

# Apache Pulsar

## 简介

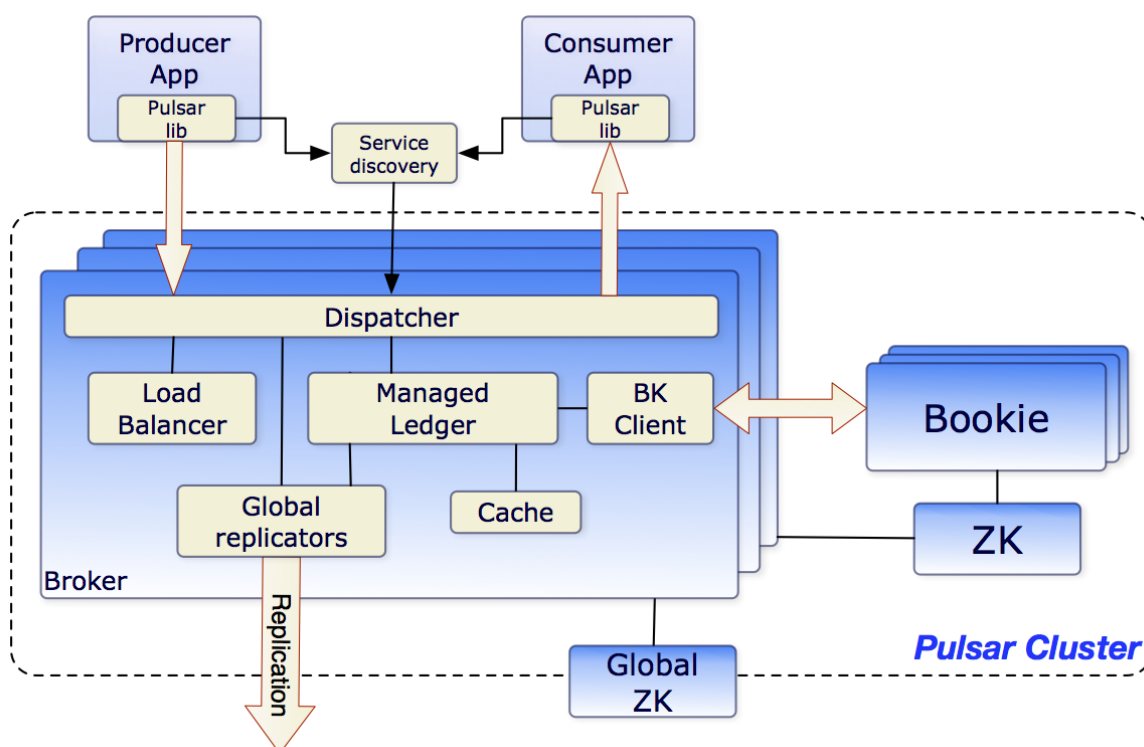
Apache Pulsar 是 Apache 软件基金会顶级项目，是下一代云原生分布式消息流平台，集消息、存储、轻量化函数式计算为一体。该系统源于 Yahoo，最初在 Yahoo 内部开发和部署，支持 Yahoo 应用服务平台 140 万个主题，日处理超过 1000 亿条消息。Pulsar 于 2016 年由 Yahoo 开源并捐赠给 Apache 软件基金会进行孵化，2018 年成为 Apache 软件基金会顶级项目。

Pulsar 作为下一代云原生分布式消息流平台，支持多租户、持久化存储、多机房跨区域数据复制，具有**强一致性、高吞吐**以及低延时的高可扩展流数据存储特性，内置诸多其他系统商业版本才有的特性，是云原生时代解决实时消息流数据传输、存储和计算的最佳解决方案。

Apache Pulsar 提供了统一的消费模型，支持 Stream（如 Kafka）和 Queue（如 RabbitMQ）两种消费模型，支持 exclusive、failover 和 shared 三种消费模式。同时，**Pulsar 提供和 Kafka 兼容的 API，以及 Kafka-On-Pulsar (KoP) 组件来兼容 Kafka 的应用程序**，KoP 在 Pulsar Broker 中解析 Kafka 协议，**用户不用改动客户端的任何 Kafka 代码就能直接使用 Pulsar。**

