# **SYNOPSIS**

The "Open Source Demystified Level 1" course is a comprehensive introduction to the world of open source software, designed for beginners who are curious about open source but have little to no prior experience or knowledge. The course is targeted towards individuals interested in understanding the concepts, principles, and practices of open source, including software developers, IT professionals, students, and technology enthusiasts. The course starts with an overview of open source, its history, and its importance in today's technology landscape. It then delves into the fundamentals of open source, covering topics such as the definition of open source, the concept of free software, open source licenses, and the benefits of using open source software. The course also explains the principles of open source development, including the concepts of collaboration, transparency, and community involvement. Next, the course explores popular open source tools and technologies, including operating systems, programming languages, databases, web servers, and content management systems. It provides practical examples and demonstrations of how to use and contribute to open source projects, including how to find, download, install, and customize open source software.

Open Source Demystified Level 1course 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 potentials and open source culture. This understanding helps one to identify, enter, contribute, learn and grow one’s competency and career. It also provides hands-on experience with the 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 the overall summary and guidance to contribute to open source projects further.

# **Glossary**

**KINETIC: -**

**Kinetics:** The study of how things change over time, including the rate and mechanism of reactions or processes.

**Kinetic data:** Data that captures changes in a system over time, such as changes in concentration, temperature, or other variables.

**Reaction rate:** The speed at which a chemical reaction occurs, often expressed as the change in concentration of a reactant or product over time.

**Rate constant:** A proportionality constant that relates the reaction rate to the concentrations of the reactants and other factors that affect the reaction.

**Reaction mechanism**: The step-by-step process by which a chemical reaction occurs, including the formation and breakdown of intermediate species.

**Activation energy:** The minimum amount of energy required for a reaction to occur, often represented as a barrier that must be overcome before the reaction can proceed.

**Half-life:** The time required for half of the reactants in a reaction to be consumed or for a substance to decay by half.

**Arrhenius equation:** A mathematical equation that relates the rate constant to the temperature and activation energy of a reaction.

**Reaction order:** The dependence of the reaction rate on the concentration of one or more reactants, often expressed as an integer value representing the power to which the concentration is raised in the rate equation.

**Reaction network:** A diagram or set of equations that represents the interconnected reactions and species involved in a chemical or biological process.

**TERRA:**

**Terra:** A cloud-based platform for scientific data analysis, developed by the Broad Institute.

**Workflow:** A series of steps or tasks that are performed to complete a specific analysis or project, often automated using software tools.

**Pipeline:** A specific type of workflow that involves the processing and analysis of data, often using a series of software tools or scripts.

**Containerization:** The process of packaging software and its dependencies into a container, which can be easily deployed and run on different systems or platforms.

**Docker:** An open-source platform for containerization, which allows users to create and deploy containers with their own software and configurations.

Kubernetes: An open-source platform for container orchestration, which allows users to manage and scale containers across multiple machines or nodes.

**Jupyter notebook:** An open-source web application for creating and sharing interactive documents that contain code, visualizations, and narrative text.

**Git:** An open-source version control system, which allows users to track changes to code and collaborate with others on software projects.

**GitHub:** An online platform for hosting and sharing code repositories, which is widely used for open-source software development.

**Continuous integration/continuous deployment (CI/CD):** A development practice that involves automating the building, testing, and deployment of software, often using a series of tools and scripts.

**ALIBABA CLOUD: -**

**Alibaba Cloud:** A cloud computing platform and service provider, offering a wide range of cloud-based products and services.

**Elastic Compute Service (ECS):** A cloud-based computing service that provides scalable virtual servers for running applications and services.

**Object Storage Service (OSS):** A cloud-based storage service that allows users to store and retrieve large amounts of data in a flexible and scalable manner.

**Relational Database Service (RDS):** A cloud-based database service that provides managed database instances for MySQL, SQL Server, and other relational database systems.

**Elastic MapReduce (EMR):** A cloud-based big data processing service that allows users to process and analyze large datasets using Hadoop and other big data tools.

**Kubernetes:** An open-source platform for container orchestration, which allows users to manage and scale containers across multiple machines or nodes.

**Virtual Private Cloud (VPC):** A cloud-based network service that allows users to create a private, isolated network within the Alibaba Cloud platform.

**Load Balancer:** A cloud-based service that distributes incoming network traffic across multiple servers or instances, improving the availability and scalability of applications and services.

Auto Scaling: A cloud-based service that automatically adjusts the number of instances or resources allocated to an application or service based on changes in demand or usage.

# **Introduction**

Open source refers to software or technology that is made available to the public with its source code openly accessible and modifiable. It is a collaborative and decentralized approach to software development that allows for free distribution, modification, and redistribution of the code. Open source software is often created and maintained by a community of developers, rather than a single company or organization.The concept of open source software originated in the 1980s and gained popularity in the 1990s with the rise of the internet. Open source software has become increasingly popular in recent years due to its flexibility, cost-effectiveness, and ability to promote innovation and collaboration. Open source technology has also been applied to areas outside of software development, such as hardware and design. Many companies, including Microsoft and Google, have embraced open source and contribute to open source projects. The use of open source software and technology is expected to continue to grow as more businesses and organizations realize its benefits.

## 

## **About this document :**

This document is divided into several sections, including an introduction, glossary, synopsis, community engagement experience, methods to join the community, ways to contribute, my contributions, open source value, and references. Each section provides detailed information on a particular aspect of the course or open source technology, and includes relevant examples, illustrations, and step-by-step guidance where appropriate. The layout of the document is designed to be easy to follow and understand. Each section is clearly labeled and organized, and key terms and concepts are defined in the glossary for easy reference. Additionally, the document includes visuals such as flowcharts, tables, and images to help illustrate complex concepts and provide additional context.

This document is designed to provide a comprehensive overview of open source contributions and their importance. It includes various sections, including an introduction to open source, ways to contribute, methods to join a community, community engagement experience, personal contributions, and the value of open source contributions. The document is structured in a way that allows readers to navigate easily and find the information they need. It includes headings and subheadings to help readers locate specific topics, and images and illustrations to enhance understanding.

## **Purpose**

The purpose of this document is to provide an overview of the Open Source Demystified Level 1 course and its contents. The document is intended to serve as a guide for individuals who are interested in learning more about open source technology and its ecosystem, as well as for those who are interested in contributing to open source projects. The document provides an overview of the key concepts and principles of open source, as well as practical guidance on how to get started with.

