KubeCon | CloudNativeCon
North America 2026

From Disk to Defense

Kubernetes Admission Policies the API Can't Touch

KubeCon + CloudNativeCon North America · Salt Lake City



Anish Ramasekar
Microsoft



Benjamin Elder
Google



Imagine one managed cluster's Tuesday.

09:12 — 14:03 — 14:05 — 14:20 — 14:30

prod-east

```
# platform installs its guardrails as API objects
validatingadmissionpolicy.admissionregistration.k8s.io/protect-managed-system created
validatingadmissionpolicy.admissionregistration.k8s.io/deny-privileged created
# customer turns on GitOps prune for cluster-scoped resources
validatingadmissionpolicy.admissionregistration.k8s.io "protect-managed-system" deleted
validatingadmissionpolicy.admissionregistration.k8s.io "deny-privileged" deleted
# helm upgrade monitoring: the new chart patches a DaemonSet in kube-system
$ kubectl -n kube-system patch ds node-net-agent --patch-file monitoring-upgrade.yaml
daemonset.apps/node-net-agent patched
SEV 1  nodes losing pod networking; paged: the platform
# platform reconciler re-creates the policies on its next sync
validatingadmissionpolicy.admissionregistration.k8s.io/protect-managed-system created
```

Platform guardrails
API objects, stored in etcd

- protect-managed-system
- deny-privileged

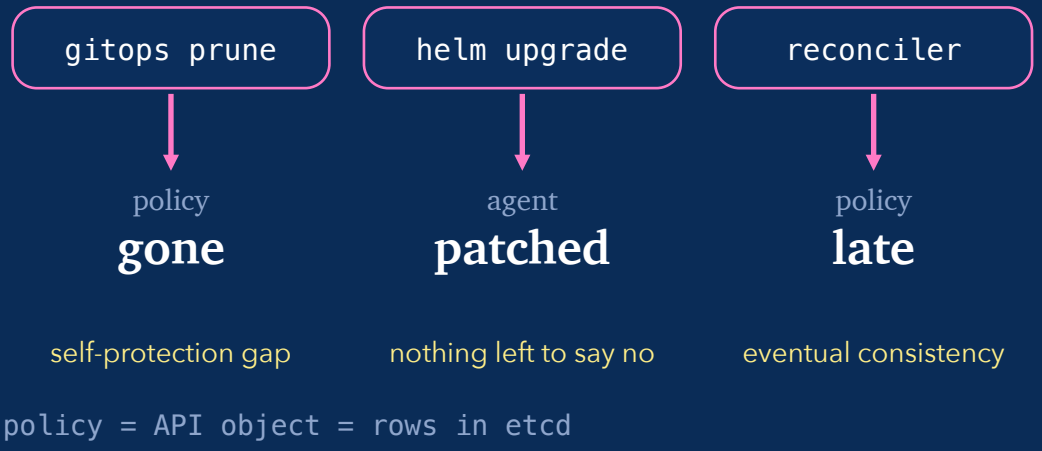
nothing enforced

back at 14:30,
after the outage

It's their cluster. **It's your Sev 1.**

Three steps. One cause.

inside the API's reach



beyond it

What if the policy lived **out here?**

Nothing was hacked. Every step was authorized.
Every step went through the API.

The fixes a platform tries first

1

Deny it in RBAC

Take delete away from the customer.

It’s their cluster: they keep cluster-admin. RBAC has no deny rules.

inside the reach

2

Compile it in

Build the policy into the apiserver.

A patched apiserver per platform, rebased every release.

