ALIBABA CLOUD

阿里云

阿里云Elasticsearch

最佳实践

文档版本：20220704

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
`Courier字体`	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.最佳实践概览

阿里云Elasticsearch为您提供各种业务场景下的最佳实践，您可以根据本文查看适合您业务的最佳实践文档。

场景	文档链接

Elasticsearch迁移	• 迁移方案选取指南
	• 阿里云Elasticsearch间数据迁移
	  ◦ 通过reindex迁移数据
	  ◦ 基于reindex实现低版本多type数据迁移
	• 自建Elasticsearch数据迁移
	  ◦ 通过OSS将自建Elasticsearch数据迁移至阿里云
	  ◦ 通过阿里云Logstash将自建Elasticsearch数据迁移至阿里云
	  ◦ 通过reindex将自建Elasticsearch数据迁移至阿里云
	  ◦ 通过实例私网打通将自建Elasticsearch数据迁移至阿里云
	• 第三方Elasticsearch数据迁移
	  ◦ 腾讯云Elasticsearch数据迁移至阿里云
	  ◦ 从AWS迁移Elasticsearch索引至阿里云

数据库同步	• RDS MySQL同步
	  ◦ 同步方案选取指南
	  ◦ 通过Logstash将RDS MySQL数据同步至Elasticsearch
	  ◦ 通过DataWorks将MySQL数据同步至Elasticsearch
	  ◦ 通过DTS将MySQL数据实时同步到阿里云Elasticsearch
	  ◦ 通过Canal将MySQL数据同步到阿里云Elasticsearch
	• PolarDB-X（DRDS）同步
	  ◦ PolarDB-X（DRDS）同步方案选取指南
	  ◦ 通过Logstash将PolarDB-X（DRDS）数据同步至Elasticsearch
	  ◦ 通过DataWorks将PolarDB-X（DRDS）数据离线同步至Elasticsearch
	• 通过DTS将PolarDB MySQL数据同步至Elasticsearch
	• 通过Monstache实时同步MongoDB数据至Elasticsearch

大数据云产品同步	• 通过DataWorks将MaxCompute数据同步至Elasticsearch • 通过阿里云Logstash将MaxCompute数据同步至Elasticsearch • 通过实时计算处理数据并同步到Elasticsearch • 通过DataWorks将Hadoop数据同步至Elasticsearch
存储产品迁移	• 从Solr集群迁移文档至阿里云Elasticsearch • 基于Logstash迁移OSS数据

场景	文档链接
ES-Hadoop使用	• 通过ES-Hadoop实现Hive读写阿里云Elasticsearch数据 • 通过ES-Hadoop将HDFS中的数据写入Elasticsearch • 通过ES-Hadoop实现Spark读写阿里云Elasticsearch数据
日志采集与分析	• 日志同步分析概述 • 通过自建Filebeat收集MySQL日志 • 使用阿里云Elasticsearch监控RabbitMQ • 使用Filebeat+Kafka+Logstash+Elasticsearch构建日志分析系统 • 查询分析RocketMQ客户端日志 • 通过Elasticsearch和rsbeat实时分析Redis slowlog
服务器数据采集	• 服务器数据采集方案概述 • 阿里云Elasticsearch数据采集解决方案 • 通过自建Metricbeat收集系统指标信息 • 使用SkyWalking和Elasticsearch实现全链路监控 • 通过Uptime实时监控阿里云Elasticsearch服务

集群管理	• 集群管理概述
	• 冷热分离与生命周期管理
	  ◦ 通过索引生命周期管理Heartbeat数据
	  ◦ 通过索引生命周期管理实现冷热数据分离
	• X-Pack高级特性应用
	  ◦ 使用跨集群复制功能迁移数据
	  ◦ X-Pack集成LDAP认证最佳实践
	  ◦ 通过Elasticsearch X-Pack角色管理实现用户权限管控
	  ◦ 配置Active Directory身份认证
	• 日志全观测应用
	  ◦ 通过Elastic实现Kubernetes容器全观测
	  ◦ 基于Indexing Service实现数据流管理
	  ◦ 通过OpenStore实现海量数据存储
	• 数据管理与可视化
	  ◦ 基于Terraform管理阿里云Elasticsearch
	  ◦ Curator操作指南
	  ◦ 通过RollUp实现流量汇总最佳实践
	  ◦ 使用DataV大屏展示阿里云Elasticsearch数据
	  ◦ 通过Cerebro访问阿里云Elasticsearch
	• 集群报警通知
	  ◦ 配置钉钉机器人接收X-Pack Watcher报警
	  ◦ 配置企业微信机器人接收X-Pack Watcher报警

2.Elasticsearch迁移

2.1. 迁移方案选取指南

您可以通过Logstash、reindex和OSS等多种方式完成阿里云Elasticsearch间数据迁移、自建Elasticsearch数据迁移至阿里云和第三方Elasticsearch迁移至阿里云。本文介绍各迁移场景对应的迁移方案、适用场景和使用限制，帮助您根据业务选择合适的方案进行迁移。

> **注意**
>
> - 2020年10月，阿里云Elasticsearch对网络架构进行了调整。2020年10月之前为旧网络架构，2020年10月及之后为新网络架构。新网络架构下的实例不支持与旧网络架构下的实例进行跨集群reindex、跨集群搜索、跨集群复制等实例互通操作。如果需要进行互通，需要确保实例创建在同一网络架构下。对于华北3（张家口）和海外地域，由于网络架构调整时间不确定，因此需要提交工单，联系阿里云Elasticsearch技术支持，校验网络是否可以互通。
> - 建议不要迁移以 `.` 开头的系统索引，例如.monitoring、.kibana、.security等，否则可能导致Kibana出现故障。

迁移场景：阿里云Elasticsearch实例间数据迁移

迁移方案	适用场景	使用限制	相关文档

OSS快照	• 源端数据量较大（GB、TB、PB级别）的场景。
	• 同账号下，同地域或跨地域快照数据迁移的场景。
	注意 对于同账号跨地域的数据迁移场景，可以通过快照备份与恢复命令实现。
		• OSS快照方式不支持迁移增量数据，建议在迁移前关闭源端待迁移索引的写入或更新。
		• 跨集群OSS仓库设置功能，对源端和目标端实例有以下限制：
		  ◦ 相同地域。
		  ◦ 归属于相同账号。
		  ◦ 源端实例的版本低于或等于目标端实例的版本。
			• 设置跨集群OSS仓库（迁移自动快照备份的数据）
			• 手动备份与恢复（迁移手动快照备份的数据）

迁移方案	适用场景	使用限制	相关文档
Logstash	• 迁移全量或同步增量数据，对实时性要求不高的场景。 **注意** ◦ 如果需要同步增量数据，需确保源端数据的ID和目标端ID一致，并且需配置schedule定时任务。 ◦ 对于跨账号、跨地域的数据迁移场景，由于Elasticsearch和Logstash不在同一专有网络下，需要配置Logstash的NAT公网数据传输，具体实现方案可参见腾讯云Elasticsearch数据迁移至阿里云。 • 仅对查询结果进行迁移的场景。 • 需对待迁移数据进行过滤的场景。 • 版本跨度较大的数据迁移场景，例如5.x迁移到6.x或7.x。版本兼容性说明请参见产品兼容性。	• 源Elasticsearch、Logstash和目标Elasticsearch实例在同一专有网络。如果不在同一专有网络，需要通过配置NAT网关实现与公网的连通，详细信息请参见配置NAT公网数据传输。 • 源Elasticsearch、Logstash和目标Elasticsearch实例版本需满足兼容性要求，详细信息请参见产品兼容性。	• 阿里云Elasticsearch间同步数据 • 基于reindex实现低版本多type数据迁移
reindex	• 源端数据量较小，且对迁移速度要求不高的场景。 • 仅对查询结果（使用查询语句在Kibana中查询出来的结果）进行迁移的场景。	两个Elasticsearch集群在同一网络架构下，详细信息请参见注意事项。	通过reindex迁移数据

迁移场景：自建Elasticsearch迁移至阿里云

迁移方案	适用场景	使用限制	相关文档

迁移方案	适用场景	使用限制	相关文档
OSS快照	• 源端数据量较大（GB、TB、PB级别）的场景。 • 同账号下，同地域或跨地域快照数据迁移的场景。	• 需要获取与源Elasticsearch版本一致的elasticsearch-repository-oss插件。 • OSS快照方式不支持迁移增量数据，建议在迁移前关闭源端待迁移索引的写入或更新。	通过OSS将自建Elasticsearch数据迁移至阿里云
Logstash	• 迁移数据时，对实时性要求不高的场景。 **注意** 如果需要同步增量数据，需确保源端数据的ID和目标端ID一致，并且需配置schedule定时任务。 • 仅对查询结果进行迁移的场景。 • 需对待迁移数据进行过滤的场景。 • 版本跨度较大的数据迁移场景，例如5.x迁移到6.x或7.x。版本兼容性说明请参见产品兼容性。	• 源Elasticsearch、Logstash和目标Elasticsearch实例在同一专有网络。如果不在同一专有网络，需要通过配置NAT网关实现与公网的连通，详细信息请参见配置NAT公网数据传输。 • 源Elasticsearch、Logstash和目标Elasticsearch实例版本需满足兼容性要求，详细信息请参见产品兼容性。	• 通过阿里云Logstash将自建Elasticsearch数据迁移至阿里云 • 基于reindex实现低版本多type数据迁移
reindex	• 源端数据量较小，且对迁移速度要求不高的场景。 • 仅对查询结果（使用查询语句在Kibana中查询出来的结果）进行迁移的场景。	两个Elasticsearch集群在同一网络架构下，详细信息请参见注意事项。	通过reindex将自建Elasticsearch数据迁移至阿里云

迁移场景：第三方Elasticsearch迁移至阿里云

迁移方案	适用场景	使用限制	相关文档
Logstash	• 迁移全量数据的场景。 • 迁移满足某些查询需求的场景。 • 跨账号、跨地域的阿里云Elasticsearch间数据迁移的场景。	• Logstash需要配置NAT网关实现与公网连通，详细信息请参见配置NAT公网数据传输。 • 源Elasticsearch、Logstash和目标Elasticsearch实例版本需满足兼容性要求，详细信息请参见产品兼容性。	腾讯云Elasticsearch数据迁移至阿里云

迁移方案	适用场景	使用限制	相关文档
OSS快照	源端数据量较大（GB、TB、PB级别）的场景。	OSS快照方式不支持迁移增量数据，建议在迁移前关闭源端待迁移索引的写入或更新。	AWS Elasticsearch数据迁移至阿里云

2.2. 阿里云Elasticsearch间数据迁移

2.2.1. 通过reindex迁移数据

当您需要将远程Elasticsearch集群中的数据迁移到本地Elasticsearch集群中时，可以通过reindex API重建索引来实现。本文介绍具体的实现方法。

背景信息

reindex的应用场景如下：

- Elasticsearch集群间迁移数据。
- 索引分片分配不合理，例如数据量太大分片数太少，可通过reindex重建索引。
- 索引中存在大量数据的情况下，需要修改索引mapping，可通过reindex复制索引数据。

> **说明** 在Elasticsearch中，定义了索引mapping且导入数据后，将不能再修改索引mapping。

前提条件

- 准备两个阿里云Elasticsearch集群，一个为本地集群，一个为远程集群。

  具体操作，请参见创建阿里云Elasticsearch实例。本地集群和远程集群需要在同一专有网络和虚拟交换机下。本文使用6.7.0版本的实例作为本地集群，6.3.2版本的实例作为远程集群。

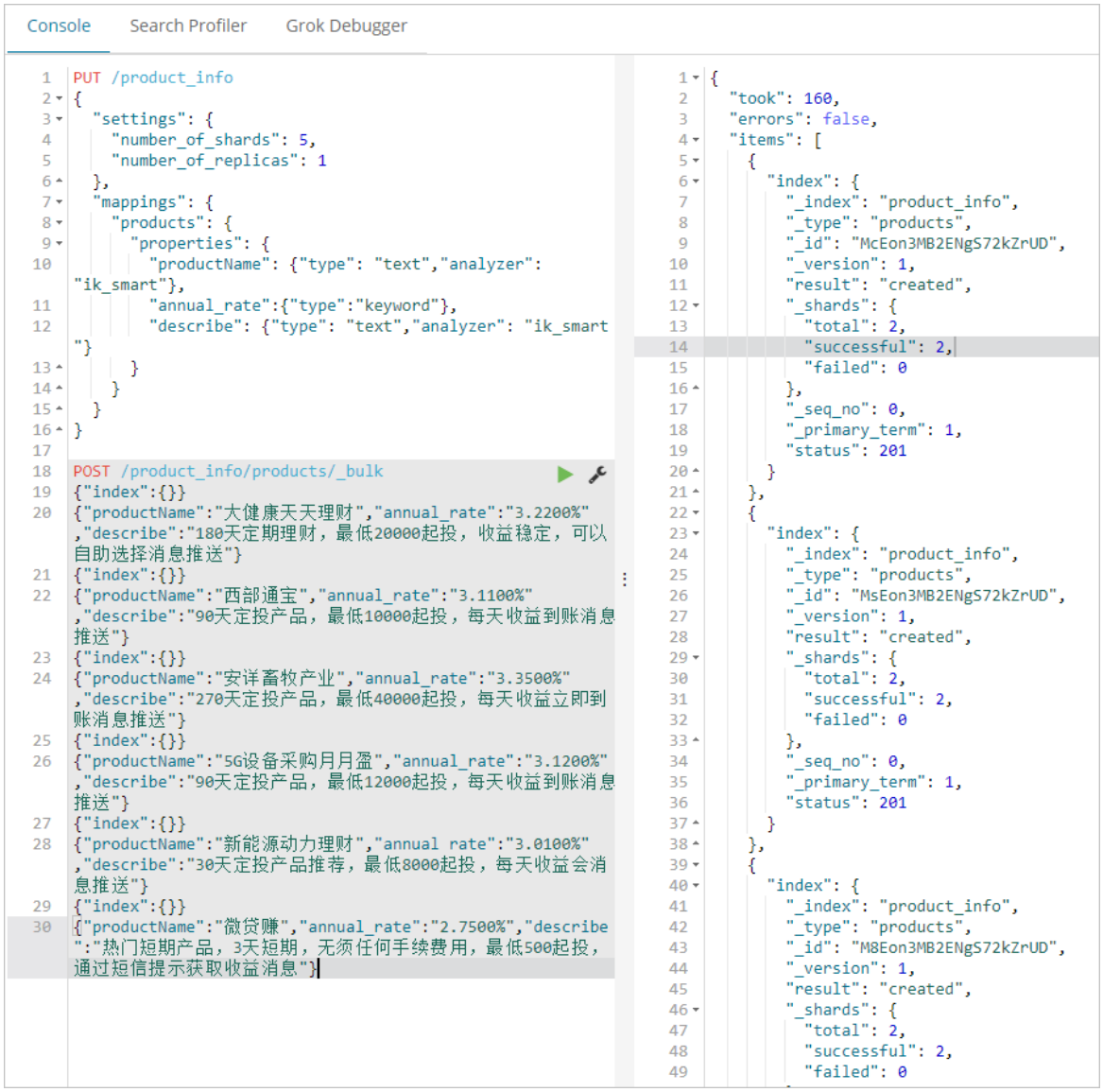
- 准备测试数据。
  - 本地集群

    在本地集群中创建目标索引。

```
PUT dest
{
  "settings": {
    "number_of_shards": 5,
    "number_of_replicas": 1
  }
}
```

- 远程集群

  在远程集群中准备待迁移的数据。本文使用快速入门章节中的数据测试，详细信息请参见快速入门。



> **注意** 如果您使用的是7.0及以上版本的集群，需要将索引类型修改为_doc。

使用限制

自2020年10月起，由于网络架构的调整，导致部分通过reindex方式跨集群迁移数据的场景受到了限制。您可以参见下表，查看您的业务场景是否支持reindex功能，并参考解决方案进行处理。

使用场景	网络状态（2020年10月之前属于旧网络架构，2020年10月及之后属于新网络架构）	是否支持reindex功能	解决方案
	两个Elasticsearch集群均创建于旧网络架构下。	是	请参见通过reindex迁移数据。

使用场景	网络状态（2020年10月之前属于旧网络架构，2020年10月及之后属于新网络架构）	是否支持reindex功能	解决方案
通过reindex方式迁移阿里云Elasticsearch集群间的数据	两个Elasticsearch集群均创建于新网络架构下。	否	不支持reindex方式，建议使用OSS或Logstash方式。具体操作请参见通过OSS将自建Elasticsearch数据迁移至阿里云和通过阿里云Logstash将自建Elasticsearch数据迁移至阿里云。
	两个Elasticsearch集群分别创建于旧网络架构下和新网络架构下。	否	
将ECS上自建的Elasticsearch集群中的数据迁移至阿里云Elasticsearch集群中	阿里云Elasticsearch集群创建于旧网络架构下。	是	请参见通过reindex将自建Elasticsearch数据迁移至阿里云。
	阿里云Elasticsearch集群创建于新网络架构下。	是	借助PrivateLink，打通ECS上自建Elasticsearch集群所处的网络与阿里云服务账号的网络，再使用终端节点域名进行reindex。详情请参见通过实例私网打通将自建Elasticsearch数据迁移至阿里云。 **说明** PrivateLink仅支持部分地域私网连接，详情请参见支持私网连接的地域和可用区。如果您的集群可用区不满足此条件，则不支持reindex功能。

操作步骤

1. 登录阿里云Elasticsearch控制台。
2. 在左侧导航栏，单击**Elasticsearch实例**。
3. 进入目标实例。
   i. 在顶部菜单栏处，选择资源组和地域。
   ii. 在左侧导航栏，单击**Elasticsearch实例**，然后在**Elasticsearch实例**中单击目标实例ID。
4. 在本地集群中，配置reindex白名单。
   i. 在左侧导航栏，选择**配置与管理 > ES集群配置**。
   ii. 在**YML文件配置**右侧，单击**修改配置**。

iii. 在**其他Configure配置**文本框中，输入reindex白名单。

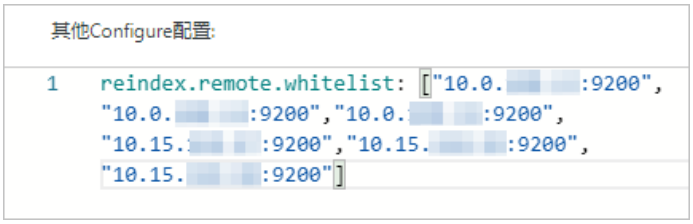
reindex白名单的配置格式与实例的可用区数量有关，具体如下：

- 单可用区实例：白名单的格式为<阿里云Elasticsearch实例的域名>:9200。

例如：

```
reindex.remote.whitelist: ["es-cn-09k1rgid9000g****.elasticsearch.aliyuncs.com:920
0"]
```

- 多可用区实例：白名单需要配置为实例中所有数据节点的IP地址与端口的组合。



例如：

```
reindex.remote.whitelist: ["10.0.xx.xx:9200","10.0.xx.xx:9200","10.0.xx.xx:9200","
10.15.xx.xx:9200","10.15.xx.xx:9200","10.15.xx.xx:9200"]
```

> **说明** 您可以在实例**基本信息**页面的**节点可视化**页签中，获取实例中所有数据节点的IP地址。详细信息，请参见查看节点的基本信息。

更多关于reindex白名单的配置说明，请参见配置reindex白名单。

iv. 选中**该操作会重启实例，请确认后操作**，单击**确定**。

确定后，Elasticsearch实例会重启。重启过程中，可在任务列表查看进度。重启成功后，即可完成配置。

5. 在本地集群中，调用reindex API重建索引。

i. 登录目标阿里云Elasticsearch实例的Kibana控制台，根据页面提示进入Kibana主页。

登录Kibana控制台的具体操作，请参见登录Kibana控制台。

> **说明** 本文以阿里云Elasticsearch 6.7.0版本为例，其他版本操作可能略有差别，请以实际界面为准。

ii. 在左侧导航栏，单击**Dev Tools**。

iii. 在Console中，执行如下命令，重建索引。

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "http://es-cn-09k1rgid9000g****.elasticsearch.aliyuncs.com:9200",
      "username": "elastic",
      "password": "your_password"
    },
    "index": "product_info",
    "query": {
      "match": {
        "productName": "理财"
      }
    }
  },
  "dest": {
    "index": "dest"
  }
}
```

类别	参数	说明
source	host	远程集群的访问地址，必须包含支持协议、域名和端口信息，例如https://otherhost:9200。host配置格式与实例的可用区数量有关，具体如下： ■ 单可用区实例：http://<实例的域名>:9200。 **说明** 实例的域名可在基本信息页面获取。详细信息，请参见查看实例的基本信息。 ■ 多可用区实例：http://<实例中任意数据节点的IP地址>:9200。
	username	可选参数，如果您所请求的远程Elasticsearch服务需要使用Basic Authentication，请在请求中一并提供此参数信息。阿里云Elasticsearch实例的默认用户名为elastic。 **注意** ■ 为确保安全性，通过Basic Authentication鉴权时建议使用HTTPS协议，否则密码信息将以文本形式进行传输。 ■ 对于阿里云Elasticsearch实例，需要开启HTTPS协议后，才可在host中使用HTTPS协议。开启HTTPS协议的具体操作请参见使用HTTPS协议。

类别	参数	说明
	password	用户对应的密码。阿里云Elasticsearch实例的elastic用户的密码在创建实例时设定，如果忘记可进行重置。重置密码的注意事项及具体操作，请参见重置实例访问密码。
	index	远程集群中的源索引。
	query	通过查询语法，指定待迁移的数据。详细信息，请参见Reindex API。
dest	index	本地集群中的目标索引。

> **说明** 从远程集群重建索引数据，不支持手动切片或自动切片。详细信息，请参见手动切片和自动切片。

执行成功后，预期结果如下。

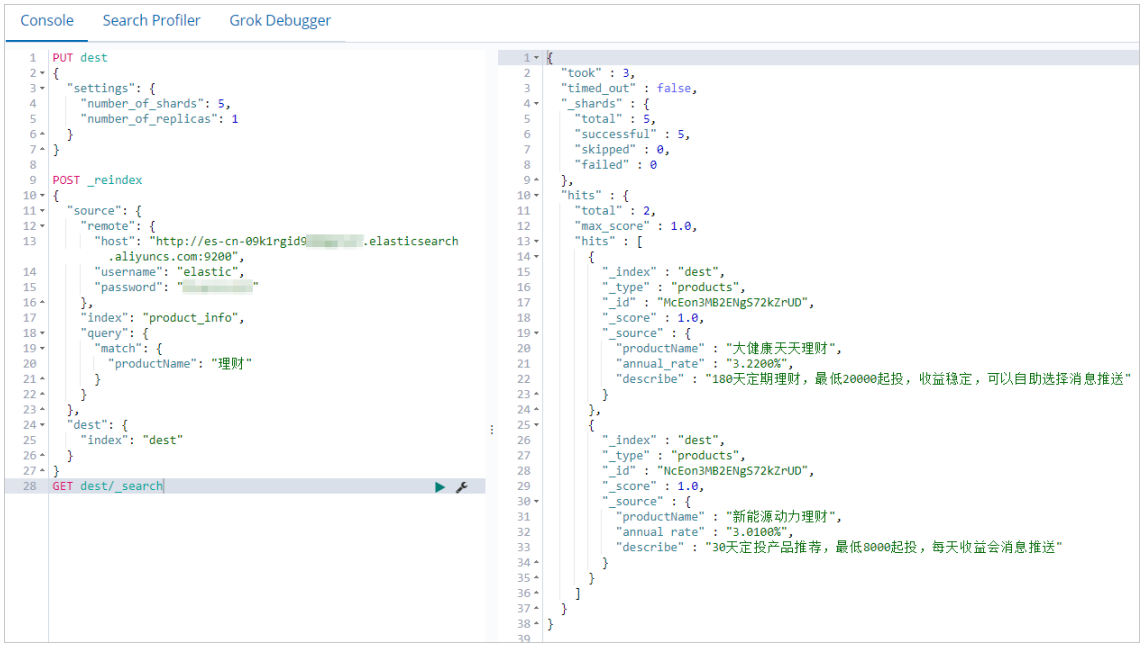
```
{
  "took" : 51,
  "timed_out" : false,
  "total" : 2,
  "updated" : 2,
  "created" : 0,
  "deleted" : 0,
  "batches" : 1,
  "version_conflicts" : 0,
  "noops" : 0,
  "retries" : {
    "bulk" : 0,
    "search" : 0
  },
  "throttled_millis" : 0,
  "requests_per_second" : -1.0,
  "throttled_until_millis" : 0,
  "failures" : [ ]
}
```

6. 查看迁移成功的数据。

```
GET dest/_search
```

预期结果如下：

- 单可用区实例



- 多可用区实例

总结

通过reindex API迁移数据时，单可用区的阿里云Elasticsearch实例和多可用区实例的配置方法大致相同，不同之处在于以下两点。

可用区类型	reindex白名单配置	host参数配置
单可用区	阿里云Elasticsearch的域名：9200	https://阿里云Elasticsearch实例的域名：9200
多可用区	实例中所有数据节点的IP地址与端口的组合	https://阿里云Elasticsearch实例中任意数据节点的IP地址：9200

更多信息

在调用reindex API重建索引时，您还可以进行批量设置和超时时间设置：

- 批量设置

  远程Elasticsearch集群使用堆缓存索引数据，默认最大值为100 MB。如果远程索引中包含大文档，请将批量数值设置为较小值。

  以下示例中，通过size设置批量数值为10。

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "http://otherhost:9200"
    },
    "index": "source",
    "size": 10,
    "query": {
      "match": {
        "test": "data"
      }
    }
  },
  "dest": {
    "index": "dest"
  }
}
```

- 超时时间设置

  您可以使用socket_timeout设置socket读取超时时间，默认为30s；使用connect_timeout设置连接超时时间，默认为1s。

  以下示例中，设置socket读取超时时间为1分钟，连接超时时间为10秒。

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "http://otherhost:9200",
      "socket_timeout": "1m",
      "connect_timeout": "10s"
    },
    "index": "source",
    "query": {
      "match": {
        "test": "data"
      }
    }
  },
  "dest": {
    "index": "dest"
  }
}
```

2.2.2. 基于reindex实现低版本多type数据迁移

本文介绍基于reindex将阿里云Elasticsearch 5.x实例中的多type数据，迁移到高版本Elasticsearch 6.x实例的单type索引中。

注意事项

自2020年10月起，由于网络架构的调整，导致部分通过reindex方式跨集群迁移数据的场景受到了限制。您可以参见下表，查看您的业务场景是否支持reindex功能，并参考解决方案进行处理。

使用场景	网络状态（2020年10月之前属于旧网络架构，2020年10月及之后属于新网络架构）	是否支持reindex功能	解决方案
通过reindex方式迁移阿里云Elasticsearch集群间的数据	两个Elasticsearch集群均创建于旧网络架构下。	是	请参见通过reindex迁移数据。
	两个Elasticsearch集群均创建于新网络架构下。	否	不支持reindex方式，建议使用OSS或Logstash方式。具体操作请参见通过OSS将自建Elasticsearch数据迁移至阿里云和通过阿里云Logstash将自建Elasticsearch数据迁移至阿里云。
	两个Elasticsearch集群分别创建于旧网络架构下和新网络架构下。	否	
将ECS上自建的Elasticsearch集群中的数据迁移至阿里云Elasticsearch集群中	阿里云Elasticsearch集群创建于旧网络架构下。	是	请参见通过reindex将自建Elasticsearch数据迁移至阿里云。
	阿里云Elasticsearch集群创建于新网络架构下。	是	借助PrivateLink，打通ECS上自建Elasticsearch集群所处的网络与阿里云服务账号的网络，再使用终端节点域名进行reindex。详情请参见通过实例私网打通将自建Elasticsearch数据迁移至阿里云。 **说明** PrivateLink仅支持部分地域私网连接，详情请参见支持私网连接的地域和可用区。如果您的集群可用区不满足此条件，则不支持reindex功能。

操作流程

1. 准备工作

   准备阿里云Elasticsearch和Logstash实例，确保两者在同一专有网络下。

   - 阿里云Elasticsearch实例：存储索引数据。
   - 阿里云Logstash实例：通过管道配置功能，迁移处理后的数据。

2. 步骤一：转换索引类型

   通过reindex，将阿里云Elasticsearch 5.x实例中的多type索引转换为单type索引。您可以通过以下两种方式来实现：

   - 合并type方式：将Elasticsearch 5.x实例中的单索引多type数据，通过reindex script方式合并成一个

- 合并type方式：将Elasticsearch 5.x实例中的单索引多type数据，通过reindex script方式合并成一个单索引单type数据。
- 拆分type方式：将Elasticsearch 5.x实例中的单索引多type数据，按照不同的type，通过reindex拆分成多个单索引单type数据的方式。

3. 步骤二：通过Logstash迁移数据

   使用阿里云Logstash，将处理后的索引数据迁移至高版本Elasticsearch 6.x实例中。

4. 步骤三：查看数据迁移结果

   在Kibana中查看迁移成功的索引。

准备工作

1. 准备低版本（5.5.3）和高版本（6.7.0）的阿里云Elasticsearch实例，并准备待迁移的多type数据。

   创建实例的具体操作，请参见创建阿里云Elasticsearch实例。

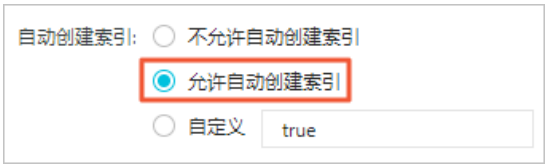
2. 创建阿里云Logstash实例，要求与阿里云Elasticsearch实例处于同一专有网络下。

   具体操作，请参见步骤一：创建阿里云Logstash实例。

步骤一：转换索引类型

以下步骤介绍通过合并type方式，将单索引多type数据合并成一个单索引单type数据。

1. 开启Elasticsearch实例的自动创建索引功能。
   i. 登录阿里云Elasticsearch控制台。
   ii. 在左侧导航栏，单击**Elasticsearch实例**。
   iii. 在顶部菜单栏处，选择资源组和地域。
   iv. 在**实例列表**中，单击低版本的实例ID。
   v. 在左侧导航栏，单击**ES集群配置**。
   vi. 单击**YML文件配置**右侧的**修改配置**。
   vii. 在**YML文件配置**页面，设置**自动创建索引**为**允许自动创建索引**。

   > **警告** 修改**自动创建索引**方式会触发实例重启，请确认后再操作。

   viii. 勾选**该操作会重启实例，请确认后操作**，单击**确定**。
2. 登录低版本Elasticsearch实例的Kibana控制台。

   具体操作，请参见登录Kibana控制台。

3. 在左侧导航栏，单击**Dev Tools**（开发工具）。
4. 在**Console**中，执行以下命令，将单索引多type数据合并成单索引单type数据。

```
POST _reindex
{
  "source": {
    "index": "twitter"
  },
  "dest": {
    "index": "new1"
  },
  "script": {
    "inline": """
    ctx._id = ctx._type + "-" + ctx._id;
    ctx._source.type = ctx._type;
    ctx._type = "doc";
    """,
    "lang": "painless"
  }
}
```

以上示例通过自定义type的方式，指定ctx._source.type在new1索引中添加type字段，将其设置为原始_type的值。并且new1索引的_id由_type-_id组成，防止存在不同类型的文档具有相同的ID而发生冲突的情况。

5. 执行 `GET new1/_mapping` 命令，查看合并后的Mapping结构。
6. 执行以下命令，查看合并后的索引数据。

```
GET new1/_search
{
   "query":{
     "match_all":{
      }
  }
}
```

以下步骤介绍通过拆分type方式，将单索引多type数据，按照不同的type，通过reindex拆分成多个单索引单type数据。

1. 在**Console**中，执行以下命令，将单索引多type拆分单索引单type。

```
POST _reindex
{
  "source": {
    "index": "twitter",
    "type": "tweet",
    "size": 10000
  },
  "dest": {
    "index": "twitter_tweet"
  }
}
POST _reindex
{
  "source": {
    "index": "twitter",
    "type": "user",
    "size": 10000
  },
  "dest": {
    "index": "twitter_user"
  }
}
```

以上示例将twitter索引按照不同type，分别拆分成twitter_tweet和twitter_user索引。

2. 执行以下命令，查看拆分后的索引数据。

```
GET twitter_tweet/_search
{
   "query":{
     "match_all":{
      }
  }
}
```

```
GET twitter_user/_search
{
   "query":{
     "match_all":{
      }
  }
}
```

步骤二：通过Logstash迁移数据

1. 登录阿里云Elasticsearch控制台。
2. 在顶部菜单栏处，选择地域。
3.
4. 在左侧导航栏，单击**管道管理**。
5. 在**管道管理**页面，使用**配置文件管理**方式配置管道。

   管道配置的具体操作，请参见通过配置文件管理管道，Config配置示例如下。

```
input {
    elasticsearch {
    hosts => ["http://es-cn-0pp1f1y5g000h****.elasticsearch.aliyuncs.com:9200"]
    user => "elastic"
    index => "*"
    password => "your_password"
    docinfo => true
  }
}
filter {
}
output {
  elasticsearch {
    hosts => ["http://es-cn-mp91cbxsm000c****.elasticsearch.aliyuncs.com:9200"]
    user => "elastic"
    password => "your_password"
    index => "test"
  }
}
```

6. 保存并部署管道配置，开始迁移数据。

   具体操作，请参见通过配置文件管理管道。

步骤三：查看数据迁移结果

1. 登录高版本Elasticsearch实例的Kibana控制台。

   具体操作，请参见登录Kibana控制台。

2. 在左侧导航栏，单击Dev Tools。

3. 在Console中，执行以下命令，查看迁移成功的索引。

```
GET _cat/indices?v
```

2.3. 自建Elasticsearch数据迁移

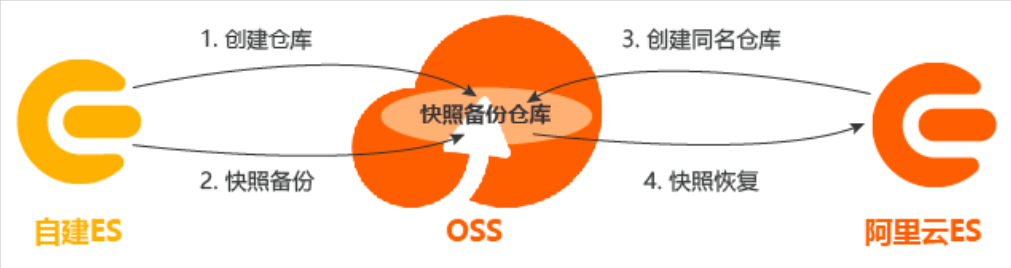
2.3.1. 通过OSS将自建Elasticsearch数据迁移至阿里云

OSS迁移自建ES数据到阿里云ES

当您需要将自建Elasticsearch数据迁移至阿里云Elasticsearch时，可以使用OSS快照的方式进行迁移。即使用Elasticsearch的snapshot API，创建自建Elasticsearch数据的快照并存储到OSS中，然后从OSS将快照数据恢复到阿里云Elasticsearch中。本文介绍具体的实现方法。

背景信息

通过OSS将自建Elasticsearch数据迁移至阿里云Elasticsearch，适用于自建Elasticsearch数据量比较大的场景，简单流程如下。



操作流程

1. 步骤一：环境准备

   部署自建Elasticsearch集群、创建OSS Bucket、创建阿里云Elasticsearch集群。

2. 步骤二：安装elasticsearch-repository-oss插件

   在自建Elasticsearch集群各节点中安装elasticsearch-repository-oss插件，插件安装后才可在自建Elasticsearch中创建OSS仓库。

3. 步骤三：在自建Elasticsearch集群中创建仓库

   使用snapshot API在自建Elasticsearch集群中创建快照备份仓库。

4. 步骤四：为指定索引创建快照

   为需要迁移的索引创建快照，并将快照备份到已创建的仓库中。

5. 步骤五：在阿里云Elasticsearch上创建相同仓库

   在阿里云Elasticsearch的Kibana控制台中，使用snapshot API创建一个与自建Elasticsearch集群相同的快照备份仓库。

6. 步骤六：在阿里云Elasticsearch上恢复快照

   将仓库中已备份的自建Elasticsearch集群的快照恢复到阿里云Elasticsearch集群中，完成数据迁移。

7. 步骤七：查看快照恢复结果

   快照恢复后，查看恢复的索引和索引数据。

步骤一：环境准备

1. 准备自建Elasticsearch集群。

   如果您还没有自建Elasticsearch集群，建议您使用阿里云ECS进行搭建，具体操作步骤请参见安装并运行Elasticsearch。

   本文以单节点集群为例进行演示，版本为6.7.0。实际生产中您可以购买多个相同专有网络VPC（Virtual Private Cloud）的ECS实例搭建多节点Elasticsearch集群，购买ECS的具体步骤请参见使用向导创建实例。

2. 开通OSS服务，并创建与部署自建Elasticsearch集群的ECS实例相同地域的Bucket。

   具体操作步骤请参见开通OSS服务和创建存储空间。

   > 注意 请创建标准存储类型的OSS Bucket，不支持归档存储类型。

3. 创建目标阿里云Elasticsearch实例，所选地域与您创建的Bucket相同。

   具体操作步骤请参见创建阿里云Elasticsearch实例。

步骤二：安装elasticsearch-repository-oss插件

1. 参见连接ECS实例，连接自建Elasticsearch集群所在的ECS。

2. 下载elasticsearch-repository-oss插件。

   本文使用6.7.0版本的插件，要求JDK为8.0及以上版本。

   ```
   wget https://github.com/aliyun/elasticsearch-repository-oss/releases/download/v6.7.0/elasticsearch-repository-oss-6.7.0.zip
   ```

   > **说明** 如需获取其他版本的elasticsearch-repository-oss插件，可参见常见问题。

3. 将安装包解压到自建Elasticsearch各节点安装路径的plugins目录下。

   ```
   unzip -d /usr/local/elasticsearch-6.7.0/plugins/elasticsearch-repository-oss elasticsearch-repository-oss-6.7.0.zip
   ```

   您也可以使用命令方式安装插件。

   ```
   ./bin/elasticsearch-plugin install file:///usr/local/elasticsearch-repository-oss-6.7.0.zip
   ```

4. 启动自建Elasticsearch集群各节点。

   ```
   cd /usr/local/elasticsearch-6.7.0/bin
   ./elasticsearch -d
   ```

步骤三：在自建Elasticsearch集群中创建仓库

连接自建Elasticsearch集群所在的ECS，执行如下命令创建仓库。

```
curl -H "Content-Type: application/json" -XPUT localhost:9200/_snapshot/<yourBackupName> -d'
{"type": "oss", "settings": { "endpoint": "http://oss-cn-hangzhou-internal.aliyuncs.com",  "
access_key_id": "<yourAccesskeyId>",  "secret_access_key":"<yourAccesskeySecret>", "bucket":
"<yourBucketName>", "compress": true }}'
```

参数	说明
<yourBackupName>	仓库名称，可自定义。
type	仓库类型，需要设置为oss。
endpoint	OSS Bucket的访问地址，可参见访问域名和数据中心获取。 **说明** 如果自建Elasticsearch所在ECS与您的OSS在同一地域，请使用内网地址，否则请使用外网地址。
access_key_id	创建OSS Bucket的账号的AccessKey ID，获取方式请参见如何获取AccessKey。
secret_access_key	创建OSS Bucket的账号的AccessKey Secret，获取方式请参见如何获取AccessKey。
bucket	您创建的OSS Bucket名称。

参数	说明
compress	是否压缩： • true：压缩 • false：不压缩

创建成功后，预期返回 `"acknowledge":true` 。

步骤四：为指定索引创建快照

在自建Elasticsearch中创建一个快照，用来备份您需要迁移的索引数据。创建快照时，默认会备份所有打开的索引。如果您不想备份系统索引，例如以.kibana、.security、.monitoring等开头的索引，可在创建快照时指定需要备份的索引。

> **注意** 建议您不要备份系统索引，因为系统索引会占用较大空间。

```
curl -H "Content-Type: application/json" -XPUT localhost:9200/_snapshot/<yourBackupName>/sna
pshot_1?pretty -d'
{
"indices": "index1,index2"
}'
```

<yourBackupName>为您在步骤三：在自建Elasticsearch集群中创建仓库中创建的仓库名称；index1和index2为您需要备份的索引名称。快照创建成功后，预期返回 `"accepted" : true` 。

步骤五：在阿里云Elasticsearch上创建相同仓库

1. 登录目标阿里云Elasticsearch实例的Kibana控制台，根据页面提示进入Kibana主页。

   登录Kibana控制台的具体操作，请参见登录Kibana控制台。

   > **说明** 本文以阿里云Elasticsearch 6.7.0版本为例，其他版本操作可能略有差别，请以实际界面为准。

2. 在左侧导航栏，单击**Dev Tools**。
3. 在**Console**中执行以下命令，创建与自建Elasticsearch相同的仓库。

   ```
   PUT _snapshot/<yourBackupName>
   {
       "type": "oss",
       "settings": {
           "endpoint": "oss-cn-hangzhou-internal.aliyuncs.com",
           "access_key_id": "<yourAccesskeyId>",
           "secret_access_key": "<yourAccesskeySecret>",
           "bucket": "<yourBucketName>",
           "compress": true
       }
   }
   ```

   <yourBackupName>和<yourBucketName>需要与步骤三：在自建Elasticsearch集群中创建仓库中保持一致。

步骤六：在阿里云Elasticsearch上恢复快照

参见步骤五：在阿里云Elasticsearch上创建相同仓库，在Kibana控制台上执行以下命令，恢复快照中的所有索引（除过 `.` 开头的系统索引）。

```
POST _snapshot/es_backup/snapshot_1/_restore
{"indices":"*,-.monitoring*,-.security_audit*","ignore_unavailable":"true"}
```

命令执行成功，预期返回 `"accepted" : true` 。

以上命令会恢复快照中的所有索引，您也可以选择需要恢复的索引。同时如果阿里云Elasticsearch集群中有同名索引，而您想在不替换现有数据的前提下恢复旧数据来验证内容，或者处理其他任务，可在恢复过程中重命名索引。

```
POST _snapshot/es_backup/snapshot_1/_restore
{
  "indices":"index1",
  "rename_pattern": "index(.+)",
  "rename_replacement": "restored_index_$1"
}
```

> **说明** 更多快照和恢复命令请参见手动备份与恢复。

步骤七：查看快照恢复结果

参见步骤五：在阿里云Elasticsearch上创建相同仓库，在Kibana控制台上执行以下命令，查看恢复结果：

- 查看恢复的索引

  ```
  GET /_cat/indices?v
  ```

  ```
  health status index
  green  open   .monitoring-kibana-6-2020.05.28
  green  open   .monitoring-es-6-2020.05.28
  green  open   index1
  green  open   .monitoring-logstash-6-2020.06.03
  green  open   .monitoring-kibana-6-2020.06.04
  green  open   .kibana_task_manager
  ```

- 查看恢复的索引数据

  ```
  GET /index1/_search
  ```

  执行成功后，预期结果如下。

```
{
  "took" : 2,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 1,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "index1",
        "_type" : "_doc",
        "_id" : "1",
        "_score" : 1.0,
        "_source" : {
          "productName" : "testpro",
          "annual_rate" : "3.22%",
          "describe" : "testpro"
        }
      }
    ]
  }
}
```

常见问题

Q：如何获取其他版本的elasticsearch-repository-oss插件？

A：可在Github上下载。如果Github上没有对应版本的插件包，建议您下载对应大版本的相近小版本的插件包，然后修改*plugin-descriptor.properties*文件中的参数值，重新打包再安装。

- version=所需插件的版本
- elasticsearch.version=自建Elasticsearch的版本

> **说明** 插件版本与自建Elasticsearch版本要保证一致。

- java.version=1.8

> **说明** 不同的Elasticsearch版本依赖的JDK版本不一样，以官方和插件要求为准。

例如，您要获取7.0.0版本的elasticsearch-repository-oss插件，可在Github上下载7.4.0版本的插件包，然后修改*plugin-descriptor.properties*文件中的参数值为：

- version=7.0.0
- elasticsearch.version=7.0.0
- java.version=1.8

2.3.2. 通过阿里云Logstash将自建Elasticsearch数据迁移至阿里云

自建es数据迁移至阿里云es

当您需要将自建Elasticsearch中的数据迁移到阿里云Elasticsearch中时，可以通过阿里云Logstash的管道配置功能实现。本文介绍具体的实现方法。

使用限制

- 自建Elasticsearch集群所在的ECS的网络类型必须是专有网络，不支持Classiclink方式打通的ECS。
- 由于阿里云Logstash实例部署在专有网络下，如果自建Elasticsearch集群与阿里云Logstash集群在同一专有网络下，可直接配置；如果不在，需要通过配置NAT网关实现与公网的连通，具体操作请参见配置NAT公网数据传输。
- 自建Elasticsearch集群所在的ECS的安全组不能限制Logstash集群的各节点IP（可在基本信息页面查看），并且需要开启9200端口。
- 本文以自建Elasticsearch 5.6.16 > 阿里云Logstash 6.7.0 > 阿里云Elasticsearch 6.7.0为例，提供的脚本仅适用于该数据迁移方案，其他方案不保证兼容。如果您使用的是其他方案，可参见产品兼容性判断是否存在兼容性问题。如果存在，可升级实例版本或新购实例。

操作流程

1. 步骤一：环境准备
2. 步骤二：配置并运行Logstash管道
3. 步骤三：查看数据迁移结果

步骤一：环境准备

1. 搭建自建Elasticsearch集群。

   建议您使用阿里云ECS搭建自建Elasticsearch集群，本文以5.6.16版本为例，具体操作请参见安装并运行Elasticsearch。

2. 创建阿里云Logstash实例。

   建议创建与部署自建Elasticsearch集群的ECS实例在同一专有网络下的Logstash实例，具体操作请参见创建阿里云Logstash实例。

3. 创建目标阿里云Elasticsearch实例，并开启实例的自动创建索引功能。
   - 建议创建与Logstash实例在同一专有网络下，且版本相同的Elasticsearch实例。本文以6.7.0版本为例，具体操作请参见创建阿里云Elasticsearch实例。
   - 开启自动创建索引功能的具体操作，请参见配置YML参数。

     > **说明** 因为Logstash只同步数据，不同步数据结构特征，所以开启自动创建索引功能后，可能会存在同步前后数据结构不一致的情况。如果您需要同步前后的数据结构一致，那么建议您先手动在目标端Elasticsearch中创建空索引，创建时，复制源端的mappings和settings结构，并合理分配shard数量。

步骤二：配置并运行Logstash管道

1. 登录阿里云Elasticsearch控制台。
2. 进入目标实例。

i. 在顶部菜单栏处，选择地域。

ii. 在左侧导航栏，单击**Logstash实例**，然后在**Logstash实例**中单击目标实例ID。

3. 在左侧导航栏，单击**管道管理**。
4. 单击**创建管道**。
5. 在**创建管道任务**页面，输入管道ID并配置管道。

本文使用的管道配置如下。

```
input {
  elasticsearch {
    hosts => ["http://<自建Elasticsearch Master节点的IP地址>:9200"]
    user => "elastic"
    index => "*,-.monitoring*,-.security*,-.kibana*"
    password => "your_password"
    docinfo => true
  }
}
filter {
}
output {
  elasticsearch {
    hosts => ["http://es-cn-mp91cbxsm000c****.elasticsearch.aliyuncs.com:9200"]
    user => "elastic"
    password => "your_password"
    index => "%{[@metadata][_index]}"
    document_type => "%{[@metadata][_type]}"
    document_id => "%{[@metadata][_id]}"
  }
  file_extend {
        path => "/ssd/1/ls-cn-v0h1kzca****/logstash/logs/debug/test"
    }
}
```

参数说明

参数	描述
hosts	自建或阿里云Elasticsearch服务的访问地址。input中为 `http://<自建Elasticsearch Master节点的IP地址>:<端口>` ；output中为 `http://<阿里云Elasticsearch实例ID>.elasticsearch.aliyuncs.com:9200` 。 **注意** 配置时，请按照格式要求，将<自建Elasticsearch Master节点的IP地址>、<端口>和<阿里云Elasticsearch实例ID>替换为对应的值。

参数	描述
user	访问自建或阿里云Elasticsearch服务的用户名。 **注意** • user和password为必选参数。如果自建Elasticsearch未安装X-Pack，可将这两个参数值设置为空。 • 访问阿里云Elasticsearch实例的用户名默认为elastic（本文以此为例）。如果想使用自建用户，需要为自建用户分配相应的角色和权限，详情请参见通过Elasticsearch X-Pack角色管理实现用户权限管控。
password	访问自建或阿里云Elasticsearch服务的密码。
index	指定同步索引名。input中设置为*,-.monitoring*,-.security*,-.kibana*，表示同步除过 . 开头的系统索引；output中设置为%{[@metadata][_index]}，表示匹配元数据中的index，即阿里云Elasticsearch生成的索引和自建Elasticsearch的索引相同。
docinfo	设置为true，阿里云Elasticsearch会提取自建Elasticsearch文档的元数据信息，例如index、type和id。
document_type	设置为%{[@metadata][_type]}，表示匹配元数据中索引的type，即阿里云Elasticsearch生成的索引的类型和自建Elasticsearch的索引类型相同。
document_id	设置为%{[@metadata][_id]}，表示匹配元数据中文档的id，即阿里云Elasticsearch生成的文档ID和自建Elasticsearch的文档ID相同。
file_extend	可选，用来开启调试日志功能，并通过path参数配置调试日志的输出路径。建议您配置该参数，配置后，可直接在控制台上查看输出结果。如果未配置，需要去目标端确认输出结果，再返回控制台修改，这样会耗费大量的时间和人力。详细信息，请参见使用Logstash管道配置调试功能。 **注意** 使用file_extend参数前，需要先安装logstash-output-file_extend插件。具体操作，请参见安装或卸载插件。其中的path参数默认为系统指定路径，请勿修改。您也可以单击**开启配置调试**获取path路径。

Elasticsearch input插件可以根据配置的查询语句，从Elasticsearch集群读取文档数据，适用于批量导入测试日志等操作。默认读取完数据后，同步动作会自动关闭。如果您想实现定时触发数据同步，可以通过cron语法配合schedule参数实现，详情请参见Logstash官网Scheduling介绍。

```
schedule => "* * * * *"
```

更多Config配置说明，请参见Logstash配置文件说明。

6. 单击**下一步**，配置管道参数。

参数	说明
管道工作线程	并行执行管道的Filter和Output的工作线程数量。当事件出现积压或CPU未饱和时，请考虑增大线程数，更好地使用CPU处理能力。默认值：实例的CPU核数。
管道批大小	单个工作线程在尝试执行Filter和Output前，可以从Input收集的最大事件数目。较大的管道批大小可能会带来较大的内存开销。您可以设置LS_HEAP_SIZE变量，来增大JVM堆大小，从而有效使用该值。默认值：125。
管道批延迟	创建管道事件批时，将过小的批分派给管道工作线程之前，要等候每个事件的时长，单位为毫秒。默认值：50ms。
队列类型	用于事件缓冲的内部排队模型。可选值： ○ **MEMORY**：默认值。基于内存的传统队列。 ○ **PERSISTED**：基于磁盘的ACKed队列（持久队列）。
队列最大字节数	请确保该值小于您的磁盘总容量。默认值：1024 MB。
队列检查点写入数	启用持久性队列时，在强制执行检查点之前已写入事件的最大数目。设置为0，表示无限制。默认值：1024。

警告 配置完成后，需要保存并部署才能生效。保存并部署操作会触发实例重启，请在不影响业务的前提下，继续执行以下步骤。

7. 单击**保存**或者**保存并部署**。
   - **保存**：将管道信息保存在Logstash里并触发实例变更，配置不会生效。保存后，系统会返回**管道管理**页面。可在**管道列表**区域，单击**操作**列下的**立即部署**，触发实例重启，使配置生效。
   - **保存并部署**：保存并且部署后，会触发实例重启，使配置生效。

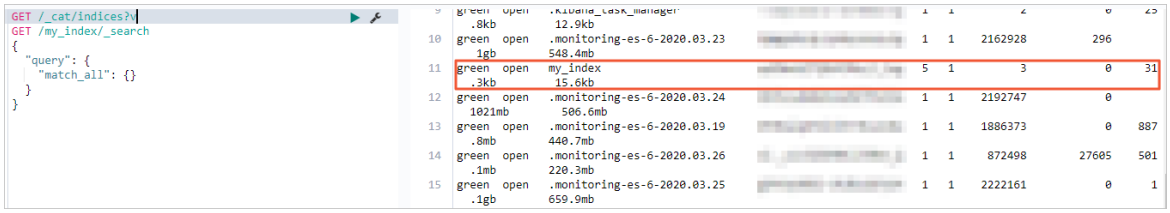
步骤三：查看数据迁移结果

1. 登录目标阿里云Elasticsearch实例的Kibana控制台，根据页面提示进入Kibana主页。

   登录Kibana控制台的具体操作，请参见登录Kibana控制台。

   说明 本文以阿里云Elasticsearch 6.7.0版本为例，其他版本操作可能略有差别，请以实际界面为准。

2. 在左侧导航栏，单击Dev Tools。
3. 在Console中，执行 `GET /_cat/indices?v` 命令，查看迁移成功的索引。



常见问题

- Q：自建Elasticsearch所在的ECS与阿里云Logstash不在同一账号下，迁移数据时，如何配置网络互通？

  A：由于ECS和Logstash不在同一账号下，那么两者的专有网络必然不同，因此需要配置两个专有网络互通。可通过云企业网来实现，具体操作请参见步骤三：加载网络实例。

- Q：Logstash数据写入时出现问题，如何处理？

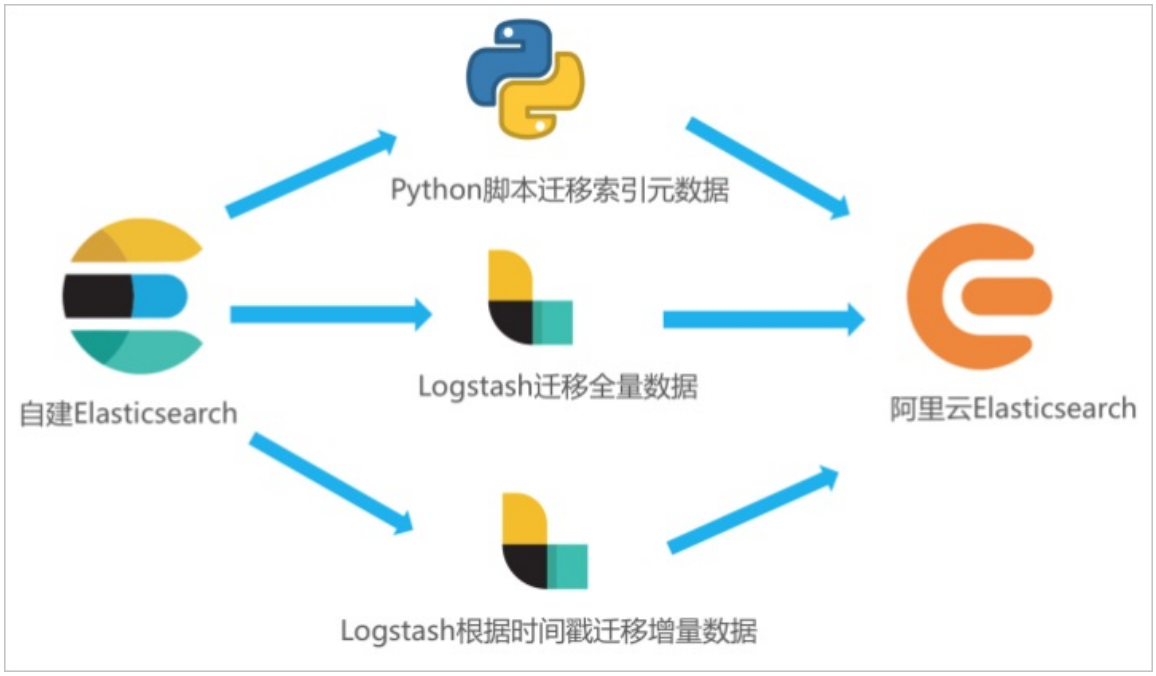
  A：参见Logstash数据写入问题排查方案进行排查处理。

2.3.3. 通过Logstash将自建Elasticsearch数据全量或增量迁移至阿里云

当您需要将自建Elasticsearch中的全量或增量数据迁移至阿里云Elasticsearch时，可通过在ECS中自建Logstash，并通过Logstash的管道配置功能实现。本文为您介绍具体的实现方法。

背景信息

本文中数据迁移的流程如下。



1. 在ECS服务器部署自建Elasticsearch并准备待迁移的数据。
2. 开通阿里云Elasticsearch服务。

3. 在ECS服务器运行Python脚本迁移索引元数据。
4. 部署Logstash，并通过Logstash管道配置功能，将自建Elasticsearch中的全量或增量数据迁移至阿里云Elasticsearch中。

注意事项

- 本文在阿里云ECS上部署自建Logstash，该Logstash所在的ECS需要与阿里云Elasticsearch集群在同一专有网络下，同时该Logstash需要能够同时访问源Elasticsearch集群（自建）和目标Elasticsearch集群（阿里云）。
- 数据迁移可以全量迁移或增量迁移，首次迁移都是全量迁移，后续写入数据选择增量迁移，增量迁移需要索引有时间戳字段。

操作流程

1. 步骤一：准备环境与实例
2. 步骤二：迁移索引元数据（设置和映射）
3. 步骤三：迁移全量数据
4. 步骤四：迁移增量数据
5. 步骤五：查看数据迁移结果

步骤一：准备环境与实例

1. 创建阿里云Elasticsearch实例。

   具体操作请参见创建阿里云Elasticsearch实例。本文使用的测试环境如下。

环境项	环境信息
地域	华东1（杭州）。
版本	通用商业版7.10.0。
实例规格配置	3个可用区、3个数据节点、单节点4核CPU、16 GB内存、100 GB ESSD云盘。

2. 创建ECS实例，用于部署自建Elasticsearch、自建Kibana和自建Logstash。

   具体操作请参见使用向导创建实例。本文使用的测试环境如下。

环境项	环境信息
地域	华东1（杭州）。
实例规格	4 vCPU 16 GiB内存。
镜像	公共镜像、CentOS、7.9 64位。
存储	系统盘、ESSD云盘、100 GiB。
网络	与阿里云Elasticsearch相同的专有网络，选中**分配公网IPv4地址**，并按使用流量计费，带宽峰值为100 Mbps。

环境项	环境信息
安全组	入方向添加5601端口（即Kibana端口），在授权对象中添加您客户端的IP地址。 **注意** • 如果您的客户端处在家庭网络或公司局域网中，您需要在授权对象中添加局域网的公网出口IP地址，而非客户端机器的IP地址。建议您通过淘宝IP地址库查看您当前使用的公网IP。 • 您也可以在授权对象中添加0.0.0.0/0，表示允许所有IPv4地址访问ECS实例。此配置会导致ECS实例完全暴露在公网中，增加安全风险，生产环境尽量避免。

3. 部署自建Elasticsearch。

   本文使用的自建Elasticsearch版本为7.6.2，1个数据节点，具体操作步骤如下：

   i. 连接ECS服务器。

      具体操作请参见通过密码或密钥认证登录Linux实例。

   ii. 创建elastic用户。

```
useradd elastic
passwd <your_password>
```

   iii. root用户切换为elastic用户。

```
su -l elastic
```

   iv. 下载Elasticsearch软件安装包并解压缩。

```
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.6.2-linux-x86_64.tar.gz
tar -zvxf elasticsearch-7.6.2-linux-x86_64.tar.gz
```

   v. 使用非root用户启动Elasticsearch。

      进入Elasticsearch的安装目录下，启动服务。

```
cd elasticsearch-7.6.2
./bin/elasticsearch -d
```

vi. 验证Elasticsearch服务是否正常运行。

```
cd ~
curl localhost:9200
```

正常情况下，返回结果中会显示Elasticsearch版本号和"You Know, for Search"。

```
[elastic@vm01 ~]$ curl localhost:9200
{
  "name" : "vm01",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "SRB4pnk4SmS-YHzsr[redacted]",
  "version" : {
    "number" : "7.6.2",
    "build_flavor" : "default",
    "build_type" : "tar",
    "build_hash" : "ef48eb35cf30adf4db14086e8aabd07ef6[redacted]",
    "build_date" : "2020-03-26T06:34:37.794943Z",
    "build_snapshot" : false,
    "lucene_version" : "8.4.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

4. 部署自建Kibana，并准备测试数据。

本文使用的自建Kibana版本为7.6.2，1个数据节点，具体操作步骤如下：

i. 连接ECS服务器。

具体操作请参见通过密码或密钥认证登录Linux实例。

ii. 下载Kibana软件安装包并解压缩。

```
wget https://artifacts.elastic.co/downloads/kibana/kibana-7.6.2-linux-x86_64.tar.gz
tar -zvxf kibana-7.6.2-linux-x86_64.tar.gz
```

iii. 修改Kibana配置文件*config/kibana.yml*，增加 `server.host: "0.0.0.0"` 配置，允许通过公网IP访问Kibana。

进入Kibana安装目录，修改*kibana.yml*。

```
cd kibana-7.6.2-linux-x86_64
vi config/kibana.yml
```

```
# Kibana is served by a back end server. This setting specifies the port to use.
#server.port: 5601

# Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid values.
# The default is 'localhost', which usually means remote machines will not be able to connect.
# To allow connections from remote users, set this parameter to a non-loopback address.
#server.host: "localhost"
server.host: "0.0.0.0"
# Enables you to specify a path to mount Kibana at if you are running behind a proxy.
# Use the `server.rewriteBasePath` setting to tell Kibana if it should remove the basePath
# from requests it receives, and to prevent a deprecation warning at startup.
# This setting cannot end in a slash.
#server.basePath: ""
```

iv. 使用非root用户启用Kibana。

```
nohup ./bin/kibana &
```

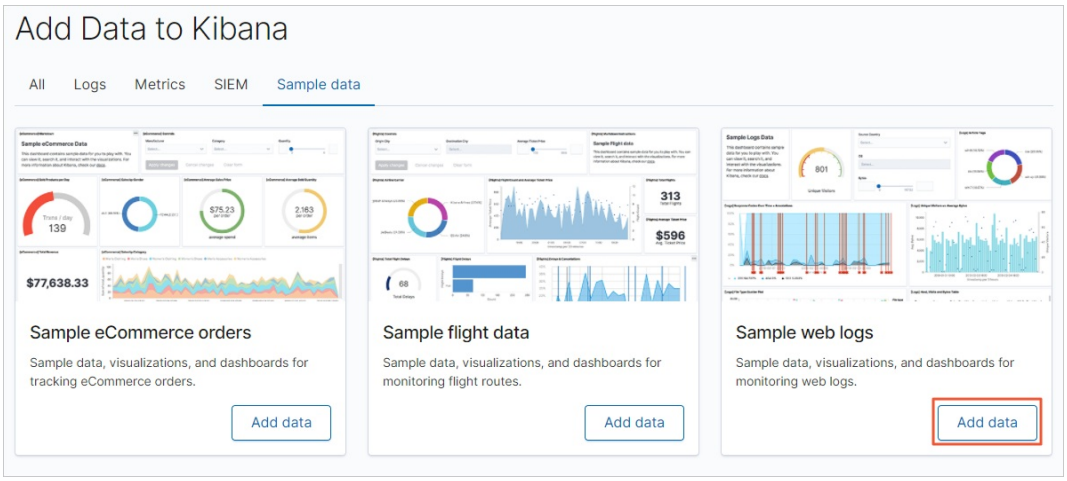
v. 登录Kibana控制台，添加示例数据。

a. 通过公网IP地址登录Kibana控制台。

公网IP地址为：**http://<ECS服务器的公网IP地址>:5601/app/kibana#/home**。

b. 在Kibana控制台首页，单击**Try our sample data**。

c. 在**Sample data**页签，单击日志示例数据模块下的**Add data**，添加对应数据。



5. 部署自建Logstash。

本文使用的Logstash版本为7.10.0，1个节点，具体操作步骤如下：

i. 连接ECS服务器。

具体操作请参见通过密码或密钥认证登录Linux实例。

ii. 回到跟目录，下载Logstash软件安装包并解压缩。

```
cd ~
wget https://artifacts.elastic.co/downloads/logstash/logstash-7.10.0-linux-x86_64.tar.gz
tar -zvxf logstash-7.10.0-linux-x86_64.tar.gz
```

iii. 修改Logstash的堆内存使用。

Logstash默认的堆内存为1 GB，您需要根据ECS规格配置合适的内存大小，加快集群数据的迁移效率。

进入Logstash的安装目录下，修改Logstash配置文件config/jvm.options，增加-Xms8g和-Xmx8g。

```
cd logstash-7.10.0
vi config/jvm.options
```

```
## JVM configuration

# Xms represents the initial size of total heap space
# Xmx represents the maximum size of total heap space

-Xms8g
-Xmx8g

################################################################
## Expert settings
################################################################
##
## All settings below this section are considered
## expert settings. Don't tamper with them unless
## you understand what you are doing
##
################################################################

## GC configuration
-XX:+UseConcMarkSweepGC
-XX:CMSInitiatingOccupancyFraction=75
-XX:+UseCMSInitiatingOccupancyOnly

## Locale
```

iv. 修改Logstash批量写入记录条数。

每批量写入5~15 MB数据，可以加快集群数据的迁移效率。

修改Logstash配置文件config/pipelines.yml，将每批量写入记录条数pipeline.batch.size从125改为5000。

```
vi config/pipelines.yml
```

```
#   # The path from where to read the configuration text
#   path.config: "/etc/conf.d/logstash/myconfig.cfg"
#
#   # How many worker threads execute the Filters+Outputs stage of the pipeline
#   pipeline.workers: 1 (actually defaults to number of CPUs)
#
   # How many events to retrieve from inputs before sending to filters+workers
   pipeline.batch.size: 5000
#
#   # How long to wait in milliseconds while polling for the next event
#   # before dispatching an undersized batch to filters+outputs
#   pipeline.batch.delay: 50
#
#   # Internal queuing model, "memory" for legacy in-memory based queuing and
#   # "persisted" for disk-based acked queueing. Defaults is memory
#   queue.type: memory
#
```

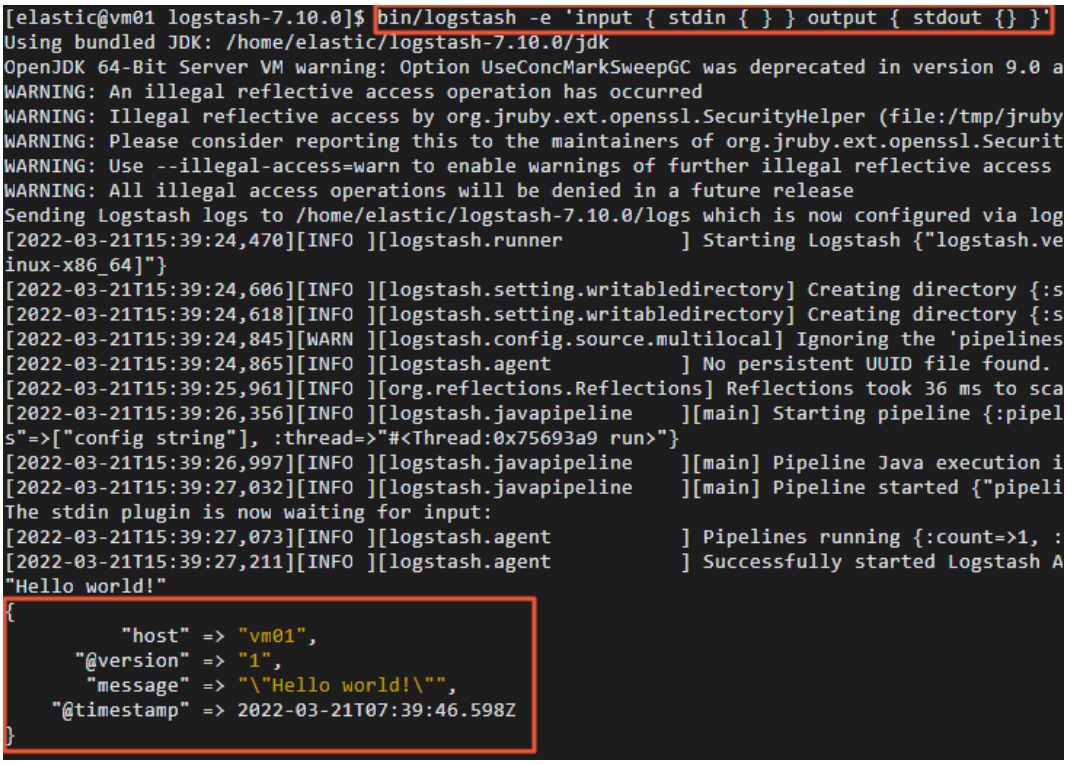
v. 验证Logstash功能。

a. 通过控制台输入输出收集数据。

```
bin/logstash -e 'input { stdin { } } output { stdout {} }'
```

b. 在控制台中输入*"Hello world!"*。

正常情况下，控制台会输出"Hello world!"。



步骤二：迁移索引元数据（设置和映射）

在进行数据迁移时，Logstash会帮助您自动创建索引，但是自动创建的索引可能与您待迁移的索引存在差异，导致迁移前后数据的格式不一致。因此建议您在数据迁移前，在阿里云Elasticsearch中手动创建目标索引，确保迁移前后索引数据完全一致。

您可以通过Python脚本创建目标索引，具体操作步骤如下：

1. 连接ECS服务器。

   具体操作请参见通过密码或密钥认证登录Linux实例。

2. 创建并打开Python脚本文件（本文以indiceCreate.py为例）。

```
vi indiceCreate.py
```

3. 修改Python脚本文件，拷贝以下代码（注意修改集群的连接地址、用户名和密码）。

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
# 文件名：indiceCreate.py
import sys
import base64
import time
import httplib
import json
```

```
## 源集群host。
oldClusterHost = "localhost:9200"
## 源集群用户名，可为空。
oldClusterUserName = "elastic"
## 源集群密码，可为空。
oldClusterPassword = "xxxxxx"
## 目标集群host，可在阿里云Elasticsearch实例的基本信息页面获取。
newClusterHost = "es-cn-zvp2m4bko0009****.elasticsearch.aliyuncs.com:9200"
## 目标集群用户名。
newClusterUser = "elastic"
## 目标集群密码。
newClusterPassword = "xxxxxx"
DEFAULT_REPLICAS = 0
def httpRequest(method, host, endpoint, params="", username="", password=""):
    conn = httplib.HTTPConnection(host)
    headers = {}
    if (username != "") :
        'Hello {name}, your age is {age} !'.format(name = 'Tom', age = '20')
        base64string = base64.encodestring('{username}:{password}'.format(username = use
rname, password = password)).replace('\n', '')
        headers["Authorization"] = "Basic %s" % base64string;
    if "GET" == method:
        headers["Content-Type"] = "application/x-www-form-urlencoded"
        conn.request(method=method, url=endpoint, headers=headers)
    else :
        headers["Content-Type"] = "application/json"
        conn.request(method=method, url=endpoint, body=params, headers=headers)
    response = conn.getresponse()
    res = response.read()
    return res
def httpGet(host, endpoint, username="", password=""):
    return httpRequest("GET", host, endpoint, "", username, password)
def httpPost(host, endpoint, params, username="", password=""):
    return httpRequest("POST", host, endpoint, params, username, password)
def httpPut(host, endpoint, params, username="", password=""):
    return httpRequest("PUT", host, endpoint, params, username, password)
def getIndices(host, username="", password=""):
    endpoint = "/_cat/indices"
    indicesResult = httpGet(oldClusterHost, endpoint, oldClusterUserName, oldClusterPass
word)
    indicesList = indicesResult.split("\n")
    indexList = []
    for indices in indicesList:
        if (indices.find("open") > 0):
            indexList.append(indices.split()[2])
    return indexList
def getSettings(index, host, username="", password=""):
    endpoint = "/" + index + "/_settings"
    indexSettings = httpGet(host, endpoint, username, password)
    print (index + "  原始settings如下：\n" + indexSettings)
    settingsDict = json.loads(indexSettings)
    ## 分片数默认和源集群索引保持一致。
    number_of_shards = settingsDict[index]["settings"]["index"]["number_of_shards"]
    ## 副本数默认为0。
    number_of_replicas = DEFAULT_REPLICAS
```

```
    number_of_replicas = DEFAULT_REPLICAS
    newSetting = "\"settings\": {\"number_of_shards\": %s, \"number_of_replicas\": %s}"
% (number_of_shards, number_of_replicas)
    return newSetting
def getMapping(index, host, username="", password=""):
    endpoint = "/" + index + "/_mapping"
    indexMapping = httpGet(host, endpoint, username, password)
    print (index + " 原始mapping如下：\n" + indexMapping)
    mappingDict = json.loads(indexMapping)
    mappings = json.dumps(mappingDict[index]["mappings"])
    newMapping = "\"mappings\" : " + mappings
    return newMapping
def createIndexStatement(oldIndexName):
    settingStr = getSettings(oldIndexName, oldClusterHost, oldClusterUserName, oldCluste
rPassword)
    mappingStr = getMapping(oldIndexName, oldClusterHost, oldClusterUserName, oldCluster
Password)
    createstatement = "{\n" + str(settingStr) + ",\n" + str(mappingStr) + "\n}"
    return createstatement
def createIndex(oldIndexName, newIndexName=""):
    if (newIndexName == "") :
        newIndexName = oldIndexName
    createstatement = createIndexStatement(oldIndexName)
    print ("新索引 " + newIndexName + " 的setting和mapping如下：\n" + createstatement)
    endpoint = "/" + newIndexName
    createResult = httpPut(newClusterHost, endpoint, createstatement, newClusterUser, ne
wClusterPassword)
    print ("新索引 " + newIndexName + " 创建结果：" + createResult)
## main
indexList = getIndices(oldClusterHost, oldClusterUserName, oldClusterPassword)
systemIndex = []
for index in indexList:
    if (index.startswith(".")):
        systemIndex.append(index)
    else :
        createIndex(index, index)
if (len(systemIndex) > 0) :
    for index in systemIndex:
        print (index + " 或许是系统索引，不会重新创建，如有需要，请单独处理～")
```

4. 执行Python脚本，创建目标索引。

```
/usr/bin/python indiceCreate.py
```

5. 参见登录Kibana控制台，登录目标集群的Kibana控制台，查看已创建的索引。

```
GET /_cat/indices?v
```

步骤三：迁移全量数据

1. 连接ECS服务器。

   具体操作请参见通过密码或密钥认证登录Linux实例。

2. 在config目录下，创建并打开Logstash配置文件。

```
cd logstash-7.10.0/config
vi es2es_all.conf
```

3. 参考以下配置，修改Logstash配置文件。

```
input{
    elasticsearch{
        # 源端ES地址。
        hosts =>  ["http://localhost:9200"]
        # 安全集群配置登录用户名密码。
        user => "xxxxxx"
        password => "xxxxxx"
        # 需要迁移的索引列表，多个索引以英文以逗号（,）分隔。
        index => "kibana_sample_data_*"
        # 以下三项保持默认即可，包含线程数和迁移数据大小和Logstash JVM配置相关。
        docinfo=>true
        slices => 5
        size => 5000
    }
}
filter {
  # 去掉一些Logstash自己加的字段。
  mutate {
    remove_field => ["@timestamp", "@version"]
  }
}
output{
    elasticsearch{
        # 目标端ES地址，可在阿里云Elasticsearch实例的基本信息页面获取。
        hosts => ["http://es-cn-zvp2m4bko0009****.elasticsearch.aliyuncs.com:9200"]
        # 安全集群配置登录用户名密码。
        user => "elastic"
        password => "xxxxxx"
        # 目标端索引名称，以下配置表示索引与源端保持一致。
        index => "%{[@metadata][_index]}"
        # 目标端索引type，以下配置表示索引类型与源端保持一致。
        document_type => "%{[@metadata][_type]}"
        # 目标端数据的id，如果不需要保留原id，可以删除以下这行，删除后性能会更好。
        document_id => "%{[@metadata][_id]}"
        ilm_enabled => false
        manage_template => false
    }
}
```

> **说明** 为了保证迁移数据的准确性，建议您创建多个Logstash管道配置文件，分批次迁移数据，每个Logstash迁移部分数据。

4. 启动Logstash全量迁移任务。

```
cd ../
nohup bin/logstash -f config/es2es_all.conf >/dev/null 2>&1 &
```

步骤四：迁移增量数据

1. 连接ECS服务器，在config目录下，创建并打开Logstash增量配置文件。

```
cd config
vi es2es_kibana_sample_data_logs.conf
```

2. 参考以下配置，修改Logstash配置文件。

   开启Logstash定时任务即可触发增量迁移，配置示例如下。

```
input{
    elasticsearch{
        # 源端ES地址。
        hosts =>  ["http://localhost:9200"]
        # 安全集群配置登录用户名密码。
        user => "xxxxxx"
        password => "xxxxxx"
        # 需要迁移的索引列表，多个索引使用英文逗号（,）分隔。
        index => "kibana_sample_data_logs"
        # 按时间范围查询增量数据，以下配置表示查询最近5分钟的数据。
        query => '{"query":{"range":{"@timestamp":{"gte":"now+8h-5m","lte":"now+8h/m"}}}
}'
        # 定时任务，以下配置表示每分钟执行一次。
        schedule => "* * * * *"
        scroll => "5m"
        docinfo=>true
        size => 5000
    }
}
filter {
  # 去掉一些Logstash自己加的字段.
  mutate {
    remove_field => ["@timestamp", "@version"]
  }
}
output{
    elasticsearch{
        # 目标端ES地址，可在阿里云Elasticsearch实例的基本信息页面获取。
        hosts => ["http://es-cn-zvp2m4bko0009****.elasticsearch.aliyuncs.com:9200"]
        # 访问集群的用户名和密码。
        user => "elastic"
        password => "xxxxxx"
        # 目标端索引名称，以下配置表示索引与源端保持一致。
        index => "%{[@metadata][_index]}"
        # 目标端索引type，以下配置表示索引类型与源端保持一致。
        document_type => "%{[@metadata][_type]}"
        # 目标端数据的id，如果不需要保留原id，可以删除以下这行，删除后性能会更好。
        document_id => "%{[@metadata][_id]}"
        ilm_enabled => false
        manage_template => false
    }
}
```

> 注意 Logstash记录的时间戳为UTC时间，如果您的本地时间为北京时间（东八区），那么两者会存在8个小时的时区差，此时需要将本地时间转化为UTC时间，可使用公式：UTC+时区差=北京时间。例如，以上示例input中的时间字段@timestamp，在进行range范围过滤查询获取增量数据时，需要在对应的时间上+8h转换为UTC时间。

3. 启动Logstash增量迁移任务。

```
nohup bin/logstash -f config/es2es_kibana_sample_data_logs.conf >/dev/null 2>&1 &
```

4. 在目标端Elasticsearch集群的Kibana中，查询最近更新的记录，验证增量数据是否同步。

   以下示例的查询条件为：索引名称为kibana_sample_data_logs、最近时间范围为5分钟。

```
GET kibana_sample_data_logs/_search
{
  "query": {
    "range": {
      "@timestamp": {
        "gte": "now-5m",
        "lte": "now/m"
      }
    }
  },
  "sort": [
    {
      "@timestamp": {
        "order": "desc"
      }
    }
  ]
}
```

步骤五：查看数据迁移结果

1. 查看是否完成全量迁移。

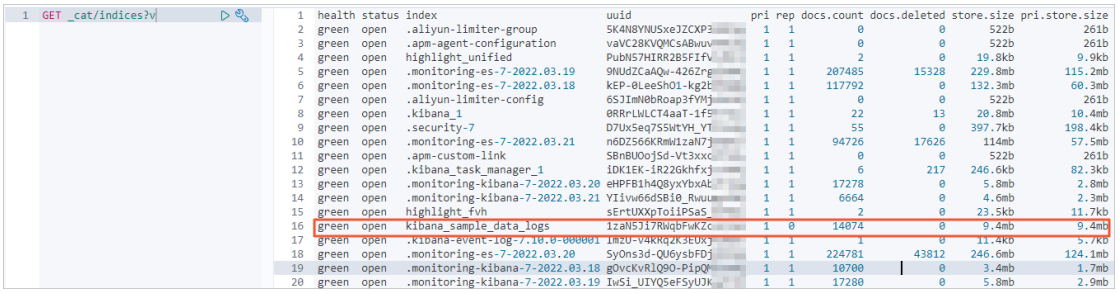
   i. 查看自建Elasticsearch集群的索引和数据量信息。

```
GET _cat/indices?v
```

   结果如下。

ii. 全量迁移后，查看阿里云Elasticsearch集群索引和数据量信息。

正常情况下，返回的记录条数应该与自建Elasticsearch集群一致。



```
GET _cat/indices?v
```

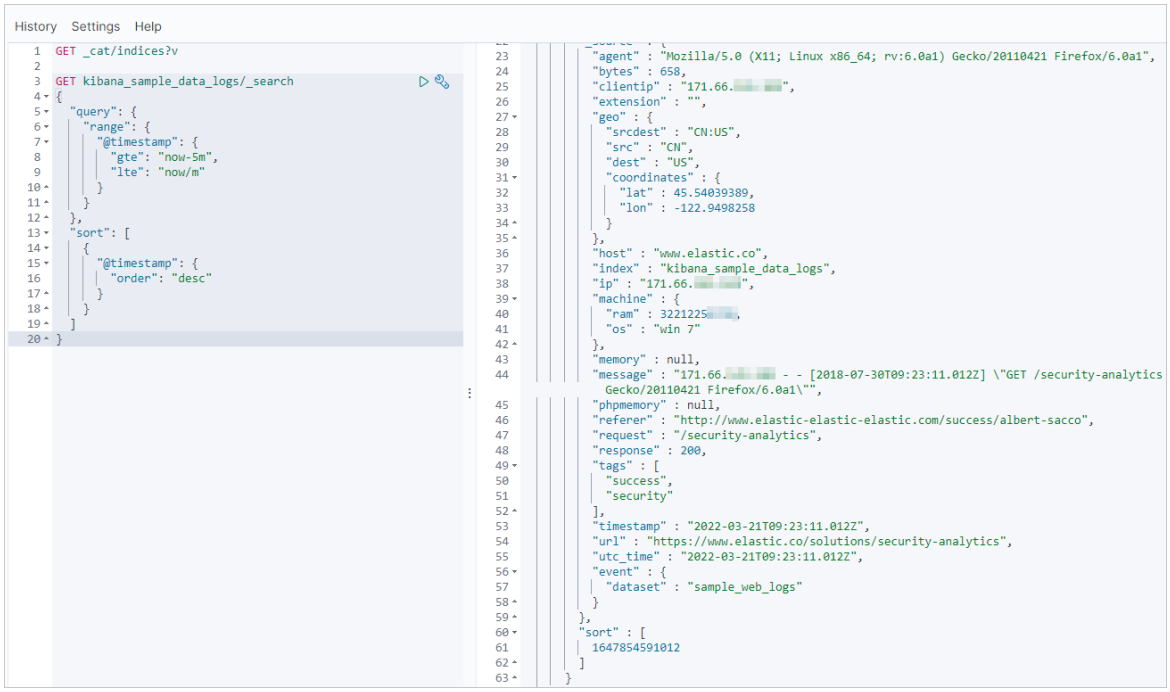
health	status	index	uuid	pri	rep	docs.count	docs.deleted	store.size	pri.store.size
green	open	.aliyun-limiter-group	5K4N8YNUSxeJZCXP3	1	1	0	0	522b	261b
green	open	.apm-agent-configuration	vaVC28KVQMCsABwuv	1	1	0	0	522b	261b
green	open	highlight_unified	PubN57HIRR2B5FIfV	1	1	2	0	19.8kb	9.9kb
green	open	.monitoring-es-7-2022.03.19	9NUdZCaAQw-426Zrg	1	1	207485	15328	229.8mb	115.2mb
green	open	.monitoring-es-7-2022.03.18	kEP-0LeeShO1-kg2E	1	1	117792	0	132.3mb	60.3mb
green	open	.aliyun-limiter-config	6SJImW0bRoap3fYMj	1	1	0	0	522b	261b
green	open	.kibana_1	0RRrLWLCT4aaT-1fS	1	1	22	13	20.8mb	10.4mb
green	open	.security-7	D7Ux5eq7S5WtYH_YI	1	1	55	0	397.7kb	198.4kb
green	open	.monitoring-es-7-2022.03.21	n6DZ566KRmW1zaN7j	1	1	94726	17626	114mb	57.5mb
green	open	.apm-custom-link	SBnBUOojSd-Vt3xxC	1	1	0	0	522b	261b
green	open	.kibana_task_manager_1	iDK1EK-iR22GkhfxJ	1	1	6	217	246.6kb	82.3kb
green	open	.monitoring-kibana-7-2022.03.20	eHPFBIh4Q8yxYbxAL	1	1	17278	0	5.8mb	2.8mb
green	open	.monitoring-kibana-7-2022.03.21	YIivw66d5Bi0_Rwuu	1	1	6664	0	4.6mb	2.3mb
green	open	highlight_fvh	sErtUXXpToiiPSaS_	1	1	2	0	23.5kb	11.7kb
green	open	kibana_sample_data_logs	1zaN5J17RWqbFwKZC	1	0	14074	0	9.4mb	9.4mb
green	open	.kibana-event-log-7.16.0-000001	1mzU-V4KRQzK3tUXj	1	1	1	0	11.4kb	5.7kb
green	open	.monitoring-es-7-2022.03.20	SyOns3d-QU6ysbFDj	1	1	224781	43812	246.6mb	124.1mb
green	open	.monitoring-kibana-7-2022.03.18	gOvcKvRlQ9O-PipQM	1	1	10700	0	3.4mb	1.7mb
green	open	.monitoring-kibana-7-2022.03.19	IwSi_UIYQ5eFSyUJK	1	1	17280	0	5.8mb	2.9mb

2. 查看是否完成增量迁移。

查看自建Elasticsearch集群的最近更新记录。

```
GET kibana_sample_data_logs/_search
{
  "query": {
    "range": {
      "@timestamp": {
        "gte": "now-5m",
        "lte": "now/m"
      }
    }
  },
  "sort": [
    {
      "@timestamp": {
        "order": "desc"
      }
    }
  ]
}
```

返回结果如下。



增量迁移完成后，使用同样命令查看阿里云Elasticsearch集群最近的更新记录。正常情况下，阿里云Elasticsearch集群的更新记录会与自建集群一致。

2.3.4. 通过reindex将自建Elasticsearch数据迁移至阿里云

本文介绍通过reindex方式，将ECS上自建Elasticsearch集群中的数据迁移至阿里云Elasticsearch中，包括创建索引和迁移数据。

背景信息

通过reindex迁移数据，仅支持单可用区实例。如果您使用的是多可用区实例，建议采用如下方案将自建Elasticsearch数据迁移至阿里云：

- 如果源端数据量较大，建议采用OSS快照方式。具体操作，请参见通过OSS将自建Elasticsearch数据迁移至阿里云。
- 如果需要对源端数据进行过滤，建议采用Logstash迁移方案。具体操作，请参见通过阿里云Logstash将自建Elasticsearch数据迁移至阿里云。

前提条件

您已完成以下操作：

- 创建单可用区的阿里云Elasticsearch实例。

  具体操作请参见创建阿里云Elasticsearch实例。

- 准备自建Elasticsearch集群和待迁移的数据。

  如果您还没有自建Elasticsearch集群，建议您使用阿里云ECS进行搭建，具体操作步骤请参见安装并运行Elasticsearch。自建Elasticsearch集群需要满足以下条件：

- 所在的ECS的网络类型必须是专有网络（不支持ClassicLink方式打通的ECS），且必须与阿里云Elasticsearch在同一个专有网络下。
- 所在的ECS的安全组不能限制阿里云Elasticsearch实例的各节点IP（Kibana控制台可查看各节点的IP），且要开启9200端口。
- 能够与阿里云Elasticsearch实例连通。可在执行脚本的机器上，使用`curl -XGET http://<host>:9200`命令验证。

> **说明** 您可以通过任意一台机器执行文档中的脚本，前提是该机器可以同时访问自建Elasticsearch和阿里云Elasticsearch集群的9200端口。

使用限制

自2020年10月起，由于网络架构的调整，导致部分通过reindex方式跨集群迁移数据的场景受到了限制。您可以参见下表，查看您的业务场景是否支持reindex功能，并参考解决方案进行处理。

使用场景	网络状态（2020年10月之前属于旧网络架构，2020年10月及之后属于新网络架构）	是否支持reindex功能	解决方案
通过reindex方式迁移阿里云Elasticsearch集群间的数据	两个Elasticsearch集群均创建于旧网络架构下。	是	请参见通过reindex迁移数据。
	两个Elasticsearch集群均创建于新网络架构下。	否	不支持reindex方式，建议使用OSS或Logstash方式。具体操作请参见通过OSS将自建Elasticsearch数据迁移至阿里云和通过阿里云Logstash将自建Elasticsearch数据迁移至阿里云。
	两个Elasticsearch集群分别创建于旧网络架构下和新网络架构下。	否	
将ECS上自建的Elasticsearch集群中的数据迁移至阿里云Elasticsearch集群中	阿里云Elasticsearch集群创建于旧网络架构下。	是	请参见通过reindex将自建Elasticsearch数据迁移至阿里云。
	阿里云Elasticsearch集群创建于新网络架构下。	是	借助PrivateLink，打通ECS上自建Elasticsearch集群所处的网络与阿里云服务账号的网络，再使用终端节点域名进行reindex。详情请参见通过实例私网打通将自建Elasticsearch数据迁移至阿里云。 **说明** PrivateLink仅支持部分地域私网连接，详情请参见支持私网连接的地域和可用区。如果您的集群可用区不满足此条件，则不支持reindex功能。

注意事项

- 2020年10月，阿里云Elasticsearch对网络架构进行了调整。2020年10月之前为旧网络架构，2020年10月及之后为新网络架构。新网络架构下的实例不支持与旧网络架构下的实例进行跨集群reindex、跨集群搜索、跨集群复制等实例互通操作。如果需要进行互通，需要确保实例创建在同一网络架构下。对于华北3（张家口）和海外地域，由于网络架构调整时间不确定，因此需要提交工单，联系阿里云Elasticsearch技

术支持，校验网络是否可以互通。

- 新网络架构下，阿里云Elasticsearch实例部署在阿里云服务账号下的VPC中，不支持访问其他网络环境下的资源；旧网络架构下，阿里云Elasticsearch部署在用户VPC中，网络访问不受影响。
- 为保证数据迁移前后一致，建议上游业务停止自建Elasticsearch集群的数据写入更新操作，确保读操作正常进行。迁移完成后，直接切换到阿里云Elasticsearch集群进行读写操作。如果不停止写操作，建议通过脚本设置循环任务减少停写服务时间，具体请参见步骤四：迁移数据中的*数据量大*、*无删除操作*、*有更新时间*章节。
- 当使用域名访问自建Elasticsearch或阿里云Elasticsearch集群时，不允许通过 `http://host:port/path` 这种带path的形式访问。

操作流程

1. 步骤一：获取终端域名（可选）
2. 步骤二：创建目标端索引
3. 步骤三：配置reindex白名单
4. 步骤四：迁移数据

步骤一：获取终端域名（可选）

如果您创建的阿里云Elasticsearch处于新网络架构下（2020年10月及之后创建的实例属于新网络架构），需要借助PrivateLink，打通ECS上自建的Elasticsearch集群所处的网络与阿里云服务账号的网络，获取终端域名，为后续配置做准备。具体操作如下：

1. 创建与阿里云Elasticsearch实例处于同一VPC下，且支持PrivateLink功能的负载均衡实例，详情请参见步骤一：创建支持PrivateLink功能的负载均衡实例。
2. 配置负载均衡实例，指定所有自建Elasticsearch集群中的ECS节点为后端服务器，并监听9200端口，详情请参见步骤二：配置负载均衡实例。
3. 创建终端节点服务，详情请参见步骤三：创建终端节点服务。
4. 获取终端节点域名，详情请参见（可选）步骤五：查看终端节点域名。

   请先记录获取到的节点域名，后续需要在步骤三：配置reindex白名单中使用。

步骤二：创建目标端索引

参考自建Elasticsearch集群中需要迁移的索引配置，提前在阿里云Elasticsearch集群中创建索引。或者为阿里云Elasticsearch集群开启自动创建索引功能（不建议）。

以Python为例，使用如下脚本，在阿里云Elasticsearch集群中批量创建自建Elasticsearch集群中需要迁移的索引。默认新创建的索引副本数为0。

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
# 文件名：indiceCreate.py
import sys
import base64
import time
import httplib
import json
## 自建Elasticsearch集群host。
oldClusterHost = "old-cluster.com"
## 自建Elasticsearch集群用户名，可为空。
oldClusterUserName = "old-username"
## 自建Elasticsearch集群密码，可为空。
```

```
oldClusterPassword = "old-password"
## 阿里云Elasticsearch集群host，可在阿里云Elasticsearch实例的基本信息页面获取。
newClusterHost = "new-cluster.com"
## 阿里云Elasticsearc集群用户名。
newClusterUser = "elastic"
## 阿里云Elasticsearc集群密码。
newClusterPassword = "new-password"
DEFAULT_REPLICAS = 0
def httpRequest(method, host, endpoint, params="", username="", password=""):
    conn = httplib.HTTPConnection(host)
    headers = {}
    if (username != "") :
        'Hello {name}, your age is {age} !'.format(name = 'Tom', age = '20')
        base64string = base64.encodestring('{username}:{password}'.format(username = usernam
e, password = password)).replace('\n', '')
        headers["Authorization"] = "Basic %s" % base64string;
    if "GET" == method:
        headers["Content-Type"] = "application/x-www-form-urlencoded"
        conn.request(method=method, url=endpoint, headers=headers)
    else :
        headers["Content-Type"] = "application/json"
        conn.request(method=method, url=endpoint, body=params, headers=headers)
    response = conn.getresponse()
    res = response.read()
    return res
def httpGet(host, endpoint, username="", password=""):
    return httpRequest("GET", host, endpoint, "", username, password)
def httpPost(host, endpoint, params, username="", password=""):
    return httpRequest("POST", host, endpoint, params, username, password)
def httpPut(host, endpoint, params, username="", password=""):
    return httpRequest("PUT", host, endpoint, params, username, password)
def getIndices(host, username="", password=""):
    endpoint = "/_cat/indices"
    indicesResult = httpGet(oldClusterHost, endpoint, oldClusterUserName, oldClusterPassword
)
    indicesList = indicesResult.split("\n")
    indexList = []
    for indices in indicesList:
        if (indices.find("open") > 0):
            indexList.append(indices.split()[2])
    return indexList
def getSettings(index, host, username="", password=""):
    endpoint = "/" + index + "/_settings"
    indexSettings = httpGet(host, endpoint, username, password)
    print index + "  原始settings如下：\n" + indexSettings
    settingsDict = json.loads(indexSettings)
    ## 分片数默认和自建Elasticsearch集群索引保持一致。
    number_of_shards = settingsDict[index]["settings"]["index"]["number_of_shards"]
    ## 副本数默认为0。
    number_of_replicas = DEFAULT_REPLICAS
    newSetting = "\"settings\": {\"number_of_shards\": %s, \"number_of_replicas\": %s}" % (n
umber_of_shards, number_of_replicas)
    return newSetting
def getMapping(index, host, username="", password=""):
    endpoint = "/" + index + "/_mapping"
```

```
    indexMapping = httpGet(host, endpoint, username, password)
    print index + " 原始mapping如下：\n" + indexMapping
    mappingDict = json.loads(indexMapping)
    mappings = json.dumps(mappingDict[index]["mappings"])
    newMapping = "\"mappings\" : " + mappings
    return newMapping
def createIndexStatement(oldIndexName):
    settingStr = getSettings(oldIndexName, oldClusterHost, oldClusterUserName, oldClusterPas
sword)
    mappingStr = getMapping(oldIndexName, oldClusterHost, oldClusterUserName, oldClusterPass
word)
    createstatement = "{\n" + str(settingStr) + ",\n" + str(mappingStr) + "\n}"
    return createstatement
def createIndex(oldIndexName, newIndexName=""):
    if (newIndexName == "") :
        newIndexName = oldIndexName
    createstatement = createIndexStatement(oldIndexName)
    print "新索引 " + newIndexName + " 的setting和mapping如下：\n" + createstatement
    endpoint = "/" + newIndexName
    createResult = httpPut(newClusterHost, endpoint, createstatement, newClusterUser, newClu
sterPassword)
    print "新索引 " + newIndexName + " 创建结果：" + createResult
## main
indexList = getIndices(oldClusterHost, oldClusterUserName, oldClusterPassword)
systemIndex = []
for index in indexList:
    if (index.startswith(".")):
        systemIndex.append(index)
    else :
        createIndex(index, index)
if (len(systemIndex) > 0) :
    for index in systemIndex:
        print index + " 或许是系统索引，不会重新创建，如有需要，请单独处理～"
```

步骤三：配置reindex白名单

1. 登录阿里云Elasticsearch控制台。
2. 在左侧导航栏，单击**Elasticsearch实例**。
3. 进入目标实例。
   i. 在顶部菜单栏处，选择资源组和地域。
   ii. 在左侧导航栏，单击**Elasticsearch实例**，然后在**Elasticsearch实例**中单击目标实例ID。
4. 在左侧导航栏，选择**配置与管理 > ES集群配置**。
5. 在**YML文件配置**区域，单击右侧的**修改配置**。
6. 在**YML文件配置**面板，修改**其他Configure配置**，配置reindex白名单。

   配置示例如下。

```
reindex.remote.whitelist: ["10.0.xx.xx:9200","10.0.xx.xx:9200","10.0.xx.xx:9200","10.15.
xx.xx:9200","10.15.xx.xx:9200","10.15.xx.xx:9200"]
```

在配置reindex白名单时，需要通过reindex.remote.whitelist参数，设置自建Elasticsearch集群的访问地址，将其添加到阿里云Elasticsearch集群的远程访问白名单中。阿里云Elasticsearch集群的网络架构不同，配置规则也不同，具体如下：

- 旧网络架构下：需要配置为host和port的组合，并使用逗号分隔多个主机配置。例如：otherhost:9200,another:9200,127.0.10.**:9200,localhost:**，不识别协议信息。
- 新网络架构下：需要配置为实例对应的终端节点域名和port的组合。例如：ep-bp1hfkx7coy8lvu4****-cn-hangzhou-i.epsrv-bp1zczi0fgoc5qtv****.cn-hangzhou.privatelink.aliyuncs.com:9200。终端节点域名可在步骤一：获取终端域名（可选）中获取，更多详细信息请参见（可选）步骤五：查看终端节点域名。

> **说明** 更多参数说明请参见配置YML参数。

7. 选中**该操作会重启实例，请确认后操作**，单击**确定**。

   确定后，Elasticsearch实例会重启。重启过程中，可在任务列表查看进度。重启成功后，即可完成配置。

步骤四：迁移数据

本文以旧网络架构下的实例为例，提供了以下三种数据迁移的方式，请根据迁移的数据量大小以及实际业务情况，选择合适的方式迁移数据。

数据量小

使用如下脚本。

```
#!/bin/bash
# file:reindex.sh
indexName="您的索引名"
newClusterUser="阿里云Elasticsearch集群用户名"
newClusterPass="阿里云Elasticsearch集群密码"
newClusterHost="阿里云Elasticsearch集群host"
oldClusterUser="自建Elasticsearch集群用户名"
oldClusterPass="自建Elasticsearch集群密码"
# 自建Elasticsearch集群host必须是[scheme]://[host]:[port]，例如http://10.37.*.*:9200。
oldClusterHost="自建Elasticsearch集群host"
curl -u ${newClusterUser}:${newClusterPass} -XPOST "http://${newClusterHost}/_reindex?pretty
" -H "Content-Type: application/json" -d'{
    "source": {
        "remote": {
            "host": "'${oldClusterHost}'",
            "username": "'${oldClusterUser}'",
            "password": "'${oldClusterPass}'"
        },
        "index": "'${indexName}'",
        "query": {
            "match_all": {}
        }
    },
    "dest": {
       "index": "'${indexName}'"
    }
}'
```

数据量大、无删除操作、有更新时间

数据量较大且无删除操作时，可以使用滚动迁移的方式，减少停止写服务的时间。滚动迁移需要有一个类似于更新时间的字段代表新数据的写时序。在数据迁移完成后，先停止业务写操作，待reindex使用最近一次更新时间快速执行一次更新后，将读写业务切换到阿里云Elasticsearch集群。

```
#!/bin/bash
# file: circleReindex.sh
# CONTROLLING STARTUP:
# 这是通过reindex操作远程重建索引的脚本，要求：
# 1. 阿里云Elasticsearch集群已经创建完索引，或者支持自动创建和动态映射。
# 2. 阿里云Elasticsearch集群必须在yml里配置IP白名单，例如reindex.remote.whitelist: 172.16.**.**:
9200。
# 3. host必须是[scheme]://[host]:[port]。
USAGE="Usage: sh circleReindex.sh <count>
       count: 执行次数，多次（负数为循环）增量执行或者单次执行
Example:
        sh circleReindex.sh 1
        sh circleReindex.sh 5
        sh circleReindex.sh -1"
indexName="您的索引名"
newClusterUser="阿里云Elasticsearch集群用户名"
newClusterPass="阿里云Elasticsearch集群密码"
oldClusterUser="自建Elasticsearch集群用户名"
oldClusterPass="自建Elasticsearch集群密码"
## http://myescluster.com
```

```
newClusterHost="阿里云Elasticsearch集群host"
# 自建Elasticsearch集群host必须是[scheme]://[host]:[port]，例如http://10.37.*.*:9200。
oldClusterHost="自建Elasticsearch集群host"
timeField="更新时间字段"
reindexTimes=0
lastTimestamp=0
curTimestamp=`date +%s`
hasError=false
function reIndexOP() {
    reindexTimes=$[${reindexTimes} + 1]
    curTimestamp=`date +%s`
    ret=`curl -u ${newClusterUser}:${newClusterPass} -XPOST "${newClusterHost}/_reindex?pret
ty" -H "Content-Type: application/json" -d '{
        "source": {
            "remote": {
                "host": "'${oldClusterHost}'",
                "username": "'${oldClusterUser}'",
                "password": "'${oldClusterPass}'"
            },
            "index": "'${indexName}'",
            "query": {
                "range" : {
                    "'${timeField}'" : {
                        "gte" : '${lastTimestamp}',
                        "lt" : '${curTimestamp}'
                    }
                }
            }
        },
        "dest": {
            "index": "'${indexName}'"
        }
    }'`
    lastTimestamp=${curTimestamp}
    echo "第${reindexTimes}次reIndex，本次更新截止时间 ${lastTimestamp} 结果：${ret}"
    if [[ ${ret} == *error* ]]; then
        hasError=true
        echo "本次执行异常，中断后续执行操作～～，请检查"
    fi
}
function start() {
    ## 负数就不停循环执行
    if [[ $1 -lt 0 ]]; then
        while :
        do
            reIndexOP
        done
    elif [[ $1 -gt 0 ]]; then
        k=0
        while [[ k -lt $1 ]] && [[ ${hasError} == false ]]; do
            reIndexOP
            let ++k
        done
    fi
```

```
}
## main
if [ $# -lt 1 ]; then
    echo "$USAGE"
    exit 1
fi
echo "开始执行索引 ${indexName} 的 ReIndex操作"
start $1
echo "总共执行 ${reindexTimes} 次 reIndex 操作"
```

数据量大、无删除操作、无更新时间

当数据量较大，且索引的Mapping中没有定义更新时间的字段时，需要由上游业务修改代码添加更新时间的字段。添加完成后可以先将历史数据迁移完，然后再使用上述第二种方案操作。

```
#!/bin/bash
# file:miss.sh
indexName="您的索引名"
newClusterUser="阿里云Elasticsearch集群用户名"
newClusterPass="阿里云Elasticsearch集群密码"
newClusterHost="阿里云Elasticsearch集群host"
oldClusterUser="自建Elasticsearch集群用户名"
oldClusterPass="自建Elasticsearch集群密码"
# 自建Elasticsearch集群host必须是[scheme]://[host]:[port]，例如http://10.37.*.*:9200
oldClusterHost="自建Elasticsearch集群host"
timeField="updatetime"
curl -u ${newClusterUser}:${newClusterPass} -XPOST "http://${newClusterHost}/_reindex?pretty
" -H "Content-Type: application/json" -d '{
    "source": {
        "remote": {
            "host": "'${oldClusterHost}'",
            "username": "'${oldClusterUser}'",
            "password": "'${oldClusterPass}'"
        },
        "index": "'${indexName}'",
        "query": {
            "bool": {
                "must_not": {
                    "exists": {
                        "field": "'${timeField}'"
                    }
                }
            }
        }
    },
    "dest": {
       "index": "'${indexName}'"
    }
}'
```

常见问题

- 问题：执行curl命令时，提示 `{"error":"Content-Type header [application/x-www-form-urlencoded]`

`is not supported","status":406}` 。

解决方法：在curl命令中，添加 `-H "Content-Type: application/json"` 脚本重试。

```
  // 获取自建Elasticsearch集群中所有索引信息，如果没有权限可去掉"-u user:pass"参数，oldClusterHost为自建Elasticsearch集群的host，注意替换。
  curl -u user:pass -XGET http://oldClusterHost/_cat/indices | awk '{print $3}'
  // 参考上面返回的索引列表，获取需要迁移的指定用户索引的setting和mapping，注意替换indexName为要查询的用户索引名。
  curl -u user:pass -XGET http://oldClusterHost/indexName/_settings,_mapping?pretty=true
  // 参考上面获取到的对应索引的_settings和_mapping信息，在阿里云Elasticsearch集群中创建对应索引，索引副本数可以先设置为0，用于加快数据同步速度，数据迁移完成后再重置副本数为1。
  //其中newClusterHost是阿里云Elasticsearch集群的host，testindex是已经创建的索引名，testtype是对应索引的type。
  curl -u user:pass -XPUT http://<newClusterHost>/<testindex> -d '{
    "testindex" : {
        "settings" : {
            "number_of_shards" : "5", //假设自建Elasticsearch集群中对应索引的shard数是5个。
            "number_of_replicas" : "0" //设置索引副本为0。
          }
        },
        "mappings" : { //假设自建Elasticsearch集群中对应索引的mappings配置如下。
            "testtype" : {
                "properties" : {
                    "uid" : {
                        "type" : "long"
                    },
                    "name" : {
                        "type" : "text"
                    },
                    "create_time" : {
                      "type" : "long"
                    }
                }
            }
        }
    }
}'
```

- 问题：单索引数据量比较大，数据同步速度比较慢时，如何处理？

  解决方法：

  - 由于reindex功能的底层实现原理是通过scroll方式实现的，所以您可以适当调大scroll size的大小或配置scroll slice，借助scroll并行化机制提升效率。详情请参见reindex API。
  - 如果源端数据量较大，建议采用OSS快照方式。详情请参见通过OSS将自建Elasticsearch数据迁移至阿里云。

- 如果单索引数据量比较大，可以在迁移前将目标索引的副本数设置为0，刷新时间设置为-1，以加快数据同步速度。待数据迁移完成后，再修改回来。

```
// 迁移索引数据前可以先将索引副本数设为0，不刷新，用于加快数据迁移速度。
curl -u user:password -XPUT 'http://<host:port>/indexName/_settings' -d' {
        "number_of_replicas" : 0,
        "refresh_interval" : "-1"
}'
// 索引数据迁移完成后，可以重置索引副本数为1，刷新时间1s（1s是默认值）。
curl -u user:password -XPUT 'http://<host:port>/indexName/_settings' -d' {
        "number_of_replicas" : 1,
        "refresh_interval" : "1s"
}'
```

2.3.5. 通过实例私网打通将自建Elasticsearch数据迁移至阿里云

本文主要介绍当您的阿里云Elasticsearch创建于新网络架构下时，如何通过实例私网连接打通网络后，使用reindex方式将ECS上自建的Elasticsearch中的数据迁移至阿里云Elasticsearch中，包括创建索引和迁移数据。

前提条件

- 自建Elasticsearch需要满足以下条件：
  -
  -
  -
  - 准备自建Elasticsearch索引数据，本文以下图中的source索引作为需要迁移的索引为例。

```
[root@elasticsearch1 ~]# curl -XGET  http://172.16.   :9200/_cat/indices?v
health status index  uuid                   pri rep docs.count docs.deleted store.size pri.store.size
green  open   source lGFcaUIgT1-Nsj9b_EezAQ   1   1          6            0     28.2kb         19.1kb
green  open   dest   Kn3Tu9TmT62J4ouMHi_37w   1   1          6            0     23.3kb          9.1kb
[root@elasticsearch1 ~]#
```

- 阿里云Elasticsearch需要满足以下条件：
  - 提前开启自动创建索引功能或者在阿里云Elasticsearch上创建好索引 `mappings` 、 `settings` 。
  - 未做白名单限制。

使用限制

自2020年10月起，由于网络架构的调整，导致部分通过reindex方式跨集群迁移数据的场景受到了限制。您可以参见下表，查看您的业务场景是否支持reindex功能，并参考解决方案进行处理。

使用场景	网络状态（2020年10月之前属于旧网络架构，2020年10月及之后属于新网络架构）	是否支持reindex功能	解决方案
	两个Elasticsearch集群均创建于旧网络架构下。	是	请参见通过reindex迁移数据。

使用场景	网络状态（2020年10月之前属于旧网络架构，2020年10月及之后属于新网络架构）	是否支持reindex功能	解决方案
通过reindex方式迁移阿里云Elasticsearch集群间的数据	两个Elasticsearch集群均创建于新网络架构下。	否	不支持reindex方式，建议使用OSS或Logstash方式。具体操作请参见通过OSS将自建Elasticsearch数据迁移至阿里云和通过阿里云Logstash将自建Elasticsearch数据迁移至阿里云。
	两个Elasticsearch集群分别创建于旧网络架构下和新网络架构下。	否	
将ECS上自建的Elasticsearch集群中的数据迁移至阿里云Elasticsearch集群中	阿里云Elasticsearch集群创建于旧网络架构下。	是	请参见通过reindex将自建Elasticsearch数据迁移至阿里云。
	阿里云Elasticsearch集群创建于新网络架构下。	是	借助PrivateLink，打通ECS上自建Elasticsearch集群所处的网络与阿里云服务账号的网络，再使用终端节点域名进行reindex。详情请参见通过实例私网打通将自建Elasticsearch数据迁移至阿里云。 **说明** PrivateLink仅支持部分地域私网连接，详情请参见支持私网连接的地域和可用区。如果您的集群可用区不满足此条件，则不支持reindex功能。

操作流程

1. 步骤一：配置支持PrivateLink功能的负载均衡实例

   目前，仅支持PrivateLink功能的负载均衡实例作为终端节点服务的服务资源。通过PrivateLink实现在VPC间私网访问服务前，您需要创建支持PrivateLink功能的负载均衡实例，并配置相关监听信息。

2. 步骤二：创建终端节点服务

   终端节点服务是可以被其他VPC通过创建终端节点建立私网连接的服务，待负载均衡实例配置完成后，您需要创建终端节点服务。

3. 步骤三：配置阿里云Elasticsearch私网互通

   在Elasticsearch控制台关联阿里云Elasticsearch实例与步骤二中创建的终端节点服务。

4. 步骤四：获取终端节点域名

   目标实例与终端节点服务关联成功后，即可获取终端节点域名，用于配置reindex白名单。

5. 步骤五：配置reindex白名单

   在Elasticsearch控制台将步骤四中获取的域名配置到目标实例的reindex白名单中，进行授权。

6. 步骤六：数据迁移

   完成以上步骤后，即完成源实例与目标实例间的网络互通，可以进行数据迁移。

步骤一：配置支持PrivateLink功能的负载均衡实例

1. 创建负载均衡实例。

确保负载均衡实例的地域和后端添加的云服务器ECS的地域相同。详情请参见创建支持PrivateLink功能的负载均衡实例。

> **说明** PrivateLink仅支持部分地域私网连接，详情请参见支持私网连接的地域和可用区。如果您的集群可用区不满足此条件，则不支持reindex功能。

2. 配置**协议&监听**。选择负载均衡协议为**TCP**，并配置端口号为**9200**。

   详情请参见配置协议&监听。

3. 配置**后端服务器**，添加自建ES所在的ECS服务器，并配置端口号为**9200**。

   详情请参见配置后端服务器。

4.

5. 配置完成后，单击**提交**，根据页面提示返回**实例管理**页面，查看后端ECS实例的健康检查状态。

   当后端ECS实例的健康检查状态为正常时，表示后端ECS实例可以正常处理负载均衡转发的请求了。

6. 在**负载均衡业务配置向导**弹框中，单击**知道了**，返回**实例管理**页面。

   当后端ECS实例的健康检查状态为正常时，表示后端ECS实例可以正常处理负载均衡转发的请求了。

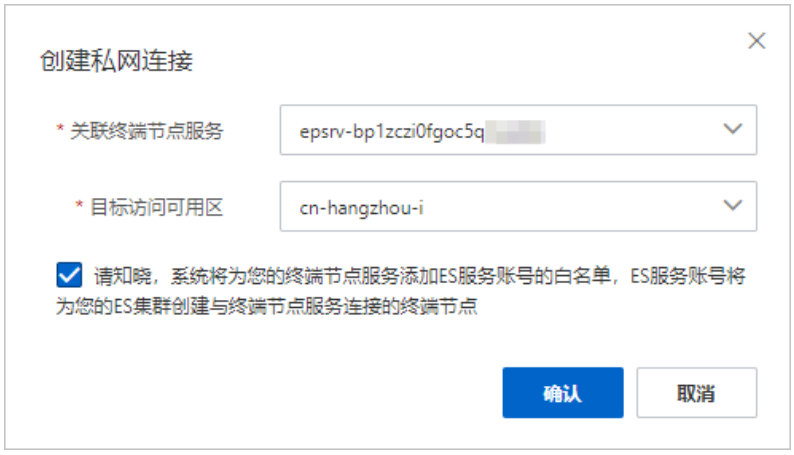
步骤二：创建终端节点服务

1. 登录专有网络管理控制台。
2.
3.
4.
5. 在**创建终端节点服务**页面，按需配置终端节点服务。

6. 单击**确定创建**。

步骤三：配置阿里云Elasticsearch私网互通

1. 登录阿里云Elasticsearch控制台。
2. 在左侧导航栏，单击**Elasticsearch实例**。
3. 进入目标实例。
   i. 在顶部菜单栏处，选择资源组和地域。
   ii. 在左侧导航栏，单击**Elasticsearch实例**，然后在**Elasticsearch实例**中单击目标实例ID。
4. 在左侧导航栏，选择**配置与管理 > 安全配置**。
5. 在**集群网络设置**区域，单击**配置实例私网连接**右侧的**修改**。
6. 在**配置实例私网连接**面板，单击**+添加私网连接**，选择步骤二中创建的终端节点服务和目标访问可用区，并选中系统提示信息。



7. 单击**确认**，终端节点服务主动连接终端节点，连接成功后会显示**已连接**。

步骤四：获取终端节点域名

执行完以上步骤后，需要获取终端节点域名用于配置reindex白名单。

1. 在**配置实例私网连接**面板中，单击目标**终端节点ID**。

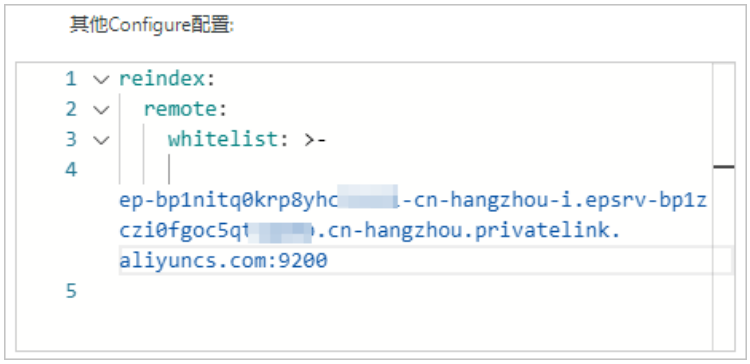
2. 在**终端节点连接**页签，单击目标终端节点ID前的+图标，即可查看终端节点对应的域名。

步骤五：配置reindex白名单

> **注意** 该操作会触发集群重启，建议在业务低峰期进行。

1. 登录阿里云Elasticsearch控制台。
2. 在左侧导航栏，单击**Elasticsearch实例**。
3. 进入目标实例。
   i. 在顶部菜单栏处，选择资源组和地域。
   ii. 在左侧导航栏，单击**Elasticsearch实例**，然后在**Elasticsearch实例**中单击目标实例ID。
4. 在左侧导航栏，选择**配置与管理 > ES集群配置**。
5. 在**YML文件配置**区域，单击右侧的**修改配置**。
6. 在**YML文件配置**面板，修改**其他Configure配置**，将从步骤四中获取的域名配置至此处。
   代码示例：

```
reindex.remote.whitelist: 'ep-bp1nitq0krp8yhcf****-cn-hangzhou-i.epsrv-bp1zczi0fgoc5qtv*
***.cn-hangzhou.privatelink.aliyuncs.com:9200'
```



7.

