Graphical user interface, text

Description automatically generated

**Open Source Demystified Level 1**

Project Report submitted in partial fulfilment of the requirements for the award of the degree of

**BACHELOR OF COMPUTER APPLICATIONS (BCA)**



**SAMARPAN DAHAL**

**21BCAA54**

Under the guidance of

**Prof. SHEETHAL AJI MANI**

**DEPARTMENT OF COMPUTER SCIENCE (UG)**

**BCA PROGRAMME**

**KRISTU JAYANTI COLLEGE (Autonomous)**

**K. Narayanapura, Kothanur P.O., Bangalore – 560077**

Graphical user interface, text

Description automatically generated

## **DEPARTMENT OF COMPUTER SCIENCE (UG)**

**CERTIFICATE OF COMPLETION**

This is to certify that the practical lab for the course titled “**Open Source Demystified Level 1”** has been satisfactorily completed by **Samarpan Dahal**, **21BCAA54** in partial fulfilment of the award of the Bachelor of Computer Applications degree requirements prescribed by Kristu Jayanti College (Autonomous) Bengaluru (Affiliated to Bangalore University) during the academic year 2022-2023.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Internal Guide***  ***External Mentor***  ***Valued by Examiners***  **1:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **2: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |  |  |  |  |  | ***Head of the Department*** |

***Centre:* Kristu Jayanti College**

***Date:***

Graphical user interface, text

Description automatically generated

**DECLARATION**

I**, SAMARPAN DAHAL, 21BCAA54** hereby declare that the practical lab work for the course titled “**Open Source Demystified Level 1**” has been completed by me, as per the course guidelines, under the guidance of **PROF.SHEETHAL AJI MANI.**

This report work has not been submitted earlier either to any University / Institution or any other body for the fulfilment of the requirement of a course of study.

**Signature**

**Bangalore**

**Date:**

**ACKNOWLEDGEMENT**

The success of the project depends upon the efforts invested. It’s my duty to acknowledge and thank the individuals who has contributed to the successful completion of the project.

I take this opportunity to express my profound and wholehearted thanks to **FR. DR. AUGUSTINE GEORGE, PRINCIPAL, KRISTU JAYANTI COLLEGE**, **BANGALORE** for providing ample facilities made to undergo my project successfully.

I express my deep sense of gratitude and sincere thanks to our **DEAN DR. CALISTUS JUDE** and Head

of the Department **PROF. SEVUGA PANDIAN** for their valuable support and advice.

I feel immense pleasure to thank my respected guide **PROF. SHEETHAL AJI MANI** for sustaining Interest and providing dynamic guidance in aiding me to complete this project immaculately and impeccably and for being the source of my strength and confidence.

It is my duty to express my thanks to all Teaching and Non-Teaching Staff members of Computer science department who offered me help directly or indirectly by their suggestions. The successful completion of my project would not have been possible without my parent’s Sacrifice, guidance, and prayers. I take this opportunity to thank everyone for their continuous Encouragement. I convey my thankfulness to all my friends who were with me to share my happiness and agony.

Last but not the least I thank almighty God for giving me strength and good health throughout my project and enabling me to complete it successfully.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **Synopsis** | **1** |
| **Glossary** | **2-5** |
| **Introduction** | **6-7** |
| **Open Source Introduction** | **8-10** |
| **Open Source Project Examples** | **11-30** |
| 1. **Dataleyk** | **11-15** |
| 1. **Terra** | **16-19** |
| 1. **Volcano** | **20-30** |
| **How to contribute to Open Source** | **31** |
| **Ways to Contribute** | **32** |
| **Methods to join the community and start contributing** | **33-34** |
| **Community Engagement Experience** | **35-36** |
| **Contribution Flowchart:** | **37** |
| **My Contributions** | **37-38** |
| **Open Source Value** | **39** |
| **References** | **40** |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Description** | **Page no** |
| **1** | Sample picture1 | **1** |
| **2** | Open source structure | **09** |
| **3** | contribution figure | **10** |
| **4** | Terra architecture diagram | **17** |
| **5** | Soda foundation org diagram | **18** |
| **6** | soda foundation chart | **19** |
| **7** | soda foundation project | **19** |
| **8** | Volcano architecture 1 | **22** |
| **9** | Volcano architecture 2 | **22** |
| **10** | volcano contributers | **30** |
| **11** | contribution open source | **31** |
| **12** | ways to contribute | **32** |
| **13** | Contribution flowchart | **37** |
| **14** | My Contribution | **38** |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Sno** | **Table name** | **Page no** |
| 1 | Dataleyk Summary | 11 |
| 2 | Terra Summary | 16 |
| 3 | Volcano Summary | 20 |
| 4 | Volvano Maintainers | 30 |

**SYNOPSIS**

**Open Source Demystified Level 1** course covers the understanding of Open source technology and ecosystem. It provides basic introduction to open source, terms, ecosystem, community, how to contribute, what are the key potential and open source culture. This understanding helps one to identify, enter, contribute, learn and grow one’s competency and career.

A picture containing shape

Description automatically generated

FIG 1:Sample picture1

It also provides the hands-on experience with given open source project community ecosystem. This project report covers the overall learning and activities completed as a part of the course. This report will give one to understand about overall summary and guidance to contribute to open source projects further.

# **Glossary**

# 

# **Dataleyk**

# **Data mining:** the process of discovering patterns in large data sets using statistical methods, machine learning, and database systems.

# **Machine learning:** a type of artificial intelligence (AI) that enables computers to learn from data without being explicitly programmed.

# **Natural language processing (NLP):** a field of computer science and AI that focuses on enabling computers to understand, interpret, and generate human language.

# **Predictive analytics:** the use of data, statistical algorithms, and machine learning techniques to identify the likelihood of future outcomes based on historical data.

# **Deep learning:** a subset of machine learning that involves training artificial neural networks to perform complex tasks such as image and speech recognition.

# **Artificial intelligence (AI)**: the simulation of human intelligence in machines that are programmed to think and learn like humans.

# **Big data:** extremely large data sets that may be analyzed computationally to reveal patterns, trends, and associations, especially relating to human behavior and interactions.

# **Data visualization:** the representation of data and information in a graphical or pictorial format to help users better understand and interpret the data.

