**INTRODUCTION**

SkyWalking: an APM(application performance monitor) system, especially designed for microservices, cloud native and container-based architectures.

SkyWalking is an open source APM system, including monitoring, tracing, diagnosing capabilities for distributed system in Cloud Native architecture.

SkyWalking is an open-source observability platform designed to help developers monitor and trace distributed systems. It provides a distributed tracing system, metrics analysis, and service mesh telemetry analysis capabilities. SkyWalking can be used to monitor complex applications deployed in cloud-native environments such as Kubernetes, Istio, and Envoy, as well as traditional application architectures.

SkyWalking collects data from various sources, such as application logs, network traffic, and system metrics. It then analyzes this data to provide insights into application performance, including identifying bottlenecks, tracing the flow of requests through the system, and detecting errors and exceptions.

SkyWalking supports multiple programming languages, including Java, .NET, Node.js, and Go, among others. It can also integrate with various logging and monitoring systems, such as Elasticsearch, Prometheus, and Grafana, to provide a comprehensive view of the entire system.

SkyWalking is widely used by developers and organizations to gain visibility into complex distributed systems and improve their application's performance and reliability.

# 

# **PROJECT SUMMARY**

| Website | https://skywalking.apache.org/docs/ |
| --- | --- |
| Organization/Foundation Name | The Apache Software Foundation. |
| License | Apache License 2.0 |
| Open/Proprietary | Apache SkyWalking is an **open-source APM** for a distributed system, Apache Software Foundation … |
| Source Path(if open source) | https://github.com/apache/skywalking |
| Brief Description | SkyWalking is an open source observability platform used to collect, analyze, aggregate and visualize data from services and cloud native infrastructures. SkyWalking provides an easy way to maintain a clear view of your distributed systems, even across Clouds. It is a modern APM, specially designed for cloud native, container based distributed systems. |

# **PROJECT DETAILS**

## Key Features

SkyWalking is an open-source APM (Application Performance Monitoring) system that provides observability and insight into distributed systems. Some key features of SkyWalking include:

Distributed Tracing: SkyWalking allows you to trace a request across multiple services and components in a distributed system. This helps you understand how different parts of your system are interacting with each other and identify bottlenecks or errors.

Metrics Analysis: SkyWalking collects metrics from your system, including system metrics like CPU usage and memory, as well as application-level metrics like response time and throughput. You can use these metrics to monitor the health of your system and identify performance issues.

Service Topology Visualization: SkyWalking provides a graphical representation of your system's architecture and how different services are connected. This makes it easy to understand the relationships between different components and identify potential points of failure.

