Elasticsearch Service

ES 集群指南

©2013–2024 Tencent Cloud. All rights reserved.

Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

The complete copyright of this document, including all text, data, images, and other content, is solely and exclusively owned by Tencent Cloud Computing (Beijing) Co., Ltd. ("Tencent Cloud"); Without prior explicit written permission from Tencent Cloud, no entity shall reproduce, modify, use, plagiarize, or disseminate the entire or partial content of this document in any form. Such actions constitute an infringement of Tencent Cloud's copyright, and Tencent Cloud will take legal measures to pursue liability under the applicable laws.

Trademark Notice

Tencent Cloud

This trademark and its related service trademarks are owned by Tencent Cloud Computing (Beijing) Co., Ltd. and its affiliated companies("Tencent Cloud"). The trademarks of third parties mentioned in this document are the property of their respective owners under the applicable laws. Without the written permission of Tencent Cloud and the relevant trademark rights owners, no entity shall use, reproduce, modify, disseminate, or copy the trademarks as mentioned above in any way. Any such actions will constitute an infringement of Tencent Cloud's and the relevant owners' trademark rights, and Tencent Cloud will take legal measures to pursue liability under the applicable laws.

Service Notice

This document provides an overview of the as-is details of Tencent Cloud's products and services in their entirety or part. The descriptions of certain products and services may be subject to adjustments from time to time.

The commercial contract concluded by you and Tencent Cloud will provide the specific types of Tencent Cloud products and services you purchase and the service standards. Unless otherwise agreed upon by both parties, Tencent Cloud does not make any explicit or implied commitments or warranties regarding the content of this document.

Contact Us

We are committed to providing personalized pre-sales consultation and technical after-sale support. Don't hesitate to contact us at 4009100100 or 95716 for any inquiries or concerns.

Contents

©2013–2024 Tencent Cloud. All rights reserved.

ES 集群指南
访问控制
CAM 访问控制

Last updated：2024-09-04 15:14:01

ES CAM 简介

访问管理（Cloud Access Management，CAM）是腾讯云提供的 Web 服务，主要用于帮助用户安全管理腾讯云账户下的资源的访问权限。用户可以通过 CAM 创建、管理和销毁用户(组)，并使用身份管理和策略管理控制其他用户使用腾讯云资源的权限，CAM 策略的详细信息及使用请参见 CAM 策略 。

ES CAM 策略

通用权限策略

ES 默认提供了两种通用策略：

- 全读写策略 QcloudElasticsearchServiceFullAccess，可以让用户拥有创建和管理 ES 所有集群实例的权限。
- 只读策略 QcloudElasticsearchServiceReadOnlyAccess，可以让用户拥有查看 ES 集群实例的权限，但是不具有创建、更新、删除等操作的权限。

您可以登录 访问管理控制台 > 策略，在搜索框中搜索“Elasticsearch”，在列表中会显示默认策略，可对需要授权的账号进行绑定。

如果默认策略不能满足需求，可单击**新建自定义策略**，进行自定义授权。

自定义权限策略

ES 可授权的资源类型如下：

资源类型	资源描述
instance	`qcs::es:$region:$account:instance/*`

各 API 支持资源级权限访问控制详情如下：

接口名	描述	是否关联资源	资源描述
获取集群列表、单个集群信息	DescribeInstances	是	`qcs::es:${Region}:uin/${ownerUin}:instance/${instanceId}`
创建集群	CreateInstance	否	`*`
更新集群	UpdateInstance	是	`qcs::es:${Region}:uin/${ownerUin}:instance/${instanceId}`
重启集群	RestartInstance	是	`qcs::es:${Region}:uin/${ownerUin}:instance/${instanceId}`

©2013-2024 Tencent Cloud. All rights reserved.

删除集群	DeleteInstance	是	qcs::es:${Region}:uin/${ownerUin}:instance/${instanceId}
更新插件	UpdatePlugins	是	qcs::es:${Region}:uin/${ownerUin}:instance/${instanceId}

支持区域如下：

区域	名称	区域 ID
华南地区	广州	ap-guangzhou
华东地区	上海	ap-shanghai
	南京	ap-nanjing
华北地区	北京	ap-beijing
西南地区	成都	ap-chengdu
	重庆	ap-chongqing
港澳台地区	中国香港	ap-hongkong
亚太东南	新加坡	ap-singapore
亚太南部	孟买	ap-mumbai
亚太东北	首尔	ap-seoul
	东京	ap-tokyo
美国西部	硅谷	na-siliconvalley
美国东部	弗吉尼亚	na-ashburn
欧洲地区	法兰克福	eu-frankfurt

自定义策略的语法如下：

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
                "Action"
            ],
            "resource": "Resource",
            "effect": "Effect"
        }
    ]
}
```

- Action：替换成要允许或拒绝的操作。
- Resource：替换成要授权的具体资源。
- Effect：替换成允许或拒绝。

ES 目前支持除了 DescribeInstances 接口之外的其他操作接口来访问控制管理，您可以将账号下某个集群的更新、重启、删除等操作赋予某个子账号进行管理。

自定义权限示例

授予某账号指定集群更新操作权限，策略语法如下：

©2013-2024 Tencent Cloud. All rights reserved.

```
{
    "version": "2.0",
    "statement": [
         {
            "action": [
                "es:Describe*"
            ],
            "resource": [
               "qcs::es:ap-guangzhou:uin/$uin:instance/$instanceID"
            ],
            "effect": "allow"
        },
        {
            "action": [
                "vpc:Describe*",
                "vpc:Inquiry*",
                "vpc:Get*"
            ],
            "resource": "*",
            "effect": "allow"
        },
        {
            "action": [
                "monitor:*",
                "cam:ListUsersForGroup",
                "cam:ListGroups",
                "cam:GetGroup"
            ],
            "resource": "*",
            "effect": "allow"
        },
        {
            "action": [
                "es:Update*"
            ],
            "resource": [
                "qcs::es:ap-guangzhou:uin/$uin:instance/$instanceID"
            ],
            "effect": "allow"
        }
    ]
}
```

©2013–2024 Tencent Cloud. All rights reserved.

ES 集群访问控制

Last updated：2024-10-15 21:58:41

ES 集群部署在逻辑隔离的私有网络 VPC 中，我们提供了丰富的能力来切实保证云上资源的安全性，包括：

- 对腾讯云账户下资源的 CAM 访问管理（参见 CAM 访问控制配置）
- ES 集群访问密码/ES 集群用户登录认证
- 设置 Kibana 和 Cerebro 外网访问 IP 黑白名单，或限制 Kibana 和 Cerebro 仅能通过内网访问。
- 对 ES 集群有限开启外网访问和设置 IP 白名单
- 提供的基于角色的访问控制（RBAC）

设置 ES 集群访问密码

在创建腾讯云 ES 集群时，会要求用户设置默认用户 elastic 的密码，该账号和密码用于 Kibana 页面登录，若集群已开启 ES 集群用户登录认证，则此用户名和密码还会用于 ES 集群的登录认证，提供进一步的安全防护，详情如下：

重置 ES 集群访问密码

用户需要调整 ES 集群访问密码时，可以通过集群详情页的密码重置功能对 ES 集群 elastic 账号的密码进行重置，操作页面如下：

设置 Kibana 公网访问 IP 黑白名单

由于 Kibana 页面是默认通过公网访问，在进行密码校验的基础上，ES 还为 Kibana 访问提供了 IP 黑白名单功能，进一步保障用户集群的访问安全性。Kibana 公网访问白名单默认为127.0.0.1，表示不允许所有 IPv4 和 IPv6 地址访问，此时单击 kibana 访问入口时，会弹窗引导用户去进行白名单设置，如下图。

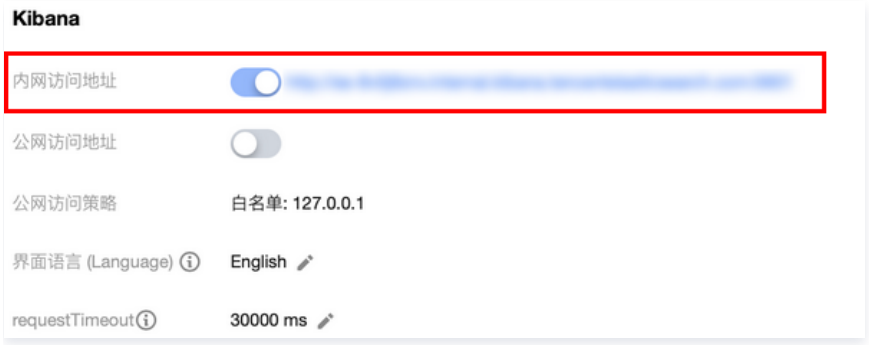
©2013-2024 Tencent Cloud. All rights reserved.

在集群的**可视化配置**界面可以设置 Kibana 公网访问 IP 黑白名单：

- 配置规则：支持多个 IP，IP 之间以英文逗号分隔，格式为192.168.0.1,192.168.0.0/24，最多支持50个。
- 黑白名单设置：客户可以设置任意一个，如果黑白名单都配置，以白名单为准。

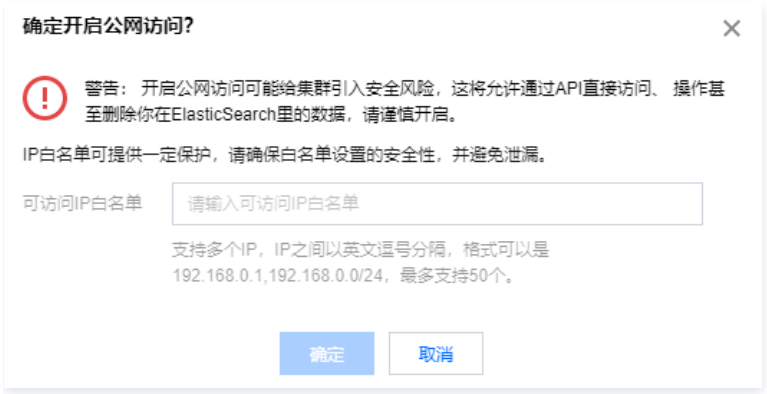
限制 Kibana 仅能通过内网访问

如果用户担心公网访问安全性，也可以关闭外网访问，设置仅允许内网访问。



开启 ES 集群公网访问

基于安全考虑，ES 集群外网访问是默认关闭的，对于已开启 ES 集群用户登录认证 的集群，允许用户基于使用便捷性的需要开启外网访问，但必须同时设置 IP 白名单以提供安全防护。



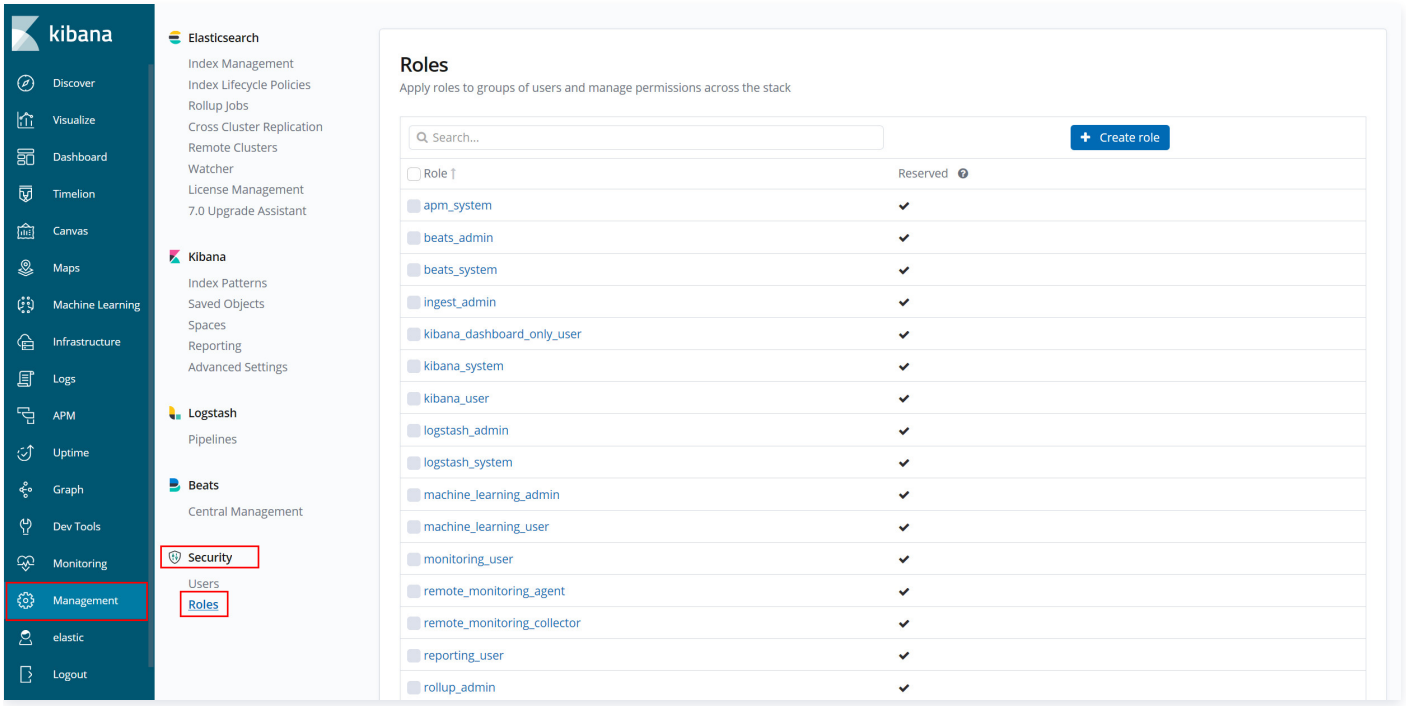
基于角色的访问控制（RBAC）

©2013–2024 Tencent Cloud. All rights reserved.

对于已开启 ES 集群用户登录认证 的集群，用户将获得更多安全管理功能。白金版还可以进一步支持基于文档、字段级别的细粒度访问控制，详情请参考 Elastic 官网文档 基于角色的访问控制 。

角色管理

用户可以在 Kibana 的 **Management > Security > Roles** 中创建、修改和删除具有不同权限组合的角色，详情如下：

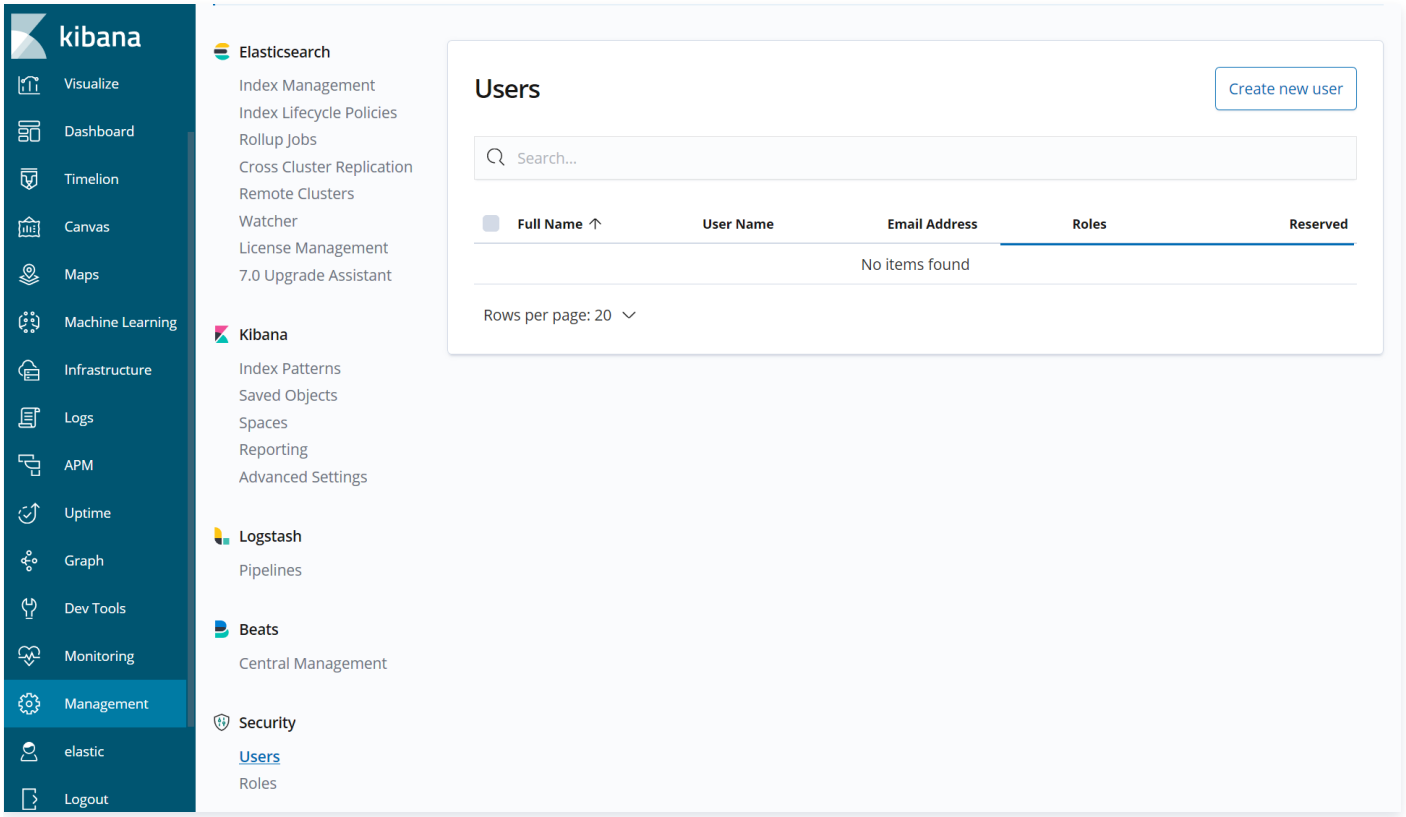


用户管理

用户可以在 Kibana 的 **Management > Security > Users** 中创建、修改（信息修改、密码修改等）和删除具有多个角色的用户，详情如下：

> **注意**
> ES 内置用户 elastic 的密码只能在官网控制台进行重置。

©2013–2024 Tencent Cloud. All rights reserved.



更多相关安全功能使用可以参考：

- 保护您在 Elastic Stack 中的数据
- Kibana XPACK SECURITY
- ES SECURITY API

©2013–2024 Tencent Cloud. All rights reserved.

LDAP 身份验证

Last updated：2024-10-15 21:58:41

本文介绍如何基于腾讯云 Elasticsearch Service 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 用户访问腾讯云 Elasticsearch Service。

使用限制

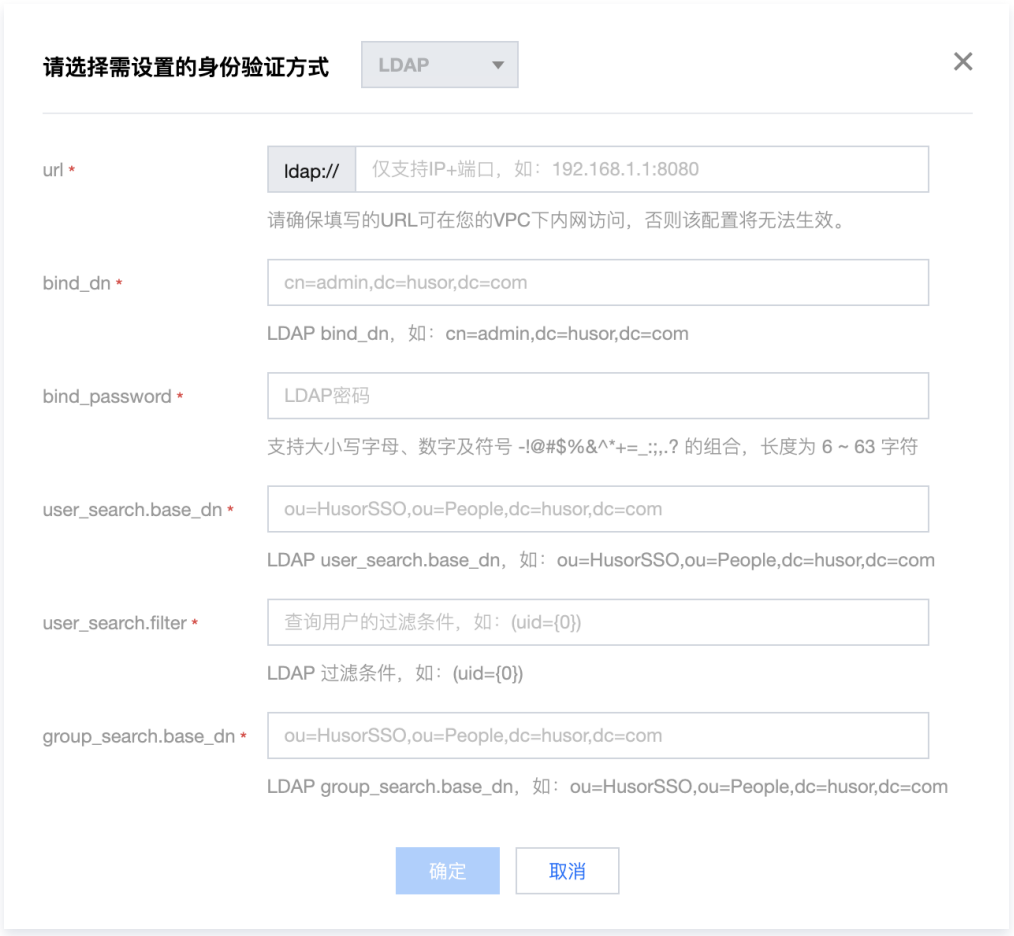
- LDAP 身份验证是 Elasticsearch 官方商业特性 X-pack 提供的高级功能，**当前仅在白金版集群支持**。其他版本集群如需使用，请先升级至白金版。
- 由于网络架构原因，2020年5月20号之前创建的集群无法访问外部服务，不支持使用 LDAP。如需开启 LDAP 身份验证服务，可重新 创建集群 。

设置 LDAP 身份验证

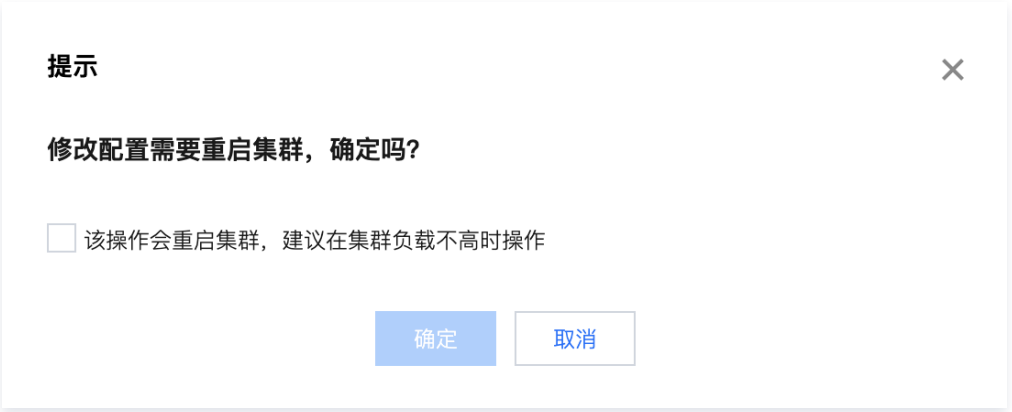
1. 登录腾讯云 Elasticsearch 控制台，单击**集群名称**访问目标集群，跳转至**基础配置**页面。
2. 在**访问控制**模块，单击**身份验证**的编辑按钮，进入身份验证设置界面。

3. 填写相关内容
   - url：LDAP 服务器地址。ldap 服务器地址，以 “ldap://” 开头，后面填写域名或者IP地址。请确保填写的 URL 可在您的 VPC 下内网访问，否则该配置将无法生效。
   - bind_dn：用于 LDAP 服务器认证的成员 DN。需满足 DN 层次型语法结构，如：cn=admin,dc=husor,dc=com，长度不超过200个字符。
   - bind_password：LDAP 服务器连接密码。支持大小写字母、数字及符号的组合，长度为6 - 63个字符。
   - user_search.base_dn：用于检索已绑定 LDAP 成员的基准 DN。需满足 DN 层次型语法结构，如：ou=HusorSSO,ou=People,dc=husor,dc=com，长度不超过200个字符。
   - user_search.filter：查询过滤条件，系统将过滤满足条件的绑定关系。如：(uid={0})。
   - group_search.base_dn：用于检索已绑定 LDAP 用户组的基准 DN。需满足 DN 层次型语法结构，如：ou=HusorSSO,ou=People,dc=husor,dc=com，长度不超过200个字符。

©2013-2024 Tencent Cloud. All rights reserved.



4. 确认填写内容无误后，单击**确定**，在弹出的对话框中，阅读注意事项，确认后，集群将会重启，可以在集群变更记录中查看变更进度。



©2013–2024 Tencent Cloud. All rights reserved.

5. 设置成功后，您将会在**身份验证**看到**LDAP 已开启**，如需修改LDAP身份验证设置，单击**LDAP 已开启**即可进行编辑。

配置 LDAP 用户角色权限

1. 设置了 LDAP 身份验证后，LDAP 用户还没有被分配任何权限，无法使用 LDAP 方式访问 ES 集群/Kibana，需要在 Kibana 中对 LDAP 用户进行角色映射。
2. 登录腾讯云 Elasticsearch 控制台，跳转进入集群的 Kibana 主页，进入 Kibana 的 **Dev Tools** 页面。
3. 参考官方文档 role mapping API，创建 LDAP 用户与角色的映射关系。例如下面的示例，为 LDAP 用户 **zhangsan** 赋予了 **superuser** 角色的权限。