步骤六：数据迁移

1. 在Kibana控制台的**Dev Tools**中执行如下命令，进行数据迁移。

   > **说明** 登录Kibana控制台的具体步骤请参见登录Kibana控制台。

   ```
   POST /_reindex?pretty
   {
     "source": {
       "remote": {
         "host": "http://ep-bp1nitq0krp8yhcf****-cn-hangzhou-i.epsrv-bp1zczi0fgoc5qtv****.cn-hangzhou.privatelink.aliyuncs.com:9200",
         "username": "elastic",
         "password": "Elastic@123***"
       },
       "index": "source",
       "size": 5000
     },
     "dest": {
       "index": "dest"
     }
   }
   ```

   更多信息说明请参见reindex api。

2. （可选）在数据迁移过程中，如果您需要获取所有正在运行的reindex请求状态，请执行以下命令。

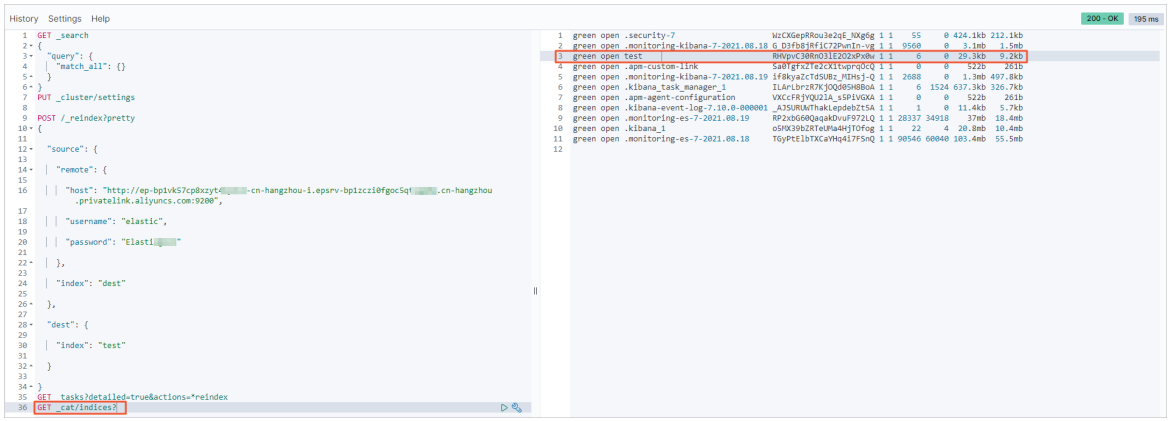
   ```
   GET _tasks?detailed=true&actions=*reindex
   ```

3. 查看数据迁移结果。

   待数据迁移完成后，您可以通过执行以下命令查看数据迁移结果。

   ```
   GET _cat/indices?
   ```

   如果目标端索引test中显示健康状态正常并且数据大小正常，则证明数据迁移成功。



常见问题

问题：单索引数据量比较大，数据同步速度比较慢时，如何处理？

解决方法：

- 由于reindex功能的底层实现原理是通过scroll方式实现的，所以您可以适当调大scroll size的大小或配置scroll slice，借助scroll并行化机制提升效率。详情请参见reindex api。
- 如果源端数据量较大，建议采用OSS快照方式。详情请参见通过OSS将自建Elasticsearch数据迁移至阿里云。
- 如果单索引数据量比较大，可以在迁移前将目标索引的副本数设置为0，刷新时间设置为-1，以加快数据同步速度。待数据迁移完成后，再修改回来。

```
// 迁移索引数据前可以先将索引副本数设为0，不刷新，用于加快数据迁移速度。
curl -u user:password -XPUT 'http://<host:port>/indexName/_settings' -d' {
        "number_of_replicas" : 0,
        "refresh_interval" : "-1"
}'
// 索引数据迁移完成后，可以重置索引副本数为1，刷新时间1s（1s是默认值）。
curl -u user:password -XPUT 'http://<host:port>/indexName/_settings' -d' {
        "number_of_replicas" : 1,
        "refresh_interval" : "1s"
}'
```

2.4. 第三方Elasticsearch数据迁移

2.4.1. 腾讯云Elasticsearch数据迁移至阿里云

本文介绍通过阿里云Logstash，将索引数据从腾讯云Elasticsearch迁移至阿里云Elasticsearch中。

场景说明

本文的操作流程同样适用于跨账号、跨地域的阿里云Elasticsearch之间迁移数据的场景。在配置时，将input中的腾讯云Elasticsearch的配置替换为源阿里云Elasticsearch的配置即可。

> **说明** 本文介绍的是如何将腾讯云Elasticsearch数据迁移至阿里云，在迁移过程中，Logstash需要连接公网才能与腾讯云Elasticsearch互通，因此会涉及到Logstash连接公网的相关操作。对于跨账号、跨地域的阿里云Elasticsearch之间迁移数据的场景，同样需要配置Logstash与公网连通，因此在此类场景下，您可以参考本文档的流程。

操作流程

1. 准备工作

   创建阿里云Elasticsearch实例、Logstash实例和腾讯云Elasticsearch实例、开启阿里云Elasticsearch实例的自动创建索引功能、搭建腾讯云代理服务器。

2. 步骤一：选取迁移方案

   选择兼容的腾讯云Elasticsearch实例、阿里云Elasticsearch实例和Logstash实例进行迁移。

3. 步骤二：创建并配置Logstash管道

   通过配置文件管理方式创建并配置Logstash管道，配置时设置input为腾讯云Elasticsearch，output为阿里云Elasticsearch。

   > **说明** 您也可以使用自建Logstash，需要在服务器上安装JDK（1.8及以上版本）以及相应版本的Logstash，然后在Logstash的conf下配置YML文件，并启动服务。

4. 步骤三：验证结果

   在Kibana中使用 `GET /_cat/indices?v` ，查看索引是否迁移成功。

准备工作

1. 创建阿里云Elasticsearch实例，并开启自动创建索引功能。

   具体操作步骤请参见创建阿里云Elasticsearch实例和快速访问与配置。

2. 创建阿里云Logstash实例。

   具体操作步骤请参见创建阿里云Logstash实例。

   由于阿里云Logstash实例部署在专有网络VPC（Virtual Private Cloud）下，但在迁移过程中，Logstash需要连接公网才能与腾讯云Elasticsearch互通，因此需要通过配置NAT网关实现与公网连通，详情请参见配置NAT公网数据传输。

   > **说明** 对于自建的Logstash，需要购买与阿里云Elasticsearch在同一VPC下的ECS实例（已符合条件的ECS不需要重复购买，需要绑定弹性公网IP）。

3. 创建腾讯云Elasticsearch实例，并搭建腾讯云代理服务器。

   由于腾讯云Elasticsearch不支持通过公网直接访问，因此需要准备cvm服务器搭建nginx代理，详情请参见通过外网访问腾讯云Elasticsearch集群。

步骤一：选取迁移方案

由于腾讯云Elasticsearch版本与阿里云Elasticsearch版本不一致，因此需要先选择兼容的版本进行迁移，本文支持的版本方案如下（其他方案不保证兼容）：

腾讯云Elasticsearch	阿里云Elasticsearch

腾讯云Elasticsearch	阿里云Elasticsearch
腾讯云Elasticsearch 5.6.4	迁移至阿里云Elasticsearch的版本要求：自建Logstash 5.6.x、阿里云Elasticsearch 5.5.6。 **说明** • 因为阿里云Elasticsearch不提供Logstash 5.6.x，所以您需要通过自建Logstash 5.6.x作为数据传输管道，实现数据迁移。 • 对于自建的Logstash，需要购买与阿里云Elasticsearch在同一VPC下的ECS实例（已符合条件的ECS不需要重复购买，需要绑定弹性公网IP）。
腾讯云Elasticsearch 6.4.3	迁移至阿里云Elasticsearch的版本有如下两种选择： • 自建Logstash 6.7.0、阿里云Elasticsearch 6.7.0 • 阿里云Logstash 6.7.0、阿里云Elasticsearch 6.7.0 **说明** • 考虑到版本兼容性，建议您使用阿里云Logstash 6.7.0或者自建Logstash 6.7.0作为数据传输管道，实现数据迁移。 • 对于自建的Logstash，需要购买与阿里云Elasticsearch在同一VPC下的ECS实例（已符合条件的ECS不需要重复购买，需要绑定弹性公网IP）。 • 本文以阿里云Logstash 6.7.0、阿里云Elasticsearch 6.7.0为例为您展开介绍。

注意

- 建议您在大版本内进行数据迁移。关于Logstash版本选取详情，请参见Support Matrix。
- 如果您使用的是其他方案，可参见产品兼容性判断是否存在兼容性问题。如果存在，可升级实例版本或新购实例。

步骤二：创建并配置Logstash管道

1. 登录阿里云Elasticsearch控制台。
2. 进入目标实例。
   i. 在顶部菜单栏处，选择地域。
   ii. 在左侧导航栏，单击**Logstash实例**，然后在**Logstash实例**中单击目标实例ID。
3. 在左侧导航栏，单击**管道管理**。
4. 单击**创建管道**。
5. 在**创建管道任务**页面，输入**管道ID**并配置管道。

   本文使用的管道配置如下。

```
input {
     elasticsearch {
       hosts => "http://xxxxxxxxx:9200"
       user  => "elastic"
       index => "*"
       password => "xxxxxx"
       docinfo => true
     }
   }
output {
      elasticsearch {
        hosts => "http://xxxxxxx.elasticsearch.aliyuncs.com:9200"
        user => "elastic"
        password => "xxxxxx"
        index => "%{[@metadata][_index]}"
        document_type => "%{[@metadata][_type]}"
        document_id => "%{[@metadata][_id]}"
  }
}
```

参数	说明
hosts	Elasticsearch服务的访问地址。input中为 `http://<腾讯云代理服务的公网地址>:<端口>` ；output中为 `http://<阿里云Elasticsearch实例ID>.elasticsearch.aliyuncs.com:9200` 。
user	访问Elasticsearch服务的用户名，默认为elastic。
password	对应用户的密码。对于阿里云Elasticsearch，elastic用户的密码在创建实例时设定，如果忘记可进行重置，重置密码的注意事项和操作步骤请参见重置实例访问密码。
index	input中为待迁移的索引名，设置为 *表示同步全部索引；output中为迁移后的索引名，设置为 *%{[@metadata][_index]}*表示匹配元数据中的index，即迁移后的索引名与源索引名相同。
document_type	迁移后索引的类型。设置为 *%{[@metadata][_type]}*表示匹配元数据中的type，即迁移后索引的类型与源索引的类型相同。
document_id	迁移后文档的ID。设置为 *%{[@metadata][_id]}*表示匹配元数据中的id，即迁移后文档的ID与源文档的ID相同。
docinfo	设置为*true*，Logstash将会提取Elasticsearch文档的元信息，例如index、type和id。

更多Config配置详情请参见Logstash配置文件说明。

以上配置将源Elasticsearch集群的所有索引同步到目标集群中，您也可以设置只同步指定的索引，input配置如下。

```
input {
     elasticsearch {
       hosts => "http://xxxxxxxx:9200"
       user  => "elastic"
       index => "alicloudtest"
       password => "xxxxxxx"
       query => '{ "query":{ "query_string": { "query": "*" } } }'
       docinfo => true
     }
```

Elasticsearch input插件可以根据配置的查询语句，从Elasticsearch集群读取文档数据，适用于批量导入测试日志等操作。默认读取完数据后，同步动作会自动关闭。如果您想实现定时触发数据同步，可以通过cron语法配合schedule参数实现，详情请参见Logstash官网Scheduling介绍。

```
schedule => "* * * * *"
```

以上示例会每分钟触发一次数据同步。如果没有指定schedule参数，那么同步仅执行一次。

6. 单击**下一步**，配置管道参数。

参数	说明
管道工作线程	并行执行管道的Filter和Output的工作线程数量。当事件出现积压或CPU未饱和时，请考虑增大线程数，更好地使用CPU处理能力。默认值：实例的CPU核数。
管道批大小	单个工作线程在尝试执行Filter和Output前，可以从Input收集的最大事件数目。较大的管道批大小可能会带来较大的内存开销。您可以设置LS_HEAP_SIZE变量，来增大JVM堆大小，从而有效使用该值。默认值：125。
管道批延迟	创建管道事件批时，将过小的批分派给管道工作线程之前，要等候每个事件的时长，单位为毫秒。默认值：50ms。
队列类型	用于事件缓冲的内部排队模型。可选值： ○ **MEMORY**：默认值。基于内存的传统队列。 ○ **PERSISTED**：基于磁盘的ACKed队列（持久队列）。
队列最大字节数	请确保该值小于您的磁盘总容量。默认值：1024 MB。
队列检查点写入数	启用持久性队列时，在强制执行检查点之前已写入事件的最大数目。设置为0，表示无限制。默认值：1024。

> 警告 配置完成后，需要保存并部署才能生效。保存并部署操作会触发实例重启，请在不影响业务的前提下，继续执行以下步骤。

7. 单击**保存**或者**保存并部署**。
   - **保存**：将管道信息保存在Logstash里并触发实例变更，配置不会生效。保存后，系统会返回**管道管理**页面。可在**管道列表**区域，单击**操作**列下的**立即部署**，触发实例重启，使配置生效。
   - **保存并部署**：保存并且部署后，会触发实例重启，使配置生效。

步骤三：验证结果

1. 登录目标阿里云Elasticsearch实例的Kibana控制台。

   具体步骤请参见登录Kibana控制台。
2. 在左侧导航栏，单击**Dev Tools**（开发工具）。
3. 在**Console**中执行 `GET /_cat/indices?v` 命令，查询索引是否迁移成功。

常见问题

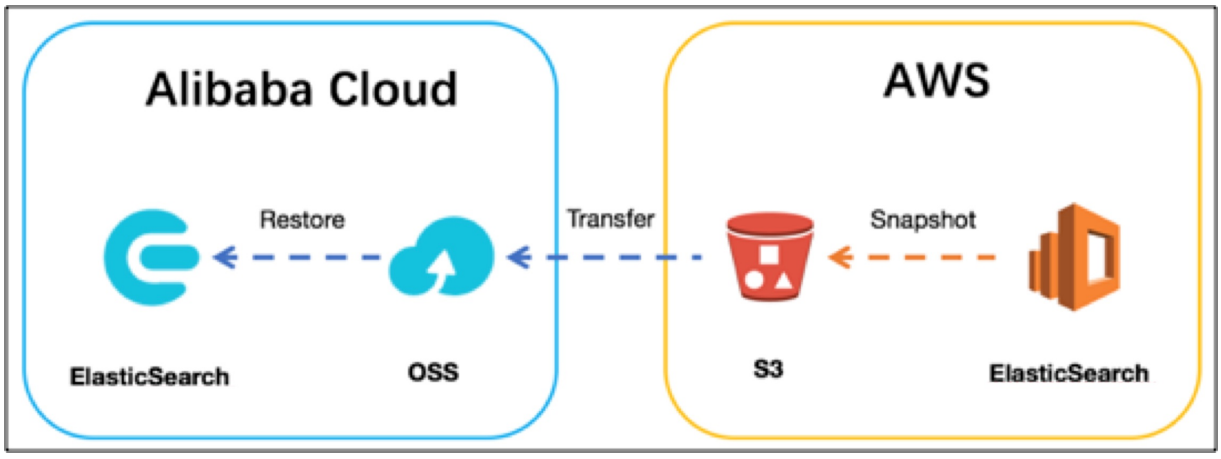
Logstash数据写入问题排查方案

2.4.2. 从AWS迁移Elasticsearch索引至阿里云

本文介绍如何将索引数据从AWS Elasticsearch迁移到阿里云Elasticsearch中。

背景信息

本次Elasticsearch索引迁移方案的参考架构如下。



相关概念

- Elasticsearch：一个分布式的RESTful风格的搜索与分析引擎，适用于各种应用场景。作为Elastic Stack的核心，Elasticsearch可以集中存储您的数据，并对数据进行搜索分析。
- Kibana：您可以使用Kibana对Elasticsearch数据进行可视化搜索分析。
- 亚马逊Elasticsearch服务（简称AWS Elasticsearch）： 一项完全托管的服务，提供了各种易于使用的Elasticsearch API和实时分析功能，还可以实现生产工作负载需要的可用性、可扩展性和安全性。您可以使用Amazon Elasticsearch Service轻松部署、保护、操作和扩展Elasticsearch，以便进行日志分析、全文搜索和应用程序监控等工作。
- 阿里云Elasticsearch服务： 提供基于开源Elasticsearch服务，致力于数据分析、数据搜索等场景服务。在

开源Elasticsearch基础上提供企业级权限管控、安全监控告警、自动报表生成等功能。本文涉及的操作主要在阿里云Elasticsearch服务的中国站上进行。

- 快照和恢复（Snapshot and Restore）：您可以使用快照和恢复功能，在远程存储库（如共享文件系统、S3或HDFS）中，创建各个索引或整个集群的快照。创建后的快照可以被恢复到对应版本的Elasticsearch中。
  - 在5.x中创建的索引快照可以恢复到6.x。
  - 在2.x中创建的索引快照可以恢复到5.x。
  - 在1.x中创建的索引快照可以恢复到2.x。

解决方案概述

您可以通过以下步骤来迁移索引数据：

1. 创建基线索引。
   i. 创建一个快照存储库，并将其与Amazon Simple Storage Service （AWS S3）存储空间相关联。
   ii. 为要迁移的索引创建第一个完整的快照。

   该快照会自动存储在步骤一中创建的AWS S3存储空间中。
   iii. 在阿里云侧创建一个对象存储服务OSS（Object Storage Service）存储空间，并将其注册到阿里云Elasticsearch实例的快照存储库中。
   iv. 使用OSS Import工具将AWS S3存储空间中的数据提取到阿里云OSS存储空间中。
   v. 将此完整快照恢复到阿里云Elasticsearch实例。
2. 定期处理增量快照 。

   重复以上步骤处理增量快照并进行恢复。
3. 确定最终快照，进行服务切换。
   i. 停止可能会修改索引数据的服务。
   ii. 创建AWS Elasticsearch实例的最终增量快照。
   iii. 将最终增量快照迁移至OSS，然后恢复到阿里云Elasticsearch实例中。
   iv. 进行服务切换，在阿里云Elasticsearch实例中，查看迁移成功的数据。

前提条件

您已完成以下操作：

- 创建AWS Elasticsearch实例，版本号为5.5.2，区域为新加坡。

  具体操作步骤请参见创建Amazon Elasticsearch域。
- 创建阿里云Elasticsearch实例，版本号为5.5.3，区域为杭州。

  具体操作步骤请参见创建阿里云Elasticsearch实例。
- 创建OSS Bucket。

  本文创建的Bucket区域为华东1（杭州）、存储类型为标准存储、读写权限为私有，其他参数保持默认，具体操作步骤请参见创建存储空间。
- 准备待迁移的索引，示例索引名称为 `movies` 。

在AWS中创建手动快照的前提条件

AWS Elasticsearch每天会自动为一个域中的主要索引分片创建快照，并将这些快照存储在预配置的AWS S3存储空间中。这些快照会保留14天，您无需额外付费。此外，您还可以使用这些快照来恢复域，但是这些自动快照不能被迁移到新域。如果要迁移，您必须使用存储在自己的存储库（S3存储空间）中的手动快照。手动快照将收取标准S3费用。

您需要使用Amazon Identity and Access Management（IAM）和AWS S3才能手动创建和恢复索引快照。创建快照之前，请确保已满足以下条件。

前提条件	描述
创建AWS S3存储空间	存储AWS Elasticsearch域的手动快照。
创建IAM角色	为AWS Elasticsearch服务授权。在给该角色添加信任关系时，必须在 `Principal` 语句中指定AWS Elasticsearch服务。使用AWS Elasticsearch注册您的快照存储库时，也需要使用该IAM角色。只有具有该角色访问权限的IAM用户才可以注册快照存储库。
创建IAM策略	指定IAM角色可以对S3存储空间执行的操作。该策略必须添加给为AWS Elasticsearch服务授权的IAM角色。您需要在该策略的Resource语句中指定S3存储空间。

- 创建S3存储空间

  创建一个AWS S3存储空间来存储手动快照，并记录其Amazon资源名称（ARN）。该ARN在以下场景中会用到：

  - 用于IAM角色添加IAM策略的Resource语句。
  - 用于注册快照存储库的Python客户端。

  AWS S3存储空间的ARN示例如下。

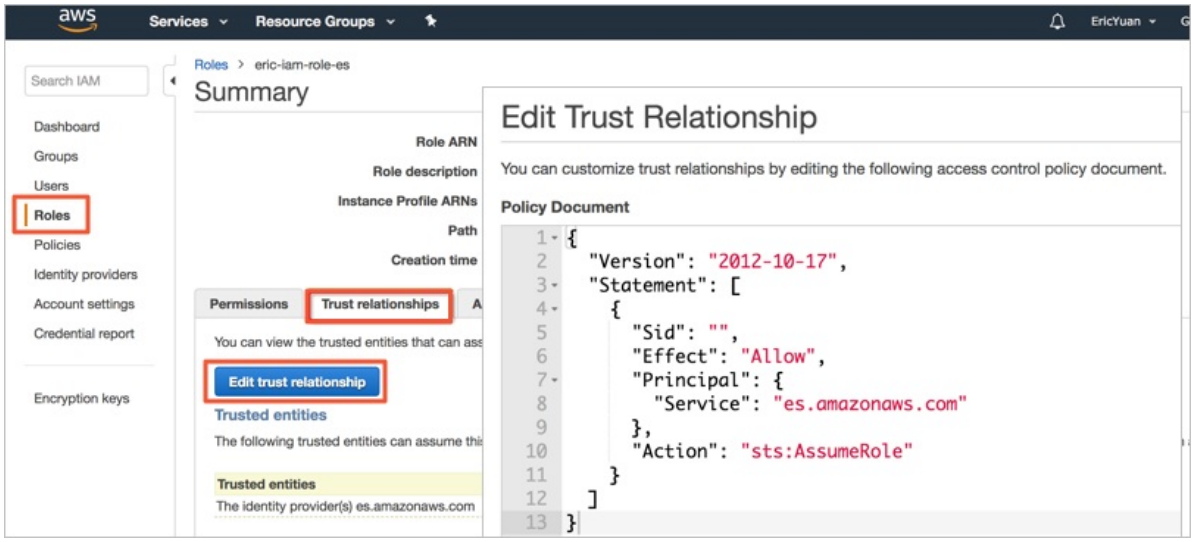
  ```
  arn:aws:s3:::eric-es-index-backups
  ```

- 创建IAM角色

  确保已经创建了一个IAM角色，且在其信任关系中的 `Service` 语句中指定该角色的服务类型为AWS Elasticsearch服务（es.amazonaws.com），如下所示。

  ```
  {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Allow",
        "Principal": {
          "Service": "es.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
      }
    ]
  }
  ```

  您可以在AWS IAM控制台查看信任关系的详细信息。



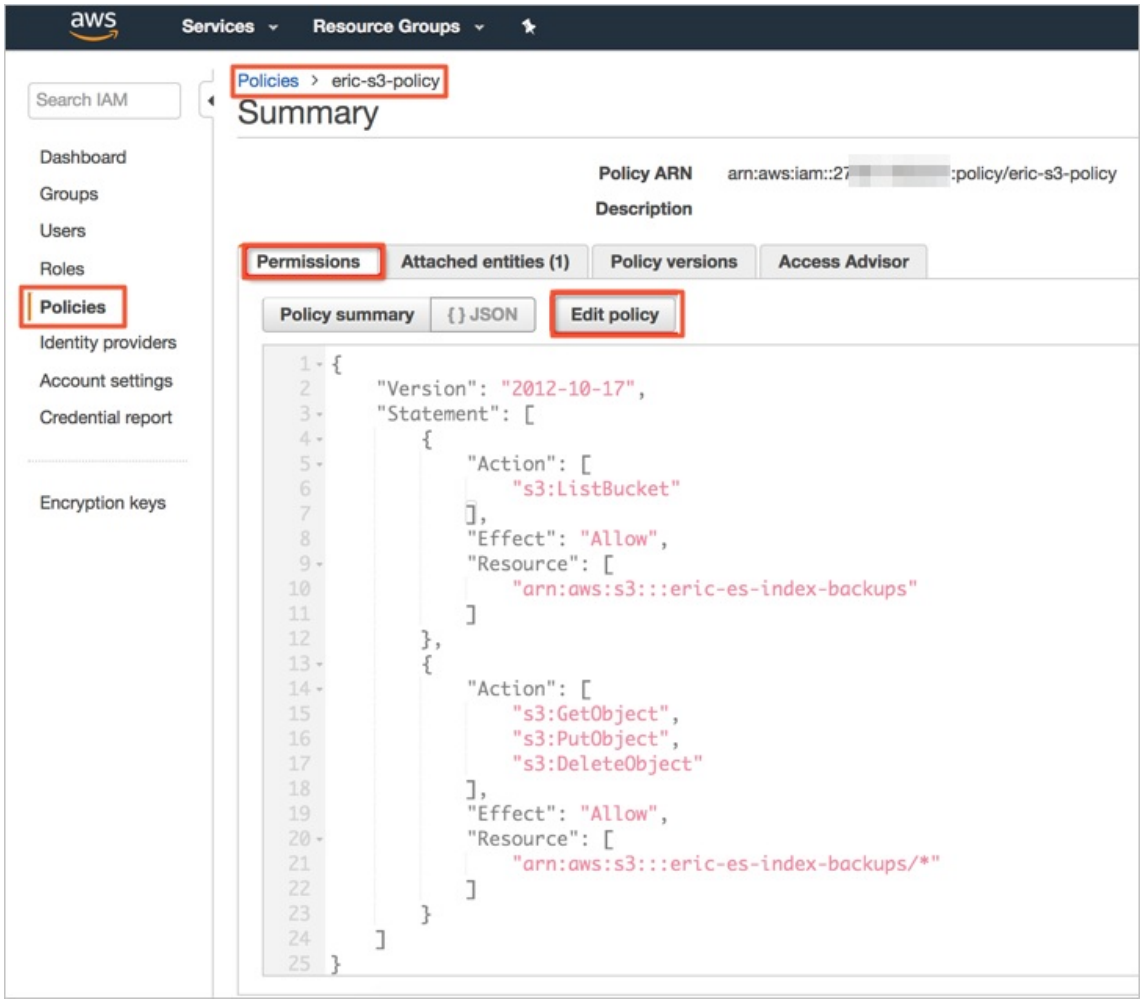
> 说明 在IAM控制台创建AWS服务角色时，Select role type下拉列表中不包含AWS Elasticsearch。但是，您可以先选择Amazon EC2，按照提示完成角色创建，然后将 `ec2.amazonaws.com` 修改为 `es.amazonaws.com` 。

- 创建IAM策略

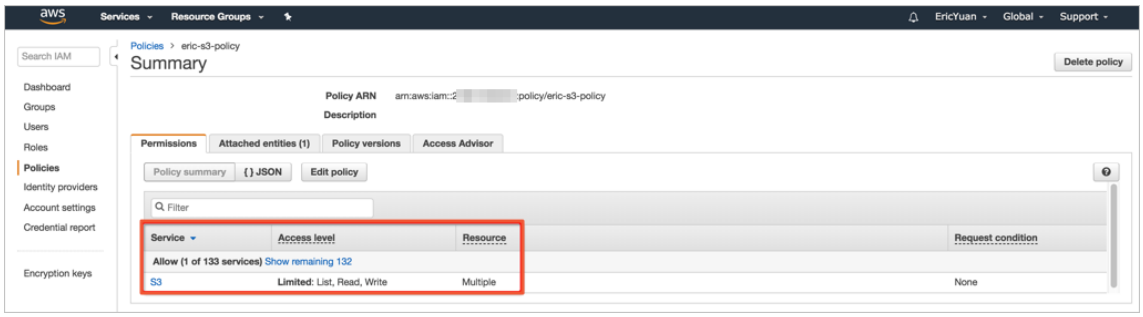
  创建IAM策略，并将IAM策略添加给IAM角色。该策略指定存储AWS Elasticsearch域的S3存储空间。以下示例指定了存储空间 `eric-es-index-backups` 的ARN。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::eric-es-index-backups"
            ]
        },
        {
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::eric-es-index-backups/*"
            ]
        }
    ]
}
```

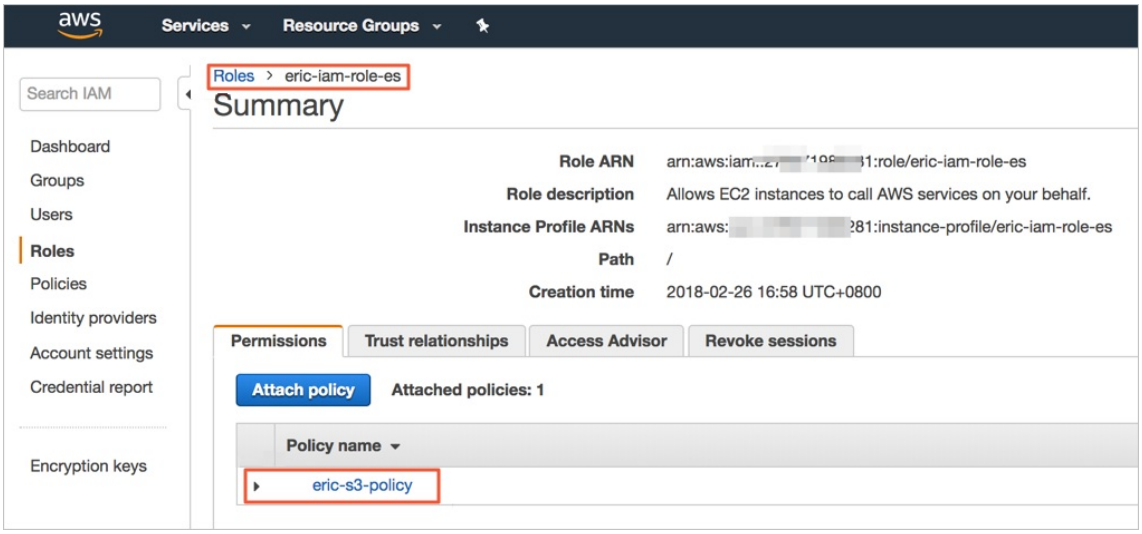
  i. 将策略内容复制到编辑策略区域。



ii. 检查策略是否正确。



iii. 为IAM角色添加IAM策略。



步骤一：注册手动快照存储库

您必须通过AWS Elasticsearch服务注册快照存储库后，才能创建手动索引快照。创建手动索引快照前，需要先为IAM角色的信任关系中指定的用户或角色签发您的AWS请求，详情请参见在AWS中创建手动快照的前提条件。

> **注意** 由于curl命令不支持AWS请求签名，因此不能使用curl命令注册快照存储库。请使用示例Python客户端注册您的快照存储库。

1. 下载示例Python客户端文件。
2. 修改示例Python客户端文件。

   修改文件中标黄的值，填入实际匹配的值。修改完成后，复制示例Python客户端文件中的内容至Python文件中，并命名为snapshot.py。

   示例Python客户端文件中的参数说明如下。

变量名	描述
region	创建快照存储库所在的AWS地域。
host	AWS Elasticsearch域的访问地址。
aws_access_key_id	IAM凭证ID。
aws_secret_access_key	IAM凭证Key。
path	快照存储库的路径。

变量名	描述
data	必须包含上文在AWS中创建手动快照的前提条件中，为IAM角色创建的S3存储空间的名字和ARN。 **注意** • 如果要为快照存储库启用S3托管密钥的服务器端加密，请将 `"server_side_encryption": true` 添加到settings JSON中。 • 如果S3存储空间在ap-southeast-1地域，请使用 `"endpoint": "s3.amazonaws.com"` 替代 `"region": "ap-southeast-1"` 。

3. 安装Amazon Web Services Library boto-2.48.0。

   上文的示例Python客户端，要求您在注册快照存储库的计算机上安装boto软件包的2.x版本。

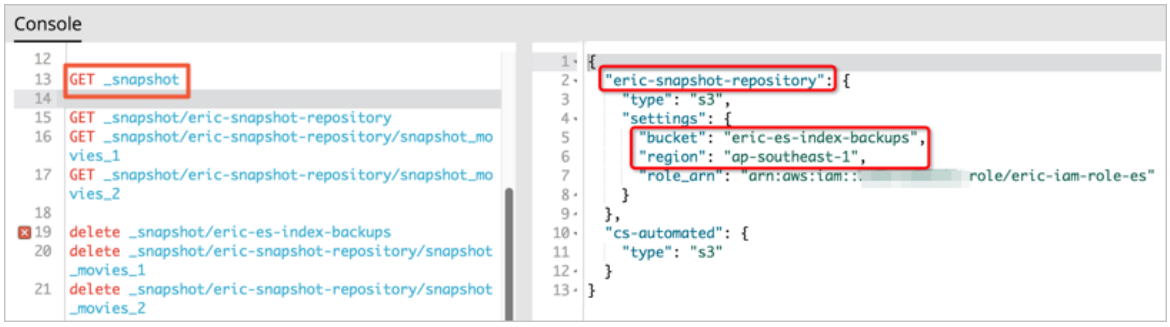
```
# wget https://pypi.python.org/packages/66/e7/fe1db6a5ed53831b53b8a6695a8f134a58833cadb5f2740802bc3730ac15/boto-2.48.0.tar.gz#md5=ce4589dd9c1d7f5d347363223ae1b970
# tar zxvf boto-2.48.0.tar.gz
# cd boto-2.48.0
# python setup.py install
```

4. 执行Python客户端，注册快照存储库。

```
# python snapshot.py
```

5. 进入对应AWS Elasticsearch的Kibana控制台，在**Dev Tools**页面的**Console**中，执行以下命令，查看请求结果。

```
GET _snapshot
```



步骤二：创建首次快照并恢复

1. 在AWS Elasticsearch上手动创建快照。

   说明 以下命令可在Kibana控制台上执行，也可以在Linux或者Mac OSX命令行中使用curl命令来执行。

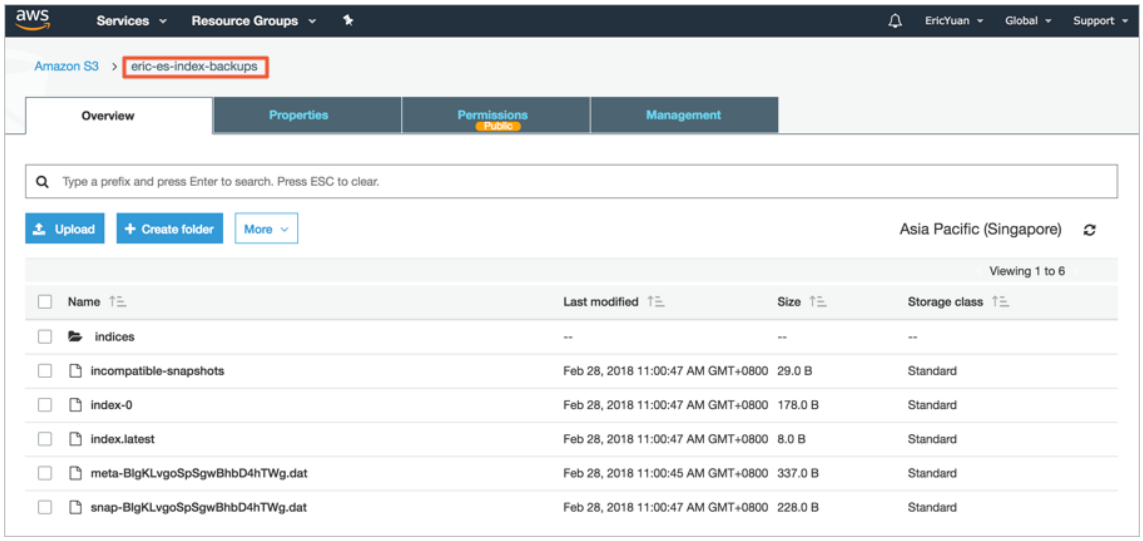
   - 为在 `eric-snapshot-repository` 存储库中的 `movies` 索引，创建名称为 `snapshot_movies_1` 的快照。

```
PUT _snapshot/eric-snapshot-repository/snapshot_movies_1
{
"indices": "movies"
}
```

- 查看快照状态。

```
GET _snapshot/ eric-snapshot-repository/snapshot_movies_1
```

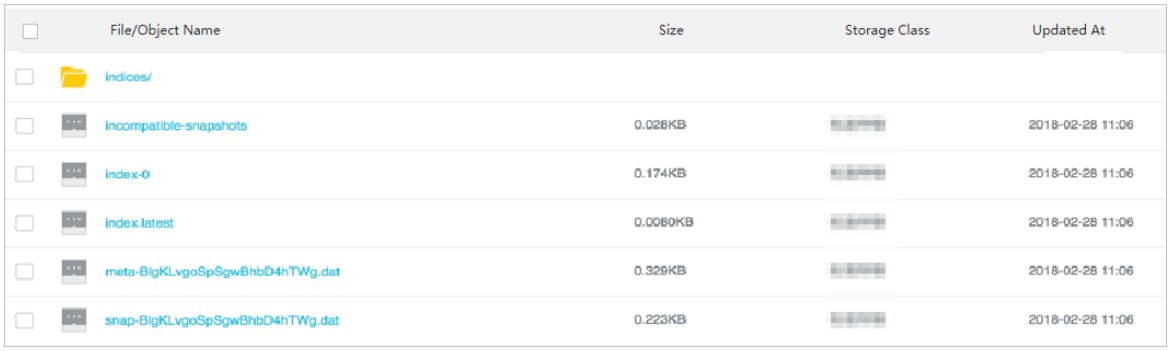
- 在AWS S3控制台中，查看快照文件。



2. 从AWS S3提取快照数据至阿里云OSS。

   详细操作方法请参见从AWS S3上的应用无缝切换至OSS。

   数据提取后，在OSS控制台中查看存储的快照数据。



File/Object Name	Size	Storage Class	Updated At
indices/			
incompatible-snapshots	0.028KB		2018-02-28 11:06
index-0	0.174KB		2018-02-28 11:06
index.latest	0.008KB		2018-02-28 11:06
meta-BlgKLvgoSpSgwBhbD4hTWg.dat	0.329KB		2018-02-28 11:06
snap-BlgKLvgoSpSgwBhbD4hTWg.dat	0.223KB		2018-02-28 11:06

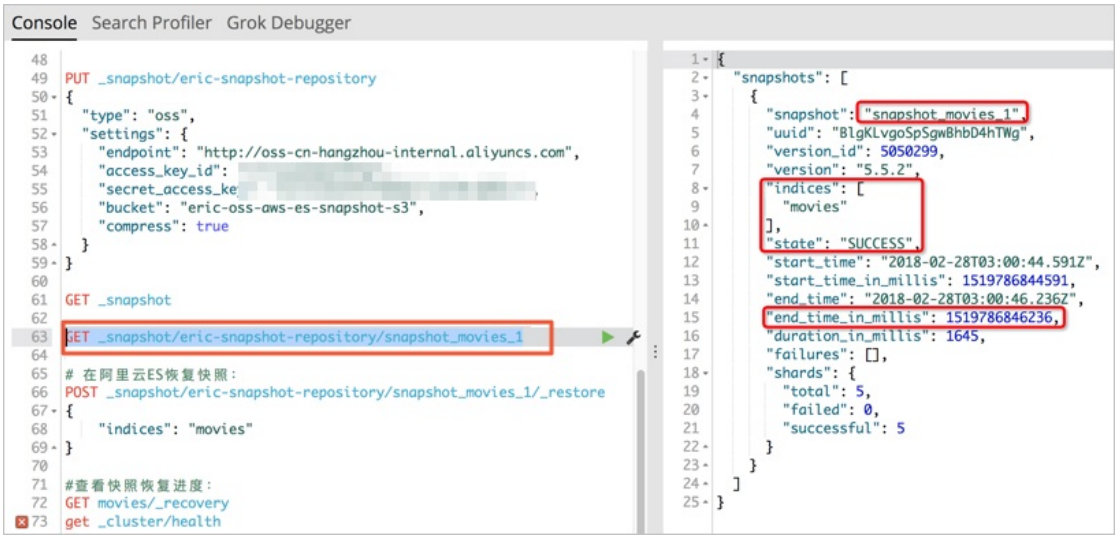
3. 还原快照至阿里云Elasticsearch实例。
   i. 创建快照存储库。

   进入目标阿里云Elasticsearch实例的Kibana控制台（登录Kibana控制台），在**Dev Tools**页面的**Console**中，执行如下命令创建一个同名的快照存储库。

```
PUT _snapshot/eric-snapshot-repository
{
"type": "oss",
"settings": {
            "endpoint": "http://oss-cn-hangzhou-internal.aliyuncs.com",
            "access_key_id": "your AccessKeyID",
            "secret_access_key": "your AccessKeySecret ",
            "bucket": "eric-oss-aws-es-snapshot-s3",
            "compress": true
      }
}
```

ii. 查看名称为 `snapshot_movies_1` 的快照状态。

```
GET _snapshot/eric-snapshot-repository/snapshot_movies_1
```



> **说明** 请记录此快照操作的起始时间和结束时间。当您使用阿里云OssImport迁移工具迁移增量快照数据时，此记录会被用到。例如：
>
> - “start_time_in_millis”：1519786844591
> - “end_time_in_millis”：1519786846236

4. 恢复快照。

   执行以下命令，查看 `movies` 索引的可用性。

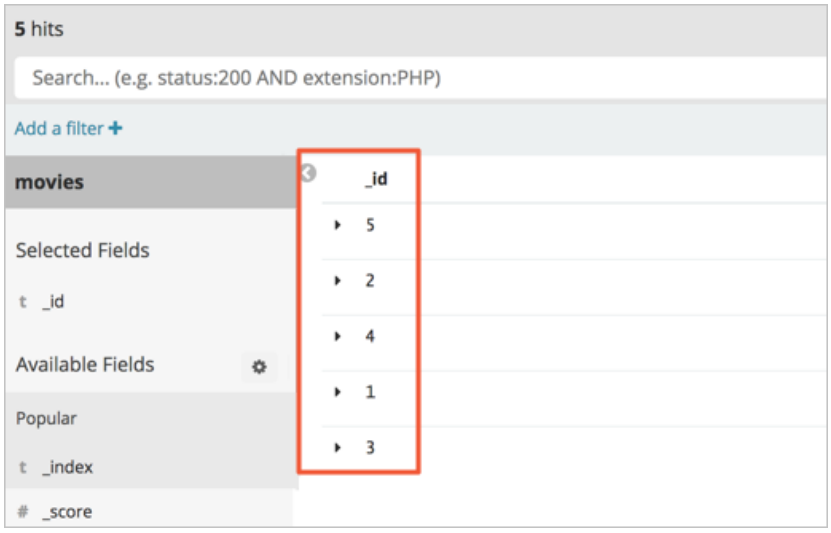
```
POST _snapshot/eric-snapshot-repository/snapshot_movies_1/_restore
{
    "indices": "movies"
}
GET movies/_recovery
```

执行成功后，可以看到 `movies` 索引中存在三组数据，且与AWS Elasticsearch实例中的数据相同。

步骤三：创建末次快照并恢复

1. 在AWS Elasticsearch的 `movies` 索引中插入数据。

`movies` 索引中已存在三组数据，您还需插入另两组数据。



使用 `GET movies/_count` 命令，可查看索引数据量。

2. 手动创建另一个快照。

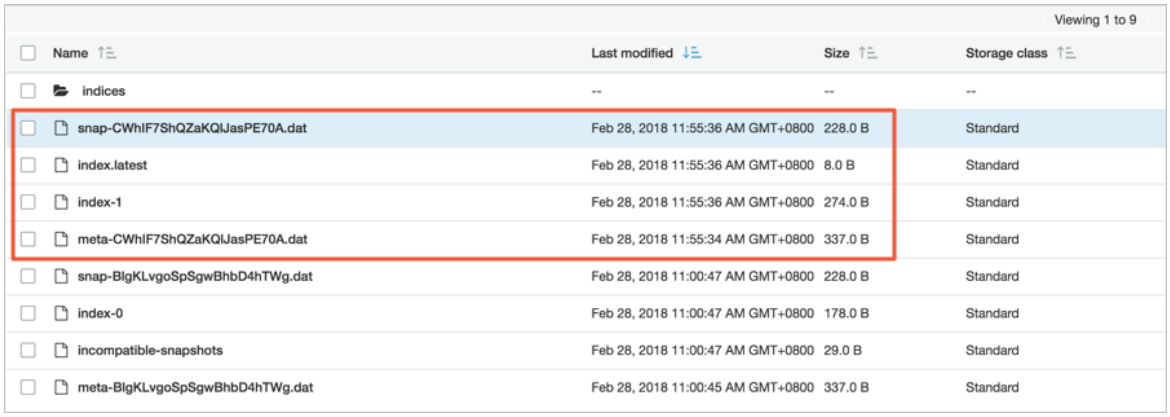
   执行以下命令手动创建快照，详情请参见上文的在AWS Elasticsearch上手动创建快照步骤。

   ```
   PUT _snapshot/eric-snapshot-repository/snapshot_movies_2
   {
   "indices": "movies"
   }
   ```

   创建成功后，执行以下命令查看快照状态。

   ```
   GET _snapshot/eric-snapshot-repository/snapshot_movies_2
   ```

   查看S3存储空间中列出的文件。



3. 从AWS S3提取增量快照数据至阿里云OSS。

   您可以使用OSSImport工具从AWS S3迁移数据至阿里云OSS。目前有两个快照文件存储在S3存储空间里，可以通过修改配置文件*local_job.cfg*中的 `isSkipExistFile` 变量来迁移新的文件。

   `isSkipExistFile` 表示数据迁移期间是否跳过现有对象，为布尔类型，默认值为false。如果设置为true，则根据 `size` 和 `LastModifiedTime` 跳过对象；如果设置为false，则覆盖现有对象。当 `jobType` 设置为 `audit` 时，此选项无效。

迁移工作完成后，您可以看到新的文件已被迁移至OSS中。

4. 恢复增量快照。

   恢复增量快照详细步骤请参见上文的步骤二：创建首次快照并恢复章节中的恢复快照步骤。但是首先需要关闭 `movies` 索引，然后再恢复快照。快照恢复后可以再次打开 `movies` 索引。

   - 关闭 `movies` 索引

     ```
     POST /movies/_close
     ```

   - 查看 `movies` 索引状态

     ```
     GET movies/_stats
     ```

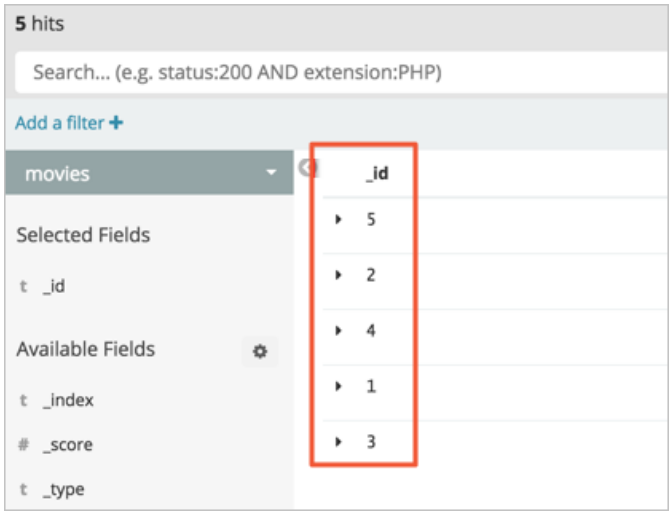
   - 恢复增量快照

     ```
     POST _snapshot/eric-snapshot-repository/snapshot_movies_2/_restore
     {
         "indices": "movies"
     }
     ```

   - 打开 `movies` 索引

     ```
     POST /movies/_open
     ```

   恢复快照步骤完成后，可以看到 `movies` 索引中文档数量为 `5` ，与AWS Elasticsearch实例中的文档数量相同。



总结

您可以通过创建和恢复快照的方法，将AWS Elasticsearch的索引数据迁移至阿里云Elasticsearch索引中。此方案要求先关闭需要迁移的AWS Elasticsearch索引，以防止迁移期间的写入和请求。

相关文档：

- Elasticsearch官方文档
- Snapshot module
- Working with Amazon Elasticsearch Service Index Snapshots
- 从AWS S3上的应用无缝切换至OSS
- OSS快照迁移
- ossimport说明及配置

3.数据库同步

3.1. RDS MySQL同步

3.1.1. 同步方案选取指南

当您在使用关系型数据库（RDS）遇到查询慢的问题时，可将RDS中的数据同步至阿里云Elasticsearch进行查询分析。阿里云Elasticsearch是一个基于Lucene的实时分布式的搜索与分析引擎，可近乎于准实时地存储、查询和分析超大数据集。您可以通过DTS、Logstash、DataWorks和Canal等多种方式将RDS中的数据同步至阿里云Elasticsearch。本文介绍各同步方案适用的场景，帮助您根据业务选择合适的场景同步数据。

同步方案	原理说明	适用场景	使用限制	相关文档
DTS实现数据实时同步	DTS采用binlog方式实现数据同步，在不影响源数据库的情况下，同步延迟可降至毫秒级别。	对数据同步的实时性要求较高的场景。	• DTS在执行全量数据初始化时将占用源库和目标库一定的读写资源，可能会导致数据库负载上升。 • 支持自定义索引Mapping，但需保证Mapping中定义的字段与MySQL中一致。 • 需要购买DTS数据同步作业。购买方式，请参见购买流程；定价说明，请参见产品定价。	通过DTS将MySQL数据实时同步到阿里云Elasticsearch
Logstash JDBC数据同步	通过logstash-input-jdbc插件实现通过Logstash批量查询RDS中的数据，并将数据迁移到Elasticsearch。实现的本质是该插件会定期对RDS中的数据进行循环轮询，从而在当前循环中找到上次插入或更改的记录，然后批量查询这些记录并迁移至Elasticsearch。与DTS同步方案相比，该方案的实时性较差，存在秒级延迟。	• 同步全量数据，接受秒级延迟的场景。 • 批量查询数据然后进行同步的场景。	• 使用前，需要先在Logstash中上传与RDS版本兼容的SQL JDBC驱动。 • 需要在RDS的白名单中加入Logstash集群中节点的IP地址。 • 需要确保Logstash和RDS在同一时区（避免同步过程中出现时间标记不符的情况）。 • 需要确保Elasticsearch中的_id字段与RDS中的id字段相同。 • 当您在RDS中插入或更新数据时，需要确保对应记录有一个包含更新或插入时间的字段。	通过Logstash将RDS MySQL数据同步至Elasticsearch

同步方案	原理说明	适用场景	使用限制	相关文档
DataWorks实现离线数据同步	DataWorks是一款提供数据集成、数据开发及数据质量等全方位的产品服务。支持引入并存储关系型数据，然后进行转化和开发，最后将处理后的数据同步到Elasticsearch或其他数据系统。	• 大数据离线同步场景（可实现最快5分钟一次的离线数据采集任务）。 • 需要自定义查询语句，以及多表联合查询后同步数据的场景。 • 同步整个数据库中数据的场景。	• 需要开通DataWorks服务。 • 对于传输速度要求较高或复杂环境中的数据源同步场景，需要自定义资源组。 • 需要在RDS的白名单中添加资源组的IP地址。	通过DataWorks将MySQL数据同步至Elasticsearch
Canal实现MySQL数据同步	通过binlog方式实现数据实时同步及订阅。	对数据同步的实时性要求较高的场景。	• 需要手动在ECS上搭建Canal环境，会增加操作成本。 • Canal 1.1.4版本暂不支持Elasticsearch 7.x版本。建议使用Canal 1.1.5版本，或者通过其他方式（例如Logstash、DTS）实现MySQL数据同步。 • 支持自定义索引Mapping，但需保证Mapping中定义的字段与MySQL中一致。	通过Canal将MySQL数据同步到阿里云Elasticsearch

3.1.2. 通过Logstash将RDS MySQL数据同步至Elasticsearch

logstash同步mysql数据到es

当您需要将RDS MySQL中的数据同步到阿里云Elasticsearch中时，可使用阿里云Logstash的logstash-input-jdbc插件（默认已安装，不可卸载），通过管道配置功能实现。通过该方案，您可以将全量或增量数据实时同步至阿里云Elasticsearch中。本文介绍具体的实现方法。

背景信息

目前，很多厂商都会在既有关系型数据库的基础上再部署Elasticsearch。在这种情况下，您可能需要确保Elasticsearch与所关联的关系型数据库中的数据保持同步。因此，本文会为您介绍如何使用Logstash将关系型数据库中的数据同步到阿里云Elasticsearch中。更多详细信息，请参见官方文档中的如何使用Logstash和JDBC确保Elasticsearch与关系型数据库保持同步。

使用限制

使用logstash-input-jdbc插件实现阿里云Elasticsearch和MySQL同步的本质是：该插件会定期对MySQL中的数据进行循环轮询，从而在当前循环中找到上次插入或更改的记录。因此要让同步任务正确运行，Elasticsearch和MySQL必须满足以下条件：

- Elasticsearch中的_id字段必须与MySQL中的id字段相同。

  该条件可以确保当您将MySQL中的记录写入Elasticsearch时，同步任务可在MySQL记录与Elasticsearch文档之间建立一个直接映射的关系。例如当您在MySQL中更新了某条记录时，同步任务会覆盖Elasticsearch中与更新记录具有相同ID的文档。

  > **说明** 根据Elasticsearch内部原理，更新操作的本质是删除旧文档然后对新文档进行索引，因此在Elasticsearch中覆盖文档的效率与更新操作的效率一样高。

- 当您在MySQL中插入或者更新数据时，对应记录必须有一个包含更新或插入时间的字段。

  Logstash每次对MySQL进行轮询时，都会保存其从MySQL所读取的最后一条记录的更新或插入时间。在读取数据时，Logstash仅读取符合条件的记录，即该记录的更新或插入时间需要晚于上一次轮询中最后一条记录的更新或插入时间。

  > **注意** logstash-input-jdbc插件无法实现同步删除，需要在Elasticsearch中执行相关命令手动删除。

- MySQL和Elasticsearch实例在同一时区，否则当同步与时间相关的数据时，同步前后的数据可能存在时区差。

操作流程

1. 步骤一：环境准备
2. 步骤二：配置Logstash管道
3. 步骤三：验证结果

步骤一：环境准备

1. 创建阿里云Elasticsearch实例，并开启自动创建索引功能。

   具体操作，请参见创建阿里云Elasticsearch实例和快速访问与配置。本文使用7.10版本的实例。

2. 创建阿里云Logstash实例，并上传与RDS MySQL版本兼容的SQL JDBC驱动（本文使用mysql-connector-java-5.1.48.jar）。

   创建时所选专有网络和版本要与目标Elasticsearch实例相同。具体操作，请参见步骤一：创建阿里云Logstash实例和配置扩展文件。

   > **说明** 您也可以使用公网环境的服务，前提是需要配置SNAT、打开RDS MySQL的公网地址并取消白名单限制。SNAT的具体配置方法，请参见配置NAT公网数据传输。

3. 准备测试数据，并在RDS MySQL的白名单中加入阿里云Logstash节点的IP地址（可在基本信息页面获取）。

   设置白名单的具体操作，请参见通过客户端、命令行连接RDS MySQL实例。

   本文使用的建表语句如下。

   ```
   CREATE table food(
   id int PRIMARY key AUTO_INCREMENT,
   name VARCHAR (32),
   insert_time DATETIME,
   update_time DATETIME );
   ```

   插入数据语句如下。

```
INSERT INTO food values(null,'巧克力',now(),now());
INSERT INTO food values(null,'酸奶',now(),now());
INSERT INTO food values(null,'火腿肠',now(),now());
```

步骤二：配置Logstash管道

1. 登录阿里云Elasticsearch控制台。
2. 进入目标实例。
   i. 在顶部菜单栏处，选择地域。
   ii. 在左侧导航栏，单击**Logstash实例**，然后在**Logstash实例**中单击目标实例ID。
3. 在左侧导航栏，单击**管道管理**。
4. 单击**创建管道**。
5. 在**创建管道任务**页面，输入**管道ID**，并进行Config配置。

   本文使用的Config配置如下。

```
input {
  jdbc {
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    jdbc_driver_library => "/ssd/1/share/<Logstash实例ID>/logstash/current/config/custom/mysql-connector-java-5.1.48.jar"
    jdbc_connection_string => "jdbc:mysql://rm-bp1xxxxx.mysql.rds.aliyuncs.com:3306/<数据库名称>?useUnicode=true&characterEncoding=utf-8&useSSL=false&allowLoadLocalInfile=false&autoDeserialize=false"
    jdbc_user => "xxxxx"
    jdbc_password => "xxxx"
    jdbc_paging_enabled => "true"
    jdbc_page_size => "50000"
    statement => "select * from food where update_time >= :sql_last_value"
    schedule => "* * * * *"
    record_last_run => true
    last_run_metadata_path => "/ssd/1/<Logstash实例ID>/logstash/data/last_run_metadata_update_time.txt"
    clean_run => false
    tracking_column_type => "timestamp"
    use_column_value => true
    tracking_column => "update_time"
  }
}
filter {
}
output {
 elasticsearch {
    hosts => "http://es-cn-0h****dd0hcbnl.elasticsearch.aliyuncs.com:9200"
    index => "rds_es_dxhtest_datetime"
    user => "elastic"
    password => "xxxxxxx"
    document_id => "%{id}"
  }
}
```

> **说明** 代码中 `<Logstash实例ID>` 需要替换为您创建的Logstash实例的ID。获取方式，请参见查看实例的基本信息。

Config配置说明

配置	说明
input	指定输入数据源。支持的数据源类型，请参见Input plugins。本文使用JDBC数据源，具体参数说明请参见input参数说明。
filter	指定对输入数据进行过滤的插件。支持的插件类型，请参见Filter plugins。
output	指定目标数据源类型。支持的数据源类型，请参见Output plugins。本文需要将MySQL中的数据同步至Elasticsearch中，因此output中需要指定目标Elasticsearch的信息。具体参数说明，请参见步骤二：创建并运行管道任务。 **注意** 如果output中使用了file_extend参数，需要先安装logstash-output-file_extend插件。具体操作，请参见安装或卸载插件。

input参数说明

参数	描述
jdbc_driver_class	JDBC Class配置。
jdbc_driver_library	指定JDBC连接MySQL驱动文件，格式为/ssd/1/share/<Logstash实例ID>/logstash/current/config/custom/<驱动文件名称>。您需要提前在控制台中上传驱动文件，阿里云Logstash支持的驱动文件及其上传方法，请参见配置扩展文件。
jdbc_connection_string	配置数据库连接的域名、端口及数据库，格式为jdbc:mysql://<MySQL的连接地址>:<端口>/<数据库名称>?useUnicode=true&characterEncoding=utf-8&useSSL=false&allowLoadLocalInfile=false&autoDeserialize=false。 **注意** ◦ <MySQL的连接地址>：需要配置MySQL的内网地址。如果使用外网地址，需要为Logstash配置NAT网关实现公网数据传输，具体操作请参见配置NAT公网数据传输。 ◦ 端口：需要与MySQL的出方向端口保持一致，一般为3306。
jdbc_user	数据库用户名。
jdbc_password	数据库密码。

参数	描述
jdbc_paging_enabled	是否启用分页，默认false。
jdbc_page_size	分页大小。
statement	指定SQL语句，多表查询可使用join语句。 **说明** sql_last_value用于计算要查询哪一行，在运行任何查询之前，此值设置为1970年1月1日星期四。详细信息，请参见Jdbc input plugin。
schedule	指定定时操作，"* * * * *"表示每分钟定时同步数据。该参数使用的是Rufus版的Cron表达式。
record_last_run	是否记录上次执行结果。如果为true，则会把上次执行到的tracking_column字段的值记录下来，保存到last_run_metadata_path指定的文件中。
last_run_metadata_path	指定最后运行时间文件存放的地址。目前后端开放了 `/ssd/1/<Logstash实例ID>/logstash/data/` 路径来保存文件。 **说明** 配置Logstash管道时，建议按照 `/ssd/1/<Logstash实例ID>/logstash/data/` 路径配置此参数。如果不按照该路径配置，会导致同步的条件记录因为权限不足而无法存放在last_run_metadata_path路径下的配置文件中。
clean_run	是否清除last_run_metadata_path的记录，默认为false。如果为true，那么每次都要从头开始查询所有的数据库记录。
use_column_value	是否需要记录某个column的值。当该值设置成true时，系统会记录tracking_column参数所指定的列的最新的值，并在下一次管道执行时通过该列的值来判断需要更新的记录。
tracking_column_type	跟踪列的类型，默认是numeric。
tracking_column	指定跟踪列，该列必须是递增的，一般是MySQL主键。

> **注意**
>
> - 以上配置按照测试数据配置，在实际业务中，请按照业务需求进行合理配置。input插件支持的其他配置选项，请参见官方Logstash Jdbc input plugin文档。
> - 如果配置中有类似last_run_metadata_path的参数，那么需要阿里云Logstash服务提供文件路径。目前后端开放了 `/ssd/1/<Logstash实例ID>/logstash/data/` 路径供您测试使用，且该目录下的数据不会被删除。因此在使用时，请确保磁盘有充足的使用空间。
> - 为了提升安全性，如果在配置管道时使用了JDBC驱动，需要在jdbc_connection_string参数后面添加allowLoadLocalInfile=false&autoDeserialize=false，否则当您在添加Logstash配置文件时，调度系统会抛出校验失败的提示，例如 `jdbc_connection_string => "jdbc:mysql://xxx.drds.aliyuncs.com:3306/<数据库名称>?allowLoadLocalInfile=false&autoDeserialize=false"` 。

更多Config配置，请参见Logstash配置文件说明。

6. 单击**下一步**，配置管道参数。

参数	说明
管道工作线程	并行执行管道的Filter和Output的工作线程数量。当事件出现积压或CPU未饱和时，请考虑增大线程数，更好地使用CPU处理能力。默认值：实例的CPU核数。
管道批大小	单个工作线程在尝试执行Filter和Output前，可以从Input收集的最大事件数目。较大的管道批大小可能会带来较大的内存开销。您可以设置LS_HEAP_SIZE变量，来增大JVM堆大小，从而有效使用该值。默认值：125。
管道批延迟	创建管道事件批时，将过小的批分派给管道工作线程之前，要等候每个事件的时长，单位为毫秒。默认值：50ms。
队列类型	用于事件缓冲的内部排队模型。可选值： - **MEMORY**：默认值。基于内存的传统队列。 - **PERSISTED**：基于磁盘的ACKed队列（持久队列）。
队列最大字节数	请确保该值小于您的磁盘总容量。默认值：1024 MB。
队列检查点写入数	启用持久性队列时，在强制执行检查点之前已写入事件的最大数目。设置为0，表示无限制。默认值：1024。

> **警告** 配置完成后，需要保存并部署才能生效。保存并部署操作会触发实例重启，请在不影响业务的前提下，继续执行以下步骤。

7. 单击**保存**或者**保存并部署**。
   - **保存**：将管道信息保存在Logstash里并触发实例变更，配置不会生效。保存后，系统会返回**管道管理**页面。可在**管道列表**区域，单击**操作**列下的**立即部署**，触发实例重启，使配置生效。
   - **保存并部署**：保存并且部署后，会触发实例重启，使配置生效。

步骤三：验证结果

1. 登录目标阿里云Elasticsearch实例的Kibana控制台，根据页面提示进入Kibana主页。

   登录Kibana控制台的具体操作，请参见登录Kibana控制台。

   > **说明** 本文以阿里云Elasticsearch 7.10.0版本为例，其他版本操作可能略有差别，请以实际界面为准。

2. 单击右上角的**Dev tools**。
3. 在**Console**中，执行如下命令，查看同步成功的索引数量。

```
GET rds_es_dxhtest_datetime/_count
{
  "query": {"match_all": {}}
}
```

   预期结果如下。

```
{
  "count" : 3,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  }
}
```

4. 更新MySQL表数据并插入表数据。

```
UPDATE food SET name='Chocolates',update_time=now() where id = 1;
INSERT INTO food values(null,'鸡蛋',now(),now());
```

5. 在Kibana控制台，查看更新后的数据。
   - 查询name为Chocolates的数据。

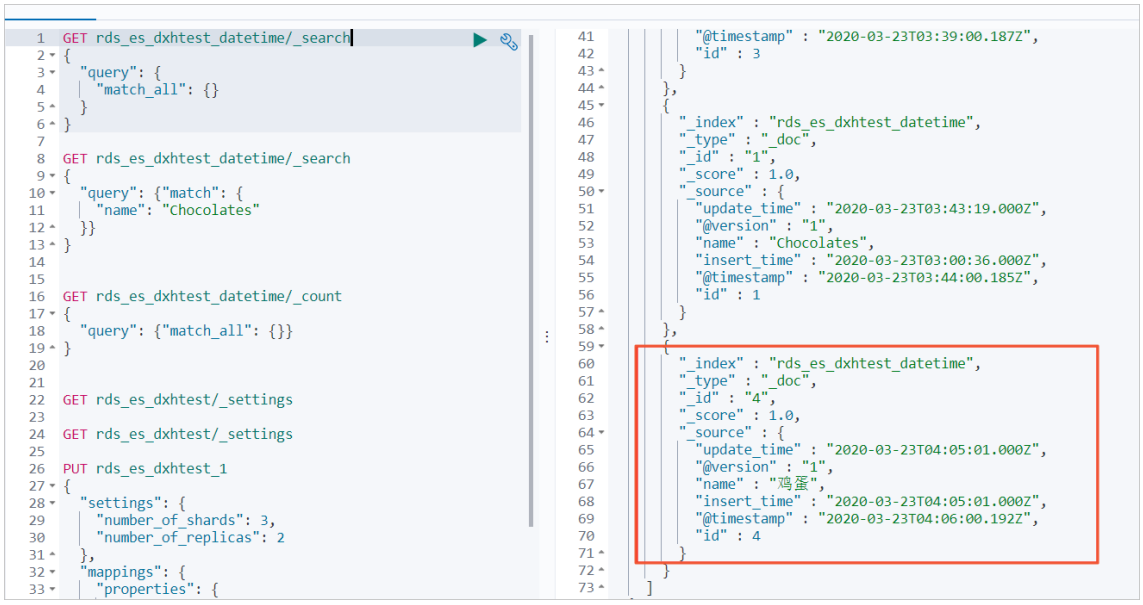
```
GET rds_es_dxhtest_datetime/_search
{
  "query": {
    "match": {
      "name": "Chocolates"
   }}
}
```

预期结果如下。

- 查询所有数据。

```
GET rds_es_dxhtest_datetime/_search
{
  "query": {
    "match_all": {}
  }
}
```

预期结果如下。



常见问题

Q：同步任务失败（例如管道一直在生效中、前后数据不一致、数据库连接不成功），如何解决？

A：查看Logstash实例的主日志是否有报错，根据报错判断原因，具体操作请参见查询日志。常见的原因及解决方法如下。

> **说明** 执行以下操作时，如果集群正在变更中，可参见查看实例任务进度详情先中断变更，操作完成后再触发重启恢复。

原因	解决方法
MySQL白名单中没有加入Logstash的IP地址。	参见通过客户端、命令行连接RDS MySQL实例，在MySQL白名单中加入Logstash节点的IP地址。 > **说明** 获取Logstash的IP地址的具体操作，请参见查看实例的基本信息。
Logstash的IP地址没有添加到对应ECS服务器的安全组中（ECS自建MySQL）。	参见添加安全组规则，在ECS安全组中添加Logstash的IP地址和端口号。 > **说明** 获取Logstash的IP地址和端口号的具体操作，请参见查看实例的基本信息。
Logstash和Elasticsearch不在同一VPC下。	选择以下任意一种方式处理： • 参见创建阿里云Elasticsearch实例，重新购买同一VPC下的Elasticsearch实例。购买后，修改现有管道配置。 • 参见配置NAT公网数据传输，配置NAT网关实现公网数据传输。
MySQL地址不正确，端口不是3306。	参见查看或修改内外网地址和端口，获取正确的地址和端口。使用正确的地址和端口，按照脚本格式替换管道配置中的jdbc_connection_string参数值。 > **注意** <MySQL的连接地址>：需要配置MySQL的内网地址。如果使用外网地址，需要为Logstash配置NAT网关实现公网数据传输，具体操作请参见配置NAT公网数据传输。
Elasticsearch未开启自动创建索引。	参见配置YML参数，开启Elasticsearch实例的自动创建索引功能。
Elasticsearch或Logstash的负载太高。	参见升配集群，升级实例规格。 > **说明** Elasticsearch负载情况可参见查看集群监控，通过控制台监控指标查看。Logstash负载情况可参见配置X-Pack监控，通过Kibana X-Pack监控查看。
没有上传JDBC连接MySQL的驱动文件。	参见配置扩展文件，下载并上传驱动文件。
管道配置中包含了file_extend，但没有安装logstash-output-file_extend插件。	选择以下任意一种方式处理： • 参见安装或卸载插件，安装logstash-output-file_extend插件。 • 在管道配置中，去掉file_extend配置。

更多问题原因及解决方法，请参见Logstash数据写入问题排查方案。

3.1.3. 通过DataWorks将MySQL数据同步至Elasticsearch

MySQL数据同步到阿里云es

阿里云上拥有丰富的云存储、云数据库产品。当您需要对这些产品中的数据进行分析和搜索时，可以通过DataWorks的数据集成服务，实现最快5分钟一次的离线数据采集任务，并同步到阿里云Elasticsearch中。本教程以阿里云RDS MySQL为例。

教程概述

1. 准备工作

   准备MySQL数据源、创建DataWorks工作空间、创建与配置阿里云Elasticsearch实例。

2. 步骤一：购买并创建独享资源组

   购买并创建一个数据集成独享资源组，并为该资源组绑定专有网络和工作空间。独享资源组可以保障数据快速、稳定地传输。

3. 步骤二：添加数据源

   将MySQL和Elasticsearch数据源接入DataWorks的数据集成服务中。

4. 步骤三：配置并运行数据同步任务

   通过向导模式配置数据同步任务，将数据集成系统同步成功的数据存储到Elasticsearch中。将独享资源组作为一个可以执行任务的资源，注册到DataWorks的数据集成服务中。这个资源组将获取数据源的数据，并执行将数据写入Elasticsearch中的任务（该任务将由数据集成系统统一下发）。

5. 步骤四：验证数据同步结果

   在Kibana控制台中，查看同步成功的数据。

准备工作

1. 创建一个数据库。

   您可以选择使用阿里云的RDS数据库，也可以在本地服务器上自建数据库。本教程以RDS MySQL数据库为例，使用JOIN获取两张表数据，同步数据到阿里云Elasticsearch中，表字段及数据如下所示。具体操作，请参见创建RDS MySQL实例。

   表一

	id	stu_id	c_name	grade
1	1	901	计算机	98
2	2	901	英语	80
3	3	902	计算机	65
4	4	902	中文	88
5	5	903	中文	95
6	6	904	计算机	70
7	7	904	英语	92
8	8	905	英语	94
9	9	906	计算机	90
10	10	906	英语	85

   表二

	id	name	sex	birth	department	address
1	901	张老大	男	1985	计算机系	北京市海淀区
2	902	张老二	男	1986	中文系	北京市昌平区
3	903	张三	女	1990	中文系	湖南省永州市
4	904	李四	男	1990	英语系	辽宁省阜新市
5	905	王五	女	1991	英语系	福建省厦门市
6	906	王六	男	1988	计算机系	湖南省衡阳市

2. 创建DataWorks工作空间。

   具体操作，请参见创建工作空间。工作空间所在地域需要与RDS MySQL一致。

3. 创建阿里云Elasticsearch实例，并开启实例的自动创建索引功能。

   创建实例时，所选专有网络需要与RDS MySQL保持一致。具体操作，请参见创建阿里云Elasticsearch实例和配置YML参数。

步骤一：购买并创建独享资源组

1. 登录DataWorks控制台。
2. 选择相应地域后，在左侧导航栏，单击**资源组列表**。
3. 参见购买资源组（创建订单），购买独享数据集成资源。

   > **注意** 购买时，所选地域需要与目标工作空间保持一致。

4. 参见新增和使用独享数据集成资源组，创建一个独享数据集成资源。

   本文使用的配置如下，其中**资源组类型**选择**独享数据集成资源组**。

5. 单击已创建的独享资源组右侧的**专有网络绑定**，参见绑定专有网络，为该独享资源组绑定专有网络。

   独享资源部署在DataWorks托管的专有网络中。DataWorks需要与MySQL和Elasticsearch实例的专有网络

连通才能同步数据。而MySQL和Elasticsearch实例在同一专有网络下，因此在绑定专有网络时，选择Elasticsearch实例所在**专有网络**和**交换机**即可。

6. 单击已创建的独享资源组右侧的**修改归属工作空间**，参见新增和使用独享数据集成资源组，为该独享资源组绑定目标工作空间。

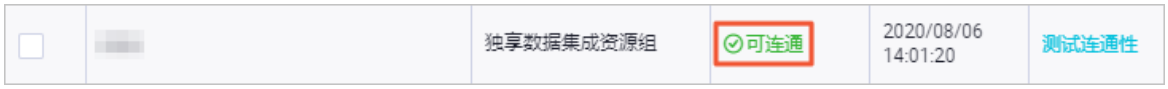
步骤二：添加数据源

1. 进入DataWorks的**数据集成**页面。
   i. 在DataWorks控制台的左侧导航栏，单击**工作空间列表**。
   ii. 找到目标工作空间，单击其右侧**操作**列下的**进入数据集成**。
2. 在左侧导航栏，选择**数据源 > 数据源列表**。
3. 在**数据源管理**页面，单击**新增数据源**。
4. 在**新增数据源**对话框中，单击**MySQL**，进入**新增MySQL数据源**页面，填写数据源信息。

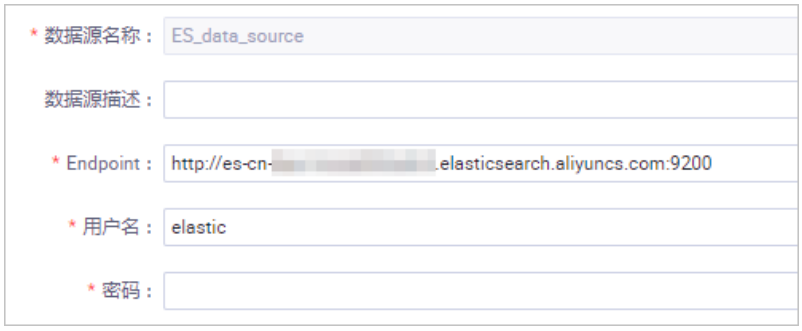
数据源类型：本教程以**阿里云实例模式**为例，您也可以选择**连接串模式**。各配置项的详细说明，请参见配置MySQL数据源。

> **注意** 如果您选择的是**连接串模式**，可以通过RDS MySQL的公网地址配置JDBC URL，但需要将独享资源组的EIP地址添加到MySQL的白名单中。详细信息，请参见设置RDS MySQL白名单和使用独享数据集成资源组执行任务需要在数据库添加的IP白名单。

配置完成后，可与独享资源组进行连通性测试。**连通状态**显示为**可连通**时，表示连通成功。



5. 单击**完成**。
6. 使用同样的方式添加Elasticsearch数据源。



参数	说明
Endpoint	阿里云Elasticsearch的访问地址，格式为：`http://<实例的内网或公网地址>:9200`。实例的内网或公网地址可在基本信息页面获取，详细信息，请参见查看实例的基本信息。 **注意** 如果您使用的是公网地址，需要将独享资源组的EIP地址添加到阿里云Elasticsearch的公网地址访问白名单中，详情请参见配置实例公网或私网访问白名单和使用独享数据集成资源组执行任务需要在数据库添加的IP白名单。
用户名	访问阿里云Elasticsearch实例的用户名，默认为elastic。
密码	对应用户的密码。elastic用户的密码在创建实例时设定，如果忘记可重置，重置密码的注意事项和操作步骤，请参见重置实例访问密码。

说明 其他未提及的参数请自定义输入。

步骤三：配置并运行数据同步任务

1. 在DataWorks的数据开发页面，新建一个业务流程。

   具体操作步骤请参见管理业务流程。

2. 新建一个离线同步任务。

   i. 展开新建的业务流程，右键单击**数据集成**，选择**新建 > 离线同步**。

   ii. 在**新建节点**对话框中，输入**节点名称**，单击**提交**。

3. 在**选择数据源**区域中，将**数据来源**指定为MySQL数据源，并填入待同步的表名称；将**数据去向**指定为Elasticsearch数据源，并填入索引名和索引类型。

> 说明
> - 您也可以使用脚本模式配置数据同步，详情请参见通过脚本模式配置离线同步任务、DRDS Reader和Elasticsearch Writer。
> - 建议在Elasticsearch数据源的**高级配置**下，将**启用节点发现**设置为**否**，否则同步过程中提示连接超时。

4. 在**字段映射**区域中，设置**来源字段**与**目标字段**的映射关系。
5. 在**通道控制**区域中，配置执行任务的相关参数。
6. 配置任务调度属性。

   在页面右侧，单击**调度配置**，按照需求配置相应的调度参数。各配置的详细说明请参见调度配置章节。

> 注意
> - 在提交任务前，必须配置任务调度**依赖的上游节点**，详情请参见配置同周期调度依赖。
> - 如果您希望对任务进行周期性调度，需要配置任务的**时间属性**，包括任务的具体执行时间、调度周期、生效周期、重跑属性等。
> - 周期任务将于配置任务开始的第二天00:00，按照您的配置规则生效执行。

7. 配置执行同步任务所使用的资源组。

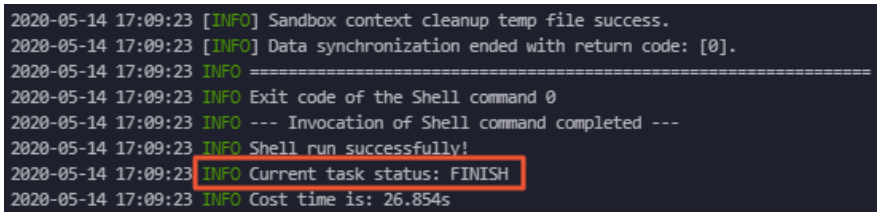
   i. 在页面右侧，单击**数据集成资源组配置**。
   ii. 选择**独享数据集成资源组**为您创建的独享资源组。

8. 提交任务。

   i. 保存当前配置，单击图标。
   ii. 在**提交新版本**对话框中，填入**备注**。
   iii. 单击**确认**。

9. 单击图标，运行任务。

任务运行过程中，可查看运行日志。运行成功后，显示如下结果。



步骤四：验证数据同步结果

1. 登录目标阿里云Elasticsearch实例的Kibana控制台。

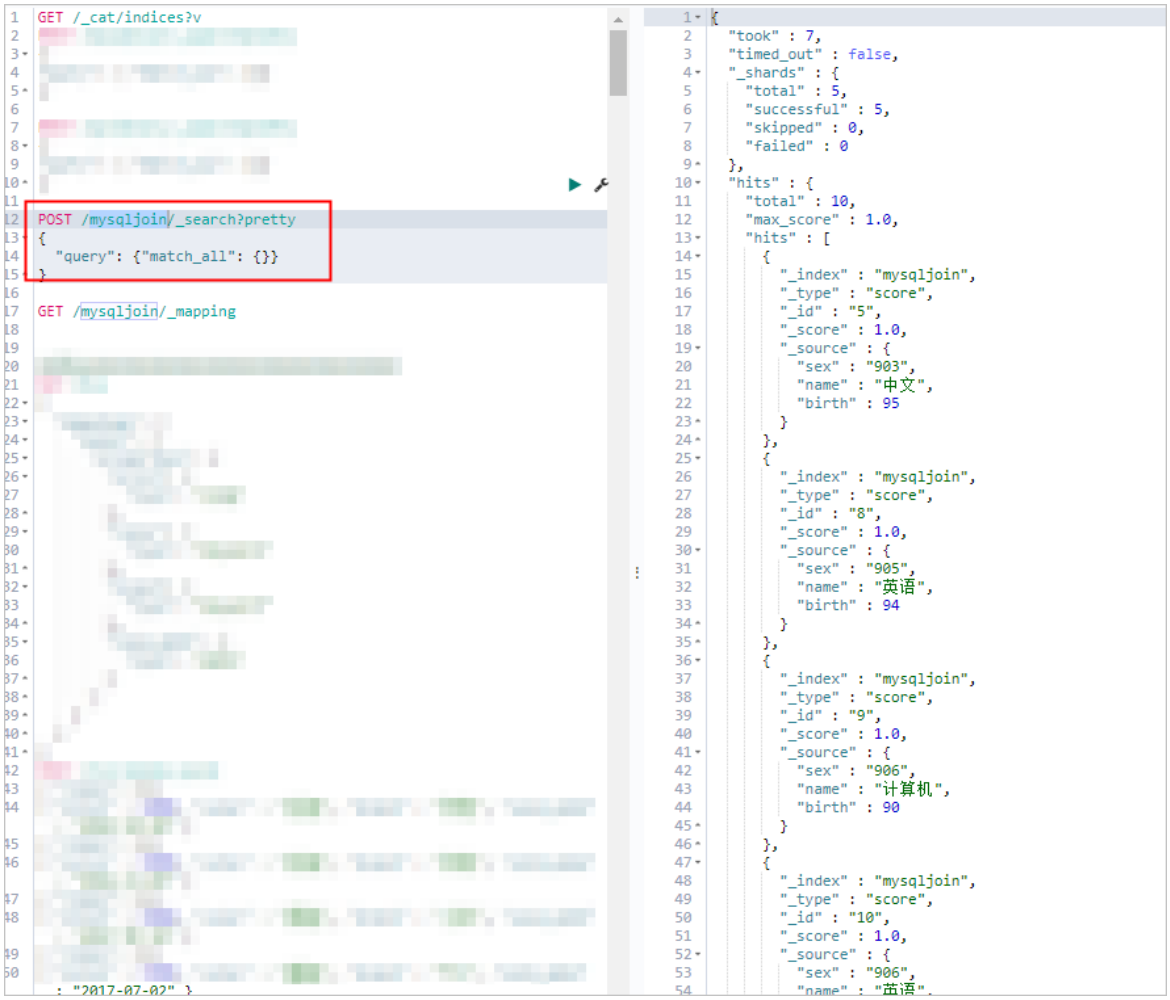
   具体操作，请参见登录Kibana控制台。

2. 在左侧导航栏，单击Dev Tools（开发工具）。
3. 在Console中，执行如下命令查看同步的数据。

```
POST /mysqljoin/_search?pretty
{
"query": { "match_all": {}}
}
```

> **说明** `mysqljoin` 为您在数据同步脚本中设置的 `index` 字段的值。

数据同步成功后，返回如下结果。



3.1.4. 通过DTS将MySQL数据实时同步到阿里云Elasticsearch

DTS实时同步MySQL数据到阿里云ES

当您需要将企业线上的RDS MySQL中的生产数据实时同步到阿里云Elasticsearch中进行搜索分析时，可通过数据传输服务DTS（Data Transmission Service），快速创建RDS MySQL到阿里云Elasticsearch的实时同步作业。本文介绍如何配置RDS MySQL到阿里云Elasticsearch的实时同步作业，并验证全量和增量数据的同步结果。

背景信息

数据传输服务DTS是一种集数据迁移、数据订阅及数据实时同步于一体的数据传输服务，详细信息请参见数据传输服务DTS。DTS支持同步的SQL操作包括Insert、Delete和Update，支持同步的数据源版本要求请参见同步方案概览。

本文适用的场景：对实时同步要求较高的关系型数据库中数据的同步场景。

使用限制

通过DTS将数据同步至阿里云Elasticsearch，不支持7.16版本的Elasticsearch实例。

注意事项

- DTS在执行全量数据初始化时将占用源库和目标库一定的读写资源，可能会导致数据库的负载上升，在数据库性能较差、规格较低或业务量较大的情况下（例如源库有大量慢SQL、存在无主键表或目标库存在死锁等），可能会加重数据库压力，甚至导致数据库服务不可用。因此您需要在执行数据同步前评估源库和目标库的性能，同时建议您在业务低峰期执行数据同步（例如源库和目标库的CPU负载在30%以下）。
- DTS不支持同步DDL操作，如果源库中待同步的表在同步的过程中已经执行了DDL操作，您需要先移除同步对象，然后在Elasticsearch实例中移除该表对应的索引，最后新增同步对象。详情请参见移除同步对象和新增同步对象。
- 如果源库中待同步的表需要执行增加列的操作，您只需先在Elasticsearch实例中修改对应表的mapping，然后在源库中执行相应的DDL操作，最后暂停并启动DTS同步实例即可。

操作流程

1. 步骤一：环境准备
2. 步骤二：配置数据同步链路
3. 步骤三：验证数据同步结果

步骤一：环境准备

1. 创建阿里云Elasticsearch实例，并开启实例的自动创建索引功能。

   具体操作步骤请参见步骤一：创建实例和配置YML参数。本文使用6.7版本的实例。

   > **说明** 阿里云Elasticsearch为了保证用户操作数据的安全性，默认把自动创建索引的配置设置为不允许。通过DTS同步数据，使用的是提交数据的方式创建索引，而不是Create index API方式。所以在同步数据前，需要先开启集群的自动创建索引功能。

2. 创建一个数据库和表，并插入数据。

   您可以选择使用阿里云的RDS数据库，也可以在本地服务器上的自建数据库。本文以RDS MySQL数据库为例，创建RDS MySQL数据库及表的详细步骤请参见快速入门。

   > **注意** 建议您创建与阿里云Elasticsearch实例同一地域下的RDS MySQL实例，不同地域的同步任务不能确保互通。

   本文使用的建表语句及数据如下：

   - 建表语句

     ```
     CREATE TABLE `es_test` (
         `id` bigint(32) NOT NULL,
         `name` varchar(32) NULL,
         `age` bigint(32) NULL,
         `hobby` varchar(32) NULL,
         PRIMARY KEY (`id`)
     ) ENGINE=InnoDB
     DEFAULT CHARACTER SET=utf8;
     ```

   - 表数据

id	name	age	hobby
1	user1	15	music
2	user2	18	game
3	user3	20	sport
4	user4	17	run
5	user5	21	basketball

步骤二：配置数据同步链路

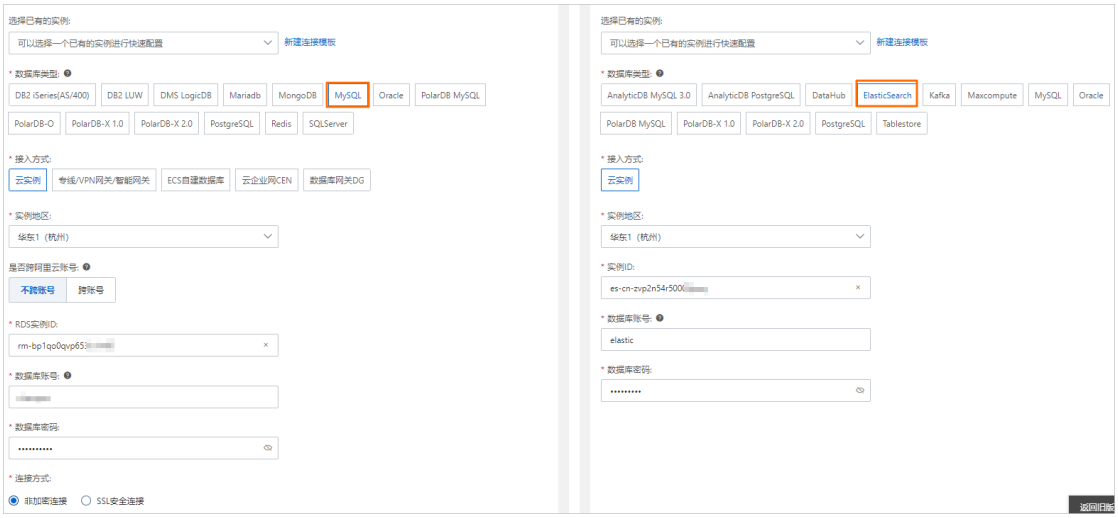
1. 登录数据传输控制台。
2. 在左侧导航栏，选择**数据传输（DTS）> 数据同步**。

   > **说明** 本文操作以DTS新版控制台为例，旧版控制台相关操作请参见数据同步操作指导。

3. 单击**创建任务**，按照页面提示创建并配置数据同步任务。

   您需要依次完成源库及目标库配置、任务对象配置、映射字段配置、高级配置和库表字段配置，本文使用的配置及相关说明如下，更多详细信息请参见MySQL为源的数据同步和PolarDB-X同步至Elasticsearch。

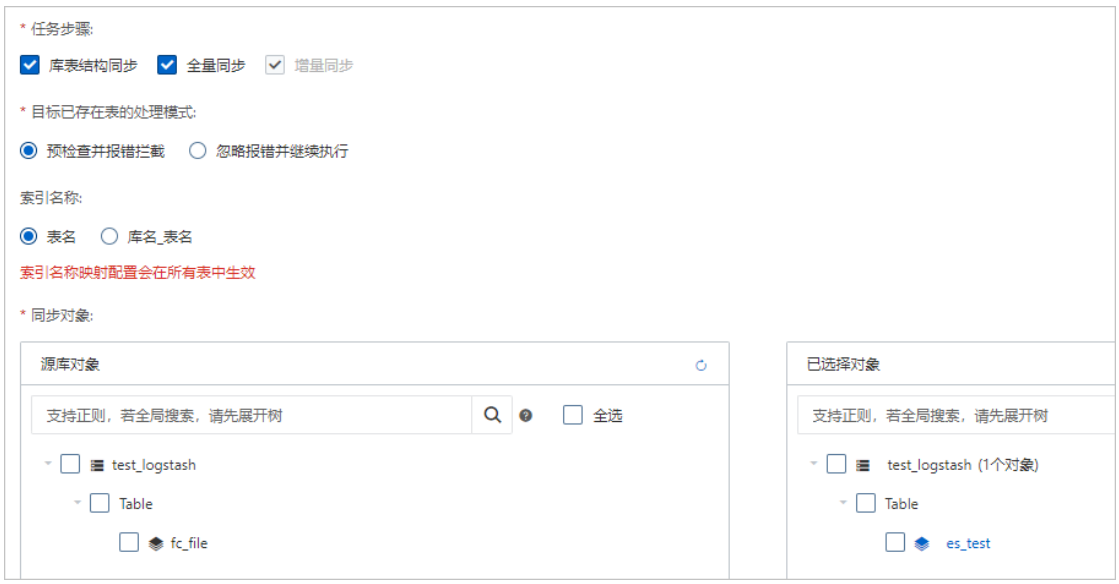
   i. 配置源库及目标库。



类别	配置	说明
无	**任务名称**	■ DTS为每个任务自动生成一个同步作业名称，该名称没有唯一性要求。 ■ 建议配置具有业务意义的名称，便于后续的识别。
	数据库类型	选择**MySQL**。
	接入方式	选择**云实例**。

类别	配置	说明
源库信息	实例地区	选择源MySQL数据库所属地域。
	是否跨阿里云账号	本场景为同一阿里云账号间同步数据，选择**不跨账号**。
	实例ID	选择源RDS MySQL实例ID。
	数据库账号	填入源RDS MySQL实例的数据库账号，需具备待同步对象的读权限。
	数据库密码	填入**数据库账号**对应的密码。
	连接方式	根据需求选择**非加密连接**或**SSL安全连接**。如果设置为**SSL安全连接**，您需要提前开启RDS MySQL实例的SSL加密功能，详情请参见设置SSL加密。
目标库信息	数据库类型	选择**ElasticSearch**。
	接入方式	固定为**云实例**。
	实例地区	选择目标Elasticsearch实例所属地域，建议与源MySQL数据库保持一致。
	实例ID	选择目标Elasticsearch实例ID。
	数据库账号	填入连接Elasticsearch实例的账号，默认账号为elastic。
	数据库密码	填入**数据库账号**对应的密码。

ii. 配置任务对象。

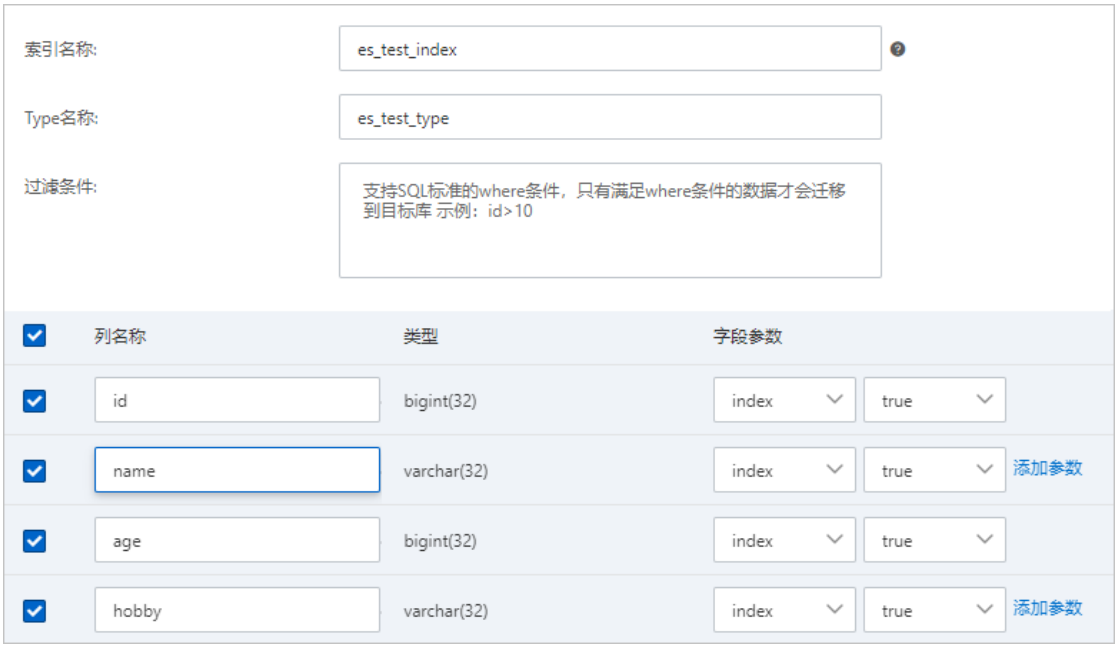


配置	说明

配置	说明
任务步骤	固定选中**增量同步**。默认情况下，您还需要同时选中**库表结构同步**和**全量同步**。预检查完成后，DTS会将源实例中待同步对象的全量数据在目标集群中初始化，作为后续增量同步数据的基线数据。
目标已存在表的处理模式	■ **预检查并报错拦截**：检查目标数据库中是否有同名的表。如果目标数据库中没有同名的表，则通过该检查项目；如果目标数据库中有同名的表，则在预检查阶段提示错误，数据同步任务不会被启动。 **说明** 如果目标库中同名的表不方便删除或重命名，您可以更改该表在目标库中的名称，请参见库表列名映射。 ■ **忽略报错并继续执行**：跳过目标数据库中是否有同名表的检查项。 **警告** 选择为**忽略报错并继续执行**，可能导致数据不一致，给业务带来风险，例如： ■ 表结构一致的情况下，如在目标库遇到与源库主键的值相同的记录： ■ 全量期间，DTS会保留目标集群中的该条记录，即源库中的该条记录不会同步至目标数据库中。 ■ 增量期间，DTS不会保留目标集群中的该条记录，即源库中的该条记录会覆盖至目标数据库中。 ■ 表结构不一致的情况下，可能会导致无法初始化数据、只能同步部分列的数据或同步失败。
索引名称	■ **表名** 选择为**表名**后，在目标Elasticsearch实例中创建的索引名称和表名一致，在本案例中即为es_test。 ■ **库名_表名** 选择为**库名_表名**后，在目标Elasticsearch实例中创建的索引名称为**库名_表名**，在本案例中即为test_logstash_es_test。
同步对象	在**源库对象**框中单击待同步对象，然后单击 › 将其移动至**已选择对象**框。
映射名称更改	■ 如需更改单个同步对象在目标实例中的名称，请右击**已选择对象**中的同步对象，设置方式，请参见库表列名单个映射。 ■ 如需批量更改同步对象在目标实例中的名称，请单击**已选择对象**方框右上方的**批量编辑**，设置方式，请参见库表列名批量映射。

iii. 配置映射字段，修改同步后的字段名称。

如果您需要修改同步后的字段名称，可在**已选择对象**区域框中，右键单击对应的表名，设置该表在目标Elasticsearch实例中的索引名称、Type名称等信息，然后单击**确定**。本文使用的配置及相关说明如下，未提及的配置保持默认，更多详细信息请参见库表列名单个映射。



配置	说明
索引名称	自定义索引名称，详情请参见基本概念。 **注意** 输入索引名称时，请确保Elasticsearch集群中不存在同名索引，否则报错 `index already exists` 。
Type名称	自定义索引类型名称，详情请参见基本概念。
过滤条件	您可以设置SQL过滤条件，过滤待同步的数据，只有满足过滤条件的数据才会被同步到目标实例，详情请参见通过SQL条件过滤任务数据。
字段参数	选择所需的**字段参数**和**字段参数值**，字段参数及取值介绍请参见Mapping parameters。 **注意** 如果添加参数时，将对应参数的index值设置为false，那么该字段将不能被查询，详情请参见index。

iv. 配置高级参数。

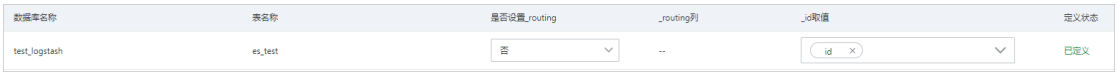
本文使用默认配置，相关说明如下。

配置	说明

配置	说明
设置告警	是否设置告警，当同步失败或延迟超过阈值后，将通知告警联系人。 ■ **不设置**：不设置告警。 ■ **设置**：设置告警，您还需要设置告警阈值和告警联系人。
源库、目标库无法连接后的重试时间	在同步任务连接失败时，DTS会立即进行持续的重试连接，默认持续重试时间为120分钟，您也可以在取值范围（10~1440分钟）内自定义重试时间，建议设置30分钟以上。如果DTS在设置的重试时间内重新连接上源库、目标库，同步任务将自动恢复。否则，同步任务将失败。 **说明** ■ 针对同源或者同目标的多个DTS实例，如DTS实例A和DTS实例B，设置网络重试时间时A设置30分钟，B设置60分钟，则重试时间以低的30分钟为准。 ■ 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。
分片配置	根据目标Elasticsearch中索引的分片配置，设置索引的主分片和副本分片的数量。Elasticsearch 7.x以下版本的索引默认包含5个主shard，1个副shard。Elasticsearch 7.x及以上版本的索引默认包含1个主shard，1个副shard。 **注意** shard大小和数量是影响Elasticsearch集群稳定性和性能的重要因素，您需要设置合理的shard数，shard的评估方法请参见规格容量评估。
字符串Index	同步至目标Elasticsearch实例中的字符串编入索引的方式： ■ analyzed：先分析字符串，再写入索引。您还需要选择具体的分析器，分析器的类型及作用，请参见分析器。 ■ not analyzed：不分析，直接使用原始值写入索引。 ■ no：不写入索引。
时区	DTS同步时间类型的数据（如DATETIME、TIMESTAMP）至目标Elasticsearch实例时，您可以选择所带时区。 **说明** 如目标实例中此类时间类型数据无需带有时区，则在同步前您需在目标实例中设置该时间类型数据的文档类型（type）。
DOCID取值	DOCID默认为表的主键，如表中无主键，则DOCID为Elasticsearch自动生成的ID列。

v. 配置库表字段，设置待同步的表在目标Elasticsearch的_routing策略和_id取值。

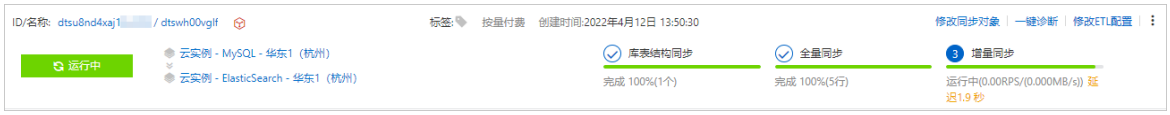
本文使用的配置及相关说明如下。



类型	说明
设置_routing	设置_routing可以将文档路由存储在目标Elasticsearch实例的指定分片上，请参见_routing。 ■ 选择为**是**，您可以自定义列进行路由。 ■ 选择为**否**，则用_id进行路由。 **说明** 创建的目标Elasticsearch实例为7.x版本时，您必须选择为**否**。
_id取值	■ **表的主键列** 联合主键合并为一列。 ■ **业务主键** 如果选择为**业务主键**，那么您还需要设置对应的**业务主键列**。

4. 配置完成后，根据页面提示保存任务、进行预检查、购买并启动任务。

购买成功后，同步任务正式开始，您可在数据同步界面查看具体任务进度。待全量同步完成，增量同步进行中时，您即可在Elasticsearch中查看同步成功的数据。



注意 由于MySQL和Elasticsearch实例支持的数据类型不同，数据类型无法一一对应。所以DTS在进行结构初始化时，会根据目标库支持的数据类型进行类型映射，详情请参见结构初始化涉及的数据类型映射关系。

步骤三：验证数据同步结果

1. 登录目标阿里云Elasticsearch实例的Kibana控制台，根据页面提示进入Kibana主页。

登录Kibana控制台的具体操作，请参见登录Kibana控制台。

说明 本文以阿里云Elasticsearch 6.7.0版本为例，其他版本操作可能略有差别，请以实际界面为准。

2. 在左侧导航栏，单击**Dev Tools**。
3. 在**Console**中，执行如下命令查看全量数据同步结果。

```
GET /es_test_index/es_test_type/_search
```

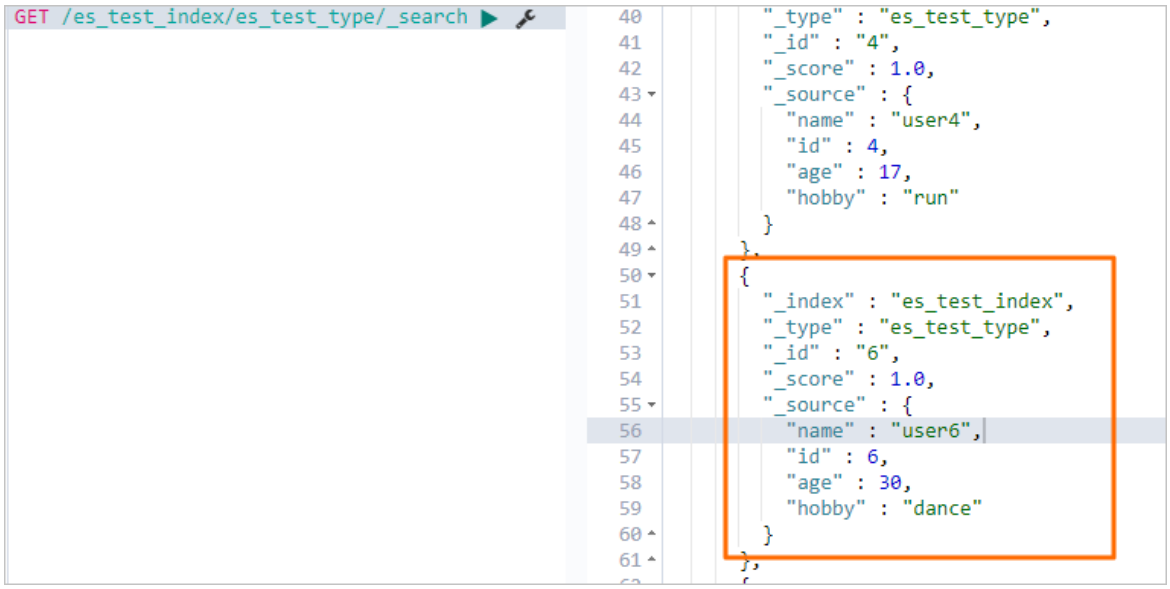
预期结果如下。

4. 在MySQL中插入一条数据，在Elasticsearch中查看增量数据同步结果。

   例如通过以下SQL语句插入一条数据。

```
INSERT INTO `test_logstash`.`es_test` (`id`,`name`,`age`,`hobby`) VALUES (6,'user6',30,'dance');
```

在Elasticsearch中查看结果，预期结果如下。



3.1.5. 通过Canal将MySQL数据同步到阿里云Elasticsearch

Canal将MySQL数据同步至es

Canal是阿里巴巴集团提供的一个开源产品，能够通过解析数据库的增量日志，提供增量数据的订阅和消费功能。当您需要将MySQL中的增量数据同步至阿里云Elasticsearch时，可通过Canal来实现。本文以阿里云RDS MySQL为例，介绍具体的实现方法。

背景信息

Canal是Github中开源的ETL（Extract Transform Load）软件，其功能原理及详细说明请参见Canal。

操作流程

1. 准备工作

   创建同一专有网络下的RDS MySQL实例、阿里云Elasticsearch实例和ECS实例，了解Canal。各模块的作用如下：

   - RDS MySQL：存放源数据和增量数据。
   - Canal：解析数据库日志，同步获取到的增量变更。
   - 阿里云Elasticsearch：接收增量数据。
   - 阿里云ECS：部署Canal-server和Canal-adapter。

2. 步骤一：准备MySQL数据源

   在RDS MySQL中，准备待同步的数据。

3. 步骤二：创建索引

   在阿里云Elasticsearch实例中，创建索引。要求Mapping中定义的字段名称和类型与待同步数据保持一致。

4. 步骤三：安装JDK

   在使用Canal前，必须先安装JDK，要求版本大于等于1.8.0。

5. 步骤四：安装并启动Canal-server

   安装Canal-server，然后修改配置文件关联RDS MySQL。Canal-server模拟MySQL集群的一个slave，获取MySQL集群Master节点的二进制日志（binary log），并将日志推送给Canal-adapter。

6. 步骤五：安装并启动Canal-adapter

   安装Canal-adapter，然后修改配置文件关联RDS MySQL和Elasticsearch，以及定义MySQL数据到Elasticsearch数据的映射字段，用来将数据同步到Elasticsearch。

7. 步骤六：验证增量数据同步

   在RDS MySQL中新增、修改或删除数据，查看数据同步结果。

准备工作

- 创建RDS MySQL实例。

  具体操作，请参见创建RDS MySQL实例。本文使用的配置如下。

- 创建阿里云Elasticsearch实例。

  具体操作，请参见创建阿里云Elasticsearch实例。本文创建的实例的**版本**为**通用商业版6.7**。

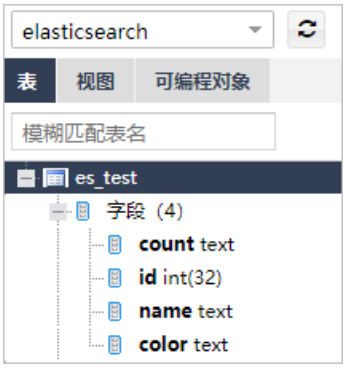
- 创建阿里云ECS实例。

  具体操作，请参见使用向导创建实例。本文创建的实例的**镜像**为**CentOS 7.6 64位**。

步骤一：准备MySQL数据源

进入RDS控制台，创建RDS MySQL数据库和表。具体操作，请参见RDS MySQL快速入门。

本文创建的表名称为**es_test**，包含的字段如下所示。



步骤二：创建索引

1. 登录目标阿里云Elasticsearch实例的Kibana控制台，根据页面提示进入Kibana主页。

   登录Kibana控制台的具体操作，请参见登录Kibana控制台。

   > **说明** 本文以阿里云Elasticsearch 6.7.0版本为例，其他版本操作可能略有差别，请以实际界面为准。

2. 在左侧导航栏，单击**Dev Tools**。
3. 在**Console**中，执行以下命令创建索引。

   以下示例创建的索引名称为es_test，包含count、id、name和color字段。

> 注意 mappings中的字段需要与步骤一：准备MySQL数据源中创建的字段（名称和类型）保持一致。

```
PUT es_test?include_type_name=true
{
    "settings" : {
      "index" : {
        "number_of_shards" : "5",
        "number_of_replicas" : "1"
      }
    },
    "mappings" : {
        "_doc" : {
            "properties" : {
              "count": {
                   "type": "text"
               },
              "id": {
                   "type": "integer"
               },
               "name": {
                    "type" : "text",
                    "analyzer": "ik_smart"
                },
                "color" : {
                    "type" : "text"
                }
            }
        }
    }
}
```

创建成功后，返回如下结果。

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : "es_test"
}
```

步骤三：安装JDK

1. 参见连接ECS实例，连接阿里云ECS实例，查看可用的JDK软件包列表。

   ```
   yum search java | grep -i --color JDK
   ```

2. 选择合适的版本，安装JDK。

   本文选择java-1.8.0-openjdk-devel.x86_64。

   ```
   yum install java-1.8.0-openjdk-devel.x86_64
   ```

3. 配置环境变量。

i. 打开etc文件夹下的profile文件。

```
vi /etc/profile
```

ii. 在文件内添加如下的环境变量。

```
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.71-2.b15.el7_2.x86_64
export CLASSPATH=.:$JAVA_HOME/jre/lib/rt.jar:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib
/tools.jar
export PATH=$PATH:$JAVA_HOME/bin
```

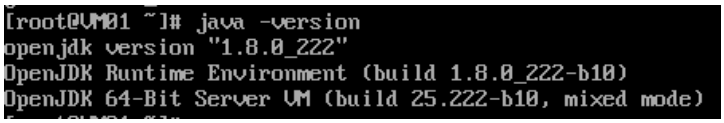
> **注意** JAVA_HOME需要替换为您JDK的安装路径，可通过 `find / -name 'java'` 命令查看。

iii. 按下Esc键，然后使用 `:wq` 保存文件并退出vi模式，随后执行以下命令使配置生效。

```
source /etc/profile
```

4. 执行以下任意命令，验证JDK是否安装成功。
   - `java`
   - `javac`
   - `java -version`

   显示如下结果说明JDK安装成功。



步骤四：安装并启动Canal-server

1. 下载Canal-server。

   本文使用1.1.4版本。

```
wget https://github.com/alibaba/canal/releases/download/canal-1.1.4/canal.deployer-1.
1.4.tar.gz
```

> **说明** 目前Canal 1.1.5版本已支持Elasticsearch 7.0版本，如果您使用的是Elasticsearch 7.0，需要下载Canal 1.1.5版本。详细信息请参见Canal release note。

2. 解压。

```
tar -zxvf canal.deployer-1.1.4.tar.gz
```

3. 修改 `conf/example/instance.properties` 文件。

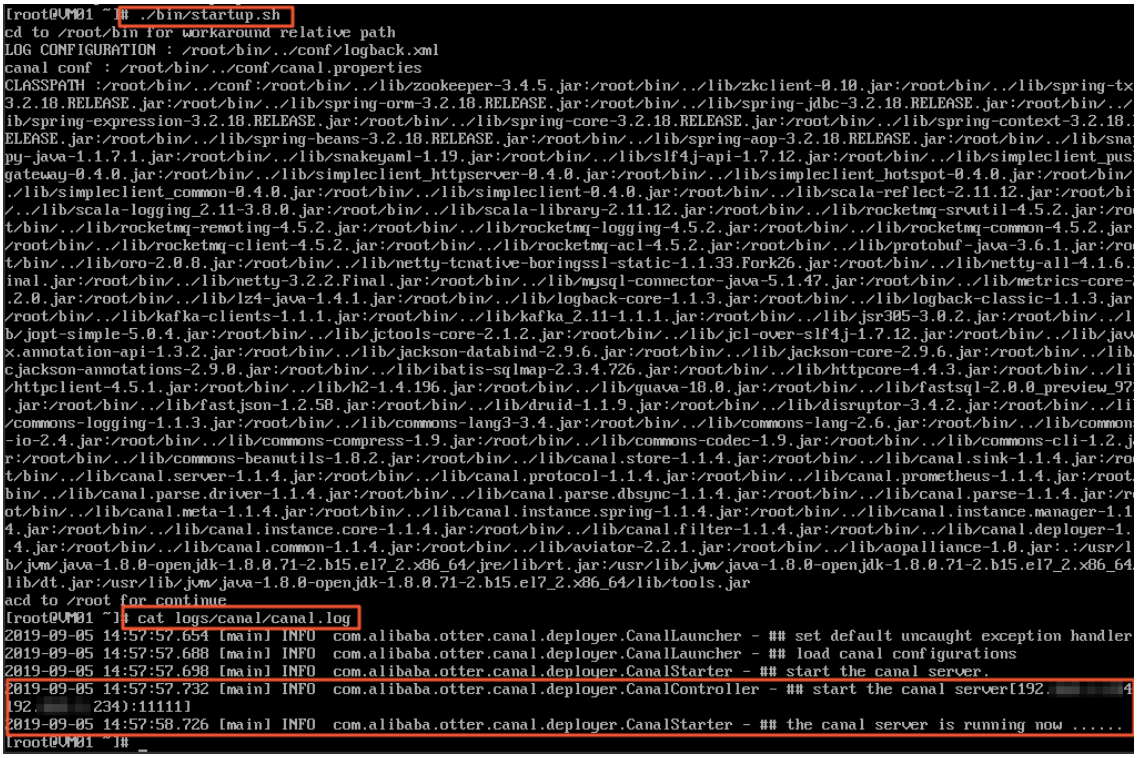
```
vi conf/example/instance.properties
```

```
#################################################
## mysql serverId , v1.0.26+ will autoGen
# canal.instance.mysql.slaveId=0

# enable gtid use true/false
canal.instance.gtidon=false

# position info
canal.instance.master.address=rm-bp[redacted]d6ph.mysql.rds.aliyuncs.com:3306
canal.instance.master.journal.name=
canal.instance.master.position=
canal.instance.master.timestamp=
canal.instance.master.gtid=

# rds oss binlog
canal.instance.rds.accesskey=
canal.instance.rds.secretkey=
canal.instance.rds.instanceId=

# table meta tsdb info
canal.instance.tsdb.enable=true
#canal.instance.tsdb.url=jdbc:mysql://127.0.0.1:3306/canal_tsdb
#canal.instance.tsdb.dbUsername=canal
#canal.instance.tsdb.dbPassword=canal

#canal.instance.standby.address =
#canal.instance.standby.journal.name =
#canal.instance.standby.position =
#canal.instance.standby.timestamp =
#canal.instance.standby.gtid=

# username/password
canal.instance.dbUsername=[redacted]
canal.instance.dbPassword=[redacted]
canal.instance.connectionCharset = UTF-8
# enable druid Decrypt database password
canal.instance.enableDruid=false
```

配置项	说明
canal.instance.master.address	需要设置为<RDS MySQL数据库的内网地址>:<内网端口>，相关信息可在RDS MySQL实例的**基本信息**页面获取。例如rm-bp1u1xxxxxxxxx6ph.mysql.rds.aliyuncs.com:3306。
canal.instance.dbUsername	RDS MySQL数据库的账号名称，可在实例的**账号管理**页面获取。
canal.instance.dbPassword	RDS MySQL数据库的密码。

4. 按下Esc键，然后使用 `:wq` 命令保存文件并退出vi模式。
5. 启动Canal-server，并查看日志。

```
./bin/startup.sh
cat logs/canal/canal.log
```



步骤五：安装并启动Canal-adapter

1. 下载Canal-adapter。

   本文使用1.1.4版本。

   ```
   wget https://github.com/alibaba/canal/releases/download/canal-1.1.4/canal.adapter-1.1
   .4.tar.gz
   ```

   > **说明** 目前Canal 1.1.5版本已支持Elasticsearch 7.0版本，如果您使用的是Elasticsearch 7.0，需要下载Canal 1.1.5版本。详细信息请参见Canal release note。

2. 解压。

   ```
   tar -zxvf canal.adapter-1.1.4.tar.gz
   ```

3. 修改 `conf/application.yml` 文件。

   ```
   vi conf/application.yml
   ```

```
canal.conf:
  mode: tcp # kafka rocketMQ
  canalServerHost: 127.0.0.1:11111
#  zookeeperHosts: slave1:2181
#  mqServers: 127.0.0.1:9092 #or rocketmq
#  flatMessage: true
  batchSize: 500
  syncBatchSize: 1000
  retries: 0
  timeout:
  accessKey:
  secretKey:
  srcDataSources:
    defaultDS:
      url: jdbc:mysql://rm-bp[redacted]ph.mysql.rds.aliyuncs.com:3306/elasticsearch?useUnicode=true
      username: [redacted]
      password: p[redacted]
  canalAdapters:
  - instance: example # canal instance Name or mq topic name
    groups:
    - groupId: g1
      outerAdapters:
      - name: logger
#      - name: rdb
#        key: mysql1
#        properties:
#          jdbc.driverClassName: com.mysql.jdbc.Driver
#          jdbc.url: jdbc:mysql://127.0.0.1:3306/mytest2?useUnicode=true
#          jdbc.username: root
#          jdbc.password: 121212
#      - name: rdb
#        key: oracle1
#        properties:
#          jdbc.driverClassName: oracle.jdbc.OracleDriver
#          jdbc.url: jdbc:oracle:thin:@localhost:49161:XE
#          jdbc.username: mytest
#          jdbc.password: m121212
#      - name: rdb
#        key: postgres1
#        properties:
#          jdbc.driverClassName: org.postgresql.Driver
#          jdbc.url: jdbc:postgresql://localhost:5432/postgres
#          jdbc.username: postgres
#          jdbc.password: 121212
#          threads: 1
#          commitSize: 3000
#      - name: hbase
#        properties:
#          hbase.zookeeper.quorum: 127.0.0.1
#          hbase.zookeeper.property.clientPort: 2181
#          zookeeper.znode.parent: /hbase
      - name: es
        hosts: es-cn-v6[redacted]dp.elasticsearch.aliyuncs.com:9200
        properties:
          mode: rest # or transport
          security.auth: elastic:[redacted]3
          cluster.name: es-cn-v6[redacted]dp
```

配置项	说明
canal.conf.canalServerHost	canalDeployer访问地址。保持默认（127.0.0.1:11111）即可。
canal.conf.srcDataSources.defaultDS.url	需要设置为jdbc:mysql://<RDS MySQL内网地址>:<内网端口>/<数据库名称>?useUnicode=true，相关信息可在RDS MySQL实例的**基本信息**页面获取。例如jdbc:mysql://rm-bp1xxxxxxxxnd6ph.mysql.rds.aliyuncs.com:3306/elasticsearch?useUnicode=true。
canal.conf.srcDataSources.defaultDS.username	RDS MySQL数据库的账号名称，可在RDS MySQL实例的**账号管理**页面获取。
canal.conf.srcDataSources.defaultDS.password	RDS MySQL数据库的密码。

配置项	说明
canal.conf.canalAdapters.groups.outerAdapters.hosts	定位到name:es的位置，将hosts替换为<Elasticsearch实例的内网地址>:<内网端口>，相关信息可在Elasticsearch实例的基本信息页面获取。例如es-cn-v64xxxxxxxx3medp.elasticsearch.aliyuncs.com:9200。
canal.conf.canalAdapters.groups.outerAdapters.mode	必须设置为rest。
canal.conf.canalAdapters.groups.outerAdapters.properties.security.auth	需要设置为<Elasticsearch实例的账号>:<密码>。例如elastic:es_password。
canal.conf.canalAdapters.groups.outerAdapters.properties.cluster.name	Elasticsearch实例的ID，可在实例的基本信息页面获取。例如es-cn-v64xxxxxxxx3medp。

4. 按下Esc键，然后使用 `:wq` 命令保存文件并退出vi模式。
5. 同样的方式，修改 `conf/es/*.yml` 文件，定义MySQL数据到Elasticsearch数据的映射字段。

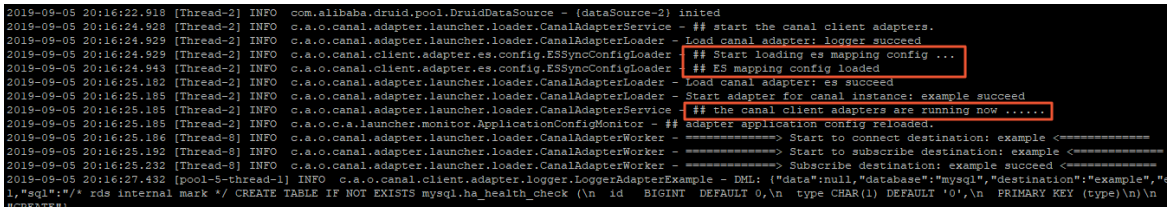
```
dataSourceKey: defaultDS
destination: example
groupId: g1
esMapping:
  _index: es_test
  _type: _doc
  _id: _id
  #pk: id
  sql: "select t.id as _id,t.id,t.count,t.name,t.color from es_test t"
  commitBatch: 3000
~
```

配置项	说明
esMapping._index	步骤二：创建索引章节中，在Elasticsearch实例中所创建的索引的名称。本文使用**es_test**。
esMapping._type	步骤二：创建索引章节中，在Elasticsearch实例中所创建的索引的类型。本文使用**_doc**。
esMapping._id	需要同步到Elasticsearch实例的文档的id，可自定义。本文使用**_id**。
esMapping.sql	SQL语句，用来查询需要同步到Elasticsearch中的字段。本文使用 `select t.id as _id,t.id,t.count,t.name,t.color from es_test t` 。

6. 启动Canal-adapter服务，并查看日志。

```
./bin/startup.sh
cat logs/adapter/adapter.log
```

服务启动正常时，结果如下所示。



步骤六：验证增量数据同步

1. 在RDS MySQL数据库中，新增、修改或删除数据库中es_test表的数据。

```
insert `elasticsearch`.`es_test`(`count`,`id`,`name`,`color`) values('11',2,'canal_te
st2','red');
```

2. 登录目标阿里云Elasticsearch实例的Kibana控制台，根据页面提示进入Kibana主页。

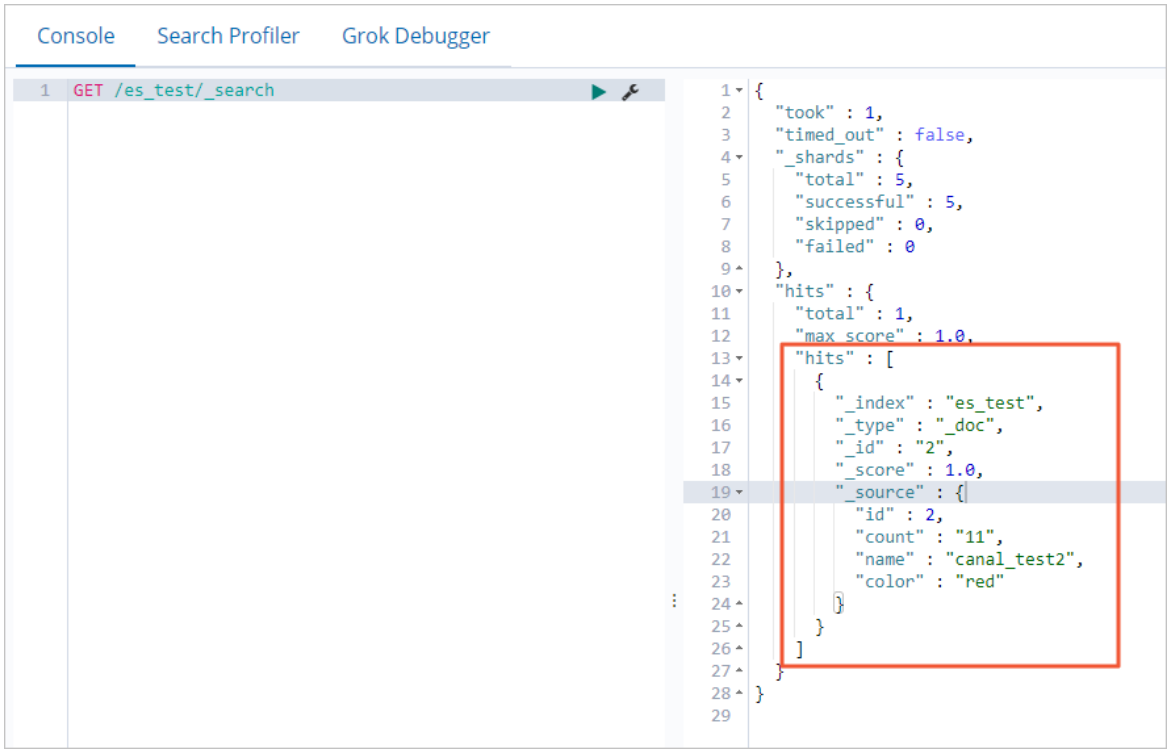
   登录Kibana控制台的具体操作，请参见登录Kibana控制台。

   > 说明 本文以阿里云Elasticsearch 6.7.0版本为例，其他版本操作可能略有差别，请以实际界面为准。

3. 在左侧导航栏，单击Dev Tools。
4. 在Console中，执行以下命令查询同步成功的数据。

```
GET /es_test/_search
```

预期结果如下。



3.2. PolarDB-X（DRDS）同步

3.2.1. PolarDB-X（DRDS）同步方案选取指南

当您在使用PolarDB-X（DRDS），需要进行全文检索和语义分析时，可将PolarDB-X中的数据同步至阿里云Elasticsearch进行查询分析。阿里云Elasticsearch是一个基于Lucene的实时分布式的搜索与分析引擎，可近乎于准实时地存储、查询和分析超大数据集。您可以通过Logstash和DataWorks两种方式将PolarDB-X中的数据同步至阿里云Elasticsearch。本文介绍各同步方案适用的场景，帮助您根据业务选择合适的方案同步数据。

同步方案	原理说明	适用场景	使用限制	相关文档
Logstash JDBC数据同步	通过logstash-input-jdbc插件实现通过Logstash批量查询PolarDB-X中的数据，并将数据迁移到Elasticsearch。实现的本质是该插件会定期对PolarDB-X中的数据进行循环轮询，从而在当前循环中找到上次插入或更改的记录，然后批量查询这些记录并迁移至Elasticsearch。	• 同步全量数据，接受秒级延迟的场景。 • 批量查询数据然后进行同步的场景。	• 使用前，需要先在Logstash中上传与PolarDB-X版本兼容的SQL JDBC驱动。 • 需要在PolarDB-X的白名单中加入Logstash集群中节点的IP地址。 • 需要确保Logstash和PolarDB-X实例在同一时区（避免同步过程中出现时间标记不符的情况）。 • 需要确保Elasticsearch中的_id字段与PolarDB-X中的_id字段相同。 • 当您在PolarDB-X中插入或更新数据时，需要确保对应记录有一个包含更新或插入时间的字段。	通过Logstash将PolarDB-X（DRDS）数据同步至Elasticsearch
DataWorks实现离线数据同步	DataWorks是一款提供数据集成、数据开发及数据质量等全方位的产品服务。支持引入并存储关系型数据，然后进行转化和开发，最后将处理后的数据同步到Elasticsearch或其他数据系统。	• 大数据离线同步场景（可实现最快5分钟一次的离线数据采集任务）。 • 需要自定义查询语句，以及多表联合查询后同步数据的场景。 • 同步整个数据库中数据的场景。	• 需要开通DataWorks服务。 • 对于传输速度要求较高或复杂环境中的数据源同步场景，需要自定义资源组。 • 需要在PolarDB-X的白名单中添加资源组的IP地址。	通过DataWorks将PolarDB-X（DRDS）数据离线同步至Elasticsearch

3.2.2. 通过Logstash将PolarDB-X（DRDS）数据同步至Elasticsearch

当您的业务数据存储在PolarDB-X（原DRDS升级版）中，需要进行全文检索和语义分析时，可借助阿里云Elasticsearch实现。本文介绍如何通过阿里云Logstash，将PolarDB-X中的数据实时同步至阿里云Elasticsearch。

背景信息

PolarDB-X是由阿里巴巴自主研发的云原生分布式数据库，融合分布式SQL引擎DRDS与分布式自研存储X-DB，基于云原生一体化架构设计，可支撑千万级并发规模及百PB级海量存储。专注解决海量数据存储、超高并发吞吐、大表瓶颈以及复杂计算效率等数据库瓶颈问题，历经各届天猫双十一及阿里云各行业客户业务的考验，助力企业加速完成业务数字化转型。详细信息，请参见PolarDB-X产品概述。

阿里云Elasticsearch兼容开源Elasticsearch的功能，以及Security、Machine Learning、Graph、APM等商业功能，致力于数据分析、数据搜索等场景服务。支持5.5.3、6.3.2、6.7.0、6.8.0、7.4.0和7.7.1等版本，并提供了商业插件X-Pack服务。在开源Elasticsearch的基础上提供企业级权限管控、安全监控告警、自动报表生成等功能。详细信息，请参见什么是阿里云Elasticsearch。

阿里云Logstash作为服务器端的数据处理管道，提供了100%兼容开源Logstash的能力。Logstash能够动态地从多个来源采集数据、转换数据，并且将数据存储到所选择的位置。通过输入、过滤和输出插件，Logstash可以对任何类型的事件加工和转换。阿里云Logstash除了支持所有官方预置插件外，还致力于打造包含logstash-input-sls、logstash-input-oss、logstash-output-oss等适用各类场景的插件中心，为您提供更为强大的数据处理和搬迁能力，实现云上数据生态打通。详细信息，请参见什么是阿里云Logstash。

使用限制

通过阿里云Logstash实现PolarDB-X和Elasticsearch数据同步，本质上是使用Logstash的logstash-input-jdbc插件（默认已安装，不可卸载），通过管道配置功能进行数据同步。该插件会定期对PolarDB-X数据库中的数据进行循环轮询，从而在当前循环中找到上次插入或更改的记录。因此要让同步任务正确运行，Elasticsearch和PolarDB-X数据库必须满足以下条件：

- Elasticsearch中的_id字段必须与数据库中的id字段相同。

  该条件可以确保当您将数据库中的记录写入Elasticsearch时，同步任务可在数据库记录与Elasticsearch文档之间建立一个直接映射的关系。例如当您在数据库中更新了某条记录时，同步任务会覆盖Elasticsearch中与更新记录具有相同ID的文档。

  > **说明** 根据Elasticsearch内部原理，更新操作的本质是删除旧文档然后对新文档进行索引，因此在Elasticsearch中覆盖文档的效率与更新操作的效率一样高。

- 当您在数据库中插入或者更新数据时，对应记录必须有一个包含更新或插入时间的字段。

  Logstash每次对数据库进行轮询时，都会保存其从数据库中所读取的最后一条记录的更新或插入时间。在读取数据时，Logstash仅读取符合条件的记录，即该记录的更新或插入时间需要晚于上一次轮询中最后一条记录的更新或插入时间。

  > **注意** logstash-input-jdbc插件无法实现同步删除。如果您要删除Elasticsearch中的数据，需要在Elasticsearch中执行相关命令，手动删除。

准备工作

1. 创建PolarDB-X 1.0实例、数据库、表，并准备测试数据。

   具体操作，请参见PolarDB-X 1.0快速入门。

   本文使用的建表语句如下。

```
CREATE table food(
id int PRIMARY key AUTO_INCREMENT,
name VARCHAR (32),
insert_time DATETIME,
update_time DATETIME );
```

插入数据语句如下。

```
INSERT INTO food values(null,'巧克力',now(),now());
INSERT INTO food values(null,'酸奶',now(),now());
INSERT INTO food values(null,'火腿肠',now(),now());
```

2. 创建阿里云Elasticsearch实例，并开启自动创建索引功能。

   创建实例时，所选专有网络要与PolarDB-X 1.0实例一致。具体操作，请参见创建阿里云Elasticsearch实例和快速访问与配置。本文使用6.7.0版本的实例。

3. 创建阿里云Logstash实例，并上传与PolarDB-X数据库版本兼容的SQL JDBC驱动。

   创建时所选专有网络和版本要与目标Elasticsearch实例一致。具体操作，请参见创建阿里云Logstash实例和配置扩展文件。本文使用mysql-connector-java-5.1.35.jar驱动。

   > **说明**
   > - 您也可以使用公网环境的服务，前提是需要通过配置NAT网关实现与公网的连通。详细信息，请参见配置NAT公网数据传输。
   > - 本文使用MySQL JDBC驱动连接PolarDB-X数据库。您也可以使用PolarDB JDBC驱动，但对于一些高版本PolarDB-X数据库，使用PolarDB JDBC驱动会有问题，建议您使用MySQL JDBC驱动。

4. 在PolarDB-X数据库白名单中加入阿里云Logstash节点的IP地址（可在基本信息页面获取）。

   具体操作，请参见设置白名单。

配置Logstash管道

1. 登录阿里云Elasticsearch控制台。
2. 进入目标实例。
   i. 在顶部菜单栏处，选择地域。
   ii. 在左侧导航栏，单击**Logstash实例**，然后在**Logstash实例**中单击目标实例ID。
3. 单击左侧导航栏的**管道管理**。
4. 单击**创建管道**。
5. 在**创建管道任务**页面，输入**管道ID**，并进行Config配置。

   本文使用的Config配置如下。

```
input {
  jdbc {
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    jdbc_driver_library => "/ssd/1/share/<Logstash实例ID>/logstash/current/config/custom/mysql-connector-java-5.1.35.jar"
    jdbc_connection_string => "jdbc:mysql://drdshbga51x6****.drds.aliyuncs.com:3306/<数据库名称>?useUnicode=true&characterEncoding=utf-8&useSSL=false&allowLoadLocalInfile=false&autoDeserialize=false"
    jdbc_user => "db_user"
    jdbc_password => "db_password"
    jdbc_paging_enabled => "true"
    jdbc_page_size => "50000"
    statement => "select * from food where update_time >= :sql_last_value"
    schedule => "* * * * *"
    record_last_run => true
    last_run_metadata_path => "/ssd/1/<Logstash实例ID>/logstash/data/last_run_metadata_update_time.txt"
    clean_run => false
    tracking_column_type => "timestamp"
    use_column_value => true
    tracking_column => "update_time"
  }
}
filter {
}
output {
  elasticsearch {
    hosts => "http://es-cn-n6w1o1x0w001c****.elasticsearch.aliyuncs.com:9200"
    user => "elastic"
    password => "es_password"
    index => "drds_test"
    document_id => "%{id}"
  }
}
```

说明 代码中 `<Logstash实例ID>` 需要替换为您创建的Logstash实例的ID。获取方式，请参见实例列表概览。

input参数说明

参数	描述
jdbc_driver_class	JDBC Class配置。
jdbc_driver_library	指定JDBC连接MySQL驱动文件。具体操作请参见配置扩展文件。
jdbc_connection_string	配置数据库连接的域名、端口及数据库。
jdbc_user	数据库用户名。
jdbc_password	数据库密码。

参数	描述
jdbc_paging_enabled	是否启用分页，默认false。
jdbc_page_size	分页大小。
statement	指定SQL语句。
schedule	指定定时操作，`"* * * * *"`表示每分钟定时同步数据。
record_last_run	是否记录上次执行结果。如果为true，则会把上次执行到的tracking_column字段的值记录下来，保存到last_run_metadata_path指定的文件中。
last_run_metadata_path	指定最后运行时间文件存放的地址。目前后端开放了*/ssd/1/<Logstash实例ID>/logstash/data/*路径来保存文件。
clean_run	是否清除last_run_metadata_path的记录，默认为false。如果为true，那么每次都要从头开始查询所有的数据库记录。
use_column_value	是否需要记录某个column的值。
tracking_column_type	跟踪列的类型，默认是numeric。
tracking_column	指定跟踪列，该列必须是递增的，一般是表的主键。

output参数说明

参数	说明
hosts	阿里云Elasticsearch实例的访问地址，格式为http://<实例的内网地址>:9200。其中实例的内网地址可在基本信息页面获取，详细信息请参见查看实例的基本信息。
user	访问阿里云Elasticsearch实例的用户名，默认为elastic。
password	对应用户的密码。elastic用户的密码在创建实例时设定，如果忘记可重置。重置密码的注意事项和操作，请参见重置实例访问密码。
index	索引名称。
document_id	文档ID。设置为%{id}，表示与源数据库中的ID字段保持一致。

> **注意**
>
> - 以上配置按照测试数据配置，在实际业务中，请按照业务需求进行合理配置。input插件支持的其他配置选项，请参见官方Logstash Jdbc input plugin
> - 如果配置中有类似 `last_run_metadata_path` 的参数，需要阿里云Logstash服务提供文件路径。目前后端开放了 `/ssd/1/<Logstash实例ID>/logstash/data/` 路径供您测试使用，且该目录下的数据不会被删除。因此在使用时，请确保磁盘有充足的使用空间。
> - 为了提升安全性，如果在配置管道时使用了JDBC驱动，需要在 `jdbc_connection_string` 参数后面添加 `allowLoadLocalInfile=false&autoDeserialize=false` ，否则当您在添加Logstash配置文件时，调度系统会抛出校验失败的提示，例如 `jdbc_connection_string => "jdbc:mysql://drdshbga51x6****.drds.aliyuncs.com:3306/<数据库名称>?allowLoadLocalInfile=false&autoDeserialize=false"` 。

更多Config配置，请参见Logstash配置文件说明。

6. 单击**下一步**，配置管道参数。

参数	说明
管道工作线程	并行执行管道的Filter和Output的工作线程数量。当事件出现积压或CPU未饱和时，请考虑增大线程数，更好地使用CPU处理能力。默认值：实例的CPU核数。
管道批大小	单个工作线程在尝试执行Filter和Output前，可以从Input收集的最大事件数目。较大的管道批大小可能会带来较大的内存开销。您可以设置LS_HEAP_SIZE变量，来增大JVM堆大小，从而有效使用该值。默认值：125。
管道批延迟	创建管道事件批时，将过小的批分派给管道工作线程之前，要等候每个事件的时长，单位为毫秒。默认值：50ms。
队列类型	用于事件缓冲的内部排队模型。可选值： - **MEMORY**：默认值。基于内存的传统队列。 - **PERSISTED**：基于磁盘的ACKed队列（持久队列）。
队列最大字节数	请确保该值小于您的磁盘总容量。默认值：1024 MB。
队列检查点写入数	启用持久性队列时，在强制执行检查点之前已写入事件的最大数目。设置为0，表示无限制。默认值：1024。

> 警告 配置完成后，需要保存并部署才能生效。保存并部署操作会触发实例重启，请在不影响业务的前提下，继续执行以下步骤。

7. 单击**保存**或者**保存并部署**。
   - **保存**：将管道信息保存在Logstash里并触发实例变更，配置不会生效。保存后，系统会返回**管道管理**页面。可在**管道列表**区域，单击**操作**列下的**立即部署**，触发实例重启，使配置生效。
   - **保存并部署**：保存并且部署后，会触发实例重启，使配置生效。

验证结果

1. 登录目标阿里云Elasticsearch实例的Kibana控制台。

   具体操作，请参见登录Kibana控制台。
2. 在左侧导航栏，单击**Dev Tools**（开发工具）。
3. 在**Console**中，执行以下命令，查看同步成功的索引数量。

```
GET drds_test/_count
{
  "query": {"match_all": {}}
}
```

