KubeCon CloudNativeCon North America 2026

#KubeCon #CloudNativeCon

Kubernetes SIG Storage Intro & Deep Dive

Xing Yang (Broadcom), Hemant Kumar (Red Hat) & Michelle Au (Google)



Agenda

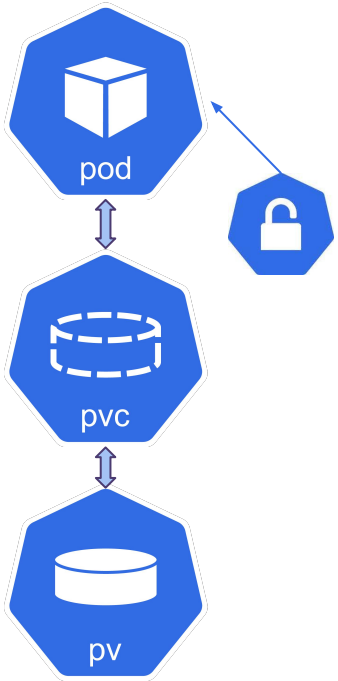
- What is SIG Storage
- Features we are working on (1.36, 1.37, 1.38+)
  - Operations
  - Security
  - Ephemeral
  - Data Protection
  - Scheduling
  - Features in design/prototyping
- How to get involved

SIG Storage

- Co-Chairs: Hemant Kumar (Red Hat) and Xing Yang (Broadcom)
- Saad Ali (Google) moved to emeritus - Thanks Saad!
- Tech Leads: Michelle Au (Google) and Jan Šafránek (Red Hat)
- #sig-storage slack channel members: 7010+
- #csi slack channel members: 1820+
- #sig-storage-cosi slack channel members: 490+
- #csi-windows slack channel members: 290+
- SIG-Storage zoom meeting attendees: 25
- Unique approvers for SIG-owned packages: 30

What we do

- Defined in SIG Storage Charter
  - Persistent Volume Claims and Persistent Volumes
  - Storage Classes and Dynamic Provisioning
  - Kubernetes volume plugins
    - NFS, iSCSI, FC, Local
    - Secret, ConfigMap, DownwardAPI, Projected and EmptyDir Volumes (co-owned with SIG-Node)
  - Container Storage Interface (CSI)
    - CSI sidecars
    - Most CSI drivers are owned by SIG Cloud Provider or other community
  - Container Object Storage Interface (COSI)
    - Targeting Beta in 1.38
    - COSI controller and sidecars
    - COSI drivers



Feature Summary

KubeCon CloudNativeCon North America 2026

Area	Feature	Stage	Release
Data Protection	VolumeGroupSnapshot	GA	1.36
	Changed Block Tracking (CBT)	Beta	1.36
	Snapshot Topology	Target Alpha	1.38*
	COSI (Object Storage)	Target Beta	1.38*
Operations	Volume Health	Alpha	1.37
	Volume reconstruction (global mount)	Target Alpha	1.38*
	Report last used time on a PVC	Beta	1.37
	CSI sidecars all-in-one	Target Alpha	1.38*
Security	CSI SA tokens via secrets field	GA	1.36
	SELinux relabeling w/ mount options	GA	1.37
	Remove gitRepo volume plugin	Deprecated	1.36

Area	Feature	Stage	Release
Scheduling	Storage capacity scoring	Beta	1.37
	CSI attach limits + autoscaler	Beta	1.37
	Mutable volume attach limits	GA	1.36
	CSI ControllerGetNodeInfo	Target Alpha	1.38*
Ephemeral	Dynamic node-local ephemeral vols	Target Alpha	1.38*
	Dynamic resize (memory-backed)	Target Beta	1.38*
	User fields (atomic-write volumes)	Target Beta	1.38*
	Sticky bit (emptyDir mode)	Target Beta	1.38*
	Bind mount options (noexec…)	Target Beta	1.38*
Design in progress	ControllerModifyVolume: topology	Design	—
	StatefulSet volume expansion	Design	—
	CSIStorageCapacity tracking	Design	—

* Targeting 1.38 (in progress). Release = version that reached the current stage. gitRepo: disabled in 1.36, code removal planned for 1.39. Feature names link to KEPs.

KubeCon CloudNativeCon North America 2026

Operations



Volume Health

Alpha in 1.37



Allow CSI drivers to report volume health

- **Volume Health from controller:** Allows CSI drivers to report per-volume health from controller.
- **Volume health from Node:** Allows CSI drivers to report per-volume health from node.
- **Storage subsystem health from Node:** Allows CSI drivers to report health of storage subsystem
- **Status:** Alpha in 1.37, staying **Alpha in 1.38**

- **Primary goal :** Health reporting
- Design APIs to support automated recovery and future extensions

Volume Health Reporting

Alpha in 1.37

KubeCon CloudNativeCon North America 2026

External-health-monitor from control-plane can report volume health in PVC objects

Cluster-Wide Polling

External health monitor can poll and report health of all PVC status in the cluster.

Control-Plane Integration

External-health-monitor running from control-plane updates health status directly in PVC objects.

pvc.yaml

```yaml
apiVersion : v1
kind: PersistentVolumeClaim
metadata :
  name: application-data
spec:
  volumeName : pvc-example-volume
status :
  healthStatus :
    healthConditions :
    - status: Degraded
      reason: HighLatency
      message: "Storage backend latency exceeds the driver's
threshold."
    lastTransitionTime : "2026-10-27T14:00:00Z"
```

Volume Health Reporting

Alpha in 1.37

KubeCon CloudNativeCon North America 2026

Kubelet can report health of degraded volumes in pod's Status

Direct Status Reporting

Kubelet can report multipath errors, iscsi errors etc directly in pod's Status.

