**mirrorship**

**部署文档**

|  |  |
| --- | --- |
| **文档版本** | 1.0 |
| **发布日期** | 2023-03-24 |



目 录

[1 产品介绍 3](#_Toc130565325)

[1.1 产品介绍 3](#_Toc130565326)

[1.2 典型业务场景 4](#_Toc130565327)

[2 系统架构 6](#_Toc130565328)

[2.1 逻辑架构 6](#_Toc130565329)

[2.2 部署架构 6](#_Toc130565330)

[2.3 部署清单和环境配置要求 7](#_Toc130565331)

[2.3.1 小规模场景 7](#_Toc130565332)

[3 成本与规格 7](#_Toc130565333)

[3.1 软件计费规格 7](#_Toc130565334)

[4 运维与支持 8](#_Toc130565335)

[4.1 补丁与升级 8](#_Toc130565336)

[4.2 许可管理 10](#_Toc130565337)

[4.3 故障排除 11](#_Toc130565338)

[4.4 支持SLA 12](#_Toc130565339)

[5 安全与隐私 13](#_Toc130565340)

[5.1 漏洞管理机制 13](#_Toc130565341)

[6 用户指南 14](#_Toc130565342)

[6.1 成功案例 15](#_Toc130565343)

[6.2 使用手册 15](#_Toc130565344)

# 产品介绍

## 产品介绍

Mirrorship 是⼀款极速全场景 MPP 企业级数据库产品，具备⽔平在线扩缩容，⾦融级⾼可⽤，兼容 MySQL 5.7 协议和 MySQL ⽣态，提供全⾯向量化引擎与多种数据源联邦查询等重要特性。Mirrorship 致⼒于在全场景 OLAP 业务上为⽤户提供统⼀的解决⽅案，适⽤于对性能、实时性、并发能⼒和灵活性有较⾼要求的各类应⽤场景。

Mirrorship 在设计之初就秉承了全新的设计理念，致⼒于在极致的性能体验下，满⾜⼤多数分析场景的需求。⽬前 Mirrorship 已在多个⾏业落地，在⾃助数据分析、实时数仓、⽤户画像、实时⻛控、订单分析等场景都有优秀的表现。

产品基本功能

1. Mirrorship 数据分布

在 Mirrorship 中数据以列的形式存储。⼀张表的列可以分成维度（key）列与指标（value）列。维度列⽤于分组和排序，指标列可以参与函数聚合。表中的数据是按照维度列进⾏排序的，在等值或者范围查询中，可以充分利⽤维度列的有序性，快速锁定需要检索的⾏。

1. Mirrorship 数据模型

在建表时可以选择不同的数据模型⽀撑不同的业务类型。⽬前 Mirrorship 提供⽤于存储明细数据的明细模型（Duplicate Key），⽤于预聚合的聚合模型（Aggregate Key）及⽤于表更新的主键模型（Primary Key）。选择合适的模型有利于数据建模与数据的读写提速。

1. Mirrorship ⾼并发能⼒

Mirrorship 提供优秀的⾼并发查询能⼒。通过合适的分区分桶策略及⼀定规模的节点数量，可以⽀撑上万的查询并发量。在经过数据分区分桶后，数据的指向性有了显著的提⾼。良好的分区分桶规则可以将业务逻辑相关的数据集中存储。当分区分桶的规则能够覆盖⼤部分的查询条件时，可以利⽤分区剪裁、分桶剪裁及前缀索引等⼿段，有效的减少数据的扫描量，精准定位并抓取查询数据。Mirrorship 提供了多种数据存储优化⼿段以避免全表扫描，减少查询对硬件资源的消耗，从⽽节省出更多的硬件资源⽀撑⾼并发的业务查询

1. Mirrorship 精简运维

Mirrorship 集群采⽤分布式架构，存储容量和计算能⼒可以近乎线性⽔平扩展。Mirrorship 单机群的节点规模可扩展到数百节点，数据规模可以达到 10PB 级别。扩容期间⽆需停服，可以正常提供查询与读写服务。Mirrorship 提供表的热变更功能，通过 SQL 命令可以动态修改表的定义，如增删列、创建物化视图等。

1. CBO：基于代价的优化器

CBO（Cost Based Optimizator），是指基于成本的优化器。与⼤部分使⽤了 RBO（Rule Based Optimizator）的 MPP 架构计算引擎不同，Mirrorship 使⽤ CBO 优化器，⽆需考虑 SQL 的预定优化规则，如常⻅的驱动表与被驱动表前后顺序，不同选择性谓词条件的前后顺序等。通过使⽤ Mirrorship CBO 优化器，可以将复杂繁琐的 SQL 优化⼯作交给优化器引擎，释放更多的时间处理 SQL 业务逻辑。

1. 资源隔离

Mirrorship ⽀持资源隔离，通过限制查询任务对计算资源的消耗，实现资源隔离和合理利⽤。资源隔离功能可为不同租户的查询任务在同⼀集群执⾏提供保障，使系统能够按需分配每⼀个⽤户的资源配额，避免不租户之间的资源争⽤。

1. 实时多表关联能⼒

在业务⽅建表时，为了和前端业务适配，传统建模⽅式直接将 schema 定义成宽表。 在 Mirrorship 中可以选择更灵活的星型模型来替代⼤宽表，如⽤⼀个视图来取代宽表进⾏建模，直接使⽤多表关联来查询。在 SSB 的标准测试集的对⽐中，Mirrorship 的多表关联性能⽐单表查询相差不⼤。

## 典型业务场景

1. **小红书**

小红书为解决数据逻辑没有很好做归拢合并，维护工作量大，新需求无法快速响应。Clickhouse的并发能力不足以及扩容复杂度在可见未来会成为整体广告系统瓶颈。Apache Flink®层逻辑落，由大量小的Flink任务构成，因此导致整个架构无法满足高可用要求，只要任何一个任务出现问题，都会影响线上业务的以上痛点从而引入MirrorShip，

