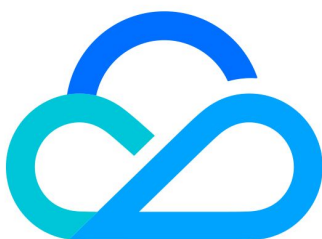


弹性 MapReduce

EMR Serverless HBase 操作指南



腾讯云

【 版权声明 】

©2013–2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

文档目录

EMR Serverless HBase 操作指南

EMR Serverless HBase 产品简介

配额与限制

规划实例

多可用区部署

管理实例

管理权限

创建实例

实例信息

修改实例

表管理

EC 策略设置

销毁实例

监控告警

实例监控

数据表分析

配置告警

开发指南

Serverless HBase 使用说明

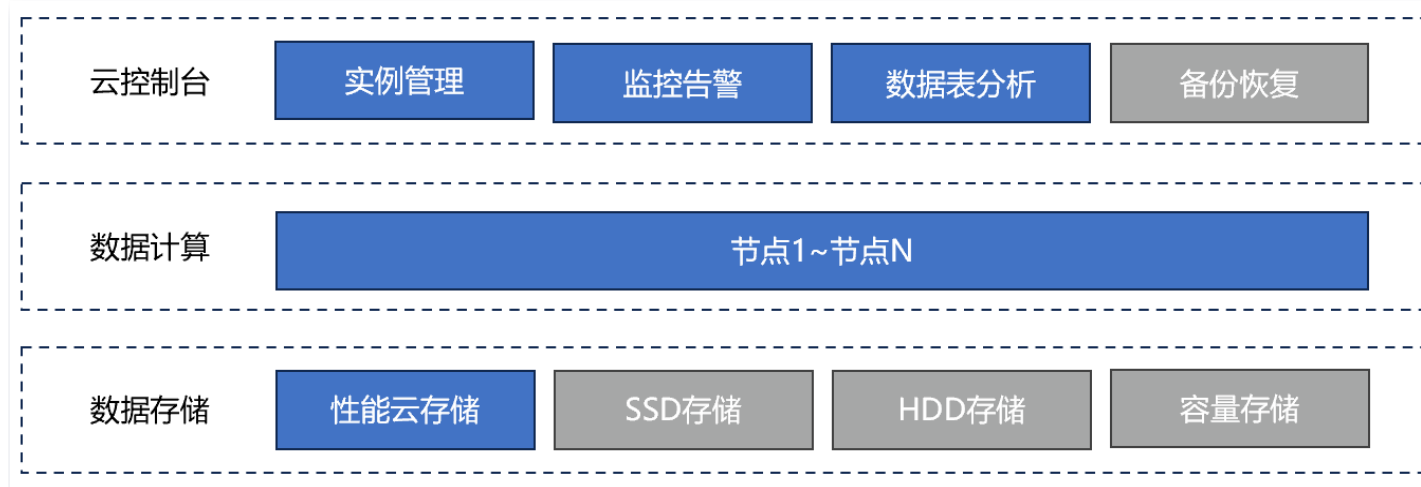
EMR Serverless HBase 操作指南

EMR Serverless HBase 产品简介

最近更新时间：2024-11-28 17:10:52

EMR Serverless HBase（原 EMR Lite HBase）提供全托管的 HBase 服务，完全兼容开源 HBase 协议，您无需关注基础硬件设施的运维，借助 Serverless 版提供的自动故障恢复、备份恢复等功能可高效进行大数据平台运维。您可以通过 EMR Serverless HBase 快速构建海量数据实时查询、实时数据计算和处理、时序数据存储存取等高吞吐、低时延的大数据存取应用。

产品架构



- **云控制台**：提供可视化的 EMR Serverless HBase 实例的管理、备份恢复和监控告警等实例运维功能。
- **数据计算**：实例计算能力由实例下多个节点提供，实例创建后可按需扩展实例数量。
- **数据存储**：数据支持存放在多种不同性能的存储介质上，邀测期间仅支持性能云存储。

产品优势

- **完全兼容**：与开源 HBase 完全兼容，无需修改代码即可无缝迁移。
- **运维简单**：无需运维底层硬件资源，内置参数调优开箱即用。
- **安全可靠**：私有网络逻辑隔离，高可用架构和备份恢复保证数据安全。

配额与限制

最近更新时间：2024-11-28 16:15:32

为确保弹性 MapReduce Serverless HBase 实例的正常运行，在使用之前，请您仔细阅读以下内容。

配额说明

对象	配额
每个实例节点数	默认不超过50个，如需提升请 提交工单 咨询。
每个节点存储量	每个节点存储量为节点核数乘以250为节点搭配存储量上限。如8核16G的节点，搭配存储量最大为8 * 250GiB = 2000GiB。