```
POST _xpack/security/role_mapping/ldap_role_mapping_zhangsan?pretty
{
  "roles": [ "superuser" ],
  "enabled": true,
  "rules": {
    "any": [
      {
        "field": {
          "username": "zhangsan"
        }
      }
    ]
  }
}
```

关闭 LDAP 身份验证

1. 登录腾讯云 Elasticsearch 控制台，单击**集群名称**访问目标集群，跳转至**基础配置**页面。
2. 在**访问控制**模块，单击**身份验证**中的**关闭**按钮，在弹出的对话框中，阅读注意事项，确认后，LDAP 关闭操作开始，集群将会重启，可以在集群变更记录中查看变更进度。

注意事项

- 如果集群健康状态为 YELLOW 或 RED，进行修改配置操作有较大风险，需要先修复集群状态，再进行操作。
- 如果集群存在无副本索引，修改集群配置时会有强制重启的提示和选项框，此时进行修改重启操作有较大风险，可能会出现部分数据短暂无法访问的情况，建议为所有索引添加副本之后，再进行修改配置操作。

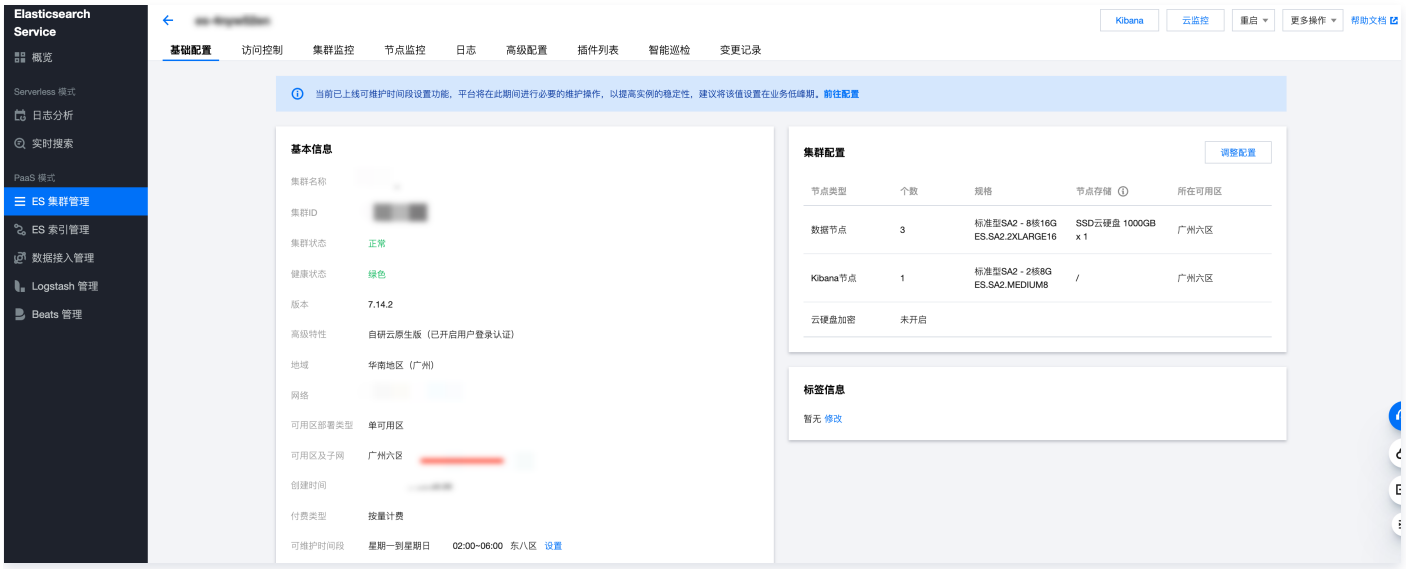
©2013–2024 Tencent Cloud. All rights reserved.

管理集群
集群状态

Last updated：2024-08-16 20:29:21

在集群列表页和集群详情页的基本信息中，均可查看集群的状态信息。

单击集群 ID/名称进入集群详情页。



集群状态是反映集群是否在变更中或正常使用的状态，包括正常、处理中等，具体含义如下：

状态	含义
正常	集群创建完成，且没有配置变更、集群重启等动作，可以正常访问和使用的状态
处理中	集群创建、集群配置变更、集群重启等操作，需要一定的时间进行处理的状态，期间部分服务访问会受到影响，如 Kibana、数据存储和查询等

其中健康状态是 ES 集群众多监控信息中非常重要的一个，用来表征集群总体上是否工作正常。健康状态种类如下：

颜色	健康状态
绿色（green）	所有的主分片和副本分片都正常运行
黄色（yellow）	所有的主分片都正常运行，但不是所有的副本分片都正常运行
红色（red）	有主分片没有正常运行

详情可查看 ES 集群健康状态 。

©2013-2024 Tencent Cloud. All rights reserved.

重启集群

Last updated：2024-10-12 21:50:11

操作场景

腾讯云 ES 为用户提供了集群重启功能，用户可以根据集群的情况使用此功能对集群进行重启操作。

注意事项

1. 重启集群为耗时操作，建议用户在非必要情况下不要进行该操作。
2. 重启过程会灰度重启各节点，因此用户应保证集群至少有1副本，否则会导致重启时集群变为 RED 状态，甚至有丢失数据的风险。
3. 当集群状态为 YELLOW 或 RED，或集群存在无副本索引等情况下，用户进行重启操作会提示需要进行强制重启。强制重启操作有较高风险，存在丢失数据的可能，在此种情况下建议用户先恢复集群状态为 GREEN，并为所有索引创建至少一个副本，然后再进行重启操作。若当前集群状态无法恢复，用户在了解强制重启风险的前提下仍要进行重启操作，可以勾选强制重启选项进行重启。

操作步骤

1. 登录 Elasticsearch Service 控制台 。
2. 重启集群有两个入口，分别位于集群列表页和集群详情页。
   - 在集群列表页，选择相应的集群，选择**操作 > 更多 > 重启**。
   - 单击集群名称进入集群详情页，在右上角单击**重启**，然后在下拉框中选择重启类型，包括集群重启、节点重启、Kibana 重启。
3. 正常集群在单击**重启**后出现如下界面，选择“操作类型”，然后单击**重启**，集群状态变为处理中，等待集群状态恢复为正常重启，操作即完成。

4. 当集群状态为 YELLOW 或 RED，或集群存在无副本索引的情况下，界面中会出现强制重启提示，例如下图只有勾选了强制重启才能对集群发起重启操作。在此种情况下用户可参考 注意事项 第三条中的描述进行操作。重启操作发起后，集群状态会变为处理中，等待集群状态恢复为正常，重启操作即完成。

©2013-2024 Tencent Cloud. All rights reserved.

销毁集群

Last updated：2024-10-16 11:20:11

操作场景

因为业务发展需要，当 ES 集群无法满足您的需求，需要退货时，您可以在控制台对集群进行销毁，避免服务继续运行而产生费用。如果是集群配置无法满足需求，您也可以通过调整集群配置把集群调整到合适的规格，操作方法请参见 调整配置 。

不同计费模式退费说明

不同计费模式下的集群，销毁集群的条件有所区别：

- 预付费包年包月的集群，如果集群还未到期，需要提前销毁时，会有条件限制：只支持1个集群的5天无理由退还和3个集群的普通自主退还，详细说明请参见 包年包月退费 。
- 后付费按量计费的集群，根据使用量计费，可以随时销毁集群，销毁后，就不再产生费用。

> **注意**
> 集群被销毁后，数据无法恢复，请谨慎操作。

操作步骤

1. 登录 Elasticsearch Service 控制台 。
2. 销毁集群有两个入口，分别位于集群列表页和集群详情页。

> **说明**
> 包年包月的集群销毁，需要满足一定的条件，详细说明请参见 包年包月退费 。

- 在集群列表页，选择相应的集群，在“操作”列选择**更多 > 销毁**。

©2013-2024 Tencent Cloud. All rights reserved.

- 单击集群名称进入集群详情页，在右上角选择**更多操作 > 销毁**。

3. 在“销毁集群”页面，单击**确定**，系统将清空集群数据，并回收资源，**数据清空后，无法恢复**。包年包月集群会把剩余未使用的费用以赠送金的形式退还到您的账户，折扣和代金券不予退还。

©2013–2024 Tencent Cloud. All rights reserved.

集群多可用区部署

Last updated：2024-10-08 14:18:21

创建支持多可用区的集群

1. 进入 腾讯云 Elasticsearch Service 购买页，选择“可用区部署模式”，设置多可用区网络。

> **说明**
> - 由于要开启多可用区容灾的集群，必须先开启专用主节点，且最小为三个，所以能支持多可用区容灾功能的地域必须最少支持三个机房。目前仅有部分大地域如北上广支持多可用区容灾的功能，其他暂不开放的地域随着腾讯云机房的建设，我们也会持续的加入这个功能。
> - 其他参数设置与单可用区基本一致。

2. 以上海地域为例，在可用区部署模式中，支持创建双可用区和三可用区集群，用户需要选择和可用区数量相等的可用区及子网。

- 节点数量会自动按可用区的倍数调整。为了保证集群的稳定性及可靠性，当用户选择多个可用区时，专用主节点默认开启，可以选择专用主节点个数为三个或五个。
- 专用主节点也会均匀的分布在三个可用区中，保证一个可用区发生不可用的情况下，不会出现超过一半的专用主节点不可用的情况，始终保持集群有超过法定的主节点选择个数，保证了集群的可靠性。

集群多可用区容灾原理

数据节点

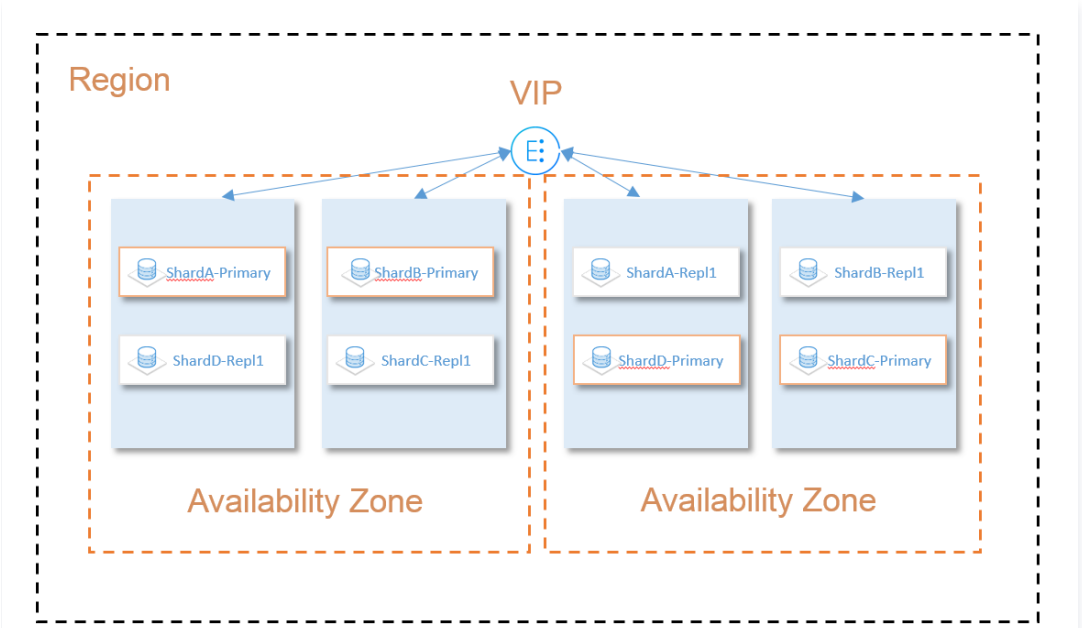
为了保证多可用区容灾的功能生效，用户需要遵守以下原则：
1. 购买集群的数据节点个数为可用区个数的倍数，例如选择两个可用区容灾，那么数据节点个数应该为2、4、6、8...以此类推。
2. 索引分片至少设置1个副本，即保证集群始终有两份以上的数据。

ES 会自动的将用户所购买的数据节点均匀的部署在用户所选择的可用区中，且所部署的数据节点含有可用区感知的功能。该功能使用户数据的副本会分布到多个可用区中，保证单个可用区仅有一份副本。
ES 提供了 VPC 内负载均衡功能，用户通过我们提供的 VIP 连接集群，通过 ES 的 API 进行数据读写及集群控制操作。

©2013-2024 Tencent Cloud. All rights reserved.

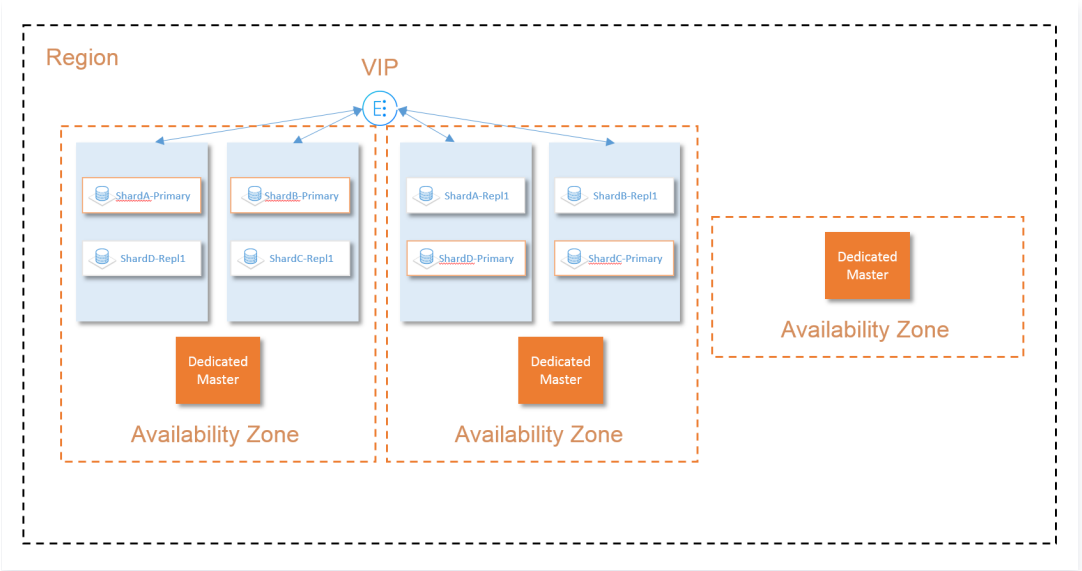
- 此 VIP 绑定了集群内部的所有数据节点，并提供负载均衡功能，用户所有请求会平均分布到集群的各个数据节点上。
- 此 VIP 还带有健康检查功能，如一个周期内多次检查确认某节点没有响应，健康检查功能会暂时从 VIP 的绑定列表中摘除有问题的节点，直到节点恢复正常。

这样就保证了当某个节点宕机，或者某个机房的可用区不可用的情况下，自动的剔除有问题的节点，保证用户的客户端不会请求到有问题的节点。从而在可用区故障的情况下，实现客户业务无感知的切换，提高了客户业务的稳定性。



专用主节点

为了提高集群的可靠性，用户在使用多可用区容灾功能时最少要创建三个专用主节点，且分布在三个不同的可用区中。即便用户选择的是双可用区部署数据节点，我们也会自动的为用户再多选择一个可用区部署专用主节点。这种部署方式，可以保证当一个可用区不可用时，集群依旧有超过半数的法定主节点选举个数，可以保证集群的正常选主。



©2013–2024 Tencent Cloud. All rights reserved.

可视化配置
Kibana

Last updated：2024-10-15 22:03:51

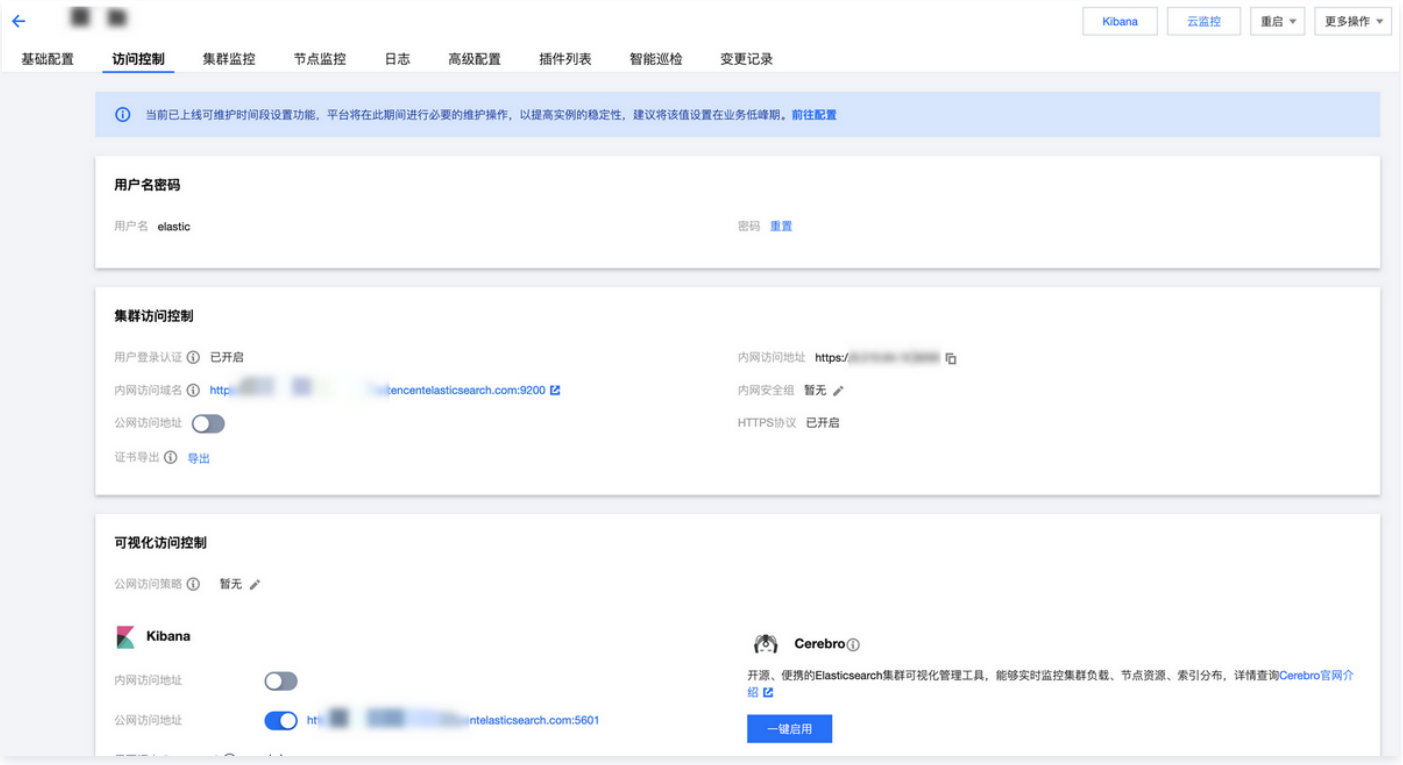
购买腾讯云 Elasticsearch 集群时，将免费提供1个1核2G 规格的 Kibana 节点，此节点将支持 Kibana、Cerebro 组件的使用，您可以根据业务需要，付费升级该 Kibana 节点至更高规格。

背景信息

Kibana 作为 Elastic Stack 的组成部分，天然契合 Elasticsearch 服务，帮助用户实时监测和管理集群状态。腾讯云 Elasticsearch Service 服务提供 Kibana 登录入口，能够调整 Kibana 基础配置，支持动态调整 Kibana 节点资源，为用户业务提供扩展性。

操作步骤

1. 登录腾讯云 Elasticsearch 控制台，单击**集群名称**访问目标集群，在访问控制页的“可视化访问控制”以及页面右上角 **Kibana** 按钮直接进入 Kibana 控制台



2. 访问 Kibana 控制台，在登录界面输入用户名和密码登录。用户名默认为 elastic，密码为集群创建时设置的密码。
3. 支持通过公网和内网访问 Kibana，并在**公网访问策略**模块进行配置，配置方法参考 ES 集群访问控制 。
4. 登入 Kibana 后，支持在控制台进行数据查询、仪表板制作等操作，详细指引请参见 Kibana Guide 。

©2013-2024 Tencent Cloud. All rights reserved.

Cerebro

Last updated：2024-07-23 11:31:21

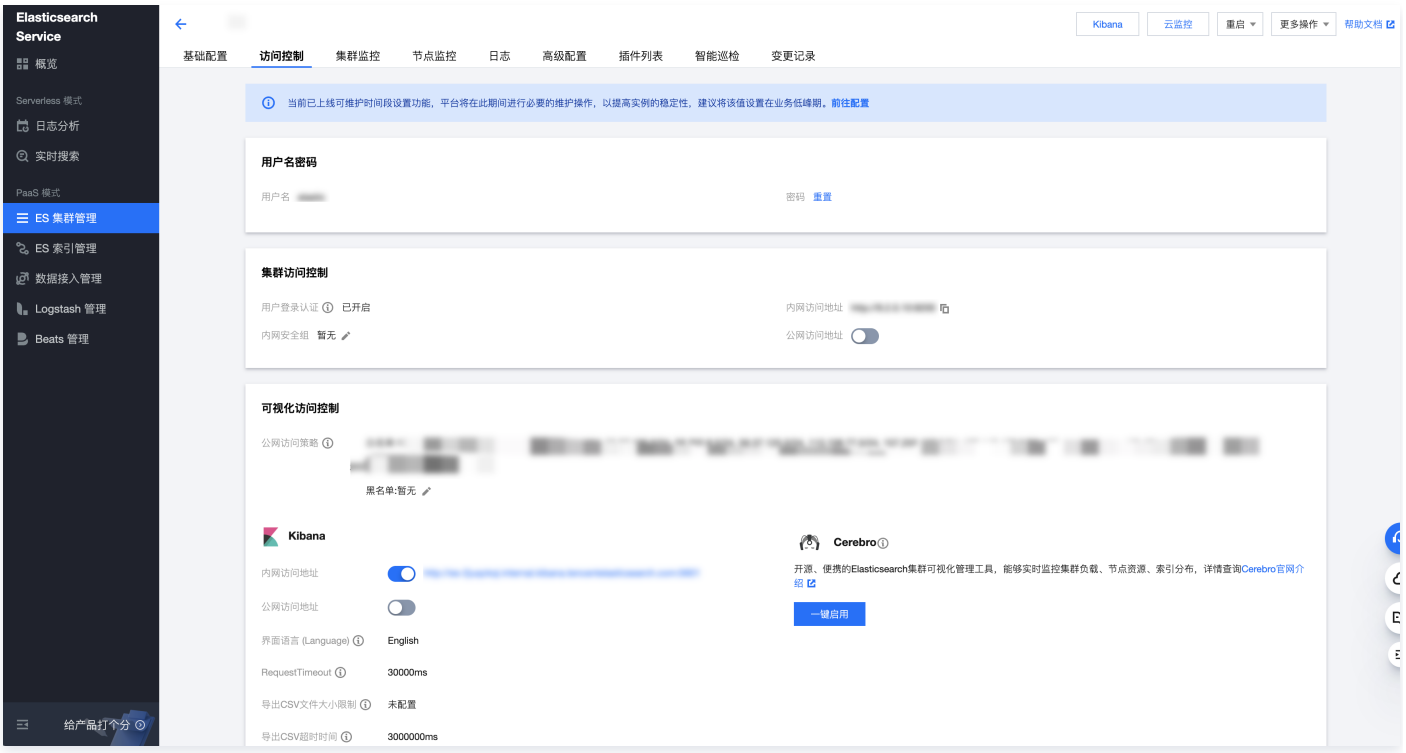
腾讯云 Elasticsearch Service 支持分钟级一键安装 Cerebro，并在控制台轻松访问，完全兼容开源，适配0.9.4版本。

背景信息

Cerebro 是开源的 Elasticsearch 可视化管理工具，支持您通过 Cerebro 对腾讯云 Elasticsearch 集群进行 Web 可视化管理，如监控实时的索引分片负载、执行 request 请求、修改 Elasticsearch 配置等。

操作步骤

1. 登录 腾讯云 Elasticsearch 控制台，单击**集群名称**访问目标集群，跳转至**访问控制**界面。



2. 在 **Cerebro** 功能模块，一键安装 Cerebro（默认未安装），Cerebro 默认安装在 Kibana 节点上，和 Kibana 共用节点资源，启用前建议前往集群变配界面升级 Kibana 节点规格至2核4G及以上。
3. 访问 Cerebro 控制台，在登录界面输入用户名和密码登录。用户名默认为 elastic，密码为集群创建时设置的密码。
4. Cerebro 支持内网和公网访问，并在公网访问策略模块进行配置，配置方案参考 ES 集群访问控制 。
5. 登入 Cerebro 后，支持在控制台进行数据查询、仪表板制作等操作。

©2013-2024 Tencent Cloud. All rights reserved.

集群扩缩容
调整配置

Last updated：2023-05-31 16:46:33

应用场景

在业务发展过程中，集群的数据量和访问量是动态的。随着业务的发展，数据量和访问量都在增大。当集群规模跟实际业务需求不太匹配时，可以动态调整集群的配置，对集群进行扩容；也可以在数据量和访问量不大时，暂时调小集群的规模。腾讯云 ES 支持对节点个数、节点类型和单个数据节点的存储量等进行配置调整。此外，也支持对集群的专用主节点和 Kibana 节点进行调整，具体见以下说明。

操作步骤

1. 登录 ES 控制台，进入集群列表页。
2. 集群调整配置有两个入口，分别位于集群列表页和集群详情页。
   - 在集群列表页，选择需要扩容的集群，选择**操作 > 更多 > 调整配置**。
   - 单击集群名称进入集群详情页，在右上角选择**更多操作 > 调整配置**。
3. 在弹出的调整配置中，根据业务需求对集群节点规格或节点数量进行调整，单击**确定**。

> **说明**
> 一次操作仅支持对节点类型（节点规格和存储）或节点数量中的一项进行扩容或缩容调整，不支持同时调整。

©2013-2024 Tencent Cloud. All rights reserved.

4. 支持根据业务需要选择不同的调配模式，了解调配模式原理查看 集群变配建议和原理介绍 。

5. 配置调整开始后，集群状态为处理中，待集群状态变为正常，即可正常使用。
6. 可以在集群详情页，查看集群的变更进度。

调整配置费用说明

不同计费模式下的集群，调整配置时，费用结算方式会有所不同。

- 预付费包年包月的集群，在调整配置时，会根据集群剩余有效期以及新的配置的定价，计算需要的费用，具体可以参考 调整配置费用说明 。
- 后付费按量计费的集群，计费周期为秒，当配置调整完成后，下一秒费用按新的配置定价进行结算。

©2013–2024 Tencent Cloud. All rights reserved.

集群变配建议和原理介绍

Last updated：2024-03-08 16:01:01

随着客户业务的不断发展，ES 集群的数据规模和访问量也越来越大，当集群规模与配置逐渐和业务实际需求不匹配时，客户可选择及时扩容 ES 集群以满足业务的需要。

变配建议综述

当业务遇到瓶颈时，采取何种方式变配是我们需要面对的问题，您可参考下表结合业务情况进行判断（表中提到的滚动模式和蓝绿模式见后文说明）：

变配方式	适用场景	实现原理介绍	特点说明
增加磁盘容量	计算资源充足，磁盘存储空间不足	直接调整，逐个对集群中节点上挂载的磁盘扩容容量（对加密云盘将采取蓝绿模式）。	平滑扩容不停服，流程时间短，每个节点预计30秒。
增加磁盘数量	计算资源充足，磁盘存储空间不足，或 IOPS 和吞吐量不足	可选择滚动模式或蓝绿模式。	滚动模式较快，蓝绿模式耗时长但对线上业务无影响。
添加更多节点	节点规格较高，但集群整体计算资源不足	直接调整，向集群中加入更多同等规格的节点。	平滑扩容不停服，流程时间短，不受节点数量影响，预计5 – 10分钟完成。
提升节点规格	节点规格较低，且集群整体计算资源不足	可选择滚动模式或蓝绿模式。	滚动模式较快，蓝绿模式耗时长但对线上业务无影响。

说明

- 上表侧重说明场景和变配方式的对应关系，其中实现原理介绍是针对单独调整当前这一项配置的情况，当同时调整多项配置时（如同时调整节点规格、磁盘容量、磁盘数量），实际的方式会有差异。
- 一次扩容操作只支持对纵向（节点规格、磁盘容量、磁盘数量）或横向（节点个数）的其中一个方向进行调整，不支持纵向横向同时调整。

变配基本概念

当集群变配时，常见的有直接调整、滚动模式、蓝绿模式等，下面将分别介绍这几种变配模式：

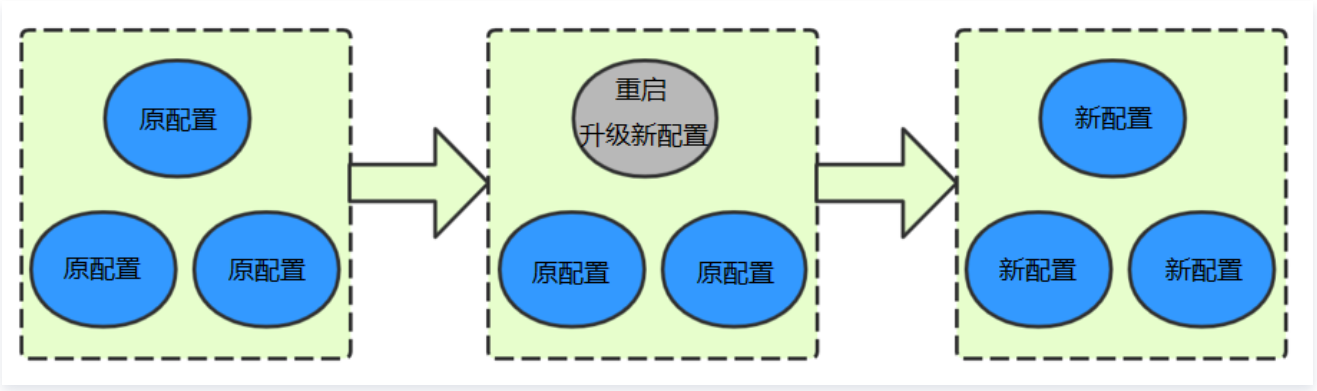
直接调整

直接调整不涉及对集群较重的操作（例如重启或拷贝数据），一般来说对线上业务无影响，且变配速度较快。例如，集群横向添加更多节点、或纵向增加磁盘容量等都是直接调整。

滚动模式

对集群中节点逐个滚动重启完成变配，期间系统服务不间断，但可能影响线上访问性能。

- **优点：**由于不需要迁移数据，扩容的时间不受集群数据规模影响，变配能够快速完成。适合于集群遇到性能瓶颈，期望快速完成扩容变配的场景。
- **缺点：**需要滚动重启集群的每一个数据节点，因此如果集群存在无副本情况下会有丢失数据的风险，且在变配过程中会不停的有节点离开集群和加入集群，会对集群的性能有一定的影响，因此不建议在业务高峰期间选择此种变配方案。

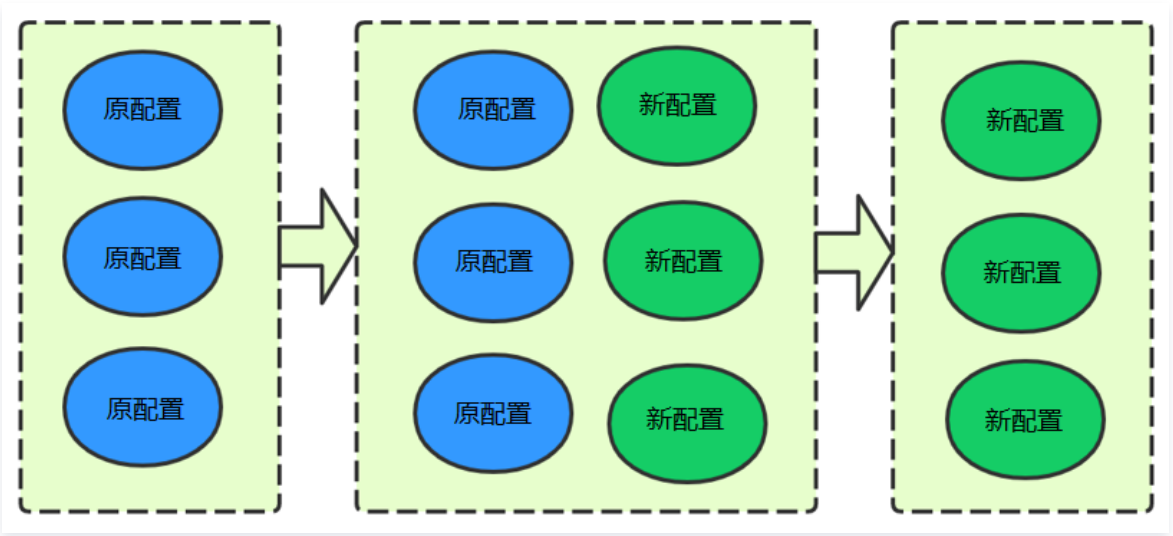


蓝绿模式

©2013-2024 Tencent Cloud. All rights reserved.

不重启集群，为原集群添加相同数量的新节点（配置为需要的节点且拷贝原节点数据），完成配置后，无缝切换并剔除掉原节点，**变配完成后节点 IP 会发生变化**。

- **优点：**变配过程中业务不停服，且扩容非常平滑，适合对集群可用性较高的场景。
- **缺点：**因涉及数据拷贝，变配耗时和集群数据量成正比，变配时间从数分钟到数天或更长。

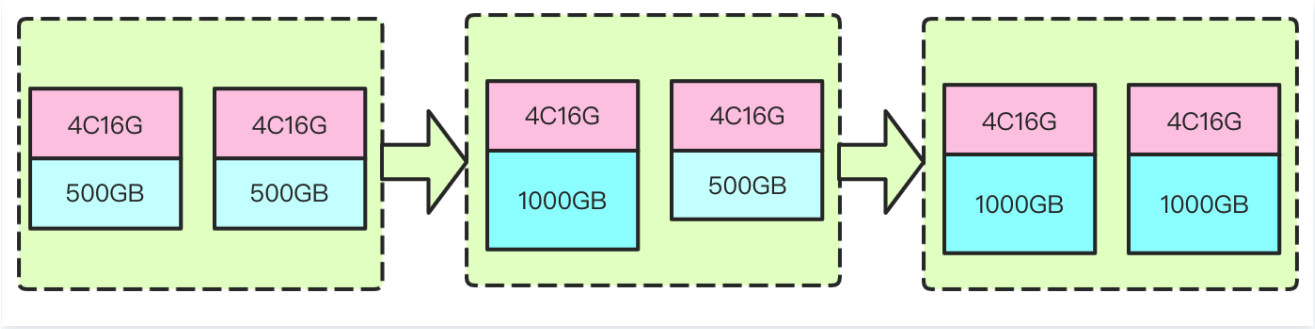


主要变配场景的原理详述

1. 增加磁盘容量

不改变计算资源配置的情况下提升每个节点的磁盘容量，以提升集群整体的存储量。

原理：扩容单盘容量的流程是对集群中每个数据节点上挂载的磁盘依次进行纵向扩容操作。以达到提升整个集群磁盘存储容量的目的。例如，磁盘原有容量为1000GB，将该磁盘容量提升至5000GB。此种扩容方式无需重启节点和集群，对业务使用集群不造成任何影响。由于集群磁盘容量的扩容是采用数据节点滚动扩容的方式，因此扩容时间与集群节点数量成正相关。预计一个节点扩容需要10秒，以10个数据节点的集群为例，预计扩容磁盘的时间在2分钟左右。流程原理示意图如下图所示：



适用场景：适用于计算资源充足，但是磁盘容量不足的情形。例如，客户集群 CPU 负载和内存使用率都较低，但是磁盘平均利用率较高，且数据都比较重要，不能通过删除数据的方式来释放磁盘空间。此时，客户可选择只扩容磁盘容量的方式来扩容集群。

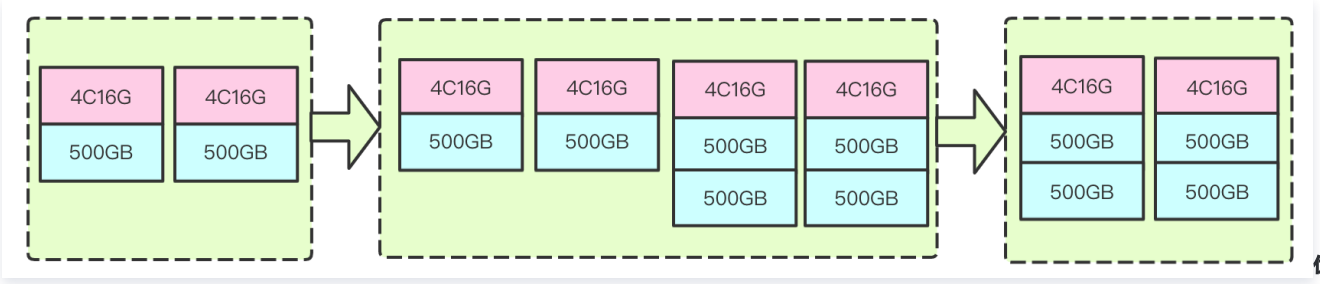
2. 增加磁盘数量

不改变计算资源配置的情况下提升每个节点的磁盘数量，以提升集群整体的存储量、IOPS 和吞吐量。

原理：将节点上挂载的云盘块数进行提升，例如，原集群中每个数据节点上只挂载了一块盘，通过此流程，能够实现集群中每个数据节点上挂载多块云盘，目前支持蓝绿模式和滚动模式两种变配方式。

- **采用蓝绿模式：**
  主要原理是向集群中加入挂载了多块云盘的同等数量的节点，然后将旧节点中的数据迁移至新节点，最后再将旧节点剔除集群。其流程原理示意图如下所示：

©2013–2024 Tencent Cloud. All rights reserved.

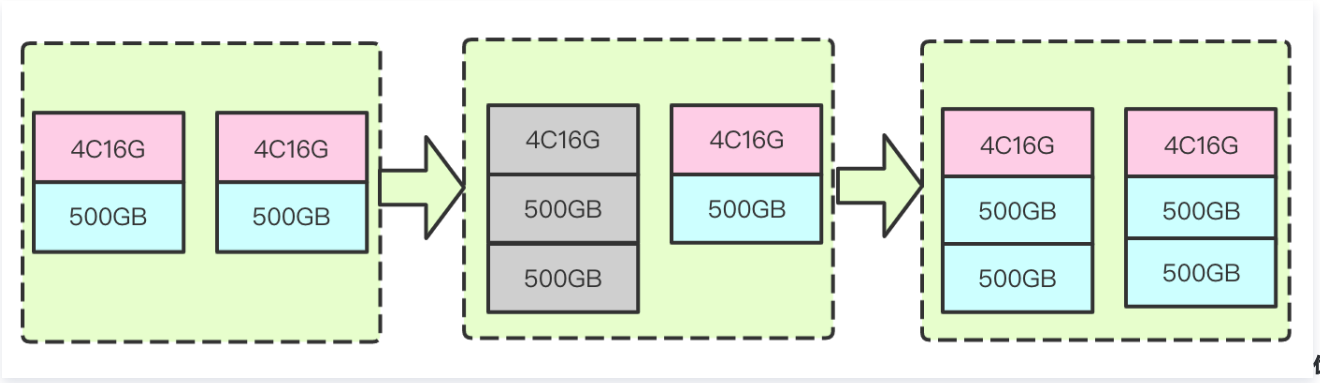


优势：在整个扩容流程中，集群平滑扩容，适用于对业务稳定性和可用性要求较高的场景。

不足：变配的时间受到集群数据量大小的影响。数据量越大，扩容流程时间越长。

- **采用滚动模式：**

主要原理是直接向原节点中加入更多的盘来达到挂载多盘的目的。由于需要修改节点的集群配置，因此节点需要重启后生效，为了最大程度减少对集群性能的影响，采用滚动重启的方式，即每次只对一个节点进行加盘、重启操作，待节点重新加入集群后再对下一个节点进行操作，以此类推。其流程原理示意图如下所示（灰色表示重启中）。



优势：滚动重启扩容的方式由于不需要迁移数据，因此扩容的流程时间不会受到集群数据规模的影响，集群变配能够快速完成。适合于集群遇到性能瓶颈，期望快速完成扩容变配的场景。

不足：此种方案需要滚动重启集群的每一个数据节点，因此，如果集群存在无副本情况下会有丢失数据的风险，且在变配过程中会不停的有节点离开集群和加入集群，会对集群的性能有一定的影响，因此不建议在业务高峰期间选择此种变配方案。

另外，由于 ES 集群不会处理单个节点中的分片再平衡问题，即 ES 不会平衡多数据路径间的分片分配。因此，如果选择了滚动重启的方式向存量节点中直接加盘的方案，ES 会优先将分片分配到最多可用存储空间的磁盘上。这会在短时间内无法发挥多盘的优势，甚至会出现 IO 负载过高的性能问题。直到新磁盘的数据量追上了存量磁盘后，该问题才会得到缓解。另外，当存量集群中单盘的使用率超过了 ES 设置的“水位线”后，意味着整个节点到达了“水位线”，此时即使新盘上还有足够的可用空间，再向节点中加盘也无法解决问题。解决该问题的方法之一是将数据迁移至其他节点或者删除一些数据。

增加磁盘数量方案对比：

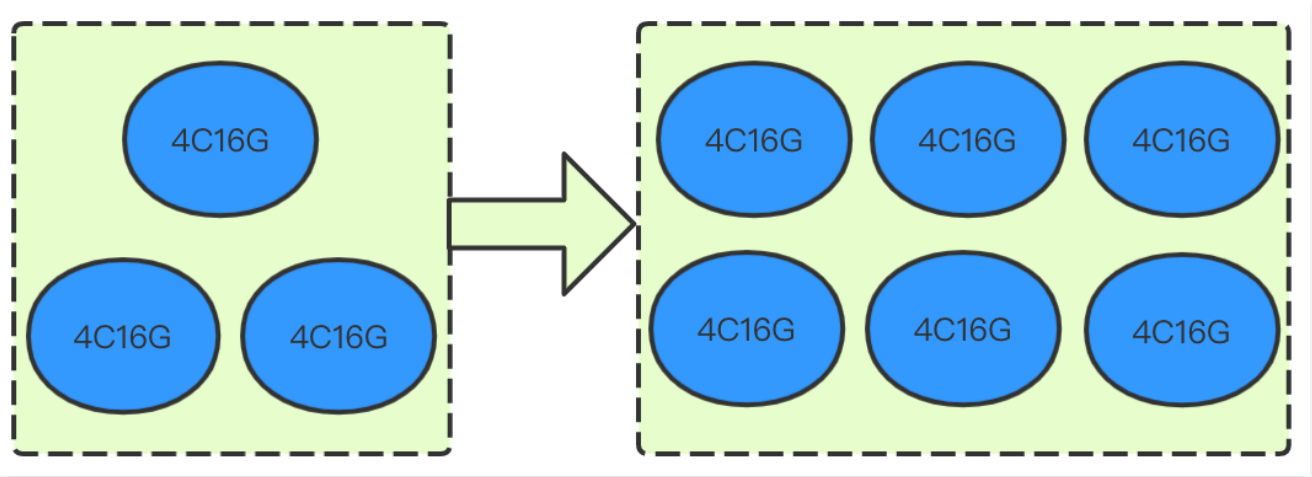
增加磁盘数量方案	优势	不足
蓝绿模式	平滑变配，业务无感知	• 蓝绿模式，流程时间不可控 • 节点 IP 发生变化
滚动模式	• 变配流程时间短，和集群数据量无关 • 节点 IP 不变	• 节点滚动重启，对业务有一定影响 • 无法在短时间内发挥多盘优势 • 无副本索引有丢数据风险

3. 添加更多节点

适用于集群的计算资源遇到瓶颈，如 CPU 负载持续较高，内存使用率居高不下，多次触发内存熔断，甚至内存出现 Out Of Memory 现象。此时您可通过对集群节点进行扩容，以提升集群整体性能。

©2013–2024 Tencent Cloud. All rights reserved.

原理：集群添加节点是指在不改变集群节点机型配置的情况下，通过向集群中加入更多节点来完成集群的扩容操作，其流程原理示意图如下所示：



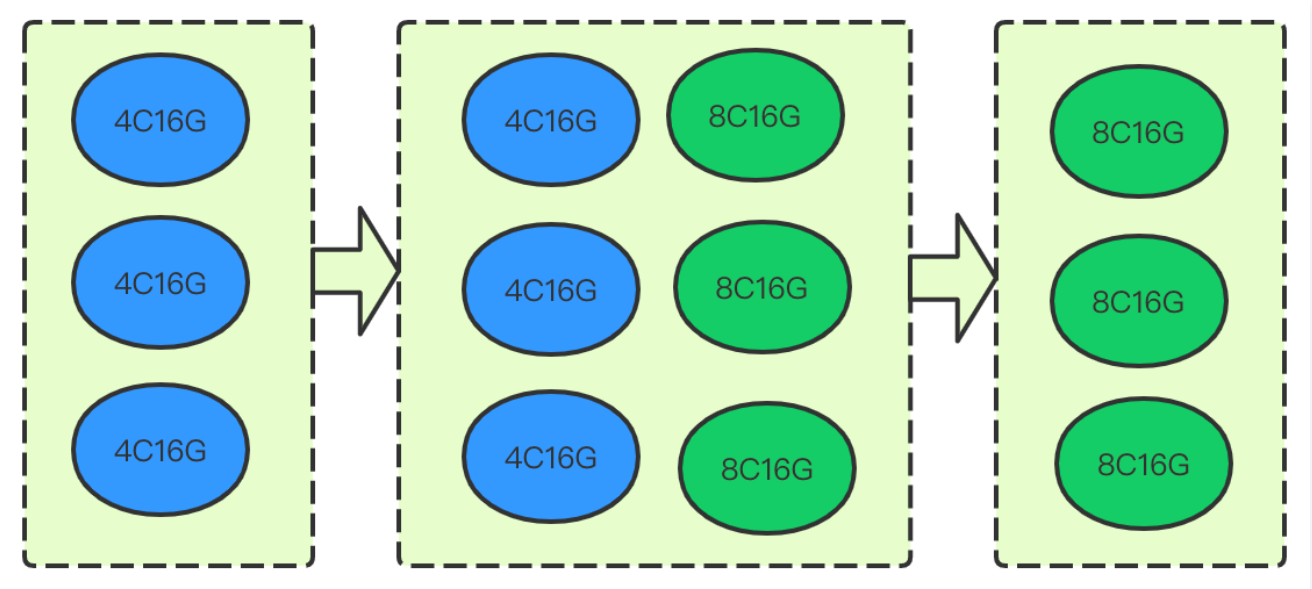
这种扩容流程的优势主要是能够做到平滑扩容，扩容过程中不影响业务使用集群，并且由于在扩容流程中不涉及到新老数据节点间的数据搬迁，因此扩容时间不受节点数量影响，通常在5 – 10分钟完成。
适用场景：节点的配置已经很高，且期望进一步提升集群整体的读写性能，同时对扩容期间集群的稳定性要求较高，这种情况下可选择向集群中添加节点方式进行扩容。

4. 提升节点规格

同样适用于集群的计算资源遇到瓶颈，如 CPU 负载持续较高，内存使用率居高不下，多次触发内存熔断，甚至内存出现 Out Of Memory 现象。可通过对集群节点提升规格，以提升集群整体性能。
原理：节点提升规格是指在不改变集群节点个数的情况下，整体提升集群中数据节点的计算资源配置，目前支持蓝绿模式和滚动模式两种变配方式。例如，将数据节点的配置从4核16G提升到8核16G。

采用蓝绿模式

原理是先将数量大小相等且更高规格的节点加入集群，然后将老节点上的数据全部搬迁到新节点上后，再将老节点剔除集群，以完成集群的扩容操作。其流程原理示意图如下所示：



这种方式的优势主要是能够做到平滑扩容，扩容过程不影响集群的使用和可用性。不足之处在于这种扩容流程需要将老节点中的数据迁移到新节点上，迁移时间受数据量的影响较大，因此如果集群的数据量在 TB 以上，且希望能够尽快完成扩容流程，可选择节点滚动模式，或者通过在 kibana 中设置如下属性，提升数据搬迁的速度：