因为MirrorShip具备能支撑大吞吐量的数据写入要求。可以支持多维度组合的灵活查询，TP99在100ms以下。有实时汇总上卷的能力，提高查询性能，支持qps达到上万的要求。通过Binlog实时同步MySQL的数据，并及时对数据进行封装。比较好的支持多表关联。

研引入MirrorShip，当前已经有五个MirrorShip集群在稳定运行中小红书目前有18个集群，节点数达到147，完成了PB级数据量的导入，其中有两个开始稳定提供线上服务，三个还在试运行。引入MirrorShip后，实现了数据服务统一化，大大简化了实时数据处理链路，同时也能保障较高的查询并发和较低的响应延迟要求，之后将用来提升更多业务场景的数据服务和查询能力。

1. **中国邮政**

中国邮政面临业务痛点为离线/实时ETL任务过多，处理逻辑大部分为简单聚合/去重，聚合表数量庞大，导致运营和运维上的成本增加；针对中等数据量、中等QPS的查询场景，如何能兼顾数据规模的同时，有较友好的查询响应延迟；大数据量下插入、更新的实时数据场景无法得到支持，例如：网点画像、实时数据导入、邮路路径、研报数据汇总等。

针对如上的问题和挑战，未解决以上业务痛点，引入MirrorShip，尽可能少的OLAP引擎，利用在明细表上现场计算来解决ETL任务、数仓表过多问题，同时需要兼顾在数据规模、查询QPS、响应耗时、查询场景方面的权衡。目前集群规模达到120个节点，数据量超过300T。

邮科院数据组引入MirrorShip生产集群，解决了数据服务层单表亿级别规模、高QPS数据场景下引擎的空白，直接开放明细表准实时查询的能力，给各项目组上层数据业务和BI系统提供了更多的选择和自由度，同时将大大减少数仓中大量ETL任务、聚合表、报表，降低了数仓ETL的运维压力和维护成本，MirrorShip综合性价比较原有的MySQL、Presto、ClickHouse等同类产品提升数倍以上。

1. **滴滴**

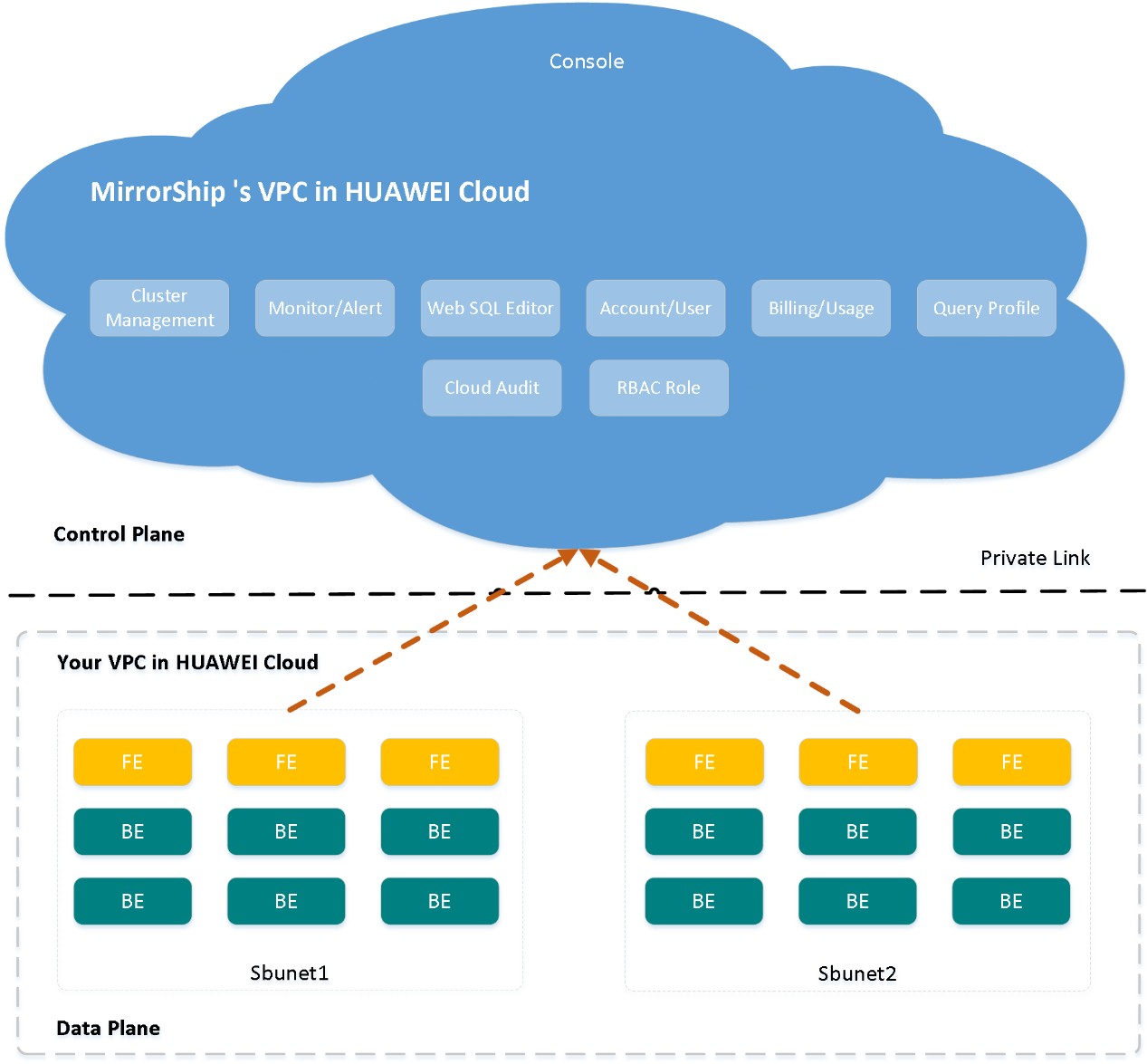
滴滴需要面对日增数据量大：日增千万级数据，支持灵活选择维度，快速地对亿级数据量进行多维分析 ；对数据分析时效性要求高：快速地基于亿级数据量精确去重，获取符合条件的用户数量。

