SEP 09 – 11, 2025

CONTAINER days CONFERENCE

Day-2’261 Migration

kubeadm+Ansible to ClusterAPI+Talos
A Swiss Bank’s Journey

Clément Nussbaumer

clement.n8r.ch

Longevity of a K8s Cluster

How old can your clusters get?

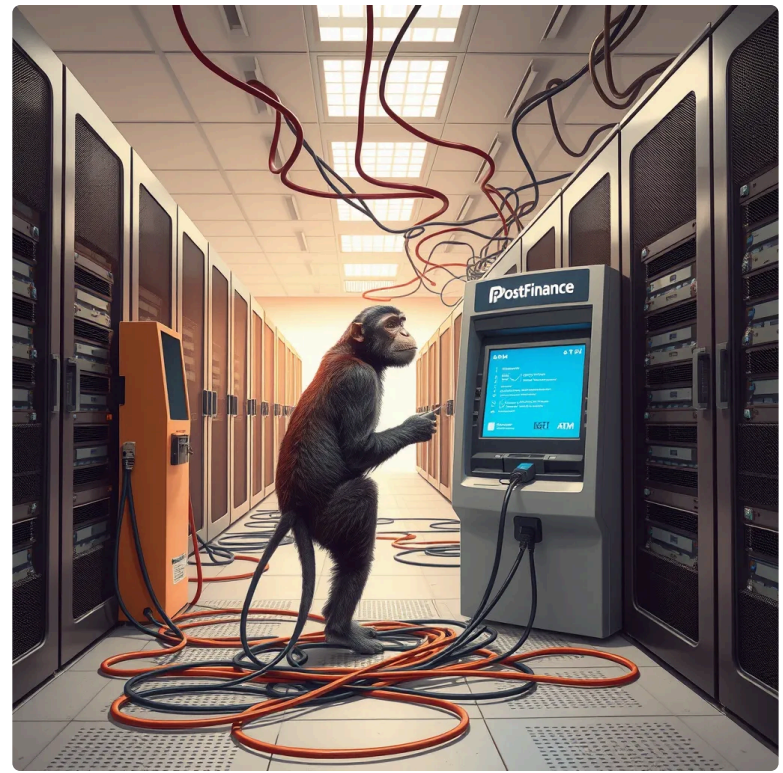
```
$ kubectl get namespace kube-system
NAME          STATUS   AGE
kube-system   Active   6y70d
```

```
     6 years (365.25 days)
 +  70 days
= 2261 days
```

Kubernetes @ PostFinance

Some context

- 35 Vanilla v1.32 Kubernetes, 450 nodes
- Oldest cluster: `AGE = 6y70d`
- 2 on-prem datacenters - VSphere Virtualization
- Chaos Monkey on all clusters 🐒
- Namespace self-service thanks to an in-house operator
- ArgoCD and GitOps for almost all deployments

