

Elasticsearch Service

快速入门

产品文档



腾讯云

【版权声明】

©2013-2019 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

快速入门

- 集群规格和容量配置评估

- 创建集群

- 访问集群

 - ES 集群用户登录认证

 - 通过客户端访问集群

 - 通过 API 访问集群

 - 通过 Kibana 访问集群

快速入门

集群规格和容量配置评估

最近更新时间：2021-10-26 16:42:13

腾讯云 Elasticsearch Service (ES) 是分布式多节点形式的集群，每个节点均是有计算和存储两部分构成，如何根据业务的需求，选择合适的配置，我们根据实际运营经验，在此提供一些 ES 常见使用场景下，配置选择的建议。您可以根据业务需要进行参考，当然，最好的方法还是需要您在业务的实际使用过程中逐步去探索。依托腾讯云 ES 提供的弹性伸缩机制，在业务规模增大，性能遇到瓶颈的时候，您可以随时扩容，调整得到合适的集群规格。

存储容量评估

影响腾讯云 ES 服务存储容量的主要因素如下：

- 副本数量：副本有利于增加数据的可靠性，但同时会增加存储成本。默认和建议的副本数量为1，对于部分可以承受异常情况导致数据丢失的场景，可考虑设置副本数量为0。
- 数据膨胀：除原始数据外，ES 需要存储索引、列存数据等，在应用编码压缩等技术后，一般膨胀10%。
- 内部任务开销：ES 占用约20%的磁盘空间，用于 segment 合并、ES Translog、日志等。
- 操作系统预留：Linux 操作系统默认为 root 用户预留5%的磁盘空间，用于关键流程处理、系统恢复、防止磁盘碎片化问题等。

因此，数据在 ES 中占用的实际空间可通过下面公式估算：

$$\begin{aligned} \text{实际空间} &= \text{源数据} \times (1 + \text{副本数量}) \times (1 + \text{数据膨胀}) / (1 - \text{内部任务开销}) / (1 - \text{操作系统预留}) \\ &\approx \text{源数据} \times (1 + \text{副本数量}) \times 1.45 \end{aligned}$$

为保证服务的稳定运行，建议至少预留15%的存储空间，因此建议申请的存储容量为：

$$\begin{aligned} \text{存储容量} &= \text{源数据} \times (1 + \text{副本数量}) \times 1.45 \times (1 + \text{预留空间}) \\ &\approx \text{源数据} \times (1 + \text{副本数量}) \times 1.67 \end{aligned}$$

对于内存与存储容量的关系，我们有如下经验比例，一般而言，为保证集群运行的性能和稳定，不建议磁盘存储超过这个比例限制。

- 对于热数据，我们推荐内存与磁盘的比例为：1:96
- 对于冷数据，我们推荐内存与磁盘的比例为：1:480

注意：

ES 广泛应用于日志、站点检索、Metrics、APM 等场景，上述存储容量评估方法较为泛化，您可以根据自身需求高度优化 ES，降低存储成本。

计算资源评估

ES 的计算资源主要消耗在写入和查询过程，而不同业务场景在写入和查询方面的复杂度不同、比重不同，导致计算资源相比存储资源较难评估。但一般情况下，存储资源会较早成为瓶颈，因此建议您优先评估存储资源量，然后初步选择计算资源，在测试过程中确认计算资源是否足够。

下面针对几种常见使用场景，介绍计算资源评估过程中的一些经验：

- 日志场景：日志属于典型的写多读少类场景，计算资源主要消耗在写入过程中。我们在日志场景的经验是：2核8GB内存的资源最大可支持0.5万次写入/s的写入能力，但注意不同业务场景可能有偏差。由于实例性能基本随计算资源总量呈线性扩容，您可以按实例资源总量估算写入能力。例如8核32GB内存的资源可支持2万次写入/s的写入能力。
- Metric 及 APM 等结构化数据场景：这也是写多读少类场景，但相比日志场景计算资源消耗较小，2核8GB内存的资源一般可支持1万次写入/s的写入能力，您可参照日志场景线性扩展的方式，评估不同规格实例的实际写入能力。
- 站内搜索及应用搜索等搜索场景：此类为读多写少类场景，计算资源主要消耗在查询过程，由于查询复杂度在不同使用场景差别非常大，计算资源也最难评估，建议您结合存储资源初步选择计算资源，然后在测试过程中验证、调整。

实例类型选择及测试

在完成存储、计算资源评估后，您可以初步选择实例类型，这里包含节点规格和节点数量两方面。选择实例类型的常用建议如下：

- 建议您至少选择3个节点，避免 ES 实例出现脑裂问题，保证 ES 实例具有较高的节点故障容错能力。

说明：

脑裂：两个节点同时认为自己是唯一处于活动状态的服务器，从而出现争用资源的情况。

- 若您有非常大的存储容量需求，建议选择高规格的节点，避免大量低规格节点，这对大实例的性能、稳定性等有较大好处。例如，若您有40核160GB内存5TB存储容量需求，建议选择8核32GB内存1TB × 5节点的实例。同理，

当您需要对实例扩容时，建议优先进行纵向扩容，把节点扩容到8核32GB或16核64GB的规格，然后再考虑横向扩容增加节点个数。

- 当完成实例类型的初步选择后，您可以使用真实数据进行测试，通过观察 CPU 使用率、写入指标（性能、拒绝率）、查询指标（QPS、拒绝率）等监控信息，进一步确认实例类型是否合适。另外，建议针对上述监控信息配置告警，方便在线上使用时，及时发现资源不足等问题。

分片数量评估

每个 ES 索引被分为多个分片，数据按哈希算法打散到不同的分片中。由于索引分片的数量影响读写性能、故障恢复速度，且通常无法轻松更改，需要提前考虑。这里给出配置分片数量的一些常用建议：

- 建议单个分片大小保持在10GB - 50GB之间，您可以据此初步确定 Index 的分片数量。分片不宜过大或过小：过大可能使 ES 的故障恢复速度变慢；过小可能导致非常多的分片，但因为每个分片使用一些数量的 CPU 和内存，从而导致读写性能、内存不足等问题。
- 在测试阶段，可以根据每个 Index 的实际大小、预期未来增长情况，适当调整分片数量。
- 当分片数量超过数据节点数量时，建议分片数量接近数据节点的整数倍，方便分片在所有数据节点均匀分布。
- 对于日志、Metric 等场景中，建议使用 ES 自带的 [Rollover Index](#) 功能，持续滚动产生新的 Index。方便在发现分片大小不合理时，通过该功能及时调整分片数量。

例如，假设实例有5个数据节点，Index 当前大小为150GB，预期半年后增长50%。如果我们控制每个单分片为30GB，则大约需要 $150\text{GB} \times (1 + 50\%) / 30 \approx 7$ 个分片，考虑到有两个数据节点支撑2/7的数据压力，节点间压力相对不均匀，我们把分片数量调整到10个。

创建集群

最近更新时间：2021-10-29 15:18:28

集群是 ES 提供托管 Elasticsearch 服务的基本单元，也是用户使用和管理 Elasticsearch 服务的主要对象。本文为您介绍通过腾讯云官网控制台，快速创建 Elasticsearch 集群。

前提条件

已创建腾讯云账号，创建账号可参考 [注册腾讯云](#)。

操作步骤

登录控制台

登录 [ES 控制台](#)，单击【新建】进入创建购买页。

创建集群

一、选择集群配置

- 计费模式：支持【按量计费】。
- 地域：目前腾讯云 ES 已在国内和境外多个地域上线。
- Elasticsearch 版本：支持5.6.4、6.4.3、6.8.2、7.5.1版本。
- 高级特性：Elasticsearch 官方商业特性（原 X-Pack 插件），包括数据权限管理、SQL JDBC、告警和机器学习等能力。不同版本包含的功能完整度不同，白金版包含所有高级特性，基础版包含一部分高级特性，开源版不包含高级特性，具体版本功能差异详见 [高级特性\(X-Pack\)](#)。

注意：

基础版从6.8版本开始支持开启 [ES 集群用户登录认证](#)，可以极大提高集群的安全性，强烈建议开启。如果购买集群时未开启，以后再开启需要全量重启集群，会对在线服务有一定影响。白金版默认开启 ES 集群用户登录认证，开源版不支持此能力。

- 网络/可用区及子网：ES 部署在 VPC 内，只有在同一个 VPC 下才能访问 ES 集群，因此为保证内网顺利访问，建议选择已有云上业务的区域位置所在 VPC。多可用区部署模式下，也是选择同一个 VPC。同一个 VPC 内，不同可用区子网之间是互通的。

注意：

ES 集群创建完成后，不支持修改调整 VPC。

特殊网络问题说明

- 跨 VPC 访问：如果需要通过同一个地域内位于其他 VPC 的 CVM 访问 ES 集群，可以考虑 [对等连接](#) 方案，将同一地域两个不同的 VPC 进行打通。
- 基础网络访问：如果您的业务都部署在基础网络内，且没有使用过 VPC，可以通过基础网络和 VPC 网络互通的方案，将位于基础网络的 CVM 绑定到 ES 集群所在 VPC，从而实现基础网络到 ES 集群的互通（具体可参见 [基础网络互通](#)）。基础网络互通功能仅支持 `10.[0~47].0.0/16` 网段内的私有网络，如果您需要从基础网络访问 ES 集群，请在创建 ES 集群时选择有 `10.[0~47].0.0/16` 网段的 VPC。

- 可用区部署模式：单可用区是 ES 集群部署在一个可用区内；多可用区是可以部署在同一个地域的两个可用区。多可用区的部署方式可以提高集群的容灾能力，保障线上业务的稳定性。可参考 [集群多可用区部署](#)。
- 数据节点部署模式：支持单一模式和冷热模式。冷热集群是可以在一个集群内包含冷、热两种属性的节点，对读写性能要求高的热点数据可以存放在热节点，对存储量需求比较大但对读写性能要求较低的索引可以存放在冷节点上，既保证了热数据的读写性能，又可以降低存储成本。详细介绍请参见 [冷热分离与索引生命周期管理](#)。
- 节点机型：集群中每个节点机型的规格，不同规格包含不同的 CPU 核数和内存，ES 支持的节点规格类型以及选择建议可参见 [节点类型存储配置建议](#)。
- 节点存储类型：支持高性能云盘和 SSD 云盘。
- 单节点存储：每个节点配置的磁盘容量，整个集群的存储量 = 单个节点存储 * 节点个数。
- 专用主节点：当集群具有一定规模时，可以 [配置专用主节点](#)，进一步保障集群稳定性。
- 专用主节点机型：专用主节点机型不要求同数据节点机型一致。
- 专用主节点个数：可选3个或5个，奇数个能保证高可用并预防脑裂风险。
- Kibana 节点：系统免费提供，1核2G，暂不支持修改。

Elasticsearch Service

1. Select Cluster Configuration
2. Set Name and Password
3. Confirm the Configuration Information

Billing Mode: Pay-as-you-go ?

