



# The Introduction Of Apache ShardingSphere

京东数科-潘娟



潘娟

京东数科高级DBA&Apache ShardingSphere PMC

主要负责京东数科分布式数据库开发、数据库运维自动化平台开发等工作。曾负责京东数科数据库自动化平台设计与开发，现专注于Apache ShardingSphere分布式数据库中间件平台的开发。主要在分布式数据库、开源、分布式架构等相关领域进行探索。多次受邀参加数据库&架构领域的相关会议并进行分享交流。



# 目录

## CONTENTS

1

Background

2

For what

3

The architecture

4

Summer 2020 Tasks



# Background

## Apache TLP

- Apache CarbonData
- Apache Dubbo
- Apache Eagle
- Apache Griffin
- Apache HAWQ
- Apache Kylin
- Apache RocketMQ
- Apache ServiceComb
- **Apache ShardingSphere**
- Apache Skywalking

## Apache Incubator

- Apache Incubator
- APISIX
- BRPC
- DolphinScheduler
- Doris
- ECharts
- IotDB
- MesaTEE
- TubeMQ
- Weex



The screenshot shows the Apache ShardingSphere website. At the top, there is a navigation bar with links for '下载' (Download), '文档' (Documentation), '社区' (Community), 'ASF', '中文' (Chinese), and 'EN'. The main header features the Apache Software Foundation logo and the text 'Incubating'. The title 'Apache ShardingSphere' is prominently displayed in orange and yellow, with the tagline '微内核 & 云原生 & 零侵入' (Micro-kernel & Cloud-native & Zero-intrusion) below it. The main content area contains a paragraph describing Apache ShardingSphere as an open-source distributed database middleware solution, followed by a paragraph defining its role as a relational database middleware. At the bottom, there are two buttons: '下载' (Download) and '了解更多' (Learn More).

下载 文档 社区 ASF 中文 EN

Incubating

## Apache ShardingSphere

微内核 & 云原生 & 零侵入

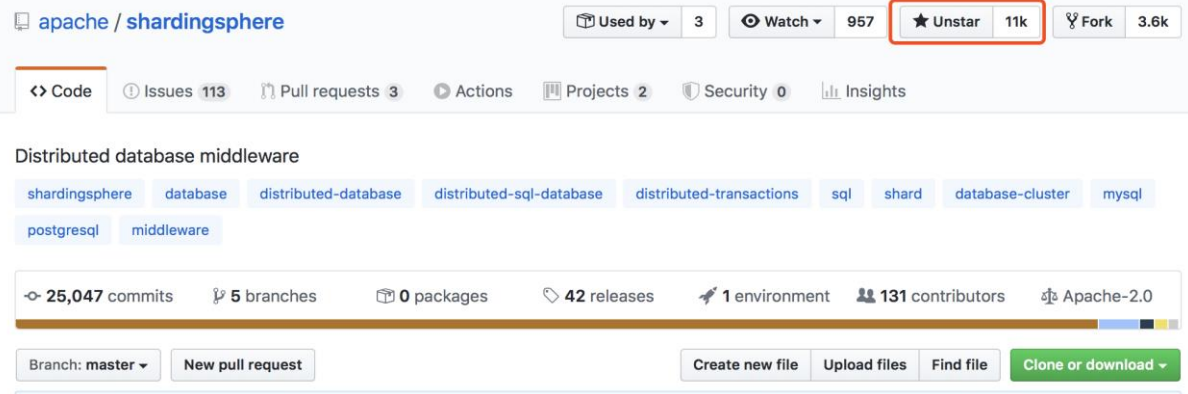
Apache ShardingSphere(Incubator) 是一套开源的分布式数据库中间件解决方案组成的生态圈，它由Sharding-JDBC、Sharding-Proxy和Sharding-Sidecar（规划中）这3款相互独立，却又能够混合部署配合使用的产品组成。它们均提供标准化的数据分片、分布式事务和数据库治理功能，可适用于如Java同构、异构语言、云原生等各种多样化的应用场景。

ShardingSphere定位为关系型数据库中间件，旨在充分合理地在分布式的场景下利用关系型数据库的计算和存储能力，而并非实现一个全新的关系型数据库。它通过关注不变，进而抓住事物本质。关系型数据库当今依然占有巨大市场，是各个公司核心业务的基石，未来也难于撼动，我们目前阶段更加关注在原有基础上的增量，而非颠覆。