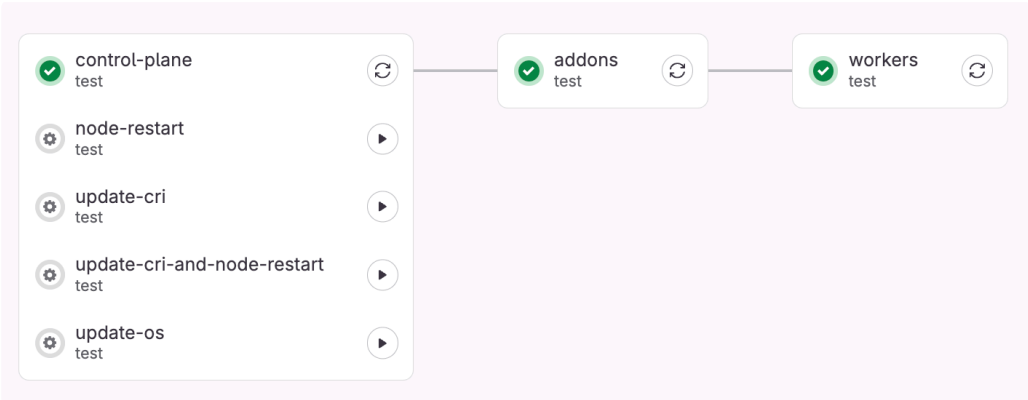


Created with FLUX.1 [dev] by Black Forest Labs

Starting Point

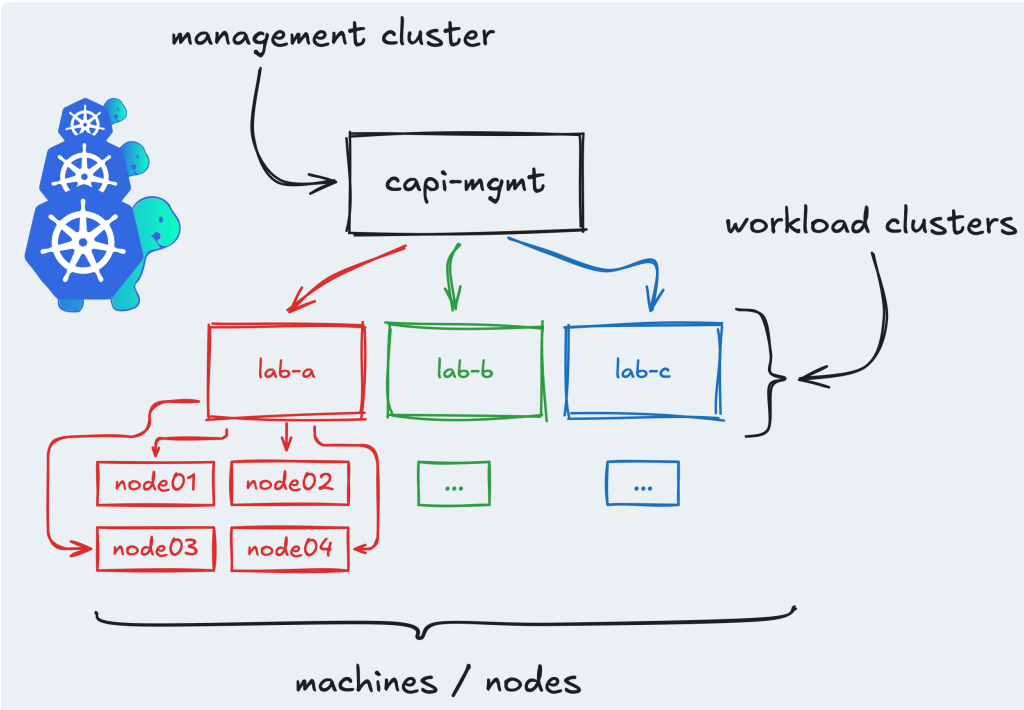
Legacy cluster provisioning

1. provision Debian VMs (terraform)
2. deploy/configure base tools
3. register nodes in `inventory.yml`
4. run Ansible playbook:
   - render config files, base manifests, …
   - run `kubeadm` commands
5. configure ArgoCD integration



Current cluster provisioning pipeline overview

ClusterAPI




```
$ kubectl get clusters --all-namespaces
NAMESPACE    NAME           AGE
capi-lab-a   e1-k8s-lab-a   29d
capi-lab-b   e1-k8s-lab-b   95d
capi-lab-c   e1-k8s-lab-c   95d
```

```
$ kubectl get machinedeployments
NAME                   REPLICAS AGE    VERSION
e1-k8s-lab-a-md-002    2        5d5h   v1.30.5
e1-k8s-lab-a-md-1      1        19d    v1.31.4
```

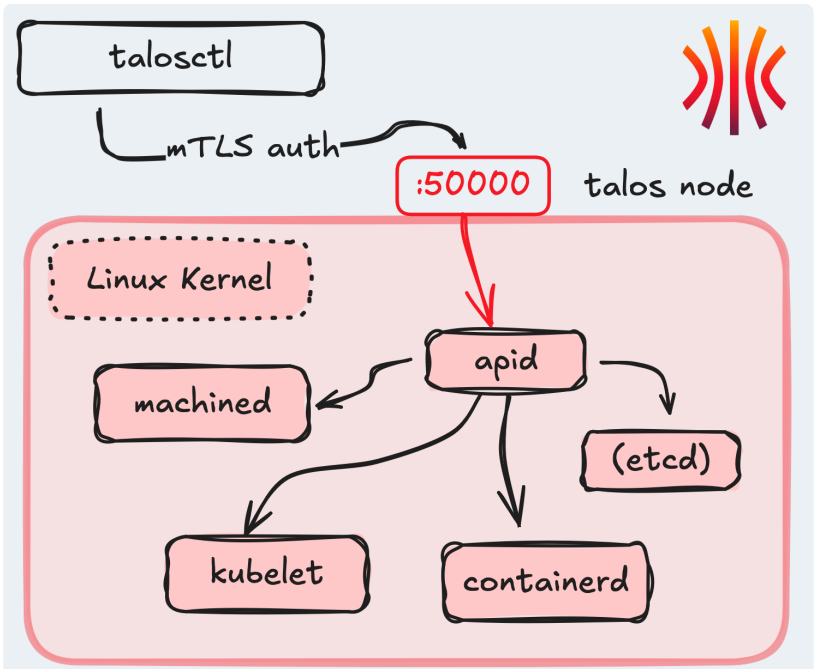
```
$ kubectl get machines
NAME                   PHASE      AGE    VERSION
lab-a-md-002-lr7ch     Running    5d4h   v1.30.5
lab-a-md-002-lwx4t     Running    4d2h   v1.30.5
lab-a-md-1-4rtb9       Running    19d    v1.31.5
```

Talos Linux

The 12 binaries' O.S.

