Dynamic Workflow in a nutshell

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



t0

control-plane node

worker nodes

t1

t2

t3

...

initialize your cluster:
kubeadm init

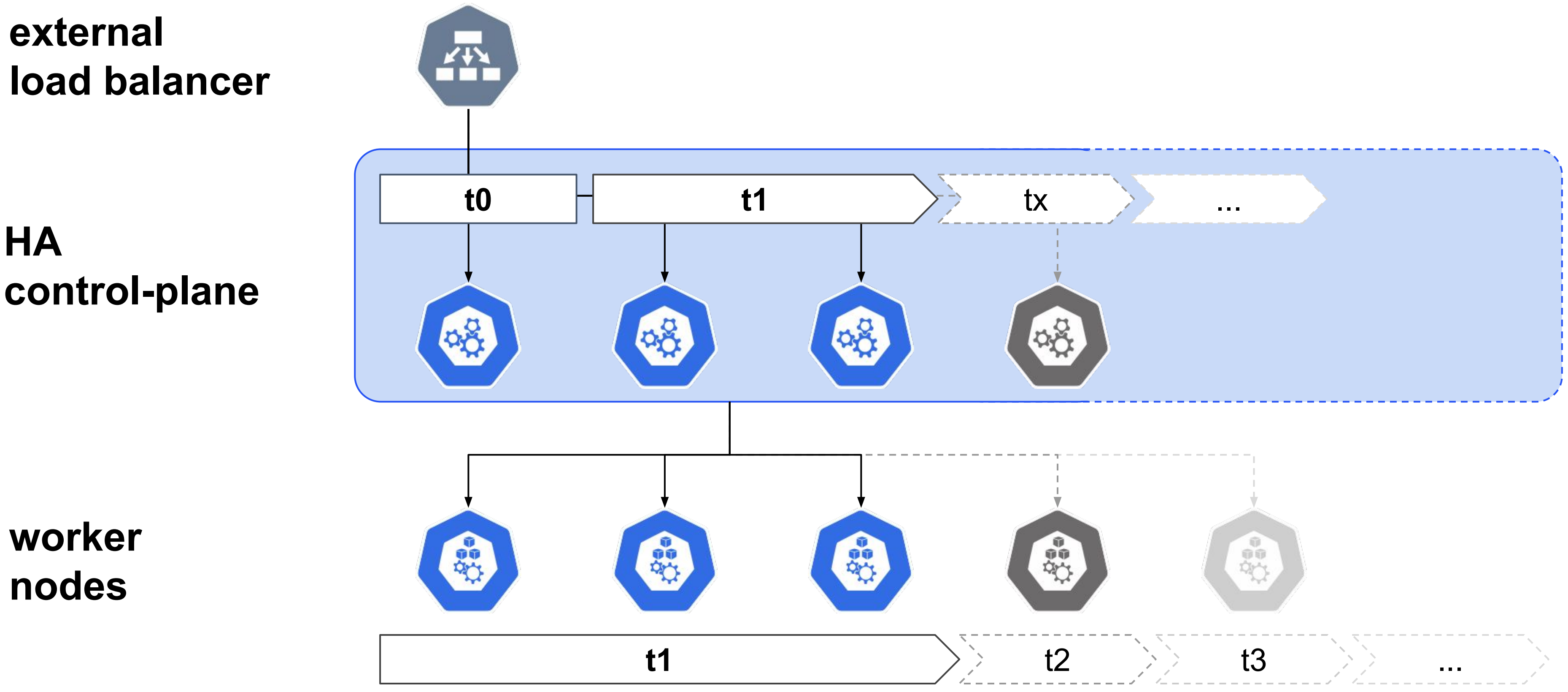
dynamically grow worker nodes:
kubeadm join

Why you should care about kubeadm Dynamic Workflow?