Comprehensive Coverage

Kubelet reports health of both mounted volumes and volumes which are somehow failing to mount.

pod-status.yaml

```yaml
kind: Pod
spec:
volumes:
- name: data
persistentVolumeClaim :
claimName : application-data
status:
  phase: Running
  volumeStatuses:
-   name: data
    healthConditions :
    - status: Degraded
      reason: MultipathLoss
      message: "One storage path is unavailable; remaining
paths are operational."
      lastTransitionTime : "2026-10-27T14:00:00Z"
```

Volume Health Reporting

Alpha in 1.37

KubeCon CloudNativeCon North America 2026

Kubelet also can report storage subsystem health via CSINode object

Unreachable Volume Reporting

Kubelet can report certain type of volumes are unreachable from the node.

Smart Scheduling

A future version of scheduler or an operator could avoid nodes which have one or more volume types inaccessible.

csinode.yaml

```yaml
kind: CSINode
metadata:
  name: worker-1
spec:
  drivers:
  - name: nfs.csi.example.com
    nodeID: worker-1
status:
  storageHealth:
  - name: nfs.csi.example.com
    healthConditions:
    - status: StorageUnreachable
      reason: NFSServerUnreachable
      message: "NFS server 192.168.1.100:2049 is unreachable
from worker-1"
      accessMode: ReadWriteMany
      lastTransitionTime: "2026-10-27T14:00:00Z"
```

Report last used time on PVC

Beta in 1.37

KubeCon CloudNativeCon North America 2026

Kubernetes will report when a PVC was last used

Unused PVC Identification

Useful for finding unused PVCs in the cluster to optimize storage usage.

Kube-state-metrics

Can report PVC transition time directly into cluster metrics.

PR #3101 Changes

pvc-status.yaml

```yaml
kind: PersistentVolumeClaim
metadata:
  name: application-data
spec:
  volumeName: pvc-example-volume
status:
  phase: Bound
  accessModes:
    - ReadWriteOnce
  conditions:
  - type: Unused
    status: "True"
    reason: NoPodsUsingPVC
    message: "No pods are currently referencing this PVC"
    lastTransitionTime: "2026-10-26T18:30:00Z"
```

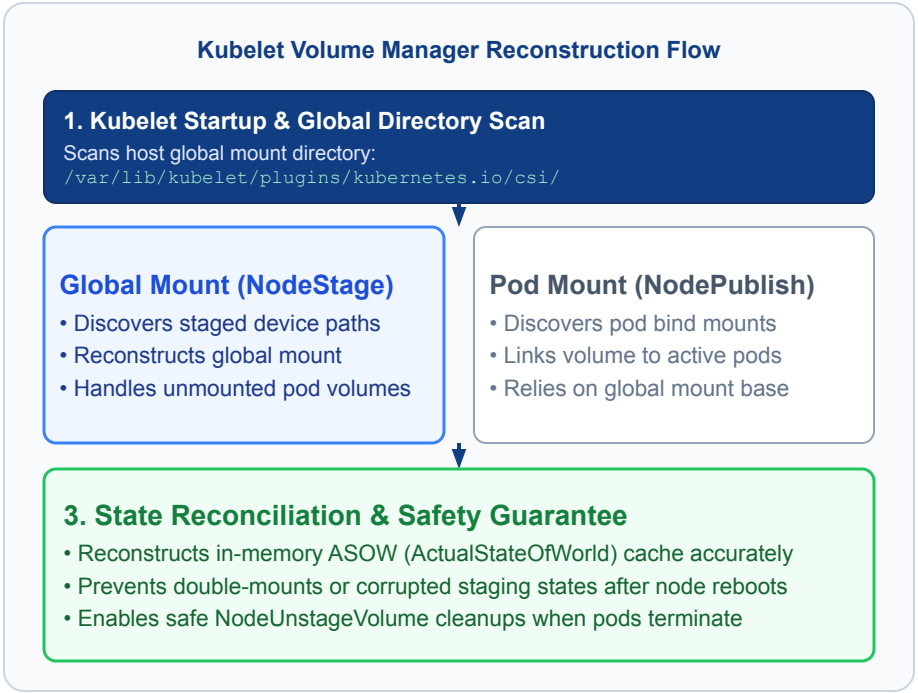
Volume Reconstruction

Alpha in 1.38

KubeCon CloudNativeCon North America 2026

Extend volume reconstruction to use global mount path

- **Reconstruct mount state:** Kubelet inspects the host filesystem on restart to detect volumes mounted at the global directory
- **Global mount path scanning:** Discovers staging mounts under `plugins/kubernetes.io/csi/` before Pod volumes are reconstructed
- **Reconcile ASOW cache:** Populates ActualStateOfWorld (ASOW) so unmount and detach controllers do not leak mounts
- **Prevent mount leaks & conflicts:** Eliminates duplicate `NodeStageVolume` calls or orphaned global mount paths across node reboots
- **Status in Kubernetes 1.38:** Introduced as **Alpha** under KEP-6059 / PR #6059



KubeCon CloudNativeCon North America 2026

Security



Remove gitRepo Volume Plugin

Disabled in 1.36

KubeCon CloudNativeCon North America 2026

Eliminate node privilege escalation vulnerability (CVE-2024-10220)

- **Security vulnerability:** gitRepo volume plugin allowed arbitrary command execution as **root on the host node** via malicious repository parameters
- **Deprecated & unmaintained:** in-tree plugin has been deprecated for many releases and superseded by modern cloud-native patterns
- **Removal timeline:** disabled by default & locked in **1.36** (feature gate `GitRepoVolumeDriver`); complete code removal planned for **1.39** (KEP-5040)
- **Modern alternatives:** migrate to **git-sync** sidecars or dedicated `initContainers` with emptyDir volumes

