


Skywalking-01：Skywalking介绍

 Switch 发布于 7 月 27 日

Skywalking介绍

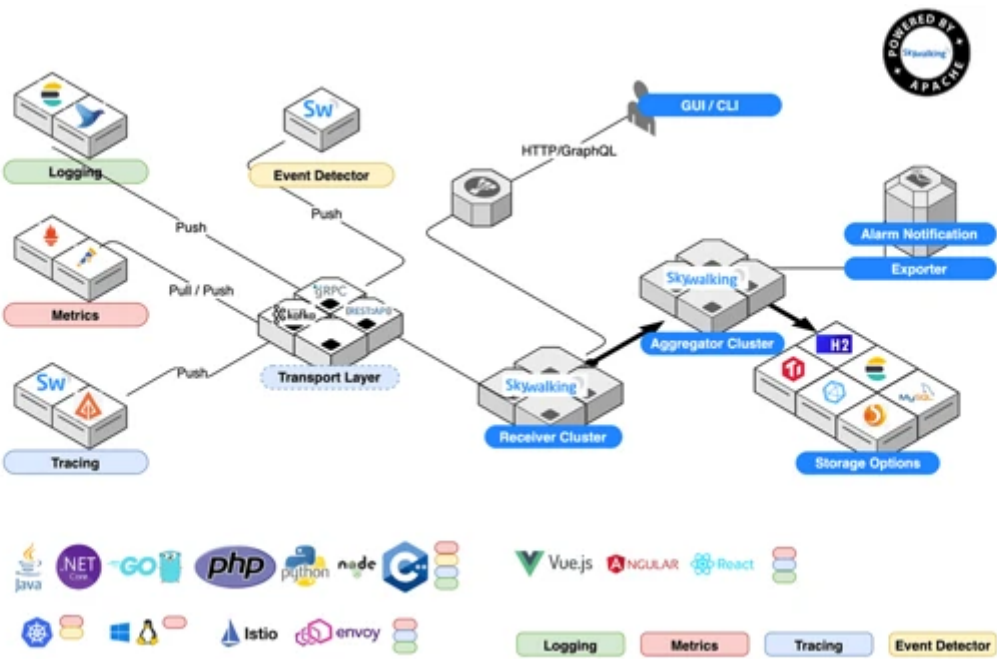
Application performance monitor tool for distributed systems, especially designed for microservices, cloud native and container-based (Docker, Kubernetes, Mesos) architectures.

SkyWalking is an Observability Analysis Platform and Application Performance Management system.

Tracing, Metrics and Logging all-in-one solution.

Java, .Net Core, PHP, NodeJS, Golang, LUA, C++ agents supported

Istio + Envoy Service Mesh supported



Skywalking文档结构

- [Welcome](#)
- [Concepts and Designs](#)
 - What is SkyWalking?
 - [Overview and Core concepts](#)
 - [Project Goals](#)
 - Project Goals
 - [Introduction](#)
 - [Service auto instrument agent](#)
 - [Manual instrument SDK](#)
 - [Service Mesh probe](#)
 - Backend
 - [Overview](#)
 - [Observability Analysis Language](#)
 - [Query in OAP](#)
 - [Meter Analysis Language](#)
 - UI
 - [Overview](#)
- [Setup](#)
 - Language agents in Service
 - [Agents](#)
 - [Java agent](#)

- [Supported middleware, framework and library](#)
 - [Agent Configuration Properties](#)
 - [Optional plugins](#)
 - [Bootstrap/JVM class plugin](#)
 - [Advanced reporters](#)
 - [Plugin development guide](#)
 - [Agent plugin tests and performance tests](#)
 - [Other language agents](#)
- Service Mesh
 - [Observe Istio Control Plane](#)
 - [Observe Service Mesh](#)
- Proxy
 - Envoy Proxy
 - [Send Envoy metrics to SkyWalking with / without Istio](#)
- [Backend, UI and CLI setup document](#)
 - [Backend setup document](#)
 - [Configuration Vocabulary](#)
 - [Overriding settings](#)
 - [IP and port setting](#)
 - [Backend init mode startup](#)
 - [Cluster management](#)
 - [Deploy in kubernetes](#)
 - [Choose storage](#)
 - [Set receivers](#)
 - [Open fetchers](#)
 - [Trace sampling](#)
 - [Slow DB statement threshold](#)
 - [OAL scripts](#)
 - [Alarm](#)
 - [Advanced deployment options](#)
 - [Metrics exporter](#)
 - [Time To Live \(TTL\)](#)
 - [Dynamic Configuration](#)
 - [Uninstrumented Gateways](#)
 - [Apdex threshold](#)
 - [Service Grouping](#)
 - [Group Parameterized Endpoints](#)
 - [OpenTelemetry Metrics Analysis](#)
 - [Meter Analysis](#)
 - [Spring Sleuth Metrics Analysis](#)
 - [UI setup document](#)
 - [CLI setup document](#)
- [UI Introduction](#)
- [Contributing Guides](#)
 - [Contact us](#)
 - [Process to become official Apache SkyWalking Committer](#)
 - [Compiling Guide](#)
 - [Agent plugin development guide](#)
- [Protocols](#)
 - [Query Protocol](#)
- [FAQs](#)