```
PUT _cluster/settings  {     "persistent": {      "cluster.routing.allocation.node_concurrent_recoveries":
"8",      "indices.recovery.max_bytes_per_sec": "80mb"    }  }
```

©2013–2024 Tencent Cloud. All rights reserved.

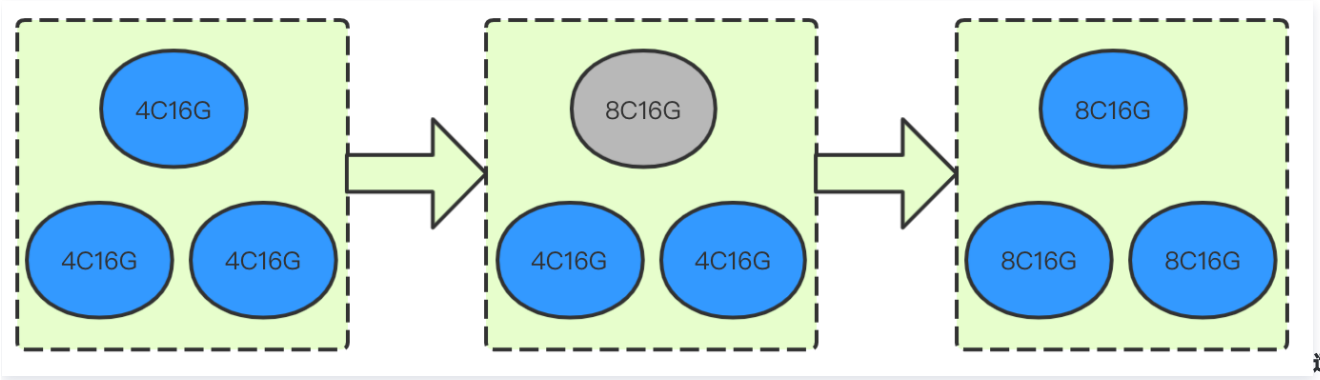
其中：

- **cluster.routing.allocation.node_concurrent_recoveries** 属性表示集群中每个节点上分片并发恢复的个数，默认是2。可根据老数据节点的 CPU 核数 × 4 来确定具体的值，但不要超过50。例如老数据节点为4核16G，则建议该值设置为16，老数据节点为16核64G，则建议该值设置为50，如果发现调大了该值后集群的稳定性受到影响，可适当减小该值。
- **indices.recovery.max_bytes_per_sec** 属性表示节点之间数据传输的最大带宽限制，默认是40mb。该值不宜设置的过高，否则会破坏集群的稳定性，客户可以5mb为步长，逐步调整该限制值，并持续观察集群的稳定性，最终选择一个相对平衡的值。

适用场景：节点的配置较低，期望提升集群整体的读写性能，但是对集群在扩容期间的稳定性要求较高，且对扩容时间不是很紧急，这种情况下可选择数据搬迁的方式进行扩容。

采用滚动模式

原理是逐个对集群中的节点通过重启的方式完成规格配置的扩容操作。由于该变配方式不涉及到数据迁移，因此变配流程时间不受集群数据规模的影响。变配时间与节点个数成正比，每个节点预计需要3 – 5分钟。其流程原理示意图如下所示（灰色表示重启中）：



适用场景：

适合集群节点配置较低，遇到性能瓶颈，对集群稳定性要求不高，期望快速提升集群性能的场景，由于集群在变配过程中节点滚动重启，会有节点不停的离开集群和加入集群，因此建议在业务低峰期操作。

提升节点规格方案对比：

提升节点规格方案	优势	不足
蓝绿模式	平滑变配，业务无感知	• 蓝绿模式，流程时间不可控 • 节点 IP 发生变化
滚动模式	• 变配流程时间短，和集群数据量无关 • 节点 IP 不变	• 节点滚动重启，对业务有一定影响 • 无副本索引有丢数据风险

©2013–2024 Tencent Cloud. All rights reserved.

集群配置
同义词配置

Last updated：2024-10-16 11:20:11

腾讯云 Elasticsearch Service 支持以下两种方式配置同义词：上传同义词文件、直接引用同义词。

方式一：上传同义词文件

注意事项

- 上传同义词文件操作将触发集群滚动重启。
- 新上传/新变更的同义词文件对老索引不生效，需要重建索引。例如，现有的索引 `myindex` 使用了 `synonym.txt` 同义词文件，当该同义词文件的内容变更并重新上传后，现有的索引 `myindex` 不会动态加载更新后的同义词，需要对现有索引进行 reindex 操作，否则更新后的同义词文件只对新建的索引生效。
- 同义词文件要求每行一个同义词表达式（表达式支持 Solr 规则 和 WordNet 规则），并且文件需要为 `utf-8` 编码，扩展名为 `.txt` 。例如：

```
快乐水,可乐 => 可乐,快乐水
elasticsearch,es => es
```

- 同义词文件单个文件最大为10M，上传文件总数最多为10个。

操作步骤

1. 登录 腾讯云 Elasticsearch Service 控制台 。
2. 在集群列表页，单击集群 ID/名称进入集群详情页。
3. 切换到**高级配置页签**，进入**同义词配置**页面，单击**更新词典**。

©2013-2024 Tencent Cloud. All rights reserved.

4. 在**更新同义词**页面上传同义词文件。

5. 上传完成后，单击**保存**。

使用同义词文件

以下实例使用 filter 过滤器配置同义词，使用 `synonym.txt` 作为测试文件，文件内容为 `elasticsearch,es => es` 。

1. 登录已上传同义词文件的集群对应的 Kibana 控制台。登录控制台的具体步骤请参考 通过 Kibana 访问集群 。
2. 单击左侧导航栏的 Dev Tools。
3. 在 Console 中执行如下的命令，创建索引。

```
PUT /my_index
{
  "settings": {
    "index": {
      "analysis": {
        "analyzer": {
          "my_ik": {
            "type": "custom",
            "tokenizer": "ik_smart",
            "filter": [
              "my_synonym"
            ]
          }
        },
        "filter": {
          "my_synonym": {
            "type": "synonym",
            "synonyms_path": "analysis/synonym.txt"
          }
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "content": {
        "type": "text",
```

©2013-2024 Tencent Cloud. All rights reserved.

```
        "analyzer": "my_ik",
        "search_analyzer": "my_ik"
      }
    }
  }
}
```

4. 执行如下命令，验证同义词配置。

```
GET /my_index/_analyze
{
  "analyzer": "my_ik",
  "text":"tencet elasticsearch service"
}
```

- 命令执行成功，将返回如下结果。

```
 {
  "tokens": [
    {
      "token": "tencet",
      "start_offset": 0,
      "end_offset": 6,
      "type": "ENGLISH",
      "position": 0
    },
    {
      "token": "es",
      "start_offset": 7,
      "end_offset": 20,
      "type": "SYNONYM",
      "position": 1
    },
    {
      "token": "service",
      "start_offset": 21,
      "end_offset": 28,
      "type": "ENGLISH",
      "position": 2
    }
  ]
}
```

- 输出结果中，`tokenes` 的类型是 `SYNONYM` 同义词。

5. 执行如下命令，添加一些文档。

```
POST /my_index/_doc/1
{
  "content": "tencet elasticsearch service"
}
POST /my_index/_doc/2
{
  "content": "hello es"
}
```

6. 执行如下命令，搜索同义词。

```
GET my_index/_search
```

©2013-2024 Tencent Cloud. All rights reserved.

```
{
  "query" : { "match" : { "content" : "es" }},
  "highlight" : {
    "pre_tags" : ["<tag1>", "<tag2>"],
    "post_tags" : ["</tag1>", "</tag2>"],
    "fields" : {"content": {}}
  }
}
```

- 命令执行成功后，返回如下结果。

```
{
  "took": 4,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 2,
    "max_score": 0.25811607,
    "hits": [
      {
        "_index": "my_index",
        "_type": "_doc",
        "_id": "2",
        "_score": 0.25811607,
        "_source": {
          "content": "hello es"
        },
        "highlight": {
          "content": [
            "hello <tag1>es</tag1>"
          ]
        }
      },
      {
        "_index": "my_index",
        "_type": "_doc",
        "_id": "1",
        "_score": 0.25316024,
        "_source": {
          "content": "tencet elasticsearch service"
        },
        "highlight": {
          "content": [
            "tencet <tag1>elasticsearch</tag1> service"
          ]
        }
      }
    ]
  }
}
```

方式二：直接引用同义词

1. 登录集群对应的 Kibana 控制台。登录控制台的具体步骤请参考 通过 Kibana 访问集群 。
2. 单击左侧导航栏的 **Dev Tools**。

©2013-2024 Tencent Cloud. All rights reserved.

3. 在 Console 中执行如下的命令，创建索引。

```
PUT /my_index
{
  "settings": {
    "index": {
      "analysis": {
        "analyzer": {
          "my_ik": {
            "type": "custom",
            "tokenizer": "ik_smart",
            "filter": [
              "my_synonym"
            ]
          }
        },
        "filter": {
          "my_synonym": {
            "type": "synonym",
            "synonyms": [
              "elasticsearch,es => es"
            ]
          }
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "content": {
        "type": "text",
        "analyzer": "my_ik",
        "search_analyzer": "my_ik"
      }
    }
  }
}
```

这里与使用同义词文件方式的区别：在 filter 中定义同义词时，直接引用了同义词，而不是同义词文件：`"synonyms": ["elasticsearch,es => es"]`。

4. 执行如下命令，验证同义词配置。

```
GET /my_index/_analyze
{
  "analyzer": "my_ik",
  "text":"tencet elasticsearch service"
}
```

命令执行成功，将返回如下结果。

```
{
  "tokens": [
    {
      "token": "tencet",
      "start_offset": 0,
      "end_offset": 6,
      "type": "ENGLISH",
      "position": 0
    },
```

©2013–2024 Tencent Cloud. All rights reserved.

```
    {
      "token": "es",
      "start_offset": 7,
      "end_offset": 20,
      "type": "SYNONYM",
      "position": 1
    },
    {
      "token": "service",
      "start_offset": 21,
      "end_offset": 28,
      "type": "ENGLISH",
      "position": 2
    }
  ]
}
```

输出结果中，`token` `es` 的类型是 `SYNONYM` 同义词。

5. 执行如下命令，添加一些文档。

```
POST /my_index/_doc/1
{
  "content": "tencet elasticsearch service"
}
POST /my_index/_doc/2
{
  "content": "hello es"
}
```

6. 执行如下命令，搜索同义词。

```
GET my_index/_search
{
  "query" : { "match" : { "content" : "es" }},
  "highlight" : {
    "pre_tags" : ["<tag1>", "<tag2>"],
    "post_tags" : ["</tag1>", "</tag2>"],
    "fields" : {"content": {}}
  }
}
```

命令执行成功后，返回如下结果。

```
{
  "took": 4,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 2,
    "max_score": 0.25811607,
    "hits": [
      {
        "_index": "my_index",
        "_type": "_doc",
```

©2013–2024 Tencent Cloud. All rights reserved.