It aims to help individuals who are new to open source understand the various ways they can contribute, the benefits of contributing, and how to join open source communities. The document also aims to encourage individuals to contribute to open source projects by providing examples of the positive impact that contributions can have on both individuals and communities.

## **Audience**

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 of open source contributions.

# **Open Source Introduction**

Open source software is defined as **“software with source code that anyone can inspect, modify and enhance”.** This type of code is developed on the premise that all software source code should be free to access and belongs to the community. The source code is open to the world and does not require a fee. The open source approach results in better code because the software is made by many hands. Open source refers to software or other products that are developed and distributed under an open-source license, which allows users to view, modify, and distribute the source code freely. In general, open source is characterized by its transparency, collaboration, and community-driven development process, which often involves a large and diverse group of developers and users.Open source software can be used for a wide range of applications, from operating systems and database software to web applications and scientific tools. Some of the key benefits of open source include lower costs, increased flexibility, and greater innovation and collaboration.

Key terms and concepts in open source include:

**Open-source license:** A legal agreement that governs the use, distribution, and modification of open-source software or other products.

**Source code:** The human-readable instructions that make up a software program, which can be viewed and modified by developers.

**Forking:** The process of creating a new version of an open-source project by copying and modifying the source code.

**Contributions:** Changes or additions to an open-source project made by developers or other contributors.

**Community:** A group of developers and users who collaborate on an open-source project, often through online forums and other communication channels.

**Governance:** The processes and structures that guide the development and management of an open-source project, such as a governing board or steering committee.

**Distribution:** The process of making open-source software or other products available to users, often through online repositories or distribution channels.

**Commercial use:** The use of open-source software or other products for commercial purposes, such as in a business or enterprise context.

**Intellectual property:** The legal rights associated with the creation and distribution of software and other products, including copyrights, patents, and trademarks.

**Open standards:** Technical standards that are developed and maintained through an open and collaborative process, often to promote interoperability and compatibility between different software systems.

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

**Linux Operating System:** Linux is a free and open-source operating system, widely used for servers, desktops, and embedded systems.

**Apache HTTP Server:** Apache is a free and open-source web server software, used to host websites and web applications.

**Mozilla Firefox:** Firefox is a free and open-source web browser, developed by the Mozilla Foundation.

**WordPress:** WordPress is a free and open-source content management system, used to create websites and blogs.

**VLC Media Player:** VLC is a free and open-source media player software, used to play a wide range of audio and video formats.

**Android:** Android is a free and open-source mobile operating system, used on a wide range of smartphones and tablets.

**PostgreSQL:** PostgreSQL is a free and open-source relational database management system, used to store and manage data.

**Open Source Structure: -**

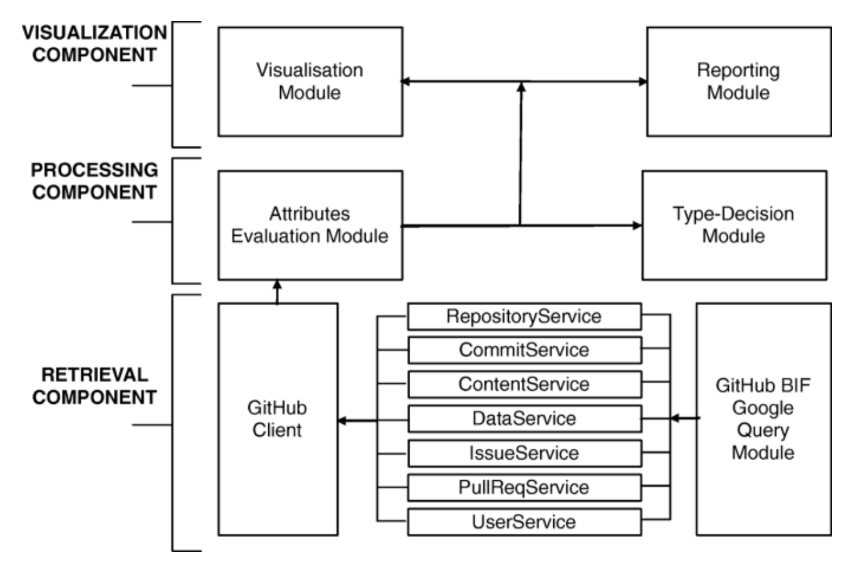


Figure 1: Open-Source Structure

**Open Source Project Examples**

1. Kinetic
2. Terra
3. Alibaba Cloud

### **KINETIC DATA**

### **Introduction: -**

### 

### Kinetic Data was built from the ground up for large, complex enterprise organizations that require a platform that is both flexible and scalable. Kinetic Data is a software company that offers a low-code platform for building enterprise service management applications. The platform allows organizations to automate and streamline their workflows, integrate their IT services, and improve their customer service. In other words, “kinetic data” in the context of the Kinetic Data software platform likely refers to the data related to the various business processes and workflows that the platform is designed to manage and optimize.The Kinetic Platform enables real change, so a time investment is required to learn the underlying philosophies and abstraction layers in order to be successful. Once understood, the power of the Kinetic Platform can be unleashed to digitize and transform all types of processes within a business. Kinetic Data has solved for the pains that you don’t know you have until you have them, such as:

### In-flight workflow versioning

### Transforming and manipulating data on-the-fly in-between workflow steps.

### Complex error handling and routing of errors to tools for automated or manual resolution.

### The ability to integrate with ANY system without waiting for the vendor to release an update or build it for you.

### Full control of the UI/UX within the end user experience to solve an accessibility issue.

### Dynamic security policies based on functions and variables, not just roles and groups.

### The ability to work within your existing CI/CD process.

### **Project Summary**

|  |  |
| --- | --- |
| Website | https://community.kineticdata.com/docs |
| Organization/Foundation Name | Kinetic community LF? |
| License | Apache License 2.0 |
| Open/Proprietary | Open |
| Source Path(if open source) | KineticCommunity/request-ce-bundle-kinetic-oscar2 (github.com |

Table 1: Kinetic Project Summary

**Brief Description: -**

Kinetic Data is a software company that offers a low-code platform for building enterprise service management applications. The platform allows organizations to automate and streamline their workflows, integrate their IT services, and improve their customer service. In other words, “kinetic data” in the context of the Kinetic Data software platform likely refers to the data related to the various business processes and workflows that the platform is designed to manage and optimize.