> immutable, minimal, ephemeral
> declarative configuration file and gRPC API [1]

```
$ talosctl services
SERVICE                 STATE     HEALTH
apid                    Running   OK
containerd              Running   OK
cri                     Running   OK
etcd                    Running   OK
kubelet                 Running   OK
machined                Running   OK
syslogd                 Running   OK
trustd                  Running   OK
udevd                   Running   OK
```



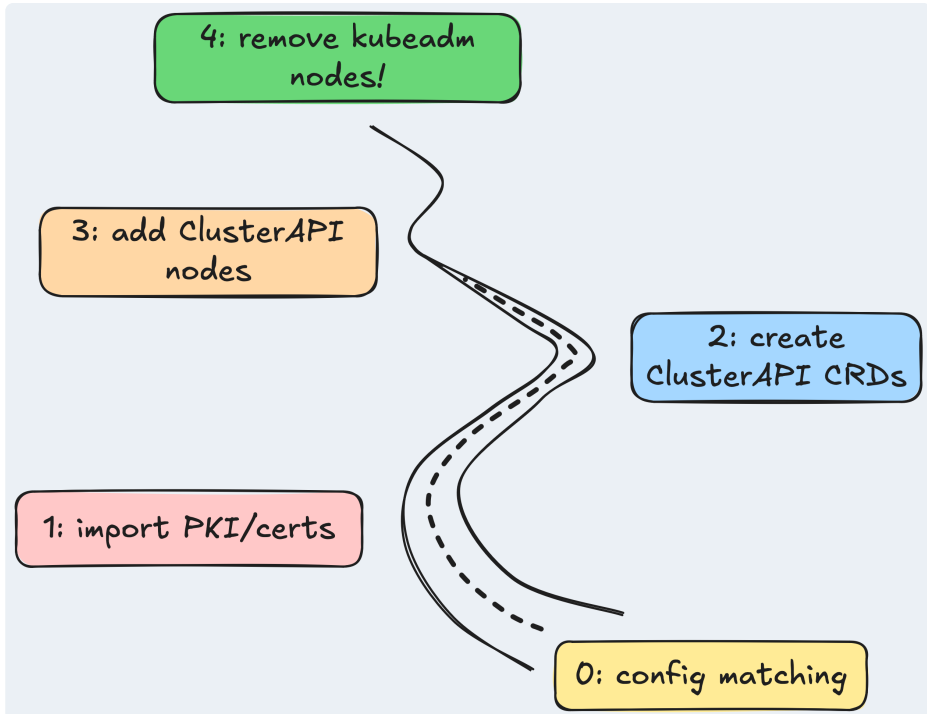
1. https://www.talos.dev/v1.11/learn-more/philosophy/ ↩

Migration

The Journey

Documentation:

- Talos `kubeadm` migration guide
- Talos `v1alpha1/config` reference
- `kube-apiserver` CLI reference
- Kubernetes `etcd` encryption doc



step 0: config matching

Kubeadm modifications:

1. `--service-account-issuer =>`

   `https://<apiserver endpoint>:6443`
2. match `etcd` encryption key names to those hard-coded in Talos template file
3. re-encrypt all secrets:

```
kubectl get secrets --all-namespaces -o json | \
  kubectl replace -f -
```

```
--- # talos EncryptionConfig template
apiVersion: v1
kind: EncryptionConfig
resources:
- resources:
  - secrets
  providers:
  {{if .Root.SecretboxEncryptionSecret}}
  - secretbox:
      keys:
      - name: key2
        secret: {{ .Root.SecretboxEncryptionSecret }}
  {{end}}
  {{if .Root.AESCBCEncryptionSecret}}
  - aescbc:
      keys:
      - name: key1
        secret: {{ .Root.AESCBCEncryptionSecret }}
  {{end}}
  - identity: {}
```