```
        "_id": "2",
        "_score": 0.25811607,
        "_source": {
          "content": "hello es"
        },
        "highlight": {
          "content": [
            "hello <tag1>es</tag1>"
          ]
        }
      },
      {
        "_index": "my_index",
        "_type": "_doc",
        "_id": "1",
        "_score": 0.25316024,
        "_source": {
          "content": "tencet elasticsearch service"
        },
        "highlight": {
          "content": [
            "tencet <tag1>elasticsearch</tag1> service"
          ]
        }
      }
    ]
  }
}
```

©2013–2024 Tencent Cloud. All rights reserved.

YML 文件配置

Last updated：2023-07-03 14:57:52

您可以通过修改腾讯云 Elasticsearch Service（ES）实例的 YML 参数配置，配置常用的参数。

使用步骤

查看配置项

登录 Elasticsearch Service 控制台，单击需要修改配置的**集群 ID/名称**，进入集群详情页，然后单击**高级配置**，即可查看配置项参数。

YML配置

修改下列YML配置需重启集群后才能生效，其中更多配置支持的配置项请查询帮助文档

修改配置

参数名	参数说明	当前值	取值说明
indices.fielddata.cache.size	指定分配到字段数据的 Java 堆空间的百分比	未配置	百分数，格式：1%-100%，默认值为15%
indices.query.bool.max_clause_count	指定 Lucene 布尔查询中允许的子句的最大数量	未配置	正整数，范围为[1, 2147483647]，默认值为1024
indices.queries.cache.size	设置过滤器缓存的内存大小	未配置	百分数，格式：1%-100%，默认值为10%
indices.queries.cache.count	缓存条目的总大小	未配置	非负整数，范围为[0, 2147483647]，默认值为10000
http.max_initial_line_length	HTTP请求URL的最大长度	未配置	正整数，范围为[1, 20000]，默认值为4KB
http.max_header_size	请求头的最大大小	未配置	正整数，范围为[1, 20000]，默认值为8KB
http.max_content_length	设置请求内容的最大大小	未配置	正整数，范围为[1, 2000]，默认值为100MB
reindex.remote.whitelist	远程 ES 集群访问地址白名单	未配置	填入合法的白名单域名或IP+端口(可选)，如:127.0.0.1:9200、127.0.0.1，通过回车键添加新白名单地址
thread_pool.write.queue_size	文档写入队列大小，适用于6.4.3及以上版本	未配置	整数，范围为[-2147483648,2147483647]，默认值为1024
thread_pool.search.queue_size	文档搜索队列大小	未配置	整数，范围为[-2147483648,2147483647]，默认值为1024

修改配置项

©2013-2024 Tencent Cloud. All rights reserved.

单击**修改配置**，可对相应的配项进行修改，配置项的输入限制，见表格最后一列“取值说明”。

单击**确定修改**，新的配置项参数将会应用到您的集群，会滚动重启集群。如果您集群中的索引有副本，重启后不会对您的业务造成影响，但建议在业务低峰期操作。

> **注意**
> 如果集群健康状态为 YELLOW 或 RED，或集群存在无副本索引时，修改集群配置项对话框会有强制重启的提示和选项框，此时进行更新配置操作有较大风险，建议修复集群状态，为所有索引添加副本，再进行修改配置操作。

如果用户已了解该操作风险，仍要进行更新配置操作，单击**确定**进行重启。详情参考下图：

确定后，腾讯云 ES 实例会进行重启，重启过程中可在集群变更记录中查看进度。重启成功后即可完成 YML 文件的配置。

支持的参数

参数	说明	默认值
indices.fielddata.cache.size	指定分配到字段数据的 Java 堆空间的百分比	15%
indices.query.bool.max_clause_count	指定 Lucene 布尔查询中允许的子句的最大数量	1024

配置 reindex 白名单

参数	说明	默认值
reindex.remote.whitelist	远程 ES 集群访问地址白名单	`[]`

自定义队列大小

©2013–2024 Tencent Cloud. All rights reserved.

参数	说明	默认值
thread_pool.bulk.queue_size	文档写入队列大小，适用于5.6.4版本	1024
thread_pool.write.queue_size	文档写入队列大小，适用于6.4.3及以上版本	1024
thread_pool.search.queue_size	文档搜索队列大小	1024

自定义 CORS 访问配置

参数	说明	默认值
http.cors.enabled	跨域资源共享配置项，`true` 表示启用跨域资源访问，`false` 表示不启用	false
http.cors.allow-origin	域资源配置项，可设置接受来自哪些域名的请求	`""`
http.cors.max-age	获取的 CORS 配置信息在浏览器中的缓存时间	1728000（20天）
http.cors.allow-methods	跨域允许的请求方法	`OPTIONS,HEAD,GET,POST,PUT,DELETE`
http.cors.allow-headers	跨域允许的请求头信息	`X-Requested-With,Content-Type,Content-Length`
http.cors.allow-credentials	是否允许响应头中返回 Access-Control-Allow-Credentials 信息	false

配置 Watcher

参数	说明	默认值
xpack.watcher.enabled	`true` 表示开启 X-Pack 的 Watcher 功能	true

具体配置项详细含义，可参见 Elasticsearch 官方文档 。如果有其他配置项自定义设置需求，可通过 售后支持 进行反馈。

©2013-2024 Tencent Cloud. All rights reserved.

集群场景化模板配置

Last updated：2024-10-16 11:20:11

腾讯云 Elasticsearch Service（ES）提供场景化模板配置功能。通过场景化配置索引模板，您可以根据业务需求选择合适的场景模板，优化该场景下集群的性能，减少由于错误的索引模板导致的性能问题。场景化索引模板支持通用场景、搜索场景和日志场景。本文介绍场景化索引模板的选择方法，并对索引模板参数进行说明。

使用须知

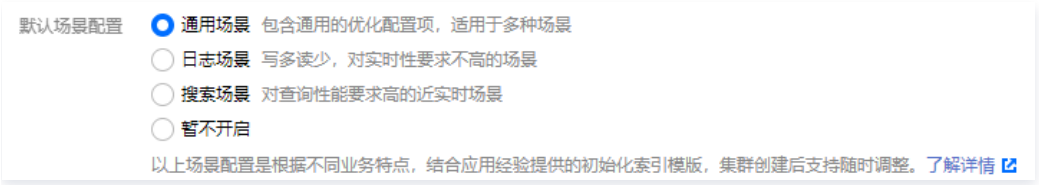
选择场景化索引模板前请注意：

- 腾讯云 ES 集群内已有为大部分场景优化的 默认索引模板，模板名为 `default@template`，您可以在 Kibana 界面的 “Dev Tools” 中通过命令 `GET _template/default@template` 查看这个模板。
- 针对通用场景、日志场景、搜索场景三种特定场景优化的索引模板名为 `scene@template`，您可以在 Kibana 界面的 “Dev Tools” 中通过命令 `GET _template/scene@template` 查看这个模板。
- 新购实例时，可在 购买页 选择场景。**默认将选择通用场景**。购买成功后，对应索引模板的配置会自动应用到集群中，在控制台可以修改场景。

使用步骤

初始化场景配置

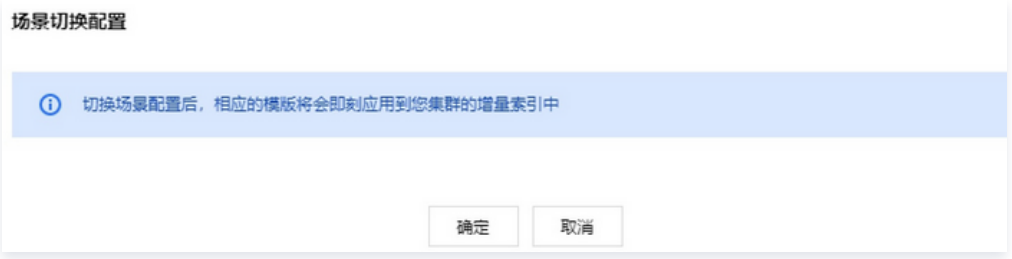
进入 腾讯云 Elasticsearch Service 购买页，在场景化配置区域，选择一个初始化场景配置模板。ES 集群购买成功后，相应索引配置将会应用到该集群。



更改场景配置

1. 登录 腾讯云 Elasticsearch Service 控制台，在**ES 集群管理**中，单击集群 **ID/名称**进入集群详情页，进入高级配置页。

2. 在场景化配置区域，选择要更改的目标场景配置模板，弹出弹窗提示，单击**取消**后，现有模板将维持不变，单击**确定**后，相应的索引配置将会即刻应用到集群。



场景化索引模板说明

场景化索引模板相关的参数说明如下：

参数	说明
order	模板优先级，数值越大优先级越高

©2013-2024 Tencent Cloud. All rights reserved.

index_patterns	索引模板匹配的索引模式，支持通配符，默认为 `*`
index.refresh_interval	索引刷新间隔，被索引的文档在该间隔后才能被查询到，如果对于查询实时性要求较高，可以适当调小该值，但是值过小将影响写入性能
index.translog.durability	translog 数据落盘方式。 • `request`：每次更新操作后都将 translog 数据落盘，保证数据的可靠性，在节点异常时 translog 数据不会丢失 • `async`：异步定时将 translog 数据落盘，牺牲了一定的数据可靠性来提高写入性能
index.translog.sync_interval	translog 落盘方式为 `async` 时配置异步刷新周期

©2013–2024 Tencent Cloud. All rights reserved.

Kona JDK

Last updated：2024-10-12 21:50:12

Kona JDK 介绍

Kona JDK 是什么?

Tencent Kona JDK，是腾讯基于社区 Open JDK 定制开发的 JDK 版本，广泛服务于腾讯内部业务和腾讯云上客户，经过了内部大数据和AI等复杂业务场景的验证，为腾讯 JAVA 生态提供专业持续的保障，具有稳定性高、安全性高、性能好等特点。

Kona JDK 优势

- 高性能和低成本：在腾讯大数据计算场景数万台服务器生产集群验证下，和 Open JDK 8 相比，Kona JDK 8 吞吐量提升8%，CPU 和内存使用率均降低10%左右。
- 开箱即用的 Vector API 支持：解决了向量指令适配导致的 JVM crash 等问题，业界率先落地，稳定支持广告训练场景。
- 多种 GC 优化：G1 GC 内存 overhead 和并行 Full GC 算法优化，同时针对强实时在线服务需求，推出生产级别的 ZGC。
- KonaFiber 协程：已经在IEG天美游戏业务合作落地，目前根据benchmark测试，协程创建/切换/调度等性能大幅超过社区 Loom。
- 支持 ARM 国产化 CPU。

数据对比

本文提供4 核16G 3个节点的腾讯云 Elasticsearch Service 集群 切换 腾讯自研 Kona JDK + G1-GC 的集群性能指标。

> **说明**
> 数据为官方提供的 geonames，包含11396503条地理位置信息，总大小接近3GB，包含了 text、long、geo 等类型数据，覆盖行列存。

对比4核16GB SSD 200G 3节点的腾讯云 ES在 Kona JDK（11.0.9.1-ga+1）+ G1-GC 和 Oracle JDK（1.8.0_181-b13）+CMS-GC 的集群性能指标，Kona JDK + G1-GC 各方面性能均有一定优势。主要得益于腾讯云自研 JDK 以及 GC 参数调优等方面的优化。

说明	Metric	Task	腾讯云ES (Open JDK+CMS-GC)	腾讯云ES (Kona JDK+G1-GC)	差异值（Kona JDK-Open JDK）	Unit	优劣
写入总耗时	Cumulative indexing time of primary shards		17.7745	17.41703333	-0.35747	min	优
GC 总次数、耗时统计	Total Young Gen GC time	-	76.597	4.988	-71.609	s	优
Total Young Gen GC count	-	4129	981	-3148	-	-	-
Total Old Gen GC time	-	0.175	0	-0.175	-	s	-
Total Old Gen GC count	-	2	0	-2	-	-	-
存储大小	Store size	-	2.885286688	2.87756444	-0.00772	GB	-
Translog size	-	3.59E-07	3.59E-07	0	-	GB	-
堆内存使用量	Heap used for segments	-	0.045909882	0.045909882	0	MB	-
Heap used for doc values	-	0.000507355	0.000507355	0	-	MB	-
Heap used for terms	-	0.037261963	0.037261963	0	-	MB	-

©2013-2024 Tencent Cloud. All rights reserved.

Heap used for norms	–	0.003967285	0.003967285	0	–	MB	–
Heap used for points	–	0	0	0	–	MB	–
Heap used for stored fields	–	0.004173279	0.004173279	0	–	MB	–
segment 总数量	Segment count	–	7	7	0	–	–
写入吞吐量、耗时统计	Min Throughput	index-append	82341.03288	83678.1806	1337.14773	docs/s	优
Mean Throughput	index-append	85463.64521	87617.83008	2154.18488	–	docs/s	–
Median Throughput	index-append	85203.41999	87749.05385	2545.63387	–	docs/s	–
Max Throughput	index-append	88282.10166	90448.56087	2166.45921	–	docs/s	–
50th percentile latency	index-append	369.9228433	360.6725829	-9.25026	–	ms	–
90th percentile latency	index-append	582.2157889	521.3938987	-60.82189	–	ms	–
99th percentile latency	index-append	2566.001355	3331.056718	765.05536	–	ms	–
99.9th percentile latency	index-append	3249.023346	4277.054507	1028.03116	–	ms	–
100th percentile latency	index-append	3677.799375	5966.865065	2289.06569	–	ms	–
50th percentile service time	index-append	369.9228433	360.6725829	-9.25026	–	ms	–
90th percentile service time	index-append	582.2157889	521.3938987	-60.82189	–	ms	–
99th percentile service time	index-append	2566.001355	3331.056718	765.05536	–	ms	–
99.9th percentile service time	index-append	3249.023346	4277.054507	1028.03116	–	ms	–
100th percentile service time	index-append	3677.799375	5966.865065	2289.06569	–	ms	–
error rate	index-append	0	0	0	–	%	–
index 指标统计	Min Throughput	index-stats	90.00917629	90.01515386	0.00598	ops/s	–
Mean Throughput	index-stats	90.01643728	90.02981242	0.01338	–	ops/s	–
Median Throughput	index-stats	90.01545276	90.03117871	0.01573	–	ops/s	–
Max Throughput	index-stats	90.035826	90.0417802	0.00595	–	ops/s	–

©2013-2024 Tencent Cloud. All rights reserved.

50th percentile latency	index-stats	2.71807611	2.706557978	-0.01152	-	ms	-
90th percentile latency	index-stats	3.542617336	3.530823253	-0.01179	-	ms	-
99th percentile latency	index-stats	4.209796032	4.047978334	-0.16182	-	ms	-
99.9th percentile latency	index-stats	10.97994745	4.319285007	-6.66066	-	ms	-
100th percentile latency	index-stats	18.87320075	4.494163208	-14.37904	-	ms	-
50th percentile service time	index-stats	1.521472819	1.515000127	-0.00647	-	ms	-
90th percentile service time	index-stats	1.832026243	1.815253124	-0.01677	-	ms	-
99th percentile service time	index-stats	2.493222319	2.149661649	-0.34356	-	ms	-
99.9th percentile service time	index-stats	3.265745808	2.558897269	-0.70685	-	ms	-
100th percentile service time	index-stats	18.49333011	2.714292146	-15.77904	-	ms	-
error rate	index-stats	0	0	0	-	%	-
node 指标统计	Min Throughput	node-stats	89.77465925	89.9748329	0.20017	ops/s	-
Mean Throughput	node-stats	89.90322336	89.99453842	0.09132	-	ops/s	-
Median Throughput	node-stats	89.92739222	89.99716573	0.06977	-	ops/s	-
Max Throughput	node-stats	89.95708367	90.01224368	0.05516	-	ops/s	-
50th percentile latency	node-stats	2.864421345	2.847921103	-0.0165	-	ms	-
90th percentile latency	node-stats	4.034234449	4.02287906	-0.01136	-	ms	-
99th percentile latency	node-stats	4.780995743	4.921176638	0.14018	-	ms	-
99.9th percentile latency	node-stats	11.95199643	8.974571229	-2.97743	-	ms	-
100th percentile latency	node-stats	19.6932666	13.60371523	-6.08955	-	ms	-
50th percentile service time	node-stats	2.031991258	2.032643184	0.00065	-	ms	-
90th percentile service time	node-stats	2.502718102	2.520979941	0.01826	-	ms	-
99th percentile service time	node-stats	3.355697962	3.726954078	0.37126	-	ms	-

©2013-2024 Tencent Cloud. All rights reserved.

99.9th percentile service time	node-stats	6.0700446 31	8.64341661	2.57337	-	ms	-
100th percentile service time	node-stats	19.1344069 3	11.63361594	-7.50079	-	ms	-
error rate	node-stats	0	0	0	-	%	-
默认查询，所有文档 score 为1（match_all）	Min Throughput	default	49.66239789	49.884253 31	0.22186	ops /s	-
Mean Throughput	default	49.8033701 9	49.93227136	0.1289	-	ops /s	-
Median Throughput	default	49.8202478	49.93857317	0.11833	-	ops /s	-
Max Throughput	default	49.8751866 9	49.9563597	0.08117	-	ops /s	-
50th percentile latency	default	3.39414924 4	3.262675833	-0.13147	-	ms	-
90th percentile latency	default	4.57803057 5	4.408122599	-0.16991	-	ms	-
99th percentile latency	default	6.25343985 9	4.992298558	-1.26114	-	ms	-
99.9th percentile latency	default	19.6619895 9	8.490681676	-11.17131	-	ms	-
100th percentile latency	default	20.0197883	8.887056261	-11.13273	-	ms	-
50th percentile service time	default	2.62252427 6	2.356512472	-0.26601	-	ms	-
90th percentile service time	default	3.10205221 2	2.851721831	-0.25033	-	ms	-
99th percentile service time	default	4.57998961 2	3.174401047	-1.40559	-	ms	-
99.9th percentile service time	default	18.4988156 7	8.051926313	-10.44689	-	ms	-
100th percentile service time	default	18.6235634 6	8.21478758	-10.40878	-	ms	-
error rate	default	0	0	0	-	%	-
term 条件查询	Min Throughput	term	98.93043451	99.7145554 5	0.78412	ops /s	-
Mean Throughput	term	99.3341338 2	99.81852777	0.48439	-	ops /s	-
Median Throughput	term	99.3845941 6	99.83007784	0.44548	-	ops /s	-
Max Throughput	term	99.5717623 5	99.88279227	0.31103	-	ops /s	-
50th percentile latency	term	3.25096957 4	3.228969406	-0.022	-	ms	-

©2013-2024 Tencent Cloud. All rights reserved.

90th percentile latency	term	3.966032993	3.853681684	-0.11235	-	ms	-
99th percentile latency	term	10.50691157	4.505703636	-6.00121	-	ms	-
99.9th percentile latency	term	17.10123536	7.033703398	-10.06753	-	ms	-
100th percentile latency	term	19.53481138	9.737900458	-9.79691	-	ms	-
50th percentile service time	term	2.501523588	2.488659229	-0.01286	-	ms	-
90th percentile service time	term	3.069854062	2.982806601	-0.08705	-	ms	-
99th percentile service time	term	7.066733902	3.509562407	-3.55717	-	ms	-
99.9th percentile service time	term	16.17278317	6.230151438	-9.94263	-	ms	-
100th percentile service time	term	19.293964844	8.562799543	-10.73117	-	ms	-
error rate	term	0	0	0	-	%	-
词组查询（query）	Min Throughput	phrase	109.1666798	109.563189	0.39651	ops/s	-
Mean Throughput	phrase	109.4885892	109.726084	0.23749	-	ops/s	-
Median Throughput	phrase	109.531043	109.7627623	0.23172	-	ops/s	-
Max Throughput	phrase	109.6776159	109.8360981	0.15848	-	ops/s	-
50th percentile latency	phrase	2.736125607	2.723197918	-0.01293	-	ms	-
90th percentile latency	phrase	3.2537736	3.277133778	0.02336	-	ms	-
99th percentile latency	phrase	5.174562978	5.252283504	0.07772	-	ms	-
99.9th percentile latency	phrase	17.949861533	11.65228278	-6.29758	-	ms	-
100th percentile latency	phrase	20.167973882	18.0053385	-2.16264	-	ms	-
50th percentile service time	phrase	1.983109396	1.964103896	-0.01901	-	ms	-
90th percentile service time	phrase	2.38582911	2.413296979	0.02747	-	ms	-
99th percentile service time	phrase	4.028498856	3.426500661	-0.602	-	ms	-
99.9th percentile service time	phrase	17.236400991	10.3977854	-6.83862	-	ms	-

©2013-2024 Tencent Cloud. All rights reserved.

100th percentile service time	phrase	19.54707783	17.02223159	-2.52485	-	ms	-
不带缓存的聚合查询（aggregation）	error rate	phrase	0	0	0	%	-
Min Throughput	country_agg_uncached	2.996727436	2.999315225	0.00259	-	ops/s	-
Mean Throughput	country_agg_uncached	2.997330498	2.999446981	0.00212	-	ops/s	-
Median Throughput	country_agg_uncached	2.997367399	2.999449396	0.00208	-	ops/s	-
Max Throughput	country_agg_uncached	2.997811835	2.999560826	0.00175	-	ops/s	-
50th percentile latency	country_agg_uncached	137.708772	138.3623141	0.65354	-	ms	-
90th percentile latency	country_agg_uncached	162.6020876	162.0003162	-0.60177	-	ms	-
99th percentile latency	country_agg_uncached	196.138452 5	190.0452044	-6.09325	-	ms	-
100th percentile latency	country_agg_uncached	208.7042201	205.8009086	-2.90331	-	ms	-
50th percentile service time	country_agg_uncached	136.977708	137.0970309	0.11932	-	ms	-
90th percentile service time	country_agg_uncached	161.4701347	160.9131827	-0.55695	-	ms	-
99th percentile service time	country_agg_uncached	195.4892302	188.7832217	-6.70601	-	ms	-
100th percentile service time	country_agg_uncached	208.3484558	204.5730753	-3.77538	-	ms	-
error rate	country_agg_uncached	0	0	0	-	%	-
带缓存的聚合查询（aggregation）	Min Throughput	country_agg_cached	98.51641526	98.62990947	0.11349	ops/s	-
Mean Throughput	country_agg_cached	98.94635299	99.03419609	0.08784	-	ops/s	-
Median Throughput	country_agg_cached	98.99545733	99.08216113	0.0867	-	ops/s	-
Max Throughput	country_agg_cached	99.247291 84	99.31333794	0.06605	-	ops/s	-
50th percentile latency	country_agg_cached	2.211962827	2.139798366	-0.07216	-	ms	-
90th percentile latency	country_agg_cached	3.517023474	3.494661488	-0.02236	-	ms	-
99th percentile latency	country_agg_cached	4.158023223	4.199306211	0.04128	-	ms	-

©2013-2024 Tencent Cloud. All rights reserved.

99.9th percentile latency	country_agg_cached	9.866942695	8.245296748	-1.62165	-	ms	-
100th percentile latency	country_agg_cached	18.06280296	12.30363548	-5.75917	-	ms	-
50th percentile service time	country_agg_cached	1.467651688	1.393478829	-0.07417	-	ms	-
90th percentile service time	country_agg_cached	1.777389366	1.689927001	-0.08746	-	ms	-
99th percentile service time	country_agg_cached	2.282693414	3.276122157	0.99343	-	ms	-
99.9th percentile service time	country_agg_cached	4.195660669	7.769071626	3.57341	-	ms	-
100th percentile service time	country_agg_cached	16.24826528	11.39958762	-4.84868	-	ms	-
error rate	country_agg_cached	0	0	0	-	%	-
分页拉取	Min Throughput	scroll	20.04421286	20.04208025	-0.00213	pages/s	-
Mean Throughput	scroll	20.05368445	20.05111381	-0.00257	-	pages/s	-
Median Throughput	scroll	20.05292541	20.05042813	-0.0025	-	pages/s	-
Max Throughput	scroll	20.0660563	20.06287951	-0.00318	-	pages/s	-
50th percentile latency	scroll	272.1138773	259.2352917	-12.87859	-	ms	-
90th percentile latency	scroll	290.9470227	265.0907522	-25.85627	-	ms	-
99th percentile latency	scroll	302.488716	284.5098141	-17.9789	-	ms	-
100th percentile latency	scroll	303.7193846	297.6893578	-6.03003	-	ms	-
50th percentile service time	scroll	270.1747189	257.1577448	-13.01697	-	ms	-
90th percentile service time	scroll	289.0668329	263.437889	-25.62894	-	ms	-
99th percentile service time	scroll	300.3281443	282.0971792	-18.23097	-	ms	-
100th percentile service time	scroll	301.2135932	296.1045038	-5.10909	-	ms	-
error rate	scroll	0	0	0	-	%	-
脚本查询（使用 expression 脚本）	Min Throughput	expression	1.500956979	1.501441158	0.00048	ops/s	-

©2013-2024 Tencent Cloud. All rights reserved.

Mean Throughput	expression	1.501160838	1.501741788	0.00058	–	ops/s	–
Median Throughput	expression	1.501147998	1.501719862	0.00057	–	ops/s	–
Max Throughput	expression	1.501414072	1.502131242	0.00072	–	ops/s	–
50th percentile latency	expression	327.3259858	295.2159694	-32.11002	–	ms	–
90th percentile latency	expression	342.0345129	317.3502734	-24.68424	–	ms	–
99th percentile latency	expression	372.6446468	378.85094	6.20629	–	ms	–
100th percentile latency	expression	396.7165332	417.1186928	20.40216	–	ms	–
50th percentile service time	expression	325.855901	293.9883978	-31.8675	–	ms	–
90th percentile service time	expression	340.6900207	316.3654443	-24.32458	–	ms	–
99th percentile service time	expression	370.736203	377.514769	6.77857	–	ms	–
100th percentile service time	expression	395.4625437	415.9661252	20.50358	–	ms	–
error rate	expression	0	0	0	–	%	–
脚本查询（使用 painless 静态脚本，不动态取字段值）	Min Throughput	painless_static	1.396916338	1.397483837	0.00057	ops/s	–
Mean Throughput	painless_static	1.397478748	1.397943395	0.00046	–	ops/s	–
Median Throughput	painless_static	1.397513941	1.397977624	0.00046	–	ops/s	–
Max Throughput	painless_static	1.397920498	1.398303982	0.00038	–	ops/s	–
50th percentile latency	painless_static	431.2919118	371.348965	-59.94295	–	ms	–
90th percentile latency	painless_static	465.1254796	391.1945282	-73.93095	–	ms	–
99th percentile latency	painless_static	512.2339443	437.3164341	-74.91751	–	ms	–
100th percentile latency	painless_static	538.9131764	465.5702729	-73.3429	–	ms	–
50th percentile service time	painless_static	429.9421017	369.791389	-60.15071	–	ms	–
90th percentile service time	painless_static	463.2926716	390.19037	-73.1023	–	ms	–
99th percentile service time	painless_static	511.3802825	434.9970652	-76.38322	–	ms	–

©2013–2024 Tencent Cloud. All rights reserved.

100th percentile service time	painless_static	537.7559569	464.3589323	-73.39702	-	ms	-
error rate	painless_static	0	0	0	-	%	-
脚本查询（使用 painless 静态脚本，动态获取字段值）	Min Throughput	painless_dynamic	1.398724661	1.396323104	-0.0024	ops/s	-
Mean Throughput	painless_dynamic	1.398964022	1.396996725	-0.00197	-	ops/s	-
Median Throughput	painless_dynamic	1.398981831	1.397038282	-0.00194	-	ops/s	-
Max Throughput	painless_dynamic	1.399149307	1.397521303	-0.00163	-	ops/s	-
50th percentile latency	painless_dynamic	432.8310895	356.7619352	-76.06915	-	ms	-
90th percentile latency	painless_dynamic	462.9847418	383.0218635	-79.96288	-	ms	-
99th percentile latency	painless_dynamic	494.9476089	428.2430825	-66.70453	-	ms	-
100th percentile latency	painless_dynamic	536.4017347	446.9218394	-89.4799	-	ms	-
50th percentile service time	painless_dynamic	431.5832192	355.6409795	-75.94224	-	ms	-
90th percentile service time	painless_dynamic	462.0900041	381.7875394	-80.30246	-	ms	-
99th percentile service time	painless_dynamic	494.3205597	425.9035249	-68.41703	-	ms	-
100th percentile service time	painless_dynamic	534.6057713	445.0034723	-89.6023	-	ms	-
error rate	painless_dynamic	0	0	0	-	%	-
地理范围查询（基于高斯衰减函数）	Min Throughput	decay_geo_gauss_function_score	1.001927565	1.002114687	0.00019	ops/s	-
Mean Throughput	decay_geo_gauss_function_score	1.002340802	1.002568849	0.00023	-	ops/s	-
Median Throughput	decay_geo_gauss_function_score	1.002308555	1.002535216	0.00023	-	ops/s	-
Max Throughput	decay_geo_gauss_function_score	1.002881625	1.003158744	0.00028	-	ops/s	-
50th percentile latency	decay_geo_gauss_function_score	387.5082242	332.3548282	-55.1534	-	ms	-

©2013-2024 Tencent Cloud. All rights reserved.

90th percentile latency	decay_geo_gauss_function_score	397.8741518	344.7444949	-53.12966	-	ms	-
99th percentile latency	decay_geo_gauss_function_score	407.4444408	356.9588375	-50.4856	-	ms	-
100th percentile latency	decay_geo_gauss_function_score	409.4531341	369.3594299	-40.0937	-	ms	-
50th percentile service time	decay_geo_gauss_function_score	386.1244814	331.0354254	-55.08906	-	ms	-
90th percentile service time	decay_geo_gauss_function_score	396.8515609	343.3262392	-53.52532	-	ms	-
99th percentile service time	decay_geo_gauss_function_score	406.6675034	355.0055938	-51.66191	-	ms	-
100th percentile service time	decay_geo_gauss_function_score	407.7369291	368.1781357	-39.55879	-	ms	-
error rate	decay_geo_gauss_function_score	0	0	0	-	%	-
地理范围查询（基于高斯衰减函数，且脚本动态获取字段值）	Min Throughput	decay_geo_gauss_script_score	1.001446744	1.001552191	0.00011	ops/s	-
Mean Throughput	decay_geo_gauss_script_score	1.001755635	1.001885174	0.00013	-	ops/s	-
Median Throughput	decay_geo_gauss_script_score	1.001731537	1.001860497	0.00013	-	ops/s	-
Max Throughput	decay_geo_gauss_script_score	1.002160032	1.002318693	0.00016	-	ops/s	-
50th percentile latency	decay_geo_gauss_script_score	411.4939715	334.8041	-76.68987	-	ms	-
90th percentile latency	decay_geo_gauss_script_score	429.658707	345.120705	-84.538	-	ms	-
99th percentile latency	decay_geo_gauss_script_score	453.6645598	355.949311	-97.71525	-	ms	-
100th percentile latency	decay_geo_gauss_script_score	454.430094	358.0469266	-96.38317	-	ms	-
50th percentile service time	decay_geo_gauss_script_s	409.7672189	333.3149343	-76.45228	-	ms	-

©2013-2024 Tencent Cloud. All rights reserved.

	core						
90th percentile service time	decay_geo_gauss_script_score	428.3069702	343.9684012	-84.33857	-	ms	-
99th percentile service time	decay_geo_gauss_script_score	451.8706388	354.2061434	-97.6645	-	ms	-
100th percentile service time	decay_geo_gauss_script_score	452.8327445	356.5891208	-96.24362	-	ms	-
error rate	decay_geo_gauss_script_score	0	0	0	-	%	-
自定义评分函数查询（基于字段值定义函数）	Min Throughput	field_value_function_score	1.503388048	1.503875481	0.00049	ops/s	-
Mean Throughput	field_value_function_score	1.504118746	1.504719285	0.0006	-	ops/s	-
Median Throughput	field_value_function_score	1.504074621	1.504659173	0.00058	-	ops/s	-
Max Throughput	field_value_function_score	1.505051463	1.505800982	0.00075	-	ops/s	-
50th percentile latency	field_value_function_score	194.2629726	134.5724794	-59.69049	-	ms	-
90th percentile latency	field_value_function_score	203.7090491	150.1895525	-53.5195	-	ms	-
99th percentile latency	field_value_function_score	214.6481861	166.6002053	-48.04798	-	ms	-
100th percentile latency	field_value_function_score	217.926288	184.5367327	-33.38956	-	ms	-
50th percentile service time	field_value_function_score	192.3880568	133.1520383	-59.23602	-	ms	-
90th percentile service time	field_value_function_score	202.3297952	148.4251454	-53.90465	-	ms	-
99th percentile service time	field_value_function_score	213.3810514	165.5014484	-47.8796	-	ms	-
100th percentile service time	field_value_function_score	215.2935127	183.1076834	-32.18583	-	ms	-
error rate	field_value_function_score	0	0	0	-	%	-
自定义评分函数查询（通过脚本动态获取字段值计算评分）	Min Throughput	field_value_script_score	1.499697369	1.500258349	0.00056	ops/s	-
Mean Throughput	field_value_script_score	1.499757694	1.500311799	0.00055	-	ops/s	-
Median Throughput	field_value_script_score	1.499759282	1.50030649	0.00055	-	ops/s	-

©2013-2024 Tencent Cloud. All rights reserved.

Max Throughput	field_value_script_score	1.499799232	1.500380821	0.00058	-	ops/s	-
50th percentile latency	field_value_script_score	240.0929602	174.8193153	-65.27364	-	ms	-
90th percentile latency	field_value_script_score	250.0571281	188.9238266	-61.1333	-	ms	-
99th percentile latency	field_value_script_score	270.1539508	215.9618342	-54.19212	-	ms	-
100th percentile latency	field_value_script_score	291.1372129	229.1083755	-62.02884	-	ms	-
50th percentile service time	field_value_script_score	238.8174967	173.5835276	-65.23397	-	ms	-
90th percentile service time	field_value_script_score	248.7244156	187.4786591	-61.24576	-	ms	-
99th percentile service time	field_value_script_score	268.9089779	214.8508051	-54.05817	-	ms	-
100th percentile service time	field_value_script_score	290.2693953	228.2811021	-61.98829	-	ms	-
error rate	field_value_script_score	0	0	0	-	%	-
大量 terms 条件查询（query）	Min Throughput	large_terms	1.101304601	1.100700963	-0.0006	ops/s	-
Mean Throughput	large_terms	1.101582867	1.100849475	-0.00073	-	ops/s	-
Median Throughput	large_terms	1.101561184	1.100839723	-0.00072	-	ops/s	-
Max Throughput	large_terms	1.101945785	1.101043	-0.0009	-	ops/s	-
50th percentile latency	large_terms	211.8277326	242.00767	30.17994	-	ms	-
90th percentile latency	large_terms	231.8088979	252.8580495	21.04915	-	ms	-
99th percentile latency	large_terms	251.2304624	265.9718477	14.74139	-	ms	-
100th percentile latency	large_terms	255.3527635	269.8639119	14.51115	-	ms	-
50th percentile service time	large_terms	203.8265727	233.9178906	30.09132	-	ms	-
90th percentile service time	large_terms	223.8224964	245.1530447	21.33055	-	ms	-
99th percentile service time	large_terms	241.849935	258.2161737	16.36624	-	ms	-
100th percentile service time	large_terms	246.3486325	262.1599194	15.81129	-	ms	-
error rate	large_terms	0	0	0	-	%	-

©2013-2024 Tencent Cloud. All rights reserved.

大量 terms 条件过滤查询（query、filter）	Min Throughput	large_filtered_terms	1.102296697	1.10245872	0.00016	ops/s	-
Mean Throughput	large_filtered_terms	1.102784885	1.102979701	0.00019	-	ops/s	-
Median Throughput	large_filtered_terms	1.102747397	1.102939052	0.00019	-	ops/s	-
Max Throughput	large_filtered_terms	1.103436209	1.103668743	0.00023	-	ops/s	-
50th percentile latency	large_filtered_terms	227.9210831	243.4361419	15.51506	-	ms	-
90th percentile latency	large_filtered_terms	249.2253724	255.9631401	6.73777	-	ms	-
99th percentile latency	large_filtered_terms	263.3142567	276.256653	12.9424	-	ms	-
100th percentile latency	large_filtered_terms	265.4732559	280.1711811	14.69793	-	ms	-
50th percentile service time	large_filtered_terms	220.1946224	235.739741	15.54512	-	ms	-
90th percentile service time	large_filtered_terms	241.1826614	248.5632175	7.38056	-	ms	-
99th percentile service time	large_filtered_terms	255.2141531	268.2613158	13.04716	-	ms	-
100th percentile service time	large_filtered_terms	256.941474	272.5524893	15.61102	-	ms	-
error rate	large_filtered_terms	0	0	0	-	%	-
大量条件取反查询（query、must not）	Min Throughput	large_prohibited_terms	1.102827031	1.102357904	-0.00047	ops/s	-
Mean Throughput	large_prohibited_terms	1.103422713	1.102860804	-0.00056	-	ops/s	-
Median Throughput	large_prohibited_terms	1.103376606	1.102821884	-0.00055	-	ops/s	-
Max Throughput	large_prohibited_terms	1.104211668	1.103525116	-0.00069	-	ops/s	-
50th percentile latency	large_prohibited_terms	202.5767318	232.1078526	29.53112	-	ms	-
90th percentile latency	large_prohibited_terms	220.4174595	247.2566163	26.83916	-	ms	-
99th percentile latency	large_prohibited_terms	234.3344817	266.7954953	32.46101	-	ms	-
100th percentile latency	large_prohibited_terms	246.6131421	268.8084021	22.19526	-	ms	-
50th percentile service time	large_prohibited_terms	193.9010601	224.5439067	30.64285	-	ms	-

©2013-2024 Tencent Cloud. All rights reserved.

90th percentile service time	large_prohibited_terms	212.6108234	239.8692667	27.25844	-	ms	-
99th percentile service time	large_prohibited_terms	226.4359237	258.9916089	32.55569	-	ms	-
100th percentile service time	large_prohibited_terms	238.984541	260.8724702	21.88793	-	ms	-
error rate	large_prohibited_terms	0	0	0	-	%	-
降序排序查询	Min Throughput	desc_sort_population	1.504037884	1.504180086	0.00014	ops/s	-
Mean Throughput	desc_sort_population	1.504907329	1.505087394	0.00018	-	ops/s	-
Median Throughput	desc_sort_population	1.504841628	1.505025118	0.00018	-	ops/s	-
Max Throughput	desc_sort_population	1.506034196	1.506249517	0.00022	-	ops/s	-
50th percentile latency	desc_sort_population	61.13778474	54.50106226	-6.63672	-	ms	-
90th percentile latency	desc_sort_population	73.92849587	69.35394919	-4.57455	-	ms	-
99th percentile latency	desc_sort_population	85.77715084	86.2006122	0.42346	-	ms	-
100th percentile latency	desc_sort_population	85.84200498	87.74290979	1.9009	-	ms	-
50th percentile service time	desc_sort_population	59.92501229	53.58439684	-6.34062	-	ms	-
90th percentile service time	desc_sort_population	72.30911367	68.09855495	-4.21056	-	ms	-
99th percentile service time	desc_sort_population	84.09957183	84.57749833	0.47793	-	ms	-
100th percentile service time	desc_sort_population	84.19063408	85.95814556	1.76751	-	ms	-
error rate	desc_sort_population	0	0	0	-	%	-
升序排序查询	Min Throughput	asc_sort_population	1.504247142	1.504528234	0.00028	ops/s	-
Mean Throughput	asc_sort_population	1.505166062	1.505508302	0.00034	-	ops/s	-
Median Throughput	asc_sort_population	1.505099341	1.505440428	0.00034	-	ops/s	-
Max Throughput	asc_sort_population	1.506349989	1.506767598	0.00042	-	ops/s	-
50th percentile latency	asc_sort_population	63.16776341	62.82690261	-0.34086	-	ms	-

©2013-2024 Tencent Cloud. All rights reserved.

90th percentile latency	asc_sort_population	78.09764324	75.61749704	-2.48015	-	ms	-
99th percentile latency	asc_sort_population	87.33172638	84.58683862	-2.74489	-	ms	-
100th percentile latency	asc_sort_population	87.89979294	85.19899659	-2.7008	-	ms	-
50th percentile service time	asc_sort_population	61.90986466	61.71039958	-0.19947	-	ms	-
90th percentile service time	asc_sort_population	76.55056985	74.45018981	-2.10038	-	ms	-
99th percentile service time	asc_sort_population	85.81453795	83.30245529	-2.51208	-	ms	-
100th percentile service time	asc_sort_population	86.60695888	84.05557927	-2.55138	-	ms	-
error rate	asc_sort_population	0	0	0	-	%	-
升序排序后 after 跳转查询	Min Throughput	asc_sort_with_after_population	1.503017792	1.503506494	0.00049	ops/s	-
Mean Throughput	asc_sort_with_after_population	1.503667166	1.504271472	0.0006	-	ops/s	-
Median Throughput	asc_sort_with_after_population	1.503620246	1.504214876	0.00059	-	ops/s	-
Max Throughput	asc_sort_with_after_population	1.504506304	1.505248472	0.00074	-	ops/s	-
50th percentile latency	asc_sort_with_after_population	79.49512405	81.97531058	2.48019	-	ms	-
90th percentile latency	asc_sort_with_after_population	94.074154118	97.05960844	2.98545	-	ms	-
99th percentile latency	asc_sort_with_after_population	115.1407234	110.1778366	-4.96289	-	ms	-
100th percentile latency	asc_sort_with_after_population	117.4867153	115.6357806	-1.85093	-	ms	-
50th percentile service time	asc_sort_with_after_population	78.126538008	80.56232799	2.43579	-	ms	-
90th percentile service time	asc_sort_with_after_population	92.577915336	96.00112103	3.42321	-	ms	-
99th percentile service time	asc_sort_with_after_popula	113.538067	108.2517642	-5.2863	-	ms	-

©2013-2024 Tencent Cloud. All rights reserved.

	tion						
100th percentile service time	asc_sort_with_after_population	116.0558164	114.5531256	-1.50269	-	ms	-
error rate	asc_sort_with_after_population	0	0	0	-	%	-
高基字段降序排序查询（基于 DistanceFeatureQuery 快速取 topK）	Min Throughput	desc_sort_geonameid	6.011806976	6.013534032	0.00173	ops/s	-
Mean Throughput	desc_sort_geonameid	6.014040004	6.016121536	0.00208	-	ops/s	-
Median Throughput	desc_sort_geonameid	6.013860893	6.015844404	0.00198	-	ops/s	-
Max Throughput	desc_sort_geonameid	6.016975785	6.019491653	0.00252	-	ops/s	-
50th percentile latency	desc_sort_geonameid	8.037513588	6.896098144	-1.14142	-	ms	-
90th percentile latency	desc_sort_geonameid	8.790209144	7.481213845	-1.309	-	ms	-
99th percentile latency	desc_sort_geonameid	20.16597	7.890859395	-12.27511	-	ms	-
100th percentile latency	desc_sort_geonameid	22.691944661	8.130467497	-14.56148	-	ms	-
50th percentile service time	desc_sort_geonameid	7.199986372	6.043605972	-1.15638	-	ms	-
90th percentile service time	desc_sort_geonameid	7.634483464	6.330675445	-1.30381	-	ms	-
99th percentile service time	desc_sort_geonameid	18.951113335	6.674837489	-12.27628	-	ms	-
100th percentile service time	desc_sort_geonameid	22.069349988	6.795545109	-15.2738	-	ms	-
error rate	desc_sort_geonameid	0	0	0	-	%	-
高基字段降序排序 after 跳转查询	Min Throughput	desc_sort_with_after_geonameid	6.003999251	6.002224615	-0.00177	ops/s	-
Mean Throughput	desc_sort_with_after_geonameid	6.00483332	6.002715504	-0.00212	-	ops/s	-
Median Throughput	desc_sort_with_after_geonameid	6.004831591	6.002684836	-0.00215	-	ops/s	-
Max Throughput	desc_sort_with_after_geonameid	6.005864935	6.003257919	-0.00261	-	ops/s	-

©2013-2024 Tencent Cloud. All rights reserved.

50th percentile latency	desc_sort_with_after_geonameid	64.12782287	69.3480419	5.22022	-	ms	-
90th percentile latency	desc_sort_with_after_geonameid	79.63361973	85.98741582	6.3538	-	ms	-
99th percentile latency	desc_sort_with_after_geonameid	87.09606319	91.30932659	4.21326	-	ms	-
100th percentile latency	desc_sort_with_after_geonameid	88.47462852	91.78488795	3.31026	-	ms	-
50th percentile service time	desc_sort_with_after_geonameid	63.23770666	68.51645093	5.27874	-	ms	-
90th percentile service time	desc_sort_with_after_geonameid	78.83979175	85.22403594	6.38424	-	ms	-
99th percentile service time	desc_sort_with_after_geonameid	86.525729	90.76162191	4.23589	-	ms	-
100th percentile service time	desc_sort_with_after_geonameid	87.29847241	91.3709281	4.07246	-	ms	-
error rate	desc_sort_with_after_geonameid	0	0	0	-	%	-
高基字段升序排序查询（基于DistanceFeatureQuery快速取 topK）	Min Throughput	asc_sort_geonameid	6.018840993	6.018470998	-0.00037	ops/s	-
Mean Throughput	asc_sort_geonameid	6.022518134	6.022078364	-0.00044	-	ops/s	-
Median Throughput	asc_sort_geonameid	6.022245684	6.021816641	-0.00043	-	ops/s	-
Max Throughput	asc_sort_geonameid	6.027178807	6.026594943	-0.00058	-	ops/s	-
50th percentile latency	asc_sort_geonameid	7.024060003	9.012220893	1.98816	-	ms	-
90th percentile latency	asc_sort_geonameid	7.69297732	9.680523816	1.98755	-	ms	-
99th percentile latency	asc_sort_geonameid	20.44826921	11.18117133	-9.2671	-	ms	-
100th percentile latency	asc_sort_geonameid	21.87036537	11.28741447	-10.58295	-	ms	-
50th percentile service time	asc_sort_geonameid	6.123304367	8.064015303	1.94071	-	ms	-
90th percentile service time	asc_sort_geonameid	6.746383384	8.737695683	1.99131	-	ms	-

©2013-2024 Tencent Cloud. All rights reserved.

99th percentile service time	asc_sort_geonameid	19.78318544	10.16213525	-9.62105	-	ms	-
100th percentile service time	asc_sort_geonameid	21.25467733	10.39039157	-10.86429	-	ms	-
error rate	asc_sort_geonameid	0	0	0	-	%	-
高基字段升序排序 after 跳转查询	Min Throughput	asc_sort_with_after_geonameid	6.013236842	6.013266563	0.00003	ops/s	-
Mean Throughput	asc_sort_with_after_geonameid	6.015849171	6.015858176	0.00001	-	ops/s	-
Median Throughput	asc_sort_with_after_geonameid	6.015618744	6.015641333	0.00002	-	ops/s	-
Max Throughput	asc_sort_with_after_geonameid	6.019167352	6.01911523	-0.00005	-	ops/s	-
50th percentile latency	asc_sort_with_after_geonameid	60.27546292	64.3463349	4.07087	-	ms	-
90th percentile latency	asc_sort_with_after_geonameid	78.63363056	85.38805693	6.75443	-	ms	-
99th percentile latency	asc_sort_with_after_geonameid	89.31191583	91.7664034	2.45449	-	ms	-
100th percentile latency	asc_sort_with_after_geonameid	90.85853212	91.9917766	1.13324	-	ms	-
50th percentile service time	asc_sort_with_after_geonameid	59.692265	63.68059153	3.98833	-	ms	-
90th percentile service time	asc_sort_with_after_geonameid	78.16235274	84.53184282	6.36949	-	ms	-
99th percentile service time	asc_sort_with_after_geonameid	88.15484255	91.29356634	3.13872	-	ms	-
100th percentile service time	asc_sort_with_after_geonameid	89.73695803	91.64701309	1.91006	-	ms	-
error rate	asc_sort_with_after_geonameid	0	0	0	-	%	-

©2013-2024 Tencent Cloud. All rights reserved.

插件配置
插件列表

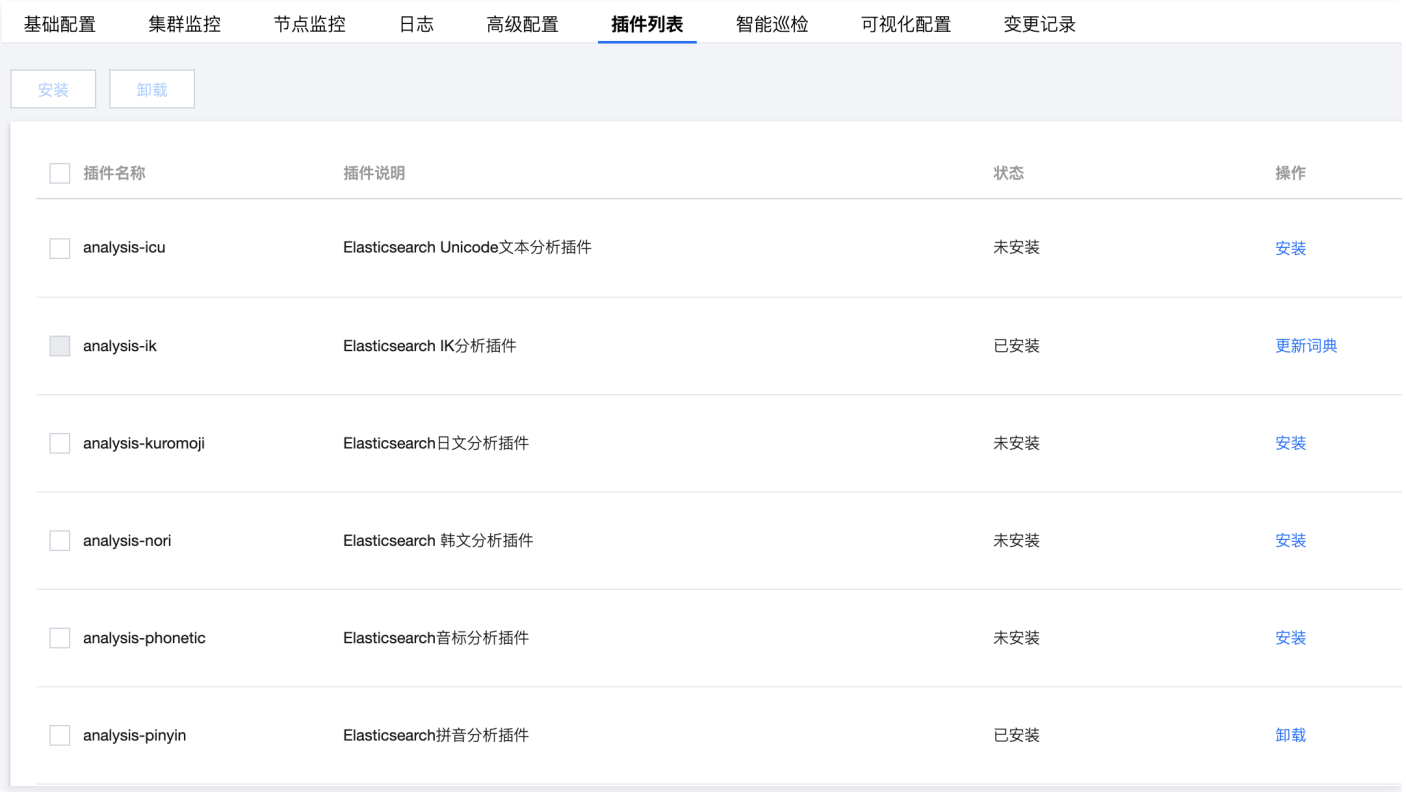
Last updated：2022-01-28 15:03:53

腾讯云 Elasticsearch Service 提供了包括开源 ES 支持插件和自研插件在内的10余款插件，为您提供丰富的插件功能。当您购买了腾讯云 ES 实例后，您可以根据需求在插件列表页面安装或者卸载这些插件。本文介绍安装或卸载腾讯云 ES 插件的方法。

> **注意**
> 安装或者卸载插件将触发集群滚动重启，请确认后操作。

操作步骤

1. 登录 腾讯云 Elasticsearch Service 控制台 。
2. 在集群列表页，单击**集群 ID** 进入集群详情页。
3. 单击**插件列表**，进入插件列表管理页面。



4. 单击对应插件右侧操作栏下的**安装**或者**卸载**。
5. 在弹出的对话框中，阅读注意事项，确认无误后单击确认。确认后，插件变更操作开始，集群将会重启，可以在**集群变更记录**中查看变更进度。

插件信息

腾讯云 Elasticsearch Service 支持的插件如下：

插件名称	默认状态	描述	支持的操作
analysis-icu	未安装	Elasticsearch Unicode 文本分析插件	安装、卸载
analysis-ik	已安装	Elasticsearch IK 分析插件，默认不能卸载。	更新词典
analysis-kuromoji	未安装	Elasticsearch 日文分析插件	安装、卸载

©2013-2024 Tencent Cloud. All rights reserved.

analysis-nori	未安装	Elasticsearch 韩文分析插件	安装、卸载
analysis-phonetic	未安装	Elasticsearch 音标分析插件	安装、卸载
analysis-pinyin	已安装	Elasticsearch 拼音分析插件	安装、卸载
analysis-qq	未安装	Elasticsearch QQ 分析插件	安装、卸载、更新词典
analysis-smartcn	未安装	Elasticsearch 智能中文分析插件	安装、卸载
analysis-stconvert	已安装	Elasticsearch 繁简体分析插件	安装、卸载
ingest-attachment	未安装	Apache Tika 信息提取插件	安装、卸载
ingest-geoip	未安装	IP 地址解析插件，6.8.2以上版本已集成在 es 模块中，不需安装	安装、卸载
ingest-user-agent	未安装	浏览器 user agent 信息提取插件，6.8.2以上版本已集成在 es 模块中，不需安装	安装、卸载
mapper-murmur3	未安装	插件用于在创建索引时计算并存储字段的 hash 值	安装、卸载
mapper-size	未安装	插件用于在创建索引时记录文档压缩前的大小	安装、卸载
repository-cos	已安装	插件用于存储 Elasticsearch Snapshot 到腾讯云 COS，详见 使用 COS 进行备份及恢复，默认不能卸载	无
repository-hdfs	未安装	插件提供了对 Hadoop 分布式文件系统（HDFS）存储库的支持	安装、卸载
sql	未安装	开源 SQL 解析插件，基础版、白金版集群已带 sql 解析功能，不需安装	安装、卸载

©2013-2024 Tencent Cloud. All rights reserved.

IK 分词插件

Last updated：2024-03-08 16:01:01

登录 腾讯云 Elasticsearch Service 控制台，选择一个集群，进入集群详情页的**插件列表**页面，可以看到系统默认插件中已经预装了 IK 中文分词插件。关于 IK 中文分词插件的介绍，详情可查看 IK Analysis for Elasticsearch，您可以利用该插件对存到 ES 集群数据中的中文关键词建立索引，实现搜索功能。

更新词典

单击**更新词典**，进入更新词典页面。有分词词典和停用词词典两项，单击**本地上传**，选择您需要更新的词典文件后，单击**保存**，即可热更新词典（不需要重启集群）。

关于词典文件的要求及说明

- **词典类型：**有两类词，“分词词典”和“停用词词典”。“分词词典”中的词是用户在向 ES 集群存入数据，建立索引的时候，指定 IK 作为分词工具。如果存入的数据中时有这类词，就会建立索引，并能通过关键词查询搜索到。“停用词词典”则会刻意回避不建立索引的词。
- **限制要求：**对于词典文件，也有一些限制和要求，需要一行一个词，utf-8 编码。为了避免混淆，分词词典和停用词词典文件名不能重复。另外，因为词典文件会加载到内存中，所以对文件的大小和个数也做了一定的限制，单个文件最大为20M，上传文件总数最多为10个。
- **更新过程：**列表会展示历史已经更新上传的词典。新上传的词典，如果不符合要求，会直接限制上传。上传完成后，词典文件会显示成“待生效”状态。所有需要更新的词典上传完成后，单击**保存**，会保存到用户的集群并生效。如果有上传失败的文件，或不是 utf-8 格式的文件，会提示失败，需要删除失败的文件后，才能单击保存生效。

©2013-2024 Tencent Cloud. All rights reserved.

QQ 分词插件

Last updated：2024-10-08 14:18:21

QQ 分词插件是由腾讯云 ES 团队与腾讯 NLP 团队联合研发的中文分词插件，在腾讯内部广泛应用于 QQ、微信、浏览器等业务。在传统词典分词的基础上，增加了 NER 命名实体识别，同时支持自定义词库功能。QQ 分词插件经过多年的应用实践和不断打磨优化，在分词准确度、分析速度等关键指标上均处于业界领先，您可以在腾讯云 ES 中使用 QQ 分词插件来完成文档的分析和检索。

使用须知

由于 QQ 分词插件团队已暂停更新该插件，新购集群已在**插件列表**模块中去掉该插件，存量集群可正常使用。
QQ 分词插件仅支持数据节点规格在4核16G及以上的集群，如果集群未安装 QQ 分词插件，请在插件列表页面安装 QQ 分词插件（analysis-qq）。
QQ 分词插件提供如下的分析器（analyzer）和分词器（tokenizer）：

- 分析器：qq_smart, qq_max, qq_smart_ner, qq_max_ner。
- 分词器：qq_smart, qq_max, qq_smart_ner, qq_max_ner。

您可以使用上述的分析器和分词器完成文档的分析和查询。您也可以通过词库配置功能，自定义更新分词词库，详情请参见下文的配置词库。

说明

- `qq_max` 和 `qq_smart` 有什么区别?
  qq_max 会对文本做最细粒度的拆分，qq_smart 会对文本做最粗粒度的拆分。
- ner 是什么？为什么 ner 功能要独立一个分词器?
  ner 是 Named Entity Recognition（命名实体识别）的简称，可以识别文本中具有特定意义的实体，主要包括人名、地名、机构名、专有名词等。对于这一类专有名词，不需要用户上传自定义词库。将 ner 功能单独保证为一个分词器，主要是因为 ner 功能需要加载一个模型，首次加载时间比较长。

注意

QQ 分词插件现无法识别中文、英文以外的词，如上传的文件中包含其他语言的内容，分词结果会过滤这些内容，类似于过滤标点符号。

使用步骤

1. 登录已安装 QQ 分词插件的集群对应的 Kibana 控制台。登录控制台的具体步骤请参考 通过 Kibana 访问集群 。
2. 单击左侧导航栏的 Dev Tools。
3. 在 Console 中使用 QQ 分词插件的分析器创建索引。