PS：附上一个[V8.0.0中文文档](#)

怎样读文档

Skywalking 官方文档的组织结构很清晰。读下[Welcome](#)，就能大致了解文档结构，然后选感兴趣的读。

Skywalking项目结构

部署项目结构

```
.
├── bin # 执行文件
│   ├── oapService.bat
│   ├── oapServiceInit.bat
│   ├── oapServiceInit.sh # 启动oap并初始化，oap集群中使用
│   ├── oapServiceNoInit.bat
│   ├── oapServiceNoInit.sh # 只启动oap，oap集群中使用
│   ├── oapService.sh # 启动oap并初始化
│   ├── startup.bat
│   ├── startup.sh # 启动oap并初始化，同时启动ui
│   ├── webappService.bat
│   └── webappService.sh # 启动ui
├── config # oap配置文件
│   ├── alarm-settings-sample.yml
│   ├── alarm-settings.yml # 告警配置
│   ├── application.yml # oap主配置文件
│   ├── component-libraries.yml
│   ├── endpoint-name-grouping.yml
│   ├── envoy-metrics-rules
│   │   └── envoy.yml
│   ├── fetcher-prom-rules # Prometheus Fetcher 配置文件
│   │   └── self.yml
│   ├── gateways.yml
│   ├── log4j2.xml
│   ├── metadata-service-mapping.yml
│   └── meter-analyzer-config Meter # Receiver配置文件
```

agent目录结构

```
.
├── activations # 工具包，提供 日志打印TID、跨线程传递TID等功能
│   ├── apm-toolkit-kafka-activation-8.4.0.jar
│   ├── apm-toolkit-log4j-1.x-activation-8.4.0.jar
│   ├── apm-toolkit-log4j-2.x-activation-8.4.0.jar
│   ├── apm-toolkit-logback-1.x-activation-8.4.0.jar
│   ├── apm-toolkit-logging-common-8.4.0.jar
│   ├── apm-toolkit-meter-activation-8.4.0.jar
│   ├── apm-toolkit-opentracing-activation-8.4.0.jar
│   └── apm-toolkit-trace-activation-8.4.0.jar
├── bootstrap-plugins # All bootstrap plugins are optional, due to unexpected risk.
│   ├── apm-jdk-http-plugin-8.4.0.jar
│   └── apm-jdk-threading-plugin-8.4.0.jar
├── config # agent配置文件
│   └── agent.config
├── logs # 日志目录
├── optional-plugins # 可选插件
│   ├── apm-customize-enhance-plugin-8.4.0.jar
│   ├── apm-gson-2.x-plugin-8.4.0.jar
│   ├── apm-kotlin-coroutine-plugin-8.4.0.jar
│   ├── apm-quartz-scheduler-2.x-plugin-8.4.0.jar
│   ├── apm-spring-annotation-plugin-8.4.0.jar
│   ├── apm-spring-cloud-gateway-2.0.x-plugin-8.4.0.jar
│   ├── apm-spring-cloud-gateway-2.1.x-plugin-8.4.0.jar
│   ├── apm-spring-tx-plugin-8.4.0.jar
│   └── apm-spring-webflux-5.x-plugin-8.4.0.jar
```

源码项目结构

```
.
├── apm-application-toolkit # 工具包
│   ├── apm-toolkit-kafka
│   ├── apm-toolkit-log4j-1.x
│   ├── apm-toolkit-log4j-2.x
│   ├── apm-toolkit-logback-1.x
│   ├── apm-toolkit-meter
│   ├── apm-toolkit-micrometer-registry
│   ├── apm-toolkit-opentracing
│   └── apm-toolkit-trace
├── apm-checkstyle # checkstyle配置目录
├── apm-commons # common包相关
│   ├── apm-datacarrier
│   └── apm-util
├── apm-dist # 构建后生成的可执行项目目录
├── apm-dist-es7 # 构建后生成的可执行项目目录 base:es7
├── apm-protocol # 协议
│   └── apm-network
├── apm-sniffer # agent相关
│   ├── apm-agent
│   ├── apm-agent-core
│   ├── apm-sdk-plugin
│   │   ├── activemq-5.x-plugin
│   │   ├── armeria-0.84.x-plugin
│   │   ├── armeria-0.85.x-plugin
│   │   └── asyncHttpClient-2.x-plugin
```