限制说明

对象	建议限制	硬性限制
每个实例表数量	建议限制在1000以内，非硬性；	-
表中数据大小	每个表的列族数量：10 单个行键：1 KB 列族：10 B 单个列限定符：100 B 表单元格中的单个值：1 MB 一行中的所有值：100 MB	单个行键：64 KB 列限定符列族：64 KB 单个列限定符：64 KB 表单元格中的单个值：10 MB 一行中的所有值：1 GB
批量请求包含的行数	建议限制在5000以内，非硬性；	-
ID 长度限制	表名：1-128 列族：1-16	实例名称：6-36

规划实例

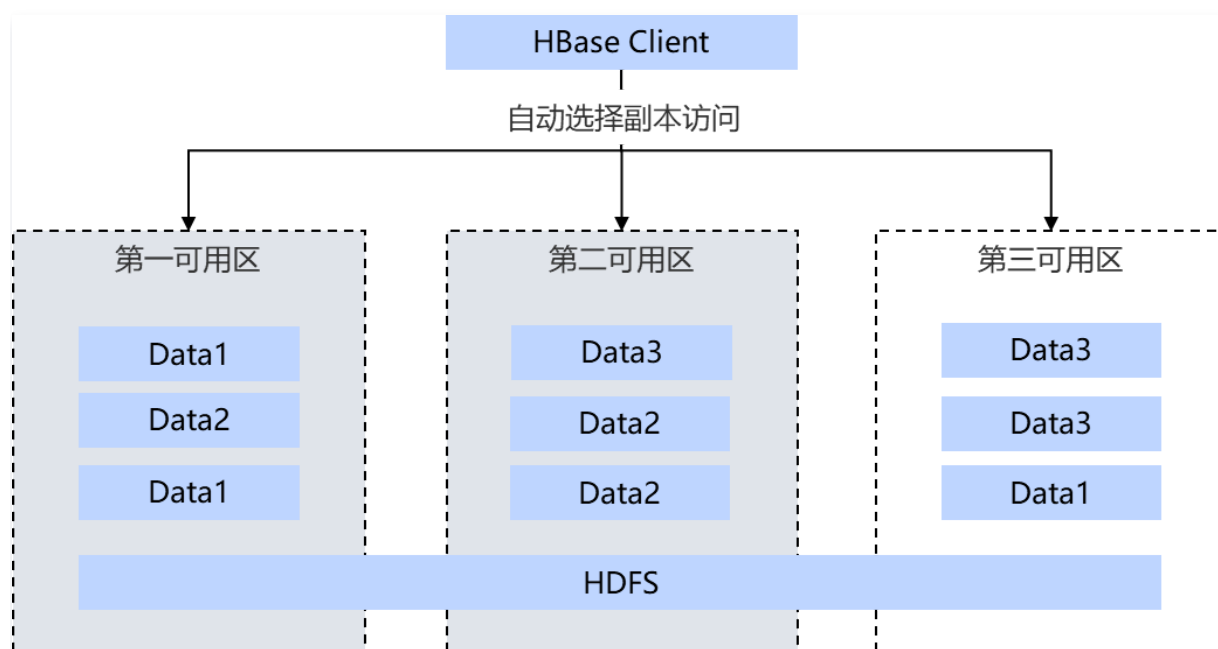
多可用区部署

最近更新时间：2024-11-18 15:28:22

EMR Serverless HBase 部署模式支持基于三个可用区部署，为单可用区故障场景提供服务可用性保障。

多可用区部署介绍

多可用区部署场景，三个可用区均可存储数据，每份数据至少会存储于两个可用区。第一可用区和第二可用区同时故障会导致服务不可用。



创建实例

创建实例时，部署模式选择多可用区部署，所有节点规格一致，第一可用区和第二可用区尽量选择资源充足的可用区。

实例扩容

实例创建完成后，三个可用区节点数量可以分别按需调整，如需数据均衡，需确保每个可用区节点数一致。

管理实例

管理权限

最近更新时间：2024-11-28 16:15:32

本文介绍使用 EMR Serverless HBase 时如何为服务角色授权和子用户授权。

服务角色授权

首次使用 EMR Serverless HBase，需要为 EMR 产品授予服务角色 EMR_QCSRole，使 EMR 产品具备调用产品功能实现必要的关联产品权限。

1. 登录 [EMR Serverless HBase 控制台](#)，首次使用时会提示您对 EMR 进行授权，单击[去授权](#)进入角色管理授权页。
2. 确认服务授权信息后，单击[授权](#)完成服务角色授权。

