**目录**

1. ElasticStack及Elasticsearch介绍
2. Elasticsearch安装
3. Elasticsearch入门
4. Elasticsearch配置
5. Elasticsearch REST API
6. Elasticsearch .NET客户端
7. Elasticsearch PasS申请流程(Accenture CIO)
8. 附录

**ElasticStack及Elasticsearch介绍**

官方网站：<https://www.elastic.co/>

Elastic Stack (旧称ELK Stack): Elasticsearch + Logstash + Kibana + Beats (New)

Elastic Stack组件：

* Elasticsearch

            搜索、分析和存储数据。Elasticsearch是一个基于JSON的分布式搜索和分析引擎，专为水平可扩展性，最高可靠性和易管理性而设计。

* Logstash

            Logstash是一个动态数据收集管道，具有可扩展的插件生态系统和强大的Elasticsearch协同作用。

* Kibana

            可视化您的数据。 导航弹性堆栈。Kibana为您的数据提供了可视化，是用于配置和管理Elastic Stack各个方面的可扩展用户界面。

* Beats

            Beats是轻量级数据运送平台，可将边缘机器的数据发送到Logstash和Elasticsearch。

**Elasticsearch安装**

最新版本为Elasticsearch 6.5.4 GA Release，Windows下载安装包如下：

zip: <https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.5.4.zip>

msi: <https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.5.4.msi> (105MB)

安装条件：64位Java虚拟机

* 选项1(较慢)：下载并安装JRE: <https://www.oracle.com/technetwork/java/javase/overview/index.html>
* 选项2(快)：可以先安装chocolatey，再通过chocolatey安装Java SE Runtime

安装步骤：

1. 下载并解压缩Elasticsearch
2. 运行bin\elasticsearch.exe
3. 使用浏览器访问http://localhost:9200

配置ES使用JVM分配的内存堆(Heap)

对于小型部署来说1G足够，最大不应超过内存的50%

安装成功验证画面

**Elasticsearch入门**

Elasticsearch是一个高度可扩展的开源全文搜索和分析引擎。 它允许您快速，近实时地存储，搜索和分析大量数据。 它通常用作底层引擎/技术，为具有复杂搜索功能和要求的应用程序提供支持。

以下是Elasticsearch可用于的一些示例用例：

* 您经营一家在线网上商店，您可以让客户搜索您销售的产品。在这种情况下，您可以使用Elasticsearch存储整个产品目录和库存，并为它们提供搜索和自动填充建议。
* 您希望收集日志或交易数据，并且希望分析和挖掘此数据以查找趋势，统计信息，摘要或异常。在这种情况下，您可以使用Logstash（Elasticsearch / Logstash / Kibana堆栈的一部分）来收集，聚合和解析数据，然后让Logstash将此数据提供给Elasticsearch。一旦数据在Elasticsearch中，您就可以运行搜索和聚合来挖掘您感兴趣的任何信息。
* 您运行价格警报平台，允许精通价格的客户指定一条规则，例如“我有兴趣购买特定的电子产品，如果小工具的价格在下个月内从任何供应商降至X美元以下，我希望收到通知” 。在这种情况下，您可以刮取供应商价格，将其推入Elasticsearch并使用其反向搜索（Percolator）功能来匹配价格变动与客户查询，并最终在发现匹配后将警报推送给客户。
* 您有分析/业务智能需求，并希望快速调查，分析，可视化并询问有关大量数据的特定问题（想想数百万或数十亿条记录）。在这种情况下，您可以使用Elasticsearch存储数据，然后使用Kibana（Elasticsearch / Logstash / Kibana堆栈的一部分）构建自定义仪表板，以便可视化对您来说重要的数据方面。此外，您可以使用Elasticsearch聚合功能针对您的数据执行复杂的商业智能查询。

基本概念

* NRT(接近实时)

            Elasticsearch是一个近实时搜索平台。 这意味着从索引文档到可搜索文档的时间有一点延迟（通常为一秒）。

* Cluster(集群) —— 按应用或业务，如HR、法务

            集群是一个或多个节点（服务器）的集合，它们共同保存您的整个数据，并提供跨所有节点的联合索引和搜索功能。 群集由唯一名称标识，默认情况下为“elasticsearch”。 此名称很重要，因为如果节点设置为按名称加入群集，则该节点只能是群集的一部分。

* Node(节点) —— 按物理服务器

            节点是作为群集一部分的单个服务器，存储数据并参与群集的索引和搜索功能。 就像集群一样，节点由名称标识，默认情况下，该名称是在启动时分配给节点的随机通用唯一标识符（UUID）。 如果不需要默认值，可以定义所需的任何节点名称。

* Index(索引)  ——  按业务领域，如订单、合同、交易记录

            索引是具有某些类似特征的文档集合。 例如，您可以拥有客户数据的索引，产品目录的另一个索引以及订单数据的另一个索引。 索引由名称标识（必须全部小写），此名称用于在对其中的文档执行索引，搜索，更新和删除操作时引用索引。

* Type(类型) —— 6.0.0已弃用
* Document(文档)

            文档是可以编制索引的基本信息单元。 例如，您可以为单个客户提供文档，为单个产品提供另一个文档，为单个订单提供另一个文档。 该文档以JSON（JavaScript Object Notation）表示，JSON是一种普遍存在的互联网数据交换格式。在索引/类型中，您可以根据需要存储任意数量的文档。 请注意，尽管文档实际上驻留在索引中，但实际上必须将文档编入索引/分配给索引中的类型。