step 1: import existing PKI

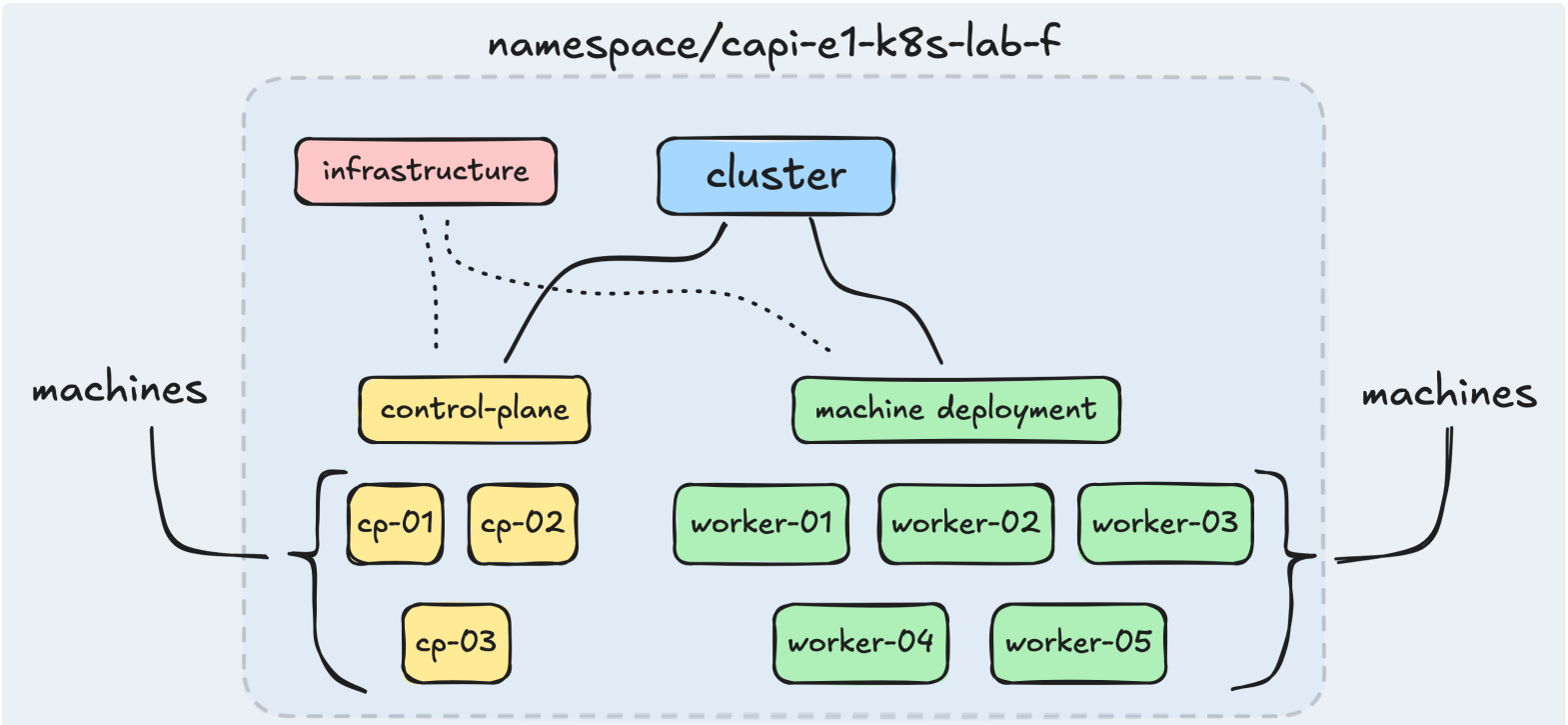
```yaml
--- # secretsbundle.yaml
secrets:
  bootstraptoken: c00ffe.0123456789abcdef
  secretboxencryptionsecret: base64-encoded-etcd-encryption-key
certs:
  etcd:
    crt: base64-encoded-crt
    key: base64-encoded-key
  k8s:
    crt: base64-encoded-crt
    key: base64-encoded-key
  k8sserviceaccount:
    key: base64-encoded-key
  os:
    crt: base64-encoded-crt
    key: base64-encoded-key
```

step 2: create ClusterAPI CRDs

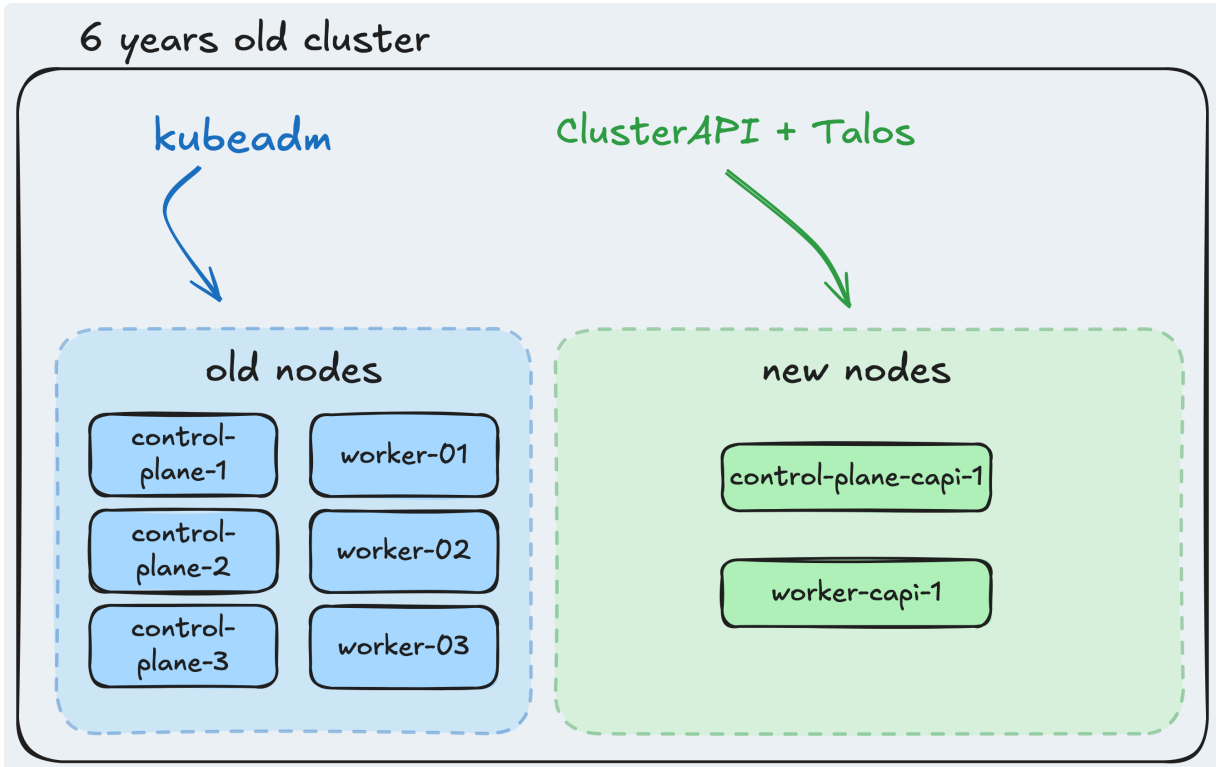
```yaml
---
apiVersion: cluster.x-k8s.io/v1beta1
kind: Cluster
metadata:
  labels:
    cluster.x-k8s.io/cluster-name: e1-k8s-lab-f
  name: e1-k8s-lab-f
spec:
  controlPlaneEndpoint:
    host: e1-k8s-lab-f-internal.tld.ch
    port: 443
  controlPlaneRef:
    apiVersion: controlplane.cluster.x-k8s.io/v1alpha3
    kind: TalosControlPlane
    name: e1-k8s-lab-f
  infrastructureRef:
    apiVersion: infrastructure.cluster.x-k8s.io/v1beta1
    kind: VSphereCluster
    name: e1-k8s-lab-f
```

```yaml
---
apiVersion: controlplane.cluster.x-k8s.io/v1alpha3
kind: TalosControlPlane
metadata:
  name: e1-k8s-lab-f
spec:
  controlPlaneConfig:
    controlplane:
      strategicPatches: [...]
      generateType: controlplane
  infrastructureTemplate:
    apiVersion: infrastructure.cluster.x-k8s.io/v1beta1
    kind: VSphereMachineTemplate
    name: control-plane-v1.9.4-pf.0
  replicas: 0
  rolloutStrategy:
    rollingUpdate:
      maxSurge: 1
    type: RollingUpdate
  version: v1.31.5
```