⚠ 注意：

子用户或协作者账户首次使用时，请确保对账户已授予权限
QcloudCamSubaccountsAuthorizeRoleFullAcces。

子用户授权

一般实际使用产品的账户主要为协作者或子用户，根据授权需求不同，可以通过以下方式子用户或协作者授予不同粒度的操作权限。

授权方案	适用场景	操作步骤
基于预设权限授权	适用于子用户对所有实例配置管理权限。	通过 EMR Serverless HBase 预设策略，快速授予子用户权限 QcloudEMRFullAccess，使子用户具有 EMR 所有资源的管理权限。
基于操作的授权	适用于为所有实例配置部分子用户配置部分操作权限。	新建 CAM 自定义策略，为子用户赋予产品部分操作权限。详见 通过策略生成器创建自定义策略 。
基于标签的普通授权	适用于通过标签为不同归属的实例配置子用户管理权限。	<ul style="list-style-type: none">● 设置实例标签：创建实例为实例添加归属标签。● 设置子用户权限：新建 CAM 自定义策略，为子用户赋予部分实例部分操作权限。详见 通过策略语法创建自定义策略。
基于标签的强制授权	适用于通过标签为不同归属的实例配置子用户管理权限，限制子用户在创建实例	<ul style="list-style-type: none">● 设置子用户权限：详见 创建资源时强制绑定固定标签键值。● 设置实例标签：创建实例为实例添加归属标签。

时只能查询有权限的资源，
为资源配置有权限的标签。

创建实例

最近更新时间：2025-02-26 14:22:52

操作场景

本文为您介绍通过 EMR 控制台创建一个 EMR Serverless HBase 实例的操作步骤和相关配置。

准备工作

1. 已为 EMR 服务授权服务角色 EMR_QCSRole。
2. 用户具备实例创建权限。

操作步骤

登录 [EMR Serverless HBase 控制台](#)，在 EMR Serverless HBase 实例列表页单击创建实例，在购买页完成相关配置，当实例列表中实例状态显示为运行中时，表示实例创建成功。

配置项	配置项说明
地域	实例部署的物理数据中心，每个地域都是一个独立的数据中心，不同地域间的云服务内网不互通。 注意：实例创建后，无法更改地域，请谨慎选择。已支持购买地域详见 计费概述。
实例名称	设置实例名称区分不同用途的实例。
计费模式	支持按量计费和包年包月计费模式。
部署模式	支持单可用区部署和 多可用区部署 。
实例网络	选择对应区域下的专有网络。如果没有，单击新建网络前往新建。
可用区及子网	选择实例部署的可用区及子网。 注意：实例创建后，无法更改可用区，请谨慎选择。
存储类型	选择存储类型，支持标准云存储和性能云存储。
节点规格	选择合适的单节点规格以及节点数量。
存储容量	每个节点搭配的存储容量，默认 100 GiB，可以按 100 GiB 倍数调整容量大小。
节点数量	设置节点数量，每个实例最小3个节点。
标签	腾讯云标签，可以使用标签区分不同的实例用途或按标签进行分账。

实例信息

最近更新时间：2025-02-26 14:22:52

操作场景

查看实例规格、网络、标签等配置信息以及状态信息。

操作步骤

1. 登录 [EMR Serverless HBase 控制台](#)，从实例列表中找到需要销毁的实例，单击实例 ID 进入实例信息页。
2. **基础设置**可以查看实例的计费以及标签等信息。
3. **配置信息**可以查看实例的存储和计算规格信息。
4. **访问方式**可以查看实例的 Zookeeper 访问地址。
5. **健康状态**可以查看实例的运行情况。
6. **高级设置**可以设置 EC 策略。

修改实例

最近更新时间：2024-11-28 16:15:32

操作场景

EMR Serverless HBase 实例创建完成后，可以在不停服状态进行实例节点数量、修改标签以及修改实例显示名称等操作。

操作步骤

调整节点数量

1. 登录 [EMR Serverless HBase 控制台](#)，从实例列表中找到需要修改的实例，单击实例 ID 进入实例信息。
2. 在实例信息-配置信息模块，单击调整节点数量。
3. 打开调整节点数量操作面板后，根据需要增加或减少节点的数量。
4. 单击确定进行变更。

扩容存储

1. 登录 [EMR Serverless HBase 控制台](#)，从实例列表中找到需要修改的实例，单击实例 ID 进入实例信息。
2. 在实例信息-配置信息模块，单击扩容存储。
3. 打开扩容存储操作面板后，根据需要调整目标存储。
4. 单击确定进行变更。

ⓘ 说明：

扩容存储当前为白名单功能，如需开通请 [提交工单](#) 咨询。

修改标签

1. 登录 [EMR Serverless HBase 控制台](#)，从实例列表中找到需要修改的实例，单击实例 ID 进入实例信息。
2. 在实例信息-基础配置模块，在标签信息后单击编辑。
3. 打开调标签编辑操作面板后，根据需要为实例添加新标签或移除旧标签。
4. 单击确定进行变更。

修改实例名

1. 登录 [EMR Serverless HBase 控制台](#)，从实例列表中找到需要修改的实例。
2. 鼠标移动至待修改实例名称上，单击编辑图标。
3. 打开修改实例名称操作面板后，填写新的实例名称。
4. 单击确定进行保存。