* Shards & Replicas (分片和复制)

            为了解决这个问题，Elasticsearch提供了将索引细分为多个称为分片的功能。 创建索引时，只需定义所需的分片数即可。 每个分片本身都是一个功能齐全且独立的“索引”，可以托管在集群中的任何节点上。

            分片很重要，主要有两个原因：

* + 允许水平拆分/扩展索引容量
  + 允许跨分片（可能在多个节点上）分布和并行化操作，从而提高性能/吞吐量

            分片的分布方式以及如何将其文档聚合回搜索请求的机制完全由Elasticsearch管理，对用户而言是透明的。

            在可以随时发生故障的网络/云环境中，非常有用，强烈建议使用故障转移机制，以防分片/节点以某种方式脱机或因任何原因消失。 为此，Elasticsearch允许您将索引的分片的一个或多个副本制作成所谓的副本分片或简称副本。

        主分片和复制分片：

* 此索引包括两个主分片和两个复制分片。你的应用会循环请求各节点。
* 写请求路由到主分片然后复制分片。
* 读请求路由到主分片或任何复制分片。

**Elasticsearch配置**

ES目录结构

* bin: 运行ES实例和插件管理的脚本
* lib: ES使用的库文件
* modules: ES主要模块
* plugins: ES插件目录

安装后可以通过Windows服务来停止或启动Elasticsearch。

ES客户端

* curl: 可以通过命令行工具curl与ES交互，Windows下curl下载地址：<https://curl.haxx.se/download.html>
* Postman: HTTP请求测试工具

**Elasticsearch REST API**

现在我们已经启动并运行了节点（和集群），下一步是了解如何与它进行通信。 幸运的是，Elasticsearch提供了一个非常全面和强大的REST API，您可以使用它与集群进行交互。 使用API可以完成的一些事项如下：

* 检查群集，节点和索引运行状况，状态和统计信息
* 管理您的群集，节点和索引数据和元数据
* 对索引执行CRUD（创建，读取，更新和删除）和搜索操作
* 执行高级搜索操作，例如分页，排序，过滤，脚本编写，聚合等等

查询集群状态：GET <http://localhost:9200/_cat/health?v>

查询节点状态：GET <http://localhost:9200/_cat/nodes?v>

查询索引列表：GET <http://localhost:9200/_cat/indices?v>

创建索引customer：PUT  <http://localhost:9200/customer?pretty>

索引文档：PUT <http://localhost:9200/customer/_doc/1?pretty>

{

"name": "John Doe"

}

查询文档：GET <http://localhost:9200/customer/_doc/1?pretty>

删除索引：DELETE <http://localhost:9200/customer?pretty>

**基本语法：<HTTP Verb> /<Index>/<Type>/<ID>**

更新/创建索引(带文档ID)：PUT <http://localhost:9200/customer/_doc/1?pretty>

创建索引(不带文档ID)：POST  <http://localhost:9200/customer/_doc?pretty>

更新文档内容：POST <http://localhost:9200/customer/_doc/1?pretty>

{

  "script" : "ctx.\_source.age += 5"

}

删除文档：DELETE <http://localhost:9200/customer/_doc/2?pretty>

**Elasticsearch .NET客户端**

Elasticsearch的.NET客户端包括[Elasticsearch.Net](http://elasticsearch.net/)和NEST：

* [Elasticsearch.Net](http://elasticsearch.net/) - 底层客户端

<https://www.elastic.co/guide/en/elasticsearch/client/net-api/current/elasticsearch-net-getting-started.html>

* NEST (推荐使用) - 高级客户都

<https://www.elastic.co/guide/en/elasticsearch/client/net-api/current/nest-getting-started.html>

**Elasticsearch PasS申请流程(CIO)**

准备材料

1. Obtain the Name Domain requirements.
2. Elastic Search Version
3. Node Configuration
   1. Instance Count
   2. Instance Type
4. Storage Configuration
   1. Storage Type
   2. Volume Type
   3. Volume Size
5. Snaposhot Configuration

申请流程

1. Application team submits an Infrastructure Architecture type RITM to ED (*INFRADELV-OPER-CIOENVDELV*).

The ticket contains the requirements above. (See I. Requirements)

1. ED assignee submits a CHG to ID (*INFRADELV-CIO-Infrastructure Design*) for TA357 update.
2. ED assignee creates the AWS ElasticSearch in AWS.
3. Application team verifies the configuration and provides sign off.

CIO Elasticsearch服务创建负责：ED Team

AWS Elasticsearch服务版本：5.3 (可以跟ED确认)

服务实例命名规则：AppName-AIRID-ENV

流程

访问控制可通过CAPP申请，如开通EC2访问ES的权限(resource based permissions)

CAPP: <https://ciocloudaccess.accenture.com/>

**附录**

参考链接：

* Elasticsearch安装及配置(旧版本): <http://blog.csdn.net/tzhuwb/article/details/77430815>
* Head插件(Web IDE)：<http://blog.csdn.net/qq3401247010/article/details/78742524>
* PHP客户端: <https://github.com/wizacha/AwsSignatureMiddleware>
* PHP创建/搜索/删除索引数据：<https://www.cnblogs.com/life_lt/p/6122767.html>