目前，Apache Pulsar 已经应用部署在国内外众多大型互联网公司 and 传统行业公司，案例分布在人工智能、金融、电信运营商、直播与短视频、物联网、零售与电子商务、在线教育等多个行业，如美国有线电视网络巨头**Comcast、Yahoo!**、**腾讯、中国电信、中国移动、BIGO、VIPKID** 等。

目前 Apache Pulsar 项目原生核心贡献者已组成创业公司 StreamNative，进一步为 Apache Pulsar 提供更好的企业级服务支持与生态建设。



上图是Pulsar Cluster的架构：

- 采用ZooKeeper存储元数据，集群配置，作为coordination
  - local zk负责Pulsar Cluster内部的配置等
  - global zk则用于Pulsar Cluster之间的数据复制等
- 采用Bookie作为存储设备(大多数MQ系统都采用本地磁盘或者DB作为存储设备)

- Broker负责负载均衡和消息的读取、写入等
- Global replicators负责集群间的数据复制

## Pulsar 安装部署【单机模式】

单机部署很简单，这里不再对每一个简单步骤进行说明。

步骤：

- 下载 pulsar

```
1 # 下载
2 wget https://mirrors.bfsu.edu.cn/apache/pulsar/pulsar-2.8.0/apache-
  pulsar-2.8.0-bin.tar.gz
3 # 解压
4 tar -zxvf apache-pulsar-2.8.0-bin.tar.gz
5 # 建立软链
6 ln -s /opt/soft/apache-pulsar-2.8.0 /usr/local/pulsar
```

- 启动单机模式 Pulsar

使用 `pulsar` 命令启动本地集群（该命令存储在 `bin` 目录中），并且可以指定为以单机模式启动 Pulsar

```
1 bin/pulsar standalone
```

成功启动 Pulsar 后，可以看到 `INFO` 级日志消息。

- 使用单机模式 Pulsar

Pulsar 中有一个名为 `pulsar-client` 的 CLI 工具。Pulsar-client 工具允许使用者在运行的集群中 consume 并 produce 消息到 Pulsar topic。

### Consume 一条消息

在 `first-subscription` 订阅中 consume 一条消息到 `my-topic` 的命令如下所示：

```
1 bin/pulsar-client consume my-topic -s "first-subscription"
```

如果消息成功发送到 topic，则会在 `pulsar-client` 日志中出现一个确认，如下所示：

```
1 09:56:55.566 [pulsar-client-io-1-1] INFO
  org.apache.pulsar.client.impl.MultiTopicsConsumerImpl -
  [TopicsConsumerFakeTopicNameee2df9] [first-subscription] Success subscribe
  new topic my-topic in topics consumer, partitions: 4,
  allTopicPartitionsNumber: 4
```

### Produce 一条消息

向名称为 `my-topic` 的 topic 发送一条简单的消息 `hello-pulsar`，命令如下所示：

```
1 bin/pulsar-client produce my-topic --messages "hello-pulsar"
```

如果消息成功发送到 topic，则会在 `pulsar-client` 日志中出现一个确认，如下所示：

```
1 13:09:39.356 [main] INFO org.apache.pulsar.client.cli.PulsarClientTool -  
1 messages successfully produced
```

- 终止单机模式 Pulsar

使用 `Ctrl+C` 终止单机模式 Pulsar 的运行。

如果服务使用 `pulsar daemon start standalone` 命令作为后台进程运行，则使用 `pulsar daemon stop standalone` 命令停止服务。