Elasticsearch Version: 7.10.1 7.5.1 6.8.2 6.4.3 5.6.4

Advanced Features: Basic edition Platinum edition Open-source edition ?

For platinum edition, the account permission verification feature of clusters will be enabled. In order to access clusters normally, you need to set an account password for actions such as Kibana login, Logstash and Beats log collecting, and access to a cluster client through API call. For details, see [Features of Each Edition](#)

Region: Guangzhou Shanghai Nanjing Hangzhou Beijing Chengdu Chongqing Hong Kong (China)

Cloud products in different regions are not interconnected over private networks. Select the region nearest to your customers to reduce the access latency. The region cannot be changed after the cluster is created.

AZ Deployment Mode: Single AZ Dual AZs Three AZs ?

Network: vpc-7vt2eoo7 | inti-test ?

If the existing VPCs do not meet your requirements, you can [create a VPC](#) on the console. The VPC cannot be changed after created.

AZ and Subnet: Guangzhou Zone 7 Select a subnet ?

If the existing subnets do not meet your requirements, you can [create a subnet](#) on the console. The subnet cannot be changed after created.

Node Deployment (Select One or More) Don't know how to set the cluster size? Click to [estimate cluster configuration](#).

Data Node x 3
StandardSA2, 2-core 4 GB
20GB Cloud SSD x 1

Warm data nodes
Disabled

Dedicated Master Node
Disabled

Kibana Node x 1
StandardSA2, 1-core 2 GB

DataNode Model: Standard High IO Big data
The model cannot be changed after purchased.

DataNode Specification: ES.SA2.MEDIUM4-2 cores 4 G Configuration Recommendations

Data Disks per Node: Cloud SSD - 20 + GB - 1 + disks ?
After purchased, the data disks cannot be changed.

DataNode Qty: - 3 +

二、设置名称及密码

单击【下一步】，设置名称及密码。

- 集群名：自定义集群名称，不作为全局唯一标示，可以设置为业务相关描述。
- 用户名：默认为 elastic，不能修改。用于 Kibana 页面登录，以及 [ES 集群用户登录认证](#)。
- 密码：上一项用户名对应的密码，请按照规则设置，并妥善保存，如果密码忘记，可在详情页重置。

- 标签：标签是腾讯云的统一服务，通过它来对腾讯云已有资源进行分类管理。在购买 ES 集群时或者购买后都可以创建或者修改标签。详细介绍请参见 [标签](#)。

1. Select Cluster Configuration
2. Set Name and Password
3. Confirm the Configuration Information

Cluster Name ?

1-50 characters of English letters, Chinese characters, numbers, dashes (-) or underscores (_) are supported.

Cluster Configuration Name ?

1-64 characters of letters, numbers, or underscores are supported

Username **elastic**

Used for Kibana login and user authentication. (Note: the open-source edition does not has the user authentication feature. For the t disable this feature. This feature is enabled by default for the platinum edition.

Password

8-16 characters, including at least three out of the following four types of characters: [a-z], [A-Z], [0-9] and [-!@#\$\$%&^*+=_~:,;.,?]

Confirm Password

Default scenario configuration

General Contains general optimization configuration items, suitable for a variety of scenarios.

Log Scenarios where there are more writes than reads and that do not require high real-time performance

Search Nearly real-time scenarios that require high query performance

Not now

The above scenario configuration is an initial index template provided based on different business characteristics and experience i after the cluster is created. [Learn More](#) 🔗

Tag

Tag key ▼

Tag value ▼

✕

+ Add

三、确认配置信息

单击【下一步】，确认配置信息。

1. Select Cluster Configuration
2. Set Name and Password

3. Confirm the Configuration Information

Configuration Information	
Billing Mode	Pay-as-you-go
Region	Guangzhou
Elasticsearch Version	7.10.1
Advanced Features	Platinum edition
Network	██████████
AZ Deployment Mode	Single AZ
AZ and Subnet	██████████
Enable Hot/Warm Deployment Mode	Not configured
Node Model	StandardSA2 2-core 4 GB(ES.SA2.MEDIUM4)
Node Storage	20GB Cloud SSD x 1
Node Qty	3
Dedicated Master Node	Not configured
Kibana Node Specification	StandardSA2 1-core 2 GB(ES.SA2.SMALL2)
Kibana Node Count	1
Cluster Name	██████████
Scenario Configuration	General

Fees **USD/hour**

Agree to "Tencent Cloud Elasticsearch Service - Service Level Agreement" [🔗](#)

Back
Purchase Now

- 对按量计费，单击【开通】直接创建集群，不需要确认订单支付，在使用过程中，会按小时进行计费及扣费。

四、创建完成

后付费集群开通成功后，就可以跳转到 [控制台](#) 查看刚创建的集群，集群将在几分钟内创建完成。

后续步骤

集群访问

ES 提供了多种类型的客户端访问集群，快速开始，详情请参见 [通过 API 访问集群](#)、[通过客户端访问集群](#)、[通过 Kibana 访问集群](#)。

集群监控

腾讯云 ES，提供了丰富的监控指标，可以帮助在集群使用过程中查看集群的状态。详情请参见 [查看监控](#)。

基础版和白金版的 Kibana 页面也有监控项，在左侧菜单的【monitor】页。

集群调整配置

随着业务数据量和访问量的增加，可以弹性地调整集群配置，详情请参见 [调整配置](#)。

访问集群

ES 集群用户登录认证

最近更新时间：2021-10-29 15:08:49

在了解如何访问 ES 集群前，您需要先了解 ES 集群用户登录认证。

ES 集群用户登录认证（User authentication）

此功能用于提高 ES 集群的数据访问安全（可参阅 [保护您在 Elastic Stack 中的数据](#)），用户必须通过用户名和密码认证，才被允许访问 ES 集群。无论是通过 Kibana、客户端或者 API 等方式访问集群都需要经过认证。详情请参考 [通过 API 访问集群](#)、[通过客户端访问集群](#)、[通过 Kibana 访问集群](#)。

说明：

在创建 ES 集群时会要求设置用户名和密码。

- 对于未开启此功能的集群，用户名和密码仅用于 Kibana 的登录。
- 对于已开启此功能的集群，用户名和密码可用于任何方式下的 ES 集群用户登录认证。

支持 ES 集群用户登录认证的 Elasticsearch 版本

并非所有的 Elasticsearch 版本都支持此功能，各版本情况如下：

- 开源版：不支持此能力。
- 基础版：从6.8版本开始，允许用户开启或关闭此功能，**6.8版本之前的版本不支持此功能**。
- 白金版：强制开启此能力，且不允许关闭。

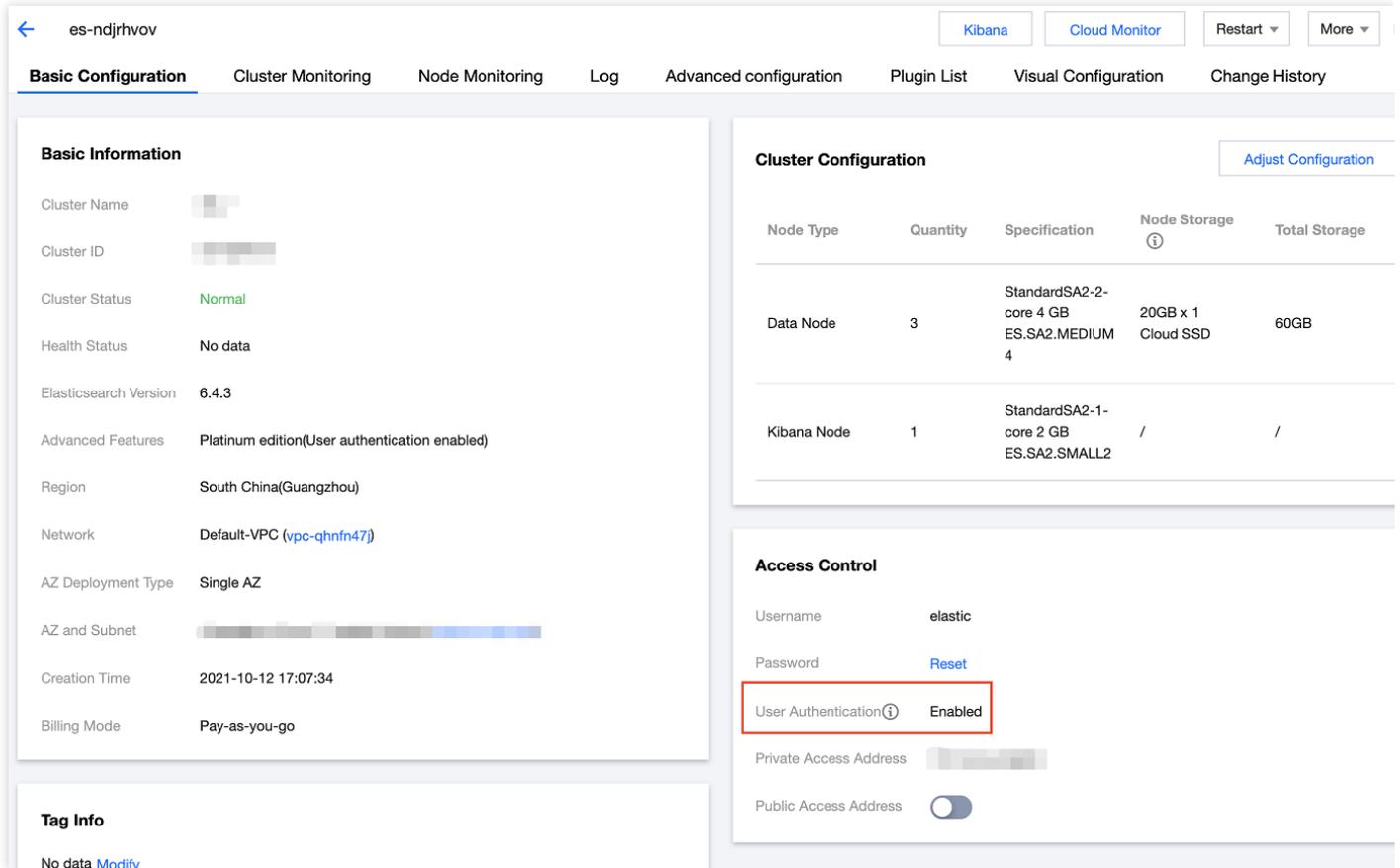
ES 集群开启用户登录认证注意事项

若 ES 集群未开启用户登录认证，需要通过升级或开启开关来开启用户登录认证时，应该先完成以下步骤：

- 提前修改业务代码，在调用 API 时传入用户名和密码，以便在开启此功能后可以正常访问集群。
- 根据 Elasticsearch 官方的设计要求，开启此能力需要全量重启集群，重启期间集群不可用，因此应谨慎选择合适的时机。