表管理

最近更新时间：2025-02-07 14:45:33

操作场景

本文为您介绍通过 EMR 控制台对 HBase 进行命名空间管理和表管理操作。

操作步骤

管理命名空间

⚠ 注意：

- 实例创建完成后，默认会创建 default 命名空间，default 命名空间不支持删除，可通过管理命名空间对自定义命名空间进行新增或删除。
- 删除自定义命名空间需要先删除空间内所有表。

- 登录 [EMR Serverless HBase 控制台](#)，从实例列表中找到需要管理的实例，单击实例 ID 进入实例信息页。
- 在左侧菜单单击表管理进入表管理页面。
- 单击命名空间管理可查看已有命名空间、新增或删除自定义命名空间。

创建表

📌 说明：

控制台仅提供简单的表创建操作，如需设置更复杂表属性，请使用 HBase API 操作。

- 登录 [EMR Serverless HBase 控制台](#)，从实例列表中找到需要管理的实例，单击实例 ID 进入实例信息页。
- 在左侧菜单单击表管理进入表管理页面。
- 单击新建表对表的属性进行配置。

配置项	配置说明
命名空间	选择表的命名空间
表名	设置表的名称
预分区	选择预分区算法，默认不分区
分区数量	选择预分区算法后，可设置分区数量，分区数量设置范围2~100
列族	设置列族名和TTL，最多可设置5个列族 注意：列族创建后不支持在控制台修改。

4. 表配置完成后单击**创建**进行表添加。

修改表

修改表仅支持添加新列族或修改已有列族TTL。

1. 登录 [EMR Serverless HBase 控制台](#)，从实例列表中找到需要管理的实例，单击**实例 ID** 进入**实例信息页**。
2. 在左侧菜单单击**表管理**进入**表管理页面**。
3. 在表列表中找到需要修改的表，在列表操作列单击**修改**。
4. 在表修改弹窗中，添加新列族或修改已有列族TTL，操作完成后单击**保存**。

启用/禁用表

启用状态的表可以进行数据表读写操作。禁用状态的表无法进行读写操作，但可以进行元数据的操作，修改表接口或者数据清理前可以通过禁用表确保表中的数据不会进行修改或访问。

1. 登录 [EMR Serverless HBase 控制台](#)，从实例列表中找到需要管理的实例，单击**实例 ID** 进入**实例信息页**。
2. 在左侧菜单单击**表管理**进入**表管理页面**。
3. 在表列表中找到需要修改的表，根据需要**启用/禁用表**，确认操作风险后单击**确认启用/确认禁用**。

查询表

1. 登录 [EMR Serverless HBase 控制台](#)，从实例列表中找到需要管理的实例，单击**实例 ID** 进入**实例信息页**。
2. 在左侧菜单单击**表管理**进入**表管理页面**。
3. 在表列表中找到需要查询的表，在列表操作列单击**查询表**进入**表查询页面**。
4. 输入查询条件后单击**查询**获取查询结果。

删除表

1. 登录 [EMR Serverless HBase 控制台](#)，从实例列表中找到需要管理的实例，单击**实例 ID** 进入**实例信息页**。
2. 在左侧菜单单击**表管理**进入**表管理页面**。
3. 在表列表中找到需要删除的表，在列表操作列单击**删除**。
4. 确认删除风险后，单击**确定删除**。

⚠ 注意：

仅支持删除禁用状态的表，操作前请先设置表状态为禁用。

EC 策略设置

最近更新时间：2025-02-19 10:26:32

EC（Erasure Coding）策略设置是实例的高级配置，可以通过开启 EC 策略降低数据副本数，进而降低资源成本。

说明：

EC 策略可按需开启或关闭，默认关闭，开启后新增数据按照 RS-6-3-1024k 策略进行存储。

EC 原理介绍

为了保证数据的可靠性，HDFS 的默认数据存储策略是 3 副本，即在写入数据的时候，会占用该数据大小 3 倍的空间。这样就造成了大量的空间浪费。对此，HDFS 引入在 RAID 磁盘阵列中已应用成熟的技术：EC（Erasure Coding，纠删码）。

EC 是一种数据保护技术，通过将原始数据分割成多个片段，并添加一定数量的校验片段，使得在部分数据丢失的情况下仍能恢复原始数据。与传统的 RAID 技术相比，EC 具有更高的存储效率和更低的冗余度。

RS-6-3-1024k 策略

不同的 EC 编解码算法、数据块大小、数据块和校验块个数，可以构成不同的 EC 策略。RS-6-3-1024k 策略相对 3 副本可以节省 50% 的存储空间，每份数据最多允许丢失 3 个数据块，数据写入和恢复时减少 50% 的网络传输量。

RS-6-3-1024k 策略的定义：

1. 使用 RS（Reed Solomon）编解码算法。
2. 有 6 个 DN 用于存储数据块。
3. 有 3 个 DN 存储校验块。
4. 最大可以容忍 3 个块丢失的异常情况。
5. 每个文件块的大小为 1024k（即 1MB）。
6. 如果使用该 EC 策略存储的文件为 100MB，则写入 DataNode 中的总数据量为 $(1+3/6) * 100\text{MB} = 150\text{MB}$ 。其中：
 - 数据块总大小为文件大小 100MB。
 - 校验块总大小为 $3/6 * 100\text{MB} = 50\text{MB}$ 。