基于此，客户搭建了3套MirrorShip集群，集群节点数达到120个，总数据量逐步增加，目前已经有300T数据。使用了MirrorShip的明细模型来建表，满足用户查询漏斗明细数据的使用场景，在明细表上根据不同的多维漏斗分析查询需求创建相应的物化视图，来满足用户选择不同维度查看漏斗模型每一步骤用户精确去重数量的使用场景。

在使用MirrorShip后，运维简单，成本低。由于MirrorShip同时支持明细和聚合模型，可以满足大多数场景，之前采用的多种引擎构建数据中心的架构，现在可以采用 MirrorShip 作为唯一引擎，架构简单明了，运维高效便捷。MirrorShip 相比于 Apache Spark™ 引擎，机器成本降低50%以上。第二 MirrorShip 查询性能优越，MirrorShip 近乎实时的查询性能，针对很多典型场景进行优化的各种特性（Colocate Shuffle Join，Bucket Shuffle Join，CBO等），给用户带来了良好的使用体验。第三 MirrorShip 既可存算一体，也可存算分离。目前 MirrorShip 是存算一体的系统，但它同时支持 ES/MySQL/Apache Hive™ 等外表功能，可以实现对 Apache Hadoop® 生态的查询，可以做到存算分离，对于节省成本，打通 Apache Hadoop® 生态很有意义。

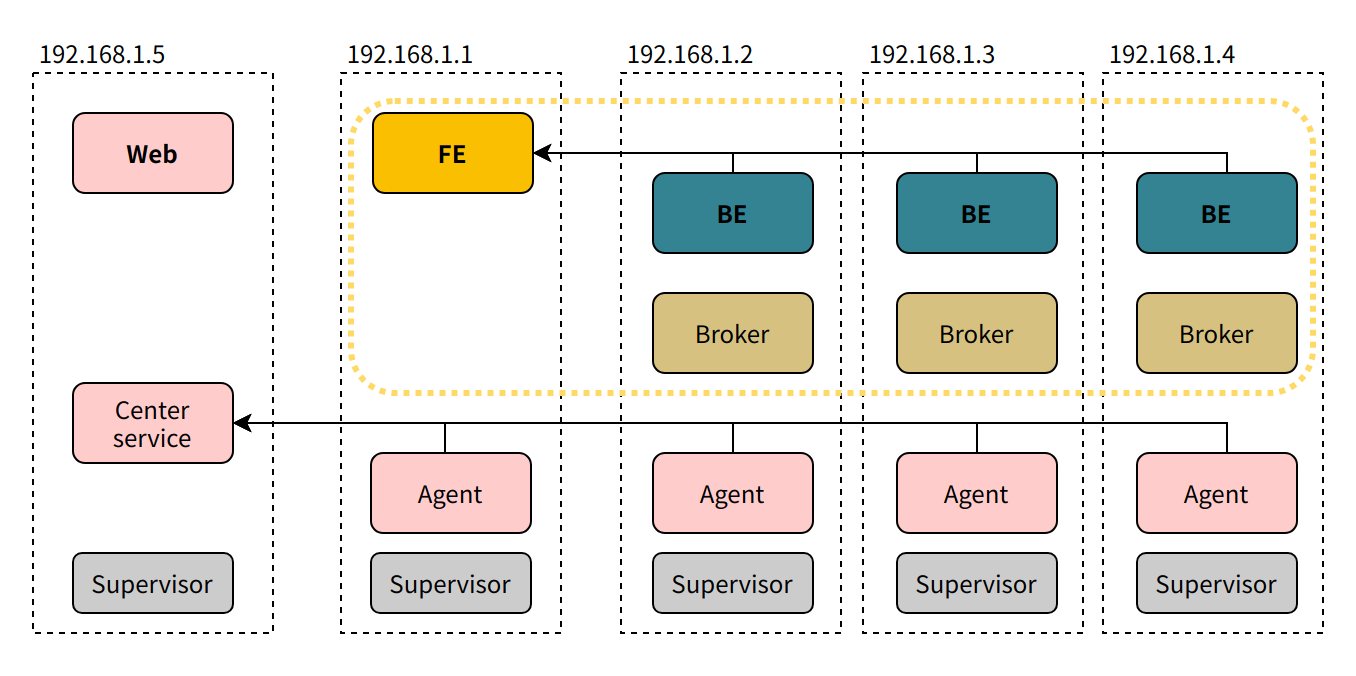
# 系统架构

## 逻辑架构

**

## 部署架构

产品部署可采用高可用模式



## 部署清单和环境配置要求

### 小规模场景

Manager服务器一台：c7.2xlarge.2 CentOS7 或以上， 500G（通用性 SSD gp3），CPU：8核 内存：8G 。

Fe节点服务器一台：c7.2xlarge.2 CentOS7 或以上， 500G（通用性 SSD gp3），CPU：8核 内存：16G。

Be节点服务器三台：c7.2xlarge.2 CentOS7 或以上， 500G（通用性 SSD gp3），CPU：16核 内存：32G。

# 成本与规格

## 软件计费规格

统一的市场定价为：￥【3000 】元/vCPU核/年（含税）

# 运维与支持

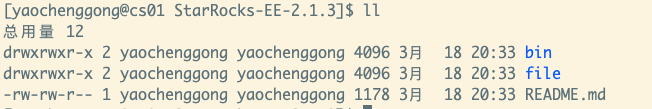
## 补丁与升级

**1 升级步骤**

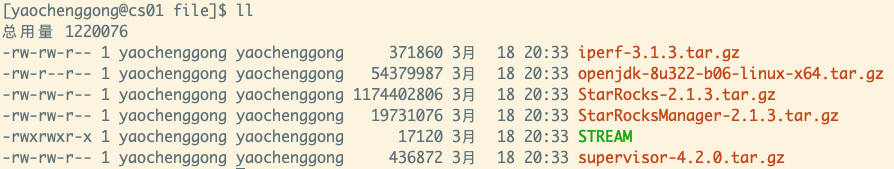
可以在 Mirrorship 官网获取对应的 Enterprise Edition (EE) 版本安装包或者联系 Mirrorship 技术人员获取，获取之后进行解压。

|  |
| --- |
| Shell  $ tar -zxvf Mirrorship-EE-x.x.x.tar.gz  $ cd Mirrorship-EE-x.x.x/file  $ ls -l |