6.8及以上基础版如何开启和关闭 ES 集群用户认证

- 在创建集群时可以选择开启或关闭 ES 集群用户认证。
- 集群创建完成后，如果需要再次调整开关状态，可以到集群详情页进行操作。



The screenshot displays the configuration page for an Elasticsearch cluster named 'es-ndjrhvov'. The page is divided into several sections:

- Basic Information:**
 - Cluster Name: [Redacted]
 - Cluster ID: [Redacted]
 - Cluster Status: **Normal**
 - Health Status: No data
 - Elasticsearch Version: 6.4.3
 - Advanced Features: Platinum edition (User authentication enabled)
 - Region: South China (Guangzhou)
 - Network: Default-VPC (vpc-qhfn47)
 - AZ Deployment Type: Single AZ
 - AZ and Subnet: [Redacted]
 - Creation Time: 2021-10-12 17:07:34
 - Billing Mode: Pay-as-you-go
- Cluster Configuration:**

Node Type	Quantity	Specification	Node Storage	Total Storage
Data Node	3	StandardSA2-2-core 4 GB ES.SA2.MEDIUM 4	20GB x 1 Cloud SSD	60GB
Kibana Node	1	StandardSA2-1-core 2 GB ES.SA2.SMALL2	/	/
- Access Control:**
 - Username: elastic
 - Password: [Redacted] [Reset](#)
 - User Authentication: **Enabled** (highlighted with a red box)
 - Private Access Address: [Redacted]
 - Public Access Address:
- Tag Info:** No data [Modify](#)

通过客户端访问集群

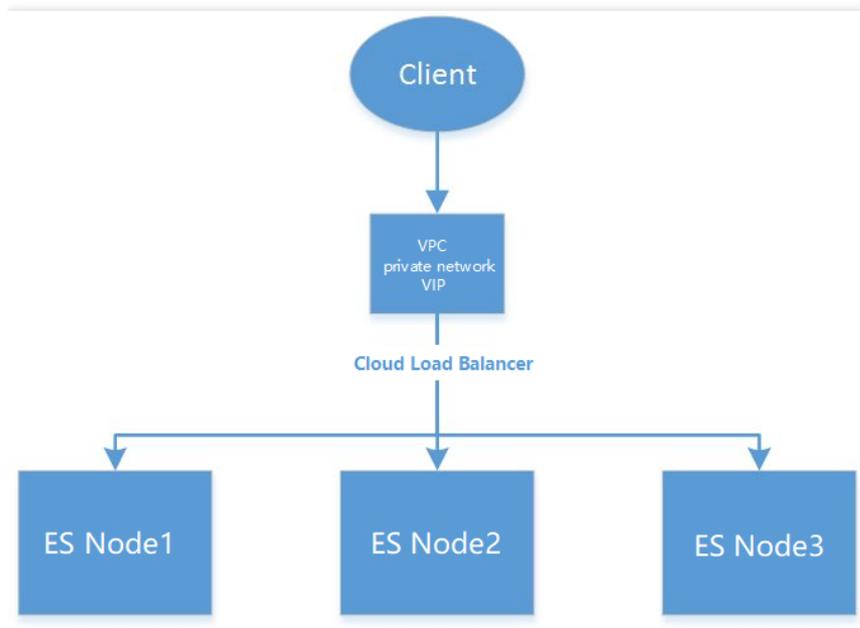
最近更新时间：2020-08-21 17:31:42

官方和社区推出了各个语言版本的 SDK 方便用户使用。

- Elasticsearch 提供了各种开发语言的客户端，例如 Java、Python 等，以满足不同开发者的需要。详情请参见 [Elasticsearch Client](#)。
- Elasticsearch 从5.6.0版本开始，发布了官方的 Java High Level REST Client，可用于执行 search、index、delete、update and bulk 操作，同 TransportClient 使用的核心 Java classes 一致，并且是为了替代 TransportClient。详情请参见 [Java High Level REST Client 说明](#)。

版本兼容性上，建议客户端版本同服务端版本保持一致，详情请参见 [兼容性说明](#)，腾讯云 ES 目前提供了多个可选的 Elasticsearch 版本，请注意选择客户端版本。

ES 在用户的 VPC 下提供了一个访问 ES 集群的 VIP，通过负载均衡的方式挂载了 ES 集群内的所有节点。这样做的目的，一方面是为了适配集群的弹性伸缩并保证其高可用性，当 ES 集群的节点发生变化时，VIP 会自动更新节点信息；另一方面，简化用户的操作，用户不用再关注集群节点 IP、port 等信息的变更。



不推荐通过 TransportClient 直接连接 Client 和 ES 节点。

ES 的 SDK 支持在创建连接时只配置一个节点的地址，并通过“节点嗅探”的功能嗅探出所有的节点来连接使用。这种使用方式有悖于我们推出 VIP 功能的初衷，增加了使用 ES 集群的复杂度。下面我们会简单介绍各个编程语言 SDK 包的使用方法，并说明如何关闭“节点嗅探”功能。

Java 客户端

ES 官方推荐使用 Java REST 客户端连接集群并进行数据操作。Java REST Client 有 Low Level 和 High Level 两种：

- Java Low Level REST Client：使用该客户端需要将 HTTP 请求的 body 手动拼成 JSON 格式，HTTP 响应也必须将返回的 JSON 数据手动封装成对象。
- Java High Level REST Client：该客户端基于 Low Level 客户端实现，提供 API 解决 Low Level 客户端需要手动转换数据格式的问题。

使用 Java High Level REST Client 访问集群，示例步骤与代码如下：

添加 Maven 依赖

建议 Java REST 客户端 API 版本，同 ES 集群服务端版本保持一致，腾讯云目前提供了不同版本的 ES，请注意选择相应版本的客户端。更多客户端 API 信息，可参考官方 [ES API 版本兼容性说明](#)。

```
<dependency>
<groupId>org.elasticsearch.client</groupId>
<artifactId>elasticsearch-rest-high-level-client</artifactId>
<version>6.4.3</version>
</dependency>
```

- Client 版本需要与 ES 集群版本保持一致，否则可能会出现兼容性问题。此处的 Demo 适用于腾讯云 ES 6.4.3 版本，其他版本的使用方法请参考 [Java High Level REST Client](#)。

- Java High REST Client 构建于 Java Low REST Client 之上，两种 Client 都是基于 HTTP 协议连接 ES 集群，Java High REST Client 可提供的 API 种类随着版本升级会越来越多，如果当前版本的 Java High REST Client 提供的 API 不满足需求，可以通过升级 ES 集群版本和 Client 版本解决。
- 使用 TCP 协议连接 ES 集群的 Transport Client 官方已经不再维护，建议使用 HTTP 协议连接集群的 Java High REST Client 或者 Java Low REST Client。详情请参考 elastic 官方文档 [由 TransportClient 迁移至 Java High REST](#)

Client。

示例代码

```
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.log4j.BasicConfigurator;
import org.apache.log4j.Logger;
import org.elasticsearch.ElasticsearchException;
import org.elasticsearch.action.DocWriteResponse;
import org.elasticsearch.action.index.IndexRequest;
import org.elasticsearch.action.index.IndexResponse;
import org.elasticsearch.action.get.GetRequest;
import org.elasticsearch.action.get.GetResponse;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.rest.RestStatus;

import java.util.Date;
import java.util.HashMap;
import java.util.Map;

public class test_es_sdk {
    private static Logger logger = Logger.getLogger(test_es_sdk.class);

    public static void main(String[] args) {
        BasicConfigurator.configure();
        // 设置验证信息, 填写账号及密码
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY,
            new UsernamePasswordCredentials("user", "passwd"));
        // 初始化 RestClient, hostName 和 port 填写集群的内网 VIP 地址与端口
        RestClientBuilder builder = RestClient.builder(new HttpHost("xx.xx.xx.xx", 9200,
            "http"));
        // 设置认证信息
        builder.setHttpClientConfigCallback(new RestClientBuilder.HttpClientConfigCallback() {
            @Override
            public HttpAsyncClientBuilder customizeHttpClient(HttpAsyncClientBuilder httpClientBuilder) {
```

```
return httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
}
});
// 设置超时时间
builder.setMaxRetryTimeoutMillis(10000);
// 由Low Level Client构造High Level Client
RestHighLevelClient client = new RestHighLevelClient(builder);

// 索引文档
Map<String, Object> jsonMap = new HashMap<String, Object>();
jsonMap.put("user", "bellen");
jsonMap.put("name", new Date());
jsonMap.put("message", "trying out Elasticsearch");
IndexRequest indexRequest = new IndexRequest("posts", "doc", "1")
    .source(jsonMap);

try {
    // 获取响应结果
    IndexResponse indexResponse = client.index(indexRequest);
    String index = indexResponse.getIndex();
    String type = indexResponse.getType();
    String id = indexResponse.getId();
    long version = indexResponse.getVersion();

    if (indexResponse.getResult() == DocWriteResponse.Result.CREATED) {
        logger.info("doc indexed, index: " + index + ", type:" + type + ",id:" + id + ",version:" + version);
    } else if (indexResponse.getResult() == DocWriteResponse.Result.UPDATED) {
        logger.info("doc updated, index: " + index + ", type:" + type + ",id:" + id + ",version:" + version);
    }
} catch (ElasticsearchException e) {
    if (e.status() == RestStatus.CONFLICT) {
        logger.error("version conflict");
    }
} catch (Exception e) {
    logger.error("execute index api failed, " + e.toString());
}

// 查询文档
GetRequest getRequest = new GetRequest(
    "posts",
    "doc",
    "1");
try {
```

```
// 获取响应结果
GetResponse getResponse = client.get(getRequest);
String index = getResponse.getIndex();
String type = getResponse.getType();
String id = getResponse.getId();
if (getResponse.isExists()) {
    long version = getResponse.getVersion();
    String sourceAsString = getResponse.getSourceAsString();
    logger.info("get doc, index: " + index + ", type:" + type + ",id:" + id + ",version:" + version + ", source:" + sourceAsString);
}
} catch (ElasticsearchException e) {
    if (e.status() == RestStatus.NOT_FOUND) {
        logger.warn("doc not found");
    }
}
}
catch (Exception e) {
    logger.error("execute get api failed, " + e.toString());
}

// 关闭客户端
try {
    client.close();
} catch (Exception e) {
    logger.error("close rest client exception:" + e.toString());
}
}
}
```