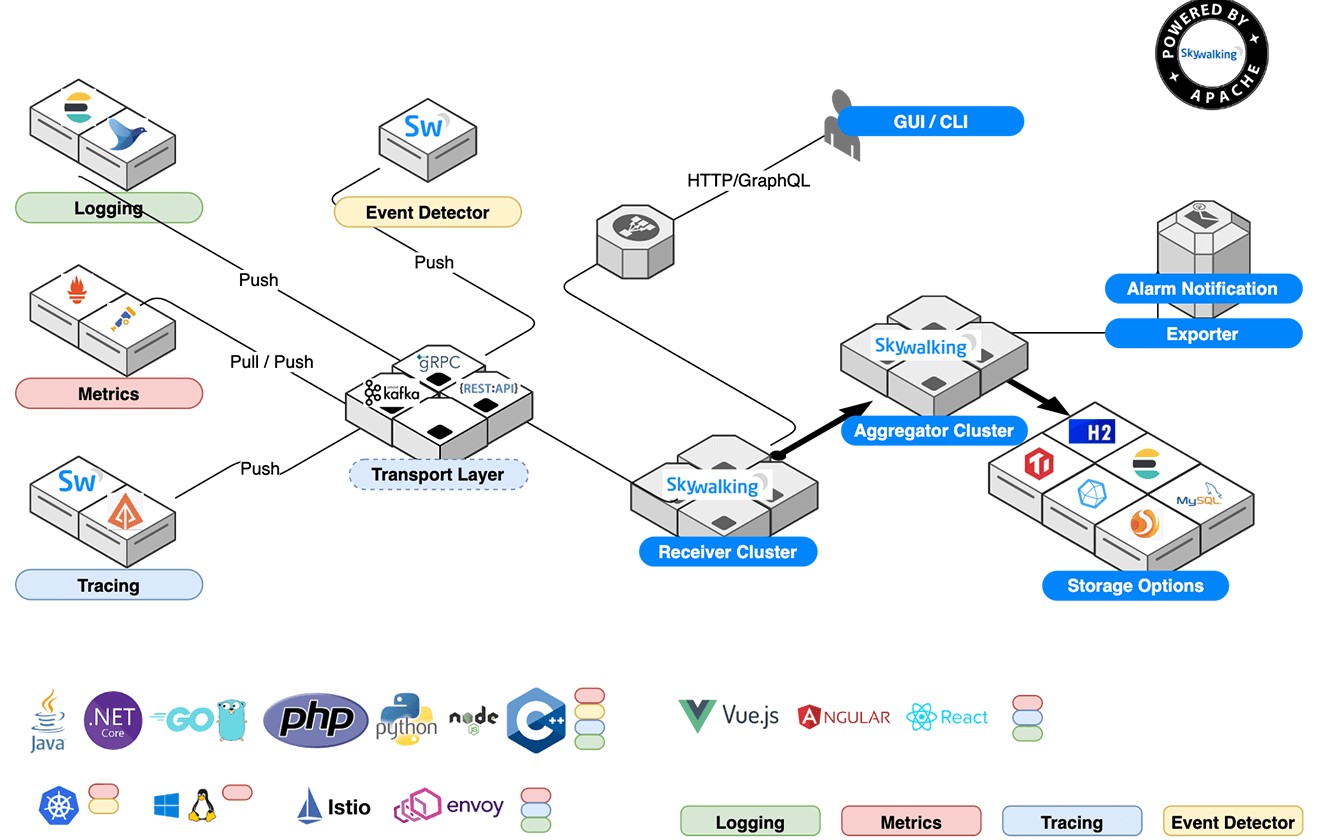
Alarm and Notification: SkyWalking can alert you when performance metrics or other indicators reach critical thresholds. This helps you quickly respond to issues and prevent downtime.

Plugin Architecture: SkyWalking has a plugin architecture that allows you to easily integrate it with other tools and frameworks. This makes it easy to get started with SkyWalking and adapt it to your specific needs.

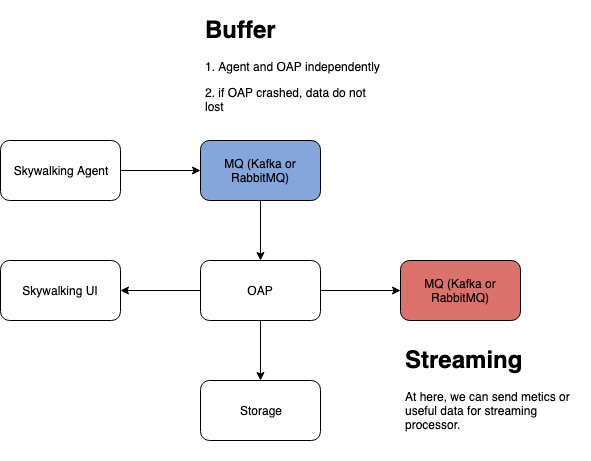
Compatibility: SkyWalking supports a wide range of programming languages, including Java, .NET, Node.js, and Python. This makes it a versatile tool that can be used in a variety of environments.

Overall, SkyWalking provides a comprehensive set of tools for monitoring and troubleshooting distributed systems, making it an essential tool for DevOps teams and system administrators.

## Architecture



* **Probes** collect data and reformat them for SkyWalking requirements (different probes support different sources).
* **Platform backend** supports data aggregation, analysis and streaming process covers traces, metrics, and logs.
* **Storage** houses SkyWalking data through an open/plugable interface. You can choose an existing implementation, such as ElasticSearch, H2, MySQL, TiDB, InfluxDB, or implement your own. Patches for new storage implementors welcome!
* **UI** is a highly customizable web based interface allowing SkyWalking end users to visualize and manage SkyWalking data.