解压完成后在 **file** 目录下会看到 **Mirrorship-x.x.x.tar.gz** 和 **MirrorshipManager-x.x.x.targ.z** 两个安装包。其中x.x.x 是3位版本号，第1位是大版本，第2位是小版本，小版本一般两周左右发布一次，您可以在 [Mirrorship社区版讨论区](https://forum.starrocks.com/c/22-category/22) 了解每个版本发布的release notes，第三位是bugfix版本。举例：

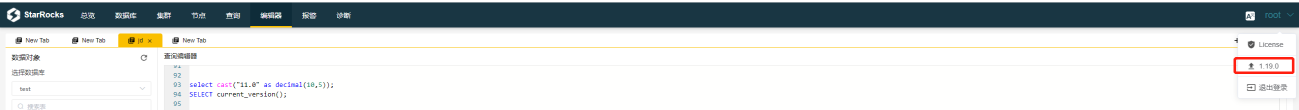


**file** 目录下的文件如下（后续升级需要选择此目录下的压缩包）：

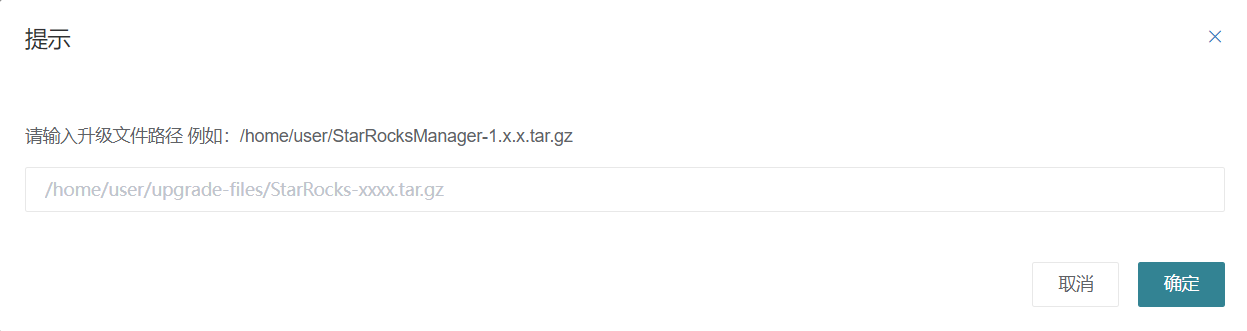


**2 升级 Mirrorship Manager**

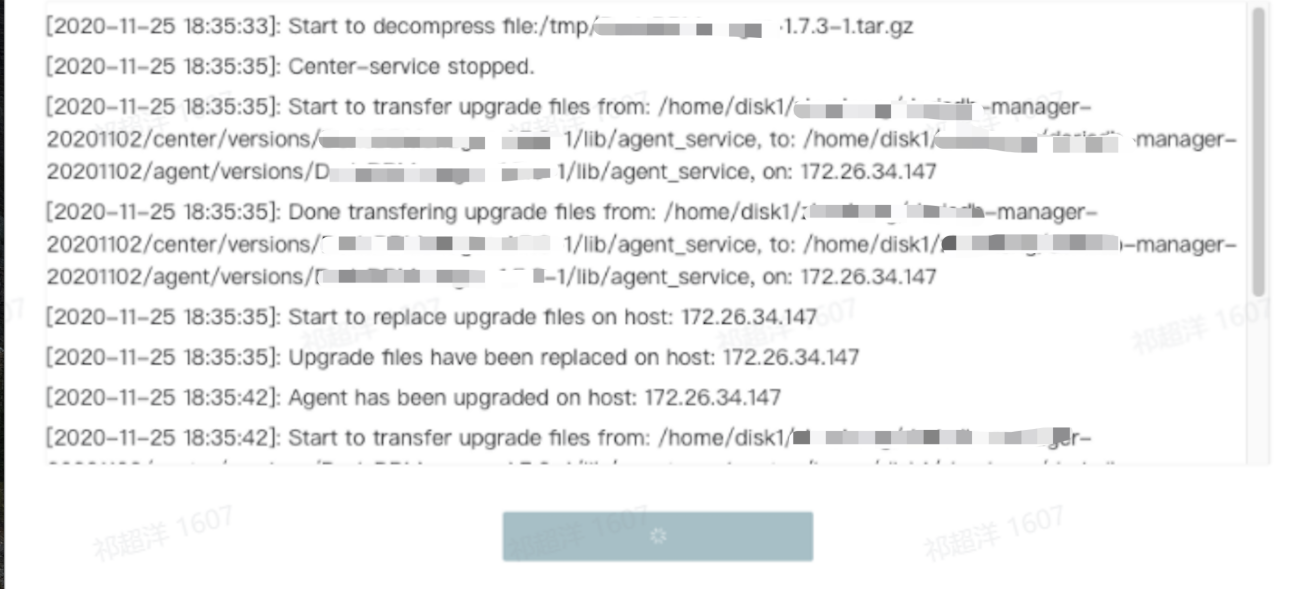
升级步骤为先升级 Mirrorship Manager，再升级 Mirrorship 节点版本。升级 Mirrorship Manager 不影响集群业务，升级成功与否都只跟Web界面展示有关。Mirrorship Manager 管理集群会有新版本功能兼容的要求，所以需要先升级Mirrorship Manager，再升级Mirrorship节点版本。