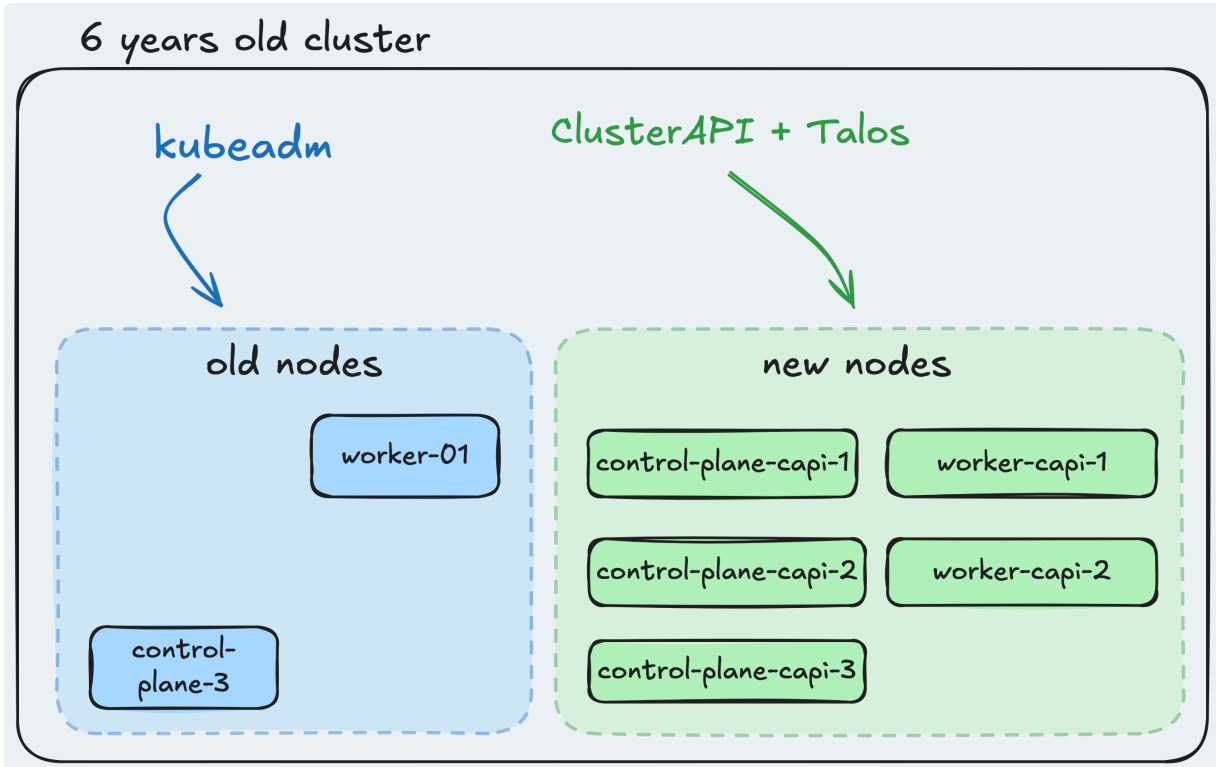
step 2: create ClusterAPI CRDs



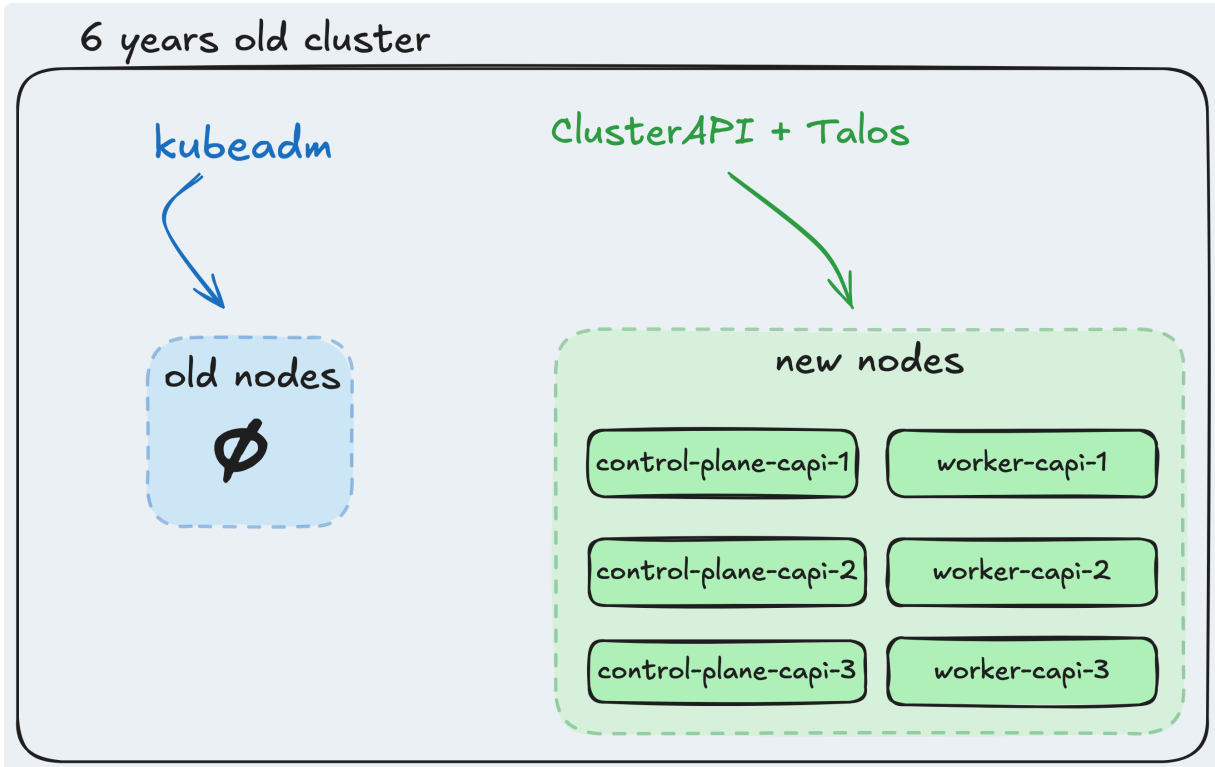
step 3: add ClusterAPI nodes



step 3: add ClusterAPI nodes



step 3: add ClusterAPI nodes



step 3: create ClusterAPI nodes

What could go wrong after all?

- mismatched `--service-account-issuer` config:

```
[authentication.go:73] "Unable to authenticate the request" err="invalid bearer token"