**Project Details**

#### **Key Features: -**

**Agent-** A portion of the Kinetic Platform that sits inside the firewall to execute calls that must take place internally for kinops systems that connect to on-prem systems (such as handlers and bridges). Agent replaced Kinetic Bridgehub and Kinetic Filehub. Agent is the harness that runs bridge adapters and file adapters. It can also be extended to run task handlers in a customer’s security context.**API-** A software intermediary that allows two applications to talk to each other.**Attribute-** User-defined metadata for Spaces, Kapps, Users, Forms, Categories, Datastore Forms, or Teams. Also sometimes used as the name for the defined values in a Bridge Model.**Bridge Mapping-** The location where the bridge data source is mapped to the bridge model properties. This tells the Kinetic Platform how to get the raw data from the data source, and how to map the source data names to data fields for Kinetic Bridge Models.**Connector-** The connection between two nodes within a workflow. May or may not contain logic. If logic is applied, the connector path is only followed if the attached logic statement evaluates to not false. Why “not false”? Your connector could return a “Name” of a person - and if a name exists we want that branch of the workflow to execute.**Kinetic App (Kapp)-** A collection of Forms within the Kinetic Request CE System. Kapps have their own metadata, categories, webAPIs, forms, and permissions. Typically a Kapp will represent a single end user experience. Example Kapps include a Service Portal or Work Order System.**Kinetic Security Language (KSL)-** A functional language created for writing and managing security policies. KSL is implemented as JavaScript code.**Workflow-** A tree or routine. A process that does the work for a given event. Sample events: Submission Created, User Defined, Form Modified, etc…

**Architecture: -**

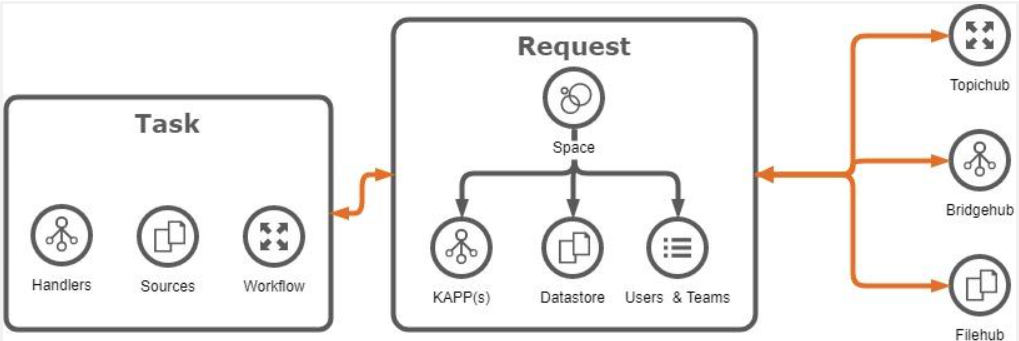


Figure 2: Architecture of Kinetic project

The Kinetic Platform consists of three primary components:Kinetic Core,Kinetic Task, andKinetic Agent.**Kinetic Core**Request is the Platform component that allows you to build powerful and beautiful front ends for your customers. It contains a drag-and-drop form editor where you can easily construct the forms you wish to present your users. It also supports the complexities of multi-system integrations, multi-tenancy, and workflow integration on all its components.You may see Kinetic Core referred to as Request CE, Request Core Edition, CE, or Core Edition. The platform is built with a version of Request (CE) that is entirely our own creation, unlike our previous versions (RE, or Remedy Edition) that relied on Remedy to be the platform.Request is the point of connection in the Platform where Task, Filehub, Bidgehub, and Request itself all come together to provide the customer with the optimal user experience.**Kinetic Task**Task is the Platform component that allows you to build and execute workflows. It contains a drag-and-drop workflow editor where you connect code snippets called handlers together to build flexible and powerful workflows. It also contains the engine that runs to execute these workflows and handle errors, providing fast workflow processing and robust error handling for all your workflow needs.There are hundreds of handlers already built and available for use. These can connect you with everything from your local Request system to your AWS provisioning service and everything in between. Also, if you desired integration has an API but no handlers currently available, there are instructions available for you to be able to build your own.While Task is a powerful workflow engine without the rest of the platform and can be used standalone, using Task with the Platform allows you to integrate that powerful workflow into your beautiful front-end experience for an optimal user experience.

**Kinetic Agent**Agent is the Platform component that provides the connectivity between the front end of the platform and other systems. This is what provides multi-system integration within the Kinetic Platform’s front ends.

#### **Current Usage: -**

Users are the primary way to identify and authorize systems and individuals within the Kinetic Platform. Users within Kinetic can be created manually by other users that have permissions to create new users, or automatically by configuring the platform to authenticate against an IDP (Identity Management Provider) using common protocols like SAML.Fundamentally, users must exist within the system in order to interact with areas of the system that require authentication. All users within Kinetic are scoped to a space, and no users are shared between spaces.

#### **Technical Details: -**

Hardware: -

Web Application Server (Minimum 2)CPU Cores - 4+RAM - 8+GBStorage - HDD or SSD - 500GB minimumOS - Ubuntu 16.04 LTSCassandra Database Server (Minimum 3)CPU Cores - 4+RAM - 8+GBStorage - Solid State Drive (SSD) - 1TB minimumOS - Ubuntu 16.04 LTS

Software: -

Database ServerJava Server JRE 8 or JDK 8Python 3 (used for cqlsh - a command line CQL tool)Apache Cassandra version 3.11Web ServerProduction Grade Relational Database (for Kinetic Task)PostgreSQL 10 (Microsoft SQL Server and Oracle are also supported)

This database does not need to be installed on the web server, it just needs to be accessible from the web server.