1. 单击右上角**root**，单击如图所示的升级按钮。
2. 输入解压后**file**文件夹下的 **MirrorshipManger-xxx.tar.gz** 文件路径。



1. 确认文件上传正确后，单击**确定**，自动执行升级。

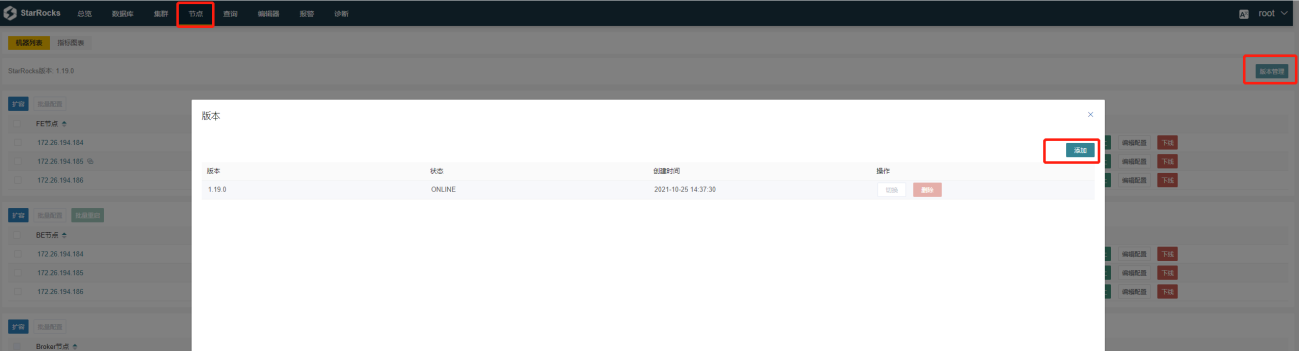


升级成功后请务必确认Mirrorship Manager版本是升级后的版本，再继续后续操作。

**3 升级 Mirrorship**

|  |
| --- |
| * 确保 Mirrorship Manager 升级成功后，再执行升级 Mirrorship 集群的操作。 |

1. 选择 **节点** > **版本管理。**在弹出的**版本**页面中，单击**添加**按钮添加Mirrorship版本。



1. 输入解压后**file**文件夹下的**Mirrorship-1.x.x.tar.gz**文件路径。
2. 确认路径正确后，单击**确定，**系统自动进行版本安装。



1. 完成后单击**切换**即可进行Mirrorship升级操作。





升级成功。

## 许可管理

产品需要使用对应的 license 才可正常进行使用。

客户需根据自身需求和商务合同情况，提供相关信息：

1. License 有效时间：固定期限
2. 公司名称及邮箱
3. 资源控制情况：CPU核数
4. 华为云为每个账户维护服务配额（以前称为服务限制），以帮助确保 华为云资源的可用性并防止意外预置超出需要的资源。有些服务配额会在您使用 华为云时随着时间自动提高。但是大多数华为云服务需要您手动请求提高配额。

## 故障排除

**Python 安装错误**

如果在安装Python时遇到问题： `\_\_init\_\_() takes 2 arguments (4 given) `，请执行如下步骤：

1. 执行 which python 命令确认 Python 安装路径为 **/usr/bin/python**。
2. 删除python-setuptools 安装包：

|  |
| --- |
| Shell  yum remove python-setuptools |

1. 删除setuptool相关文件。

|  |
| --- |
| Shell  rm /usr/lib/python2.7/site-packages/setuptool\* -rf |

1. 获取**ez\_setup.py**文件。

|  |
| --- |
| Shell  wget <https://bootstrap.pypa.io/ez_setup.py> -O - | python |

**UTC 错误**

如果在配置中心服务过程中出现时区UTC错误，将 **export TZ='Asia/Shanghai';** 添加到 **~/.bashrc** 文件中执行**。**该操作设置环境变量 TZ 为 Asia/Shanghai，将该设置加入到系统变量里。

安装**web**服务前确认时区为**cst。**

[ycg@Mirrorship-sandbox04 ~]# export TZ=Asia/Shanghai

[ycg@Mirrorship-sandbox04 ~]# date

Sat Apr 10 10:04:46 CST 2021

## 支持SLA

|  |  |  |
| --- | --- | --- |
| 技术服务类型 | 描述 | 支持时间 |
| 产品技术支持 | 为甲方提供集群安装部署支持以及因产品BUG导致的问题定位和技术支持，以确保Mirrorship稳定运行，具体包括：  常规Mirrorship数据库集群安装部署；  提供持续交付的产品补丁，漏洞修复，产品更新，大小版本升级；  Linux系统内核参数调优配置指导；  Mirrorship配置参数调优配置指导；  数据备份与恢复指导；  集群扩缩容指导；  慢SQL查询的调优建议；  系统巡检（一月一次含巡检报告）。 | 5\*10 |
| 数据库故障响应 | 采用远程方式对数据库故障进行技术响应。响应时间和故障级别定义参见《表2：故障等级界定及响应时间》  注：P1、P2、P3故障响应时间仅限工作日9:00~19:00。 | 5\*10 |
| 应急技术支持 | 对P0级故障提供7\*24小时远程专属应急响应通道支持。  应急响应通道包括：电话、微信、腾讯会议等即时通讯工具；远程桌面、TeamViewer、VPN等远程工具。 | 7\*24 |

备注：

1）5\*10小时是指工作日9:00~19:00；

2）7\*24小时是指任意时间；

日常时间故障等级界定及响应间

|  |  |  |
| --- | --- | --- |
| 故障等级 | 故障定义 | 响应时间 |
| P0 | 软件无法启动或启动后无法进行任何操作，业务生产中断 | <=1小时 |
| P1 | 出现的故障对业务有轻微的影响，在受限情况下仍可继续操作或使用。 | <=1小时 |
| P2 | 出现的故障不直接影响业务，但可能影响数据库的管理和维护。 | <=2工作日 |
| P3 | 软件使用过程中的需求和改进建议，对业务影响较小。 | <=2工作日 |