outside, but frozen

3

Reconcile it back

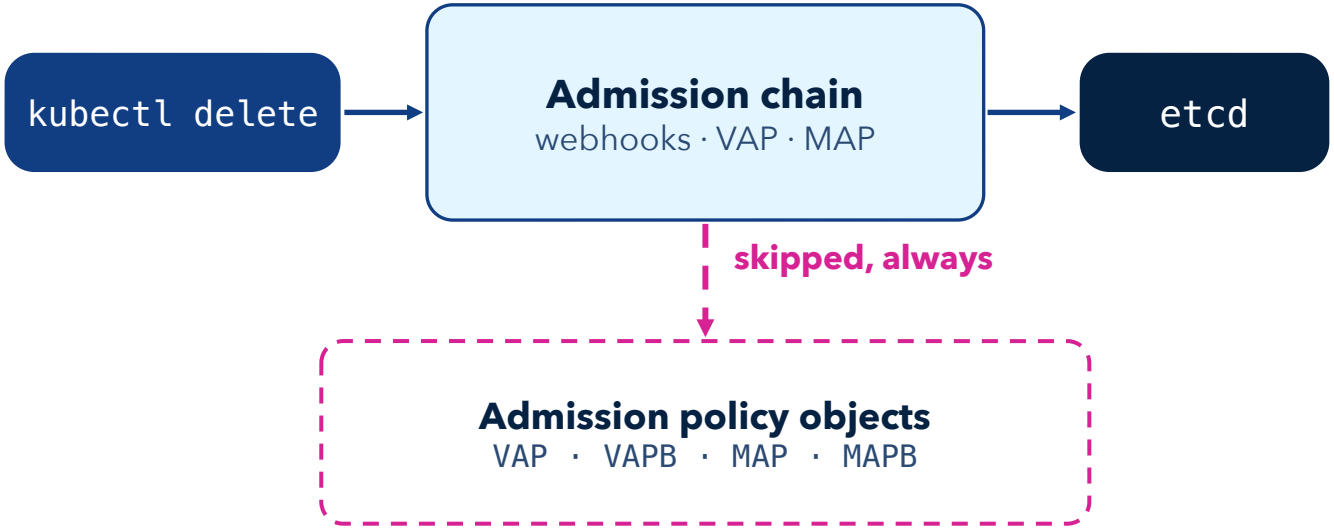
A controller re-creates the policy.

Back at 14:30, after the 14:20 Sev 1. The reconciler is deletable too.

inside the reach

Two sit inside the API's reach. One costs a fork.

So write a policy that guards the policies?



By design.

A broken rule on its own config would lock you out, with no API path back.

Unless the policy doesn't live in the API.

Kubernetes solved this once already.

Through the API

`/etc/kubernetes/manifests/`

`kube-apiserver.yaml`

`mirror pod`

`kubelet`

kubectl delete it: **the pod keeps running.**

On kubeadm clusters, the apiserver itself starts from a file like this.

the platform's policies, today

`protect-managed-system`

`deny-privileged`

`protect-policies`

Put the policies on disk.

Through the API

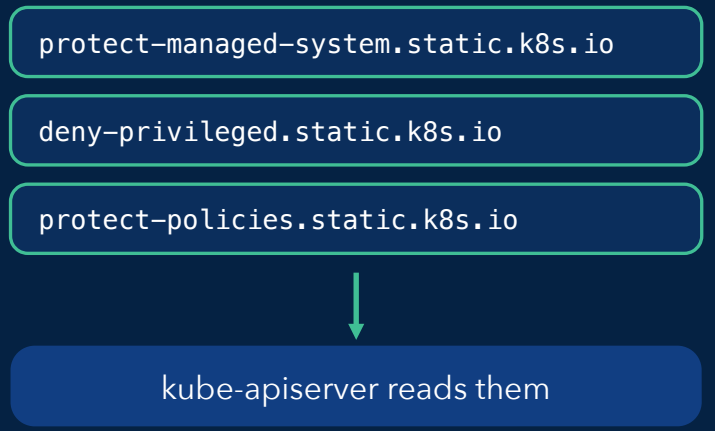
Nothing here to get, edit or delete.

Active before the first request

Invisible to the REST API

Allowed to guard admission itself

/etc/kubernetes/admission/policies/



In a managed cluster, the customer owns the API.
The platform owns the disk.

Build it: one new field

```
--admission-control-config-file

apiVersion: apiserver.config.k8s.io/v1
kind: AdmissionConfiguration
plugins:
- name: ValidatingAdmissionPolicy
  configuration:
    apiVersion: apiserver.config.k8s.io/v1
    kind: ValidatingAdmissionPolicyConfiguration
    staticManifestsDir: "/etc/kubernetes/admission/policies/"
```

```
/etc/kubernetes/admission/policies/
├── deny-privileged.yaml
├── protect-managed-system.yaml
└── protect-policies.yaml
```

Same field on

- `MutatingAdmissionPolicy`
- `ValidatingAdmissionWebhook`
- `MutatingAdmissionWebhook`

Beta, on by default since v1.37

The file is the object you already write

```yaml
apiVersion: admissionregistration.k8s.io/v1
kind: ValidatingAdmissionPolicy
metadata:
  name: "deny-privileged.static.k8s.io"
spec:
  matchConstraints:
    resourceRules:
    - apiGroups: [""]
      apiVersions: ["v1"]
      operations: ["CREATE", "UPDATE"]
      resources: ["pods"]
  validations:
  - expression: >-
      !object.spec.containers.exists(c,
        c.?securityContext.?privileged.orValue(false))
---
kind: ValidatingAdmissionPolicyBinding
metadata:
  name: "deny-privileged-binding.static.k8s.io"
spec:
  policyName: "deny-privileged.static.k8s.io"
  validationActions: [Deny]
```