Apache官方发布从4.0.0版本开始。

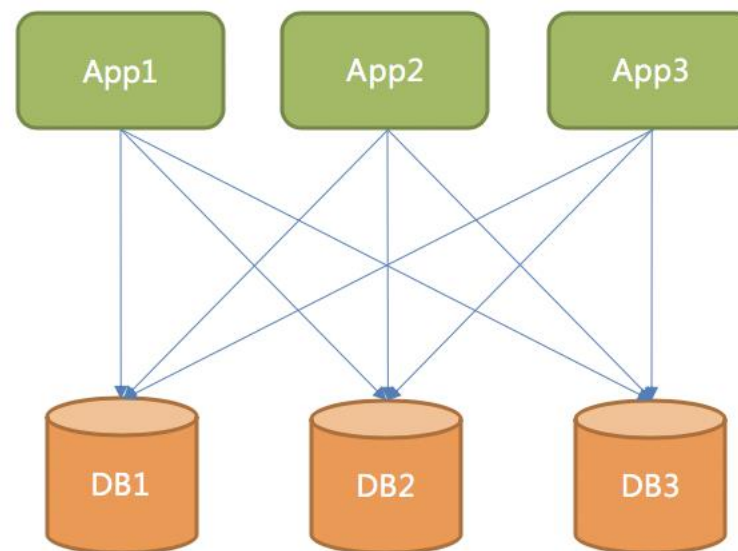
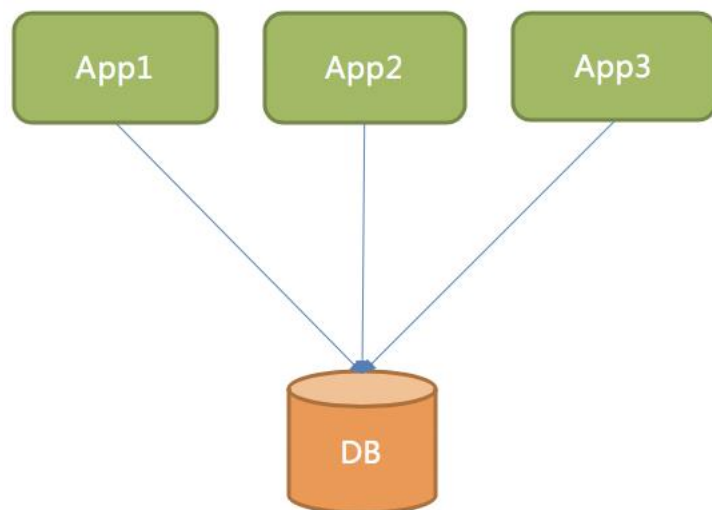
下载 了解更多

# Background

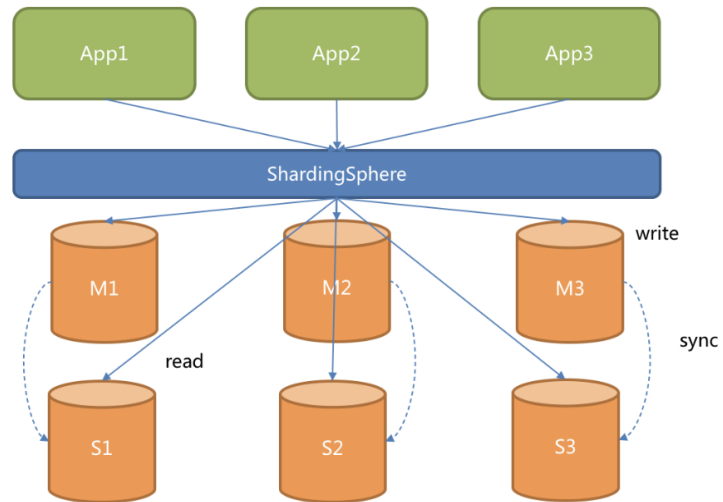
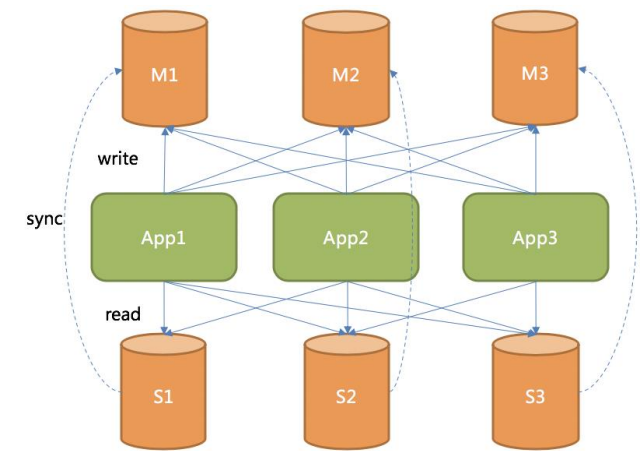