```

- `etcd` encryption key missing:

```
[reflector.go:561] storage/cacher.go:/secrets: failed to list *core.Secret: unable
to transform key "/registry/secrets/appl-titi/toto-secret": no matching prefix found
```

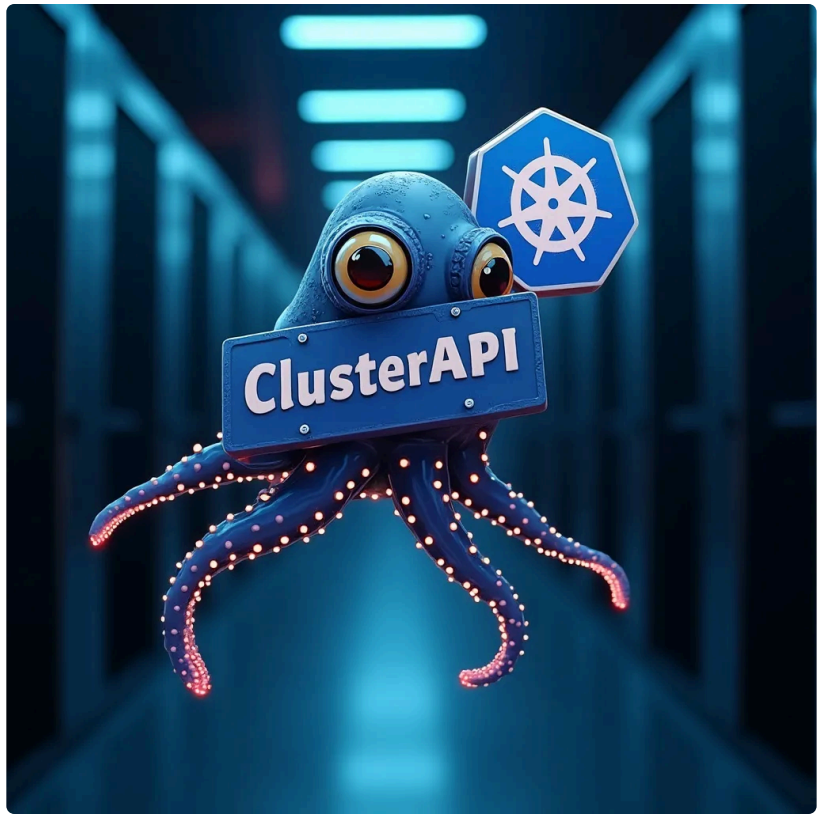
- wrong `etcd` encryption key

```
[transformer.go:163] "failed to decrypt data" err="output array was not large enough for encryption"
```