# 

# **Terra**

1. **Stablecoin:** A cryptocurrency that is designed to maintain a stable value by being pegged to the value of an underlying asset, such as a fiat currency or a commodity.
2. **Luna:** The native cryptocurrency of the Terra network, used for staking and governance.
3. **Anchor:** A decentralized finance (DeFi) application on the Terra network that offers a savings platform with stable interest rates and yield optimization.
4. **Mirror Protocol:** A DeFi platform on the Terra network that enables the creation of synthetic assets, such as stocks and commodities, that track the price of their real-world counterparts.
5. **Terra Station:** A wallet and blockchain explorer for the Terra network.
6. **Validator:** A network node on the Terra network responsible for validating transactions and adding new blocks to the blockchain.
7. **Staking:** The process of holding cryptocurrency in a wallet to support the operation of a blockchain network and earn rewards in return.
8. **Governance:** The process by which stakeholders in a blockchain network can propose, vote on, and implement changes to the network’s protocol and parameters.
9. **Oracle:** A third-party service or system that provides real-world data to a blockchain network.
10. **Decentralized Exchange (DEX):** A cryptocurrency exchange that operates on a decentralized blockchain network, allowing users to trade cryptocurrencies without the need for a centralized intermediary. The Terra network has its own DEX called Terraswap.
11. **SDS software defined storage :** Software-defined storage is a marketing term for computer data storage software for policy-based provisioning and management of data storage independent of the underlying hardware.
12. **Open Stack :** OpenStack is an open source platform that uses pooled virtual resources to build and manage private and public clouds.
13. **Cryptocurrencies:** a digital currency produced by a public network, rather than any government, that uses cryptography to make sure payments are sent and received safely.
14. **OPEX :** OPEX is an Open source Exchange platform for cryptocurrencies.

**Volcano**

1. **Kubernetes:** An open-source platform for automating the deployment, scaling, and management of containerized applications.
2. **Workload:** A specific type of task or job that needs to be executed on a computing system, such as running a machine learning model.
3. **Scheduler:** A component of the Volcano project that is responsible for scheduling workloads to run on the Kubernetes cluster based on their resource requirements and availability.
4. **Job:** A specific instance of a workload that is scheduled to run on the cluster.
5. **Pod:** The smallest deployable unit in Kubernetes, which contains one or more containers and the resources they need to run.
6. **Executor:** A component of the Volcano project that is responsible for managing the lifecycle of a workload, including launching and terminating pods as needed.
7. **Resource:** A specific amount of CPU, memory, or other computing resources that are required for a workload to run.
8. **Queue:** A mechanism used by the Volcano scheduler to manage the order in which workloads are executed.
9. **Priority:** A measure of the importance or urgency of a workload relative to other workloads in the queue.
10. **Plugin:** An extension to the Volcano project that provides additional functionality, such as support for specific machine learning frameworks or hardware accelerators.
11. **CRD (Custom Resource Definition):** A Kubernetes extension that allows users to define their own custom resources, such as new types of workloads or resources.
12. **Preemption:** A mechanism used by the Volcano scheduler to reclaim resources from low-priority workloads to make them available for high-priority workloads.

# **Introduction**

# 

# **About this document :**

# Open source refers to a software development approach in which the source code of the software is made publicly available and can be accessed, modified, and redistributed by anyone. This means that the software is free to use, modify, and distribute, without any restrictions from the original copyright holders.The idea of open source software emerged in the 1980s as a response to the growing commercialization of software and the restrictions imposed by proprietary software licenses. The term "open source" was coined in 1998 by a group of developers who wanted to create a more collaborative and decentralized approach to software development.One of the key benefits of open source software is that it allows for greater collaboration among developers, who can work together to improve the software and fix bugs. This can result in higher-quality software that meets the needs of a wider range of users. Open source software is often developed by a community of developers who share a common interest in the software, rather than a single company or organization.

# Another benefit of open source software is its flexibility. Because the source code is openly accessible, users can modify the software to meet their specific needs. This can be particularly valuable for businesses or organizations that have unique requirements that cannot be met by commercial software.

# Open source software is also cost-effective, as it is often free to use and distribute. This can result in significant cost savings for businesses and organizations that rely on software for their operations. Additionally, because the software is not tied to a single vendor or licensing agreement, users have more freedom and flexibility in how they use and distribute it. Open source software has become increasingly popular in recent years, with many companies, including Microsoft, Google, and IBM, contributing to open source projects. Open source technology has also been applied to areas outside of software development, such as hardware and design. The use of open source software and technology is expected to continue to grow as more businesses and organizations realize its benefits..

# **Purpose :**

This open source document provides comprehensive information about open source technology, its benefits, its impact on the technology industry, and the various open source projects.The document consists of an introduction that explains what open source is and its significance in the technology industry. It provides a brief history of open source and how it has evolved over time.

# 

# **Audience:**

This open source document provides comprehensive information about open source technology, its benefits, its impact on the technology industry, and the various open source projects. The document consists of an introduction that explains what open source is and its significance in the technology industry. It provides a brief history of open source and how it has evolved over time.

The audience for this document includes individuals who are interested in open source technology and its ecosystem, as well as those who are interested in contributing to open source projects. This may include students, software developers, IT professionals, and anyone else who is interested in learning more about open source technology and how to contribute to open source projects. The document is designed to be accessible to individuals with a basic understanding of software development and programming concepts, but no prior experience with open source is required. The document provides detailed explanations of key concepts and terminology, and includes step-by-step guidance on how to contribute to open source projects. The primary audience for this document is individuals who are new to open source and are interested in learning more about how to contribute.

It is also relevant for experienced contributors who want to expand their knowledge of open source contributions and the value they provide. Additionally, organizations and businesses that are considering using open source software may find this document useful in understanding the benefits

# 

# **Open Source Introduction**

Open source software is a type of software whose source code is freely available to anyone. This approach is based on the belief that access to source code should be free and open to everyone, and that software development should be collaborative and decentralized. The community is key to the success of open source software, as the code is developed by a community of volunteers who share their knowledge and skills to make the software better.

One of the advantages of open source software is that the source code is open to everyone, so anyone can inspect, modify, and enhance it. This can lead to higher quality software that meets the needs of a wider range of users. The open source approach also results in more secure software, as the source code can be reviewed by anyone and vulnerabilities can be identified and fixed quickly.

While open source software is free, it is not necessarily without cost. Developers and companies may choose to donate their time and resources to the project, but there may still be costs associated with maintaining and supporting the software. However, the benefits of open source software can outweigh the costs, especially for businesses and organizations that require flexibility and customization.