Python 客户端

pip 安装

```
pip install elasticsearch
```

示例代码

在 `Elasticsearch` 函数的参数中设置如下3个参数关闭节点嗅探：

- `sniff_on_start=False`
- `sniff_on_connection_fail=False`
- `sniffer_timeout=None`

```
from elasticsearch import Elasticsearch

es = Elasticsearch(["http://xx.xx.xx.xx:9200"],
http_auth=('user', 'passwd'),
sniff_on_start=False,
sniff_on_connection_fail=False,
sniffer_timeout=None)

res = es.index(index="my_index", doc_type="my_type", id=1, body={"title": "One",
"tags": ["ruby"]})
print(res)
res = es.get(index="my_index", doc_type="my_type", id=1)
print(res['_source'])
```

PHP 客户端

为了避免节点的嗅探，您可以设置显示的连接池类为 `StaticConnectionPool`，不要使用节点嗅探连接池。

示例代码

```
$client = ClientBuilder::create()
->setConnectionPool('\Elasticsearch\ConnectionPool\StaticConnectionPool', ["htt
p://user:passwd@xx.xx.xx.xx:9200"])
->build();
```

Go 客户端

`gopkg.in/olivere/elastic` 是在 Go 语言中广泛使用的一个由社区贡献的 SDK。此处的 Demo 适用于腾讯云 ES 6.4.3版本，其他版本的使用方法请参考 [Elastic](#)。

安装 elastic

```
go get github.com/olivere/elastic
```

示例代码

在 `NewClient` 函数的参数中设置 `elastic.SetSniff(false)` 关闭节点嗅探，设置 `elastic.SetHealthcheck(false)` 关闭对节点的健康检查。

```
import (  
    "context"  
    "fmt"  
  
    "github.com/olivere/elastic"  
)  
  
func main() {  
    client, err := elastic.NewClient(elastic.SetURL("http://user:passwd@xx.xx.xx.xx:9  
200"),  
    elastic.SetSniff(false), elastic.SetHealthcheck(false))  
    if err != nil {  
        panic(err)  
    }  
    exists, err := client.IndexExists("twitter").Do(context.Background())  
    if err != nil {  
        panic(err)  
    }  
    fmt.Println(exists)  
}
```

通过 API 访问集群

最近更新时间：2021-05-18 16:02:49

Elasticsearch 提供了功能全面的 RESTful API 与集群交互，详情请参见 Elasticsearch 官方的 [API 文档](#)。

腾讯云 ES 构建在用户 VPC 内，用户可以通过位于同一 VPC 下的 CVM 作为客户端访问 ES 集群。可通过**内网访问**和**外网访问**两种方式访问 ES 集群，**外网访问存在安全风险**，需谨慎开启。

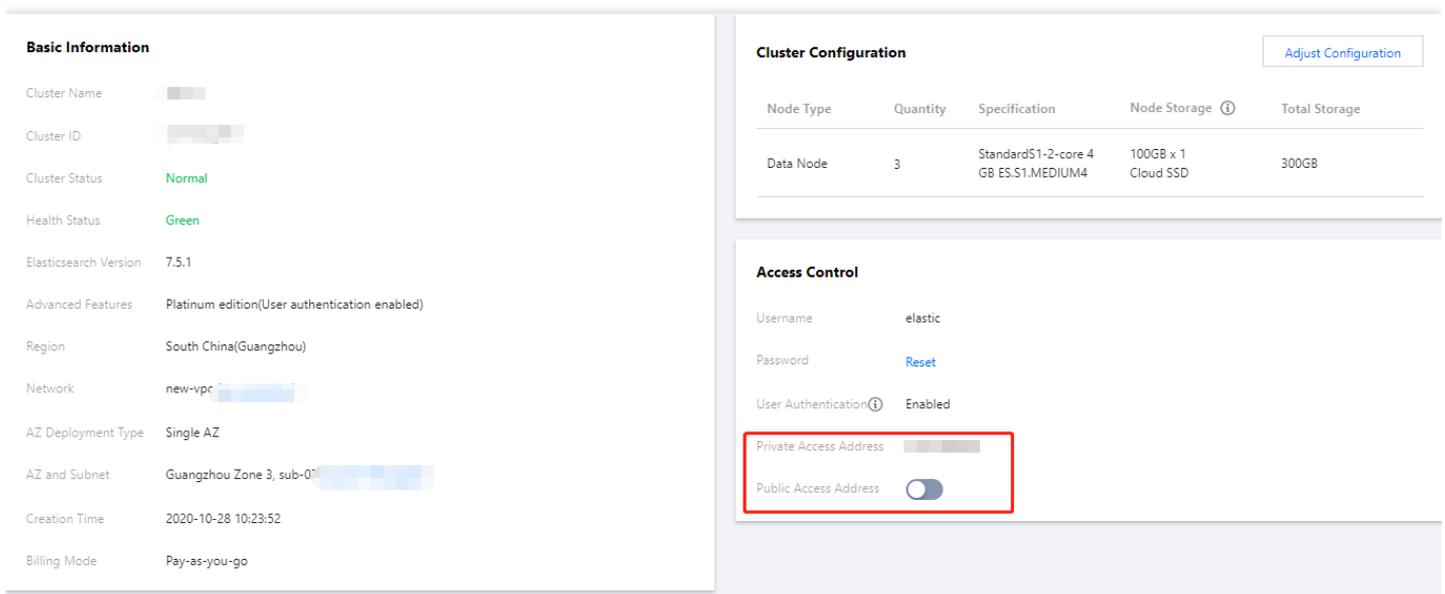
说明：

- 外网访问仅用于开发调试，因系统会限制调用频次，所以不能用于生产环境。
- 当前 ES 公网访问不计费，带宽10M。

查看内外网访问地址

在 [集群列表页](#)，单击【集群 ID】进入详情页：

- 对于内网地址，在基础配置中可直接查看。
- 对于外网地址，出于安全考虑默认是关闭的。对于已开启 [ES 集群用户登录认证](#) 的集群，支持开启公网地址。开启公网访问可能会为集群引入安全风险，同时也将允许通过 API 直接访问、操作甚至删除在 ElasticSearch 集群中的数据，请谨慎开启。



The screenshot displays the console interface for an Elasticsearch cluster. It is divided into two main sections: Basic Information and Cluster Configuration.

Basic Information:

- Cluster Name: [Redacted]
- Cluster ID: [Redacted]
- Cluster Status: Normal
- Health Status: Green
- Elasticsearch Version: 7.5.1
- Advanced Features: Platinum edition(User authentication enabled)
- Region: South China(Guangzhou)
- Network: new-vpc
- AZ Deployment Type: Single AZ
- AZ and Subnet: Guangzhou Zone 3, sub-01
- Creation Time: 2020-10-28 10:23:52
- Billing Mode: Pay-as-you-go

Cluster Configuration:

Node Type	Quantity	Specification	Node Storage	Total Storage
Data Node	3	StandardS1-2-core 4 GB ES.S1.MEDIUM4	100GB x 1 Cloud SSD	300GB

Access Control:

- Username: elastic
- Password: Reset
- User Authentication: Enabled
- Private Access Address:
- Public Access Address:

测试访问

可通过 curl 的方式测试访问集群，不支持通过 ping 的方式测试连通性。

测试服务是否可访问

说明：

对于已开启 [ES 集群用户登录认证](#) 的集群，登录时需要用户名和密码认证，具体规则为 `curl action -u user:password host ...`，需要将 `user`、`password` 替换为自己实际的用户名和密码，将 `host` 替换为自己的 IP。

下面将以内网地址访问来演示各访问操作。

输入命令：

```
curl -XGET http://10.0.17.2:9200
若开启了ES集群用户登录认证，请注意输入用户名密码
curl -XGET -u user:password http://10.0.17.2:9200
```

返回如下，表示集群访问正常，具体参数的值会根据集群的版本有所不同：

```
{
  "name": "1558982657000*****",
  "cluster_name": "es-*****",
  "cluster_uuid": "NGIm1M_zRw-L3o_gH*****",
  "version": {
    "number": "6.4.3",
    "build_flavor": "default",
    "build_type": "zip",
    "build_hash": "fe40335",
    "build_date": "2019-05-17T14:22:47.286024Z",
    "build_snapshot": false,
    "lucene_version": "7.4.0",
    "minimum_wire_compatibility_version": "5.6.0",
    "minimum_index_compatibility_version": "5.0.0"
  },
  "tagline": "You Know, for Search"
}
```

创建文档

创建单个文档

- 若集群未开启用户登录认证，输入命令行：

```
curl -XPUT http://10.0.0.2:9200/china/city/beijing -H 'Content-Type: application/json' -d'
{
  "name": "北京市",
  "province": "北京市",
  "lat": 39.9031324643,
  "lon": 116.4010433787,
  "x": 6763,
  "level.range": 4,
  "level.level": 1,
  "level.name": "一线城市",
  "y": 6381,
  "cityNo": 1
}
```

- 若集群已开启用户登录认证，需要将下文中的 **user**、**password** 替换为自己集群实际的用户名和密码。输入命令行：

```
curl -XPUT -u user:password http://10.0.0.2:9200/china/city/beijing -H 'Content-Type: application/json' -d'
{
  "name": "北京市",
  "province": "北京市",
  "lat": 39.9031324643,
  "lon": 116.4010433787,
  "x": 6763,
  "level.range": 4,
  "level.level": 1,
  "level.name": "一线城市",
  "y": 6381,
  "cityNo": 1
}
```

响应如下：

```
{
  "_index": "china",
  "_type": "city",
  "_id": "beijing",
  "_version": 1,
  "result": "created",
```

```
"_shards":{
  "total":2,
  "successful":1,
  "failed":0
},
"created":true
}
```