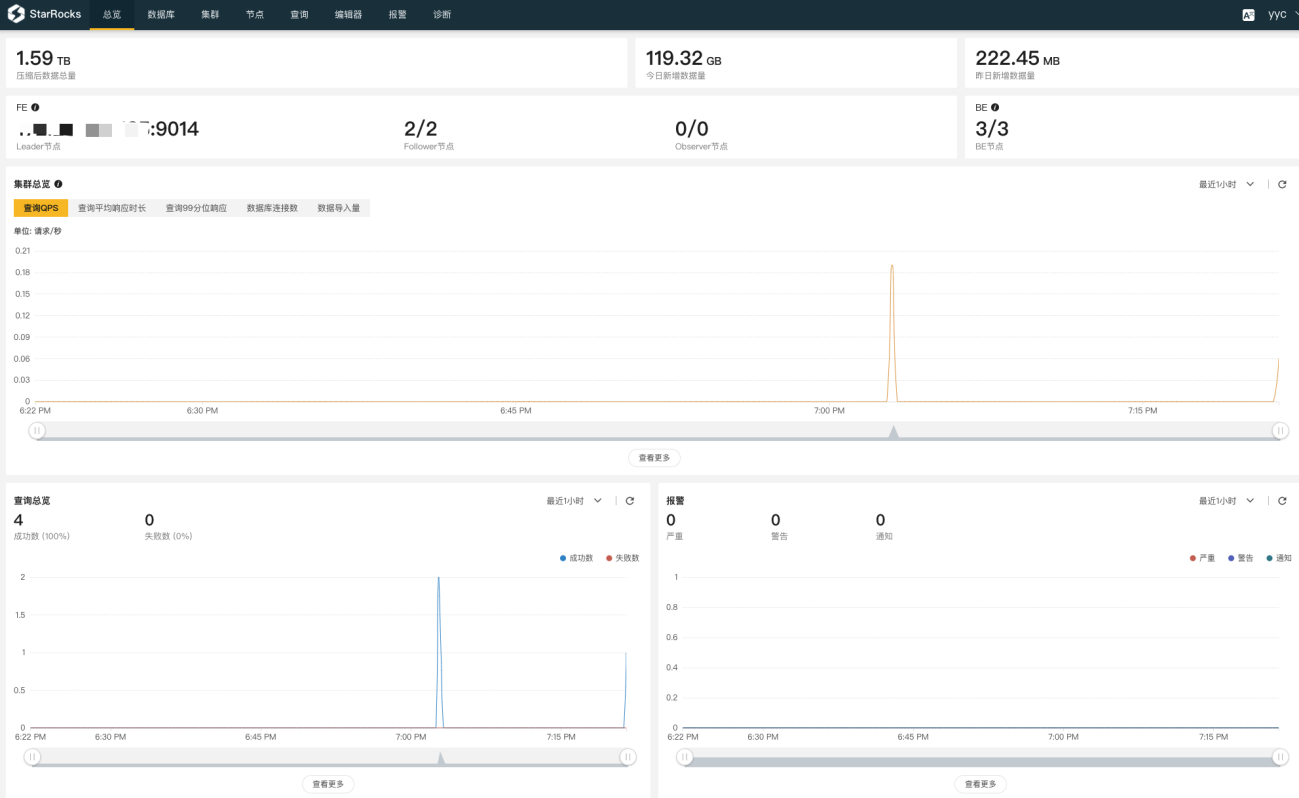
1. 电话支持：提供专门的热线电话响应用户的售后技术支持请求，电话号码：400-608-5626；
2. 邮件支持：提供专门的售后支持邮箱响应用户的售后技术支持请求；
3. 远程支持：提供远程支持的工具，通过该工具可以实时连接到在线的客户设备上进行远程支持和诊断；
4. 现场支持：对于通过以电话，邮件和网站方式无法解决的问题，根据问题性质和级别提供相应的工程师现场服务；
5. 产品培训：提供的相关运维培训；
6. 数据库售后的技术服务的技术手段包含但不限于以上手段。

# 安全与隐私

## 漏洞管理机制

完成安装后可以使用 root 用户名和空密码访问控制台， 登录后建议立即修改密码并创建用户。

|  |
| --- |
| SQL  SET PASSWORD FOR 'root'@'%' = PASSWORD('Mirrorship2021') |



Mirrorship Manager 提供如下功能：

* **总览**： 集群整体概况，包括集群FE/BE信息、集群数据量、导入量、查询、报警、网络、磁盘IO等
* **数据库**：查看数据库以及表、任务、分片等元数据信息
* **集群**：各类集群监控信息
* **节点**：各类节点详细监控信息
* **查询**：查询/慢查询历史记录
* **编辑器**：交互式查询控制台
* **报警**：报警监控页面
* **诊断**：集群硬件检测工具

# 用户指南

## 成功案例

详见产品文档：

中信建投：<https://www.mirrorship.cn/zh-CN/blog/zhongxinjiantou>

中原银行：https://www.mirrorship.cn/zh-CN/blog/zhongyuanyinhang

顺丰科技：https://www.mirrorship.cn/zh-CN/blog/shunfengkeji

## 使用手册

**1. 安装Mirrorship Manager**

1. 下载并解压 **Mirrorship-EE-x.x.x.tar.gz** 安装包。

|  |
| --- |
| Apache  tar -zxvf Mirrorship-EE-x.x.x.tar.gz |

1. 安装Web页面和Center servcie。

解压完成后，需要到解压目录下，执行一个**install\_path**的脚本来生成一个web界面，进行集群的Mirrorship Manager安装。

**install\_path** 是 Mirrorship Manager 在机器上的安装目录，一般默认即可。如果您需要安装多个 Mirrorship Manager 到同一台机器或需要进行目录定制，可以根据需要修改**安装目录**或者**端口。**