## **CURRENT USAGE**

SkyWalking is an open-source application performance monitoring (APM) system that can be used to monitor and analyze the performance of distributed systems and microservices-based applications. It is commonly used by organizations and products in various industries, including:

**Alibaba Cloud** - SkyWalking was originally developed by the Alibaba Group, and it is still actively used by Alibaba Cloud to monitor their systems.

**Huawei** **-** Huawei, a global technology company, uses SkyWalking to monitor the performance of its products and services.

**Didi Chuxing -** Didi Chuxing, a ride-hailing company based in China, uses SkyWalking to monitor their distributed systems and microservices.

**Zalando -** Zalando, a European e-commerce company, uses SkyWalking to monitor the performance of their systems and applications.

**Apache RocketMQ -** SkyWalking can be used to monitor and analyze the performance of Apache RocketMQ, an open-source distributed messaging and streaming platform.

**Apache ShardingSphere -** SkyWalking can be used to monitor and analyze the performance of Apache ShardingSphere, an open-source database sharding and scaling solution.

**Apache Dubbo -** SkyWalking can be used to monitor and analyze the performance of Apache Dubbo, an open-source distributed service framework.

|  |  |
| --- | --- |

## **TECHNICAL DETAIL**

* Scalability

SkyWalking is an open-source application performance monitoring (APM) system that provides insights into the performance and health of distributed systems. As with any system, scalability is an important factor to consider when evaluating SkyWalking.

SkyWalking has been designed with scalability in mind and offers various features that allow it to scale to handle large-scale distributed systems. Some of these features include:

Horizontal scalability: SkyWalking can be easily scaled horizontally by adding more instances of the collector and/or storage components. This allows for a distributed architecture that can handle large amounts of data.

Distributed tracing: SkyWalking's distributed tracing capabilities allow it to trace transactions across multiple services and nodes, making it easier to identify performance bottlenecks and optimize the system.

Sampling: SkyWalking offers configurable sampling rates, which allows users to reduce the amount of data collected without sacrificing the quality of insights provided.

High availability: SkyWalking is designed to be highly available, with multiple instances of each component running in a cluster. This ensures that the system is always available, even in the event of component failures.

* Performance

SkyWalking is an open source application performance monitoring (APM) system designed to collect, analyze, and visualize distributed tracing data for cloud-native applications. Its performance can be evaluated based on several factors, including:

Ease of installation and configuration: SkyWalking is relatively easy to install and configure, and can be integrated with various programming languages and frameworks.

Scalability: SkyWalking can handle large-scale distributed systems with ease, and its architecture is designed to be scalable.

Performance impact: SkyWalking has a low performance impact on the monitored applications, which means that it does not significantly slow down the applications it is monitoring.

Data visualization: SkyWalking provides an intuitive web-based UI that enables users to visualize the collected data and quickly identify performance bottlenecks.

Compatibility: SkyWalking is compatible with various cloud-native environments, including Kubernetes, Istio, and more.

Overall, SkyWalking is a reliable and efficient APM system that can help developers and operations teams to monitor the performance of their cloud-native applications and improve their overall reliability and availability.

* Technical Information

SkyWalking is an open-source application performance monitoring (APM) system that uses a distributed tracing architecture to monitor and analyze the performance of distributed systems and microservices-based applications. Here are some technical details about SkyWalking:

Architecture: SkyWalking has a distributed architecture, which means that it is designed to monitor and analyze the performance of distributed systems and microservices-based applications. It consists of three main components: the data collectors, the analysis and storage system, and the user interface.

Data collectors: SkyWalking supports multiple data collection methods, including agents, service mesh, and log files. The agents can be deployed on each application instance to collect metrics and trace data, while the service mesh can automatically collect data from the entire system. SkyWalking also provides log analysis capabilities to extract data from log files.