创建多个文档

输入命令行：

```
curl -XPOST http://10.0.0.2:9200/_bulk -H 'Content-Type: application/json' -d'
{ "index" : { "_index": "china", "_type" : "city", "_id" : "beijing" } }
{"name":"北京市","province":"北京市","lat":39.9031324643,"lon":116.4010433787,"x":6
763,"level.range":4,"level.level":1,"level.name":"一线城市","y":6381,"cityNo":1}
{ "index" : { "_index": "china", "_type" : "city", "_id" : "shanghai" } }
{"name":"上海市","province":"上海市","lat":31.2319526784,"lon":121.469443249,"x":77
79,"level.range":4,"level.level":1,"level.name":"一线城市","y":4409,"cityNo":2}
{ "index" : { "_index": "china", "_type" : "city", "_id" : "guangzhou" } }
{"name":"广州市","province":"广东省越秀区吉祥路79号","lat":23.1317146641,"lon":113.25
95185241,"x":6173,"level.range":4,"level.level":1,"level.name":"一线城市","y":256
0,"cityNo":3}
{ "index" : { "_index": "china", "_type" : "city", "_id" : "shenzhen" } }
{"name":"深圳市","province":"广东省福田区新园路37号","lat":22.5455465546,"lon":114.05
27779134,"x":6336,"level.range":4,"level.level":1,"level.name":"一线城市","y":242
9,"cityNo":4}
{ "index" : { "_index": "china", "_type" : "city", "_id" : "chengdu" } }
{"name":"成都市","province":"四川省锦江区红星路4段-88号-附1号","lat":30.6522796787,"lo
n":104.0725574128,"x":4387,"level.level":2,"level.range":19,"level.name":"新一线城市
市","y":4304,"cityNo":5}
{ "index" : { "_index": "china", "_type" : "city", "_id" : "hangzhou" } }
{"name":"杭州市","province":"浙江省拱墅区环城北路316号","lat":30.2753694112,"lon":12
0.1509063337,"x":7530,"level.level":2,"level.range":19,"level.name":"新一线城市","
y":4182,"cityNo":6}
'
```

响应如下：

```
"took":9,"errors":false,"items":[{"index":{"_index":"china","_type":"city","_id":
"beijing","_version":4,"result":"updated","_shards":{"total":2,"successful":2,"fa
iled":0},"created":false,"status":200}},{ "index":{"_index":"china","_type":"city"
,"_id":"shanghai","_version":2,"result":"updated","_shards":{"total":2,"successfu
l":2,"failed":0},"created":false,"status":200}},{ "index":{"_index":"china","_typ
e":"city","_id":"guangzhou","_version":1,"result":"created","_shards":{"total":2,
"successful":2,"failed":0},"created":true,"status":201}},{ "index":{"_index":"chin
```

```
a", "_type": "city", "_id": "shenzhen", "_version": 1, "result": "created", "_shards": {"total": 2, "successful": 2, "failed": 0}, "created": true, "status": 201}}, {"index": {"_index": "china", "_type": "city", "_id": "chengdu", "_version": 2, "result": "updated", "_shards": {"total": 2, "successful": 2, "failed": 0}, "created": false, "status": 200}}, {"index": {"_index": "china", "_type": "city", "_id": "hangzhou", "_version": 2, "result": "updated", "_shards": {"total": 2, "successful": 2, "failed": 0}, "created": false, "status": 200}}]
```

更新文档

重复上文创建单个文档的输入代码，即可更新指定 ID `beijing` 的文档。响应如下：

```
{"_index": "china", "_type": "city", "_id": "beijing", "_version": 2, "result": "updated", "_shards": {"total": 2, "successful": 2, "failed": 0}, "created": false}
```

查询文档

查询指定 ID

输入命令行：

```
curl -XGET 'http://10.0.0.2:9200/china/city/beijing?pretty' -H 'Content-Type: application/json'
```

响应如下：

```
{
  "_index" : "china",
  "_type" : "city",
  "_id" : "beijing",
  "_version" : 4,
  "found" : true,
  "_source" : {
    "name" : "北京市",
    "province" : "北京市",
    "lat" : 39.9031324643,
    "lon" : 116.4010433787,
    "x" : 6763,
    "level.range" : 4,
    "level.level" : 1,
    "level.name" : "一线城市",
    "y" : 6381,
```

```
"cityNo" : 1
}
}
```

查询某个索引

输入命令行：

```
curl -XGET 'http://10.0.0.2:9200/china/city/_search?pretty' -H 'Content-Type: application/json'
```

响应如下：

```
{
  "took" : 0,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 6,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "china",
        "_type" : "city",
        "_id" : "guangzhou",
        "_score" : 1.0,
        "_source" : {
          "name" : "广州市",
          "province" : "广东省越秀区吉祥路79号",
          "lat" : 23.1317146641,
          "lon" : 113.2595185241,
          "x" : 6173,
          "level.range" : 4,
          "level.level" : 1,
          "level.name" : "一线城市",
          "y" : 2560,
          "cityNo" : 3
        }
      }
    ]
  },
}
```

```
.....  
}
```

复杂查询

模拟 SQL：

```
select * from city where level.level=2  
curl -XGET http://10.0.0.2:9200/china/city/_search?pretty -H 'Content-Type: application/json' -d'  
{  
  "query" : {  
    "constant_score" : {  
      "filter" : {  
        "term" : {  
          "level.level" : 2  
        }  
      }  
    }  
  }  
'
```

响应如下：

```
{  
  "took" : 2,  
  "timed_out" : false,  
  "_shards" : {  
    "total" : 5,  
    "successful" : 5,  
    "skipped" : 0,  
    "failed" : 0  
  },  
  "hits" : {  
    "total" : 2,  
    "max_score" : 1.0,  
    "hits" : [  
      {  
        "_index" : "china",  
        "_type" : "city",  
        "_id" : "chengdu",  
        "_score" : 1.0,  
        "_source" : {  
          "name" : "成都市",  
          "province" : "四川省锦江区红星路4段-88号-附1号",  
          "lat" : 30.6522796787,  
          "lon" : 104.0669642777,  
          "level.level" : 2  
        }  
      }  
    ]  
  }  
}
```

```
"lon" : 104.0725574128,
"x" : 4387,
"level.level" : 2,
"level.range" : 19,
"level.name" : "新一线城市",
"y" : 4304,
"cityNo" : 5
},
{
  "_index" : "china",
  "_type" : "city",
  "_id" : "hangzhou",
  "_score" : 1.0,
  "_source" : {
    "name" : "杭州市",
    "province" : "浙江省拱墅区环城北路316号",
    "lat" : 30.2753694112,
    "lon" : 120.1509063337,
    "x" : 7530,
    "level.level" : 2,
    "level.range" : 19,
    "level.name" : "新一线城市",
    "y" : 4182,
    "cityNo" : 6
  }
}
]
```

聚合查询

模拟 SQL :

```
select level.level, count(1) from city group by level.level
curl -XGET http://10.0.0.2:9200/china/city/_search?pretty -H 'Content-Type: application/json' -d'
{
  "size" : 0,
  "aggs" : {
    "city_level" : {
      "terms" : {
        "field" : "level.level"
      }
    }
  }
}
```

```
}  
}'
```

响应如下：

```
{  
  "took" : 10,  
  "timed_out" : false,  
  "_shards" : {  
    "total" : 5,  
    "successful" : 5,  
    "skipped" : 0,  
    "failed" : 0  
  },  
  "hits" : {  
    "total" : 7,  
    "max_score" : 0.0,  
    "hits" : [ ]  
  },  
  "aggregations" : {  
    "city_level" : {  
      "doc_count_error_upper_bound" : 0,  
      "sum_other_doc_count" : 0,  
      "buckets" : [  
        {  
          "key" : 1,  
          "doc_count" : 4  
        },  
        {  
          "key" : 2,  
          "doc_count" : 3  
        }  
      ]  
    }  
  }  
}
```

删除文档

删除单个文档

输入命令行：

```
curl -XDELETE 'http://10.0.0.2:9200/china/city/beijing?pretty' -H 'Content-Type: application/json'
```

响应如下：

```
{
  "found" : true,
  "_index" : "china",
  "_type" : "city",
  "_id" : "beijing",
  "_version" : 5,
  "result" : "deleted",
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  }
}
```

删除类型

```
curl -XDELETE 'http://10.0.0.2:9200/china/city?pretty' -H 'Content-Type: application/json'
```

删除索引

```
curl -XDELETE 'http://10.0.0.2:9200/china?pretty' -H 'Content-Type: application/json'
```

通过 Kibana 访问集群

最近更新时间：2021-07-20 16:53:35

腾讯云 ES 包含 Kibana 模块，用户可以访问集群对应的 Kibana 页面，进行数据的可视化查询分析，以及数据的管理操作，您可以通过本教程开始 Kibana 的快速入门。

访问 Kibana 页面

入口

Kibana 页面有两个入口，分别位于集群列表页和集群详情页，详情如下图。单击对应的入口，会跳转到 Kibana 登录页面。

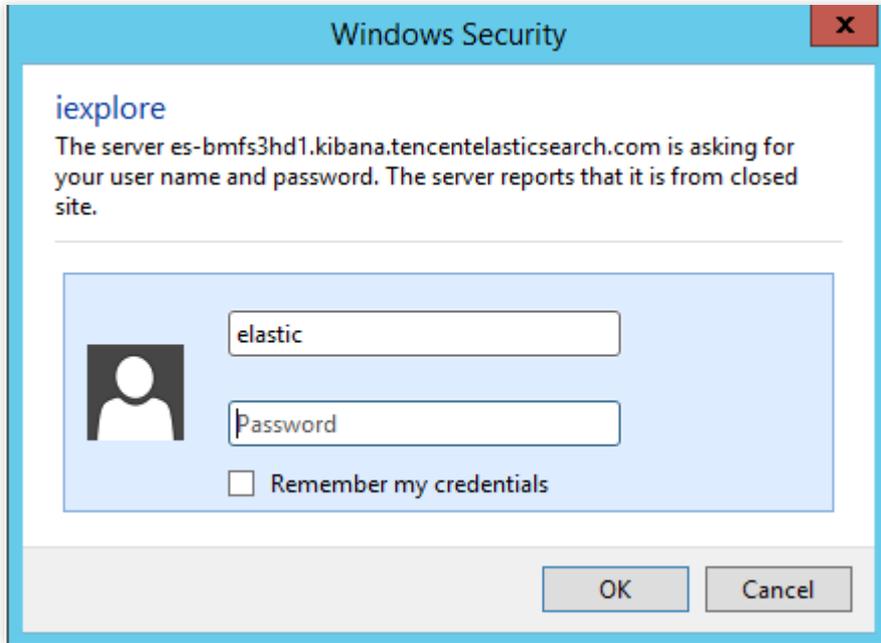
默认情况下，Kibana 通过公网地址访问，如果您担心通过公网访问 Kibana 会造成安全问题，我们也支持在集群详情页关闭 Kibana 公网地址，开启 Kibana 内网地址来进行访问。

The screenshot displays the Tencent Cloud Elasticsearch Service console. The top section is the 'Cluster List' page, which includes a navigation bar with various regions (Guangzhou, Shanghai, Beijing, etc.) and a search bar. Below the navigation bar is a table with columns: ID/Name, Status, Node Specs, Nodes, Health Status, Availability Zo..., Network, ES Version, Billing Type, and Operation. A red box highlights the 'Kibana' link in the 'Operation' column of the first cluster row. The bottom section shows the 'Cluster Configuration' page for a specific cluster, with a red box highlighting the 'Kibana' link in the top navigation bar. The 'Cluster Configuration' page is divided into two panels: 'Basic Info' and 'Cluster Configuration'. The 'Basic Info' panel shows fields like Name, Cluster Configuration Name, ID, Private Network Address, Cluster Status (Normal), and Health Status (Green). The 'Cluster Configuration' panel shows fields like Node Model, Node Storage Type, Node Storage, Nodes, Total Cluster Storage, and Dedicated Master Node.