   运行成功后，结果如下。

```
{
  "count" : 3,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  }
}
```

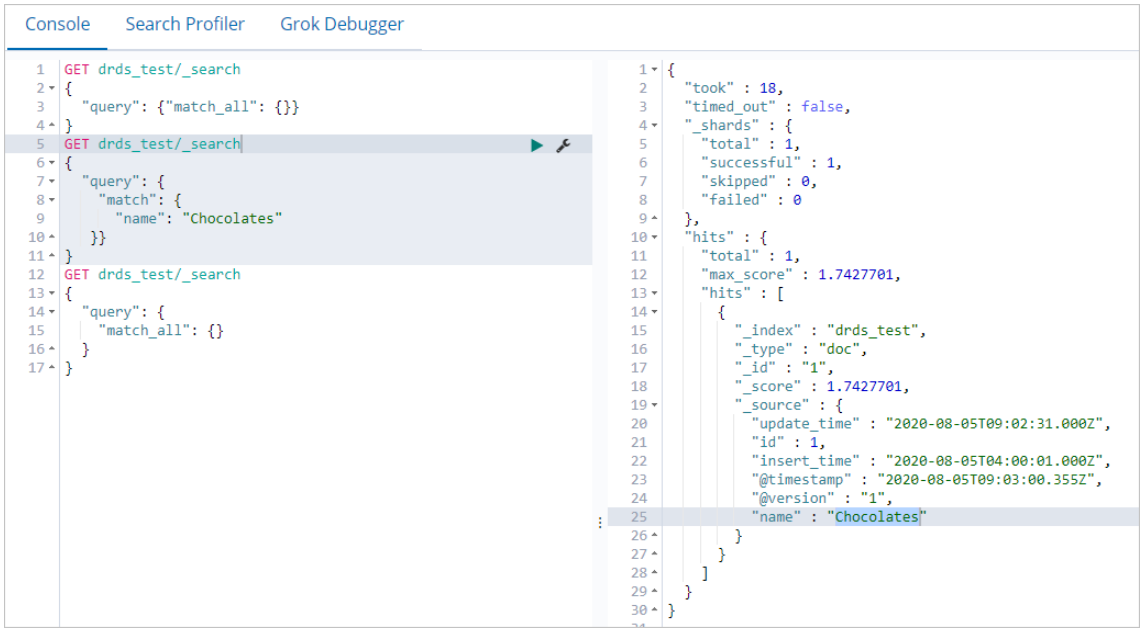
4. 在表中更新并插入数据。

```
UPDATE food SET name='Chocolates',update_time=now() where id = 1;
INSERT INTO food values(null,'鸡蛋',now(),now());
```

5. 在Kibana控制台，查看更新后的数据。
   - 查询name为Chocolates的数据。

```
GET drds_test/_search
{
  "query": {
    "match": {
      "name": "Chocolates"
    }}
}
```

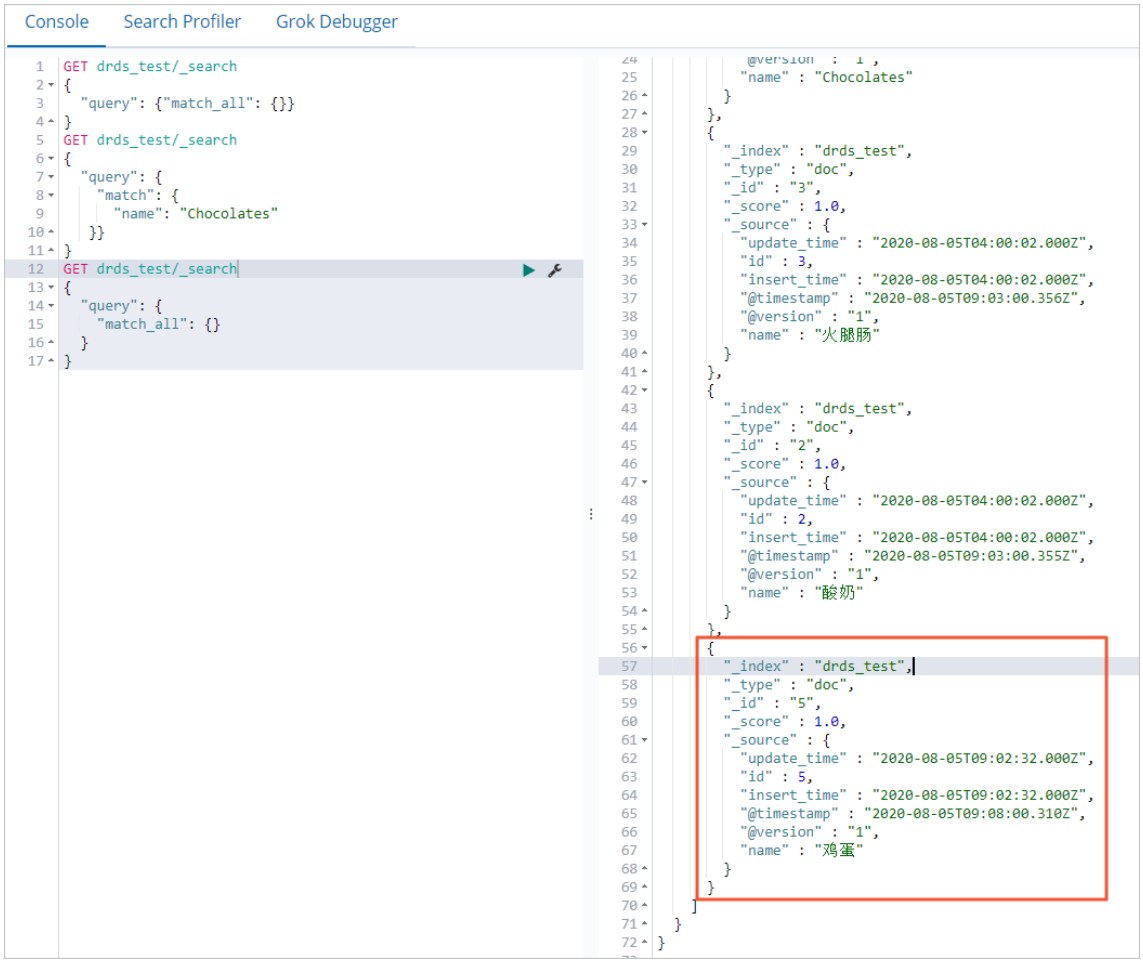
返回结果如下。



- 查询所有数据。

```
GET drds_test/_search
{
  "query": {
    "match_all": {}
  }
}
```

返回结果如下。



常见问题

Logstash数据写入问题排查方案

3.2.3. 通过DataWorks将PolarDB-X（DRDS）数据离线同步至Elasticsearch

当您的业务数据存储在PolarDB-X（原DRDS升级版）中，需要进行全文检索和语义分析时，可借助阿里云Elasticsearch实现。本文介绍如何通过DataWorks，将PolarDB-X中的数据离线同步至Elasticsearch，并进行检索分析。

背景信息

PolarDB-X是由阿里巴巴自主研发的云原生分布式数据库，融合分布式SQL引擎DRDS与分布式自研存储X-DB，基于云原生一体化架构设计，可支撑千万级并发规模及百PB级海量存储。专注解决海量数据存储、超高并发吞吐、大表瓶颈以及复杂计算效率等数据库瓶颈问题，历经各届天猫双十一及阿里云各行业客户业务的考验，助力企业加速完成业务数字化转型，详情请参见PolarDB-X产品概述。

阿里云Elasticsearch兼容开源Elasticsearch的功能，以及Security、Machine Learning、Graph、APM等商业功能，致力于数据分析、数据搜索等场景服务。支持5.5.3、6.3.2、6.7.0、6.8.0、7.4.0和7.7.1等版本，并提供了商业插件X-Pack服务。在开源Elasticsearch的基础上提供企业级权限管控、安全监控告警、自动报表生成等功能，详情请参见什么是阿里云Elasticsearch。

操作流程

1. 准备工作

   准备同一专有网络VPC（Virtual Private Cloud）下的阿里云PolarDB-X实例和Elasticsearch实例、在PolarDB-X实例中准备待同步的数据、开通DataWorks的数据集成和数据开发服务。

   > **说明** 建议您在同一VPC下进行数据同步，这样可以提高同步任务的稳定性。

2. 步骤一：购买并创建独享资源组

   在DataWorks中，购买并创建独享资源组。为确保网络互通，您还需要为独享资源组绑定PolarDB-X实例所在的专有网络。

   > **说明** 独享资源组可以保障数据快速、稳定地传输。

3. 步骤二：添加数据源

   在DataWorks中，创建DRDS和Elasticsearch数据源。

4. 步骤三：配置并运行数据同步任务

   通过向导模式配置数据同步任务，将数据集成系统同步成功的数据存储到Elasticsearch中。将独享资源组作为一个可以执行任务的资源，注册到DataWorks的数据集成服务中。这个资源组将获取数据源的数据，并执行将数据写入Elasticsearch中的任务（该任务将由数据集成系统统一下发）。

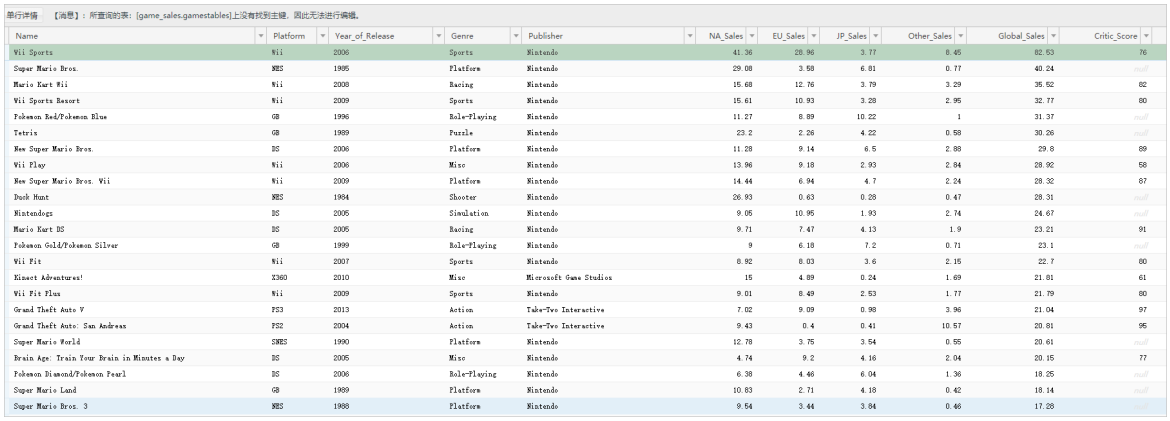
5. 步骤四：查看数据同步结果

   在Elasticsearch实例的Kibana控制台中，查看同步成功的数据量，并对指定字段进行数据检索。

准备工作

1. 创建阿里云PolarDB-X 1.0实例、构建数据库和表，并插入数据。

   具体操作步骤请参见PolarDB-X快速入门。本文使用的测试数据如下。



Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Critic_Score
Wii Sports	Wii	2006	Sports	Nintendo	41.36	28.96	3.77	8.45	82.53	76
Super Mario Bros.	NES	1985	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24	null
Mario Kart Wii	Wii	2008	Racing	Nintendo	15.68	12.76	3.79	3.29	35.52	82
Wii Sports Resort	Wii	2009	Sports	Nintendo	15.61	10.93	3.28	2.95	32.77	80
Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	Nintendo	11.27	8.89	10.22	1	31.37	null
Tetris	GB	1989	Puzzle	Nintendo	23.2	2.26	4.22	0.58	30.26	null
New Super Mario Bros.	DS	2006	Platform	Nintendo	11.28	9.14	6.5	2.88	29.8	89
Wii Play	Wii	2006	Misc	Nintendo	13.96	9.18	2.93	2.84	28.92	58
New Super Mario Bros. Wii	Wii	2009	Platform	Nintendo	14.44	6.94	4.7	2.24	28.32	87
Duck Hunt	NES	1984	Shooter	Nintendo	26.93	0.63	0.28	0.47	28.31	null
Nintendogs	DS	2005	Simulation	Nintendo	9.05	10.95	1.93	2.74	24.67	null
Mario Kart DS	DS	2005	Racing	Nintendo	9.71	7.47	4.13	1.9	23.21	91
Pokemon Gold/Pokemon Silver	GB	1999	Role-Playing	Nintendo	9	6.18	7.2	0.71	23.1	null
Wii Fit	Wii	2007	Sports	Nintendo	8.92	8.03	3.6	2.15	22.7	80
Kinect Adventures!	X360	2010	Misc	Microsoft Game Studios	15	4.89	0.24	1.69	21.81	61
Wii Fit Plus	Wii	2009	Sports	Nintendo	9.01	8.49	2.53	1.77	21.79	80
Grand Theft Auto V	PS3	2013	Action	Take-Two Interactive	7.02	9.09	0.98	3.96	21.04	97
Grand Theft Auto: San Andreas	PS2	2004	Action	Take-Two Interactive	9.43	0.4	0.41	10.57	20.81	95
Super Mario World	SNES	1990	Platform	Nintendo	12.78	3.75	3.54	0.55	20.61	null
Brain Age: Train Your Brain in Minutes a Day	DS	2005	Misc	Nintendo	4.74	9.2	4.16	2.04	20.15	77
Pokemon Diamond/Pokemon Pearl	DS	2006	Role-Playing	Nintendo	6.38	4.46	6.04	1.36	18.25	null
Super Mario Land	GB	1989	Platform	Nintendo	10.83	2.71	4.18	0.42	18.14	null
Super Mario Bros. 3	NES	1988	Platform	Nintendo	9.54	3.44	3.84	0.46	17.28	null

   > **注意** 数据库创建完成后，默认允许所有IP地址访问。出于数据安全考虑，建议仅将待访问机器的IP地址加入白名单，详情请参见设置白名单。

2. 创建DataWorks工作空间。

   具体操作步骤请参见创建工作空间。创建时，所选地域需要与阿里云PolarDB-X实例所在地域保持一致。

3. 创建阿里云Elasticsearch实例，并开启自动创建索引功能。

   具体操作请参见创建阿里云Elasticsearch实例和快速访问与配置。创建实例时，所选专有网络和虚拟交

换机需要与PolarDB-X实例保持一致。

步骤一：购买并创建独享资源组

1. 登录DataWorks控制台。
2. 选择相应地域后，在左侧导航栏，单击**资源组列表**。
3. 参见购买资源组（创建订单），购买独享数据集成资源。

   > **注意** 购买时，所选地域需要与目标工作空间保持一致。

4. 参见新增和使用独享数据集成资源组，创建一个独享数据集成资源。

   本文使用的配置如下，其中**资源组类型**选择**独享数据集成资源组**。

5. 单击已创建的独享资源组右侧的**专有网络绑定**，参见绑定专有网络，为该独享资源组绑定专有网络。

   独享资源部署在DataWorks托管的专有网络中，需要与PolarDB-X和Elasticsearch实例的专有网络连通才能同步数据，因此在绑定专有网络时，需要选择PolarDB-X实例所在**专有网络**和**交换机**。

6. 单击已创建的独享资源组右侧的**修改归属工作空间**，参见新增和使用独享数据集成资源组，为该独享资源组绑定目标工作空间。

步骤二：添加数据源

1. 进入DataWorks的**数据集成**页面。
   i. 在DataWorks控制台的左侧导航栏，单击**工作空间列表**。
   ii. 找到目标工作空间，单击其右侧**操作**列下的**进入数据集成**。
2. 在左侧导航栏，单击**数据源**。
3. 在**数据源管理**页面，单击**新增数据源**。
4. 在**新增数据源**对话框中，单击**DRDS**。
5. 在**新增DRDS数据源**对话框中，填写数据源信息并测试连通性。连通成功后，单击**完成**。

参数	说明
数据源类型	本文使用**连接串模式**。您也可以选择**阿里云数据库（DRDS）**类型，详情请参见配置DRDS数据源。
数据源名称	必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
JDBC URL	JDBC连接信息，格式为jdbc:mysql://ServerIP:Port/Database。此处需要将ServerIP:Port替换为PolarDB-X实例的VPC地址：VPC端口；Database替换为您已创建的PolarDB-X数据库名称。 **说明** VPC地址和端口的获取方式请参见快速入门。
用户名	数据库对应的用户名。
密码	数据库对应的密码。

6. 使用同样的方式，添加Elasticsearch数据源。

参数	说明
数据源名称	必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
Endpoint	格式为http://<Elasticsearch实例的内网地址>:9200。可参见查看实例的基本信息，获取Elasticsearch实例的内网地址。
用户名	Elasticsearch实例的用户名，默认为**elastic**。
密码	对应用户的密码。**elastic**用户的密码在创建实例时设定，如果忘记可进行重置，重置密码的注意事项和操作步骤请参见重置实例访问密码。

步骤三：配置并运行数据同步任务

1. 在DataWorks的数据开发页面，新建一个业务流程。

   具体操作步骤请参见管理业务流程。

2. 新建一个离线同步任务。

   i. 展开新建的业务流程，右键单击**数据集成**，选择**新建 > 离线同步**。

ii. 在**新建节点**对话框中，输入**节点名称**，单击**提交**。

3. 在**选择数据源**区域中，将**数据来源**指定为DRDS数据源，并填入待同步的表名称；将**数据去向**指定为Elasticsearch数据源，并填入索引名和索引类型。

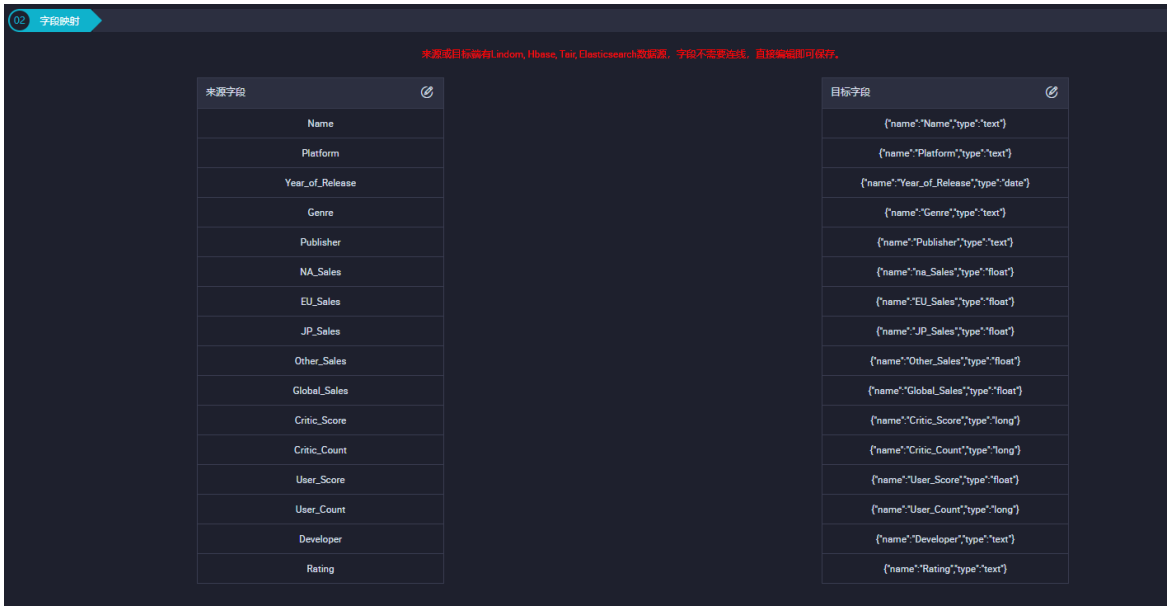
> **说明**
>
> - 您也可以使用脚本模式配置数据同步，详情请参见通过脚本模式配置离线同步任务、DRDS Reader和Elasticsearch Writer。
> - 建议在Elasticsearch数据源的**高级配置**下，将**启用节点发现**设置为**否**，否则同步过程中提示连接超时。

4. 在**字段映射**区域中，设置**来源字段**与**目标字段**的映射关系。

   本示例中，**来源字段**保持默认，仅修改**目标字段**。在**目标字段**右侧，单击图标，在对话框中输入如下字段配置。

```
{"name":"Name","type":"text"}
{"name":"Platform","type":"text"}
{"name":"Year_of_Release","type":"date"}
{"name":"Genre","type":"text"}
{"name":"Publisher","type":"text"}
{"name":"na_Sales","type":"float"}
{"name":"EU_Sales","type":"float"}
{"name":"JP_Sales","type":"float"}
{"name":"Other_Sales","type":"float"}
{"name":"Global_Sales","type":"float"}
{"name":"Critic_Score","type":"long"}
{"name":"Critic_Count","type":"long"}
{"name":"User_Score","type":"float"}
{"name":"User_Count","type":"long"}
{"name":"Developer","type":"text"}
{"name":"Rating","type":"text"}
```

配置完成后，效果如下。



5. 在**通道控制**区域中，配置执行任务的相关参数。
6. 配置任务调度属性。

   在页面右侧，单击**调度配置**，按照需求配置相应的调度参数。各配置的详细说明请参见调度配置章节。

   > 注意
   > - 在提交任务前，必须配置任务调度**依赖的上游节点**，详情请参见配置同周期调度依赖。
   > - 如果您希望对任务进行周期性调度，需要配置任务的**时间属性**，包括任务的具体执行时间、调度周期、生效周期、重跑属性等。
   > - 周期任务将于配置任务开始的第二天00:00，按照您的配置规则生效执行。

7. 配置执行同步任务所使用的资源组。

i. 在页面右侧，单击**数据集成资源组配置**。

ii. 选择**独享数据集成资源组**为您创建的独享资源组。

8. 提交任务。

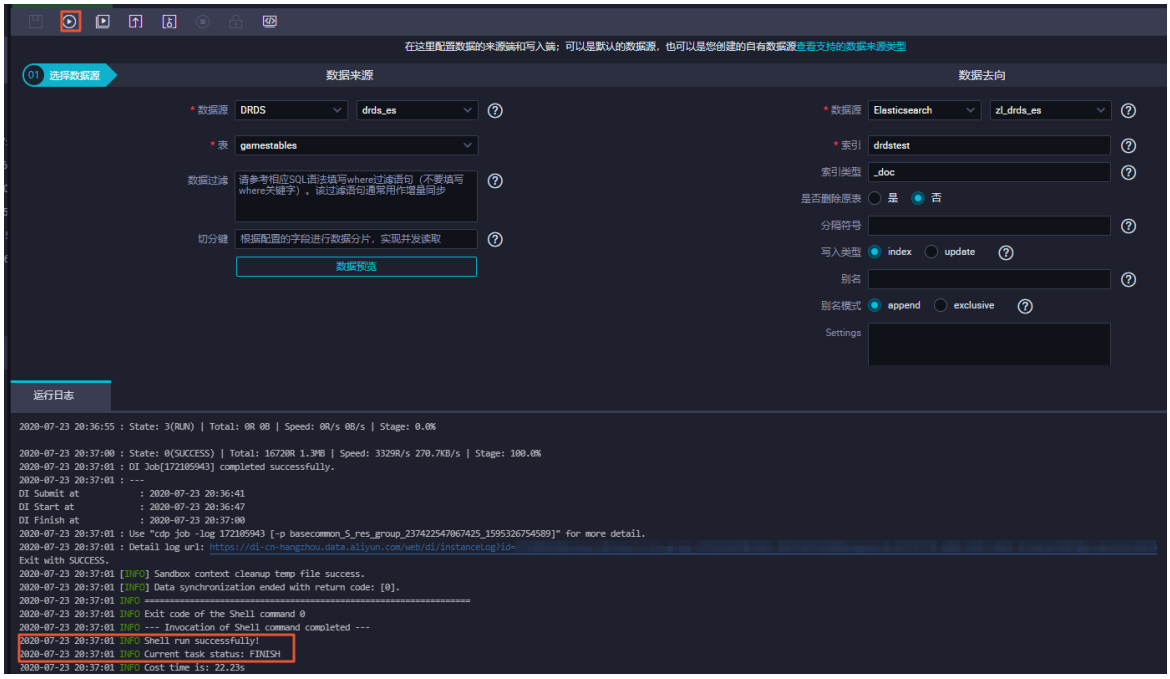
i. 保存当前配置，单击图标。

ii. 在**提交新版本**对话框中，填入**备注**。

iii. 单击**确认**。

9. 单击图标，运行任务。

任务运行过程中，可查看运行日志。日志输出**successfully**表示任务运行成功；输出**FINISH**表示任务运行完成。



> **说明** 在运行任务前，您还可以配置任务的调度属性和运行任务所使用的资源组，详细信息请参见调度配置章节和资源规划和配置。

步骤四：查看数据同步结果

1. 登录目标阿里云Elasticsearch实例的Kibana控制台。

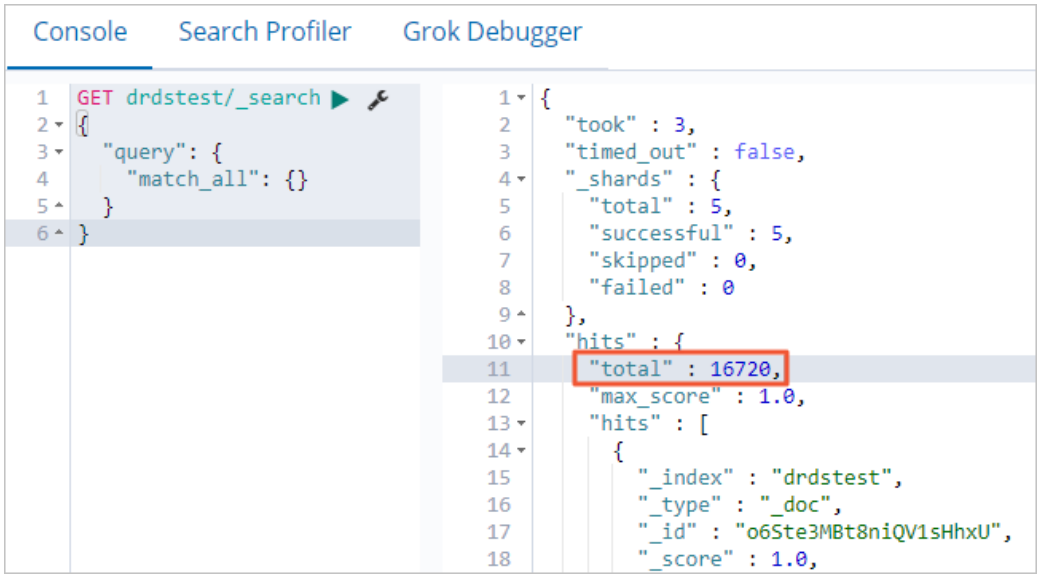
具体操作步骤请参见登录Kibana控制台。

2. 在左侧导航栏，单击**Dev Tools**。
3. 在**Console**中，执行如下命令，查询目标端数据量。

> **说明** 您可以将目标端数据量与源端数据量进行对比，确认数据是否全部同步成功。

```
GET drdstest/_search
{
  "query": {
    "match_all": {}
  }
}
```

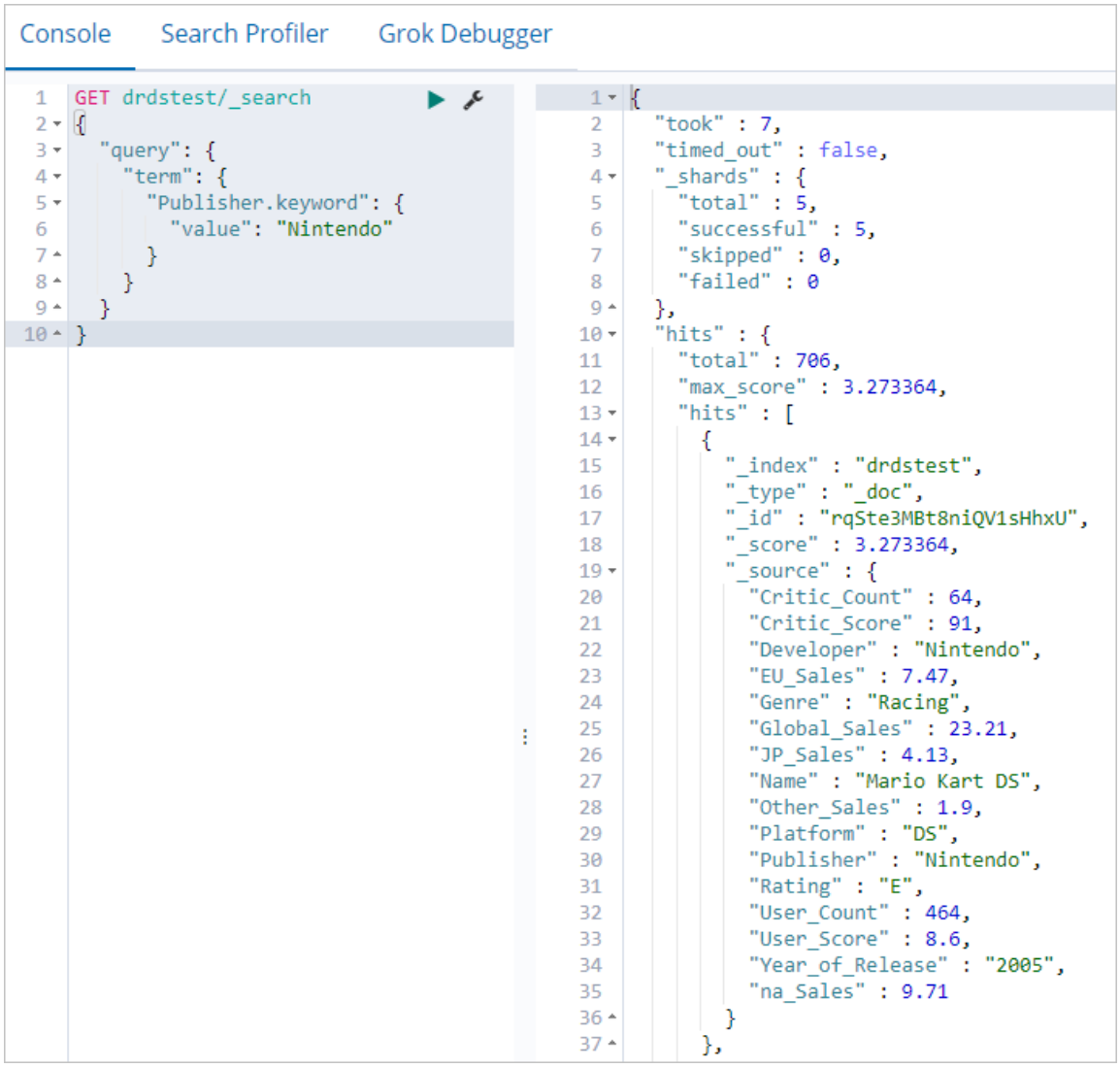
执行成功后，结果如下。



4. 执行如下命令，对指定字段进行数据检索。

```
GET drdstest/_search
{
  "query": {
    "term": {
      "Publisher.keyword": {
        "value": "Nintendo"
      }
    }
  }
}
```

执行成功后，返回如下结果。



3.3. 通过DTS将PolarDB MySQL数据同步至阿里云Elasticsearch

当您在使用PolarDB MySQL遇到查询慢的问题时，可以通过数据传输服务DTS（Data Transmission Service），将企业线上的PolarDB MySQL中的生产数据实时同步到阿里云Elasticsearch中进行搜索分析。本文介绍具体的实现方法。

背景信息

本案例需要使用以下三个云产品，相关介绍如下：

- 数据传输服务DTS是一种集数据迁移、数据订阅及数据实时同步于一体的数据传输服务，详情请参见数据传输服务DTS。DTS支持同步的SQL操作包括：Insert、Delete、Update。

  > 注意 进行数据同步时，请选择DTS支持的数据源及其版本，详情请参见同步方案概览。

- PolarDB是阿里云自研的下一代关系型云数据库，有三个独立的引擎，分别可以100%兼容MySQL、100%兼

容PostgreSQL、高度兼容Oracle语法。存储容量最高可达100TB，单库最多可扩展到16个节点，适用于企业多样化的数据库应用场景，详情请参见PolarDB MySQL概述。

- Elasticsearch是一个基于Lucene的实时分布式的搜索与分析引擎，它提供了一个分布式服务，可以使您快速的近乎于准实时的存储、查询和分析超大数据集，通常被用来作为构建复杂查询特性和需求强大应用的基础引擎或技术，详情请参见什么是阿里云Elasticsearch。

本文适用的场景：对实时同步要求较高的关系型数据库中数据的同步场景。

注意事项

- DTS在执行全量数据初始化时将占用源库和目标库一定的读写资源，可能会导致数据库的负载上升，在数据库性能较差、规格较低或业务量较大的情况下（例如源库有大量慢SQL、存在无主键表或目标库存在死锁等），可能会加重数据库压力，甚至导致数据库服务不可用。因此您需要在执行数据同步前评估源库和目标库的性能，同时建议您在业务低峰期执行数据同步（例如源库和目标库的CPU负载在30%以下）。
- DTS不支持同步DDL操作，如果源库中待同步的表在同步的过程中已经执行了DDL操作，您需要先移除同步对象，然后在Elasticsearch实例中移除该表对应的索引，最后新增同步对象。详情请参见移除同步对象和新增同步对象。
- 如果源库中待同步的表需要执行增加列的操作，您只需先在Elasticsearch实例中修改对应表的mapping，然后在源库中执行相应的DDL操作，最后暂停并启动DTS同步实例即可。

使用限制

通过DTS将数据同步至阿里云Elasticsearch，不支持7.16版本的Elasticsearch实例。

操作流程

1. 步骤一：环境准备

   完成创建Elasticsearch实例、创建PolarDB MySQL集群、准备测试数据等任务。

2. 步骤二：配置数据同步链路

   通过数据传输服务DTS，快速创建并启动PolarDB MySQL到阿里云Elasticsearch的实时同步作业。

3. 步骤三：查看数据同步结果

   在阿里云Elasticsearch的Kibana控制台中查看同步成功的数据。

4. 步骤四：验证增量数据同步

   验证在PolarDB MySQL数据库中新增数据时，数据的同步效果。

步骤一：环境准备

1. 创建阿里云Elasticsearch实例，并开启实例的自动创建索引功能。

   具体操作步骤请参见步骤一：创建实例和配置YML参数。本文使用6.7版本的实例。

   > **说明** 阿里云Elasticsearch为了保证用户操作数据的安全性，默认把自动创建索引的配置设置为不允许。通过DTS同步数据，使用的是提交数据的方式创建索引，而不是Create index API方式。所以在同步数据前，需要先开启集群的自动创建索引功能。

2. 创建云数据库PolarDB MySQL集群，并开启Binlog。

   具体操作步骤请参见购买按量付费集群、开启Binlog。

3. 创建PolarDB MySQL数据库和表，并插入测试数据。

   具体操作步骤请参见数据库管理。本文使用的表结构和测试数据如下。

- 建表语句

```
CREATE TABLE `product` (
    `id` bigint(32) NOT NULL AUTO_INCREMENT,
    `name` varchar(32) NULL,
    `price` varchar(32) NULL,
    `code` varchar(32) NULL,
    `color` varchar(32) NULL,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB
DEFAULT CHARACTER SET=utf8;
```

- 测试数据

```
INSERT INTO `estest`.`product` (`id`,`name`,`price`,`code`,`color`) VALUES (1,'mobile phone A','2000','amp','golden');
INSERT INTO `estest`.`product` (`id`,`name`,`price`,`code`,`color`) VALUES (2,'mobile phone B','2200','bmp','white');
INSERT INTO `estest`.`product` (`id`,`name`,`price`,`code`,`color`) VALUES (3,'mobile phone C','2600','cmp','black');
INSERT INTO `estest`.`product` (`id`,`name`,`price`,`code`,`color`) VALUES (4,'mobile phone D','2700','dmp','red');
INSERT INTO `estest`.`product` (`id`,`name`,`price`,`code`,`color`) VALUES (5,'mobile phone E','2800','emp','silvery');
```

步骤二：配置数据同步链路

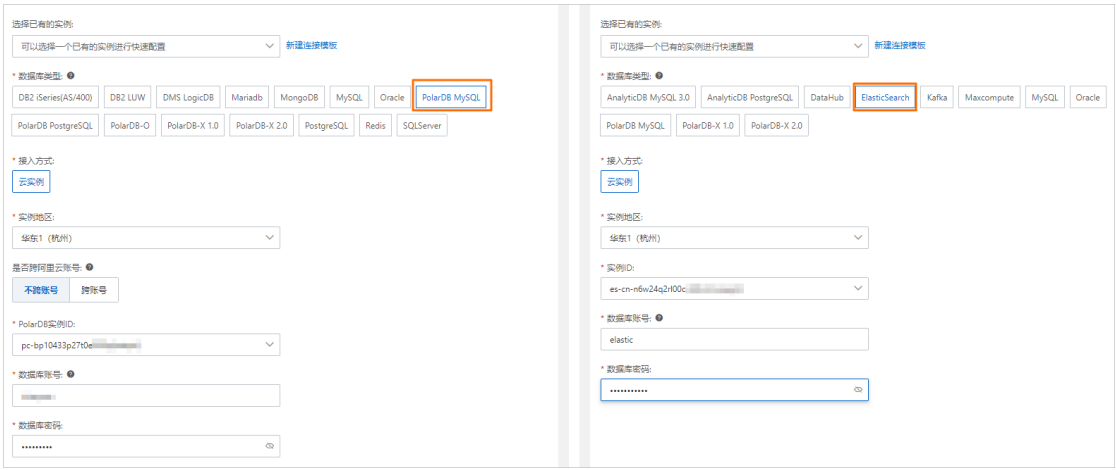
1. 登录数据传输控制台。
2. 在左侧导航栏，选择**数据传输（DTS）> 数据同步**。

> **说明** 本文操作以DTS新版控制台为例，旧版控制台相关操作请参见数据同步操作指导。

3. 单击**创建任务**，按照页面提示创建并配置数据同步任务。

   您需要依次完成源库及目标库配置、任务对象配置、映射字段配置、高级配置和库表字段配置，本文使用的配置及相关说明如下，更多详细信息请参见PolarDB MySQL为源的数据同步和PolarDB-X同步至Elasticsearch。

   i. 配置源库及目标库。



类别	配置	说明
无	**任务名称**	■ DTS为每个任务自动生成一个同步作业名称，该名称没有唯一性要求。 ■ 建议配置具有业务意义的名称，便于后续的识别。
源库信息	**数据库类型**	选择**PolarDB MySQL**。
	接入方式	选择**云实例**。
	实例地区	选择源PolarDB MySQL数据库所属地域。
	是否跨阿里云账号	本场景为同一阿里云账号间同步数据，选择**不跨账号**。
	实例ID	选择源PolarDB MySQL实例ID。
	数据库账号	填入源PolarDB MySQL实例的数据库账号，需具备待同步对象的读权限。
	数据库密码	填入**数据库账号**对应的密码。
目标库信息	**数据库类型**	选择**ElasticSearch**。
	接入方式	固定为**云实例**。
	实例地区	选择目标Elasticsearch实例所属地域，建议与源MySQL数据库保持一致。
	实例ID	选择目标Elasticsearch实例ID。
	数据库账号	填入连接Elasticsearch实例的账号，默认账号为elastic。
	数据库密码	填入**数据库账号**对应的密码。

ii. 配置任务对象。

配置	说明
任务步骤	固定选中**增量同步**。默认情况下，您还需要同时选中**库表结构同步**和**全量同步**。预检查完成后，DTS会将源实例中待同步对象的全量数据在目标集群中初始化，作为后续增量同步数据的基线数据。
目标已存在表的处理模式	■ **预检查并报错拦截**：检查目标数据库中是否有同名的表。如果目标数据库中没有同名的表，则通过该检查项目；如果目标数据库中有同名的表，则在预检查阶段提示错误，数据同步任务不会被启动。 **说明** 如果目标库中同名的表不方便删除或重命名，您可以更改该表在目标库中的名称，请参见库表列名映射。 ■ **忽略报错并继续执行**：跳过目标数据库中是否有同名表的检查项。 **警告** 选择为**忽略报错并继续执行**，可能导致数据不一致，给业务带来风险，例如： ■ 表结构一致的情况下，如在目标库遇到与源库主键的值相同的记录： ■ 全量期间，DTS会保留目标集群中的该条记录，即源库中的该条记录不会同步至目标数据库中。 ■ 增量期间，DTS不会保留目标集群中的该条记录，即源库中的该条记录会覆盖至目标数据库中。 ■ 表结构不一致的情况下，可能会导致无法初始化数据、只能同步部分列的数据或同步失败。
索引名称	■ **表名** 选择为**表名**后，在目标Elasticsearch实例中创建的索引名称和表名一致，在本案例中即为product。 ■ **库名_表名** 选择为**库名_表名**后，在目标Elasticsearch实例中创建的索引名称为**库名_表名**，在本案例中即为estest_product。
同步对象	在**源库对象**框中单击待同步对象，然后单击 › 将其移动至**已选择对象**框。
映射名称更改	■ 如需更改单个同步对象在目标实例中的名称，请右击**已选择对象**中的同步对象，设置方式，请参见库表列名单个映射。 ■ 如需批量更改同步对象在目标实例中的名称，请单击**已选择对象**方框右上方的**批量编辑**，设置方式，请参见库表列名批量映射。

iii. 配置映射字段，修改同步后的字段名称。

如果您需要修改同步后的字段名称，可在**已选择对象**区域框中，右键单击对应的表名，设置该表在目标Elasticsearch实例中的索引名称、Type名称等信息，然后单击**确定**。本文使用的配置及相关说明如下，未提及的配置保持默认，更多详细信息请参见库表列名单个映射。

配置	说明
索引名称	自定义索引名称，详情请参见基本概念。 **注意** 输入索引名称时，请确保Elasticsearch集群中不存在同名索引，否则报错 `index already exists` 。
Type名称	自定义索引类型名称，详情请参见基本概念。
过滤条件	您可以设置SQL过滤条件，过滤待同步的数据，只有满足过滤条件的数据才会被同步到目标实例，详情请参见通过SQL条件过滤任务数据。
字段参数	选择所需的**字段参数**和**字段参数值**，字段参数及取值介绍请参见Mapping parameters。 **注意** 如果添加参数时，将对应参数的index值设置为false，那么该字段将不能被查询，详情请参见index。

iv. 配置高级参数。

本文使用默认配置，相关说明如下。

配置	说明
设置告警	是否设置告警，当同步失败或延迟超过阈值后，将通知告警联系人。 ■ **不设置**：不设置告警。 ■ **设置**：设置告警，您还需要设置告警阈值和告警联系人。
源库、目标库无法连接后的重试时间	在同步任务连接失败时，DTS会立即进行持续的重试连接，默认持续重试时间为120分钟，您也可以在取值范围（10~1440分钟）内自定义重试时间，建议设置30分钟以上。如果DTS在设置的重试时间内重新连接上源库、目标库，同步任务将自动恢复。否则，同步任务将失败。 **说明** ■ 针对同源或者同目标的多个DTS实例，如DTS实例A和DTS实例B，设置网络重试时间时A设置30分钟，B设置60分钟，则重试时间以低的30分钟为准。 ■ 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。
分片配置	根据目标Elasticsearch中索引的分片配置，设置索引的主分片和副本分片的数量。Elasticsearch 7.x以下版本的索引默认包含5个主shard，1个副shard。Elasticsearch 7.x及以上版本的索引默认包含1个主shard，1个副shard。 **注意** shard大小和数量是影响Elasticsearch集群稳定性和性能的重要因素，您需要设置合理的shard数，shard的评估方法请参见规格容量评估。
字符串Index	同步至目标Elasticsearch实例中的字符串编入索引的方式： ■ analyzed：先分析字符串，再写入索引。您还需要选择具体的分析器，分析器的类型及作用，请参见分析器。 ■ not analyzed：不分析，直接使用原始值写入索引。 ■ no：不写入索引。
时区	DTS同步时间类型的数据（如DATETIME、TIMESTAMP）至目标Elasticsearch实例时，您可以选择所带时区。 **说明** 如目标实例中此类时间类型数据无需带有时区，则在同步前您需在目标实例中设置该时间类型数据的文档类型（type）。
DOCID取值	DOCID默认为表的主键，如表中无主键，则DOCID为Elasticsearch自动生成的ID列。

v. 配置库表字段，设置待同步的表在目标Elasticsearch的_routing策略和_id取值。

本文使用的数据库名称为estest，表名称为product，设置_routing为否，_id取值为id，相关说明如下。

类型	说明
设置_routing	设置_routing可以将文档路由存储在目标Elasticsearch实例的指定分片上，请参见_routing。 ■ 选择为**是**，您可以自定义列进行路由。 ■ 选择为**否**，则用_id进行路由。 **说明** 创建的目标Elasticsearch实例为7.x版本时，您必须选择为**否**。
_id取值	■ **表的主键列** 联合主键合并为一列。 ■ **业务主键** 如果选择为**业务主键**，那么您还需要设置对应的**业务主键列**。

4. 配置完成后，根据页面提示保存任务、进行预检查、购买并启动任务。

购买成功后，同步任务正式开始，您可在数据同步界面查看具体任务进度。待全量同步完成，增量同步进行中时，您即可在Elasticsearch中查看同步成功的数据。

步骤三：查看数据同步结果

1. 登录目标阿里云Elasticsearch实例的Kibana控制台，根据页面提示进入Kibana主页。

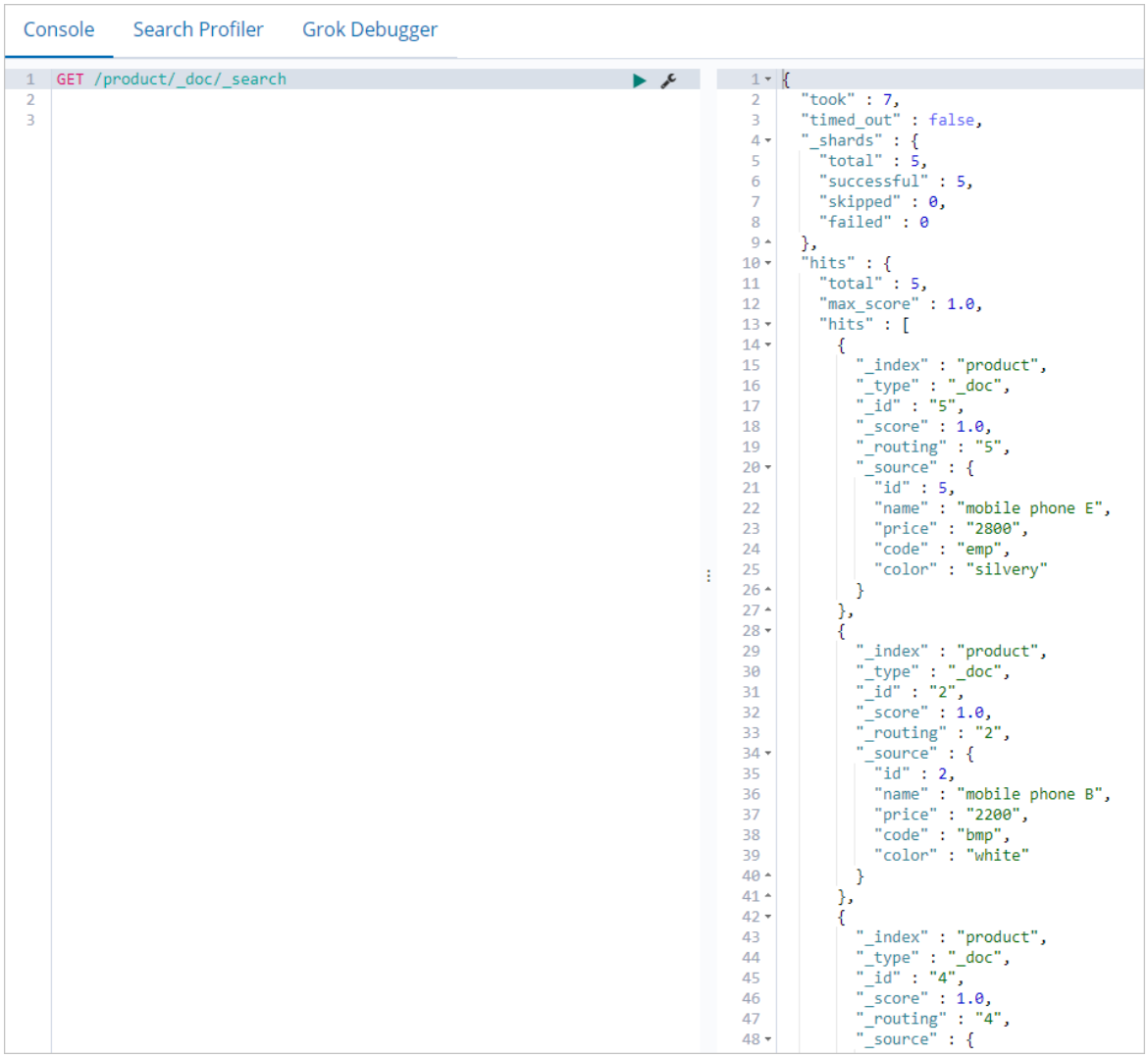
登录Kibana控制台的具体操作，请参见登录Kibana控制台。

说明 本文以阿里云Elasticsearch 6.7.0版本为例，其他版本操作可能略有差别，请以实际界面为准。

2. 在左侧导航栏，单击**Dev Tools**。
3. 在**Console**中，执行如下命令查看全量数据同步结果。

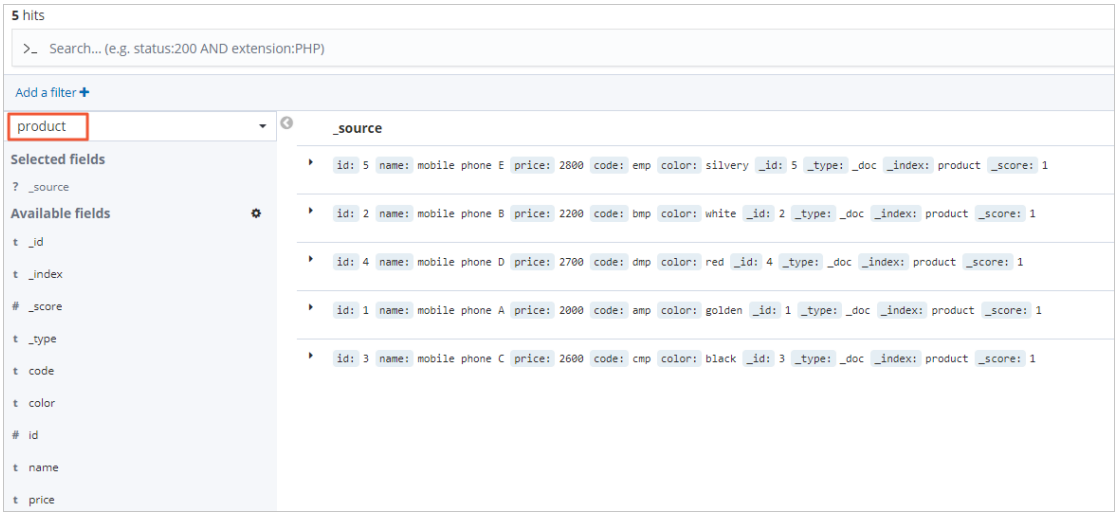
```
GET /product/_doc/_search
```

预期结果如下。



4. 通过控制台操作查看同步成功的数据。
   i. 为目标索引创建索引模式。
      a. 在左侧导航栏，单击**Management**。
      b. 在**Kibana**区域，单击**Index Patterns**。
      c. 单击**Create index pattern**。
      d. 在**Create index pattern**页面的**Index pattern name**中，输入与同步成功的索引相匹配的索引模式名称（例如product、product*）。
      e. 单击**Next step**。
      f. 单击**Create index pattern**。
   ii. 在左侧导航栏，单击**Discover**。

iii. 选择您已创建的索引模式，查看同步成功的数据。

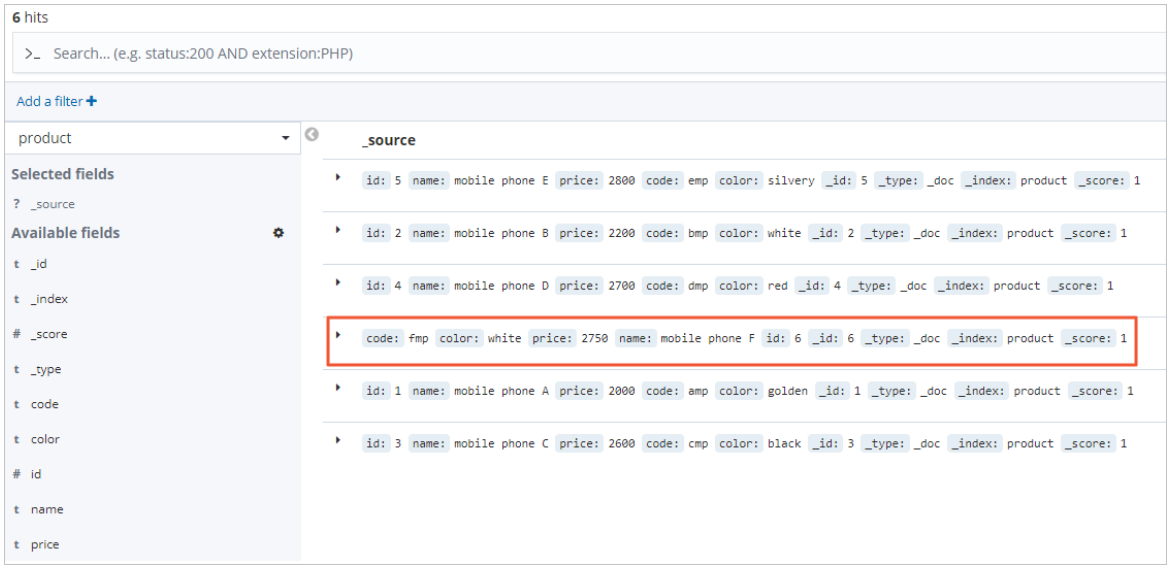


步骤四：验证增量数据同步

1. 进入PolarDB控制台。
2. 在PolarDB MySQL数据库中插入一条数据。

```
INSERT INTO `estest`.`product` (`id`,`name`,`price`,`code`,`color`) VALUES (6,'mobile phone F','2750','fmp','white');
```

3. 登录Kibana控制台，在左侧导航栏，单击**Discover**，选择您已创建的索引模式，查看同步成功的增量数据。



> **说明** 您也可以使用同样的方式，在PolarDB MySQL数据库中删除或修改数据，然后查看数据同步结果。

3.4. 通过Monstache实时同步MongoDB数据至Elasticsearch

Monstache同步MongoDB数据到ES

当您的业务数据存储在MongoDB中，并且需要进行语义分析和大图展示时，可借助阿里云Elasticsearch实现全文搜索、语义分析、可视化展示等。本文介绍如何通过Monstache将MongoDB数据实时同步至阿里云Elasticsearch，并对数据进行分析及展示。

背景信息

本文以解析及统计热门电影数据为例，提供的解决方案可以帮助您完成以下需求：

- 通过Monstache快速同步及订阅全量或增量数据。
- 将MongoDB数据实时同步至高版本Elasticsearch。
- 解读Monstache常用配置参数，应用于更多的业务场景。

方案优势

- MongoDB、阿里云Elasticsearch及Monstache服务部署在专有网络VPC（Virtual Private Cloud）内，所有数据私网通信，高速且安全。
- Monstache基于MongoDB的oplog实现实时数据同步及订阅，支持MongoDB与高版本Elasticsearch之间的数据同步，同时支持MongoDB的变更流和聚合管道功能，并且拥有丰富的特性。
- Monstache不仅支持软删除和硬删除，还支持数据库删除和集合删除，能够确保Elasticsearch端实时与源端数据保持一致。

操作流程

1. 步骤一：环境准备

   准备同一专有网络下的阿里云MongoDB实例、阿里云Elasticsearch实例和ECS实例。其中ECS实例用来安装Monstache。

   > 注意 请准备版本兼容的Monstache工具、阿里云Elasticsearch和MongoDB实例，版本兼容性详情请参见Monstache version。

2. 步骤二：搭建Monstache环境

   在ECS实例中安装Monstache，用来将MongoDB中的数据同步至阿里云Elasticsearch。安装前需要先配置Go环境变量。

3. 步骤三：配置实时同步任务

   修改默认的Monstache配置文件，在配置文件中指定MongoDB和Elasticsearch的访问地址、待同步的集合、Elasticsearch的用户名和密码等参数。配置完成后，运行Monstache服务，即可将MongoDB中的数据实时同步至阿里云Elasticsearch中。

4. 步骤四：验证数据同步结果

   分别在MongoDB数据库中添加、更新、删除数据，验证数据是否实时同步。

5. 步骤五：通过Kibana分析并展示数据

   在Kibana控制台中，分析数据并使用Pie图展示分析结果。

步骤一：环境准备

1. 创建阿里云Elasticsearch实例，并开启实例的自动创建索引功能。

   具体操作步骤请参见创建阿里云Elasticsearch实例和配置YML参数。本文使用的实例版本为通用商业版6.7。

2. 创建阿里云MongoDB实例，并准备测试数据。

具体操作步骤请参见MongoDB快速入门。本文以4.2版本的副本集MongoDB实例为例，部分数据如下。

> 注意 MongoDB实例必须是副本集或分片集架构，不支持单节点架构。

3. 创建ECS实例。

   具体操作步骤请参见使用向导创建实例。该ECS实例用来安装Monstache，需要与阿里云Elasticsearch实例在同一专有网络下。

步骤二：搭建Monstache环境

1. 参见连接ECS实例，连接ECS实例。
2. 安装Go，并配置环境变量。

   > 说明 由于Monstache数据同步依赖于Go语言，因此需要先在ECS中准备Go环境。

   i. 下载Go安装包并解压。

   ```
   wget https://dl.google.com/go/go1.14.4.linux-amd64.tar.gz
   tar -C /usr/local -xzf go1.14.4.linux-amd64.tar.gz
   ```

   ii. 配置环境变量。

   使用 `vim /etc/profile` 命令打开环境变量配置文件，并将如下内容写入该文件中。其中 `GOPROXY` 用来指定阿里云Go模块代理。

   ```
   export GOROOT=/usr/local/go
   export GOPATH=/home/go/
   export PATH=$PATH:$GOROOT/bin:$GOPATH/bin
   export GOPROXY=https://mirrors.aliyun.com/goproxy/
   ```

   iii. 应用环境变量配置。

   ```
   source /etc/profile
   ```

3. 安装Monstache。

i. 进入安装路径。

```
cd /usr/local/
```

ii. 从Git库中下载安装包。

```
git clone https://github.com/rwynn/monstache.git
```

> **说明** 如果出现 `git: command not found` 的错误提示，需要先执行 `yum install -y git` 命令安装Git。

iii. 进入*monstache*目录。

```
cd monstache
```

iv. 切换版本。

本文以rel5版本为例。

```
git checkout rel5
```

v. 安装Monstache。

```
go install
```

vi. 查看Monstache版本。

```
monstache -v
```

执行成功后，预期结果如下。

```
5.5.5
```

步骤三：配置实时同步任务

Monstache配置使用TOML格式，默认情况下，Monstache会使用默认端口连接本地主机上的Elasticsearch和MongoDB，并追踪MongoDB oplog。在Monstache运行期间，MongoDB的任何更改都会同步到Elasticsearch中。

由于本文使用阿里云MongoDB和Elasticsearch，并且需要指定同步对象（mydb数据库中的hotmovies和col集合），因此要修改默认的Monstache配置文件。修改方式如下：

1. 进入Monstache安装目录，创建并编辑配置文件。

```
cd /usr/local/monstache/
vim config.toml
```

2. 参考以下示例，修改配置文件。

简单的配置示例如下，详细配置请参见Monstache Usage。

```
# connection settings
# connect to MongoDB using the following URL
mongo-url = "mongodb://root:<your_mongodb_password>@dds-bp1aadcc629******.mongodb.rds.aliyuncs.com:3717"
# connect to the Elasticsearch REST API at the following node URLs
elasticsearch-urls = ["http://es-cn-mp91kzb8m00******.elasticsearch.aliyuncs.com:9200"]
# frequently required settings
```

```
# If you need to seed an index from a collection and not just listen and sync changes ev
ents
# you can copy entire collections or views from MongoDB to Elasticsearch
direct-read-namespaces = ["mydb.hotmovies","mydb.col"]
# if you want to use MongoDB change streams instead of legacy oplog tailing use change-s
tream-namespaces
# change streams require at least MongoDB API 3.6+
# if you have MongoDB 4+ you can listen for changes to an entire database or entire depl
oyment
# in this case you usually don't need regexes in your config to filter collections unles
s you target the deployment.
# to listen to an entire db use only the database name.  For a deployment use an empty s
tring.
#change-stream-namespaces = ["mydb.col"]
# additional settings
# if you don't want to listen for changes to all collections in MongoDB but only a few
# e.g. only listen for inserts, updates, deletes, and drops from mydb.mycollection
# this setting does not initiate a copy, it is only a filter on the change event listene
r
#namespace-regex = '^mydb\.col$'
# compress requests to Elasticsearch
#gzip = true
# generate indexing statistics
#stats = true
# index statistics into Elasticsearch
#index-stats = true
# use the following PEM file for connections to MongoDB
#mongo-pem-file = "/path/to/mongoCert.pem"
# disable PEM validation
#mongo-validate-pem-file = false
# use the following user name for Elasticsearch basic auth
elasticsearch-user = "elastic"
# use the following password for Elasticsearch basic auth
elasticsearch-password = "<your_es_password>"
# use 4 go routines concurrently pushing documents to Elasticsearch
elasticsearch-max-conns = 4
# use the following PEM file to connections to Elasticsearch
#elasticsearch-pem-file = "/path/to/elasticCert.pem"
# validate connections to Elasticsearch
#elastic-validate-pem-file = true
# propogate dropped collections in MongoDB as index deletes in Elasticsearch
dropped-collections = true
# propogate dropped databases in MongoDB as index deletes in Elasticsearch
dropped-databases = true
# do not start processing at the beginning of the MongoDB oplog
# if you set the replay to true you may see version conflict messages
# in the log if you had synced previously. This just means that you are replaying old do
cs which are already
# in Elasticsearch with a newer version. Elasticsearch is preventing the old docs from o
verwriting new ones.
#replay = false
# resume processing from a timestamp saved in a previous run
resume = true
# do not validate that progress timestamps have been saved
#resume-write-unsafe = false
```

```
# override the name under which resume state is saved
#resume-name = "default"
# use a custom resume strategy (tokens) instead of the default strategy (timestamps)
# tokens work with MongoDB API 3.6+ while timestamps work only with MongoDB API 4.0+
resume-strategy = 0
# exclude documents whose namespace matches the following pattern
#namespace-exclude-regex = '^mydb\.ignorecollection$'
# turn on indexing of GridFS file content
#index-files = true
# turn on search result highlighting of GridFS content
#file-highlighting = true
# index GridFS files inserted into the following collections
#file-namespaces = ["users.fs.files"]
# print detailed information including request traces
verbose = true
# enable clustering mode
cluster-name = 'es-cn-mp91kzb8m00******'
# do not exit after full-sync, rather continue tailing the oplog
#exit-after-direct-reads = false
[[mapping]]
namespace = "mydb.hotmovies"
index = "hotmovies"
type = "movies"
[[mapping]]
namespace = "mydb.col"
index = "mydbcol"
type = "collection"
```

参数	说明
mongo-url	MongoDB实例的主节点访问地址。可在实例的基本信息页面获取，获取前需配置MongoDB实例的白名单，即在白名单中添加安装Monstache的ECS实例的内网IP地址，详情请参见设置白名单。
elasticsearch-urls	阿里云Elasticsearch实例的访问地址，格式为 `http://<阿里云Elasticsearch实例的内网地址>:9200` 。阿里云Elasticsearch实例的内网地址可在实例的基本信息页面获取，详情请参见查看实例的基本信息。
direct-read-namespaces	指定待同步的集合，详情请参见direct-read-namespaces。本文同步的数据集为mydb数据库下的hotmovies和col集合。
change-stream-namespaces	如果要使用MongoDB变更流功能，需要指定此参数。启用此参数后，oplog追踪会被设置为无效，详情请参见change-stream-namespaces。
namespace-regex	通过正则表达式指定需要监听的集合。此设置可以用来监控符合正则表达式的集合中数据的变化。

参数	说明
elasticsearch-user	访问阿里云Elasticsearch实例的用户名，默认为elastic。 **注意** 实际业务中不建议使用elastic用户，这样会降低系统安全性。建议使用自建用户，并给予自建用户分配相应的角色和权限，详情请参见通过Elasticsearch X-Pack角色管理实现用户权限管控。
elasticsearch-password	对应用户的密码。elastic用户的密码在创建实例时指定，如果忘记可进行重置，重置密码的注意事项和操作步骤请参见重置实例访问密码。
elasticsearch-max-conns	定义连接Elasticsearch的线程数。默认为4，即使用4个Go线程同时将数据同步到Elasticsearch。
dropped-collections	默认为true，表示当删除MongoDB集合时，会同时删除Elasticsearch中对应的索引。
dropped-databases	默认为true，表示当删除MongoDB数据库时，会同时删除Elasticsearch中对应的索引。
resume	默认为false。设置为true，Monstache会将已成功同步到Elasticsearch的MongoDB操作的时间戳写入monstache.monstache集合中。当Monstache因为意外停止时，可通过该时间戳恢复同步任务，避免数据丢失。如果指定了cluster-name，该参数将自动开启，详情请参见resume。
resume-strategy	指定恢复策略。仅当resume为true时生效，详情请参见resume-strategy。
verbose	默认为false，表示不启用调试日志。
cluster-name	指定集群名称。指定后，Monstache将进入高可用模式，集群名称相同的进程将进行协调，详情请参见cluster-name。
mapping	指定Elasticsearch索引映射。默认情况下，数据从MongoDB同步到Elasticsearch时，索引会自动映射为 `数据库名.集合名` 。如果需要修改索引名称，可通过该参数设置，详情请参见Index Mapping。

说明 Monstache支持丰富的参数配置，以上配置仅使用了部分参数完成数据实时同步，如果您有更复杂的同步需求，请参见Monstache config和Advanced进行配置。

3. 运行Monstache。

```
monstache -f config.toml
```

说明 通过-f参数，您可以显式运行Monstache，系统会打印所有调试日志（包括对Elasticsearch的请求追踪）。

步骤四：验证数据同步结果

1. 分别进入MongoDB的DMS控制台和阿里云Elasticsearch实例的Kibana控制台，查看同步前后对应文档的数量。

> 说明
> - 登录DMS控制台的方法请参见通过DMS连接MongoDB副本集实例。
> - 登录Kibana控制台的方法请参见登录Kibana控制台。

- MongoDB

```
db.hotmovies.find().count()
```

预期结果如下。

```
[
10000
]
```

- 阿里云Elasticsearch

```
GET hotmovies/_count
```

预期结果如下。通过以下结果可以看到同步前后的文档的数量都为10000条。

```
{
  "count" : 10000,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  }
}
```

2. 在MongoDB数据库中插入数据，查看该数据是否同步到阿里云Elasticsearch实例中。

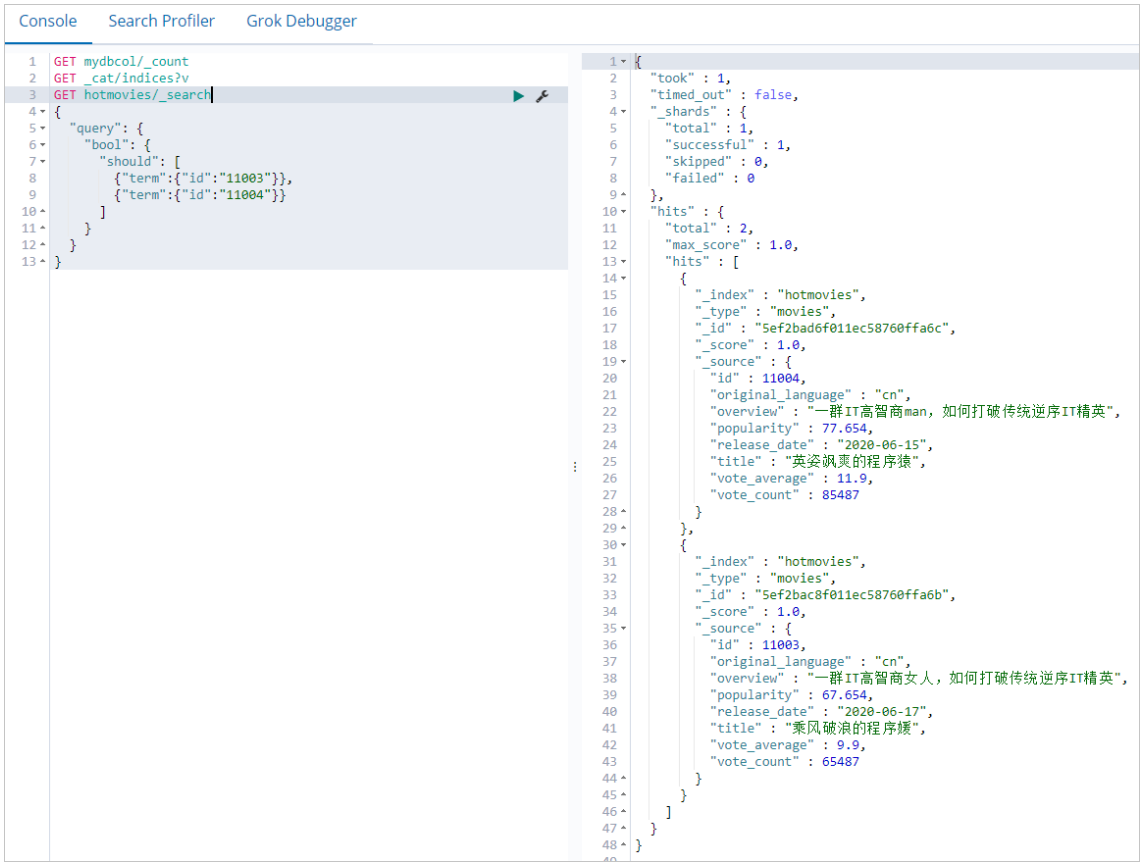
- MongoDB

```
db.hotmovies.insert({id: 11003,title: "乘风破浪的程序媛",overview: "一群IT高智商女人，如何打破传统逆序IT精英",original_language:"cn",release_date:"2020-06-17",popularity:67.654,vote_count:65487,vote_average:9.9})
db.hotmovies.insert({id: 11004,title: "英姿飒爽的程序猿",overview: "一群IT高智商man，如何打破传统逆序IT精英",original_language:"cn",release_date:"2020-06-15",popularity:77.654,vote_count:85487,vote_average:11.9})
```

- 阿里云Elasticsearch

```
GET hotmovies/_search
{
  "query": {
    "bool": {
      "should": [
        {"term":{"id":"11003"}},
        {"term":{"id":"11004"}}
      ]
    }
  }
}
```

预期结果如下。



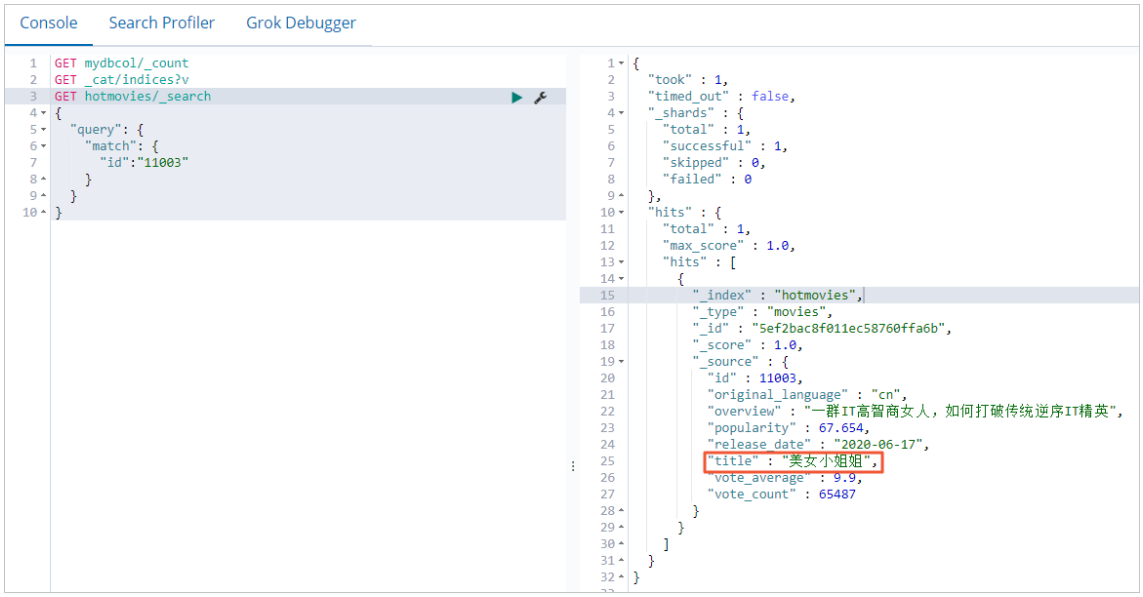
3. 在MongoDB数据库中更新数据，查看阿里云Elasticsearch实例中对应的数据是否会同步更新。
   - MongoDB

     ```
     db.hotmovies.update({'title':'乘风破浪的程序媛'},{$set:{'title':'美女小姐姐'}})
     ```

   - 阿里云Elasticsearch

     ```
     GET hotmovies/_search
     {
       "query": {
         "match": {
           "id":"11003"
         }
       }
     }
     ```

预期结果如下。



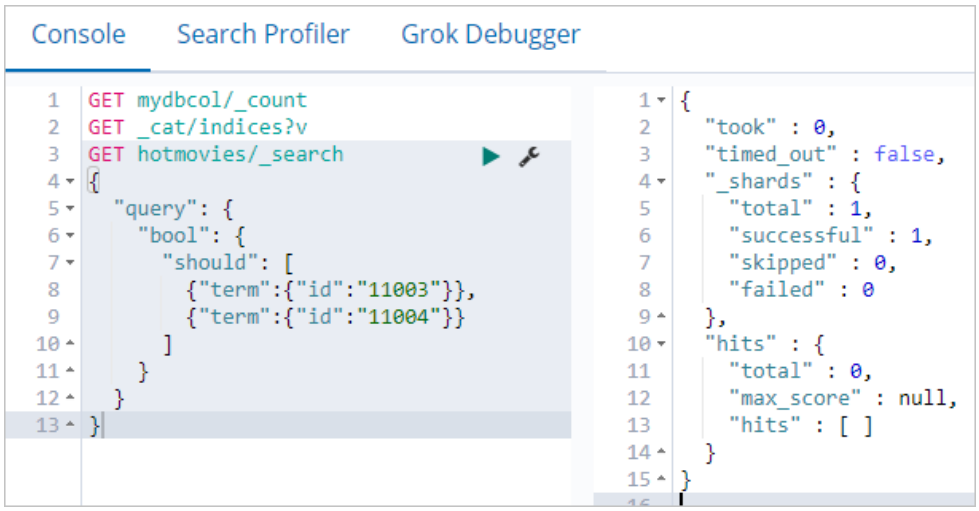
4. 在MongoDB数据库中删除数据，查看阿里云Elasticsearch实例中对应的数据是否会同步删除。
   - MongoDB

```
db.hotmovies.remove({id: 11003})
db.hotmovies.remove({id: 11004})
```

   - 阿里云Elasticsearch

```
GET hotmovies/_search
{
  "query": {
    "bool": {
      "should": [
        {"term":{"id":"11003"}},
        {"term":{"id":"11004"}}
      ]
    }
  }
}
```

预期结果如下。



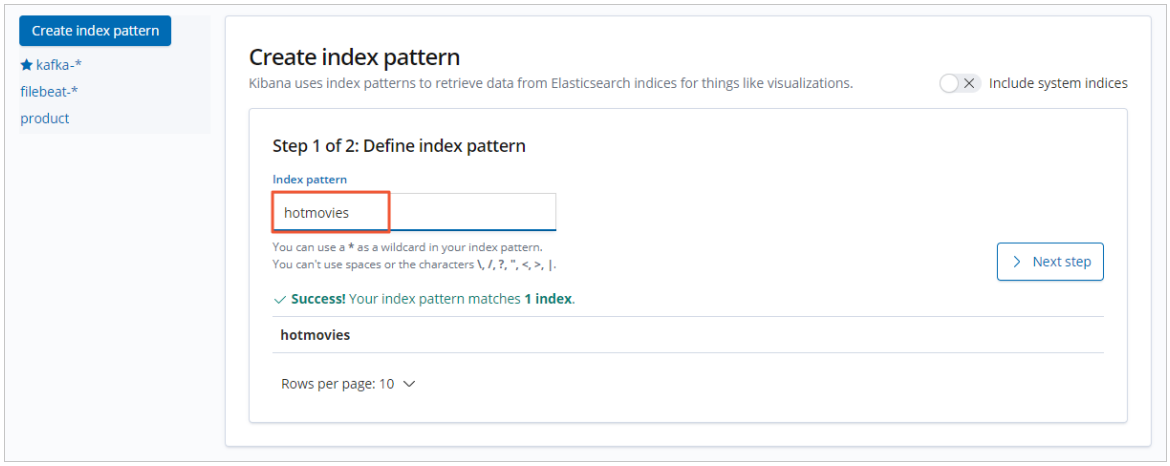
步骤五：通过Kibana分析并展示数据

1. 登录目标阿里云Elasticsearch实例的Kibana控制台，根据页面提示进入Kibana主页。

   登录Kibana控制台的具体操作，请参见登录Kibana控制台。

   > 说明 本文以阿里云Elasticsearch 6.7.0版本为例，其他版本操作可能略有差别，请以实际界面为准。

2. 创建索引模式。



   i. 在左侧导航栏，单击Management。

   ii. 在Kibana区域，单击Index Patterns。

   iii. 单击Create index pattern。

   iv. 输入Index pattern名称，单击Next step。

   v. 从Time Filter field name中，选择时间过滤器字段名（本文选择I don't want to use the Time Filter）。

   vi. 单击Create index pattern。

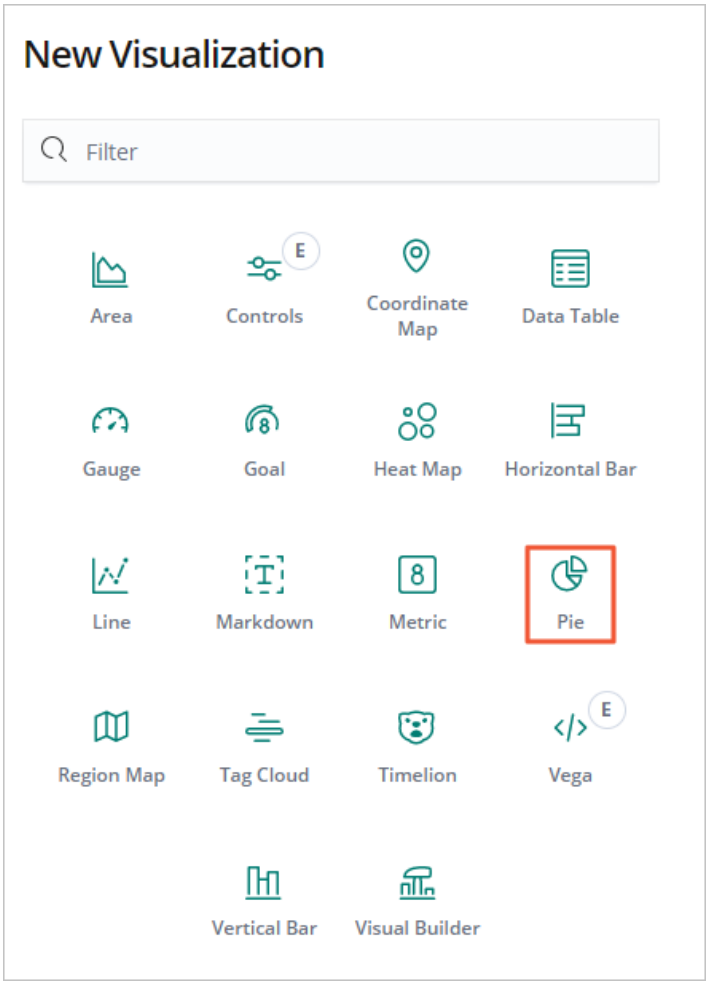
3. 配置Kibana大图。

   本文以配置最受欢迎的Top10电影的Pie图为例，操作步骤如下：

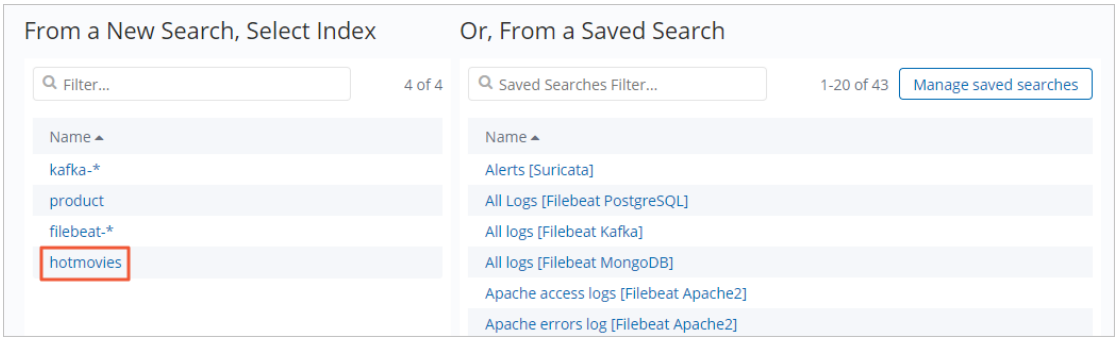
i. 在左侧导航栏，单击Visualize。

ii. 在搜索框右侧，单击+。

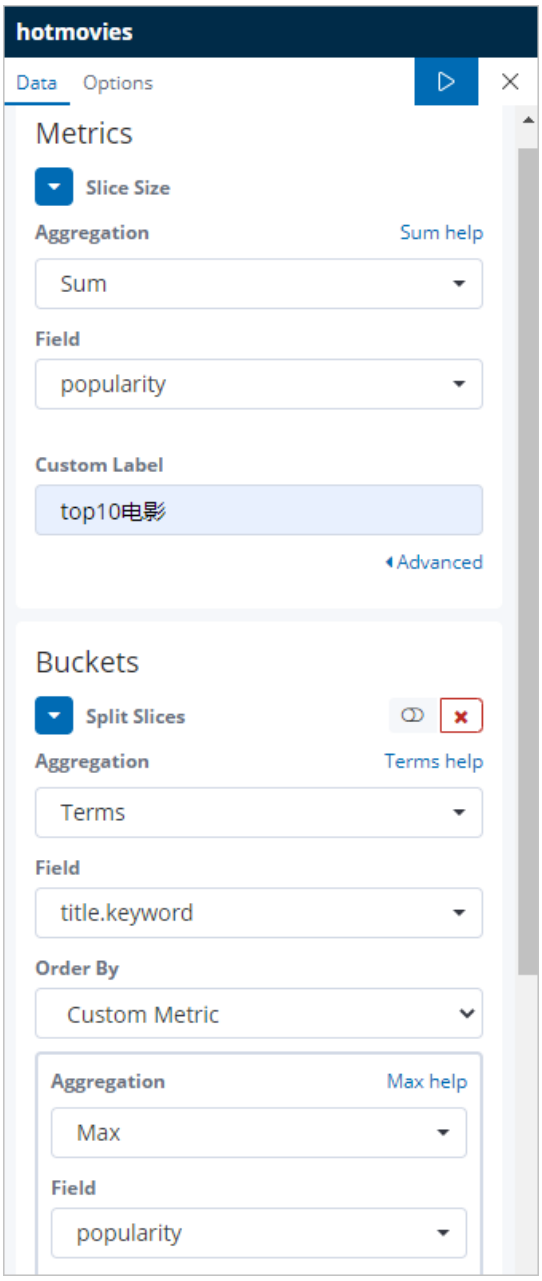
iii. 在New Visualization对话框中，单击Pie。



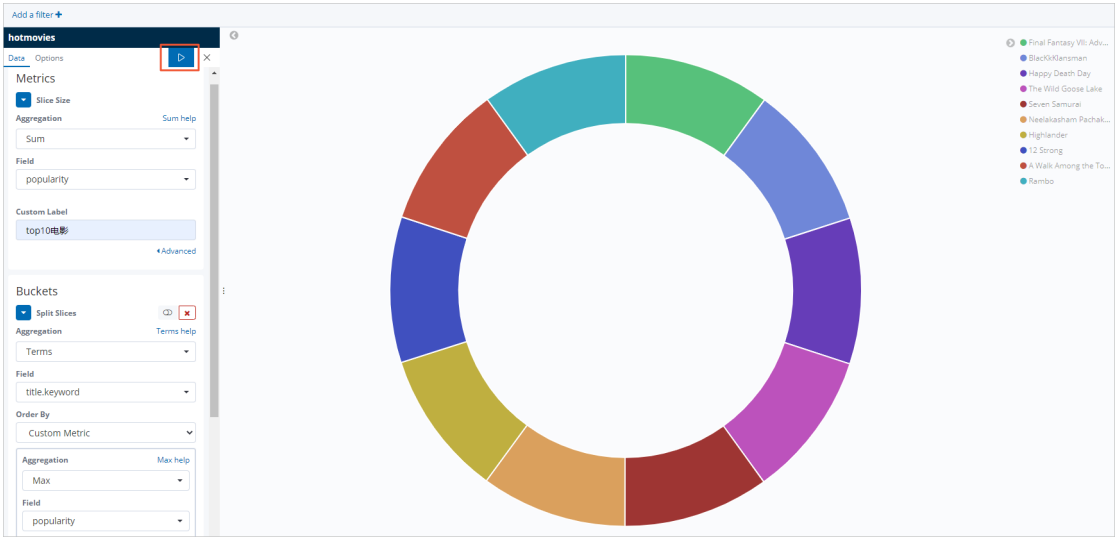
iv. 单击hotmovies索引模式。



v. 按照下图配置Metrics和Buckets。



vi. 单击 图标，应用配置，查看数据展示结果。



常见问题

- 问题

  阿里云Elasticsearch实例开启高可用、高并发功能后，数据有丢失现象，如何排查？

- 解决方案

  查看阿里云Elasticsearch集群的整体情况是否正常：

  - 正常：需要排查Monstache服务的问题，详细信息请参见Monstache官网。
  - 不正常：参见热点文章，排查阿里云Elasticsearch集群的问题。同时降低并发数，观察数据是否正常。

  如果您的问题仍未解决，可以提交工单咨询。

4.大数据云产品同步方案

4.1. 通过DataWorks将MaxCompute数据同步至Elasticsearch

MaxCompute数据同步到阿里云es

阿里云上拥有丰富的云存储、云数据库产品。当您需要对这些产品中的数据进行分析和搜索时，可以通过DataWorks的数据集成服务实现最快5分钟一次的离线数据采集，并同步到阿里云Elasticsearch中。本文以阿里云大数据计算服务MaxCompute（原名ODPS）为例。

背景信息

阿里云Elasticsearch支持同步的离线数据源包括：

- 阿里云云数据库（MySQL、PostgreSQL、SQL Server、PPAS、MongoDB、HBase）
- 阿里云PolarDB-X（原DRDS升级版）
- 阿里云MaxCompute
- 阿里云OSS
- 阿里云Tablestore
- 自建HDFS、Oracle、FTP、DB2，及以上数据库的自建版本

操作流程

1. 准备工作

   创建DataWorks工作空间并开通MaxCompute服务、准备MaxCompute数据源、创建阿里云Elasticsearch实例。

2. 步骤一：购买并创建独享资源组

   购买并创建一个数据集成独享资源组，并为该资源组绑定专有网络和工作空间。独享资源组可以保障数据快速、稳定地传输。

3. 步骤二：添加数据源

   将MaxCompute和Elasticsearch数据源接入DataWorks的数据集成服务中。

4. 步骤三：配置并运行数据同步任务

   通过向导模式配置数据同步任务，将数据集成系统同步成功的数据存储到Elasticsearch中。将独享资源组作为一个可以执行任务的资源，注册到DataWorks的数据集成服务中。这个资源组将获取数据源的数据，并执行将数据写入Elasticsearch中的任务（该任务将由数据集成系统统一下发）。

5. 步骤四：验证数据同步结果

   在Kibana控制台中，查看同步成功的数据，并按条件查询数据。

准备工作

1. 创建DataWorks工作空间。创建时选择MaxCompute计算引擎服务。

   具体操作，请参见创建MaxCompute项目。

2. 创建MaxCompute表并导入测试数据。

   具体操作，请参见创建表、导入数据。

   本文使用的表结构和部分数据如下。

表结构

表数据

	A	B	C	D	E	F	G	H
1	create_time	category	brand	buyer_id	trans_num	trans_amount	click_cnt	pt
2	2020/6/1 8:00	外套	品牌A	user1	10	150.0	50	1
3	2020/6/2 8:00	生鲜	品牌B	user2	11	180.0	60	1
4	2020/6/3 8:00	外套	品牌D	user3	12	150.0	50	1
5	2020/6/4 8:00	外套	品牌A	user4	14	150.0	50	1
6	2020/6/5 8:00	电器	品牌C	user5	10	1500.0	90	1
7	2020/6/6 8:00	外套	品牌F	user6	10	190.0	69	1
8	2020/6/7 8:00	外套	品牌E	user7	88	150.0	80	1
9	2020/6/8 8:00	外套	品牌A	user8	10	180.0	67	1
10	2020/6/9 8:00	卫浴	品牌G	user9	18	3500.0	50	1
11	2020/6/10 8:00	外套	品牌F	user10	10	150.0	45	1
12	2020/6/11 8:00	卫浴	品牌G	user11	22	4500.0	70	1
13	2020/6/12 8:00	外套	品牌A	user12	10	150.0	55	1
14	2020/6/13 8:00	生鲜	品牌B	user13	20	220.0	110	1

> **说明** 本文提供的数据仅供测试。实际情况中，您可以将Hadoop数据迁移到MaxCompute后再进行同步。详细信息，请参见Hadoop数据迁移MaxCompute最佳实践。

3. 创建阿里云Elasticsearch实例，并开启实例的自动创建索引功能。

   具体操作，请参见创建阿里云Elasticsearch实例和配置YML参数。创建实例时，所选地域与DataWorks工作空间所在地域保持一致。

步骤一：购买并创建独享资源组

1. 登录DataWorks控制台。
2. 选择相应地域后，在左侧导航栏，单击**资源组列表**。
3. 参见购买资源组（创建订单），购买独享数据集成资源。

> **注意** 购买时，所选地域需要与目标工作空间保持一致。

4. 参见新增和使用独享数据集成资源组，创建一个独享数据集成资源。

   本文使用的配置如下，其中**资源组类型**选择**独享数据集成资源组**。

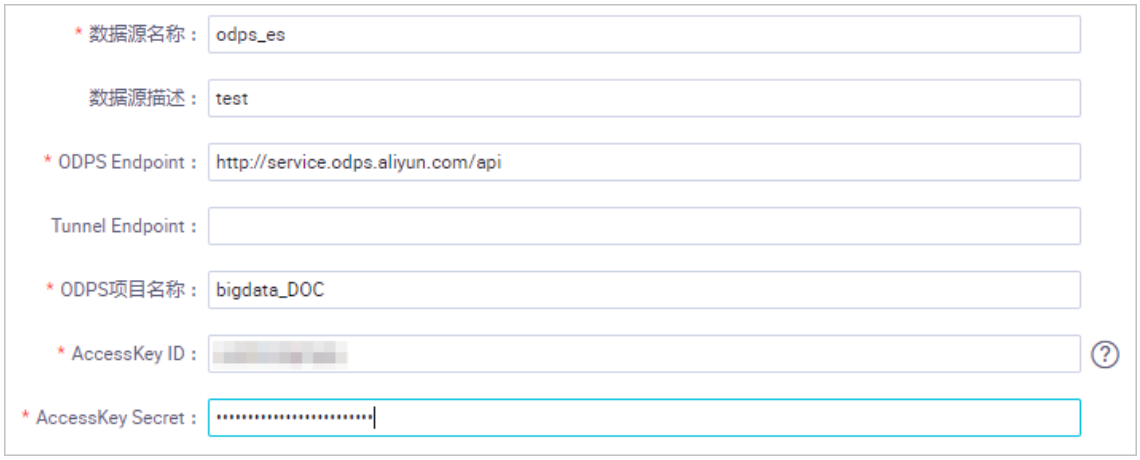
5. 单击已创建的独享资源组右侧的**网络设置**，参见绑定专有网络，为该独享资源组绑定专有网络。

   独享资源部署在DataWorks托管的专有网络中。DataWorks需要与Elasticsearch实例的专有网络连通才能同步数据，因此在绑定专有网络时，需要选择Elasticsearch实例所在**专有网络**和**交换机**。

6. 单击已创建的独享资源组右侧的**修改归属工作空间**，参见新增和使用独享数据集成资源组，为该独享资源组绑定目标工作空间。

步骤二：添加数据源

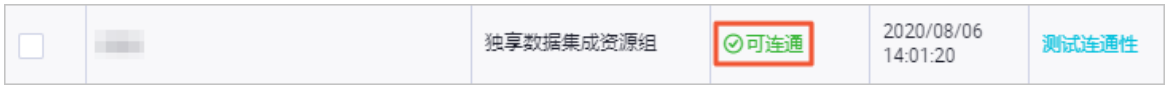
1. 进入DataWorks的**数据集成**页面。
   i. 在DataWorks控制台的左侧导航栏，单击**工作空间列表**。
   ii. 找到目标工作空间，单击其右侧**操作**列下的**进入数据集成**。
2. 在左侧导航栏，选择**数据源 > 数据源列表**。
3. 在**数据源管理**页面，单击**新增数据源**。
4. 在**新增数据源**对话框中，单击**MaxCompute**，进入**新增MaxCompute数据源**页面，填写数据源信息。



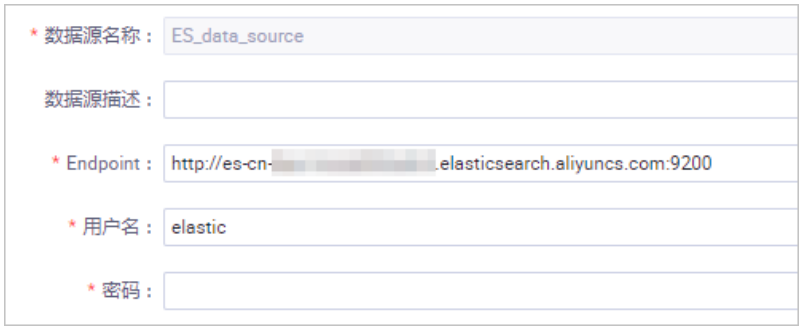
参数	说明
ODPS Endpoint	MaxCompute服务的访问地址，不同区域的访问地址不同，详情请参见Endpoint。
ODPS项目名称	进入DataWorks控制台，在左侧导航栏，单击**计算引擎列表**下的**MaxCompute**获取。
AccessKey ID	鼠标移至您的用户头像上，单击**AccessKey 管理**获取。
AccessKey Secret	鼠标移至您的用户头像上，单击**AccessKey 管理**获取。

> **说明** 以上未提及的配置请自定义输入或保持默认。

配置完成后，可与独享资源组进行连通性测试。**连通状态**显示为**可连通**时，表示连通成功。



5. 单击**完成**。
6. 使用同样的方式添加Elasticsearch数据源。



参数	说明
Endpoint	阿里云Elasticsearch的访问地址，格式为：`http://<实例的内网或公网地址>:9200`。实例的内网或公网地址可在基本信息页面获取，详细信息，请参见查看实例的基本信息。 **注意** 如果您使用的是公网地址，需要将独享资源组的EIP地址添加到阿里云Elasticsearch的公网地址访问白名单中，详情请参见配置实例公网或私网访问白名单和使用独享数据集成资源组执行任务需要在数据库添加的IP白名单。
用户名	访问阿里云Elasticsearch实例的用户名，默认为elastic。
密码	对应用户的密码。elastic用户的密码在创建实例时设定，如果忘记可重置，重置密码的注意事项和操作步骤，请参见重置实例访问密码。

说明 其他未提及的参数请自定义输入。

步骤三：配置并运行数据同步任务

1. 在DataWorks的数据开发页面，新建一个业务流程。

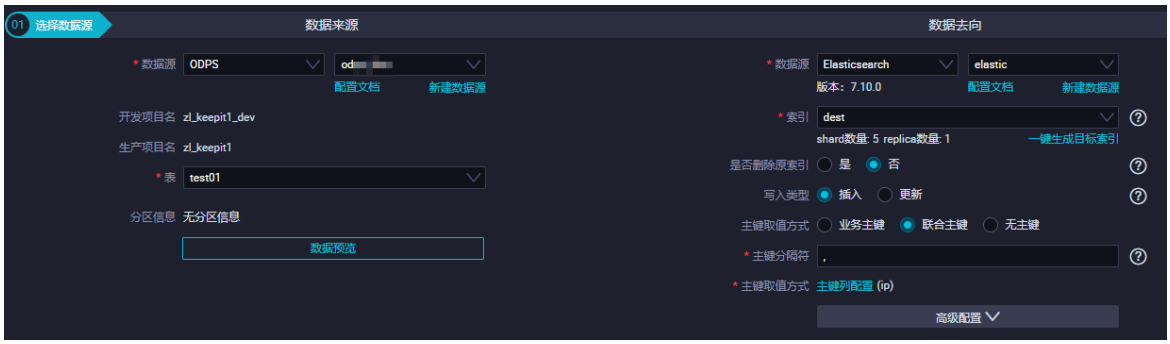
   具体操作步骤请参见管理业务流程。

2. 新建一个离线同步任务。

   i. 展开新建的业务流程，右键单击**数据集成**，选择**新建 > 离线同步**。

   ii. 在**新建节点**对话框中，输入**节点名称**，单击**提交**。

3. 在**选择数据源**区域中，将**数据来源**指定为ODPS数据源，并选择待同步的表名称；将**数据去向**指定为Elasticsearch数据源，并填入索引名和索引类型。



> 说明
> - 您也可以使用脚本模式配置数据同步，详情请参见通过脚本模式配置离线同步任务、DRDS Reader和Elasticsearch Writer。
> - 建议在Elasticsearch数据源的**高级配置**下，将**启用节点发现**设置为**否**，否则同步过程中提示连接超时。

4. 在**字段映射**区域中，设置**来源字段**与**目标字段**的映射关系。
5. 在**通道控制**区域中，配置执行任务的相关参数。
6. 配置任务调度属性。

   在页面右侧，单击**调度配置**，按照需求配置相应的调度参数。各配置的详细说明请参见调度配置章节。

> 注意
> - 在提交任务前，必须配置任务调度**依赖的上游节点**，详情请参见配置同周期调度依赖。
> - 如果您希望对任务进行周期性调度，需要配置任务的**时间属性**，包括任务的具体执行时间、调度周期、生效周期、重跑属性等。
> - 周期任务将于配置任务开始的第二天00:00，按照您的配置规则生效执行。

7. 配置执行同步任务所使用的资源组。

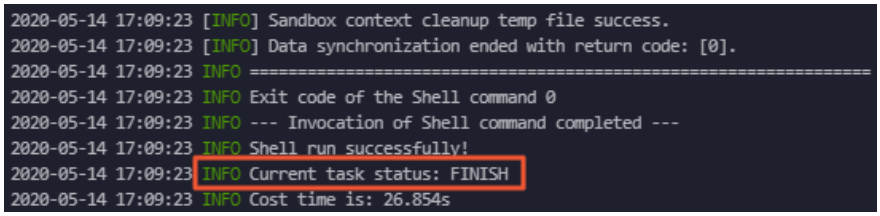
   i. 在页面右侧，单击**数据集成资源组配置**。
   ii. 选择**独享数据集成资源组**为您创建的独享资源组。

8. 提交任务。

   i. 保存当前配置，单击图标。
   ii. 在**提交新版本**对话框中，填入**备注**。
   iii. 单击**确认**。

9. 单击图标，运行任务。

任务运行过程中，可查看运行日志。运行成功后，显示如下结果。



步骤四：验证数据同步结果

1. 登录目标阿里云Elasticsearch实例的Kibana控制台。

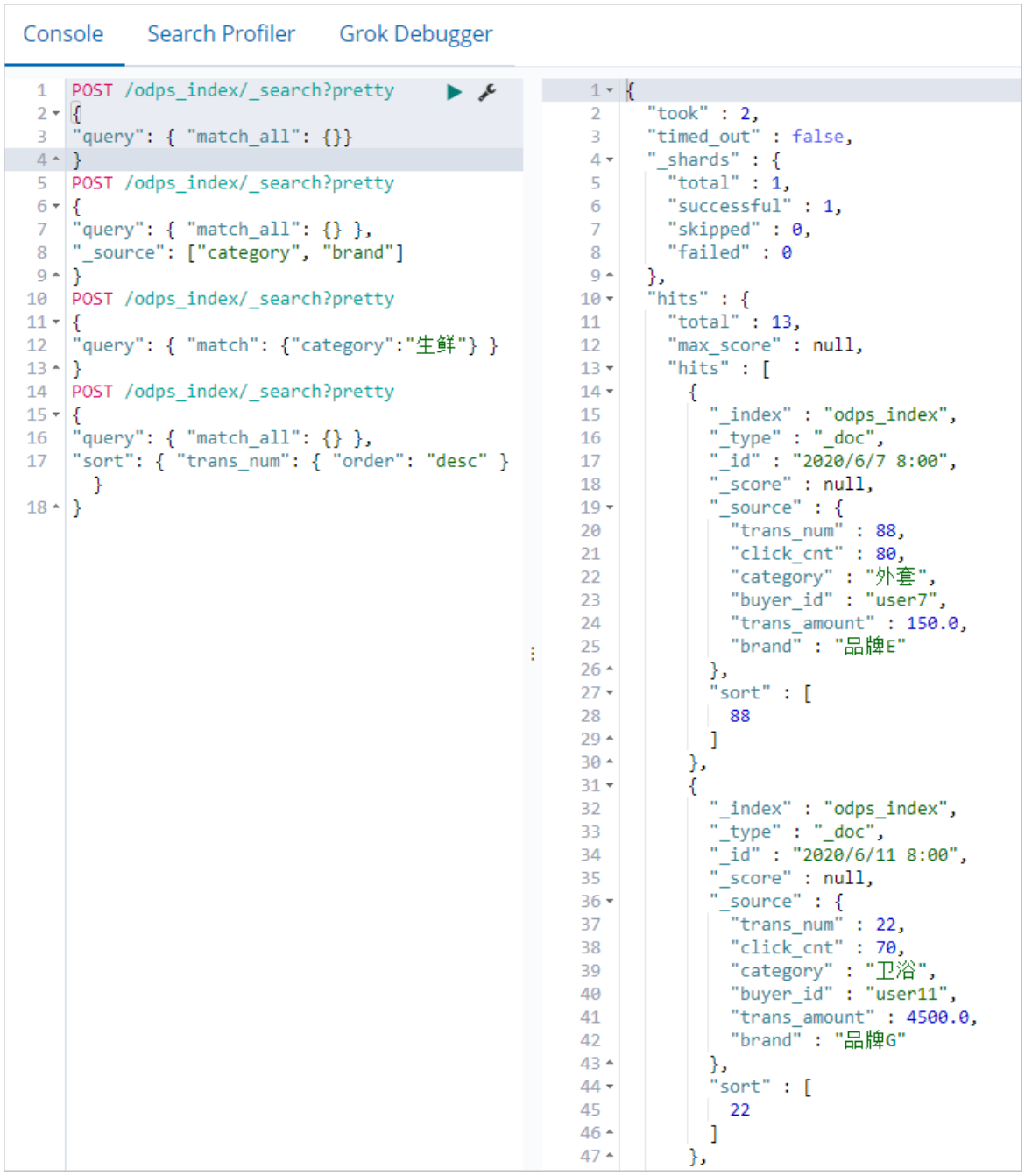
   具体操作，请参见登录Kibana控制台。

2. 在左侧导航栏，单击Dev Tools（开发工具）。
3. 在Console中，执行如下命令查看同步的数据。

```
POST /odps_index/_search?pretty
{
"query": { "match_all": {}}
}
```

> **说明** `odps_index` 为您在数据同步脚本中设置的 `index` 字段的值。

数据同步成功后，返回如下结果。



4. 执行如下命令，搜索文档中的 `category` 和 `brand` 字段。

```
POST /odps_index/_search?pretty
{
"query": { "match_all": {} },
"_source": ["category", "brand"]
}
```

5. 执行如下命令，搜索 `category` 为 `生鲜` 的文档。

```
POST /odps_index/_search?pretty
{
"query": { "match": {"category":"生鲜"} }
}
```

6. 执行如下命令，按照 `trans_num` 字段对文档进行排序。

```
POST /odps_index/_search?pretty
{
"query": { "match_all": {} },
"sort": { "trans_num": { "order": "desc" } }
}
```

更多命令和访问方式，请参见Elastic.co官方帮助中心。

4.2. 通过阿里云Logstash将MaxCompute数据同步至Elasticsearch

logstash同步MaxCompute数据到es

当您需要将MaxCompute离线表中的数据同步到阿里云Elasticsearch中时，可以通过阿里云Logstash的logstash-input-maxcompute插件和管道配置功能实现。本文介绍对应的配置方法。

前提条件

您已完成以下操作：

- 开通阿里云MaxCompute产品，并完成创建项目、创建表和导入数据的任务。

  具体操作步骤请参见MaxCompute官方文档的准备工作和快速入门章节。
- 创建阿里云Logstash实例，并安装logstash-input-maxcompute插件。

  具体操作步骤请参见步骤一：创建阿里云Logstash实例和安装或卸载插件。
- 创建目标阿里云Elasticsearch实例，并开启实例的自动创建索引功能。

  具体操作步骤请参见创建阿里云Elasticsearch实例和快速访问与配置。本文以6.7.0版本为例。
- 请确保网络能够互通。即MaxCompute、阿里云Logstash、阿里云Elasticsearch处于同一专有网络VPC（Virtual Private Cloud）下。

  > **说明** 您也可以使用公网环境的服务，前提是需要通过配置NAT网关实现与公网的连通，详情请参见配置NAT公网数据传输。

配置Logstash管道

1. 登录阿里云Elasticsearch控制台。
2. 进入目标实例。
   i. 在顶部菜单栏处，选择地域。
   ii. 在左侧导航栏，单击**Logstash实例**，然后在**Logstash实例**中单击目标实例ID。
3. 在左侧导航栏，单击**管道管理**。
4. 单击**创建管道**。
5. 在**创建管道任务**页面，输入**管道ID**，并进行Config配置。

本文使用的Config配置如下。

```
input {
    maxcompute {
        access_id => "LTAIezFX********"
        access_key => "SCm8xcF3bwdRbGY7AdU1sH********"
        endpoint => "http://service.cn-hangzhou.maxcompute.aliyun-inc.com/api"
        project_name => "mXXX"
        table_name => "sale_detail"
        partition => "sale_date='201911', region='hangzhou'"
        thread_num => 1
        dirty_data_file => "/ssd/1/share/XXXXX.txt"
    }
}
output {
  elasticsearch {
    hosts => ["http://es-cn-4591f5ja9000j****.elasticsearch.aliyuncs.com:9200"]
    index => "odps_index"
    user => "elastic"
    password => "Adm******"
  }
}
```

input参数说明

参数	类型	是否必选	说明
access_id	string	是	您阿里云账号的AccessKey ID。
access_key	string	是	您阿里云账号的Access Key Secret。
endpoint	string	是	MaxCompute对外服务的访问域名，详情请参见各地域Endpoint对照表（外网连接方式）。
project_name	string	是	MaxCompute的项目名称。
table_name	string	是	MaxCompute的表名称。
partition	string	是	分区字段。分区表按照字段来定义，例如：`sale_date='201911'`，`region='hangzhou'`。
thread_num	number	是	线程数，默认为1。
dirty_data_file	string	是	指定文件，用于记录处理失败的日志。

注意

- 以上配置仅作为测试使用，在实际业务中，请按照业务需求进行合理配置。Input插件支持的其他配置选项请参见Logstash Jdbc input plugin。
- logstash-input-maxcompute插件会全量同步MaxCompute中的数据到阿里云Elasticsearch中。

Config配置详情请参见Logstash配置文件说明。

6. 单击**下一步**，配置管道参数。

参数	说明
管道工作线程	并行执行管道的Filter和Output的工作线程数量。当事件出现积压或CPU未饱和时，请考虑增大线程数，更好地使用CPU处理能力。默认值：实例的CPU核数。
管道批大小	单个工作线程在尝试执行Filter和Output前，可以从Input收集的最大事件数目。较大的管道批大小可能会带来较大的内存开销。您可以设置LS_HEAP_SIZE变量，来增大JVM堆大小，从而有效使用该值。默认值：125。
管道批延迟	创建管道事件批时，将过小的批分派给管道工作线程之前，要等候每个事件的时长，单位为毫秒。默认值：50ms。
队列类型	用于事件缓冲的内部排队模型。可选值： ○ **MEMORY**：默认值。基于内存的传统队列。 ○ **PERSISTED**：基于磁盘的ACKed队列（持久队列）。
队列最大字节数	请确保该值小于您的磁盘总容量。默认值：1024 MB。
队列检查点写入数	启用持久性队列时，在强制执行检查点之前已写入事件的最大数目。设置为0，表示无限制。默认值：1024。

警告 配置完成后，需要保存并部署才能生效。保存并部署操作会触发实例重启，请在不影响业务的前提下，继续执行以下步骤。

7. 单击**保存**或者**保存并部署**。
   - **保存**：将管道信息保存在Logstash里并触发实例变更，配置不会生效。保存后，系统会返回**管道管理**页面。可在**管道列表**区域，单击**操作**列下的**立即部署**，触发实例重启，使配置生效。
   - **保存并部署**：保存并且部署后，会触发实例重启，使配置生效。

验证结果

1. 登录目标阿里云Elasticsearch实例的Kibana控制台。

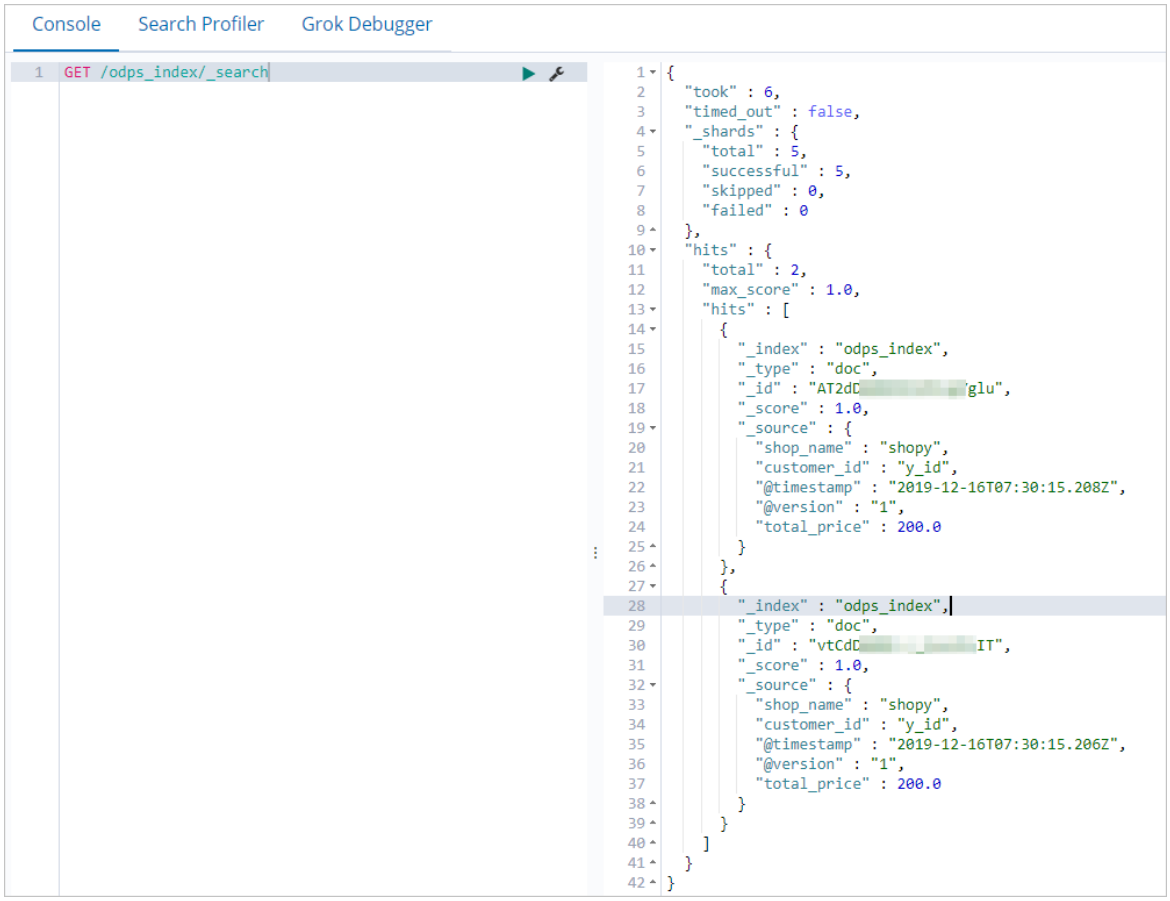
   具体步骤请参见登录Kibana控制台。

2. 在左侧导航栏，单击**Dev Tools**（开发工具）。

3. 在**Console**中，执行以下命令，查看同步成功的索引数据。

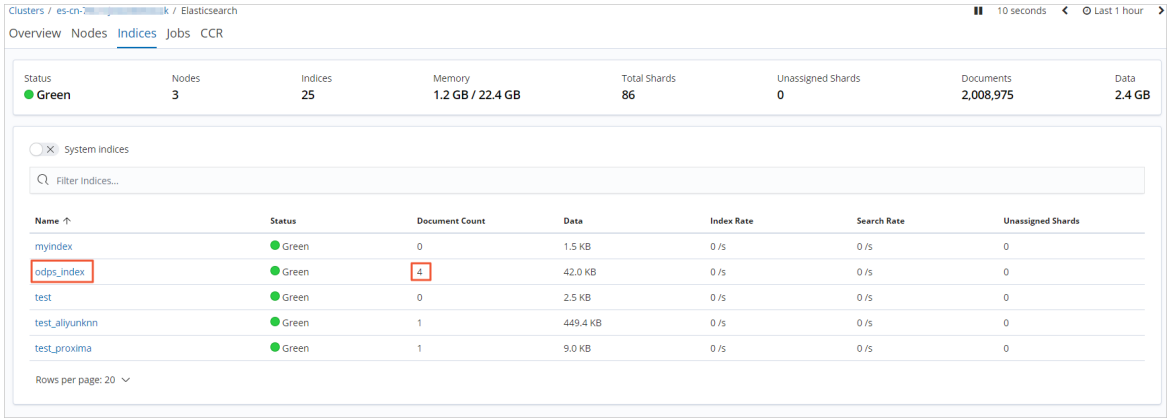
```
GET /odps_index/_search
```

运行成功后，结果如下。



> **说明** 如果运行失败，可在**日志查询**页面查看相关日志进行排查修复，详情请参见查询日志。

4. 切换到**Monitoring**（监控）页面，单击**Indices**（索引）。
5. 在**Indices**页面，查看同步成功的索引，以及写入文档的数量。



常见问题

Logstash数据写入问题排查方案

4.3. 通过实时计算处理数据并同步到Elasticsearch

Flink es

当您需要构建一个日志检索系统时，可通过实时计算Flink对日志数据进行计算后，输出到Elasticsearch进行搜索。本文以阿里云日志服务SLS（Log Service）为例，为您介绍具体的实现方法。

前提条件

您已完成以下操作：

- 开通阿里云实时计算服务并创建项目。

  具体操作，请参见开通服务和创建项目。

- 创建阿里云Elasticsearch实例。

  具体操作，请参见创建阿里云Elasticsearch实例。

- 开通SLS服务、创建Project和Logstore。

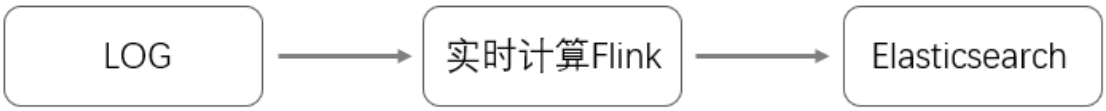
  具体操作，请参见开通阿里云日志服务、创建Project和创建Logstore。

背景信息

阿里云实时计算Flink是阿里云官方支持的Flink产品，支持包括Kafka、Elasticsearch等多种输入输出系统。实时计算Flink与Elasticsearch结合，能够满足典型的日志检索场景。

Kafka或LOG等系统中的日志，经过Flink进行简单或者复杂计算之后，输出到Elasticsearch进行搜索。结合Flink的强大计算能力与Elasticsearch的强大搜索能力，可为业务提供实时数据加工及查询，助力业务实时化转型。

实时计算Flink为您提供了非常简单的方式来对接Elasticsearch。例如当前业务中的日志或者数据被写入了LOG中，并且需要对LOG中的数据进行计算之后再写到Elasticsearch中进行搜索，可通过以下链路实现。



操作步骤

1. 登录实时计算控制台。
2. 创建实时计算作业。

   具体操作，请参见创建作业。

3. 编写Flink SQL。

i. 创建日志服务LOG源表。

```
create table sls_stream(
  a int,
  b int,
  c VARCHAR
)
WITH (
  type ='sls',
  endPoint ='<yourEndpoint>',
  accessId ='<yourAccessId>',
  accessKey ='<yourAccessKey>',
  startTime = '<yourStartTime>',
  project ='<yourProjectName>',
  logStore ='<yourLogStoreName>',
  consumerGroup ='<yourConsumerGroupName>'
);
```

WITH参数说明如下表。

变量	说明
endPoint	阿里云日志服务的公网服务入口，即访问对应LOG项目及其内部日志数据的URL。详细信息，请参见服务入口。 例如杭州区域的日志服务入口为：**http://cn-hangzhou.log.aliyuncs.com**。需要在对应的服务入口前加**http://**。
accessId	您账号的AccessKey ID。
accessKey	您账号的AccessKey Secret。
startTime	消费日志开始的时间点。运行Flink作业时所选时间要大于此处设置的时间。
project	LogService的项目名称。
logStore	LogService项目下具体的LogStore名称。
consumerGroup	日志服务的消费组名称。

更多WITH参数及其说明，请参见创建日志服务SLS源表。

ii. 创建Elasticsearch结果表。

> **注意**
>
> - 实时计算3.2.2及以上版本增加了Elasticsearch结果表功能。创建Flink作业时，请注意所选的版本。
> - Elasticsearch结果表的实现使用了REST API，可以兼容Elasticsearch的各个版本。

```
CREATE TABLE es_stream_sink(
  a int,
  cnt BIGINT,
  PRIMARY KEY(a)
)
WITH(
  type ='elasticsearch',
  endPoint = 'http://<instanceid>.public.elasticsearch.aliyuncs.com:<port>',
  accessId = '<yourAccessId>',
  accessKey = '<yourAccessSecret>',
  index = '<yourIndex>',
  typeName = '<yourTypeName>'
);
```

WITH参数说明如下。

参数	说明
endPoint	阿里云Elasticsearch实例的公网地址，格式为http://<instanceid>.public.elasticsearch.aliyuncs.com:9200。可在实例的基本信息页面获取，详细信息请参见查看实例的基本信息。
accessId	访问阿里云Elasticsearch实例的用户名，默认为elastic。
accessKey	对应用户的密码。elastic用户的密码在创建实例时设定，如果忘记可进行重置，重置密码的注意事项和操作步骤，请参见重置实例访问密码。
index	索引名称。如果您还未创建过索引，需要先创建一个索引。具体操作，请参见步骤三：创建索引。您也可以开启自动创建索引功能，自动创建对应索引。具体操作，请参见配置YML参数。
typeName	索引类型。7.0及以上版本的Elasticsearch实例必须为_doc。

更多WITH参数及其说明，请参见创建Elasticsearch结果表。

说明

- Elasticsearch支持根据PRIMARY KEY更新文档，且 `PRIMARY KEY` 只能为1个字段。指定 `PRIMARY KEY` 后，文档的ID为 `PRIMARY KEY` 字段的值。未指定 `PRIMARY KEY` ，文档的ID由系统随机生成。详细信息，请参见Index API。
- Elasticsearch支持多种更新模式，对应WITH中的参数为updateMode：
  - 当 `updateMode=full` 时，新增的文档会完全覆盖已存在的文档。
  - 当 `updateMode=inc` 时，Elasticsearch会根据输入的字段值更新对应的字段。
- Elasticsearch所有的更新默认为UPSERT语义，即INSERT或UPDATE。

iii. 处理业务逻辑并同步数据。

```
INSERT INTO es_stream_sink
SELECT
  a,
  count(*) as cnt
FROM sls_stream GROUP BY a
```

4. 上线并启动作业。

   具体操作步骤请参见作业上线和生产运维。

   上线并启动作业后，即可将日志服务中的数据进行简单聚合后写入阿里云Elasticsearch中。实时计算Flink还支持更多的计算操作，详细信息请参见概述。

更多信息

使用实时计算Flink+Elasticsearch，可帮助您快速创建实时搜索链路。如果您有更复杂的Elasticsearch写入需求，可以使用实时计算Flink的自定义Sink功能来实现。详细信息，请参见创建自定义结果表。

4.4. 通过DataWorks将Hadoop数据同步至Elasticsearch

当您基于Hadoop进行交互式大数据分析查询，遇到查询延迟的问题时，可以将数据同步至阿里云Elasticsearch中再进行查询分析。Elasticsearch对于多种查询类型，特别是即席查询（Ad Hoc），基本可以达到秒级响应。本文介绍如何通过DataWorks的数据同步功能，将Hadoop数据同步到阿里云Elasticsearch中，并进行搜索分析。

操作流程

1. 准备工作

   搭建Hadoop集群、创建DataWorks工作空间、创建与配置阿里云Elasticsearch实例。

2. 步骤一：准备数据

   在Hadoop集群中创建测试数据。

3. 步骤二：购买并创建独享资源组

   购买并创建一个数据集成独享资源组，并为该资源组绑定专有网络和工作空间。独享资源组可以保障数据快速、稳定地传输。

4. 步骤三：添加数据源

   将Elasticsearch和Hadoop的HDFS数据源接入DataWorks的数据集成服务中。

5. 步骤四：配置并运行数据同步任务

   通过向导模式配置数据同步任务，将数据集成系统同步成功的数据存储到Elasticsearch中。将独享资源组作为一个可以执行任务的资源，注册到DataWorks的数据集成服务中。这个资源组将获取数据源的数据，并执行将数据写入Elasticsearch中的任务（该任务将由数据集成系统统一下发）。

6. 步骤五：验证数据同步结果

   在Kibana控制台中，查看同步成功的数据，并按条件查询数据。

准备工作

1. 搭建Hadoop集群。

进行数据同步前，请确保您的Hadoop集群环境正常。本文使用阿里云E-MapReduce服务自动化搭建Hadoop集群。详细信息，请参见创建集群。

E-MapReduce Hadoop集群配置信息如下（未提到的信息，本文均保持默认，您也可以根据自身需求修改配置）：

- **集群类型**：Hadoop
- **产品版本**：EMR-3.26.3
- **挂载公网**：开启

2. 创建阿里云Elasticsearch实例，并开启实例的自动创建索引功能。

   具体操作，请参见创建阿里云Elasticsearch实例和快速访问与配置。创建实例时，请选择与E-MapReduce集群相同的区域、可用区以及专有网络。本文使用的阿里云Elasticsearch版本为通用商业版6.7.0。

3. 创建DataWorks工作空间。

   创建工作区间时，所选区域需要与阿里云Elasticsearch一致，具体操作，请参见创建工作空间。

步骤一：准备数据

1. 进入E-MapReduce控制台。
2. 在顶部菜单栏，选择地域。
3. 在上方菜单栏，单击**数据开发**。
4. 在**数据开发**页面，新建一个数据开发项目，其中资源组选择默认资源组。

   具体操作，请参见项目管理。

5. 在**项目列表**中，单击目标项目右侧**操作**列下的**作业编辑**，新建一个作业。

   具体操作，请参见作业编辑。其中作业类型选择Hive。

6. 创建数据表并插入数据。

   i. 在代码编辑区域中，输入Hive建表语句，单击**运行**。

   本文档使用的建表语句如下。

```
CREATE TABLE IF NOT
EXISTS hive_esdoc_good_sale(
 create_time timestamp,
 category STRING,
 brand STRING,
 buyer_id STRING,
 trans_num BIGINT,
 trans_amount DOUBLE,
 click_cnt BIGINT
 )
 PARTITIONED BY (pt string) ROW FORMAT
 DELIMITED FIELDS TERMINATED BY ',' lines terminated by '\n'
```

   ii. 在**运行作业**对话框中配置运行参数，单击**确定**。

   - 资源组：选择默认资源组。
   - 执行集群：选择您已创建的集群。

iii. 重新新建一个作业，输入如下SQL语句，插入测试数据。

您可以选择从OSS或其他数据源导入测试数据，也可以手动插入少量的测试数据。本文使用手动插入数据的方法，脚本如下。

```
insert into
hive_esdoc_good_sale PARTITION(pt =1 ) values('2018-08-21','外套','品牌A','lilei',3,5
00.6,7),('2018-08-22','生鲜','品牌B','lilei',1,303,8),('2018-08-22','外套','品牌C','ha
nmeimei',2,510,2),(2018-08-22,'卫浴','品牌A','hanmeimei',1,442.5,1),('2018-08-22','生
鲜','品牌D','hanmeimei',2,234,3),('2018-08-23','外套','品牌B','jimmy',9,2000,7),('2018
-08-23','生鲜','品牌A','jimmy',5,45.1,5),('2018-08-23','外套','品牌E','jimmy',5,100.2,
4),('2018-08-24','生鲜','品牌G','peiqi',10,5560,7),('2018-08-24','卫浴','品牌F','peiqi
',1,445.6,2),('2018-08-24','外套','品牌A','ray',3,777,3),('2018-08-24','卫浴','品牌G',
'ray',3,122,3),('2018-08-24','外套','品牌C','ray',1,62,7) ;
```