Trace data: SkyWalking collects trace data from each application instance and uses it to construct a distributed tracing tree. This trace data includes information about the application's execution path, the duration of each operation, and any errors or exceptions that occurred during execution.

Metrics data: SkyWalking collects various metrics from each application instance, including CPU usage, memory usage, network traffic, and database performance.

Analysis and storage system: SkyWalking uses a backend analysis and storage system to store and analyze the collected data. This system includes a database, an alerting system, and an analytics engine. SkyWalking supports various databases, including Elasticsearch, MySQL, and PostgreSQL.

User interface: SkyWalking provides a web-based user interface that allows users to visualize and analyze the collected data. The user interface includes dashboards, metrics charts, and distributed tracing trees.

Integrations: SkyWalking supports integrations with various technologies and frameworks, including Java, .NET, PHP, Node.js, and Kubernetes.

Overall, SkyWalking is a powerful and flexible APM system that provides a wealth of data collection, analysis, and visualization capabilities to help users monitor and optimize the performance of their distributed systems and microservices-based applications.

### **PROJECT COMPARISON**

Scouter and Skywalking are both open-source distributed tracing systems that are used to monitor and troubleshoot microservices and cloud-native applications. While both tools have similar goals, they differ in their approach and features.

Scouter is a lightweight Java-based monitoring and profiling tool that is designed to be easy to install and use. It uses a simple architecture that consists of a collector and agents that are deployed on each application server. Scouter supports tracing, monitoring, and profiling of Java-based applications, and it can be used to identify performance bottlenecks and errors in real-time.

Skywalking, on the other hand, is a more feature-rich distributed tracing system that is written in Java and Go. It supports tracing of multiple programming languages, including Java, .NET, and Node.js, and it has an extensive set of plugins that can be used to integrate with various technologies and frameworks. Skywalking also includes features like service mesh telemetry, distributed tracing visualization, and alerting.

In terms of architecture, Skywalking uses a more complex approach that includes multiple components, including agents, collectors, and a web UI. This complexity can make it more challenging to set up and use compared to Scouter, which has a simpler architecture.

Overall, Scouter is a good option if you're looking for a simple, lightweight monitoring tool for Java-based applications, while Skywalking is a more comprehensive solution that provides more extensive features and broader language support. The choice between the two ultimately depends on your specific needs and preferences.

### **OTHER INFORMATION**

SkyWalking is an open source APM system, including monitoring, tracing, diagnosing capabilities for distributed system in Cloud Native architecture.

Distributed Tracing

End-to-end distributed tracing. Service topology analysis, service-centric observability and APIs dashboards.

Agents for your stack

Java, .Net Core, PHP, NodeJS, Golang, LUA, Rust, C++, Client JavaScript and Python agents with active development and maintenance.

eBPF early adoption

Rover agent works as metrics collector and profiler powered by eBPF to diagnose CPU and network performance.

Scaling

100+ billion telemetry data could be collected and analyzed from one SkyWalking cluster.

Mature Telemetry Ecosystems Supported

Metrics, Traces, and Logs from mature ecosystems are supported, e.g. Zipkin, OpenTelemetry, Prometheus, Zabbix, Fluentd

Native APM Database

BanyanDB, an observability database, created in 2022, aims to ingest, analyze and store telemetry/observability data.

Consistent Metrics Aggregation

SkyWalking native meter format and widely known metrics format(OpenCensus, OTLP, Telegraf, Zabbix, e.g.) are processed through the same script pipeline.

Log Management Pipeline

Support log formatting, extract metrics, various sampling policies through script pipeline in high performance.

Alerting and Telemetry Pipelines

Support service-centric, deployment-centric, API-centric alarm rule setting. Support forwarding alarms and all telemetry data to 3rd party.

SkyWalking is an open source APM system, including monitoring, tracing, diagnosing capabilities for distributed system in Cloud Native architecture.

### **Reference**

* <https://skywalking.apache.org/docs/#SkyWalking>
* <https://github.com/apache/skywalking>
* <https://skywalking.apache.org/>
* <https://www.apache.org/>

**-----------------------------End of the template--------------------------------**