操作步骤

1. 登录 [EMR Serverless HBase 控制台](#)，从实例列表中找到需要销毁的实例，单击实例 ID 进入实例信息页。
2. 在高级设置中找到 EC 策略，单机开关打开 EC 策略设置弹框。
3. 在 EC 策略设置弹框中，确认开启或关闭策略的风险后，单机确认或确认关闭进行操作确认。

注意：

仅支持使用的 EC 策略为 RS-6-3-1024k，策略开启需满足实例节点数不低于10个。

销毁实例

最近更新时间：2024-11-28 16:15:32

操作场景

业务变化不需要 EMR Serverless HBase 实例时，可以通过销毁实例释放资源。

⚠ 注意：

销毁实例后实例内的数据会全部清除，确定风险后谨慎操作。

操作步骤

1. 登录 [EMR Serverless HBase 控制台](#)，从实例列表中找到需要销毁的实例，在操作列单击**销毁**按钮。
2. 打开销毁实例确认面板后，确认实例可销毁后单击**确定**开始销毁实例。

监控告警

实例监控

最近更新时间：2024-11-28 16:15:31

操作场景

查看 HBase 实例的运行指标。

操作步骤

1. 登录 [EMR Serverless HBase 控制台](#)，从实例列表中找到需要查看的实例，单击 [监控](#) 进入 [实例监控页](#)。
2. 进入实例监控页后，可以按需查看 HBase 运行指标。

更新信息请查阅 [HBase 监控指标](#)。

数据表分析

最近更新时间：2024-11-28 16:15:31

操作场景

查看 HBase 表和 Region 请求情况趋势，分析数据热点来源。

操作步骤

1. 登录 [EMR Serverless HBase 控制台](#)，从实例列表中找到需要查看的实例，单击实例 ID 进入实例信息页。
2. 单击左侧菜单**诊断分析**，选择**数据表分析**进行查看。

配置告警

最近更新时间：2024-11-28 16:15:31

操作场景

弹性 MapReduce 已接入腾讯云可观测平台，用户可在腾讯云可观测平台控制台配置弹性 MapReduce 节点和服务监控指标的告警策略。

操作步骤

1. 登录 [腾讯云可观测平台](#) 控制台，选择左侧导航栏中的告警管理>告警配置。
2. 在策略管理页面中，单击新建策略。
3. 在弹出的新建告警策略窗口中，参考下表说明配置基本信息、告警规则和新建通知模板。

配置类型	配置项	说明	
基本信息	策略名称	自定义策略名称	
	备注	自定义策略备注	
配置告警规则	监控类型	选择云产品监控	
	策略所属项目	设置所属项目后，您可以在告警策略列表快速筛选该项目下的告警策略	
	策略类型	选择您需要监控的云产品策略类型	
	告警对象	选择实例 ID，则该告警策略绑定所选实例 选择实例组，则该告警策略绑定所选实例分组 选择全部对象，则该告警策略绑定当前账号拥有权限的全部实例	
	触发条件	手动配置（指标告警）	告警触发条件：指标、比较关系、阈值、统计周期和持续周期组成的一个有语义的条件。您可以展开触发条件详情，根据图表中指标变化趋势设置合适的告警阈值
		手动配置（事件告警）	在云产品资源或底层基础设施服务发生异常时，可以创建事件告警及时通知您采取措施
选择模板		选择模板按钮，并在下拉列表选择已配置的模板，具体配置请参阅 配置触发条件模板	
配置告警通知	通知模板	默认绑定系统预设通知模板（接收人为主账号管理员，接收渠道为短信和邮件），每个告警策略最多只能绑定三个通知模	

板。如需了解更多通知模板配置请参考
新建通知模板

4. 填写完后，单击**完成**即可。

开发指南

Serverless HBase 使用说明

最近更新时间：2024-11-28 16:15:32

操作场景

通过 Shell、Java 以及 Golang 访问 EMR Serverless HBase 实例并进行建表和查询等操作。

准备工作

- 已经创建了一个 EMR Serverless HBase 实例且实例状态为运行中。
- 已经创建一台与 EMR Serverless HBase 实例同 VPC 和子网的 CVM 实例（后面简称 CVM 客户机）。
- 在 CVM 客户机上已安装 Java 环境并配置环境变量，安装要求 JDK1.8 以上版本，如未安装，[获取安装并配置 Java 环境](#)。
- 已在 EMR Serverless HBase 实例信息页获取 Serverless HBase 实例的 Zookeeper 地址。

ⓘ 说明：

更多 HBase API 使用请查阅 Apache HBase 官网：
<https://hbase.apache.org/2.4/apidocs/index.html>。