Legacy gitRepo (Vulnerable & Deprecated):

```
volumes:
- name: git-vol
  gitRepo:  # Disabled in 1.36, Removed 1.39
    repository: "https://github.com/..."
```

Migrate to secure initContainer pattern:

Recommended: initContainer / git-sync

```
initContainers:
- name: git-clone
  image: alpine/git:latest
  args: ["clone", "https://github.com/...", "/data"]
  volumeMounts: [{name: "data", mountPath: "/data"}]
```

Pod SA Tokens via CSI Secrets

GA in 1.36

KubeCon CloudNativeCon North America 2026

Securely pass bound ServiceAccount tokens to CSI drivers via standard secrets field

- **Security enhancement:** pass audience-bound Pod ServiceAccount tokens directly through the CSI secrets field
- **Use cases:** CSI drivers that authenticate with Pod's identity. Most commonly used in secret-provider CSI drivers.
- **NodePublishVolumeRequest:** tokens delivered in **secrets** payload instead of unencrypted volume context
- **CSIDriver opt-in:** drivers opt in via **serviceAccountTokenInSecrets: true** in driver spec
- **Status:** Graduated to **GA in 1.36** (**KEP-5538**); essential for secret-store CSI drivers

```yaml
apiVersion: storage.k8s.io/v1
kind: CSIDriver
metadata:
  name: secrets-store.csi.k8s.io
spec:
  tokenRequests:
  - audience: "vault.corp.internal"
    expirationSeconds: 3600
  # Opt-in to receive tokens in secrets field:
  serviceAccountTokenInSecrets: true
```

SELinux Relabeling with Mount Options

GA in 1.37

KubeCon CloudNativeCon North America 2026

Speed up container startup by mounting volumes directly with SELinux context

- **Eliminates startup bottleneck:** avoids recursive filesystem traversal (`chcon`) across millions of files at pod start
- **Pod opt-out:** configure `spec.securityContext.seLinuxChangePolicy = Recursive` to retain recursive relabeling if needed
- CSI driver needs to support it and opt-in via `CSIDriver spec.SELinuxMount`
- **Status:** SELinuxMountReadWriteOncePod reached GA in 1.36; full **SELinuxMount** is **GA in 1.37** (**KEP-1710**)

KubeCon CloudNativeCon North America 2026

Ephemeral Storage



Dynamic Node-Local Ephemeral Volumes

Target Alpha in 1.38

KubeCon CloudNativeCon North America 2026

Add and remove node-local ephemeral volumes without recreating pods

- **Supports in-place updates:** works with ConfigMaps, Secrets, Projected, and EmptyDir volumes
- **No pod restarts:** eliminates downtime when rotating credentials, updating configs, or resizing scratch space
- Add new volume to new container or existing container with a restart.
- **Target Alpha in 1.38:** guarded by DynamicNodeLocalVolumes feature gate

```
spec:
containers:
- name: app
  volumeMounts:
  - name: token-vol
    mountPath: /tokens
volumes:
- name: token-vol
  secret:
    secretName:
```

In-place pod update:
Kubelet mounts new volume

```
spec:
containers:
- name: app
  volumeMounts:
  - name: token-vol
    mountPath: /tokens
  - name: cache-vol # Mount added!
    mountPath: /cache
volumes:
- name: token-vol
  secret:
    secretName:
- name: cache-vol # New volume added
```

Dynamic Resize Memory-Backed emptyDir

Target Beta in 1.38

KubeCon CloudNativeCon North America 2026

Dynamically expand emptyDir tmpfs sizeLimit at runtime without pod disruption

- **Mutable sizeLimit:** pod spec permits updating **emptyDir.sizeLimit** in place without disruption
- **Memory cgroups integration:** keeps memory quotas aligned with container resource allocations
- **Status:** Alpha in 1.37, targeting **Beta in 1.38**
- **Use cases:** cache bursts, ML model loading into memory, and high-performance in-memory pipelines

```
volumes:
- name: ram-cache
  emptyDir:
    medium: Memory
    sizeLimit: 1Gi   # Initial
```

In-place pod update:
kubelet remounts tmpfs

```
volumes:
- name: ram-cache
  emptyDir:
    medium: Memory
    sizeLimit: 4Gi   # Resized!
```

User Fields: Atomic-Write Volumes

Target Beta in 1.38

Control ownership UID/GID for files projected from Secrets, ConfigMaps, and DownwardAPI

- **Problem:** projected files previously defaulted to root (UID 0), forcing non-root containers to run chmod/chown initContainers
- **Granular ownership:** explicit **defaultUser** and **user** fields on volume sources and projected items
- **Security & simplicity:** removes the need for root initContainers to fix permissions in restricted Pod Security Standards
- **Status:** Alpha in 1.37, targeting **Beta in 1.38** (KEP-5936)

```yaml
volumes:
- name: secret-vol
  secret:
    secretName: certs
    defaultUser: 10001  # UID
```

Or override per-item in projected volume:

```yaml
volumes:
- name: app-cfg
  configMap:
    name: app-config
    items:
    - key: tls.key
      path: cfg.json
      user: 10001  # UID
```