Open source software stands in contrast to closed source software, which is developed and distributed by companies like Microsoft for a profit. Closed source software typically does not allow users to view or modify the source code, which can limit its flexibility and functionality.

In conclusion, open source software is a collaborative and decentralized approach to software development that relies on the contributions of communities. Its success depends on the involvement of developers, companies, and users who believe in the benefits of open source software.

**Here are some key terms related to open source:**

* Source code: This refers to the human-readable instructions that make up a software program. In open source software, the source code is freely available for others to view, modify, and redistribute.
* License: A legal agreement that outlines the terms and conditions under which open source software can be used, modified, and distributed.
* Community: A group of developers, users, and enthusiasts who collaborate on the development and maintenance of open source software.
* Fork: A process in which a new project is created from an existing open source project by copying its source code and creating a new repository.
* Pull request: A proposed change to an open source project made by a contributor. The change is submitted for review by the project maintainers, who can then accept or reject it.
* Distribution: The process of making an open source software program available to others for use or modification.
* Contribution: Any addition, modification, or improvement made to an open source project by a member of the community.
* Bug fix: A correction made to an open source program to address a defect or issue.
* Repository: A location where the source code of an open source project is stored and managed, often using version control software.

**Open source Structure**

Diagram

Description automatically generated

Fig 2:Open source structure

**Some famous examples of open-source products are :**

1. Operating system : Android , Ubuntu, Linux .
2. Internet Browsers : Mozilla FireFox, Chromium.
3. Integrated Development Environment : Vs code(Visual studio code),Android Studio,PyCharm, Xcode
4. Database Management Systems: MySQL, PostgreSQL, MongoDB
5. Content Management Systems: WordPress, Drupal, Joomla
6. Graphics and Design Software: GIMP, Inkscape, Blender
7. Office Suites: LibreOffice, Apache OpenOffice
8. Web Servers: Apache HTTP Server, Nginx

Shape

Description automatically generated**Open-source community and Contributions :**

The open-source community is a worldwide community of programmers and software developers who are continuously working on various open-source projects to make our lives better. This community is self-governing and self-organizing, there are no executives to take the decisions solely. This community plays a very crucial role in the sustainability of various open-source organizations.

# 

# **Open Source Project Examples** FIG 3:contribution figure

1. Dataleyk
2. Terra
3. Volcano

A picture containing text, tableware, dishware, plate

Description automatically generated

**DATALEYK**

**INTRODUCTION:**

Dataleyk is a cloud data lake platform that makes big data analytics simple and accessible to all.

Dataleyk is a provider of cloud-based data lake and data analytics solutions. The product offered by the company is a data lake platform that provides visualization, BI tools, and built-in graphs. The features of the product include data encryption, infrastructure management, modernization of data warehousing, etc. The clients of the company include ELEVEN KINGS, Keepface, HELLO MARY, etc.

**Dataleyk Summary**

|  |  |
| --- | --- |
| **Website** | www.dataleyk.com |
| **Organization/Foundation Name** | Dataleyk |
| **License** | Not open Source |
| **Open/Proprietary** | Proprietary |
| **SourcePath(if open source)** | \_\_\_\_\_\_\_\_\_\_\_\_ |
| **Brief Description** | Dataleyk is a fully managed data platform that ingests and analyzes a wide variety of data, enabling businesses of all sizes to benefit from the vast amount of data accessible to them. |

Table 1: Dataleyk summary

**Project Details :**

**Key Features**

* **Security & Scalability**

Dataleyk runs on AWS data centers around the world. This means Dataleyk users enjoy the high availability of AWS servers as well as the best in data security along with automatic server-side encryption (AES-256) and GDPR compliance.

* **Smart Pricing**

Dataleyk offers a pay-as-you-go structure for pricing for computing and ingestion services. By paying for services on an as needed basis, you can redirect your focus and funds to innovation and invention.

* **Free trial**

Can discover the benefits of cloud-based data analytics for free with Dataleyk

#### **Architecture**

#### No architecture has been provided

**Current Usage**

Dataleyk is the missing link in reaching your data-driven goals. Our platform makes it quick and easy to have a stable, flexible and reliable cloud data lake with near-zero technical knowledge. Bring all of your company data from every single source, explore with SQL and visualize with your favorite BI tool or our advanced built-in graphs.

#### **Ready For Takeoff**

Modernize your data warehousing with Dataleyk. Our state of the art cloud data platform is ready to handle your scalable structured and unstructured data.

**Highly Secure**

Data is an asset. Dataleyk is a secure, cloud data platform which encrypts all of your data and offers on demand data warehousing.

#### **Zero Maintenance**

Zero maintenance, as an objective, may not be easy to achieve. But as an initiative, it can be a driver for significant delivery improvements and transformational results.

#### **Cost Effective**

Handling big data should not be a burden for SMB’s. Dataleyk offers flexible pricing options and only requires you to pay for what is used.

## **How Dataleyk Works?**

#### **Getting Started**

Getting started with Dataleyk is as simple as clicking the ‘free trial’ button. You can create an account with 30 days free trial period to test our value. Pricing is transparent and easy to understand, allowing SMBs to quickly get started.

#### **Import Data**

Simply point the repository to the platform and Dataleyk will get to work. Dataleyk enables syncing the historical data from a wide variety of sources, including streaming data over REST API.

#### **Build Model**

Dataleyk offers a user-friendly interface for building model. Once the data is ingested and processed, simply make a model with support of our data experts based on the raw data from different sources with no stress or hassle.

#### **Gain Insights**

Our end-to-end data platform does not stop at data processing. Users also have access to visualize results from Dataleyk’s AI powered graphs that convert numbers to insights or easy integrations with popular BI tools right out of the box.

**Dataleyk Funding & Investment**

Dataleyk has not raised any funding rounds yet.

## **Core Values**

Dataleyk shares a fundamental set of core values which sets the working principles of our company.

#### **Customer Commitment**

We believe in always putting the customer first, to enable genuine development and adaptation for their unique market requirements.

#### **Strive to be better**

Understand and then solve. Without communication or mutual feedback, we cannot bring true data driven solutions to our customers.

#### **Integrity**

Fairness in all aspects of business. We expect the same from all parties we do business with.

**Mission**

Make Big Data Analytics easy and accessible to all.