- It simplify cluster lifecycle (grow the number of nodes, replace nodes)
- Because HA is implemented by dynamically growing the control-plane nodes, and requires some special considerations

The external load balancer

In order to dynamically grow the control-plane nodes you need an **external load balancer** and a stable control-plane address. Use kubeadm join --experimental-control-plane to add control-plane nodes



initialize your cluster:
kubeadm init

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

dynamically grow worker nodes:
kubeadm join

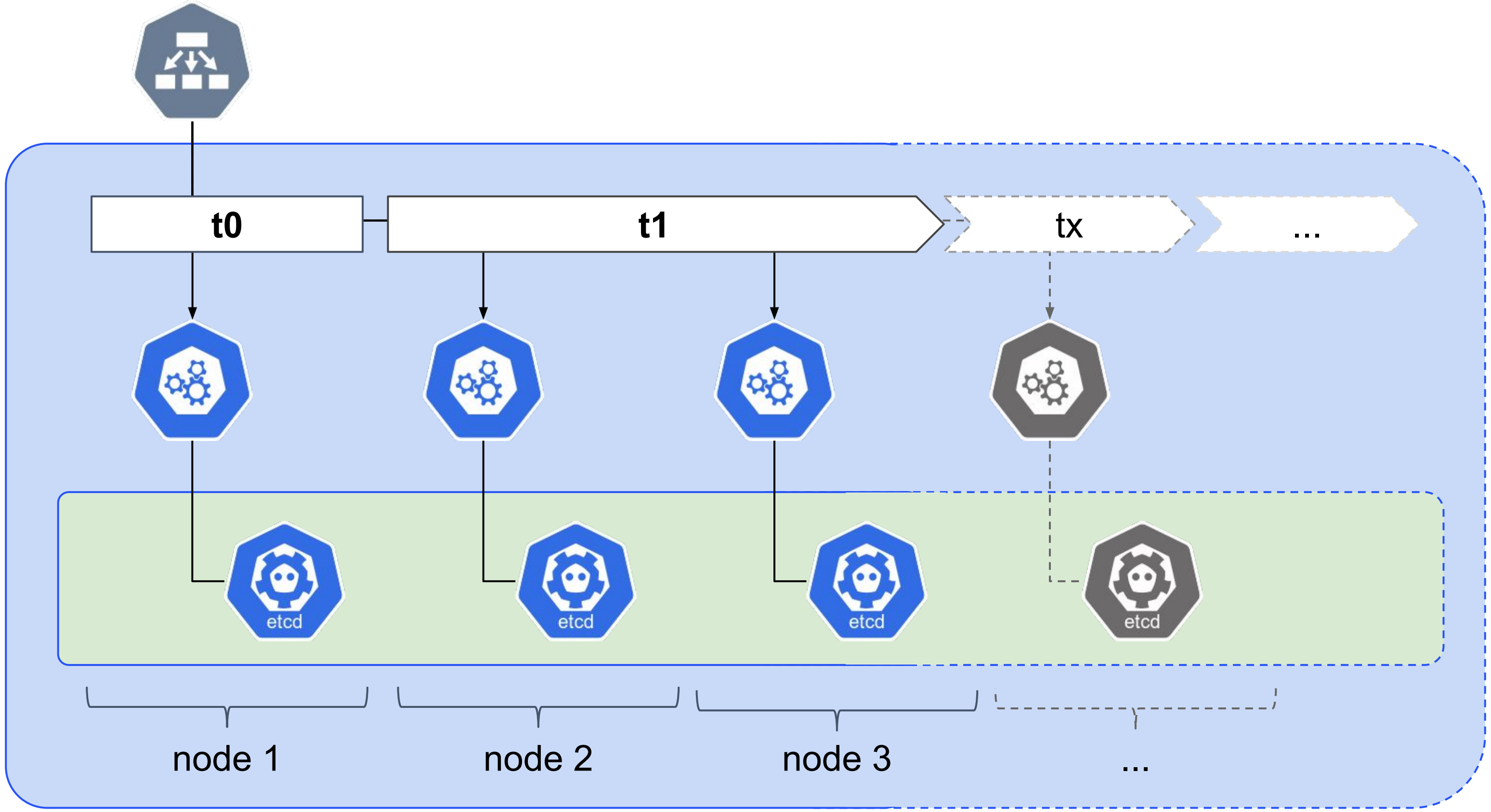
Stacked etcd

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

external load balancer

HA control-plane

The stacked etcd cluster



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

Key takeaways!

In order to create an HA cluster you need an **external load balancer** and a stable IP address
Then use kubeadm join --experimental-control-plane to dynamically grow control-plane nodes

In case you are not providing an external etcd, **a stacked etcd cluster is automatically generated**

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

⚠ Each apiserver 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.