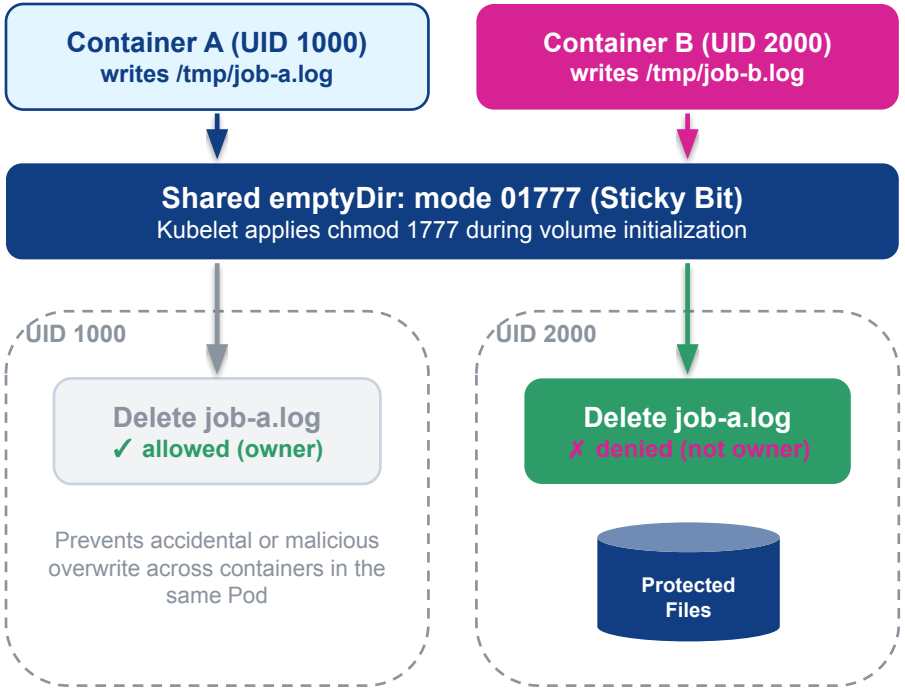
Permission Bits for emptyDir

Target Beta in 1.38

KubeCon CloudNativeCon North America 2026

Set POSIX permission bits, including sticky bit (mode 01777) on shared emptyDir volumes for safe multi-user sharing

- **Multi-tenant / multi-container safety:** when multiple containers run with different UIDs and share an emptyDir volume (e.g. /tmp), any container could delete other containers' files
- **POSIX sticky bit semantics:** only the file owner or root can delete or rename files in the shared directory
- **Configurable permission mode:** adds **emptyDir.defaultMode: 01777** support directly in pod specification
- **Target Beta in 1.38:** Alpha in 1.37, moves to Beta enabled by default (KEP-5502)



Bind Mount Options on any VolumeMount

Target Beta in 1.38

KubeCon CloudNativeCon North America 2026

Apply OS-level security hardening (noexec, nosuid, nodev) directly to container volume mounts

- **Fine-grained security hardening:** enforce Linux mount options on individual container **volumeMounts**
- **Supported options:** **noexec** (blocks execution), **nosuid** (blocks setuid escalation), **nodev** (blocks special device creation)
- **Applies across all volume types:** emptyDir, ConfigMap, Secret, PVC, and HostPath
- **Status:** Alpha in 1.37, targeting **Beta in 1.38** (KEP-5855)

```yaml
containers:
- name: web-app
  image: app:v1.2
  volumeMounts:
  - name: scratch-vol
    mountPath: /tmp
    mountOptions:
    - noexec
    - nosuid
    - nodev
```

KubeCon CloudNativeCon
North America 2026

Data Protection



VolumeGroupSnapshot

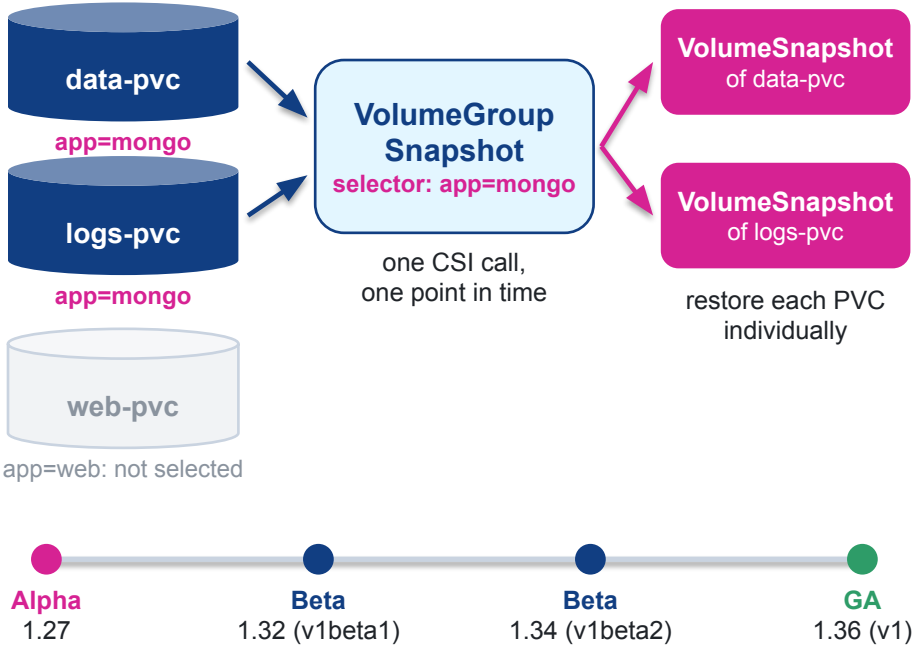
GA in 1.36

KubeCon CloudNativeCon North America 2026

Snapshot multiple volumes at one point in time: crash-consistent, no app quiesce

- Group PVCs with a **label selector**; one VolumeSnapshot is created per PVC
- Restore any PVC from its own VolumeSnapshot
- CSI drivers implement the optional **Group Controller Service**

```yaml
apiVersion: groupsnapshot.storage.k8s.io/v1
kind: VolumeGroupSnapshot
metadata:
  name: mongo-groupsnap
spec:
  volumeGroupSnapshotClassName: csi-group
  source:
    selector:
      matchLabels:
        app: mongo
```