* Java Server JRE 8 or JDK 8
* Apache Tomcat Apache Tomcat 8.5 w/adjusted Java Heap size to 2 GB+
* Kinetic Request CE
* Kinetic Bridgehub
* Kinetic Filehub
* Kinetic Task
* Port Summary
* User Ports

**Other Information**

**Forms vs Datastore Form Comparison**There are a few key differences between regular (Kapp) Forms and Datastore Forms. The primary difference is in the way one queries for Submissions (or records created) of one of these forms.**Security: -**Security for platform components is handled by Security Policies. Security policies are something defined by a system admin and can be as simple or complex as necessary. These policies can incorporate a variety of information, including the user’s indentity, teams, roles, and more.Security policies can be used, in some cases, to determine access to the platform consoles themselves.**Creating Security Policies**: - Security Policies are created in the same manner, regardless of the level they are created for. You may have different information depending on the level, but creating the policy is the same.Rules are built (written) using KSL, a scripting language created for such things. See the related KSL article for more information on KSL and for sample security policies.Each Policy Rule console has the same fields:Name. Descriptive name for the ruleType. Drop-down list of options for the rule depending on where you are in the Request console (Space or Kapp).Message. Message presented if the rule resolves to FalseRule. KSL that resolves to True or False**Security Policy Precedence**: - Security policies are defined in different places, depending on where they are applied. If no policy is defined, then the policy is used from the next highest level.In level order:

* Form
* Kapp
* Space
* Consoles

In general, you need to be a Space Admin to have access to the consoles.If you allow a user access to Form Modification, they have access to the kapp the form is part of. They can see other consoles in the kapp, but cannot make changes. Each kapp also has a setting for kapp visibility and kapp modification. While kapp visibility is just to see the forms for the kapp, kapp modification allows a user to have access to the kapp consoles. kapp security settings are in the Settings console for the kapp, under the Security tab.**Form Security Policies**:- There are four options for Security Policies for Forms. They control form modification and visibility, and submision access and modification.Another option is to set the form to anonymous. This option is featured in another article because it’s not specifically a security option.Each of the four options has the same list of choices.

* Form Display.
* Form Modification.
* Subbmission Access.
* Submission Modification.

**Alternatives & Competitors to Kinetic: -**Kinetic stacks up to the competition, the reviews from current & previous users in industries like Machinery, Automotive, and Mechanical or Industrial Engineering, and find the best product for your business.

* **SAP ERP** software is a proven foundation for the world’s largest organizations. Streamline procurement, manufacturing, service, sales, finance, and HR processes.
* **NetSuite** is a cloud ERP solution, providing a suite of applications, from accounting and financial planning, to warehouse management, ecommerce, inventory management and beyond.
* **SYSPRO** is a global leader in the production of world-class ERP software, the company now caters to the specialized needs of over 15,000 licensed companies in more than 60 countries worldwide
* **Microsoft Dynamics 365** for Operations is the complete ERP solution for enterprises and offers solutions for manufacturing, distribution, retail, services, and public sector.
* **Acumatica** is a leading provider of Cloud ERP and Cloud Accounting software for mid-size businesses and non-for-profits.
* **SAP Business** One single, integrated solution to manage your entire business.

These are some of alternatives or competitors to Kinetic

**Reference Links: -** [docs.kineticdata.com](https://docs.kineticdata.com/docs)

[www.linkedin.com](https://www.linkedin.com/company/kinetic-data/)

[kineticdata.com](https://kineticdata.com/platform/)

**TERRA SDS CONTROLLER**

**Introduction: -**

Terra is a member project of the SODA Foundation, which is a Linux Foundation project focused on open-source data management and storage technologies. As a member project of the SODA Foundation, Terra is part of a larger community of open-source projects and contributors working together to advance data management and storage technologies.

In the context of the SODA Foundation, Terra is specifically focused on providing a cloud-native platform for deploying and managing distributed applications and services across multiple cloud providers. This aligns with the overall mission of the SODA Foundation, which is to promote the adoption and development of open-source data management and storage solutions, particularly in the areas of data mobility, data protection, and data control.

As a member project of the SODA Foundation, Terra benefits from access to a large and active community of open-source contributors, as well as the resources and support provided by the Linux Foundation. It also helps to promote collaboration and interoperability between different open-source projects in the data management and storage space.

**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 cloud-native platform that simplifies the deployment and management of distributed applications and services across multiple cloud providers. It provides a unified interface for managing resources and deploying workloads on public and private clouds |

Table 2: Terra Project Summary

**Project Details**

**Key Features:**

* **Multi-cloud management:** Terra enables users to manage data across multiple cloud environments, including public, private, and hybrid clouds. This helps organizations avoid vendor lock-in and gain flexibility in deploying their workloads.
* **Hybrid cloud deployment:** Terra allows for seamless integration between on-premises and cloud-based storage resources, providing a hybrid cloud infrastructure. This can help organizations optimize their storage costs and improve performance by leveraging the benefits of both environments.
* **Simplified application deployment:** Terra provides a unified interface for deploying applications across different storage environments, simplifying the deployment process and reducing complexity.
* **Scalability and flexibility:** Terra is designed to be scalable and flexible, allowing organizations to easily adapt to changing business requirements and storage needs.
* **Integration with Kubernetes:** Terra integrates with Kubernetes, a popular open-source container orchestration platform, enabling users to manage storage for their Kubernetes-based applications. This helps organizations streamline their storage management processes and improve application performance.