```
PUT /index
{
  "mappings": {
    "_doc": {
      "properties": {
        "content": {
          "type": "text",
          "analyzer": "qq_max",
          "search_analyzer": "qq_smart"
        }
      }
    }
  }
}
```

上面的语句创建了一个名称为 `index` 的索引，类型为 `_doc`（ES 7及以上版本需要在创建索引时加入 `?include_type_name=true` 才能支持类型）。包含了一个 `content` 属性，类型为 `text`，并使用了 `qq_max` 和 `qq_smart` 分析器。执行成功后，将返回如下结果。

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

©2013-2024 Tencent Cloud. All rights reserved.

```
}
```

4. 添加文档。

```
POST /index/_doc/1
{
  "content": "我从微信上下载了王者荣耀"
}
POST /index/_doc/2
{
  "content": "住建部:9月底前完成名镇名村景观资源登记"
}
POST /index/_doc/3
{
  "content": "中国气象台最新天气预报"
}
POST /index/_doc/4
{
  "content": "我家住在中国古建筑保护协会附近"
}
```

上面的语句导入了4个文档，将使用 `qq_max` 分析器对文档进行分析。

5. 使用关键词高亮的方式查询文档。

```
GET index/_search
{
  "query" : { "match" : { "content" : "中国" }},
  "highlight" : {
    "pre_tags" : ["<tag1>", "<tag2>"],
    "post_tags" : ["</tag1>", "</tag2>"],
    "fields" : {"content": {}}
  }
}
```

上面的语句在所有 `_doc` 类型的文档中，使用 `qq_smart` 分析器，搜索 `content` 字段中包含 `中国` 的文档。执行成功后，返回如下结果。

```
{
  "took" : 108,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 2,
      "relation" : "eq"
    },
    "max_score" : 0.7199211,
    "hits" : [
      {
        "_index" : "index",
        "_type" : "_doc",
        "_id" : "4",
        "_score" : 0.7199211,
        "_source" : {
          "content" : "我家住在中国古建筑保护协会附近"
```

©2013–2024 Tencent Cloud. All rights reserved.