ℹ The stacked etcd cluster is subject to the usual etcd operational considerations
e.g. quorum

⚠ If you override defaults for kube-apiserver or for etcd using the ClusterConfiguration extraArg config object, you will override settings on all nodes.

Bonus pack

Photo by Math on Unsplash

The Starting Point

Creating a control-plane node with kubeadm => create certificates, kubeconfig files, manifests, etc.

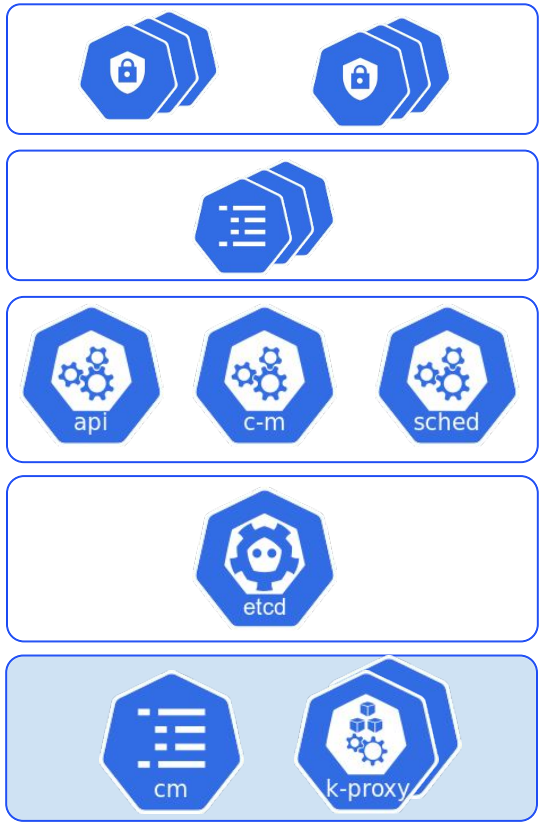
1 certificates files in /etc/kubernetes/pki

2 kubeconfig files in /etc/kubernetes

3 static pod manifests in /etc/kubernetes/manifest

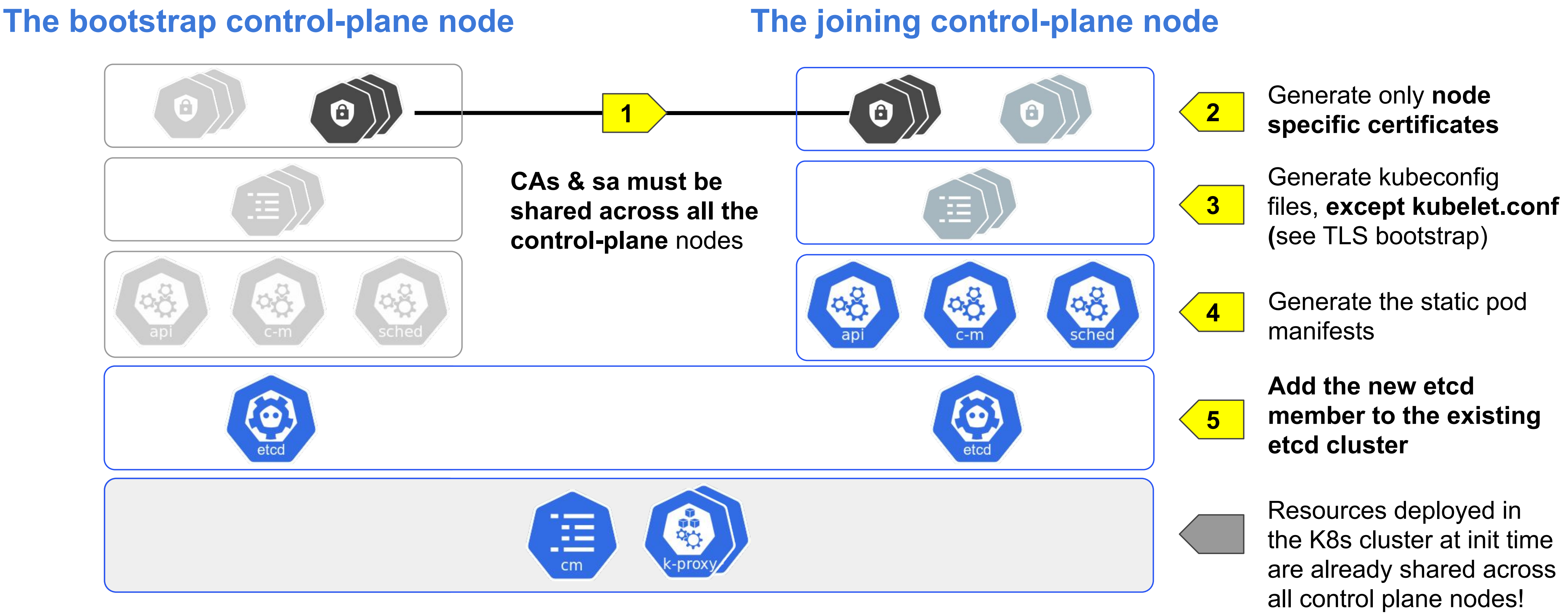
4 kubeadm ConfigMap + core addons + RBAC rules, bootstrap-tokens **are deployed in the K8s cluster**

```
$ 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!
```