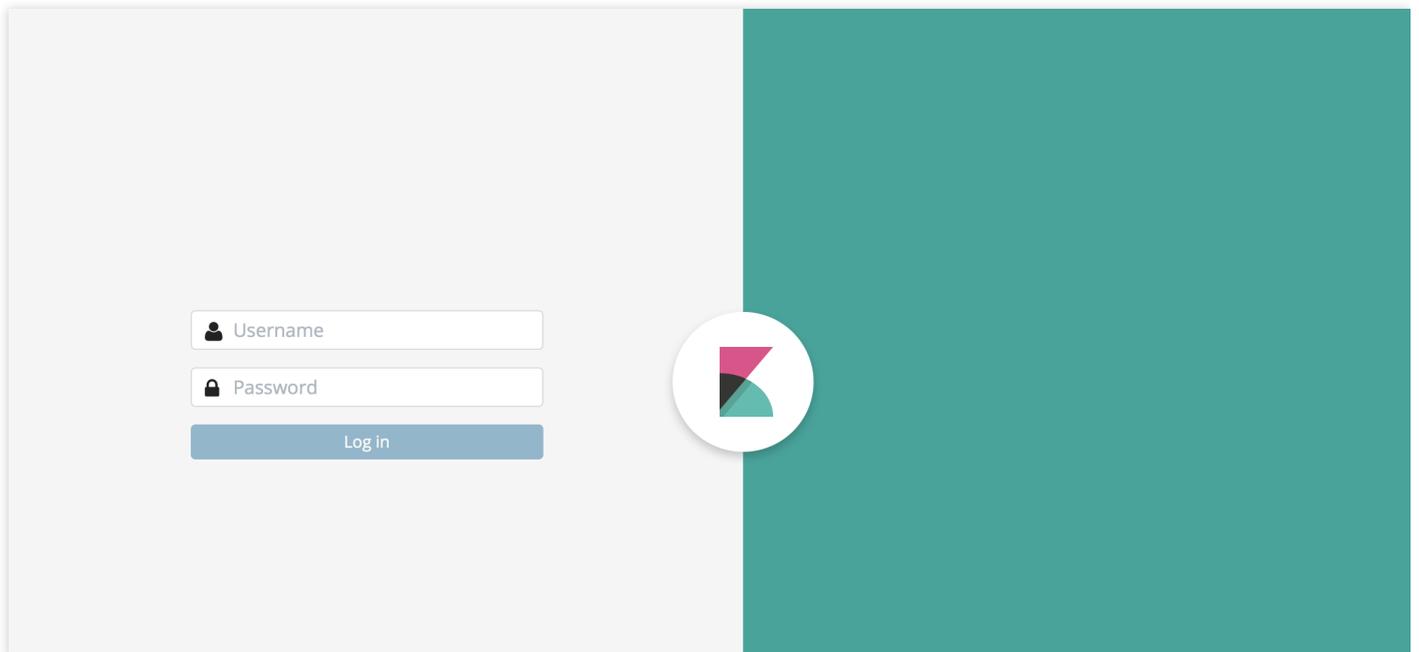
登录

Kibana 页面访问需要登录，帐号为 elastic，密码为用户创建集群时设置的 Kibana 密码。如果忘记密码，可以在集群详情页重置密码。出于安全考虑，用户可以配置 Kibana 公网地址的访问黑白名单来提高安全防护，详见 [Kibana 访问设置](#)。

- 若“ES 集群用户登录认证”未开启，Kibana 登录页如下图所示：

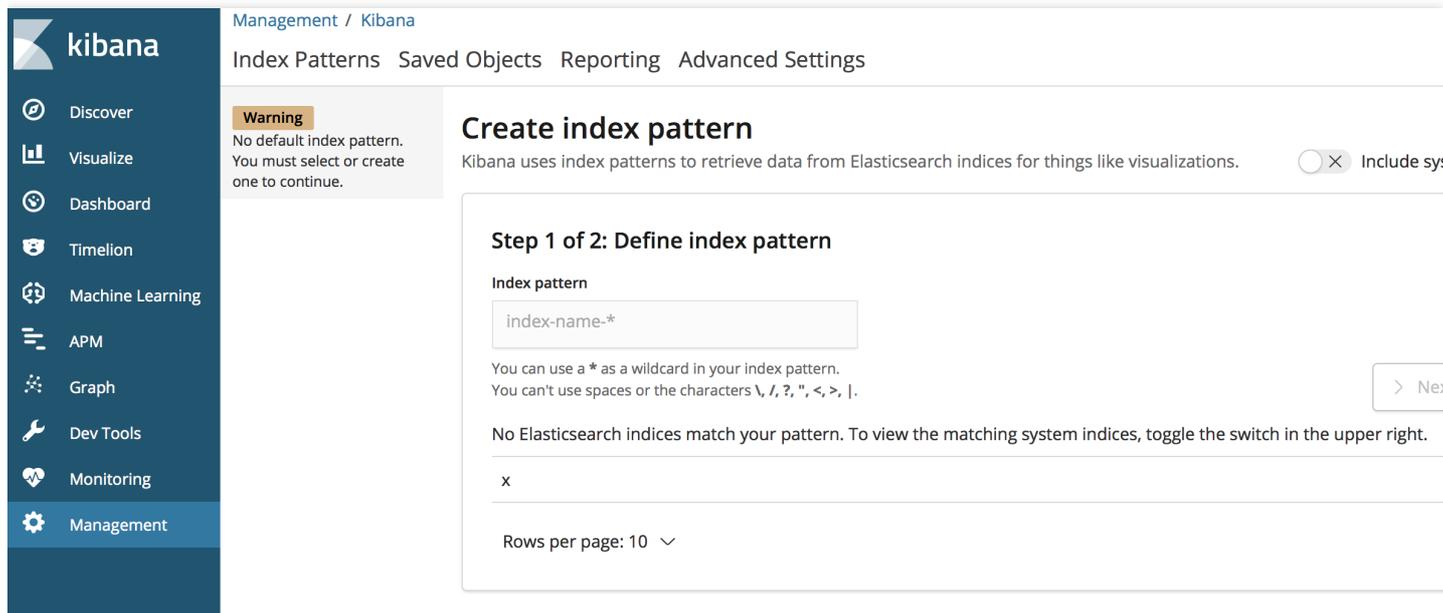


- 若“ES 集群用户登录认证”已开启，Kibana 登录页如下图所示：



访问

登录 Kibana 页面后，如果用户是第一次使用，集群尚未存入用户自定义的索引数据，页面会提示用户配置索引，具体参考 [索引添加及访问](#)。



The screenshot shows the Kibana Management interface for creating an index pattern. The left sidebar contains navigation options: Discover, Visualize, Dashboard, Timelion, Machine Learning, APM, Graph, Dev Tools, Monitoring, and Management (selected). The main content area is titled 'Create index pattern' and includes a warning message: 'Warning: No default index pattern. You must select or create one to continue.' Below the warning is a text input field containing 'index-name-*'. A message states: 'No Elasticsearch indices match your pattern. To view the matching system indices, toggle the switch in the upper right.' There is a toggle switch labeled 'Include system indices' which is currently turned off. At the bottom, there is a 'Rows per page: 10' dropdown menu.

索引添加及访问（存储数据）

在 Kibana 页面左侧菜单，单击【Dev Tools】进入开发工具页面，用户可以通过控制台，向集群发送各种操作请求。下面将通过城市信息的数据存储操作的示例，演示如何通过控制台操作集群和存储数据。

添加索引

定义索引的 mapping

索引名称为 china，类型名称为 city，以及详细的字段及类型信息。其中字段 location 的类型是 geo_point，可以表示地理位置信息；level 是对象类型，包含二级字段信息。关于字段类型说明，可查看官方文档 [Field Datatypes](#)。

kibana
Dev Tools History Settings

- Discover
- Visualize
- Dashboard
- Timelion
- Machine Learning
- APM
- Graph
- Dev Tools
- Monitoring
- Management

Console
Search Profiler
Grok Debugger

```

1 PUT china
2 {
3   "mappings": {
4     "city": {
5       "properties": {
6         "name": { "type": "keyword" },
7         "province": { "type": "keyword" },
8         "location": { "type": "geo_point" },
9         "x": { "type": "integer" },
10        "level": {
11          "properties": {
12            "level": { "type": "integer" },
13            "range": { "type": "integer" },
14            "name": { "type": "keyword" }
15          }
16        },
17        "y": { "type": "integer" },
18        "cityNo": { "type": "integer" }
19      }
20    }
21  }
22 }
23
```

```

1  #! Deprecation: the default number of shards will change from
2  [5] to [1] in 7.0.0; if you wish to continue using the defau
3  of [5] shards, you must manage this on the create index
4  request or with an index template
5  #! Deprecation: [_all] is deprecated in 6.0+ and will be remov
6  in 7.0. As a replacement, you can use [copy_to] on mapping
7  fields to create your own catch all field.
8  #! Deprecation: [_default_] mapping is deprecated since it is
9  not useful anymore now that indexes cannot have more than on
10 type
11 {
12   "acknowledged": true,
13   "shards_acknowledged": true,
14   "index": "china"
15 }
```

```

PUT china
{
  "mappings": {
    "city": {
      "properties": {
        "name": { "type": "keyword" },
        "province": { "type": "keyword" },
        "location": { "type": "geo_point" },
        "x": { "type": "integer" },
        "level": {
          "properties": {
            "level": { "type": "integer" },
            "range": { "type": "integer" },
            "name": { "type": "keyword" }
          }
        },
        "y": { "type": "integer" },
        "cityNo": { "type": "integer" }
      }
    }
  }
}