```
        },
        "highlight" : {
          "content" : [
            "我家住在<tag1>中国</tag1>古建筑保护协会附近"
          ]
        }
      },
      {
        "_index" : "index",
        "_type" : "_doc",
        "_id" : "3",
        "_score" : 0.6235748,
        "_source" : {
          "content" : "中国气象台最新天气预报"
        },
        "highlight" : {
          "content" : [
            "<tag1>中国</tag1>气象台最新天气预报"
          ]
        }
      }
    ]
  }
}
```

使用自定义词典

1. 登录 Elasticsearch Service 控制台，在集群列表页，单击集群 ID/名称进入集群详情页。

2. 单击**插件列表**，进入插件列表管理页面。

3. 找到 QQ 分词插件（analysis-qq），单击右侧**更新词典**。
4. 词典文件要求如下。
   - 词典文件必须为 GBK 编码，一行一个词，且文件扩展名为 `.dic` 。
   - 单个文件最大为10M，上传文件总数最多为10个。

©2013-2024 Tencent Cloud. All rights reserved.

5. 单击**保存**。保存后，不会触发集群重启，但需要若干分钟触发集群变更使词典文件生效。

排查测试

如果您在使用 QQ 分词插件时，得到的结果不符合预期，可以通过下面的语句对分析器和分词器进行排查测试。

```
GET _analyze
{
  "text": "我家住在中国古建筑保护协会附近",
  "analyzer": "qq_max"
}

GET _analyze
{
  "text": "我家住在中国古建筑保护协会附近",
  "tokenizer": "qq_smart"
}
```

©2013–2024 Tencent Cloud. All rights reserved.

监控与告警
查看监控

Last updated：2023-04-07 10:33:11

操作场景

腾讯云 Elasticsearch Service 对运行中的 ES 集群，提供了多项监控指标，用以监测集群的运行情况，如存储、IO、CPU、内存使用率等。您可以根据这些指标实时了解集群服务的运行状况，针对可能存在的风险及时处理，保障集群的稳定运行。本文为您介绍通过 Elasticsearch Service 控制台查看集群监控的操作。

操作步骤

1. 登录 Elasticsearch Service 控制台，在集群列表单击集群 **ID/名称**，进入集群详情页。
2. 选择**集群监控**页，可以查看集群整体的运行情况，选择**指标分组**，支持拆分查看数据节点、冷数据节点、专用主节点的集群监控指标。
3. 选择**节点监控**页，可以查看集群内各节点的运行情况和性能指标。

集群监控

在集群监控页，可以进行告警策略设置，同时也可以看到集群的监控数据信息。可通过选择不同的时间范围、指标分组和时间粒度查看集群总体状态和性能指标。

> **说明**
> 也可通过 腾讯云可观测平台控制台 查看 ES 集群完整的监控指标。

©2013-2024 Tencent Cloud. All rights reserved.

节点监控

©2013–2024 Tencent Cloud. All rights reserved.

- **节点列表**
  展示集群各个节点部分实时运行指标。

- **单个节点状态**
  提供每个节点，各项指标详细的历史运行情况。并支持导出数据到本地。

部分指标含义及说明

ES 集群一般由多个节点构成，为反映集群整体的运行情况，部分监控指标提供了两类值：平均值、最大值。

- 平均值表示集群所有节点该指标值的平均数。
- 最大值表示集群所有节点该指标的最大值。

所有指标的统计周期均为1分钟，即每1分钟对集群的指标采集1次。具体各指标含义说明如下：

监控指标	统计方式	详情
集群健康状态	ES 集群健康状态：0：表示绿色，集群正常；1：表示黄色，告警，部分副本分片不可用；2：表示红色，异常，部分主分片不可用。	green：表示所有的主分片和副本分片都可用，集群处于最健康的状态。 yellow：表示所有的主分片均可用，但部分副本分片不可用，此时搜索结果仍然是完整的。但集群的高可用性在一定程度上受到影响，数据面临较高的丢失风险。在集群健康状态变为 yellow 后，应及时调查和定位问题，并修复，防止数据丢失。 red：表示至少一个主分片以及它的全部副本均不可用。集群处于 red 状态意味着已有部分数据丢失：搜索只能返回部分数据，而分配到丢失分片上的写入请求会返回异常。在集群健康状态变为 red 后，应及时定位异常分片，并进行修复。

©2013–2024 Tencent Cloud. All rights reserved.

平均磁盘使用率	每单位统计周期内（1分钟），集群各个节点的磁盘使用率的平均值。	磁盘使用率过高会导致数据无法正常写入。解决方法：及时清理无用的索引。对集群进行扩容，增加单节点的磁盘容量或增加节点个数。
最大磁盘使用率	每单位统计周期内（1分钟），集群各个节点中最大磁盘使用率。	-
平均 JVM 内存使用率	每单位统计周期内（1分钟），集群各个节点的 JVM 内存使用率的平均值。	该值过高会导致集群节点 GC 频繁，甚至有出现 OOM。导致该值过高的原因，一般是节点上 ES 处理任务超出节点 JVM 的负载能力。您需要注意观察集群正在执行的任务，或调整集群的配置。
最大 JVM 内存使用率	每单位统计周期内（1分钟），集群各个节点中最大 JVM 内存使用率。	-
平均 CPU 使用率	每单位统计周期内（1分钟），集群各个节点的 CPU 使用率的平均值。	当集群各节点处理的读写任务超出节点 CPU 的负载能力时，该指标就会过高，CPU 使用率过高会导致集群节点处理能力下降，甚至宕机。您可以从以下几点解决平均 CPU 使用率过高的问题： 观察该指标是持续性较高，还是临时飙升。若是临时飙升，确定是否有临时性复杂任务正在执行。 若该指标持续较高，分析业务对集群的读写操作是否可以优化，降低读写频率，减小数据量，从而减轻节点负载。 对于节点配置无法满足业务吞吐量的情况，建议对集群节点进行纵向扩容，提高单节点的负载能力。
最大 CPU 使用率	每单位统计周期内（1分钟），集群各个节点中最大 CPU 使用率。	-
集群1分钟平均负载	集群1分钟所有节点的平均负载 load_1m，指标来源：ES 节点状态 api：_nodes/stats/os/cpu/load_average/1m。	load_1m 过高时，建议降低集群负载或调大集群节点规格。
集群1分钟最大负载	集群1分钟所有节点的最大平均负载 load_1m。	-
平均写入延迟	写入延迟（index_latency），指单次 index 请求耗时（ms/次），集群平均写入延迟是统计周期内（1分钟）所有节点单次 index 请求耗时的平均值。 节点单次 index 请求耗时计算规则：每隔一个统计周期（1分钟）记录一次节点的两个指标，节点历史 index 总次数（_nodes/stats/indices/indexing/index_total），节点历史 index 总耗时（_nodes/stats/indices/indexing/index_time_in_millis），取相邻两次记录的差值，即一个周期内的绝对值并进行计算：index 耗时 / index 次数，得出统计周期内（1分钟）单次 index 平均耗时。	写入延迟，是指单个文档写入平均耗时。集群平均写入延迟，是指统计周期内，所有节点的写入用时的平均值。写入延迟过高时，建议调大节点规格或增加节点个数。
最大写入延迟	写入延迟（index_latency），指单次 index 请求耗时（ms/次），集群最大写入延迟是统计周期内（1分钟）所有节点中单次 index 请求耗时的最大值。 节点单次 index 请求耗时计算规则：见平均写入延迟。	-
平均查询延迟	查询延迟（search_latency），指单次查询请求耗时（ms/次），集群平均查询延迟是统计周期内（1分钟）所有节点单次查询请求耗时的平均值。 节点单次查询请求耗时计算规则：每隔一个统计周期（1分钟）记录一次节点的两个指标，节点历史查询总次数（_nodes/stats/indices/search/query_total），节点历史查询总耗时（_nodes/stats/indices/search/query_time_in_millis），取相邻两次记录的差值，即一个周期内的绝对值并进行计算：query 耗时 / query 次数，得出统计周期内（1分钟）单次查询平均耗时。	查询延迟，是指单个查询平均耗时。集群平均查询延迟，就是统计周期内，所有节点查询用时的平均值。查询延迟过高时，建议调大节点规格或增加节点个数。
最大查询延迟	查询延迟（search_latency），指单次查询请求耗时（ms/次），集群最大查询延迟是统计周期内（1分钟）所有节点中单次查	-

©2013-2024 Tencent Cloud. All rights reserved.

	询请求耗时的最大值。 节点单次查询请求耗时计算规则：见平均查询延迟。	
平均每秒写入次数	集群所有节点接收到的每秒 index 请求次数的平均值。节点每秒 index 请求次数计算规则：每隔一个统计周期（1分钟）记录一次节点历史 index 总次数（_nodes/stats/indices/indexing/index_total），取相邻两次记录的差值，即一个周期内的绝对值并进行计算：index 次数 / 60秒，得出统计周期内每秒 index 请求次数的平均值。	-
平均每秒完成查询次数	集群所有节点接收到的每秒查询请求次数的平均值。节点每秒查询请求次数计算规则：每个统计周期（1分钟）记录一次节点历史查询总次数（_nodes/stats/indices/search/query_total），取相邻两次记录的差值，即一个周期内的绝对值并进行计算：query 次数 / 60秒，得到统计周期内每秒查询请求次数的平均值。	-
写入拒绝率	单位周期内，集群写入请求被拒绝次数 ÷ 总写入请求次数，得到的比率。具体计算规则：每隔一个统计周期采集两个指标：历史写入请求被拒绝次数（5.6.4版本：_nodes/stats/thread_pool/bulk/rejected，6.4.3及之后版本：_nodes/stats/thread_pool/write/rejected）、历史写入请求次数（5.6.4版本：_nodes/stats/thread_pool/bulk/completed，6.4.3及之后版本：_nodes/stats/thread_pool/write/completed），取相邻两次记录的差值，即一个周期内的绝对值并进行计算，写入请求拒绝次数 / 写入请求完成次数。	写入 QPS 过大，CPU、内存、磁盘使用率过高时，可能会造成集群写入拒绝率会增加。一般地，是集群当前配置无法满足业务写操作需求。对于节点配置过低的场景，可以通过提高节点规格或降低写入操作次数来解决。对于磁盘使用率过高的情况，可以通过扩容集群磁盘或删除无用数据来解决。
查询拒绝率	单位周期内，集群查询请求被拒绝次数 ÷ 总查询请求次数，得到的比率。具体计算规则：每隔一个统计周期采集两个指标：历史查询请求被拒绝次数（_nodes/stats/thread_pool/search/rejected）、历史查询请求次数（_nodes/stats/thread_pool/search/completed），取相邻两次记录的差值，即一个周期内的绝对值并进行计算：查询请求拒绝次数 / 查询请求完成次数。	写入 QPS 过大，CPU、内存使用率过高，可能会造成集群查询拒绝率上升。一般地，是集群当前配置无法满足业务读操作需求，该值过高时建议对集群调大集群节点配置，提高集群节点的处理能力。
集群总文档数	集群中的文档总数，此数字可能包括来自嵌套字段的文档。计算规则：ES 集群文档数 API：_cluster/stats/indices/docs/count，详情请参见 Cluster stats API 。	-
自动快照备份状态	集群开启自动快照备份后的备份结果：0：未开启自动备份；1：自动备份正常；-1：自动备份失败。	自动快照备份，会把集群的数据定时备份到 COS，以便在需要的时候可以对数据进行恢复，从而更全面保障数据安全，建议开启，更多信息可查看：自动快照备份 。
分片中最大文档数	集群中全部索引下分片，按文档数统计后的单个分片最大值。	-
集群最大分片存储量	集群中全部索引分片，按存储量统计后的单个分片最大值。	-
集群最大分片文档 delete 次数	集群中全部索引下分片文档数中，单个分片中被标记为已删除状态的文档数量经统计后，各个分片中的最大值。	-
集群最大分片文档 delete 次数占比	集群中全部索引下分片文档数中，单个分片中被标记为已删除状态的文档数量相较于该分片文档总数的比值，各个分片下的比值中的最大值。	-
活跃查询 context 数量	集群下全部节点统计活跃查询 context 的均值。	-

©2013-2024 Tencent Cloud. All rights reserved.

查询任务耗时90分位	每单位统计周期内，集群最大节点执行查询任务90分位时延。	-
写入任务耗时90分位	每单位统计周期内，集群最大节点执行写入任务90分位时延。	-

©2013-2024 Tencent Cloud. All rights reserved.

配置告警

Last updated：2024-09-04 15:14:01

操作场景

腾讯云 ES 提供一些关键指标的配置告警功能，配置告警可帮助您及时发现集群问题并进行处理。本文为您介绍通过控制台配置告警的操作。

操作步骤

查看集群是否已配置告警

1. 登录 Elasticsearch Service 控制台，在集群列表单击集群 ID/名称，进入集群详情页。
2. 在左侧菜单 ES 集群管理界面，选择**集群监控 > 告警策略**，可查看集群是否已经配置了告警。

> **警告：**
> 如果您暂未配置告警，强烈建议进行告警策略配置，以便及时获取并处理集群运行的状况及风险，保障服务的稳定。

3. 点击**去查看**按钮。

4. 进入策略详情页面，您可以查看告警策略的基本信息，同时也可以修改策略名称和备注。

> **说明**
> 您也可登录 腾讯云可观测平台控制台 ，在**告警管理 > 策略管理**，通过筛选策略和产品，查询某个集群是否已经配置了告警策略。

自定义告警配置

1. 在管理告警策略界面，点击**返回上一级**按钮。

©2013-2024 Tencent Cloud. All rights reserved.

2. 选择**告警管理** > **策略管理**，并单击**新建策略**。

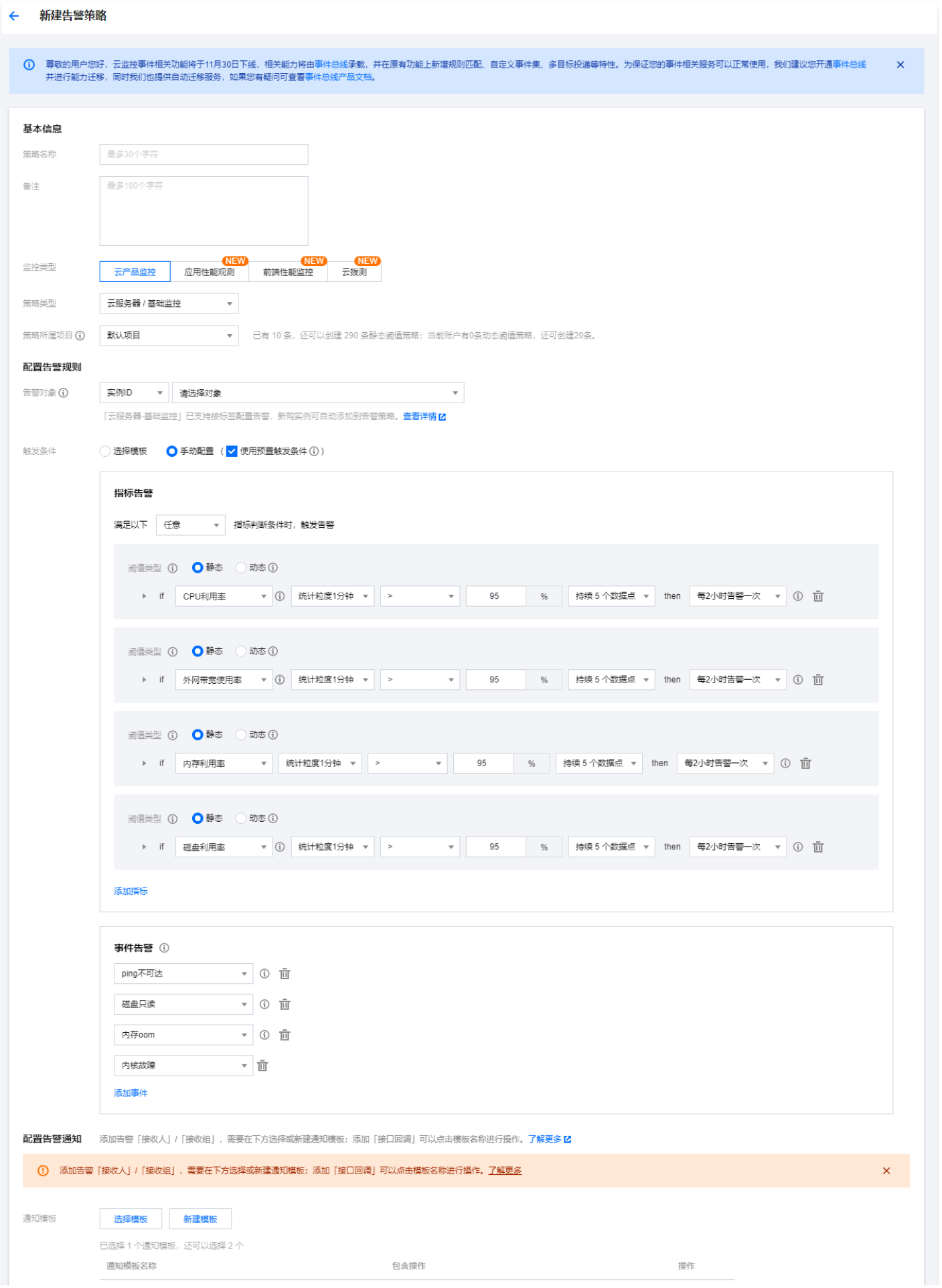
3. 在新建策略页面，配置策略参数。
   - **策略类型**：选择 **Elasticsearch 服务**。
   - **告警对象**：选择需要配置告警策略的集群。
   - **触发条件**：支持触发条件模板和配置触发条件，默认选择自定义配置触发条件，自定义配置参见以下说明，新建模板参见 新建触发条件模板 。

     说明
     - 指标：例如“CPU 使用率”，统计周期为1分钟或5分钟。因 ES 集群的各项指标都是1分钟采集1次，所以选择统计周期是1分钟时，当集群出现一次超过阈值就会触发告警，如果选择5分钟，则5分钟内，连续超过阈值才会触发告警。
     - 告警频次：例如“每30分钟警告一次”，指每30分钟内，连续多个统计周期指标都超过了阈值，如果有一次告警，30分钟内就不会再次进行告警，直到下一个30分钟，如果指标依然超过阈值，才会再次告警。

   - **告警渠道**：选择接收组、有效时段、接收渠道。
4. 配置完成后，单击**完成**，返回到**告警策略**列表，即可查看到刚配置的告警策略。

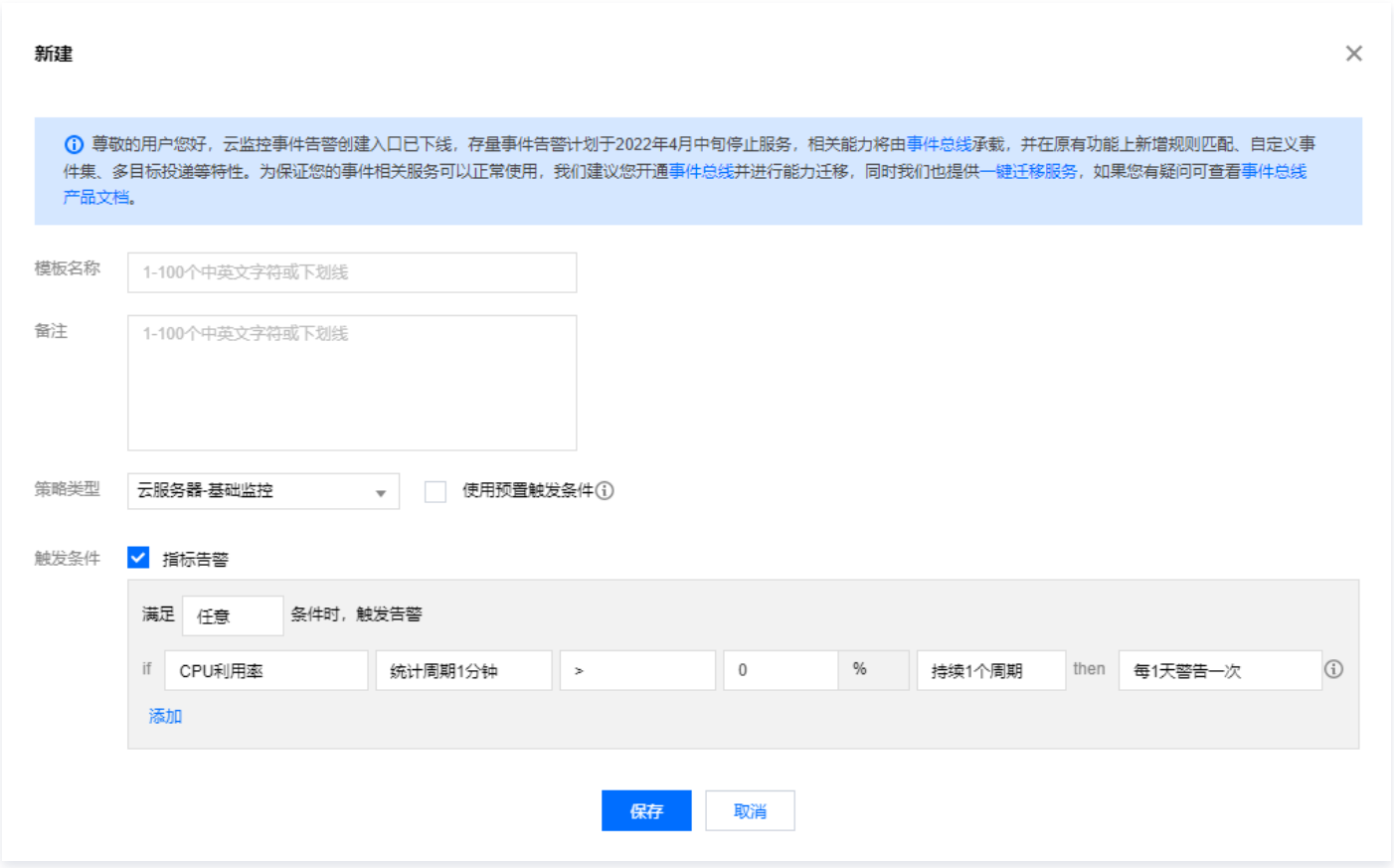
说明
告警策略更详细配置教程可参见 告警策略配置 。

©2013–2024 Tencent Cloud. All rights reserved.


新建告警策略
尊敬的用户您好，云监控事件相关功能将于11月30日下线，相关能力将由事件总线承载，并在原有功能上新增规则匹配、自定义事件集、多目标投递等特性。为保证您的事件相关服务可以正常使用，我们建议您开通事件总线并进行能力迁移，同时我们也提供自动迁移服务，如果您有疑问可查看事件总线产品文档。
基本信息
策略名称
最多30个字符
备注
最多100个字符
监控类型
云产品监控
应用性能观测
前端性能监控
云拨测
NEW
策略类型
云服务器 / 基础监控
策略所属项目
默认项目
已有 10 条，还可以创建 290 条静态阈值策略；当前账户有0条动态阈值策略，还可创建20条。
配置告警规则
告警对象
实例ID
请选择对象
「云服务器-基础监控」已支持按标签配置告警，新购实例可自动添加到告警策略。查看详情
触发条件
选择模板
手动配置
使用预置触发条件
指标告警
满足以下
任意
指标判断条件时，触发告警
阈值类型
静态
动态
if
CPU利用率
统计粒度1分钟
>
95
%
持续 5 个数据点
then
每2小时告警一次
外网带宽使用率
内存利用率
磁盘利用率
添加指标
事件告警
ping不可达
磁盘只读
内存oom
内核故障
添加事件
配置告警通知
添加告警「接收人」/「接收组」，需要在下方选择或新建通知模板；添加「接口回调」可以点击模板名称进行操作。了解更多
通知模板
选择模板
新建模板
已选择 1 个通知模板，还可以选择 2 个
通知模板名称
包含操作
操作

新建触发条件模板

1. 在触发条件页面，单击**新增触发条件模板**。
2. 在触发条件模板页，单击**新建**。
3. 在新建模板页，配置策略类型。
   - **策略类型**：选择 **Elasticsearch 服务**。
   - **使用预置触发条件**：勾选此选项，会出现系统建议的告警策略。
4. 确认无误后，单击**保存**。



5. 返回新建告警策略页，单击 ，即可出现刚配置的告警策略模板。

©2013–2024 Tencent Cloud. All rights reserved.

监控告警配置建议

Last updated：2024-10-08 14:18:21

腾讯云 ES 不仅为运行中的 ES 集群提供了多项监控指标，用于监测集群的运行情况，还提供了一些关键指标的配置告警功能，帮助您及时发现集群问题并进行处理。具体使用方法可参考 查看监控 和 配置告警 。

本文为您介绍在使用 ES 集群过程中需要重点关注的一些指标及其告警建议配置：

指标	告警建议配置	详细说明
集群健康状态	统计周期1分钟，≥1，持续5个周期，每30分钟告警一次	集群健康状态取值为： • 0：绿色，表示集群所有主分片和副本分片都可用，集群处于最健康的状态。 • 1：黄色，表示所有的主分片均可用，但存在不可用副本分片。此时，搜索结果仍然是完整的，但集群的高可用性在一定程度上受到影响，数据面临较高的丢失风险。 • 2：红色，表示至少一个主分片以及它的全部副本分片均不可用。集群处于红色状态意味着已有部分数据不可用，搜索只能返回部分数据，而分配到丢失分片上的请求会返回异常。 • 集群健康状态是集群当前运行情况的最直接体现，当集群处于黄色或红色状态时，应立即排查产生原因，并及时修复，防止数据丢失和服务不可用。
平均磁盘使用率	统计周期1分钟，>80%，持续5个周期，每30分钟告警一次	平均磁盘使用率表示集群各节点磁盘使用率的平均值。磁盘使用率过高会导致节点没有足够的磁盘空间容纳分配到该节点上的分片，从而导致创建索引，添加文档等基本操作执行失败。建议在平均磁盘使用率超过75%时及时清理数据或扩容集群。另外可以参考 使用 Curator 在腾讯云 Elasticsearch 中自动删除过期数据 ，为集群配置定时清理任务。
平均 JVM 内存使用率	统计周期1分钟，>85%，持续5个周期，每30分钟告警一次	平均 JVM 堆内内存使用率表示集群各节点 JVM 内存使用率的平均值。JVM 内存使用率过高会导致读写操作被拒绝，集群 GC 频繁，甚至出现 OOM 等问题。当发现 JVM 内存使用率超过阈值时，建议通过纵向扩容的方式提高集群节点的规格。
平均 CPU 使用率	统计周期1分钟，>90%，持续5个周期，每30分钟告警一次	平均 CPU 使用率表示集群各节点 CPU 使用率的平均值。该值过高会导致集群节点处理能力下降，甚至宕机。发现 CPU 过高时，应根据集群当前节点配置情况和业务情况，提高节点规格或降低业务请求量。
bulk 拒绝率	统计周期1分钟，>0%，持续1个周期， 每30分钟告警一次	bulk 拒绝率表示单周期内集群执行 bulk 操作被拒绝次数占 bulk 总操作次数的百分比。当 bulk 拒绝率大于0%，即出现 bulk 拒绝时，说明集群已经达到了 bulk 操作处理能力的上限，或集群出现异常，应及时排除出现 bulk 拒绝的原因并及时解决，否则会影响业务的 bulk 操作，甚至出现数据丢失情况。
查询拒绝率	统计周期1分钟，>0%，持续1个周期，每30分钟告警一次	查询拒绝率表示单周期内集群执行查询操作被拒绝次数占查询总操作数的百分比。当查询拒绝率大于0%，即出现查询拒绝时，说明集群已经达到了查询操作处理能力的上限，或集群出现异常，应及时排查出现查询拒绝的原因并及时解决，否则会影响业务的查询操作。

©2013-2024 Tencent Cloud. All rights reserved.

日志查询
查询集群日志

Last updated：2024-10-16 11:20:11

本文为您介绍腾讯云 Elasticsearch Service 集群运行日志的使用说明。用户可以通过集群的运行日志，了解集群的运行状况，定位问题，辅助集群的应用开发和运维。

查询集群日志

1. 登录 Elasticsearch Service 控制台，单击集群 ID/名称，进入集群详情页。
2. 选择日志页签，可查看集群的运行日志。
   - ES 共有四种类型的日志：主日志、搜索慢日志、索引慢日志和 GC 日志，日志内容包括日志时间、日志级别，以及具体信息等。
   - ES 默认提供集群7天内的运行日志，按时间倒序展示，用户可以按时间和关键字进行查询。此外，用户也可调用 ES API 调整日志相关配置，例如对于慢日志，设定查询或索引数据时，认为响应较慢的时间阈值。
3. 在日志页面的搜索框，可以按照时间范围和关键字查询相关日志，关键字查询语法同 lucene 查询语法一致。
   - 输入关键词查询，例如：“YELLOW”。
   - 指定字段设置关键词，例如：`message:YELLOW`。
   - 多个条件组合：`level:INFO and ip:10.0.1.1`，可以查询相关日志。

日志说明

主日志

展示集群运行产生日志的时间、级别、信息等，有 INFO、WARN、DEBUG 等不同级别。

```
2019-2-14 08:00:00 10.0****
INFO
[o.e.c.r.a.AllocationService] [1550199698000783811] Cluster health status changed from [YELLOW] to [GREEN]
(reason: [shards started [[filebeat-2019.02.19][2]] ...]).

2019-2-14 02:30:02 10.0****
DEBUG
[o.e.a.a.i.d.TransportDeleteIndexAction] [1550199698000783811] failed to delete indices [[[filebeat-
2019.02.09/7VZM6Fa-Twmj8pVaAyyrxg]]]
org.elasticsearch.index.IndexNotFoundException: no such index
    at org.elasticsearch.cluster.metadata.MetaData.getIndexSafe(MetaData.java:475) ~[elasticsearch-
5.6.4.jar:5.6.4]
    at
org.elasticsearch.cluster.metadata.MetaDataDeleteIndexService.lambda$deleteIndices$0(MetaDataDeleteIndexSe
rvice.jav
```

慢日志

慢日志的目的是捕获超过指定时间阈值的查询和索引请求，用于跟踪分析由用户产生的很慢的请求。

©2013-2024 Tencent Cloud. All rights reserved.