The Grand Theory of HA in kubeadm

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


The bootstrap control-plane node
The joining control-plane node
1
api
c-m
sched
api
c-m
sched
etcd
etcd
cm
k-proxy
2
3
4
5

CAs & sa must be shared across all the control-plane nodes

2 Generate only **node specific certificates**

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

4 Generate the static pod manifests

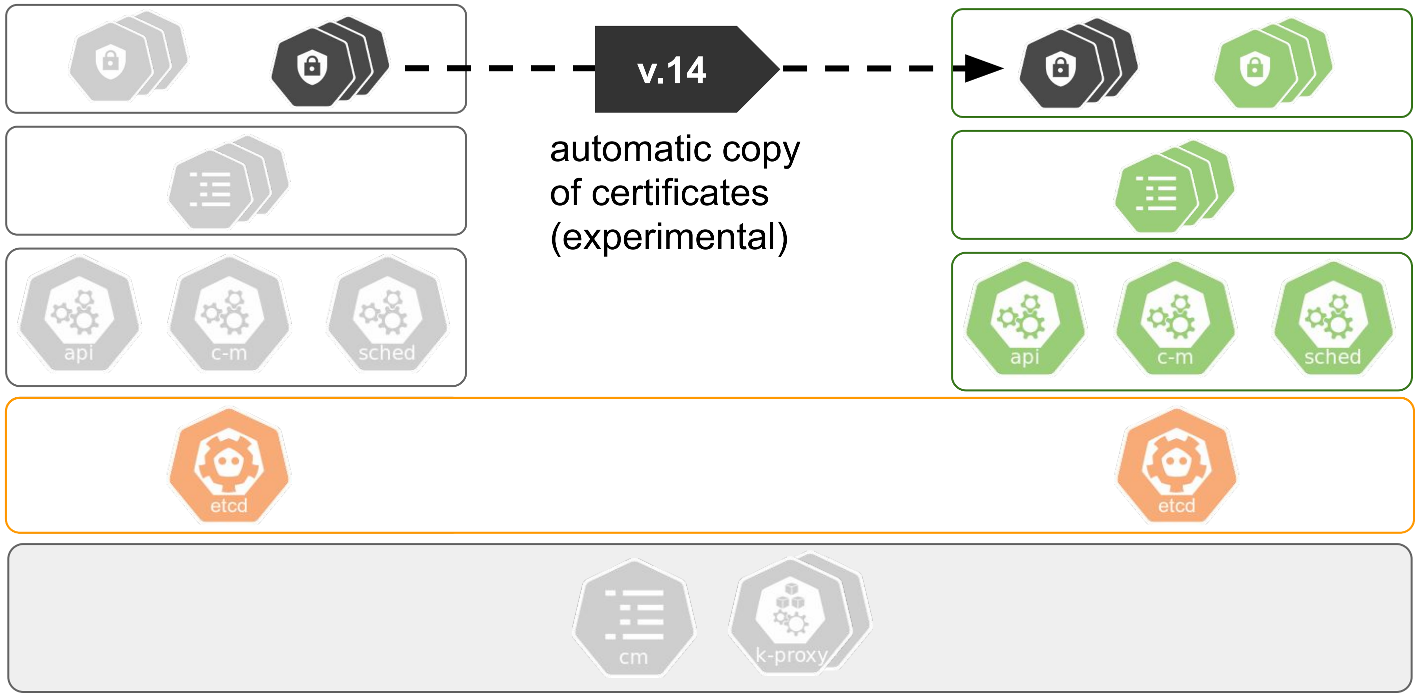
5 **Add the new etcd member to the existing etcd cluster**