**Dataleyk Competitors**

Dataleyk’s competitors include

* [ThoughtSpot](https://tracxn.com/d/companies/thoughtspot/__j0eT_Leq8_wjW3vvd9FqpizTJQ8ZdOcLZdd38ouF0fs)
* [Tableau](https://tracxn.com/d/companies/tableau/__mE9oz1qtnZEh2yO7sS-NGHsGc2E2hbPiW2YrKqYRQoE)
* [Grafana](https://tracxn.com/d/companies/grafana/___YLnv1__Y26usD-QBbRWcBhyQ87C4VBrvKqkMV7TkUQ)
* [Devo](https://tracxn.com/d/companies/devo/__DpgWmAQnVELpjuouR3tlBubu5w74nVD1jrynuPS4mfM)

Dataleyk ranks 180th among 550 active competitors.

**How SMBs are Benefiting from Dataleyk**

#### **Versatile Database**

Dataleyk can ingest data from a wide variety of sources. It is equipped to extract semi-structured, binary data and even structured data from relational databases. Our platform can handle data in whatever form it comes in and from multiple sources.

#### **Data Analytics Hub**

While Dataleyk offers initial AI-powered graphs for quick analysis, our platform can also easily integrate to your favorite BI tool. Simply connect Dataleyk to Tableau (or one of your choosing) and visualize your clean, processed data all in a centralized location.

#### **Extreme Scalability**

Our user-friendly interface makes data ingestion faster and more efficient. Dataleyk is equipped to handle terabyte-scale data and ingest billions of data points. We also are powered by Hetzner Cloud, meaning we are readily scalable and highly available.

#### **Transparent Pricing**

Dataleyk’s transparent pricing ensures no billing surprises even as you scale. Reduce costs by up to **%70** with our incredibly fast and highly reliable platform.

#### **Security of the Cloud**

All the data get encrypted before storing. You can manage restrictions with advanced access control to selective data, and secure PII data with advanced hashing.

### **Project References**

* [https://Dataleyk.io/](https://cognera.io/)
* <https://www.crunchbase.com/organization/dataleyk>



**Introduction :**

Terra is an open source storage management and automation project. Terra is a universal SDS(software-defined storage) controller for connecting storage to Kubernetes, OpenStack, and VMware environments.

SDS controller software provides storage access services, networking, and connectivity. The most important characteristic of SDS controller software is that it makes no assumptions about the capacity or usefulness of the underlying hardware.

SDS is a software controller that manages and virtualizes your physical storage. As a result, you can control where and how data is stored. A software-defined storage controller provides networking, storage access, and connectivity services. The SDS controller does not rely on the usefulness or capacity of hardware.

**Project Summary :**

|  |  |
| --- | --- |
| **Website** | https://www.sodafoundation.io/projects/terra/ |
| **Organization** | Soda Foundation |
| **License** | Apache License 2.0 |
| **Open/Proprietary** | Open |
| **Source Path** | https://github.com/sodafoundation/controller |
| **Brief Description** | Terra is a universal SDS(software-defined storage) controller for connecting storage to Kubernetes, OpenStack, and VMware environments. |

Table 2:Terra summary

**Architecture :**

Graphical user interface

Description automatically generated

**Use Cases** : FIG 4:Terra architecture diagram

* Storage Provisioning: automate block and file storage provisioning
* Storage-as-a-Service (STaaS): self-service catalog empowers users and reduces OPEX **Features :**
* Standardized API, Controller for metadata and Dock for Drivers to provide seamless data management across various storage vendors
* Supports to connect different platforms like Kubernetes, Open Stack, VMware through plugins
* Supports custom vendors drivers and CSI plugins for heterogeneous storages

**Community Channels**

1. Website: <https://sodafoundation.org/>
2. GitHub: <https://github.com/sodafoundation/>
3. Discord: <https://discord.gg/KjzgswS/>
4. Twitter: <https://twitter.com/SodaFoundation/>
5. Telegram: <https://t.me/SodaFoundation/>
6. Slack: <http://www.sodafoundation.io/slack/>

**Project Organization :**

Graphical user interface, website

Description automatically generatedTerra are the projects from Soda Framework. Como and Terra are a part of the foundation. Soda Foundation is a non-profit organization. SODA Foundation is an open source project under Linux Foundation that aims to establish an open, unified, and autonomous data management framework for data mobility from the edge, to core, to cloud.

FIG 5: Soda foundation org diagram

**SODA Foundation Chart**

SODA (SODA Open Data Autonomy) is evolving to realize a challenging goal of building a unified framework for data and storage management. It connects the application platforms and solutions to the backend storages seamlessly, be it on prem or cloud through a unified API layer. This enables the application platforms to focus on building more valuable use cases rather than worrying about managing the underlying storage backends and data management. SODA Foundation India group is to address the storage and data management challenges in the industry and provide an open and unified “One Data Framework”.

**SODA Foundation Chart**

Graphical user interface

Description automatically generated

FIG 6: soda foundation chart

**SODA Foundation Projects :**

A picture containing logo

Description automatically generated

### 

FIG 7:Soda foundation Projects

## Background pattern Description automatically generated

## 

## 

## **Volcano**

## **INTRODUCTION:**

Volcano is a cloud native system for high-performance workloads, which has been accepted by [Cloud Native Computing Foundation (CNCF)](https://www.cncf.io/) as its first and only official container batch scheduling project. Volcano supports popular computing frameworks such as [Spark](https://spark.apache.org/), [TensorFlow](https://tensorflow.google.en/), [PyTorch](https://pytorch.org/), [Flink](https://flink.apache.org/), [Argo](https://argoproj.github.io/), [MindSpore](https://www.mindspore.cn/en), and [PaddlePaddle](https://www.paddlepaddle.org.cn/). Volcano also supports scheduling of computing resources on different architecture, such as x86, Arm, and Kunpeng.

### 

### **Project Summary**

|  |  |
| --- | --- |
| **Website** | https://volcano.sh/en/ |
| **Organization/Foundation Name** | Volcano was accepted to [Cloud Native Computing Foundation](https://www.cncf.io/) (CNCF) on April 9, 2020 |
| **License** | Apache License version 2.0 |
| **Open/Proprietary** | Open Source |
| **Source Path (if open source)** | https://github.com/volcano-sh/volcano |
| **Brief Description** | Volcano is system for running high-performance workloads on Kubernetes. It features powerful batch scheduling capability that Kubernetes cannot provide but is commonly required by many classes of high-performance workloads, including:   * Machine learning/Deep learning * Bioinformatics/Genomics * Other big data applications |

Table 3:Volcano Summary

### **Project Details**

#### **Key Features**

### **Rich scheduling policies**

Volcano supports a variety of scheduling policies:

* Gang scheduling
* Fair-share scheduling
* Queue scheduling
* Preemption scheduling
* Topology-based scheduling
* Reclaim
* Backfill
* Resource reservation

You can also configure plug-ins and actions to use custom scheduling policies.