7. 查看数据是否插入成功。

i. 新建一个临时查询作业。

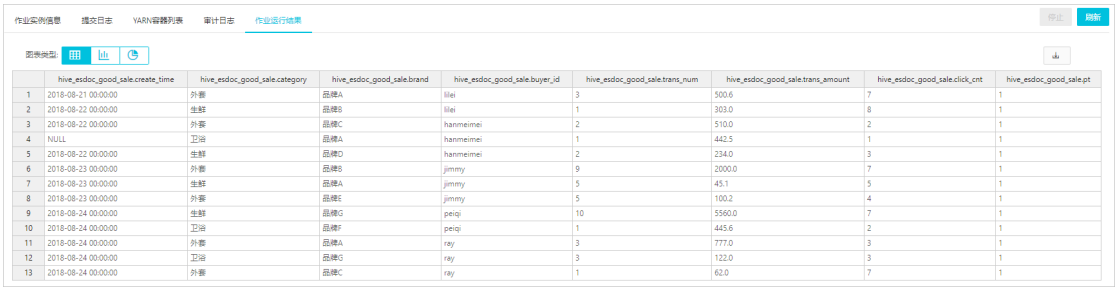
具体操作，请参见临时查询。

ii. 输入如下SQL语句，单击**运行**。

```
select * from hive_esdoc_good_sale where pt =1;
```

iii. 在页面下方，单击**运行记录**，再单击**操作**列下的**详情**。

iv. 在**运维中心**，单击**作业运行结果**。
此操作可以检查Hadoop集群表中是否已存在数据可用于同步，运行成功后的结果如下。



	hive_esdoc_good_sale.create_time	hive_esdoc_good_sale.category	hive_esdoc_good_sale.brand	hive_esdoc_good_sale.buyer_id	hive_esdoc_good_sale.trans_num	hive_esdoc_good_sale.trans_amount	hive_esdoc_good_sale.click_cnt	hive_esdoc_good_sale.pt
1	2018-08-21 00:00:00	外套	品牌A	lilei	3	500.6	7	1
2	2018-08-22 00:00:00	生鲜	品牌B	lilei	1	303.0	8	1
3	2018-08-22 00:00:00	外套	品牌C	hanmeimei	2	510.0	2	1
4	NULL	卫浴	品牌A	hanmeimei	1	442.5	1	1
5	2018-08-22 00:00:00	生鲜	品牌D	hanmeimei	2	234.0	3	1
6	2018-08-23 00:00:00	外套	品牌B	jimmy	9	2000.0	7	1
7	2018-08-23 00:00:00	生鲜	品牌A	jimmy	5	45.1	5	1
8	2018-08-23 00:00:00	外套	品牌E	jimmy	5	100.2	4	1
9	2018-08-24 00:00:00	生鲜	品牌G	peiqi	10	5560.0	7	1
10	2018-08-24 00:00:00	卫浴	品牌F	peiqi	1	445.6	2	1
11	2018-08-24 00:00:00	外套	品牌A	ray	3	777.0	3	1
12	2018-08-24 00:00:00	卫浴	品牌G	ray	3	122.0	3	1
13	2018-08-24 00:00:00	外套	品牌C	ray	1	62.0	7	1

步骤二：购买并创建独享资源组

1. 登录DataWorks控制台。
2. 选择相应地域后，在左侧导航栏，单击**资源组列表**。
3. 参见购买资源组（创建订单），购买独享数据集成资源。

> **注意** 购买时，所选地域需要与目标工作空间保持一致。

4. 参见新增和使用独享数据集成资源组，创建一个独享数据集成资源。

本文使用的配置如下，其中**资源组类型**选择**独享数据集成资源组**。

5. 单击已创建的独享资源组右侧的**网络设置**，参见绑定专有网络，为该独享资源组绑定专有网络。

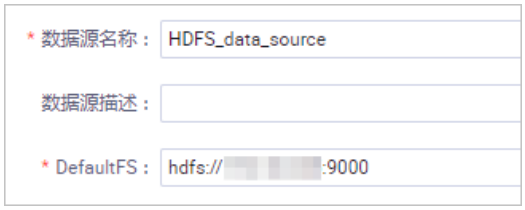
   独享资源部署在DataWorks托管的专有网络中。DataWorks需要与Hadoop集群和Elasticsearch实例的专有网络连通才能同步数据。而Hadoop集群和Elasticsearch实例在同一专有网络下，因此在绑定专有网络时，选择Elasticsearch实例所在**专有网络**和**交换机**即可。

6. 单击已创建的独享资源组右侧的**修改归属工作空间**，参见新增和使用独享数据集成资源组，为该独享资源

组绑定目标工作空间。

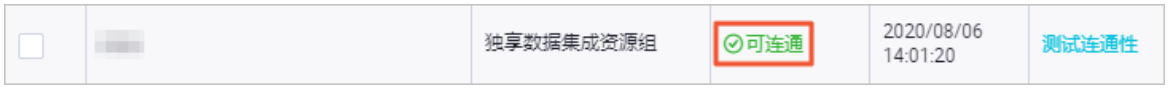
步骤三：添加数据源

1. 进入DataWorks的**数据集成**页面。
   i. 在DataWorks控制台的左侧导航栏，单击**工作空间列表**。
   ii. 找到目标工作空间，单击其右侧**操作**列下的**进入数据集成**。
2. 在左侧导航栏，选择**数据源 > 数据源列表**。
3. 在**数据源管理**页面，单击**新增数据源**。
4. 在**新增数据源**对话框的**半结构化存储**区域中，单击**HDFS**。
5. 在**新增HDFS数据源**对话框中，填写**数据源名称**和**Default FS**。

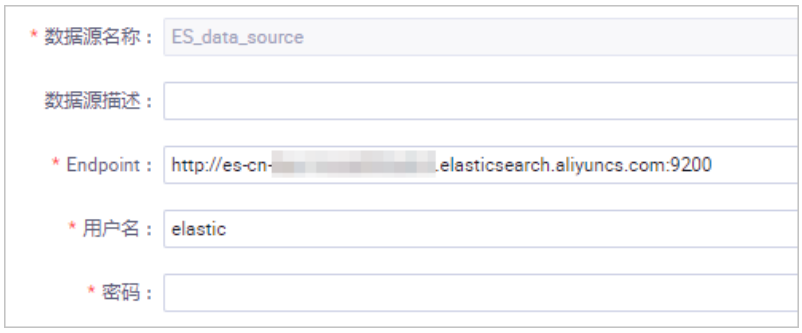


Default FS：对于E-MapReduce Hadoop集群而言，如果Hadoop集群为非HA集群，则此处地址为 `hdfs://emr-header-1的IP:9000` 。如果Hadoop集群为HA集群，则此处地址为 `hdfs://emr-header-1的IP:8020` 。在本文中，emr-header-1与DataWorks通过专有网络连接，因此此处填写内网IP。

配置完成后，可与独享资源组进行连通性测试。**连通状态**显示为**可连通**时，表示连通成功。



6. 单击**完成**。
7. 使用同样的方式添加Elasticsearch数据源。



参数	说明
Endpoint	阿里云Elasticsearch的访问地址，格式为：`http://<实例的内网或公网地址>:9200` 。实例的内网或公网地址可在基本信息页面获取，详细信息，请参见查看实例的基本信息。 **注意** 如果您使用的是公网地址，需要将独享资源组的EIP地址添加到阿里云Elasticsearch的公网地址访问白名单中，详情请参见配置实例公网或私网访问白名单和使用独享数据集成资源组执行任务需要在数据库添加的IP白名单。

参数	说明
用户名	访问阿里云Elasticsearch实例的用户名，默认为elastic。
密码	对应用户的密码。elastic用户的密码在创建实例时设定，如果忘记可重置，重置密码的注意事项和操作步骤，请参见重置实例访问密码。

> **说明** 其他未提及的参数请自定义输入。

步骤四：配置并运行数据同步任务

1. 在DataWorks的数据开发页面，新建一个业务流程。

   具体操作步骤请参见管理业务流程。

2. 新建一个离线同步任务。

   i. 展开新建的业务流程，右键单击**数据集成**，选择**新建 > 离线同步**。

   ii. 在**新建节点**对话框中，输入**节点名称**，单击**提交**。

3. 在**选择数据源**区域中，将**数据来源**指定为HDFS数据源，并选择待同步的文件；将**数据去向**指定为Elasticsearch数据源，并填入索引名和索引类型。

> **说明**
>
> - 您也可以使用脚本模式配置数据同步，详情请参见通过脚本模式配置离线同步任务、DRDS Reader和Elasticsearch Writer。
> - 建议在Elasticsearch数据源的**高级配置**下，将**启用节点发现**设置为否，否则同步过程中提示连接超时。

4. 在**字段映射**区域中，设置**来源字段**与**目标字段**的映射关系。
5. 在**通道控制**区域中，配置执行任务的相关参数。
6. 配置任务调度属性。

   在页面右侧，单击**调度配置**，按照需求配置相应的调度参数。各配置的详细说明请参见调度配置章节。

> 注意
>
> - 在提交任务前，必须配置任务调度**依赖的上游节点**，详情请参见配置同周期调度依赖。
> - 如果您希望对任务进行周期性调度，需要配置任务的**时间属性**，包括任务的具体执行时间、调度周期、生效周期、重跑属性等。
> - 周期任务将于配置任务开始的第二天00:00，按照您的配置规则生效执行。

7. 配置执行同步任务所使用的资源组。

   i. 在页面右侧，单击**数据集成资源组配置**。

   ii. 选择**独享数据集成资源组**为您创建的独享资源组。

8. 提交任务。

   i. 保存当前配置，单击图标。

   ii. 在**提交新版本**对话框中，填入**备注**。

   iii. 单击**确认**。

9. 单击图标，运行任务。

   任务运行过程中，可查看运行日志。运行成功后，显示如下结果。

```
2020-05-14 17:09:23 [INFO] Sandbox context cleanup temp file success.
2020-05-14 17:09:23 [INFO] Data synchronization ended with return code: [0].
2020-05-14 17:09:23 INFO ================================================================
2020-05-14 17:09:23 INFO Exit code of the Shell command 0
2020-05-14 17:09:23 INFO --- Invocation of Shell command completed ---
2020-05-14 17:09:23 INFO Shell run successfully!
2020-05-14 17:09:23 INFO Current task status: FINISH
2020-05-14 17:09:23 INFO Cost time is: 26.854s
```

步骤五：验证数据同步结果

1. 登录目标阿里云Elasticsearch实例的Kibana控制台。

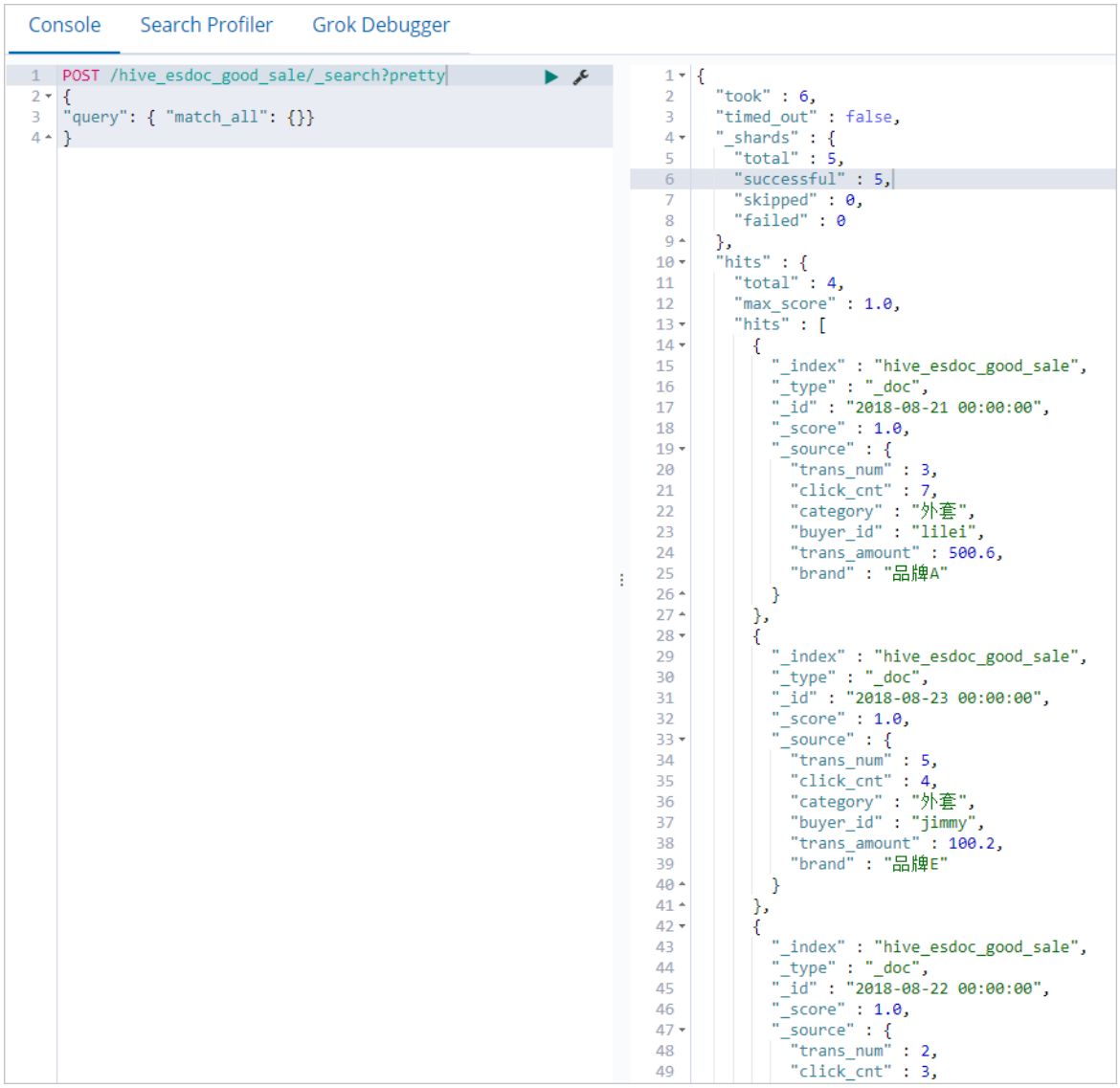
   具体操作，请参见登录Kibana控制台。

2. 在左侧导航栏，单击**Dev Tools**（开发工具）。
3. 在**Console**中，执行如下命令查看同步的数据。

```
POST /hive_esdoc_good_sale/_search?pretty
{
"query": { "match_all": {}}
}
```

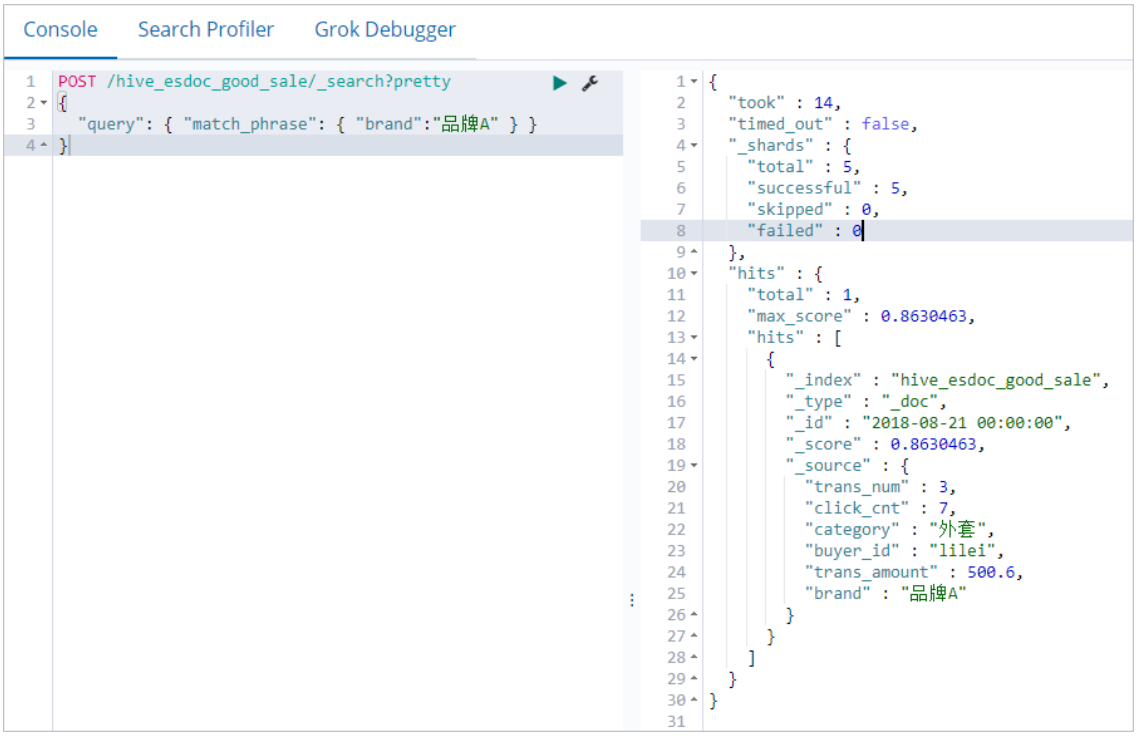
> **说明** `hive_esdoc_good_sale` 为您在数据同步脚本中设置的 `index` 字段的值。

数据同步成功后，返回如下结果。



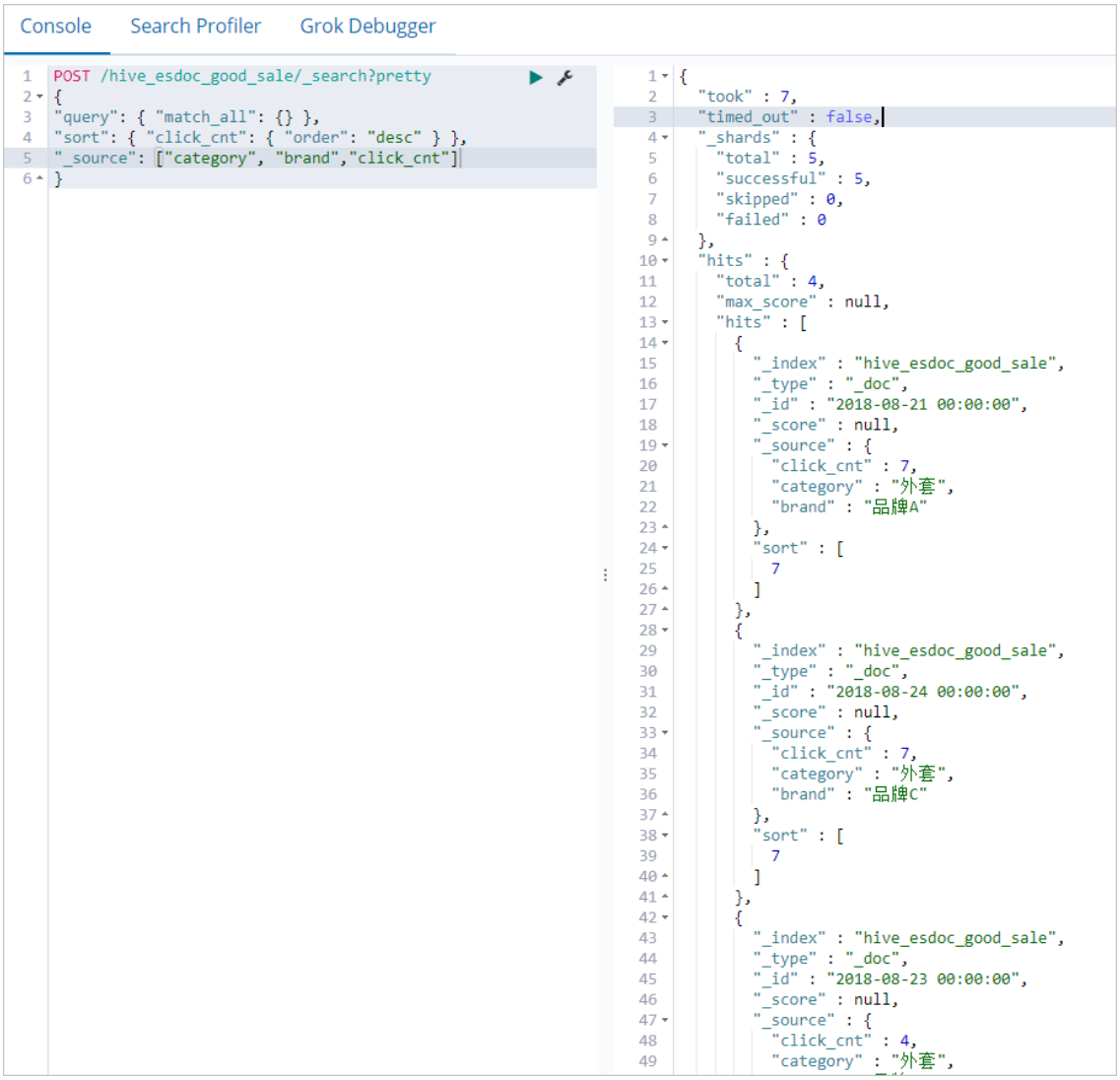
4. 执行如下命令，搜索品牌为A的所有文档。

```
POST /hive_esdoc_good_sale/_search?pretty
{
  "query": { "match_phrase": { "brand":"品牌A" } }
}
```



5. 执行如下命令，按照**点击次数**排序，判断各品牌产品的热度。

```
POST /hive_esdoc_good_sale/_search?pretty
{
"query": { "match_all": {} },
"sort": { "click_cnt": { "order": "desc" } },
"_source": ["category", "brand","click_cnt"]
}
```



更多命令和访问方式，请参见阿里云Elasticsearch官方文档和Elasticsearch官方文档。

5.存储产品数据迁移

5.1. 基于Logstash迁移OSS数据

MNS事件通知迁移OSS数据

当对象存储服务OSS（Object Storage Service）文件发生变更，触发阿里云消息服务MNS（Message Notification Service）事件通知时，您可以通过阿里云Logstash的logstash-input-oss插件获取OSS变更事件，再通过logstash-output-oss插件将数据同步到OSS Bucket中。本文介绍对应的配置方法。

前提条件

您已完成以下操作：

- 安装logstash-input-oss和logstash-output-oss插件。

  具体操作请参见安装或卸载插件。
- 开通阿里云OSS服务并准备数据。

  具体操作请参见开通OSS服务。
- 开通消息服务MNS，并确保与OSS服务在同一区域。

  具体操作请参见开通消息服务MNS并授权。

> **说明** 本示例设置事件通知类型为PostObject、PutObject，即触发Post、Put事件请求，logstash-output-oss插件同步此数据到对端OSS。

操作步骤

1. 步骤一：配置事件通知规则
2. 步骤二：配置Logstash管道
3. 步骤三：查看数据同步结果

步骤一：配置事件通知规则

1. 登录OSS管理控制台。
2. 在左侧导航栏，单击**Bucket列表**，再单击目标Bucket名称。
3. 选择**基础设置** > **事件通知**。
4. 在**事件通知**区域，单击**设置**。
5. 单击**创建规则**。
6. 在**创建规则**面板，配置事件通知规则。

   本示例配置的事件通知如下。

配置参数详情请参见设置事件通知规则。

> **说明** 资源描述目录中不能包含特殊字符，更多事件类型请参见事件通知概述。

7. 单击**确定**。
8. （可选）查看事件通知规则。
   i. 进入消息服务MNS控制台。
   ii. 在顶部菜单栏选择地域。
   iii. 在左侧导航栏，单击**事件通知**。

步骤二：配置Logstash管道

1. 登录阿里云Elasticsearch控制台。
2. 进入目标实例。
   i. 在顶部菜单栏处，选择地域。
   ii. 在左侧导航栏，单击**Logstash实例**，然后在**Logstash实例**中单击目标实例ID。
3. 在左侧导航栏，单击**管道管理**。
4. 单击**创建管道**。
5. 在**创建管道任务**页面，输入管道ID并配置管道。

   本文使用的管道配置如下。

```
input {
  oss {
    endpoint => "oss-cn-hangzhou-internal.aliyuncs.com"
    bucket => "zl-ossoutoss"
    access_key_id => "LTAIaX42ddd******"
    access_key_secret => "zuyBRUUnddddddds3e6i******"
    mns_settings => {
       endpoint => "18185036364****.mns.cn-hangzhou-internal.aliyuncs.com"
       queue => "zl-test"
    }
    codec => json {
      charset => "UTF-8"
    }
  }
}
output {
  oss {
    endpoint => "http://oss-cn-hangzhou-internal.aliyuncs.com"
    bucket => "zl-log-output-test"
    access_key_id => "LTAIaxxxxx******"
    access_key_secret => "zuxxxx8hBpXs3e6i******"
    prefix => "oss/database"
    recover => true
    rotation_strategy => "size_and_time"
    time_rotate => 1
    size_rotate => 1000
    temporary_directory => "/ssd/1/ls-cn-0pp1cwec****/logstash/data/22"

    encoding => "gzip"
    additional_oss_settings => {
      max_connections_to_oss => 1024
      secure_connection_enabled => false
    }
  }
}
```

注意

- MNS Endpoint不能加HTTP前缀，并且是internal域名，否则报错。
- 通过time_rotate及size_rotate，设置当临时目录数据保存1分钟或者大小达到1000 Bytes时，触发数据上传。
- input参数说明请参见logstash-input-oss插件参数说明，output参数说明请参见logstash-output-oss插件参数说明。

6. 单击**下一步**，配置管道参数。

参数	说明
管道工作线程	并行执行管道的Filter和Output的工作线程数量。当事件出现积压或CPU未饱和时，请考虑增大线程数，更好地使用CPU处理能力。默认值：实例的CPU核数。
管道批大小	单个工作线程在尝试执行Filter和Output前，可以从Input收集的最大事件数目。较大的管道批大小可能会带来较大的内存开销。您可以设置LS_HEAP_SIZE变量，来增大JVM堆大小，从而有效使用该值。默认值：125。
管道批延迟	创建管道事件批时，将过小的批分派给管道工作线程之前，要等候每个事件的时长，单位为毫秒。默认值：50ms。
队列类型	用于事件缓冲的内部排队模型。可选值： ○ **MEMORY**：默认值。基于内存的传统队列。 ○ **PERSISTED**：基于磁盘的ACKed队列（持久队列）。
队列最大字节数	请确保该值小于您的磁盘总容量。默认值：1024 MB。
队列检查点写入数	启用持久性队列时，在强制执行检查点之前已写入事件的最大数目。设置为0，表示无限制。默认值：1024。

警告 配置完成后，需要保存并部署才能生效。保存并部署操作会触发实例重启，请在不影响业务的前提下，继续执行以下步骤。

7. 单击**保存**或者**保存并部署**。
   - **保存**：将管道信息保存在Logstash里并触发实例变更，配置不会生效。保存后，系统会返回**管道管理**页面。可在**管道列表**区域，单击**操作**列下的**立即部署**，触发实例重启，使配置生效。
   - **保存并部署**：保存并且部署后，会触发实例重启，使配置生效。

步骤三：查看数据同步结果

1. 进入OSS管理控制台。
2. 上传新的文件到MNS监控的目录中（步骤一：配置事件通知规则中定义的**资源描述**）。

   上传文件的具体操作请参见上传文件。
3. 进入目标端OSS Bucket，查看同步成功的数据。

> 注意 OSS插件不是以文档或目录形式进行数据同步，而是按照原文档中的数据一条一条进行读写。根据rotate规则，logstash-output-oss插件先将数据存储在本地临时文件中，当达到一定的时间或者大小再进行上传，所以不能保证同一个文档的数据保存到相同的文档下，例如可能会出现多个文档的部分数据同步到一个文档中的情况。

5.2. 从Solr集群迁移文档至阿里云Elasticsearch

本文介绍通过第三方社区提供的solr-to-es工具，将Solr节点中的文档迁移到阿里云Elasticsearch（简称ES）中的方法。

环境准备

1. 创建阿里云ES实例，要求版本为6.x，本文使用6.3.2 版本，详情请参见创建阿里云Elasticsearch实例。

   > 注意 本文使用的solr-to-es迁移工具仅支持阿里云ES 6.x版本，其他版本需自行测试。

2. 开启目标阿里云ES的自动创建索引功能，详情请参见快速访问与配置。
3. 创建阿里云ECS实例，本文使用CentOS 7.3版本，详情请参见步骤一：创建ECS实例。

   > 注意 ECS实例需要与阿里云ES实例在同一区域和可用区，以及同一专有网络VPC（Virtual Private Cloud）下。

4. 在ECS上安装Solr，本文使用5.0.0版本的Solr，详情请参见Solr官方文档。
5. 在ECS上安装Python，要求3.0及以上版本，本文使用Python 3.6.2。
6. 在ECS上安装PySolr，要求3.3.3及以上，4.0以下版本。

安装solr-to-es工具

1. 连接ECS服务器，下载solr-to-es工具。
2. 进入*setup.py*所在的目录，运行 `python setup.py install` 命令，安装solr-to-es工具。
3. 安装成功后，参考以下命令进行文档迁移。

```
python __main__.py <solr_url>:8983/solr/<my_core>/select http://<username>:<password>
@<elasticsearch_url>:9200 <elasticsearch_index> <doc_type>
```

参数说明

参数	说明
`<solr_url>`	Solr集群的完整访问地址。例如，http://116.62.**.**。
`<my_core>`	迁移文档对应的SolrCore的名称。
`<username>`	阿里云ES的访问用户名，默认为elastic。
`<password>`	阿里云ES的访问密码，在创建实例时设定。

参数	说明
<elasticsearch_url>	阿里云ES实例的内网或外网访问地址。可在实例的基本信息页面获取，详情请参见查看实例的基本信息。
<elasticsearch_index>	待写入的Solr文档对应的索引名称。
<doc_type>	对应索引的类型名称。

注意 如果您使用的是其他版本的solr-to-es工具，也可以尝试使用如下命令进行文档迁移，参数详情请参见solr-to-es。

```
solr-to-es [-h] [--solr-query SOLR_QUERY] [--solr-fields COMMA_SEP_FIELDS]
                [--rows-per-page ROWS_PER_PAGE] [--es-timeout ES_TIMEOUT]
                solr_url elasticsearch_url elasticsearch_index doc_type
```

本案例使用以上命令会输出 `-bash: solr-to-es.py: command not found` 的错误。

操作示例

通过以下命令，查询名称为 `my_core` 的SolrCore的所有文档，写入到阿里云ES实例中。对应的索引为 `elasticsearch_index` ，索引类型为 `doc_type` 。

1. 在Solr环境中，进入*solr-to-es-master/solr_to_es*文件夹下。
2. 执行以下命令。

```
python __main__.py 'http://116.62.**.**:8983/solr/my_core/select?q=*%3A*&wt=json&indent=true' 'http://elastic:替换密码@es-cn-so4lwf40ubsrf****.public.elasticsearch.aliyuncs.com:9200' elasticsearch_index doc_type
```

参数	说明
q	Solr的查询语法，必选，可以使用运算符。 `*%3A*` 表示查询所有文档。
wt	返回的数据类型，支持JSON、XML、Python、Ruby、CSV等格式。
indent	返回结果是否需要格式化展示，默认为 `false` 。

其他参数说明请参见参数说明。

3. 登录目标阿里云ES实例的Kibana控制台。

   登录控制台的具体操作步骤请参见登录Kibana控制台。
4. 单击左侧导航栏的**Dev Tools**（开发工具），在**Console**中执行以下命令，查看阿里云ES集群服务中是否已成功创建 `elasticsearch_index` 索引。

```
GET _cat/indices?v
```

5. 执行以下命令，查看迁移成功的文档详情。

```
GET /elasticsearch_index/doc_type/_search
```

查询成功后，返回如下结果。


```
{
  "took" : 12,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 2,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "elasticsearch_index",
        "_type" : "doc_type",
        "_id" : "Tz8WNW4BwRjcQciJ****",
        "_score" : 1.0,
        "_source" : {
          "id" : "2",
          "title" : [
            "test"
          ],
          "_version_" : 1648195017403006976
        }
      },
      {
        "_index" : "elasticsearch_index",
        "_type" : "doc_type",
        "_id" : "Tj8WNW4BwRjcQciJ****",
        "_score" : 1.0,
        "_source" : {
          "id" : "1",
          "title" : [
            "change.me"
          ],
          "_version_" : 1648195007391203328
        }
      }
    ]
  }
}
```

6.ES-Hadoop使用

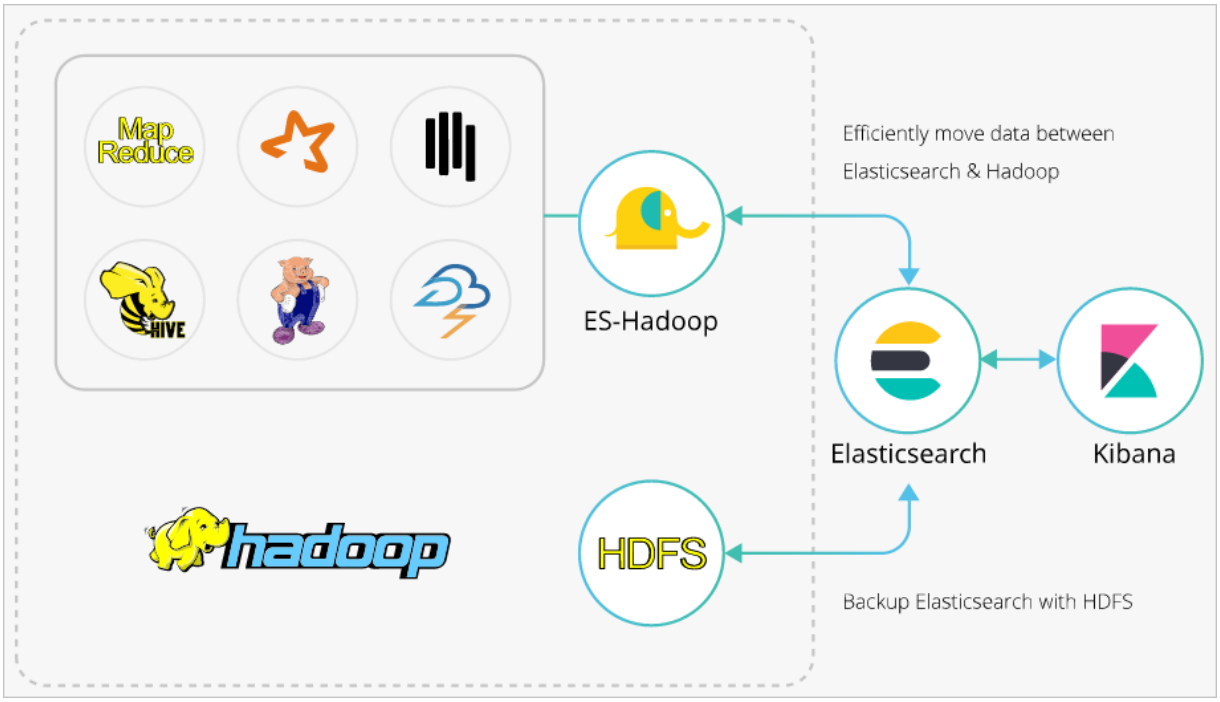
6.1. 通过ES-Hadoop实现Hive读写阿里云Elasticsearch数据

ES-Hadoop是Elasticsearch推出的专门用于对接Hadoop生态的工具，可以让数据在Elasticsearch和Hadoop之间双向移动，无缝衔接Elasticsearch与Hadoop服务，充分使用Elasticsearch的快速搜索及Hadoop批处理能力，实现交互式数据处理。本文介绍如何通过ES-Hadoop实现Hadoop的Hive服务读写阿里云Elasticsearch数据。

背景信息

Hadoop生态的优势是处理大规模数据集，但是其缺点也很明显，就是当用于交互式分析时，查询时延会比较长。而Elasticsearch擅长于交互式分析，对于很多查询类型，特别是对于Ad-hoc查询（即席查询），可以达到秒级。ES-Hadoop的推出提供了一种组合两者优势的可能性。使用ES-Hadoop，您只需要对代码进行很小的改动，即可快速处理存储在Elasticsearch中的数据，并且能够享受到Elasticsearch带来的加速效果。

ES-Hadoop的原理是将Elasticsearch作为MR、Spark或Hive等数据处理引擎的数据源，在计算存储分离的架构中扮演存储的角色。这和 MR、Spark或Hive的数据源并无差异，但相对于这些数据源，Elasticsearch具有更快的数据选择过滤能力。这种能力正是分析引擎最为关键的能力之一。



操作流程

1. 准备工作

   创建同一专有网络下的阿里云Elasticsearch和E-MapReduce（以下简称EMR）实例、关闭Elasticsearch实例的自动创建索引功能并创建索引和Mapping、下载与Elasticsearch实例版本一致的ES-Hadoop安装包。

2. 步骤一：上传ES-Hadoop JAR包至HDFS

   将已下载的ES-Hadoop安装包上传至EMR Master节点的HDFS目录下。

3. 步骤二：创建Hive外表

   创建Hive外表，与Elasticsearch索引中的字段进行映射。

4. 步骤三：通过Hive写入索引数据

   通过HiveSQL，向Elasticsearch实例的索引中写入数据。

5. 步骤四：通过Hive读取索引数据

   通过HiveSQL，读取Elasticsearch实例中的索引数据。

准备工作

1. 创建阿里云Elasticsearch实例。

   本文使用6.7.0版本的实例，具体操作步骤请参见创建阿里云Elasticsearch实例。

2. 关闭实例的自动创建索引功能，并提前创建索引和Mapping。

   开启自动创建索引功能后，可能会导致Elasticsearch自动创建的索引类型和您预期的类型不一致。例如您定义了一个字段age，为INT类型，开启自动创建索引后，可能将其索引成了LONG类型，因此建议手动创建索引。本文使用的索引和Mapping如下。

```
PUT company
{
  "mappings": {
    "_doc": {
      "properties": {
        "id": {
          "type": "long"
        },
        "name": {
          "type": "text",
          "fields": {
            "keyword": {
              "type": "keyword",
              "ignore_above": 256
            }
          }
        },
        "birth": {
          "type": "text"
        },
        "addr": {
          "type": "text"
        }
      }
    }
  },
  "settings": {
    "index": {
      "number_of_shards": "5",
      "number_of_replicas": "1"
    }
  }
}
```

3. 创建与Elasticsearch实例在同一专有网络下的EMR集群。

> 注意 Elasticsearch实例的私网访问白名单默认为0.0.0.0/0，您可在安全配置页面查看，如果未使用默认配置，您还需要在白名单中加入EMR集群的内网IP地址：
>
> - 请参见查看集群列表与详情，获取EMR集群的内网IP地址。
> - 请参见配置实例公网或私网访问白名单，配置Elasticsearch实例的VPC私网访问白名单。

步骤一：上传ES-Hadoop JAR包至HDFS

1. 下载ES-Hadoop安装包，其版本需要与Elasticsearch实例保持一致。

   本文使用elasticsearch-hadoop-6.7.0.zip。

2. 登录E-MapReduce控制台，获取Master节点的IP地址，并通过SSH登录对应的ECS机器。

   具体操作步骤请参见登录集群。

3. 将已下载的elasticsearch-hadoop-6.7.0.zip上传至Master节点，并解压获得elasticsearch-hadoop-hive-6.7.0.jar。

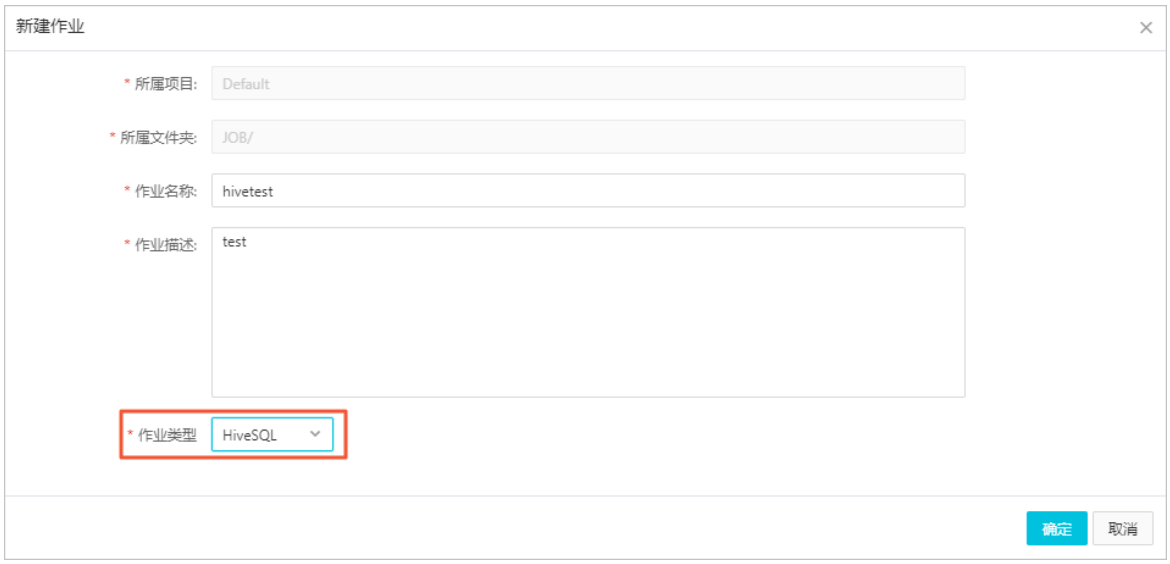
4. 创建HDFS目录，将elasticsearch-hadoop-hive-6.7.0.jar上传至该目录下。

```
hadoop fs -mkdir /tmp/hadoop-es
hadoop fs -put elasticsearch-hadoop-6.7.0/dist/elasticsearch-hadoop-hive-6.7.0.jar /tmp/hadoop-es
```

步骤二：创建Hive外表

1. 在EMR控制台的**数据开发**模块中，创建**HiveSQL**类型的作业。

   具体操作步骤请参见Hive SQL作业配置。



2. 配置作业，创建外表。

   作业配置如下。

```
####添加jar包，仅对当前会话有效########
add jar hdfs:///tmp/hadoop-es/elasticsearch-hadoop-hive-6.7.0.jar;
####创建hive外表，与es索引进行映射#####
CREATE EXTERNAL table IF NOT EXISTS company(
   id BIGINT,
   name STRING,
   birth STRING,
   addr STRING
)
STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'
TBLPROPERTIES(
    'es.nodes' = 'http://es-cn-mp91kzb8m0009****.elasticsearch.aliyuncs.com',
    'es.port' = '9200',
    'es.net.ssl' = 'true',
    'es.nodes.wan.only' = 'true',
    'es.nodes.discovery'='false',
    'es.input.use.sliced.partitions'='false',
    'es.input.json' = 'false',
    'es.resource' = 'company/_doc',
    'es.net.http.auth.user' = 'elastic',
    'es.net.http.auth.pass' = 'xxxxxx'
);
```

ES-Hadoop相关参数说明

参数	默认值	说明
es.nodes	localhost	指定阿里云Elasticsearch实例的访问地址，建议使用内网地址，可在实例的基本信息页面查看，详情请参见查看实例的基本信息。
es.port	9200	Elasticsearch实例的访问端口号。
es.net.http.auth.user	elastic	Elasticsearch实例的访问用户名。 **说明** 如果程序中指定elastic账号访问Elasticsearch服务，后续在修改elastic账号对应密码后需要一些时间来生效，在密码生效期间会影响服务访问，因此不建议通过elastic来访问。建议在Kibana控制台中创建一个符合预期的Role角色用户进行访问，详情请参见通过Elasticsearch X-Pack角色管理实现用户权限管控。
es.net.http.auth.pass	/	Elasticsearch实例的访问密码。
es.nodes.wan.only	false	开启Elasticsearch集群在云上使用虚拟IP进行连接，是否进行节点嗅探： ○ true：设置 ○ false：不设置

参数	默认值	说明
es.nodes.discovery	true	是否禁用节点发现： ○ true：禁用 ○ false：不禁用 注意 使用阿里云Elasticsearch，必须将此参数设置为false。
es.input.use.sliced.partitions	true	是否使用slice分区： ○ true：使用。设置为true，可能会导致索引在预读阶段的时间明显变长，有时会远远超出查询数据所耗费的时间。建议设置为false，以提高查询效率。 ○ false：不使用。
es.index.auto.create	true	通过Hadoop组件向Elasticsearch集群写入数据，是否自动创建不存在的index： ○ true：自动创建 ○ false：不会自动创建
es.resource	/	指定要读写的index和type。
es.mapping.names	/	表字段与Elasticsearch的索引字段名映射。
es.read.metadata	false	操作Elasticsearch字段涉及到_id之类的内部字段，请开启此属性。

更多的ES-Hadoop配置项说明，请参见官方配置说明。

3. 保存并运行作业。

HIVE_SQL FJ-36DA8E09A86AA0D9 作业内容:

```
add jar hdfs:///tmp/hadoop-es/elasticsearch-hadoop-hive-6.7.0.jar;

CREATE EXTERNAL table IF NOT EXISTS company(
    id BIGINT,
    name STRING,
    birth STRING,
    addr STRING
)
STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'
TBLPROPERTIES(
    'es.nodes' = 'http://es-cn-n6w[redacted].elasticsearch.aliyuncs.com',
    'es.port' = '9200',
    'es.net.ssl' = 'true',
    'es.nodes.wan.only' = 'true',
    'es.nodes.discovery'='false',
    'es.input.json' = 'false',
    'es.resource' = 'company/_doc',
    'es.net.http.auth.user' = 'elastic',
    'es.net.http.auth.pass' = '[redacted]'
);
```

运行成功后，结果如下。

步骤三：通过Hive写入索引数据

1. 创建一个**HiveSQL**类型的写数据作业。

   作业配置如下。

```
add jar hdfs:///tmp/hadoop-es/elasticsearch-hadoop-hive-6.7.0.jar;
INSERT INTO TABLE company VALUES (1, "zhangsan", "1990-01-01","No.969, wenyixi Rd, yuhang, hangzhou");
INSERT INTO TABLE company VALUES (2, "lisi", "1991-01-01", "No.556, xixi Rd, xihu, hangzhou");
INSERT INTO TABLE company VALUES (3, "wangwu", "1992-01-01", "No.699 wangshang Rd, binjiang, hangzhou");
```

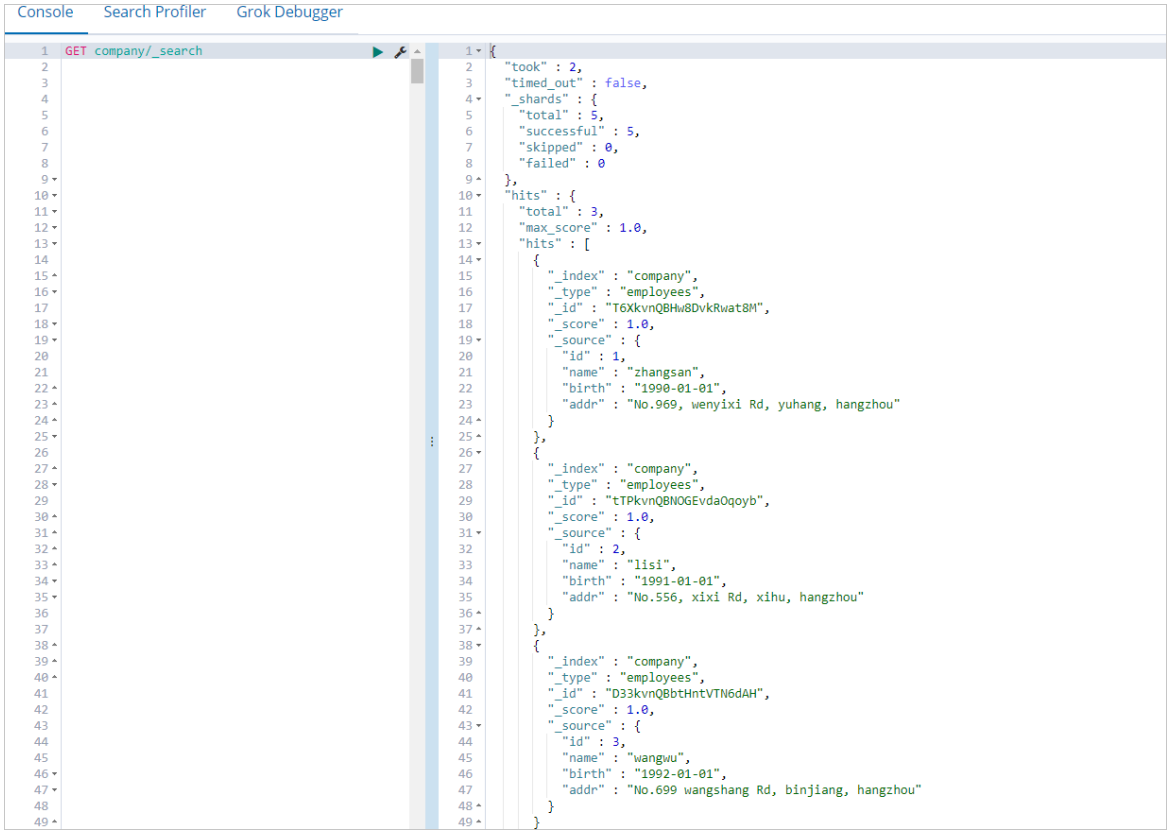
2. 保存并运行作业。

3. 运行成功后，登录Elasticsearch实例的Kibana控制台，查看company索引数据。

   登录Kibana控制台的具体操作步骤，请参见登录Kibana控制台。您可以在Kibana控制台中，执行以下命令查看company索引数据。

```
GET company/_search
```

执行成功后，返回结果如下。



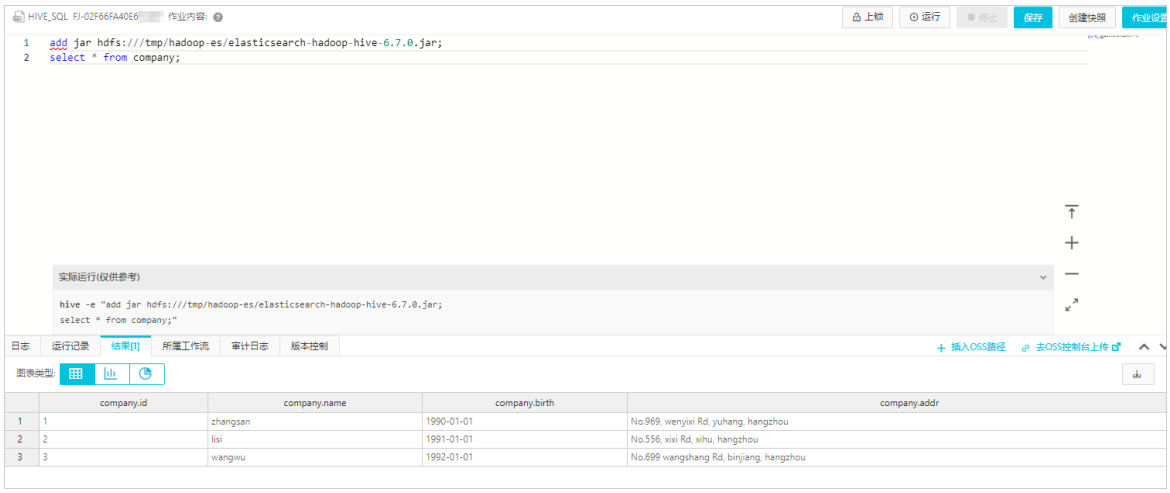
步骤四：通过Hive读取索引数据

1. 创建一个HiveSQL类型的读数据作业。

   作业配置如下。

   ```
   add jar hdfs:///tmp/hadoop-es/elasticsearch-hadoop-hive-6.7.0.jar;
   select * from company;
   ```

2. 保存并运行作业。



总结

本文以阿里云EMR和Elasticsearch为例，介绍了如何通过Elasticsearch强大的ES-Hadoop组件，在Hive上进行数据的查询和写入，可以帮助您将Elasticsearch与Hadoop生态组件结合起来，实现更灵活的数据分析。如果您需要了解ES-Hadoop与Hive更高级的配置，请参见Elasticsearch官方说明文档。

6.2. 通过ES-Hadoop将HDFS中的数据写入Elasticsearch

ES-Hadoop是Elasticsearch推出的专门用于对接Hadoop生态的工具，可以让数据在Elasticsearch和Hadoop之间双向移动，无缝衔接Elasticsearch与Hadoop服务，充分使用Elasticsearch的快速搜索及Hadoop批处理能力，实现交互式数据处理。对于一些较复杂的分析任务，需要通过MapReduce任务读取HDFS上的JSON文件，写入Elasticsearch集群。本文介绍如何通过ES-Hadoop，借助MapReduce任务向Elasticsearch写入数据。

操作流程

1. 准备工作

   创建同一专有网络下的阿里云Elasticsearch和E-MapReduce（以下简称EMR）实例、开启Elasticsearch实例的自动创建索引功能、准备测试数据和Java环境。

2. 步骤一：上传ES-Hadoop JAR包至HDFS

   下载ES-Hadoop安装包，并上传至EMR Master节点的HDFS目录下。

3. 步骤二：配置pom依赖

   创建Java Maven工程，并配置pom依赖。

4. 步骤三：编写并运行MapReduce任务

   编写MapReduce写数据到Elasticsearch的Java代码，并打成Jar包上传至EMR集群，最后运行代码完成写数据任务。

5. 步骤四：验证结果

   在Elasticsearch的Kibana控制台上，查看通过MapReduce写入的数据。

准备工作

1. 创建阿里云Elasticsearch实例，并开启自动创建索引功能。

   具体操作步骤请参见创建阿里云Elasticsearch实例和配置YML参数。本文以6.7.0版本的实例为例。

   > 注意 在生产环境中，建议关闭自动创建索引功能，提前创建好索引和Mapping。由于本文仅用于测试，因此开启了自动创建索引功能。

2. 创建与Elasticsearch实例在同一专有网络下的EMR实例。

   实例配置如下：

   - 产品版本：EMR-3.29.0
   - 必选服务：HDFS(2.8.5)，其他服务保持默认

   具体操作步骤请参见创建集群。

> 注意 Elasticsearch实例的私网访问白名单默认为0.0.0.0/0，您可在安全配置页面查看，如果未使用默认配置，您还需要在白名单中加入EMR集群的内网IP地址：
>
> - 请参见查看集群列表与详情，获取EMR集群的内网IP地址。
> - 请参见配置实例公网或私网访问白名单，配置Elasticsearch实例的VPC私网访问白名单。

3. 准备JSON测试数据，将其写入到*map.json*文件中，并上传至HDFS的*/tmp/hadoop-es*目录下。

   本文使用的测试数据如下。

```
{"id": 1, "name": "zhangsan", "birth": "1990-01-01", "addr": "No.969, wenyixi Rd, yuhang, hangzhou"}
{"id": 2, "name": "lisi", "birth": "1991-01-01", "addr": "No.556, xixi Rd, xihu, hangzhou"}
{"id": 3, "name": "wangwu", "birth": "1992-01-01", "addr": "No.699 wangshang Rd, binjiang, hangzhou"}
```

4. 准备Java环境，要求JDK版本为1.8.0及以上。

步骤一：上传ES-Hadoop JAR包至HDFS

1. 下载ES-Hadoop安装包，其版本需要与Elasticsearch实例保持一致。

   本文使用elasticsearch-hadoop-6.7.0.zip。

2. 登录E-MapReduce控制台，获取Master节点的IP地址，并通过SSH登录对应的ECS机器。

   具体操作步骤请参见登录集群。

3. 将已下载的elasticsearch-hadoop-6.7.0.zip上传至Master节点，并解压获得elasticsearch-hadoop-6.7.0.jar。

4. 创建HDFS目录，将elasticsearch-hadoop-6.7.0.jar上传至该目录下。

```
hadoop fs -mkdir /tmp/hadoop-es
hadoop fs -put elasticsearch-hadoop-6.7.0/dist/elasticsearch-hadoop-6.7.0.jar /tmp/hadoop-es
```

步骤二：配置pom依赖

创建Java Maven工程，并将如下的pom依赖添加到Java工程的pom.xml文件中。

```
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-shade-plugin</artifactId>
            <version>2.4.1</version>
            <executions>
                <execution>
                    <phase>package</phase>
                    <goals>
                        <goal>shade</goal>
                    </goals>
                    <configuration>
                        <transformers>
                            <transformer
                                    implementation="org.apache.maven.plugins.shade.resource.
```

```
ManifestResourceTransformer">
                                <mainClass>WriteToEsWithMR</mainClass>
                            </transformer>
                        </transformers>
                    </configuration>
                </execution>
            </executions>
        </plugin>
    </plugins>
</build>
<dependencies>
    <dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-hdfs</artifactId>
        <version>2.8.5</version>
    </dependency>
    <dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-mapreduce-client-jobclient</artifactId>
        <version>2.8.5</version>
    </dependency>
    <dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-common</artifactId>
        <version>2.8.5</version>
    </dependency>
    <dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-auth</artifactId>
        <version>2.8.5</version>
    </dependency>
    <dependency>
        <groupId>org.elasticsearch</groupId>
        <artifactId>elasticsearch-hadoop-mr</artifactId>
        <version>6.7.0</version>
    </dependency>
    <dependency>
        <groupId>commons-httpclient</groupId>
        <artifactId>commons-httpclient</artifactId>
        <version>3.1</version>
    </dependency>
</dependencies>
```

注意 请确保pom依赖中版本与云服务对应版本保持一致，例如elasticsearch-hadoop-mr版本与阿里云Elasticsearch版本一致；hadoop-hdfs与HDFS版本一致。

步骤三：编写并运行MapReduce任务

1. 编写示例代码。

   以下代码会读取HDFS上*/tmp/hadoop-es*目录下的JSON文件，并将这些JSON文件中的每一行作为一个文档写入Elasticsearch。写入过程由EsOutputFormat在Map阶段完成。

```
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.util.GenericOptionsParser;
import org.elasticsearch.hadoop.mr.EsOutputFormat;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;
public class WriteToEsWithMR extends Configured implements Tool {
    public static class EsMapper extends Mapper<Object, Text, NullWritable, Text> {
        private Text doc = new Text();
        @Override
        protected void map(Object key, Text value, Context context) throws IOException,
InterruptedException {
            if (value.getLength() > 0) {
                doc.set(value);
                System.out.println(value);
                context.write(NullWritable.get(), doc);
            }
        }
    }
    public int run(String[] args) throws Exception {
        Configuration conf = new Configuration();
        String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
        conf.setBoolean("mapreduce.map.speculative", false);
        conf.setBoolean("mapreduce.reduce.speculative", false);
        conf.set("es.nodes", "es-cn-4591jumei000u****.elasticsearch.aliyuncs.com");
        conf.set("es.port","9200");
        conf.set("es.net.http.auth.user", "elastic");
        conf.set("es.net.http.auth.pass", "xxxxxx");
        conf.set("es.nodes.wan.only", "true");
        conf.set("es.nodes.discovery","false");
        conf.set("es.input.use.sliced.partitions","false");
        conf.set("es.resource", "maptest/_doc");
        conf.set("es.input.json", "true");
        Job job = Job.getInstance(conf);
        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(EsOutputFormat.class);
        job.setMapOutputKeyClass(NullWritable.class);
        job.setMapOutputValueClass(Text.class);
        job.setJarByClass(WriteToEsWithMR.class);
        job.setMapperClass(EsMapper.class);
        FileInputFormat.setInputPaths(job, new Path(otherArgs[0]));
        return job.waitForCompletion(true) ? 0 : 1;
    }
    public static void main(String[] args) throws Exception {
        int ret = ToolRunner.run(new WriteToEsWithMR(), args);
        System.exit(ret);
    }
```

```
}
```

ES-Hadoop相关参数说明

参数	默认值	说明
es.nodes	localhost	指定阿里云Elasticsearch实例的访问地址，建议使用内网地址，可在实例的基本信息页面查看，详情请参见查看实例的基本信息。
es.port	9200	Elasticsearch实例的访问端口号。
es.net.http.auth.user	elastic	Elasticsearch实例的访问用户名。 **说明** 如果程序中指定elastic账号访问Elasticsearch服务，后续在修改elastic账号对应密码后需要一些时间来生效，在密码生效期间会影响服务访问，因此不建议通过elastic来访问。建议在Kibana控制台中创建一个符合预期的Role角色用户进行访问，详情请参见通过Elasticsearch X-Pack角色管理实现用户权限管控。
es.net.http.auth.pass	/	Elasticsearch实例的访问密码。
es.nodes.wan.only	false	开启Elasticsearch集群在云上使用虚拟IP进行连接，是否进行节点嗅探： ○ true：设置 ○ false：不设置
es.nodes.discovery	true	是否禁用节点发现： ○ true：禁用 ○ false：不禁用 **注意** 使用阿里云Elasticsearch，必须将此参数设置为false。
es.input.use.sliced.partitions	true	是否使用slice分区： ○ true：使用。设置为true，可能会导致索引在预读阶段的时间明显变长，有时会远远超出查询数据所耗费的时间。建议设置为false，以提高查询效率。 ○ false：不使用。
es.index.auto.create	true	通过Hadoop组件向Elasticsearch集群写入数据，是否自动创建不存在的index： ○ true：自动创建 ○ false：不会自动创建

参数	默认值	说明
es.resource	/	指定要读写的index和type。
es.input.json	false	输入是否已经是JSON格式： ○ true：是JSON格式 ○ false：不是JSON格式
es.mapping.names	/	表字段与Elasticsearch的索引字段名映射。
es.read.metadata	false	操作Elasticsearch字段涉及到_id之类的内部字段，请开启此属性。

更多的ES-Hadoop配置项说明，请参见官方配置说明。

2. 将代码打成Jar包，上传至EMR客户端机器（例如Gateway或EMR集群主节点）。
3. 在EMR客户端机器上，运行如下命令执行MapReduce程序。

```
hadoop jar es-mapreduce-1.0-SNAPSHOT.jar /tmp/hadoop-es/map.json
```

> 说明 es-mapreduce-1.0-SNAPSHOT.jar需要替换为您已上传的Jar包名称。

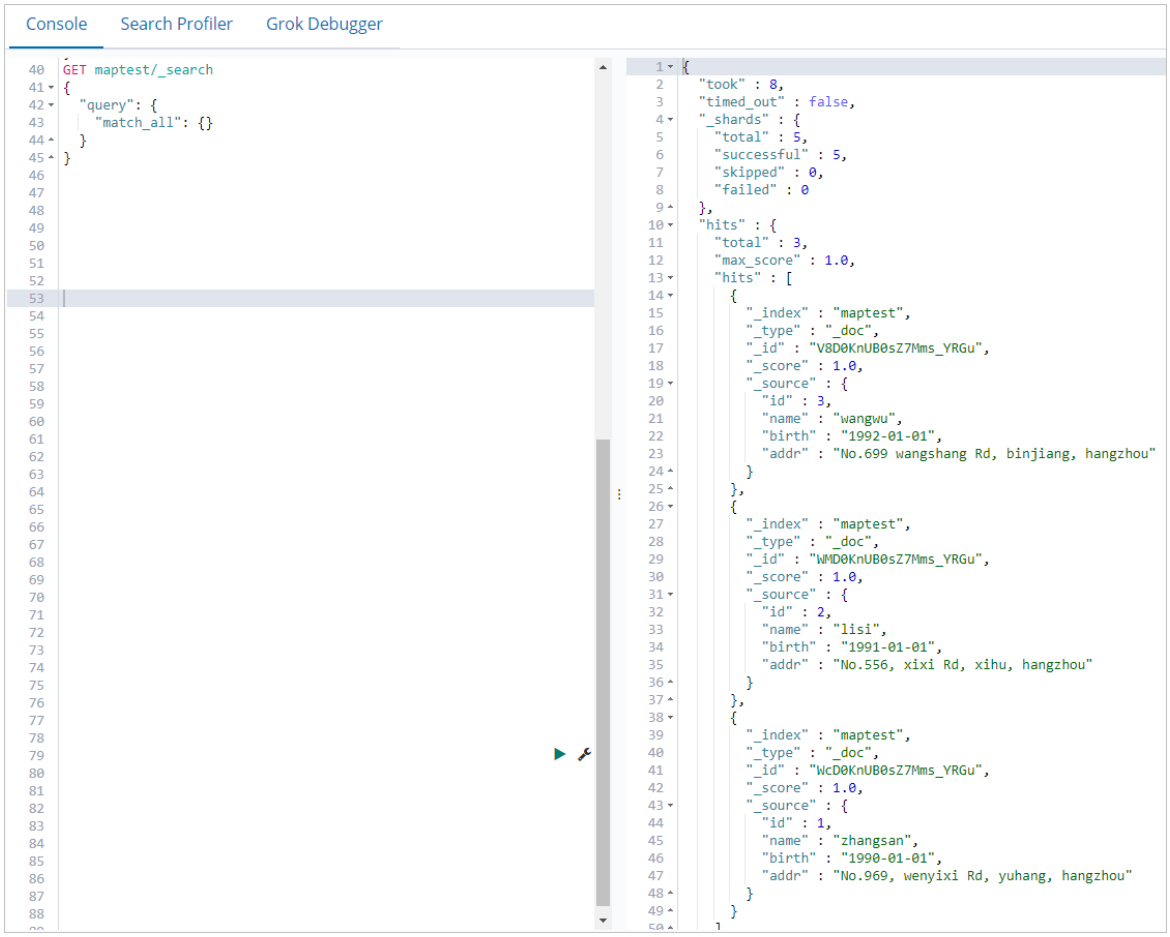
步骤四：验证结果

1. 登录对应阿里云Elasticsearch实例的Kibana控制台。

   具体操作步骤请参见登录Kibana控制台。
2. 在左侧导航栏，单击Dev Tools。
3. 在Console页签下，执行以下命令，查看通过MapReduce任务写入的数据。

```
GET maptest/_search
{
  "query": {
    "match_all": {}
  }
}
```

查询成功后，返回结果如下。



总结

本文以阿里云Elasticsearch和EMR为例，介绍了如何通过ES-Hadoop，借助MapReduce任务向Elasticsearch写入数据。相反，您也可以借助MapReduce任务查询Elasticsearch数据。查询配置和写入类似，详细说明可参见官方Reading data from Elasticsearch说明。

6.3. 通过ES-Hadoop实现Spark读写阿里云Elasticsearch数据

Spark是一种通用的大数据计算框架，拥有Hadoop MapReduce所具有的计算优点，能够通过内存缓存数据为大型数据集提供快速的迭代功能。与MapReduce相比，减少了中间数据读取磁盘的过程，进而提高了处理能力。本文介绍如何通过ES-Hadoop实现Hadoop的Spark服务读写阿里云Elasticsearch数据。

准备工作

1. 创建阿里云Elasticsearch实例，并开启自动创建索引功能。

   具体操作步骤请参见创建阿里云Elasticsearch实例和配置YML参数。本文以6.7.0版本的实例为例。

   > **注意** 在生产环境中，建议关闭自动创建索引功能，提前创建好索引和Mapping。由于本文仅用于测试，因此开启了自动创建索引功能。

2. 创建与Elasticsearch实例在同一专有网络下的E-MapReduce（以下简称EMR）实例。

实例配置如下：

- 产品版本：EMR-3.29.0
- 必选服务：Spark(2.4.5)，其他服务保持默认

具体操作步骤，请参见创建集群。

> 注意 Elasticsearch实例的私网访问白名单默认为0.0.0.0/0，您可在安全配置页面查看，如果未使用默认配置，您还需要在白名单中加入EMR集群的内网IP地址：
>
> - 请参见查看集群列表与详情，获取EMR集群的内网IP地址。
> - 请参见配置实例公网或私网访问白名单，配置Elasticsearch实例的VPC私网访问白名单。

3. 准备Java环境，要求JDK版本为8.0及以上。

编写并运行Spark任务

1. 准备测试数据。
   i. 登录E-MapReduce控制台，获取Master节点的IP地址，并通过SSH登录对应的ECS机器。

   具体操作步骤，请参见登录集群。

   ii. 将测试数据写入文件中。

   本文使用的JSON数据示例如下，将该数据保存在*http_log.txt*文件中。

   ```
   {"id": 1, "name": "zhangsan", "birth": "1990-01-01", "addr": "No.969, wenyixi Rd, yuhang, hangzhou"}
   {"id": 2, "name": "lisi", "birth": "1991-01-01", "addr": "No.556, xixi Rd, xihu, hangzhou"}
   {"id": 3, "name": "wangwu", "birth": "1992-01-01", "addr": "No.699 wangshang Rd, binjiang, hangzhou"}
   ```

   iii. 执行以下命令，将测试数据上传至EMR Master节点的*tmp/hadoop-es*文件中。

   ```
   hadoop fs -put http_log.txt /tmp/hadoop-es
   ```

2. 配置pom依赖。

   创建Java Maven工程，并将如下的pom依赖添加到Java工程的pom.xml文件中。

```
<dependencies>
    <dependency>
        <groupId>org.apache.spark</groupId>
        <artifactId>spark-core_2.12</artifactId>
        <version>2.4.5</version>
    </dependency>
    <dependency>
        <groupId>org.apache.spark</groupId>
        <artifactId>spark-sql_2.11</artifactId>
        <version>2.4.5</version>
    </dependency>
    <dependency>
        <groupId>org.elasticsearch</groupId>
        <artifactId>elasticsearch-spark-20_2.11</artifactId>
        <version>6.7.0</version>
    </dependency>
</dependencies>
```

注意 请确保pom依赖中版本与云服务对应版本保持一致，例如elasticsearch-spark-20_2.11版本与阿里云Elasticsearch版本一致；spark-core_2.12与HDFS版本一致。

3. 编写示例代码。

   i. 写数据

   以下示例代码用来将测试数据写入Elasticsearch的company索引中。

```
import java.util.Map;
import java.util.concurrent.atomic.AtomicInteger;
import org.apache.spark.SparkConf;
import org.apache.spark.SparkContext;
import org.apache.spark.api.java.JavaRDD;
import org.apache.spark.api.java.function.Function;
import org.apache.spark.sql.Row;
import org.apache.spark.sql.SparkSession;
import org.elasticsearch.spark.rdd.api.java.JavaEsSpark;
import org.spark_project.guava.collect.ImmutableMap;
public class SparkWriteEs {
    public static void main(String[] args) {
        SparkConf conf = new SparkConf();
        conf.setAppName("Es-write");
        conf.set("es.nodes", "es-cn-n6w1o1x0w001c****.elasticsearch.aliyuncs.com");
        conf.set("es.net.http.auth.user", "elastic");
        conf.set("es.net.http.auth.pass", "xxxxxx");
        conf.set("es.nodes.wan.only", "true");
        conf.set("es.nodes.discovery","false");
        conf.set("es.input.use.sliced.partitions","false");
        SparkSession ss = new SparkSession(new SparkContext(conf));
        final AtomicInteger employeesNo = new AtomicInteger(0);
        //以下的/tmp/hadoop-es/http_log.txt需要替换为您测试数据的路径。
        JavaRDD<Map<Object, ?>> javaRDD = ss.read().text("/tmp/hadoop-es/http_log.tx
t")
                .javaRDD().map((Function<Row, Map<Object, ?>>) row -> ImmutableMap.o
f("employees"   employeesNo.getAndAdd(1), row.mkString()));
        JavaEsSpark.saveToEs(javaRDD, "company/_doc");
    }
}
```

ii. 读数据

以下示例代码用来读取上一步写入Elasticsearch的数据，并进行打印。

```
import org.apache.spark.SparkConf;
import org.apache.spark.api.java.JavaPairRDD;
import org.apache.spark.api.java.JavaSparkContext;
import org.elasticsearch.spark.rdd.api.java.JavaEsSpark;
import  java.util.Map;
public class ReadES {
    public static void main(String[] args) {
        SparkConf  conf = new SparkConf().setAppName("readEs").setMaster("local[*]")
                .set("es.nodes", "es-cn-n6w1o1x0w001c****.elasticsearch.aliyuncs.com
")
                .set("es.port", "9200")
                .set("es.net.http.auth.user", "elastic")
                .set("es.net.http.auth.pass", "xxxxxx")
                .set("es.nodes.wan.only", "true")
                .set("es.nodes.discovery","false")
                .set("es.input.use.sliced.partitions","false")
                .set("es.resource", "company/_doc")
                .set("es.scroll.size","500");
        JavaSparkContext sc = new JavaSparkContext(conf);
        JavaPairRDD<String, Map<String, Object>> rdd = JavaEsSpark.esRDD(sc);
        for ( Map<String, Object> item : rdd.values().collect()) {
            System.out.println(item);
        }
        sc.stop();
    }
}
```

参数说明

参数	默认值	说明
es.nodes	localhost	指定阿里云Elasticsearch实例的访问地址，建议使用内网地址，可在实例的基本信息页面查看。更多信息，请参见查看实例的基本信息。
es.port	9200	Elasticsearch实例的访问端口号。
es.net.http.auth.user	elastic	Elasticsearch实例的访问用户名。 **说明** 如果程序中指定elastic账号访问Elasticsearch服务，后续在修改elastic账号对应密码后需要一些时间来生效，在密码生效期间会影响服务访问，因此不建议通过elastic来访问。建议在Kibana控制台中创建一个符合预期的Role角色用户进行访问，详情请参见通过Elasticsearch X-Pack角色管理实现用户权限管控。

参数	默认值	说明
es.net.http.auth.pass	/	对应用户的密码，在创建实例时指定。如果忘记可进行重置，具体操作步骤，请参见重置实例访问密码。
es.nodes.wan.only	false	开启Elasticsearch集群在云上使用虚拟IP进行连接，是否进行节点嗅探： ○ true：设置 ○ false：不设置
es.nodes.discovery	true	是否禁用节点发现： ○ true：禁用 ○ false：不禁用 注意 使用阿里云Elasticsearch，必须将此参数设置为false。
es.input.use.sliced.partitions	true	是否使用slice分区： ○ true：使用。设置为true，可能会导致索引在预读阶段的时间明显变长，有时会远远超出查询数据所耗费的时间。建议设置为false，以提高查询效率。 ○ false：不使用。
es.index.auto.create	true	通过Hadoop组件向Elasticsearch集群写入数据，是否自动创建不存在的index： ○ true：自动创建 ○ false：不会自动创建
es.resource	/	指定要读写的index和type。
es.mapping.names	/	表字段与Elasticsearch的索引字段名映射。

更多的ES-Hadoop配置项说明，请参见官方配置说明。

4. 将代码打成Jar包，上传至EMR客户端机器（例如Gateway或EMR集群主节点）。
5. 在EMR客户端机器上，运行如下命令执行Spark程序：
   - 写数据

     ```
     cd /usr/lib/spark-current
     ./bin/spark-submit  --master yarn --executor-cores 1 --class "SparkWriteEs" /root/spark_es.jar
     ```

     注意 */root/spark_es.jar*需要替换为您Jar包上传的路径。
   - 读数据

```
cd /usr/lib/spark-current
./bin/spark-submit  --master yarn --executor-cores 1 --class "ReadES"  /root/spark_es.jar
```

读数据成功后，打印结果如下。

```
20/11/09 14:52:19 INFO [Executor task launch worker for task 0] Executor: Adding file:/tmp/spark-383278cd-c582-4c1c-[redacted]/userFiles-8d764
20/11/09 14:52:19 INFO [Executor task launch worker for task 0] Executor: Finished task 0.0 in stage 0.0 (TID 0). 1274 bytes result sent to driver
20/11/09 14:52:19 INFO [task-result-getter-0] TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 544 ms on localhost (executor driver) (1/1)
20/11/09 14:52:19 INFO [task-result-getter-0] TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
20/11/09 14:52:19 INFO [dag-scheduler-event-loop] DAGScheduler: ResultStage 0 (collect at ReadES.java:26) finished in 0.691 s
20/11/09 14:52:19 INFO [main] DAGScheduler: Job 0 finished: collect at ReadES.java:26, took 0.761197 s
{employees0={"id": 1, "name": "zhangsan", "birth": "1990-01-01", "addr": "No.969, wenyixi Rd, yuhang, hangzhou"}}
{employees1={"id": 2, "name": "lisi", "birth": "1991-01-01", "addr": "No.556, xixi Rd, xihu, hangzhou"}}
{employees2={"id": 3, "name": "wangwu", "birth": "1992-01-01", "addr": "No.699 wangshang Rd, binjiang, hangzhou"}}
20/11/09 14:52:19 INFO [main] SparkUI: Stopped Spark web UI at http://[redacted]:4041
20/11/09 14:52:19 INFO [dispatcher-event-loop-1] MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
20/11/09 14:52:19 INFO [main] MemoryStore: MemoryStore cleared
20/11/09 14:52:19 INFO [main] BlockManager: BlockManager stopped
20/11/09 14:52:19 INFO [main] BlockManagerMaster: BlockManagerMaster stopped
20/11/09 14:52:19 INFO [dispatcher-event-loop-1] OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
20/11/09 14:52:19 INFO [main] SparkContext: Successfully stopped SparkContext
20/11/09 14:52:19 INFO [pool-1-thread-1] ShutdownHookManager: Shutdown hook called
20/11/09 14:52:19 INFO [pool-1-thread-1] ShutdownHookManager: Deleting directory /tmp/spark-383278cd-c582-4c1c-b4b6-[redacted]
20/11/09 14:52:19 INFO [pool-1-thread-1] ShutdownHookManager: Deleting directory /tmp/spark-72515c56-2ba4-4b36-942c-[redacted]
```

验证结果

1. 登录对应阿里云Elasticsearch实例的Kibana控制台。

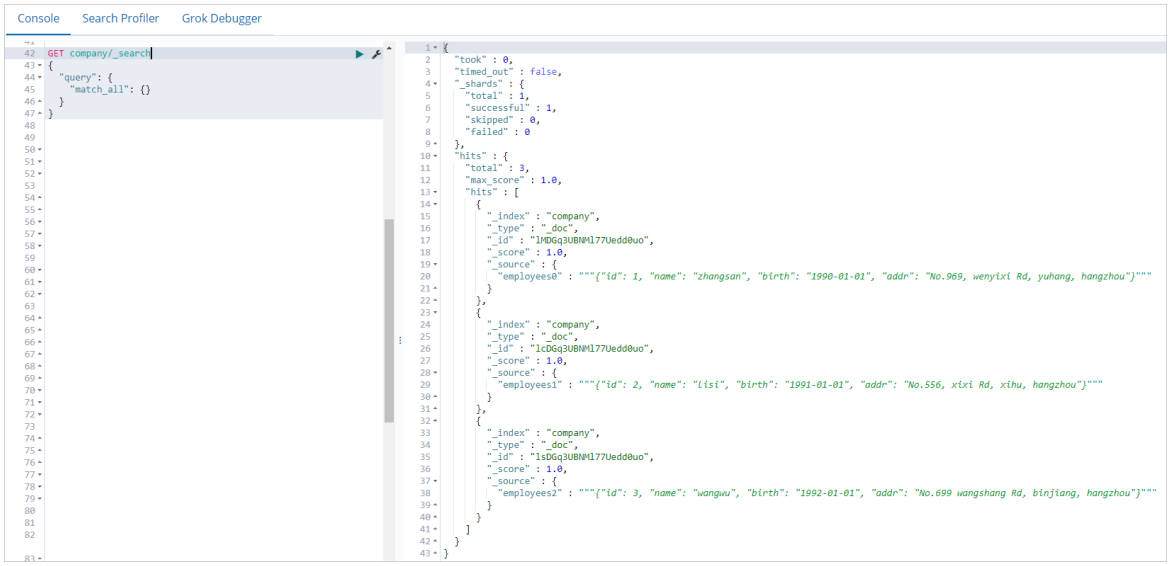
   具体操作步骤请参见登录Kibana控制台。

2. 在左侧导航栏，单击**Dev Tools**。

3. 在**Console**中，执行以下命令，查看通过Spark任务写入的数据。

```
GET company/_search
{
  "query": {
    "match_all": {}
  }
}
```

查询成功后，返回结果如下。



总结

本文以阿里云Elasticsearch和EMR为例，介绍了如何通过ES-Hadoop，实现Spark读写阿里云Elasticsearch数据。与其他EMR组件相比，ES-Hadoop与Spark的集成，不仅包括RDD，还包括Spark Streaming、scale、DataSet与Spark SQL等，您可以根据需求进行配置。详细信息，请参见Apache Spark support。

7.日志采集与分析

7.1. 日志同步分析概述

应用系统在提供服务过程中，会产生各种各样的日志数据。针对这些数据，可以根据业务的需求和环境，选择对应的方案采集数据并传输到Elasticsearch服务中进行查询分析。本文对日志同步分析的方案进行了汇总。

相关文档	方案描述
通过Filebeat采集Apache日志数据	典型的ELK日志采集模式。使用阿里云Filebeat采集Apache日志数据，通过阿里云Logstash过滤采集后的日志数据，传输到Elasticsearch中进行查询分析。
通过logstash-input-sls插件从日志服务获取日志	通过logstash-input-sls插件，使用Logstash从日志服务获取日志，传输到到阿里云Elasticsearch中进行查询分析。
通过自建Filebeat收集MySQL日志	通过自建Filebeat采集MySQL日志并发送到阿里云Elasticsearch中，然后在Kibana控制台中进行可视化查询、分析和展示。
使用阿里云Elasticsearch监控RabbitMQ	通过阿里云Filebeat将RabbitMQ的日志采集到阿里云Elasticsearch中，并对日志进行可视化分析与监控。
使用Filebeat+Kafka+Logstash+Elasticsearch构建日志分析系统	使用Filebeat采集日志数据，将Kafka作为Filebeat的输出端。Kafka实时接收到Filebeat采集的数据后，输出到Logstash中进行过滤处理，最终将满足需求的数据输出到Elasticsearch中进行分布式检索，并通过Kibana进行分析与展示。
查询分析RocketMQ客户端日志	使用Beats、Elasticsearch、Logstash和Kibana，在分布式环境下采集、汇聚、解析阿里云RocketMQ客户端SDK日志，帮助您在消息队列开发场景中快速定位并解决应用开发问题。
通过Elasticsearch和rsbeat实时分析Redis slowlog	使用rsbeat将Redis slowlog采集到Elasticsearch中，然后在Kibana中进行图形化分析。

7.2. 通过自建Filebeat收集MySQL日志

当您需要查看并分析MySQL的日志信息（例如慢日志、error日志等）时，可通过Filebeat采集MySQL日志并发送到阿里云Elasticsearch中，然后在Kibana控制台中完成可视化查询、分析和展示。本文介绍具体的实现方法。

操作流程

1. 准备工作

   创建阿里云Elasticsearch和ECS实例。Elasticsearch实例用来接收Filebeat采集的MySQL日志，并提供一个Kibana控制台进行可视化查询、分析与展示；ECS实例用来安装MySQL和Filebeat。

2. 步骤一：安装并配置MySQL

   安装MySQL，并在MySQL的配置文件中开启error和慢查询日志文件配置。开启后，Filebeat才可采集到对应的日志。

3. 步骤二：安装并配置Filebeat

安装Filebeat，用来将MySQL中的日志采集到阿里云Elasticsearch集群中。需要启用Filebeat的MySQL模块，并在配置文件中指定Kibana和阿里云Elasticsearch的访问地址。

4. 步骤三：使用Kibana Dashboard展示MySQL日志

   进行查询测试，并通过Kibana Dashboard展示对应的error和慢查询日志。

准备工作

- 创建阿里云Elasticsearch实例。

  具体操作步骤请参见创建阿里云Elasticsearch实例，本文以通用商业版6.7.0版本为例。

- 创建阿里云ECS实例。

  具体操作步骤请参见使用向导创建实例，本文以CentOS操作系统为例。

步骤一：安装并配置MySQL

1. 连接ECS实例。

   具体操作步骤请参见通过密码认证登录Linux实例。

2. 下载并安装MySQL源。

   ```
   wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm
   rpm -ivh mysql-community-release-el7-5.noarch.rpm
   ```

3. 安装MySQL。

   ```
   yum install mysql-server
   ```

4. 启动MySQL并查看服务状态。

   ```
   systemctl start mysqld
   systemctl status mysqld
   ```

5. 在my.cnf文件中配置error日志文件和慢查询日志文件。

   > **说明** 默认情况下MySQL会禁用日志文件配置，因此您需要手动开启。您也可通过MySQL命令开启临时慢日志。

   i. 打开my.cnf文件。

   ```
   vim /etc/my.cnf
   ```

ii. 配置日志文件。

```
[mysqld]
log_queries_not_using_indexes = 1
slow_query_log=on
slow_query_log_file=/var/log/mysql/slow-mysql-query.log
long_query_time=0
[mysqld_safe]
log-error=/var/log/mysql/mysqld.log
```

参数	说明
log_queries_not_using_indexes	是否将未使用索引的查询记录到慢查询日志中。1表示记录，0表示不记录。
slow_query_log	是否开启慢查询日志。on表示开启，off表示关闭。
slow_query_log_file	指定慢查询日志的存储路径。
long_query_time	慢查询阈值，单位为秒。当查询时间超过设定的阈值时，MySQL会将日志写入 slow_query_log_file 指定的文件中。 **注意** 本文为方便测试，将该参数值设置为0，实际情况中请根据业务需要合理设置该参数值。

iii. 创建日志文件。

```
mkdir /var/log/mysql
touch /var/log/mysql/mysqld.log
touch /var/log/mysql/slow-mysql-query.log
```

注意 MySQL不会主动创建日志文件，所以您需要手动创建对应文件。

iv. 为所有用户授予日志文件的读写权限。

```
chmod 777 /var/log/mysql/slow-mysql-query.log /var/log/mysql/mysqld.log
```

步骤二：安装并配置Filebeat

1. 登录目标阿里云Elasticsearch实例的Kibana控制台。

   具体操作步骤请参见登录Kibana控制台。
2. 在左侧导航栏，单击Logs。
3. 单击View setup instructions。
4. 在Add Data to Kibana页面，单击MySQL logs。
5. 在Self managed页面，单击RPM。

说明 由于本文使用的是Linux系统，因此选择RPM。实际情况中，请根据您的操作系统类型选择合适的安装方式。

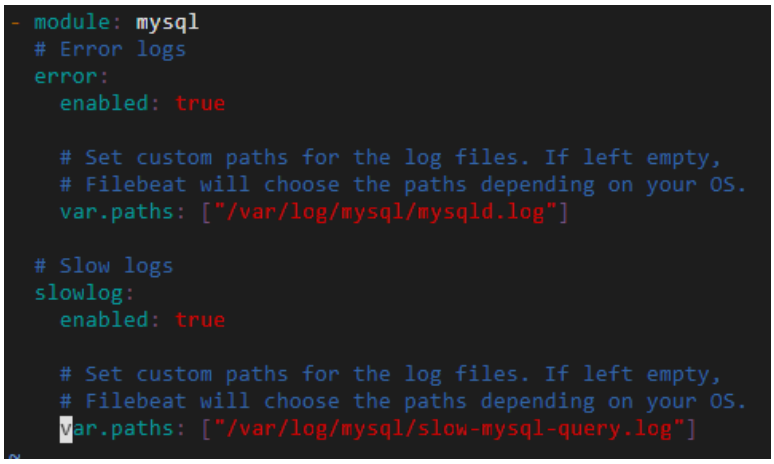
6. 按照页面提示，在ECS中安装Filebeat。
7. 修改MySQL模块配置，分别指定待采集的error和slow日志文件。
   i. 启用MySQL模块。

```
sudo filebeat modules enable mysql
```

   ii. 打开mysql.yml文件。

```
vim /etc/filebeat/modules.d/mysql.yml
```

   iii. 修改MySQL模块配置。




```
- module: mysql
  # Error logs
  error:
    enabled: true
    var.paths: ["/var/log/mysql/mysqld.log"]
    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    #var.paths:
  # Slow logs
  slowlog:
    enabled: true
    var.paths: ["/var/log/mysql/slow-mysql-query.log"]
    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    #var.paths:
```

参数	说明
`enabled`	设置为 `true` 。
`var.paths`	设置为对应日志文件的路径。需要与MySQL配置文件中设置的路径保持一致，详情请参见步骤一：安装并配置MySQL。

8. 配置filebeat.yml文件。
   i. 打开filebeat.yml文件。

```
vim /etc/filebeat/filebeat.yml
```

ii. 修改Filebeat modules配置。

```
#============================== Filebeat modules ===============================

filebeat.config.modules:
  # Glob pattern for configuration loading
  path: /etc/filebeat/modules.d/mysql.yml

  # Set to true to enable config reloading
  reload.enabled: true

  # Period on which files under path should be checked for changes
  reload.period: 1s
```

```
filebeat.config.modules:
  # Glob pattern for configuration loading
  path: /etc/filebeat/modules.d/mysql.yml
  # Set to true to enable config reloading
  reload.enabled: true
  # Period on which files under path should be checked for changes
  reload.period: 1s
```

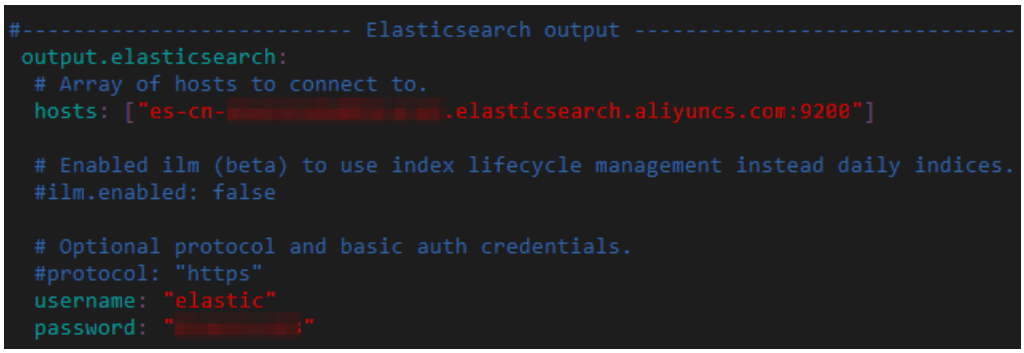
iii. 修改Kibana配置。

```
#============================== Kibana =====================================

# Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.
# This requires a Kibana endpoint configuration.
setup.kibana:

  # Kibana Host
  # Scheme and port can be left out and will be set to the default (http and 5601)
  # In case you specify and additional path, the scheme is required: http://localhost:5601/path
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
  host: "https://es-cn-[redacted].kibana.elasticsearch.aliyuncs.com:5601"
```

```
setup.kibana:
host: "https://es-cn-0pp1jxvcl000*****.kibana.elasticsearch.aliyuncs.com:5601"
```

`host`：Kibana的访问地址。可在Kibana配置页面获取，详情请参见查看Kibana公网地址，格式为`<Kibana公网地址>:5601`。

iv. 修改Elasticsearch output配置。




```
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["es-cn-0pp1jxvcl000*****.elasticsearch.aliyuncs.com:9200"]
  # Optional protocol and basic auth credentials.
  #protocol: "https"
  username: "elastic"
  password: "<your_password>"
```

参数	说明
hosts	阿里云Elasticsearch的访问地址，格式为 <实例的内网或公网地址>:9200 。Elasticsearch实例的内网或公网地址可在实例的基本信息页面获取，详情请参见查看实例的基本信息。 **说明** 如果ECS和Elasticsearch实例在同一专有网络VPC（Virtual Private Cloud）下，请使用内网地址；如果不在同一VPC下，请使用公网地址。使用公网地址需要配置公网地址访问白名单，详情请参见配置实例公网或私网访问白名单。
username	阿里云Elasticsearch实例的访问用户名，默认为elastic。
password	对应用户的密码，一般在创建实例时设定。如果忘记，可重置，重置密码的注意事项和操作步骤请参见重置实例访问密码。

9. 执行以下命令，将Dashboard等信息上传到Elasticsearch和Kibana中，并启用Filebeat服务。

```
sudo filebeat setup
sudo service filebeat start
```

步骤三：使用Kibana Dashboard展示MySQL日志

1. 在ECS中重启MySQL数据库，并查询任意数据进行测试。

   重启命令如下。

```
systemctl restart mysqld
```

2. 查看日志内容。

   本文检测到的日志内容如下。

慢日志内容

```
[root@zl-test003 ~]# tail -50 /var/log/mysql/slow-mysql-query.log
# Query_time: 0.000385  Lock_time: 0.000000 Rows_sent: 0  Rows_examined: 0
SET timestamp=1590720469;
;
# User@Host: root[root] @ localhost []  Id:     2
# Query_time: 0.000138  Lock_time: 0.000000 Rows_sent: 0  Rows_examined: 0
SET timestamp=1590720469;
;
# User@Host: root[root] @ localhost []  Id:     2
# Query_time: 0.000214  Lock_time: 0.000048 Rows_sent: 5  Rows_examined: 5
SET timestamp=1590720469;
SELECT * from student;
# Time: 200529 10:47:52
# User@Host: root[root] @ localhost []  Id:     2
# Query_time: 0.000231  Lock_time: 0.000117 Rows_sent: 2  Rows_examined: 10
SET timestamp=1590720472;
SELECT * from runoob_tbl WHERE runoob_author='菜鸟教程';
# User@Host: root[root] @ localhost []  Id:     2
# Query_time: 0.000171  Lock_time: 0.000075 Rows_sent: 2  Rows_examined: 10
SET timestamp=1590720472;
SELECT * from runoob_tbl WHERE runoob_author='菜鸟教程';
# Time: 200529 10:47:53
# User@Host: root[root] @ localhost []  Id:     2
# Query_time: 0.000149  Lock_time: 0.000075 Rows_sent: 2  Rows_examined: 10
SET timestamp=1590720473;
SELECT * from runoob_tbl WHERE runoob_author='菜鸟教程';
# Time: 200529 10:47:56
# User@Host: root[root] @ localhost []  Id:     2
# Query_time: 0.000039  Lock_time: 0.000000 Rows_sent: 0  Rows_examined: 0
SET timestamp=1590720476;
sleep(4);
```

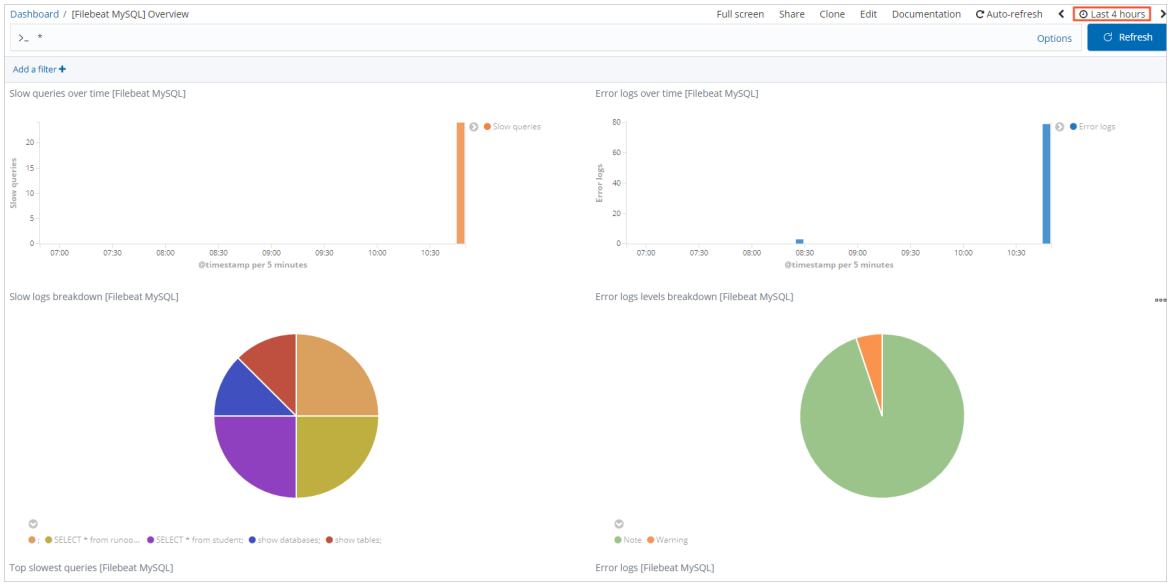
error日志内容

```
[root@zl-test003 ~]# tail -50 /var/log/mysql/mysqld.log
2020-05-29 10:47:43 4240 [Note] Shutting down plugin 'INNODB_LOCKS'
2020-05-29 10:47:43 4240 [Note] Shutting down plugin 'INNODB_TRX'
2020-05-29 10:47:43 4240 [Note] Shutting down plugin 'InnoDB'
2020-05-29 10:47:43 4240 [Note] InnoDB: FTS optimize thread exiting.
2020-05-29 10:47:43 4240 [Note] InnoDB: Starting shutdown...
2020-05-29 10:47:45 4240 [Note] InnoDB: Shutdown completed; log sequence number 1666138
2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'BLACKHOLE'
2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'ARCHIVE'
2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'MRG_MYISAM'
2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'MyISAM'
2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'MEMORY'
2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'CSV'
2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'sha256_password'
2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'mysql_old_password'
2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'mysql_native_password'
2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'binlog'
2020-05-29 10:47:45 4240 [Note] /usr/sbin/mysqld: Shutdown complete

200529 10:47:45 mysqld_safe mysqld from pid file /var/run/mysqld/mysqld.pid ended
200529 10:47:45 mysqld_safe Logging to '/var/log/mysql/mysqld.log'.
200529 10:47:45 mysqld_safe Starting mysqld daemon with databases from /var/lib/mysql
2020-05-29 10:47:45 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_def
_timestamp server option (see documentation for more details).
2020-05-29 10:47:45 0 [Note] /usr/sbin/mysqld (mysqld 5.6.48-log) starting as process 21710 ...
2020-05-29 10:47:45 21710 [Warning] Buffered warning: Changed limits: max_open_files: 1024 (requested 5000)
```

3. 登录目标Elasticsearch实例的Kibana控制台。

   具体操作步骤请参见登录Kibana控制台。

4. 在左侧导航栏，单击Dashboard。
5. 在Dashboards列表中，单击[Filebeat MySQL] Overview。
6. 在页面右上角选择查询时间，查看对应时间段内的日志。



7.3. 使用阿里云ES监控RabbitMQ

阿里云ES监控RabbitMQ

RabbitMQ是一个开源的消息代理服务器，能够为您的应用提供一个通用的消息发送和接收平台，并且保证消息在传输过程中的安全性。本文介绍如何通过Filebeat，将RabbitMQ的日志采集到阿里云Elasticsearch（简称ES）中，并对日志进行可视化分析与监控。

操作流程

1. 准备工作

   完成环境准备，包括创建实例、安装RabbitMQ等。

2. 步骤一：配置RabbitMQ

   配置RabbitMQ的日志级别和文件名。

3. 步骤二：安装RabbitMQ示例应用

   基于Spring Boot，使用RabbitMQ的JMS客户端生产日志数据到RabbitMQ中。

4. 步骤三：创建并配置阿里云Filebeat

   配置Filebeat采集器，将RabbitMQ的日志数据发送到Logstash的8100端口。

5. 步骤四：配置阿里云Logstash管道

   配置Logstash管道，使用基本的Grok模式从原始消息中分离出时间戳、日志级别和消息，然后发送到阿里云ES中。

6. 步骤五：通过Kibana查看日志数据

   通过Kibana查看经Logstash处理后的数据。

7. 步骤六：通过Kibana过滤日志数据

   在Kibana控制台的Discover页面，通过Filter过滤出RabbitMQ相关的日志。

8. 步骤七：配置Metricbeat采集RabbitMQ指标

   通过Metricbeat采集RabbitMQ日志，并通过Kibana实现可视化指标监控。

准备工作

1. 创建阿里云ES和Logstash实例，两者版本相同，并且在同一专有网络VPC（Virtual Private Cloud）下。

具体操作步骤请参见创建阿里云Elasticsearch实例、创建阿里云Logstash实例，本文以6.7版本为例。

2. 开启阿里云ES实例的**自动创建索引**功能。

   具体操作步骤请参见快速访问与配置。

3. 创建阿里云ECS实例，要求与阿里云ES实例和Logstash实例处于同一VPC下。

   具体操作步骤请参见使用向导创建实例。

   > **注意** 该ECS实例用来安装Beats和RabbitMQ，由于Beats目前仅支持Aliyun Linux、RedHat和CentOS这三种操作系统，因此在创建时请选择其中一种操作系统。

4. 在目标ECS实例上安装云助手和Docker服务。

   具体操作步骤请参见安装云助手客户端和部署并使用Docker（Alibaba Cloud Linux 2）。

5. 在目标ECS实例上安装RabbitMQ。

   具体操作步骤请参见Downloading and Installing RabbitMQ。

步骤一：配置RabbitMQ

配置RabbitMQ的日志记录级别和文件名，步骤如下：

1. 连接安装了RabbitMQ的ECS服务器。

   具体操作步骤请参见连接实例。

2. 执行以下命令，打开RabbitMQ的配置文件。

   ```
   vim /etc/rabbitmq/rabbitmq.config
   ```

3. 修改RabbitMQ的配置文件。

   ```
   {
     lager, [
     %%
     %% Log directory, taken from the RABBITMQ_LOG_BASE env variable by default.
     %% {log_root, "/var/log/rabbitmq"},
     %%
     %% All log messages go to the default "sink" configured with
     %% the `handlers` parameter. By default, it has a single
     %% lager_file_backend handler writing messages to "$nodename.log"
     %% (ie. the value of $RABBIT_LOGS).
      {handlers, [
        {lager_file_backend, [{file, "rabbit.log"},
                              {level, info},
                              {date, ""},
                              {size, 0}]}
     ]},
   {extra_sinks, [
        {rabbit_channel_lager_event, [{handlers, [
                                         {lager_forwarder_backend,
                                          [lager_event, info]}]}]},
        {rabbit_conection_lager_event, [{handlers, [
                                           {lager_forwarder_backend,
                                            [lager_event, error]}]}]}
     ]}
   ```

修改后，RabbitMQ的日志文件名变为*rabbit.log*。

> **说明** 日志级别（level）默认为info，您也可以将其设置为error，即记录出错日志。本文以info日志为例。

修改配置文件后，需要重新启动RabbitMQ才能生效。

4. 重启RabbitMQ服务。

   进入RabbitMQ安装目录的bin文件夹下，重启服务。

   ```
   cd /usr/lib/rabbitmq/bin
   rabbitmq-server start
   ```

   重启成功后，可在*/var/log/rabbitmq*下看到*rabbit.log*文件。

步骤二：安装RabbitMQ示例应用

本节基于Spring Boot，使用RabbitMQ的JMS客户端进行演示。具体操作步骤如下：

> **注意** 由于示例需要编译应用，因此您必须先安装Java 8。

1. 在ECS中，执行以下命令克隆示例。

   ```
   git clone https://github.com/rabbitmq/rabbitmq-jms-client-spring-boot-trader-demo
   ```

2. 进入该应用所在根目录。

   ```
   cd rabbitmq-jms-client-spring-boot-trader-demo
   ```

3. 执行以下命令打包并运行应用。

   ```
   mvn clean package
   java -jar target/rabbit-jms-boot-demo-1.2.0-SNAPSHOT.jar
   ```

   运行成功后，返回结果如下。

```
.   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v1.5.8.RELEASE)
2020-05-11 10:16:46.089  INFO 28119 --- [main] com.rabbitmq.jms.sample.StockQuoter
: Starting StockQuoterxxxxxx
2020-05-11 10:16:46.092  INFO 28119 --- [main] com.rabbitmq.jms.sample.StockQuoter
: No active profile set,xxxxxx
2020-05-11 10:16:46.216  INFO 28119 --- [main] s.c.a.AnnotationConfigApplicationContext
: Refreshing org.springframework.xxxxxx
2020-05-11 10:16:47.224  INFO 28119 --- [main] com.rabbitmq.jms.sample.StockConsumer
: connectionFactory => RMQConnectionFactoryxxxxxx
2020-05-11 10:16:48.054  INFO 28119 --- [main] o.s.j.e.a.AnnotationMBeanExporter
: Registering beans for JMX exposurexxxxxx
2020-05-11 10:16:48.062  INFO 28119 --- [main] o.s.c.support.DefaultLifecycleProcessor
: Starting beans in phase 0xxxxxx
 ......
```

4. 进入log目录下查看RabbitMQ日志。

```
cd /var/log/rabbitmq
ls
```

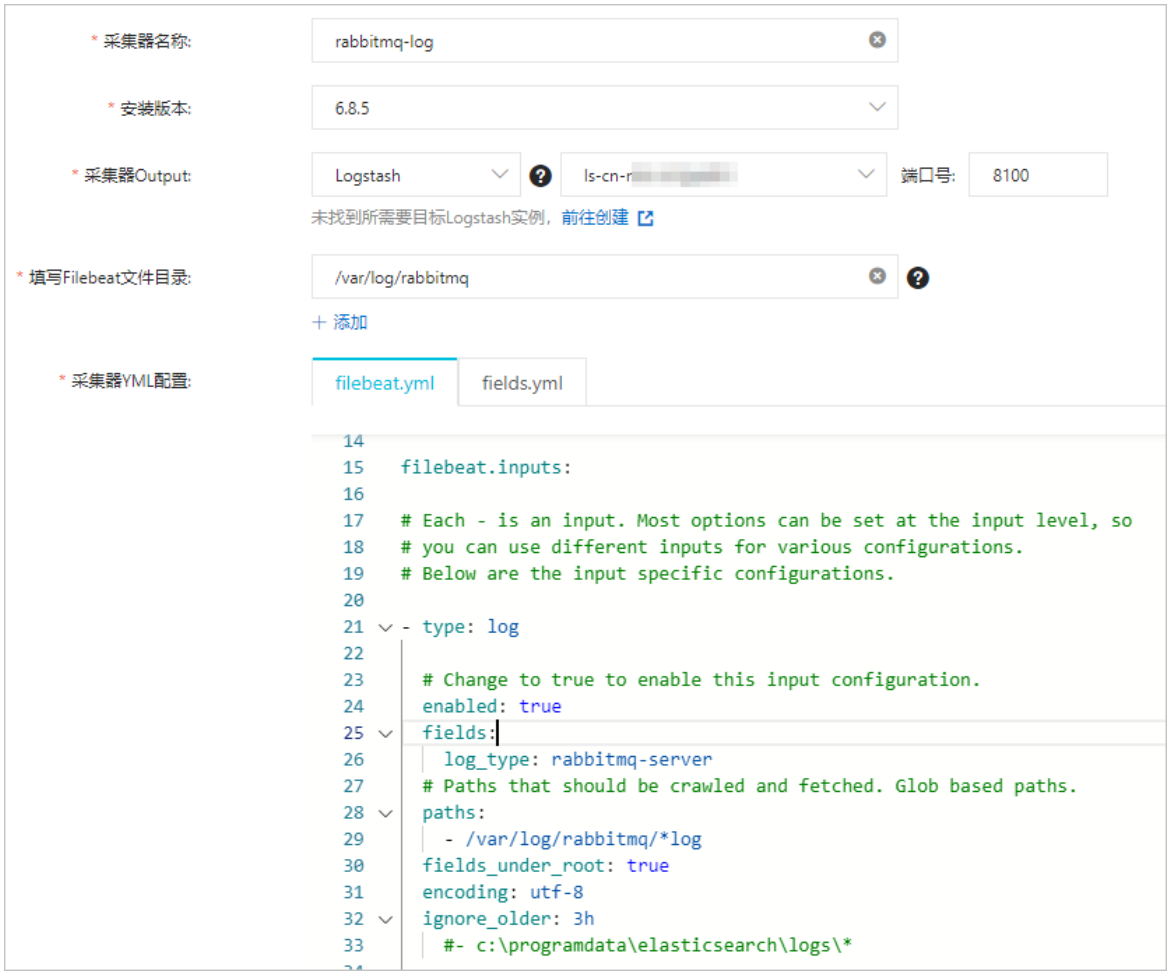
返回结果中的*rabbit.log*即为步骤一：配置RabbitMQ中配置的文件名。

步骤三：创建并配置阿里云Filebeat

配置Filebeat采集器，将RabbitMQ的日志信息发送到阿里云Logstash的8100端口。

1. 登录阿里云Elasticsearch控制台。
2. 在**创建采集器**区域，单击**Filebeat**。
3. 配置采集器。

   具体操作步骤及详细参数说明，请参见采集ECS服务日志和采集器YML配置，本文使用的配置如下。



说明

- 采集器Output需要指定目标阿里云Logstash的实例ID，在YML配置中不需要重新指定Output。
- Filebeat文件目录需要填写RabbitMQ日志所在的目录，同时需要在YML配置中开启log数据采集，并配置log路径。

本文使用的**filebeat.yml**中的 `filebeat.inputs` 配置如下。

```
filebeat.inputs:
# Each - is an input. Most options can be set at the input level, so
# you can use different inputs for various configurations.
# Below are the input specific configurations.
- type: log
  # Change to true to enable this input configuration.
  enabled: true
  fields:
    log_type: rabbitmq-server
  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /var/log/rabbitmq/*log
  fields_under_root: true
  encoding: utf-8
  ignore_older: 3h
```

4. 安装并启动采集器。

   具体操作步骤请参见采集ECS服务日志。安装时所选的ECS实例要与阿里云Logstash在同一VPC下。

5. 查看采集器安装情况。

   采集器状态变为**已生效1/1**，且查看**采集器安装情况**后显示为**心跳正常**，说明采集器安装成功。

步骤四：配置阿里云Logstash管道

配置阿里云Logstash管道，使用基本的Grok模式从原始消息中分离出时间戳、日志级别和消息，然后发送到阿里云ES实例的指定索引中。

1. 登录阿里云Elasticsearch控制台。
2. 在左侧导航栏，单击**Elasticsearch实例**。
3.
4. 在左侧导航栏，单击**管道管理**。

5. 单击**创建管道**。
6. 在**创建管道任务**页面，输入**管道ID**并配置管道。

   本文使用的管道配置如下。

```
input {
  beats {
    port => 8100
  }
}
filter {
  grok {
    match => { "message" => ["%{TIMESTAMP_ISO8601:timestamp} \[%{LOGLEVEL:log_level}\
] \<%{DATA:field_misc}\> %{GREEDYDATA:message}"] }
  }
}
output {
   elasticsearch {
      hosts => "es-cn-4591jumei000u****.elasticsearch.aliyuncs.com:9200"
      user => "elastic"
      password => "your_password"
      index => "rabbitmqlog-%{+YYYY.MM.dd}"
  }
}
```

   管道配置详情请参见Logstash配置文件说明。

7. 单击**下一步**，配置管道参数。

参数	说明
管道工作线程	并行执行管道的Filter和Output的工作线程数量。当事件出现积压或CPU未饱和时，请考虑增大线程数，更好地使用CPU处理能力。默认值：实例的CPU核数。
管道批大小	单个工作线程在尝试执行Filter和Output前，可以从Input收集的最大事件数目。较大的管道批大小可能会带来较大的内存开销。您可以设置LS_HEAP_SIZE变量，来增大JVM堆大小，从而有效使用该值。默认值：125。
管道批延迟	创建管道事件批时，将过小的批分派给管道工作线程之前，要等候每个事件的时长，单位为毫秒。默认值：50ms。

参数	说明
队列类型	用于事件缓冲的内部排队模型。可选值： ○ **MEMORY**：默认值。基于内存的传统队列。 ○ **PERSISTED**：基于磁盘的ACKed队列（持久队列）。
队列最大字节数	请确保该值小于您的磁盘总容量。默认值：1024 MB。
队列检查点写入数	启用持久性队列时，在强制执行检查点之前已写入事件的最大数目。设置为0，表示无限制。默认值：1024。

警告 配置完成后，需要保存并部署才能生效。保存并部署操作会触发实例重启，请在不影响业务的前提下，继续执行以下步骤。

8. 单击**保存**或者**保存并部署**。
   - **保存**：将管道信息保存在Logstash里并触发实例变更，配置不会生效。保存后，系统会返回**管道管理**页面。可在**管道列表**区域，单击**操作**列下的**立即部署**，触发实例重启，使配置生效。
   - **保存并部署**：保存并且部署后，会触发实例重启，使配置生效。

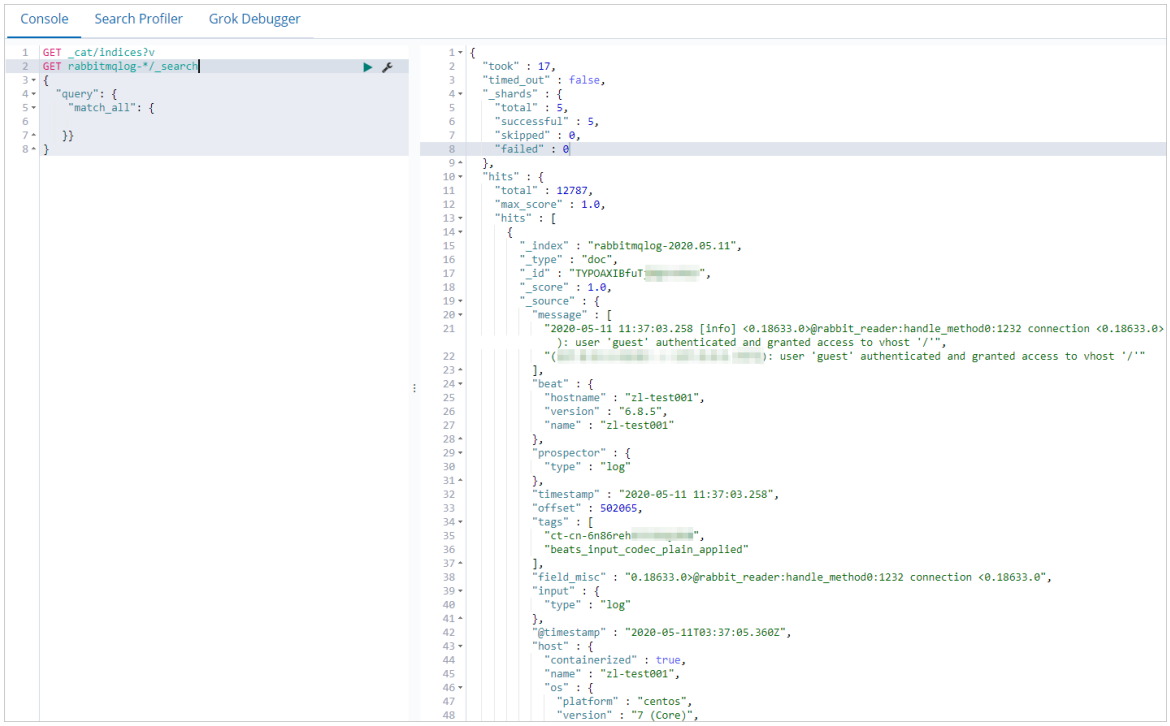
步骤五：通过Kibana查看日志数据

1. 登录目标阿里云ES的Kibana控制台。

   具体步骤请参见登录Kibana控制台。
2. 在左侧导航栏，单击**Dev Tools**。
3. 在**Console**中执行以下命令，查看Logstash处理后的RabbitMQ日志数据。

```
GET rabbitmqlog-*/_search
{
  "query": {
    "match_all": {
   }}
}
```

执行成功后，返回如下结果。



步骤六：通过Kibana过滤日志数据

1. 登录目标阿里云ES实例的Kibana控制台。

   具体步骤请参见登录Kibana控制台。

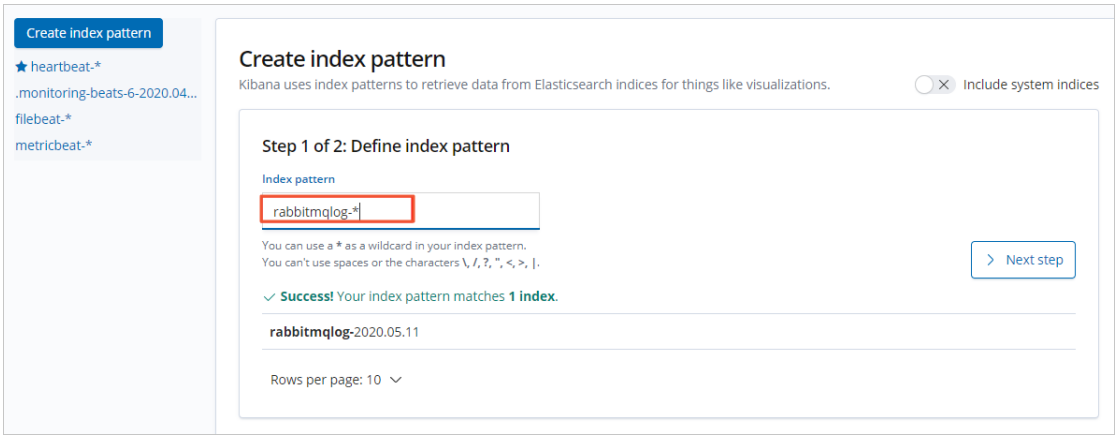
2. 创建一个索引模式。

   i. 在左侧导航栏，单击**Management**。

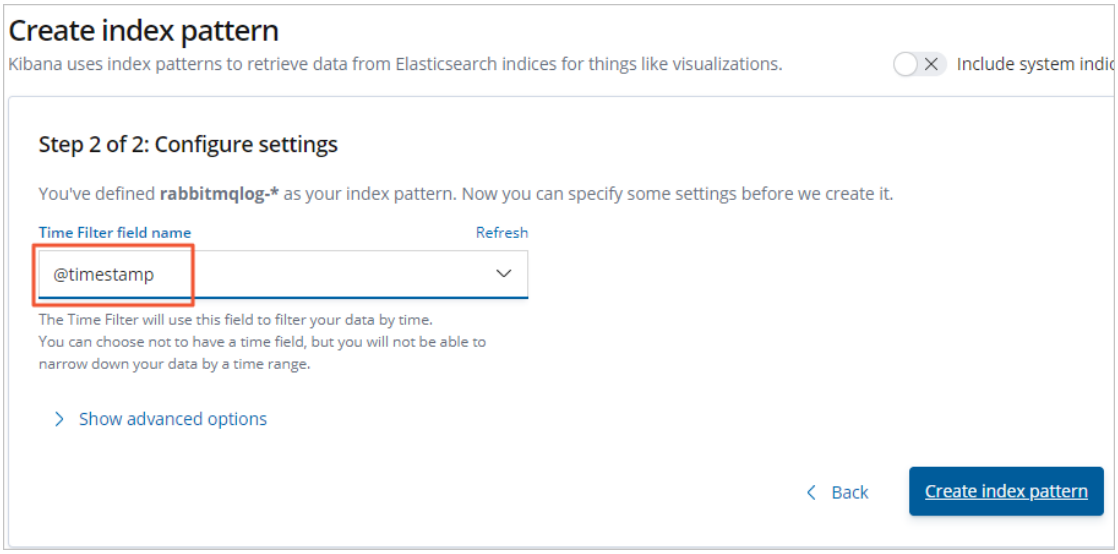
   ii. 在Kibana区域，单击**Index Patterns**。

   iii. 单击**Create index pattern**。

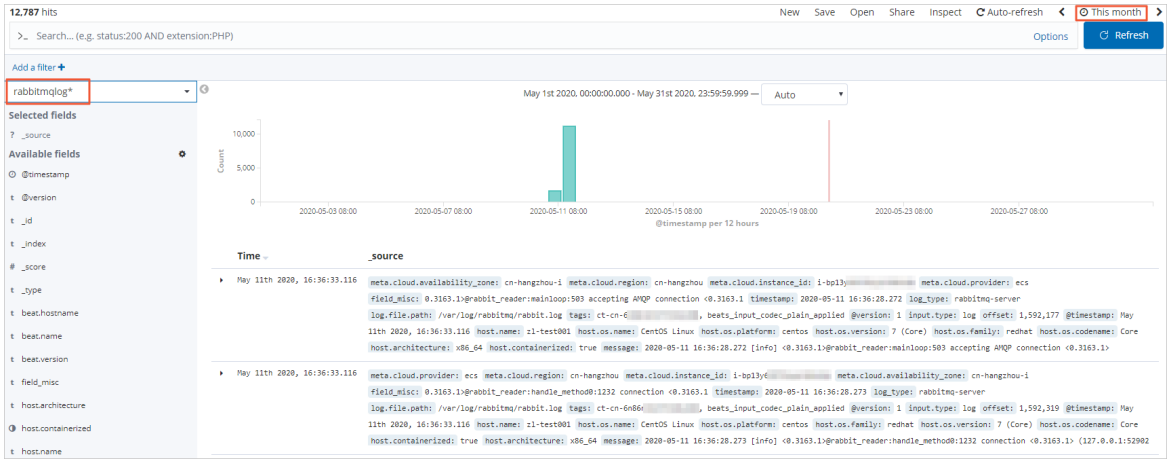
   iv. 输入**Index pattern**（本文使用rabbitmqlog-*），单击**Next step**。



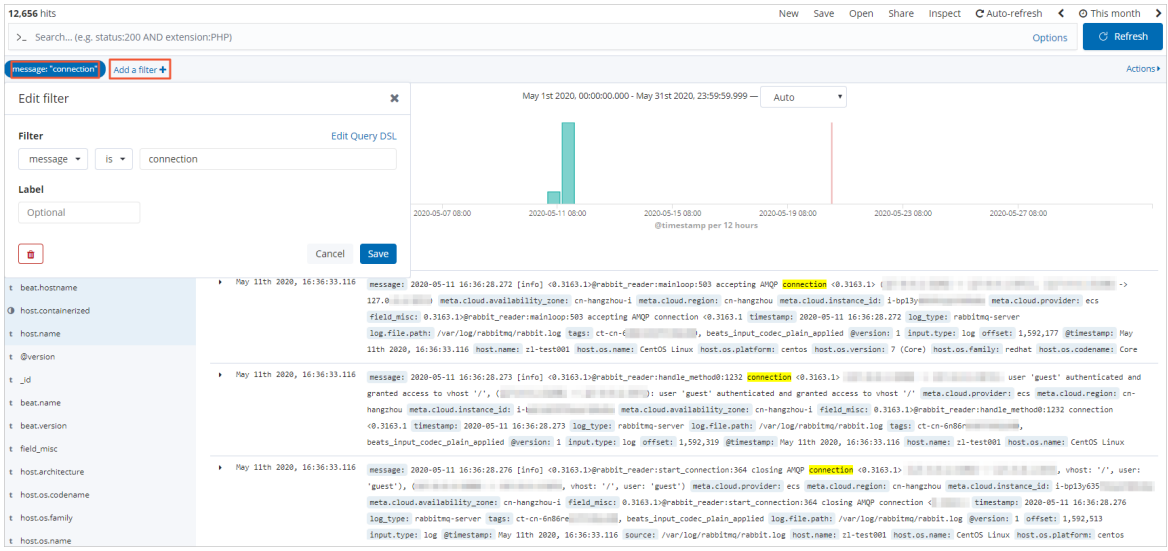
v. 选择Time Filter field name（本文选择@timestamp），单击Create index pattern。



3. 在左侧导航栏，单击Discover。
4. 从页面左侧的下拉列表中，选择您已创建的索引模式（rabbitmqlog-*）。
5. 在页面右上角，选择一段时间，查看对应时间段内的RabbitMQ日志数据。



6. 单击Add a filter，设置过滤条件，查看符合条件的日志数据。



步骤七：配置Metricbeat采集RabbitMQ指标

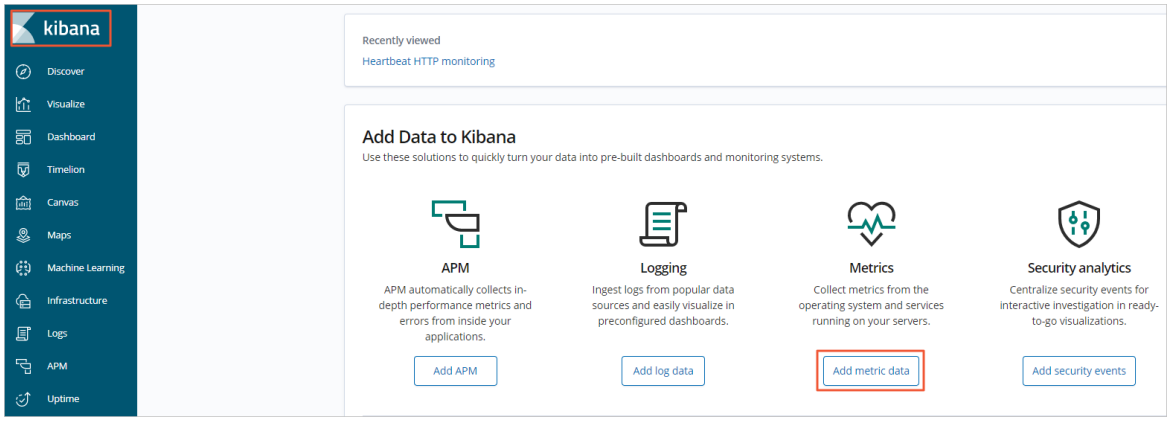
您也可以通过Metricbeat采集RabbitMQ日志数据，并通过Kibana可视化监控RabbitMQ的各项指标。

1. 登录目标阿里云ES的Kibana控制台。

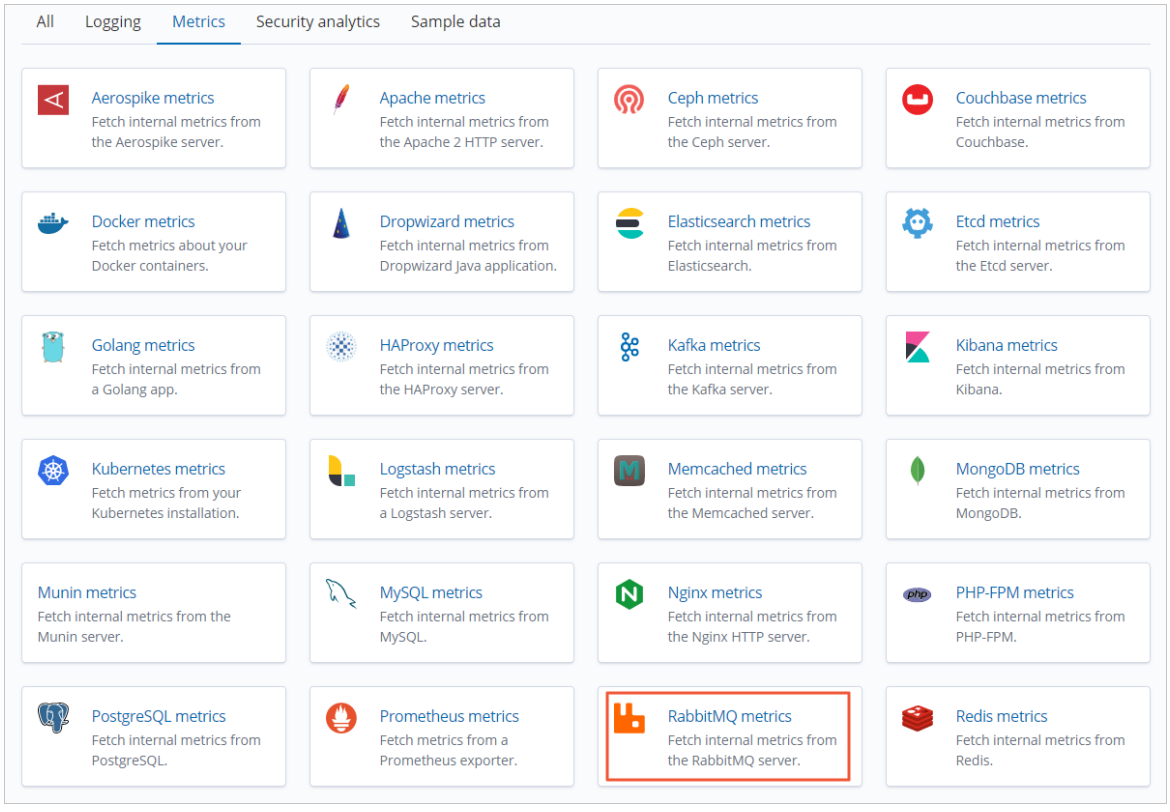
   具体步骤请参见登录Kibana控制台。

2. 在页面左上角，单击Kibana。

3. 在Add Data to Kibana区域，单击Add metric data。



4. 在Metrics页签，单击RabbitMQ metrics。



5. 单击RPM，按照页面提示在ECS中安装并配置Metricbeat。

   配置Metricbeat时，需要修改*/etc/metricbeat/metricbeat.yml*文件，设置Kibana和ES集群的连接信息，本文使用的配置如下。

```
setup.kibana:
  # Kibana Host
  # Scheme and port can be left out and will be set to the default (http and 5601)
  # In case you specify and additional path, the scheme is required: http://localhost:56
01/path
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
   host: "https://es-cn-4591jumei000u****.kibana.elasticsearch.aliyuncs.com:5601"
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["es-cn-4591jumei000u****.elasticsearch.aliyuncs.com:9200"]
  # Enabled ilm (beta) to use index lifecycle management instead daily indices.
  #ilm.enabled: false
  # Optional protocol and basic auth credentials.
  #protocol: "https"
  username: "elastic"
  password: "your_password"
```

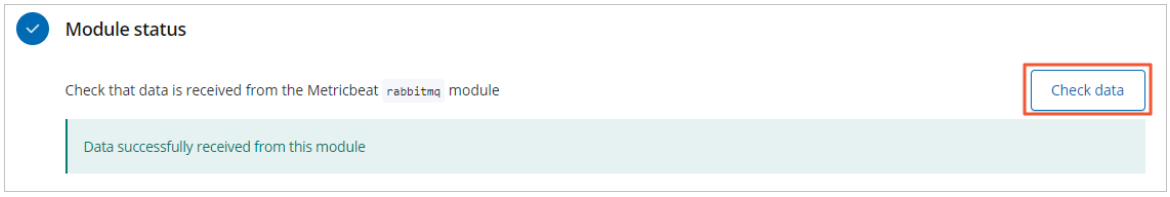
6. 启动RabbitMQ模块及Metricbeat服务。
   - 启动RabbitMQ模块