[1] <https://shardingsphere.apache.org/community/cn/users/>

For what

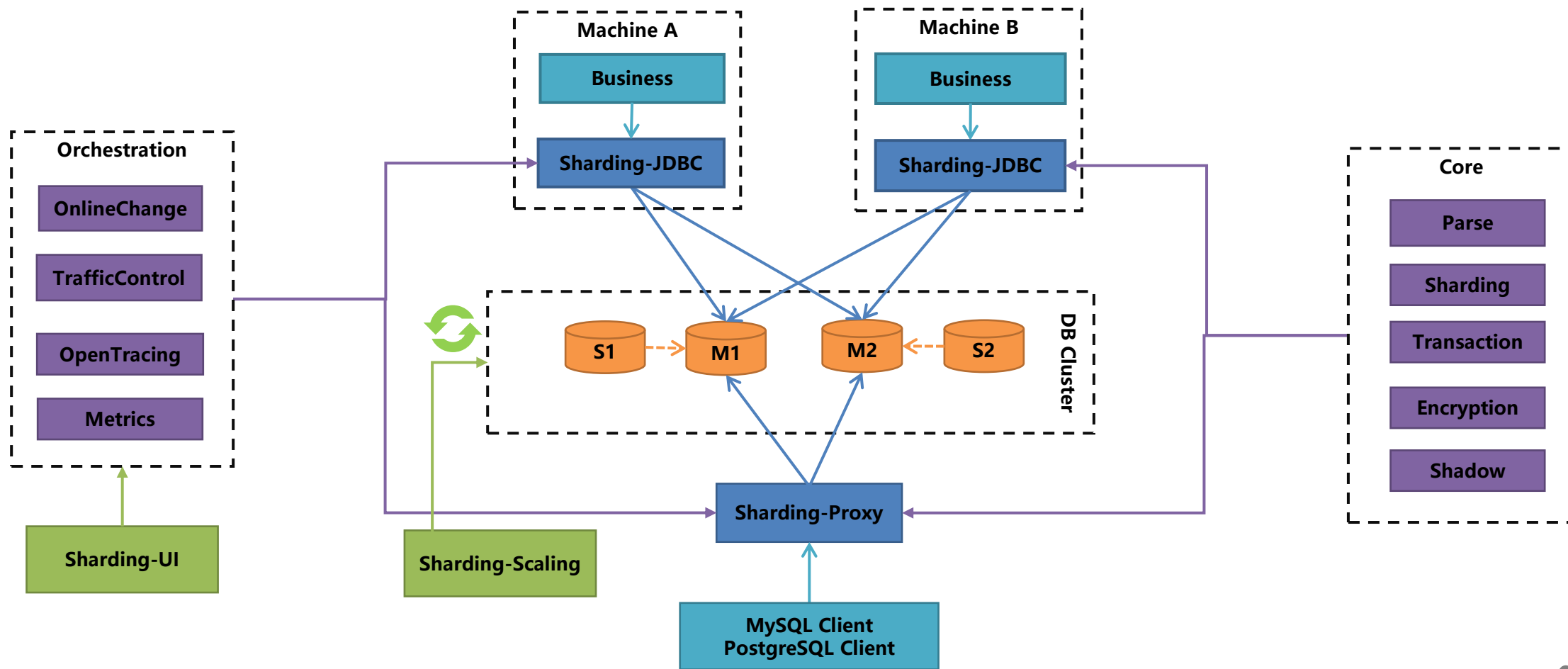


# For what



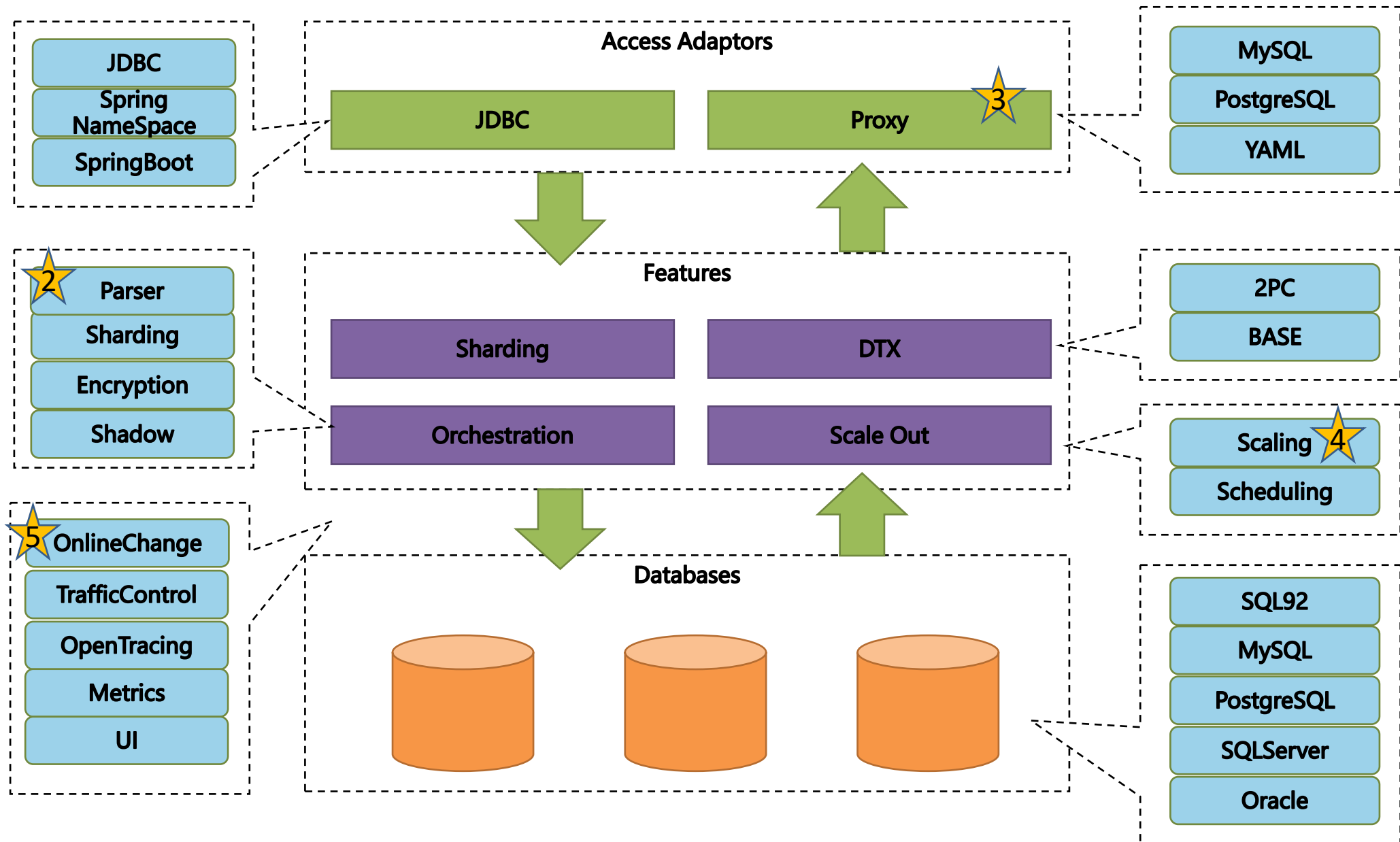
- 1 数据分片
- 2 分布式事务
- 3 分布式治理
- 4 弹性伸缩

# Architecture

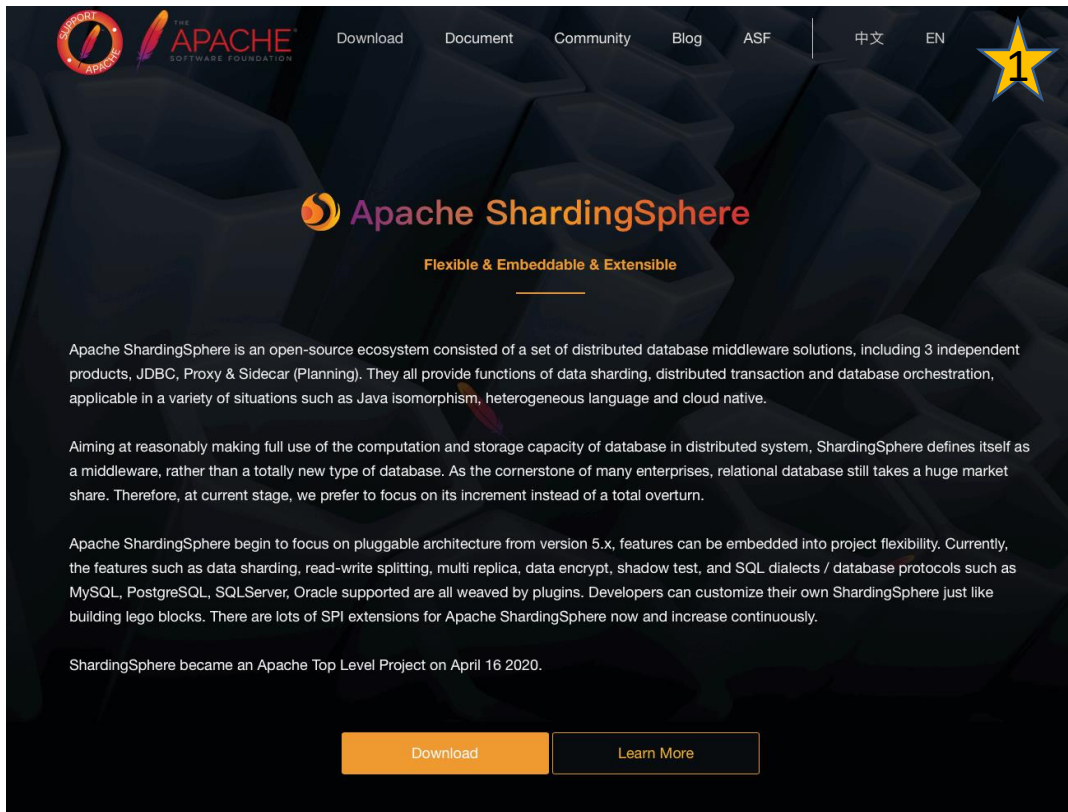




# Summer 2020 tasks



# Summer 2020 tasks



The screenshot shows the Apache ShardingSphere website. At the top, there's a navigation bar with links for Download, Document, Community, Blog, ASF, and language options (中文, EN). A yellow star with the number '1' is placed over the 'EN' link. The main header features the Apache ShardingSphere logo and the tagline 'Flexible & Embeddable & Extensible'. Below this, there are three paragraphs of text describing the project. The first paragraph states it's an open-source ecosystem for distributed database middleware. The second paragraph mentions its goal of making full use of computation and storage capacity. The third paragraph talks about its pluggable architecture and supported features like data sharding and replication. At the bottom, there are 'Download' and 'Learn More' buttons.

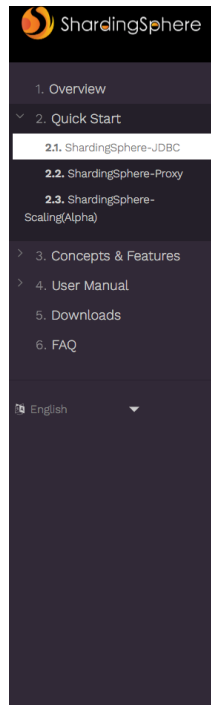
Apache ShardingSphere is an open-source ecosystem consisted of a set of distributed database middleware solutions, including 3 independent products, JDBC, Proxy & Sidecar (Planning). They all provide functions of data sharding, distributed transaction and database orchestration, applicable in a variety of situations such as Java isomorphism, heterogeneous language and cloud native.

Aiming at reasonably making full use of the computation and storage capacity of database in distributed system, ShardingSphere defines itself as a middleware, rather than a totally new type of database. As the cornerstone of many enterprises, relational database still takes a huge market share. Therefore, at current stage, we prefer to focus on its increment instead of a total overturn.