## Pulsar 安装部署【集群模式】

### 1. 准备资源

- 一台主机
- 3 台 Linux 虚拟机服务器，配置 1G，2核
- JDK 8 运行环境
- Pulsar 最新版本安装包

### 2. 集群说明

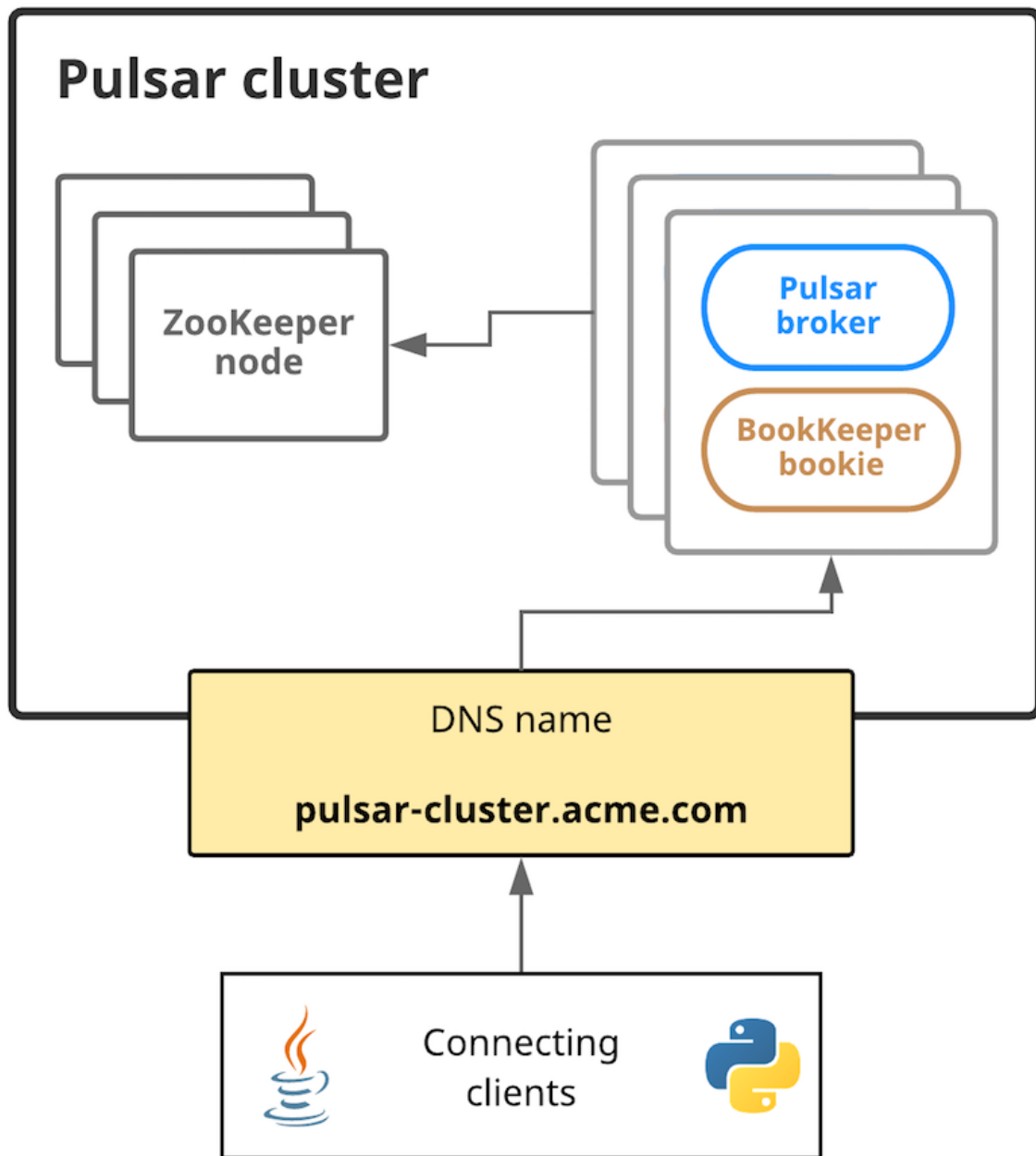
1. 搭建 Pulsar 集群至少需要 3 个组件：

- **ZooKeeper 集群** (3 个 ZooKeeper 节点组成)
- **bookie 集群** (也称为 BookKeeper 集群，3 个 BookKeeper 节点组成)
- **broker 集群** (3 个 Pulsar 节点组成，broker 是 Pulsar 的自身实例)