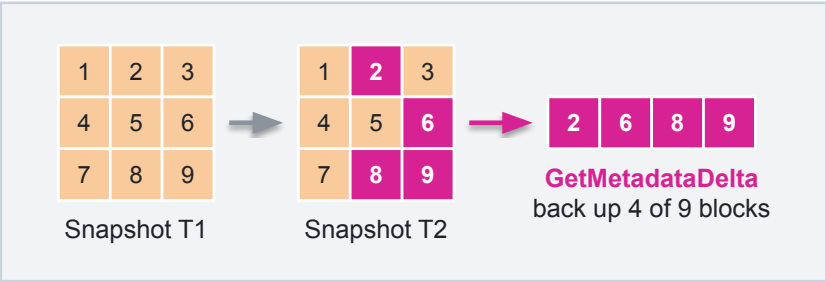
Changed Block Tracking

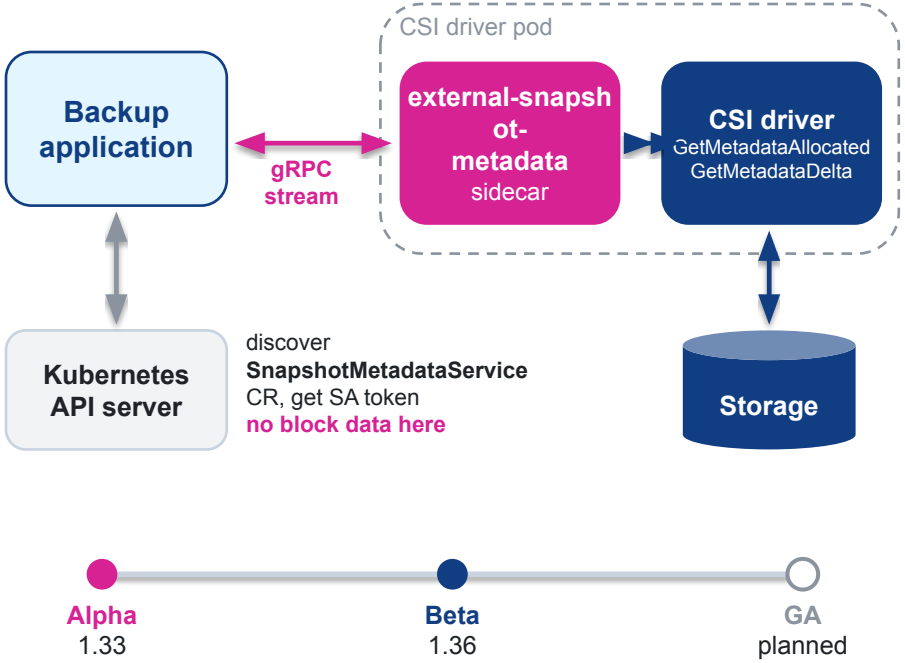
Beta in 1.36

KubeCon CloudNativeCon North America 2026

Back up only the blocks that changed between snapshots

- Optional CSI **SnapshotMetadata** service: **allocated** blocks and **changed** blocks
- Streams over gRPC, **bypassing the API server**
- Secured with audience-scoped SA tokens





COSI

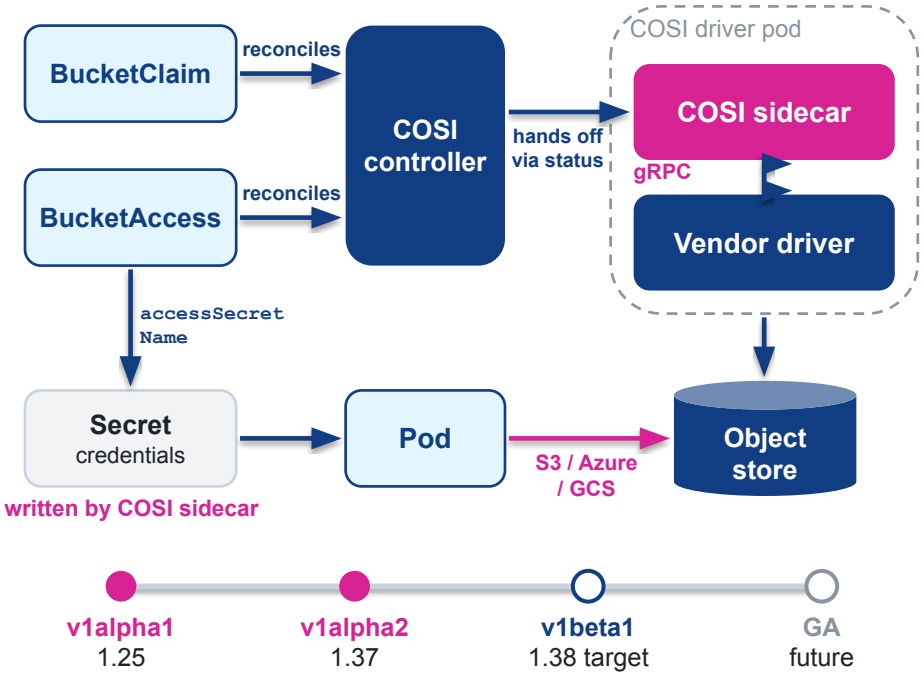
v1alpha2 in 1.37

KubeCon CloudNativeCon North America 2026

Container Object Storage Interface: CSI-style, vendor-neutral buckets for Kubernetes

- Self-service **buckets** and **access credentials**
- Portable across clusters and vendors
- Protocols: S3, Azure Blob, GCS
- Out of tree: CRDs, controller, sidecar