使用 HBase Shell

配置环境

1. 登录 CVM 客户机，然后下载 HBase 客户端，解压并切换到 `emr-serverless-hbase-client` 目录。

```
cd /usr/local
wget https://emr-serverless-hbase-1259353343.cos.ap-guangzhou.myqcloud.com/emr-serverless-hbase-client.tar.gz
tar -zxvf emr-serverless-hbase-client.tar.gz
cd emr-serverless-hbase-client
```

2. 修改 `conf/hbase-site.xml` 中的 `hbase.zookeeper.quorum` 参数配置，`$quorum` 是 Zookeeper 地址。

```
vi conf/hbase-site.xml
<property>
  <name>hbase.zookeeper.quorum</name>
  <value>$quorum</value>
```

```
</property>
```

HBase 基本操作

1. 通过如下命令可以进入 HBase Shell。

```
./bin/hbase shell
```

2. 在 HBase shell 下输入 help 可以查看基本的使用信息和示例的指令。可以使用以下指令建立一个新表。

```
hbase:001:0> create 'test', 'cf'
Created table test
Took 1.5703 seconds
=> Hbase::Table - test
```

3. 表格建立后，可以使用 list 指令来查看您建立的表是否存在。

```
hbase:002:0> list 'test'
TABLE
test
1 row(s)
Took 0.0158 seconds
=> ["test"]
```

4. 使用 put 指令来为您创建的表加入数据。在创建的表中加入了三个值，第一次在“row1”行“cf:a”列插入了一个值“value1”，以此类推。

```
hbase:003:0> put 'test', 'row1', 'cf:a', 'value1'
Took 0.1766 seconds
hbase:004:0> put 'test', 'row2', 'cf:b', 'value2'
Took 0.0160 seconds
hbase:005:0> put 'test', 'row3', 'cf:c', 'value3'
Took 0.0149 seconds
```

5. 使用 scan 指令来遍历整个表。

```
hbase:006:0> scan 'test'
ROW                                COLUMN+CELL
  row1                                column=cf:a, timestamp=2024-07-
16T10:56:28.027, value=value1
```

```
row2                                column=cf:b, timestamp=2024-07-
16T10:56:40.658, value=value2
row3                                column=cf:c, timestamp=2024-07-
16T10:56:51.744, value=value3
3 row(s)
Took 0.0682 seconds
```

6. 使用 get 指令来取得表中指定行的值。

```
hbase:007:0> get 'test', 'row1'
COLUMN                                CELL
cf:a                                timestamp=2024-07-16T10:56:28.027,
value=value1
1 row(s)
Took 0.0361 seconds
```

7. 使用 drop 指令来删除一个表，在删除表之前需要先使用 disable 指令来禁用一个表。

```
hbase:008:0> disable 'test'
Took 0.6919 seconds
hbase:009:0> drop 'test'
Took 0.3451 seconds
```

8. 最后可以使用 exit 指令来关闭 HBase shell。

使用 Java API

确保开发环境下有合适的 JDK 版本，安装配置 Maven 的环境变量，如果您使用 IDE，请在 IDE 中设置 Maven 相关配置。如有需要，使用腾讯云镜像源加速 Maven，在 Maven 配置文件 settings.xml 追加如下 repository mirror。

```
<mirror>
  <id>nexus-tencentyun</id>
  <mirrorOf>*</mirrorOf>
  <name>Nexus tencentyun</name>
  <url>http://mirrors.cloud.tencent.com/nexus/repository/maven-
public/</url>
</mirror>
```

新建一个 Maven 工程

在命令行下进入您想要新建工程的目录，输入如下命令新建一个 Maven 工程。


```
mvn archetype:generate -DgroupId=com.tencent.cloud-DartifactId=test-serverless-hbase -Dversion=1.0 -DarchetypeArtifactId=maven-archetype-quickstart-DarchetypeVersion=1.1
```

创建成功后，在工程目录下就会生成一个名为 `test-serverless-hbase` 的工程文件夹。其中的文件结构如下所示。`pom.xml` 文件主要用于依赖管理和打包配置，`Java` 文件夹下放置您的源代码。

```
test-hbase
├─ pom.xml
└─ src
   └─ main
      └─ java
         └─ com
            └─ tencent
               └─ cloud
                  └─ App.java
└─ test
   └─ java
      └─ com
         └─ tencent
            └─ cloud
               └─ AppTest.java
```