### **Enhanced job management**

You can use enhanced job features of Volcano for high-performance computing:

* Multi-pod jobs
* Improved error handling
* Indexed jobs

### **Multi-architecture computing**

Volcano can schedule computing resources from multiple architectures:

* x86
* Arm
* Kunpeng
* Ascend
* GPU

### **Faster scheduling**

Compared with existing queue schedulers, Volcano shortens the average scheduling delay through a series of optimizations.

#### **Architecture**

## image**Overall Architecture**

Fig 8: Volcano architecture 1

#### **Application scenarios of Volcano**

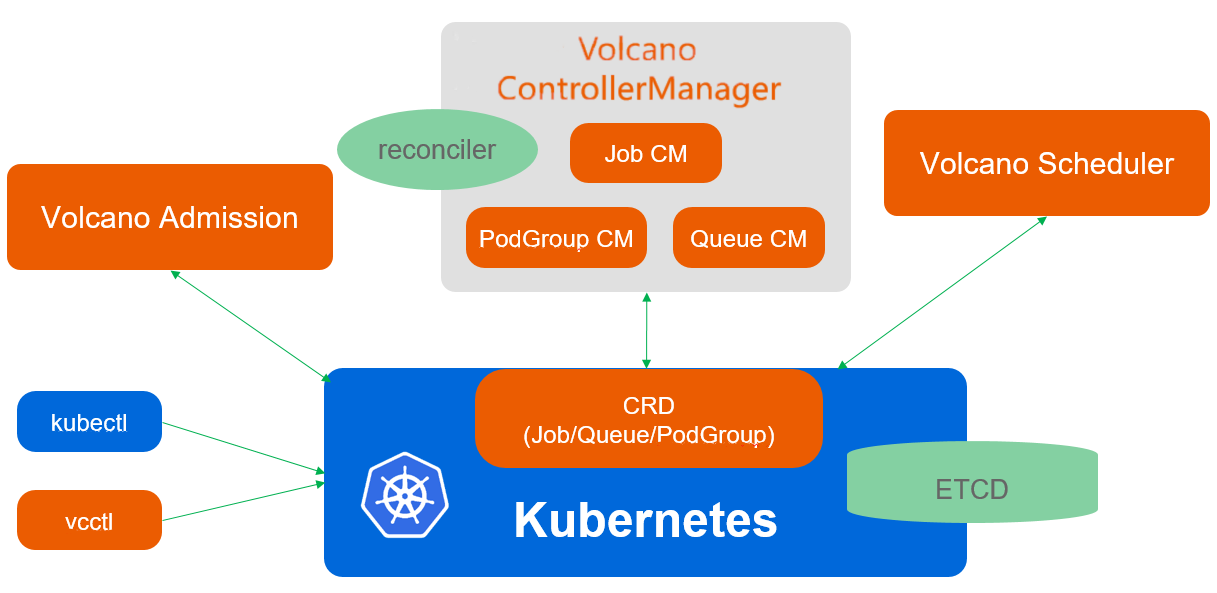
Volcano is designed for high-performance workloads running on Kubernetes. It follows the design and mechanisms of Kubernetes.

Fig 9: Volcano architecture 2

#### **Volcano architecture**

Volcano consists of **scheduler** / **controller manager** / **admission** / **vcctl**:

##### **Scheduler**

Volcano Scheduler schedules jobs to the most suitable node based on actions and plug-ins. Volcano supplements Kubernetes to support multiple scheduling algorithms for jobs.

##### **Controller Manager (CM)**

Volcano CMs manage the lifecycle of Custom Resource Definitions (CRDs). You can use the **Queue CM, PodGroup CM**, and **VCJob CM.**

##### **Admission**

Volcano Admission is responsible for the CRD API validation.

##### **vcctl**

Volcano vcctl is the command line client for Volcano.

## **Ecosystem**

Volcano allows you to use mainstream computing frameworks.

Volcano has been commercially used as the infrastructure scheduling engine by companies and organizations.

### **Kubeflow on Volcano**

Volcano is an enhanced high performance computing task batch processing system built on Kubernetes. As a platform for high performance computing scenarios, it makes up for Kubernetes’ lack of basic capabilities in machine learning, deep learning, HPC, and big data computing scenarios, including gang-schedule scheduling capability, computational task queue management, task-topology, and GPU affinity scheduling. In addition, Volcano has enhanced the batch creation and life cycle management of computing tasks, fair-share, binpack scheduling and other aspects on the basis of the native Kubernetes capability. Volcano has fully solved the problem of distributed training in Kubeflow.

# **Flink on Volcano**

Apache Flink is an open-source streaming framework developed by the Apache Software Foundation. At its core, Apache Flink is a distributed streaming data streaming engine written in Java and Scala. Flink executes any stream data program in data parallelism and pipelining. Flink’s pipelined runtime system can execute both batch and stream programs. In addition, the Flink runtime itself supports the execution of iterative algorithms.

# **MindSpore on Volcano**

MindSpore is a new generation of deep learning framework launched by Huawei. It is derived from the best practices of the whole industry, which best matches the computing power of the rise processor, supports the flexible deployment of the whole scene of terminal, edge and cloud, and creates a brand new AI programming paradigm and lowers the threshold of AI development.

### **Spark on Volcano**

Currently, there are two ways to support the integration of Spark on Kubernetes and volcano. - Spark on Kubernetes native support: maintained by the [Apache Spark community](https://github.com/apache/spark) and Volcano community - Spark Operator support: maintained by the [GoogleCloudPlatform community](https://github.com/GoogleCloudPlatform/spark-on-k8s-operator) and Volcano community.

# **Volcano Roadmap**

## **v1.0(Released on July 8, 2020)**

The major target of this release to make Volcano more stable for product.

Stability and Resilience

Improve test coverage. In v1.0, more test cases will be added to improve Volcano stability for product.