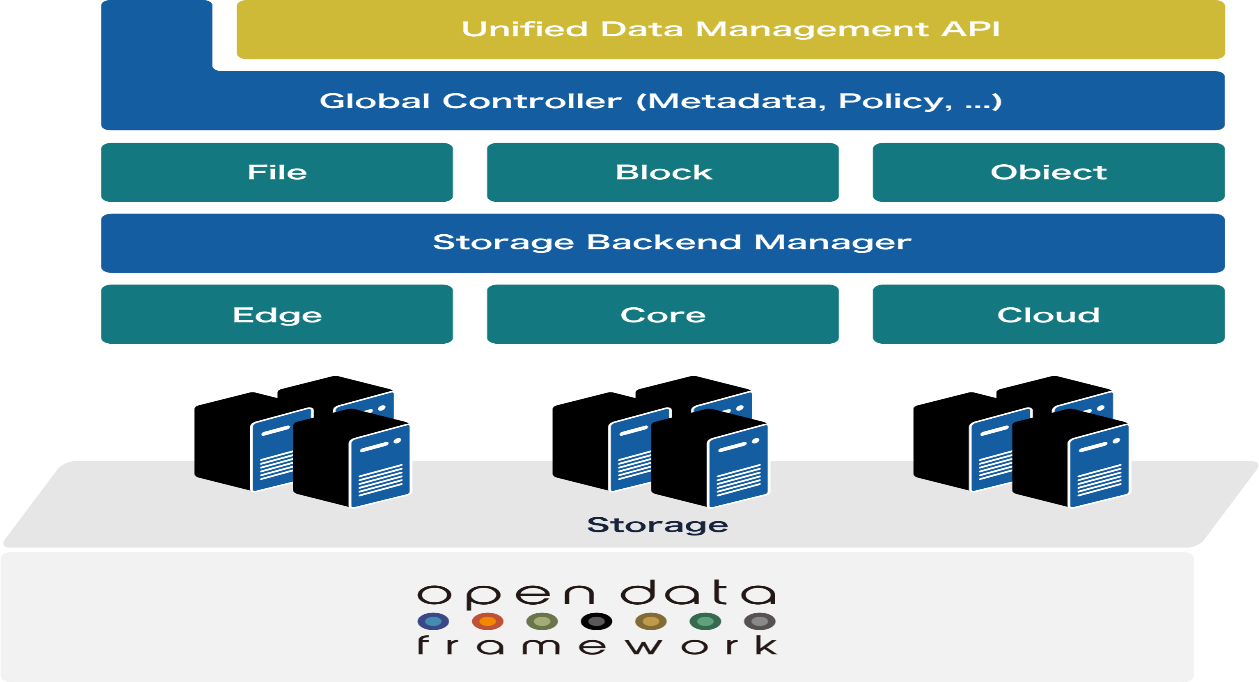
**Architecture:**

Figure 3: Architecture of Terra project

Terra is built on top of Kubernetes and leverages a number of open-source technologies, including Helm, Istio, and Prometheus. It uses a declarative approach to infrastructure management, allowing users to specify desired state configurations and automatically deploying and managing resources to achieve that state.

**Current Usage:**

Terra is used by a number of organizations to manage their cloud-based applications and services, including Tencent, China Mobile, and China Telecom.

The Terra project, which is part of the SODA Foundation, aims to provide a unified platform for managing data storage across multiple cloud environments. It is designed to simplify the deployment and management of applications and workloads in hybrid cloud environments, enabling organizations to optimize their storage resources and improve performance.

The project provides a range of features, including multi-cloud management, simplified application deployment, scalability and flexibility, and integration with Kubernetes. These features are intended to make it easier for organizations to manage their storage infrastructure and provide a seamless experience for their users.The goal of the Terra project is to provide a standardized, open-source platform for managing data storage in hybrid cloud environments, helping organizations to reduce complexity, improve performance, and lower costs.

**Technical Details:**

Terra is highly scalable and can be used to manage applications and services of any size. It provides automated scaling and self-healing capabilities, making it easy to manage large and complex workloads.

**Use Cases:**

* 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

**Project References:**

<https://terra.project.com>

<https://github.com/sodafoundation/terra>

<https://www.sodafoundation.io/projects/terra/>

<https://www.linuxfoundation.org/projects/soda-foundation/>

**ALIBABA CLOUD**

**Introduction: -**

Alibaba Cloud is a cloud computing service provided by Alibaba Group, a multinational technology company based in China. It offers a wide range of cloud products and services, including computing, storage, databases, networking, security, and analytics.

Alibaba Cloud, also known as Aliyun, is one of the world's leading cloud computing providers. It was established in 2009 as a subsidiary of Alibaba Group, and has since grown rapidly to become a global leader in cloud technology. Alibaba Cloud offers a wide range of cloud products and services, including infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) solution. Its products and services are designed to help businesses of all sizes to improve their agility, flexibility, and scalability, and to achieve greater efficiency and cost savings. With its state-of-the-art technology, advanced security features, and global network of data centres, Alibaba Cloud is trusted by millions of customers worldwide, from startups and small businesses to large enterprises and government organizations. Whether you need to run your applications in the cloud, store and manage your data, or analyse your business insights, Alibaba Cloud has the tools and expertise to help you

succeed.

**Project Summary**

|  |  |
| --- | --- |
| Website | https://www.alibabacloud.com/ |
| Organization/Foundation Name | Alibaba Group |
| License | Proprietary |
| Open/Proprietary | Proprietary |
| Source Path(if open source) | N\A |
| Brief Description | Alibaba Cloud is a secure, reliable, and scalable cloud platform that enables businesses to run their applications and services with ease and efficiency. |