If you know CSI…		COSI
StorageClass	→	**BucketClass**
PersistentVolumeClaim	→	**BucketClaim**
PersistentVolume	→	**Bucket**
Volume mount	→	**BucketAccess + Secret**



COSI: What's New

v1beta1 target 1.38

KubeCon CloudNativeCon North America 2026

v1alpha2 in 1.37: the most requested features, plus a leaner sidecar

- One BucketAccess, **multiple buckets**
- Per-bucket access modes: **ReadWrite, ReadOnly, WriteOnly**
- Secrets use individual fields (no JSON blob), easy to load as env vars
- Sidecar only reconciles Bucket and BucketAccess: **fewer RBAC permissions**, better version skew
- **v1beta1 (1.38 target):** no API changes; adds controller and sidecar metrics




```yaml
apiVersion: objectstorage.k8s.io/v1alpha2
kind: BucketAccess
metadata:
  name: app-access
spec:
  bucketAccessClassName: s3-access
  protocol: S3
  bucketClaims:
  - bucketClaimName: logs
    accessSecretName: logs-creds
    accessModes: {objectData:ReadWrite}
  - bucketClaimName: images
    accessSecretName: images-creds
    accessModes: {objectData:ReadOnly}
```

Snapshot Topology

Alpha target 1.38

Snapshots record where they can be restored, so restores land in the right place

- **Problem:** snapshots have no topology, so a restore in the wrong zone can fail or leave data unreachable
- **Scope:** only CSI drivers whose snapshots aren't accessible from every zone or region
- CSI: new **SNAPSHOT_ACCESSIBILITY_CONSTRAINTS** capability; topology returned in CreateSnapshotResponse
- VolumeSnapshotContent gets **nodeAffinity** (mutable)
- VolumeSnapshotClass gets **allowedTopologies** to choose where snapshots live
- Feature gate: **VolumeSnapshotTopology**
- Foundation for future **cross-region snapshot cloning**

```
kind: VolumeSnapshotClass
driver: csi.example.com
allowedTopologies:
- matchLabelExpressions:
  - key: topology.kubernetes.io/zone
    values: [zone-b]
```

CreateSnapshot: driver returns snapshot topology

```
kind: VolumeSnapshotContent
spec:
  driver: csi.example.com
  nodeAffinity:
  - matchLabelExpressions:
    - key: topology.kubernetes.io/zone
      values: [zone-b]
```

Use cases: audit snapshot placement for DR, pin snapshots to a location, restore without manual zone matching

Snapshot Topology: Restore via Webhook

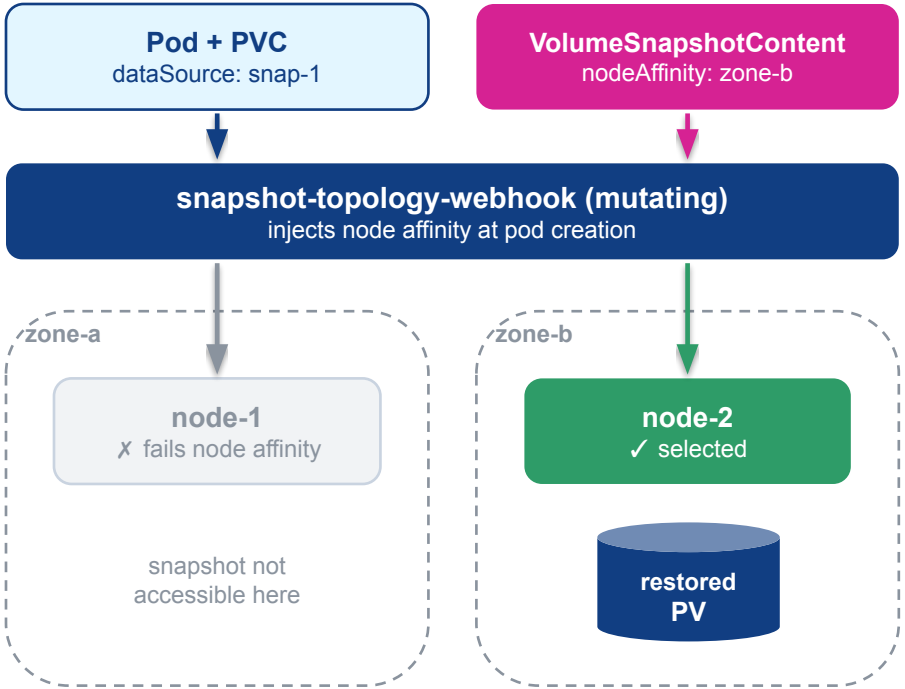
The pod carries the snapshot's topology to the scheduler

WaitForFirstConsumer: webhook

- **Mutating pod webhook:** merges the snapshot's nodeAffinity into the pod at creation
- **Topology not known yet:** gates the pod; the webhook's leader-elected topology controller adds the affinity, then ungates
- **Validating PVC webhook:** rejects a selected-node the snapshot can't be restored on
- Ships in **external-snapshotter**; works with the default scheduler

Immediate: external-provisioner

- Intersects the snapshot's nodeAffinity with StorageClass allowedTopologies



KubeCon CloudNativeCon North America 2026

Storage Scheduling



Autoscaler and Scheduler Changes

Beta in 1.37

KubeCon CloudNativeCon North America 2026

Scheduler can now prevent pod placement to nodes without CSI driver

CSIDriver Spec (YAML)

```yaml
apiVersion: storage.k8s.io/v1
kind: CSIDriver
metadata:
  name: nfs.csi.example.com
spec:
  attachRequired: false
  podInfoOnMount: false
  volumeLifecycleModes:
    - Persistent
  preventPodSchedulingIfMissing:true
```

Opt-in Flag per CSIDriver

Enabled by default in 1.37, but requires explicit per-driver opt-in using the `preventPodSchedulingIfMissing` flag in the `CSIDriver` object.