     ```
     sudo metricbeat modules enable rabbitmq
     ```

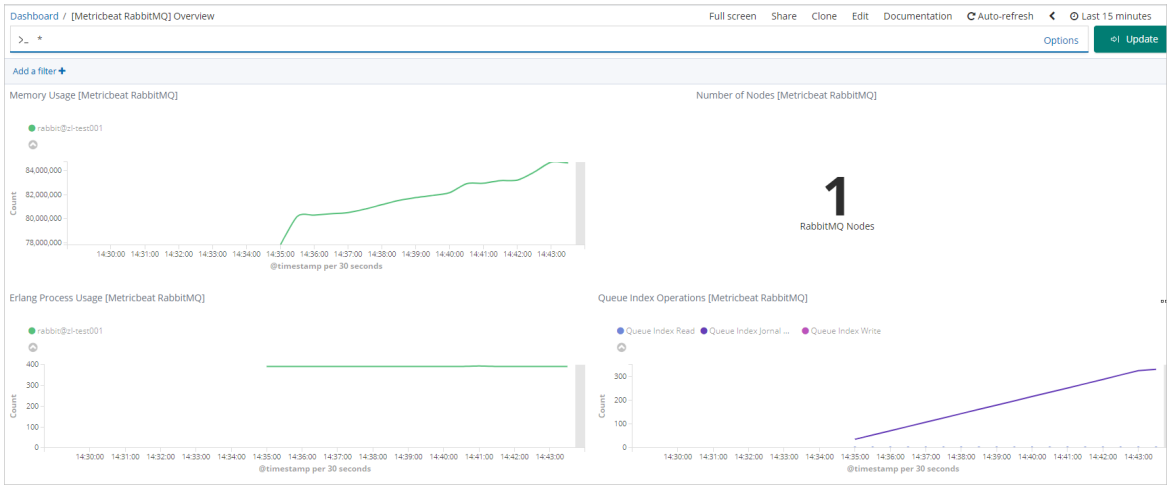
   - 启动Metricbeat服务并设置仪表盘

     ```
     sudo metricbeat setup
     sudo service metricbeat start
     ```

7. 在**RPM**页签中，单击**Check data**。



8. 单击**RabbitMQ metrics dashboard**，查看Dashboard监控大盘。



7.4. 使用Filebeat+Kafka+Logstash+Elasticsearch构建日志分析系统

随着时间的积累，日志数据会越来越多，当您需要查看并分析庞杂的日志数据时，可通过Filebeat+Kafka+Logstash+Elasticsearch采集日志数据到阿里云Elasticsearch中，并通过Kibana进行可视化展示与分析。本文介绍具体的实现方法。

背景信息

Kafka是一种分布式、高吞吐、可扩展的消息队列服务，广泛用于日志收集、监控数据聚合、流式数据处理、在线和离线分析等大数据领域，已成为大数据生态中不可或缺的部分。本文使用阿里云消息队列Kafka版，详情请参见什么是消息队列Kafka版。

在实际应用场景中，为了满足大数据实时检索的需求，您可以使用Filebeat采集日志数据，并输出到Kafka中。Kafka实时接收Filebeat采集的数据，并输出到Logstash中。输出到Logstash中的数据在格式或内容上可能不能满足您的需求，此时可以通过Logstash的filter插件过滤数据。最后将满足需求的数据输出到Elasticsearch中进行分布式检索，并通过Kibana进行数据分析与展示。简单流程如下。

操作流程

1. 准备工作

   完成环境准备，包括创建阿里云Elasticsearch、Logstash、ECS和消息队列Kafka版实例、创建Topic和Consumer Group等。

   > 注意 建议您使用同一专有网络VPC（Virtual Private Cloud）下的阿里云Elasticsearch、Logstash、ECS和消息队列Kafka版实例。

2. 步骤一：安装并配置Filebeat

   安装并配置Filebeat，设置input为系统日志，output为Kafka，将日志数据采集到Kafka的指定Topic中。

3. 步骤二：配置Logstash管道

   配置Logstash管道的input为Kafka，output为阿里云Elasticsearch，使用Logstash消费Topic中的数据并传输到阿里云Elasticsearch中。

4. 步骤三：查看日志消费状态

   在消息队列Kafka中查看日志数据的消费的状态，验证日志数据是否采集成功。

5. 步骤四：通过Kibana过滤日志数据

   在Kibana控制台的Discover页面，通过Filter过滤出Kafka相关的日志。

准备工作

1. 创建阿里云Elasticsearch实例，并开启实例的自动创建索引功能。

   具体操作步骤请参见创建阿里云Elasticsearch实例和配置YML参数。本文以6.7版本为例。

2. 创建阿里云Logstash实例。

   具体操作步骤请参见创建阿里云Logstash实例。创建的实例需要满足：

   - 版本：与阿里云Elasticsearch实例的版本要满足兼容性要求，详细信息请参见产品兼容性。本文使用与Elasticsearch相同的版本，即6.7版本。
   - 网络：与阿里云Elasticsearch实例在同一VPC下，否则需要配置NAT网关实现与公网的连通，详细信息请参见配置NAT公网数据传输。

3. 购买并部署阿里云消息队列Kafka版实例、创建Topic和Consumer Group。

   本文使用VPC实例，并要求该实例与阿里云Elasticsearch实例在同一VPC下，具体操作步骤请参见VPC接入。

   创建Topic和Consumer Group的具体步骤请参见步骤三：创建资源。

4. 创建阿里云ECS实例。

   具体操作步骤请参见使用向导创建实例。本文的ECS实例与阿里云Elasticsearch实例在同一VPC下，否则需要配置公网访问白名单实现网络互通，详细信息请参见配置实例公网或私网访问白名单。

   > 注意 该ECS实例用来安装Filebeat，由于Filebeat目前仅支持Aliyun Linux、RedHat和CentOS这三种操作系统，因此在创建时请选择其中一种操作系统。

步骤一：安装并配置Filebeat

1. 连接ECS服务器。

   具体操作步骤请参见连接实例。

2. 安装Filebeat。

   本文以6.8.5版本为例，安装命令如下，详细信息请参见Install Filebeat。

   ```
   curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-6.8.5-linux-x86_64.tar.gz
   tar xzvf filebeat-6.8.5-linux-x86_64.tar.gz
   ```

3. 执行以下命令，进入Filebeat安装目录，创建并配置filebeat.kafka.yml文件。

   ```
   cd filebeat-6.8.5-linux-x86_64
   vi filebeat.kafka.yml
   ```

   filebeat.kafka.yml配置如下。

   ```
   filebeat.prospectors:
     - type: log
       enabled: true
       paths:
           - /var/log/*.log
   output.kafka:
       hosts: ["alikafka-post-cn-zvp2n4v7****-1-vpc.alikafka.aliyuncs.com:9092"]
       topic: estest
       version: 0.10.2
   ```

   > 注意 当Filebeat为7.0及以上版本时，filebeat.prospectors需要替换为filebeat.inputs。

参数	说明
type	输入类型。设置为log，表示输入源为日志。
enabled	设置配置是否生效： ◦ true：生效 ◦ false：不生效
paths	需要监控的日志文件的路径。多个日志可在当前路径下另起一行写入日志文件路径。
hosts	消息队列Kafka实例的单个接入点，可在实例详情页面获取，详情请参见查看接入点。由于本文使用的是VPC实例，因此使用默认接入点中的任意一个接入点。
topic	日志输出到消息队列Kafka的Topic，请指定为您已创建的Topic。
version	Kafka的版本，可在消息队列Kafka的实例详情页面获取。 注意 ◦ 不配置此参数会报错。 ◦ 由于不同版本的Filebeat支持的Kafka版本不同，例如8.2及以上版本的Filebeat支持的Kafka版本为2.2.0，因此version需要设置为Filebeat支持的Kafka版本，否则会出现类似报错：`Exiting: error initializing publisher: unknown/unsupported kafka vesion '2.2.0' accessing 'output.kafka.version' (source:'filebeat.kafka.yml')`，详细信息请参见version。

4. 启动Filebeat。

```
./filebeat -e -c filebeat.kafka.yml
```

步骤二：配置Logstash管道

1. 登录阿里云Elasticsearch控制台。
2. 进入目标实例。
   i. 在顶部菜单栏处，选择地域。
   ii. 在左侧导航栏，单击**Logstash实例**，然后在**Logstash实例**中单击目标实例ID。
3. 在左侧导航栏，单击**管道管理**。
4. 单击**创建管道**。
5. 在**创建管道任务**页面，输入**管道ID**并配置管道。

   本文使用的管道配置如下。

```
input {
  kafka {
    bootstrap_servers => ["alikafka-post-cn-zvp2n4v7****-1-vpc.alikafka.aliyuncs.com:9092,alikafka-post-cn-zvp2n4v7****-2-vpc.alikafka.aliyuncs.com:9092,alikafka-post-cn-zvp2n4v7****-3-vpc.alikafka.aliyuncs.com:9092"]
    group_id => "es-test"
    topics => ["estest"]
    codec => json
 }
}
filter {
}
output {
  elasticsearch {
    hosts => "http://es-cn-n6w1o1x0w001c****.elasticsearch.aliyuncs.com:9200"
    user =>"elastic"
    password =>"<your_password>"
    index => "kafka-%{+YYYY.MM.dd}"
 }
}
```

input参数说明

参数	说明
bootstrap_servers	消息队列Kafka实例的接入点，可在实例详情页面获取，详情请参见查看接入点。由于本文使用的是VPC实例，因此使用默认接入点。
group_id	指定为您已创建的Consumer Group的名称。
topics	指定为您已创建的Topic的名称，需要与Filebeat中配置的Topic名称保持一致。
codec	设置为json，表示解析JSON格式的字段，便于在Kibana中分析。

output参数说明

参数	说明
hosts	阿里云Elasticsearch的访问地址，取值为 `http://<阿里云Elasticsearch实例的内网地址>:9200` 。 **说明** 您可在阿里云Elasticsearch实例的基本信息页面获取其内网地址，详情请参见查看实例的基本信息。
user	访问阿里云Elasticsearch的用户名，默认为elastic。您也可以使用自建用户，详情请参见通过Elasticsearch X-Pack角色管理实现用户权限管控。
password	访问阿里云Elasticsearch的密码，在创建实例时设置。如果忘记密码，可进行重置，重置密码的注意事项及操作步骤请参见重置实例访问密码。

参数	说明
index	索引名称。设置为 `kafka-%{+YYYY.MM.dd}` 表示索引名称以kafka为前缀，以日期为后缀，例如 `kafka-2020.05.27` 。

更多Config配置详情请参见Logstash配置文件说明。

6. 单击**下一步**，配置管道参数。

参数	说明
管道工作线程	并行执行管道的Filter和Output的工作线程数量。当事件出现积压或CPU未饱和时，请考虑增大线程数，更好地使用CPU处理能力。默认值：实例的CPU核数。
管道批大小	单个工作线程在尝试执行Filter和Output前，可以从Input收集的最大事件数目。较大的管道批大小可能会带来较大的内存开销。您可以设置LS_HEAP_SIZE变量，来增大JVM堆大小，从而有效使用该值。默认值：125。
管道批延迟	创建管道事件批时，将过小的批分派给管道工作线程之前，要等候每个事件的时长，单位为毫秒。默认值：50ms。
队列类型	用于事件缓冲的内部排队模型。可选值： ○ **MEMORY**：默认值。基于内存的传统队列。 ○ **PERSISTED**：基于磁盘的ACKed队列（持久队列）。
队列最大字节数	请确保该值小于您的磁盘总容量。默认值：1024 MB。
队列检查点写入数	启用持久性队列时，在强制执行检查点之前已写入事件的最大数目。设置为0，表示无限制。默认值：1024。

警告 配置完成后，需要保存并部署才能生效。保存并部署操作会触发实例重启，请在不影响业务的前提下，继续执行以下步骤。

7. 单击**保存**或者**保存并部署**。
   - **保存**：将管道信息保存在Logstash里并触发实例变更，配置不会生效。保存后，系统会返回**管道管理**页面。可在**管道列表**区域，单击**操作**列下的**立即部署**，触发实例重启，使配置生效。
   - **保存并部署**：保存并且部署后，会触发实例重启，使配置生效。

步骤三：查看日志消费状态

1. 进入消息队列Kafka控制台。
2. 参见查看消费状态，查看详细消费状态。

   预期结果如下：

分区ID	owner	最大位点	消费位点	堆积量	最近消费时间
0	logstash-0_/	76	76	0	2020年5月27日 14:12:07
1	logstash-0_/	51	51	0	2020年5月27日 14:12:07
2	logstash-1_/	67	67	0	2020年5月27日 14:12:07
3	logstash-1_/	52	52	0	2020年5月27日 14:12:07
4	logstash-2_/	69	69	0	2020年5月27日 14:12:07
5	logstash-2_/	54	54	0	2020年5月27日 14:12:07

步骤四：通过Kibana过滤日志数据

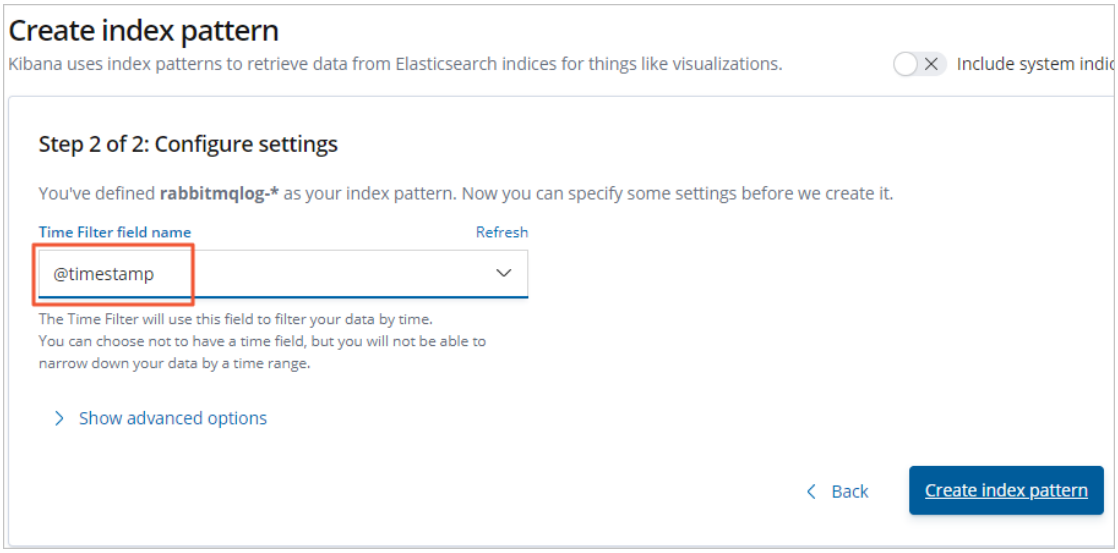
1. 登录目标阿里云Elasticsearch实例的Kibana控制台，根据页面提示进入Kibana主页。

   登录Kibana控制台的具体操作，请参见登录Kibana控制台。

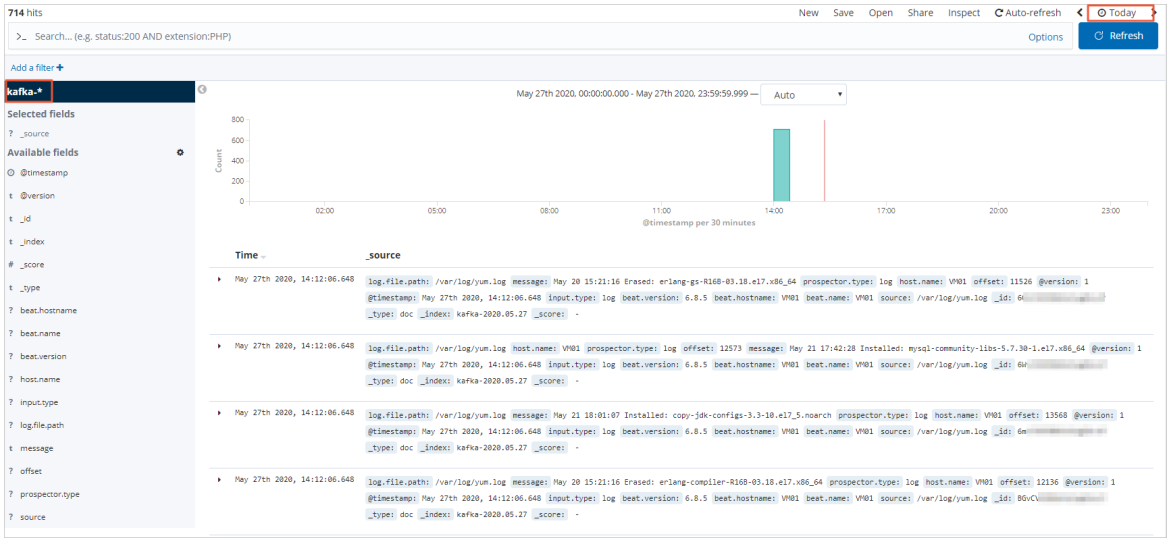
   > **说明** 本文以阿里云Elasticsearch 6.7.0版本为例，其他版本操作可能略有差别，请以实际界面为准。

2. 创建一个索引模式。
   i. 在左侧导航栏，单击**Management**。
   ii. 在Kibana区域，单击**Index Patterns**。
   iii. 单击**Create index pattern**。
   iv. 输入**Index pattern**（本文使用kafka-*），单击**Next step**。

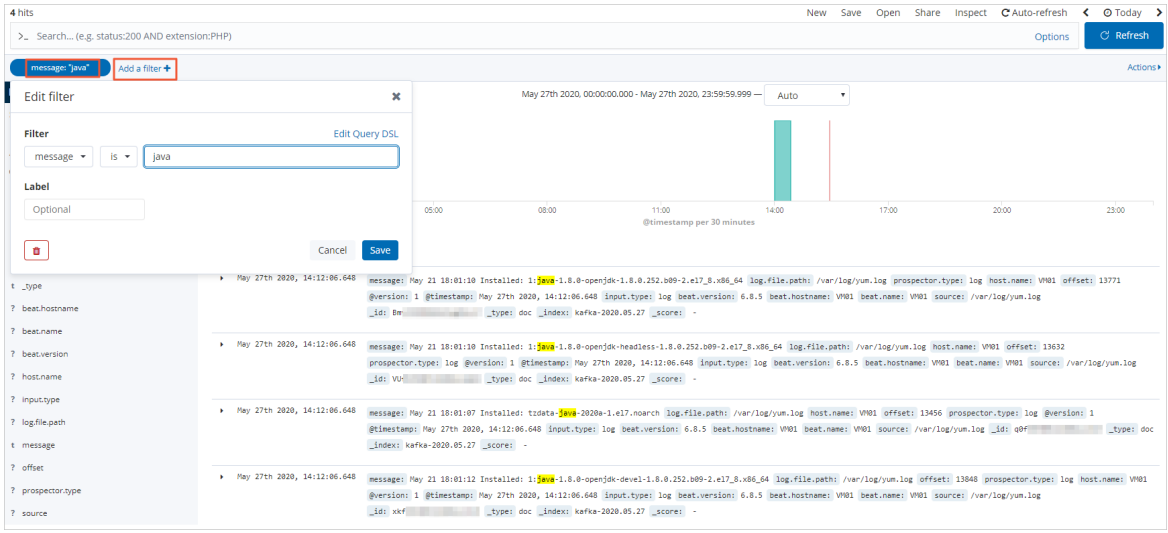
v. 选择Time Filter field name（本文选择@timestamp），单击Create index pattern。



3. 在左侧导航栏，单击Discover。
4. 从页面左侧的下拉列表中，选择您已创建的索引模式（kafka-*）。
5. 在页面右上角，选择一段时间，查看对应时间段内的Filebeat采集的日志数据。



6. 单击Add a filter，在Add filter页面中设置过滤条件，查看符合对应过滤条件的日志数据。



常见问题

Q：同步日志数据出现问题，管道一直在生效中，无法将数据导入Elasticsearch，如何解决？

A：查看Logstash实例的主日志是否有报错，根据报错判断原因，具体操作请参见查询日志。常见的原因及解决方法如下。

原因	解决方法
Kafka的接入点不正确。	参见查看接入点获取正确的接入点。完成后，修改管道配置替换错误接入点。
Logstash与Kafka不在同一VPC下。	重新购买同一VPC下的实例。购买后，修改现有管道配置。 **说明** VPC实例只能通过专有网络VPC访问消息队列Kafka版。
Kafka或Logstash集群的配置太低，例如使用了测试版集群。	升级集群规格，完成后，刷新实例，观察变更进度。升级Logstash实例规格的具体操作，请参见升配集群；升级Kafka实例规格的具体操作，请参见升级实例配置。
管道配置中包含了file_extend，但没有安装logstash-output-file_extend插件。	选择以下任意一种方式处理： • 安装logstash-output-file_extend插件。具体操作，请参见 安装或卸载插件。 • 中断变更，等到实例处于变更中断状态后，在管道配置中，去掉file_extend配置，触发重启恢复。

更多问题原因及解决方法，请参见Logstash数据写入问题排查方案。

7.5. 查询分析RocketMQ客户端日志

es分析阿里云RocketMQ客户端日志

本文使用Beats、Elasticsearch、Logstash和Kibana，在分布式环境下采集、汇聚、解析阿里云RocketMQ客户端SDK日志，帮助您在消息队列开发场景中快速定位并解决应用开发问题。

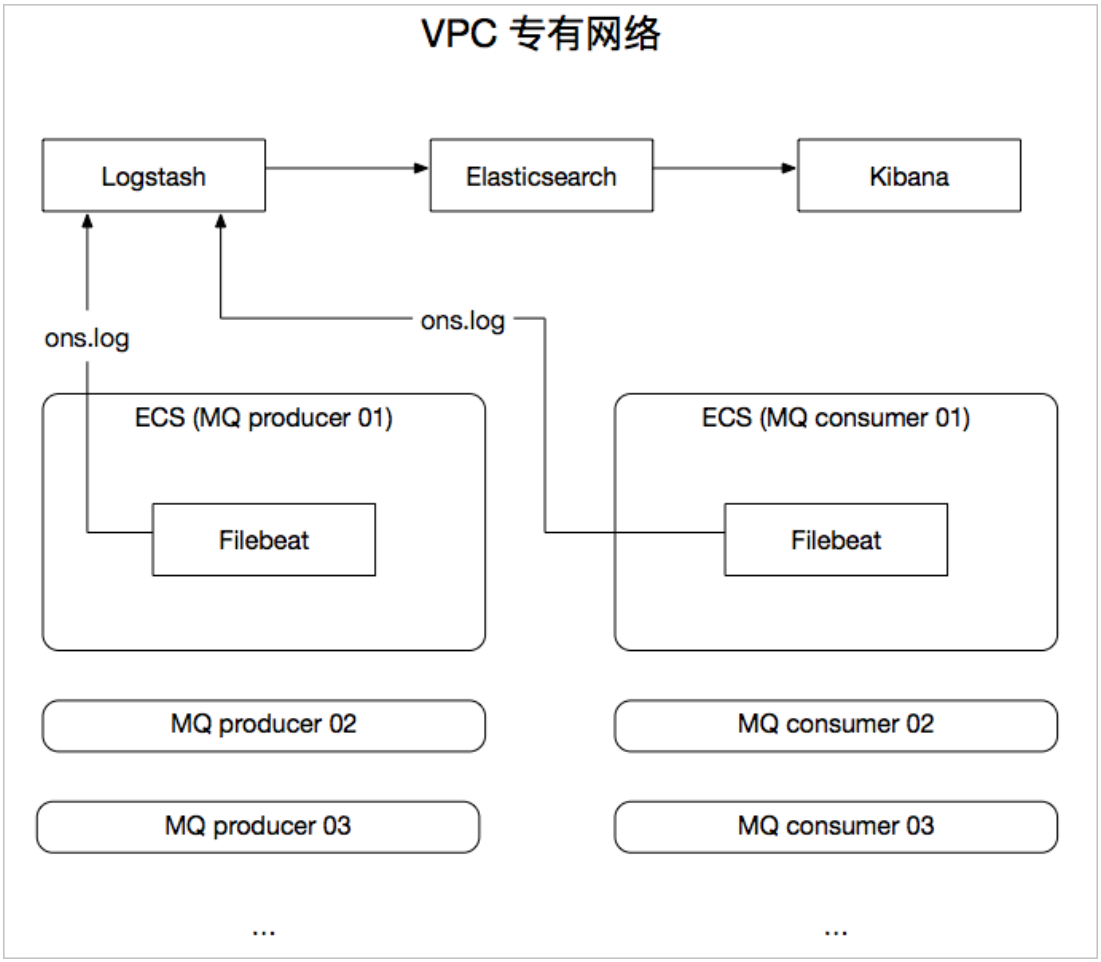
背景信息

阿里云Elasticsearch（简称ES）已具备Elastic Stack全栈套件：Elasticsearch、Logstash、Kibana和Beats，

具有日志汇聚、快速分析、可视化展示等能力，与开源产品相比具有以下优势：

- Beats和Logstash服务部署在专有网络VPC（Virtual Private Cloud）中，所有数据私网通信，高速且安全。
- Elastic Stack全栈套件具有“0部署，轻运维”的特性，省去了在分布式环境下逐个节点安装与配置Beats采集器的繁琐工作，同时解决了分布式消息中间件日志分散、采集困难的问题。
- 支持在控制台快速添加和移除Beats实例，能够适应RocketMQ的弹性伸缩特性。

本方案使用的产品包括：阿里云消息队列RocketMQ版和阿里云Elasticsearch，方案架构如下。



操作流程

1. 准备工作

   完成创建阿里云ES实例和Logstash实例、开通RocketMQ消息队列服务、创建ECS实例并在实例中安装云助手和Docker服务。各实例或服务的功能如下：

   - RocketMQ消息队列服务：提供RocketMQ资源，包括实例、Topic、Group。
   - ECS实例：安装Filebeat并运行RocketMQ测试工程，生成客户端日志。
   - 阿里云Logstash实例：通过管道配置将Filebeat采集的日志同步到阿里云ES中。
   - 阿里云ES实例：对日志进行分析，并进行可视化展示。

   > 注意 请确保阿里云ES实例、Logstash实例和ECS实例在同一专有网络VPC（Virtual Private Cloud）下。

2. 步骤一：创建并配置Filebeat采集器

   通过Filebeat将RocketMQ的客户端日志采集到Logstash中。

3. 步骤二：创建并运行Logstash管道

   通过管道配置，使用Grok filter插件处理Filebeat采集的RocketMQ客户端日志，然后同步到阿里云ES中。

4. 步骤三：模拟RocketMQ客户端日志

   在安装Filebeat的ECS上运行RocketMQ客户端测试工程，生成日志数据。

5. 步骤四：通过Kibana查看日志

   创建索引模式，并在Kibana的Discover页面查看日志的详细信息。

6. 步骤五：通过Kibana分析日志

   以筛选ERROR级别日志为例，演示在Kibana中分析并解读日志数据的方法。

准备工作

1. 创建阿里云ES实例，并开启自动创建索引功能。

   具体操作步骤请参见创建阿里云Elasticsearch实例和快速访问与配置。

2. 创建阿里云Logstash实例，要求与阿里云ES实例在同一VPC下。

   具体操作步骤请参见创建阿里云Logstash实例。

3. 开通RocketMQ消息队列服务，并创建所需资源，包括实例、Topic、Group。

   具体操作步骤请参见开通消息队列服务并授权和创建资源。

4. 创建一个或多个ECS实例，并且该实例与阿里云ES实例和Logstash实例处于同一VPC下。

   具体操作步骤请参见使用向导创建实例。

   > 注意 ECS的操作系统必须为Aliyun Linux、RedHat或CentOS。因为Beats仅支持这三种操作系统。

5. 在ECS实例中安装云助手和Docker服务。

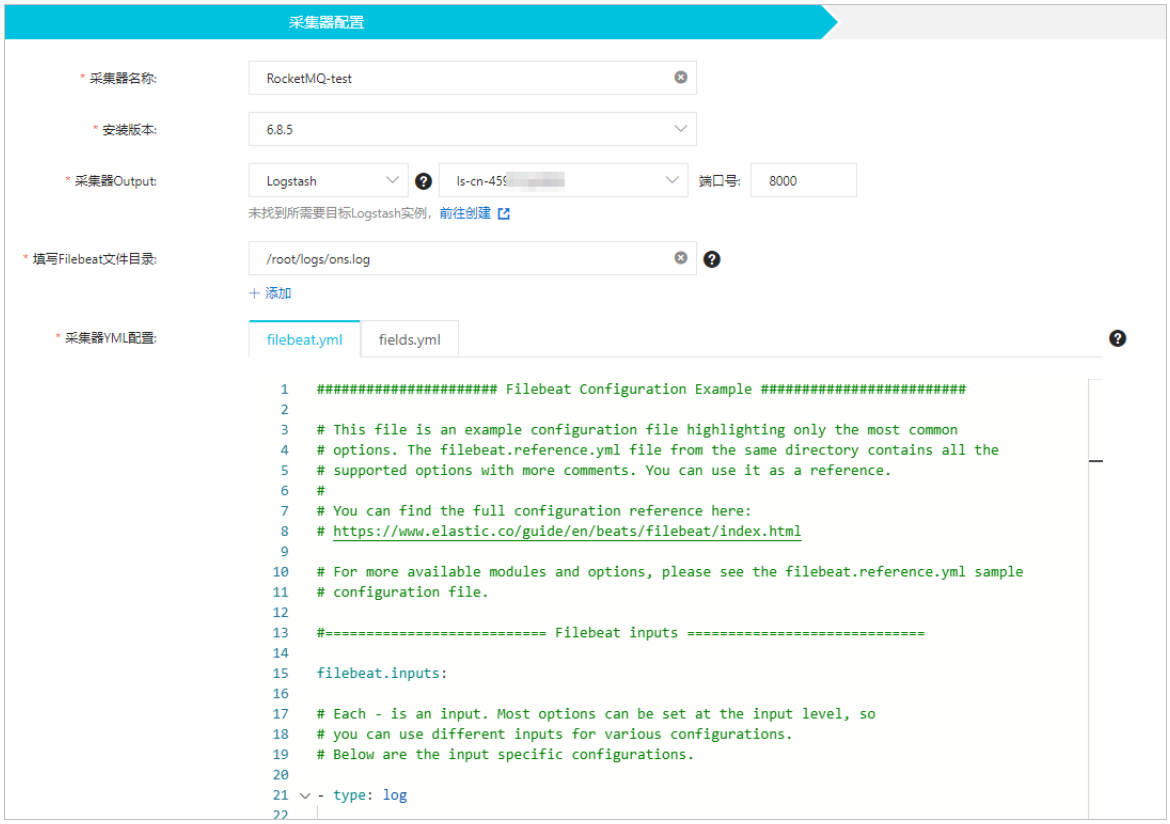
   具体操作步骤请参见安装云助手客户端和部署并使用Docker（Alibaba Cloud Linux 2）。

   > 注意 所选ECS必须安装云助手和Docker，且对应服务已正常运行。因为安装Beats时会依赖这两个服务。

步骤一：创建并配置Filebeat采集器

1. 登录阿里云Elasticsearch控制台。
2. 在**新建采集器**区域中，单击**Filebeat**。
3. 安装并配置采集器。

   详情请参见采集ECS服务日志和采集器YML配置，本文使用的配置如下。



参数	说明
采集器名称	自定义采集器名称。
安装版本	目前只支持6.8.5版本。
采集器Output	指定目标阿里云Logstash的实例ID，在YML配置中不需要重新指定Output。
填写Filebeat文件目录	填写数据源所在的目录，同时需要在YML配置中开启log数据采集，并配置log路径。

采集器Yml配置	○ 开启log数据采集，将 `enabled` 修改为 `true` 。
	○ 修改 `paths` 为具体的日志文件路径，与Filebeat文件目录保持一致。
	○ 调整Multiline options相关配置，解决多行日志及java stacktrace的情况。
	`multiline.pattern: '^([0-9]{4}-[0-9]{2}-[0-9]{2})'`
	`multiline.negate: true`
	`multiline.match: after`
	`multiline.timeout: 120s #default 5s`
	`multiline.max_lines: 10000 #default 500`

完整的filebeat.yml配置如下。

filebeat.yml fields.yml

```
14
15  filebeat.inputs:
16
17  # Each - is an input. Most options can be set at the input level, so
18  # you can use different inputs for various configurations.
19  # Below are the input specific configurations.
20
21  - type: log
22
23    # Change to true to enable this input configuration.
24    enabled: true
25
26    # Paths that should be crawled and fetched. Glob based paths.
27    paths:
28      - /root/logs/ons.log
49    ### Multiline options
50    multiline.pattern: '^([0-9]{4}-[0-9]{2}-[0-9]{2})'
51    multiline.negate: true
52    multiline.match: after
53    multiline.timeout: 120s #default 5s
54    multiline.max_lines: 10000 #default 500
```

```
###################### Filebeat Configuration Example #########################
#============================ Filebeat inputs ===============================
filebeat.inputs:
  # Each - is an input. Most options can be set at the input level, so
  # you can use different inputs for various configurations.
  # Below are the input specific configurations.
  - type: log
    # Change to true to enable this input configuration.
    enabled: true
    # Paths that should be crawled and fetched. Glob based paths.
    paths:
      - /root/logs/ons.log
      #- c:\programdata\elasticsearch\logs\*
    ### Multiline options
    # Multiline can be used for log messages spanning multiple lines. This is common
    # for Java Stack Traces or C-Line Continuation
    multiline.pattern: '^([0-9]{4}-[0-9]{2}-[0-9]{2})'
    multiline.negate: true
    multiline.match: after
    multiline.timeout: 120s #default 5s
    multiline.max_lines: 10000 #default 500
#==================== Elasticsearch template setting ==========================
setup.template.settings:
  index.number_of_shards: 3
  #index.codec: best_compression
  #_source.enabled: false
#================================ Processors =====================================
# Configure processors to enhance or manipulate events generated by the beat.
processors:
  - add_host_metadata: ~
  - add_cloud_metadata: ~
```

4. 单击**下一步**。
5. 在**采集器安装**配置向导中，选择安装采集器的ECS实例。

6. 启动采集器并查看采集器安装情况。
   i. 单击**启动**。
   启动成功后，系统弹出**启动成功**对话框。
   ii. 单击**前往采集中心查看**，返回**Beats数据采集中心**页面，在**采集器管理**区域中，查看启动成功的Filebeat采集器。
   iii. 等待**采集器状态**变为**已生效1/1**后，单击右侧**操作**栏下的**查看运行实例**。

iv. 在**查看运行实例**页面，查看**采集器安装情况**，当显示为**心跳正常**时，说明采集器安装成功。

步骤二：创建并运行Logstash管道

1. 登录阿里云Elasticsearch控制台。
2. 进入目标实例。
   i. 在顶部菜单栏处，选择地域。
   ii. 在左侧导航栏，单击**Logstash实例**，然后在**Logstash实例**中单击目标实例ID。
3. 在左侧导航栏，单击**管道管理**。
4. 单击**创建管道**。
5. 在**创建管道任务**页面，输入管道ID并配置管道。

   本文使用的Config配置如下。

```
input {
    beats {
        port => 8000
    }
}
filter {
    grok    {
          match => {
              "message" => "%{TIMESTAMP_ISO8601:log_time} %{LOGLEVEL:log_level} %{GRE
EDYDATA:log_message}"
        }
    }
}
output {
  elasticsearch {
    hosts => "http://es-cn-4591jumei000u****.elasticsearch.aliyuncs.com:9200"
    user =>"elastic"
    password =>"<your_password>"
    index => "rocketmq-%{+YYYY.MM.dd}"
 }
}
```

   - input：输入插件。以上配置使用beats插件，指定8000端口。
   - filter：过滤插件。以上配置提供了Grok filter示例，解析RocketMQ客户端SDK日志，提取`log_time`、`log_level`以及`log_message`三个字段信息，方便您分析日志。您也可以根据需求修改`filter.grok.match`的内容。
   - output：输出插件。以上配置使用elasticsearch插件，相关参数说明如下。

参数	说明
hosts	阿里云ES的访问地址，设置为 `http://<阿里云ES实例的内网地址>:9200` 。 **说明** 您可在阿里云ES实例的基本信息页面获取其内网地址，详情请参见查看实例的基本信息。
user	访问阿里云ES的用户名，默认为elastic。您也可以使用自建用户，详情请参见通过Elasticsearch X-Pack角色管理实现用户权限管控。
password	访问阿里云ES的密码，在创建实例时设置。如果忘记密码，可进行重置，重置密码的注意事项及操作步骤请参见重置实例访问密码。
index	索引名称。设置为 `rocketmq-%{+YYYY.MM.dd}` 表示索引名称以rocketmq为前缀，以日期为后缀，例如 `rocketmq-2020.05.27` 。

更多Config配置详情请参见Logstash配置文件说明。

6. 单击**下一步**，配置管道参数。

参数	说明
管道工作线程	并行执行管道的Filter和Output的工作线程数量。当事件出现积压或CPU未饱和时，请考虑增大线程数，更好地使用CPU处理能力。默认值：实例的CPU核数。
管道批大小	单个工作线程在尝试执行Filter和Output前，可以从Input收集的最大事件数目。较大的管道批大小可能会带来较大的内存开销。您可以设置LS_HEAP_SIZE变量，来增大JVM堆大小，从而有效使用该值。默认值：125。
管道批延迟	创建管道事件批时，将过小的批分派给管道工作线程之前，要等候每个事件的时长，单位为毫秒。默认值：50ms。
队列类型	用于事件缓冲的内部排队模型。可选值： ○ **MEMORY**：默认值。基于内存的传统队列。 ○ **PERSISTED**：基于磁盘的ACKed队列（持久队列）。

参数	说明
队列最大字节数	请确保该值小于您的磁盘总容量。默认值：1024 MB。
队列检查点写入数	启用持久性队列时，在强制执行检查点之前已写入事件的最大数目。设置为0，表示无限制。默认值：1024。

警告 配置完成后，需要保存并部署才能生效。保存并部署操作会触发实例重启，请在不影响业务的前提下，继续执行以下步骤。

7. 单击**保存**或者**保存并部署**。
   - **保存**：将管道信息保存在Logstash里并触发实例变更，配置不会生效。保存后，系统会返回**管道管理**页面。可在**管道列表**区域，单击**操作**列下的**立即部署**，触发实例重启，使配置生效。
   - **保存并部署**：保存并且部署后，会触发实例重启，使配置生效。

步骤三：模拟RocketMQ客户端日志

1. 连接安装了Filebeat的ECS实例。

   具体操作步骤请参见连接实例。

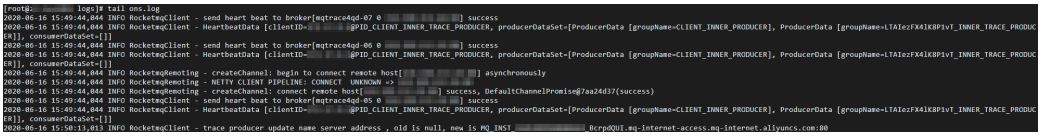
2. 搭建并运行消息队列RocketMQ版测试工程，发送若干条测试消息，生成日志。

   具体操作步骤请参见RocketMQ Demo工程。

   发送消息

```
[root@ 1.0-SNAPSHOT]# java -jar mq-demo-producer-1.0-SNAPSHOT-jar-with-dependencies.jar
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Producer Started
testes
Tue Jun 16 15:49:43 CST 2020 Send mq message success! Topic is:testes msgId is: AC1100010F55070DEA4E50A4E4440000
testes
Tue Jun 16 15:49:43 CST 2020 Send mq message success! Topic is:testes msgId is: AC1100010F55070DEA4E50A4E4A70002
testes
Tue Jun 16 15:49:43 CST 2020 Send mq message success! Topic is:testes msgId is: AC1100010F55070DEA4E50A4E4C70005
testes
Tue Jun 16 15:49:43 CST 2020 Send mq message success! Topic is:testes msgId is: AC1100010F55070DEA4E50A4E4E60008
testes
Tue Jun 16 15:49:43 CST 2020 Send mq message success! Topic is:testes msgId is: AC1100010F55070DEA4E50A4E505000B
testes
Tue Jun 16 15:49:43 CST 2020 Send mq message success! Topic is:testes msgId is: AC1100010F55070DEA4E50A4E524000E
testes
Tue Jun 16 15:49:43 CST 2020 Send mq message success! Topic is:testes msgId is: AC1100010F55070DEA4E50A4E5600011
testes
Tue Jun 16 15:49:43 CST 2020 Send mq message success! Topic is:testes msgId is: AC1100010F55070DEA4E50A4E57E0014
testes
Tue Jun 16 15:49:43 CST 2020 Send mq message success! Topic is:testes msgId is: AC1100010F55070DEA4E50A4E59B0017
testes
Tue Jun 16 15:49:44 CST 2020 Send mq message success! Topic is:testes msgId is: AC1100010F55070DEA4E50A4E5B9001A
```

   查看日志



步骤四：通过Kibana查看日志

1. 登录目标阿里云ES实例的Kibana控制台。

   具体操作步骤请参见登录Kibana控制台。

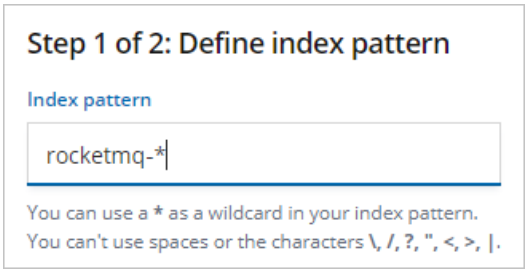
2. 创建一个索引模式。

   i. 在左侧导航栏，单击Management。

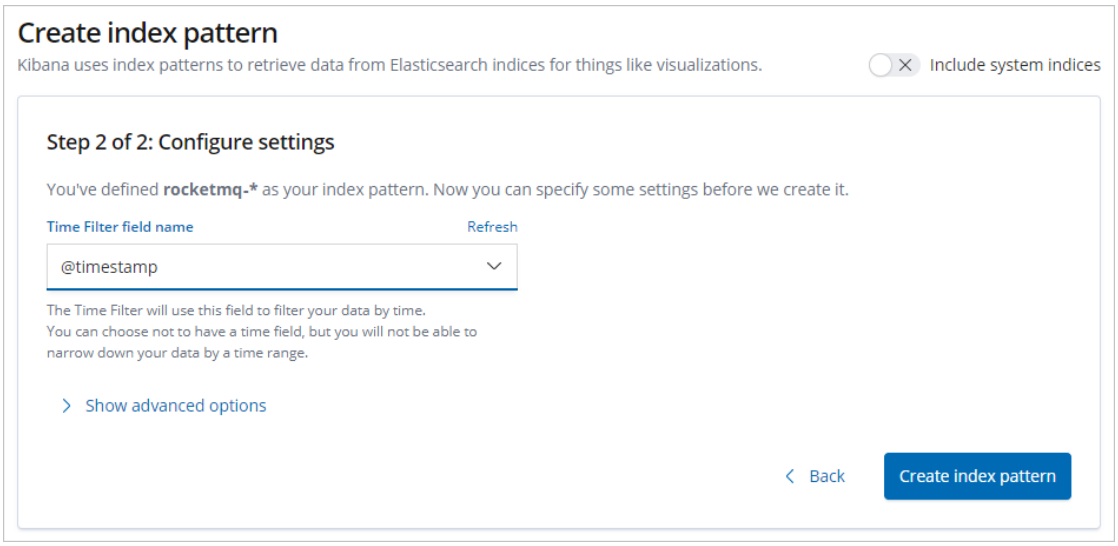
ii. 在Kibana区域，单击**Index Patterns**。

iii. 单击**Create index pattern**。

iv. 输入**Index pattern**（本文使用rocketmq-*），单击**Next step**。

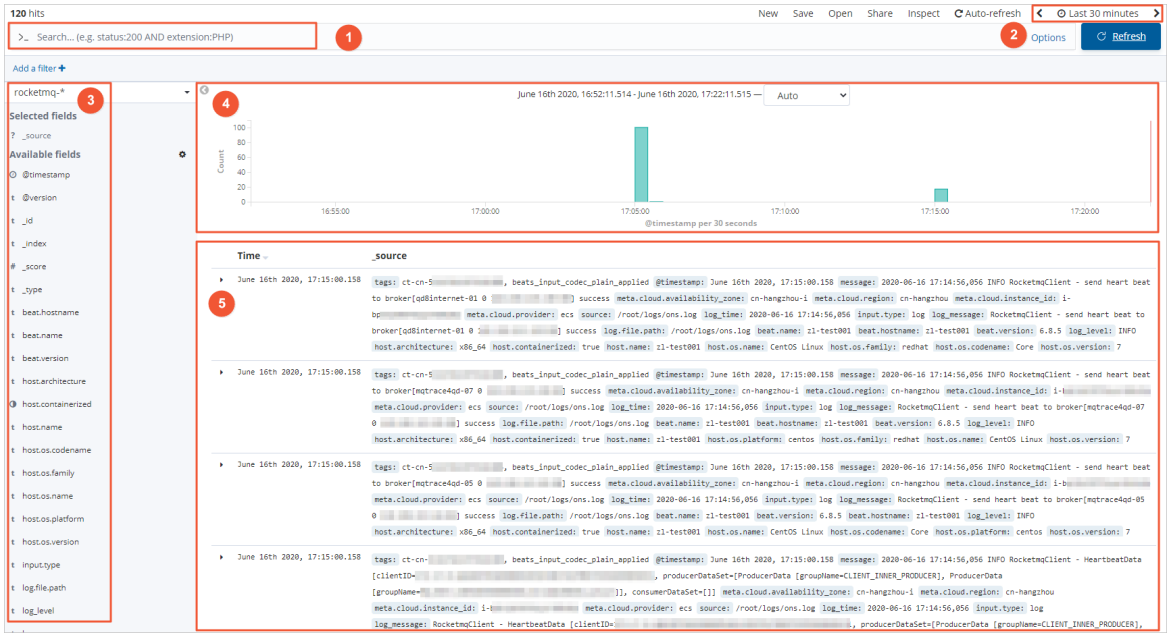


v. 选择**Time Filter field name**（本文选择@timestamp），单击**Create index pattern**。



> **说明** 选择@timestamp作为时间过滤器，可以方便通过直方图等其他可视化图表展示日志数据。

3. 在左侧导航栏，单击**Discover**。
4. 从页面左侧的下拉列表中，选择您已创建的索引模式（rocketmq-*）。
5. 在页面右上角，选择一段时间，查看对应时间段内的Filebeat采集的日志数据。



区域	说明
①	数据查询区域。您可根据需求在此区域输入查询语句，查询语句需符合Kibana Query Language要求，例如 `log_level:ERROR` 。
②	时间和刷新频率选择区域。您可在此区域选择展示哪个时间段内的数据，以及数据刷新的频率。
③	字段选择区域。您可根据需求在此区域选择要展示的字段。
④	直方图展示区域。系统会根据创建索引模式时指定的@timestamp字段，通过直方图在此区域汇聚展示所选字段的数据。
⑤	数据展示区域。您可在此区域查看所选字段的数据。

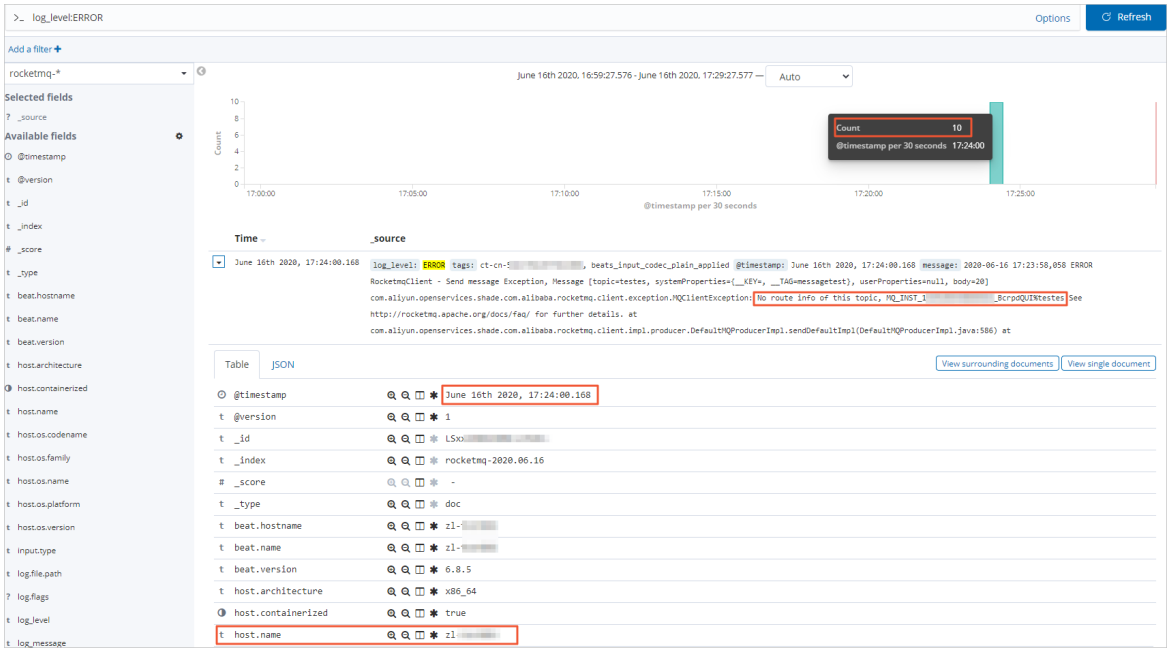
步骤五：通过Kibana分析日志

对于Java开发类日志，开发者在运维时最为关心的是ERROR日志，以下示例演示如何筛选ERROR级别日志。为了演示需要，本例在RocketMQ控制台将topic删除，制造ERROR场景。具体分析方法如下：

1. 在**Discover**页面的数据查询区域输入如下搜索语句，筛选出ERROR级别日志。

```
log_level:ERROR
```

2. 解读筛选的日志信息。



从筛选结果可以看到：

- 所选时间段内有10条ERROR日志。
- 具体的报错信息是 `No route info of this topic, MQ_INST_**********_BcrpdQUI%testes` ，从报错信息可以判断出现ERROR日志的原因是该实例（MQ_INST_**********_BcrpdQUI）中topic（testes）的路由信息丢失，可能原因是topic配置错误、topic被误删等。
- 报错的时间点为 `June 16th 2020, 17:24:00.168` 。
- 将单个报错信息展开后，还能看到具体抛出ERROR日志的主机（ `host.name` ）信息，进而精确定位到具体的应用实例，然后通过远程登录该实例快速定位排查问题。

7.6. 通过Elasticsearch和rsbeat实时分析Redis slowlog

Redis是目前流行的高性能key-value数据库，但如果使用不当，很容易出现慢查询。慢查询过多或者一个时间较长（例如20s）的慢查询会导致操作队列（Redis是单进程）堵塞，可能会导致服务不可用。因此您需要实时收集并分析Redis slowlog，在出现问题时快速定位解决。本文介绍如何通过Elasticsearch和rsbeat实时分析Redis slowlog。

背景信息

通过Elasticsearch和rsbeat实时分析Redis slowlog的原理为：使用rsbeat将Redis slowlog采集到Elasticsearch中，然后在Kibana中进行图形化分析。相关概念说明如下：

- Elasticsearch：是一个基于Lucene的实时分布式的搜索与分析引擎，是遵从Apache开源条款的一款开源产品，是当前主流的企业级搜索引擎。它提供了一个分布式服务，可以使您快速的近乎于准实时的存储、查询和分析超大数据集，通常被用来作为构建复杂查询特性和需求强大应用的基础引擎或技术。

  阿里云Elasticsearch兼容开源Elasticsearch的功能，以及Security、Machine Learning、Graph、APM等商业功能，致力于数据分析、数据搜索等场景服务，支持5.5.3、6.3.2、6.7.0、6.8.0和7.4.0等版本，并提供了商业插件X-Pack服务。在开源Elasticsearch的基础上提供企业级权限管控、安全监控告警、自动报表生成等功能。本文使用阿里云Elasticsearch进行演示，详情请参见什么是阿里云Elasticsearch。
- rsbeat：用来收集和分析Redis慢日志的采集器，详情请参见rsbeat官方文档。

- Redis：是一个开源的、基于内存的数据结构存储器，可以用作数据库、缓存和消息中间件，详情请参见Redis官方说明。

  云数据库Redis版（ApsaraDB for Redis）是兼容开源Redis协议标准、提供内存加硬盘的混合存储方式的数据库服务，基于高可靠双机热备架构及可平滑扩展的集群架构，满足高读写性能场景及弹性变配的业务需求。本文使用云数据库Redis版进行演示，更多详情请参见什么是云数据库Redis版。

操作流程

1. 准备工作

   创建阿里云Elasticsearch实例、云数据库Redis版实例（以下简称Redis实例）和ECS实例，三者在同一专有网络VPC（Virtual Private Cloud）下。

2. 步骤一：配置Redis慢查询参数

   根据需求设置Redis slowlog生成的条件，以及可记录的slowlog的最大条数。

3. 步骤二：安装并配置rsbeat

   在ECS中安装rsbeat，并在其配置文件中指定Redis和Elasticsearch服务。

4. 步骤三：通过Kibana图形化分析slowlog

   通过Kibana查看日志详细信息，并根据需求进行统计分析。

准备工作

1. 创建阿里云Elasticsearch实例，并开启自动创建索引功能。

   具体操作步骤请参见创建阿里云Elasticsearch实例和配置YML参数。本文使用的实例版本为通用商业版6.7。

2. 创建Redis实例。

   具体操作步骤请参见步骤1：创建实例。本文使用的实例版本为Redis 5.0社区版，并且与阿里云Elasticsearch实例在同一VPC下，便于内网访问。

3. 创建ECS实例。

   具体操作步骤请参见使用向导创建实例。本文使用的实例镜像为CentOS 7.6 64位，并且与Redis和Elasticsearch实例在同一VPC下。

4. 配置Redis实例的访问白名单。

   将ECS实例的内网IP地址添加到Redis实例的白名单中，具体操作步骤请参见设置白名单。

步骤一：配置Redis慢查询参数

1. 登录Redis管理控制台。
2. 在顶部菜单栏处，选择地域。
3. 在**实例列表**页面，单击目标实例ID或者其右侧**操作**列下的 **> 管理**。
4. 在左侧导航栏，单击**参数设置**。
5. 在参数设置列表中，找到**slowlog-log-slower-than**和**slowlog-max-len**参数，将其修改为您期望的值。

参数	说明	示例

参数	说明	示例
slowlog-log-slower-than	当命令执行时间（不包括排队时间）超过该参数值时，该命令会被定义为慢查询，并记录到slowlog中。单位为微秒，默认为10000，即10毫秒。 注意 负数表示关闭慢查询日志功能，0表示记录所有命令操作。	本文将该参数值设置为20000。表示在slowlog中记录执行时长超过20毫秒的命令。
slowlog-max-len	slowlog中可以记录的最大慢查询命令的条数。当slowlog中的记录数超过最大值后，Redis会将最早的slowlog删除。	本文将该参数值设置为100。表示在slowlog中记录最近100条慢查询命令。

步骤二：安装并配置rsbeat

1. 连接ECS实例。

   具体操作步骤请参见连接实例。

2. 安装rsbeat。

   本文使用5.3.2版本。

   ```
   wget https://github.com/Yourdream/rsbeat/archive/master.zip
   unzip master.zip
   ```

3. 修改rsbeat配置。

   i. 执行以下命令打开rsbeat.yml文件。

   ```
   cd rsbeat-master
   vim rsbeat.yml
   ```

   ii. 按照以下说明修改rsbeat和output.elasticsearch参数配置，并保存。

```
############################# Rsbeat ######################################

rsbeat:
  # Defines how often an event is sent to the output
  period: 1s
  redis: ["r-[redacted].redis.rds.aliyuncs.com:6379"]
  slowerThan: 100

#================================ General =====================================

# The name of the shipper that publishes the network data. It can be used to group
# all the transactions sent by a single shipper in the web interface.
#name:

# The tags of the shipper are included in their own field with each
# transaction published.
#tags: ["service-X", "web-tier"]

# Optional fields that you can specify to add additional information to the
# output.
#fields:
#  env: staging

#================================ Outputs =====================================

# Configure what outputs to use when sending the data collected by the beat.
# Multiple outputs may be used.

#-------------------------- Elasticsearch output ------------------------------
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["es-cn-[redacted].elasticsearch.aliyuncs.com:9200"]

  # Optional protocol and basic auth credentials.
  #protocol: "https"
  username: "elastic"
  password: "[redacted]"

  # Overwrite existing template
  template.overwrite: true
```

rebeat配置

参数	说明
period	每隔多久将slowlog输出到Elasticsearch。
redis	Redis实例的连接地址，获取方式请参见查看连接地址。 注意 由于配置文件中没有定义Redis实例的密码，因此在获取连接地址后，您还需要开启免密访问，才能确保rsbeat能够访问Redis实例，开启方法请参见开启专有网络免密访问。
slowerThan	定义将 `config set slowlog-log-slower-than` 命令发送到Redis服务器的时间。单位为微秒。

output.elasticsearch配置

参数	说明
hosts	阿里云Elasticsearch实例的连接地址，可在实例的基本信息页面获取，详情请参见查看实例的基本信息。
username	阿里云Elasticsearch实例的访问用户名，默认为elastic。
password	对应用户的密码。elastic用户的密码在创建实例时设定，如果忘记可重置，重置密码的注意事项和操作步骤请参见重置实例访问密码。
template.overwrite	是否覆盖已存在的同名模板，默认为true。

4. 启动rsbeat服务。

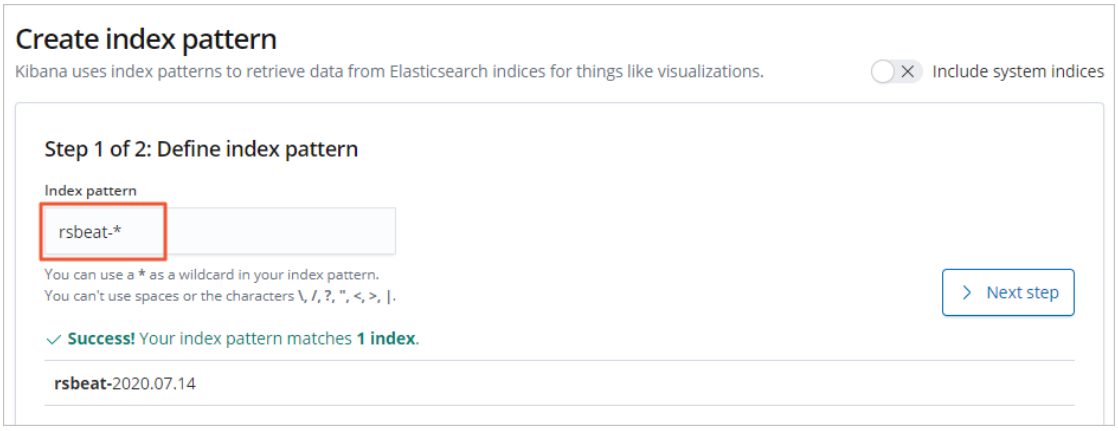
```
./rsbeat.linux.amd64 -c rsbeat.yml -e -d "*"
```

步骤三：通过Kibana图形化分析slowlog

1. 登录目标阿里云Elasticsearch实例的Kibana控制台。

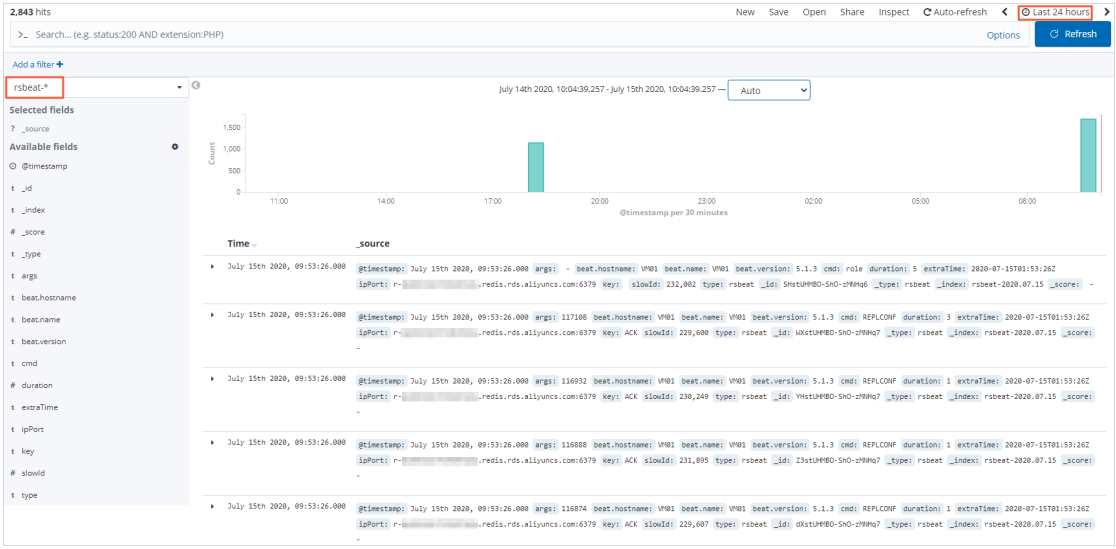
   具体操作步骤请参见登录Kibana控制台。

2. 创建索引模式。
   i. 在左侧导航栏，单击Management。
   ii. 在Kibana区域，单击Index Patterns。
   iii. 单击Create index pattern。
   iv. 输入Index pattern名称，单击Next step。



   v. 从Time Filter field name中，选择时间过滤器字段名（本文选择@timestamp）。
   vi. 单击Create index pattern。
3. 查看slowlog的详细信息。
   i. 在左侧导航栏，单击Discover。
   ii. 在Discover页面左侧，选择目标索引模式rsbeat-*。

iii. 在页面右上角，选择一段时间，查看该时间段内的slowlog信息。

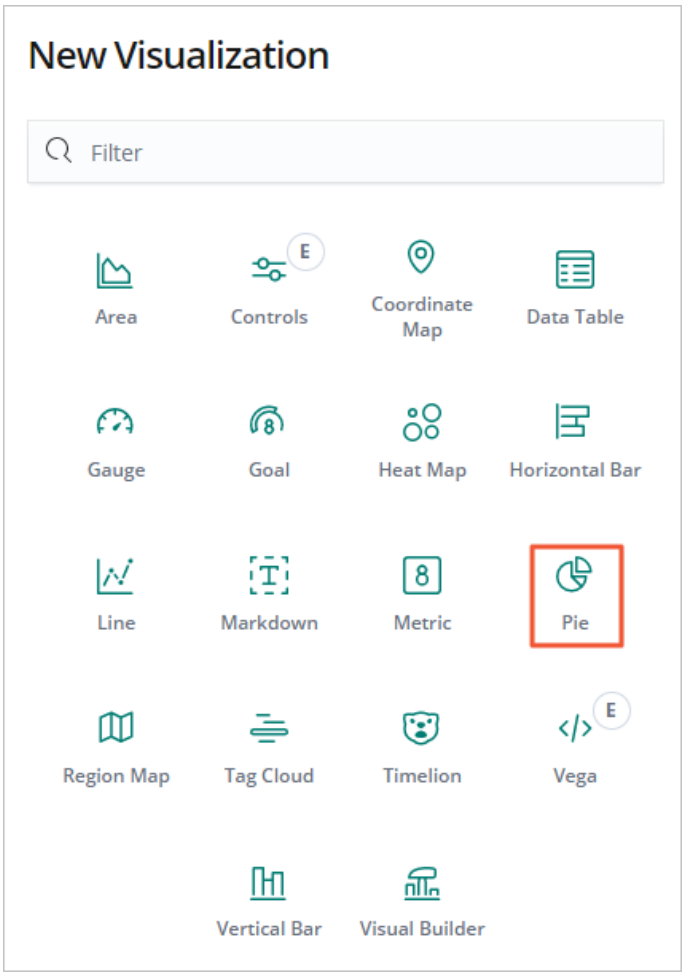


4. 统计slowlog数量最多的前10个key，并以降序排列展示。

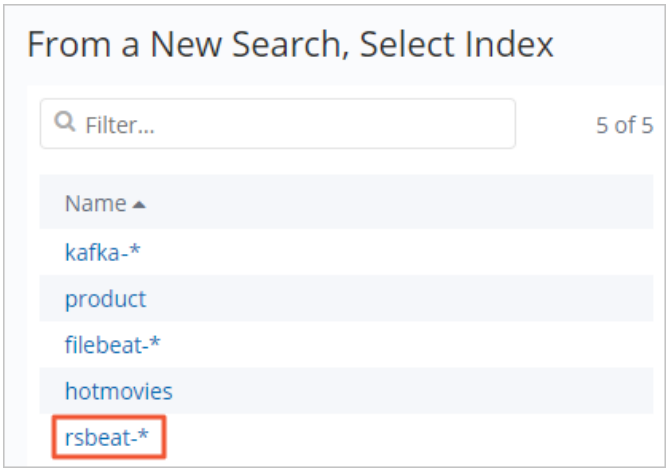
i. 在左侧导航栏，单击**Visualize**。

ii. 在**Visualize**页面，单击 + 图标。

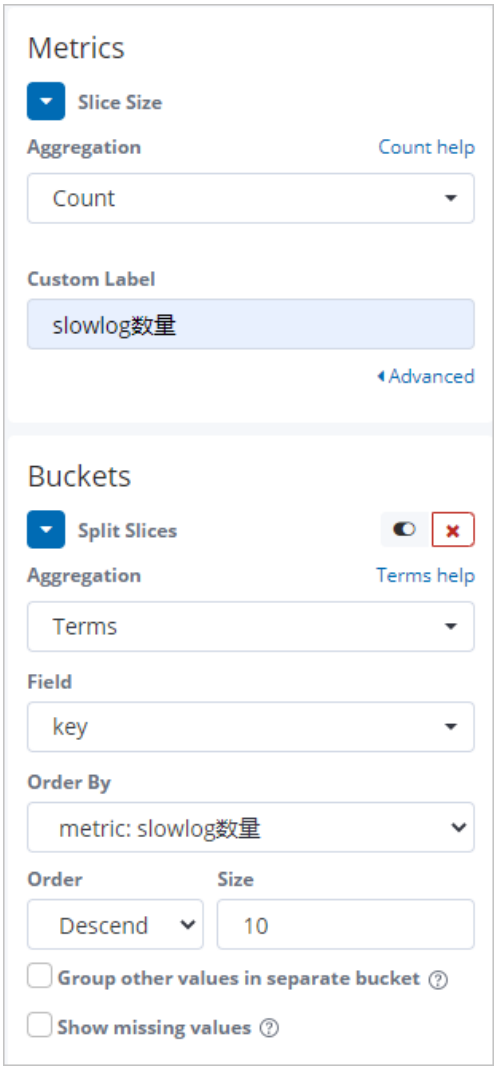
iii. 在**New Visualization**对话框中，单击**Pie**。



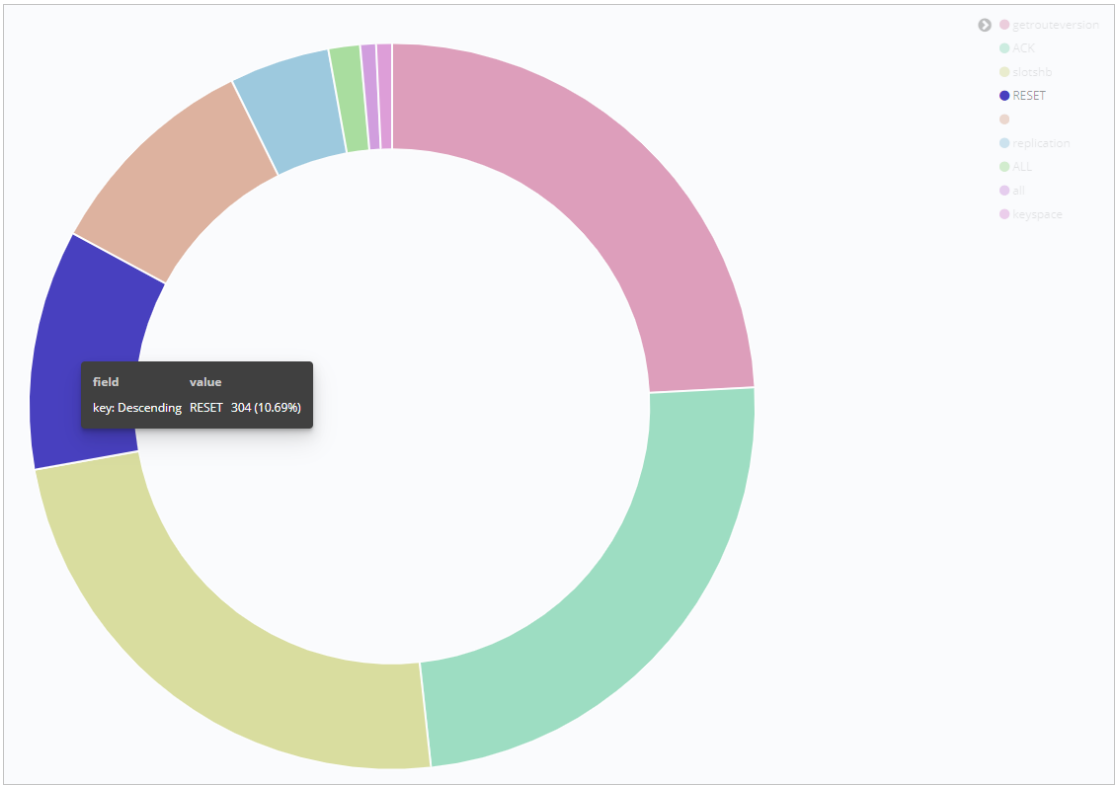
iv. 选择索引模式rsbeat-*。



v. 按照下图配置Metrics和Buckets。



vi. 单击 图标，查看结果。



说明 更多Kibana的使用方法，请参见Kibana官方文档。

8.服务器数据采集

8.1. 服务器数据采集方案概述

应用系统在提供服务过程中，会产生日志数据、系统指标数据、审计框架数据、检测状态数据以及各类APM监控数据。针对这些数据，可以根据业务的需求和环境，选择对应的方案进行数据采集并传输到Elasticsearch服务。本文对服务器数据采集方案进行了汇总。

相关文档	方案描述
阿里云Elasticsearch数据采集解决方案	本文提供Beats、Logstash、语言客户端和Kibana开发工具四种方法，您可以根据需求和环境，选择合适的方法或工具来采集数据。
通过Filebeat采集Apache日志数据	您可以使用Filebeat采集日志数据，并通过Logstash过滤采集的日志数据，最后传输到Elasticsearch中进行分析。
通过Metricbeat收集系统数据及Nginx服务数据	您可以通过阿里云Metricbeat采集器收集系统数据和Nginx服务数据，并生成可视化图表。
通过Auditbeat收集系统审计数据并监控文件更改	您可以通过阿里云Auditbeat收集Linux系统的审计框架数据，监控系统文件的更改情况，并生成可视化图表。
通过Heartbeat检测ICMP及HTTP服务	您可以通过阿里云Heartbeat检测ICMP及HTTP服务的状态，并生成可视化图表。
通过自建Metricbeat收集系统指标信息	您可以使用Metricbeat采集对应机器的指标信息，推送到阿里云Elaticsearch上，然后通过Kibana进行搜索分析，生成对应的图表。
使用SkyWalking和Elasticsearch实现全链路监控	您可以通过SkyWalking APM工具，将采集的监控数据存储到阿里云Elasticsearch中，然后通过Kibana查看并分析相应的全链路监控数据。
通过Uptime实时监控阿里云Elasticsearch服务	您可以通过Heartbeat检测HTTP/HTTPS、TCP、ICMP服务的状态，将采集的检测数据输出到Kibana的Uptime应用中，在业务受到影响前检测出问题。

8.2. 阿里云Elasticsearch数据采集解决方案

本文提供了将数据采集到阿里云Elasticsearch服务中的几种解决方案。

背景信息

对于数据搜索和分析来说，Elasticsearch无处不在。开发人员和社区可使用Elasticsearch寻找各种各样的使用场景，从应用程序搜索和网站搜索，到日志、基础架构监测、APM和安全分析，不一而足。虽然现在有针对这些场景的免费解决方案，但是开发人员仍需要先将其数据提供给Elasticsearch。

本文提供了以下几种常见的将数据采集到阿里云Elasticsearch中的方法：

- Elastic Beats
- Logstash
- 语言客户端
- Kibana开发工具

Elasticsearch提供了灵活的RESTful API，用于与客户端应用程序进行通信。因此通过调用RESTful API，可以完成数据采集、搜索和分析，以及管理集群及其索引的操作。

Elastic Beats

Elastic Beats是一组轻量型的数据采集器，可以方便地将数据发送给Elasticsearch服务。由于是轻量型的，Beats不会产生太多的运行开销，因此，可以在硬件资源有限的设备（如 IoT 设备、边缘设备或嵌入式设备）上运行和收集数据。如果您需要收集数据，但没有资源来运行资源密集型数据收集器，那么Beats会是您最佳的选择。这种无处不在（涵盖所有联网设备）的数据收集方式，能够让您快速检测到异常情况并做出反应，例如系统范围内的问题和安全事件等。

当然，Beats并不局限于资源有限的系统，它们还可用于具有更多可用硬件资源的系统。

Beats有多种风格，可以收集不同类型的数据：

- Filebeat

  Filebeat支持从文件形式的数据源中读取、预处理和传输数据。虽然大多数用户使用Filebeat来读取日志文件，但它也支持非二进制文件格式。Filebeat还支持多种其他数据源，包括TCP/UDP、容器、Redis和Syslog。借助丰富的模块，可以轻松收集Apache、MySQL和Kafka等常见应用程序的日志，并且能够根据日志格式解析出对应的数据。

- Metricbeat

  Metricbeat可以收集并预处理系统和服务指标数据。系统指标包括运行中的进程的相关信息，以及CPU、内存、磁盘、网络利用率等方面的数据。借助丰富的模块，可以收集来自不同服务的数据，包括Kafka、Palo Alto Networks、Redis等。

- Packetbeat

  Packetbeat可以收集并预处理实时网络数据，从而支持应用程序监测、安全和网络性能分析。此外，Packetbeat还支持DHCP、DNS、HTTP、MongoDB、NFS和TLS协议。

- Winlogbeat

  Winlogbeat可以从Windows操作系统捕获事件日志，包括应用程序事件、硬件事件，以及安全和系统事件。

- Auditbeat

  Auditbeat可以检测对关键文件的更改，并从Linux的审计框架中收集事件，主要应用于安全分析场景中。

- Heartbeat

  Heartbeat可以使用探测来监测系统和服务的可用性，因此可以应用于很多场景中，例如基础架构监测和安全分析。同时支持ICMP、TCP和HTTP协议。

- Functionbeat

  Functionbeat可以从无服务器环境（如AWS Lambda）中收集日志和指标。

请参见通过自建Metricbeat收集系统指标信息学习Beats的使用方法，其他Beats的使用方法与此类似。

Logstash

Logstash是一个强大而灵活的工具，可以读取、处理和传送任何类型的数据，并且具有丰富的功能，但对设备资源的要求较高。目前Beats还不支持Logstash提供的一些丰富的功能，或者通过Beats执行成本太高，例如通过查找外部数据源来丰富文档。但是Logstash的硬件要求显著高于Beats，因此Logstash通常不应部署在低资源设备上，在Beats功能不足以满足特定应用场景要求的情况下，可使用Logstash进行代替。

一种常见的架构模式是将Beats和Logstash组合起来：使用Beats来收集数据，并使用Logstash来执行Beats无法执行的数据处理任务。

阿里云Elasticsearch提供了Logstash服务。阿里云Logstash作为服务器端的数据处理管道，提供了100%兼容开源的Logstash功能，能够动态地从多个来源采集数据、转换数据，并且将数据存储到所选择的位置。通过输入、过滤和输出插件，Logstash可以对任何类型的事件进行加工和转换。

Logstash通过事件处理管道来执行任务，其中每个管道至少包含以下各项中的一个：

- 输入

  从数据源读取数据。官方支持多种数据源，包括文件、http、imap、jdbc、kafka、syslog、tcp和udp。

- 过滤器

  以多种方式处理和丰富数据。在许多情况下，需要先将非结构化的日志行解析为更加结构化的格式。因此，除其他功能外，Logstash还在正则表达式的基础上，提供了解析CSV、JSON、键/值对、分隔的非结构化数据、复杂的非结构化数据的过滤器（grok过滤器）。Logstash还提供了更多的过滤器，通过执行DNS查找，添加关于IP地址的地理信息，或通过查找自定义目录或Elasticsearch索引来丰富数据。通过这些附加的过滤器，能够对数据进行各种转换，例如重命名、删除、复制数据字段和值（mutate过滤器）。

- 输出

  输出是Logstash处理管道的最后阶段，可以将解析后并加以丰富的数据写入数据接收器。虽然有很多输出插件可用，但本文主要讨论如何使用Elasticsearch输出，将数据采集到Elasticsearch服务中。

以下提供了一个示例Logstash管道，该管道能够：

- 读取Elastic博客RSS源。
- 通过复制或重命名字段、删除特殊字符以及HTML标记，来执行一些简单的数据预处理。
- 将文档采集到Elasticsearch。

1. 按照如下示例配置阿里云Logstash管道。

```
input {
  rss {
    url => "/blog/feed"
    interval => 120
  }
}
filter {
  mutate {
    rename => [ "message", "blog_html" ]
    copy => { "blog_html" => "blog_text" }
    copy => { "published" => "@timestamp" }
  }
  mutate {
    gsub => [
      "blog_text", "<.*?>", "",
      "blog_text", "[\n\t]", " "
    ]
    remove_field => [ "published", "author" ]
  }
}
output {
  stdout {
    codec => dots
  }
  elasticsearch {
    hosts => [ "https://<your-elsaticsearch-url>" ]
    index => "elastic_blog"
    user => "elastic"
    password => "<your-elasticsearch-password>"
  }
}
```

`hosts` 需要替换为 `<对应阿里云Elasticsearch实例的内网地址>:9200`；`password` 需要替换为对应阿里云Elasticsearch的访问密码。详细配置方法请参见步骤二：创建并运行管道任务。

2. 在Kibana控制台中，查看索引数据。

```
POST elastic_blog/_search
```

详细操作方法请参见查看数据同步结果。

语言客户端

通过Elasticsearch提供的客户端，您可以将数据采集与自定义应用程序代码集成。这些客户端是抽象出数据采集低层细节的库，使您能够专注于特定应用程序的实际工作。目前Elasticsearch支持多种客户端语言，包括Java、JavaScript、Go、.NET、PHP、Perl、Python和Ruby等，更多支持的客户端语言以及您所选语言的详细信息和代码示例，请参见官方支持的Elasticsearch客户端。

如果您的应用程序所使用的语言不在官方支持的客户端语言中，可以在社区贡献的客户端中查找相关文档。

Kibana开发工具

推荐您使用Kibana开发控制台开发并调试Elasticsearch请求。Kibana开发工具公开了通用的Elasticsearch RESTful API的全部功能，同时抽象出了底层HTTP请求的技术细节。您可以使用Kibana开发工具，将原始JSON文档添加到Elasticsearch中。

```
PUT my_first_index/_doc/1
{
    "title" :"How to Ingest Into Elasticsearch Service",
    "date" :"2019-08-15T14:12:12",
    "description" :"This is an overview article about the various ways to ingest into Elasti
csearch Service"
}
```

> **说明** 除了Kibana开发工具以外，您也可以使用其他工具，通过RESTful API与Elasticsearch通信并采集文档。例如curl是一款经常使用的工具，用于Elasticsearch请求的开发、调试或与自定义脚本集成。

总结

将数据采集到Elasticsearch服务的方法不胜枚举。您需要根据特定用例、需求和环境，选择合适的方法或工具来采集数据。

- Beats提供了一种方便、轻量级的开箱即用型解决方案，可以从许多不同的来源采集数据。与Beats封装在一起的模块为许多常见数据库、操作系统、容器环境、Web服务器、缓存等提供了数据获取、解析、索引和可视化的配置。通过这些模块，可以实现五分钟将数据制作成仪表板进行展示。因为Beats是轻量型的，所以非常适合资源受限的嵌入式设备，例如IoT设备或防火墙。
- Logstash是一种灵活的工具，可用于读取、转换和采集数据，提供了大量的过滤器、输入和输出插件。如果Beats的功能对于某些用例来说还不够，那么一种常见的架构模式是使用Beats来收集数据，并通过Logstash做进一步处理，然后再采集到Elasticsearch中。
- 当您需要直接从应用程序采集数据时，建议使用官方支持的客户端库。
- 当您需要对Elasticsearch请求进行开发或调试时，建议您使用Kibana开发工具。

相关文档

- 如何将数据采集到 Elasticsearch服务
- 应该使用Logstash还是Elasticsearch采集节点数据
- 使用Beats系统模块将系统日志和指标获取到Elasticsearch

8.3. 通过自建Metricbeat收集系统指标信息

Metricbeat收集机器指标信息

当您需要查看并分析一台机器的指标信息时，可以使用Metricbeat采集该机器的指标信息，推送到阿里云Elasticsearch上。然后在Kibana中搜索分析，并生成对应的Dashborard。本文以Mac电脑为例，介绍具体的实现方法。

前提条件

您已完成以下操作：

- 创建阿里云Elasticsearch实例。

  具体操作，请参见创建阿里云Elasticsearch实例。

> **说明** 如果您需要通过内网地址来访问阿里云Elasticsearch实例，还需要购买一个与实例在相同地域和可用区，以及相同专有网络下的云服务器ECS实例。具体操作，请参见使用向导创建实例。

- 下载Metricbeat：
  - MAC系统的Metricbeat安装包下载地址。
  - 32位Linux系统的Metricbeat安装包下载地址。
  - 64位Linux系统的Metricbeat安装包下载地址。
  - 32位Windows系统的Metricbeat安装包下载地址。
  - 64位Windows系统的Metricbeat安装包下载地址。

背景信息

Beats是一个集合了多种单一用途的数据采集器的平台，这些采集器安装后可用作轻量型代理，从成百上千或成千上万台机器向Logstash或Elasticsearch实例发送数据。

Metricbeat是一个轻量级的指标采集器，可以从系统和服务中收集指标。从CPU到内存，从Redis到Nginx，Metricbeat能够以一种轻量型的方式采集各种系统和服务的统计数据。

操作流程

1. 配置阿里云Elasticsearch
2. 配置Metricbeat
3. 在Kibana中查看Dashboard

> **说明** 您也可以参考本案例的操作流程，使用Metricbeat采集一台Linux系统或Windows系统电脑的指标信息，并推送到阿里云Elasticsearch上。

配置阿里云Elasticsearch

1. 登录阿里云Elasticsearch控制台。
2. 在左侧导航栏，单击**Elasticsearch实例**。
3. 进入目标实例。
   i. 在顶部菜单栏处，选择资源组和地域。
   ii. 在左侧导航栏，单击**Elasticsearch实例**，然后在**Elasticsearch实例**中单击目标实例ID。
4. 在左侧导航栏，单击**安全配置**。
5. 打开**公网地址**开关，待配置生效后，单击**公网地址访问白名单**右侧的**修改**，将您MAC机器对外的公网IP地址配置到公网地址访问白名单中。

注意 如果您使用的是WIFI等网络，需要将公网出口的跳板机IP地址配置进去。如果获取不到，建议配置 `0.0.0.0/1,128.0.0.0/1` 来开放尽可能多的IP地址（本篇以此为例）。需要注意这个配置将导致您的阿里云Elasticsearch服务完全暴露在公网中，需要先评估下是否可以接受这个风险。

6. 在左侧导航栏，单击**基本信息**。在**基本信息**区域，获取Elasticsearch实例的**公网地址**备用。

7. 在左侧导航栏，单击**ES集群配置**。在**YML文件配置**区域，单击右侧的**修改配置**，将**自动创建索引**设置为**允许自动创建索引**。

警告 此配置需要重启Elasticsearch实例才能生效，为保证您的业务不受影响，请确认后再进行后续操作。

8. 勾选**该操作会重启实例，请确认后操作**，单击**确定**。

   重启过程中，可在任务列表查看重启进度。重启完成后，即可完成实例的配置。

配置Metricbeat

1. 将您下载的Metricbeat安装包解压缩，并进入Metricbeat文件夹。

```
MacBook-Pro:Desktop $ cd metricbeat-6.3.1-darwin-x86_64
MacBook-Pro:metricbeat-6.3.1-darwin-x86_64 $ ls
LICENSE.txt                 data                        logs                        metricbeat.yml
NOTICE.txt                  fields.yml                  metricbeat                  modules.d
README.md                   kibana                      metricbeat.reference.yml
deMacBook-Pro:metricbeat-6.3.1-darwin-x86_64 $
```

2. 打开*metricbeat.yml*文件，定位到 `Elasticsearch output` 部分，取消对应内容的注释。

```
#================================ Outputs =====================================

# Configure what output to use when sending the data collected by the beat.

#-------------------------- Elasticsearch output ------------------------------
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["es-cn-[redacted].public.elasticsearch.aliyuncs.com:9200"]

  # Optional protocol and basic auth credentials.
  protocol: "http"
  username: "[redacted]:"
  password: "[redacted]"
```

参数	说明
hosts	Elasticsearch实例的公网或内网地址（本文以公网地址为例）。
protocol	需要配置为http。
username	默认是elastic。
password	对应用户的密码。elastic用户的密码在创建实例时设定，如果忘记可重置。重置密码的注意事项和操作步骤，请参见重置实例访问密码。

3. 执行以下命令，启动Metricbeat。

```
./metricbeat -e -c metricbeat.yml
```

```
deMacBook-Pro:metricbeat-6.3.1-darwin-x86_64 $ ./metricbeat -e -c metricbeat.yml
```

启动成功后，Metricbeat开始向Elasticsearch实例推送数据。

在Kibana中查看Dashboard

1. 登录目标Elasticsearch实例的Kibana控制台。

   具体操作，请参见登录Kibana控制台。

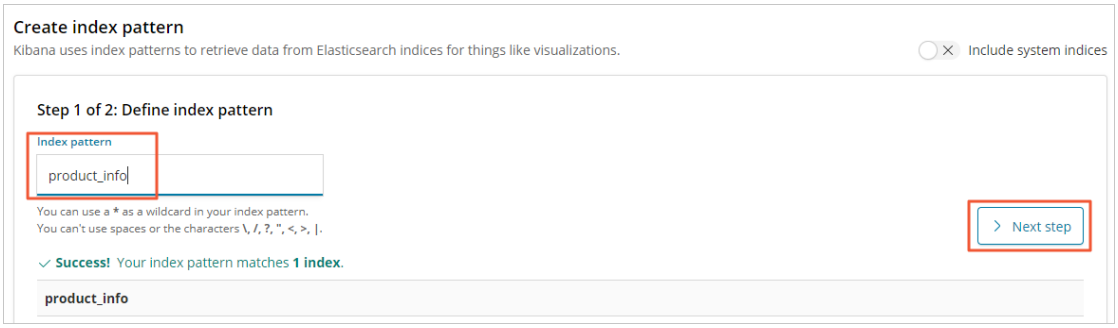
2. （可选）在左侧导航栏，单击**Management**，按照以下步骤创建一个索引模式。

   > **注意** 如果已经创建了索引模式，可忽略此步骤。

   i. 在**Management**页面，单击**Kibana**区域中的**Index Patterns**。

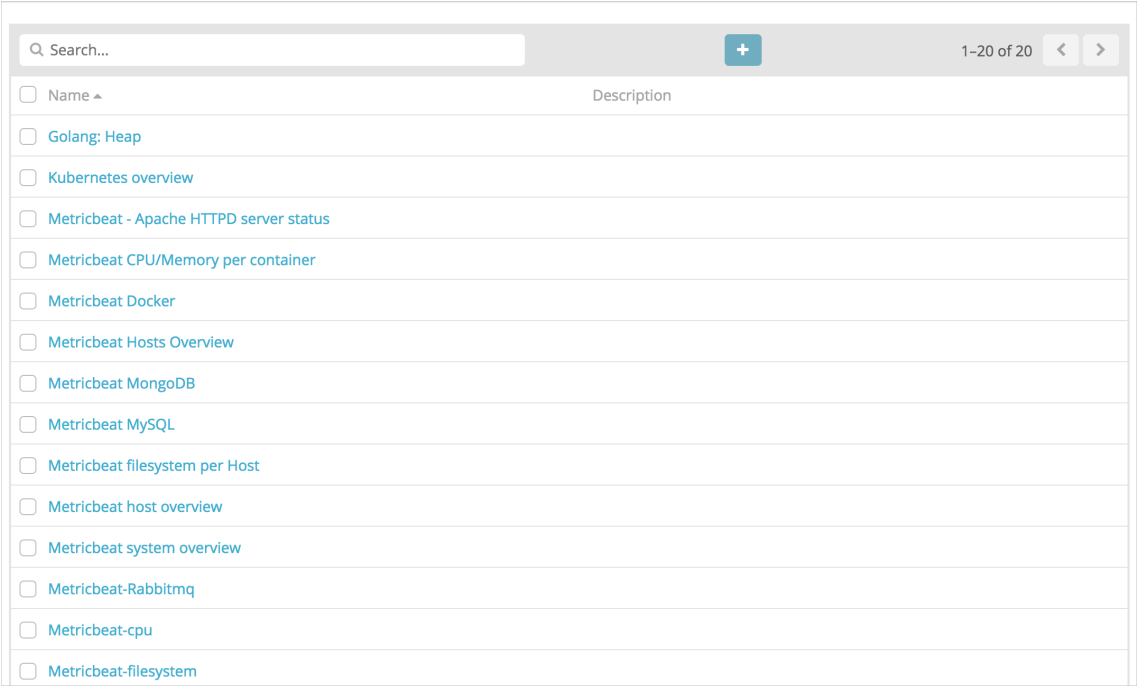
   ii. 在**Create index pattern**页面，输入索引模式名称（待查询的索引名称）。

iii. 单击Next step。

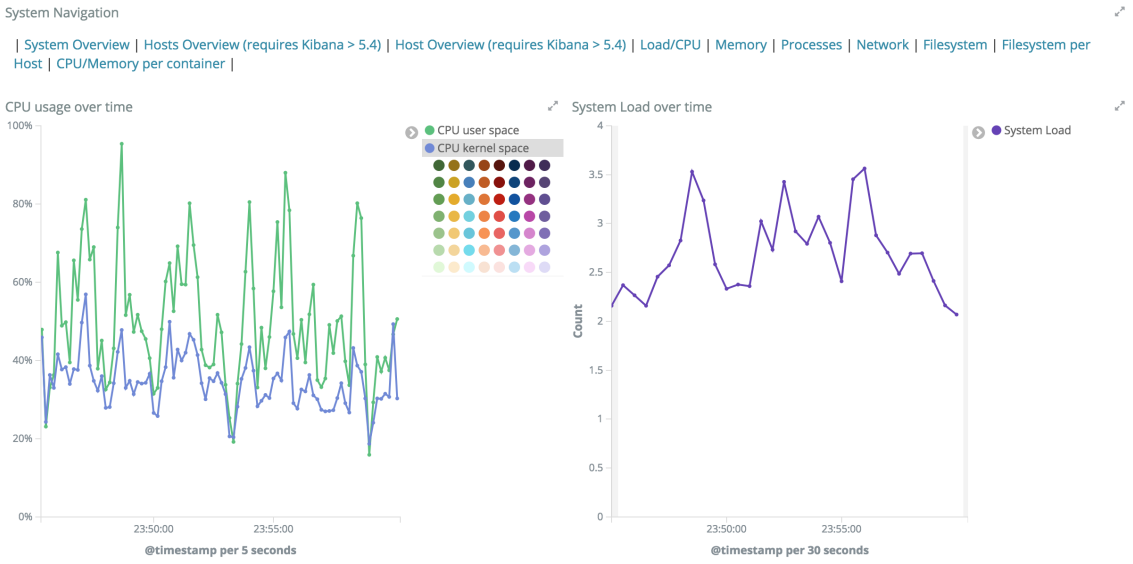


iv. 单击Create index pattern。

3. 在左侧导航栏，单击Dashboard。
4. 在Dashboard页面查看相关信息。
   - 各类相关指标列表



   - 单击Metricbeat-cpu，查看CPU指标信息



> **说明** 您可以将数据定义为5s刷新一次，然后生成对应的报表，并接入WebHook进行异常告警。

相关文档

借助Beats快速搭建可视化运维系统

8.4. 使用SkyWalking和Elasticsearch实现全链路监控

SkyWalking和es全链路监控

SkyWalking是分布式的应用性能管理APM（Application Performance Monitoring）工具，也被称为分布式追踪系统。本文介绍使用阿里云Elasticsearch 7.4版本的实例与SkyWalking，实现对实例的全链路监控。

背景信息

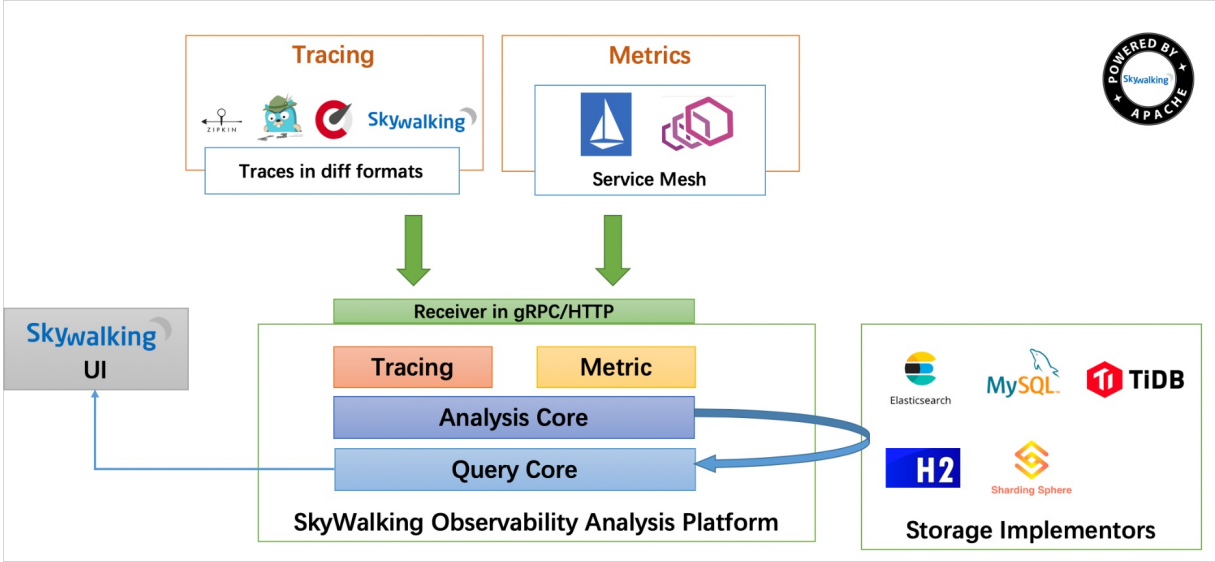
SkyWalking具有以下特性：

- 全自动探针监控，不需要修改应用程序代码。
- 手动探针监控，提供了支持OpenTracing标准的SDK。覆盖范围扩大到OpenTracing-Java支持的组件。

  > **说明** OpenTracing支持的组件请参见OpenTracing Registry。

- 自动监控和手动监控可以同时使用，使用手动监控弥补自动监控不支持的组件，甚至私有化组件。
- 纯Java后端分析程序，提供RESTful服务，可为其他语言探针提供分析能力。
- 高性能纯流式分析。

SkyWalking的架构图如下。



SkyWalking的核心在于数据分析和度量结果的存储平台部分，通过HTTP或gRPC方式向SkyWalking Collector提交分析和度量数据。SkyWalking Collector对数据进行分析和聚合，存储到Elasticsearch、H2、MySQL、TiDB等其一即可，最后通过SkyWalking UI的可视化界面查看分析结果。Skywalking支持从多个来源和多种格式收集数据，支持多种语言的Skywalking Agent 、Zipkin v1/v2 、Istio勘测、Envoy度量等数据格式。

> **说明** 本文介绍SkyWalking与阿里云Elasticsearch 7.4版本的集成配置，您也可以通过Skywalking客户端上报Java应用数据，详细信息，请参见通过SkyWalking上报Java应用数据。SkyWalking支持的中间件和组件，请参见SkyWalking官方文档。

前提条件

您已完成以下操作：

- 创建阿里云Elasticsearch实例，本文使用7.4.0版本。

  具体操作步骤，请参见创建阿里云Elasticsearch实例。
- 准备一台Linux服务器，并在服务器中安装JDK，要求JDK版本为1.8.0及以上版本。

  建议您使用阿里云ECS服务器。购买ECS服务器的方法，请参见步骤一：创建ECS实例。

  > **说明** 安装JDK的方式，请参见步骤三：安装JDK。如果未正确安装JDK，启动SkyWalking后查看日志，可能会显示Java not found或者java-xxx: No such file or directory报错。
- 确保Linux服务器的8080、10800、11800、12800端口不被占用。
- 关闭Linux服务器的防火墙及SELinux。

操作流程

1. 步骤一：下载并安装SkyWalking
2. 步骤二：配置SkyWalking与Elasticsearch连通
3. 步骤三：验证结果

步骤一：下载并安装SkyWalking

1. 在Linux服务器中，下载SkyWalking。

   建议选择最新的7.0.0版本。由于本文使用的是Elasticsearch 7.4.0版本，因此选择Binary Distribution for ElasticSearch 7二进制包。下载命令如下。

   ```
   wget https://archive.apache.org/dist/skywalking/7.0.0/apache-skywalking-apm-es7-7.0.0.tar.gz
   ```

2. 解压。

   ```
   tar -zxvf apache-skywalking-apm-es7-7.0.0.tar.gz
   ```

3. 查看解压后的文件。

   ```
   ll apache-skywalking-apm-bin-es7/
   ```

   返回结果如下。

   ```
   total 92
   drwxrwxr-x 8 1001 1002   143 Mar 18 23:50 agent
   drwxr-xr-x 2 root root   241 Apr 10 16:03 bin
   drwxr-xr-x 2 root root   221 Apr 10 16:03 config
   -rwxrwxr-x 1 1001 1002 29791 Mar 18 23:37 LICENSE
   drwxrwxr-x 3 1001 1002  4096 Apr 10 16:03 licenses
   -rwxrwxr-x 1 1001 1002 32838 Mar 18 23:37 NOTICE
   drwxrwxr-x 2 1001 1002 12288 Mar 19 00:00 oap-libs
   -rw-rw-r-- 1 1001 1002  1978 Mar 18 23:37 README.txt
   drwxr-xr-x 3 root root    30 Apr 10 16:03 tools
   drwxr-xr-x 2 root root    53 Apr 10 16:03 webapp
   ```

步骤二：配置SkyWalking与Elasticsearch连通

1. 在config目录下，打开*application.yml*文件。

   ```
   cd apache-skywalking-apm-bin-es7/config/
   vi application.yml
   ```

2. 定位到 `storage` 部分，将默认的H2存储库改为elasticsearch7，并按照以下说明配置。

   ```
   storage:
     selector: ${SW_STORAGE:elasticsearch7}
     elasticsearch7:
       nameSpace: ${SW_NAMESPACE:"skywalking-index"}
       clusterNodes: ${SW_STORAGE_ES_CLUSTER_NODES:es-cn-4591kzdzk000i****.public.elasticsearch.aliyuncs.com:9200}
       protocol: ${SW_STORAGE_ES_HTTP_PROTOCOL:"http"}
      # trustStorePath: ${SW_SW_STORAGE_ES_SSL_JKS_PATH:"../es_keystore.jks"}
      # trustStorePass: ${SW_SW_STORAGE_ES_SSL_JKS_PASS:""}
       enablePackedDownsampling: ${SW_STORAGE_ENABLE_PACKED_DOWNSAMPLING:true} # Hour and Day metrics will be merged into minute index.
       dayStep: ${SW_STORAGE_DAY_STEP:1} # Represent the number of days in the one minute/hour/day index.
       user: ${SW_ES_USER:"elastic"}
       password: ${SW_ES_PASSWORD:"es_password"}
   ```

> **说明** SkyWalking服务默认使用H2存储，不具有持久存储的特性，所以需要将存储组件修改为elasticsearch。

参数	说明
selector	存储选择器。本文设置为elasticsearch7。
nameSpace	命名空间。Elasticsearch实例中，所有索引的命名会使用此参数值作为前缀。
clusterNodes	指定Elasticsearch实例的访问地址。由于实例与SkyWalking不在同一专有网络VPC（Virtual Private Cloud）下，因此要使用公网访问地址，获取方式请参见查看实例的基本信息。
user	Elasticsearch实例的访问用户名，默认为elastic。
password	对应用户的密码。elastic用户的密码在创建实例时指定，如果忘记可重置。重置密码的注意事项和操作步骤，请参见重置实例访问密码。

> **注意** 配置中仅指定用户名和密码即可，请注释trustStorePath和trustStorePass，否则会报错NoSuchFileException:../es_keystore.jks。

3. （可选）修改监听的IP地址或端口号。

   SkyWalking默认使用12800作为Rest API通信端口，11800为gRPC API端口，可在*application.yml*文件的core中修改，本文使用默认配置。

```
core:
  selector: ${SW_CORE:default}
  default:
    # Mixed: Receive agent data, Level 1 aggregate, Level 2 aggregate
    # Receiver: Receive agent data, Level 1 aggregate
    # Aggregator: Level 2 aggregate
    role: ${SW_CORE_ROLE:Mixed} # Mixed/Receiver/Aggregator
    restHost: ${SW_CORE_REST_HOST:0.0.0.0}
    restPort: ${SW_CORE_REST_PORT:12800}
    restContextPath: ${SW_CORE_REST_CONTEXT_PATH:/}
    gRPCHost: ${SW_CORE_GRPC_HOST:0.0.0.0}
    gRPCPort: ${SW_CORE_GRPC_PORT:11800}
```

4. （可选）在webapp目录下，修改*webapp.yml*配置。

   本文使用默认配置，您也可以根据具体需求修改。

```
server:
  port: 8080
collector:
  path: /graphql
  ribbon:
    ReadTimeout: 10000
    # Point to all backend's restHost:restPort, split by ,
    listOfServers: 127.0.0.1:12800
```

步骤三：验证结果

1. 在Linux服务器中，启动SkyWalking。

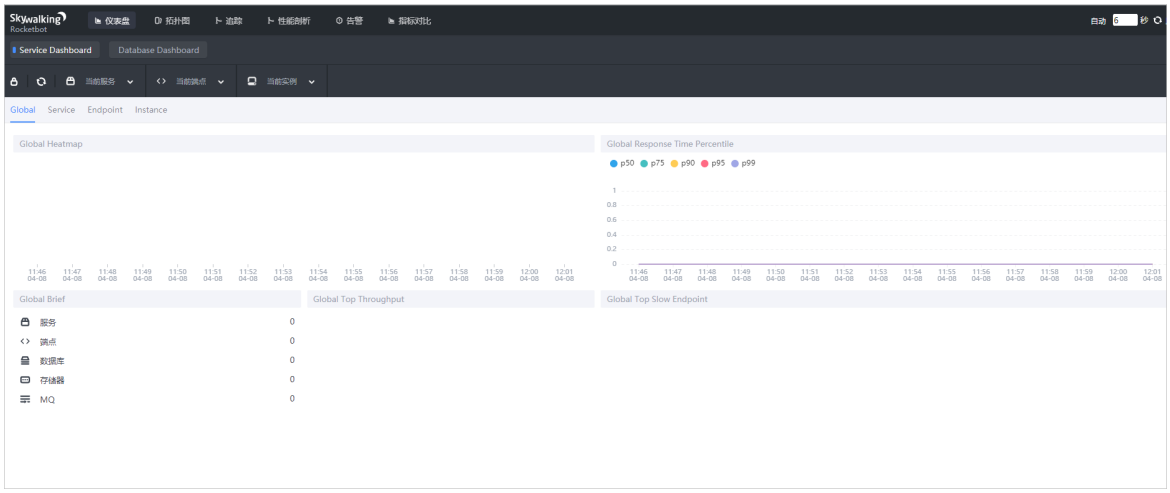
```
cd ../bin
./startup.sh
```

> **注意**
> - 在启动SkyWalking前，请确保Elasticsearch实例为正常状态。
> - 执行 `./startup.sh` 命令，会同时启动Collector和UI。

启动成功后，返回如下结果。

```
SkyWalking OAP started successfully!
SkyWalking Web Application started successfully!
```

2. 在浏览器中，访问http://<Linux服务器的IP地址>:8080/。



> **说明** 初次使用SkyWalking连接Elasticsearch服务，启动会比较慢。因为SkyWalking需要向Elasticsearch服务创建大量的index，所以在未创建完成之前，访问这个页面会显示空白。此时您可以通过查看日志来判断启动是否完成，日志路径为 `<SkyWalking的安装路径>logs/skywalking-oap-server.log` 。

3. 参见登录Kibana控制台，登录对应Elasticsearch实例的Kibana控制台，执行 `GET _cat/indices?v` 命令查看索引数据。
根据返回结果，可以看到Elasticsearch实例中包含了大量以 `skywalking-index` 开头的索引。

```
green  open   skywalking-index_all_percentile-20200408
green  open   skywalking-index_endpoint_inventory
green  open   skywalking-index_service_apdex-20200408
green  open   skywalking-index_database_access_resp_time_month-202004
green  open   skywalking-index_service_relation_client_cpm-20200408
green  open   skywalking-index_database_access_sla_month-202004
green  open   skywalking-index_service_relation_server_call_sla-20200408
green  open   skywalking-index_endpoint_sla-20200408
green  open   skywalking-index_instance_jvm_memory_noheap-20200408
green  open   skywalking-index_service_relation_server_percentile-20200408
green  open   skywalking-index_service_instance_relation_server_resp_time-20200408
green  open   skywalking-index_profile_task_segment_snapshot-20200408
green  open   skywalking-index_endpoint_relation_cpm-20200408
green  open   skywalking-index_instance_jvm_old_gc_time-20200408
green  open   skywalking-index_service_sla-20200408
green  open   skywalking-index_top_n_database_statement-20200408
green  open   skywalking-index_service_relation_client_call_sla-20200408
green  open   skywalking-index_endpoint_percentile_month-202004
green  open   apm-agent-configuration
```

8.5. 通过Uptime实时监控阿里云Elasticsearch服务

Heartbeat支持通过HTTP/HTTPS、TCP和ICMP服务，定期检测网络端点状态，并将采集的检测数据，输出到Kibana的Uptime应用中，实时监控应用程序及服务的可用性和响应时间，在业务受到影响前检测出问题。本文介绍如何通过Uptime实时监控阿里云Elasticsearch服务。

背景信息

Uptime需要与以下服务结合使用：

- Heartbeat
- 阿里云Elasticsearch
- Kibana

> **说明** 您还可以通过Kibana 7.7的Alerting and Actions实现监控报警通知。

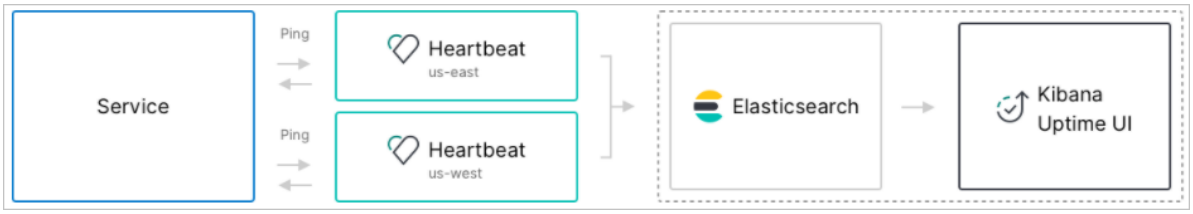
部署架构

- 单实例部署

  单个Heartbeat实例部署在单个监控位置，监控单个服务。Heartbeat发送监控数据给阿里云Elasticsearch，与此同时，可以使用Kibana Uptime查看心跳数据并确定服务状态。

- 多实例部署

两个Heartbeat部署在不同的监控位置，监控同一个服务。Heartbeat发送监控数据给阿里云Elasticsearch，与此同时，可以使用Kibana Uptime查看心跳数据并确定服务状态。当某个区域的Heartbeat发生故障，多个监视位置可以帮助您定位Heartbeat故障的区域。



更多部署架构，请参见Deployment Architecture。

准备工作

1. 创建阿里云Elasticsearch实例，并开启自动创建索引功能。

   具体操作，请参见创建阿里云Elasticsearch实例和配置YML参数。

2. 创建ECS实例，作为Heartbeat的部署机器。要求该ECS实例与阿里云Elasticsearch实例处于同一专有网络下。

   具体操作，请参见使用向导创建实例。

   > 注意 在创建ECS实例时，需要选择Aliyun Linux、RedHat和CentOS这三种操作系统，因为Beats（包含Heartbeat）目前仅支持这三种操作系统。

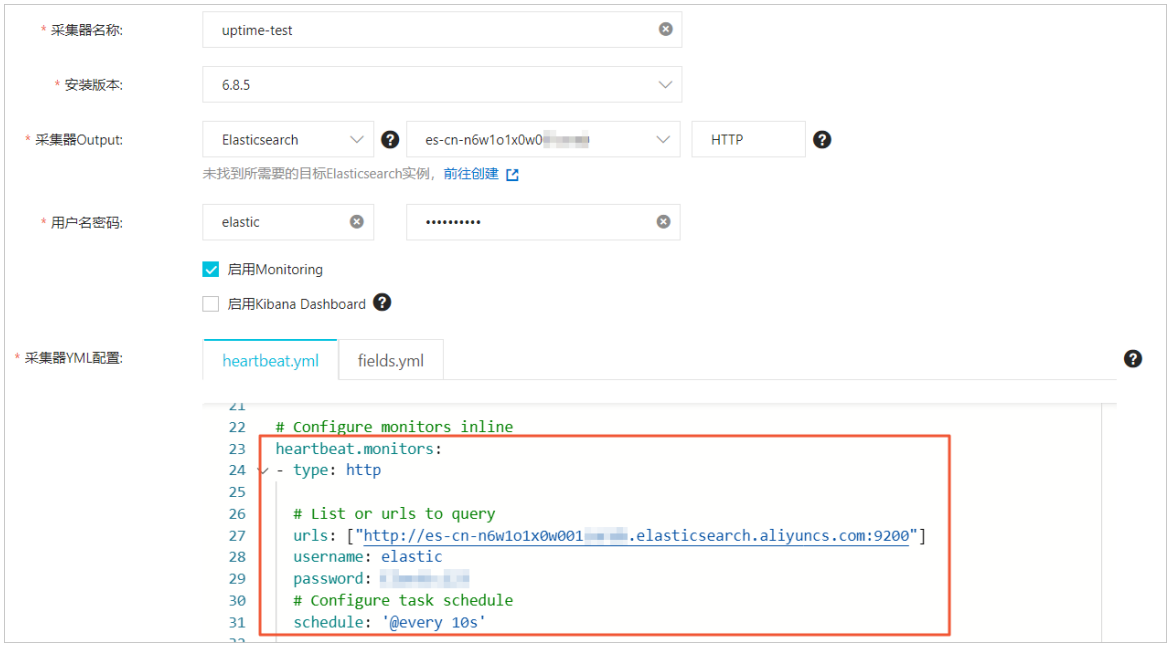
3. 在ECS实例上安装云助手和Docker服务。

   具体操作，请参见安装云助手客户端和部署并使用Docker（Alibaba Cloud Linux 2）。

创建Heartbeat采集器

1. 登录阿里云Elasticsearch控制台。
2. 在**创建采集器**区域，单击**Heartbeat**。
3. 安装并配置采集器。

   具体操作，请参见采集ECS服务日志和采集器YML配置。



heartbeat.monitors配置说明

参数	说明
type	本文指定为http。 **说明** Heartbeat支持检查HTTP/HTTPS、TCP和ICMP服务。例如使用HTTP/HTTPS监视器，可以检查响应代码（code）、正文（body）和头信息（header）； 使用TCP监视器，可以检查端口和字符串。
urls	待检查的URL列表，可以指定多个HTTP服务。本文以检查阿里云Elasticsearch服务为例，此处需要配置为待检查实例的私网访问地址。
schedule	检查间隔。@every 10s表示每10s检查一次。

4. 单击**下一步**。
5. 在**采集器安装**配置向导中，选择安装采集器的ECS实例。

6. 启动采集器并查看采集器安装情况。

   具体操作，请参见采集ECS服务日志。

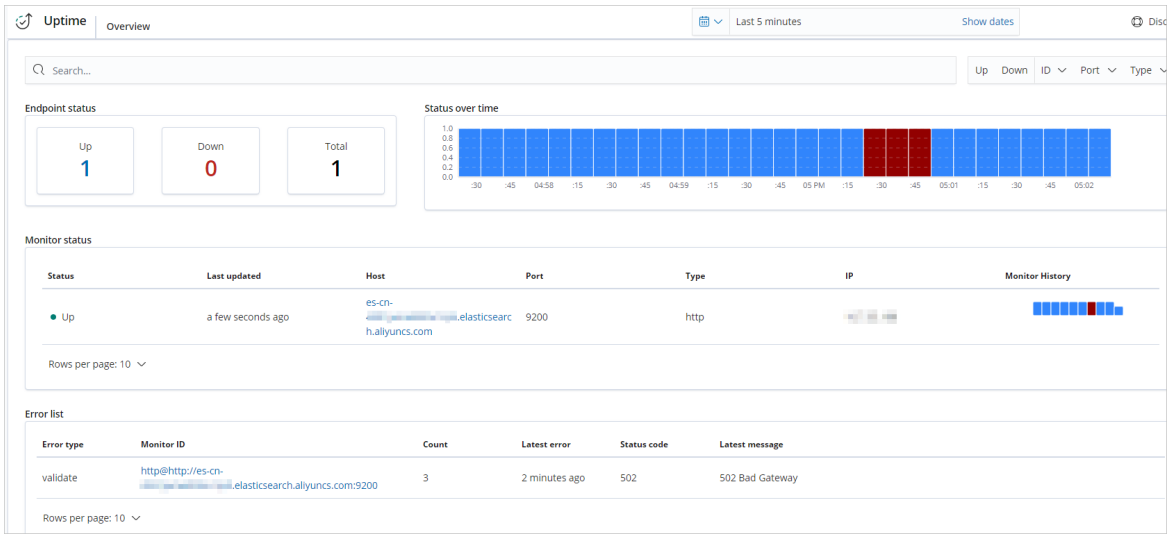
   当**采集器状态**为**已生效**，且**采集器安装情况**显示为**心跳正常**时，说明采集器安装成功。

查看Uptime监控信息

1. 登录Kibana控制台。

   此Kibana控制台为：创建采集器时，**采集器Output**指定的阿里云Elasticsearch实例对应的Kibana控制台。具体操作，请参见登录Kibana控制台。

2. 在左侧导航栏，单击**Uptime**，查看监控数据。



   - 红色：异常状态，请检查Heartbeat通信或阿里云Elasticsearch状态。
   - 蓝色：正常状态。

9.集群管理

9.1. 集群管理概述

在使用阿里云Elasticsearch集群时，针对不同的使用场景会涉及到对集群不同的管理方式。本文对集群管理方式以及功能描述进行了汇总，以便帮您快速了解各功能。

类别	相关文档	功能描述
冷热分离与生命周期管理	通过索引生命周期管理Heartbeat数据	对于时间序列数据，会随着时间的积累越来越大，您可以通过索引生命周期管理ILM（Index Lifecycle Management）定期将数据滚动到新索引，防止因数据过大影响查询效率和成本。随着索引的老化和查询频率的降低，您可以将其转移到价格较低的磁盘上，并减少分片和副本的数量。
	通过索引生命周期管理实现冷热数据分离	冷热集群是指在集群中包含冷、热两种属性的节点，可以提高Elasticsearch的处理性能和服务稳定性。 在使用阿里云Elasticsearch集群时，您可以通过使用生命周期管理ILM（Index Lifecycle Management）功能，实现冷热数据分离目标。该功能可以为您实现在保证集群读写性能的基础上，自动维护集群上的冷热数据，又能通过优化集群架构，降低企业生产成本。
X-Pack高级特性应用	使用跨集群复制功能迁移数据	通过跨集群复制功能，您可以将本地Elasticsearch集群中的索引数据迁移到一个远程集群中，或者将一个远程集群中的索引数据迁移到本地集群，实现集群高可用及容灾备份，或跨地域数据的就近访问。
	X-Pack集成LDAP认证最佳实践	在使用阿里云Elasticsearch集群时，您可以通过配置轻量目录访问协议LDAP（Lightweight Directory Access Protocol）认证，实现相应角色的LDAP用户访问目标阿里云Elasticsearch实例。
	通过Elasticsearch X-Pack角色管理实现用户权限管控	当您需要设置集群、索引、字段或其他操作的访问权限时，可以通过Elasticsearch X-Pack的RBAC（Role-based Access Control）机制，为自定义角色分配权限，并将角色分配给用户，实现权限管控。Elasticsearch提供了多种内置角色，您可以在内置角色的基础上扩展自定义角色，以满足特定需求。
	配置Active Directory身份认证	基于阿里云Elasticsearch配置活动目录AD（Active Directory）身份认证，以实现AD域下相应角色的用户访问阿里云Elasticsearch。
	通过Elastic实现Kubernetes容器全观测	Elastic可观测性是通过Kibana可视化能力，将日志、指标及APM数据结合在一起，实现对容器数据的观测和分析。当您的应用程序以Pods方式部署在Kubernetes中，可以在Kibana中查看Pods生成日志、主机和网络上的事件指标及APM数据，逐步缩小排查范围进行故障排查。

典型全观测应用	相关文档	功能描述
	基于Indexing Service实现数据流管理	通过使用阿里云Elasticsearch 7.10日志增强版Indexing Service系列，可以为您实现云托管写入加速和按流量付费（即您无需按集群峰值写入吞吐预留资源），能够极低成本实现海量时序日志分析。
	通过OpenStore实现海量数据存储	OpenStore存储是阿里云Elasticsearch团队自研的针对日志场景的低成本、高效、弹性存储解决方案，能够为您在日志场景中提供海量存储服务。
数据管理与可视化	基于Terraform管理阿里云Elasticsearch最佳实践	通过Terraform，您可以使用代码配置实现物理机等资源的分配。即通过Terraform，写一个配置文件，就可以帮助您购买一台云服务器，或者申请到阿里云Elasticsearch、OSS等云资源。通过Terraform管理阿里云Elasticsearch的方法，包括创建、更新、查看、删除实例等操作。
	Curator操作指南	Curator是Elasticsearch官方提供的一个索引管理工具，该工具为您提供了删除、创建、关闭、段合并索引等功能。
	通过RollUp实现流量汇总最佳实践	对于时序数据场景，随着时间的积累数据量会越来越大。如果一直保留详细数据，会导致存储成本线性增长，此时您可以通过Elasticsearch的RollUp机制节省数据存储成本。
	使用DataV大屏展示阿里云Elasticsearch数据	通过在DataV中添加阿里云Elasticsearch数据源，您可以使用DataV访问阿里云Elasticsearch服务，完成数据的查询与展示。
	通过Cerebro访问阿里云ES	除了Kibana、curl命令、客户端等方式，您还可以通过Elasticsearch-Head、Cerebro等第三方插件或工具访问阿里云Elasticsearch实例。
集群报警通知	配置钉钉机器人接收X-Pack Watcher报警	通过为阿里云Elasticsearch添加X-Pack Watcher，可以实现当满足某些条件时执行某些操作。例如当logs索引中出现error日志时，触发系统自动发送报警邮件、钉钉消息或机器人消息。可以简单地理解为X-Pack Watcher是一个基于Elasticsearch实现的监控报警服务。
	配置企业微信机器人接收X-Pack Watcher报警	

9.2. 冷热分离与生命周期管理

9.2.1. 通过索引生命周期管理Heartbeat数据

es索引生命周期管理

对于时间序列数据，会随着时间的积累越来越大，您可以通过索引生命周期管理ILM（Index Lifecycle Management）定期将数据滚动到新索引，防止因数据过大影响查询效率和成本。随着索引的老化和查询频率的降低，您可以将其转移到价格较低的磁盘上，并减少分片和副本的数量。本文介绍通过ILM管理Heartbeat数据的方法。

背景信息

本文使用的测试场景如下：

业务场景中存在大量的heartbeat-*时序索引，并且每天单个索引大小都为4 MB左右。当数据越来越多时，shard数量也会越来越多，导致集群负载增加。所以需要指定不同的滚动更新策略，在hot阶段滚动更新heartbeat-*开头的历史监控索引，warm阶段对索引进行分片收缩及段合并，cold阶段将数据从hot节点迁移到warm（冷数据）节点，delete阶段定期删除索引数据。

注意事项

- 索引必须定义模板和别名后，才可以设置生命周期管理策略。
- 如果在滚动更新索引时修改了生命周期管理策略，新策略将在下一次滚动更新时生效。

操作流程

1. 步骤一：创建并配置冷热集群

   在创建集群时配置集群中节点的冷热属性、开启自动创建索引功能、配置公网地址访问白名单。

2. 步骤二：在Heartbeat下配置ILM

   在*heartbeat.yml*中开启阿里云Elasticsearch的生命周期管理功能，并配置其参数。配置完成并启动后，系统会自动在对应阿里云Elasticsearch实例中生成Heartbeat索引模板。

3. 步骤三：创建ILM策略

   通过ilm policy API创建生命周期管理策略，该策略用来定义索引滚动更新和归档的条件。

4. 步骤四：为ILM策略关联索引模板

   为上一步创建的生命周期管理策略关联Heartbeat索引模板。

5. 步骤五：为索引关联ILM策略

   为第一个Heartbeat索引关联生命周期管理策略，以便将该策略应用到整个Heartbeat索引模板覆盖的索引下。

6. 步骤六：查看各阶段索引

   查看归档在各阶段（hot、warm、cold、delete）的索引。

步骤一：创建并配置冷热集群

1. 创建冷热集群并查看集群的冷热节点属性。

   冷热集群是指在集群中包含冷、热两种属性的节点，可以提高Elasticsearch的处理性能和服务稳定性。两者区别如下。

节点类型	存储数据要求	读写性能要求	规格要求	存储要求
热节点（hot）	近期数据，例如最近2天的日志数据。	高	高，例如32核64 GB	建议使用SSD云盘存储数据，存储空间大小需根据数据大小进行设置。

节点类型	存储数据要求	读写性能要求	规格要求	存储要求
冷节点（warm）	历史数据，例如2天之前的日志数据。	低	低，例如8核32 GB	○ 建议使用高效云盘存储数据，存储空间大小需根据数据大小进行设置。 ○ 建议使用阿里云自研OpenStore存储功能，实现海量冷数据Serverless存储，详细信息请参见通过OpenStore实现海量数据存储。

i. 在购买阿里云Elasticsearch实例时，启用冷数据节点，即可创建冷热集群。

当您启用了冷数据节点并购买后，系统会在节点启动参数中加入-Enode.attr.box_type参数。

- 热数据节点：-Enode.attr.box_type=hot
- 冷数据节点：-Enode.attr.box_type=warm

> **说明**
>
> - 购买实例时，只有当启用了冷数据节点后，数据节点才会变成热节点。
> - 本文以阿里云Elasticsearch 6.7.0版本为例，所涉及的操作及图片仅适用于该版本，其他版本以实际界面为准。

ii. 登录该集群的Kibana控制台，在左侧导航栏，单击Dev Tools。

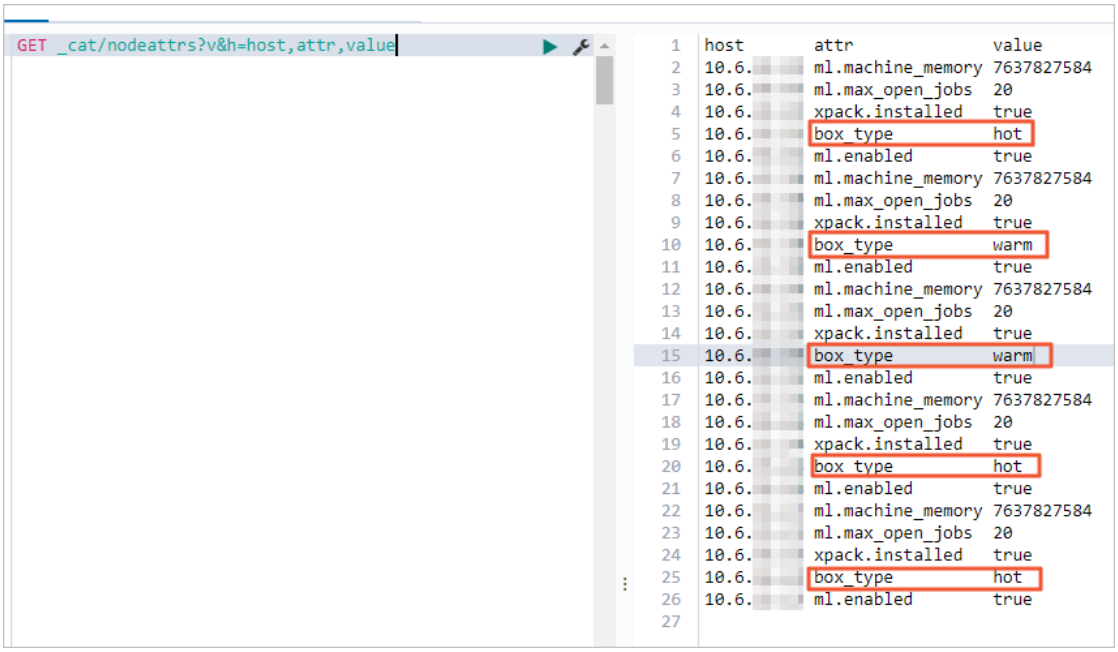
登录Kibana控制台的具体操作请参见登录Kibana控制台。

iii. 在左侧导航栏，单击Dev Tools。

iv. 在Console中，执行如下命令，查看集群冷热节点属性。

```
GET _cat/nodeattrs?v&h=host,attr,value
```

结果显示集群中包含3个hot节点，2个warm节点，支持冷热架构。



2. 开启目标集群的自动创建索引功能。

   具体操作步骤请参见配置YML参数。

3. 配置集群的公网地址访问白名单，将安装Heartbeat服务器的IP地址添加到白名单中。

   具体操作步骤请参见配置实例公网或私网访问白名单。

步骤二：在Heartbeat下配置ILM

为了使Heartbeat与Elasticsearch的ILM无缝衔接，可在heartbeat.yml配置中定义Elasticsearch的ILM，详细配置请参见Set up index lifecycle management。

1. 下载Heartbeat安装包，并解压缩。
2. 编辑heartbeat.yml，分别定义heartbeat.monitors、setup.template.settings、setup.kibana和output.elasticsearch。

   本文使用的配置如下。

```
heartbeat.monitors:
- type: icmp
  schedule: '*/5 * * * * * *'
  hosts: ["47.111.xx.xx"]
setup.template.settings:
  index.number_of_shards: 3
  index.codec: best_compression
  index.routing.allocation.require.box_type: "hot"
setup.kibana:
  # Kibana Host
  # Scheme and port can be left out and will be set to the default (http and 5601)
  # In case you specify and additional path, the scheme is required: http://localhost:56
01/path
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
  host: "https://es-cn-4591jumei00xxxxxx.kibana.elasticsearch.aliyuncs.com:5601"
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["es-cn-4591jumei00xxxxxx.elasticsearch.aliyuncs.com:9200"]
  ilm.enabled: true
  setup.template.overwrite: true
  ilm.rollover_alias: "heartbeat"
  ilm.pattern: "{now/d}-000001"
  # Enabled ilm (beta) to use index lifecycle management instead daily indices.
  #ilm.enabled: false
  # Optional protocol and basic auth credentials.
  #protocol: "https"
  username: "elastic"
  password: "<your_password>"
```