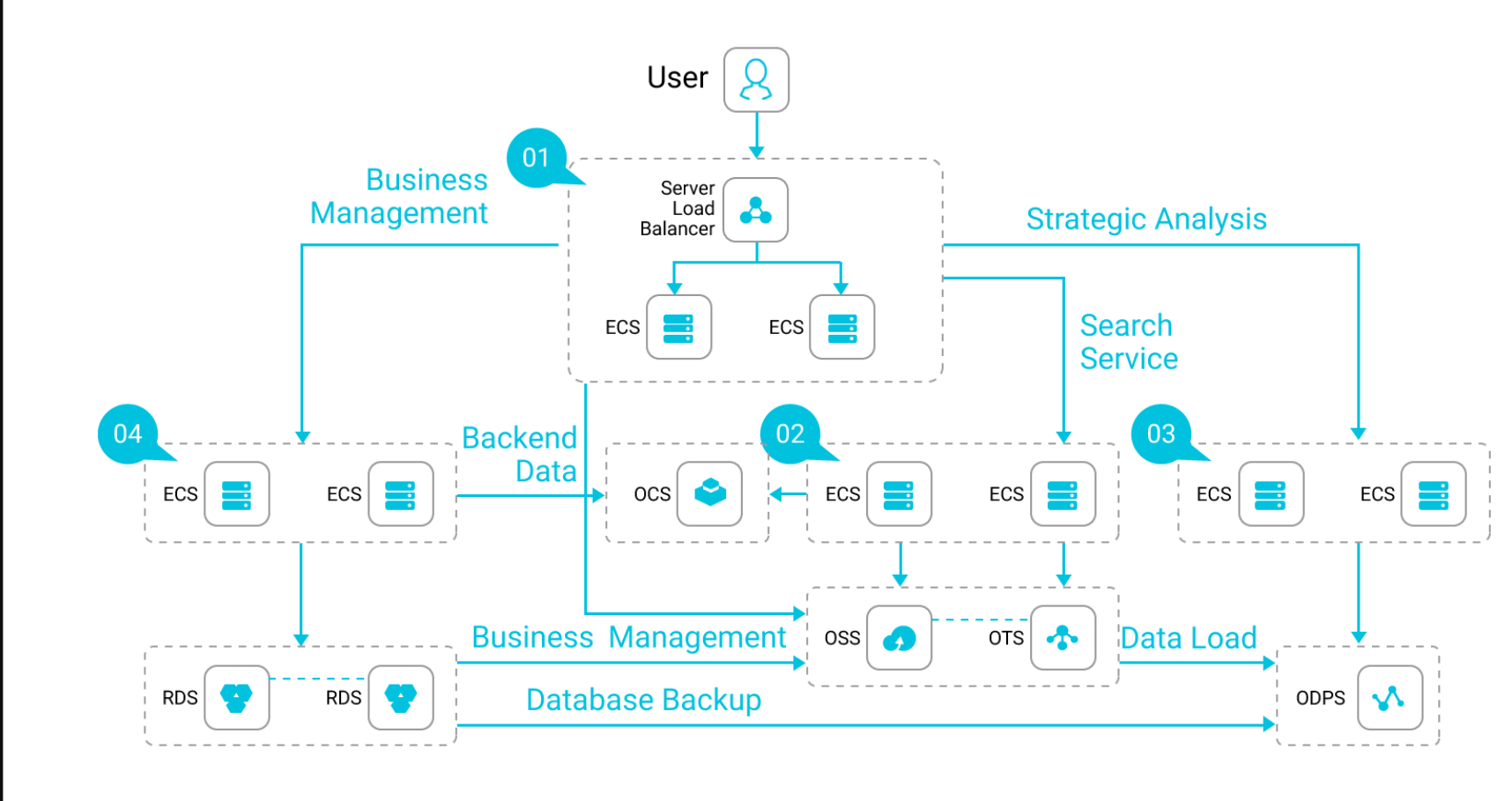
Table 3: Alibaba Cloud Summary

**Project Details**

**Key Features**

* Elastic Compute Service (ECS)
* Object Storage Service (OSS)
* Relational Database Service (RDS)
* Virtual Private Cloud (VPC)
* Content Delivery Network (CDN)
* Anti-DDoS Pro
* Security Center
* Data Works
* **Elastic Compute Service (ECS):** This is Alibaba Cloud's flagship computing service that enables users to create and manage virtual machines (VMs) in the cloud. ECS provides flexible and scalable computing resources that can be quickly deployed and customized to meet the specific needs of different workloads. It also offers a variety of features such as auto-scaling, load balancing, and security to ensure high availability and performance of the VMs.
* **Object Storage Service (OSS):** This is Alibaba Cloud's scalable and highly available object storage service that allows users to store and retrieve large amounts of data in the cloud. OSS supports various types of data such as images, videos, audio files, and documents, and provides features such as data replication, backup, and disaster recovery to ensure data durability and availability.
* **Relational Database Service (RDS):** This is Alibaba Cloud's managed database service that enables users to create and manage relational databases in the cloud. RDS supports various types of databases such as MySQL, SQL Server, PostgreSQL, and Oracle, and provides features such as automated backup, high availability, and disaster recovery to ensure data reliability and security.
* **Virtual Private Cloud (VPC):** This is Alibaba Cloud's private network service that enables users to create and manage isolated virtual networks in the cloud. VPC provides a secure and scalable network environment that can be customized to meet the specific needs of different workloads, and also allows users to connect their cloud resources with their on-premises infrastructure using VPN or dedicated connections.
* **Content Delivery Network (CDN):** This is Alibaba Cloud's globally distributed content delivery service that enables users to deliver content to their end-users with low latency and high availability. CDN provides a network of edge nodes that cache and distribute content across the globe, and also provides features such as SSL encryption, DDoS protection, and traffic control to ensure optimal performance and security of the content delivery.
* **Anti-DDoS Pro:** This is Alibaba Cloud's DDoS protection service that provides real-time protection against various types of DDoS attacks such as SYN flood, UDP flood, and HTTP flood. Anti-DDoS Pro provides features such as traffic scrubbing, blacklisting, and whitelisting to mitigate the impact of DDoS attacks and ensure the availability and performance of the online services.
* **Security Center:** This is Alibaba Cloud's security management service that provides centralized security management and monitoring for cloud resources. Security Center provides features such as vulnerability scanning, intrusion detection, and log analysis to help users identify and mitigate security risks in the cloud.
* **DataWorks:** This is Alibaba Cloud's big data processing and analytics platform that enables users to process, analyze, and visualize large amounts of data in the cloud. DataWorks provides features such as data integration, data processing, and data analysis to help users extract insights from their data and make informed decisions.

**Architecture: -**

Figure 4: Architecture of Alibaba Cloud

Alibaba Cloud's architecture is based on a multi-tenant, shared infrastructure model, where resources are dynamically allocated and optimized based on user demand. This enables users to scale up or down their resources quickly and easily, without having to worry about the underlying infrastructure.

The architecture also incorporates advanced technologies such as virtualization, containerization, and software-defined networking to provide a highly scalable and flexible computing environment. Virtualization enables the creation of multiple virtual machines (VMs) on a single physical server, while containerization enables the creation of lightweight, portable containers that can run on any infrastructure. Software-defined networking provides a highly flexible and programmable network infrastructure, enabling users to easily configure and manage their network resources.

Alibaba Cloud's architecture is also designed to be highly available and fault-tolerant, with multiple layers of redundancy and failover mechanisms. For example, the Elastic Compute Service (ECS) includes features such as auto-scaling, load balancing, and automatic failover, to ensure that applications and services are always available and responsive.

In terms of security, Alibaba Cloud's architecture includes multiple layers of protection against various types of threats, including distributed denial-of-service (DDoS) attacks, malware, and data breaches. For example, the Security Center provides centralized security management and monitoring for cloud resources, while the Anti-DDoS Pro service provides real-time protection against DDoS attacks.

Overall, Alibaba Cloud's architecture is designed to provide a highly scalable, flexible, and secure computing environment that can meet the needs of a wide range of workloads and use cases, from small-scale web applications to large-scale enterprise applications.

**Current Usage: -**

Alibaba Cloud is used by millions of customers worldwide, including startups, enterprises, and government agencies. Some of the notable customers include SAP, Siemens, and Sinopec. Alibaba Cloud is also popular among startups and small businesses due to its cost-effective pricing and flexible payment options. Alibaba Cloud’s pay-as-you-go model allows users to only pay for the resources they use, without any upfront costs or long-term commitments.

Alibaba Cloud is also widely used in the e-commerce industry, particularly in China, where it is the leading cloud provider. Alibaba Group, the parent company of Alibaba Cloud, is one of the largest e-commerce companies in the world, and its cloud services play a critical role in supporting its online marketplaces and digital platforms.

