**Kadalu**

Introduction:

Kadalu is an open source project that provides storage solutions for Kubernetes.

It aims to provide highly scalable and reliable storage for containerized applications running on a Kubernetes cluster. Kadalu integrates with existing storage solutions, such as GlusterFS, to provide a unified storage system that can be easily managed and scaled.

The project's main goal is to simplify storage management for Kubernetes users and provide a flexible and scalable solution for their storage needs.

Project Summary:

| Website | https://kadalu.tech/ |
| --- | --- |
| Organization / Foundation Name | Kadalu Technologies |
| License | Apache License, Version 2.0 |
| Open / Proprietary | Open Source |
| Source Path (if open source) | https://github.com/kadalu/kadalu.git |
| Brief Description | Open-source project that aims to provide persistent and scalable storage solutions for containers and virtual machines on Kubernetes. |

Project Details

* Key Features:
* Persistent Volume (PV) and Persistent Volume Claim (PVC) management for stateful applications.
* Support for multiple storage backends, including GlusterFS, NFS, and iSCSI.
* Data migration between different storage backends and cluster nodes.
* Automatic data tiering and data replication to ensure high availability and data durability.
* Dynamic provisioning of storage volumes for stateful applications.
* Integration with Kubernetes for easy management and orchestration of storage resources
* Support for advanced storage features like snapshots, quotas, and backups.
* Architecture:



The architecture of Kadalu consists of three main components:

* GlusterFS: A scalable, distributed file system that is used to store data.
* Kadalu Daemon: A background process that communicates with GlusterFS and other Kadalu daemons to manage storage.
* Kadalu CLI: A command-line interface that allows users to interact with the Kadalu daemon and manage storage.

The GlusterFS storage is made up of nodes, which are physical or virtual machines that host the storage. The data is stored on multiple nodes and is automatically replicated for reliability and availability.

The Kadalu daemon communicates with GlusterFS and the Kadalu CLI to manage storage, such as creating and deleting volumes, adding and removing nodes, and monitoring storage usage.

The Kadalu CLI provides a simple, user-friendly interface for managing storage and interacting with the Kadalu daemon.

* Technical Details:

Some of the technical details of Kadalu are:

* Supports multiple storage backends: Kadalu supports multiple storage backends like GlusterFS, Ceph, NFS and can be extended to support other storage solutions.
* Kubernetes integration: Kadalu integrates with Kubernetes to manage storage volumes and provide persistent storage for containers.
* Snapshot and Backup: Kadalu provides snapshot and backup functionality to enable recovery from data loss and data protection.
* Data tiering: Kadalu supports data tiering that helps to manage the data life-cycle by moving data between different storage tiers.
* Scalability: Kadalu is designed to scale horizontally to accommodate growth in storage needs.
* Performance: Kadalu provides high performance and low latency storage access to containers.
* Open Source: Kadalu is open source software released under the Apache License 2.0.