Skywalking环境搭建

部署文件结构

```
.
├── docker-compose.yml # docker-compose部署文件
├── es # es集群目录
│   ├── es01
│   ├── es02
│   └── es03
├── kibana # kibana配置目录
│   └── kibana.yml
├── mysql # mysql存储目录
├── nacos # nacos配置目录
│   ├── init.d
│   │   └── custom.properties
│   ├── nacos01
│   ├── nacos02
│   └── nacos03
├── prometheus # prometheus配置目录
│   └── prometheus-cluster.yaml
├── skywalking # skywalking配置目录
│   ├── oap # oap服务配置
│   │   ├── oap01
│   │   │   └── config
│   │   └── oap02
│   │       └── config
│   └── ui # ui服务配置
│       └── webapp.yml
└── startup.sh # docker-compose脚本，提供易用性功能
```

docker-compose.yml

```
version: '3.5'

networks:
  skywalking_networks:
    ipam:
      config:
        - subnet: 192.168.200.0/24

# elastic search参考: https://www.elastic.co/guide/en/elasticsearch/reference/7.13/docker.html
# kibana参考: https://www.elastic.co/guide/en/kibana/7.13/docker.html
# nacos参考: https://github.com/nacos-group/nacos-docker/tree/2.0.1
# nacos prometheus grafana监控参考: https://nacos.io/zh-cn/docs/monitor-guide.html
services:
  es01:
    image: docker.elastic.co/elasticsearch/elasticsearch:${ES_TAG}
    container_name: es01
    environment:
      - node.name=es01
      - cluster.name=es-docker-cluster
      - discovery.seed_hosts=es02,es03
      - cluster.initial_master_nodes=es01,es02,es03
      - bootstrap.memory_lock=true
      - "ES_JAVA_OPTS=-Xms1024m -Xmx1024m"
      - cluster.max_shards_per_node=100000
      - thread_pool.write.queue_size=1000
```

部署文件：[docker.zip](#)

PS：可用内存未大于11G，切勿尝试，如想尝试，请限制容器内存使用。

参考文档

- 1. [Skywalking官网](#)
- 2. [Skywalking项目README文档v8.4.0](#)

分享并记录所学所见



阅读 200 • 发布于 7 月 27 日

👍 赞

🔖 收藏

🔗 分享

本作品系原创，采用《署名-非商业性使用-禁止演绎 4.0 国际》许可协议



Switch

9 声望 0 粉丝

关注作者

0 条评论

得票数 最新



撰写评论 ...



提交评论

你知道吗？

我没有写过诗，但有人说我写过的代码，像诗一样优雅。

注册登录

继续阅读

使用 docker 部署 spring boot 并接入 skywalking

最近在研究skywalking，打算使用k8s部署 skywalking 并将 pod 中的应用接入 skywalking 进行服务链路追踪。这篇文章先不介绍 s...
[惜鸟](#) · 阅读 3.4k · 17 赞

skywalking实现分布式系统链路追踪

随着微服务的越来越流行，我们服务之间的调用关系就显得越来越复杂，我们急需一个APM工具来分析系统中存在的各种性能指标...
[huan1993](#) · 阅读 3.9k · 2 赞

Skywalking系列博客2-Skywalking使用

Java agent支持的框架、中间件等，可在 <https://github.com/apache/skywalking/blob/v6.6.0/docs/en/setup/service-agent/java-ag...>
[eacdy](#) · 阅读 958

聊聊skywalking的kafka-plugin

[skywalking-6.6.0/apm-sniffer/apm-sdk-plugin/kafka-plugin/src/main/resources/skywalking-plugin.def](#)
[codecraft](#) · 阅读 991

聊聊skywalking的sharding-sphere-plugin

[skywalking-6.6.0/apm-sniffer/apm-sdk-plugin/sharding-sphere-4.x-plugin/src/main/resources/skywalking-plugin.def](#)
[codecraft](#) · 阅读 751

史上最全SpringCloud整合skywalking

SkyWalking 是什么？FROM [\[链接\]](#)分布式系统的应用程序性能监视工具，专为微服务、云原生架构和基于容器（Docker、K8s、Me...
[liumang](#) · 阅读 663

聊聊skywalking的lettuce-plugin

[skywalking-6.6.0/apm-sniffer/optional-plugins/lettuce-5.x-plugin/src/main/resources/skywalking-plugin.def](#)
[codecraft](#) · 阅读 598

Skywalking-03：Skywalking本地调试

在打开的 skywalking-live-demo 项目中使用 Module from Existing Sources 的方式导入 Skywalking。
[Switch](#) · 阅读 216

[热门专栏](#)
[热门课程](#)
[最新活动](#)
[技术圈](#)
[酷工作](#)

[PHP 开发课程](#)
[Python 开发课程](#)
[前端开发课程](#)
[移动开发课程](#)

[用户排行榜](#)
[勋章](#)
[帮助中心](#)
[声望与权限](#)
[社区服务中心](#)
[建议反馈](#)

[广告投放](#)
[职位发布](#)
[讲师招募](#)
[联系我们](#)
[合作伙伴](#)

[社区运营日志](#)
[市场运营日志](#)
[团队日志](#)
[社区访谈](#)

[隐私政策](#)
[下载 App](#)

Copyright © 2011-2021 SegmentFault. 当前呈现版本 21.09.09



[浙ICP备15005796号-2](#) [浙公网安备33010602002000号](#) ICP 经营许可 [浙B2-20201554](#)

杭州堆栈科技有限公司版权所有