### **Preemption/Reclaim Enhancement**

Preemption and Reclaim are two import features for resource sharing; there're two actions for now, but unstable. In v1.0, those two features are going to be enhanced for elastic workload, e.g. stream job, bigdata batch job.

### **GPU Share (**[**#624**](https://github.com/volcano-sh/volcano/issues/624)**)**

A better performance has its cost, including GPU; and there are several scenarios that a Pod can not consume one GPU, e.g. inference workload, dev environment. One of solutions is to support GPU share, including related enhancement to both scheduler and kubelet.

### **Integrate with Apache Flink**

Flink is a widely used for Stateful Computations over Data Streams, but flink on kubernetes has some gaps now.

### **Integrate with argo to support job dependencies**

Investigate to cooperate with argo to support job dependencies.

### **Support running MindSpore jobs**

[MindSpore](https://www.mindspore.cn/) is a deep learning training and inference framework, support running MindSpore training with volcano job.

## **v1.2(Released on Feb 27, 2021)**

### **Queue Resource Reservation(Delay)**

* Description: Support reserve specified resource for queue without restart Volcano.
* Priority: High
* Issue: [#1101](https://github.com/volcano-sh/volcano/issues/1101)

### **Fair Scheduling For Jobs Of Same Priority And Different Queue**

* Description: Schedule jobs of same priority but from different queue accord to create time.
* Priority: High
* Issue: [#1098](https://github.com/volcano-sh/volcano/issues/1098)
* Owner: @alcorj-mizar

### **Differentiated Scheduling Strategies For Different Queue**

* Description: Support configure actions and plugins for different queues.
* Priority: High
* Issue: [#1035](https://github.com/volcano-sh/volcano/issues/1035)
* Owner: @sresthas

### **Support Hierarchy Queue(Delay)**

* Description: Support Hierarchy Queue algorithm.
* Priority: High
* Issue: [#1033](https://github.com/volcano-sh/volcano/issues/1033)
* Owner: @My-pleasure

### **Job PriorityClassName Update**

* Description: Support update vcjob priorityClassName update when job has not been scheduled.
* Priority: Middle
* Issue: [#1097](https://github.com/volcano-sh/volcano/issues/1097)
* Owner: @merryzhou

### **Status Message Enhanced For CRD(Delay)**

* Description: Provide more status detail for CRD status when use CLI such job fail reason.
* Priority: Middle
* Issue: [#1094](https://github.com/volcano-sh/volcano/issues/1094)
* Owner:@mikechengwei

### **Support MinAvailable For Task**

* Description: Support MinAvailable for task
* Priority: Middle
* Issue: [#988](https://github.com/volcano-sh/volcano/issues/988)

## **v1.3(Released on May 27, 2021)**

### **Task-Topology**

* Description: Support task topology scheduling
* Priority: High
* Issue: [#1349](https://github.com/volcano-sh/volcano/issues/1349)

### **Support multiple scheduler**

* Description: Support multiple scheduler by admission controller.
* Priority: High
* Issue: [#1322](https://github.com/volcano-sh/volcano/issues/1322)
* Owner: @Thor-wl @zen-xu

### **Stability and Resilience**

* Description: Improve the UT/E2E test coverage and add the stress test to improve stability.
* Priority: High
* Issue: [#1284](https://github.com/volcano-sh/volcano/issues/1284)
* Owner: @rudeigerc

### **Volcano Device Plugin enhancement**

* Description: Support container using multiples GPU as well as part of GPU card.
* Priority: High
* Issue: [volcano-sh/devices#12](https://github.com/volcano-sh/devices/issues/12)
* Owner: @peiniliu

## **v1.4(Released on Sep 18, 2021)**

### **Support NUMA-Awareness scheduling in Volcano**

* Description: Support NUMA-Awareness scheduling in Volcano.
* Priority: High
* Issue: [#1707](https://github.com/volcano-sh/volcano/issues/1707)
* Owner: @huone1 @william-wang

### **Support multi-scheduler by admission controller**

* Description: Use default scheduler for system daemon and Volcano scheduler for biz workload.
* Priority: High
* Issue: [#1322](https://github.com/volcano-sh/volcano/issues/1322)
* Owner: @huone1 @william-wang

### **Support scheduling with proportion of resources**

* Description: Add scheduling policy with proportion of resources.
* Priority: Middle
* Issue: [#1368](https://github.com/volcano-sh/volcano/issues/1368)
* Owner: @king-jingxiang

### **Enhance the resource comparison functions for various of scenarios**

* Description: Improve the Fundamental comparison functions
* Priority: Middle
* Issue: [#1525](https://github.com/volcano-sh/volcano/issues/1525)
* Owner: @Thor-wl

### **System Stability Enhancement**

* Description: Add UT/E2E to cover more scenarios and add basic stress test.
* Priority: Middle
* Issue: [#1284](https://github.com/volcano-sh/volcano/issues/1284)
* Owner: @rudeigerc

## **v1.5(Released on Feb 20, 2022)**

### **Support Hierarchy Queue(Delay)**

* Description: Support Hierarchy Queue algorithm.
* Priority: High
* Issue: [#1033](https://github.com/volcano-sh/volcano/issues/1033)
* Owner: @Thor-wl

### **Support Volcano scheduler in Spark community(Delay)**

* Description: Support Volcano scheduler in Spark community.
* Priority: High
* Issue: [#1704](https://github.com/volcano-sh/volcano/issues/1704)
* Owner: @william-wang @Yikun

### **Monitoring: Cluster Resource(Delay)**

* Description: Support monitoring metrics at cluster level
* Priority: Middle
* Issue: [#1586](https://github.com/volcano-sh/volcano/issues/1586)

### **Task Dag scheduling**

* Description: Support Dag for task level
* Priority: High
* Issue: [#1627](https://github.com/volcano-sh/volcano/issues/1627)
* Owner: @hwdef @shinytang6 @Thor-wl

### **Support configuration hot update(Delay)**

* Description: Add hot update for Volcano components arguments.
* Issue: [#1326](https://github.com/volcano-sh/volcano/issues/1326)

## **v1.6(To Be Released around May 15, 2022)**

### **Support Rescheduling Based on Realtime Load**

* Description: Support rescheduling based on realtime load.
* Priority: High
* Issue: [#1777](https://github.com/volcano-sh/volcano/issues/1777)
* Owner: @Thor-wl

### **Support Elastic Scheduling**

* Description: Support elastic scheduling for workloads.
* Priority: High
* Issue: TO BE ADDED
* Owner: @qiankunli @Thor-wl

## **Later (To be updated)**

### **Monitoring:**

* Description: Support monitoring metrics at queue and job level
* Priority: Middle
* Issue: [#1586](https://github.com/volcano-sh/volcano/issues/1586)
* Owner: @yanglilangqun @Tammy-kunyu

### **Improve resource calculation accuracy**

* Description: Support high accurate resource calculation.
* Issue: [#1196](https://github.com/volcano-sh/volcano/issues/1196)

### **Support job backfill**

* Description: Add backfill functionality to improve the resource utilization.