- loss of quorum: use the `--force-new-cluster` flag to recover one node

Demo

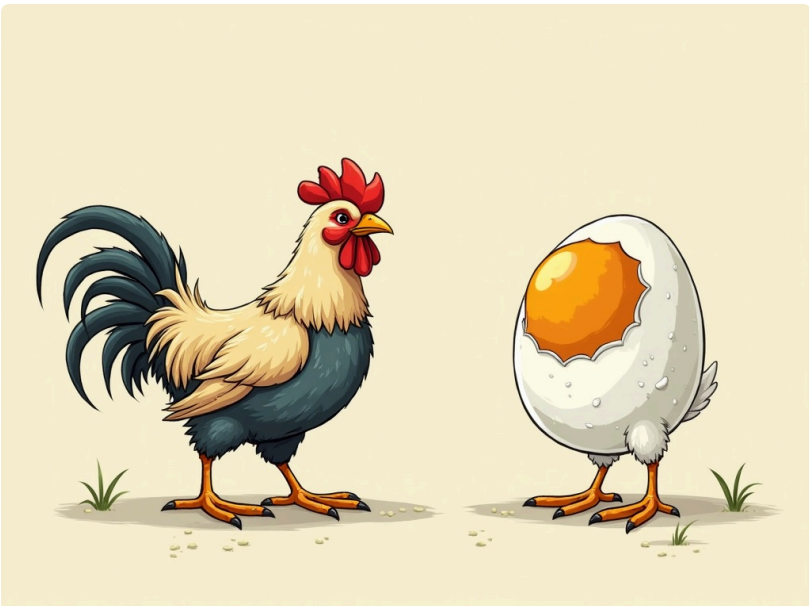
Live migration 🚀



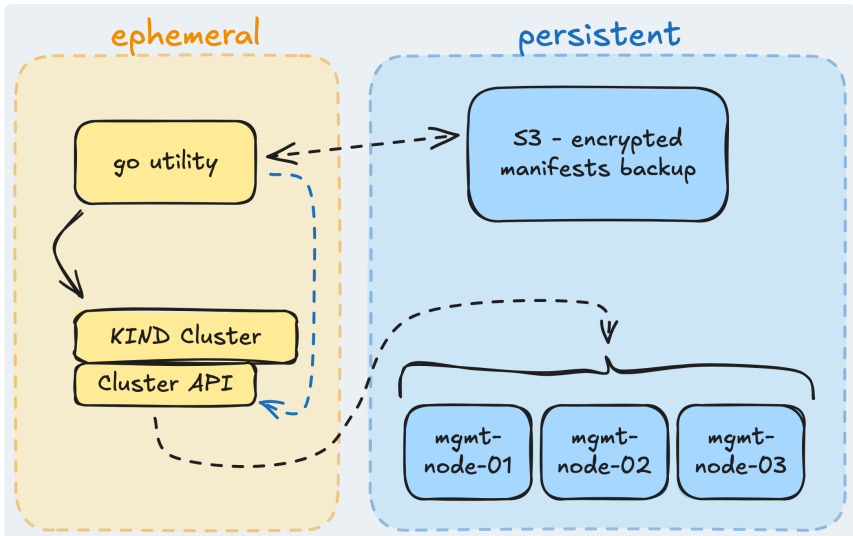
Created with FLUX.1 [dev] by Black Forest Labs

Bootstrapping Issue

The Chicken & Egg problem



Created with FLUX.1 [dev] by Black Forest Labs



Change of Plans

Why we're not implementing ClusterAPI (now)

Siderolabs' Strategic Shift

- Talos ClusterAPI providers are now "**low priority**"
- No new features planned for CAPI providers
- Focus shifted to **Omni** platform

Technical Concerns

- Kubernetes as dependency for cluster management
- In-place upgrades vs. machine replacement
- CAPI complexity vs. simplicity goals

Alternative Approaches

- **siderolabs'** solution (Omni)
- using **ansible** to wrap `talosctl` operations? 😬
- **Purpose-built tool** for Talos? 🪴

References:

- GitHub Issue #193
- Siderolabs Blog

Conclusion

Migration Takeaways

- **ClusterAPI remains a valid option** for many organizations
- **Key lessons learned:**
  - PKI import and secret matching are critical
  - Configuration alignment is crucial
  - Step-by-step node replacement minimizes risk and permits to migrate without downtime

Looking Forward

We're building an **open-source tool**:

- **Purpose-built for Talos** Linux clusters
- **Stateless & minimal** Go binary
- **Plugin-based architecture** for infrastructure flexibility

Coming soon - stay tuned for the open-source release! 🪴

The migration journey continues...

