

# Distributed Filesystem in Kubernetes with CSI

# Distributed Filesystem in Kubernetes with CSI

## Initial Question

**User:**

If I would want to write a filesystem (I am not sure if it would be a user space only filesystem or kernel space too), which programming language would you recommend?

**ChatGPT:**

Hey! That's an awesome project idea building a filesystem is a deep and rewarding challenge.

(Explanation of language choices for kernel-space and user-space FS)

## Distributed Filesystem Idea

**User:**

Yeah, I would like to write a clustered filesystem or a distributed FS, I am not sure yet.

**ChatGPT:**

Nice! That's getting into really interesting territory.

(Details about clustered vs distributed FS, language recommendations)

## Kubernetes + CSI Integration

**User:**

How would you change the above things if this filesystem would need to be in a Kubernetes environment and a CSI driver might be also needed?

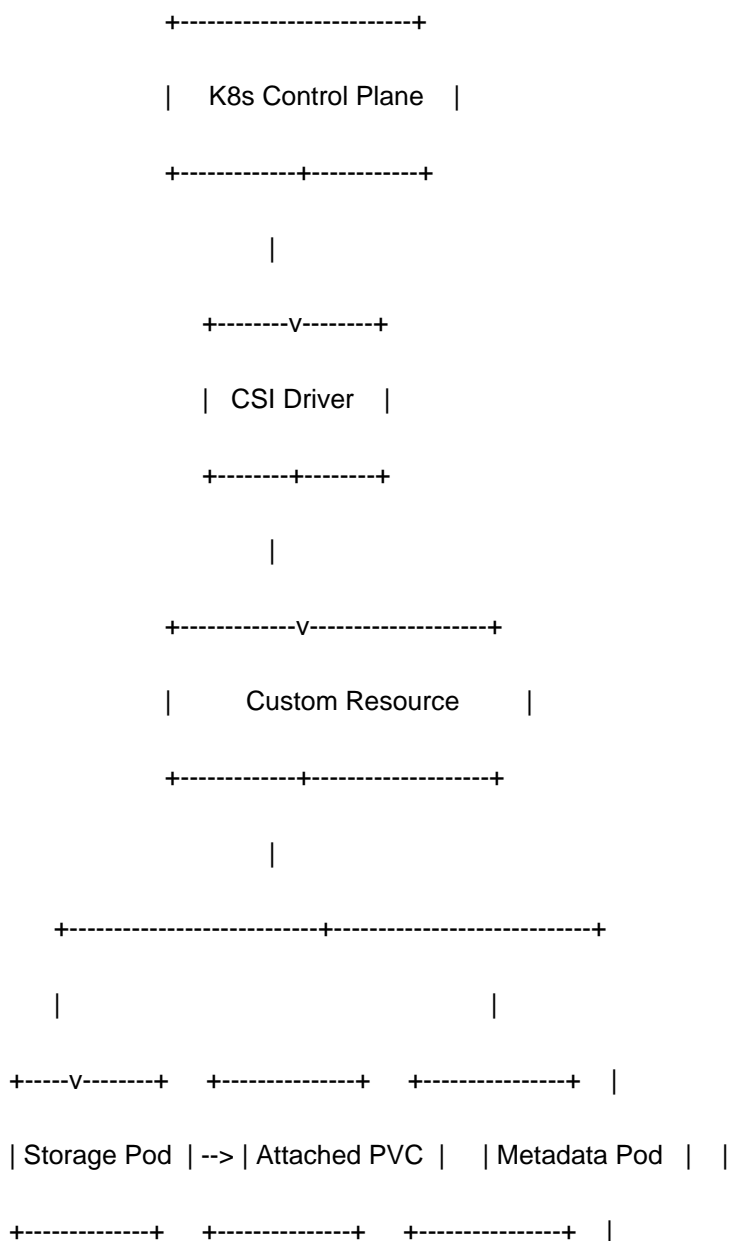
## Distributed Filesystem in Kubernetes with CSI

**\*\*ChatGPT:\*\***

You're basically looking to:

- Build a distributed/clustered FS that runs inside Kubernetes
- Integrates with the CSI to provide volumes via PVs/PVCs
- Spreads or shards data across multiple PVCs and pods

### ### Architecture Overview



## Distributed Filesystem in Kubernetes with CSI

| | | |

+-----K8s Pods with Your Filesystem-----+-----+

(FUSE Mount or in-kernel driver)

### ### Project Directory Structure

my-cluster-fs/

cmd/

csi-controller/

csi-node/

deploy/

helm-chart/

storageclass.yaml

fs-core/

metadata/

storage/

fuse/

lib.rs

proto/

scripts/

Dockerfile.\*

go.mod / go.sum

Cargo.toml

## Distributed Filesystem in Kubernetes with CSI

### ## MVP Workflow

1. User creates a PVC
2. CSI Controller provisions PVCs, spawns storage + metadata pods
3. Node plugin mounts FS using FUSE
4. FS logic (in Rust) handles chunking, replication, metadata