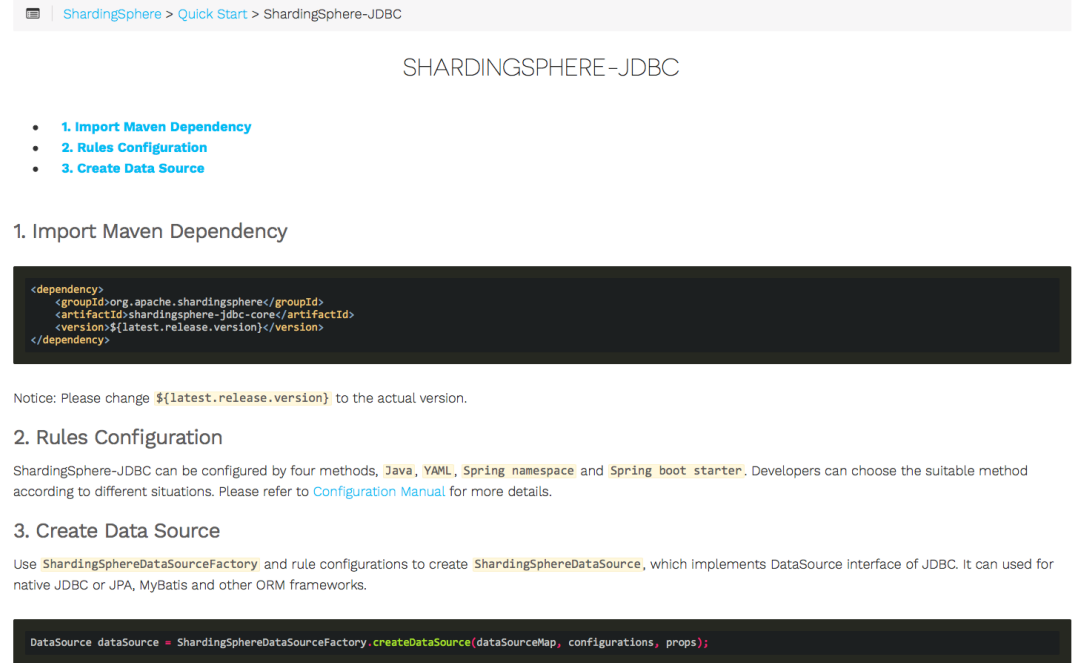
Apache ShardingSphere begin to focus on pluggable architecture from version 5.x, features can be embedded into project flexibility. Currently, the features such as data sharding, read-write splitting, multi replica, data encrypt, shadow test, and SQL dialects / database protocols such as MySQL, PostgreSQL, SQLServer, Oracle supported are all weaved by plugins. Developers can customize their own ShardingSphere just like building lego blocks. There are lots of SPI extensions for Apache ShardingSphere now and increase continuously.

ShardingSphere became an Apache Top Level Project on April 16 2020.

[Download](#) [Learn More](#)



This is a vertical navigation menu for ShardingSphere. It includes a logo at the top and a list of links: Overview, Quick Start (expanded), ShardingSphere-JDBC (2.1), ShardingSphere-Proxy (2.2), ShardingSphere-Scaling(Alpha) (2.3), Concepts & Features (3), User Manual (4), Downloads (5), and FAQ (6). There is also a language selector at the bottom set to 'English'.



The screenshot shows the 'SHARDINGSPHERE-JDBC' page in the 'Quick Start' section. It lists three steps: 1. Import Maven Dependency, 2. Rules Configuration, and 3. Create Data Source. Step 1 is highlighted. Below it, a code block shows the Maven dependency XML snippet. To the right of the code block, there's a note about replacing the version placeholder. Step 2 is also visible, mentioning four configuration methods: Java, YAML, Spring namespace, and Spring boot starter. Step 3 is partially visible, mentioning the use of ShardingSphereDataSourceFactory.

[ShardingSphere](#) > [Quick Start](#) > ShardingSphere-JDBC

## SHARDINGSPHERE-JDBC

- 1. [Import Maven Dependency](#)
- 2. [Rules Configuration](#)
- 3. [Create Data Source](#)

### 1. Import Maven Dependency

```
<dependency>
<groupId>org.apache.shardingsphere</groupId>
<artifactId>shardingsphere-jdbc-core</artifactId>
<version>${latest.release.version}</version>
</dependency>
```

Notice: Please change `${latest.release.version}` to the actual version.

### 2. Rules Configuration

ShardingSphere-JDBC can be configured by four methods, `Java`, `YAML`, `Spring namespace` and `Spring boot starter`. Developers can choose the suitable method according to different situations. Please refer to [Configuration Manual](#) for more details.

### 3. Create Data Source

Use `ShardingSphereDataSourceFactory` and rule configurations to create `ShardingSphereDataSource`, which implements `DataSource` interface of JDBC. It can be used for native JDBC or JPA, MyBatis and other ORM frameworks.

```
DataSource dataSource = ShardingSphereDataSourceFactory.createDataSource(dataSourceMap, configurations, props);
```

<https://cwiki.apache.org/confluence/display/SHARDINGSPHERE/Open+Source+Promotion+Plan+-+Summer+2020+Tasks>



# Thank you

<https://shardingsphere.apache.org/>

<https://github.com/apache/shardingsphere>

<mailto:dev-subscribe@shardingsphere.apache.org>