```
2018-10-28 12:04:17
WARN
[index.indexing.slowlog.index] [1540298502000001009] [pmc/wCALr6BfRm-sr3qOQuGX
Xw] took[18.6ms], took_millis[18], type[articles], id[AWa41-J9c0s1mOPvR6F3], routing[] , source[]
```

开启和调整慢日志

默认情况，慢日志不开启。开启慢日志需要定义具体动作（query、fetch 或 index）、期望的事件记录等级（INFO、WARN、DEBUG 等）、以及时间阈值。用户可以根据业务场景，开启和调整相关配置。

如需开启慢日志，可在集群详情页的右上角单击 **Kibana** 进入 Kibana 页面，通过 Dev Tools 调用 Elasticsearch 相关 API，或通过客户端调用配置修改 API。

配置所有索引：

```
PUT */_settings
{
"index.indexing.slowlog.threshold.index.debug" : "5ms",
"index.indexing.slowlog.threshold.index.info" : "50ms",
"index.indexing.slowlog.threshold.index.warn" : "100ms",
"index.search.slowlog.threshold.fetch.debug" : "10ms",
"index.search.slowlog.threshold.fetch.info" : "50ms",
"index.search.slowlog.threshold.fetch.warn" : "100ms",
"index.search.slowlog.threshold.query.debug" : "100ms",
"index.search.slowlog.threshold.query.info" : "200ms",
"index.search.slowlog.threshold.query.warn" : "1s"
}
```

配置单个索引：

```
PUT /my_index/_settings
{
"index.indexing.slowlog.threshold.index.debug" : "5ms",
"index.indexing.slowlog.threshold.index.info" : "50ms",
"index.indexing.slowlog.threshold.index.warn" : "100ms",
"index.search.slowlog.threshold.fetch.debug" : "10ms",
"index.search.slowlog.threshold.fetch.info" : "50ms",
"index.search.slowlog.threshold.fetch.warn" : "100ms",
"index.search.slowlog.threshold.query.debug" : "100ms",
"index.search.slowlog.threshold.query.info" : "200ms",
"index.search.slowlog.threshold.query.warn" : "1s"
}
```

GC 日志

ES 默认开启 GC 日志，以下两条具体的 GC 日志，分别展示日志的时间、节点 IP、日志级别等。

```
2019-2-14 20:48:22
10.0.***
INFO
[o.e.m.j.JvmGcMonitorService] [1550199698000783711] [gc][380573] overhead, spent [307ms] collecting in the
last [1s]
2019-2-14 10:04:09
10.0.***
WARN
[o.e.m.j.JvmGcMonitorService] [1550199698000784111] [gc][341943] overhead, spent [561ms] collecting in the
last [1s]
```

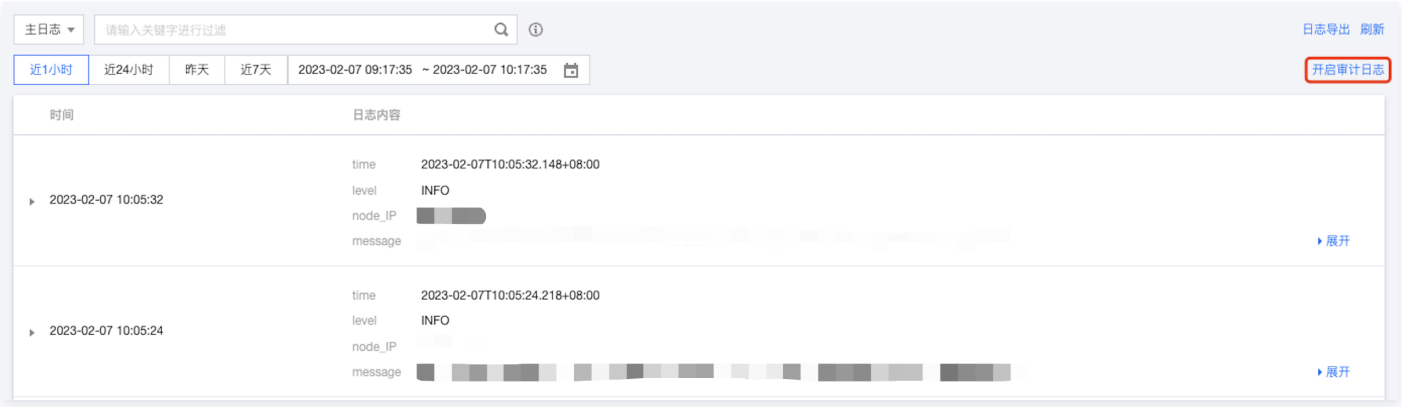
审计日志

©2013-2024 Tencent Cloud. All rights reserved.

> **注意**
> 审计日志为 X-Pack 高级特性能力，仅在白金版支持。

审计日志主要展示 Elasticsearch 实例对应的增、删、改、查等操作产生的日志，您可以通过如下步骤开启审计日志：

1. 在**日志**页面，单击右侧的**开启审计日志**。



2. 在出现的弹窗中，勾选对应提示，并单击**确认**。

> **注意**
> - 开启审计日志采集，文件将输出到当前 ES 集群中，并使用 security_audit_log-* 开头的索引名称，您可以在 Kibana 中查询到该集群的审计日志。
> - ES 6.8.2及以上版本，数据默认保存3天，如需存储更长时间，您可修改对应生命周期管理策略；ES 6.4.3版本，数据默认永久保存，请及时进行清理操作；ES 5.6.4版本，不支持审计日志功能。
> - 若您需要修改采集的审计事件类型，可参见 文档 。
> - 开启或关闭审计日志采集会触发集群重启，建议在集群负载不高时操作。

3. 确认后，集群会进行重启，您可在**变更记录**中查看进度。重启成功后，即可开启审计日志采集。

> **说明**
> 审计日志信息会占用集群的磁盘空间，同时也会影响性能。如果您不需要查看审计日志，可使用同样的方式关闭审计日志采集功能。

4. 进入 Kibana Discover 界面，找到对应索引，即可查看审计日志。

©2013-2024 Tencent Cloud. All rights reserved.

集群 IP 溯源

Last updated：2024-08-08 11:48:31

本文介绍如何在腾讯云 Elasticsearch Service 上，开启 IP 溯源日志，进行请求来源的 IP 溯源。以帮助用户解决无法确定有哪些客户端在使用的情况。

使用限制

IP 溯源日志功能为腾讯云开发的ES内核能力，仅在腾讯云 Elasticsearch Service 的 PaaS 模式支持。当前支持的版本号为 6.8.2 / 7.10.1 / 7.14.2。

> **说明**
> 2023年6月1日后创建的集群可直接使用该能力。在此之前创建的集群需要在控制台滚动重启集群，以将内核更新至最新才可使用。

设置 IP 溯源日志开启

集群维度动态配置开关

```
# req/rsp均支持独立开启、关闭
"http.tracer.request.enable"
"http.tracer.response.enable"

# 支持使用uri前缀过滤
"http.tracer.include"
"http.tracer.exclude"

# 支持ip过滤
"http.tracer.remote_ip.include"
"http.tracer.remote_ip.exclude"

# body打印支持独立开启、关闭
"http.tracer.request.body.enable"
"http.tracer.response.body.enable"
```

> **说明**
> - 强烈建议您尽可能多的设置过滤条件，避免日志打印过多，影响集群性能，甚至引发磁盘写满风险。
> - 避免长期打开此开关，随用随开，用完及时关闭。

关闭示例

```
PUT _cluster/settings
{
  "transient": {
    "http.tracer.request.enable": false,
    "http.tracer.response.enable": false,
    "http.tracer.include": null,
    "http.tracer.exclude": null,
    "http.tracer.remote_ip.include": null,
    "http.tracer.remote_ip.exclude": null,
    "http.tracer.request.body.enable": false,
    "http.tracer.response.body.enable": false
  }
}

——关闭ip溯源日志——
不打印
```

开启示例

©2013-2024 Tencent Cloud. All rights reserved.

```
PUT _cluster/settings
{
  "transient": {
    "http.tracer.request.enable": true,
    "http.tracer.response.enable": true,
    "http.tracer.include": [
      "/.kibana/_search"
    ]
  }
}

---日志，打印req，rsp---
[2023-04-17T15:39:26,952][INFO ][o.e.h.HttpTracer         ] [1681467780000360532] [29940][null][GET]
[/.kibana/_search] received request from [Netty4HttpChannel{localAddress=/9.10.64.164:9200,
remoteAddress=/9.2.1.38:41696}]
[2023-04-17T15:39:26,959][INFO ][o.e.h.HttpTracer         ] [1681467780000360532] [29940][null][OK]
[application/json; charset=UTF-8][2600808] sent response to
[Netty4HttpChannel{localAddress=/9.10.64.164:9200, remoteAddress=/9.2.1.38:41696}] success [true]
---公网访问情况---
[2023-04-25T11:36:13,197][INFO ][o.e.h.HttpTracer         ] [1681703897000800432] [74][null][GET]
[/.kibana/_search] received request from [Netty4HttpChannel{localAddress=/9.10.66.6:9200,
remoteAddress=/9.10.65.111:53349}], body [], x-forwarded-for [113.108.77.60]
[2023-04-25T11:36:13,238][INFO ][o.e.h.HttpTracer         ] [1681703897000800432] [74][null][OK]
[application/json; charset=UTF-8][2607129] sent response to
[Netty4HttpChannel{localAddress=/9.10.66.6:9200, remoteAddress=/9.10.65.111:53349}] success [true]
```

```
PUT _cluster/settings
{
  "transient": {
    "http.tracer.request.enable": true,
    "http.tracer.response.enable": false,
    "http.tracer.request.body.enable": true,
    "http.tracer.include": [
      "/.kibana/_search"
    ]
  }
}

---日志，打印body，不打印rsp---
[2023-04-25T10:36:08,090][INFO ][o.e.h.HttpTracer         ] [1681703897000800432] [148][null][GET]
[/.kibana/_search] received request from [Netty4HttpChannel{localAddress=/9.10.66.6:9200,
remoteAddress=/9.2.1.38:53310}], body [{"query":{"query_string":{"query":"*"}}}]
```

```
PUT _cluster/settings
{
  "transient": {
    "http.tracer.request.enable": true,
    "http.tracer.response.enable": false,
    "http.tracer.request.body.enable": true,
    "http.tracer.remote_ip.exclude": "9.2.1.38",
    "http.tracer.include": [
      "/.kibana/_search"
    ]
  }
}

---日志，过滤指定ip---
```

©2013–2024 Tencent Cloud. All rights reserved.

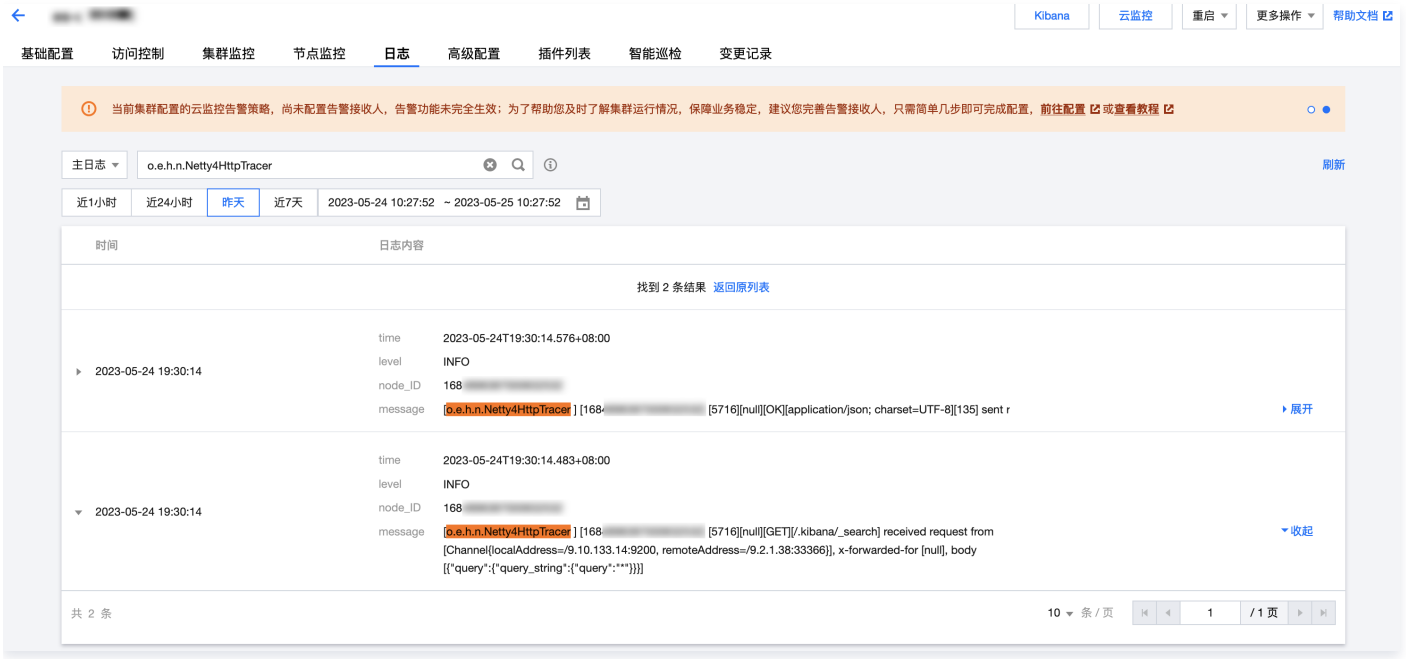
不打印

注意事项：

1. 公网访问溯源
   1.1 内网访问情况，remoteAddress 就是真实 IP；公网访问情况，remoteAddress 是负载均衡的网关 IP。
   1.2 取 header 中的 x-forwarded-for 可以拿到真实的公网访问 IP，req 日志会输出。
2. IP 过滤
   2.1 不填写则不过滤，可数组形式填写多个 IP 过滤。
   2.2 不支持通过公网 IP 过滤（x-forwarded-for）。
3. http.tracer.include 定义的 uri 路径，末尾可以尽量加 * 来增加匹配，很多查询 uri 后面都会带？参数。

在腾讯云控制台查看 IP 溯源日志

1. 登录腾讯云 Elasticsearch 控制台，单击**集群名称**访问目标集群，跳转至**日志**页面。
2. 输入关键字查找 IP 溯源日志，6.8.2 版本输入 "o.e.h.n.Netty4HttpTracer" 查找，7.10.1/7.14.2 版本输入 "o.e.h.HttpTracer" 查找。



©2013-2024 Tencent Cloud. All rights reserved.

数据备份
自动快照备份

Last updated：2024-03-13 16:25:31

腾讯云 ES 提供自动备份，自动创建集群的主索引分片的快照并备份到 COS，进而根据需要将备份数据恢复到集群。

> **说明**
> 此功能目前为免费试用。

备份说明

- 腾讯云 ES 自动快照备份，只保留最近7天的快照数据。
- 自动快照备份的数据只能用于恢复到原集群，如果需要跨集群恢复，请参考 手动快照 。
- 默认情况下，自动快照备份在凌晨进行，建议您根据业务需求选择集群访问压力不大的时间进行。
- 集群第一个快照是集群数据的完整拷贝，执行时间视具体数据量而定，之后的快照保留的是已存快照和新数据之间的增量差异。
- 集群健康状态为红色时，自动快照服务将停止创建，建议您关注集群的健康状态。

操作步骤

开启自动快照备份

1. 登录 Elasticsearch Service 控制台，在集群列表，单击集群 ID/名称进入集群详情页。
2. 在“备份管理”页面，单击**自动备份设置**，可以开启自动快照备份以及配置自动快照备份的时间。

查看快照存储库

在集群详情页单击 **Kibana**，进入 Kibana 控制台，在“Dev Tools”页可以通过 ES 的 API 查看集群的所有快照存储库。

- 查看集群的快照存储库。

```
GET _snapshot?pretty
```

如果只有自动快照，将得到如下的返回信息：

```
{
"ES_AUTO_BACKUP": {
  "type": "cos",
  "settings": {
    "bucket": "es-ap-guangzhou",
    "base_path": "/es_backup/es-2s8x1b9u",
    "chunk_size": "500mb",
    "region": "ap-guangzhou",
```

©2013-2024 Tencent Cloud. All rights reserved.

```
    "compress": "true"
  }
}
}
```

- 查看自动快照存储库中的快照信息。

```
GET _snapshot/ES_AUTO_BACKUP/_all?pretty
```

返回结果如下：

```
如果没有快照，列表为空。
{
"snapshots": []
}
{
"snapshots": [
  {
    "snapshot": "es-2s8x1b9u_20181220",
    "uuid": "gsXPyWb1SNOlTuj3eNs2gA",
    "version_id": 5060499,
    "version": "5.6.4",
    "indices": [
      ".kibana"
    ],
    "state": "SUCCESS",
    "start_time": "2018-12-20T08:00:12.336Z",
    "start_time_in_millis": 1545292812336,
    "end_time": "2018-12-20T08:00:12.945Z",
    "end_time_in_millis": 1545292812945,
    "duration_in_millis": 609,
    "failures": [],
    "shards": {
      "total": 1,
      "failed": 0,
      "successful": 1
    }
  }
]
}
```

恢复数据

```
POST _snapshot/ES_AUTO_BACKUP/es-2s8x1b9u_20181220/_restore
```

©2013-2024 Tencent Cloud. All rights reserved.

使用 COS 进行备份及恢复

Last updated：2024-02-20 19:49:41

创建仓库

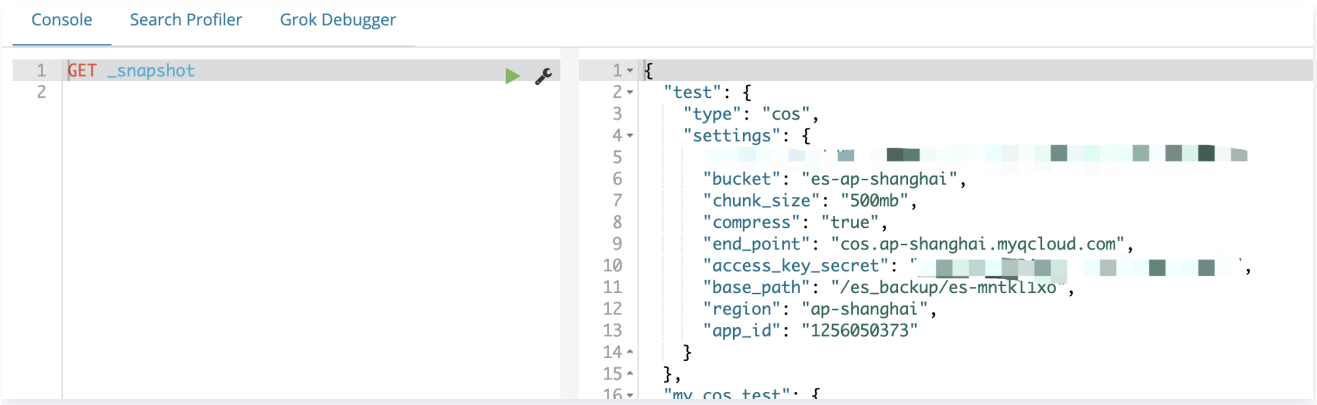
您可以通过如下命令创建仓库：

```
PUT _snapshot/my_cos_backup
{
    "type": "cos",
    "settings": {
        "app_id": "xxxxxxx",
        "access_key_id": "xxxxxx",
        "access_key_secret": "xxxxxxx",
        "bucket": "xxxxxx",
        "region": "ap-guangzhou",
        "compress": true,
        "chunk_size": "500mb",
        "base_path": "/"
    }
}
```

- app_id：腾讯云账号 APPID。
- access_key_id：腾讯云 API 密钥 SecretId。
- access_key_secret：腾讯云 API 密钥 SecretKey。
- bucket：COS Bucket 名字，**名字不能带 `-{appId}` 后缀**，创建 COS 存储桶的详细步骤请参见 创建存储桶 。
- region：COS Bucket 地域，**此地域必须与 ES 集群为同一地域**。地域编码可参考 地域和可用区 。
- compress：默认为 true，对索引元数据存储进行压缩。
- base_path：备份目录。

列出仓库信息

您可通过 `GET _snapshot` 获取仓库信息，也可使用 `GET _snapshot/my_cos_backup` 获取指定的仓库信息。
下图中右侧部分即仓库信息，例如： "base_path": "/es_backup/es-mntkl1xo","end_point": "cos.ap-shanghai.myqcloud.com"。



创建快照备份

备份所有索引

将 ES 集群内所有索引备份到 `my_cos_backup` 仓库下，并命名为 `snapshot_1` 。

```
PUT _snapshot/my_cos_backup/snapshot_1
```

©2013-2024 Tencent Cloud. All rights reserved.

这个命令会立刻返回，并在后台异步执行直到结束。如果希望创建快照命令阻塞执行，可以添加 `wait_for_completion` 参数。命令执行的时间与索引大小相关。

```
PUT _snapshot/my_cos_backup/snapshot_1?wait_for_completion=true
```

备份指定索引

您可以在创建快照的时候指定要备份的索引。参数 indices 的值为多个索引的时候，需要用 `,` 隔开且不能有空格。

```
PUT _snapshot/my_cos_backup/snapshot_2
{
    "indices": "index_1,index_2"
}
```

查询快照

查询单个快照信息：

```
GET _snapshot/my_cos_backup/snapshot_1
```

该命令会返回快照的相关信息：

> **说明**
> 当信息中的 `state` 字段为 `SUCCESS` 时，说明快照备份完成。

```
{
    "snapshots": [
        {
            "snapshot": "snapshot_1",
            "uuid": "zUSugNiGR-OzH0CCcgcLmQ",
            "version_id": 5060499,
            "version": "5.6.4",
            "indices": [
                "index_1",
                "index_2"
            ],
            "state": "SUCCESS",
            "start_time": "2018-05-04T11:44:15.975Z",
            "start_time_in_millis": 1525434255975,
            "end_time": "2018-05-04T11:45:29.395Z",
            "end_time_in_millis": 1525434329395,
            "duration_in_millis": 73420,
            "failures": [],
            "shards": {
                "total": 3,
                "failed": 0,
                "successful": 3
            }
        }
    ]
}
```

删除快照

删除指定的快照：

©2013–2024 Tencent Cloud. All rights reserved.

```
DELETE _snapshot/my_cos_backup/snapshot_1
```

> **注意**
> 如果存在还未完成的快照，删除快照命令依旧会执行，并取消未完成快照的创建进程。

从快照恢复

1. 将快照中备份的所有索引都恢复到 ES 集群中。

```
POST _snapshot/my_cos_backup/snapshot_1/_restore
```

- 这里是恢复所有的索引数据，通常会有一些系统索引已经存在，那就会出现异常的情况。
- 如果 snapshot_1 包括5个索引，则这5个索引都会被恢复到 ES 集群中。
- 您还可以使用附加的选项对索引进行重命名。该选项允许您通过模式匹配索引名称，并通过恢复进程提供一个新名称。如果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进行其他操作，则可以使用该选项。

2. 从快照里恢复单个索引并提供一个替换的名称。

```
POST /_snapshot/my_cos_backup/snapshot_1/_restore
{
 "indices": "index_1",
 "rename_pattern": "index_(.+)",
 "rename_replacement": "restored_index_$1"
}
```

- indices：只恢复 index_1 索引，忽略快照中存在的其他索引。
- rename_pattern：查找所提供的模式能匹配上的正在恢复的索引。
- rename_replacement：将匹配的索引重命名成替代的模式。

查询快照恢复状态

可通过执行 `_recovery` 命令，查看快照恢复的状态并监控快照恢复的进度。

1. 您可以在恢复的指定索引中单独调用下述 API。

```
GET index_1/_recovery
```

2. 该命令会返回指定索引各分片的恢复状况。

```
{
 "sonested": {
    "shards": [
        {
            "id": 1,
            "type": "SNAPSHOT",
            "stage": "INDEX",
            "primary": true,
            "start_time_in_millis": 1525766148333,
            "total_time_in_millis": 8718,
            "source": {
                "repository": "my_cos_backup",
                "snapshot": "snapshot",
                "version": "5.6.4",
                "index": "index_1"
            },
            "target": {
                "id": "xxxxxxxxxxx",
                "host": "10.0.0.6",
```

©2013-2024 Tencent Cloud. All rights reserved.

```
                "transport_address": "10.0.0.6:9300",
                "ip": "10.0.0.6",
                "name": "node-1"
            },
            "index": {
                "size": {
                    "total_in_bytes": 1374967573,
                    "reused_in_bytes": 0,
                    "recovered_in_bytes": 160467084,
                    "percent": "11.7%"
                },
                "files": {
                    "total": 132,
                    "reused": 0,
                    "recovered": 20,
                    "percent": "15.2%"
                },
                "total_time_in_millis": 8716,
                "source_throttle_time_in_millis": 0,
                "target_throttle_time_in_millis": 0
            },
            "translog": {
                "recovered": 0,
                "total": 0,
                "percent": "100.0%",
                "total_on_start": 0,
                "total_time_in_millis": 0
            },
            "verify_index": {
                "check_index_time_in_millis": 0,
                "total_time_in_millis": 0
            }
        }
    ]
  }
}
```


- type：描述恢复的本质，该分片是在从一个快照恢复。
- source：哈希描述作为恢复来源的特定快照和仓库。
- percent：描述恢复的状态。该特定分片目前已经恢复了94%的文件，即将全部恢复。

输出会列出所有目前正在恢复的索引，以及这些索引里的所有分片。每个分片里会有启动/停止时间、持续时间、恢复百分比、传输字节数等统计值。

取消快照恢复

```
DELETE /restored_index_1
```

如果 restored_index_1 正在恢复中，则该删除命令会停止恢复，同时删除所有已经恢复到集群中的数据。

©2013-2024 Tencent Cloud. All rights reserved.

升级
升级 ES 集群

Last updated：2024-10-08 14:18:21

腾讯云 Elasticsearch Service 提供了版本升级的功能，支持 ES 从低版本到高版本的升级，以及高级特性的升级，可根据业务需要对集群进行升级，实现业务的无缝过渡。

支持的升级操作

ES 支持以下两种类型的升级。

1. Elasticsearch 版本升级

源 Elasticsearch 版本	目标 Elasticsearch 版本
低版本（如6.8版本）	高版本（如7.10版本）

2. 高级特性升级

源高级特性版本	目标高级特性版本
开源版	基础版、白金版
基础版	白金版

高级特性版本说明：
基础版和白金版集成了官方高级商业特性插件（原 X-Pack），包含安全（Security）、SQL、机器学习（Machine Learning）、监控（Monitor）等高级功能，其中基础版只包含 SQL 的一部分和监控，白金版则包含所有高级特性，详细介绍可查看 高级特性（X-Pack）。

> **说明**
> - 以上两种类型的升级一次只能操作一种，不能同时进行。低版本/开源版升级到高版本时，可同时选择升级到基础版。
> - 5.x 版本只有开源版，没有基础版和白金版。

关于升级的重启

1. 对于升级重启情况，在执行升级操作时，集群可能：
- 不重启
- 滚动重启（节点依次重启，期间服务可以正常访问，但性能可能受到部分影响，建议在集群负载不高时进行）
- 全量重启（所有节点完全关闭后重启，期间服务不可访问，需谨慎选择）

2. 根据您的选择，升级界面对这些情况均会给出提示说明。
对于集群全量重启的情况，因为会涉及业务不可访问，这里做一个特别说明：
- 集群全量重启与 ES 集群用户登录认证 从关闭到开启有关（可极大提升集群访问安全性），当您的集群从未开启此功能状态变化为开启此功能状态时，根据 Elastic 官方的设计要求，集群需要全量重启，期间集群不可用，因此建议您选择不影响业务的时机来慎重开启此功能。

> **警告**
> 从关闭到开启 ES 集群用户登录认证，您还需要提前改造业务代码，通过 API 访问集群，在调用 API/SDK 时传入用户名和密码参数，否则升级后集群无法正常访问。

3. 对 ES 集群用户登录认证功能的版本支持情况说明如下：
- 开源版：采用腾讯云自研 O-Pack 鉴权认证组件，默认开启。
- 6.8及以上基础版支持 X-Pack 登录认证（可选择是否开启）。
- 白金版默认开启此功能。

举例：
- 从6.4开源版（无此功能）升级到6.8基础版（选择不开启此功能），不需要全量重启集群。
- 从6.4开源版（无此功能）升级到6.8基础版（选择开启此功能），需要全量重启集群。

©2013-2024 Tencent Cloud. All rights reserved.

- 从6.8基础版（未开启此功能）升级到6.8白金版（默认开启此功能），需要全量重启集群。
- 从6.8基础版（已开启此功能）升级到6.8白金版（默认开启此功能），不需要全量重启集群。

升级注意事项

Elasticsearch 5.x版本与6.x版本兼容及使用说明

1. index 多 type
   Elasticsearch 从 6.x 开始，不再支持一个索引多个 type。从 5.x 版本升级到 6.x 版本后，新建多 type 的索引将会报错，原有在 5.x 版本创建的多 type 索引，不受影响，可以正常写入。
2. curl 访问集群用 curl 访问集群时，需要增加请求 `header -H 'Content-Type: application/json'` 。