### **Improve the Autoscaling efficiency**

* Description: Combine the Autoscaler and scheduler to improve the scaling efficiency.

Graphical user interface

Description automatically generated**Volcano Contributors**

FIG 10: volcano contributers

# **Volcano Maintainers**

|  |  |  |
| --- | --- | --- |
| **Maintainer** | **GitHub ID** | **Affiliation** |
| Klaus Ma | [k82cn](https://github.com/k82cn) | Huawei |
| Kevin Wang | [kevin-wangzefeng](https://github.com/kevin-wangzefeng) | Huawei |
| Zhonghu Xu | [hzxuzhonghu](https://github.com/hzxuzhonghu) | Huawei |
| Thor-wl | [Thor-wl](https://github.com/Thor-wl) | Hjmicro |
| William-wang | [william-wang](https://github.com/william-wang) | Huawei |
| Liang Tang | [shinytang6](https://github.com/shinytang6) | Baidu |

Table 4: Volvano Maintainers

### 

### **Project References**

* <https://github.com/volcano-sh>
* <https://volcano.sh/en/#home_slider>
* <https://volcano.sh/en/docs/>

# **How to contribute to Open Source?**

Graphical user interface, text, application

Description automatically generated

Open-Source Software is a type of software whose code is publicly available to use and modify. Open-Source Contribution involves contributing to the development or improvement of open-source software. This is where open source contribution comes in. Contributors from around the world help develop and improve the software for every FIG 11:contribution open source

one of us who use it. Being a contributor will give you the superpower to be a part of something that is impacting so many lives.

Apart from the impact that you get to create, it also helps you become a better developer and with time a good mentor, leader, and passionate team player.

If you have already done some projects, open-sourcing your projects might be a good way to get started. Put it on GitHub and seek contributions from the community. This will not only add value to your project but will also help you to collaborate with many developers around the world.

Many widely-used frameworks and libraries were open-sourced by individual developers. Several people started collaborating and maintaining these projects after the projects got popular.

**Here are some steps :**

**Choose a project:** Find a project that you're interested in and that aligns with your skillset. There are many websites that list open source projects, such as GitHub, GitLab, and Bitbucket.

**Read the documentation:** Before you start contributing, read the project's documentation to understand its purpose, structure, and coding standards.

**Start small:** Don't try to tackle a large feature or bug fix right away. Instead, start with a small issue or improvement that you can handle.

**Fork the repository:** Create a copy of the project on your own GitHub account, and make changes to your forked repository.

**Make changes and submit a pull request:** Once you've made your changes, submit a pull request to the project's repository. Be sure to include a detailed description of your changes and any relevant documentation.

**Communicate with the community:** Engage with the project's community by joining mailing lists, forums, and chat rooms. Ask questions, seek feedback, and be respectful.

**Be patient:** Remember that open source projects are often run by volunteers, so it may take some time to get feedback on your contributions.

### **Ways to Contribute**

Diagram, venn diagram

Description automatically generated

There are many ways to contribute to open source projects, even if you're not a coding expert. Here are some ways to get involved

**Code contributions:**

One of the main ways to contribute to open source is by writing code. This can involve fixing bugs, implementing new features, or improving existing code. Many open source projects use version control systems, such as Git, to manage code contributions. This allows developers to easily collaborate on code changes, review each other's code, and track changes over time. FIG 12: ways to contribute

**Testing and bug reporting:**

Another important way to contribute to open source is by testing and bug reporting. Developers rely on users to test their software and report any issues they encounter. This feedback is crucial in identifying and fixing bugs, and improving the overall quality of the software.

**Documentation:**

Documentation is also a key aspect of open source software. Good documentation helps users understand how to use the software, and how to contribute to the project. Documentation can include user manuals, developer guides, and technical specifications.

**Translation:**

Translation is another important way to contribute to open source. By translating the software and documentation into other languages, you can help make the software accessible to a wider audience.

**Design and user experience:**

Design and user experience are also important aspects of open source software. Good design can make software easier to use and more visually appealing. If you have design skills, you can help by creating icons, graphics, or improving the user interface.

**Community building:**

Building and growing the community around the project is also an important contribution. This can involve answering questions, mentoring new contributors, and organizing events. By building a strong community, the project can attract more contributors and users, and continue to grow and improve over time.

**Marketing and promotion:**

Marketing and promotion are also important for open source projects. By creating blog posts, videos, or social media posts, you can help promote the project and attract new contributors and users.

## 

## **Methods to join the community and start contributing**

Joining an open source community and making your first contribution can seem daunting, but there are many ways to get started. Here is a step-by-step approach with tips for joining an open source community and making your first contribution.

**Step 1: Find a project**

The first step is to find a project that interests you and aligns with your skills. There are many resources for finding open source projects, including GitHub, GitLab, and Bitbucket. You can also search for open source projects related to your field or interests.

**Tip:** Look for projects with active communities and good documentation. This can help you get started and connect with other contributors.

**Step 2: Explore the project**

Once you have identified a project, explore its website, documentation, and code. This can help you understand the project's purpose, structure, and coding standards.

**Tip:** Take some time to read the project's documentation, code of conduct, and contributing guidelines. This can help you get a sense of the project's values and expectations.

**Step 3: Introduce yourself**

Join the project's communication channels, such as mailing lists, forums, or chat rooms, and introduce yourself. You can tell the community about your skills, interests, and goals.

**Tip:** Be respectful and professional in your communication. Avoid asking questions that have already been answered, and try to contribute to ongoing discussions.

**Step 4: Start small**

Don't try to tackle a large feature or bug fix right away. Instead, start with a small issue or improvement that you can handle.

**Tip:** Look for issues labeled as "beginner-friendly" or "good first issue." These are typically smaller tasks that are a good fit for new contributors.