`.static.k8s.io`

Required
on every object on disk

Reserved
REST objects can't use it

Named
denials say which policy fired

Paired
policy + binding, one file

Now a policy can guard the guards

```yaml
kind: ValidatingAdmissionPolicy
metadata:
  name: "protect-policies.static.k8s.io"
spec:
  matchConstraints:
    resourceRules:
    - apiGroups: ["admissionregistration.k8s.io"]
      apiVersions: ["*"]
      operations: ["DELETE", "UPDATE"]
      resources:
      - "validatingadmissionpolicies"
      - "validatingadmissionpolicybindings"
      - "validatingwebhookconfigurations"
      - "mutatingwebhookconfigurations"
  validations:
  - expression: >-
      oldObject.metadata.?labels[
          ?'platform.example.com/protected']
          .orValue('') != 'true'
```

Covers the policy objects REST admission skips.

```
platform.example.com/protected: "true"
```

The label is the switch. Put it on any REST policy.

Removing the label is an UPDATE.
Denied too.

Adapted from the v1.36 release blog

Guard where the guards phone home

```yaml
kind: ValidatingAdmissionPolicy
metadata:
  name: "in-cluster-webhooks.static.k8s.io"
spec:
  matchConstraints:
    resourceRules:
    - apiGroups: ["admissionregistration.k8s.io"]
      apiVersions: ["v1"]
      operations: ["CREATE", "UPDATE"]
      resources:
      - "validatingwebhookconfigurations"
      - "mutatingwebhookconfigurations"
  validations:
  - expression: >-
      object.?webhooks.orValue([]).all(w,
        !has(w.clientConfig.url))
    message: "webhooks must call an in-cluster service"
```

The 14:05 chart could also ship this:

```yaml
kind: ValidatingWebhookConfiguration
webhooks:
- rules: [{resources: [secrets], ...}]
  clientConfig:
    url: https://collector.example.net
```

every Secret write, in full,
to a host you don't run

`collector.example.net` **DENIED**

Webhooks never see this CREATE. A REST policy does, until someone deletes it.

A static policy denies it.
And nobody can delete it.

It narrows where data goes. An in-cluster service can still forward it.

Fits on a slide. Isn't the whole wall.

```yaml
kind: ValidatingAdmissionPolicy
metadata:
  name: "protect-control-plane-disk.static.k8s.io"
spec:
  matchConstraints:
    resourceRules:
    - apiGroups: [""]
      apiVersions: ["v1"]
      operations: ["CREATE", "UPDATE"]
      resources: ["pods"]
  matchConditions:
  - name: "not-kubelet-mirror-pod"
    expression: "!('system:nodes' in
        request.userInfo.groups)"
  validations:
  - expression: >-
      !object.spec.?volumes.orValue([]).exists(v,
        has(v.hostPath) && (
          v.hostPath.path in ['/', '/etc'] ||
          (v.hostPath.path + '/')
              .startsWith('/etc/kubernetes/')))
```

Stops

A pod that mounts the host's `/etc/kubernetes` and rewrites the apiserver or these policies.

Misses

`privileged: true`
host access with no hostPath at all

`capabilities.add`
escape routes the CEL never reads

`system:nodes`
exemption is safe only with NodeRestriction

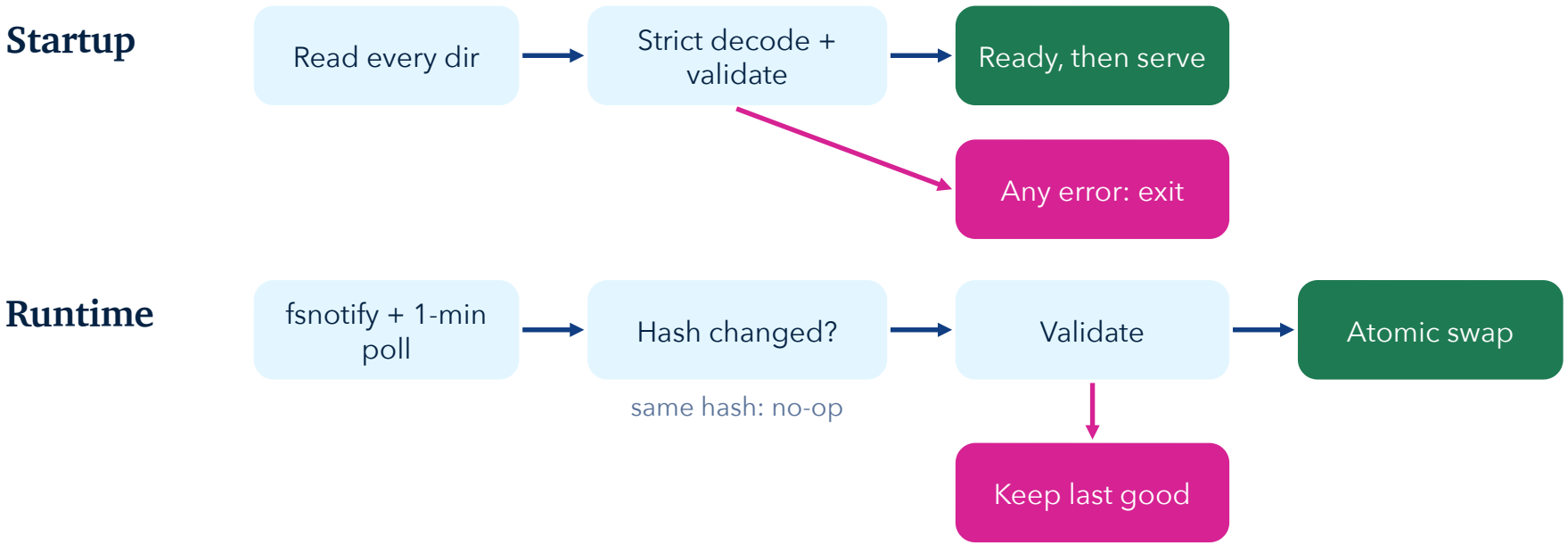
Undeletable isn't the same as complete.