```
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
}'
```

3. 配置项兼容 Elasticsearch 不同版本之间会存在一些不兼容的配置，如果您有设置，升级后可能会影响集群的使用。ES 升级功能提供了配置项的检查流程，也提供了调整说明，见下文 升级检查 。
4. 更多说明请参考 Breaking changes in 6.0 版本 。

升级处理流程说明

Elasticsearch 版本升级，需要先进行升级检查、数据备份两个步骤。前两个成功后，才会开始执行升级操作。

1. 升级检查

> **说明**
> 只有 Elasticsearch 版本升级会有。

检查升级前后两个版本是否有不兼容的配置，如果检查不通过，流程就会终止。如果升级过程中，遇到升级检查不通过的情况，您可以查看具体的检查项和相应的解决方案，详情可参见 升级检查 。您也可以在升级操作前，只选择升级检查操作，查看集群是否满足升级条件。

2. 快照备份

> **说明**
> 仅 Elasticsearch 版本升级会有。

升级操作前，ES 会先对您的集群进行快照备份，以防升级操作失败时，可以用快照还原集群。所以，如果快照这一步失败，升级流程也会终止。**快照备份耗时同集群数据量有关，如果集群未开启自动快照备份，且数据量较大，第一次快照时间会比较长。**

3. 升级过程和集群重启 6.x 及以上版本的集群，支持高级特性的升级（从开源版到基础版或白金版），升级期间服务需要重启，方式如下：

- 若升级过程不涉及 ES 集群用户登录认证从关闭到开启，集群需滚动重启，需满足健康状态是绿色，并且不存在只配置了单个副本的索引，以及不能有 close 的索引，才能执行升级操作。升级期间，服务可以正常访问，但性能可能受到部分影响，建议在集群负载不高时进行。
- 若升级过程涉及 ES 集群用户登录认证从关闭到开启，集群需要全量重启，期间集群会停服，服务不可访问，升级完成后才能正常访问，请谨慎选择。

集群升级操作步骤

©2013-2024 Tencent Cloud. All rights reserved.

1. 登录 ES 控制台，进入集群详情页，单击右上角**更多操作 > 升级**。

2. 在升级对话框中可以选择升级 Elasticsearch 版本或选择升级高级特性。

升级 Elasticsearch 版本

1. 在升级页面中选择升级类型为**升级版本**。
2. 在 **Elasticsearch 版本**下拉框中选择要升级到的版本。

> **说明**
> - 若不选择同时升级高级特性，则默认升级到高版本时保持原高级特性。
> - ES 主版本升级，例如 ES 5.x 版本升级到 6.x 版本，可以同时把**高级特性**从开源版升级到基础版。我们也建议您选择**基础版**高级特性，其包含监控、SQL 等高级版功能。升级到6.8及以上基础版，还可以选择勾选 ES 集群用户登录认证，此时需要全量重启，期间集群会停服，服务不可访问，请谨慎选择。

3. 升级前，会先对集群状态进行检查，判断集群是否可以升级，包括检查集群的配置、集群的状态等。**操作选择**可以选择**仅做升级检查，暂不升级**，在单击**确定**后，仅做升级检查，不执行升级命令，您可以在详情页**变更记录**页签中查看检查结果。

> **说明**
> ES 主版本升级，例如 5.x 版本到 6.x 版本，有些集群级别的配置项和索引级别的配置项不兼容，需要通过**升级检查**判断集群是否可升级。整个升级检查会检查一些配置项，错误的配置项需要您进行调整，告警的配置项可以选择性调整，具体可查看 ES 版本升级检查 。

©2013–2024 Tencent Cloud. All rights reserved.

4. 选择完成后，单击**确定**开始版本升级。

升级高级特性

1. 在升级页面中选择升级类型为**升级[高级特性]版本**。
2. 在**高级特性**中选择要升级到的高级特性版本。
3. 单击**确定**开始升级。

> **注意**
> - 高级特性升级须知：
>   目前支持 6.x 及以上版本的高级特性升级，5.x 版本不支持（5.x 版本只有开源版，无基础和白金版）。
> - 升级到不同的版本，过程有所不同，具体如下：
> - 若升级过程不涉及 ES 集群用户登录认证从关闭到开启，集群需要滚动重启，期间服务访问会受短暂影响，请在服务访问量不大时进行操作。
> - 若升级过程涉及 ES 集群用户登录认证从关闭到开启，集群需要全量重启，期间集群会停服，服务不可访问，请谨慎选择。

4. 开始升级后，可以在集群详情页的**变更记录**页签中查看升级进度。

©2013–2024 Tencent Cloud. All rights reserved.

ES 版本升级检查

Last updated：2024-10-24 11:49:32

Elasticsearch 不同版本之间有部分不兼容的配置，如果您已设置，升级后，使用集群可能会受到影响。您可通过升级检查的功能，检查是否有不兼容的配置，并进行调整。以下是 ES 对 Elasticsearch 版本升级时检查的配置说明。
升级检查操作入口，在 Elasticsearch Service 控制台 详情页，单击右上角**升级**，操作步骤详见 升级 ES 集群 。

> **警告**
> 需特别留意本文中需要人工检查的部分（无法自动检查），并在升级前完成代码兼容性改造，避免升级后造成集群无法正常访问，例如，6.8版本升级到7.5版本后默认无法创建包含 type 的索引，详情见下文 需人工自查的不兼容配置 。

Elasticsearch 5.x 版本升级到 6.x 版本配置检查

检查的配置项列表

序号	配置维度	配置信息	兼容性	详细说明
1	集群级别	集群快照设置（Snapshot settings）	CRITICAL	集群设置 `cluster.routing.allocation.snapshot.relocation_enabled`，在6.0版本开始被废弃，详见 Breaking changes in 6.0
2	集群级别	集群存储限流设置（Store throttling settings）	CRITICAL	集群设置 `indices.store.throttle.type` 和 `indices.store.throttle.max_bytes_per_sec`，在6.0版本开始被废弃，详见 Breaking changes in 6.0
3	索引级别	索引相似性设置（Similarity settings）	WARNING	索引设置 `index.similarity.base`，在6.0版本开始被废弃，详见 Breaking changes in 6.0
4	索引级别	索引影子副本设置（Shadow Replicas settings）	CRITICAL	索引设置 `index.shared_filesystem` 和 `index.shadow_replicas`，在6.0版本开始被废弃，详见 Breaking changes in 6.0
5	索引级别	索引存储设置（Index Store settings）	CRITICAL	索引设置 `index.store.type` 为 default，在6.0版本开始被废弃，详见 Breaking changes in 6.0
6	索引级别	索引存储限流设置（Index Store throttling settings）	CRITICAL	索引设置 `index.store.throttle.type` 和 `index.store.throttle.max_bytes_per_sec`，在6.0版本开始被废弃，详见 Breaking changes in 6.0
7	索引级别	索引 Mapping 参数 include_in_all	WARNING	索引 mapping 参数 `include_in_all`，在6.0版本之后创建的索引中无法使用（5.x 版本创建包含此设置的索引在升级 6.x 版本后可以兼容），详见 Breaking changes in 6.0
8	索引级别	索引创建版本	CRITICAL	索引创建版本 `index.version.created` 不允许跨 ES 主版本，例如，无法将在 5.x 版本创建的索引直接升级到 7.x 版本，需要将失败的索引 reindex 迁移到新索引并删除后再进行升级。
9	索引模板级别	索引模板相似性设置（Similarity settings）	CRITICAL	索引设置 `index.similarity.base`，在6.0版本开始被废弃，详见 Breaking changes in 6.0，模板中存在该设置将导致版本升级后该模板无法用于创建新索引。
10	索引模板级别	索引模板影子副本设置（Shadow Replicas settings）	CRITICAL	索引设置 `index.shared_filesystem` 和 `index.shadow_replicas`，在6.0版本开始被废弃，详见 Breaking changes in 6.0，模板中存在该设置将导致版本升级后该模板无法用于创建新索引。
11	索引模板级别	索引模板存储设置（Index Store settings）	CRITICAL	索引设置 `index.store.type` 为 default，在6.0版本开始被废弃，详见 Breaking changes in 6.0，模板中存在该设置将导致版本升级后该模板无法用于创建新索引。

©2013-2024 Tencent Cloud. All rights reserved.

12	索引模板级别	索引模板存储限流设置（Index Store throttling settings）	CRITICAL	索引设置 `index.store.throttle.type` 和 `index.store.throttle.max_bytes_per_sec`，在6.0版本开始被废弃，详见 Breaking changes in 6.0，模板中存在该设置将导致版本升级后该模板无法用于创建新索引。
13	索引模板级别	索引模板 Mapping 参数 `include_in_all`	CRITICAL	索引 Mapping 参数 `include_in_all`，在6.0版本开始被废弃，详见 Breaking changes in 6.0，模板中存在该设置将导致版本升级后该模板无法用于创建新索引。
14	索引模板级别	索引模板 Mapping 元字段 `_all`	CRITICAL	索引 Mapping 元字段 `_all`，在6.0版本开始被废弃，详见 Breaking changes in 6.0，模板中存在该设置将导致版本升级后该模板无法用于创建新索引。
15	索引模板级别	索引模板 Mapping 多个 Types	CRITICAL	索引 Mapping 包含多个 Types，在6.0版本开始被废弃，详见 Removal of mapping types，模板中存在该设置将导致版本升级后该模板无法用于创建新索引。

说明

- 警告（WARNING）：检查失败时仍可以升级。此类型检查项对应设置在升级后将被忽略。
- 错误（CRITICAL）：检查失败时无法升级。此类型检查项对应设置在目标版本无法兼容。

配置不兼容调整方法

集群级别

- 集群快照设置（Snapshot settings）
  通过 ES 集群设置更新接口 `PUT _cluster/settings` 取消此设置（包括 persistent 和 transient）：

```
PUT _cluster/settings
{
  "persistent": {
    "cluster.routing.allocation.snapshot.relocation_enabled": null
  },
  "transient": {
    "cluster.routing.allocation.snapshot.relocation_enabled": null
  }
}
```

- 集群存储限流设置（Store throttling settings）
  通过 ES 集群设置更新接口 `PUT _cluster/settings` 取消此设置（包括 persistent 和 transient）：

```
PUT _cluster/settings
{
  "persistent": {
    "indices.store.throttle.type": null,
    "indices.store.throttle.max_bytes_per_sec": null
  },
  "transient": {
    "indices.store.throttle.type": null,
    "indices.store.throttle.max_bytes_per_sec": null
  }
}
```

索引级别

- 索引相似性设置（Similarity settings）
  此设置在升级 6.x 后被忽略，但不会影响升级。如需取消设置，可按照如下步骤操作：
- 此设置需要关闭索引后修改，关闭后的索引无法被读写。关闭索引：

©2013–2024 Tencent Cloud. All rights reserved.

```
POST my_index/_close
```

- 使用 ES 索引设置更新接口取消此设置：

```
PUT my_index/_settings
{
    "index.similarity.base.*": null
}
```

- 最后打开索引：

```
POST my_index/_open
```

- 索引影子副本设置（Shadow Replicas settings）
- 此设置需要关闭索引后修改，关闭后的索引无法被读写。关闭索引：

```
POST my_index/_close
```

- 使用 ES 索引设置更新接口取消此设置：

```
PUT my_index/_settings
{
     "index.shared_filesystem": null,
     "index.shadow_replicas": null
}
```

- 最后打开索引：

```
POST my_index/_open
```

- 索引存储设置（Index Store settings）
- 此设置需要关闭索引后修改，关闭后的索引无法被读写。关闭索引：

```
POST my_index/_close
```

- 使用 ES 索引设置更新接口取消此设置：

```
PUT my_index/_settings
{
    "index.store.type": null
}
```

- 最后打开索引：

```
POST my_index/_open
```

- 索引存储限流设置（Index Store throttling settings）
  使用 ES 索引设置更新接口取消此设置：

```
PUT my_index/_settings
{
  "settings": {
    "index.store.throttle.type": null,
    "index.store.throttle.max_bytes_per_sec": null
```

©2013–2024 Tencent Cloud. All rights reserved.

```
  }
}
```

- 索引 Mapping 参数 include_in_all
  对已创建的包含此参数的索引，升级后可以兼容，无需修复。

索引模板级别

1. 使用 API `GET _template/my_template` 获取不兼容模板 `my_template` 。存在以下三个不兼容设置：索引模板存储设置（Index Store settings）、索引模板 Mapping 参数 `include_in_all` 、索引模板 Mapping 元字段 `_all` 。

```
 {
"my_template": {
  "order": 0,
  "template": "my_*",
  "settings": {
    "index": {
      "store": {
        "throttle": {
          "max_bytes_per_sec": "10m"
        }
      }
    }
  },
  "mappings": {
    "my_type": {
      "_all": {
        "enabled": true
      },
      "properties": {
        "my_field": {
          "type": "text",
          "include_in_all": true
        }
      }
    }
  },
  "aliases": {}
}
 }
```

2. 拷贝并去除模板中不兼容配置后，使用 API `PUT _template/my_template` 更新模板。

```
PUT _template/my_template
{
 "order": 0,
 "template": "my_*",
 "settings": {
 },
 "mappings": {
   "my_type": {
     "properties": {
       "my_field": {
         "type": "text"
       }
     }
   }
 },
 "aliases": {}
```

©2013–2024 Tencent Cloud. All rights reserved.

```
}
```

Elasticsearch 6.8 版本升级到 7.x 版本配置检查

需人工自查的不兼容配置

- 对于超过1024个字段数的索引，查询时如果不指定字段会非常消耗性能。建议设置默认查询字段 `index.query.default_field` 。如果仍要查询所有字段，那么可以通过提高“布尔查询最大子句数”设置 `indices.query.bool.max_clause_count` 来实现。
- `include_type_name` 默认为 false，升级后默认将无法创建包含 type 的索引，需要显式指定参数 `include_type_name` ：

> **警告**
> 需重点关注集群是否有此项不兼容配置，避免升级后造成集群无法正常访问。

```
PUT my_index?include_type_name
{
  "mappings": {
    "type1": {
      "properties": {
        "name": {
          "type": "text"
        }
      }
    }
  }
}
```

可自动检查的配置项列表

序号	配置维度	配置信息	兼容性	详细说明
1	集群级别	用户代理预处理器（User agent processor）设置	WARNING	用户代理预处理器（User agent processor）的格式默认值变为 `ecs` ，详见 Breaking changes in 7.0
2	集群级别	集群分片数	WARNING	每个节点最大 shard 数 `cluster.max_shards_per_node` 默认1000，shard 数超过在升级后将无法创建新分片，详见 Breaking changes in 7.0
3	集群级别	集群设置（discovery.zen.no_master_block）	WARNING	集群设置 `discovery.zen.no_master_block` 改名为 `cluster.no_master_block` ，详见 Breaking changes in 7.0
4	集群级别	预处理器管道（Ingest pipeline）日期格式	WARNING	预处理器管道（Ingest pipeline） `date` 或 `date_index_name` 处理器使用废弃的日期格式，详见 Breaking changes in 7.0
5	索引级别	索引创建版本	CRITICAL	不允许6.0版本之前创建的索引升级到7.x版本，详见 Breaking changes in 7.0
6	索引级别	Delimited payload 词汇单元过滤器（Token filter）索引设置	WARNING	Delimited payload 语汇单元过滤器（Token filter）从 `delimited_payload_filter` 改名为 `delimited_payload` ，详见 Breaking changes in 7.0
7	索引级别	索引设置（index.percolator.map_unmapped_fields_as_string）	CRITICAL	索引设置 `index.percolator.map_unmapped_fields_as_string` 改名为 `index.percolator.map_unmapped_fields_as_text` ，详见 Breaking changes in 7.0
8	索引级别	索引名	WARNING	索引名不能包含“:”，详见 Breaking changes in 7.0

©2013–2024 Tencent Cloud. All rights reserved.

9	索引级别	索引设置（index.unassigned.node_left.delayed_timeout）	CRITICAL	索引设置 `index.unassigned.node_left.delayed_timeout` 负数值被废弃，详见 Breaking changes in 7.0
10	索引级别	索引设置（index.shard.check_on_startup）	CRITICAL	索引设置 `index.shard.check_on_startup` 的值为 `fix` 被废弃，详见 Breaking changes in 7.0
11	索引级别	经典相似性（classic similarity）索引 Mapping 参数	WARNING	经典相似性（classic similarity）索引 Mapping 参数被废弃，详见 Breaking changes in 7.0
12	索引级别	经典相似性（classic similarity）索引设置	WARNING	经典相似性（classic similarity）索引设置被废弃，详见 Breaking changes in 7.0
13	索引级别	索引字段数	WARNING	对于超过1024个字段的索引，建议设置默认查询字段（`index.query.default_field`）或提升最大字段数设置（`indices.query.bool.max_clause_count`），否则升级后将导致不指定字段的查询无法正常使用，详见 Breaking changes in 7.0
14	索引级别	索引 Mapping 日期格式	WARNING	Joda-Time 格式改为 Java Time，详见 Breaking changes in 7.0
15	索引模板级别	Delimited payload 语汇单元过滤器（Token filter）索引模板设置	WARNING	Delimited payload 语汇单元过滤器（Token filter）从 `delimited_payload_filter` 改名为 `delimited_payload`，详见 Breaking changes in 7.0
16	索引模板级别	索引模板设置（index.percolator.map_unmapped_fields_as_string）	CRITICAL	索引模板设置 `index.percolator.map_unmapped_fields_as_string` 改名为 `index.percolator.map_unmapped_fields_as_text`，升级后不兼容，详见 Breaking changes in 7.0
17	索引模板级别	索引模板设置（index.unassigned.node_left.delayed_timeout）	CRITICAL	索引模板设置 `index.unassigned.node_left.delayed_timeout` 负数值被废弃，升级后不兼容，详见 Breaking changes in 7.0
18	索引模板级别	索引模板设置（index.shard.check_on_startup）	CRITICAL	索引模板设置 `index.shard.check_on_startup` 的值为 `fix` 被废弃，升级后不兼容，详见 Breaking changes in 7.0
19	索引模板级别	经典相似性（classic similarity）索引模板 Mapping 参数	CRITICAL	经典相似性（classic similarity）索引模板 Mapping 参数被废弃，升级后不兼容，详见 Breaking changes in 7.0
20	索引模板级别	经典相似性（classic similarity）索引模板设置	CRITICAL	经典相似性（classic similarity）索引模板设置被废弃，升级后不兼容，详见 Breaking changes in 7.0

©2013-2024 Tencent Cloud. All rights reserved.

21	索引模板级别	索引模板字段数	CRITICAL	对于超过1024个字段的索引模板，建议设置默认查询字段（`index.query.default_field`）或提升最大字段数设置（`indices.query.bool.max_clause_count`），否则升级后将导致不指定字段的查询无法正常使用，详见 Breaking changes in 7.0 **说明：** 对于超过1024个字段的索引模板，我们提供以下建议： • 查询时如果不指定字段会非常消耗性能。建议设置默认查询字段 `index.query.default_field`。 • 如果仍要查询所有字段，那么可以通过提高“布尔查询最大子句数”设置 `indices.query.bool.max_clause_count` 来实现（该操作是集群层面的设置）。
22	索引模板级别	索引模板 Mapping 日期格式	WARNING	Joda-Time 格式改为 Java Time，详见 Breaking changes in 7.0

配置不兼容调整方法

集群级别

- 用户代理预处理器（User agent processor）设置
  使用非 `ecs` 格式的 user agent 将被废弃，应调整为：

```
PUT _ingest/pipeline/my_pipeline1
{
"processors" : [
  {
    "user_agent" : {
      "field" : "agent",
      "ecs": true
    }
  }
]
}
```

- 集群分片数
  对于超过分片数限制的集群，有两种调整方法：
- 降低集群分片数（通过清理过期索引、shrink 索引 等）。
- 修改集群设置提高分片数限制。

```
PUT _cluster/settings
{
  "persistent": {
 "cluster.max_shards_per_node": 5000
  }
}
```

- 集群设置（discovery.zen.no_master_block）
  对于使用了集群设置 `discovery.zen.no_master_block` 的集群，在升级完成后调整为 `cluster.no_master_block`，例如：

```
PUT _cluster/settings
{
"persistent": {
  "cluster.no_master_block": "write"
}
}
```

©2013-2024 Tencent Cloud. All rights reserved.

- 预处理器管道（Ingest pipeline）日期格式
  可参考 Joda based date formatters are replaced with java ones 进行调整。

索引级别

- 索引创建版本
  不允许6.0版本之前创建的索引升级到7.x版本，对于需要保留的索引，需要在升级前执行 reindex ， 例如：

```
POST _reindex
{
"source": {
  "index": "my_index"
},
"dest": {
  "index": "new_my_index"
}
}
```

- Delimited payload 语汇单元过滤器（Token filter）索引模板设置
  将索引设置中的 `delimited_payload_filter` 改为 `delimited_payload`，例如原始索引如下：

```
PUT my_index
{
"settings": {
  "analysis": {
    "analyzer": {
      "whitespace_plus_delimited": {
        "tokenizer": "whitespace",
        "filter": [ "plus_delimited" ]
      }
    },
    "filter": {
      "plus_delimited": {
        "type": "delimited_payload_filter",
        "delimiter": "+",
        "encoding": "int"
      }
    }
  }
}
}
```

  修改后的索引：

```
PUT my_index
{
"settings": {
  "analysis": {
    "analyzer": {
      "whitespace_plus_delimited": {
        "tokenizer": "whitespace",
        "filter": [ "plus_delimited" ]
      }
    },
    "filter": {
      "plus_delimited": {
        "type": "delimited_payload",
        "delimiter": "+",
        "encoding": "int"
```

©2013–2024 Tencent Cloud. All rights reserved.

```
      }
    }
  }
}
}
```

- **索引设置（index.percolator.map_unmapped_fields_as_string）**
  将索引设置 `index.percolator.map_unmapped_fields_as_string` 调整为 `index.percolator.map_unmapped_fields_as_text`（此设置调整前需要 close 索引）。

```
PUT my_index/_settings
{
"index.percolator.map_unmapped_fields_as_text": true
}
```

- **索引名**
  对于包含“:”的索引，需执行 reindex，例如：

```
POST _reindex
{
"source": {
  "index": "my_index"
},
"dest": {
  "index": "new_my_index"
}
}
```

- **索引设置（index.unassigned.node_left.delayed_timeout）**
  将索引设置 `index.unassigned.node_left.delayed_timeout` 调整为正整数：

```
PUT my_index/_settings
{
"index.unassigned.node_left.delayed_timeout": 0
}
```

- **索引设置（index.shard.check_on_startup）**
  将索引设置 `index.shard.check_on_startup` 值为 `fix` 调整为其它值，例如：

```
PUT my_index/_settings
{
"index.shard.check_on_startup": true
}
```

- **经典相似性（classic similarity）索引 Mapping 参数、经典相似性（classic similarity）索引设置**
  对于包含经典相似性（classic similarity）的索引设置或 Mapping，调整为其他类型，例如原始索引如下：

```
PUT my_index
{
"mappings": {
  "doc": {
    "properties": {
      "field1": {
        "properties": {
          "field4": {
            "type": "text",
            "similarity": "classic"
```

©2013–2024 Tencent Cloud. All rights reserved.

```
          }
        }
      }
    }
  }
}
}
```

修改后的索引为：

```
PUT my_index
{
"mappings": {
  "doc": {
    "properties": {
      "field1": {
        "properties": {
          "field4": {
            "type": "text",
            "similarity": "BM25"
          }
        }
      }
    }
  }
}
}
```

- 索引字段数
  对于超过1024个字段的索引，有两种调整方法：
- 设置默认查询字段设置 `index.query.default_field` 。

```
PUT my_index/_settings
{
  "index.query.default_field": "field1"
}
```

- 通过修改 ES YML 配置文件提升最大字段数设置 `indices.query.bool.max_clause_count` （后续支持）。
- 索引 Mapping 日期格式
  可参考 Joda based date formatters are replaced with java ones 进行调整。

索引模板级别

索引模板级别的不兼容设置调整请参考 索引级别 。

©2013–2024 Tencent Cloud. All rights reserved.

智能巡检

Last updated：2024-10-12 21:50:12

智能巡检支持基于集群、节点、索引等维度对腾讯云 Elasticsearch（ES）集群进行健康检测，能够主动排查集群问题和隐患，并基于腾讯云专家经验沉淀给出问题解决方案，自动归纳集群巡检结果生成报告。智能巡检服务能够为用户提取集群关键信息、高效定位集群问题、提供专业解决建议，实现运维体验闭环。

功能概述

- 开启智能巡检
- 查看巡检概览
  查看集群巡检健康概况和趋势变化。
- 集群巡检
  支持选择需要巡检的索引和巡检项，支持自动巡检和手动巡检集群，并产生巡检报告。
- 查看巡检报告
  支持查看最近一个月的历史巡检报告，报告中包含巡检项说明、巡检结果、巡检建议、详情。
- 关闭智能巡检

开启智能巡检

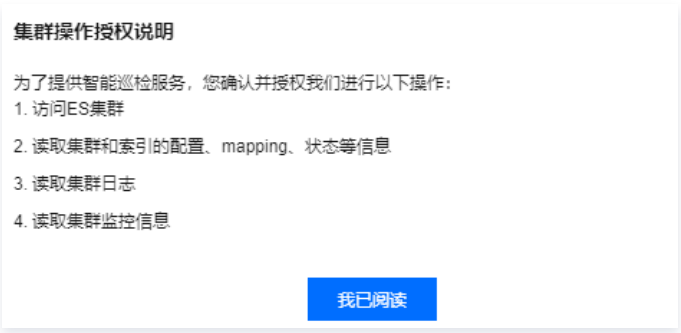
智能巡检需要访问用户集群的基本信息和日志，因此当首次启用智能巡检服务时，需要用户对访问授权。

操作步骤

1. 登录 ES 控制台，在**集群列表**中单击需要开启智能巡检的**实例 ID/名称**，进入实例详情页。
2. 在实例详情页，切换到**智能巡检**页签，界面内展示智能巡检功能的初始化页面。

©2013-2024 Tencent Cloud. All rights reserved.

3. 在初始化页面中，可单击**《集群操作授权说明》**，阅读相关说明和提示事项。



4. 确认无误后，勾选**我已阅读并充分理解《集群操作授权说明》**。然后单击**授权开启智能巡检服务**，即可开启智能巡检功能。

查看巡检概览

巡检概览统计并展示了集群的巡检结果及其趋势变化，方便用户查看集群近期的健康状况。巡检结果通过高风险、低风险和安全3种状态来展示集群的健康状况：

- 高风险：表示集群已经出现了严重的问题或隐患，已经影响集群可用性，需要立即处理，否则会导致数据丢失、集群故障等问题。
- 低风险：表示集群存在较严重的问题或隐患，可能会影响集群可用性，建议尽快处理。
- 安全：表示集群健康。

操作步骤

1. 登录 ES 控制台，在**集群列表**中单击**实例 ID/名称**（实例已开启智能巡检），进入实例详情页。
2. 在实例详情页，切换到**智能巡检**页签，在巡检概览部分，展示了最新集群巡检结果、巡检项趋势和近7天巡检结果累计分布。
   - 最新集群巡检结果：会展示最近一次的集群巡检结果。
   - 巡检项趋势：展示最近7天内每天最后一次巡检的巡检项健康状态。
   - 近7天巡检结果累计分布：展示最近7天内全部巡检次数累计的巡检结果，帮助您判断巡检项的健康状况。

巡检集群

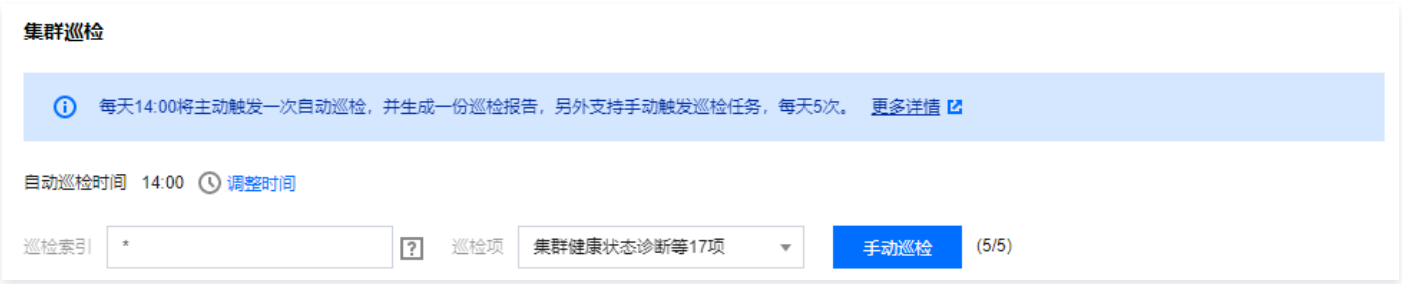
开启智能巡检功能后，支持巡检集群，并产生巡检报告。支持自动巡检和手动巡检，用户可选择需要巡检的索引和巡检项。每次巡检约耗时2分钟。

- 自动巡检：系统将随机生成一个固定巡检时间，每天在该时间点定时巡检集群，默认选择全部巡检项并生成巡检报告，每天一次，支持更改巡检时间。
- 手动巡检：除了每天的自动巡检外，支持手动随时触发巡检任务，用户可以指定部分巡检项和索引，并立即生成巡检报告，每天限制5次。

操作步骤

1. 登录 ES 控制台，在**集群列表**中单击**实例 ID/名称**（实例已开启智能巡检），进入实例详情页。
2. 在实例详情页，切换到**智能巡检**页签。
3. 在“智能巡检”页的**集群巡检**中选择巡检索引和巡检项。系统默认选择所有巡检索引和所有巡检项，用户也可自定义选择，支持通过正则表达式选择巡检索引。选择完成后，单击**手动巡检**。

©2013-2024 Tencent Cloud. All rights reserved.



4. 诊断完成后，系统会展示最新的智能巡检报告。查看智能巡检报告，获取集群的最新状况，包括巡检项说明、巡检结果和巡检建议。

查看巡检报告

巡检报告中包含巡检项说明、巡检结果、巡检建议以及详情，报告内容由用户选择的巡检项构成。最多支持查看最近一个月的历史巡检报告。

操作步骤

1. 登录 ES 控制台，在**集群列表**中单击**实例 ID/名称**（实例已开启智能巡检且至少已生成一份报告），进入实例详情页。
2. 在实例详情页，切换到**智能巡检**页签。
3. 在“智能巡检”页的**巡检报告**中，可单击右侧目录查看不同日期下的历史报告，在左侧查看对应的历史报告详情。

关闭智能巡检

当用户不再需要使用智能巡检功能时，可关闭该服务，关闭后，系统将不会再定时巡检集群并生成新的巡检报告。关闭后再开启，系统将保留开启日期前30天的历史巡检报告。

©2013–2024 Tencent Cloud. All rights reserved.

操作步骤

1. 登录 ES 控制台，在**集群列表**中单击**实例 ID/名称**（实例已开启智能巡检），进入实例详情页。
2. 在实例详情页，切换到**智能巡检**页签，单击**关闭智能巡检**。

3. 在**关闭智能巡检**中，阅读注意事项，确认无误后，单击**确定**即可关闭智能巡检。

©2013–2024 Tencent Cloud. All rights reserved.