2. Pulsar 的安装包已包含了搭建集群所需的各个组件库。无需单独下载 ZooKeeper 安装包和 BookKeeper 安装包。

3. 3 台 Linux 服务器 IP 分别为 192.168.18.22，192.168.18.24 和 192.168.18.27。

默认关闭服务器防火墙。



上面是一个显示部署架构的图。

在图中，客户端必须能够使用单个 URL 连接到 Pulsar 集群。在这种情况下，`pulsar-cluster.acme.com` 包含了所有的broker 的地址。Pulsar消息brokers与BookKeeper bookies一起运行；反过来，brokers 和 bookies 都依赖 ZooKeeper。

### 3.安装JDK8

在 3 台 Linux 服务器上安装JDK（要求版本不低于 JDK 8），并配置环境变量。

这里就不再详细介绍安装 jdk 步骤

### 4.Pulsar 二进制包安装

```
1 # 下载
2 wget https://mirrors.bfsu.edu.cn/apache/pulsar/pulsar-2.8.0/apache-pulsar-2.8.0-bin.tar.gz
3 # 解压
4 tar -zxvf apache-pulsar-2.8.0-bin.tar.gz
5 # 建立软链
6 ln -s /opt/soft/apache-pulsar-2.8.0 /usr/local/pulsar
```

## 5.配置部署Zookeeper集群

这里使用前面安装 kafka 的时候的 zookeeper 集群，不再介绍。

如果你规划 Zookeeper 和 Bookie部署在同一台机器上，你必须配置 zookeeper 监听其他的端口，或者停用内嵌的管理服务，然后再启动。

```
1 # 修改端口【方案一】
2 -Dzookeeper.admin.serverPort=8001
3 # 禁用服务【方案二】
4 -Dzookeeper.admin.enableServer=false
5 # 修改zoo.cfg,禁用服务【方案三】我们采用这种
6 admin.enableServer=false
```

启动 zookeeper

```
1 bin/zkServer.sh start
```

## 6.初始化集群元数据

在任一个 zooKeeper 节点上执行一次就行

```
1 # 进入Apache-pulsar 目录，执行命令初始化集群元数据
2 bin/pulsar initialize-cluster-metadata \
3 --cluster pulsar-cluster \
4 --zookeeper 192.168.18.22:2181 \
5 --configuration-store 192.168.18.22:2181/ledgers \
6 --web-service-url http://pulsar.cluster.com:8080 \
7 --web-service-url-tls https://pulsar.cluster.com:8443 \
8 --broker-service-url pulsar://http://pulsar.cluster.com:6650 \
9 --broker-service-url-tls pulsar+ssl://http://pulsar.cluster.com:6651
```

- 查看zookeeper配置是否成功

```
1 # 执行 ZooKeeper 客户端连接命令
2 bin/zkCli.sh
3 # Enter键，使用 ZooKeeper 命令查看 ls /ledgers。如果有数据表示成功
```

从上面的示例中可以看出，您需要指定以下内容：

标记	说明
<code>--cluster</code>	集群名字
<code>--zookeeper</code>	A "local" ZooKeeper connection string for the cluster. This connection string only needs to include <i>one</i> machine in the ZooKeeper cluster.
<code>--configuration-store</code>	整个集群实例的配置存储连接字符串。 As with the <code>--zookeeper</code> flag, this connection string only needs to include <i>one</i> machine in the ZooKeeper cluster.
<code>--web-service-url</code>	The web service URL for the cluster, plus a port. This URL should be a standard DNS name. The default port is 8080 (you had better not use a different port).
<code>--web-service-url-tls</code>	If you use <a href="#">TLS</a> , you also need to specify a TLS web service URL for the cluster. The default port is 8443 (you had better not use a different port).
<code>--broker-service-url</code>	Broker服务的URL，用于与集群中的brokers进行交互。这个 URL 不应该使用和 web 服务 URL 同样的 DNS名称，而应该用 <code>pulsar</code> 方案。默认端口是6650(我们不建议使用其他端口)。
<code>--broker-service-url-tls</code>	如果使用 <a href="#">TLS</a> ，你必须为集群指定一个 TLS web 服务URL，以及用于集群中 broker TLS 服务的URL。默认端口是6651(不建议使用其他端口)。