“安装多个 Mirrorship Manager 到同一台机器”的含义：可以在同一台机器上部署多个Mirrorship Manager来管理不同的集群。）

|  |
| --- |
| Bash  cd Mirrorship-EE-x.x.x  sh bin/install.sh -h |

**install.sh** 支持的配置如下：

|  |
| --- |
| Apache  -[d install\_path] install\_path(default: /home/disk1/mirrorship/Mirrorship-manager-20200101)  -[y python\_bin\_path] python\_bin\_path(default: /usr/bin/python)  -[p admin\_console\_port] admin\_console\_port(default: 19321)  -[s supervisor\_http\_port] supervisor\_http\_port(default: 19320) |

对于一个机器上安装多个 Mirrorship Manager 的场景，注意设置每个 Mirrorship Manager 的安装路径，一般以集群名或 IP 区别。

举例：如下命令用于修改Mirrorship Manager 的文件目录及端口。

|  |
| --- |
| Apache  sh bin/install.sh -d /home/disk1/SRManager-Manager -p 19123 -s 19023 |

执行成功后会生成指定的目录文件。

举例：生成的**SRManager-Manager**目录下有如下3个文件，这个目录是Center service的安装目录。

|  |
| --- |
| Apache  drwxrwxr-x 9 4096 Feb 17 17:37 center  -rwxrwxr-x 1 366 Feb 17 17:37 centerctl.sh  drwxrwxr-x 3 4096 Feb 17 17:37 temp\_supervisor |

安装完成后，可以访问web界面。

* 如果是本机安装，使用[localhost:19321](http://localhost:19321)。
* 如果通过IP访问，使用 <http://192.168.x.x:19321>。19321为 -p指定的 admin\_console\_port 端口。 默认端口：19321。

如果显示执行成功，但无法正常访问，请排查隧道是否打通。

如果需要关闭Web服务和管理Web的Supervisor（比如原来的 Superviser 端口被占用或者遇到其他问题），或者需要修改 **Web** 端口，可以进一步使用 **centerctl.sh** 脚本来进行操作：

|  |
| --- |
| Bash  cd Mirrorship-manager-xxx  ./centerctl.sh -i  # 进入 Supervivisor 交互式界面。  help # 查看有哪些命令。  status # 查看当前服务，stop、shutdown 等其他命令。  # 也可以重新启动 Mirrorship Manager 的安装。  ./centerctl.sh daemon |

**2. 通过 Web 部署 Mirrorship 集群**

1. 访问Web界面，首先需要配置一个 MySQL 数据库，用于存储Mirrorship Manager的管理、查询、报警等信息。

有多个 Mirrorship 集群时，强烈建议为不同集群配置不同的 mysql 账号，以防止配错信息时，导致不可预期的问题）。



填写完成后，单击**测试连接**，连接成功后单击**下一步**。

1. 选择需要部署的节点，以及Agent和Supervisor的安装目录。

* 主机 IP：可以批量配置，多个IP之间使用半角分号（;）分隔。
* Supervisor用于管理进程的启动停止。
* Agent负责采集机器的统计信息。

所有安装都在用户环境，不会影响系统环境。主机IP需填写内网IP，其他参数采用默认配置即可。



**注意和解释**

* 系统中 Supervisor 有两类：一类用于管理 Agent、BE、FE、Broker；一类用于管理 Web和Center service。
* 如果您需要在部署 Web 和 Center service 的机器上同时部署 Agent、FE、BE、Broker，则需要注意Supervisor 端口冲突的情况。如果有冲突，**推荐如下操作**：重新修改前面的bin/install.sh -s ${new\_port} 配置来指定 Mirrorship Manager 所需要的 Superviser 端口，并保持所有管理 FE/BE/Broker 的 Supervisor/Agent 端口是默认的。

1. 单击**下一步，**在弹出的对话框选择**安装新的集群**或**迁移已有集群**。

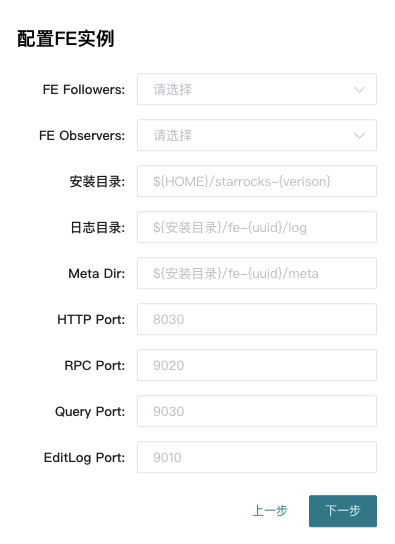
如果您之前已经部署 FE和BE，但没有部署 Mirrorship Manager，则选择**迁移已有集群**。如果您从零开始部署，即新机器上无任何 Mirrorship 程序，则选择**安装新的集群**。