部分参数说明如下，更多参数说明请参见官方Heartbeat配置文档。

参数	说明
index.number_of_shards	设置主分片数，默认是1。
index.routing.allocation.require.box_type	设置将索引数据写入hot节点。
host	需要替换为您Kibana服务的公网访问地址，可在Kibana的配置页面获取。
hosts	需要替换为您Elasticsearch集群的公网或私网访问地址，可在集群的基本信息页面获取，详细信息请参见查看实例的基本信息。 **说明** 如果设置为公网地址，需要配置集群的公网地址访问白名单，具体操作请参见配置实例公网或私网访问白名单；如果设置为私网地址，需要确保集群与安装Heartbeat的服务器在同一专有网络下。
ilm.enabled	设置为true，表示启用索引生命周期管理ILM。

参数	说明
setup.template.overwrite	设置是否覆盖原索引模板。如果您已经将此版本的索引模板加载到Elasticsearch中，则必须将该参数设置为true，使用此版本的索引模板覆盖原索引模板。
ilm.rollover_alias	设置滚动更新索引时，生成的索引的别名，默认是heartbeat-\{beat.version\}。
ilm.pattern	设置滚动更新索引时，生成的索引的模式。支持date math，默认是{now/d}-000001。当触发索引滚动更新条件后，新的索引名称会在最后一位数字上加1。 例如第一次滚动更新产生的索引名称是heartbeat-2020.04.29-000001，当满足索引滚动更新条件后触发滚动，Elasticsearch会创建新的索引，名称为heartbeat-2020.04.29-000002。
username	用户名默认为elastic。
password	elastic用户的密码在创建实例时设定，如果忘记可重置。重置密码的注意事项和操作步骤，请参见重置实例访问密码。

> **注意** 如果在加载索引模板后修改ilm.rollover_alias或ilm.pattern，则必须设置setup.template.overwrite为true，重写索引模板。

3. 启动Heartbeat服务。

```
sudo ./heartbeat -e
```

步骤三：创建ILM策略

Elasticsearch支持通过API和Kibana控制台操作两种方式创建ILM策略。以下示例以API方式为例，介绍通过ilm policy API创建hearbeat-policy策略。

> **说明** Heartbeat支持通过 `./heartbeat setup --ilm-policy` 命令加载默认的策略并写入Elasticsearch，默认策略可通过 `./heartbeat export ilm-policy` 命令到stdout。您可以修改该默认策略，实现手动创建策略。

在Kibana控制台中，执行以下命令，创建ILM策略。

```
PUT /_ilm/policy/hearbeat-policy
{
  "policy": {
    "phases": {
      "hot": {
        "actions": {
          "rollover": {
            "max_size": "5mb",
            "max_age": "1d",
            "max_docs": 100
          }
        }
      },
      "warm": {
        "min_age": "60s",
        "actions": {
          "forcemerge": {
                "max_num_segments":1
              },
          "shrink": {
                "number_of_shards":1
              }
        }
      },
      "cold": {
        "min_age": "3m",
        "actions": {
          "allocate": {
            "require": {
              "box_type": "warm"
            }
          }
        }
      },
      "delete": {
        "min_age": "1h",
        "actions": {
          "delete": {}
        }
      }
    }
  }
}
```

参数	说明

参数	说明
hot	该策略设置索引只要满足其中任一条件：数据写入达到5 MB、使用超过1天、doc数超过100，就会触发索引滚动更新。此时系统将创建一个新索引，该索引将重新启动策略，而旧索引将在滚动更新后等待60秒进入warm阶段。 **注意** 目前Elasticsearch支持在rollover中配置三种归档策略：max_docs、max_size、max_age，满足其中任何一个条件都会触发索引归档操作。
warm	索引进入warm阶段后，ILM会将索引收缩到1个分片，强制合并为1个段。完成该操作后，索引将在3分钟（从滚动更新时算起）后进入cold阶段。
cold	索引进入cold阶段后，ILM将索引从hot节点移动到warm（冷数据）节点。完成操作后，索引将在1小时后进入delete阶段。
delete	索引进入delete阶段后被删除。

说明

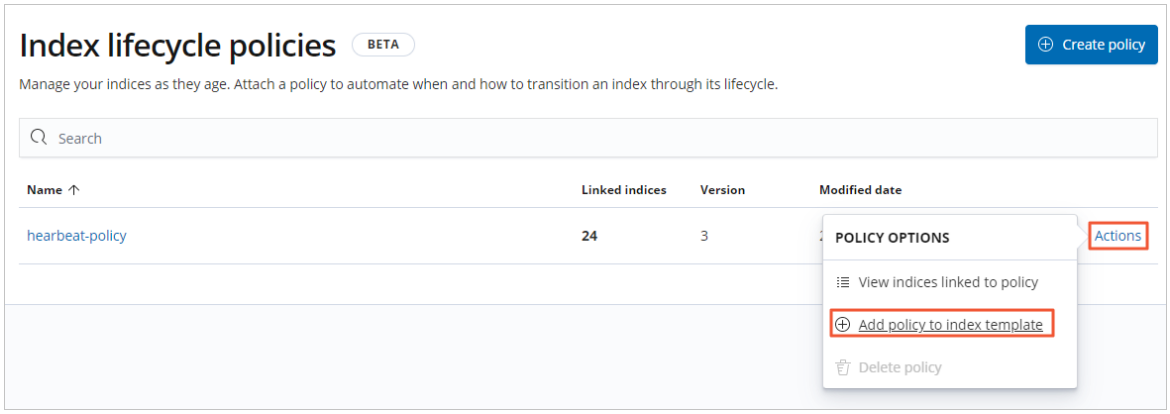
- 策略名创建后将无法更改。
- 您也可以在Kibana控制台上创建策略，但是Kibana上指定的max_age最小单位为小时，而通过API方式，可指定最小单位为秒。

步骤四：为ILM策略关联索引模板

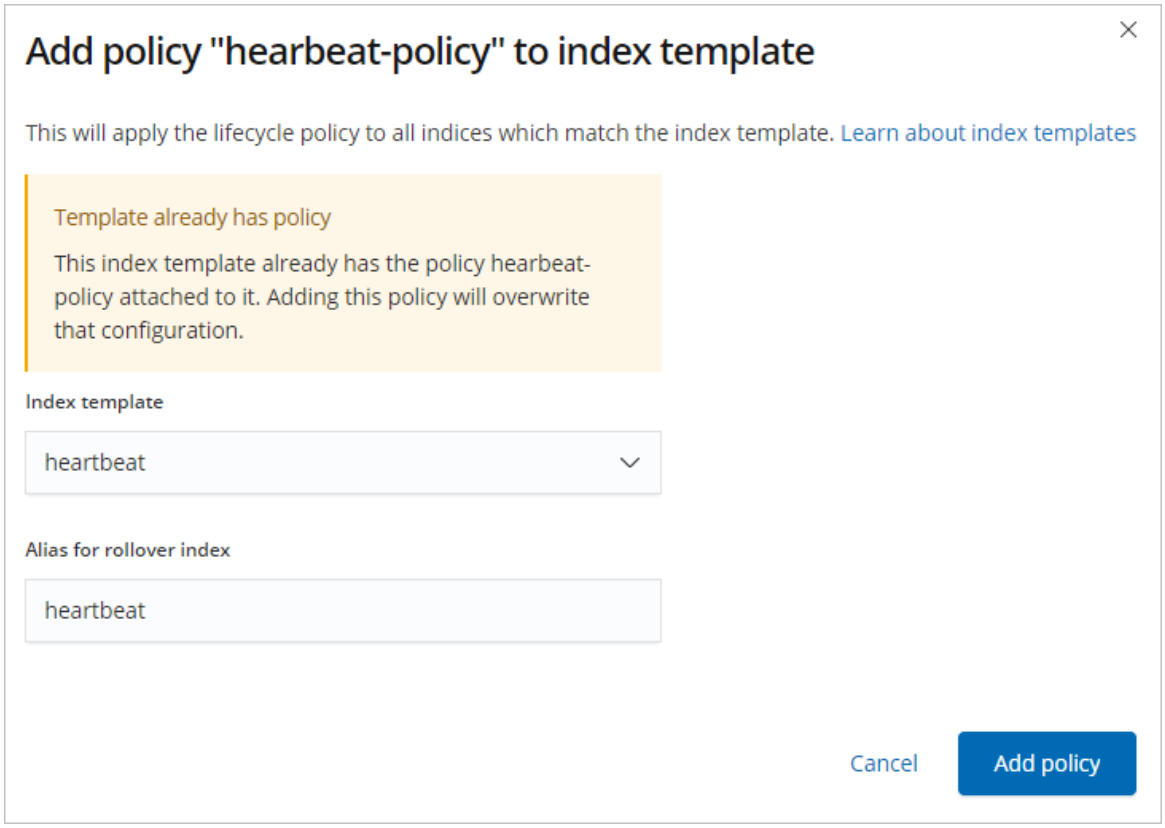
启动Heartbeat后，系统会自动在对应的Elasticsearch中创建Heartbeat索引模板。您需要为步骤三：创建ILM策略中创建的自定义ILM策略关联该索引模板。

1. 登录目标阿里云Elasticsearch实例的Kibana控制台。

   具体步骤请参见登录Kibana控制台。
2. 在左侧导航栏，单击**Management**。
3. 在**Elasticsearch**区域，单击**Index Lifecycle Policies**。
4. 在**Index lifecycle policies**列表中，选择**Actions > Add policy to index template**。



5. 在弹出的对话框中，从**Index template**列表中选择索引模板，并在**Alias for rollover index**文本框中输入索引别名。

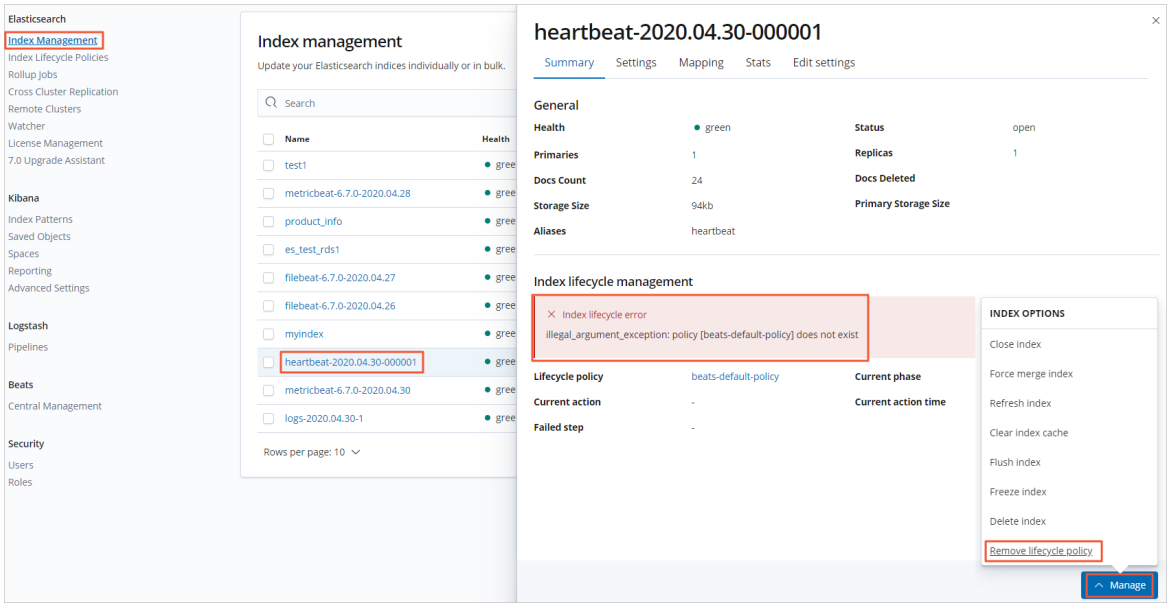


6. 单击Add policy。

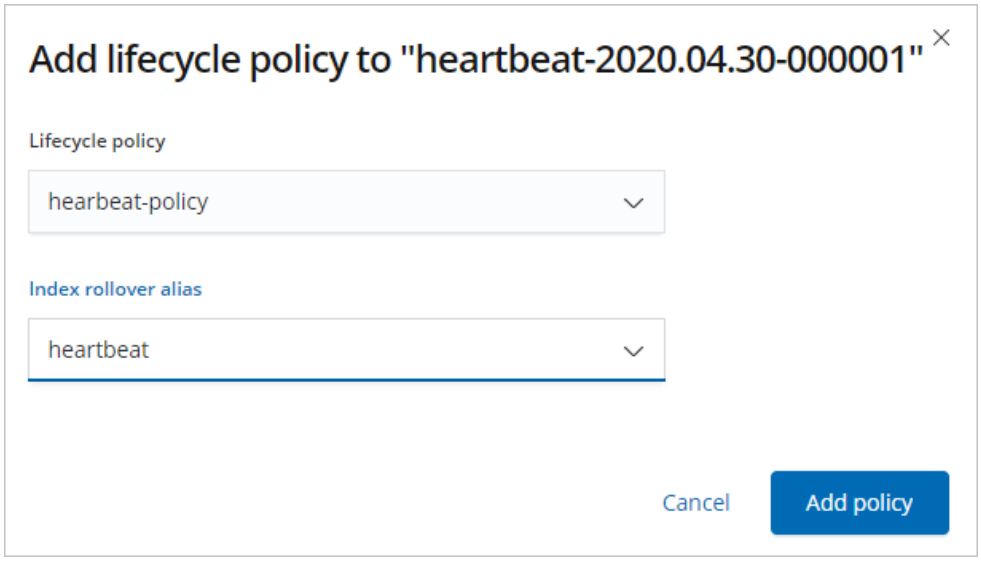
步骤五：为索引关联ILM策略

启动Heartbeat后，系统会自动在对应的Elasticsearch中创建索引。您需要为第一个索引关联对应的ILM策略，该ILM策略已经关联了索引模板（步骤四：为ILM策略关联索引模板）。

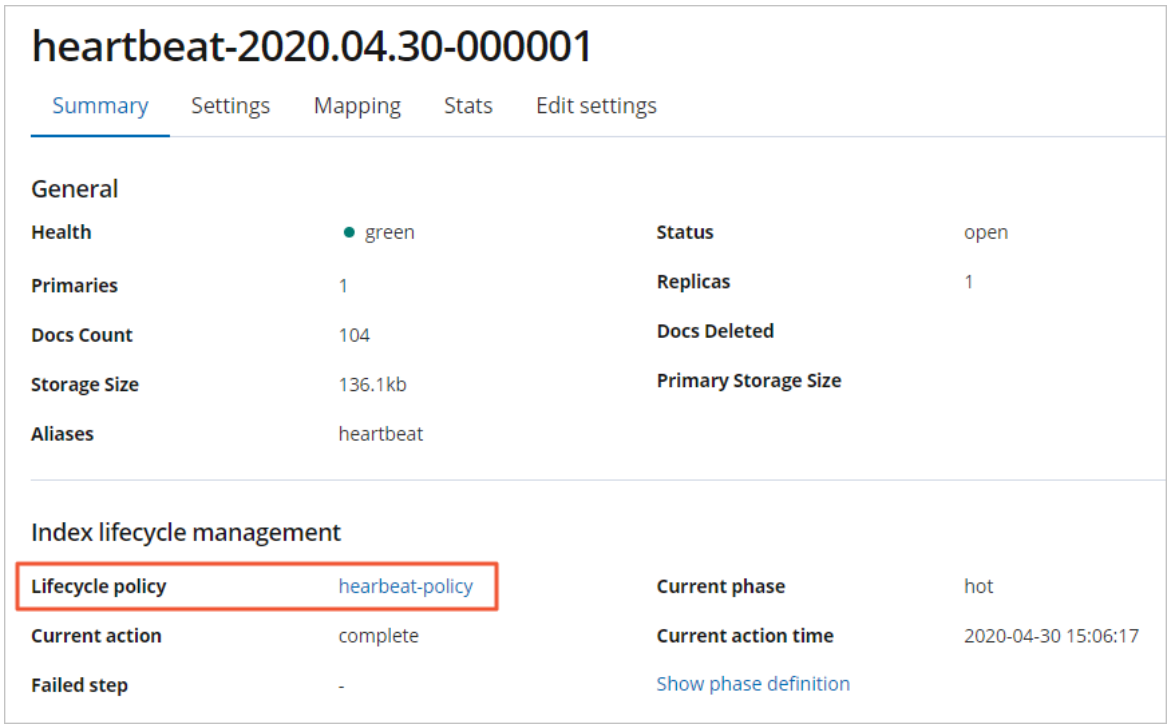
1. 在Management页面的Elasticsearch区域中，单击Index Management。
2. 在Index management列表中，找到目标索引，单击索引名称。
3. 在Summary页面，选择Manage > Remove lifecycle policy，移除Heartbeat自带的默认策略。



4. 在弹出的对话框中，单击Remove policy。
5. 再选择Manage > Add lifecycle policy。
6. 在弹出的对话框中，从Lifecycle policy列表中选择步骤三：创建ILM策略中创建的生命周期策略，并在Index rollover alias输入框中输入步骤四：为ILM策略关联索引模板中定义的索引别名，单击Add policy。

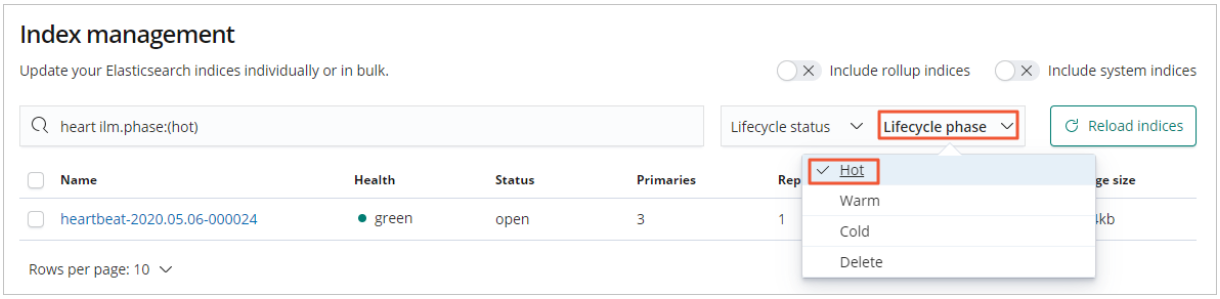


关联成功后，结果如下图。



步骤六：查看各阶段索引

查看hot阶段的索引：在Index management页面，选择Lifecycle phase > Hot。



您也可以使用同样的方式，查看其他阶段的索引。

常见问题

Q：如何设置ILM策略周期？

A：由于索引生命周期策略默认是10分钟检查一次符合策略的索引，因此在这10分钟内索引中的数据可能会超出指定的阈值。例如在步骤三：创建ILM策略时，设置max_docs为100，但doc数量在超过100后才触发索引滚动更新，此时可通过修改indices.lifecycle.poll_interval参数来控制检查频率，使索引在阈值范围内滚动更新。

> 注意 请慎重修改该参数值，避免时间间隔太短给节点增加不必要的负载，本测试中将其改成了1m。

```
PUT _cluster/settings
{
  "transient": {
    "indices.lifecycle.poll_interval":"1m"
  }
}
```

9.2.2. 通过索引生命周期管理实现冷热数据分离

本文介绍在阿里云Elasticsearch集群上，通过生命周期管理ILM（Index Lifecycle Management）功能，实现冷热数据分离的实践流程。通过本实践，您既可以实现在保证集群读写性能的基础上，自动维护集群上的冷热数据，又能通过优化集群架构，降低企业生产成本。

背景信息

当今大数据时代，数据时刻在更新变化。尤其是随着时间的积累，存储在阿里云Elasticsearch中的数据会越来越多，当数据达到一定量时，必然会造成服务的内存、CPU、IO等指标上涨，影响Elasticsearch的全文检索能力。为此Elasticsearch 6.6.0及以上版本提供了索引生命周期管理ILM功能，将索引生命周期分为4个阶段：hot、warm、cold、delete。其中hot阶段主要负责对索引进行滚动更新操作，warm、cold、delete阶段主要负责进一步处理索引数据，详细说明如下。

阶段	描述
hot	热数据阶段。主要处理时序数据的实时写入，可根据索引的文档数、大小、时长决定是否调用rollover API来滚动更新索引。
warm	warm阶段。索引不再写入，主要用来提供查询。

阶段	描述
cold	冷数据阶段。索引不再更新，查询很少，查询速度会变慢。
delete	删除数据阶段。索引将被删除。

您可以通过两种方式为索引添加生命周期管理策略：

- 为索引模板添加生命周期管理策略：将策略应用到整个别名覆盖的索引下，本文以此为例。
- 为单个索引添加生命周期管理策略：只能覆盖当前索引，新滚动的索引不再受策略影响。

在时序和冷热数据场景上应用ILM，可以大幅度节约存储成本。本文以冷热数据场景为例，介绍如何使用ILM功能。配置场景如下：

1. 将索引数据实时写入Elasticsearch。当索引数据增加到一定量时，数据自动写入新索引。
2. 旧索引在hot阶段停留30分钟，进入warm阶段。
3. warm阶段完成Merge及Shrink操作后，索引等待1小时（从滚动更新时算起），进入cold阶段。
4. cold阶段将热节点数据迁移到冷节点，实现冷热数据分离后，索引会在2个小时（从滚动更新时算起）后被删除。

注意事项

- Elasticsearch索引生命周期策略需密切贴近业务模型。例如，对多个不同结构的索引进行生命周期管理，建议各个索引配置独立的别名和生命周期策略，以便于管理。
- 使用rollover滚动索引，初始索引应以自增数字结尾（-000001），否则策略不生效，长度要求为6。例如，定义初始索引为myindex-000001，则rollover后的索引是myindex-000002，以此类推进行递增。如果集群中索引名不符合规范，建议进行索引重建。
- hot阶段主要处理数据写入。业务中需保证数据是按照时间顺序写入的，处于warm和cold阶段的索引不建议进行数据写入。例如，在warm阶段配置actions为shrink或read only，那么索引进入warm阶段后将处于只读状态，数据无法写入。

操作流程

1. 步骤一：创建冷热集群并查看节点的冷热属性

   在创建集群时设置节点的冷热属性。

2. 步骤二：为索引配置生命周期管理策略

   定义ILM策略，并将该策略应用到别名覆盖的所有索引下。

3. 步骤三：验证数据分布

   验证cold阶段索引的shard是否分布在冷数据节点上。

4. 步骤四：更新ILM策略

   更新已有策略。

5. 步骤五：切换ILM策略

   在不同策略间实现滚动切换。

步骤一：创建冷热集群并查看节点的冷热属性

冷热集群是指在集群中包含冷、热两种属性的节点，可以提高Elasticsearch的处理性能和服务稳定性。两者区别如下。

节点类型	存储数据要求	读写性能要求	规格要求	存储要求
热节点（hot）	近期数据，例如最近2天的日志数据。	高	高，例如32核64 GB	建议使用SSD云盘存储数据，存储空间大小需根据数据大小进行设置。
冷节点（warm）	历史数据，例如2天之前的日志数据。	低	低，例如8核32 GB	• 建议使用高效云盘存储数据，存储空间大小需根据数据大小进行设置。 • 建议使用阿里云自研OpenStore存储功能，实现海量冷数据Serverless存储，详细信息请参见通过OpenStore实现海量数据存储。

1. 在购买阿里云Elasticsearch实例时，启用冷数据节点，即可创建冷热集群。

   当您启用了冷数据节点并购买后，系统会在节点启动参数中加入-Enode.attr.box_type参数。

   - 热数据节点：-Enode.attr.box_type=hot
   - 冷数据节点：-Enode.attr.box_type=warm

   > **说明**
   > - 购买实例时，只有当启用了冷数据节点后，数据节点才会变成热节点。
   > - 本文以阿里云Elasticsearch 6.7.0版本为例，所涉及的操作及图片仅适用于该版本，其他版本以实际界面为准。

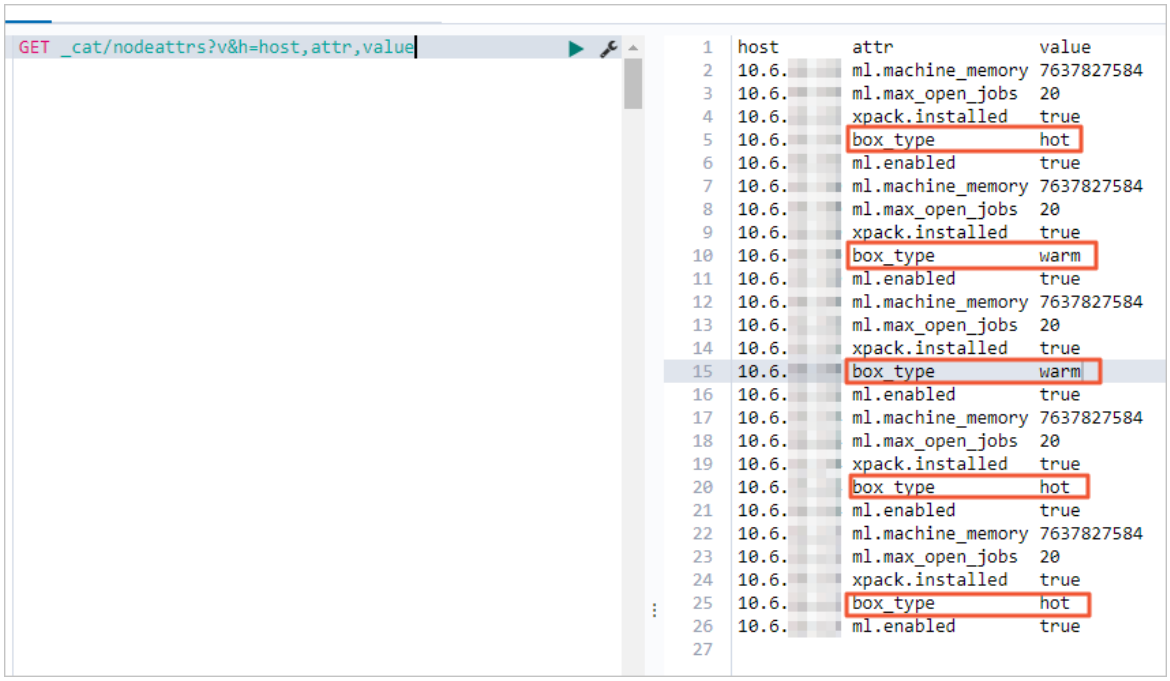
2. 登录该集群的Kibana控制台，在左侧导航栏，单击**Dev Tools**。

   登录Kibana控制台的具体操作请参见登录Kibana控制台。

3. 在**Console**中，执行如下命令，查看集群冷热节点属性。

   ```
   GET _cat/nodeattrs?v&h=host,attr,value
   ```

结果显示集群中包含3个hot节点，2个warm节点，支持冷热架构。



步骤二：为索引配置生命周期管理策略

1. 在Kibana控制台中，执行如下命令，通过API方式定义ILM策略。

```
PUT /_ilm/policy/game-policy
{
  "policy": {
    "phases": {
      "hot": {
        "actions": {
          "rollover": {
            "max_size": "1GB",
            "max_age": "1d",
            "max_docs": 1000
          }
        }
      },
      "warm": {
        "min_age": "30m",
        "actions": {
          "forcemerge": {
                "max_num_segments":1
              },
          "shrink": {
                "number_of_shards":1
              }
        }
      },
      "cold": {
        "min_age": "1h",
        "actions": {
          "allocate": {
            "require": {
              "box_type": "warm"
            }
          }
        }
      },
      "delete": {
        "min_age": "2h",
        "actions": {
          "delete": {}
        }
      }
    }
  }
}
```

参数	说明

参数	说明
hot	该策略设置索引只要满足其中任一条件：数据写入达到1 GB、使用超过1天、doc数超过1000，就会触发索引滚动更新。此时系统将创建一个新索引，该索引将重新启动策略，而旧索引将在滚动更新后等待30分钟进入warm阶段。 **注意** 目前Elasticsearch支持在rollover中配置三种归档策略：max_docs、max_size、max_age，满足其中任何一个条件都会触发索引归档操作。
warm	索引进入warm阶段后，ILM会将索引收缩到1个分片，强制合并为1个段。完成该操作后，索引将在1小时（从滚动更新时算起）后进入cold阶段。
cold	索引进入cold阶段后，ILM将索引从hot节点移动到warm（冷数据）节点。完成操作后，索引将在2小时后进入delete阶段。
delete	索引进入delete阶段后被删除。

说明

- 策略名创建后将无法更改。
- 您也可以在Kibana控制台上创建策略，但是Kibana上指定的max_age最小单位为小时，而通过API方式，可指定最小单位为秒。

2. 创建索引模板。

   在settings中指定冷热属性，数据写入后存储在hot节点上。

```
PUT _template/gamestabes_template
{
  "index_patterns" : ["gamestabes-*"],
  "settings": {
    "index.number_of_shards": 5,
    "index.number_of_replicas": 1,
    "index.routing.allocation.require.box_type":"hot",
    "index.lifecycle.name": "game-policy",
    "index.lifecycle.rollover_alias": "gamestabes"
  }
}
```

参数	说明
index.routing.allocation.require.box_type	指定索引新建时所分配的节点。
index.lifecycle.name	指定生命周期策略名称。
index.lifecycle.rollover_alias	指定rollover别名。

3. 基于序号创建初始索引。

```
PUT gamestabes-000001
{
"aliases": {
    "gamestabes":{
       "is_write_index": true
        }
      }
}
```

您也可以基于时间创建索引，详情请参见using date math。

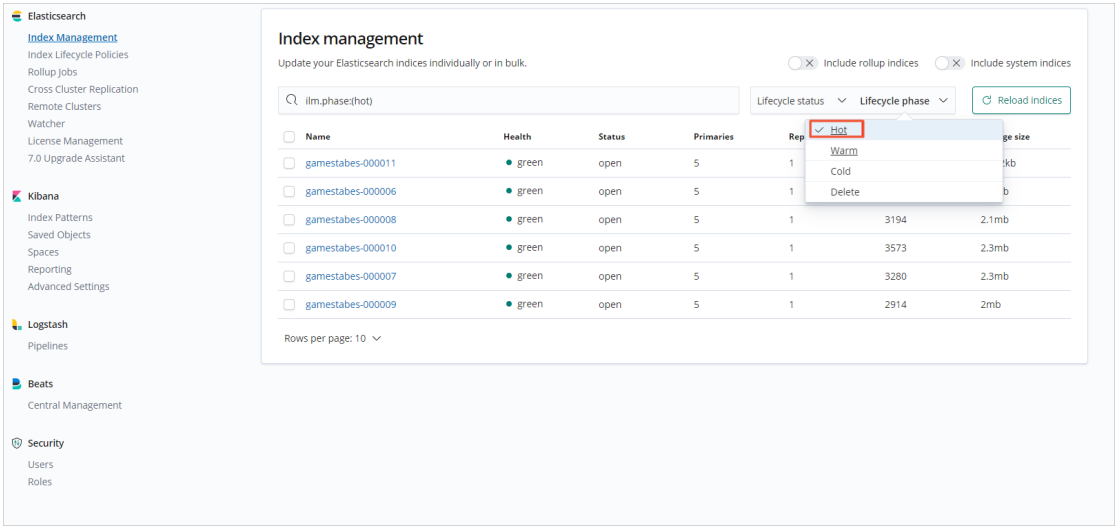
4. 通过别名写入数据。

   当数据达到rollover条件，并触发ILM检测周期后，索引将进行滚动更新。

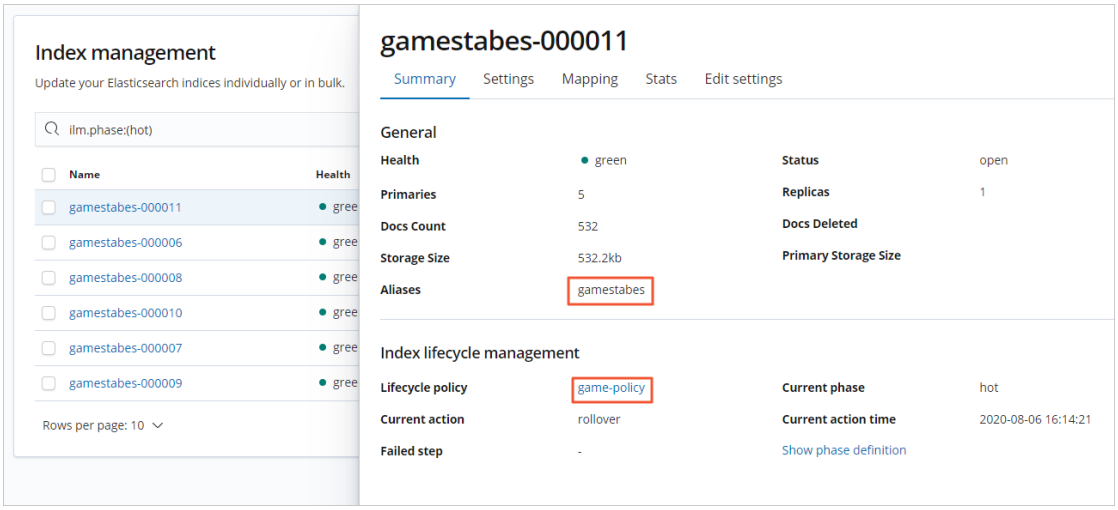
```
PUT gamestabes/_doc/1
{
    "EU_Sales" : 3.58,
    "Genre" : "Platform",
    "Global_Sales" : 40.24,
    "JP_Sales" : 6.81,
    "Name" : "Super Mario Bros.",
    "Other_Sales" : 0.77,
    "Platform" : "NES",
    "Publisher" : "Nintendo",
    "Year_of_Release" : "1985",
    "na_Sales" : 29.08
}
```

> **说明** ILM默认10分钟检测一次符合策略标准的索引，当达到rollover条件后，索引将滚动到下一阶段，同时配置indices.lifecycle.poll_interval参数，修改检测周期。

5. 根据生命周期阶段过滤索引，并查看索引详细配置。
   i. 在左侧导航栏，单击**Management**。
   ii. 在**Elasticsearch**区域中，单击**Index Management**。
   iii. 在**Index management**中，单击**Lifecycle status**右侧的**Lifecycle phase**，从下拉列表中选择生命周期阶段进行过滤。

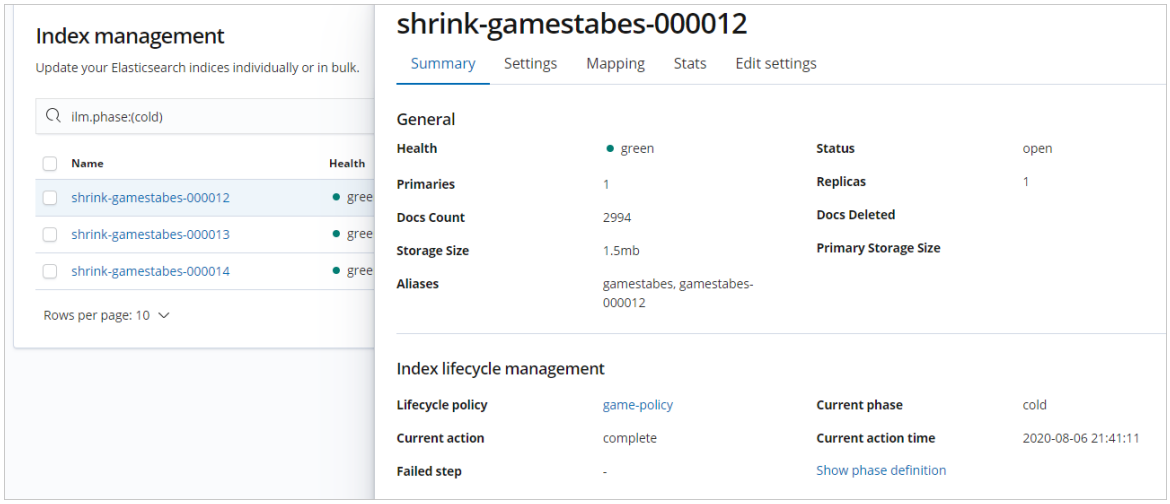


iv. 单击过滤后的索引，查看索引详细信息。



步骤三：验证数据分布

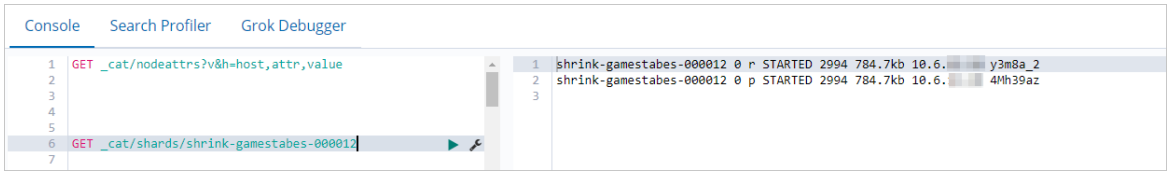
1. 查询进入cold阶段的索引，并查看其配置信息。



2. 查询处于cold阶段索引的shard分布。

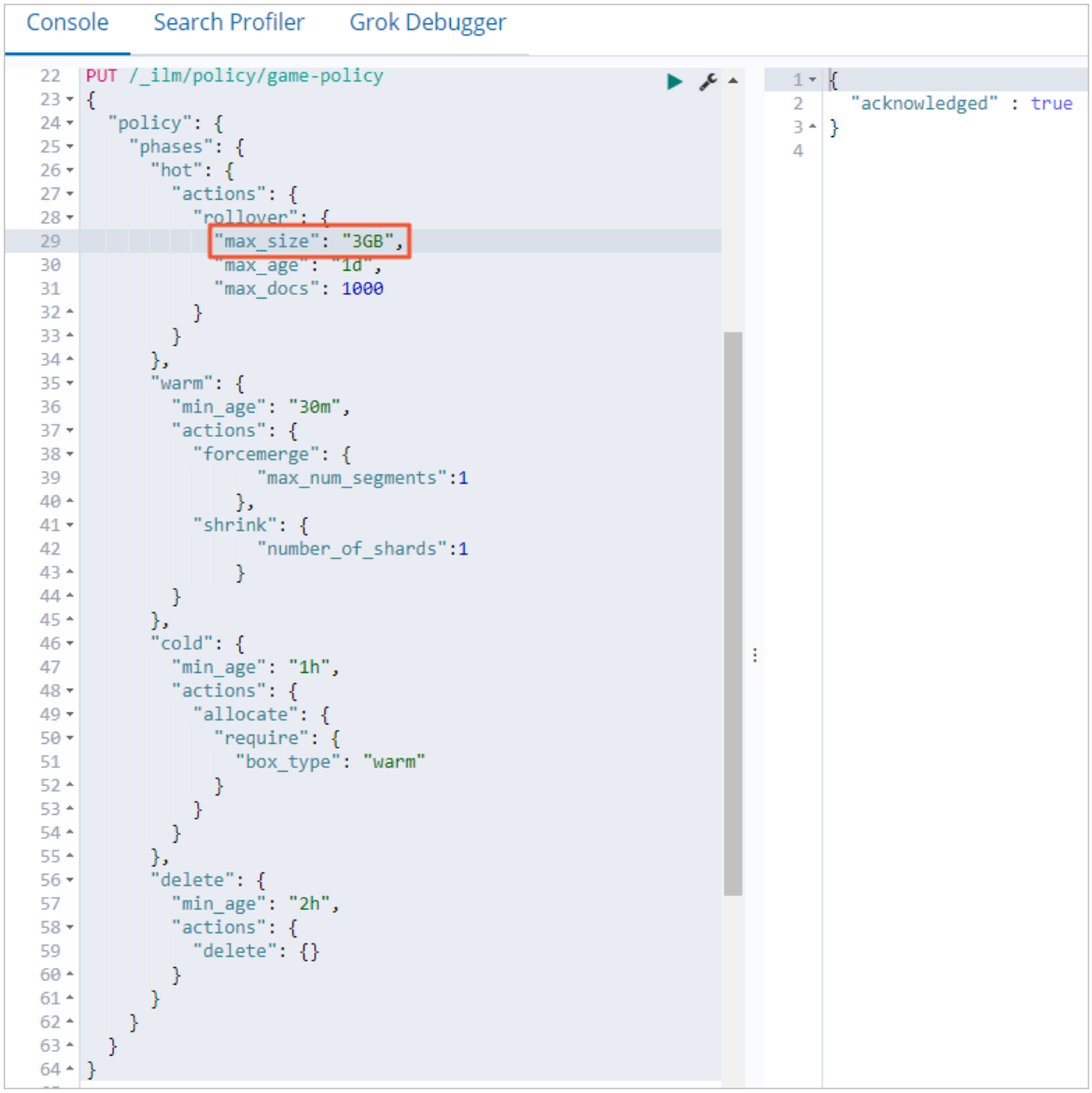
```
GET _cat/shards?shrink-gamestables-000012
```

返回结果如下。根据返回结果可知，cold阶段的索引数据主要分布在冷数据节点上。



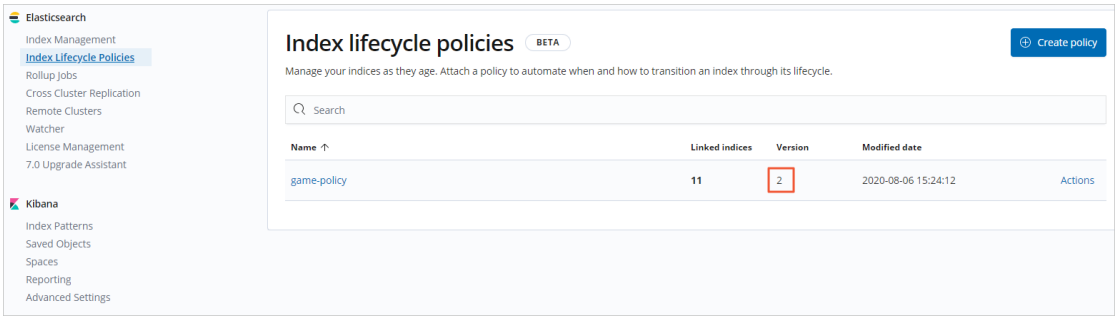
步骤四：更新ILM策略

1. 更新正在运行的ILM策略。



2. 查看更新后的策略版本。
   i. 在左侧导航栏，单击Management。
   ii. 在Elasticsearch区域中，单击Index Lifecycle Policies。
   iii. 在Index lifecycle policies中，查看更新后的策略版本。

   更新后策略的版本号增加1，此时正在滚动写入的索引依旧使用旧策略，新策略将在下次滚动更新时生效。



步骤五：切换ILM策略

1. 创建新策略。

```
PUT /_ilm/policy/game-new
{
  "policy": {
    "phases": {
      "hot": {
        "actions": {
          "rollover": {
            "max_size": "3GB",
            "max_age": "1d",
            "max_docs": 1000
          }
        }
      },
      "warm": {
        "min_age": "30m",
        "actions": {
          "forcemerge": {
                "max_num_segments":1
              },
          "shrink": {
                "number_of_shards":1
              }
        }
      },
      "cold": {
        "min_age": "1h",
        "actions": {
          "allocate": {
            "require": {
              "box_type": "warm"
            }
          }
        }
      },
      "delete": {
        "min_age": "2h",
        "actions": {
          "delete": {}
        }
      }
    }
  }
}
```

2. 为模板绑定新策略。

```
PUT _template/gamestabes_template
{
  "index_patterns" : ["gamestabes-*"],
  "settings": {
    "index.number_of_shards": 5,
    "index.number_of_replicas": 1,
    "index.routing.allocation.require.box_type":"hot",
    "index.lifecycle.name": "game-new",
    "index.lifecycle.rollover_alias": "gamestabes"
  }
}
```

注意

- 切换策略后，新策略不会立即生效。当前正在滚动写入的索引依旧使用旧策略，直到当前索引rollover生成新索引，新策略才会生效。
- 应用旧策略创建的索引，依旧绑定旧策略。如果您需要为这些索引绑定新策略，可执行 `PUT gamestabes-*/_settings` 命令，详情请参见switching policies for an index。

常见问题

Q：如何设置ILM策略周期？

A：由于索引生命周期策略默认是10分钟检查一次符合策略的索引，因此在这10分钟内索引中的数据可能会超出指定的阈值。例如在步骤二：为索引配置生命周期管理策略时，设置max_docs为1000，但doc数量在超过1000后才触发索引滚动更新，此时可通过修改indices.lifecycle.poll_interval参数来控制检查频率，使索引在阈值范围内滚动更新。

注意 请慎重修改该参数值，避免时间间隔太短给节点增加不必要的负载，本测试中将其改成了1m。

```
PUT _cluster/settings
{
  "transient": {
    "indices.lifecycle.poll_interval":"1m"
  }
}
```

9.3. X-Pack高级特性应用

9.3.1. 使用跨集群复制功能迁移数据

当您需要将本地Elasticsearch集群中的索引数据迁移到一个远程集群中，或者将一个远程集群中的索引数据迁移到本地集群，可通过跨集群复制CCR（Cross Cluster Replication）功能实现。本文介绍具体的实现方法。

背景信息

CCR是开源Elasticsearch在platinum版本中发布的一个商业特性。购买阿里云Elasticsearch实例后，您无需额外付费，只需要简单配置，即可使用CCR功能。CCR的应用场景如下：

- 灾难恢复及高可用性

  对于分布在不同地域的Elasticsearch集群，您可以通过CCR进行数据备份。当其中一个集群发生故障时，您可以通过访问其他集群来获取故障集群的数据，保证数据不丢失。

- 就近访问数据

  例如A集团下有多个子公司，各子公司所分布的地域不同。为了提高业务处理速度，可按照地理位置划分子公司要承担的业务，并通过CCR将业务数据分发给各地域中的Elasticsearch集群。子公司在处理业务时，可直接访问当前所在地域的集群。

- 集中报告

  通过CCR，将多个数据量较小的集群中的数据复制到一个中央集群中，进行可视化分析与报告。

使用CCR功能，需要准备两种类型的集群。一个是远程集群，即提供源数据（Leader index）的集群；一个是本地集群，即订阅数据（Follower index）的集群。该功能为被动复制，即所有复制任务都是由本地集群执行。同时支持批量实时迁移数据，更多详情请参见Cross-cluster replication。

使用限制

- 新网络架构下的实例不支持与旧网络架构下的实例进行跨集群reindex、跨集群搜索、跨集群复制等实例互通操作。如果需要进行互通，请确保实例创建在同一网络架构下。

  > 说明
  > - 2020年10月，阿里云Elasticsearch对网络架构进行了调整。2020年10月之前为旧网络架构，2020年10月及之后为新网络架构。
  > - 对于华北3（张家口）和海外地域，由于网络架构调整时间不确定，因此需要提交工单，联系阿里云Elasticsearch技术支持，校验网络是否可以互通。

- 在旧网络架构下，仅6.7.0及以上版本的单可用区的阿里云Elasticsearch实例支持跨集群复制功能。新网络架构下，只有版本限制，没有可用区限制。

操作流程

1. 准备工作

   准备远程和本地集群，以及待迁移的索引。

2. 步骤一：配置实例网络互通

   连通远程和本地集群的网络。

3. 步骤二：添加远程集群

   在本地集群的Kibana控制台中，添加远程集群。

4. 步骤三：配置跨集群复制

   在本地集群的Kibana控制台中，配置待迁移和迁移后的索引。

5. 步骤四：验证数据迁移结果

   在远程集群中插入数据，在本地集群中，验证数据是否迁移成功。

准备工作

1. 准备远程和本地Elasticsearch集群。

   具体操作，请参见创建阿里云Elasticsearch实例。要求两个实例为相同版本（6.7.0及以上），且在同一专有网络和虚拟交换机下。

2. 登录远程集群的Kibana控制台，在左侧导航栏，单击Dev Tools。

   登录Kibana控制台的具体操作，请参见登录Kibana控制台。

   > 说明 本文以阿里云Elasticsearch 6.7.0版本为例，其他版本操作可能略有差别，请以实际界面为准。

3. 在Console中执行如下命令，创建待迁移的索引。

   > 注意
   > - 对于7.0及以下版本的Elasticsearch实例，在创建索引时，需要开启soft_deletes属性，否则会报错。您可以通过 `GET /<yourIndexName>/_settings?pretty` 命令，查看是否开启了soft_deletes属性。开启时，您可以在返回结果中看到soft_deletes属性的配置。
   > - 如果您需要迁移已创建的索引，需要通过重建索引来开启soft_deletes属性。

   ```
   PUT myindex
   {
     "settings": {
       "index.soft_deletes.retention.operations": 1024,
       "index.soft_deletes.enabled": true
     }
   }
   ```

4. 关闭待迁移的索引的物理复制功能。

   对于6.7.0版本的阿里云Elasticsearch实例，系统会默认为新建索引开启物理复制功能。使用CCR功能时，需要先关闭物理复制功能。

   i. 关闭索引。

   ```
   POST myindex/_close
   ```

   ii. 更新索引settings，关闭物理复制功能。

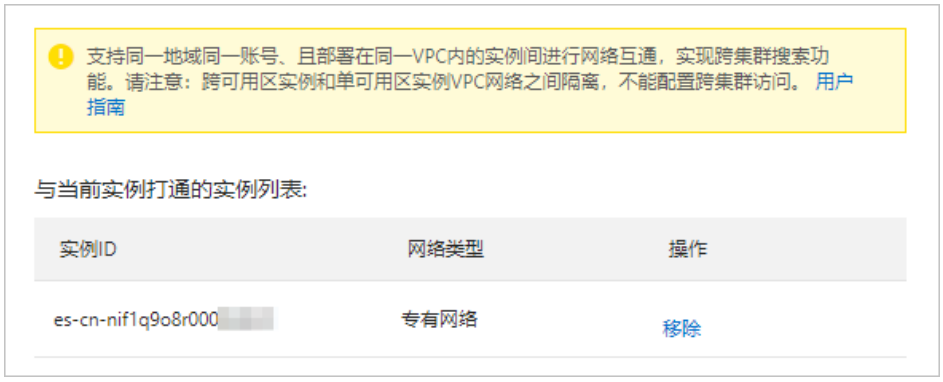
   ```
   PUT myindex/_settings
   {
   "index.replication.type" : null
   }
   ```

   iii. 打开索引。

   ```
   POST myindex/_open
   ```

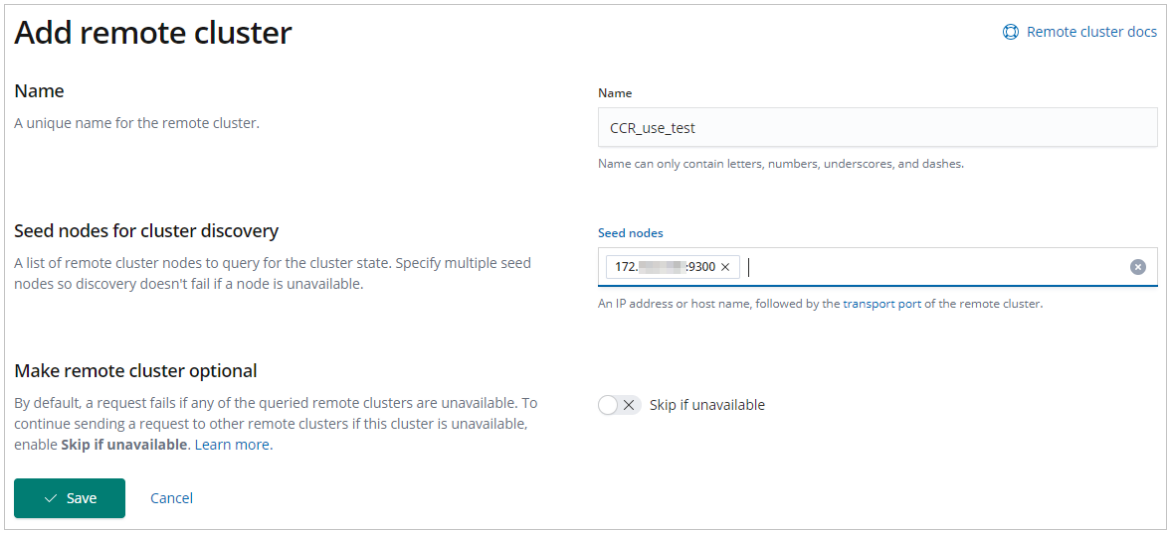
步骤一：配置实例网络互通

参见通过配置实例网络互通使用跨集群搜索功能，在远程集群中添加需要进行网络互通的本地集群。最终配置如下。

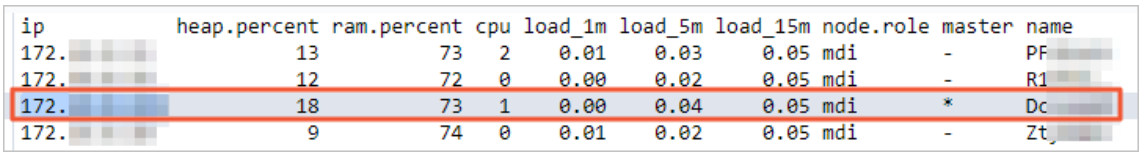


步骤二：添加远程集群

1. 登录本地集群的Kibana控制台。
2. 在左侧导航栏，单击Management。
3. 在Elasticsearch区域中，单击Remote Clusters。
4. 单击Add a remote cluster。
5. 在Add remote cluster页面中，输入远程集群信息。



- **Name**：远程集群的名称，不可重复。
- **Seed nodes**：需要配置为远程集群的节点的IP地址:9300。远程集群的节点的IP地址，可在远程集群的Kibana控制台中，使用 `GET /_cat/nodes?v` 命令获取。所配置的节点中必须包含主节点，建议您配置多个子节点，确保当主节点不可用时，可以继续使用跨集群复制功能。

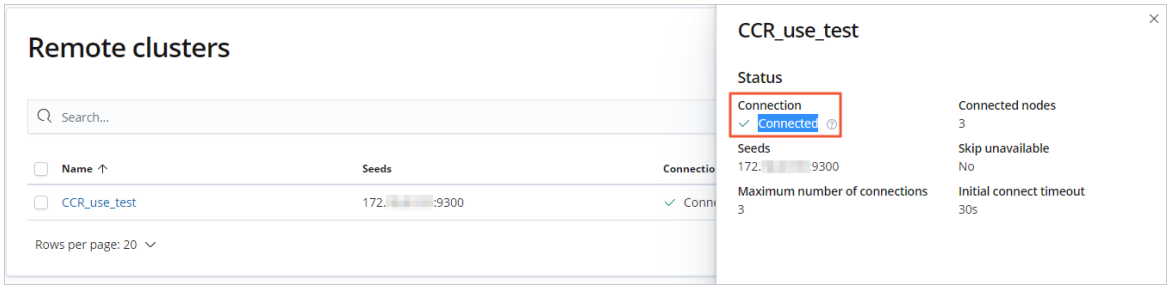


```
ip          heap.percent ram.percent cpu load_1m load_5m load_15m node.role master name
172.                  13          73   2    0.01    0.03     0.05 mdi       -      PF
172.                  12          72   0    0.00    0.02     0.05 mdi       -      R1
172.                  18          73   1    0.00    0.04     0.05 mdi       *      Dc
172.                   9          74   0    0.01    0.02     0.05 mdi       -      Zt
```

> **注意** 由于CCR功能是Kibana通过数据节点之间的TCP端口（9300）访问数据节点IP的形式来进行网络互通，因此不支持HTTP端口（9200）访问。

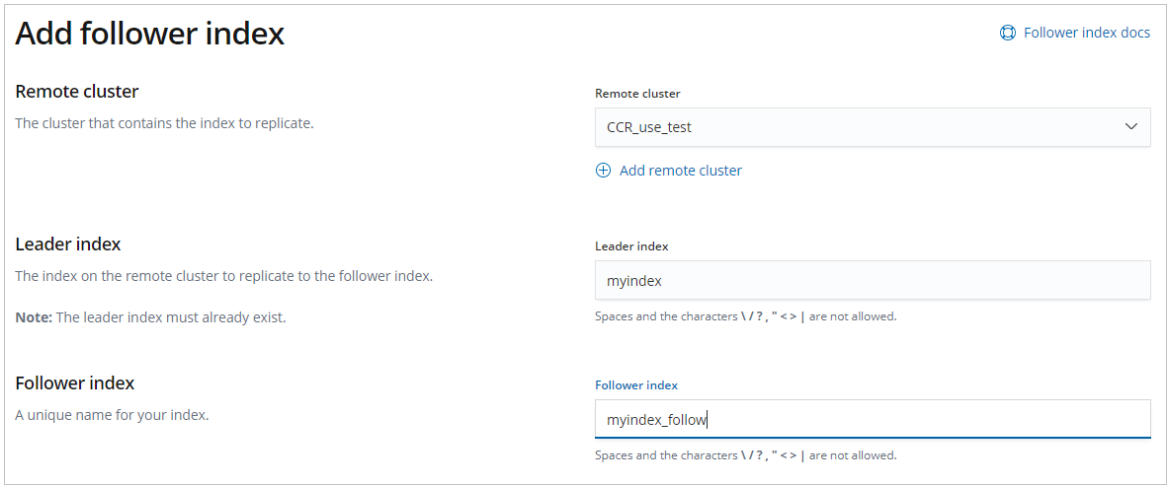
6. 单击Save。

   保存后，系统会自动连接远程集群。连接成功后，显示Connected。



步骤三：配置跨集群复制

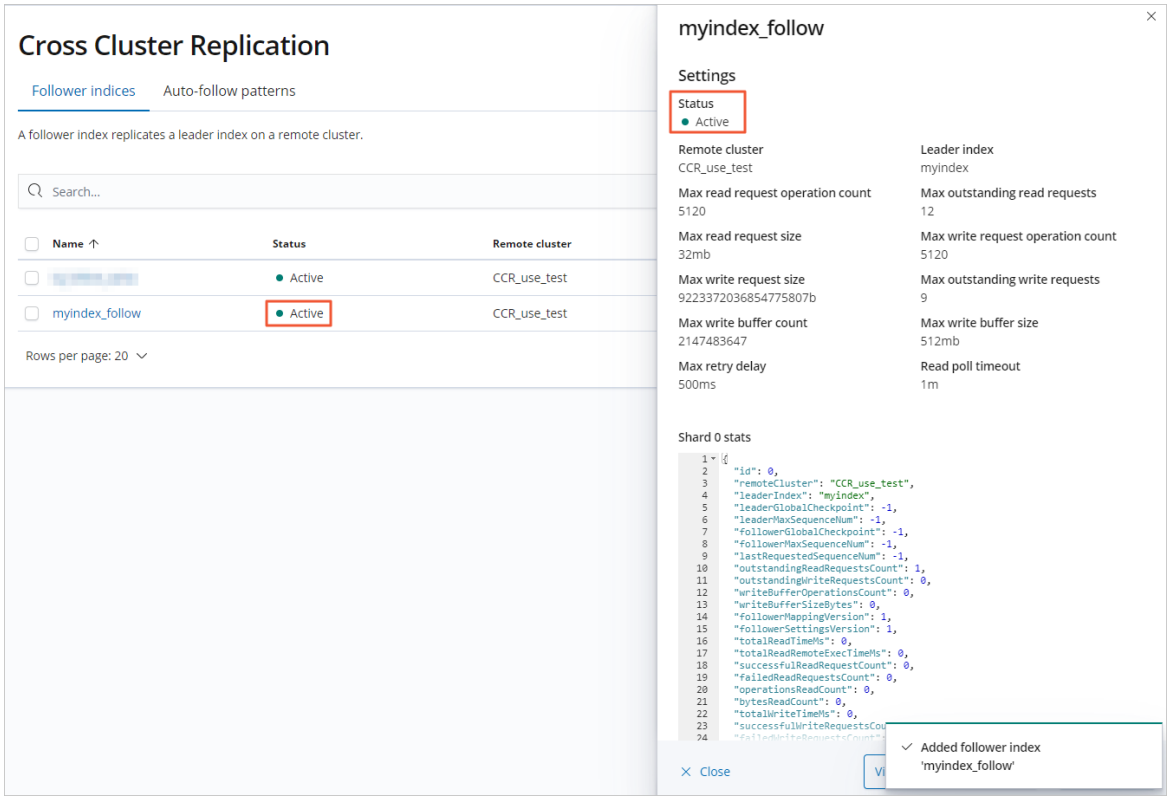
1. 在本地集群Kibana控制台的Management页面，单击Elasticsearch区域中的Cross Cluster Replication。
2. 单击Create a follower index。
3. 在Add follower index页面，配置跨集群复制信息。



参数	说明
Remote cluster	选择您在步骤二：添加远程集群中添加的集群。
Leader index	待迁移的索引。本文使用准备工作中创建的myindex索引。
Follower index	迁移数据生成的索引。索引名称不可重复。

4. 单击Create。

   创建成功后，索引状态显示为Active。



步骤四：验证数据迁移结果

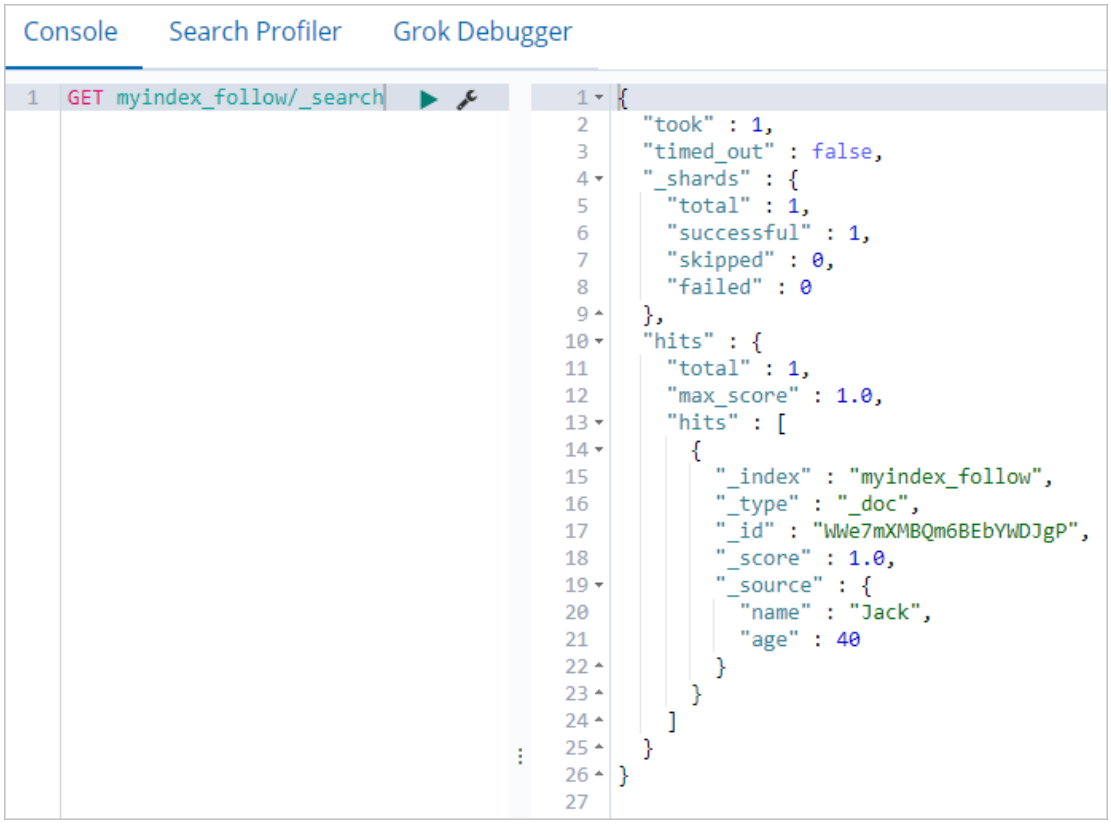
1. 在远程集群的Kibana控制台，执行如下命令，在远程集群中插入数据。

```
POST myindex/_doc/
{
  "name":"Jack",
  "age":40
}
```

2. 在本地集群的Kibana控制台，执行如下命令，验证数据是否迁移成功。

```
GET myindex_follow/_search
```

迁移成功后，返回如下结果。



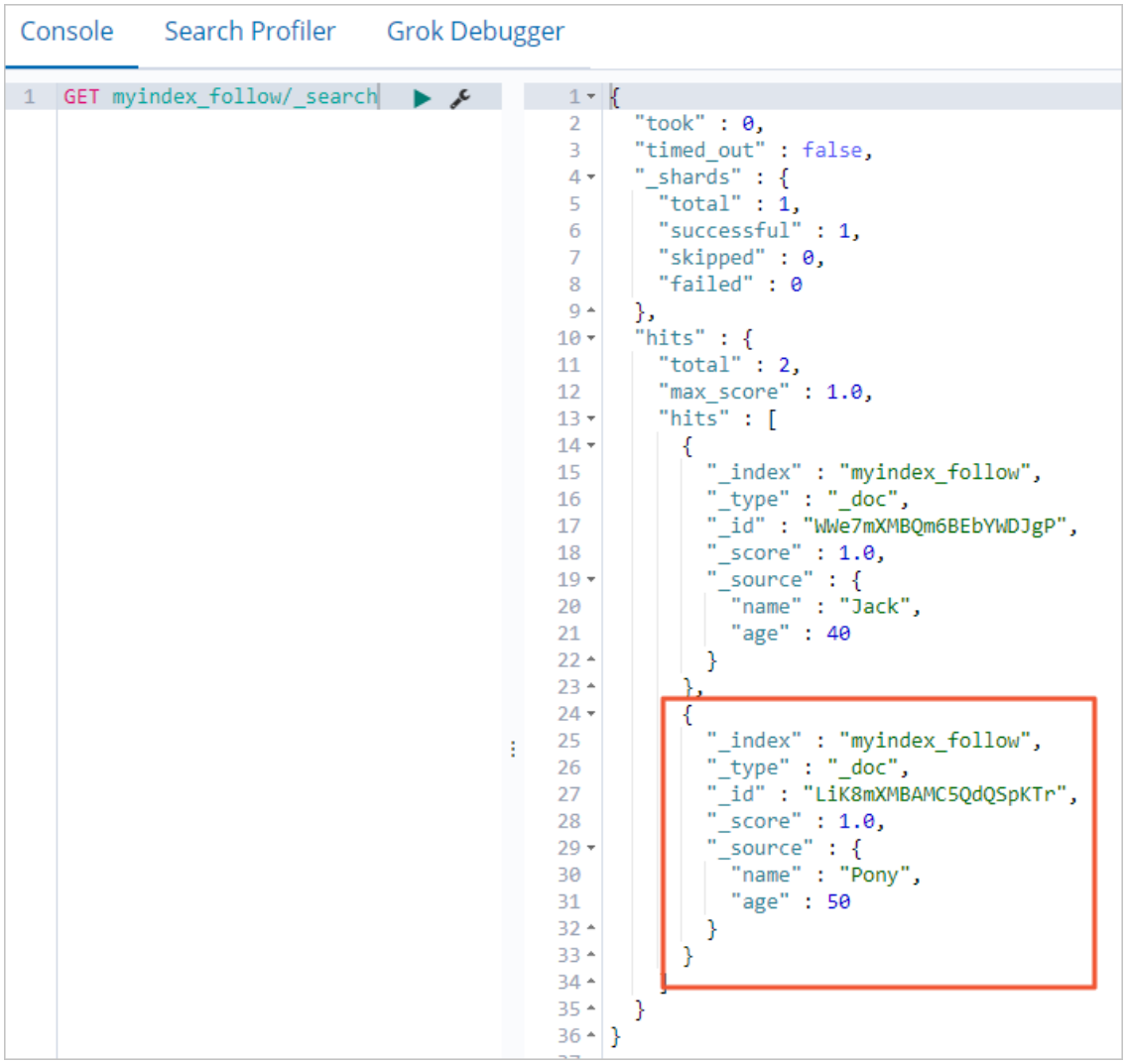
从以上结果可以看到，远程集群的Leader索引（myindex）中的数据，已通过CCR功能复制到了本地集群的Follower索引（myindex_follow）中。

> **注意** Follower索引为只读状态，如果需要和普通索引一样写入数据，需要先转换成普通索引。详细信息，请参见使用Elasticsearch跨集群复制进行跨数据中心复制。

3. 在远程集群中，重新插入一条数据，随即在本地集群中进行查看，验证增量数据是否实时同步。

```
POST myindex/_doc/
{
  "name":"Pony",
  "age":50
}
```

数据插入后，立即在本地集群中进行查看，结果如下。



从以上结果可以看到，通过CCR可以实现增量数据的实时同步。

> **说明** 您也可以通过CCR功能的API，进行跨集群复制相关操作。详细信息，请参见Cross-cluster replication APIs。

常见问题

Q：为什么在添加远程集群时，可以使用9300端口。但是通过域名访问Elasticsearch集群时，只能使用9200端口？

A：9300端口实际上是开放的。因为涉及到产品的安全策略，在SLB端口校验过程中，外网访问Elasticsearch域名的时候，只开放了9200端口。

9.3.2. X-Pack集成LDAP认证最佳实践

本文介绍如何基于阿里云Elasticsearch配置轻量目录访问协议LDAP（Lightweight Directory Access Protocol）认证，以实现相应角色的LDAP用户访问阿里云Elasticsearch。

前提条件

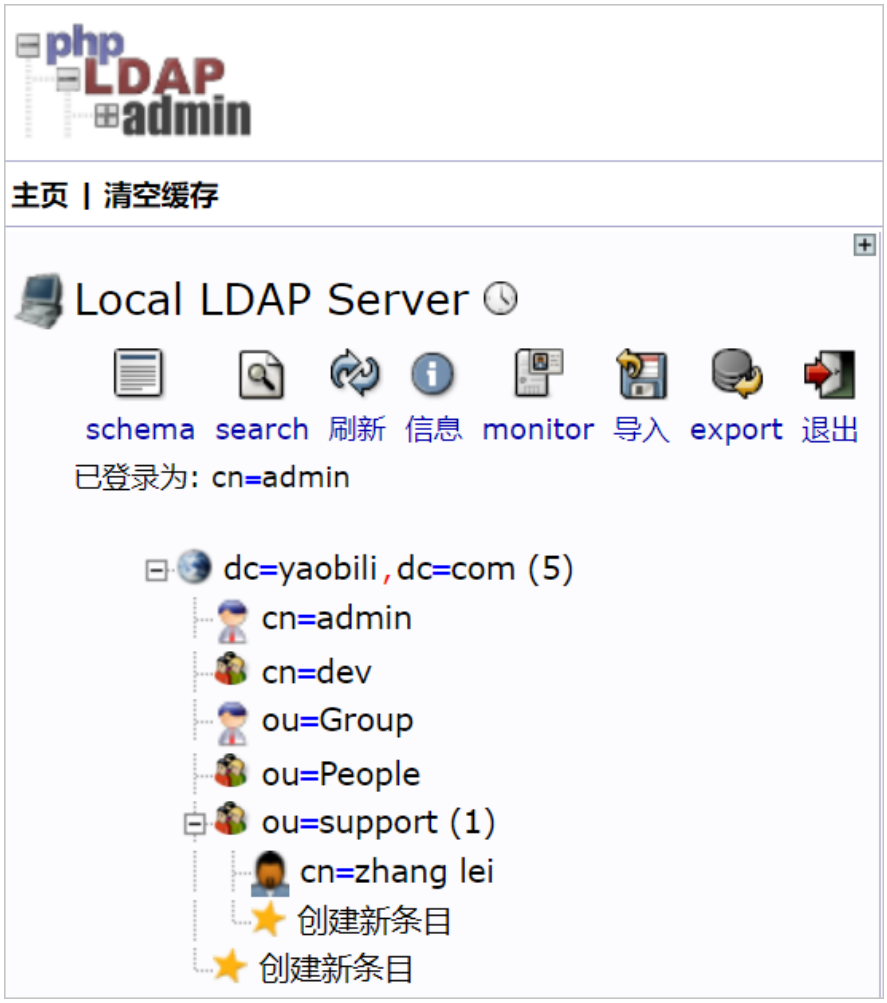
您已完成以下操作：

- 创建阿里云Elasticsearch实例。本文以6.7版本为例。

  具体操作，请参见创建阿里云Elasticsearch实例。

- 准备和阿里云Elasticsearch同VPC下的LDAP服务和用户数据，本文以OpenLDAP 2.4.44版本为例。

  具体操作，请参见LDAP官方文档。



注意事项

自2020年10月起，阿里云Elasticsearch对不同地域进行了网络架构的调整，对创建的实例有以下影响：

- 2020年10月之前创建的实例均在旧网络架构下，即Elasticsearch实例处于用户VPC下，如果需要访问公网，可以直接使用SNAT功能或自建Nginx代理。
- 2020年10月及之后创建的实例均在新网络架构下，即Elasticsearch实例处于Elasticsearch服务VPC下，X-Pack Watcher功能受到网络限制，为解决此问题，阿里云Elasticsearch提供了实例私网连接方案，详细信息请参见配置实例私网连接。如果您还需要将报警信息推送至公网环境，在通过实例私网连接打通Elasticsearch服务VPC和用户VPC的基础上，还需对负载均衡后端服务配置Nginx代理或开启SNAT功能实现公网信息推送。

> 注意 实例私网连接方案是新网络架构下X-Pack Watcher、reindex、LDAP和AD（Active Directory）身份认证等功能受限的唯一解决方案，为保证功能使用不受影响，请严格按照文档配置。

-
- 2020年10月及之后创建的实例均为新网络架构，LDAP功能受到网络限制。为解决此问题，您可以使用PrivateLink进行VPC网络打通，具体配置请参见配置实例私网连接。如果您需要访问公网，则需要配置Nginx代理进行请求转发。
- 旧网络架构下，阿里云Elasticsearch仅支持单可用区的LDAP认证，不支持多可用区，新网络架构不受此限制。

操作流程

1. 步骤一：获取终端节点域名（可选）
2. 步骤二：配置LDAP认证
3. 步骤三：为域账号信息映射角色
4. 步骤四：验证结果

步骤一：获取终端节点域名（可选）

如果您创建的阿里云Elasticsearch处于新网络架构下（2020年10月及之后创建的实例属于新网络架构），需要借助PrivateLink，打通用户VPC与阿里云服务账号VPC，获取终端域名，为后续配置做准备。具体操作如下：

1. 创建与阿里云Elasticsearch实例处于同一VPC下，且支持PrivateLink功能的负载均衡实例。

   具体操作，请参见步骤一：创建支持PrivateLink功能的负载均衡实例。
2. 配置负载均衡实例。

   配置时，需要指定LDAP所在的服务器为后端服务器，监听端口为389。

   具体操作，请参见步骤二：配置负载均衡实例。
3. 创建终端节点服务。

   具体操作，请参见步骤三：创建终端节点服务。
4. 配置阿里云Elasticsearch私网互通。

   具体操作，请参见步骤四：配置阿里云Elasticsearch私网互通。
5. 获取终端节点域名。

   具体操作，请参见（可选）步骤五：查看终端节点域名。

   > **说明** 请先记录获取到的节点域名，该域名会在后续配置中使用。

步骤二：配置LDAP认证

目前，X-Pack集成LDAP认证支持通过以下两种方式配置：

- 用户搜索模式。
- 带有用户DNs特定模板的模式。

其中，用户搜索模式是最常见的操作方式。在此模式中，具有搜索LDAP目录权限的特定用户，根据X-Pack提供的用户名和LDAP属性搜索进行身份验证的用户的DN。一旦找到，X-Pack将使用找到的DN和提供的密码，尝试绑定到LDAP目录来验证用户。详细信息，请参见Configure an LDAP realm。

以下为LDAP管理DN的映射方式，需要在Elasticsearch的YML文件中添加如下配置，具体操作请参见配置YML参数。阿里云Elasticsearch实例的版本不同，添加的配置也不同，具体如下：

- 6.7版本

```
xpack.security.authc.realms.ldap1.type: ldap
xpack.security.authc.realms.ldap1.order: 0
xpack.security.authc.realms.ldap1.url: "ldap://ep-bp1dhpobznlgjhj9****-cn-hangzhou-i.epsrv
-bp1q8tcj2jjt5dwr****.cn-hangzhou.privatelink.aliyuncs.com:389"
xpack.security.authc.realms.ldap1.bind_dn: "cn=zhang lei,ou=support,dc=yaobili,dc=com"
xpack.security.authc.realms.ldap1.bind_password: "yourPassword"
xpack.security.authc.realms.ldap1.user_search.base_dn: "ou=support,dc=yaobili,dc=com"
xpack.security.authc.realms.ldap1.user_search.filter: "(cn={0})"
xpack.security.authc.realms.ldap1.group_search.base_dn: "ou=support,dc=yaobili,dc=com"
xpack.security.authc.realms.ldap1.unmapped_groups_as_roles: false
```

- 7.10版本

```
xpack.security.authc.realms.ldap.ldap1.order: 0
xpack.security.authc.realms.ldap.ldap1.url: "ldap://ep-bp1dhpobznlgjhj9****-cn-hangzhou-i.
epsrv-bp1q8tcj2jjt5dwr****.cn-hangzhou.privatelink.aliyuncs.com:389"
xpack.security.authc.realms.ldap.ldap1.bind_dn: "cn=srd_artddffctory,ou=github,ou=All User
s,dc=motenta,dc=ai"
xpack.security.authc.realms.ldap.ldap1.bind_password: "yourPassword"
xpack.security.authc.realms.ldap.ldap1.user_search.base_dn: "ou=support,dc=yaobili,dc=com"
xpack.security.authc.realms.ldap.ldap1.user_search.filter: "(cn={0})"
xpack.security.authc.realms.ldap.ldap1.group_search.base_dn: "ou=support,dc=yaobili,dc=com
"
xpack.security.authc.realms.ldap.ldap1.unmapped_groups_as_roles: false
```

参数	说明
type	设置域。此处必须设置为ldap。
order	域的优先级，数值越小，优先级越高。当配置中指定多个域时，建议配置此参数，系统会先访问order值较小的域。
url	指定LDAP服务器的URL及端口。ldap协议表示使用普通连接，端口为389；ldaps表示使用SSL安全连接，端口为636。 **注意** 新网络架构下需要配置为终端节点域名:端口，**终端节点域名**可在步骤一：获取终端节点域名（可选）中获取。本文以 `ep-bp1dhpobznlgjhj9****-cn-hangzhou-i.epsrv-bp1q8tcj2jjt5dwr****.cn-hangzhou.privatelink.aliyuncs.com:389` 为例。
bind_dn	用于绑定到LDAP并执行搜索的用户的DN，仅适用于用户搜索模式。
bind_password	用于绑定到LDAP目录的用户的密码。
user_search.base_dn	用户搜索的容器DN。
group_search.base_dn	用于搜索用户具有成员资格的容器DN。当此参数不存在时，Elasticsearch将搜索user_group_attribute指定的属性，来确定成员身份。
unmapped_groups_as_roles	默认false。如果设置为true，则任何未映射的LDAP组的名称都将用作角色名称分配给用户。

更多参数的详细信息，请参见Security settings in Elasticsearch。

步骤三：为域账号信息映射角色

1. 登录目标阿里云Elasticsearch实例的Kibana控制台，根据页面提示进入Kibana主页。

   登录Kibana控制台的具体操作，请参见登录Kibana控制台。

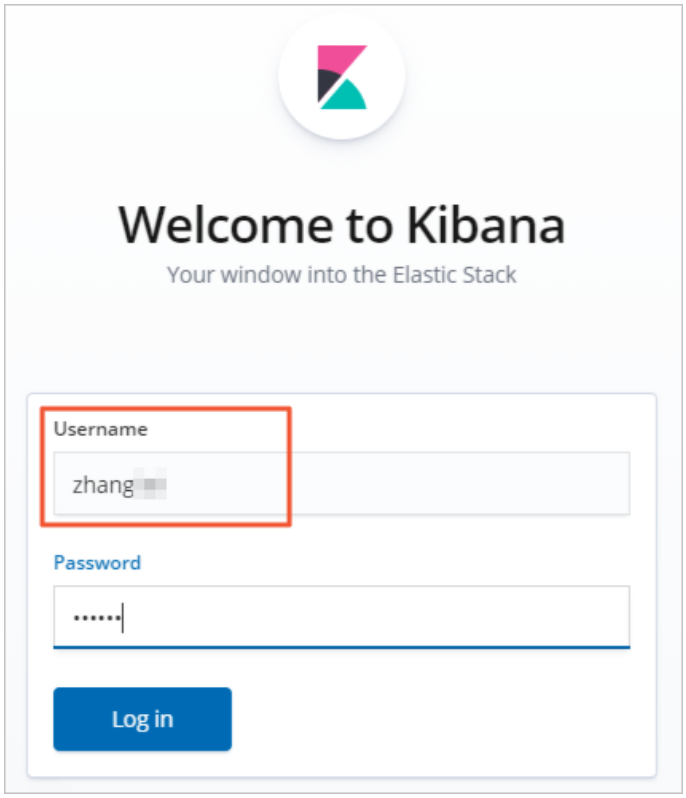
   > **说明** 本文以阿里云Elasticsearch 6.7.0版本为例，其他版本操作可能略有差别，请以实际界面为准。

2. 在左侧导航栏，单击**Dev Tools**。
3. 在**Console**中执行如下命令，映射LDAP下的zhang*账户为管理员角色。

```
POST _xpack/security/role_mapping/ldap_super_user1?pretty
{
  "roles": [ "superuser" ],
  "enabled": true,
  "rules": {
    "any": [
      {
        "field": {
          "username": "zhang*"
        }
      }
    ]
  }
}
```

步骤四：验证结果

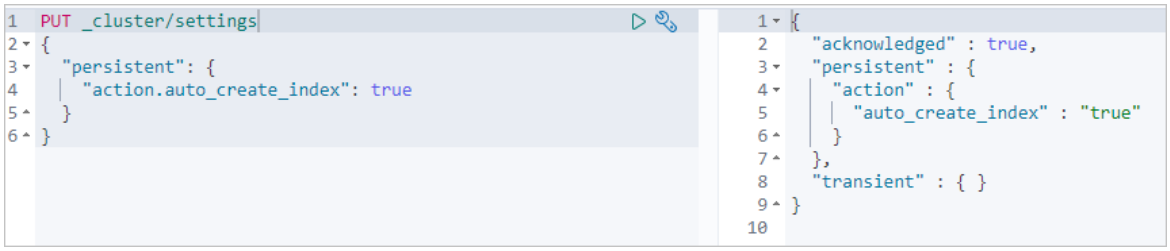
1. 使用已授权的zhang*账号登录阿里云Elasticsearch的Kibana控制台。



2. 在左侧导航栏，单击Dev Tools。
3. 在Console中执行如下命令，验证zhang*用户是否有修改集群配置的权限。

```
PUT _cluster/settings
{
  "persistent": {
    "action.auto_create_index": true
  }
}
```

授权成功后，预期结果如下。



9.3.3. 通过Elasticsearch X-Pack角色管理实现用户权限管控

当您需要设置集群、索引、字段或其他操作的访问权限时，可以通过Elasticsearch X-Pack的RBAC（Role-based Access Control）机制，为自定义角色分配权限，并将角色分配给用户，实现权限管控。Elasticsearch提供了多种内置角色，您可以在内置角色的基础上扩展自定义角色，以满足特定需求。本文介绍几种常见的角色配置，以及如何通过角色配置实现权限管控。

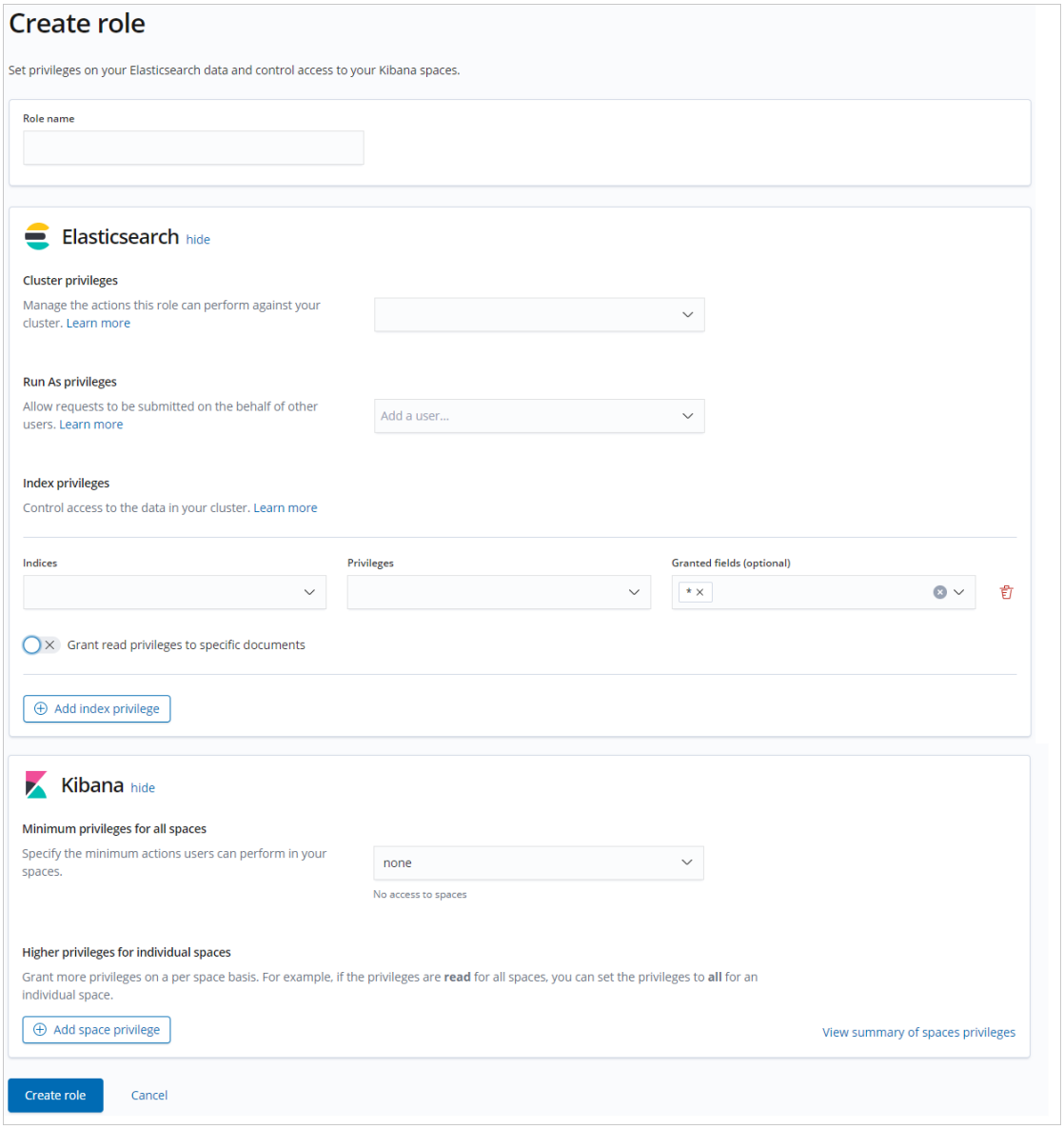
背景信息

- Elasticsearch支持X-Pack RBAC机制。详细信息，请参见User authorization。
- Elasticsearch支持多种安全认证功能。详细信息，请参见Elasticsearch身份认证和授权。

操作步骤

> 说明 本文操作以Elasticsearch 6.7.0版本为例，其他版本操作可能略有不同，具体以实际界面为准。

1. 创建角色。
   i. 登录Kibana控制台。
      具体步骤，请参见登录Kibana控制台。
   ii. 在左侧导航栏，单击**Management**。
   iii. 在**Security**区域，单击**Roles**。
   iv. 单击**Create role**，然后输入相关参数配置。



参数	说明
Role name	角色名称。
Cluster privileges	定义集群的操作权限，例如查看集群健康度和Settings、创建快照等。详细信息，请参见Cluster privileges。
Run As privileges	扮演该角色的用户，可选。如果此处未选择，可在创建用户时，为该用户指定对应角色，具体操作，请参见创建用户。

参数	说明
Index privileges	定义索引的操作权限，例如只读查看所有索引的所有字段（索引名设置为*后，授予read索引），索引名支持通配符（*）及正则表达式。详细信息，请参见Indices privileges。配置时，需要填写以下参数： ■ Indices：选择对应的索引模式。例如heartbeat-*。 说明 如果没有索引模式，请先在Management页面，单击Kibana中的Index Pattern，按照页面提示创建一个索引模式。 ■ Privileges：为角色分配的权限。 ■ Granted fields (optional)：授权的字段，可选。
Kibana privileges	定义Kibana操作权限。 注意 Kibana 7.0以下版本仅支持Base privileges，默认为所有空间授权；7.0及以上版本在Base privileges的基础上，还支持Feature privileges。即对Kibana特定功能授权，需要指定Kibana空间。

创建角色时，需要为该角色分配对应权限。本文的角色权限配置示例如下：

- 为普通用户授予指定索引的只读（Read）权限。设置用户仅能访问指定索引，不能进行其他操作。

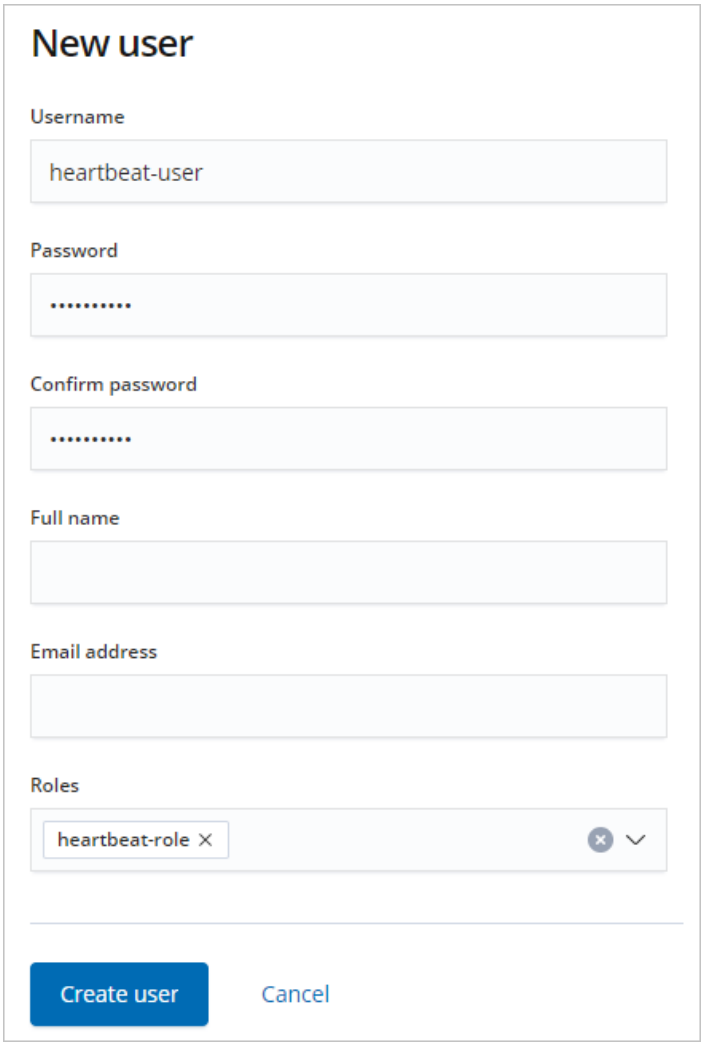
  详细信息，请参见配置索引只读权限。
- 为普通用户授予查看所有或部分Dashboard的权限。

  详细信息，请参见配置Dashboard操作权限。
- 为普通用户授予部分索引的读写权限，及所有集群的只读权限。例如查看集群健康度、快照及Settings，写入索引数据和更新索引Mapping等。

  详细信息，请参见配置索引读写和集群只读权限。

v. 单击Create role。

2. 创建用户，并为该用户分配对应角色，为其授予该角色拥有的权限。
   i. 在Kibana控制台的左侧导航栏，单击Management。
   ii. 在Security区域，单击Users。
   iii. 单击Create new user，然后输入相关参数配置。



参数	说明
Username	用户名称，用来登录Kibana控制台。自定义输入。
Password	该用户的密码，用来登录Kibana控制台。自定义输入。
Confirm password	确认密码，与Password保持一致。
Full name	用户全名，自定义输入。
Email address	用户的Email地址。
Roles	为用户分配角色。选择已创建的角色，或系统预置的角色，可选择多个。 **注意** 在创建角色时，如果已选择了对应用户，此处依然需要选择角色，否则登录时会报错。

iv. 单击Create user。

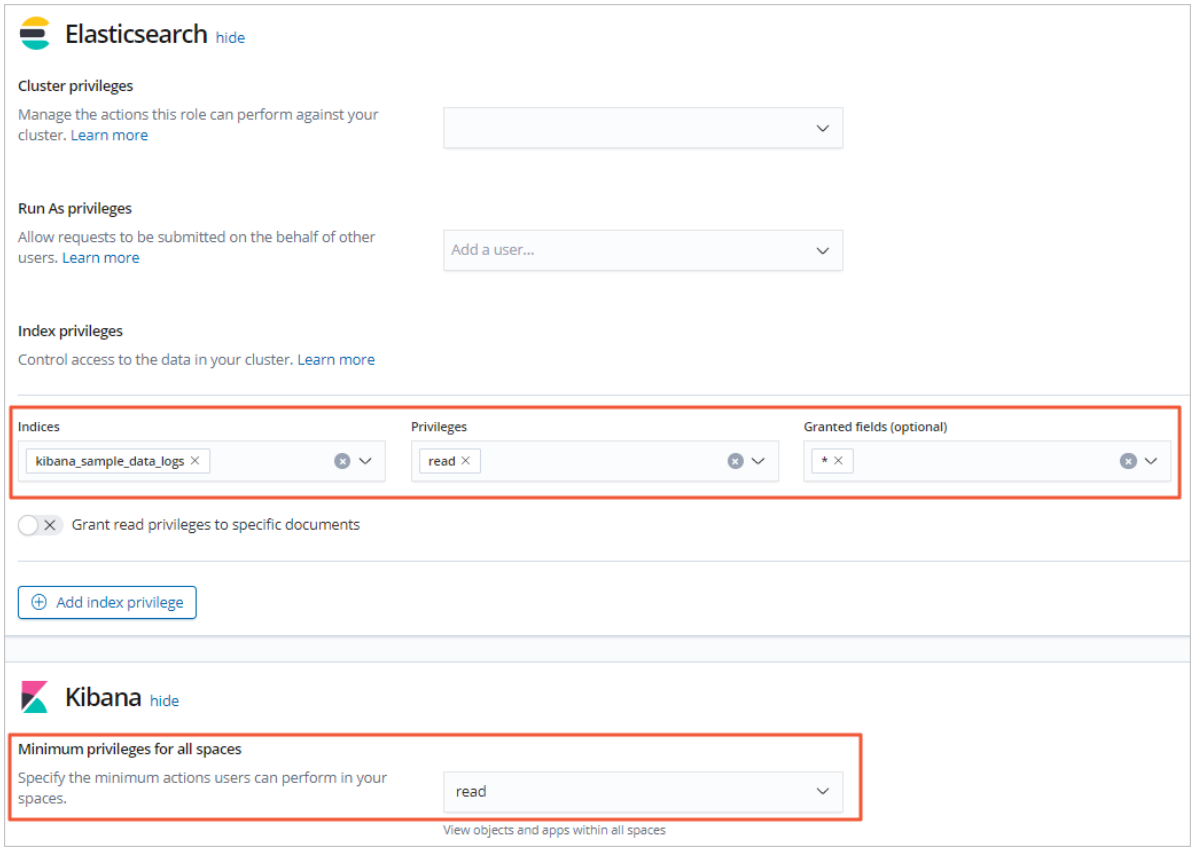
3. 通过自定义用户登录Kibana控制台，执行相关操作验证权限是否生效。

配置索引只读权限

- 场景描述

  为普通用户授予指定索引的只读权限。该用户可通过Kibana查询索引数据，但无权访问集群。

- 角色配置



权限说明

权限类型	权限Key	权限Value	描述
Index privileges	indices	kibana_sample_data_logs	指定索引名称，支持索引全名、别名、通配符及正则表达式。详细信息，请参见Indices Privileges。
	privileges	read	设置索引只读权限。只读权限包括count、explain、get、mget、scripts、search、scroll等操作权限。详细信息，请参见privileges-list-indices。
	Granted fields (optional)	*	索引字段。*表示索引的所有字段。

权限类型	权限Key	权限Value	描述
Kibana privileges	privileges	read	为所有空间授予Kibana只读权限。默认为none，表示所有空间无权限访问Kibana。 注意 Kibana 7.0以下版本仅支持Base privileges，默认为所有空间授权；7.0及以上版本在Base privileges的基础上，还支持Feature privileges。即对Kibana特定功能授权，需要指定Kibana空间。

- 验证

  通过普通用户登录Kibana控制台，执行读索引命令，返回结果正常。执行写索引命令，返回未授权的错误信息。

```
GET /kibana_sample_data_logs/_search
```

```
POST /kibana_sample_data_logs/_doc/1
{
    "productName": "testpro",
    "annual_rate": "3.22%",
    "describe": "testpro"
}
```

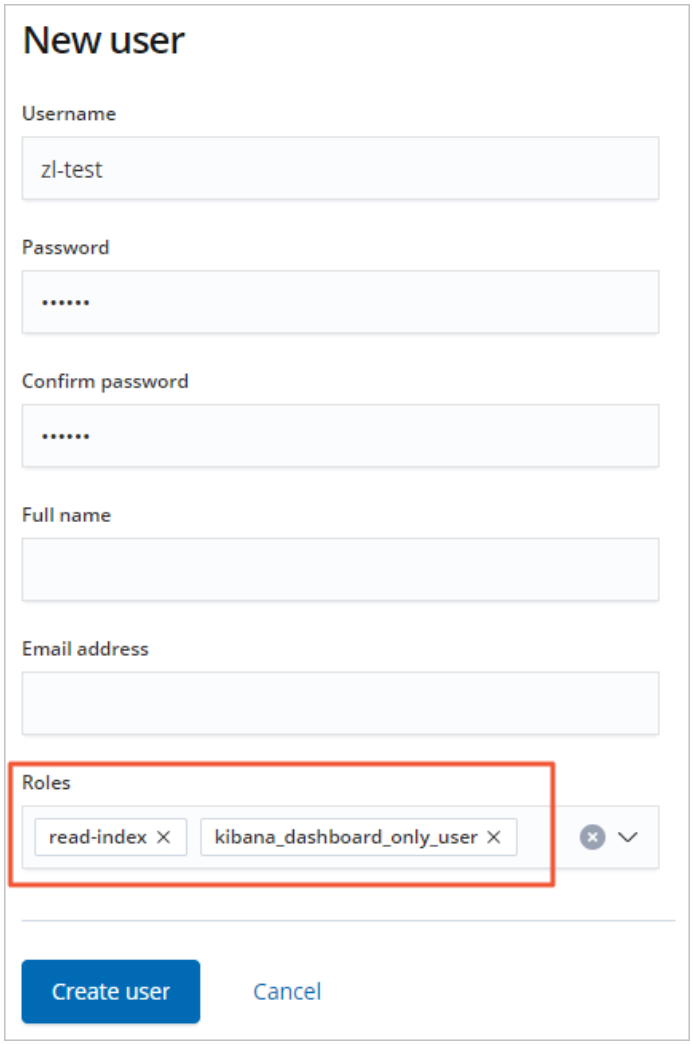
配置Dashboard操作权限

- 场景描述

  授予普通用户指定索引的只读权限，且可查看该索引对应的Dashboard数据。

- 角色配置

  在创建用户时，为该用户分配角色：read-index和kibana_dashboard_only_user。

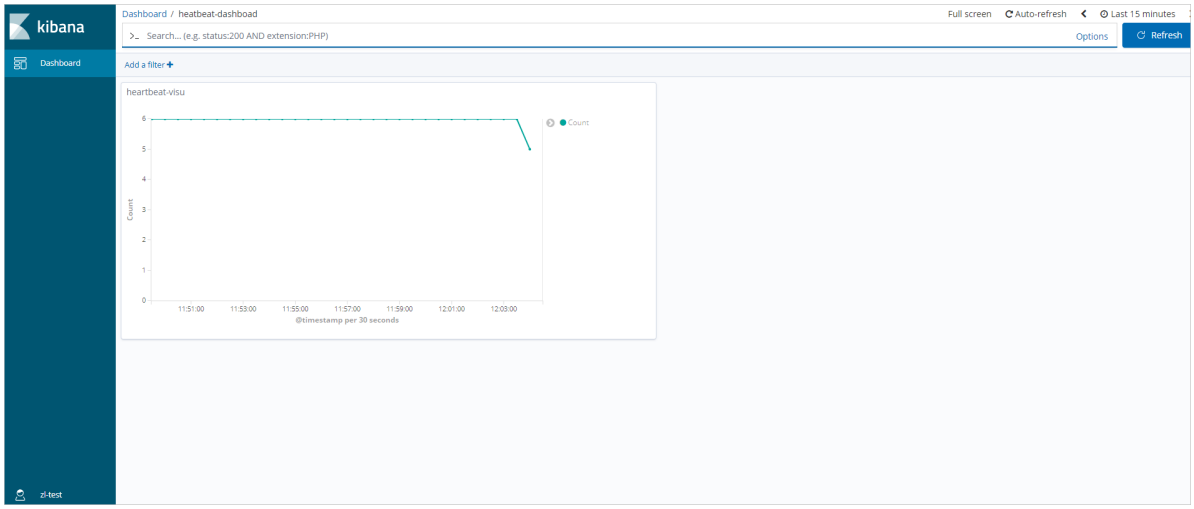


- read-index：自定义角色示例，需要您自行创建。该角色拥有指定索引的只读权限。
- kibana_dashboard_only_user：Kibana内置角色，该角色拥有查看指定索引Dashboard数据的权限。

> 注意
>
> - 在Kibana 7.0及以上版本中，kibana_dashboard_only_user角色已经被废弃。如果要查看指定索引的Dashboard，只需要为该索引配置读权限，详细信息，请参见配置索引只读权限。
> - kibana_dashboard_only_user角色与自定义角色配合使用可应用于很多场景。如果您仅需为自定义角色设定Dashboards only roles功能，可在Management页面的Kibana区域，单击Advanced Settings，找到Dashboard部分，绑定自定义角色（默认是kibana_dashboard_only_user角色）。

- 验证

通过普通用户登录Kibana控制台，可查看对应索引的Dashboard大盘。

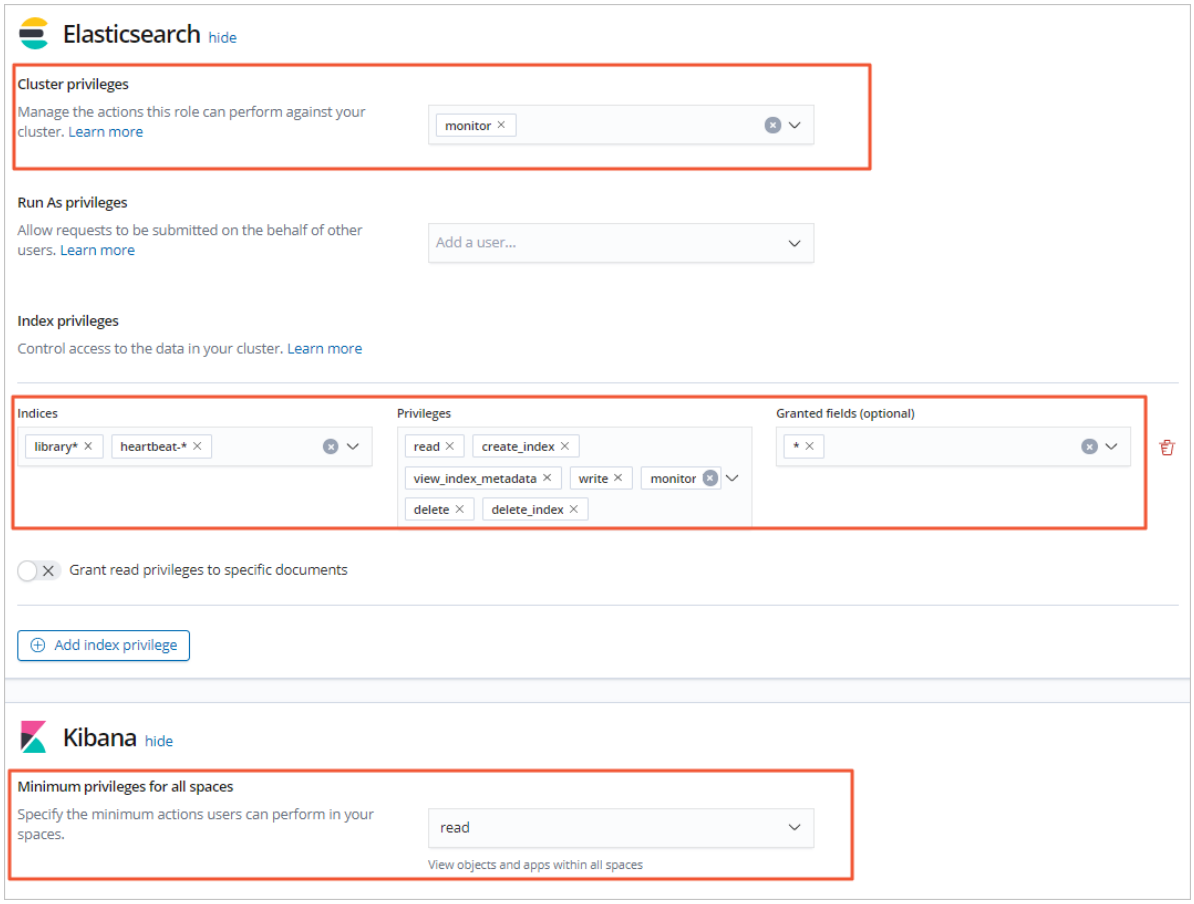


配置索引读写和集群只读权限

- 场景描述

  为普通用户授予指定索引的读、写和删除权限，以及集群和Kibana的只读权限。

- 角色配置

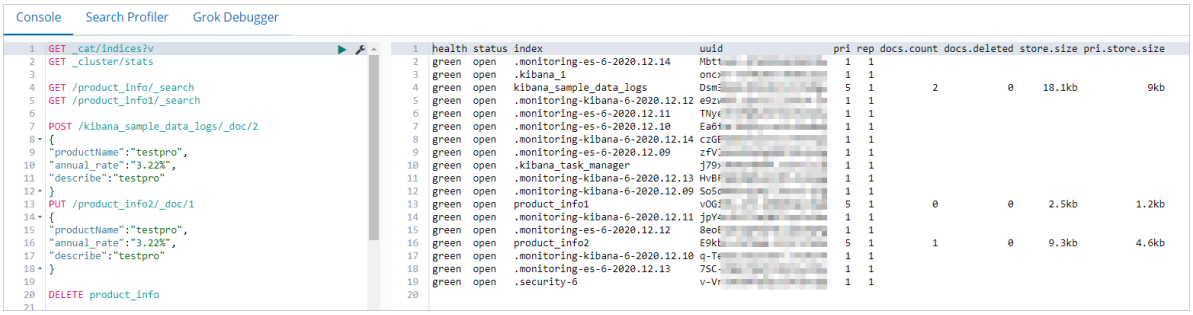


权限说明

权限类型	权限Key	权限Value	描述
Cluster privileges	**cluster**	monitor	集群只读权限。例如查看集群的健康度、状态、热线程、节点信息、阻塞的任务等。
Index privileges	indices	heartbeat-*,library*	指定索引名称，支持索引全名、别名、通配符及正则表达式。详细信息，请参见roles-indices-priv。
	privileges	read	设置索引只读权限。只读权限包括count、explain、get、mget、scripts、search、scroll等操作权限。详细信息，请参见privileges-list-indices。
		create_index	创建索引权限。如果在创建索引时，定义了索引别名，还需要授予索引manage权限。 **注意** 索引别名需要同时满足indices下定义的匹配规则。
		view_index_metadata	索引元数据的只读权限，包括aliases、aliases exists、get index、exists、field mappings、mappings、search shards、type exists、validate、warmers、settings和ilm。
		write	对文档执行所有写操作的权限，包括index、update、delete、bulk和更新mapping操作。与create和index权限相比，该权限的覆盖面更大。
		monitor	监控所有操作的权限，包括index recovery、segments info、index stats和status。
		delete	删除索引文档权限。
		delete_index	删除索引权限。
	granted fields	*	待授权的索引字段，*表示索引的所有字段。
Kibana privileges	privileges	read	为所有空间授予Kibana只读权限。默认为none，表示所有空间无权限访问Kibana。 **注意** Kibana 7.0以下版本仅支持Base privileges，默认为所有空间授权；7.0及以上版本在Base privileges的基础上，还支持Feature privileges。即对Kibana特定功能授权，需要指定Kibana空间。

- 验证

  通过普通用户登录Kibana控制台，执行如下命令均正常。



  - 查看集群中所包含索引的详细信息

    ```
    GET /_cat/indices?v
    ```

  - 查看集群状态

    ```
    GET /_cluster/stats
    ```

  - 查询product_info索引中的数据

    ```
    GET /product_info/_search
    ```

  - 查询product_info1索引中的数据

    ```
    GET /product_info1/_search
    ```

  - 通过POST方式，向kibana_sample_data_logs索引中写入数据

    ```
    POST /kibana_sample_data_logs/_doc/2
    {
        "productName": "testpro",
        "annual_rate": "3.22%",
        "describe": "testpro"
    }
    ```

  - 通过PUT方式，向product_info2索引中写入数据

    ```
    PUT /product_info2/_doc/1
    {
        "productName": "testpro",
        "annual_rate": "3.22%",
        "describe": "testpro"
    }
    ```

  - 删除product_info索引

    ```
    DELETE product_info
    ```

9.3.4. 配置Active Directory身份认证

本文介绍如何基于阿里云Elasticsearch配置活动目录AD（Active Directory）身份认证，以实现AD域下相应角色的用户访问阿里云Elasticsearch。

前提条件

您已完成以下操作：

- 创建阿里云Elasticsearch实例。

  具体操作，请参见创建阿里云Elasticsearch实例。本文以7.10版本实例为例。

- 如果您的阿里云Elasticsearch实例是2020年10月及之后创建的，由于网络架构的调整，您需要完成以下操作：

  i.

  ii. 配置负载均衡实例。详情请参见步骤二：配置负载均衡实例。

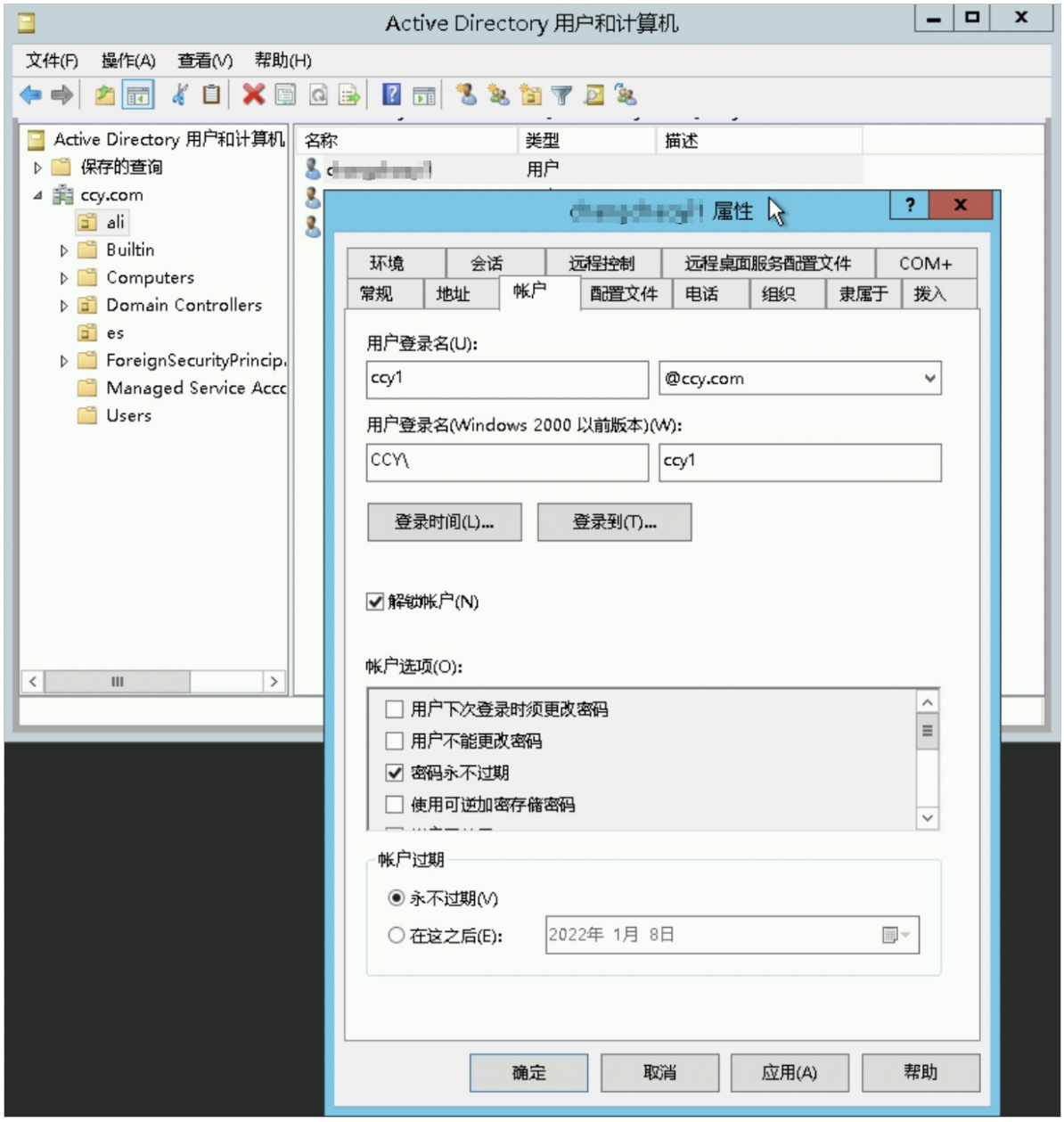
  iii.

  iv.

  v.