Avoids Pod Crowding

Prevents scheduling too many pods (with volumes) to a new node even before CSI driver is started. This frequently resulted in scheduling volumes that is beyond attach limit of node.

Allows Cluster Heterogeneity

Replaces complex, custom pod scheduling rules. Seamlessly mix and match different storage solutions across nodes without extra overhead.

Autoscaler and Scheduler Changes

Beta in 1.37

KubeCon CloudNativeCon North America 2026

Autoscaler now takes into account attach limits when scaling nodes for pending pods

Configuration & Purpose

Enabled by default behind CLI flag:
`–enable-csi-node-aware-scheduling`

Goal: Achieves more accurate use of node resources when scaling nodes for pending pods.

Node Group Scaling Modes

Existing Nodes: Uses attach limits from active nodes to synthesize volume capacity for new nodes.

Scaling from 0: Allows CloudProviders to report supported volume count via established APIs.

Interaction with `preventPodSchedulingIfMissing`

Scaling from Zero Caution

Do not enable `preventPodSchedulingIfMissing` if using Cluster Autoscaler (CA) and your driver's cloudprovider lacks integration for scaling from zero.

When safe to Enable

Generally safe to enable in CA deployments that do not support scaling from zero, even without cloudprovider integration.

Generally safe to enable in CA deployments where CloudProvider reports CSI attach limits via TemplateNodeInfo. For example - AWS, ClusterAPI

Generally safe if **NOT** using CA but you are using **Karpenter** because Karpenter does not run scheduler plugins.

Mutable Attach Limits

GA in 1.36

KubeCon CloudNativeCon North America 2026

Kubelet can poll periodically for any attach limit changes

CSIDriver & CSINode Specs (YAML)

```yaml
# CSIDriver config
apiVersion: storage.k8s.io/v1
kind: CSIDriver
spec:
  nodeAllocatableUpdatePeriodSeconds: 60

# CSINode status
apiVersion: storage.k8s.io/v1
kind: CSINode
spec:
  drivers:
    - name: block.csi.example.com
      nodeID: worker-1
      allocatable:
        count: 16
```

Dynamic Attach Limits

Attach limits reported in CSINode used to be static after driver registration. Drivers can now update previously reported attach limits dynamically.

Pod Eviction Handling

Kubelet can evict pods if drivers report `ResourceExhausted` errors during volume attach operations.

Scheduler Reschedule Caveat

Scheduler may reschedule evicted pods back to the same node unless CSINode spec is explicitly updated to reflect new limits.

CSI ControllerGetNodeInfo

Alpha in 1.38

KubeCon CloudNativeCon North America 2026

Allows control-plane via external-attacher to report attach limit and topology

Current Architecture

- Currently kubelet reports attach limit and topology info of a node via `CSINode` object.
- **Requires API access to cloudprovider from every node.**
- Usually getting attach limit and topology information requires CSI driver to talk to cloudprovider or control-plane of storage.

Key Challenges

Scalability Issue

Does not scale very well across large clusters due to per-node API communication overhead.

Security Risk

A compromised node can leak access credentials of the storage control-plane.

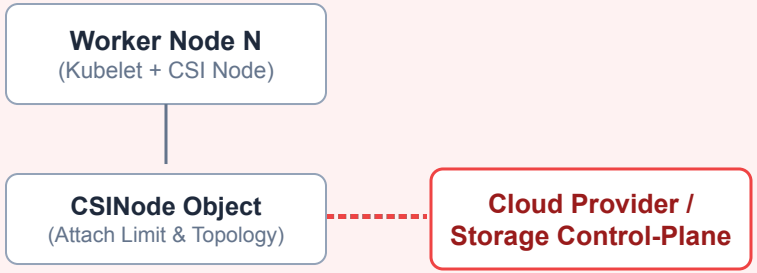
CSI ControllerGetNodeInfo Flow

Alpha in 1.38

KubeCon CloudNativeCon North America 2026

Architecture Comparison: Legacy Node-Direct vs. Centralized Control-Plane

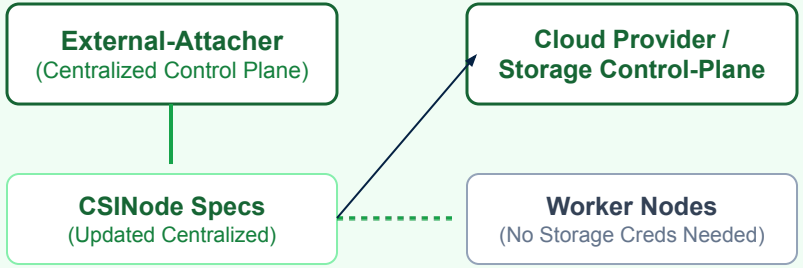
Legacy: Per-Node Direct Access



Drawbacks & Vulnerabilities:

- **Scalability Bottleneck:** Every node independently polls storage APIs, causing massive overhead.
- **Security Risk:** Storage credentials must be distributed across all worker nodes.
- **Attack Surface:** A compromised node leaks full storage control-plane access.

New: ControllerGetNodeInfo (1.38+)



Key Advantages:

- **Centralized Reporting:** External-attacher queries ControllerGetNodeInfo on behalf of nodes.
- **Enhanced Security:** Worker nodes no longer require direct credentials to cloud/storage control-plane.
- **High Scalability:** Dramatically reduces API communication traffic across large clusters.