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 !

v.11 Split ClusterConfiguration from Init/JoinConfiguration



automatic copy of certificates (experimental)

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

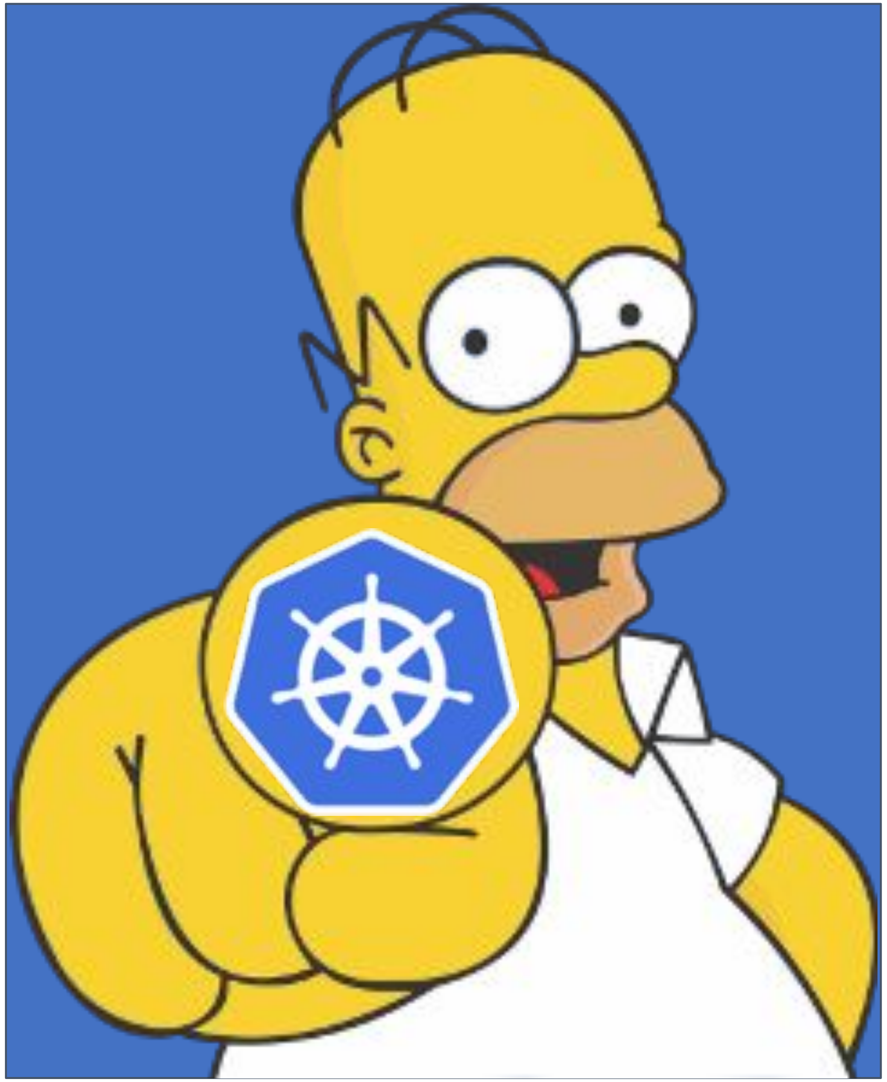
v.13 Join control plane with stacked etcd (with manual copy of certificates)