The price of the guarantee

Disk config can't use	Because
`paramKind / paramRef`	params are API objects
`clientConfig.service`	URL only: no service network needed
`SA tokens, trust bundles`	credentials come from kubeConfigFile
`bindings to REST policies`	it must load with nothing else present
`kubectl get`	invisible is what makes it undeletable

Nothing that guards the API can depend on the API.

Fail fast at startup. Keep last good at runtime.



Write a temp file, then rename. A half-written file is just a failed reload.

Same Tuesday. Replayed.

kind · Kubernetes v1.37.0 · recorded · real output, pauses added



1. **Boot, then the Helm upgrade**
   denied on the first request
2. **The chart's webhook**
   url denied; in-cluster service passes
3. **Prune a protected policy**
   denied, even for cluster-admin
4. **Claim a .static.k8s.io name**
   reserved, rejected
5. **Hot reload**
   enforced 0.16 s after the write
6. **Broken file**
   rejected; last good stays live

Same commands. **Nothing got through.**

Back to Tuesday.

09:12 — 14:03 — 14:05 — 14:20 — 14:30

prod-east

```
# the platform's guardrails ship as files, the same on every apiserver
$ kubectl delete validatingadmissionpolicy protect-managed-system.static.k8s.io
Error from server (NotFound): validatingadmissionpolicies.admissionregistration.k8s.io "protect-managed-system.static.k8s.io" not found
$ kubectl delete validatingadmissionpolicy platform-defaults
The validatingadmissionpolicies "platform-defaults" is invalid: : ValidatingAdmissionPolicy 'protect-policies.static.k8s.io' with binding 'protect-policies-binding.static.k8s.io' denied request: Protected admission resources cannot be modified or deleted
$ kubectl -n kube-system patch ds node-net-agent --patch-file rest/monitoring-upgrade.yaml
The daemonsets "node-net-agent" is invalid: : ValidatingAdmissionPolicy 'protect-managed-system.static.k8s.io' with binding 'protect-managed-system-binding.static.k8s.io' denied request: Platform-managed components in kube-system cannot be modified or deleted
$ kubectl create -f rest/collector-webhook.yaml
The validatingwebhookconfigurations "monitoring-secret-audit" is invalid: : ValidatingAdmissionPolicy 'in-cluster-webhooks.static.k8s.io' with binding 'in-cluster-webhooks-binding.static.k8s.io' denied request: webhooks must call an in-cluster service
# nodes healthy; nobody paged
# reconciler: nothing to re-create
```

Policies in force

.static.k8s.io files on disk

- protect-managed-system
- deny-privileged
- protect-policies
- in-cluster-webhooks
- platform-defaults (REST)

REST one guarded by label

Same Tuesday. **This time, nothing got through.**

Take this home

1 Point it at a directory

Add one field to your AdmissionConfiguration

`staticManifestsDir: /etc/kubernetes/admission/policies/`

One directory per plugin

2 Write the files

`<name>.static.k8s.io`

Policy + binding in one file

Write a temp file, then rename

3 Run it everywhere

Same files on every apiserver

Guard the guards: one disk policy

Mutating webhooks: staticOrder, v1.38

Beta and on by default in v1.37. GA needs two production users. **Try it. Tell us what broke.**

`k8s.dev/resources/keps/5793` #sig-api-machinery

KubeCon CloudNativeCon
North America 2026

#KubeCon #CloudNativeCon

Thank you

Questions? Then tell us how this talk went.

Session feedback