- 准备与阿里云Elasticsearch相同专有网络下的AD域环境和数据，本文以Windows Server 2012为例。

  具体操作，请参见ECS实例搭建Windows系统AD域。本文配置的用户名称为ccy1，根域为ccy.com，如下图所示。



使用限制

自2020年10月起，阿里云Elasticsearch对不同地域进行了网络架构的调整，对创建的实例有以下影响：

- 2020年10月之前创建的实例均在旧网络架构下，即Elasticsearch实例处于用户VPC下，如果需要访问公网，可以直接使用SNAT功能或自建Nginx代理。
- 2020年10月及之后创建的实例均在新网络架构下，即Elasticsearch实例处于Elasticsearch服务VPC下，X-Pack Watcher功能受到网络限制，为解决此问题，阿里云Elasticsearch提供了实例私网连接方案，详细信息请参见配置实例私网连接。如果您还需要将报警信息推送至公网环境，在通过实例私网连接打通Elasticsearch服务VPC和用户VPC的基础上，还需对负载均衡后端服务配置Nginx代理或开启SNAT功能实现公网信息推送。

> **注意** 实例私网连接方案是新网络架构下X-Pack Watcher、reindex、LDAP和AD（Active Directory）身份认证等功能受限的唯一解决方案，为保证功能使用不受影响，请严格按照文档配置。

-
- 2020年10月及之后创建的实例均为新网络架构下，AD功能受到网络限制，为解决此问题，您可以使用PrivateLink进行VPC网络打通，具体配置请参见配置实例私网连接。如果您需要访问公网，则需要配置Nginx代理进行请求转发。
- 旧网络架构下，阿里云Elasticsearch仅支持单可用区的AD认证，不支持多可用区。新网络架构不受此限制。

操作流程

1. 步骤一：配置AD认证
2. 步骤二：为域账号映射角色
3. 步骤三：验证结果

步骤一：配置AD认证

您可以通过Elasticsearch的安全功能与AD域通信，实现用户身份认证。安全功能基于LDAP与AD域进行通信，因此active_directory域类似于ldap域。与LDAP目录一样，AD域分层存储用户和组。AD域通过发送LDAP绑定请求，验证用户的身份。验证后，AD域会通过搜索查找对应用户在Active Directory中的条目。一旦找到该用户，AD域就会从Active Directory中用户条目的tokenGroups属性中检索该用户的组成员身份。详细信息，请参见Configuring an Active Directory realm。

如果您的目标阿里云Elasticsearch为6.x版本，可参见配置YML参数，在目标Elasticsearch实例的YML文件中添加如下配置，设置对应用户的AD认证。如果为7.x版本，则需要通过提交工单，将相关配置提交给技术人员帮您配置。

```
xpack.security.authc.realms.active_directory.my_ad.order: 0
xpack.security.authc.realms.active_directory.my_ad.domain_name: ccy.com
xpack.security.authc.realms.active_directory.my_ad.url: ldap://ep-bp1i321219*********-cn-han
gzhou-h.epsrv-bp15571d5ps*********.cn-hangzhou.privatelink.aliyuncs.com:389
xpack.security.authc.realms.active_directory.my_ad.bind_dn: ccy1@ccy.com
xpack.security.authc.realms.active_directory.my_ad.secure_bind_password: your_password
```

参数	说明
order	进行身份验证时，检查已配置的AD域的顺序。
domain_name	根域的名称。
url	AD域与ECS实例进行私网连接的URL及端口号，详细信息请参见Configuring an Active Directory realm。 **注意** 新网络架构下需要设置为 `ldap://<终端节点域名>:<端口>`，本文以ldap://ep-bp1i321219*********-cn-hangzhou-h.epsrv-bp15571d5ps*********.cn-hangzhou.privatelink.aliyuncs.com:389为例。
bind_dn	执行所有AD搜索请求的用户。
secure_bind_password	验证AD域中身份信息的密码。

步骤二：为域账号映射角色

1. 登录目标Elasticsearch实例的Kibana控制台。

   具体操作，请参见登录Kibana控制台。

   > **说明** 本文以阿里云Elasticsearch 7.10.0版本为例，其他版本操作可能略有差别，请以实际界面为准。

2. 根据页面提示进入Kibana主页，单击右上角的**Dev tools**。
3. 在**Console**页签，执行如下命令，将AD域下的ccy1用户设置为管理员角色。

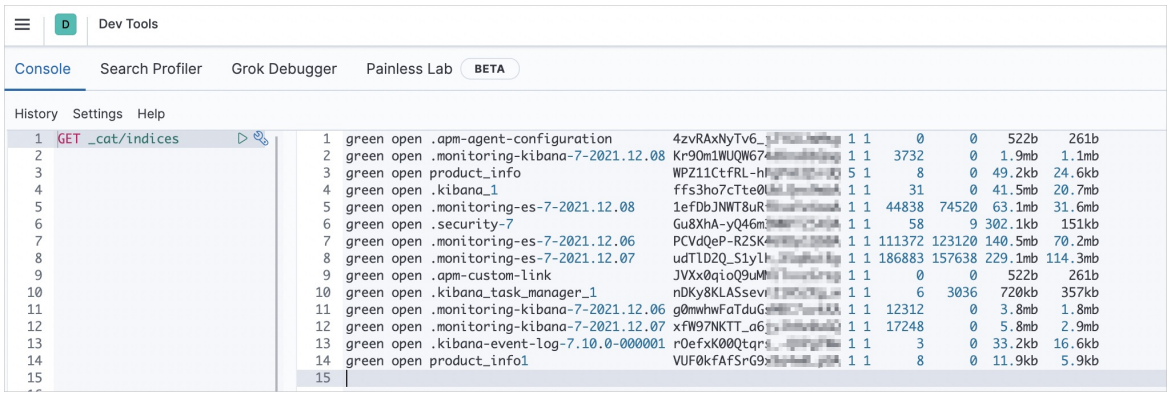
```
PUT /_security/role_mapping/basic_users
{
  "roles": [ "superuser" ],
  "enabled": true,
  "rules": {
    "any": [
      {
        "field": {
          "groups": "cn=ali,dc=ccy,dc=com"
        }
      },
      {
        "field": {
          "dn": "cn=ccy1,cn=ali,dc=ccy,dc=com"
        }
      }
    ]
  }
}
```

步骤三：验证结果

1. 使用已授权的ccy1用户登录目标Elasticsearch的Kibana。
2. 根据页面提示进入Kibana主页，单击右上角的**Dev tools**。
3. 在**Console**页签，执行如下命令，验证ccy1用户是否有执行对应操作的权限。

```
GET _cat/indices
```

如果ccy1用户有权限，会返回如下结果，说明AD域下ccy1用户权限设置成功。



9.4. 集群安全配置

9.4.1. 配置阿里云IDaaS单点登录Kibana控制台

本文介绍如何通过配置安全断言标记语言SAML（Security Assertion Markup Language）的身份提供者IDaaS（Alibaba Cloud Identity as a Service）和服务提供者Elasticsearch及Kibana，实现单点登录Kibana控制台。

背景信息

Elasticsearch支持SAML单点登录（SSO）到Kibana。在SAML术语中，Elasticsearch和Kibana作为服务提供者SP（Service Provider），支持SAML 2.0 协议的Web浏览器SSO和SAML 2.0 Single Logout配置，这使您能够使用任何符合SAML 2.0 的身份提供者IDP（IDentity Provider）访问阿里云Elasticsearch和Kibana，例如阿里云应用身份服务IDaaS（Alibaba Cloud Identity as a Service）、联合身份验证服务ADFS（Active Directory Federation Services）等。本文以IDaaS为例进行介绍。

本文中涉及的术语解释如下：

- 阿里云应用身份服务IDaaS：是阿里云为企业用户提供的一套集中式身份、权限、应用管理服务，IDaaS支持多种产品，例如EIAM、CIAM等。详细信息请参见什么是IDaaS。
- 安全断言标记语言SAML：基于XML协议，使用包含断言（Assertion）的安全令牌，在身份提供者IDP和服务提供者SP之间传递身份信息，实现基于网络跨域的单点登录。SAML协议是成熟的认证协议，在公有云和私有云中有非常广泛的运用。详细信息请参见SAML。
- 单点登录SSO（Single Sign On）：是指在多个应用系统中，用户只需要登录一次，就可以访问所有相互信任的应用系统。详细信息请参见单点登录和身份联邦。

目前，SAML单点登录仅支持后端手动配置，您需要先参考本文，在测试环境配置并测试成功后，再提交工单将配置提供给技术人员帮您配置。

> **说明** 本文包含配置IDaaS SAML应用（用户侧）和创建自定义角色并配置elastic SAML（后端）两部分。其中配置IDaaS SAML应用（用户侧）需要您手动操作，创建自定义角色并配置elastic SAML（后端）需要阿里云Elasticsearch技术人员在后端为您配置。本文给出后端操作旨在帮助您测试配置并了解配置原理。

前提条件

- 创建阿里云Elasticsearch实例7.10版本，并开启HTTPS访问协议。

  创建实例的具体操作，请参见创建阿里云Elasticsearch实例。本文以7.10版本实例为例，其他版本的操作配置可能存在差异，具体以实际界面为准。

  开启HTTPS的具体操作，请参见使用HTTPS协议。

  > **注意** 只有包含协调节点的实例才支持开启HTTPS，请确保实例中已包含协调节点。

- 开通IDaaS服务的EIAM实例。

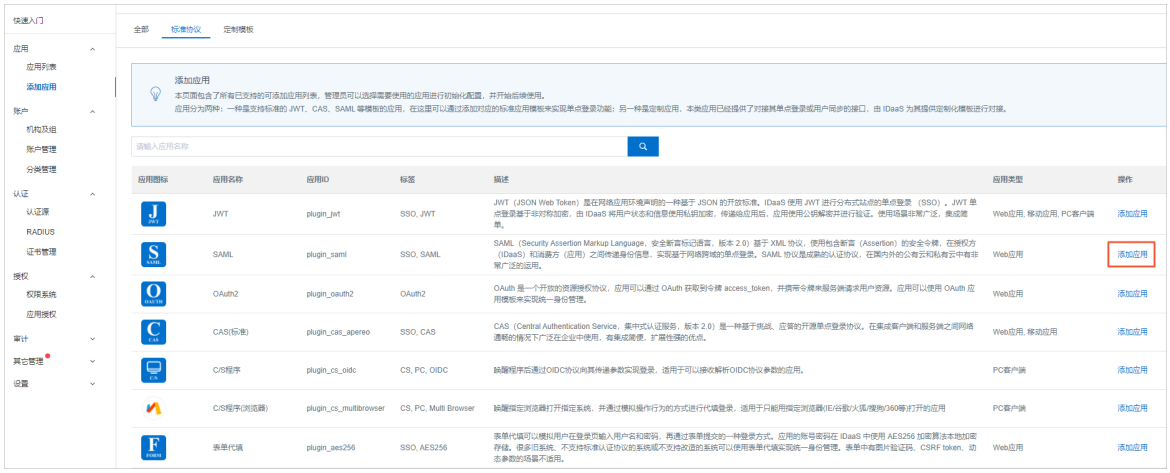
  具体操作，请参见开通和试用流程。

  > **说明** elastic仅支持SAML身份验证的HTTP-Redirect binding方式，不支持HTTP-POST binding及其他方式，因此只需保证PC端可以访问IDP及SP服务即可。

配置IDaaS SAML应用（用户侧）

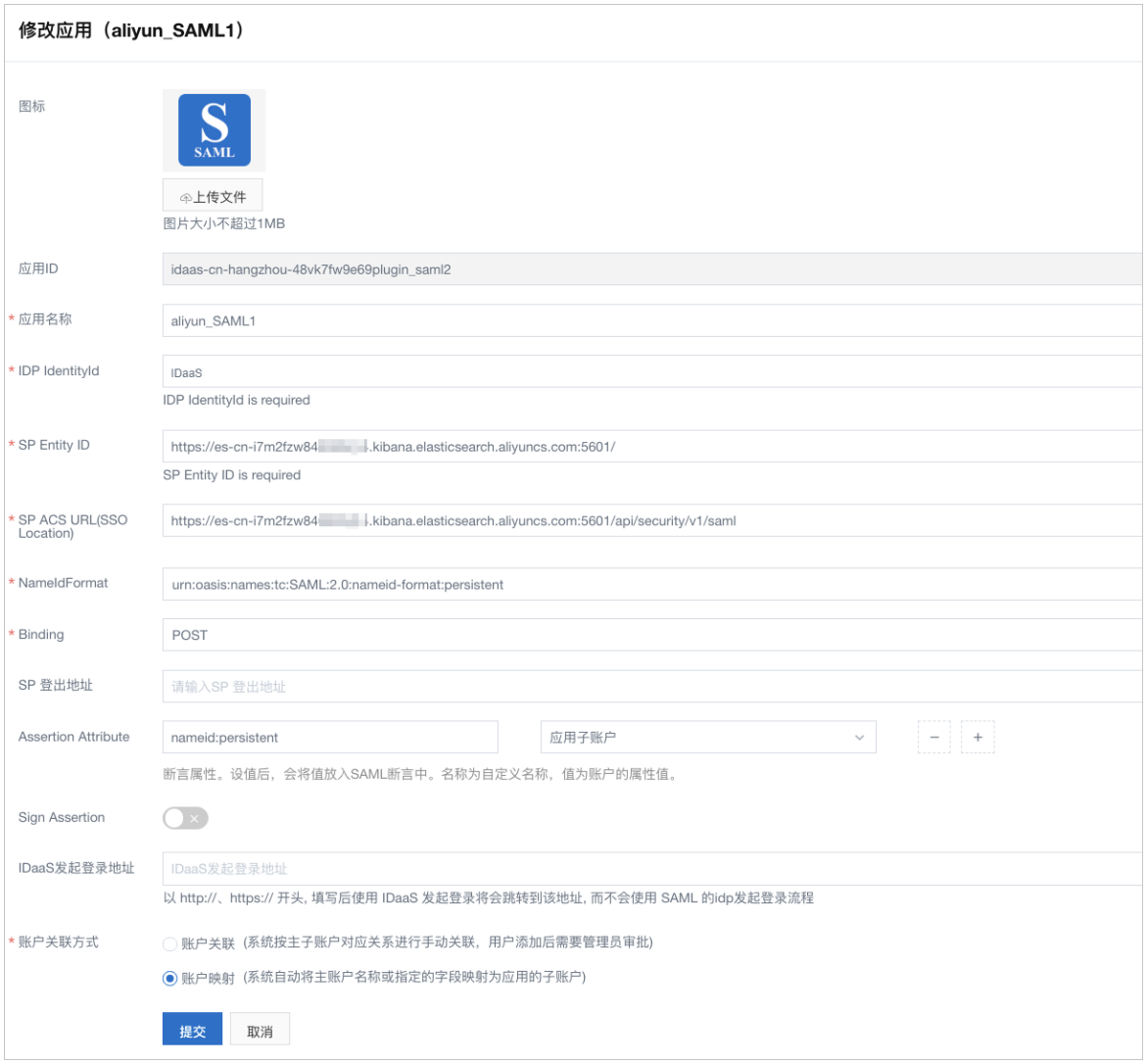
1. 登录IDaaS管理平台，进入EIAM实例，添加SAML应用。

   具体操作请参见添加应用。



2. 在**添加应用（SAML）**面板，单击目标SigningKey右侧的**选择**，配置IDP认证ID及SP认证信息。

> 说明 如果没有SigningKey，需要先添加或导入一个SigningKey，具体操作请参见Gitlab对接（SAML）。



您需要配置以下参数，其他参数保持默认。

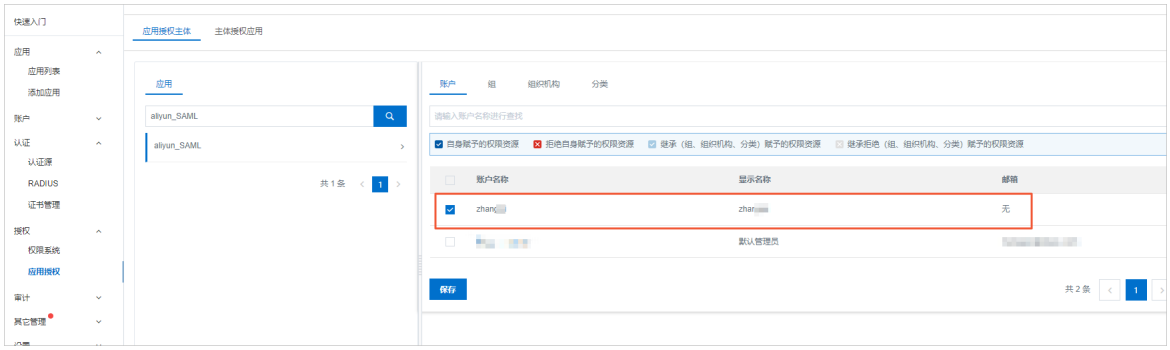
参数	说明
应用名称	自定义SAML应用的名称。
IDP IdentityId	在IDaaS中设置的认证参数，需要将此参数配置到SP中，可设置为IDaaS。
SP Entity ID	服务提供者SP的访问URL。本文的服务提供者为Kibana，因此需要配置为Kibana服务的基础访问URL，并使用HTTPS协议。
SP ACS URL(SSO Location)	断言消费服务ACS端点，一般为Kibana的URL，用来接收来自IDP的身份验证消息。此ACS端点仅支持SAML HTTP-POST绑定，通常配置为 `${kibana-url}/api/security/v1/saml` ，其中 `${kibana-url}` 为Kibana的基础访问URL。
NameIdFormat	名称标识格式类型。请设置为**urn:oasis:names:tc:SAML:2.0:nameid-format:persistent**。
Binding	选择默认的**POST**方式。

参数	说明
Assertion Attribute	断言属性。名称可自定义，值需要选择**应用子账户**。
账户关联方式	选择**账户映射**。

3. 单击**提交**。
4. 在**系统提示**弹出框中，单击**立即授权**，为应用授权。

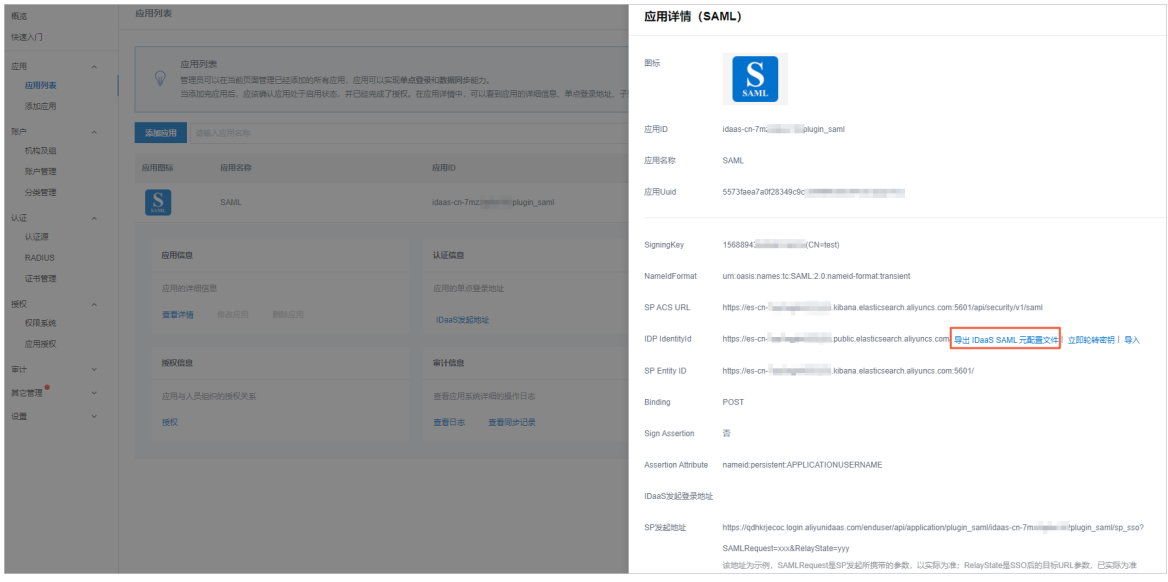
> 注意 应用授权之前，请确保已经同步或创建应用侧账户信息，详细信息请参见IDaaS账户。

5. 参考下图，选择对应账户，单击**保存**，确认后完成授权。



6. 从已创建的IDaaS应用中导出IDaaS SAML元配置文件。

   具体操作请参见Gitlab对接（SAML)。

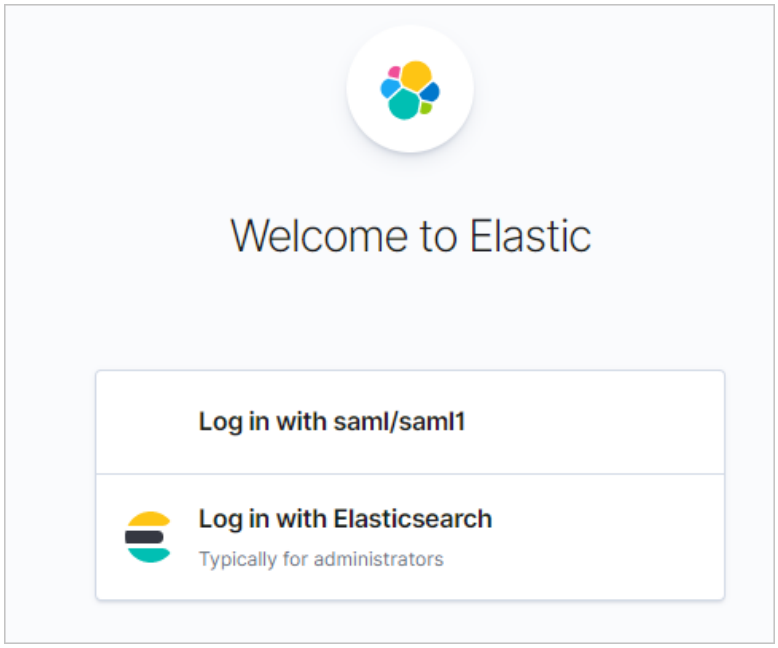


7. 将导出的文件通过提交工单的形式提交给阿里云Elasticsearch技术支持人员。

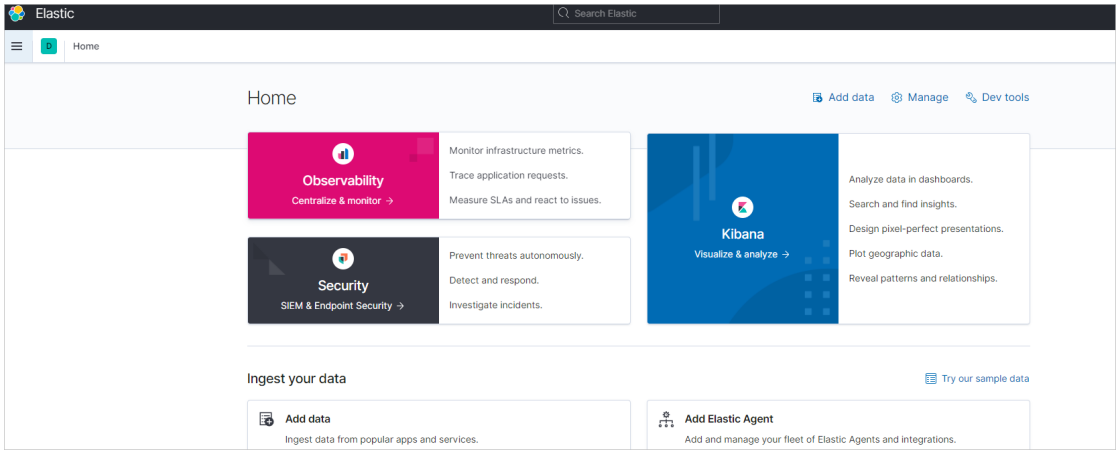
   提交后，技术人员会按照创建自定义角色并配置elastic SAML（后端）中的操作为您配置elastic SAML。您也可以参考这部分内容，在您的自建Elasticsearch环境中进行测试。

8. 待技术人员配置完成后，验证Kibana单点登录。

i. 参见登录Kibana控制台，进入Kibana控制台登录页面，单击Log in with saml/saml1。

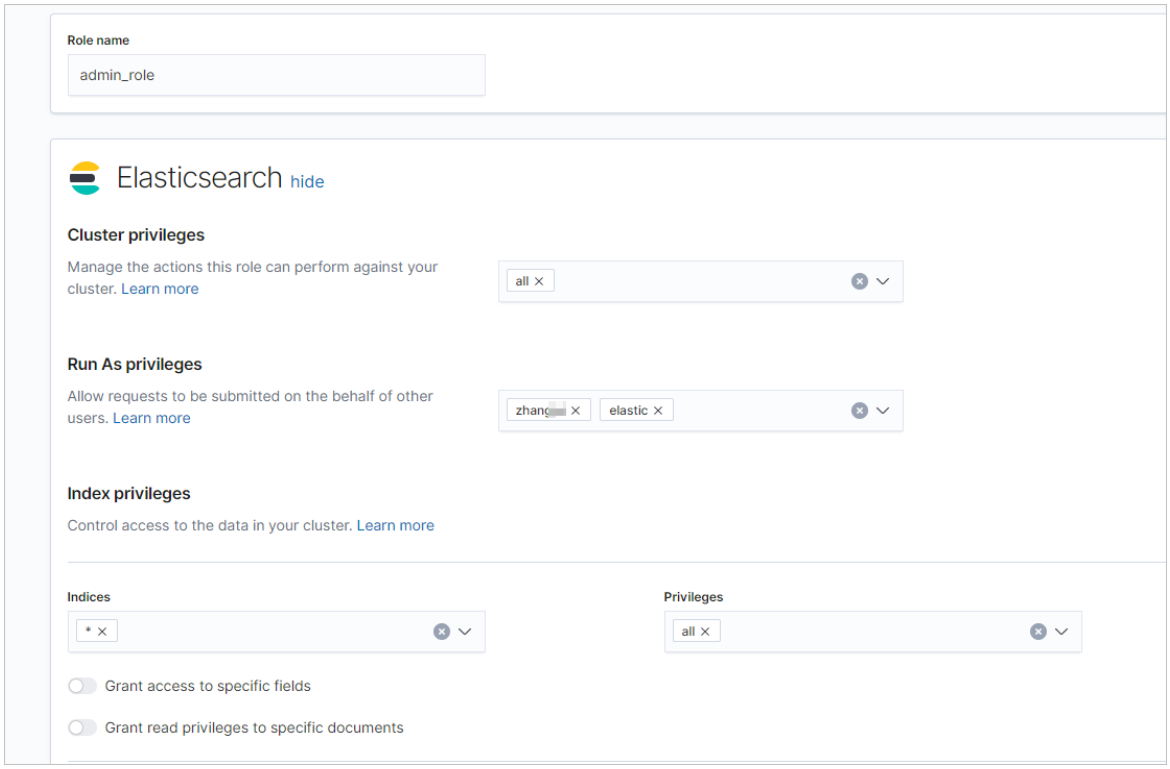


ii. 输入IDaaS关联的账号，单击**提交**。
登录成功后的页面如下图。



创建自定义角色并配置elastic SAML（后端）

1. 登录目标实例的Kibana控制台。
2. 创建自定义角色。



3. 将角色与SAML进行映射。

```
PUT /_security/role_mapping/idaas-test
{
  "roles": [ "admin_role" ],
  "enabled": true,
  "rules": {
    "field": { "realm.name": "saml1" }
  }
}
```

> **说明** roles参数值需要替换为上一步创建的角色名称。

4. 将配置IDaaS SAML应用（用户侧）中导出的IDaaS SAML元配置文件上传至elasticsearch的*config/saml*路径下。
5. 分别在Elasticsearch和Kibana的YML配置文件中，添加SAML信息。

> **注意** YML参数信息需要与配置IDaaS SAML应用（用户侧）中配置的SAML信息保持一致。

   - elasticsearch.yml配置

```
#elasticsearch.yml配置
xpack.security.authc.token.enabled: 'true'
xpack.security.authc.realms.saml.saml1:
  order: 0
  idp.metadata.path: saml/metadata.xml
  idp.entity_id: "https://es-cn-n6xxxxxx1d.elasticsearch.aliyuncs.com/"
  sp.entity_id: "https://es-cn-n6xxxxxx1d.kibana.elasticsearch.aliyuncs.com:5601/"
  sp.acs: "https://es-cn-n6xxxxxx1d.kibana.elasticsearch.aliyuncs.com:5601/api/security/v1/saml"
  attributes.principal: "nameid:persistent"
  attributes.groups: "roles"
```

参数	说明
xpack.security.authc.token.enabled	是否开启Token服务。需要设置为true，才可配置SAML单点登录，详细信息请参见saml-enable-token。
xpack.security.authc.realms.saml.saml1	定义的身份认证领域，本文示例为saml1。领域的详细信息请参见Realms。
order	领域优先级，数值越低，优先级越高。
idp.metadata.path	IDP元数据文件路径。
idp.entity_id	IDP使用的身份标识符，和元数据文件中的EntityID相匹配。
sp.entity_id	Kibana实例唯一标识符，如果将Kibana添加为IDP的服务提供商，需要设置此值，推荐配置为Kibana的URL。 注意 请确保参数值与业务环境保持一致。如果您配置了反向代理访问Kibana，而不是直接通过URL访问，需要将该参数配置为反向代理的地址和端口。
sp.acs	断言消费服务ACS端点，一般为Kibana的URL，用来接收来自IDP的身份验证消息。此ACS端点仅支持SAML HTTP-POST绑定，通常配置为 `${kibana-url}/api/security/v1/saml` ，其中 `${kibana-url}` 为Kibana的基础访问URL。
sp.logout	Kibana接收来自IDP的注销信息的URL，类似sp.acs，需要设置为 `${kibana-url}/logout` ，其中 `${kibana-url}` 为Kibana的基础访问URL。
attributes.principal	断言信息，具体请参见Attribute mapping。
attributes.groups	断言信息，具体请参见Attribute mapping。

- kibana.yml配置

```
# kibana配置
xpack.security.authc.providers:
  saml.saml1:
    order: 0
    realm: "saml1"
  basic.basic1:
    order: 1
    icon: "logoElasticsearch"
    hint: "Typically for administrators"
```

参数	说明
xpack.security.authc.providers	添加SAML提供程序，以设置Kibana使用SAML SSO作为身份验证方法。
xpack.security.authc.providers.saml.<provider-name>.realm	设置SAML领域名称。请将<provider-name>替换为elasticsearch.yml中配置的领域名称。本文示例为saml1。
xpack.security.authc.providers.basic.basic1	Kibana配置SAML后，仅支持符合SAML身份验证的用户登录Kibana。为了便于在Kibana登录页面支持basic身份登录（尤其在测试环节，可能需要使用elastic用户名和密码登录集群，创建角色及角色映射），可指定basic.basic1部分配置。指定后，Kibana登录页面会添加基本身份登录入口，详细信息请参见Authentication in kibana。 **说明** 如果用户不需要使用basic身份登录Kibana控制台，可在kibana.yml中不设置basic身份登录。

9.5. 日志全观测应用

9.5.1. 日志全观测应用概述

在复杂的业务场景下，往往存在着结构分散、种类多样、规模庞大的各类指标，日志和APM数据。针对这些数据，可以根据业务的需求和环境，选择对应的方案，分析并定位全链路的异常问题并进行日常运维。本文对阿里云Elasticsearch的日志全观测应用进行了汇总。

相关文档	描述
通过Elastic实现Kubernetes容器全观测	Elastic可观测性是通过Kibana可视化能力，将日志、指标及APM数据结合在一起，实现对容器数据的观测和分析。当您的应用程序以Pods的方式部署在Kubernetes中时，可以通过Kibana查看Pods生成的日志、主机和网络上的事件指标及APM数据，逐步缩小排查范围进行故障排查。
基于Indexing Service实现数据流管理	当面对海量时序数据和日志数据写入出现性能瓶颈时，您可以根据业务需求选择使用阿里云Elasticsearch 7.10日志增强版Indexing Service系列，此功能基于读写分离架构以及写入按量付费的Serverless模式，实现了Elasticsearch集群的云端写入托管和降本提效的目标。
通过OpenStore实现海量数据存储	当您面临着查询复杂度大、海量存储成本高的问题时，可通过阿里云Elasticsearch的OpenStore存储功能，实现基于计算存储分离的超低成本的弹性存储，即根据实际数据的存储量按量计费，无须提前预留集群存储容量。

9.5.2. 通过Elastic实现Kubernetes容器全观测

Elastic可观测性是通过Kibana可视化能力，将日志、指标及APM数据结合在一起，实现对容器数据的观测和分析。当您的应用程序以Pods方式部署在Kubernetes中，可以在Kibana中查看Pods生成日志、主机和网络上的事件指标及APM数据，逐步缩小排查范围进行故障排查。本文介绍具体的实现方法。

前提条件

- 创建阿里云Elasticsearch实例，版本为6.8，并开启白名单和自动创建索引功能。

  具体操作，请参见创建阿里云Elasticsearch实例、配置实例公网或私网访问白名单和配置YML参数。
- 创建ACK集群，并运行Pod服务。本文的测试场景使用的Kubernetes版本为1.18.8-aliyun.1，ECS规格为2C8G。

  具体操作，请参见创建Kubernetes托管版集群。
- 配置kubectl客户端，可通过kubectl连接Kubernetes集群。

  具体操作，请参见通过kubectl工具连接集群。

背景信息

本文介绍如何使用Elastic实现对Kubernetes容器的全方位检测，具体内容如下：

- 通过Metricbeat实现指标采集
- 通过Filebeat实现日志采集
- 通过Elastic APM实现应用程序性能监测

关于Metricbeat、Filebeat及APM更多的特性说明，请参见Infrastructure monitoring、Log monitoring和Elastic APM。

通过Metricbeat实现指标采集

在Kubernetes上部署Metricbeat，有以下两种方式：

- DaemonSet：保证每个节点运行一个Pod，可以实现对Host指标、System指标、Docker统计信息以及Kubernetes上运行的所有服务指标的采集。
- Deployment：部署单个Metricbeat实例，该实例用于检索整个集群的唯一指标，如kubernetes event或者kube-state-metrics。

> 注意
>
> - 本文以同时使用DaemonSet和Deployment的部署方式为例介绍如何部署Metricbeat容器，您也可以仅使用DaemonSet方式或Deployment方式进行部署。
> - Metricbeat依赖kube-state-metrics监控，部署前需要确保已完成kube-state-metrics的部署。阿里云ACK容器默认在arms-prom命名空间下部署了kube-state-metrics监控。

1. 通过kubectl访问云容器，下载Metricbeat配置文件。

   ```
   curl -L -O https://raw.githubusercontent.com/elastic/beats/6.8/deploy/kubernetes/metricbeat-kubernetes.yaml
   ```

2. 修改Metricbeat配置文件。

> **注意**
>
> 官方下载的YML文件中，DaemonSets和Deployments的资源使用extensions/v1beta1，而v1.18及以上版本的Kubernetes，DaemonSets、Deployments和Replicasets资源的extensions/v1beta1 API将被废弃，请使用apps/v1。

i. 修改kind: Deployment和kind: DaemonSet下的配置信息。

- 修改环境变量，具体内容如下：

```
env:
- name: ELASTICSEARCH_HOST
  value: es-cn-nif23p3mo0065****.elasticsearch.aliyuncs.com
- name: ELASTICSEARCH_PORT
  value: "9200"
- name: ELASTICSEARCH_USERNAME
  value: elastic
- name: ELASTICSEARCH_PASSWORD
  value: ****
- name: KIBANA_HOST
  value: es-cn-nif23p3mo0065****-kibana.internal.elasticsearch.aliyuncs.com
- name: KIBANA_PORT
  value: "5601"
```

> **注意** 下载的Metricbeat配置文件中默认未定义Kibana相关变量，可以通过容器env传入变量信息。

参数	说明
ELASTICSEARCH_HOST	阿里云Elasticsearch实例的私网地址。
ELASTICSEARCH_PORT	阿里云Elasticsearch实例的私网端口。
ELASTICSEARCH_USERNAME	阿里云Elasticsearch的用户名，默认值elastic。
ELASTICSEARCH_PASSWORD	elastic用户的密码。
KIBANA_HOST	Kibana私网地址。
KIBANA_PORT	Kibana私网端口。

- 增加spec.selector配置信息，具体内容如下：

```
## kind: DaemonSet
spec:
  selector:
    matchLabels:
       k8s-app: metricbeat
  template:
    metadata:
      labels:
        k8s-app: metricbeat
## kind: Deployment
spec:
  selector:
    matchLabels:
       k8s-app: metricbeat
  template:
    metadata:
      labels:
        k8s-app: metricbeat
```

ii. 分别在name: metricbeat-daemonset-config和name: metricbeat-deployment-config下，配置Kibana Output信息，调用文件中配置的环境变量。

```
output.elasticsearch:
      hosts: ['${ELASTICSEARCH_HOST:elasticsearch}:${ELASTICSEARCH_PORT:9200}']
      username: ${ELASTICSEARCH_USERNAME}
      password: ${ELASTICSEARCH_PASSWORD}
setup.kibana:
      host: "https://${KIBANA_HOST}:${KIBANA_PORT}"
setup.dashboards.enabled: true
```

iii. 修改metricbeat-daemonset-modules配置，定义system模块监控的cpu、load、memory、network等系统指标，以及kubernetes模块可以获取的监控指标。

> **说明** 关于Metricbeat更多的模块配置及指标说明，请参见System module和Kubernetes module。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: metricbeat-daemonset-modules
  namespace: kube-system
  labels:
    k8s-app: metricbeat
data:
  system.yml: |-
    - module: system
      period: 10s
      metricsets:
        - cpu
        - load
        - memory
        - network
        - process
        - process_summary
        - core
        - diskio
        - socket
      processes: ['.*']
      process.include_top_n:
        by_cpu: 5      # include top 5 processes by CPU
        by_memory: 5   # include top 5 processes by memory
    - module: system
      period: 1m
      metricsets:
        - filesystem
        - fsstat
      processors:
      - drop_event.when.regexp:
          system.filesystem.mount_point: '^/(sys|cgroup|proc|dev|etc|host|lib)($|/)'
  kubernetes.yml: |-
    - module: kubernetes
      metricsets:
        - node
        - system
        - pod
        - container
        - volume
      period: 10s
      host: ${NODE_NAME}
      hosts: ["localhost:10255"]
```

iv. 修改metricbeat-deployment-modules配置，获取kube-state-metric监控指标和event服务指标。

> **注意** Metricbeat服务创建在kube-system的namespace下，kube-state-metrics默认创建在arms-prom的namespace下，由于namespace不同，所以hosts的命名格式为kube-state-metrics.<namespace>:8080。如果Metricbeat服务与kube-state-metrics模块都创建在同一namespace下，则hosts的命名格式为kube-state-metrics:8080。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: metricbeat-deployment-modules
  namespace: kube-system
  labels:
    k8s-app: metricbeat
data:
  # This module requires `kube-state-metrics` up and running under `kube-system` nam
espace
  kubernetes.yml: |-
    - module: kubernetes
      metricsets:
        - state_node
        - state_deployment
        - state_replicaset
        - state_pod
        - state_container
      period: 10s
      host: ${NODE_NAME}
      hosts: ["kube-state-metrics.arms-prom:8080"]
    # Uncomment this to get k8s events:
    - module: kubernetes
      metricsets:
        - event
```

v. 配置RBAC权限声明，保证Metricbeat可以获取到Kubernetes集群资源信息。

```
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: metricbeat
subjects:
- kind: ServiceAccount
  name: metricbeat
  namespace: kube-system
roleRef:
  kind: ClusterRole
  name: metricbeat
  apiGroup: rbac.authorization.k8s.io
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRole
metadata:
  name: metricbeat
  labels:
    k8s-app: metricbeat
rules:
- apiGroups: [""]
  resources:
  - nodes
  - namespaces
  - events
  - pods
  verbs: ["get", "list", "watch"]
- apiGroups: ["extensions"]
  resources:
  - replicasets
  verbs: ["get", "list", "watch"]
- apiGroups: ["apps"]
  resources:
  - statefulsets
  - deployments
  verbs: ["get", "list", "watch"]
- apiGroups:
  - ""
  resources:
  - nodes/stats
  verbs:
  - get
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: metricbeat
  namespace: kube-system
  labels:
    k8s-app: metricbeat
---
```

3. 部署Metricbeat，并查看资源状态。

通过kubectl执行以下命令：

```
kubectl apply -f metricbeat-kubernetes.yaml
kubectl get pods -n kube-system
```

> 注意 请确保Pods资源均处于Running状态，否则在Kibana平台有可能会看不到相应数据。

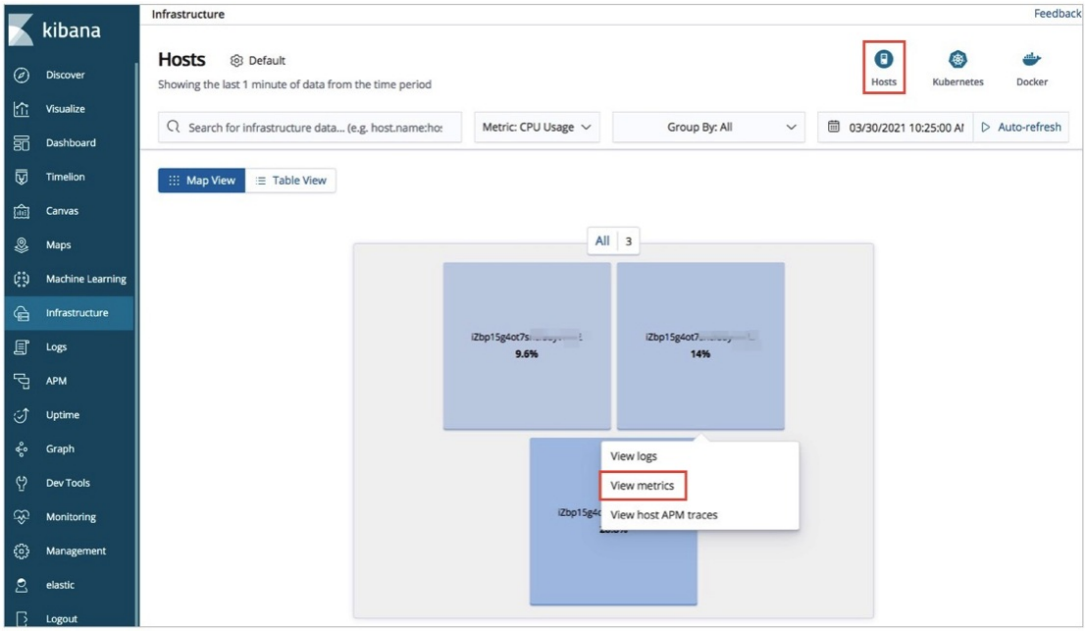
4. 在Kibana查看监测数据。
   i. 登录目标阿里云Elasticsearch实例的Kibana控制台。

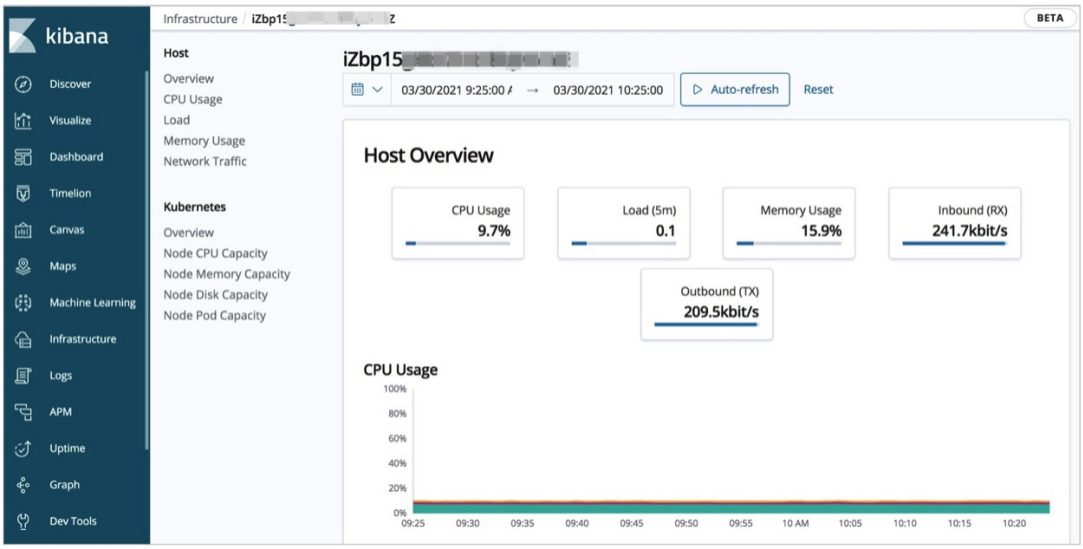
      具体操作步骤请参见登录Kibana控制台。

   ii. 单击左侧菜单**Infrastructure**。

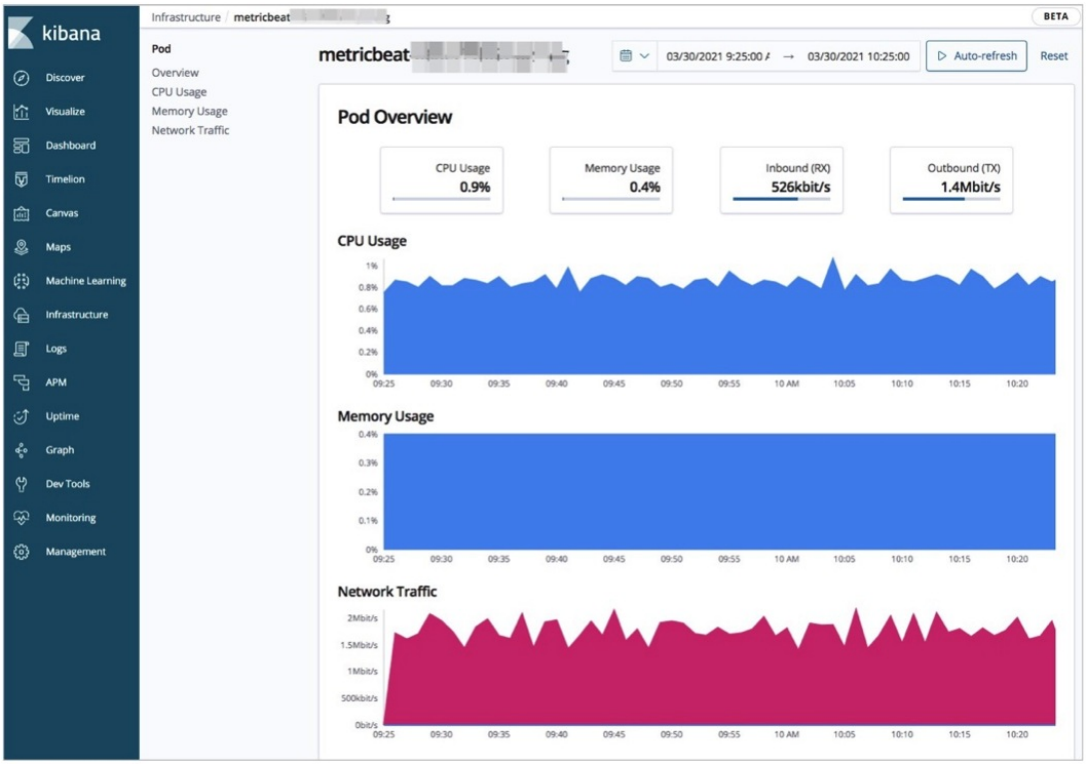
   iii. 查看Hosts、Kubernetes Pods对应的Metrics信息。

   - 查看Hosts对应的Metrics信息：单击右上角**Hosts**，在**Map View**页签下，单击指定Host，选择**View metrics**，就可以查看对应的CPU、Load、Memory等指标数据。



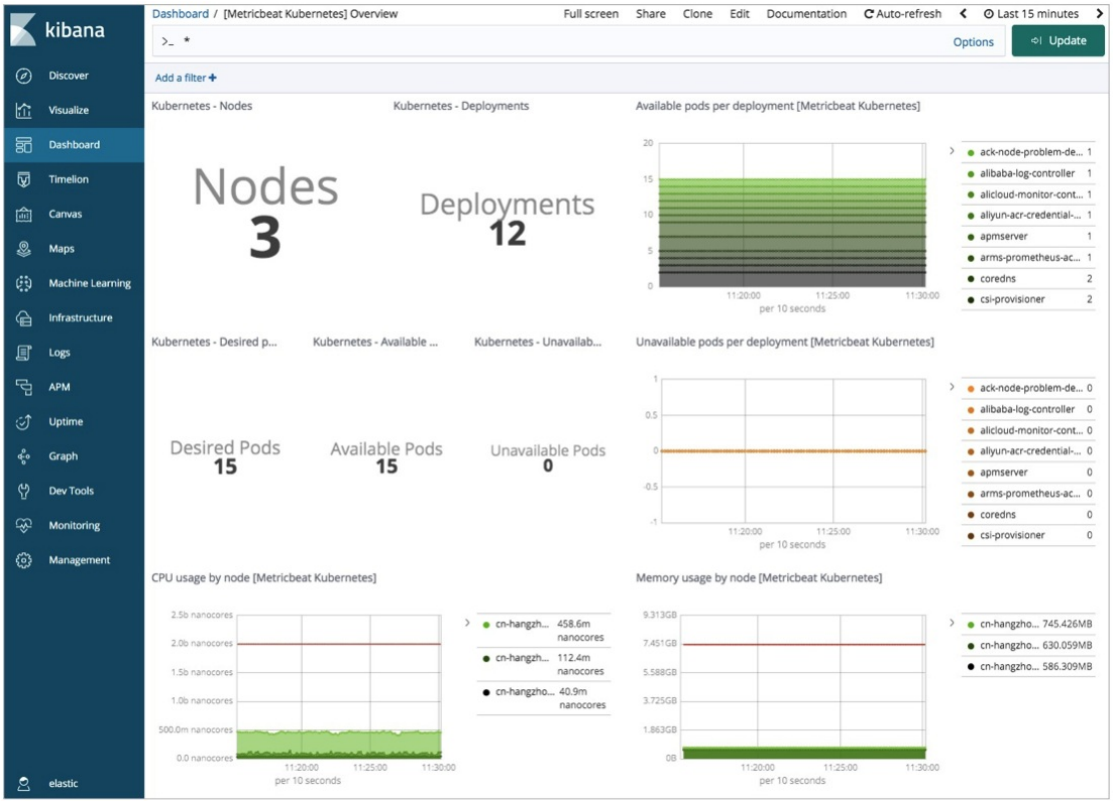


- 查看Kubernetes Pods对应的Metrics信息：单击右上角**Kubernetes**，在**Map View**页签下，单击指定Pod，选择**View metrics**，就可以查看对应的CPU、Memory、Network等指标数据。



iv. 查看Kubernetes集群资源总览数据。

单击左侧菜单**Dashboard**，选择**[Metricbeat Kubernetes] Overview**，就可以查看集群资源总览数据。



通过Filebeat实现日志采集

本文示例中，Filebeat容器使用DaemonSet控制器部署，确保集群上每一个节点都有一个正在运行的实例采集数据，并且配置文件中的资源部署在kube-system命名空间下。您如果需要改变，可以手动更改配置文件。

1. 下载Filebeat配置文件。

   通过kubectl访问云容器，下载Filebeat配置文件。

   ```
   curl -L -O https://raw.githubusercontent.com/elastic/beats/6.8/deploy/kubernetes/filebeat-kubernetes.yaml
   ```

2. 修改Filebeat配置文件。

   i. 修改kind: DaemonSet下的环境变量。

   ```
   env:
   - name: ELASTICSEARCH_HOST
     value: es-cn-nif23p3mo0065****.elasticsearch.aliyuncs.com
   - name: ELASTICSEARCH_PORT
     value: "9200"
   - name: ELASTICSEARCH_USERNAME
     value: elastic
   - name: ELASTICSEARCH_PASSWORD
     value: ****
   - name: KIBANA_HOST
     value: es-cn-nif23p3mo0065****-kibana.internal.elasticsearch.aliyuncs.com
   - name: KIBANA_PORT
     value: "5601"
   - name: NODE_NAME
     valueFrom:
        fieldRef:
           fieldPath: spec.nodeName
   ```

参数	说明
ELASTICSEARCH_HOST	阿里云Elasticsearch实例的私网地址。
ELASTICSEARCH_PORT	阿里云Elasticsearch实例的私网端口。
ELASTICSEARCH_USERNAME	阿里云Elasticsearch的用户名，默认值elastic。
ELASTICSEARCH_PASSWORD	elastic用户的密码。
KIBANA_HOST	Kibana私网地址。
KIBANA_PORT	Kibana私网端口。
NODE_NAME	Kubernetes集群Host。

ii. 修改name: filebeat-config对应的ConfigMap配置信息，配置Kibana Output信息，调用文件中配置的环境变量。

```
output.elasticsearch:
      hosts: ['${ELASTICSEARCH_HOST:elasticsearch}:${ELASTICSEARCH_PORT:9200}']
      username: ${ELASTICSEARCH_USERNAME}
      password: ${ELASTICSEARCH_PASSWORD}
setup.kibana:
      host: "https://${KIBANA_HOST}:${KIBANA_PORT}"
```

iii. 配置Kubernetes，采集容器日志。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: filebeat-inputs
  namespace: kube-system
  labels:
    k8s-app: filebeat
data:
  kubernetes.yml: |-
    - type: docker
      containers.ids:
      - "*"
      processors:
        - add_kubernetes_metadata:
            host: ${NODE_NAME}
            in_cluster: true
---
```

3. 在Kubernetes中部署Filebeat，并查看资源状态。

   通过kubectl执行以下命令：

```
kubectl apply -f filebeat-kubernetes.yaml
kubectl get pods -n kube-system
```

> **注意** 请确保Pods资源均处于Running状态，否则在Kibana平台有可能会看不到相应数据。

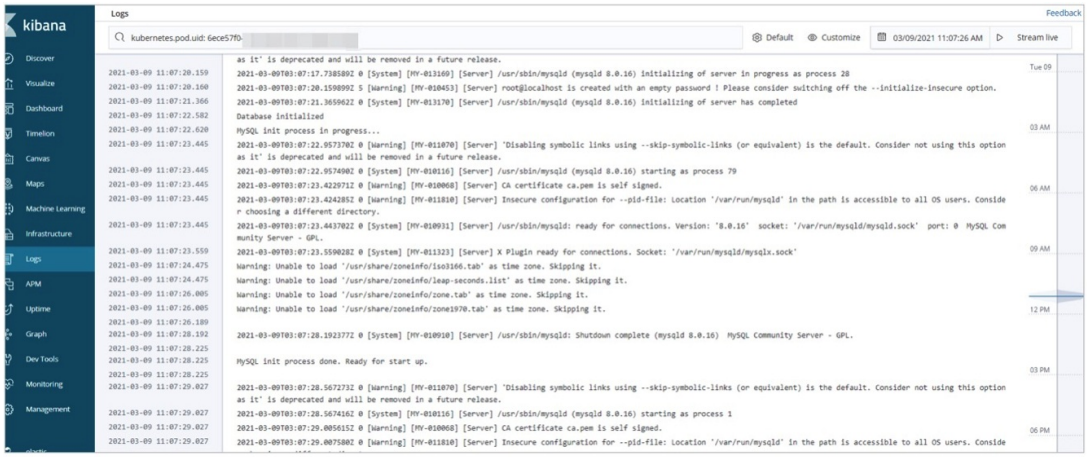
4. 在Kibana查看实时日志。

   i. 登录目标阿里云Elasticsearch实例的Kibana控制台。

   具体操作步骤请参见登录Kibana控制台。

ii. 查看Hosts、Kubernetes Pods对应的日志。

- 查看Hosts对应的日志信息：单击右上角**Hosts**，在**Map View**页签下，单击指定Host，选择**View logs**，就可以查看Host实时日志。
- 查看Kubernetes Pods对应的日志信息：单击右上角**Kubernetes**，在**Map View**页签下，单击指定Pod，选择**View logs**，就可以查看Pod实时日志。



通过Elastic APM实现应用程序性能监测

Elastic APM是基于Elastic Stack构建的应用程序性能监控系统。它提供实时监控软件服务和应用程序的功能，采集传入请求的响应时间和数据库查询、高速缓存调用及外部HTTP请求等的详细性能信息，帮助您更快速的查明并修复性能问题。Elastic APM还可以自动收集未处理的错误、异常及调用栈，帮助您识别出新错误，并关注对应错误发生的次数。

关于Elastic APM更多的介绍，请参见Elastic APM Overview。

1. 部署APM Server容器。

   本文示例使用Kubernetes部署，通过ConfigMap控制器定义apm-server.yml文件，并初始化Pods启动，通过service实现服务自动发现和负载均衡。

   i. 配置apm-server.yml文件。

   完整的配置文件内容如下：

```
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: apm-deployment-config
  namespace: kube-system
  labels:
    k8s-app: apmserver
data:
  apm-server.yml: |-
      apm-server.host: "0.0.0.0:8200"
      output.elasticsearch:
         hosts: ['${ELASTICSEARCH_HOST:elasticsearch}:${ELASTICSEARCH_PORT:9200}']
         username: ${ELASTICSEARCH_USERNAME}
         password: ${ELASTICSEARCH_PASSWORD}
      setup.kibana:
         host: "https://${KIBANA_HOST}:${KIBANA_PORT}"
---
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: apmserver
  namespace: kube-system
  labels:
    k8s-app: apmserver
spec:
  selector:
    matchLabels:
       k8s-app: apmserver
  template:
    metadata:
      labels:
        k8s-app: apmserver
    spec:
      serviceAccountName: apmserver
      hostNetwork: true
      dnsPolicy: ClusterFirstWithHostNet
      containers:
      - name: apmserver
        image: docker.elastic.co/apm/apm-server:6.8.14
        args: [
          "-c", "/etc/apm-server.yml",
          "-e",
        ]
        env:
        - name: ELASTICSEARCH_HOST
          value: es-cn-oew20i5h90006****.elasticsearch.aliyuncs.com
        - name: ELASTICSEARCH_PORT
          value: "9200"
        - name: ELASTICSEARCH_USERNAME
          value: elastic
        - name: ELASTICSEARCH_PASSWORD
          value: ****
        - name: KIBANA_HOST
          value: es-cn-oew20i5h90006****-kibana.internal.elasticsearch.aliyuncs.com
        - name: KIBANA_PORT
          value: "5601"
        - name: NODE_NAME
          valueFrom:
            fieldRef:
              fieldPath: spec.nodeName
        securityContext:
          runAsUser: 0
        resources:
          limits:
            memory: 50Mi
          requests:
            cpu: 20m
            memory: 30Mi
        volumeMounts:
        - name: config
          mountPath: /etc/apm-server.yml
          readOnly: true
```

```
          subPath: apm-server.yml
      volumes:
      - name: config
        configMap:
          defaultMode: 0600
          name: apm-deployment-config
---
apiVersion: v1
kind: Service
metadata:
  name: apmserver
  namespace: kube-system
  labels:
    k8s-app: apmserver
spec:
  clusterIP: None
  ports:
  - name: http-metrics
    port: 8200
    targetPort: 8200
  selector:
    k8s-app: apmserver
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: apmserver
  namespace: kube-system
  labels:
    k8s-app: apmserver
---
```

注意

- Deployment部署资源中，使用docker.elastic.co/apm/apm-server:6.8.14镜像部署Pods容器，镜像版本需要和阿里云Elasticsearch实例版本一致。
- 通过service对集群暴露8200服务端口，保证APM Agents可与APM Server通信。

参数	说明
ELASTICSEARCH_HOST	阿里云Elasticsearch实例的私网地址。
ELASTICSEARCH_PORT	阿里云Elasticsearch实例的私网端口。
ELASTICSEARCH_USERNAME	阿里云Elasticsearch的用户名，默认值elastic。
ELASTICSEARCH_PASSWORD	elastic用户的密码。
KIBANA_HOST	Kibana私网地址。
KIBANA_PORT	Kibana私网端口。

参数	说明
NODE_NAME	Kubernetes集群Host。

ii. 部署APM Server容器，并查看资源状态。

通过kubectl执行以下命令：

```
kubectl apply -f apm-server.yml
kubectl get pods -n kube-system
```

> **注意** 请确保Pods资源均处于Running状态，否则在Kibana平台有可能会看不到相应数据。

2. 配置APM Agents。

本文示例是通过Spring Boot实现一个简单的Web应用并打包为JAR包，并将JAR包和从Maven Central下载的最新Java Agent上传到服务器。详细信息，请参见Spring Boot和Maven Central。

i. 登录Kubernetes节点，在工作目录中创建Dockerfile文件，文件名为myapply。

Dockerfile文件内容如下：

```
FROM frolvlad/alpine-oraclejdk8
MAINTAINER peterwanghao.com
VOLUME /tmp
ADD spring-boot-0.0.1-SNAPSHOT.jar spring-boot-0.0.1-SNAPSHOT.jar
ADD elastic-apm-agent-1.21.0.jar elastic-apm-agent-1.21.0.jar
EXPOSE 8080
ENTRYPOINT ["java","-javaagent:/elastic-apm-agent-1.21.0.jar","-Delastic.apm.service_name=my-application","-Delastic.apm.server_url=http://apmserver:8200","-Delastic.apm.application_packages=com.example","-jar","/spring-boot-0.0.1-SNAPSHOT.jar"]
```

ENTRYPOINT定义容器启动时运行的Java命令及参数如下：

参数	说明
-javaagent	指定APM Agent代理JAR包。
-Delastic.apm.service_name	APM Service Name，允许以下字符：a-z、A-Z、0-9、-、_及空格。
-Delastic.apm.server_url	APM Server URL，*http://apmserver:8200*是在apm-server.yml文件中的service定义。
-Delastic.apm.application_packages	应用程序的基础软件包。
-jar	指定应用JAR包。

ii. 通过 `docker build` 命令和Dockerfile定义的myapply文件构建镜像。

在当前路径下，执行以下命令：

```
docker build -t myapply .
```

iii. 将构建好的镜像加载到其他容器节点。

iv. 配置Pods部署文件，文件名为my-application.yaml。

文件内容如下：

```
---
apiVersion: v1
kind: Pod
metadata:
  name: my-apply
  namespace: kube-system
  labels:
    app: my-apply
spec:
  containers:
    - name: my-apply
      image: myapply:latest
      ports:
        - containerPort: 8080
      imagePullPolicy: Never
---
apiVersion: v1
kind: Service
metadata:
  name: my-apply
  namespace: kube-system
  labels:
    app: my-apply
spec:
  type: NodePort
  ports:
  - name: http-metrics
    port: 8080
    nodePort: 30000
  selector:
    app: my-apply
---
```

> **说明** image为构建好的镜像文件。

v. 通过kubectl执行以下命令，部署Pods。

```
kubectl apply -f my-application.yaml
```

vi. 待所有Pods资源均处于Running状态后，使用Curl访问主机的30000端口。

执行命令如下：

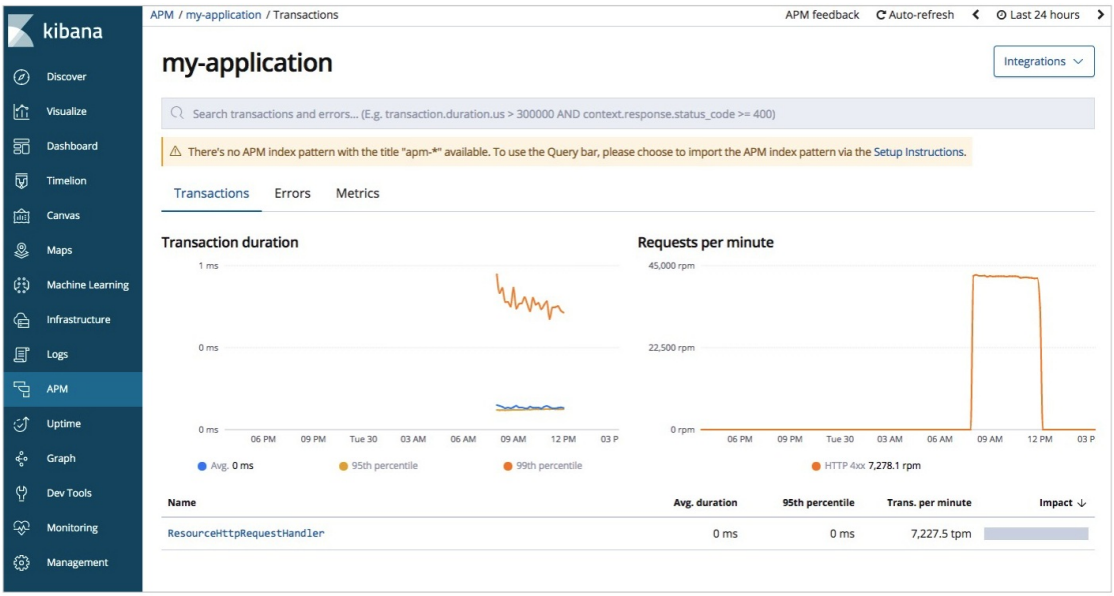
```
curl http://10.7.XX.XX:30000
```

> **说明** 10.7.XX.XX为Kubernetes的节点IP地址。

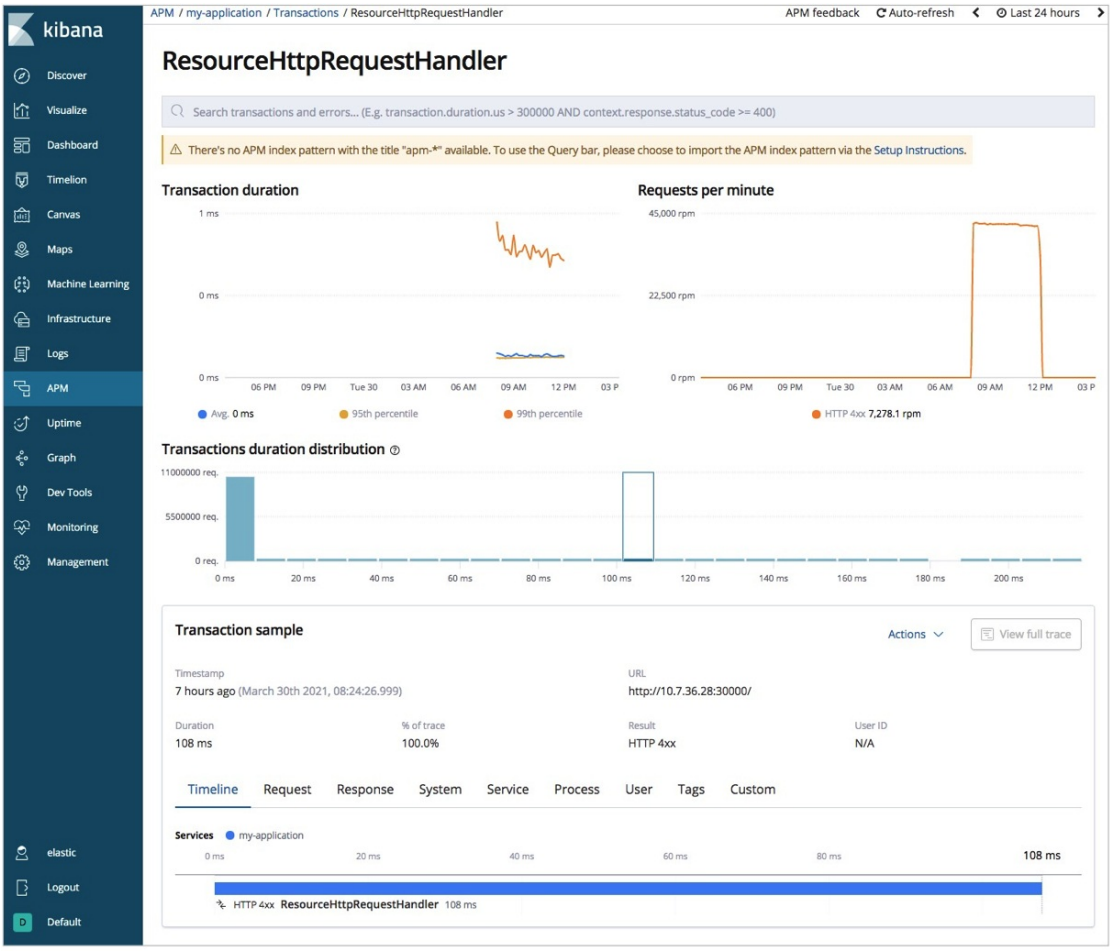
能够访问对应主机后，APM Agents部署成功。

3. 在Kibana控制台查看APM监控数据。
   i. 登录目标阿里云Elasticsearch实例的Kibana控制台。

      具体操作步骤请参见登录Kibana控制台。
   ii. 单击左侧菜单**APM**。
   iii. 单击目标应用程序，本文示例为my-application，可以看到服务的整体性能数据。

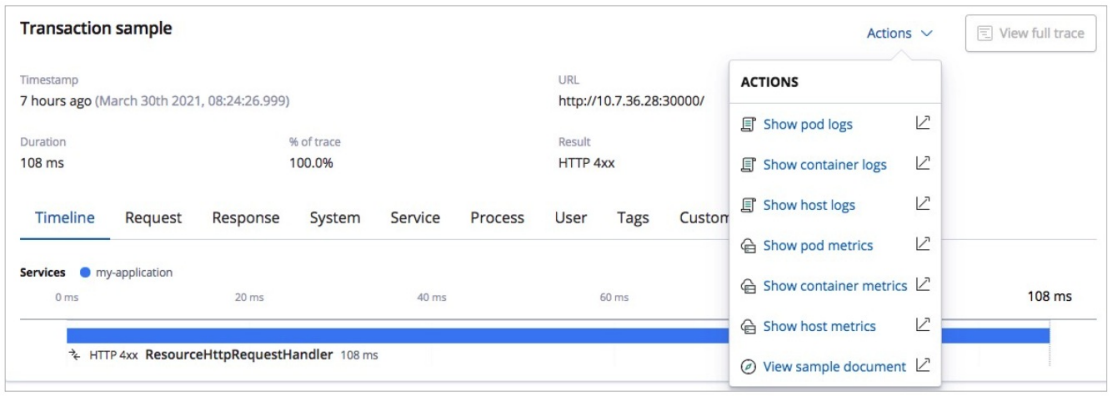


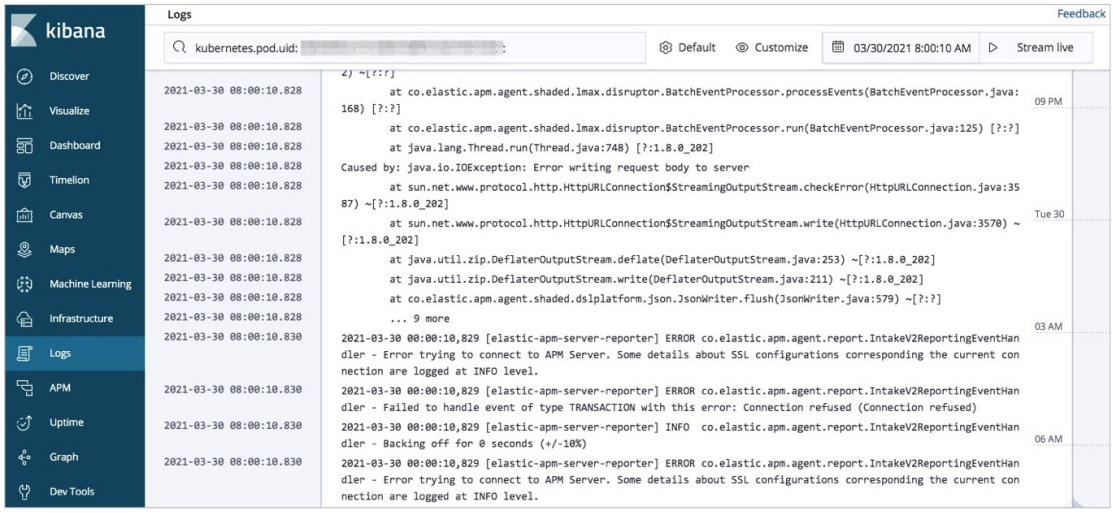
iv. 单击对应的请求接口，可以看到具体的请求信息。



v. 查看Hosts、Pods相关日志和指标数据。

单击Actions，选择Show pod logs、Show pod metrics等，查看对应的日志和指标数据。





常见问题

- 问题：Kubernetes配置文件中resources.requests下资源设置较大，Pods无法启动成功。

  解决方案：Metricbeat、Filebeat、APM配置文件中都需要设置resources.requests，建议根据Kubernetes集群规格适当调整设置的值。

- 问题：在部署Metricbeat、Filebeat、APM容器时，一直报错。报错内容类似：`no matches for kind "DaemonSet" in version "extensions/v1beat1"` 。

  解决方案：官方下载的YML文件中，Daemonsets和Deployments的资源使用extensions/v1beta1，而v1.18及以上版本的Kubernetes，Daemonsets、Deployments和Replicasets资源的extensions/v1beta1 API将被废弃，请使用apps/v1。

9.5.3. 基于Indexing Service实现数据流管理

通过使用阿里云Elasticsearch 7.10日志增强版Indexing Service系列，可以为您实现云托管写入加速和按流量付费（即您无需按集群峰值写入吞吐预留资源），能够极低成本实现海量时序日志分析。本文为您介绍如何基于Indexing Service系列实现数据流管理以及日志场景分析。

背景信息

在复杂业务场景下，海量服务器、物理机、Docker容器、移动设备和IoT传感器等设备中往往存在着结构分散、种类多样且规模庞大的各类指标和日志数据，而除了底层系统的各类指标和日志数据外，往往还存在着规模庞大的业务数据，例如用户行为、行车轨迹等。当面对海量时序数据和日志数据写入出现性能瓶颈时，您可以根据业务需求选择使用阿里云Elasticsearch 7.10日志增强版Indexing Service系列，此功能基于读写分离架构以及写入按量付费的Serverless模式，实现了Elasticsearch集群的云端写入托管和降本提效的目标。

在阿里云Elasticsearch 7.10日志增强版Indexing Service系列中，推荐使用数据流管理，可以帮您实现跨多索引存储仅追加时间序列数据，为请求提供唯一的命名资源；并且您可以根据关联的索引模板和Rollover策略实现自动取消托管，从而达到云端托管数据的自动清理和成本优化。数据流管理非常适用于日志、事件、指标和其他连续生成数据的场景。除此之外，您还可以通过使用索引生命周期管理（ILM）定期管理后备索引，帮助您降低成本及开销。

Elasticsearch集群中既可以存在数据流（Data Stream），也可以存在独立索引（Index）对象。除系统索引不托管外，其他索引均默认开启云端托管功能。独立索引支持增、删、改、查操作，操作前需要您手动取消云端托管。为了帮助您更好的使用数据流管理云端托管索引，阿里云Elasticsearch控制台分别提供了数据流管理、索引管理和索引模板管理功能模块，通过白屏化的方式为您实现数据流一站式管理。

使用场景

本文通过将采集到的nginx服务日志数据，写入到阿里云Elasticsearch 7.10日志增强版Indexing Service系列实例中，通过数据流管理和索引生命周期管理，实现日志数据的分析和检索。

注意事项

- 因为数据流写入依赖时间字段@timestamp，所以请确保写入数据中存在@timestamp字段的数据，否则数据流写入过程中会报错。如果源数据中没有@timestamp字段数据，您可以使用ingest pipeline指定_ingest.timestamp，获取元数据值，从而引入@timestamp字段数据。
- Indexing Service提供了写入Serverless保护机制，因此使用前请参见使用限制，提前优化配置，以避免使用过程中出现不合规的情况。
- Indexing Service日志增强版实例与用户集群进行数据同步时，依赖于apack/cube/metadata/sync任务（可通过 `GET _cat/tasks?v` 命令获取该任务信息），不建议手动清理该任务。如果被清理，请尽快使用 `POST /_cube/meta/sync` 命令恢复，否则会影响业务写入。

操作流程

1. 步骤一：创建Indexing Service实例

   创建一个阿里云Elasticsearch 7.10日志增强版Indexing Service系列的实例。

2. 步骤二：创建索引模板

   在使用数据流之前，需要创建索引模板，通过模板对数据流后备索引进行结构配置。

3. 步骤三：创建数据流

   创建数据流并写入数据。

4. 步骤四：管理托管索引

   对数据流或者独立索引进行云端托管管理。

5. 步骤五：查看集群信息

   在节点可视化页面，查看集群当天写入的总流量以及写入托管总数量。

6. 步骤六：分析日志

   在Kibana控制台中，查看基于Indexing Service实现的数据流管理的实时日志流和实时数据指标。

步骤一：创建Indexing Service实例

1. 前往实例创建页面。
2. 根据页面提示，按照以下说明选择实例配置，完成购买。

   购买实例的具体操作请参见创建阿里云Elasticsearch实例。下表为购买页的部分参数说明，未提及的参数保持默认，详细信息请参见购买页面参数（增强版）。

参数	示例值	说明
付费模式	按量付费	○ **按量付费**：在前期程序研发或功能测试期间，建议购买**按量付费**实例进行测试。支持在控制台手动释放实例。 ○ **包年包月**：目前在购买包年包月实例时，可以享受优惠条件。购买后，支持5天内退余款。超过5天后，将不再支持退款。 支持手动续费和自动续费，详细信息，请参见Elasticsearch续费。不支持在控制台手动释放实例。
选择服务	日志增强版	仅阿里云Elasticsearch日志增强版7.10支持Indexing Service系列介绍。本文以7.10版本为例。
系列	Indexing Service	阿里云Elasticsearch 7.10日志增强版Indexing Service提供云端托管能力，能够在低成本下，提高数据写入速度。
场景初始化配置	日志场景	阿里云Elasticsearch 7.10日志增强版默认应用**日志场景**模板，使集群配置适配于日志场景。
地域和可用区	华东1（杭州） 杭州可用区I	购买页仅显示日志增强版Indexing Service系列支持的地域和可用区。 **说明** 建议您选择和云端其他业务相同的地域和可用区，提高业务的集中化管理。
可用区数量	单可用区	○ **单可用区**：普通部署模式，适用于非关键任务型工作（默认）。 ○ **两个可用区**：跨可用区容灾部署模式，适用于生产型工作。 ○ **三个可用区**：高可用部署模式，适用于具有更高可用性要求的生产型工作。
写入Serverless资源	开启	日志增强版Indexing Service默认开启写入Serverless模块，开启后集群无写入计算压力，您可以减少数据节点的配置规格及数量，并通过OpenStore存储实现海量日志存储分析。 写入Serverless模块按实际写入流量及托管存储空间进行按量计费，详情请参见Elasticsearch计费项。

参数	示例值	说明
查询集群资源	开启	当您配置集群资源时，可以按照查询业务要求配置冷、热节点资源，具体说明如下： ○ 写入云端托管场景下，当您查询集群无写入计算压力时，推荐使用冷数据节点配置，降低资源成本。 ○ 如果您对日志数据查询时延有较高要求，可以对冷、热数据节点进行规划，选择规格合适的热数据节点。

3. 提示开通成功后，单击**管理控制台**，进入阿里云Elasticsearch实例的控制台**概览**页面。
4. 在左侧导航栏，单击**Elasticsearch实例**。在顶部菜单栏，选择资源组和地域，然后在**实例列表**页面查看创建成功的日志增强版Indexing Service实例。

> **说明**
> - 实例创建后，需要一段时间才能生效。时间长短与您的集群规格、数据结构和大小等相关，一般在小时级别。
> - 当实例的信息没有及时更新时，例如刚创建完成的实例状态显示失败，可在**基本信息**页面，单击**刷新**，手动刷新页面中的状态信息。

步骤二：创建索引模板

1. 登录阿里云Elasticsearch控制台。
2. 进入目标实例。
   i. 在顶部菜单栏处，选择资源组和地域。
   ii. 在左侧导航栏，单击**Elasticsearch实例**，然后在**Elasticsearch实例**中单击目标实例ID。
3. 在左侧导航栏，选择**配置与管理 > 索引管理中心**。
4. 单击**索引模板管理**页签。
5. 单击**创建索引模板**。
6. （可选）在**创建索引模板**面板，参考下图配置索引生命周期策略。

> **说明** 如果您无需对数据流后备索引进行生命周期策略管理，单击**跳过此步**即可。

部分参数说明如下。未提及参数请参考页面上的具体说明。

参数	示例值	说明
索引生命周期策略	新建索引生命周期策略	○ **新建索引生命周期策略**：创建新的索引生命周期策略。 **说明** Indexing Service架构下，不支持在索引生命周期中自定义freeze。 ○ **选择已有索引生命周期策略**：集群中存在服务业务逻辑策略，点击下拉框选择即可。

参数	示例值	说明
策略名称	nginx_policy	新建索引生命周期策略时，需要自定义输入；选择已有索引生命周期策略时，需要在下拉列表中选择集群中已存在的生命周期策略。
删除时间	7天	设置索引保留多少天后会被自动删除。

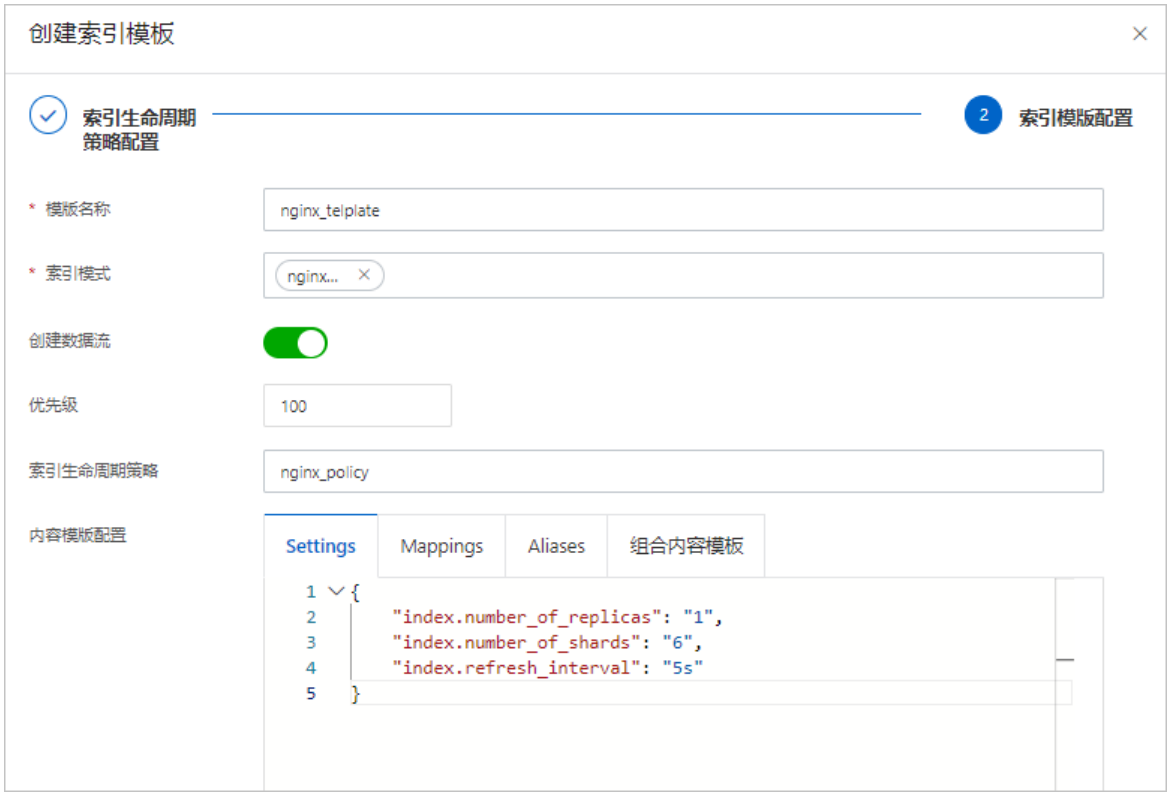
本步骤使用的命令示例如下。

```
PUT /_ilm/policy/nginx_policy
{
  "policy": {
    "phases": {
      "hot": {
        "actions": {
          "rollover": {
            "max_size": "30GB",
            "max_age": "1d",
            "max_docs": 1000000
          },
          "set_priority" : {
            "priority": 1000
          }
        }
      },
      "delete": {
        "min_age": "7d",
        "actions": {
          "delete": {}
        }
      }
    }
  }
}
```

以上新建的索引生命周期策略表示，当托管索引满足以下任意条件时，将触发滚动更新，生成新的后备索引，原索引保留7天后将自动删除：

- 写入文件数超过1000000。
- 索引大小达到30 GB。
- 索引从创建开始满1天。

7. 单击**保存并下一步**，配置索引模版信息。



参数	示例值	说明
模板名称	nginx_telplate	定义的模板名称。
索引模式	nginx-*	定义索引模式，使用通配符（*）表达式匹配数据流及索引名称，不允许使用空格和字符 `\/?"<>|` 。
创建数据流	开启	开启数据流模式。如果未开启，索引模式无法生成数据流。详细信息，请参见Data stream。
优先级	100	定义模板优先级，数值越大，优先级越高。
索引生命周期策略	nginx_policy	只能引用一个索引生命周期策略。

参数	示例值	说明

内容模板配置	Settings配置如下：
	{
	"index.number_of_replicas": "1",
	"index.number_of_shards": "6",
	"index.refresh_interval": "5s"
	}
		配置索引Settings、Mappings、Aliases和组合内容模板。
		注意
		○ 写入到数据流中的每个文档都要求包含一个@timestamp字段，建议在索引模板中为@timestamp字段指定映射。如果不指定，该字段会映射为Elasticsearch中的date或者date_nanos类型的字段。
		○ 配置格式严格按照Elastic官方配置。

本步骤使用的命令示例值如下：

```
PUT /_index_template/nginx_telplate
{
  "index_patterns": [ "nginx-*" ],
  "data_stream": { },
  "template": {
    "settings": {
      "index.number_of_replicas": "1",
      "index.number_of_shards": "6",
      "index.refresh_interval": "5s",
      "index.lifecycle.name": "nginx_policy",
      "index.apack.cube.following_index": true
    }
  },
  "priority": 100
}
```

注意

- 通过命令创建模板时，务必将index.apack.cube.following_index设置为true。
- 云端托管集群上index.refresh_interval参数已默认配置最优，手动配置不生效。如果需要通过手动配置index.refresh_interval生效，需要先取消云托管功能。

8. 单击**确认**，索引模板列表中会显示您创建的模板。

步骤三：创建数据流

1. 在**索引管理中心**页面，单击**数据流管理**页签。
2. 单击**创建数据流**。
3. 在**创建数据流**面板，单击**预览已有索引模板**，根据对应的索引模板，输入可匹配索引模板的数据流名称。

本步骤使用的命令示例值如下。

```
PUT /_data_stream/nginx-log
```

> **注意**
>
> - 创建数据流之前必须存在数据流可匹配的索引模板，该模板包含用于配置数据流的后备索引映射及设置。
> - 数据流名称支持以短划线（-）结尾，不支持通配符星号（*）。

4. 单击**确定**，系统会自动生成数据流及后备索引。

   每个数据流创建成功后，都会自动生成一个统一格式的后备索引，格式如下。

```
.ds-<data-stream>-<yyyy.MM.dd>-<generation>
```

参数	说明
.ds	隐藏索引名统一标识，数据流生成的后备索引名，默认均以.ds开头。
<data-stream>	数据流名称。
<yyyy.MM.dd>	后备索引创建日期。

参数	说明
<generation>	每个数据流都会生成一个六位数，默认从000001开始的累积整数值，generation值更大的后备索引包含更多新数据。

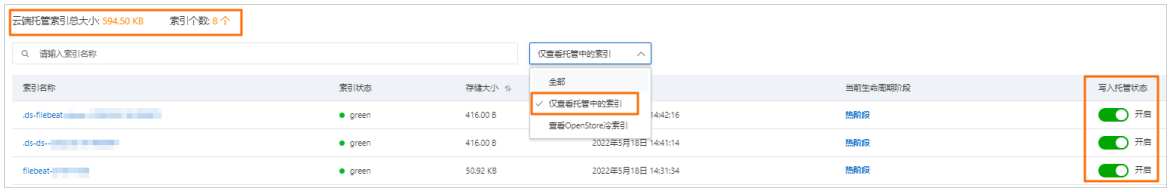
5. 写入数据，具体操作请参见最佳实践。

   数据写入过程中，必须带@timestamp字段，否则写入失败。本场景采用filebeat+kafka+logstash架构将日志采集写入到Elasticsearch实例中，采集过程中会自动生成@timestamp字段。命令示例如下。

```
POST /nginx-log/_doc/
{
  "@timestamp": "2099-03-07T11:04:05.000Z",
  "user": {
    "id": "vlb44hny"
  },
  "message": "Login attempt failed"
}
```

步骤四：管理托管索引

1. 在**索引管理中心**页面，单击**索引管理**页签，查看处于云托管状态的索引。



参数	说明
仅查看托管中的索引	系统默认展示集群中的所有索引（不包括系统索引），选择**仅查看托管中的索引**后，系统仅展示托管中的索引，帮助您快速获取处于托管的数据。
云端托管索引总大小	当前时刻，正处于云端写入托管中的索引总大小。 **注意** 云端托管索引总大小为实时变化数值，不是历史索引总大小。
索引个数	当前时刻，正处于云端写入托管中的索引总个数。 该数值为当前系统中的实时数值。 **注意** 索引个数为实时变化数值，不是历史索引总个数。

参数	说明
写入托管状态	○ 开启：该索引的云端写入托管处于开启状态。默认开启。 ○ 关闭：取消该索引的云端写入托管。支持手动关闭，关闭后不支持开启。 **说明** ○ 手动关闭某一索引的云端写入托管，数据将直接写入用户集群中。请在关闭前确认该索引是否持续有数据写入，以及用户集群负载情况，否则可能出现用户集群负载较高风险。 ○ Indexing Service按照写入托管索引总大小和写入流量进行按量计费，业务上建议使用数据流（Data Stream）和索引生命周期管理（ILM）滚动策略实现云端托管空间最优化。 ○ Indexing Service场景，索引处于托管状态，不兼容ILM Action中的shrink操作，建议ILM取消shrink配置或取消托管后再执行shrink操作，详细信息请参见ILM-shrink。 ○ 在独立索引的云端写入托管过程中，索引数据会全量存储在云托管服务Indexing Service中，将会增加云托管费用。请根据业务使用场景（如索引是否仍有数据写入）评估是否需要手动关闭该索引的写入托管。

说明 由于数据流nginx-log配置了索引滚动策略，所以在云托管服务上，每次仅保存最新生成的后备索引（本场景中的.ds-nginx-log-2021.04.26-000004），旧的后备索引会自动从云托管上关闭。

2. 取消索引托管。

   独立索引或未设置滚动策略的索引将一直在云托管服务保存，需要手动关闭。关闭后，对应索引的写入托管状态会处于**关闭**状态。

注意

○ 取消云托管后，无法再次开启云端Indexing Service写入托管功能。

○ Elasticsearch集群中既可以存在数据流（Data Stream），又可以存在独立索引（Index）对象，除系统索引不托管外，其他索引均默认开启托管功能。

○ 您可以通过Indexing Service API获取更多Indexing Service托管集群信息。

i. 在**索引管理**页签中，单击对应索引右侧**写入托管状态**列下的**开启**开关。

ii. 在**取消托管**弹框中，单击**确认**。

本步骤对应的命令示例如下。

```
POST /.ds-nginx-log-xxx/_cube/unfollow
```

步骤五：查看集群信息

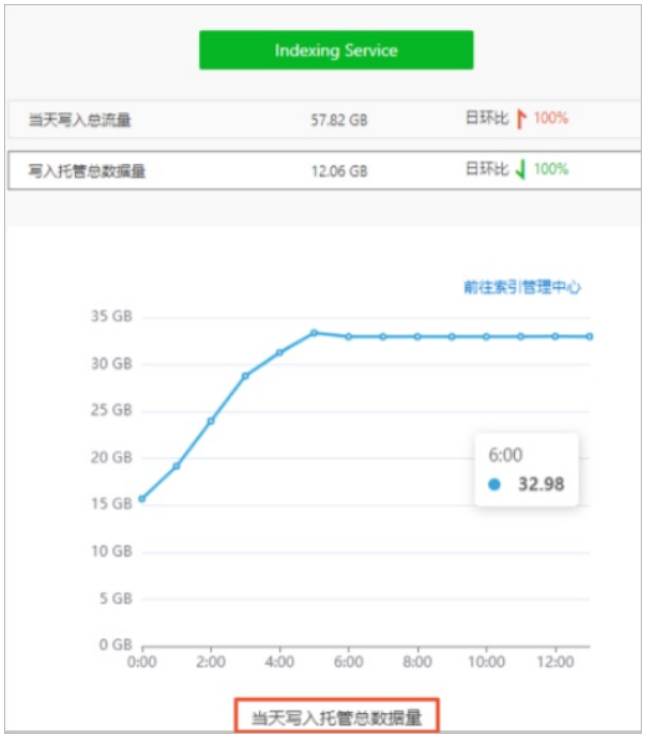
1. 进入节点可视化页面，查看写入Indexing Service实时写入流量和数据量信息。
2. 在**Indexing Service**区域，单击**当天写入总流量**，即可查看**每小时平均写入吞吐量**的曲线图。

> **说明** Indexing Service写入总流量监控为非实时整点展示的静态趋势监控图，监控数据展示延时最长为1小时。例如在14:00~14:59间写入的总流量，需要等到15:10后，在监控页面的14:00处获取。

3. 单击**查看监控详情**，将跳转至Grafana监控展示更详细的监控数据。

> **注意** Grafana的登录名和密码请从高级监控报警获取。

4. 在**Indexing Service**页面，单击**写入托管总数据量**，即可查看**当天写入托管总数据量**。



> **说明** Indexing Service写入总流量监控为非实时整点展示的静态趋势监控图，监控数据展示延时最长为1小时，例如在14:00~14:59间写入的总数据量，需要等到15:10后，在监控页面的14:00处获取。

步骤六：分析日志

1. 登录目标阿里云Elasticsearch实例的Kibana控制台，根据页面提示进入Kibana主页。

   登录Kibana控制台的具体操作，请参见登录Kibana控制台。

   > **说明** 本文以阿里云Elasticsearch 7.10.0版本为例，其他版本操作可能略有差别，请以实际界面为准。

2. 创建索引模板。

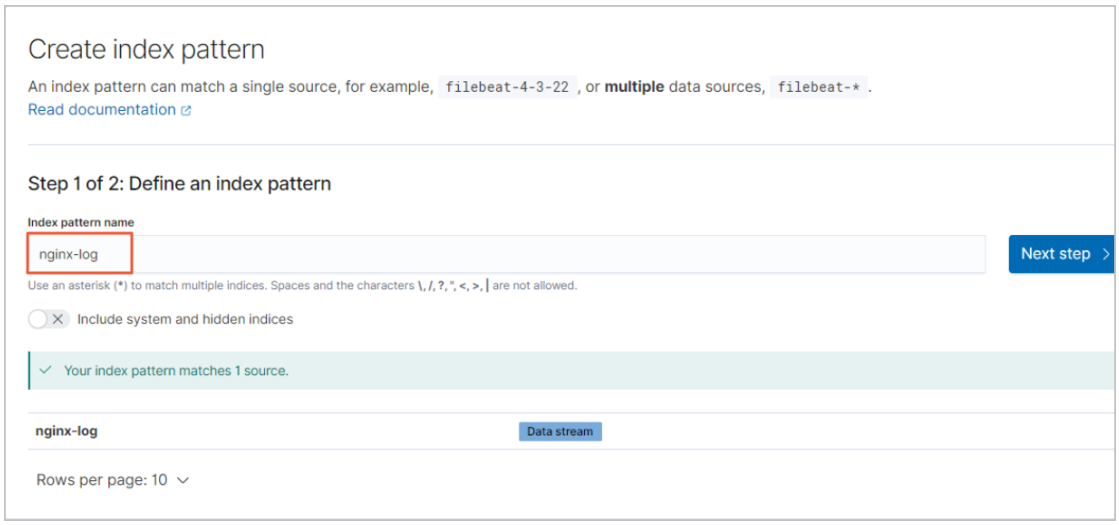
   i. 单击左上角的 ☰ 。

   ii. 在左侧导航栏，选择Management > Stack Management。

   iii. 在Stack Management页面的Kibana区域，单击Index Patterns。

   iv. 单击Create index pattern。

v. 在Create index pattern页面的Index pattern name文本框中，输入索引模板名称。



> **说明** Index pattern name不仅可以指定为数据流名称，也可以指定为后备索引名称。

3. 设置Settings。

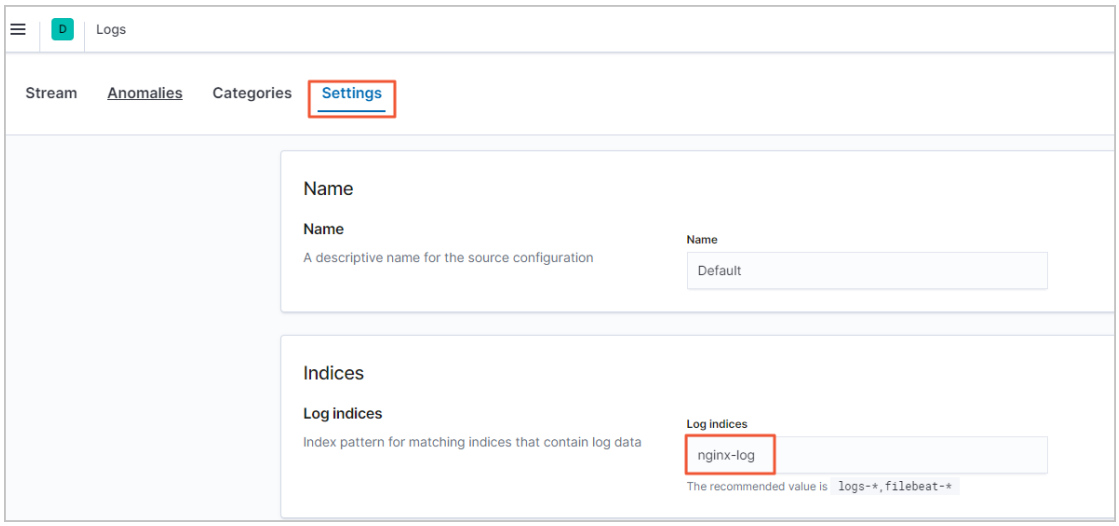
i. 单击左上角 ☰ 。

ii. 在左侧导航栏，选择Observability > Logs。

iii. 在Logs页面，单击Settings页签。

iv. 在Log indices文本框中，输入数据流名称。

本文以nginx-log数据流名称为例，其他字段的默认配置符合数据流数据要求，可不修改。



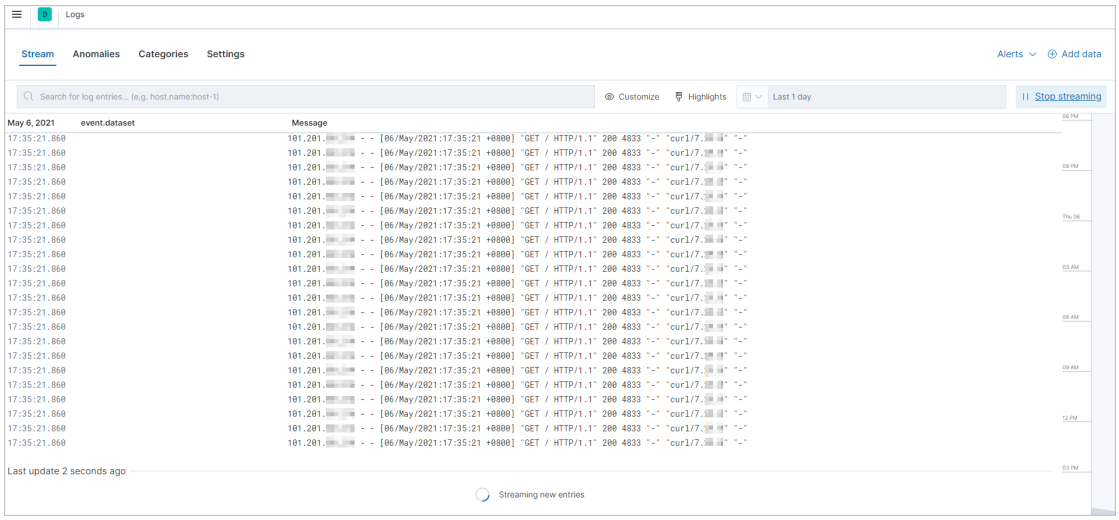
v. 在右下角，单击Apply。

4. 获取实时日志流数据。

i. 在Logs页面，单击Stream页签。

ii. 在页面右侧，单击Stream live。

iii. 在Stream页签中，查看获取到的实时数据流。

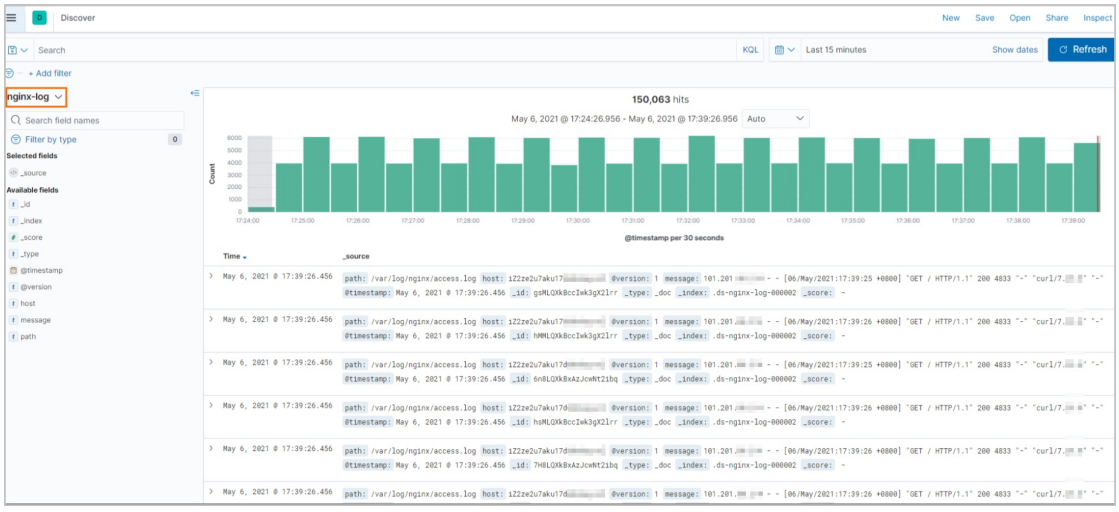


5. 获取实时数据指标。

i. 单击左上角 ☰ 。

ii. 在左侧导航栏，选择Kibana > Discover。

iii. 在Discover页面，选择对应索引，获取该索引的实时数据指标。



更多Kibana日志分析功能请参见Kibana Guide。

常见问题

Q：为Indexing Service实例中的写入托管索引配置refresh、merge等写入参数，是否会生效？

A：不会生效。Indexing Service实例中的写入托管索引已使用默认写入参数配置，用户侧配置不生效。默认写入参数配置如下。

```
"index.merge.policy.max_merged_segment" : "1024mb",
"index.refresh_interval" : "3s",
"index.translog.durability" : "async",
"index.translog.flush_threshold_size" : "2gb",
"index.translog.sync_interval" : "100s"
```

9.5.4. 通过OpenStore实现海量数据存储

OpenStore存储是阿里云Elasticsearch团队自研的针对日志场景的低成本、高效、弹性存储解决方案，能够为您在日志场景中提供海量存储服务。本文介绍在不同的场景下，如何开启OpenStore存储，以及其使用方式。

背景信息

当您有长时间存储数据、归档审计数据的需求时，通常需要通过阿里云Elasticsearch集群快照的方式将数据存储在对象存储OSS上，该方式虽然能够帮助您存储日志数据，但是存储后不能够直接进行信息查询。查询信息前，需要您调用相关API把快照信息恢复到集群中，等待快照中的索引初始化完成后才可以去查询。该场景面临着查询复杂度大、海量存储成本高的问题。为解决此问题，阿里云Elasticsearch团队自研OpenStore存储功能，该功能实现了基于计算存储分离的超低成本、弹性存储，可以帮助您实现根据实际数据的存储量按量计费，无须提前预留集群存储容量，并100%兼容Elasticsearch原生查询能力。真正做到有多少用多少，用多少付多少。在提升集群易用性的同时，大大降低了云上Elasticsearch海量数据的存储资源成本。

使用限制

购买和使用OpenStore存储时，存在以下使用限制。

类别	限制项	限制说明
地域	华东2（上海）、华北3（张家口）、华东1（杭州）、华南 1（深圳）、华北 2（北京）	目前仅开放华东2（上海）、华北3（张家口）、华东1（杭州）、华南 1（深圳）、华北 2（北京）地域。
实例版本	仅7.10版本实例支持开启OpenStore存储	仅支持以下两种方式开启OpenStore存储： • 新购7.10日志增强版Index Service实例开启。 • 已购7.10通用商业版，通过集群升配功能开启。 **注意** ◦ 已购7.10通用商业版，内核小版本需要升级到1.5.0及以上，才可开启OpenStore存储。 ◦ OpenStore属于阿里云Elasticsearch日志增强版特性，商业版升级后，实例类型将变为日志增强版，不会因为开启OpenStore存储而将实例升级为Indexing Service实例。
实例存储	单节点最大存储数据量	单节点最大存储数据量为20 TB。 **说明** 如果您有更大的单节点存储需求，请提交工单申请，最大支持50 TB。

类别	限制项	限制说明
shard数	shard副本数	开启OpenStore存储，索引数据会存储至OpenStore，索引shard副本数默认为0，数据的可靠性将由底层存储保证，请放心使用。 **注意** 如果将OpenStore存储的索引副本数设置为1，索引将处于yellow状态，所以不建议手动设置副本，保持默认值即可。
索引模板	OpenStore定制索引模板	开启OpenStore存储，阿里云Elasticsearch会默认提供定制模板openstore-index-template，模板默认使用openstore_default_ilm_policy策略。 **说明** 手动删除OpenStore存储索引时，需要将索引及索引对应的别名一起删除才可删除成功。
索引生命周期配置	不支持在索引生命周期中自定义freeze	无。

操作流程

1. 步骤一：开启OpenStore存储
2. 步骤二：管理OpenStore索引模板
3. 步骤三：将数据流写入OpenStore索引

步骤一：开启OpenStore存储

阿里云Elasticsearch支持通用商业版和日志增强版两种类型的实例。您可以通过创建7.10日志增强版Indexing Service系列实例开启OpenStore存储功能；也可以将实例从7.10通用商业版升级至7.10日志增强版，开启OpenStore存储功能。

新购实例开启OpenStore存储

1. 登录阿里云Elasticsearch控制台。
2. 在左侧导航栏，单击**Elasticsearch实例**。
3. 在**Elasticsearch实例**页面，单击**创建**。
4. 在购买页面中，选择实例类型为日志增强版7.10，**系列**为Indexing Service。

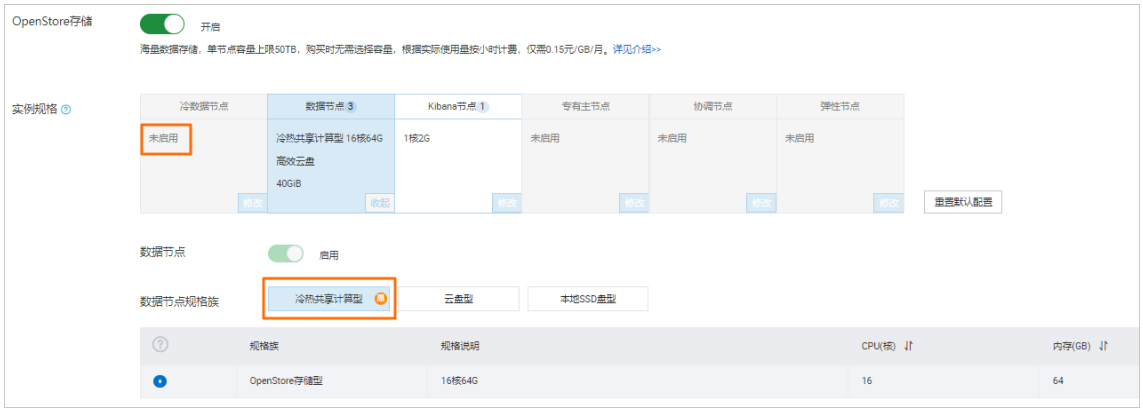
5. 单击**下一步：集群配置**，在**集群配置**中打开**OpenStore存储**开关，并选择实例规格。

   您可以通过以下方式选择实例规格，开启**OpenStore存储**：

   - 开启独立冷数据节点（默认）：冷数据节点+数据节点+Kibana节点，其中冷数据节点规格固定为OpenStore存储型16核64 GB。

> **说明** 开启OpenStore存储，默认会开启数据节点，集群采用冷热分离架构。OpenStore存储仅应用在冷数据节点上，且仅支持16核64 GB的固定规格。

   - 选择冷热共享型数据节点：数据节点+Kibana节点，其中**数据节点规格族**需要选择**冷热共享计算型**。



> 说明 每个OpenStore节点的最大存储数据量为20 TB。如果您有更大的单节点存储需求，请提交工单申请，最大支持50 TB。

6. 参见购买页面参数（增强版），配置其他参数，完成购买。

已购通用商业版实例开启OpenStore存储

1. 登录阿里云Elasticsearch控制台。
2. 在左侧导航栏，单击**Elasticsearch实例**。
3. 进入目标实例。
   i. 在顶部菜单栏处，选择资源组和地域。
   ii. 在左侧导航栏，单击**Elasticsearch实例**，然后在**Elasticsearch实例**中单击目标实例ID。
4. （可选）升级内核版本。

   在**基本信息**页面，查看实例的内核版本是否过低。如果页面没有**有可更新的内核补丁**提示，说明当前内核版本为最新版本，无需升级，可忽略此步骤。如果有，请执行以下步骤升级内核版本：

   i. 单击**有可更新的内核补丁**。

   ii. 参见升级版本，将内核版本升级到1.5.0及以上。

5. 在**基本信息**页面的**节点可视化**区域，开启OpenStore存储。

i. 在OpenStore存储模块，打开OpenStore存储用量开关。

ii. 在弹出的对话框中，单击**立即开启**。

> 注意
>
> - 如果集群已开启非OpenStore的冷数据节点，则系统不展示OpenStore模块，且不支持切换至OpenStore存储。
> - 如果您实例的内核版本过低，在开启OpenStore存储时，系统会提示您升级版本，请先按照提示升级实例的内核版本，具体操作请参见升级版本。

iii. 在升配页面开启**OpenStore存储**，勾选服务协议，单击**立即购买**，按照提示完成购买，并返回OpenStore存储模块。

升配的具体操作，请参见升配集群。

步骤二：管理OpenStore索引模板

通过控制台管理索引模板

1. 进入已开启OpenStore存储的目标实例，在左侧导航栏选择**配置与管理 > 索引管理中心**。

   > 说明 本章节以7.10日志增强版Indexing Service实例为例进行介绍，商业版实例不支持索引管理中心功能，具体以页面显示为准。

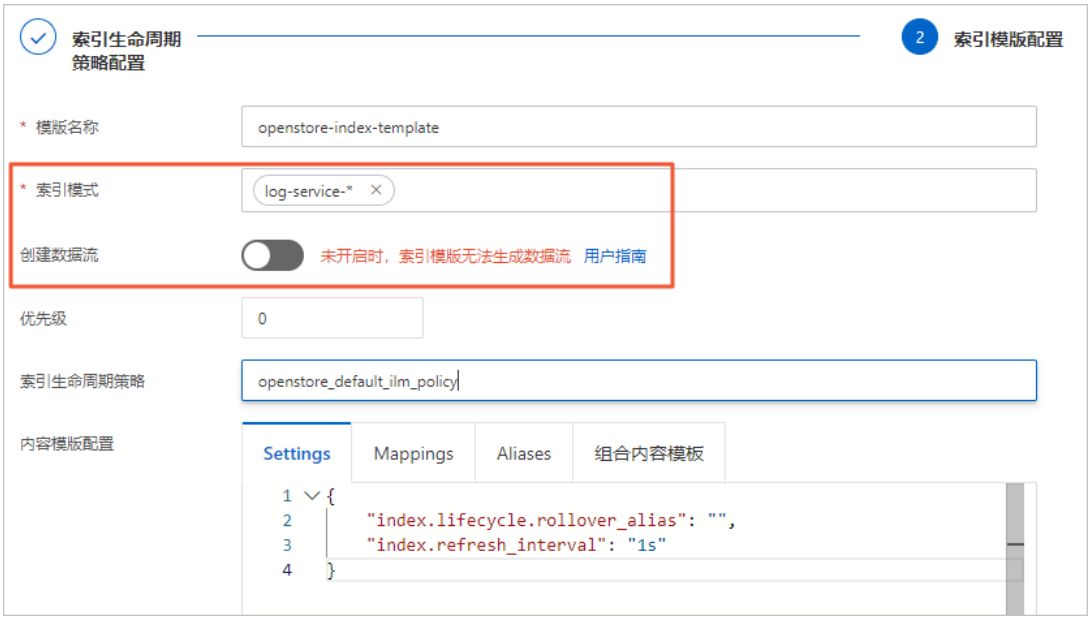
2. 单击**索引模板管理**页签。
3. 在**索引模板管理**页面，管理您集群中的索引模板。

   阿里云Elasticsearch为您提供了OpenStore定制索引模板，您可以查看并修改OpenStore定制的索引模板和生命周期策略，将定制模板快速应用到您的业务索引上。您也可以创建自定义的索引模板，并配置生命周期策略。具体说明如下：

   - 查看并修改OpenStore定制的索引模板和生命周期策略：
     a. 单击**openstore-index-template**或**openstore_default_ilm_policy**名称，查看OpenStore定制的索引模板和生命周期策略。

b. 单击**openstore-index-template**右侧的**修改**，修改定制的索引模板和生命周期策略。

定制化的OpenStore模板默认仅对**log-service-***索引进行管理。修改时可将**索引模式**指定为业务索引名称，并开启**创建数据流**，定制模板即可快速应用到业务索引上。



c. 单击**确认**。

- 创建自定义索引模板，并配置生命周期策略：

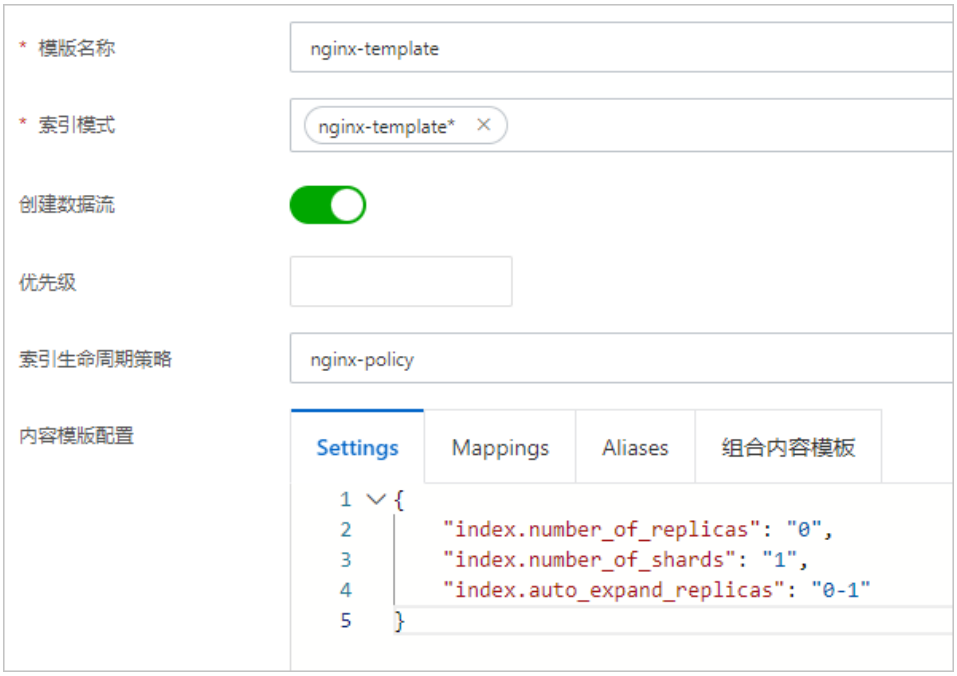
a. 单击**创建索引模板**。

b. 在**索引生命周期策略配置**向导中，参考下图配置索引生命周期策略。

详细参数说明，请参见索引模板管理。

> **注意**
> - Indexing Service实例建议开启滚动更新，保证数据滚动更新后自动取消云托管能力。
> - 系统默认开启**冷阶段**和**OpenStore存储**，如果您的索引需要进行冷热生命周期配置，请勿关闭**冷阶段**和**OpenStore存储**开关。

c. 单击**保存并下一步**，在**索引模板**配置向导中，参考下图配置索引模板。



详细参数说明，请参见索引模板管理。

> **注意** Indexing Service实例开启**创建数据流**后，才可在**数据流管理**页面管理写入的索引。详细信息请参见步骤三：将数据流写入OpenStore索引。

d. 单击**确认**。

通过API管理索引模板

从Elasticsearch 7.10版本开始，索引模板默认会使用优先级最高的配置，不会自动组合两个索引模板内的配置。由于集群中已经存在默认的OpenStore索引模板，如果您需要再通过API自定义索引模板，而这两个模板的配置不会自动组合，因此可能会影响OpenStore功能的正常使用，所以建议您使用Elasticsearch组合模板进行配置。

开启OpenStore存储后，集群中会默认添加OpenStore的组合模板component-openstore-index-template。在自定义模板时，您只需要在脚本的composed_of参数中配置依赖的组合模板，即可使用OpenStore存储。自定义索引模板的示例脚本如下。

> **说明**
>
> - 本文中的脚本均可在Kibana控制台上运行，具体操作请参见登录Kibana控制台。
> - 以下脚本中的 `...` 表示省略部分配置。
> - 如果您使用了自定义的组合模板和策略，请确保索引模板中配置的组合模板名称与您自定义的组合模板名称保持一致，即composed_of参数值配置为您自定义的组合模板名称。

```
PUT _index_template/template_instance-sls
{
  "index_patterns" : [
    "-.*"
  ],
  "template" : {
    "settings" : {
            ...
    },
    "mappings" : {
            ...
    }
  },
  # 如果您使用了自定义的组合模板和策略，此处需要配置为您自定义的组合模板名称。
  "composed_of" : ["component-openstore-index-template"],
  "priority" : 100
}
```

开启OpenStore存储的实例默认提供其依赖的内容模板和生命周期策略，您可以通过 `GET _component_template/component-openstore-index-template` 和 `GET _ilm/policy/openstore_default_ilm_policy` 命令获取默认的配置信息。获取到的配置信息如下：

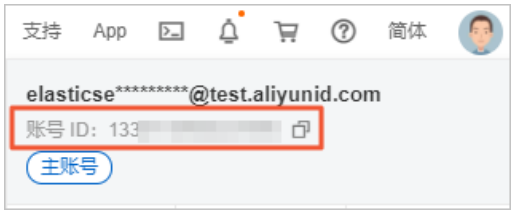
- component-openstore-index-template

```
{
  "component_templates" : [
    {
      "name" : "component-openstore-index-template",
      "component_template" : {
        "template" : {
          "settings" : {
            "index" : {
              "lifecycle" : {
                "name" : "openstore_default_ilm_policy",
                "rollover_alias" : ""
              },
              "apack" : {
                "cube" : {
                  "following_index" : "true"
                }
              },
              "codec" : "OpenIndex",
              "refresh_interval" : "1s"
            }
          }
        }
      }
    }
  ]
}
```

- openstore_default_ilm_policy

```
{
  "openstore_default_ilm_policy" : {
    "version" : 2,
    "modified_date" : "2022-03-16T06:33:42.802Z",
    "policy" : {
      "phases" : {
        "hot" : {
          "min_age" : "0s",
          "actions" : { }
        },
        "cold" : {
          "min_age" : "3d",
          "actions" : {
            "openstore" : {
              "openstore_repository" : "aliyun_auto_snapshot",
              "force_merge_index" : true,
              "user_id" : "1330710960******",
              "region_id" : "cn-hangzhou",
              "instance_id" : "es-cn-7mz2lpnaf0012****"
            },
            "set_priority" : {
              "priority" : 50
            }
          }
        }
      }
    }
  }
}
```

以下参数均为actions中的必选参数，详细说明如下。

参数	说明
openstore_repository	OpenStore存储仓库名称，固定为*aliyun_auto_snapshot*，不支持其他名称。
user_id	您阿里云账号的ID。在控制台上，将鼠标移至右侧头像处，获取**账号ID**。 
region_id	目标实例所在的地域ID。参见查看实例的基本信息，在实例的基本信息页面查看实例所在地域，并参见参数说明获取地域ID。
instance_id	目标实例的ID。参见查看实例的基本信息，在实例的基本信息页面查看实例的ID。
force_merge_index	是否执行forcemerge，必须设置为true，不能为false。设置为true，表示数据存储至OpenStore后，将处于只读状态。

> **注意** 阿里云Elasticsearch不支持通过Kibana Stack Management管理OpenStore内容模板及策略，建议使用API操作或控制台索引管理中心进行配置管理。

集群中已经配置了OpenStore部分的索引模板和生命周期策略名称，建议不要修改。如果您的业务需要自定义组合模板和策略，可以参考默认模板结构进行配置，示例如下：

- 索引模板

  以下示例创建了名称为zlcomponent-openstore-index-template的OpenStore索引模板，并使用了自定义的zlopenstore_default_ilm_policy策略。

```
PUT _component_template/zlcomponent-openstore-index-template
{
  "template" : {
    "settings" : {
      "index" : {
        "lifecycle" : {
          "name" : "zlopenstore_default_ilm_policy"
        },
        "apack" : {
          "cube" : {
            "following_index" : "true"
          }
        },
        "codec" : "OpenIndex87",
        "refresh_interval" : "1s"
      }
    }
  }
}
```

- 生命周期策略

  以下示例创建了名称为zlopenstore_default_ilm_policy的策略，并在模板的基础上添加了delete阶段的配置。

```
PUT _ilm/policy/zlopenstore_default_ilm_policy
{
    "policy" : {
      "phases" : {
        "hot" : {
          "min_age" : "0ms",
          "actions" : { }
        },
        "cold" : {
          "min_age" : "3d",
          "actions" : {
            "openstore" : {
              "openstore_repository" : "aliyun_auto_snapshot",
              "force_merge_index" : true,
              "user_id" : "1330710960******",
              "region_id" : "cn-hangzhou",
              "instance_id" : "es-cn-7mz2lpnaf0012****"
            },
            "set_priority" : {
              "priority" : 50
            }
          }
        },
          "delete": {
            "min_age": "40d",
            "actions": {
              "delete": {
                "delete_searchable_snapshot": true
              }
            }
          }
      }
    }
}
```

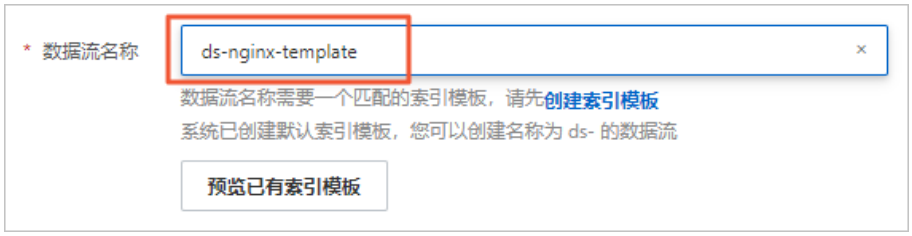
步骤三：将数据流写入OpenStore索引

1. 切换到**数据流管理**页签，单击**创建数据流**。

   > **注意** 仅7.10日志增强版Indexing Service系列的实例支持在控制台进行数据流管理，通用商业版7.10升级到日志增强版后不支持Indexing Service，因此也不支持在控制台进行数据流管理。建议您使用数据流相关API（CreateDataStream、RolloverDataStream、ListDataStreams、DeleteDataStream）管理数据。

2. 输入与索引模板匹配的**数据流名称**，单击**确定**。

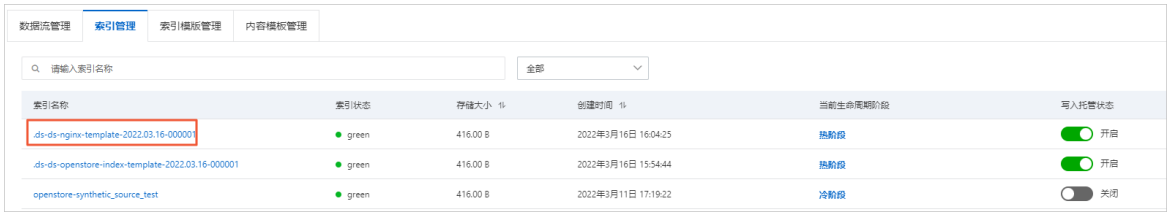
   数据流名称需要一个匹配的索引模板，此处需要输入ds-您在步骤二：管理OpenStore索引模板中定义的OpenStore索引模板名称。如果您忘记已创建的索引模板名称，可单击**预览已有索引模板**查看。



3. 登录Kibana控制台，通过bulk批量写入数据。

   在写入数据时，您可以通过设置写入的文档数量超过生命周期配置中**文件数限制**参数设置的值进行测试。**文件数限制**参数设置的值，可在步骤二：管理OpenStore索引模板中查看。

4. 切换至**索引管理**页签，查看OpenStore索引的写入托管状态、当前生命周期阶段等信息。



> **说明** 如果是非数据流场景，数据写入到OpenStore存储后，索引名称将以 `openstore-*` 开头，不影响业务查询。

常见问题

- Q：索引数据存储至OpenStore后，为什么无法写入更新，只能读取？

  A：集群提供的openstore_default_ilm_policy的actions中指定了force_merge_index参数为true，不可更改。当索引force_merge后，索引将处于只读状态，无法写入。

- Q：索引数据存储至OpenStore的过程中，集群状态为什么会变为异常（红色）？

  A：因为在索引数据复制进OpenStore冷存储的过程中，冷索引会处于写入状态，导致索引状态变为red，所以集群状态会变成红色。而在该复制过程中，热索引处于正常状态，并且是可持续对外提供服务的。当写入完成后，热索引被删除，冷索引恢复正常，此时集群也会恢复正常状态。

9.5.5. 基于TimeStream对接Prometheus+Grafana实现可观测性

TimeStream是阿里云Elasticsearch团队自研，并结合Elastic社区时序类产品特性共建的时序引擎。阿里云Elasticsearch支持无缝对接Prometheus+Grafana，支持Prometheus Query相关的API，可以直接将TimeStream索引作为Grafana的Prometheus数据源使用，能够提高时序指标数据存储与查询分析的性能，同时节约成本。本文介绍如何基于ElasticSearch TimeStream时序引擎对接Prometheus+Grafana实现云原生的可观测性。

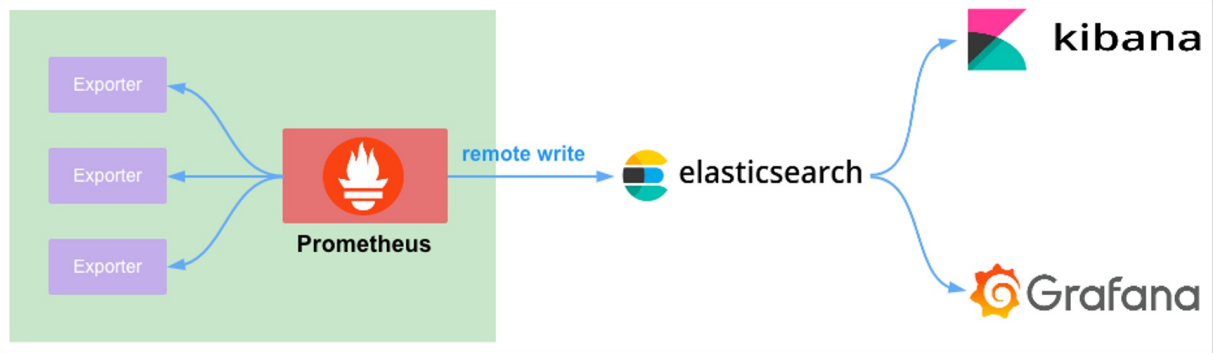
背景信息

Prometheus本地存储会遇到以下问题：

- 存储无副本，本地集群机器宕机后，Prometheus将无法访问。
- 单机存储，随着数据量不断上涨，可能遇到硬件瓶颈，无法横向扩容。
- 数据无备份能力，硬盘损坏后，数据可能无法恢复。
- 本地磁盘存储成本高，无法进行冷热分离存储。

因此在Prometheus的高可用方案中，在存储侧，推荐使用分布式、高可用的远端存储。阿里云Elasticsearch的TimeStream引擎提供了对Prometheus远端存储和查询的能力，基于Elasticsearch的分布式、弹性、高可用、备份和冷热分层存储等能力，可以作为Prometheus远端存储的最佳选择之一。

阿里云Elasticsearch与Prometheus和Grafana的结合方式如下图所示。



原理说明如下：

1. Prometheus收集各个Exporter的数据。
2. Prometheus通过remote write的方式将收集的数据同步到Elasticsearch。
3. 用户通过Kibana和Grafana查看Prometheus同步到Elasticsearch中的数据。

> **说明** 在使用Grafana访问Elasticsearch中的数据时，除了可以使用原生的Elasticsearch DataSource，还可以使用Prometheus的DataSource直接访问Elasticsearch数据，并使用PromQL来查看指标数据。

前提条件

已创建阿里云Elasticsearch实例，且实例版本为通用商业版7.16及以上，内核版本为1.7.0及以上。具体操作请参见创建阿里云Elasticsearch实例。

操作流程

1. 步骤一：环境准备
2. 步骤二：下载并启动node_exporter
3. 步骤三：下载、配置并启动Prometheus
4. 步骤四：下载、启动并配置Grafana Dashboard

步骤一：环境准备

1. 创建通用商业版7.16版本的阿里云Elasticsearch实例。

   具体操作请参见创建阿里云Elasticsearch实例。

2. 创建一个ECS实例，该实例要与步骤一中创建的Elasticsearch实例在相同专有网络下。

   创建ECS实例的具体操作，请参见使用向导创建实例。该ECS实例用来访问阿里云Elasticsearch实例，并部署Prometheus和Grafana，实现阿里云Elasticsearch与Prometheus和Grafana的结合。

3. 创建一个接收Prometheus数据的Elasticsearch TimeStream索引。

   i. 登录目标阿里云Elasticsearch实例的Kibana控制台，根据页面提示进入Kibana主页。

   登录Kibana控制台的具体操作，请参见登录Kibana控制台。

ii. 在左侧导航栏，单击  图标，然后选择**Management > 开发工具**。

iii. 在控制台中，执行 `PUT _time_stream/prom_index` 命令，创建名称为prom_index的TimeStream索引。

步骤二：下载并启动node_exporter

node_exporter用于收集各种与硬件和内核相关的指标，并提供给Prometheus进行读取，详细信息请参见node_exporter。

1. 连接ECS实例。

   具体操作请参见通过密码或密钥认证登录Linux实例。

2. 下载node_exporter安装包。

   本示例以node_exporter 1.3.1版本为例，下载命令如下。

   ```
   wget https://github.com/prometheus/node_exporter/releases/download/v1.3.1/node_exporter-1.3.1.linux-amd64.tar.gz
   ```

3. 解压安装包并启动node_exporter。

   ```
   tar xvfz node_exporter-1.3.1.linux-amd64.tar.gz
   cd node_exporter-1.3.1.linux-amd64
   ./node_exporter
   ```

步骤三：下载、配置并启动Prometheus

1. 连接ECS实例。

   具体操作请参见通过密码或密钥认证登录Linux实例。

2. 在根目录下载Prometheus安装包。

   本示例以Prometheus 2.36.2版本为例，下载命令如下。

   ```
   cd ~
   wget https://github.com/prometheus/prometheus/releases/download/v2.36.2/prometheus-2.36.2.linux-amd64.tar.gz
   ```

3. 解压Prometheus安装包。

   ```
   tar xvfz prometheus-2.36.2.linux-amd64.tar.gz
   ```

4. 在Prometheus目录的*prometheus.yml*文件中，配置node_exporter和remote_write。

   ```
   cd prometheus-2.36.2.linux-amd64
   vim prometheus.yml
   ```

   配置示例如下。

```
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from thi
s config.
  - job_name: 'prometheus'
    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.
    static_configs:
    - targets: ['localhost:9090']
  #配置node_exporter
  - job_name: "node"
    static_configs:
    - targets: ["127.0.0.1:9100"]
#配置remote_write，确保Prometheus能够访问Elasticsearch集群，即网络是通的。
remote_write:
  - url: "http://xxx:9200/_time_stream/prom_write/prom_index"
    basic_auth:
     username: elastic
     password: xxxx
```

参数	说明
node_exporter	配置node_exporter的连接信息。targets需要配置为node_exporter的访问地址:端口。 由于本示例使用同一个ECS实例部署Prometheus和node_exporter，因此node_exporter的访问地址使用本地访问IP地址127.0.0.1，端口使用默认的9100端口。

参数	说明
remote_write	配置Elasticsearch实例的TimeStream索引的连接信息。需要配置以下基础参数，更多高级参数请可见remote_write。 ○ url：访问TimeStream索引的URL，格式为：http://<Elasticsearch实例的公网或私网访问地址>:9200/_time_stream/prom_write/<yourTimeStreamIndex>。 说明 ■ Elasticsearch实例的公网或私网访问地址：可在Elasticsearch实例的基本信息页面获取。如果Prometheus所部署的ECS实例与Elasticsearch实例在同一VPC下，可使用私网访问地址（本文以此为例）；如果不在同一VPC下，需要使用公网访问地址，并且要配置公网访问白名单，详细信息请参见配置实例公网或私网访问白名单。 ■ <yourTimeStreamIndex>：用于接收Prometheus数据的Elasticsearch TimeStream索引，该索引需要提前创建，本文以prom_index索引为例。 ○ username：访问TimeStream索引的用户名，默认为管理员账号elastic。您也可以使用自建用户，但需确保自建用户具有访问与操作TimeStream索引的权限，详细信息请参见通过Elasticsearch X-Pack角色管理实现用户权限管控。 ○ password：访问TimeStream索引的用户对应的密码。elastic账号的密码在创建实例时设定，如果忘记可重置，重置密码的注意事项和操作步骤请参见重置实例访问密码。

5. 启动Prometheus。

```
./prometheus
```

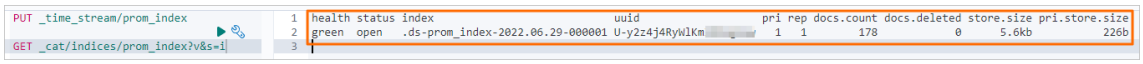
6. 验证Prometheus数据是否已经同步到Elasticsearch的TimeStream索引中。

   在Elasticsearch的Kibana控制台中，执行以下命令进行验证：

   ○ 查看prom_index索引是否已经有数据。

   ```
   GET _cat/indices/prom_index?v&s=i
   ```

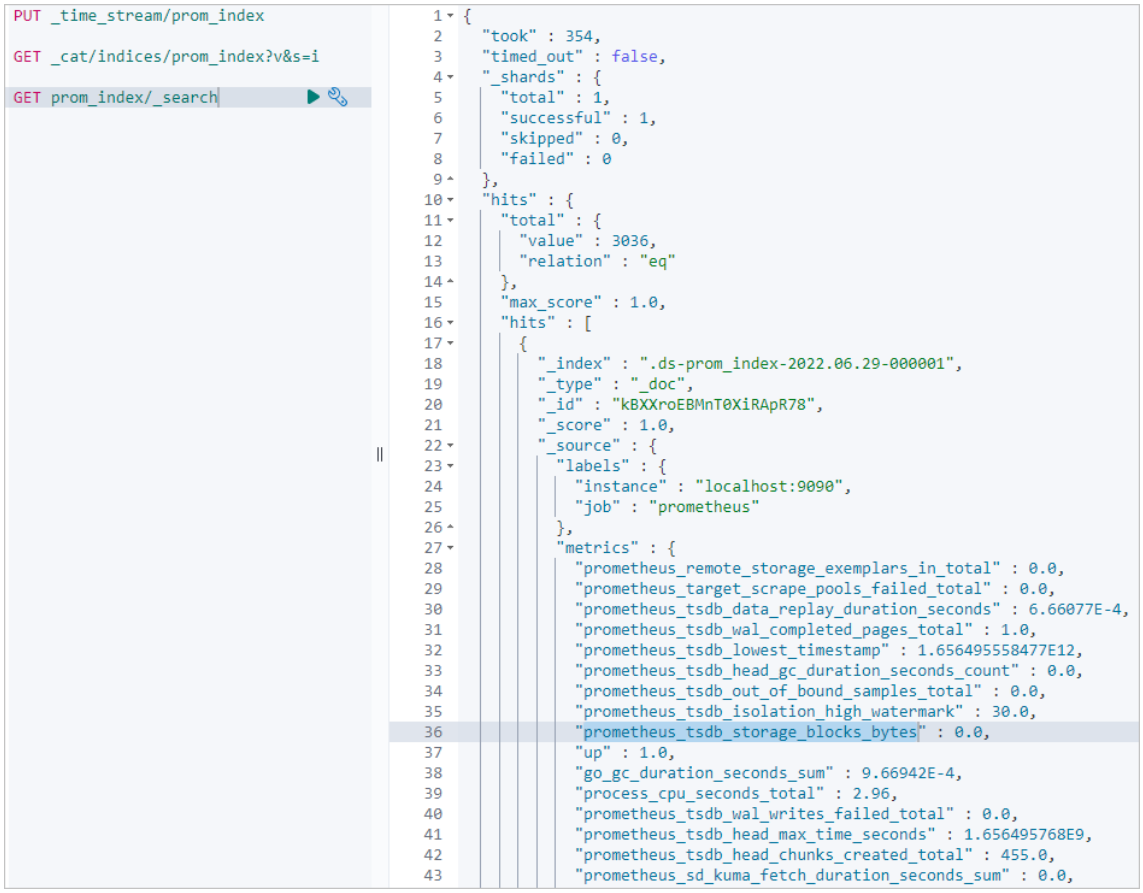
   预期结果如下。

   ○ 确认是否能查询到数据并查看数据内容。

   ```
   GET prom_index/_search
   ```

预期结果如下。



步骤四：下载、启动并配置Grafana Dashboard

1. 连接ECS实例。

   具体操作请参见通过密码或密钥认证登录Linux实例。

2. 在根目录下载Grafana安装包。

   本示例以Grafana 9.0.2版本为例，下载命令如下。

   ```
   cd ~
   wget https://dl.grafana.com/enterprise/release/grafana-enterprise-9.0.2.linux-amd64.tar.gz
   ```

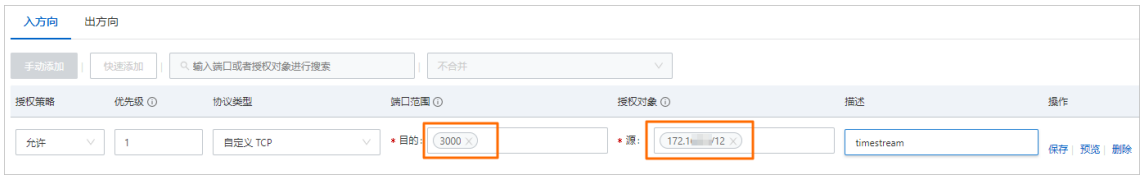
3. 解压Grafana安装包并启动。

   ```
   tar xvfz grafana-enterprise-9.0.2.linux-amd64.tar.gz
   cd grafana-9.0.2
   ./bin/grafana-server
   ```

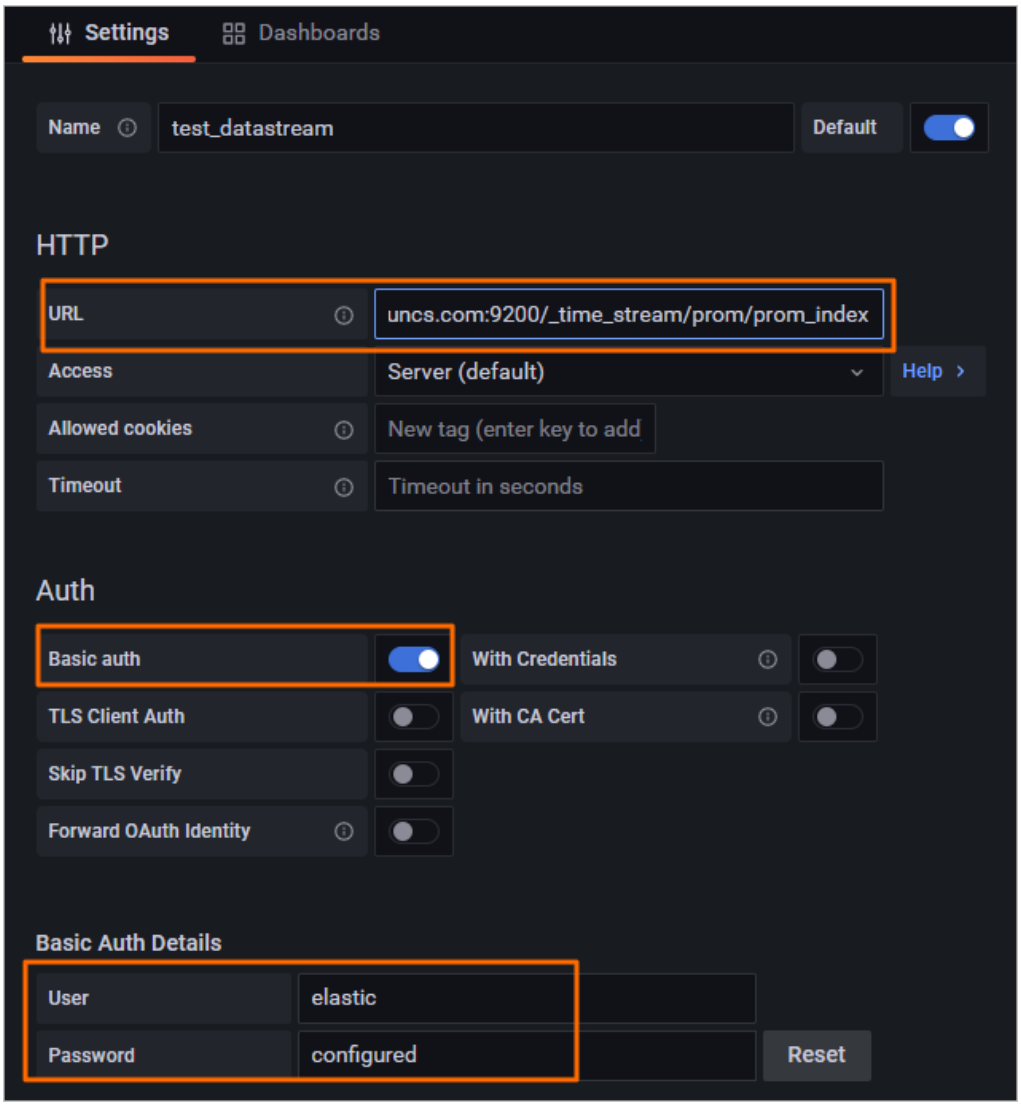
4. 在浏览器中输入Grafana的访问地址 `http://<ECS的公网IP地址>:3000` ，进入Grafana登录页面，输入用户名和密码进入Grafana控制台。

   - 首次登录Grafana控制台需要使用默认用户名和密码，均为admin。登录后，系统会提示修改密码，密码修改完成后即可进入Grafana控制台。
   - ECS的公网IP地址：进入ECS管理控制台，在对应实例的**IP地址**列获取。

- Grafana的默认访问端口号为3000，在浏览器中访问3000端口时，需要配置ECS的入方向的安全组规则，设置**目的**为3000，**源**为您客户端的IP地址。详细信息请参见添加安全组规则。



5. 在Grafana中，创建Prometheus的DataSource。
   i. 在Grafana控制台的左侧导航栏，选择 > **Data sources**。
   ii. 在**Data sources**页签，单击**Add data source**。
   iii. 在**Time series databases**列表中，单击**Prometheus**。
   iv. 在**Settings**页签中，配置**Prometheus**数据源信息。



本文中必须配置的参数说明如下。

参数	说明
URL	访问TimeStream索引的URL，格式为：http://<Elasticsearch实例的公网或私网访问地址>:9200/_time_stream/prom/<yourTimeStreamIndex>。 **说明** ■ Elasticsearch实例的公网或私网访问地址：可在Elasticsearch实例的基本信息页面获取。如果Prometheus所部署的ECS实例与Elasticsearch实例在同一VPC下，可使用私网访问地址（本文以此为例）；如果不在同一VPC下，需要使用公网访问地址，并且要配置公网访问白名单，详细信息请参见配置实例公网或私网访问白名单。 ■ <yourTimeStreamIndex>：用于接收Prometheus数据的Elasticsearch TimeStream索引，该索引需要提前创建，本文以prom_index索引为例。
Basic auth	是否开启Elasticsearch实例的Basic auth认证。开启后需要配置访问Elasticsearch实例的用户名和密码。
User	访问TimeStream索引的用户名，默认为管理员账号elastic。您也可以使用自建用户，但需确保自建用户具有访问与操作TimeStream索引的权限，详细信息请参见通过Elasticsearch X-Pack角色管理实现用户权限管控。
Password	访问TimeStream索引的用户对应的密码。elastic账号的密码在创建实例时设定，如果忘记可重置，重置密码的注意事项和操作步骤请参见重置实例访问密码。

v. 单击Save&test。

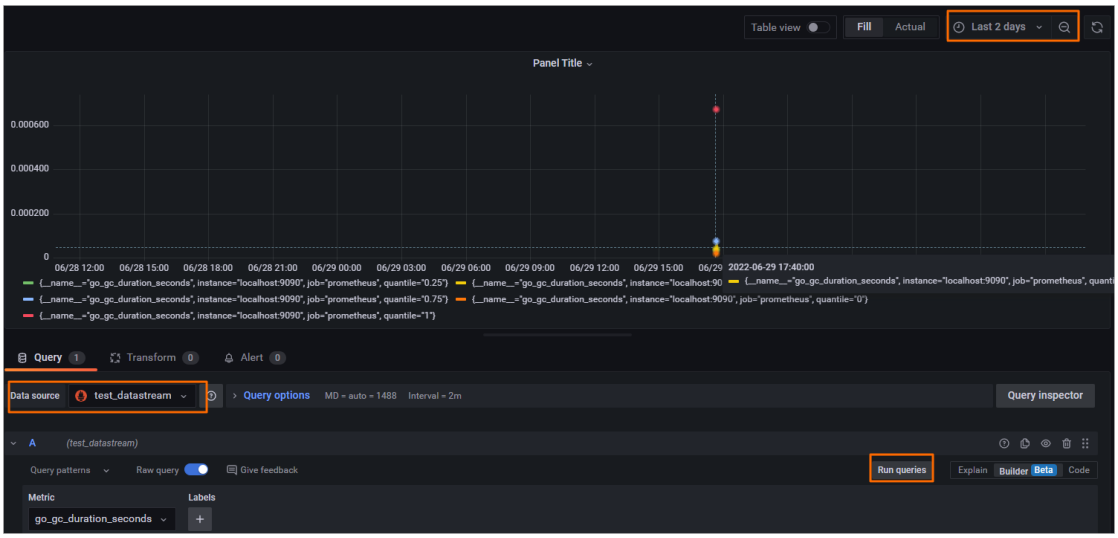
配置成功后，系统提示Data source is working。

6. 在Grafana中，创建展示Prometheus数据源的Dashboard。

i. 在Grafana控制台的左侧导航栏，选择  > New dashboard。

ii. 单击Add a new panel。

iii. 选择Data source和查询时间，单击Run queries查询数据。

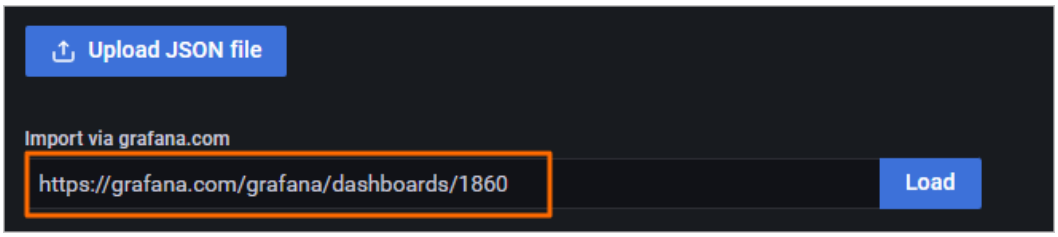


iv. 单击右上角的Save，保存Dashboard。

7. 在Grafana中，导入node_exporter自带的Grafana Dashboard，并配置Prometheus数据源，生成指标监控Dashboard。

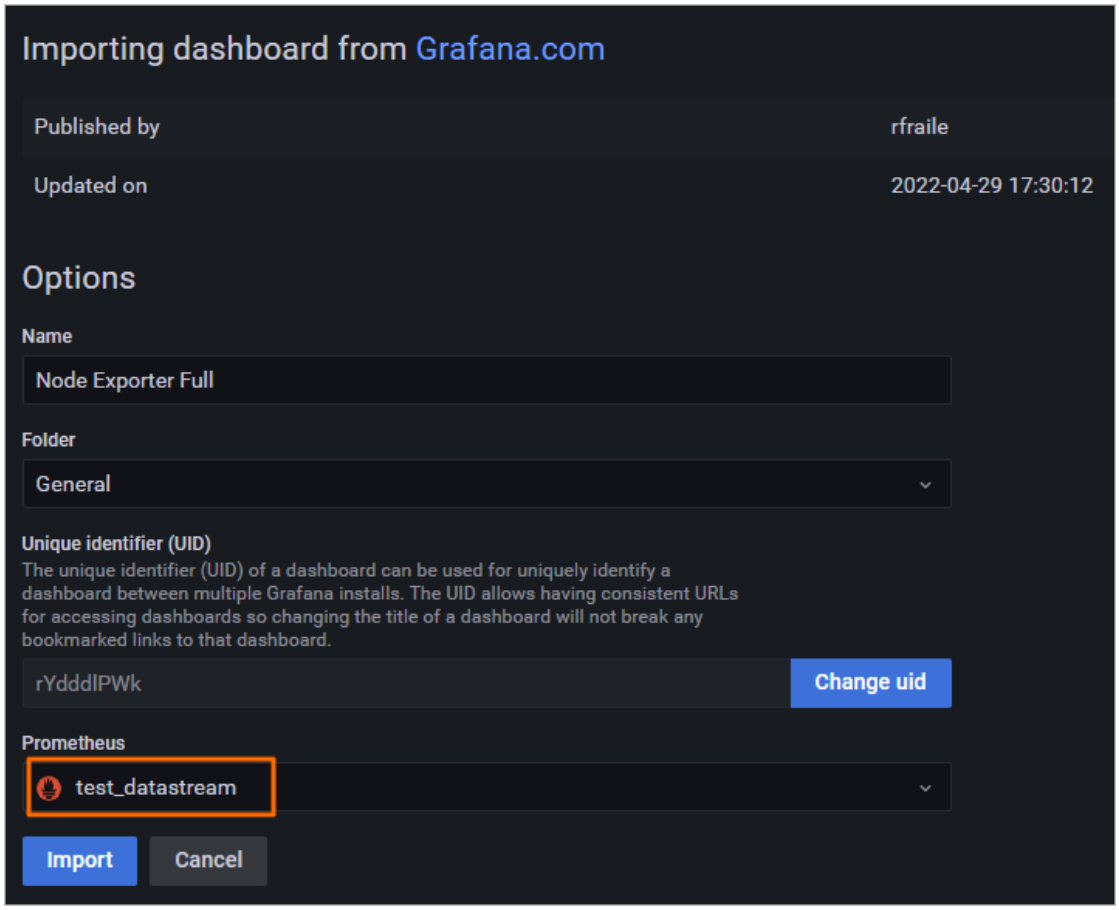
i. 在Grafana控制台的左侧导航栏，选择 > Import 。

ii. 在Import via grafana.com文本框中，填写node_exporter的Grafana地址或ID：即https://grafana.com/grafana/dashboards/1860或1860。



iii. 单击Load。

iv. 在配置页面选择Prometheus数据源为您已创建的数据源。



v. 单击Import。

vi. 在Dashboard页面右上角，选择查询时间，查看对应时间段内的指标监控Dashboard。

> **说明** 关于Grafana更详细的操作教程，请参见Grafana documentation。

相关文档

- TimeStream管理Elasticsearch时序数据快速入门

- 使用Aliyun-TimeStream插件
- TimeStream集成Prometheus接口

9.6. 数据管理与可视化

9.6.1. 基于Terraform管理阿里云Elasticsearch最佳实践

通过Terraform，您可以使用代码配置实现物理机等资源的分配。也就是说通过Terraform，写一个配置文件，就可以帮助您购买一台云服务器，或者申请到阿里云Elasticsearch、OSS等云资源。本文介绍通过Terraform管理阿里云Elasticsearch的方法，包括创建、更新、查看、删除实例等操作。

背景信息

您可以通过以下两种方式安装并配置Terraform环境：

- 在本地安装和配置Terraform（本文以此为例）。
- 在Cloud Shell中使用Terraform。

本文介绍通过Terraform管理阿里云Elasticsearch的方法，主要包括：

- 安装并配置Terraform
- 通过Terraform创建阿里云Elasticsearch实例
- 通过Terraform更新Elasticsearch资源配置
- 将Elasticsearch资源导入Terraform
- 查看Terraform管理的所有Elasticsearch资源
- 通过Terraform删除Elasticsearch实例

安装并配置Terraform

1. 前往Terraform官网，下载适用于您的操作系统的程序包。

   本文以Linux系统为例。如果您还没有Linux环境，可购买阿里云ECS实例，详情请参见步骤一：创建ECS实例。

2. 将程序包解压到*/usr/local/bin*目录。

   如果您需要将可执行文件解压到其他目录，请按照以下方法为其定义全局路径：

   - Linux：参见在Linux系统定义全局路径
   - Windows：参见在Windows系统定义全局路径
   - Mac：参见在Mac系统定义全局路径

3. 执行 `terraform` 命令验证路径配置。

   执行成功后，返回如下结果。

```
[root@elastic64 ~]# terraform
Usage: terraform [-version] [-help] <command> [args]

The available commands for execution are listed below.
The most common, useful commands are shown first, followed by
less common or more advanced commands. If you're just getting
started with Terraform, stick with the common commands. For the
other commands, please read the help and docs before usage.