**Step 5: Fork the repository**

Create a copy of the project on your own GitHub account, and make changes to your forked repository.

**Tip:** Make sure to follow the project's contribution guidelines, such as coding style and formatting, and write clear commit messages.

**Step 6: Submit a pull request**

Once you have made your changes, submit a pull request to the project's repository. Be sure to include a detailed description of your changes and any relevant documentation.

**Tip:** Be patient, as it may take some time to get feedback on your pull request. If you don't hear back after a week or two, you can follow up politely.

**Step 7: Get feedback and iterate**

Once you have submitted your pull request, be prepared to receive feedback from other contributors. Take the feedback into account, and iterate on your changes as needed.

**Tip:** Be open to feedback, and don't take it personally. Remember that the goal is to improve the software, and that everyone is working towards the same goal.

**Step 8: Continue contributing**

Once you have made your first contribution, continue contributing to the project. Look for more issues to work on, engage with the community, and help others get started.

**Tip:** Make sure to communicate with the community and ask for help if you need it. Don't be afraid to ask questions, and be respectful of others' time and expertise

**Community Engagement Experience**

Throughout my training, I had the privilege of engaging with the open-source community in various meaningful ways. One of the most notable examples of my community engagement experience was participating in regular community meetings hosted by the foundation. These meetings provided an opportunity for contributors to share updates on their work, discuss pressing issues, and collaborate on new ideas. Through these meetings, I was able to connect with other members of the community and gain valuable insights into the project's development.

Another essential communication channel within the SODA Foundation was Slack channels. These channels enabled contributors to engage with one another on a more casual level, ask questions, share ideas, and seek support. The Slack channels proved to be an excellent platform for building strong relationships and fostering a sense of community within the project.

I also had the chance to participate in several community events, such as conferences and meetups, which were organized or supported by the foundation. These events provided an excellent opportunity to promote the project and engage with stakeholders. By attending such events, I learned about the latest trends in open-source software development and gained insights into how other projects were approaching similar challenges.

The SODA Foundation also provided online forums where contributors could discuss technical issues, collaborate on development projects, and share their knowledge with one another. The forums were a great resource for learning and growing within the community.

Finally, the foundation's commitment to comprehensive documentation and tutorials was another way that I engaged with the community. The extensive resources provided by the foundation helped me understand the project's features and capabilities and get started with the project more efficiently.

Overall, community engagement is an essential part of the SODA Foundation's approach to open-source development. By fostering a collaborative and inclusive community, the foundation can continue to develop and improve its data management and storage tools for the benefit of all stakeholders**.**

**Community Engagement Experience**

**Soda Design Meeting**

Date:17-01-23

Meeting time: 12:30 - 2:00

Meeting Agenda: Soda Design

Organizer: Sanil Kumar

Soda design meeting was organised for the department of computer science by soda foundation. The meeting was meeting was held on JAN 17 organized and hosted by Sanil kumar.

It was organized for the detailed instructions about the project architecture and design. Each project maintainer gave a detailed instructions about various project of soda such as Terra, Delfin, Como etc.

Further the session was carried out by “Pravin Rajan” who gave a brief introduction to Strato Roadmap, uses of GitHub, slack etc. Maintainers briefed the project progress and its next plans ,various features of Roadmap 2023 H1 and H2 focus.

**Soda Global Community Meeting**

Date:03-02-23

Meeting time: 12:00 - 1:00

Meeting Agenda: Soda Global Community Meeting

Organizer: Sanil Kumar, Anvith K.S

Soda Global Community meeting was organized to provide Project updates of kahu delfin and Como. They explained about extending the drivers for the collection of metadata, AWS and manual syncing done in last iteration.

The community member interested in contributing could help with AWS driver testing and API’s. Later there was discussion based on Kahu new repository, Kahu operation, management and monitoring.

Later they explained bug fixing regarding ubuntu 2.0, data lake projects.

## 

## Diagram Description automatically generated**Contribution Flowchart:**

FIG 13: Contribution flowchart

**My Contributions**

During the course, I had the opportunity to contribute to an open-source project by creating documentation for the Dataleyk Project. To begin, I took the initiative to explore the Dataleyk Project by researching information related to the project.

After gathering relevant information from the Dataleyk website, I thoroughly analyzed the project's features, architecture, and technical aspects. Based on my analysis, I created comprehensive documentation that included all the necessary details. To ensure that the documentation was structured and presented effectively, I utilized the proper documentation techniques using stack edit.

Following the documentation creation, I forked the Architecture Analysis repository of SODA Foundation to my GitHub account. I then uploaded my documentation to the forked repository and committed the changes.

Once the changes were made, I created a pull request on GitHub and requested a review from the project's maintainers.

Overall, this experience allowed me to gain valuable knowledge and experience in contributing to open-source projects, such as the importance of proper documentation techniques and effective communication with project maintainers.

**A screenshot of a computer

Description automatically generatedHere is a screenshot of my pull request**

FIG 14:My contribution

**Open Source Value**

Open source contributions bring value to both individuals and communities in various ways. One of the main benefits is the collaborative approach to software development. Open source projects often involve developers from different parts of the world working together to improve the software, resulting in higher quality software that meets the needs of a wider range of users.

Another value of open source is transparency. The source code is freely available, which means that users can see how it works and make modifications if needed. This transparency builds trust in the software and ensures that it is free from hidden vulnerabilities or malicious code.

Flexibility is another key value of open source. The software is often more flexible than proprietary software, allowing users to customize it to meet their specific needs. This flexibility can be particularly valuable for businesses or organizations that have unique requirements.

Cost savings is also a significant value of open source software. Open source software is often free to use, which can result in significant cost savings compared to proprietary software. Additionally, since the source code is freely available, users are not locked into a specific vendor or licensing agreement.

Finally, open source can foster innovation by providing a platform for developers to experiment with new ideas and collaborate on new projects. This can lead to the creation of new technologies and solutions that would not be possible with proprietary software.

In summary, the value of open source lies in its collaborative, transparent, flexible, cost-effective, and innovative approach to software development. It provides users with the freedom and flexibility to use, modify, and distribute software as they see fit, leading to better quality software that meets the needs of a wider range of users.

**References**

* [https://Dataleyk.io/](https://cognera.io/)
* <https://www.crunchbase.com/organization/dataleyk>
* <https://github.com/volcano-sh>
* <https://volcano.sh/en/#home_slider>
* <https://volcano.sh/en/docs/>