Another area where Alibaba Cloud is gaining traction is in the field of artificial intelligence (AI) and machine learning (ML). Alibaba Cloud offers a range of AI and ML services, including natural language processing (NLP), computer vision, and speech recognition, which are being used by enterprises to build intelligent applications and services.

Overall, Alibaba Cloud’s broad range of services and flexible pricing options have made it a popular choice for a wide range of customers, from small businesses to large enterprises, across a variety of industries and use cases.

**Technical Details: -**

* **Scalability:** Alibaba Cloud is designed to be highly scalable, allowing users to quickly and easily scale up or down their computing resources as needed. The Elastic Compute Service (ECS) includes features such as auto-scaling and load balancing, which enable users to automatically adjust their computing resources based on traffic and demand.
* **Performance:** Alibaba Cloud is optimized for high performance, with a range of services and features designed to maximize the speed and efficiency of computing workloads. For example, the ECS includes a range of instance types optimized for different use cases, including compute-optimized, memory-optimized, and storage-optimized instances. Alibaba Cloud also offers a range of high-performance storage options, including Object Storage Service (OSS), which provides fast, reliable, and secure object storage.
* **Reliability:** Alibaba Cloud is designed to be highly reliable, with multiple layers of redundancy and failover mechanisms built in. For example, the ECS includes automatic failover capabilities, which can quickly and automatically switch over to a backup instance in the event of a hardware failure. Alibaba Cloud also offers a range of backup and disaster recovery options, including backup and recovery services for databases and file systems.
* **Security:** Alibaba Cloud is built with security as a top priority, and includes a range of security features and services designed to protect users' data and applications. For example, the Security Center provides centralized security management and monitoring for cloud resources, while the Anti-DDoS Pro service provides real-time protection against DDoS attacks. Alibaba Cloud also offers a range of compliance certifications, including ISO 27001 and SOC 2, to help ensure that users' data is handled in a secure and compliant manner.

In addition to these technical details, Alibaba Cloud also offers a range of developer tools and APIs, which enable users to easily integrate their applications with Alibaba Cloud services and build custom solutions on top of the platform..

**Other informations: -**

Alibaba Cloud offers a wide range of cloud certifications and training programs to help users develop the skills and knowledge they need to build and manage cloud applications on the platform. These certifications include the Alibaba Cloud Certified Associate (ACA), Alibaba Cloud Certified Professional (ACP), and Alibaba Cloud Certified Expert (ACE) certifications, which cover a range of cloud-related topics and technologies. Alibaba Cloud also offers a range of training programs, including online courses, workshops, and bootcamps, which provide users with hands-on experience building and deploying cloud applications on the platform. These training programs are designed to help users develop the skills and knowledge they need to succeed in the cloud computing industry.

Alibaba Cloud has also built a vibrant community of developers and partners who can help customers with their cloud projects. The Alibaba Cloud Developer Community provides a platform for developers to share knowledge, collaborate on projects, and learn from each other. In addition, Alibaba Cloud has a network of partners, including system integrators, value-added resellers, and independent software vendors, who can provide customers with additional resources and expertise to help them build and deploy cloud applications on the platform. Alibaba Cloud is committed to providing its customers with the resources and support they need to succeed in the cloud computing industry, whether through its cloud certifications and training programs, its developer community, or its network of partners and service providers.

**Project References: -**

* <https://www.alibabacloud.com/>
* <https://www.alibabacloud.com/certification>
* <https://www.alibabacloud.com/community>
* <https://www.alibabacloud.com/product/kubernetes>

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

Open source software is software whose source code is available to the public, allowing anyone to inspect, modify, and distribute the code. Contributing to open source projects is a great way to improve your coding skills, gain experience working on real-world projects, and give back to the community.

There are several ways to contribute to open source projects, including code contributions, documentation, testing, bug reporting, and community engagement. Contributing to open-source can be a rewarding experience and is a great way to develop skills and give back to the community. Here are some steps you can follow to contribute to open-source:

1. **Choose a project**: Look for a project that interests you and aligns with your skills. You can browse popular platforms such as GitHub or GitLab to find projects that are seeking contributors.
2. **Understand the project**: Once you have identified a project, spend some time understanding its purpose, the codebase, and the community. Read the documentation and code to understand how the project works.
3. **Interact with the community**: Join the community and engage with other contributors. This can include participating in forums, mailing lists, or chat rooms. This will give you an opportunity to ask questions, learn from others and get feedback on your contributions.
4. **Find an issue**: Look for issues or features that the project needs help with. These issues can be labelled as "beginner-friendly" or "good first issue" and are usually easy to tackle. You can also look for issues that align with your skills and interests.
5. **Contribute**: Once you have found an issue, start working on it. Make sure to follow the project's guidelines and coding standards. Submit your code for review and incorporate feedback from other contributors.
6. **Stay involved**: Once you have contributed, continue to stay involved with the project. This can include providing support to other contributors, testing new features, or identifying and fixing bugs.

**Ways to Contribute:**

**Code contributions:** One of the most obvious ways to contribute to open source projects is by writing code. This can include fixing bugs, adding new features, and improving performance. Before you start coding, it's important to read the project's documentation and get familiar with the project's coding style and conventions.

**Documentation:** Many open source projects have documentation that needs improvement. This can include updating existing documentation, creating new documentation, or translating documentation into other languages.

**Testing:** Testing is a critical part of software development, and open source projects often need help with testing. This can include writing automated tests, manual testing, and reporting bugs.

**Bug reporting:** If you find a bug in an open source project, you can report it to the project's issue tracker. Make sure to include detailed information about the bug, including steps to reproduce it and any error messages.

**Community engagement**: Contributing to open source projects also involves engaging with the community. This can include answering questions on forums, participating in mailing lists or chat rooms, and helping new contributors get started.

**Methods to Join the Community and Start Contributing:**

**Find a project:** The first step to contributing to open source is to find a project that interests you. You can search for projects on GitHub or other code hosting platforms, or browse directories like Open Source at Microsoft or the Apache Software Foundation.

**Read the documentation:** Once you've found a project you're interested in, read the documentation to get familiar with the project's goals, coding style, and conventions.

**Look for beginner-friendly issues:** Many open source projects have "good first issues" or "beginner-friendly" issues that are designed for new contributors. Look for these issues to get started.

**Introduce yourself:** Once you've found an issue you're interested in working on, introduce yourself to the community. This can include joining the project's mailing list or chat room and letting people know that you're interested in contributing.

