

57个经典实例  
3个综合项目案例

# Elasticsearch

## 搜索引擎开发实战

罗刚 张子宪◎编著

深入剖析大规模分布式搜索引擎的实现原理，详解Elasticsearch开发搜索引擎的相关技术  
涵盖大数据搜索引擎融合、自然语言处理与搜索引擎融合、Spring Boot与Vue.js前端融合等相关技术

详解多个搜索算法，每个算法都有广泛的应用前景  
通过大量实例和综合案例手把手带领读者快速上手  
书中的实例和综合案例大多来源于作者负责的实际项目



机械工业出版社  
China Machine Press





# Elasticsearch

## 搜索引擎开发实战



罗刚 张子宪 编著



机械工业出版社  
China Machine Press

## 图书在版编目 ( CIP ) 数据

Elasticsearch搜索引擎开发实战/罗刚, 张子宪编著. —北京: 机械工业出版社, 2018.6

ISBN 978-7-111-60348-1

I. E… II. ①罗… ②张… III. 搜索引擎—程序设计 IV. ①TP391.3

中国版本图书馆CIP数据核字 ( 2018 ) 第145805号

本书结合Elasticsearch在工程中的实际应用, 详细介绍了使用Elasticsearch开发支持中文和英文搜索引擎的相关技术, 从而实现系统监控。

本书共分为8章, 内容涵盖了Elasticsearch搜索引擎开发的环境安装与配置; 实现一个简单的网站搜索; 开发中文搜索引擎; Mapping详解; 源代码分析; 提高搜索相关性; 使用Spring Boot开发搜索界面; 使用Elasticsearch和相关软件实现系统监控; 搜索引擎开发案例分析。

本书非常适合信息检索技术爱好者、搜索引擎开发人员和搜索引擎优化 ( SEO ) 人员阅读, 也适合作为高等院校信息检索课程的教材或教学参考书。

## Elasticsearch 搜索引擎开发实战

出版发行: 机械工业出版社 ( 北京市西城区百万庄大街22号 邮政编码: 100037 )

责任编辑: 欧振旭 李华君

责任校对: 姚志娟

印刷: 中国电影出版社印刷厂

版次: 2018年7月第1版第1次印刷

开本: 186mm×240mm 1/16

印张: 15.75

书号: ISBN 978-7-111-60348-1

定价: 69.00元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: ( 010 ) 88379426 88361066

投稿热线: ( 010 ) 88379604

购书热线: ( 010 ) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光/邹晓东

# 前言

搜索引擎在人们的日常生活中发挥着越来越重要的作用。随着开源软件的普及与发展，涌现出了许多优秀的搜索软件，如 Elasticsearch、Solr 等。其中，Elasticsearch 以大规模分布式搜索见长，而 Solr 则以分面搜索见长。

本书选择 Elasticsearch 作为实现搜索引擎的工具。Elasticsearch 具有强大的分布式搜索和可视化功能，不仅丰富了实现搜索引擎的方法，而且还使复杂抽象的数据结构与算法变得直观而鲜活，因此在国外被迅速地引入到人工智能的相关课程中。

本书全面、系统地介绍了分布式搜索引擎的相关内容 & Elasticsearch 中的 Java 代码实现。本书内容既注重基础知识，又非常注重实践，每章都提供了大量的实例程序。读者可以通过这些实例快速上手，并迅速提高搜索引擎开发技术。通过对本书内容的学习，读者不仅可以掌握搜索引擎开发的基本知识，而且还可以灵活地将 Elasticsearch 运用到解决实际问题当中，从而提升工作效率。

## 本书特色

### 1. 内容全面，结构合理

本书首先介绍了 Elasticsearch 的安装和基本使用方法，然后介绍了从搜索到内容监控等方方面面的知识。在内容安排上，本书根据读者的认知规律对学习梯度做了合理安排，降低了学习难度。

### 2. 讲解详尽，实例丰富

本书对每个技术要点都做了细致入微的介绍，并且在讲解的过程中提供了丰富的实例，而且每个实例都经过精挑细选，具有很强的针对性，特别是本书最后的应用案例，更是对相关技术的一个全面应用。另外，书中所有实例的实现代码都考虑了通用性，读者可以直接将代码移植过来加以修改，即可解决自己的实际问题。