```

添加单个文档

Console Search Profiler Grok Debugger

```

1 PUT china/city/wuhan
2 {"name":"Wuhan","province":"No. 188 Yanjiang Road, Jiang'an District, Hubei Province","location":{"lat":30.5952548577,"lon":114.2999398195},"x":6384,"level":{"level":2,"range":19,"name":"new first-tier city"},"y":4231,"cityNo":7}

```

```

1 {
2   "_index" : "china",
3   "_type" : "city",
4   "_id" : "wuhan",
5   "_version" : 4,
6   "result" : "updated",
7   "_shards" : {
8     "total" : 2,
9     "successful" : 2,
10    "failed" : 0
11  },
12  "_seq_no" : 4,
13  "_primary_term" : 1
14 }
15

```

```

PUT china/city/wuhan
{"name":"武汉市","province":"湖北省江岸区沿江大道188号","location":{"lat":30.5952548577,"lon":114.2999398195},"x":6384,"level":{"level":2,"range":19,"name":"新一线城市"},"y":4231,"cityNo":7}

```

查询单个文档

```
GET /china/city/wuhan
```

添加多个文档

```

POST _bulk
{ "index" : { "_index": "china", "_type": "city", "_id": "beijing" } }
{"name":"北京市","province":"北京市","location":{"lat":39.9031324643,"lon":116.4010433787},"x":6763,"level":{"range":4,"level":1,"name":"一线城市"},"y":6381,"cityNo":1}
{ "index" : { "_index": "china", "_type": "city", "_id": "shanghai" } }
{"name":"上海市","province":"上海市","location":{"lat":31.2319526784,"lon":121.469443249},"x":7779,"level":{"range":4,"level":1,"name":"一线城市"},"y":4409,"cityNo":2}
{ "index" : { "_index": "china", "_type": "city", "_id": "guangzhou" } }
{"name":"广州市","province":"广东省越秀区吉祥路79号","location":{"lat":23.1317146641,"lon":113.2595185241},"x":6173,"level":{"range":4,"level":1,"name":"一线城市"},"y":2560,"cityNo":3}
{ "index" : { "_index": "china", "_type": "city", "_id": "shenzhen" } }
{"name":"深圳市","province":"广东省福田区新园路37号","location":{"lat":22.5455465546,"lon":114.0527779134},"x":6336,"level":{"range":4,"level":1,"name":"一线城市"},"y":2429,"cityNo":4}
{ "index" : { "_index": "china", "_type": "city", "_id": "chengdu" } }
{"name":"成都市","province":"四川省锦江区红星路4段-88号-附1号","location":{"lat":30.6522796787,"lon":104.0725574128},"x":4387,"level":{"level":2,"range":19,"name":"新一线城市"},"y":4304,"cityNo":5}
{ "index" : { "_index": "china", "_type": "city", "_id": "hangzhou" } }

```

```
{ "name": "杭州市", "province": "浙江省拱墅区环城北路316号", "location": { "lat": 30.2753694112, "lon": 120.1509063337 }, "x": 7530, "level": { "level": 2, "range": 19, "name": "新一线城市" }, "y": 4182, "cityNo": 6 }
```

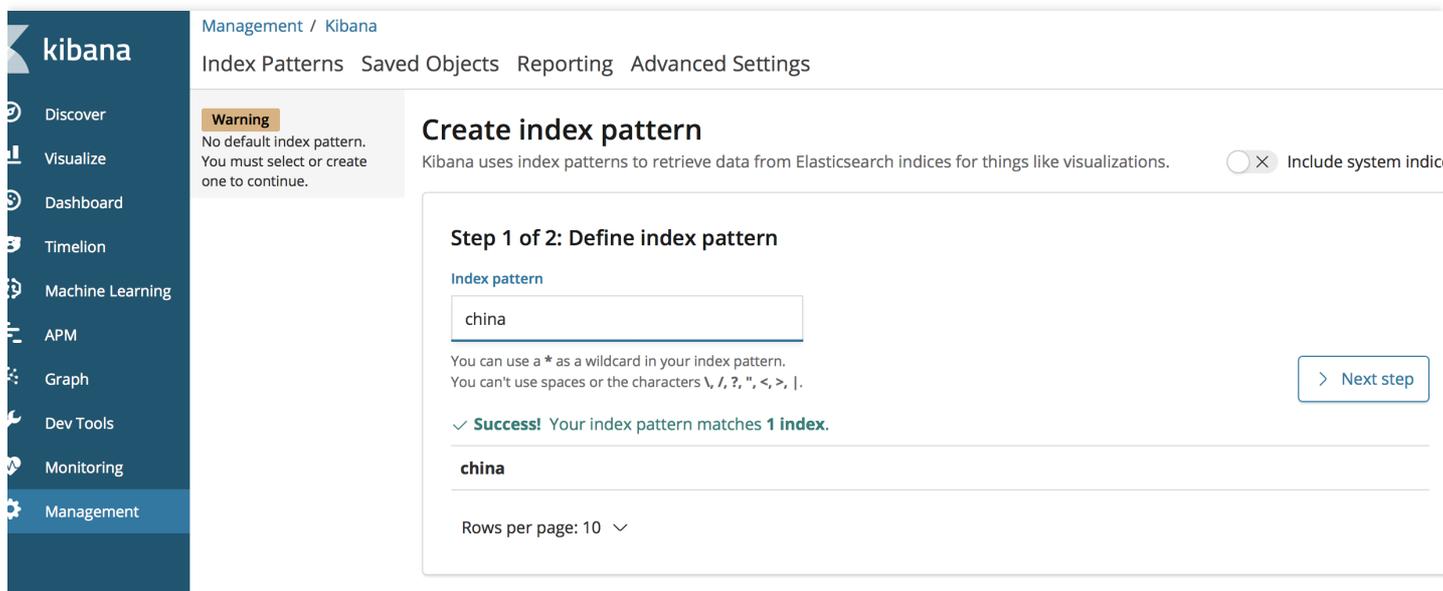
查询多个文档：

```
GET /china/city/_search
```

访问索引

配置 Kibana 访问索引

为了使用 Kibana，需要配置至少一个可以匹配到的索引。输入上文创建的索引 china，单击【Next step】进入下一步。



The screenshot shows the Kibana Management interface for creating an index pattern. The left sidebar contains navigation options like Discover, Visualize, Dashboard, etc. The main content area is titled 'Create index pattern' and includes a warning message: 'Warning: No default index pattern. You must select or create one to continue.' Below this, there is a text input field containing 'china'. A success message states: 'Success! Your index pattern matches 1 index.' A 'Next step' button is visible on the right. The interface also shows 'Include system indices' as an unchecked option and 'Rows per page: 10' at the bottom.

配置时间过滤字段用于通过时间过滤索引中的数据，如果索引中没有表示时间的字段，可以选择不使用时间过滤功能。单击【Create index pattern】创建索引模式。

Management / Kibana

Index Patterns Saved Objects Reporting Advanced Settings

★ kibana_sample_data_flig...
bellen*
filebeat-6.4.3-*
nginx_access-*

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations. Include system indices

Step 2 of 2: Configure settings

You've defined **bellen-*** as your index pattern. Now you can specify some settings before we create it.

Time Filter field name [Refresh](#)

I don't want to use the Time Filter

The Time Filter will use this field to filter your data by time. You can choose not to have a time field, but you will not be able to narrow down your data by a time range.

[Show advanced options](#)

[Back](#) [Create index pattern](#)

查看索引对应的字段。

Management / Kibana

Index Patterns Saved Objects Reporting Advanced Settings

+ Create Index Pattern

★ china

★ china

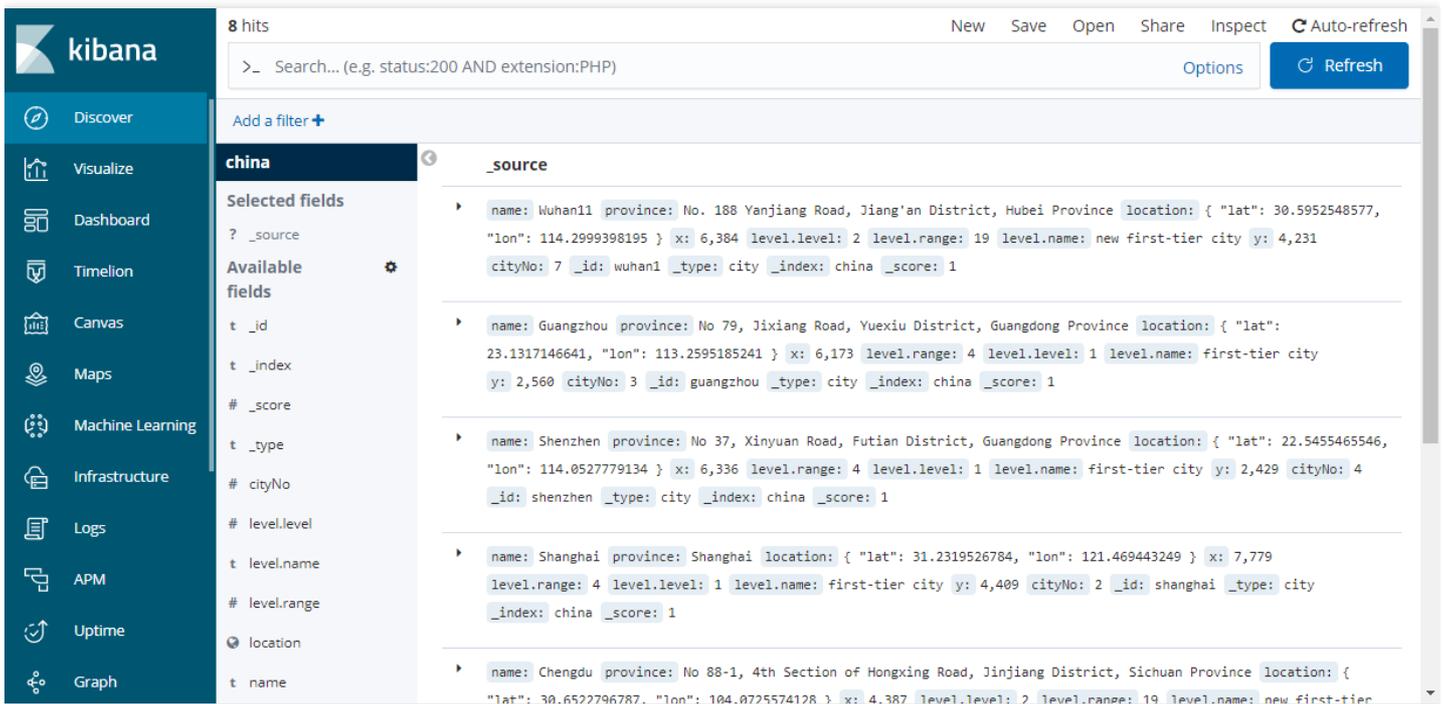
This page lists every field in the **china** index and the field's associated core type as recorded by Elasticsearch. To change a field type the Elasticsearch [Mapping API](#)