添加配置和样例代码

在 `pom.xml` 文件中添加 Maven 依赖、打包和编译插件。

```
<dependencies>
  <!-- https://mvnrepository.com/artifact/org.apache.hbase/hbase-client -->
  <dependency>
    <groupId>org.apache.hbase</groupId>
    <artifactId>hbase-client</artifactId>
    <version>2.4.5</version>
  </dependency>

  <!-- https://mvnrepository.com/artifact/log4j/log4j -->
  <dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.17</version>
  </dependency>
</dependencies>
```

```
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.10.1</version>
      <configuration>
        <source>8</source>
        <target>8</target>
        <encoding>utf-8</encoding>
      </configuration>
    </plugin>
    <plugin>
      <artifactId>maven-assembly-plugin</artifactId>
      <configuration>
        <descriptorRefs>
          <descriptorRef>jar-with-dependencies</descriptorRef>
        </descriptorRefs>
        <archive>
          <manifest>
            <!-- 此处指定main方法入口的class -->
            <mainClass>com.tencent.cloud.App</mainClass>
          </manifest>
        </archive>
      </configuration>
      <executions>
        <execution>
          <id>make-assembly</id>
          <phase>package</phase>
          <goals>
            <goal>single</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

【可选】如果在之后的调试中遇到样例执行失败需要从标准输出中获取日志定位错误，或者您希望保存操作日志到指定目录 `target/serverless-hbase.log`，您可以在 `main` 文件夹下创建 Java 配置文件目录 `resources`，创建 `log4j.properties` 文件并在 `log4j.properties` 文件中添加 `log4j` 日志信息打印模块的配置信息。

```
log4j.rootLogger=INFO, stdout
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d %p [%c] - %m%n
log4j.appender.logfile=org.apache.log4j.FileAppender
log4j.appender.logfile.File=target/serverless-hbase.log
log4j.appender.logfile.layout=org.apache.log4j.PatternLayout
log4j.appender.logfile.layout.ConversionPattern=%d %p [%c] - %m%n
```

App.java 样例代码如下。

```
package com.tencent.cloud;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.*;
import org.apache.hadoop.hbase.client.*;
import org.apache.hadoop.hbase.util.Bytes;

import java.io.IOException;

/**
 * 通过 HBase Java API 访问 Serverless HBase 实例
 */
public class App {
    public static void main(String[] args) throws IOException {
        // $quorum 在控制台实例信息页面通过访问方式模块 zookeeper 访问地址获取。
        // 示例: conf.set("hbase.zookeeper.quorum",
        "10.0.0.8,10.0.0.11,10.0.0.5");
        Configuration conf = HBaseConfiguration.create();
        conf.set("hbase.zookeeper.quorum", "$quorum");

        // 建立 EMR Lite HBase 客户端和数据之间的连接
        Connection connection =
        ConnectionFactory.createConnection(conf);
        Admin admin = connection.getAdmin();

        // 连接建立后,可以使用HBase Java API 访问 Serverless HBase 实例。
        TableDescriptorBuilder tableBuilder =
        TableDescriptorBuilder.newBuilder(TableName.valueOf("test1"));
        ColumnFamilyDescriptor cf =
        ColumnFamilyDescriptorBuilder.of("cf");
        tableBuilder.setColumnFamily(cf);
    }
}
```

```
TableDescriptor table = tableBuilder.build();

System.out.println("Creating Table.");
if (admin.tableExists(table.getTable_name())) {
    admin.disableTable(table.getTable_name());
    admin.deleteTable(table.getTable_name());
}
admin.createTable(table);

System.out.println("Inserting Data.");
Table table1 = connection.getTable(TableDescriptor.valueOf("test1"));
Put put1 = new Put(Bytes.toBytes("row1"));
put1.addColumn(Bytes.toBytes("cf"), Bytes.toBytes("a"),
    Bytes.toBytes("value1"));
```

编译代码并打包上传

使用本地命令行进入工程目录，执行以下指令对工程进行编译打包。

```
mvn package
```

日志打印 build success 表示操作成功，在工程目录下的 target 文件夹中能够看到打包好的 jar 包，您需要将以 with-dependencies 为尾缀的 jar 包通过 CVM 实例控制台中的上传文件功能上传到 CVM 客户机中。若 CVM 客户机可以通过公网 IP 访问，也可以在本地命令行模式下运行以下命令上传文件。

```
scp $localfile root@公网IP地址:$remotefolder
```

其中，\$localfile 是您的本地文件的路径加名称，root 为 CVM 服务器用户名，公网 IP 可以在 CVM 客户机控制台查看。\$remotefolder 是您想存放文件的 CVM 服务器路径。上传完成后，在 CVM 客户机命令行中即可查看对应文件夹下是否有相应文件。

执行样例并查看结果

登录 CVM 客户机，切换到对应文件夹下，使用如下命令执行样例。

```
java -jar $package.jar
```

运行代码，在控制台输出“Done”后，说明所有的操作已完成。您可以切换到 HBase shell 中通过 list 命令来查看使用 API 创建的 HBase 表是否成功，若成功可通过 scan 命令来查看表的具体内容。

```
hbase:001:0> list 'test1'
TABLE
```

```
test1
1 row(s)
Took 0.4315 seconds
=> ["test1"]
hbase:002:0> scan 'test1'
ROW                                COLUMN+CELL
  row1                                column=cf:a,
timestamp=2024-07-16T16:20:38.685, value=value1
  row2                                column=cf:b,
timestamp=2024-07-16T16:20:38.690, value=value2
  row3                                column=cf:c,
timestamp=2024-07-16T16:20:38.695, value=value3
3 row(s)
Took 0.1231 seconds
hbase:003:0> exit
```

使用 Golang API

添加环境和样例代码

在 CVM 客户机上 [安装 Golang](#) 并配置环境变量。以 go1.17.13 版本为例。