### 3. 语言通俗，图文并茂

本书用通俗易懂的语言进行讲解，尽量避免生疏的专业术语。在讲解一些重要知识点

时，书中给出了大量的图示及实例运行结果，帮助读者更加直观、高效地理解所学内容。

#### 4. 提供配套教学PPT，使学习更高效

为了便于读者高效、直观地学习本书内容，作者特意针对每章的重点内容制作了教学PPT，这些PPT和本书的实例源文件都会免费提供给读者下载。

## 本书内容

本书共分8章，具体内容介绍如下：

第1章 Elasticsearch 开发搜索引擎应用，主要介绍了搜索引擎开发方面的一些基础知识和 Elasticsearch 开发环境的安装，并对 Java API 与 Elasticsearch 搜索集群的交互也做了介绍。

第2章开发中文搜索引擎，主要介绍了中文搜索引擎开发的相关内容，包括中文分词原理和中文分词插件开发等。

第3章 Mapping 详解，主要介绍了 Mapping 概念及如何使用 Mapping，包括 Mapping 索引、Mapping 数据类型、Mapping 参数和动态 Mapping 等。

第4章深入源码分析，详细分析了 Elasticsearch 源代码，主要内容包括 Lucene 源码分析、启动搜索服务、Guice 框架、日期和时间库、Transport 模块、线程池、模块、Netty 通信框架、缓存、分布式、Zen 发现机制、联合搜索和 JVM 字节码等。

第5章提高搜索相关性，主要介绍了向量空间检索模型、BM25 检索模型、学习评分、查询意图识别和图像特征提升检索体验等内容。

第6章搜索界面开发，涵盖的主要内容包括使用 Searchkit 实现搜索界面；Spring Boot 入门；Java 模板引擎 Pebble 介绍；通过 Spring-data-elasticsearch 项目访问 Elasticsearch；REST 基本概念；使用 Vue.js 开发搜索界面；使用 Vue.js Paginator 插件实现翻页；实现搜索接口；Suggester 搜索词提示；Word2vec 挖掘相关搜索词；部署网站；使用 Rust 开发搜索界面等。

第7章 Elastic 栈系统监控，主要介绍了使用 Elasticsearch 和相关软件实现系统监控，包括管理 Elasticsearch 集群、Logstash 数据处理工具、Filebeats 文件收集器、消息过期、Kibana 可视化平台、Flume 日志收集系统、Kafka 分布式流平台和 Graylog 日志管理平台等内容。

第8章案例分析，主要介绍了双语句对搜索、内容管理系统站内检索，以及使用 Elasticsearch 搜索公开的药物临床试验项目信息等几个案例。

## 本书读者对象

- 信息检索技术爱好者；

- 搜索引擎开发人员；
- 搜索引擎优化（SEO）人员；
- 从事算法研究的技术人员；
- 高等院校理工科专业的学生和老师。

## 本书配套资源及获取方式

为了方便读者高效学习，本书特意提供了以下配套资源：

- 本书配套教学 PPT；
- 本书源代码文件；
- 本书涉及的一些开发工具的安装包。

这些配套资源需要读者自行下载，请登录机械工业出版社华章公司的网站 [www.hzbook.com](http://www.hzbook.com)，搜索到本书，然后在页面上的“资料下载”模块下载即可。

## 本书作者

本书由罗刚主笔编写，其他参与编写的人员有张子宪、沙芸、柳若边、崔智杰、石天盈、张继红、罗庭亮。

在此感谢我的家人、同事及所有在本书写作过程中提供过帮助的人！另外，本书在编写过程中参考了一些开源代码，在此对相关作者也一并表示感谢！

虽然我们对书中所述内容都尽量核实，并进行了多次校对，但由于写作时间仓促，加之作者水平所限，书中可能还存在疏漏和错误之处，恳请广大读者批评、指正。联系我们，请发电子邮件到 [hzbook2017@163.com](mailto:hzbook2017@163.com)。

罗刚  
于北京

# 目录

## 前言

第 1 章 Elasticsearch 开发搜索引擎应用	1
1.1 搜索引擎开发需求	1
1.2 准备开发环境	1
1.2.1 Windows 命令行 cmd	1
1.2.2 在 Windows 下使用 Java	3
1.2.3 Linux 终端	5
1.2.4 在 Linux 下使用 Java	9
1.2.5 Eclipse 集成开发环境	10
1.3 了解 Elasticsearch	10
1.3.1 JSON 数据格式	11
1.3.2 Elasticsearch 基本概念	12
1.3.3 HTTP 协议	13
1.4 Elasticsearch 安装和配置	16
1.4.1 安装 Elasticsearch	16
1.4.2 运行 Elasticsearch 作为服务进程	19
1.5 实现一个简单的网站搜索	21
1.5.1 定义索引结构	23
1.5.2 导入数据	26
1.5.3 查询 API	27
1.5.4 实现搜索界面	29
1.6 本章小结	35
第 2 章 开发中文搜索引擎	36
2.1 中文分词原理	36
2.1.1 最长匹配方法	36
2.1.2 自己写分析器	42
2.1.3 概率语言模型的分词方法	44
2.1.4 中文分词插件原理	52
2.1.5 开发中文分词插件	54
2.1.6 支持 Elasticsearch 的插件	57
2.1.7 中文分析器提供者	59

