KubeCon CloudNativeCon North America 2026

#KubeCon #CloudNativeCon

5,000 Nodes in 3 Minutes, Twice in a Week

Alex Yuskauskas, NVIDIA



KubeCon CloudNativeCon North America 2026

Alex Yuskauskas

Senior System Software Engineer, NVIDIA

- **NodeWright maintainer**: the controller behind this talk
- Wrote the `copy-fail` and `dirty-frag` mitigation packages
- Node packages behind NVIDIA's kernel CVE mitigations and GPU node bring-up



github.com/NVIDIA/nodewright ·
github.com/NVIDIA/nodewright-packages

Two kernel CVEs, one half-working workaround

CERT-EU 2026-005 workaround

```
echo 'install algif_aead /bin/false' \
  > /etc/modprobe.d/disable-algif.conf
rmmod algif_aead
```

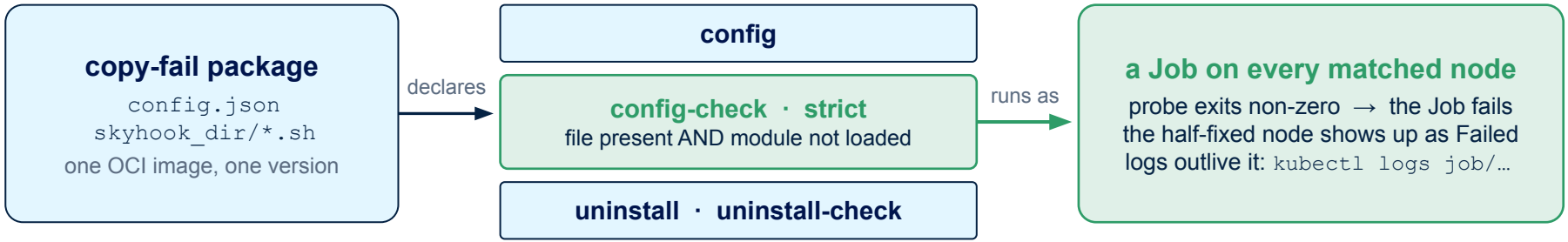
Module was idle: rmmod succeeded
denylist file written · algif_aead **unloaded** → mitigated

Module was in use: rmmod silently skipped
denylist file written · algif_aead **STILL LOADED** → looks fixed, vulnerable until reboot

Copy-Fail `CVE-2026-31431` (algif_aead) · Dirty Frag `CVE-2026-43284` · days apart, no patched kernel yet

The half-fixed node Any audit that greps for the file marks this node done, but the module is *still loaded*. Across a fleet, nothing exposes the gap and nothing is definitively fixed until the next reboot.

Ship the change and the check that fails without it




```
# copy-fail/skyhook_dir/config_check.sh  -- the two assertions that matter
grep -Fxq "install algif_aead /bin/false"  /etc/modprobe.d/disable-algif.conf ||  exit 1
grep -q '^algif_aead '  /proc/modules &&  exit 1   # still loaded -> still vulnerable
```

Escape hatch, visible on the CR `ALLOW_LOADED_MODULE=true` downgrades the second assertion to a warning. Set it in the package `env` on a CR scoped to the nodes SRE has triaged. One package for all distros: `/etc/modprobe.d` is honored on Ubuntu 20.04–24.04, AL2023, RHEL 10.1 and SUSE 16.