## 7.配置部署 BookKeeper 集群

### 1.修改配置文件 bookkeeper.conf

通过配置文件 [conf/bookkeeper.conf](#) 去配置 BookKeeper bookies。配置 bookies 最重要的一步，是要确保 `zkServers` 设置为 Zookeeper 集群的连接信息。

```

1 vim bookkeeper.conf
2 # 修改zk地址和端口信息
  zkServers=192.168.18.22:2181,192.168.18.24:2181,192.168.18.25:2181
3 zkServers=192.168.18.22:2181,192.168.18.24:2181,192.168.18.25:2181

```

### 2. 启动 bookie 集群

```

1 # 以后台进程启动bookie
2 bin/pulsar-daemon start bookie
3
4 # 控制台启动bookie
5 bin/pulsar bookie

```

你可以通过运行BookKeeper shell上的 `bookiesanity` 命令验证 bookie 是否正常工作：

```

1 # 验证是否启动成功
2 bin/bookkeeper shell bookiesanity

```

这个命令会在本地的 bookie 创建一个临时的 BookKeeper ledger，往里面写一些条目，然后读取它，最后删除这个 ledger。

启动了所有的 bookie 后，你能够在任意一台bookie上，使用[BookKeeper shell]的 `simpletest` 命令，去校验集群内所有的 bookie 是否都已经启动。

## 8.部署配置 Pulsar broker 集群

### 1. 修改配置文件 broker.conf

```
1 # 修改集群名，和 zookeeper 里初始化元数据时指定的集群名(--cluster pulsar-cluster)相同
  clusterName=pulsar-cluster
2 # 修改如下两个配置，指定的都是 zookeeper 集群地址和端口号
3 zookeeperservers=192.168.18.22:2181,192.168.18.24:2181,192.168.18.25:2181
4 configurationStoreServers=192.168.18.22:2181,192.168.18.24:2181,192.168.18.25:2181
5 # 修改如下参数为本服务器ip地址，另外两个 broker 节点配置文件也做对应修改
6 advertisedAddress=192.168.18.22
```

### 2. 启动 broker 节点。

```
1 # 以后台进程启动 broker
2 bin/pulsar-daemon start broker
3
4 # 查看集群 brokers 节点情况
5 bin/pulsar-admin brokers list pulsar-cluster
```

至此，集群 ZooKeeper，Broker，Bookie 节点启动完毕，集群部署成功。

## 9.连接集群

`pulsar-client` 工具能够从 Pulsar 主题生产消息和消费消息，并提供了一个简单的方式来确保你的集群运行正常。

To use the `pulsar-client` tool, first modify the client configuration file in `conf/client.conf` in your binary package. 你需要修改 `webServiceUrl` 和 `brokerServiceUrl` 的值，将默认 `localhost` 值替换为你给 broker/bookie 指定的 DNS 名称。如下所示：

```
1 webServiceUrl=http://pulsar.cluster.com:8080
2 brokerServiceUrl=pulsar://pulsar.cluster.com:6650
```

如果没用dns就修改hosts文件

```
1 vim /etc/hosts
2 # 增加映射关系
3 192.168.18.22 pulsar.cluster.com
```

完成后，你能够发送消息到 Pulsar 主题：

```
1 bin/pulsar-client produce \  
2   persistent://public/default/test \  
3   -n 1 \  
4   -m "Hello Pulsar"
```

消费:

```
1 bin/pulsar-client consume \  
2   persistent://public/default/test \  
3   -n 100 \  
4   -s "consumer-test" \  
5   -t "Exclusive"
```

一旦您成功地向主题发送了上面的消息，您应该在标准输出中看到它:

----- got message -----

key:[null], properties:[], content:Hello Pulsar