2.1.8 字词混合索引	61
2.2 提高分词准确度	63
2.3 本章小结	65
<b>第3章 Mapping 详解</b>	<b>66</b>
3.1 索引模式	66
3.1.1 创建模式	66
3.1.2 修改模式	68
3.2 Mapping 数据类型	69
3.3 Mapping 参数	70
3.4 动态 Mapping	71
3.4.1 使用动态 Mapping	72
3.4.2 实现原理	72
3.5 本章小结	74
<b>第4章 深入源码分析</b>	<b>75</b>
4.1 Lucene 源码分析	75
4.1.1 使用 Lucene	75
4.1.2 Ivy 管理依赖项	77
4.1.3 源码结构介绍	77
4.1.4 并发控制	82
4.2 启动搜索服务	88
4.3 Guice 框架	89
4.4 日期和时间库——Joda-Time	91
4.5 Transport 模块	91
4.6 线程池	92
4.7 模块	93
4.8 Netty 通信框架	93
4.9 缓存	94
4.10 分布式	95
4.11 Zen 发现机制	95
4.12 联合搜索	97
4.13 JVM 字节码	98
4.13.1 编译代码	99
4.13.2 同步相关指令	99
4.14 本章小结	100
<b>第5章 提高搜索相关性</b>	<b>102</b>
5.1 向量空间检索模型	102
5.2 BM25 检索模型	105
5.2.1 使用 BM25 检索模型	108
5.2.2 参数调优	108



5.3	学习评分	109
5.3.1	基本原理	109
5.3.2	准备数据	110
5.3.3	Elasticsearch 学习排名	112
5.4	查询意图识别	112
5.5	图像特征提升检索体验	113
5.6	本章小结	116
<b>第 6 章</b>	<b>搜索界面开发</b>	<b>118</b>
6.1	使用 Searchkit 实现搜索界面	118
6.2	Spring Boot 入门	122
6.2.1	可执行的 WAR	125
6.2.2	spring-boot-devtools 模块实现热部署	136
6.3	Java 模板引擎 Pebble 介绍	136
6.4	通过 Spring-data-elasticsearch 项目访问 Elasticsearch	141
6.5	REST 基本概念	149
6.6	使用 Vue.js 开发搜索界面	154
6.7	使用 Vue.js Paginator 插件实现翻页	157
6.8	实现搜索接口	161
6.8.1	编码识别	161
6.8.2	布尔搜索	163
6.8.3	搜索结果重定向	164
6.8.4	搜索结果排序	165
6.8.5	实现相似文档搜索	166
6.9	Suggester 搜索词提示	167
6.9.1	拼音提示	169
6.9.2	部署总结	169
6.9.3	相关搜索	170
6.9.4	再次查找	172
6.9.5	搜索日志	172
6.10	Word2vec 挖掘相关搜索词	174
6.11	部署网站	179
6.11.1	部署到 Web 服务器	179
6.11.2	防止攻击	181
6.12	使用 Rust 开发搜索界面	184
6.13	本章小结	184
<b>第 7 章</b>	<b>Elastic 栈系统监控</b>	<b>186</b>
7.1	管理 Elasticsearch 集群	186
7.1.1	写入权限控制	187
7.1.2	使用 X-Pack	188

7.1.3	快照	189
7.2	Logstash 数据处理工具	190
7.2.1	使用 Logstash	190
7.2.2	插件	192
7.2.3	数据库输入插件	192
7.2.4	开发插件	193
7.3	Filebeat 文件收集器	193
7.4	消息过期	194
7.5	Kibana 可视化平台	195
7.6	Flume 日志收集系统	196
7.7	Kafka 分布式流平台	197
7.8	Graylog 日志管理平台	198
7.9	本章小结	202
<b>第 8 章</b>	<b>案例分析</b>	<b>204</b>
8.1	双语句对搜索	204
8.1.1	爬虫抓取双语句对	204
8.1.2	英文分词	205
8.1.3	句子切分	205
8.1.4	标注词性	207
8.1.5	词对齐	209
8.1.6	索引数据	213
8.2	内容管理系统站内检索	214
8.2.1	MySQL 数据库	214
8.2.2	RESTful API 管理索引	215
8.2.3	自动客服机器人	217
8.3	搜索文档	225
8.3.1	爬虫抓取信息	225
8.3.2	在 Linux 下使用.NET	233
8.3.3	NEST 客户端	235
8.4	本章小结	239
<b>参考文献</b>		<b>240</b>