**Start small:** When you're first starting out, it's important to start small. Pick an issue that you think you can handle and work on that. As you gain experience, you can tackle more complex issues.

## **Contribution Flowchart**

# 

Figure 5: Contribution Flowchart

# **Community Engagement Experience**

During the course of my training, I had the opportunity to engage with the open-source community in a number of ways. Here are a few examples of my community engagement experience:Regular community meetings: The foundation hosts regular meetings where contributors can share updates on their work, discuss issues, and collaborate on new ideas.

* Slack channels: The SODA Foundation has multiple Slack channels where contributors can ask questions, share ideas, and engage with other members of the community.
* Community events: The foundation hosts and participates in various community events, such as conferences and meetups, to promote the project and engage with stakeholders.
* Online forums: The SODA Foundation has online forums where contributors can discuss technical issues and collaborate on development projects.
* Documentation and tutorials: The foundation provides comprehensive documentation and tutorials to help contributors get started with the project and learn more about its features and capabilities.

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.

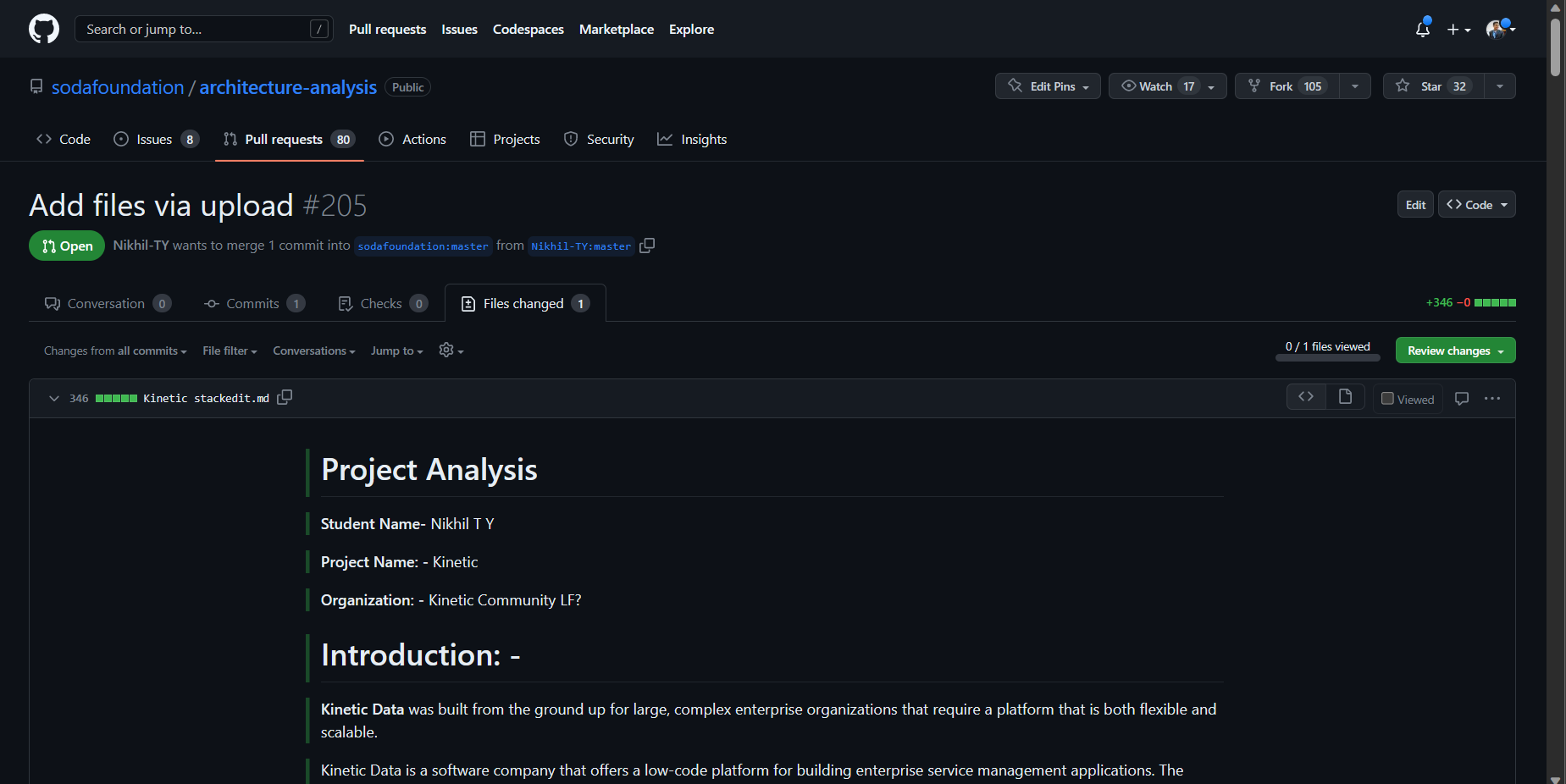
# **My Contributions**

During the course, I contributed to an open source project by creating a documentation for Kinetic Project. I started analysing about the project by browsing the information related to Kinetic project.

I started gathering information from Kinetic’s website and github repositories. After gathering information, I started analysing the project’s features, architecture and technical aspects, and made a documentation including the above details. I was able to get to know the proper method of documentation in stackedit.

After creating the documentation for the same, I forked the Architecture analysis repository of SODA Foundation to my GitHub account. Then I uploaded my documentation to the forked repository and commit the changes. Once I had made the changes, I created a pull request on GitHub and requested a review from the project's maintainers.

Here is a screenshot of my pull request:

Figure 6: My Contribution

# **Open Source Value**

The value of open source contributions is multi-fold and extends to both individuals and communities.

Here are some of the key values of open source contributions:

* **Collaboration:** Open source projects are often collaborative efforts, with contributors from all around the world working together to improve the software. This collaborative approach can result in higher quality software that meets the needs of a wider range of users.
* **Transparency:** Open source software is transparent, which means that users can see how it works and modify it if necessary. This transparency can help build trust in the software and ensure that it is free from hidden vulnerabilities or malicious code.
* **Flexibility:** Open source 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:** 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.
* **Innovation:** Open source software 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.

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, which can lead to better quality software that meets the needs of a wider range of users.

# **References: -**

* <https://github.com/MOJO-007>
* <https://github.com/sodafoundation>
* <https://landscape.cncf.io/>
* <https://opensource.google/projects>