Questions?

SEP 09 – 11, 2025

CONTAINER days CONFERENCE

clement.n8r.ch

Templating Talos Configs

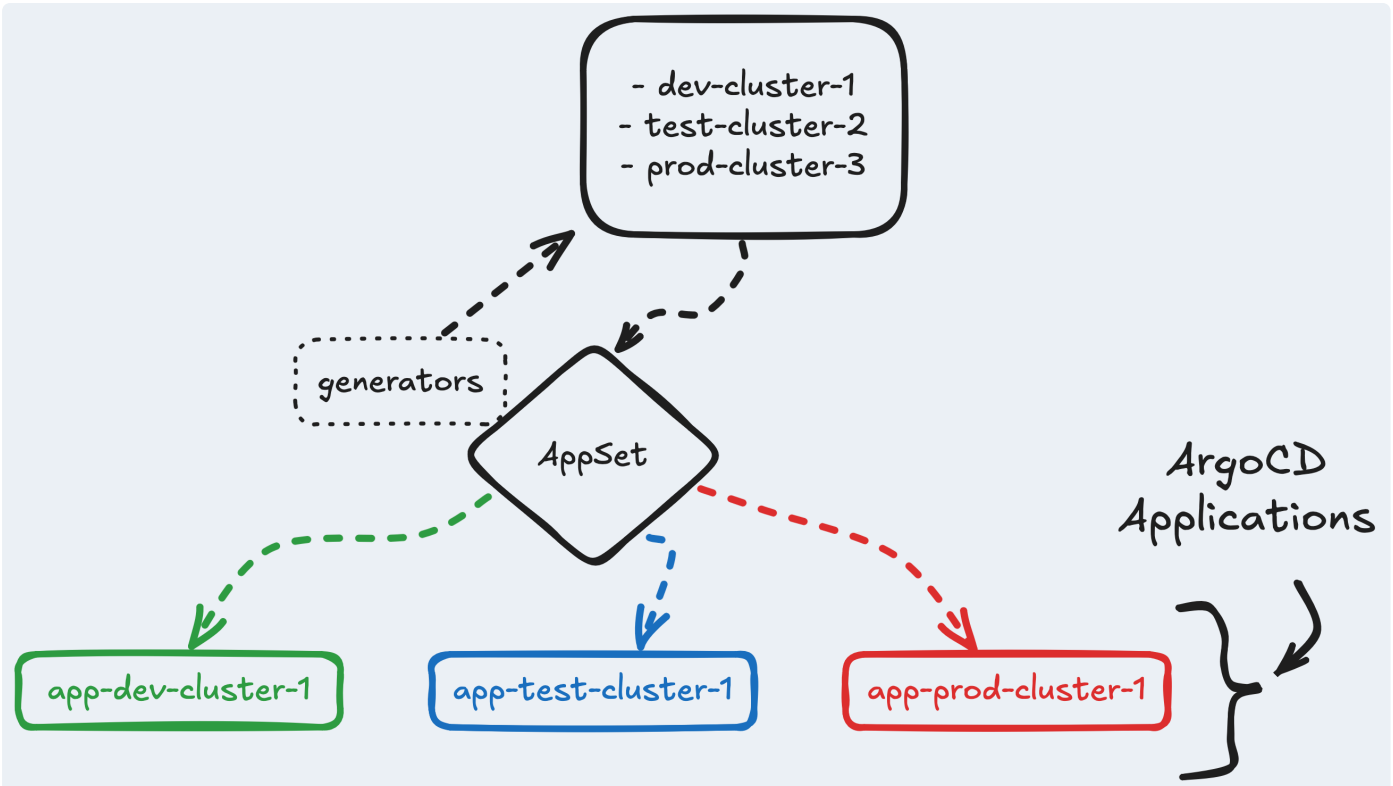
Additional content

```
yq '(.. | select(has("yq_load_str"))) |= load_str(.yq_load_str)'
```

```
---
apiVersion: bootstrap.cluster.x-k8s.io/v1alpha3
kind: TalosConfigTemplate
metadata:
  name: config-template-20250326001
spec:
  template:
    spec:
      generateType: worker
      strategicPatches:
        - | # strategic-patches.yaml content
          cluster:
            apiServer:
              extraArgs:
          ...
        - | # registries.yaml content
          ...
```

ArgoCD AppSets

Additional content



ArgoCD AppSet for Cluster management

Additional content

```
spec.generators:
  - matrix:
      generators:
        - git:
            repoURL: git@your.repo/abcdef.git
            revision: HEAD
            directories:
              - path: envs/lab/*
              - path: envs/test/*
spec.template:
  sources:
    - repoURL: git@your.repo/abcdef.git
      plugin:
        name: lovely-ytt
        parameters:
          - name: lovely_preprocessors
            string: |
              set -euxo && cd ../../../ && \
              find config -name "*.yaml" -exec yq -i 'explode(.) | del(.".*")' {} \; && \
              find {{ .path.path }} -name "*.yaml" -exec yq -i '(.. | select(has("yq_load_str"))) |= load_str(.yq_load_s
```