```
cd /usr/local
wget https://go.dev/dl/go1.17.13.linux-amd64.tar.gz
tar -zxvf go1.17.13.linux-amd64.tar.gz
vi /etc/profile
```

在/etc/profile中追加

```
export GO_HOME=/usr/local/go
export PATH=${GO_HOME}/bin:$PATH
```

后保存返回命令行

```
source /etc/profile
```

确保开发环境下有一致的 Golang 版本，进入工程目录，如果工程目录下没有 go.mod 文件，执行以下命令。

```
go mod init test-serverless-hbase
```

根据 Golang 版本下载 gohbase 依赖分支([查看 gohbase 项目地址](#))。

Golang Version	commit_id
----------------	-----------

1.22.x	731f0bdb6be56dfe0b5a5a79582161bc7fbed32b
1.18.x~1.21.x	1fee39f343954ca7501a6b8f25abd9f86eaf618b
1.13.x~1.17.x	05795eede1cb2442e6cc0313db93a4b544b49148

更多 Golang 版本对应的 Commit 可以在 [Commits](#) · [tsuna/gohbase](#) · [GitHub](#) 中查看。

```
go env -w GOPROXY=https://goproxy.cn
go get github.com/tsuna/gohbase@commit_id
# 以1.17.13版本为例: go get
github.com/tsuna/gohbase@05795eede1cb2442e6cc0313db93a4b544b49148
```

在项目目录下创建样例代码文件 demo.go，此时工程目录结构如下。

```
test-serverless-hbase/
├─ demo.go
├─ go.mod
└─ go.sum
```

demo.go 样例代码如下。

```
package main

import (
    "context"
    "fmt"
    "github.com/tsuna/gohbase"
    "github.com/tsuna/gohbase/hrpc"
    "log"
)

func main() {
    // $quorum 在控制台实例信息页面通过访问方式模块 zookeeper 访问地址获取。
    // 示例: quorum := "10.0.0.8,10.0.0.11,10.0.0.5"
    quorum := $quorum
    // 创建表格
    tableName := "test2"
    var cfs = map[string]map[string]string{
        "cf1": {
            "MIN_VERSIONS": "1",
        },
    },
```

```
    }
    admin := gohbase.NewAdminClient(quorum)
    crt := hrpc.NewCreateTable(context.Background(),
[]byte(tableName), cfs)
    err := admin.CreateTable(crt)
    if err != nil {
        log.Fatal(err)
    }

    client := gohbase.NewClient(quorum)

    // 写入数据
    putRequest, err := hrpc.NewPutStr(context.Background(),
tableName, "my_row_key", map[string]map[string][]byte{
        "cf1": map[string][]byte{
            "col1": []byte("value1"),
            "col2": []byte("value2"),
        },
    })
    if err != nil {
        log.Fatal(err)
    }
}
```

编译代码并打包上传

如果您当前的开发环境是 Linux 环境，请直接在命令行执行如下命令。

```
go build demo.go
```

如果是非 Linux 环境，请执行如下命令，打包成 Linux 文件后，切换原来的环境，确保其他文件正常编译。

```
go env -w GOOS=linux
go build demo.go
```

在工程目录下能够看到编译得到的二进制文件 demo，将该文件通过 CVM 实例控制台中的上传文件功能上传到 CVM 客户机中。

若 CVM 客户机可以通过公网 IP 访问，也可以在本地命令行模式下运行以下命令上传文件。

```
scp $localfile root@公网IP地址:$remotefolder
```

其中，\$localfile 是您的本地文件的路径加名称，root 为 CVM 服务器用户名，公网 IP 可以在 CVM 客户机控制台查看。\$remotefolder 是您想存放文件的 CVM 服务器路径。上传完成后，在 CVM 客户机命令行中即可查看对应文件夹下是否有相应文件。

执行样例并查看结果

切换到该文件所在的目标目录下，赋予该二进制文件可执行权限并执行。

```
chmod +x demo
./demo
```

运行代码后，切换到 HBase shell 中通过 list 命令来查看使用 API 创建的 HBase 表是否成功，若成功可通过 scan 命令来查看表的具体内容。

```
hbase:001:0> list 'test2'
TABLE
test2
1 row(s)
Took 0.3814 seconds
=> ["test2"]
hbase:002:0> scan 'test2'
ROW                                COLUMN+CELL
  my_row_key                        column=cf1:col1, timestamp=2024-07-
17T21:16:49.302, value=value1
  my_row_key                        column=cf1:col2, timestamp=2024-07-
17T21:16:49.302, value=value2
1 row(s)
Took 0.1268 seconds
hbase:003:0> exit
```