Rollout status: one kubectl get, failed Jobs, two gauges

```
$ kubectl get nodewright
NAME        STATUS        PRIORITY   AGE   NODES IN-PROGRESS   COMPLETE NODES   PACKAGES          POLICY
copy-fail   in_progress   100        2m    120                 4880/5000        copy-fail:1.0.0   rolling

$ kubectl get jobs -n nodewright -l nodewright.nvidia.com/name=copy-fail
NAME                                                STATUS    COMPLETIONS   DURATION   AGE
copy-fail-copy-fail-1-0-0-config-ip-10-0-22-86-3f9a1c2e   Failed    0/1           74s        2m
$ kubectl logs -n nodewright job/copy-fail-copy-fail-1-0-0-config-ip-10-0-22-86-3f9a1c2e -c copy-fail-configcheck
ERROR: algif_aead is still loaded in the running kernel; node remains vulnerable to CVE-2026-31431 until next reboot
```

`.status.status` complete · in_progress · erroring · blocked · waiting · paused · disabled · unknown

`.status.nodeStatus{}` per node

`.status.nodeState{}` per package + stage

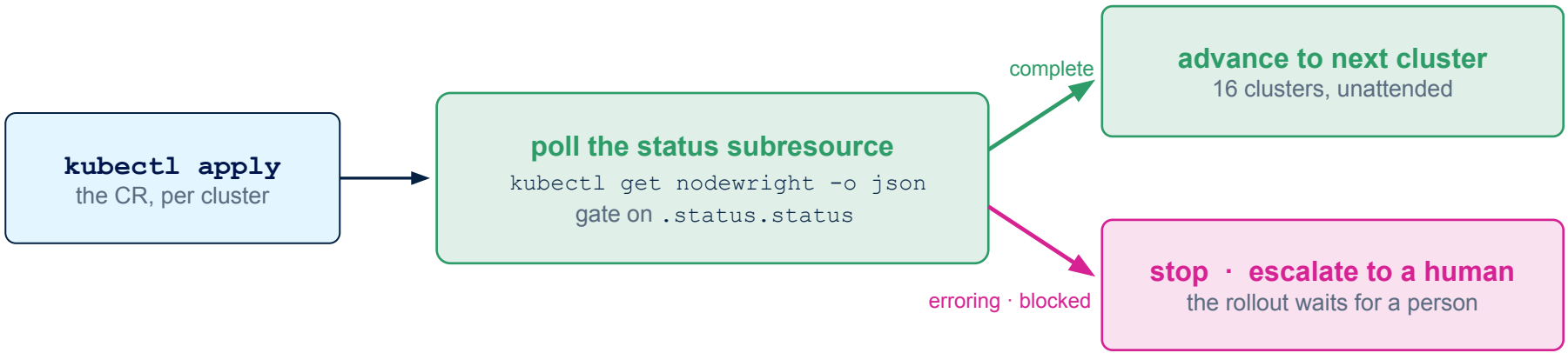
`nodewright_rollout_progress_percent`

`nodewright_node_status_count`

`kubectl delete` = rollback

Done is an enum value on the CR. The nodes that need attention are the failed Jobs.

The status enum let an agent drive the rollout



The agent gates on the CR status, the same field `kubectl` and the dashboards show.

What it caught On clusters running NVIDIA's `forge` control plane, tainted control-plane nodes left our Jobs unschedulable and the CR never reached *complete*. The agent escalated. We added tolerations to the package config and re-applied.

Two CVEs, thousands of nodes, one controller

16
clusters in the most recent run

~5,600
nodes converged in 30 minutes
agent-driven

3 min
largest single cluster: 5,000 nodes
one kubectl apply

2 CVEs
Copy-Fail and Dirty Frag, days apart, same controller

What changed afterward A new package per advisory was too much work, so both mitigations merged into *one* generic kernel-module package configured by YAML in the CR. The controller never changed. **The next CVE is a YAML change.**

KubeCon CloudNativeCon
North America 2026

Ship the change and the check. Let kubectl get be the audit.

github.com/NVIDIA/nodewright: controller and documentation

github.com/NVIDIA/nodewright-packages: the copy-fail package and strict probe

cert.europa.eu/publications/security-advisories/2026-005/: CERT-EU advisory

Portable: `kured` + a CRD, or `system-upgrade-controller`
Works with: Argo CD · Flux · plain `kubectl`