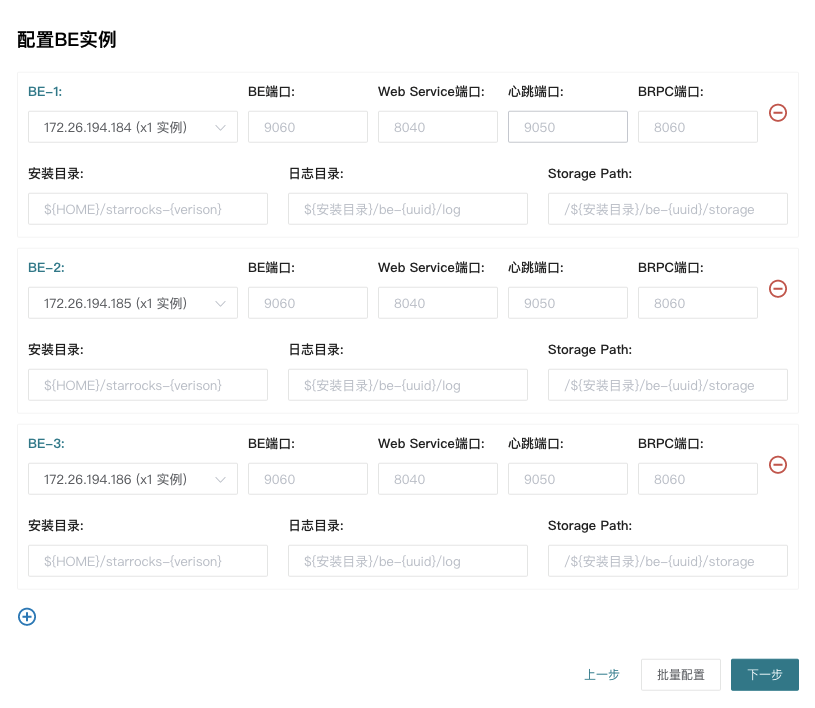
**安装新集群**

1. 安装FE**。**在**配置FE实例**对话框中，配置如下参数。

* **FE Followers：**建议配置1个或者3个。
* **FE Observers：**可以不配置，在请求压力比较大的情况可以酌情增加。
* **Meta Dir：**Mirrorship的元数据目录，和手动安装类似，建议配置独立的Meta和FE log 目录。
* 安装目录、日志目录、端口号请使用默认值。



1. 安装**BE。**



1. 安装**Broker**，建议在所有节点上都安装Broker。



1. 安装**Center service。**

Center service负责从Agent拉取信息、汇总存储、并提供监控报警。**邮件服务**是用来发送通知报警的邮箱，可以不填，后续再配置。



在配置中心服务过程中可能会出现时区UTC错误，请参考[UTC错误](https://starrocks.feishu.cn/docs/doccnKJZC1CBz7UKLYvXApkmhlf#6T6Wny)进行操作。



至此Mirrorship集群安装完成，使用默认的用户root、密码为空就可以登录控制台（可参考 MySQL 的修改密码、创建用户方式进行密码修改和用户添加）。后续进行系统配置优化，请参考Mirrorship企业版文档。

2.1.3版本后的Mirrorship Manager会生成临时root用户及随机密码，在部署的最后一步页面中会提示密码为\*\*\*\*\*\*，需要记录。如果没有及时获取到密码信息，可以在日志中进行查询，日志记录中有打印的临时密码。命令如下：

|  |
| --- |
| SQL  grep -r password manager/center/log/web/\* |

**端口列表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实例名称 | 端口名称 | 默认端口 | 通讯方向 | 说明 |
| BE | be\_port | 9060 | FE --> BE | BE 上 Thrift server 的端口，用于接收来自 FE 的请求 |
| webserver\_port | 8040 | BE <--> BE | BE 上 HTTP server 的端口 |
| heartbeat\_service\_port | 9050 | FE --> BE | BE 上心跳服务端口（thrift），用于接收来自 FE 的心跳 |
| brpc\_port | 8060 | FE<-->BE BE <--> BE | BE 上的 brpc 端口，用于 BE 之间通讯 |
| FE | http\_port | 8030 | FE <--> FE 用户 | FE 上的 HTTP server 端口 |
| rpc\_port | 9020 | BE --> FE FE <--> FE | FE 上的 Thrift server 端口 |
| query\_port | 9030 | 用户 | FE 上的 MySQL Server 端口 |
| edit\_log\_port | 9010 | FE <--> FE | FE 上的 BDBJE之间通信用的端口 |
| Broker | broker\_ipc\_port | 8000 | FE --> Broker BE --> Broker | Broker 上的 Thrift server，用于接收请求 |
| drms (Web) | admin\_console\_port | 19321 | drms 对外 | 对外Web端口 |
| Agent/ MirrorshipManager | supervisor\_http\_port | 19320 | Supervisor内部 | Supervisor管理进程 |
| Agent | agent\_port | 9932 | Agent-->Center | Agent和Center servic通讯，用户上报监控信息 |

**安装目录**

安装成功以后一般会显示如下目录：

* Deploy 目录：安装文件的目录，安装完成后可以备份（如果要装新集群可以复用）或者清除。
* Mirrorship-manager-xxx 目录：管理工具的安装目录。

一般来说有 agent、center 两个目录。Agent是信息采集的代理程序，采集metrics等信息；Center service 负责汇总Agent上报的信息。Center service中有一个web服务，对外提供图形化界面。

* Mirrorship-xxxx 目录：Mirrorship的安装目录，有如下目录：
* be-91bb4d6b-397f-480d-b100-f7f55bd83af4
* fe-0e990400-3910-480d-8529-d422ab83cf7b
* broker-12312313123-12313121

分别是本机的 BE、FE、broker 目录，这三个目录一般只会有一个，除非安装失败。可以在本机部署多个BE但是线上环境并不建议这么做，Mirrorship可以支持单个BE管理多块磁盘。

升级后，每个BE、FE、broker 目录中会有一个**versions**目录保存所有升级后的版本。

**启动 Mirrorship**

安装完成后，Mirrorship 已经完成部署，所有的进程都使用Supervisor进行管理。其中**center**目录下的Supervisor管理Center service和Web；**agent** 目录下的 Supervisor 管理 Agent，BE，FE，Broker。

完成后可以进行使用测试，正式上线前请参考 Mirrorship 企业版文档调整相关参数。

如果机器重启后需要再次启动 Mirrorship 进程，首先进入 Mirrorship Manager 所在的目录，执行**./centerctl.sh**脚本。命令示例：

|  |
| --- |
| Bash  cd mirrorship-manager-20201102/center  ./centerctl.sh daemon |

**centerctl.sh**脚本会启动所有 Agent 及其管理的进程。

如果只有个别 Agent 机器重启了，进入agent目录，执行 **agentctl.sh** 脚本。

|  |
| --- |
| Bash  cd mirrorship-manager-20201102/agent  ./agentctl.sh daemon |

重启web服务也可以自动拉起所有Agent机器的 Supervisor。

|  |
| --- |
| Bash  cd mirrorship-manager-20201102/center  ./centerctl.sh restart web |