v.15 **Graduate to beta!**

Coming Soon… 2019 Roadmap

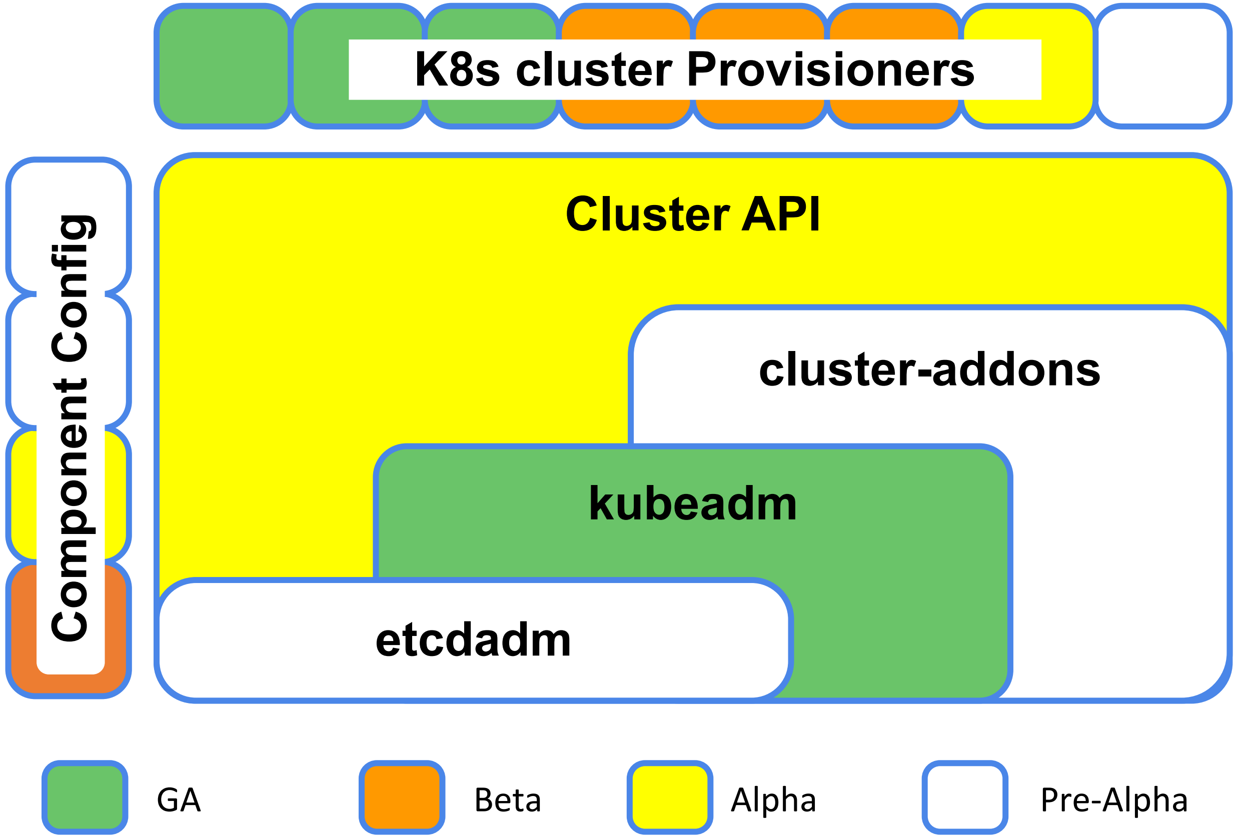
Photo by Samuel Zeller on Unsplash

The SCL Roadmap



We need your help!

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



K8s cluster Provisioners

Component Config

Cluster API

cluster-addons

kubeadm

etcdadm

🟩 GA

🟧 Beta

🟨 Alpha

⬜ Pre-Alpha

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**
- …

Getting Involved!

SIG Cluster Lifecycle

- 100s of contributors across several companies
- We’re working on growing the contributor/reviewers pool
- We’re EMEA contributors friendly

The “kubeadm” team

- Smaller core group of active maintainers
  - Tim, Lubomir, Ross, Jason, Liz, Chuck (VMWare)
  - Marek, Rafael (SUSE)
  - Alex, Ed (Intel)
  - Luxas, Fabrizio, Yago (Other/Independent)
- Large user community on #kubeadm

How can you Contribute

- SIG Cluster Lifecycle New Contributor Onboarding
- Look for “good first issue”, “help wanted” and “sig/cluster-lifecycle” labeled issues in our repositories (in k/k or in various project repository)
- Attend our Zoom meetings / be around on Slack
- We have “Office Hours” for our projects: weekly for kubeadm and Cluster API, bi-weekly for kops and kubespray
- Full list of SIG meetings and links to minutes and recordings can be found on SIG page
- 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!

Questions and Answers

Thank You!

Q & A

Photo by Nik Shuliahin on Unsplash