Storage capacity scoring

Beta in 1.37

KubeCon CloudNativeCon North America 2026

Allow volume provisioning during scheduling to prefer nodes with less or more free space

Prefer Nodes with less free space available- Scheduler

```yaml
apiVersion: kubescheduler.config.k8s.io/v1
kind: KubeSchedulerConfiguration
profiles:
- schedulerName: default-scheduler
- schedulerName: storage-packing
  pluginConfig:
  - name: VolumeBinding
    args:
      shape:
      - utilization: 0
        score: 0
      - utilization: 100
        score: 10
```

Storage Bin-Packing Strategy

Scores nodes with less free space higher (shape: 0% util → score 0, 100% util → score 10). Maximizes storage density and packs pods onto fuller nodes.

VolumeBinding & Capacity Reporting

Configured via `VolumeBinding` plugin args with `WaitForFirstConsumer`. Remaining node capacity is dynamically tracked via `CSIStorageCapacity` objects.

Multi-Profile & Custom Scoring Shapes

Targeted pods select this profile via `schedulerName: storage-packing`. Linear or non-linear scoring shapes (0–10) balance storage utilization against pod placement priority.

Storage capacity scoring

Beta in 1.37

KubeCon CloudNativeCon North America 2026

Allow volume provisioning during scheduling to prefer nodes with less or more free space

Prefer nodes with more free space -Scheduler Config

```yaml
apiVersion: kubescheduler.config.k8s.io/v1
kind: KubeSchedulerConfiguration
profiles:
- schedulerName: default-scheduler
- schedulerName: storage-spread
  pluginConfig:
  - name: VolumeBinding
    args:
      shape:
      - utilization: 0
        score: 10
      - utilization: 100
        score: 0
```

Storage Spreading Strategy

Scores nodes with more free space higher (shape: 0% util → score 10, 100% util → score 0). Distributes I/O workload and avoids early disk exhaustion on individual nodes.

Capacity Unreported & Default Behavior

When CSI drivers do not publish `CSIStorageCapacity` or capacity is unknown, the scheduler scores the node using a configurable fallback score (default: 0).

Workload-Specific Scheduler Profiles

Allows clusters to run bin-packing and spreading concurrently under separate profiles. Batch jobs can pack storage tightly, while I/O-heavy databases spread across nodes.

KubeCon CloudNativeCon
North America 2026

Designs in Progress



PV Node Affinity & Topology Modification

Design / Alpha

KubeCon CloudNativeCon North America 2026

Allow VolumeAttributesClass to alter PersistentVolume nodeAffinity dynamically without workload downtime

- **Mutable PV Node Affinity (KEP-5381):** permits storage drivers to update volume accessibility constraints
- **Use cases:** converting volume from zonal to regional
- **Non-disruptive operations:** CSI drivers ensure active pods and workloads continue running uninterrupted

```yaml
apiVersion: storage.k8s.io/v1alpha1
kind: VolumeAttributesClass
metadata:
  name: regional-vac
driverName: my-driver
parameters:
  regional: true
```

Apply VAC to PVC:
ControllerModifyVolume updates PV topology

```yaml
# PV nodeAffinity dynamically updated to regional
nodeAffinity:
  required:
    nodeSelectorTerms:
    - matchExpressions:
      - key: topology.kubernetes.io/region
        operator: In
        values: ["us-east-1"]
```

StatefulSet Volume Expansion

Design / KEP-4650

KubeCon CloudNativeCon North America 2026

Automate storage resizing across StatefulSet pods via volumeClaimTemplates updates

- **In-place template expansion:** update storage requests directly in StatefulSet volumeClaimTemplates
- **Automatic PVC propagation:** StatefulSet controller reconciles and modifies child PVC sizes automatically
- **Eliminates manual toil:** replaces complex manual PVC patching and STS deletion with --cascade=orphan
- **Safety & ordering:** honors StatefulSet rolling update partitioning and volume expansion safeguards
- **No current owner:** If you are interested in contributing, let us know!

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: kafka-cluster
spec:
  replicas: 3
  volumeClaimTemplates:
  - metadata:
      name: datadir
    spec:
      accessModes: ["ReadWriteOnce"]
      resources:
        requests:
          # Seamless in-place expansion:
          storage: 100Gi  # Was 20Gi
  template: ...
```

CSIStorageCapacity Tracking Enhancements

Design / KEP-5925

KubeCon CloudNativeCon North America 2026

High-scale storage capacity reporting to optimize topology-aware pod scheduling

- **Multi-PVC scheduling:** Evaluates all of a Pod's PVCs together
- **Per-pool capacity reporting:** Model multiple identical pools (e.g., separate disks) on a node.
- **Capacity reservation:** Marks scheduled capacity as reserved in the scheduler until CSI driver refreshes it, preventing concurrent overprovisioning.
- **Capacity-aware rescheduling:** Support rebuilding volumes on a new node with sufficient space.

```
apiVersion: storage.k8s.io/v1
kind: CSIStorageCapacity
metadata:
name: csi-pool-node-1
storageClassName: fast-nvme
# Multi-pool capacity reporting
availableCapacities:
- 5Ti
- 1Ti
- 2Ti
nodeTopology:
 kubernetes.io/hostname: node-1
```

KubeCon CloudNativeCon North America 2026

How to get involved



How to get involved

- Start at the SIG Storage page:
  - https://github.com/kubernetes/community/tree/master/sig-storage
- Attend our meetings:
  - Bi-weekly meeting: 9am PT / 6pm CET every second Thursday.
  - Sub-project meetings: COSI, CSI, Data protection
- Mailing List:
  - sig-storage@kubernetes.io
- Slack channel:
  - #sig-storage
  - #csi
  - #sig-storage-cosi

Resources

- SIG Storage page
- Storage concepts
- CSI driver docs
- CSI spec
- CSI sample driver hostpath deployment example
- SIG Storage annual report

KubeCon CloudNativeCon
North America 2026

Thank You