Fields (14) Scripted fields (0) Source filters (0)

Q Filter All field type

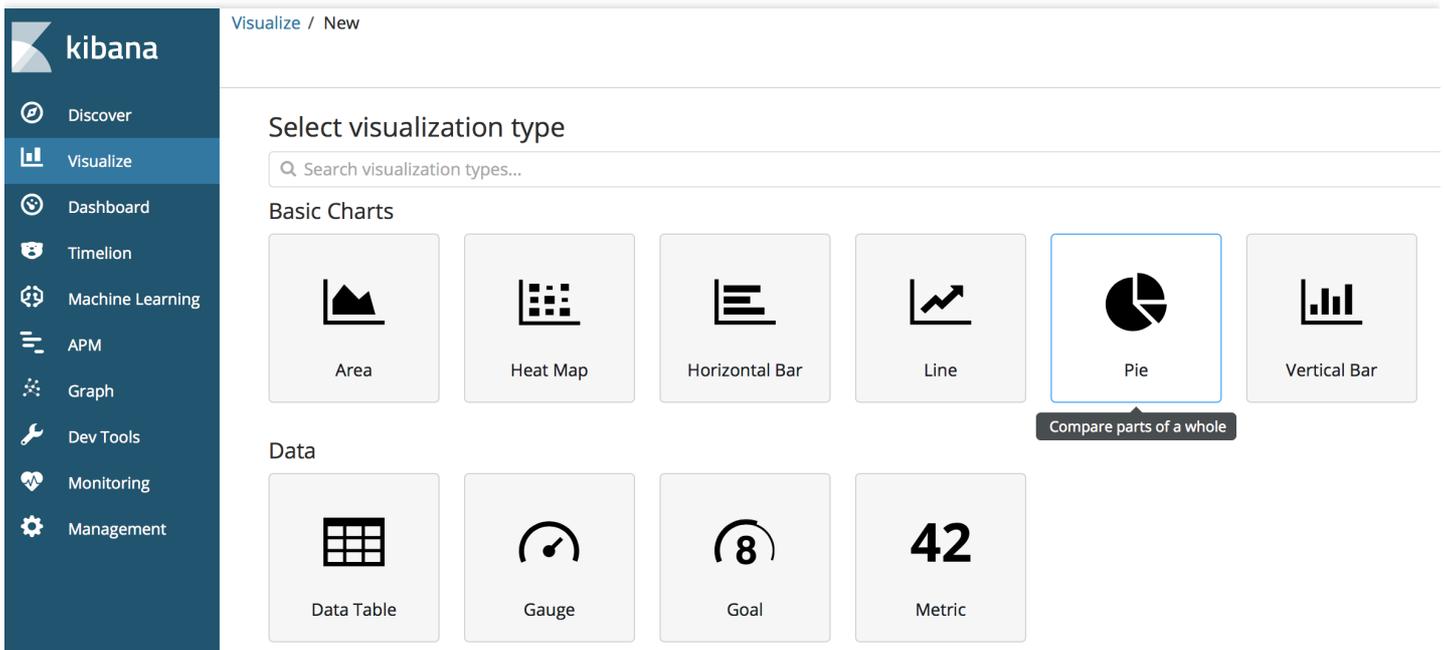
Name	Type	Format	Searchable	Aggregatable	Excluded
_id	string		●	●	
_index	string		●	●	
_score	number				
_source	_source				
_type	string		●	●	

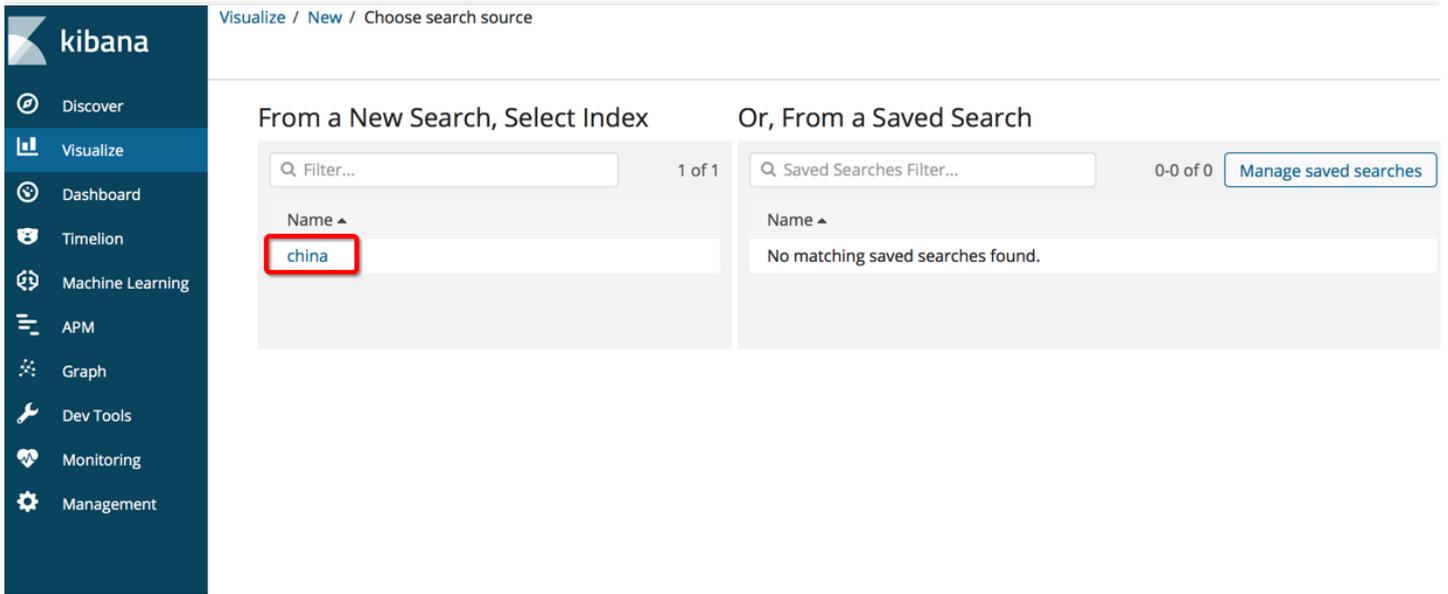
单击左侧菜单【Discover】，查看该索引下已经添加的文档。



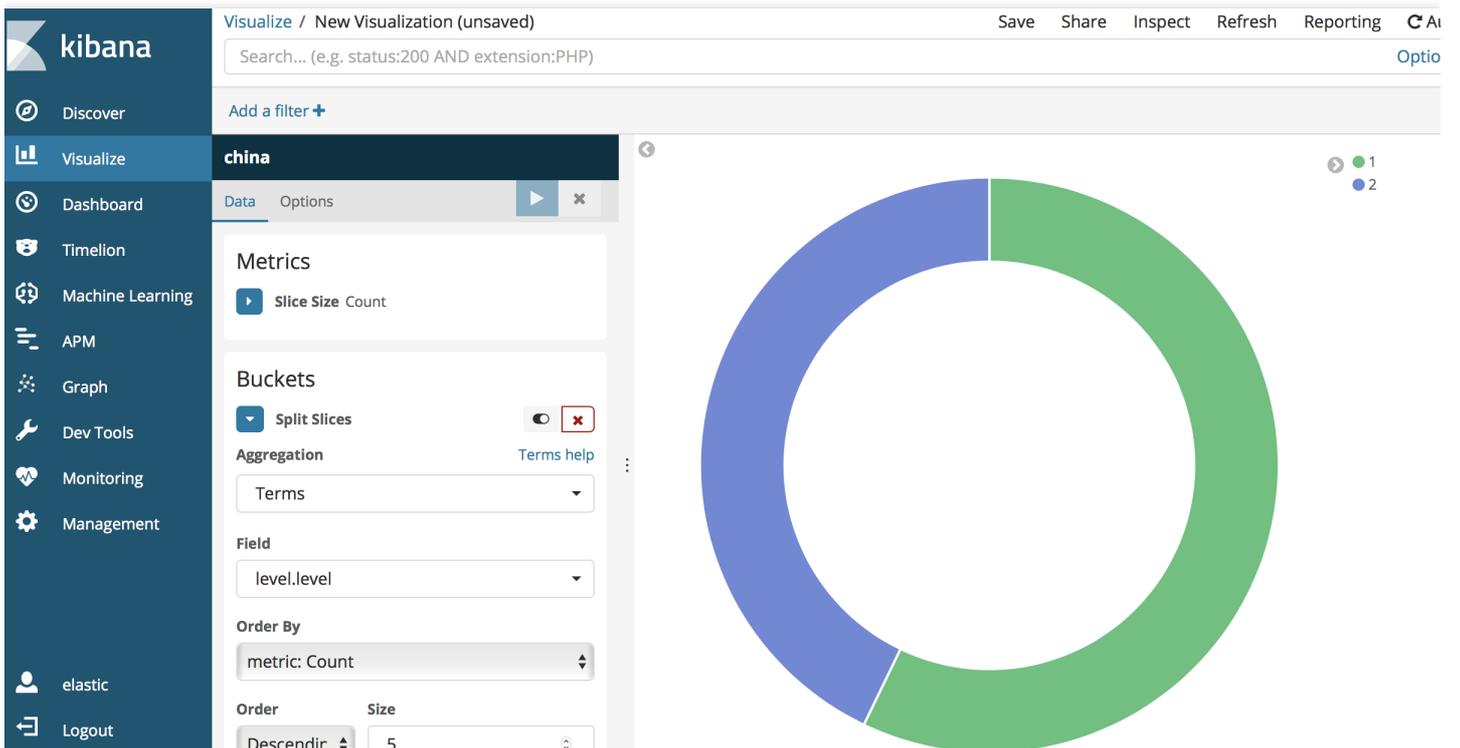
可视化查询分析

Kibana 拥有可视化统计分析数据的能力，单击左侧菜单【Visualize】，可以配置各种可视化的图表进行数据的分析。例如：要统计上文中，china 索引下的不同等级。





配置指标是 count，按字段 level.level 进行分组聚合统计，单击【Save】保存。



Kibana 其他使用方式，可查阅 [Kibana 官方文档](#)。