Common commands:
    apply              Builds or changes infrastructure
    console            Interactive console for Terraform interpolations
    destroy            Destroy Terraform-managed infrastructure
    env                Workspace management
    fmt                Rewrites config files to canonical format
    get                Download and install modules for the configuration
    graph              Create a visual graph of Terraform resources
    import             Import existing infrastructure into Terraform
    init               Initialize a Terraform working directory
    output             Read an output from a state file
    plan               Generate and show an execution plan
    providers          Prints a tree of the providers used in the configuration
    refresh            Update local state file against real resources
    show               Inspect Terraform state or plan
    taint              Manually mark a resource for recreation
    untaint            Manually unmark a resource as tainted
    validate           Validates the Terraform files
    version            Prints the Terraform version
    workspace          Workspace management
```

4. 创建RAM用户，并为其授权。

   为提高权限管理的灵活性和安全性，建议您创建RAM用户，并为其授权。

   i. 登录RAM控制台。

   ii. 创建名为Terraform的RAM用户，并为该用户创建AccessKey。

   具体操作方法请参见创建RAM用户。

   > **注意** 请不要使用阿里云账号的AccessKey配置Terraform工具。

   iii. 为RAM用户授权。

   本示例为用户Terraform授予AliyunElasticsearchFullAccess和AliyunVPCFullAccess权限，具体操作方法请参见为RAM用户授权。

5. 创建测试目录。

   因为每个Terraform项目都需要创建一个独立的执行目录，所以需要先创建一个测试目录。以下创建一个名为terraform-test的测试目录。

   ```
   mkdir terraform-test
   ```

6. 进入terraform-test目录。

   ```
   cd terraform-test
   ```

7. 创建配置文件，并配置身份认证信息。

   Terraform在运行时，会读取该目录下所有的*.tf和*.tfvars文件。请按照实际需求，将配置信息写入到不同的文件中。以下列出几个常用的配置文件。

配置文件	说明
provider.tf	provider配置。
terraform.tfvars	配置provider要用到的变量。
varable.tf	通用变量。
resource.tf	资源定义。
data.tf	包文件定义。
output.tf	输出文件定义。

例如创建provider.tf文件时，使用 `vim provider.tf` 打开文件，并按照以下格式配置您的身份认证信息。

```
provider "alicloud" {
    region      = "cn-hangzhou"
    access_key  = "LTA**********NO2"
    secret_key   = "MOk8x0**********wwff"
    }
```

更多配置信息请参见alicloud_elasticsearch_instance。

8. 使用 `mkdir -p plugh` 命令，在当前目录下创建plugh目录，下载provider插件并解压到plugh目录下。
9. 初始化工作目录，使用-plugin-dir指定provider所在的路径，完成配置。

```
terraform init -plugin-dir=./plugh/
```

返回Terraform has been successfully initialized表示初始化成功。

> 注意 每个Terraform项目在新建Terraform工作目录并创建配置文件后，都需要初始化工作目录。

通过Terraform创建阿里云Elasticsearch实例

1. 在测试目录下，创建一个elastic.tf配置文件。
2. 参考以下脚本配置elastic.tf文件，创建一个跨可用区的通用商业版6.7版本的阿里云Elasticsearch实例。

```
resource "alicloud_elasticsearch_instance" "instance" {
  description          = "testInstanceName"
  instance_charge_type = "PostPaid"
  data_node_amount     = "2"
  data_node_spec       = "elasticsearch.sn2ne.large"
  data_node_disk_size  = "20"
  data_node_disk_type  = "cloud_ssd"
  vswitch_id           = "vsw-bp1f7r0ma00pf9h2l****"
  password             = "es_password"
  version              = "6.7_with_X-Pack"
  master_node_spec     = "elasticsearch.sn2ne.large"
  zone_count           = "1"
}
```

provider插件支持的所有参数说明如下。

参数	是否必选	描述
description	否	实例自定义名称的描述。
instance_charge_type	否	计费模式。可选值： ○ PostPaid（默认）：按量付费 ○ PrePaid：包年包月
period	否	购买时长（单位：月），当instance_charge_type为PrePaid时有效。可选值：1~9、12、24、36，默认是1个月。
data_node_amount	是	ES集群的数据节点的个数。可选值：2~50。
data_node_spec	是	数据节点实例规格。
data_node_disk_size	是	指定磁盘空间。不同类型的磁盘，支持的最大存储空间大小不同： ○ cloud_ssd：SSD云盘，支持最大存储2048 GB（2 TB）。 ○ cloud_efficiency：高效云盘，支持最大5 TB的存储空间，提供较为低廉的存储能力，适合大规模数据量的日志及分析场景。高效云盘超过2048 GB时，只能取：2560、3072、3584、4096、4608、5120。
data_node_disk_type	是	存储类型。可选值： ○ cloud_ssd：SSD云盘。 ○ cloud_efficiency：高效云盘。
vswitch_id	是	虚拟交换机的实例ID。
password	否	实例密码，支持大小写字母、数字、特殊字符，长度为8~32位字符。特殊字符包括 `!@#$%^&*()_+-=` 。
kms_encrypted_password	否	KMS加密密码。如果配置了password，该字段将被忽略。password和kms_encrypted_password必须配置一个。
kms_encryption_context	否	KMS加密上下文。只有设置了kms_encrypted_password时才有效。用于对使用kms_encrypted_password加密创建或更新的实例进行解密，详情请参见encryption context。

参数	是否必选	描述
version	是	Elasticsearch版本。可选值： ○ 5.5.3_with_X-Pack：5.5.3版本。 ○ 6.3_with_X-Pack：6.3.0版本。 ○ 6.7_with_X-Pack：6.7.0版本。
private_whitelist	否	设置实例的专有网络VPC（Virtual Private Cloud）网络白名单。
kibana_whitelist	否	设置Kibana访问白名单。
master_node_spec	否	Master节点规格。
advancedDedicateMaster	否	用于表示是否创建专有主节点，取值含义如下： ○ true：创建专有主节点。如果部署多可用区并且启用专有主节点，则需要将该参数设置为true。 ○ false（默认值）：不创建专有主节点。
zone_count	否	可用区数量。取值为1~3，data_node_amount必须是该值的整数倍。

更多参数详情请参见alicloud_elasticsearch_instance。

> **注意**
> - kms_encrypted_password和kms_encryption_context参数要求provider插件版本在1.57.1及以上；zone_count参数要求provider插件版本在1.44.0及以上。
> - 如果需要购买除数据节点外的其他属性节点，请参见createInstance参数开启其他节点属性。例如，购买多可用区专有主节点，脚本中需要加入 `advancedDedicateMaster="true"` 。

3. 执行 `terraform plan` 命令，查看将会执行的操作。
   执行成功后，返回如下结果。

```
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.
------------------------------------------------------------------------
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create
Terraform will perform the following actions:
  # alicloud_elasticsearch_instance.instance will be created
  + resource "alicloud_elasticsearch_instance" "instance" {
      + description          = "testInstanceName"
      + data_node_amount     = 2
      + data_node_disk_size  = 20
      + data_node_disk_type  = "cloud_ssd"
      + data_node_spec       = "elasticsearch.sn2ne.large"
      + domain               = (known after apply)
      + id                   = (known after apply)
      + instance_charge_type = "PostPaid"
      + kibana_domain        = (known after apply)
      + kibana_port          = (known after apply)
      + kibana_whitelist     = (known after apply)
      + master_node_spec     = "elasticsearch.sn2ne.large"
      + password             = (sensitive value)
      + port                 = (known after apply)
      + private_whitelist    = (known after apply)
      + public_whitelist     = (known after apply)
      + status               = (known after apply)
      + version              = "6.7_with_X-Pack"
      + vswitch_id           = "vsw-bp1f7r0ma00pf9h21****"
      + zone_count           = 1
    }
Plan: 1 to add, 0 to change, 0 to destroy.
------------------------------------------------------------------------
Note: You didn't specify an "-out" parameter to save this plan, so Terraform
can't guarantee that exactly these actions will be performed if
"terraform apply" is subsequently run.
```

4. 执行 `terraform apply` 命令，运行工作目录中的配置文件，输入*yes*。

   执行成功后，返回如下结果。

```
Plan: 1 to add, 0 to change, 0 to destroy.
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.
  Enter a value: yes
 alicloud_elasticsearch_instance.instance: Creating...
 alicloud_elasticsearch_instance.instance: Still creating... [10s elapsed]
alicloud_elasticsearch_instance.instance: Still creating... [20s elapsed]
 ..............
 Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

5. 登录阿里云Elasticsearch控制台，查看创建成功的Elasticsearch集群。

通过Terraform更新Elasticsearch资源配置

1. 进入测试目录，修改elastic.tf配置文件。

   例如修改data_node_disk_size为50。

```
resource "alicloud_elasticsearch_instance" "instance" {
  instance_charge_type = "PostPaid"
  data_node_amount     = "2"
  data_node_spec       = "elasticsearch.sn2ne.large"
  data_node_disk_size  = "50"
  data_node_disk_type  = "cloud_ssd"
  vswitch_id           = "vsw-bp1f7r0ma00pf9h2l****"
  password             = "es_password"
  version              = "6.7_with_X-Pack"
  master_node_spec     = "elasticsearch.sn2ne.large"
  zone_count           = "1"
}
```

   注意

   - 实例创建成功后，version无法修改。
   - 每次请求，只支持修改一项配置。例如同时修改data_node_spec和data_node_disk_size，系统将会出现错误响应。

2. 执行 `terraform plan` 查看资源配置信息。
3. 执行 `terraform apply` 等待资源升配结束。

将Elasticsearch资源导入Terraform

如果阿里云Elasticsearch实例不是通过Terraform创建的，可通过命令，将阿里云Elasticsearch导入到Terraform的state目录下进行管理。

1. 在测试目录下，创建一个main.tf文件。

```
vim main.tf
```

2. 进行资源声明，指定所要导入的资源在state中的存放路径。

```
resource "alicloud_elasticsearch_instance" "test" {}
```

3. 开始资源导入操作。

```
terraform import alicloud_elasticsearch_instance.test  es-cn-0pp1f1y5g000h****
```

   执行成功后，返回如下结果。

```
alicloud_elasticsearch_instance.test: Importing from ID "es-cn-0pp1f1y5g000h****"...
alicloud_elasticsearch_instance.test: Import prepared!
  Prepared alicloud_elasticsearch_instance for import
alicloud_elasticsearch_instance.test: Refreshing state... [id=es-cn-0pp1f1y5g000h****]
Import successful!
The resources that were imported are shown above. These resources are now in
your Terraform state and will henceforth be managed by Terraform.
```

> 说明 有关import如何实现存量资源的管理，请参见一文揭秘存量云资源的管理难题。

查看Terraform管理的所有Elasticsearch资源

使用 `terraform show` 命令，查看当前state中所有被管理的资源及其所有属性值。

```
# alicloud_elasticsearch_instance.instance:
resource "alicloud_elasticsearch_instance" "instance" {
    data_node_amount     = 2
    data_node_disk_size  = 20
    data_node_disk_type  = "cloud_ssd"
    data_node_spec       = "elasticsearch.sn2ne.large"
    domain               = "es-cn-dssf9op81lz4q****.elasticsearch.aliyuncs.com"
    id                   = "es-cn-dssf9op81lz4q****"
    instance_charge_type = "PostPaid"
    kibana_domain        = "es-cn-dssf9op81lz4q****.kibana.elasticsearch.aliyuncs.com"
    kibana_port          = 5601
    kibana_whitelist     = []
    master_node_spec     = "elasticsearch.sn2ne.large"
    password             = (sensitive value)
    port                 = 9200
    private_whitelist    = []
    public_whitelist     = []
    status               = "active"
    version              = "6.7.0_with_X-Pack"
    vswitch_id           = "vsw-bp1f7r0ma00pf9h2l****"
    zone_count           = 1
}
# alicloud_elasticsearch_instance.test:
resource "alicloud_elasticsearch_instance" "test" {
    data_node_amount     = 3
    data_node_disk_size  = 51
    data_node_disk_type  = "cloud_ssd"
    data_node_spec       = "elasticsearch.r5.large"
    domain               = "es-cn-0pp1f1y5g000h****.elasticsearch.aliyuncs.com"
    id                   = "es-cn-0pp1f1y5g000h****"
    instance_charge_type = "PostPaid"
    kibana_domain        = "es-cn-0pp1f1y5g000h****.kibana.elasticsearch.aliyuncs.com"
    kibana_port          = 5601
    kibana_whitelist     = []
    port                 = 9200
    private_whitelist    = []
    public_whitelist     = []
    status               = "active"
    version              = "6.7.0_with_X-Pack"
    vswitch_id           = "vsw-bp1f7r0ma00pf9h2l****"
    zone_count           = 1
    timeouts {}
}
```

通过Terraform删除Elasticsearch实例

警告 实例删除后将不可恢复，实例中的所有数据将被清空。

进入测试目录，执行 `terraform destroy` 命令，输入*yes*，即可删除该实例。

```
# terraform destroy
alicloud_elasticsearch_instance.instance: Refreshing state... [id=es-cn-v3x49h5397fau****]
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  - destroy
Terraform will perform the following actions:
  # alicloud_elasticsearch_instance.instance will be destroyed
  - resource "alicloud_elasticsearch_instance" "instance" {
      - data_node_amount     = 2 -> null
      - data_node_disk_size  = 20 -> null
      - data_node_disk_type  = "cloud_ssd" -> null
      - data_node_spec       = "elasticsearch.sn2ne.large" -> null
      - domain               = "es-cn-v3x49h5397fau****.elasticsearch.aliyuncs.com" -> null
      - id                   = "es-cn-v3x49h5397fau****" -> null
      - instance_charge_type = "PostPaid" -> null
      - kibana_domain        = "es-cn-v3x49h5397fau****.kibana.elasticsearch.aliyuncs.com" -
> null
      - kibana_port          = 5601 -> null
      - kibana_whitelist     = [] -> null
      - master_node_spec     = "elasticsearch.sn2ne.large" -> null
      - password             = (sensitive value)
      - port                 = 9200 -> null
      - private_whitelist    = [] -> null
      - public_whitelist     = [] -> null
      - status               = "active" -> null
      - version              = "6.7.0_with_X-Pack" -> null
      - vswitch_id           = "vsw-bp1f7r0ma00pf9h2l****" -> null
      - zone_count           = 1 -> null
    }
Plan: 0 to add, 0 to change, 1 to destroy.
Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.
  Enter a value: yes
alicloud_elasticsearch_instance.instance: Destroying... [id=es-cn-v3x49h5397fau****]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****, 1
0s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****, 2
0s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****, 3
0s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****, 4
0s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****, 5
0s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****, 1
m0s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****, 1
m10s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****, 1
```

```
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****, 1
m20s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****, 1
m30s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****, 1
m40s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****, 1
m50s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****, 2
m0s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****, 2
m10s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****, 2
m20s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****, 2
m30s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****, 2
m40s elapsed]
alicloud_elasticsearch_instance.instance: Destruction complete after 10m2s
Destroy complete! Resources: 1 destroyed.
```

9.6.2. 通过_split API快速拆分主分片

当您使用Elasticsearch集群出现索引分片设置不合理（例如索引主分片设置不合理、每个分片存在大量数据等）引发集群性能问题时，可通过 `_split` API在线扩大主分片数，将现有索引拆分为具有更多主分片的索引。本文介绍如何通过 `_split` API快速拆分主分片。

背景信息

索引创建后，Elasticsearch不支持修改索引主分片的数量，如果需要修改，一般会使用reindex重建索引，耗时太久。而在6.x版本开始，Elasticsearch支持在线扩大主分片数的Split index API，支持将现有索引拆分为具有更多主分片的索引。

reindex与 `_split` API的性能测试信息如下：

- 测试环境：
  - 数据节点：数量为5个，规格为8核16 GB。
  - 索引数据：数据量为183 GB。
  - 分片数：原主分片数为5，目标分片数为20，副本数为0。
- 测试结果

方式	耗时	资源占用
reindex	2.5小时	集群中有大量的写QPS，索引所占节点资源高。
`_split` API	3分钟	集群数据节点CPU使用率为78%左右，load_1m为10左右。

前提条件

- Elasticsearch集群状态健康，且负载处于正常水位。
- 根据集群数据节点个数、集群磁盘容量等因素，合理评估索引可拆分的分片数。详细信息请参见Shard评

估。

- 待拆分的索引禁止写入，且拆分后生成的新索引不存在。
- Elasticsearch集群有足够的磁盘空间存储拆分后生成的新索引。

操作步骤

1. 登录目标阿里云Elasticsearch实例的Kibana控制台，根据页面提示进入Kibana主页。

   登录Kibana控制台的具体操作，请参见登录Kibana控制台。

   > **说明** 本文以阿里云Elasticsearch 7.10.0版本为例，其他版本操作可能略有差别，请以实际界面为准。

2. 单击右上角的**Dev tools**。
3. 在Console中，执行如下命令，在创建索引时指定index.number_of_routing_shards，设置索引可拆分的分片数。

   以下示例以7.10版本实例为例，在创建dest1索引时指定分片路由数number_of_routing_shards，对主分片进行拆分。可拆分的主分片数需要为number_of_routing_shards的一个因数且为number_of_shards（主分片数）的倍数。如下number_of_shards为2，number_of_routing_shards为24，则可拆分的主分片数支持：4、6、8、12、24。

   > **说明** 使用时需要将dest1替换为您的业务索引名。

   ```
   PUT /dest1
   {
     "settings": {
       "index": {
         "number_of_routing_shards": 24,
         "number_of_shards":2
       }
     }
   }
   ```

参数	说明

参数	说明
number_of_routing_shards	路由分片数，定义索引可拆分的次数或原始分片可拆分的分片数。创建索引指定该参数，要求索引主分片数必须是路由分片数的一个因数。 **说明** • Elasticsearch 7.0以下版本，创建索引时必须指定index.number_of_routing_shards，最大值为1024；7.0及以上版本，index.number_of_routing_shards值默认依赖主分片数，如果创建索引时未指定，默认按因子2拆分，并且最多可拆分为1024个分片。例如原索引主分片数为1，则可拆分为1~1024中的任意数；原索引主分片为5，则支持拆分的分片数为：10、20、40、80、160、320以及最大数640（不能超过1024）。 • 经过shrink后的主分片再进行split时，主分片数为原主分片数的倍数即可。例如原分片数为5，进行split时，支持拆分的分片数为：10、15、20、25、30...，最大不能超过1024。
number_of_shards	索引的主分片数。

4. 插入数据。

   说明 以下数据仅供测试。

```
POST /dest1/_doc/_bulk
{"index":{}}
{"productName":"大健康天天理财","annual_rate":"3.2200%","describe":"180天定期理财，最低20000起投，收益稳定，可以自助选择消息推送"}
{"index":{}}
{"productName":"西部通宝","annual_rate":"3.1100%","describe":"90天定投产品，最低10000起投，每天收益到账消息推送"}
{"index":{}}
{"productName":"安详畜牧产业","annual_rate":"3.3500%","describe":"270天定投产品，最低40000起投，每天收益立即到账消息推送"}
{"index":{}}
{"productName":"5G设备采购月月盈","annual_rate":"3.1200%","describe":"90天定投产品，最低12000起投，每天收益到账消息推送"}
{"index":{}}
{"productName":"新能源动力理财","annual rate":"3.0100%","describe":"30天定投产品推荐，最低8000起投，每天收益会消息推送"}
{"index":{}}
{"productName":"微贷赚","annual_rate":"2.7500%","describe":"热门短期产品，3天短期，无须任何手续费用，最低500起投，通过短信提示获取收益消息"}
```

5. 禁止对原索引的写入操作。

```
PUT /dest1/_settings
{
  "settings": {
    "index.blocks.write": true
  }
}
```

6. 拆分原索引并配置新索引，取消新索引的禁止写入限制。

```
POST dest1/_split/dest3
{
  "settings": {
    "index.number_of_shards": 12,
    "index.blocks.write": null
  }
}
```

以上示例使用 `_split` API从原索引dest1拆分出新索引dest3，设置新索引的分片数为12，且取消新索引禁止写限制。

> 注意
>
> - 由于原索引的主分片数为2，index.number_of_routing_shards为24，则拆分后生成的新索引的主分片数需要为原索引主分片数的整数倍且不能超过24，否则Kibana会报错。
> - split过程会进行segment merge操作，此操作会消耗集群计算资源，增加集群负载。因此在操作前需确保集群有充足的磁盘空间，并建议在业务低峰期操作。
> - 使用时需要将dest1和dest3替换为您的业务索引名。

7. 测试结果。

通过 `_cat recovery` API查看分片拆分进度，当无拆分分片相关的recovey，且集群状态健康，则分片拆分完成。

- 查看分片拆分进度

```
GET _cat/recovery?v&active_only
```

当返回结果的index列没有待拆分的索引时，说明无拆分分片相关的recovey。

- 查看集群健康状态

```
GET _cluster/health
```

当返回结果中包含 `"status" : "green"` 时，说明集群状态健康。

常见问题

Q：split完成后，为什么集群的CPU使用率、节点load_1m没有降下来？

A：split过程涉及到对文档进行重新路由，新索引会存在大量的docs.deleted文档。通过 `GET _nodes/hot_threads` 可看到索引在进行merge操作，所以计算资源负载会比较高，建议在业务低峰期操作。

9.6.3. 通过_shrink API快速减少主分片数

在Elasticsearch集群中创建索引时，如果您无法评估实际数据量，可能导致设置的主分数很大，但实际业务的数据量并不多，此时您需要减少主分片数，防止因主分片数太多导致集群资源消耗过大或影响查询写入速率等。本文为您介绍如何通过`_shrink` API快速减少主分片数。

背景信息

使用Elasticsearch需要密切关注集群分片总数及索引分片数设置。集群分片总数越多，对应分片会占用大量的文件句柄耗用大量的集群资源。同样，索引分片数设置不合理，会对查询和写入均造成潜在的影响。

在创建索引时，如果您无法评估实际数据量，可能导致设置的索引主分数很大，但实际业务数据量并不多。通过reindex减少主分片数耗时太久，elastic提供了`_shrink` API可快速减少索引主分片数。shrink操作不会在原索引上直接缩小分片，基本流程如下：

1. 创建一个和原索引配置相同的新索引，新索引主分片比原索引少，所有分片需汇集在一个节点，该节点预留的磁盘空间需要大于原索引主分片上的数据大小。
2. 从原索引到新索引创建segments硬链接。
3. 对新索引执行恢复操作，类似关闭的索引执行打开操作。

reindex与`_shrink` API的性能测试信息如下：

- 测试环境：
  - 数据节点：数量为5个，规格为8核16 GB。
  - 索引数据：数据量为182 GB。
  - 分片数：原主分片数为30，目标分片数为5，副本数为0。
- 测试结果

方式	耗时	资源占用
reindex	3小时22分钟	集群中有大量的写QPS，索引所占节点资源高。
`_shrink` API	15分钟	shrink节点计算资源较高。

前提条件

- Elasticsearch集群状态健康，且负载处于正常水位。
- 根据集群数据节点个数、集群磁盘容量等因素，合理评估索引可减少的分片数。详细信息请参见Shard评估。
- 原索引必须处于green状态。
- 原索引的文档总数不能超过2,147,483,519。
- Elasticsearch集群中没有新索引的同名索引。

操作步骤

1. 登录目标阿里云Elasticsearch实例的Kibana控制台，根据页面提示进入Kibana主页。

   登录Kibana控制台的具体操作，请参见登录Kibana控制台。

   > **说明** 本文以阿里云Elasticsearch 7.10.0版本为例，其他版本操作可能略有差别，请以实际界面为准。

2. 单击右上角的**Dev tools**。
3. 在Console中，执行如下命令，将原索引设置为禁止写状态，副本置为0，且将分片汇集到集群中的一个

节点上。

以下示例以原索引shrink5为例，使用时需要替换为您的业务索引名。

```
PUT shrink5/_settings
{
    "index.routing.allocation.require._name": "es-cn-zvp25yhyy000y****-1ab7****-0001",
    "index.blocks.write": true,
    "index.number_of_replicas": 0
}
```

参数	说明
index.routing.allocation.require._name	设置分片汇集的目标节点name。节点name可通过 `GET _cat/nodes?v` 命令获取。 **说明** 在通过 `_shrink` API减少主分片数之前，原索引的每个分片必须汇集到集群中的一个节点上。
index.blocks.write	是否禁用对索引的写操作，必须为true，即禁止写操作。 **说明** 在通过 `_shrink` API减少主分片数之前，必须设置原索引为禁止写状态。
index.number_of_replicas	索引的副本分片数。

4. 通过 `_shrink` API减少主分片数。

以下示例以将原索引shrink5的主分片数从30减少到5，并生成新索引shrink_hk5e_cn。使用时请替换索引名。

```
POST shrink5/_shrink/shrink_hk5e_cn
{
  "settings": {
    "index.blocks.write": null,
    "index.number_of_shards": 5,
    "index.number_of_replicas": 0,
    "index.routing.allocation.require._name": null
  }
}
```

参数	说明
index.blocks.write	是否禁用对索引的写操作。在通过 `_shrink` API减少主分片数后，需要将新索引的index.blocks.write置为null，即清除从原索引复制的配置。

参数	说明
index.number_of_shards	新索引的主分片数。 **注意** • 触发shrink后，shrink_node节点CPU使用率和load_1m将比较高，建议在业务低峰期操作。 • 原索引的主分片数一定要大于新索引，新索引的主分片数必须可被原索引的主分片数整除。例如，原索引的主分片数为8，则可以减少到4、2、1；原索引的主分片数为15，则可以减少到5、3、1。如果原索引的主分片数是质数，则只能减少到1。
index.number_of_replicas	新索引的副本分片数。
index.routing.allocation.require._name	设置将分片汇集的目标节点。在通过`_shrink` API减少主分片数后，需要将新索引的index.routing.allocation.require._name置为null清除从原索引复制的配置，或者删除原索引。

5. 查看结果。

   通过`_cat recovery` API查看shrink进度，当无shrink相关的recovey，且集群状态健康，则shrink完成。

   - 查看shrink进度

     ```
     GET _cat/recovery?v&active_only
     ```

     当返回结果的index列没有待shrink的索引时，说明无shrink相关的recovey。

   - 查看集群健康状态

     ```
     GET _cluster/health
     ```

     当返回结果中包含`"status" : "green"`时，说明集群状态健康。

常见问题

Q：为什么要使用硬链接，而不使用软链接？

A：通过软链接创建索引，待数据写入，将原索引删除后，目标索引数据也会被删除，而硬链接会保证索引的独立性。

9.6.4. Curator操作指南

Curator

Curator是Elasticsearch官方提供的一个索引管理工具，提供了删除、创建、关闭、段合并索引等功能。本文介绍Curator的使用方法，包括安装Curator、单命令行接口、crontab定时执行、冷热数据分离实践以及跨节点迁移索引。

安装Curator

在安装Curator前，请先完成以下准备工作：

- 创建阿里云Elasticsearch实例。

  详情请参见创建阿里云Elasticsearch实例。

- 创建阿里云ECS实例。

  本文以CentOS 7.3 64位的ECS为例，所购买的实例需要与阿里云ES实例在同一地域和可用区，以及同一专有网络VPC（Virtual Private Cloud）下。详情请参见使用向导创建实例。

连接ECS实例，执行以下命令安装Curator。

```
pip install elasticsearch-curator
```

> **说明** 建议您安装5.6.0版本的Curator，它可以支持阿里云ES 5.5.3和6.3.2版本。关于Curator版本与Elasticsearch版本的兼容性，请参见Version Compatibility。

安装成功后，执行以下命令查看Curator版本。

```
curator --version
```

正常情况下，返回结果如下。

```
curator, version 5.6.0
```

> **说明** 更多关于Curator的详细说明请参见Curator。

单命令行接口

您可以使用curator_cli命令执行单个操作，使用方式请参见Singleton Command Line Interface。

> **说明**
> - curator_cli命令只能执行一个操作。
> - 并不是所有的操作都适用于单命令行执行，例如Alias和Restore操作。

crontab定时执行

您可以通过crontab和curator命令实现定时执行一系列操作。

curator命令格式如下。

```
curator [OPTIONS] ACTION_FILE
Options:
  --config PATH  Path to configuration file. Default: ~/.curator/curator.yml
  --dry-run      Do not perform any changes.
  --version      Show the version and exit.
  --help         Show this message and exit.
```

执行curator命令时需要指定config.yml文件和action.yml文件，详情请参见config.yml官方文档和action.yml官方文档。

冷热数据分离实践

详细操作方法请参见使用Curator进行冷热数据迁移。

将索引从hot节点迁移到warm节点

1. 在*/usr/curator/*路径下创建*config.yml*文件，配置内容参考如下示例。

```
client:
  hosts:
    - http://es-cn-0pxxxxxxxxxxx234.elasticsearch.aliyuncs.com
  port: 9200
  url_prefix:
  use_ssl: False
  certificate:
  client_cert:
  client_key:
  ssl_no_validate: False
  http_auth: user:password
  timeout: 30
  master_only: False
logging:
  loglevel: INFO
  logfile:
  logformat: default
  blacklist: ['elasticsearch', 'urllib3']
```

   - `hosts`：替换为对应阿里云ES实例的内网或外网地址（此处以内网地址为例）。
   - `http_auth`：替换为对应阿里云ES实例的账号和密码。

2. 在*/usr/curator/*路径下创建*action.yml*文件，配置内容参考如下示例。

```
actions:
  1:
    action: allocation
    description: "Apply shard allocation filtering rules to the specified indices"
    options:
      key: box_type
      value: warm
      allocation_type: require
      wait_for_completion: true
      timeout_override:
      continue_if_exception: false
      disable_action: false
    filters:
    - filtertype: pattern
      kind: prefix
      value: logstash-
    - filtertype: age
      source: creation_date
      direction: older
      timestring: '%Y-%m-%dT%H:%M:%S'
      unit: minutes
      unit_count: 30
```

以上示例按照索引创建时间，将30分钟前创建在 `hot` 节点以 `logstash-` 开头的索引迁移到 `warm` 节点中。您也可以根据实际场景自定义配置*action.yml*文件。

3. 执行以下命令，验证curator命令能否正常执行。

```
curator --config /usr/curator/config.yml /usr/curator/action.yml
```

正常情况下会返回类似如下所示的信息。

```
2019-02-12 20:11:30,607 INFO      Preparing Action ID: 1, "allocation"
2019-02-12 20:11:30,612 INFO      Trying Action ID: 1, "allocation": Apply shard allocat
ion filtering rules to the specified indices
2019-02-12 20:11:30,693 INFO      Updating index setting {'index.routing.allocation.requ
ire.box_type': 'warm'}
2019-02-12 20:12:57,925 INFO      Health Check for all provided keys passed.
2019-02-12 20:12:57,925 INFO      Action ID: 1, "allocation" completed.
2019-02-12 20:12:57,925 INFO      Job completed.
```

4. 执行以下命令，使用crontab实现每隔15分钟定时执行curator命令。

```
*/15 * * * * curator --config /usr/curator/config.yml /usr/curator/action.yml
```

9.6.5. 通过RollUp实现流量汇总最佳实践

RollUp实现流量汇总

对于时序数据场景，随着时间的积累数据量会越来越大。如果一直保留详细数据，会导致存储成本线性增长，此时您可以通过Elasticsearch（简称ES）的RollUp机制节省数据存储成本。本文以汇总Logstash流量为例介绍RollUp的使用方法。

前提条件

- 确保您已拥有manage或manage_rollup权限。

  使用RollUp必须要有manage或manage_rollup权限，详情请参见Security Privileges。

- 已创建阿里云ES实例。

  详情请参见创建阿里云Elasticsearch实例，本文使用的实例版本为通用商业版7.4。

> **说明** 下文中的RollUp命令适用于ES 7.4版本，6.x版本的命令请参见RollUp Job。

背景信息

本文的需求场景如下：

- 每15分钟定时汇总整小时内instanceId的networkoutTraffic、networkinTraffic流量。
- 通过Kibana大图展示指定instanceId的入口流量和出口流量。

本文以monitordata-logstash-sls-*为前缀的索引为例，该索引以每天创建一个索引的规则切分索引。索引的Mapping格式如下。

```
"monitordata-logstash-sls-2020-04-05" : {
    "mappings" : {
      "properties" : {
        "@timestamp" : {
          "type" : "date"
        },
        "__source__" : {
          "type" : "text",
          "fields" : {
            "keyword" : {
              "type" : "keyword",
              "ignore_above" : 256
            }
          }
        },
        "disk_type" : {
          "type" : "text",
          "fields" : {
            "keyword" : {
              "type" : "keyword",
              "ignore_above" : 256
            }
          }
        },
        "host" : {
          "type" : "keyword"
        },
        "instanceId" : {
          "type" : "keyword"
        },
        "metricName" : {
          "type" : "keyword"
        },
        "monitor_type" : {
          "type" : "keyword"
        },
```

```
        "networkinTraffic" : {
          "type" : "double"
        },
        "networkoutTraffic" : {
          "type" : "double"
        },
        "node_spec" : {
          "type" : "keyword"
        },
        "node_stats_node_master" : {
          "type" : "keyword"
        },
        "resource_uid" : {
          "type" : "keyword"
        }
      }
    }
  }
}
```

> **说明** 本文中的命令均可在Kibana控制台上执行，详情请参见登录Kibana控制台。

操作流程

1. 步骤一：创建RollUp作业
2. 步骤二：启动RollUp作业并查看作业信息
3. 步骤三：查询汇总索引的数据
4. 步骤四：创建Rollup索引模式
5. 步骤五：创建Kibana流量监控大图
6. 步骤六：创建Kibana流量监控仪表板

步骤一：创建RollUp作业

RollUp作业配置包含该作业如何运行、何时索引文档及将来对汇总索引执行哪些查询的详情信息。以下示例通过 `PUT _rollup/job` 命令定义1小时内汇总的作业。

```
PUT _rollup/job/ls-monitordata-sls-1h-job1
{
    "index_pattern": "monitordata-logstash-sls-*",
    "rollup_index": "monitordata-logstash-rollup-1h-1",
    "cron": "0 */15 * * * ?",
    "page_size" :1000,
    "groups" : {
      "date_histogram": {
        "field": "@timestamp",
        "fixed_interval": "1h"
      },
      "terms": {
        "fields": ["instanceId"]
      }
    },
    "metrics": [
        {
            "field": "networkoutTraffic",
            "metrics": ["sum"]
        },
        {
            "field": "networkinTraffic",
            "metrics": ["sum"]
        }
    ]
}
```

参数	是否必选	类型	说明
index_pattern	是	string	汇总的索引或索引模式。支持通配符（*）。
rollup_index	是	string	汇总结果的索引。不支持通配符，必须是一个完整的名称。
cron	是	string	执行汇总作业任务的时间间隔。与汇总数据的时间间隔无关。
page_size	是	integer	汇总索引每次迭代中处理的存储桶的结果数。值越大，执行越快，但是处理过程中需要更多的内存。
groups	是	object	为汇总作业定义分组字段和聚合。
└ date_histogram	是	object	将date字段汇总到基于时间的存储桶中。
└ field	是	string	需要汇总的date字段。
└ fixed_interval	是	time units	数据汇总的时间间隔。例如设置为1h，表示按照1小时汇总field指定的时间字段。该参数定义了数据能够聚合的最小时间间隔。

参数	是否必选	类型	说明
terms	否	object	无。
└ fields	是	string	定义terms字段集。此数组字段可以是keyword也可以是numerics类型，无顺序要求。
metrics	否	object	无。
└ field	是	string	定义需要采集的指标的字段。例如以上示例是分别对networkoutTraffic、networkinTraffic进行采集。
└ metrics	是	array	定义聚合算子。设置为sum，表示对networkinTraffic进行sum运算。仅支持min、max、sum、average、value count。

> **说明** └表示子参数。

更多参数说明请参见Create rollup jobs API。配置参数时，请注意：

- index_pattern 中指定通配符时，请确保不会匹配到 rollup_index 指定的汇总索引名，否则报错。
- 由于汇总索引的Mapping是object类型，请确保集群中不存在与汇总索引相匹配的索引模板，否则报错。
- 字段分组聚合仅支持Date Histogram aggregation、Histogram aggregation、Terms aggregation，详细限制说明请参见Rollup aggregation limitations。

步骤二：启动RollUp作业并查看作业信息

1. 启动RollUp作业。

```
POST _rollup/job/ls-monitordata-sls-1h-job1/_start
```

2. 查看RollUp作业的配置、统计和状态信息。

```
GET _rollup/job/ls-monitordata-sls-1h-job1/
```

更多详细说明请参见Get rollup jobs API。

执行成功后，返回如下结果。

```
{
      ........
       "status" : {
         "job_state" : "indexing",
         "current_position" : {
           "@timestamp.date_histogram" : 1586775600000,
           "instanceId.terms" : "ls-cn-ddddez****"
         },
         "upgraded_doc_id" : true
       },
       "stats" : {
         "pages_processed" : 3,
         "documents_processed" : 11472500,
         "rollups_indexed" : 3000,
         "trigger_count" : 1,
         "index_time_in_ms" : 766,
         "index_total" : 3,
         "index_failures" : 0,
         "search_time_in_ms" : 68559,
         "search_total" : 3,
         "search_failures" : 0
       }
}
```

步骤三：查询汇总索引的数据

在Rollup内部，由于汇总文档使用的文档结构和原始数据不同，Rollup查询端口会将标准查询DSL重写为与汇总文档匹配的格式，然后获取响应并将其重写回给原始查询的客户端所期望的格式。

1. 使用match_all获取汇总索引的所有数据。

```
GET monitordata-logstash-rollup-1h-1/_search
{
  "query": {
    "match_all": {}
  }
}
```

   - 查询仅能指定一个汇总索引，即不支持模糊匹配。对实时索引数据查询没有限制要求，查询可指定多个索引。
   - 查询仅支持Term、Terms、Range query、MatchAll query、Any compound query（Boolean、Boosting、ConstantScore等），更多限制请参见Rollup search limitations。

2. 使用 `_rollup_search` 聚合出口流量总数据。

```
GET /monitordata-logstash-rollup-1h-1/_rollup_search
{
    "size": 0,
    "aggregations": {
        "sum_temperature": {
            "sum": {
                "field": "networkoutTraffic"
            }
        }
    }
}
```

`_rollup_search` 支持常规的Search API特性子集：

- query：指定DSL查询参数，但受一些限制，详情请参见Rollup search limitations和Rollup aggregation limitations。
- aggregations：指定聚合参数。

`_rollup_search` 不可用功能包括：

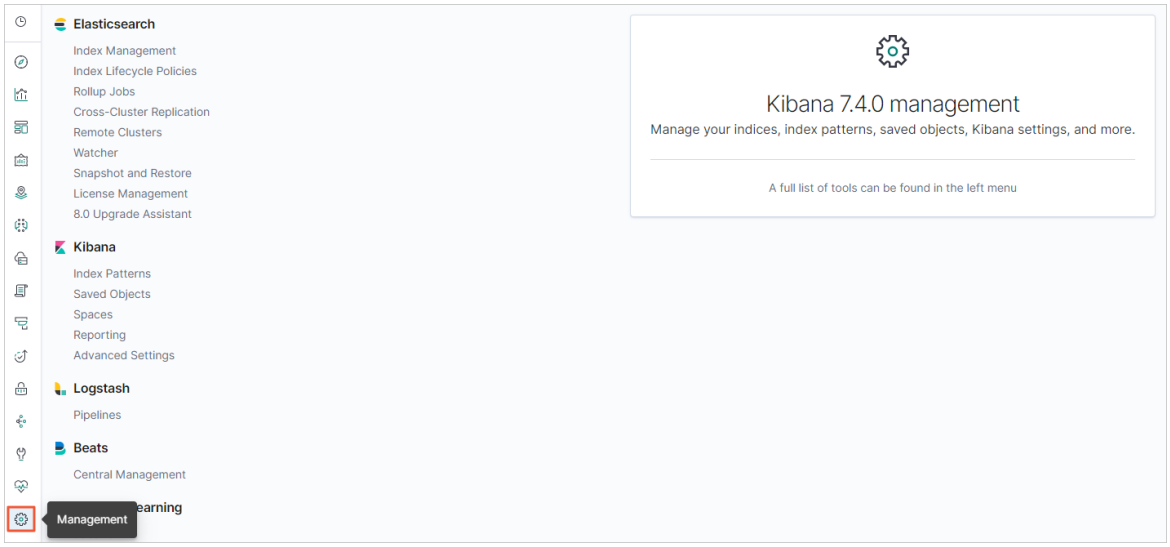
- size：由于汇总适用于聚合数据，无法返回查询结果，因此将size设置为0或者完全省略。
- 不支持highlighter、suggestors、post_filter、profile、explain等参数。

步骤四：创建Rollup索引模式

1. 登录Kibana控制台。

   登录控制台的具体步骤请参见登录Kibana控制台。

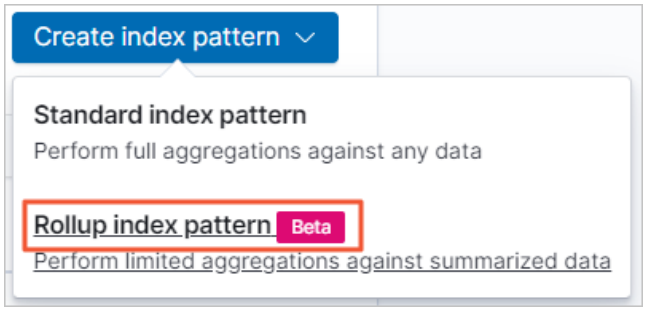
2. 在左侧导航栏，单击Management图标。



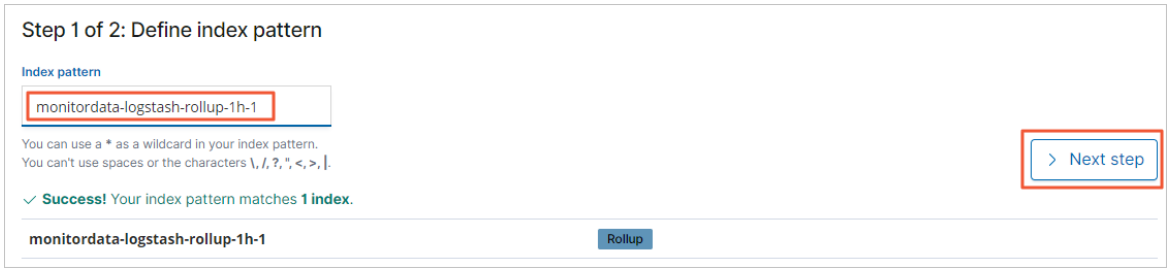
3. 在Kibana区域，单击Index Patterns。
4. （可选）关闭About index patterns页面。

   > **说明** 如果您不是首次创建索引模式，可忽略此步骤。

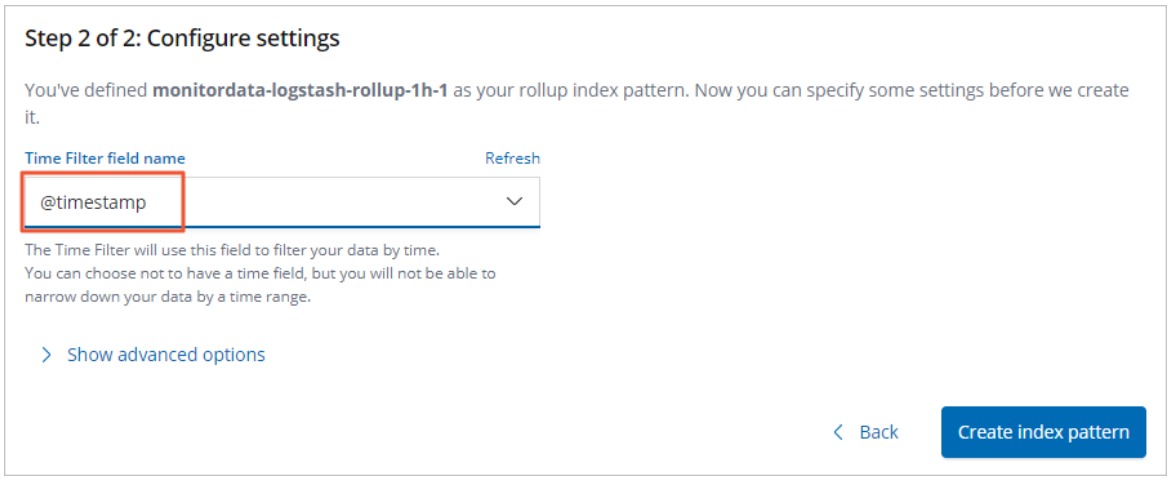
5. 单击Create index pattern > Rollup index pattern。



6. 输入索引模式名称（monitordata-logstash-rollup-1h-1），然后单击**Next step**。



7. 从**Time Filter field name**列表中，选择**@timestamp**。



8. 单击**Create index pattern**。

步骤五：创建Kibana流量监控大图

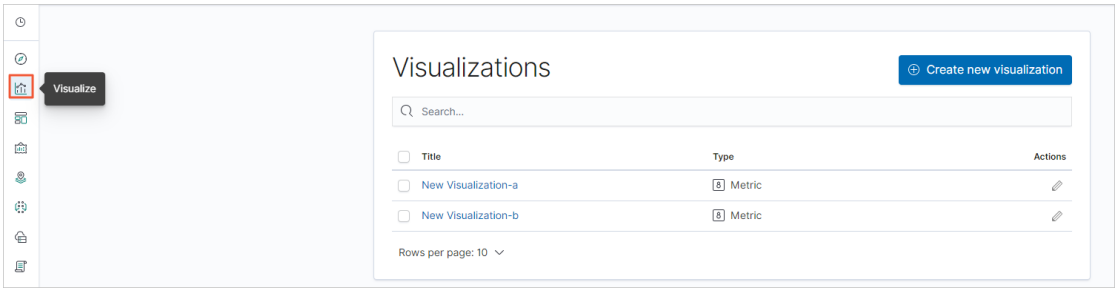
在Kibana上分别配置汇总索引的入口流量及出口流量监控大图，操作步骤如下：

1. 登录Kibana控制台。

   登录控制台的具体步骤请参见登录Kibana控制台。

2. 创建**Line**曲线图。

i. 在左侧导航栏，单击Visualize图标。



ii. 单击Create new visualization。

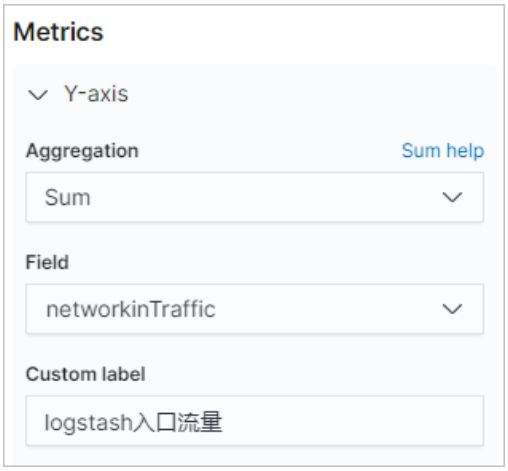
iii. 在New Visualization对话框中，单击Line。

iv. 在索引模式列表中，单击选择已创建的Rollup索引模式。

3. 设置Metrics和Buckets。

i. 在Metrics区域，单击 > 。

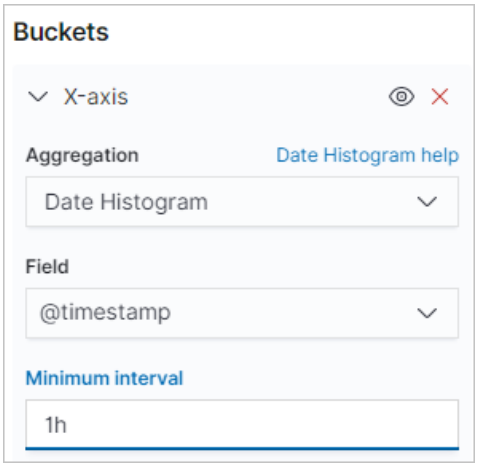
ii. 设置Y-axis参数。



参数	说明
Aggregation	选择Sum。
Field	选择networkinTraffic或networkoutTraffic。
Custom label	自定义Y轴标签。

iii. 在Buckets区域，单击Add > X-axis。

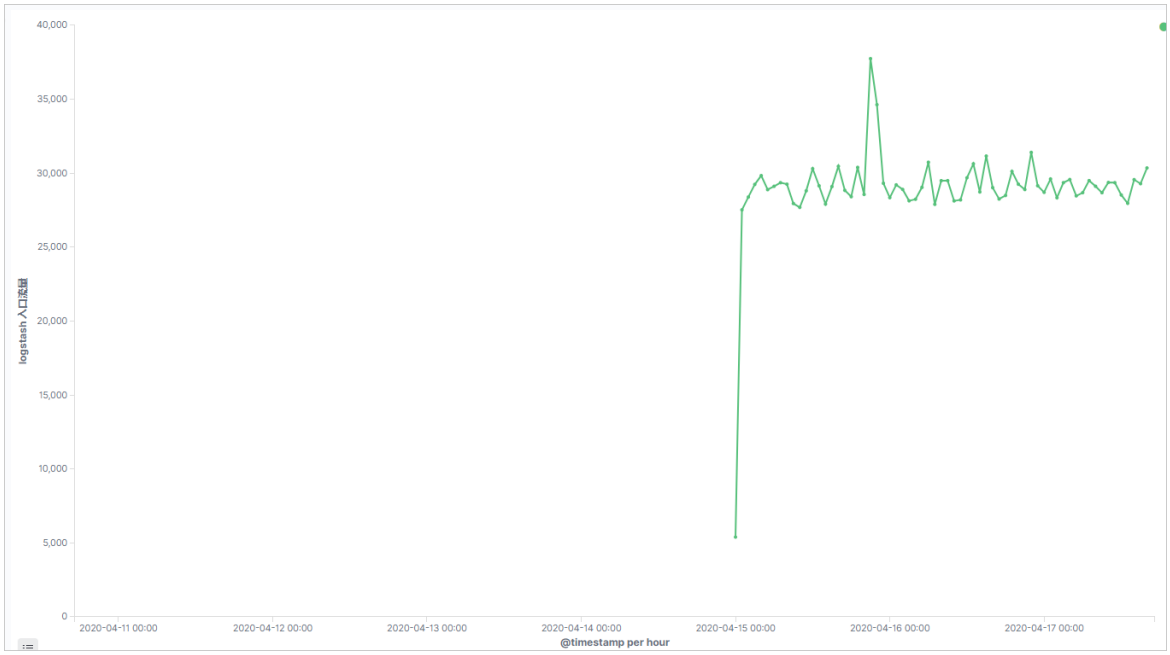
iv. 设置X-axis参数。



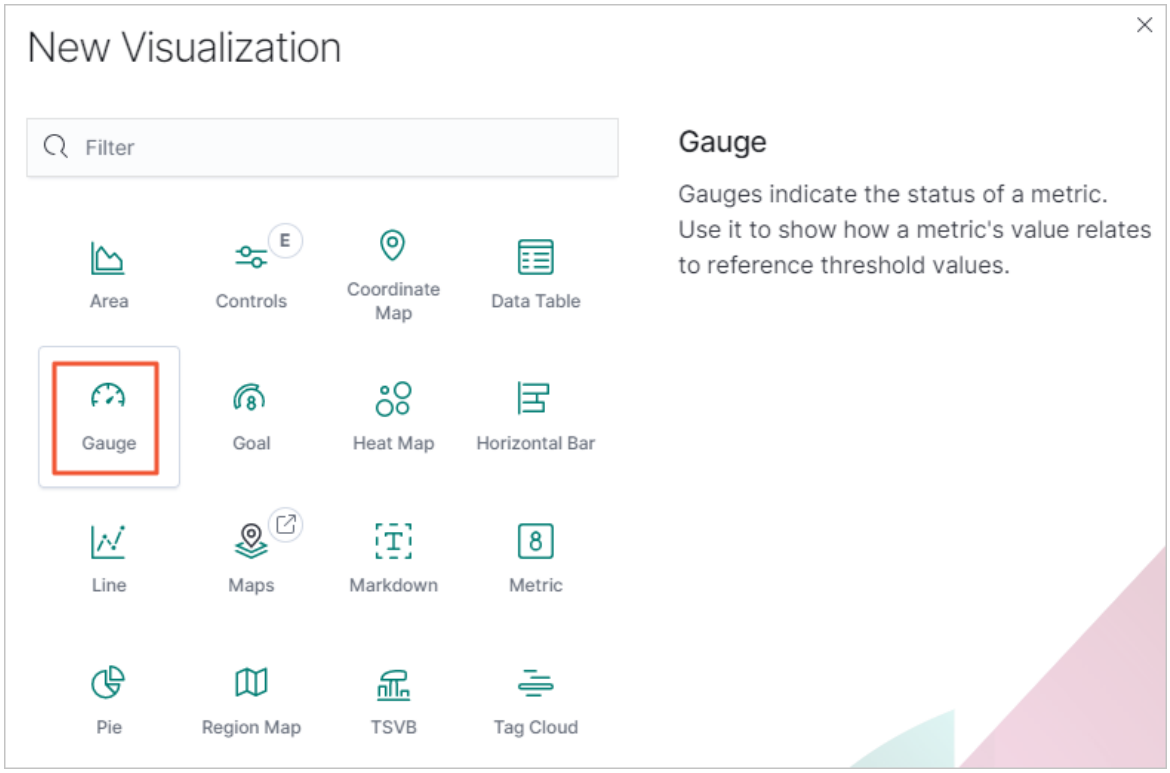
参数	说明
Aggregation	设置为步骤一：创建RollUp作业时，`group`中定义的`date_histogram`。
Field	选择@timestamp。
Minimum interval	默认为RollUp作业中定义的聚合时间粒度。必须是汇总配置间隔的整数倍，例如2h、3h等。

v. 单击图标。

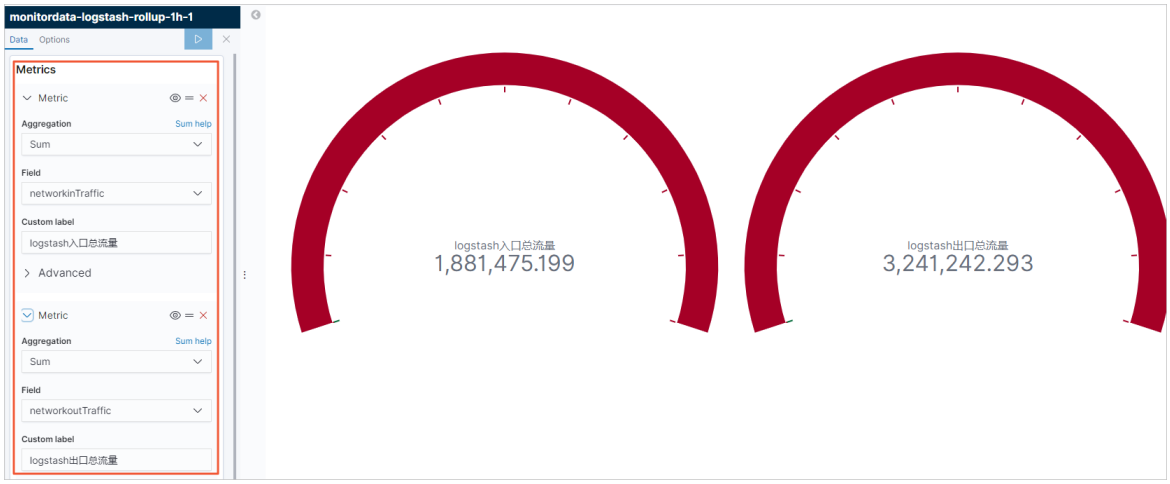
4. 在顶部菜单栏，单击Save。
配置成功后，可看到如下结果。



5. 使用同样的方式创建Gauge图。

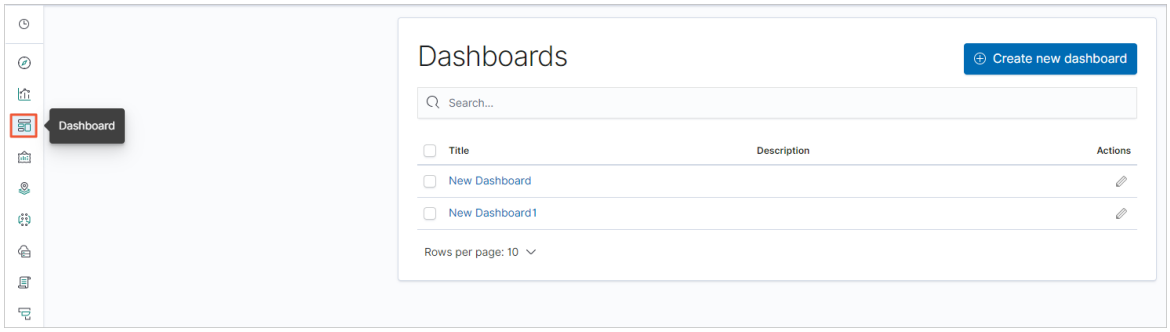


6. 参考下图配置Gauge图。



步骤六：创建Kibana流量监控仪表板

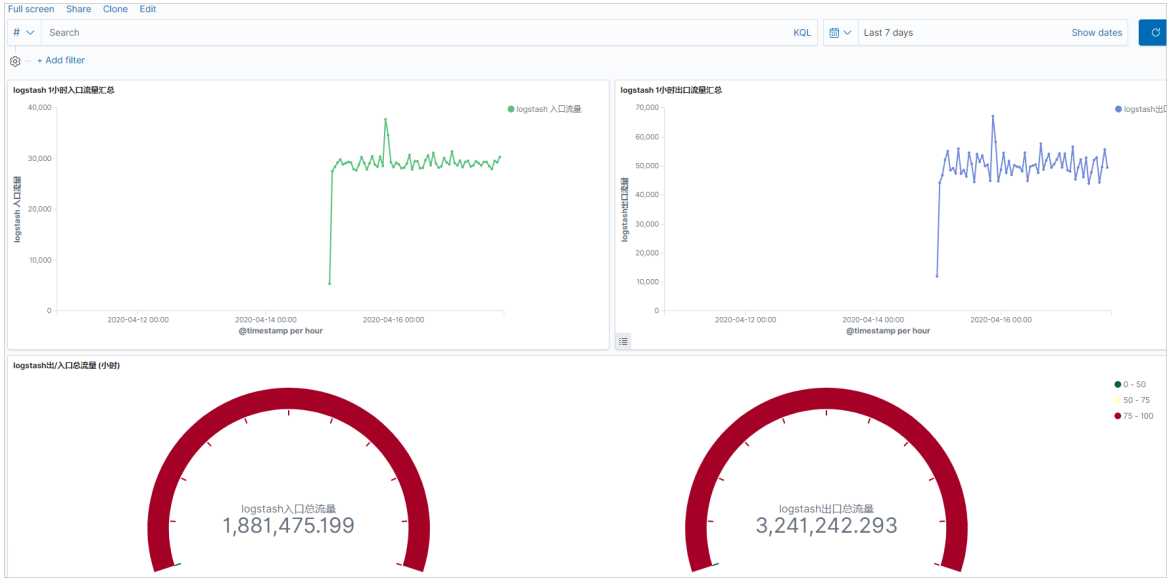
1. 在Kibana控制台中，单击左侧导航栏的**Dashboard**图标。



2. 单击**Create new dashboard**。

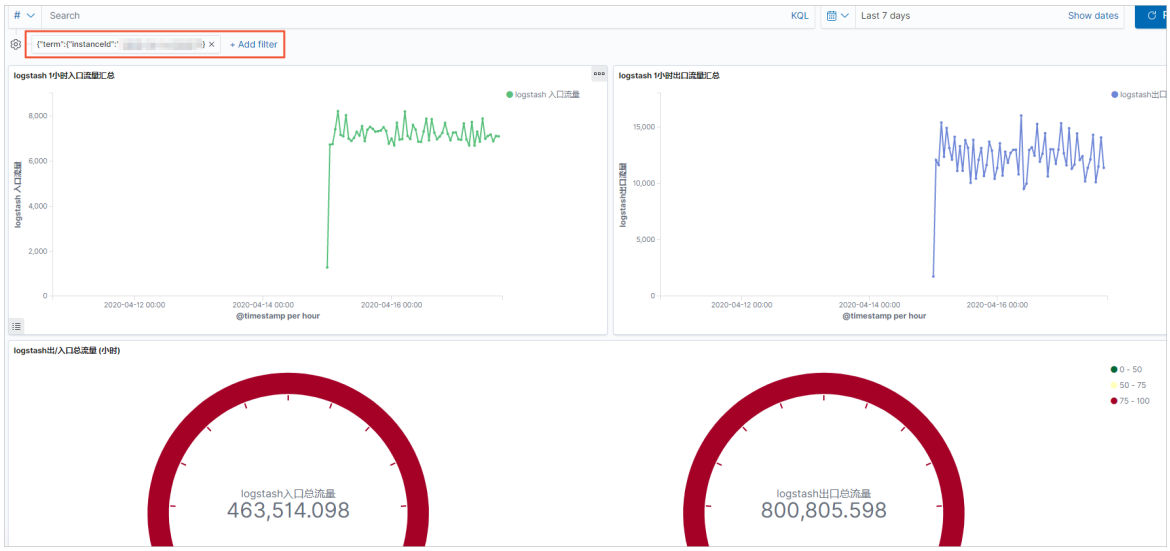
3. 在顶部菜单栏，单击Add。
4. 在Add panels页面，单击选择Visualize中配置的可视化大图。
5. 关闭Add panels页面，在顶部菜单栏，单击Save。
6. 修改仪表板名称，单击Confirm Save。

   仪表板保存成功后，可查看仪表板展示结果。



7. 单击+ Add filter，选择一个过滤项并配置过滤条件，单击Save。

   本文使用term过滤项，指定查询某个instance的流量，最终结果如下。



9.6.6. 使用DataV大屏展示阿里云Elasticsearch数据

使用DataV展示es数据

通过在DataV中添加阿里云Elasticsearch数据源，您可以使用DataV访问阿里云Elasticsearch服务，完成数据的查询与展示。本文介绍如何使用DataV大屏展示阿里云Elasticsearch数据。

前提条件

您已完成以下操作：

- 创建阿里云Elasticsearch实例。

  具体操作，请参见创建阿里云Elasticsearch实例。

- 开通DataV服务，且版本为企业版或以上版本。

  具体操作，请参见开通DataV服务。

- 准备待展示的索引数据。

  具体操作，请参见快速入门。

  本文使用如下命令创建索引和添加数据：

  - 创建索引

```
PUT /my_index
{
    "settings" : {
      "index" : {
        "number_of_shards" : "5",
        "number_of_replicas" : "1"
      }
    },
    "mappings" : {
        "my_type" : {
            "properties" : {
              "post_date": {
                   "type": "date"
               },
              "tags": {
                   "type": "keyword"
               },
                "title" : {
                    "type" : "text"
                }
            }
        }
    }
}
```

- 添加数据

```
PUT /my_index/my_type/1?pretty
{
  "title": "One",
  "tags": ["ruby"],
  "post_date":"2009-11-15T13:00:00"
}
PUT /my_index/my_type/2?pretty
{
  "title": "Two",
  "tags": ["ruby"],
  "post_date":"2009-11-15T14:00:00"
}
PUT /my_index/my_type/3?pretty
{
  "title": "Three",
  "tags": ["ruby"],
  "post_date":"2009-11-15T15:00:00"
}
```

在DataV中添加Elasticsearch数据源

1. 登录阿里云Elasticsearch控制台。
2. 在左侧导航栏，单击**Elasticsearch实例**。
3. 进入目标实例。
   i. 在顶部菜单栏处，选择资源组和地域。
   ii. 在左侧导航栏，单击**Elasticsearch实例**，然后在**Elasticsearch实例**中单击目标实例ID。
4. 在左侧导航栏，单击**可视化控制**。
5. 在**DataV**区域中，单击**进入控制台**。
6. 在DataV控制台中，添加Elasticsearch数据源。

> **注意** DataV企业版及以上版本才支持添加Elasticsearch数据源。

   i. 进入**我的数据**页面，单击**添加数据**。
   ii. 从**添加数据**对话框的**类型**列表中，选择**Elastic Search**。

iii. 单击**使用前请授权DataV访问**。

iv. 在**云资源访问授权**页面，单击**同意授权**。

v. 返回DataV控制台，单击**我的数据**。

vi. 单击**添加数据**。

vii. 从**添加数据**对话框的**类型**列表中，选择**Elastic Search**，并填写阿里云Elasticsearch实例信息。

参数	说明
自定义数据源名称	数据源的显示名称，可自定义。
Region	实例的地域。
实例ID	实例ID，可在实例的基本信息页面获取。详细信息，请参见查看实例的基本信息。 授权DataV访问阿里云Elasticsearch服务后，单击**获取实例列表**，可在右侧下拉列表选择某一实例。
密码	实例的访问密码。

viii. 单击**确定**。
确定后系统会自动进行测试连接，测试连接成功后即可完成数据源的添加。

使用Elasticsearch数据源

在使用阿里云Elasticsearch数据源之前，需要先在DataV中添加Elasticsearch数据源。

1. 进入DataV控制台。
2. 在**我的可视化**页面，移动鼠标移至您的大屏项目上，单击**编辑**。

> **说明** 如果还没有大屏项目，请先创建一个大屏项目并添加组件。具体操作，请参见DataV官方文档的快速入门章节。

3. 在大屏编辑页面的画布中，单击选择某一组件。

   本文以双十一轮播列表柱状图组件为例。

4. 单击**数据**页签，再单击**配置数据源**。

5. 在**设置数据源**页面，选择**数据源类型**为**Elastic Search**，**已有数据源**为您已经添加的阿里云Elasticsearch数据源。

6. 在**index**输入框中填写查询索引。

   查询索引通常为一个字符串，本文使用my_index索引。

7. 在**Query**输入框中填写查询体。

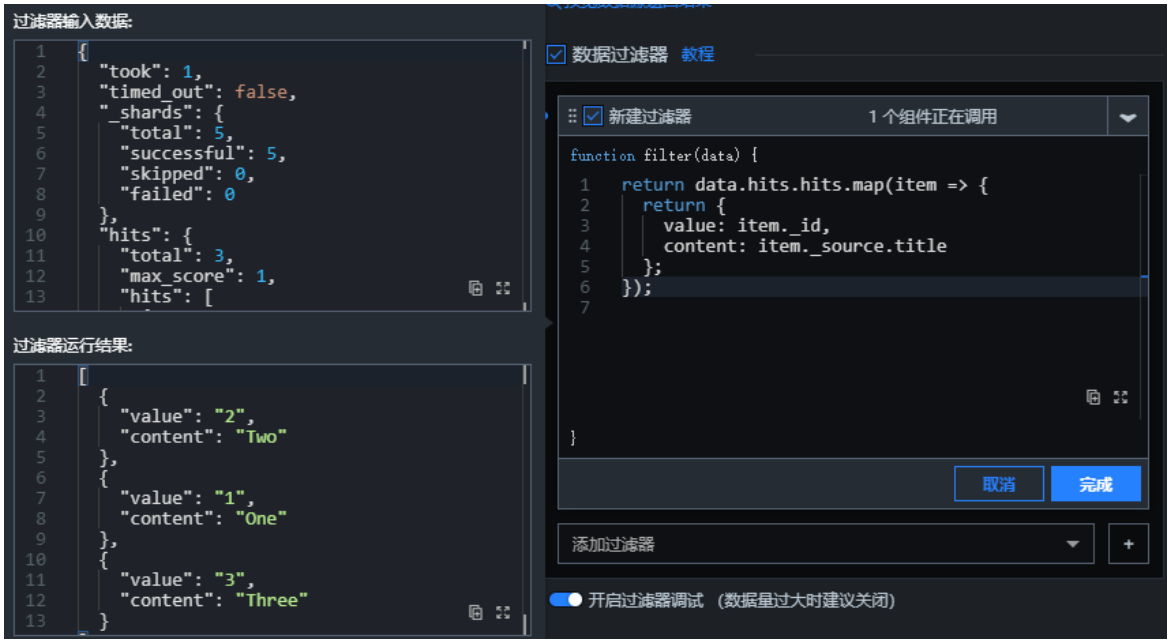
   查询体通常为一个JSON对象，默认是{} 。

8. 启用并配置数据过滤器。

本文使用的过滤器脚本如下，具体配置方法请参见组件过滤器使用介绍。

```
return data.hits.hits.map(item => {
  return {
    value: item._id,
    content: item._source.title
  };
});
```

9. 在数据过滤器脚本编辑区域，单击空白处，查看过滤器运行结果。



后续步骤

预览并发布大屏，展示对应Elasticsearch实例的索引数据。具体操作，请参见发布PC端可视化应用。

9.6.7. 通过Cerebro访问阿里云ES

cerebro访问es

除了Kibana、curl命令、客户端等方式，您还可以通过Elasticsearch-Head、Cerebro等第三方插件或工具访问阿里云Elasticsearch（简称ES）实例。由于Elasticsearch-Head插件在5.x版本之后已不再维护，因此建议您使用Cerebro访问阿里云ES实例。本文介绍具体的操作方法。

前提条件

- 创建阿里云ES实例。

  具体操作步骤请参见创建阿里云Elasticsearch实例。

- 创建ECS实例，要求该实例与阿里云ES实例在同一专有网络VPC（Virtual Private Cloud）下。

  具体操作步骤请参见使用向导创建实例。该ECS实例用来安装Cerebro。

  > **说明** 如果您的ECS实例与阿里云ES实例不在同一VPC中，或者您需要在本机安装cerebro，此时可通过公网访问阿里云ES实例。通过公网访问阿里云ES时需要注意：
  >
  > - 公网访问的安全性较低。
  > - 当网络延迟时可能会造成服务抖动。
  > - 需要开启阿里云ES的公网地址并配置公网访问白名单，详情请参见配置实例公网或私网访问白名单。

- 在ECS实例中安装JDK，要求版本为1.8及以上。

背景信息

- Cerebro是第三方支持的工具。

- 在公网环境下，Cerebro只能通过阿里云ES实例的公网地址和端口访问集群。

操作步骤

1. 连接ECS实例。

   具体操作步骤请参见连接ECS实例。

2. 下载Cerebro安装包并解压。
   - 下载

     ```
     wget https://github.com/lmenezes/cerebro/releases/download/v0.9.0/cerebro-0.9.0.tgz
     ```

   - 解压

     ```
     tar -zxvf cerebro-0.9.0.tgz
     ```

3. 修改Cerebro配置文件，关联待访问的阿里云ES实例。

   i. 打开application.conf文件。

   ```
   vim cerebro-0.9.0/conf/application.conf
   ```

ii. 按照以下说明配置 `hosts` 。

```
# A list of known hosts
hosts = [
  #{
  #  host = "http://localhost:9200"
  #  name = "Localhost cluster"
  #  headers-whitelist = [ "x-proxy-user", "x-proxy-roles", "X-Forwarded-For" ]
  #}
  # Example of host with authentication
  {
    host = "http://es-cn-r                    .elasticsearch.aliyuncs.com:9200"
    name = "es-cn-r                    "
    auth = {
      username = "elastic"
      password = "                    "
    }
  }
```

> **说明** 您也可以关联多个实例，多个实例之间用英文逗号（,）分隔。

参数	说明
host	阿里云ES实例的访问地址，格式为 `http://<阿里云ES实例的内网地址>:9200` 。实例的内网地址可在基本信息页面获取，详情请参见查看实例的基本信息。
name	阿里云ES实例的ID，可在基本信息页面获取，详情请参见查看实例的基本信息。
username	访问阿里云ES实例的用户名，默认为elastic。 **注意** 实际业务中不建议使用elastic用户，这样会降低系统安全性。建议使用自建用户，并给予自建用户分配相应的角色和权限，详情请参见通过Elasticsearch X-Pack角色管理实现用户权限管控。
password	对应用户的密码。elastic用户的密码在创建实例时指定，如果忘记可进行重置，重置密码的注意事项和操作步骤请参见重置实例访问密码。

iii. 保存文件后，启动Cerebro服务。

```
cd cerebro-0.9.0
bin/cerebro
```

启动成功后，返回如下结果。

```
[root@VM01 cerebro-0.9.0]# bin/cerebro
[info] play.api.Play - Application started (Prod) (no global state)
[info] p.c.s.AkkaHttpServer - Listening for HTTP on /0.0.0.0:9000
```

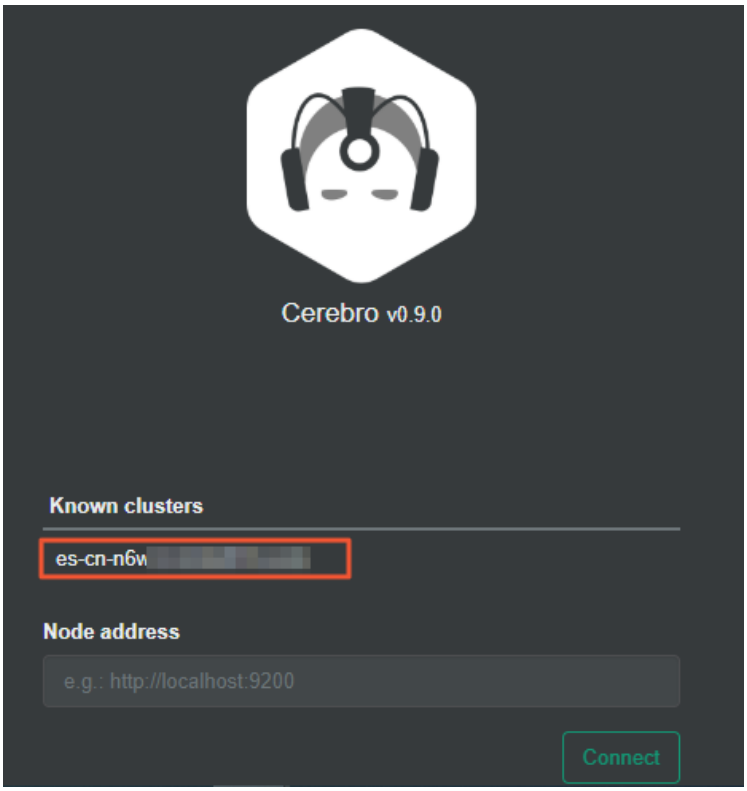
4. 通过Cerebro访问阿里云ES。

i. 配置ECS实例的安全组，在**入方向**中，添加待访问机器的IP地址并开放9000端口。

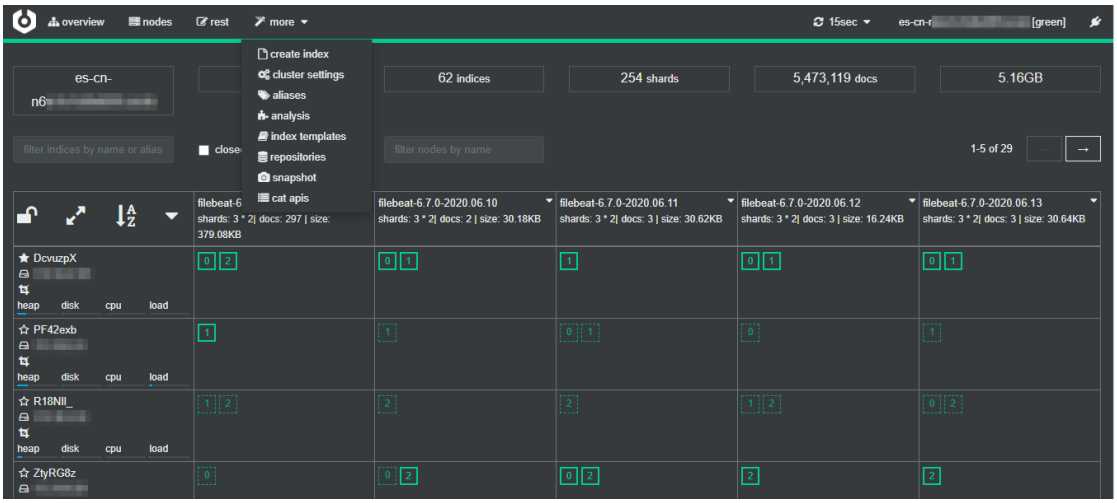
具体操作步骤请参见添加安全组规则。

ii. 在浏览器中输入http://<ECS的外网IP地址>:9000。

iii. 在Cerebro登录页面，单击您要访问的阿里云ES实例的ID。



iv. 在Cerebro控制台中，查看集群状态以及索引、分片和文档数量等，并根据业务进行相关操作。



> **说明** Cerebro的使用方法请参见Getting Started with Cerebro。

9.7. 集群报警通知

9.7.1. 配置钉钉机器人接收X-Pack Watcher报警

配置X-Pack Watcher

通过为阿里云Elasticsearch添加X-Pack Watcher，可以实现当满足某些条件时执行某些操作。例如当logs索引中出现error日志时，触发系统自动发送报警邮件或钉钉消息。可以简单地理解为X-Pack Watcher是一个基于Elasticsearch实现的监控报警服务。本文介绍如何配置钉钉机器人接收X-Pack Watcher报警。

背景信息

X-Pack Watcher功能主要由Trigger、Input、Condition和Actions组成：

- Trigger

  Watcher定时触发器，即多久触发一次Watcher，相当于多久执行一次input。支持多种调度触发器，详细信息请参见Schedule Trigger。

- Input

  Input将数据加载到执行上下文，用于后续的Watcher执行阶段，如果input没有指定，将会加载一个空上下文，详细信息请参见Inputs。Watcher支持以下input类型：

  - simple：将输入静态数据加载到执行上下文。例如手动输入一段简单的数据进行报警。
  - search：将搜索结果加载到执行上下文。例如全文搜索关键词，对搜索结果进行统计实现报警。
  - http：将HTTP请求结果加载到执行上下文。例如通过Elasticsearch请求接口获取集群健康状态、节点状态等实现报警。
  - chain：将一系列的输入数据加载到执行上下文，这些数据一般是来自多个源。

- Condition

  执行Actions的条件。即满足条件将会触发下一步操作，如果不指定条件，默认为always，详细信息请参见Conditions。Watcher支持以下condition类型：

  - always：条件总为true，始终执行Watcher Actions。

- nerver：条件总为false，从不执行Watcher Actions。
- compare：对Watcher有效负载中的值进行简单比较，以确定是否执行Watcher Actions。
- array_compare：将Watcher有效负载中的值数组与给定值进行比较，以确定是否执行Watcher Actions。
- script：使用脚本确定是否执行Watcher Actions。

- Actions

  报警接收对象，常见的报警接收对象包括邮件、Webhook 、index和logging等，详细信息请参见Actions。

  > **说明** 通过邮件接收报警存在端口限制，阿里云Elasticsearch不支持，建议通过Webhook进行邮件转发。

前提条件

您已完成以下操作：

- 创建单可用区的阿里云Elasticsearch实例。

  具体操作，请参见创建阿里云Elasticsearch实例。

  > **说明** 旧网络架构下X-Pack Watcher功能仅支持单可用区Elasticsearch实例，不支持多可用区实例，新网络架构下没有限制。

- 开启Elasticsearch实例的X-Pack Watcher功能（默认关闭）。

  具体操作，请参见配置YML参数。

- 在用户VPC下创建ECS服务，并部署相关应用。

  具体操作，请参见使用向导创建实例。

  > **说明**
  > - 使用PrivateLink打通网络时，ECS服务器会作为后端服务器，主要接收通过负载均衡实例所转发的请求，没有可用区的限制，但是在创建时需要与负载均衡实例部署在同一地域且同一VPC下。
  > - 阿里云Elasticsearch的X-Pack Watcher功能不支持直接与公网通讯，需要基于实例的内网地址通讯（专有网络VPC环境），因此您需要对VPC网络下的ECS配置SNAT或弹性公网IP，作为代理去转发请求。

注意事项

自2020年10月起，阿里云Elasticsearch对不同地域进行了网络架构的调整，对创建的实例有以下影响：

- 2020年10月之前创建的实例均在旧网络架构下，即Elasticsearch实例处于用户VPC下，如果需要访问公网，可以直接使用SNAT功能或自建Nginx代理。
- 2020年10月及之后创建的实例均在新网络架构下，即Elasticsearch实例处于Elasticsearch服务VPC下，X-Pack Watcher功能受到网络限制，为解决此问题，阿里云Elasticsearch提供了实例私网连接方案，详细信息请参见配置实例私网连接。如果您还需要将报警信息推送至公网环境，在通过实例私网连接打通Elasticsearch服务VPC和用户VPC的基础上，还需对负载均衡后端服务配置Nginx代理或开启SNAT功能实现公网信息推送。

> 注意 实例私网连接方案是新网络架构下X-Pack Watcher、reindex、LDAP和AD（Active Directory）身份认证等功能受限的唯一解决方案，为保证功能使用不受影响，请严格按照文档配置。

操作流程

1. 步骤一：创建并配置钉钉机器人
2. （可选）步骤二：配置Elasticsearch实例私网连接
3. 步骤三：配置ECS安全组和Nginx代理
4. 步骤四：配置Watcher报警
5. 步骤五：查看报警结果

步骤一：创建并配置钉钉机器人

1. 创建一个钉钉报警接收群。

   具体操作，请参见钉钉入门教程。

2. 在群的右上角找到群机器人，然后添加一个自定义通过Webhook接入的机器人并进行安全设置，同时获取Webhook地址。

   详细信息，请参见获取自定义机器人Webhook和安全设置。

   安全设置

> 注意 安全设置中的关键词必须包含在您设置的报警信息中。

获取Webhook地址

> 注意 请保管好此Webhook地址，以备后用。同时不要将其公布在外部网站上，泄露后有安全风险。

（可选）步骤二：配置Elasticsearch实例私网连接

旧网络架构下创建的实例，无需配置私网连接；新网络架构下创建的实例，需要配置私网连接。

1. 登录阿里云Elasticsearch控制台。
2. 配置Elasticsearch实例的私网连接，获取终端节点域名作为访问外部服务的网络连接。

   具体操作，请参见配置实例私网连接。

步骤三：配置ECS安全组和Nginx代理

1. 配置ECS安全组。
   i. 登录阿里云ECS控制台。
   ii. 在左侧导航栏，单击**实例**。
   iii. 在实例列表页面，选择目标实例右侧**操作**列下的**更多 > 网络和安全组 > 安全组配置**。
   iv. 在**安全组列表**页签下，单击目标安全组右侧**操作**列下的**配置规则**。
   v. 在**入方向**页签，单击**手动添加**。

vi. 填写相关参数。

参数	说明
授权策略	选择**允许**。
优先级	保持默认。
协议类型	选择**自定义TCP**。
端口范围	填写您常用的端口（配置Nginx时需要用到，本文以8080为例）。
授权对象	添加您购买的阿里云Elasticsearch实例所有节点的IP地址。 **说明** 参见查看节点的基本信息，获取Elasticsearch实例中所有节点的IP地址。
描述	输入对规则的描述。

vii. 单击**保存**。

2. 配置Nginx代理。

i. 在ECS上安装Nginx。

具体安装方法请参见Nginx安装配置。

ii. 配置nginx.conf文件。

使用以下配置替换nginx.conf文件中 `server` 部分的配置。

```
#下面是server虚拟主机的配置
server
 {
   listen 8080;#监听端口
   server_name localhost;#域名
   index index.html index.htm index.php;
   root /usr/local/webserver/nginx/html;#站点目录
     location ~ .*\.(php|php5)?$
   {
     #fastcgi_pass unix:/tmp/php-cgi.sock;
     fastcgi_pass 127.0.0.1:9000;
     fastcgi_index index.php;
     include fastcgi.conf;
   }
   location ~ .*\.(gif|jpg|jpeg|png|bmp|swf|ico)$
   {
     expires 30d;
     # access_log off;
   }
   location / {
     proxy_pass https://oapi.dingtalk.com/robot/send?access_token=[redacted];
   }
   location ~ .*\.(js|css)?$
   {
     expires 15d;
     # access_log off;
   }
   access_log off;
 }
```

```
server
  {
    listen 8080;#监听端口
    server_name localhost;#域名
    index index.html index.htm index.php;
    root /usr/local/webserver/nginx/html;#站点目录
      location ~ .*\.(php|php5)?$
    {
      #fastcgi_pass unix:/tmp/php-cgi.sock;
      fastcgi_pass 127.0.0.1:9000;
      fastcgi_index index.php;
      include fastcgi.conf;
    }
    location ~ .*\.(gif|jpg|jpeg|png|bmp|swf|ico)$
    {
      expires 30d;
      # access_log off;
    }
    location / {
      proxy_pass <钉钉机器人Webhook地址>;
    }
    location ~ .*\.(js|css)?$
    {
      expires 15d;
      # access_log off;
    }
    access_log off;
  }
```

<钉钉机器人Webhook地址>：请替换为接收报警消息的钉钉机器人的Webhook地址，获取方式请参见步骤一：创建并配置钉钉机器人。

iii. 加载修改后的配置文件并重启Nginx。

```
/usr/local/webserver/nginx/sbin/nginx -s reload            # 重新载入配置文件
/usr/local/webserver/nginx/sbin/nginx -s reopen            # 重启Nginx
```

步骤四：配置Watcher报警

1. 登录目标阿里云Elasticsearch实例的Kibana控制台，根据页面提示进入Kibana主页。

   登录Kibana控制台的具体操作，请参见登录Kibana控制台。

   > **说明** 本文以阿里云Elasticsearch 6.7.0版本为例，其他版本操作可能略有差别，请以实际界面为准。

2. 在左侧导航栏，单击**Dev Tools**。
3. 在**Console**中，执行如下命令创建一个报警文档。

   以下示例以创建log_error_watch文档为例，每隔10s查询logs索引中是否出现error日志，如果出现0次以上则触发报警。

```
PUT _xpack/watcher/watch/log_error_watch
{
  "trigger": {
    "schedule": {
      "interval": "10s"
    }
  },
  "input": {
    "search": {
      "request": {
        "indices": ["logs"],
        "body": {
          "query": {
            "match": {
              "message": "error"
            }
          }
        }
      }
    }
  },
  "condition": {
    "compare": {
      "ctx.payload.hits.total": {
        "gt": 0
      }
    }
  },
  "actions" : {
  "test_issue" : {
    "webhook" : {
      "method" : "POST",
      "url" : "http://<yourAddress>:8080",
      "body" : "{\"msgtype\": \"text\", \"text\": { \"content\": \"error 日志出现了，请尽快处理\"}}"
    }
  }
}
}
```

关键参数说明

参数	网络类型	配置对象	说明
<yourAddress>	新网络	终端节点域名地址	新网络架构下，即通过配置实例私网连接进行网络打通，通过终端节点域名实现请求转发。 **注意** 此处需要配置为终端节点域名，而非服务域名。获取终端节点域名的具体操作，请参见（可选）步骤五：查看终端节点域名。

参数	网络类型	配置对象	说明
	旧网络	Nginx代理IP地址	通过同VPC下Nginx代理经公网进行请求转发。
		url配置为钉钉机器人的Webhook地址	开启SNAT网关，使处于用户VPC下的Elasticsearch实例可以访问外网，详细信息请参见使用公网NAT网关SNAT功能访问互联网。

注意

- 如果在执行以上命令时，出现 `No handler found for uri [/_xpack/watcher/watch/log_error_watch_2] and method [PUT]` 异常，表示您购买的阿里云Elasticsearch实例未开启X-Pack Watcher功能，请开启后再执行以上命令。具体步骤，请参见配置YML参数。
- 以上代码中的body参数需要根据钉钉机器人的安全设置配置，详细信息请参见步骤一：创建并配置钉钉机器人。例如本文选择**安全设置**方式为**自定义关键词**，且添加了一个自定义关键词：error，那么body中的content字段必须包含error，钉钉机器人才会推送报警信息。

步骤五：查看报警结果

正常情况下，当集群中的数据达到步骤四：配置Watcher报警中配置的报警条件时，您可以在钉钉群中收到类似 `error 日志出现了，请尽快处理` 的报警信息。

说明 如果您不再需要执行报警任务，可执行以下命令删除该报警任务。

```
DELETE _xpack/watcher/watch/log_error_watch
```

9.7.2. 配置企业微信机器人接收X-Pack Watcher报警

通过为阿里云Elasticsearch添加X-Pack Watcher，可以实现当满足某些条件时执行某些操作。例如当logs索引中出现error日志时，触发系统自动发送报警邮件或机器人消息。可以简单地理解为X-Pack Watcher是一个基于Elasticsearch实现的监控报警服务。本文介绍如何配置企业微信机器人接收X-Pack Watcher报警。

背景信息

X-Pack Watcher功能主要由Trigger、Input、Condition和Actions组成：

- Trigger

  Watcher定时触发器，即多久触发一次Watcher，相当于多久执行一次input。支持多种调度触发器，详细信息请参见Schedule Trigger。

- Input

  Input将数据加载到执行上下文，用于后续的Watcher执行阶段，如果input没有指定，将会加载一个空上下文，详细信息请参见Inputs。Watcher支持以下input类型：

  - simple：将输入静态数据加载到执行上下文。例如手动输入一段简单的数据进行报警。
  - search：将搜索结果加载到执行上下文。例如全文搜索关键词，对搜索结果进行统计实现报警。

- http：将HTTP请求结果加载到执行上下文。例如通过Elasticsearch请求接口获取集群健康状态、节点状态等实现报警。
- chain：将一系列的输入数据加载到执行上下文，这些数据一般是来自多个源。

- Condition

  执行Actions的条件。即满足条件将会触发下一步操作，如果不指定条件，默认为always，详细信息请参见Conditions。Watcher支持以下condition类型：

  - always：条件总为true，始终执行Watcher Actions。
  - nerver：条件总为false，从不执行Watcher Actions。
  - compare：对Watcher有效负载中的值进行简单比较，以确定是否执行Watcher Actions。
  - array_compare：将Watcher有效负载中的值数组与给定值进行比较，以确定是否执行Watcher Actions。
  - script：使用脚本确定是否执行Watcher Actions。

- Actions

  报警接收对象，常见的报警接收对象包括邮件、Webhook 、index和logging等，详细信息请参见Actions。

  > **说明** 通过邮件接收报警存在端口限制，阿里云Elasticsearch不支持，建议通过Webhook进行邮件转发。

前提条件

您已完成以下操作：

- 创建单可用区的阿里云Elasticsearch实例。

  具体操作，请参见创建阿里云Elasticsearch实例。

  > **说明** 旧网络架构下X-Pack Watcher功能仅支持单可用区Elasticsearch实例，不支持多可用区实例，新网络架构下没有限制。

- 开启Elasticsearch实例的X-Pack Watcher功能（默认关闭）。

  具体操作，请参见配置YML参数。

- 在用户VPC下创建ECS服务，并部署相关应用。

  具体操作，请参见使用向导创建实例。

  > **说明**
  > - 使用PrivateLink打通网络时，ECS服务器会作为后端服务器，主要接收通过负载均衡实例所转发的请求，没有可用区的限制，但是在创建时需要与负载均衡实例部署在同一地域且同一VPC下。
  > - 阿里云Elasticsearch的X-Pack Watcher功能不支持直接与公网通讯，需要基于实例的内网地址通讯（专有网络VPC环境），因此您需要对VPC网络下的ECS配置SNAT或弹性公网IP，作为代理去转发请求。

- 创建企业微信机器人，并获取其Webhook地址。

注意事项

自2020年10月起，阿里云Elasticsearch对不同地域进行了网络架构的调整，对创建的实例有以下影响：

- 2020年10月之前创建的实例均在旧网络架构下，即Elasticsearch实例处于用户VPC下，如果需要访问公网，可以直接使用SNAT功能或自建Nginx代理。
- 2020年10月及之后创建的实例均在新网络架构下，即Elasticsearch实例处于Elasticsearch服务VPC下，X-Pack Watcher功能受到网络限制，为解决此问题，阿里云Elasticsearch提供了实例私网连接方案，详细信息请参见配置实例私网连接。如果您还需要将报警信息推送至公网环境，在通过实例私网连接打通Elasticsearch服务VPC和用户VPC的基础上，还需对负载均衡后端服务配置Nginx代理或开启SNAT功能实现公网信息推送。

> 注意 实例私网连接方案是新网络架构下X-Pack Watcher、reindex、LDAP和AD（Active Directory）身份认证等功能受限的唯一解决方案，为保证功能使用不受影响，请严格按照文档配置。

操作流程

1. （可选）步骤一：配置Elasticsearch实例私网连接
2. 步骤二：配置ECS安全组和Nginx代理
3. 步骤三：配置Watcher报警
4. 步骤四：查看报警结果

（可选）步骤一：配置Elasticsearch实例私网连接

旧网络架构下创建的实例，无需配置私网连接；新网络架构下创建的实例，需要配置私网连接。

1. 登录阿里云Elasticsearch控制台。
2. 配置Elasticsearch实例的私网连接，获取终端节点域名作为访问外部服务的网络连接。

   具体操作，请参见配置实例私网连接。

步骤二：配置ECS安全组和Nginx代理

1. 配置ECS安全组。
   i. 登录阿里云ECS控制台。
   ii. 在左侧导航栏，单击**实例**。
   iii. 在实例列表页面，选择目标实例右侧**操作**列下的**更多 > 网络和安全组 > 安全组配置**。
   iv. 在**安全组列表**页签下，单击目标安全组右侧**操作**列下的**配置规则**。
   v. 在**入方向**页签，单击**手动添加**。

vi. 填写相关参数。

参数	说明
授权策略	选择**允许**。
优先级	保持默认。
协议类型	选择**自定义TCP**。
端口范围	填写您常用的端口（配置Nginx时需要用到，本文以8080为例）。
授权对象	添加您购买的阿里云Elasticsearch实例所有节点的IP地址。 **说明** 参见查看节点的基本信息，获取Elasticsearch实例中所有节点的IP地址。
描述	输入对规则的描述。

vii. 单击**保存**。

2. 配置Nginx代理。

i. 在ECS上安装Nginx。

具体安装方法请参见Nginx安装配置。

ii. 配置nginx.conf文件。

使用以下配置替换nginx.conf文件中 `server` 部分的配置。

```
server {
        listen 8080;
        server_name _;
        root /usr/share/nginx/html;
        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;
          location / {
            proxy_pass <企业微信机器人Webhook地址>;
          }
        error_page 404 /404.html;
            location = /40x.html {
        }
        error_page 500 502 503 504 /50x.html;
            location = /50x.html {
        }
    }
```

<企业微信机器人Webhook地址>：请替换为接收报警消息的企业微信机器人的Webhook地址。

iii. 加载修改后的配置文件并重启Nginx。

```
/usr/local/webserver/nginx/sbin/nginx -s reload            # 重新载入配置文件
/usr/local/webserver/nginx/sbin/nginx -s reopen            # 重启Nginx
```

步骤三：配置Watcher报警

1. 登录目标阿里云Elasticsearch实例的Kibana控制台，根据页面提示进入Kibana主页。

   登录Kibana控制台的具体操作，请参见登录Kibana控制台。

   > **说明** 本文以阿里云Elasticsearch 6.7.0版本为例，其他版本操作可能略有差别，请以实际界面为准。

2. 在左侧导航栏，单击**Dev Tools**。
3. 在**Console**中，执行如下命令创建一个报警文档。

   以下示例以创建developer_count_watch文档为例，每隔10s查询zl-testgaes索引中的developer字段，如果字段值为Nintendo且出现次数大于158974则触发报警。

```
PUT _xpack/watcher/watch/developer_count_watch
{
  "trigger": {
    "schedule": {
      "interval": "10s"
    }
  },
  "input": {
    "search": {
      "request": {
        "indices": ["zl-testgaes"],
        "body": {
          "query": {
    "bool": {
      "must": [
        {"match":
         {
           "developer" : "Nintendo"
        }
        },
        {
        "range": {
          "year_of_release": {
            "gte": "2011-09-20T16:00:00.000Z",
            "lte": "2011-12-31T16:00:00.000Z"
                  }
              }
        }
      ]
    }
  }
        }
      }
    }
  },
  "condition": {
    "compare": {
      "ctx.payload.hits.total": {
        "gt": 158974
      }
    }
  },
  "actions" : {
  "test_issue" : {
    "webhook" : {
      "method" : "POST",
      "url" : "http://<yourAddress>:8080",
      "body" : "{\"msgtype\": \"text\", \"text\": { \"content\": \"developer is Nintendo
, More than 158974\"}}"
    }
  }
}
}
```

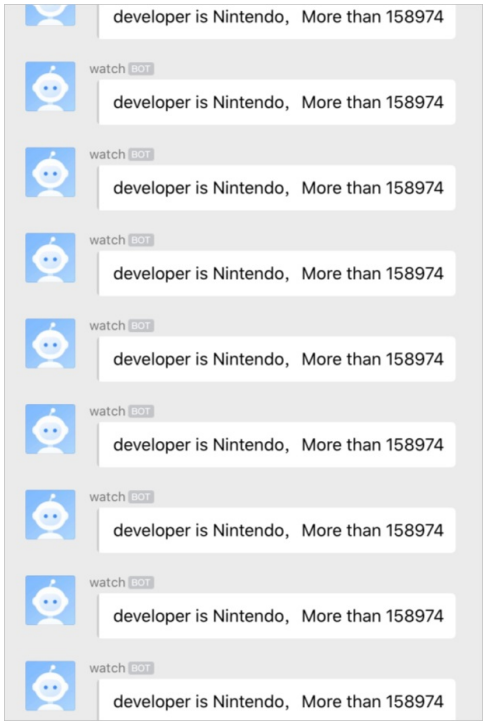
关键参数说明

参数	网络类型	配置对象	说明
<yourAddress>	新网络	终端节点域名地址	新网络架构下，即通过配置实例私网连接进行网络打通，通过终端节点域名实现请求转发。 **注意** 此处需要配置为终端节点域名，而非服务域名。获取终端节点域名的具体操作，请参见（可选）步骤五：查看终端节点域名。
	旧网络	Nginx代理IP地址	通过同VPC下Nginx代理经公网进行请求转发。
		url配置为企业微信机器人Webhook地址	开启SNAT网关，使处于用户VPC下的Elasticsearch实例可以访问外网，详细信息请参见使用公网NAT网关SNAT功能访问互联网。

注意 如果在执行以上命令时，出现 `No handler found for uri [/_xpack/watcher/watch/log_error_watch_2] and method [PUT]` 异常，表示您购买的阿里云Elasticsearch实例未开启X-Pack Watcher功能，请开启后再执行以上命令。具体步骤，请参见配置YML参数。

步骤四：查看报警结果

正常情况下，当集群中的数据达到步骤三：配置Watcher报警中配置的报警条件时，您可以企业微信中收到如下报警信息。



> **说明** 如果您不再需要执行报警任务，可执行以下命令删除该报警任务。

```
DELETE _xpack/watcher/watch/developer_count_watch
```

9.7.3. 通过X-Pack Watcher实现CCR异常报警通知

Elasticsearch X-Pack Watcher可跟踪网络，具备对基础设施、索引数据和集群健康等指标的监控和报警能力。您可以在Kibana控制台上获取跨集群复制CCR（Cross Cluster Replication）功能相关的监控，并通过X-Pack Watcher监控CCR异常实现报警。本文介绍如何将CCR获取的读数据请求耗时及CCR Checkpoint作为预警条件，实现CCR异常报警通知。

背景信息

X-Pack Watcher功能主要由Trigger、Input、Condition和Actions组成：

- Trigger

  Watcher定时触发器，即多久触发一次Watcher，相当于多久执行一次input。支持多种调度触发器，详细信息请参见Schedule Trigger。

- Input

  Input将数据加载到执行上下文，用于后续的Watcher执行阶段，如果input没有指定，将会加载一个空上下文，详细信息请参见Inputs。Watcher支持以下input类型：

  - simple：将输入静态数据加载到执行上下文。例如手动输入一段简单的数据进行报警。
  - search：将搜索结果加载到执行上下文。例如全文搜索关键词，对搜索结果进行统计实现报警。
  - http：将HTTP请求结果加载到执行上下文。例如通过Elasticsearch请求接口获取集群健康状态、节点状态等实现报警。
  - chain：将一系列的输入数据加载到执行上下文，这些数据一般是来自多个源。

- Condition

  执行Actions的条件。即满足条件将会触发下一步操作，如果不指定条件，默认为always，详细信息请参见Conditions。Watcher支持以下condition类型：

  - always：条件总为true，始终执行Watcher Actions。
  - nerver：条件总为false，从不执行Watcher Actions。
  - compare：对Watcher有效负载中的值进行简单比较，以确定是否执行Watcher Actions。
  - array_compare：将Watcher有效负载中的值数组与给定值进行比较，以确定是否执行Watcher Actions。
  - script：使用脚本确定是否执行Watcher Actions。

- Actions

  报警接收对象，常见的报警接收对象包括邮件、Webhook、index和logging等，详细信息请参见Actions。

  > **说明** 通过邮件接收报警存在端口限制，阿里云Elasticsearch不支持，建议通过Webhook进行邮件转发。

前提条件

您已完成以下操作：

- 创建单可用区的阿里云Elasticsearch实例。

  具体操作，请参见创建阿里云Elasticsearch实例。

  > **说明** 旧网络架构下X-Pack Watcher功能仅支持单可用区Elasticsearch实例，不支持多可用区实例，新网络架构下没有限制。

- 开启Elasticsearch实例的X-Pack Watcher功能（默认关闭）。

  具体操作，请参见配置YML参数。

- 在用户VPC下创建ECS服务，并部署相关应用。

  具体操作，请参见使用向导创建实例。

  > **说明**
  > - 使用PrivateLink打通网络时，ECS服务器会作为后端服务器，主要接收通过负载均衡实例所转发的请求，没有可用区的限制，但是在创建时需要与负载均衡实例部署在同一地域且同一VPC下。
  > - 阿里云Elasticsearch的X-Pack Watcher功能不支持直接与公网通讯，需要基于实例的内网地址通讯（专有网络VPC环境），因此您需要对VPC网络下的ECS配置SNAT或弹性公网IP，作为代理去转发请求。

注意事项

自2020年10月起，阿里云Elasticsearch对不同地域进行了网络架构的调整，对创建的实例有以下影响：

- 2020年10月之前创建的实例均在旧网络架构下，即Elasticsearch实例处于用户VPC下，如果需要访问公网，可以直接使用SNAT功能或自建Nginx代理。
- 2020年10月及之后创建的实例均在新网络架构下，即Elasticsearch实例处于Elasticsearch服务VPC下，X-Pack Watcher功能受到网络限制，为解决此问题，阿里云Elasticsearch提供了实例私网连接方案，详细信息请参见配置实例私网连接。如果您还需要将报警信息推送至公网环境，在通过实例私网连接打通Elasticsearch服务VPC和用户VPC的基础上，还需对负载均衡后端服务配置Nginx代理或开启SNAT功能实现公网信息推送。

  > **注意** 实例私网连接方案是新网络架构下X-Pack Watcher、reindex、LDAP和AD（Active Directory）身份认证等功能受限的唯一解决方案，为保证功能使用不受影响，请严格按照文档配置。

操作流程

1. 步骤一：创建并配置钉钉机器人
2. （可选）步骤二：配置Elasticsearch实例私网连接
3. 步骤三：配置ECS安全组和Nginx代理
4. 步骤四：配置Watcher报警
5. 步骤五：查看报警结果

步骤一：创建并配置钉钉机器人

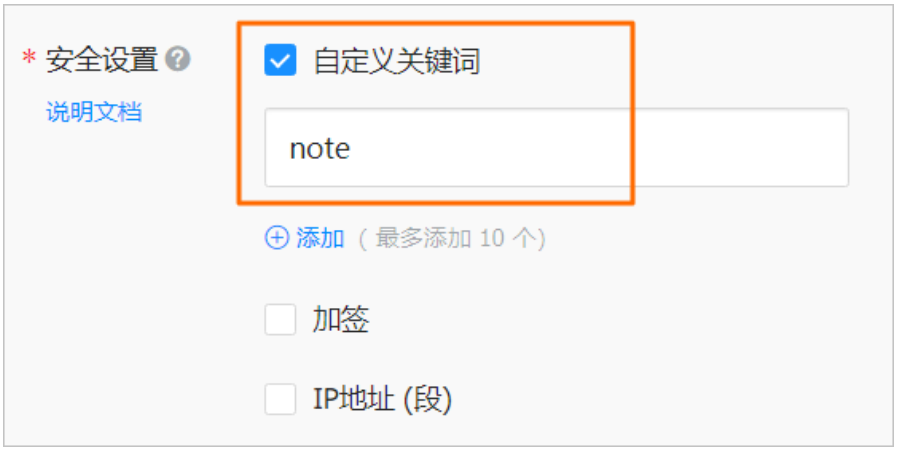
1. 创建一个钉钉报警接收群。

   具体操作，请参见钉钉入门教程。

2. 在群的右上角找到群机器人，然后添加一个自定义通过Webhook接入的机器人并进行安全设置，同时获取Webhook地址。

   详细信息，请参见获取自定义机器人Webhook和安全设置。

   安全设置



> 注意 安全设置中的关键词必须包含在您设置的报警信息中。

   获取Webhook地址

> 注意 请保管好此Webhook地址，以备后用。同时不要将其公布在外部网站上，泄露后有安全风险。

（可选）步骤二：配置Elasticsearch实例私网连接

旧网络架构下创建的实例，无需配置私网连接；新网络架构下创建的实例，需要配置私网连接。

1. 登录阿里云Elasticsearch控制台。
2. 配置Elasticsearch实例的私网连接，获取终端节点域名作为访问外部服务的网络连接。

   具体操作，请参见配置实例私网连接。

步骤三：配置ECS安全组和Nginx代理

1. 配置ECS安全组。

i. 登录阿里云ECS控制台。

ii. 在左侧导航栏，单击**实例**。

iii. 在实例列表页面，选择目标实例右侧**操作**列下的**更多 > 网络和安全组 > 安全组配置**。

iv. 在**安全组列表**页签下，单击目标安全组右侧**操作**列下的**配置规则**。

v. 在**入方向**页签，单击**手动添加**。

vi. 填写相关参数。

参数	说明
授权策略	选择**允许**。
优先级	保持默认。
协议类型	选择**自定义TCP**。
端口范围	填写您常用的端口（配置Nginx时需要用到，本文以8080为例）。
授权对象	添加您购买的阿里云Elasticsearch实例所有节点的IP地址。 **说明** 参见查看节点的基本信息，获取Elasticsearch实例中所有节点的IP地址。
描述	输入对规则的描述。

vii. 单击**保存**。

2. 配置Nginx代理。

i. 在ECS上安装Nginx。

具体安装方法请参见Nginx安装配置。

ii. 配置nginx.conf文件。

使用以下配置替换nginx.conf文件中 `server` 部分的配置。

```
#下面是server虚拟主机的配置
server
 {
   listen 8080;#监听端口
   server_name localhost;#域名
   index index.html index.htm index.php;
   root /usr/local/webserver/nginx/html;#站点目录
     location ~ .*\.(php|php5)?$
   {
     #fastcgi_pass unix:/tmp/php-cgi.sock;
     fastcgi_pass 127.0.0.1:9000;
     fastcgi_index index.php;
     include fastcgi.conf;
   }
   location ~ .*\.(gif|jpg|jpeg|png|bmp|swf|ico)$
   {
     expires 30d;
     # access_log off;
   }
   location / {
     proxy_pass https://oapi.dingtalk.com/robot/send?access_token=[redacted];
   }
   location ~ .*\.(js|css)?$
   {
     expires 15d;
     # access_log off;
   }
   access_log off;
 }
}
```

```
server
  {
    listen 8080;#监听端口
    server_name localhost;#域名
    index index.html index.htm index.php;
    root /usr/local/webserver/nginx/html;#站点目录
      location ~ .*\.(php|php5)?$
    {
      #fastcgi_pass unix:/tmp/php-cgi.sock;
      fastcgi_pass 127.0.0.1:9000;
      fastcgi_index index.php;
      include fastcgi.conf;
    }
    location ~ .*\.(gif|jpg|jpeg|png|bmp|swf|ico)$
    {
      expires 30d;
      # access_log off;
    }
    location / {
      proxy_pass <钉钉机器人Webhook地址>;
    }
    location ~ .*\.(js|css)?$
    {
      expires 15d;
      # access_log off;
    }
    access_log off;
  }
```

<钉钉机器人Webhook地址>：请替换为接收报警消息的钉钉机器人的Webhook地址，获取方式请参见步骤一：创建并配置钉钉机器人。

iii. 加载修改后的配置文件并重启Nginx。

```
/usr/local/webserver/nginx/sbin/nginx -s reload            # 重新载入配置文件
/usr/local/webserver/nginx/sbin/nginx -s reopen            # 重启Nginx
```

步骤四：配置Watcher报警

1. 登录目标阿里云Elasticsearch实例的Kibana控制台，根据页面提示进入Kibana主页。

   登录Kibana控制台的具体操作，请参见登录Kibana控制台。

   > **说明** 本文以阿里云Elasticsearch 6.7.0版本为例，其他版本操作可能略有差别，请以实际界面为准。

2. 在左侧导航栏，单击**Dev Tools**。
3. 在**Console**中，执行如下命令创建一个报警文档。

```
PUT _watcher/watch/ccr_watcher
{
  "trigger": {
    "schedule": {
      "interval": "10s"
    }
  },
  "input": {
    "search": {
      "request": {
        "indices": [
          ".monitoring-es*"
        ],
        "body": {
          "size": 0,
          "sort": [
            {
              "timestamp": {
                "order": "desc"
              }
            }
          ],
          "query": {
            "bool": {
              "must": [
                {
                  "range": {
                    "timestamp": {
                      "gte": "now-10m"
                    }
                  }
                },
                {
                  "term": {
                    "type": {
                      "value": "ccr_stats"
                    }
                  }
                },
                {
                  "bool": {
                    "should": [
                      {
                        "range": {
```

```
                          "ccr_stats.time_since_last_read_millis": {
                            "gte": 600000
                          }
                        }
                      },
                      {
                        "script": {
                          "script": "long gap = doc['ccr_stats.leader_global_checkpoint'
].value - doc['ccr_stats.follower_global_checkpoint'].value;\n            return gap>100
0;"
                        }
                      }
                    ]
                  }
                }
              ]
            }
          },
          "aggs": {
            "NAME": {
              "terms": {
                "field": "ccr_stats.follower_index",
                "size": 1000
              }
            }
          }
        }
      }
    }
  },
  "condition": {
    "compare": {
      "ctx.payload.hits.total": {
        "gt": 0
      }
    }
  },
  "transform": {
    "script": """
    StringBuilder message = new StringBuilder();
for (def bucket : ctx.payload.aggregations.NAME.buckets) {
  message.append(bucket.key).append('  ')
}
    return [ 'delay_indices' : message.toString().trim()  ]
"""
  },
  "actions" : {
     "add_index": {
      "index": {
        "index": "ccr_delay_indices",
        "doc_type": "doc"
      }
    },
     "my_webhook": {
     "webhook" : {
```

```
        "method" : "POST",
        "url" : "http://<yourAddress>:8080",
        "body" : "{\"msgtype\": \"text\", \"text\": { \"content\": \"Please note:  {{ctx
.payload}}\"}}"
      }
    }
  }
}
```

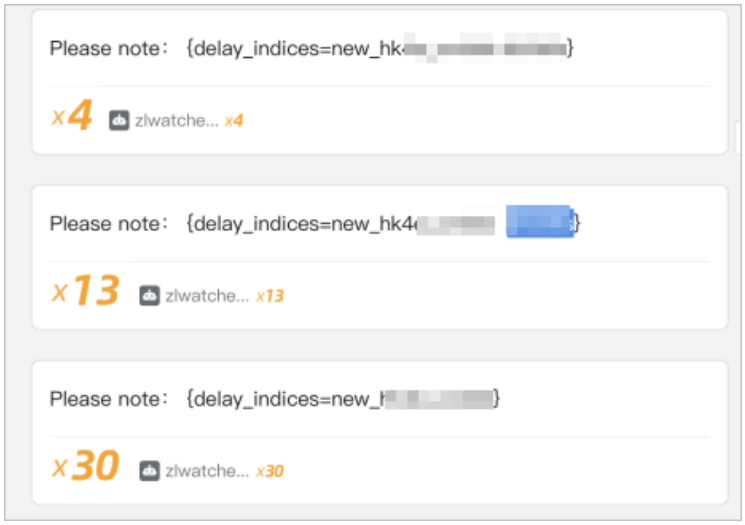
部分关键参数说明如下。

参数	说明
trigger	检测周期，请根据实际业务进行配置。以上示例设置为每10s检测一次。
input.search.request.indices	查询检测的目标索引。.monitoring-es*索引保存集群支持的所有监控指标，其中也包括CCR指标。
input.search.request.body	查询体。以上示例从系统监控索引查询近10分钟的CCR状态信息，当查询满足以下条件中的其中一个，CCR将进入下一步： ○ ccr_stats.time_since_last_read_millis > 600000ms（10min）：读请求发送到Leader节点的延时时间大于10min。请根据您CCR的实际业务使用进行配置。 ○ ccr_stats.leader_global_checkpoint-ccr_stats.follower_global_checkpoint > 1000：Follower Checkpoint落后于Leader Checkpoint的程度大于1000。请根据您CCR的实际业务使用进行配置。
condition	报警条件。以上示例表示满足input.search.request.body中的条件，且返回的聚合文档数大于0即可报警。
transform	预处理。以上示例表示循环获取buckets key值，并通过空格进行分割，过滤出延迟索引名。
actions	满足条件时，执行的具体操作。以上示例设置两个action： ○ add_index：将上面返回的结果写入索引中，在调试Watcher配置环节，定义index进行调试。 ○ my_webhook：通过Webhook方式发送报警信息。
<yourAddress>	接收报警信息的服务器的访问地址。该访问地址与阿里云Elasticsearch的网络架构相关，具体说明如下： ○ 新网络架构：需要配置为终端节点域名地址。即通过配置实例私网连接进行网络打通，通过终端节点域名实现请求转发。获取终端节点域名的具体操作，请参见（可选）步骤五：查看终端节点域名。 ○ 旧网络架构：需要配置为Nginx代理IP地址或钉钉机器人Webhook地址。 ■ Nginx代理IP地址：通过同VPC下Nginx代理经公网进行请求转发。 ■ url配置为钉钉机器人Webhook地址：开启SNAT网关，使处于用户VPC下的Elasticsearch实例可以访问外网，详细信息请参见使用公网NAT网关SNAT功能访问互联网。

> 说明
>
> - 如果在执行以上命令时，出现 `No handler found for uri [/_xpack/watcher/watch/log_error_watch_2] and method [PUT]` 异常，表示您购买的阿里云Elasticsearch实例未开启X-Pack Watcher功能，请开启后再执行以上命令。具体步骤，请参见配置YML参数。
> - 以上代码中的body参数需要根据钉钉机器人的安全设置配置，详细信息请参见步骤一：创建并配置钉钉机器人。例如本文选择**安全设置**方式为**自定义关键词**，且添加了一个自定义关键词：note，那么body中的content字段必须包含note，钉钉机器人才会推送报警信息。

步骤五：查看报警结果

正常情况下，当CCR状态达到步骤四：配置Watcher报警中配置的报警条件时，您可以钉钉群中收到如下报警信息。



> 说明 如果您不再需要执行报警任务，可执行以下命令删除该报警任务。

```
DELETE _xpack/watcher/watch/ccr_watcher
```