# 第 1 章 Elasticsearch 开发搜索引擎应用

信息时代，可供获取的数据大量涌现。那么如何通过搜索引擎从这些数据中挖掘出有价值的信息呢？正是基于这个需求，开源大数据搜索引擎 Elasticsearch 应运而生。

## 1.1 搜索引擎开发需求

网站搜索的一般需求有如下几点。

- 支持微服务：微服务架构模式可以用来构建复杂应用。
- 弹性负载：通过将搜索访问请求自动分发到多个服务节点上来扩展搜索系统对外的服务能力，实现应用程序容错。
- 容易部署：即集成的功能，不依赖第三方的分布式应用程序协调服务。
- 安全控制：控制非法的外部访问。
- 管理界面：管理搜索集群的健康状况，方便查看数据分布情况等。

## 1.2 准备开发环境

Elasticsearch 采用 Java 语言开发，所以我们需要先准备基本的 Java 开发工具 JDK，然后再准备运行在 JDK 上的 Eclipse。


### 1.2.1 Windows 命令行 cmd

假设有一个标准件工厂，在车间生产产品，在工地使用这些产品。与之类似，一般是在集成开发环境中开发软件，如果在 Windows 操作系统中运行开发的软件，则往往通过 Windows 命令行来运行。

在图形化用户界面出现之前，人们就是用命令行来操作计算机的。Windows 命令行是通过 Windows 系统目录下的 cmd.exe 程序执行的。执行这个程序最直接的方式是找到该程序，然后双击，但 cmd.exe 程序并没有桌面快捷启动图标，所以启动时比较麻烦。

鉴于此，可以在“开始”菜单的运行窗口中直接输入程序名，回车后运行这个程序。

具体操作方法：单击“开始”|“运行”命令，打开资源管理器中的运行程序窗口；或者直接使用快捷键——窗口键+R 键，打开运行程序窗口。然后输入程序名 `cmd` 后单击“确定”按钮，弹出命令提示窗口。因为可以通过这个黑屏的窗口直接输入相应命令来控制计算机，所以也称其为控制台窗口。

说明：Console，即控制台。遥控器上有控制面板，更复杂的设备往往有控制台。例如，一台机床或者数控设备的控制箱，通常会被称为控制台。顾名思义，控制台就是一个直接控制设备的台面，往往是一个上面有很多控制按钮的面板。在计算机里，把直接连接在计算机上的键盘和显示器叫做控制台。

通常用扩展名来表示文件的类别，如 `exe` 表示可执行文件。文件名称由文件名和扩展名组成，文件名和扩展名之间由小数点分隔，如 `java.exe`。

当我们建立或修改一个文件时，必须向 Windows 指明该文件的位置。文件的位置由三部分组成：驱动器、文件所在路径和文件名。路径是由一系列路径名组成的，这些路径名之间用“\”分开，如 `C:\Program Files\Java\jdk1.8.0_03\bin\java.exe`。

开始的路径一般是 `C:\Users\Administrator`，就像公园的地图上往往会标出游客的当前位置。Windows 命令行也有当前路径的概念，如 `C:\Users\Administrator` 就是当前路径。

可以用 `cd` 命令改变当前路径，例如，改变到 `C:\Program Files\Java\jdk1.8.0_03` 路径，可以用如下命令：

```
C:\Users\Administrator>cd C:\Program Files\Java\jdk1.8.0_03
```

如果输入“`cd d:`”命令，这样的效果是改变当前路径到“`d:`”目录下。所以切换盘符不能使用 `cd` 命令，而是直接输入盘符的名称。例如想要切换到 D 盘，可以使用如下命令：

```
C:\Users\Administrator>d:
```

系统约定从指定的路径找可执行文件，这个路径通过 `PATH` 环境变量指定。环境变量是一个“变量名=变量值”的对应关系，每一个变量都有一个或者多个值与之对应。如果是多个值，则这些值之间用分号隔开。例如，`PATH` 环境变量可能对应这样的值：“`C:\Windows\system32;C:\Windows`”，表示 Windows 会从 `C:\Windows\system32` 和 `C:\Windows` 两个路径下寻找可执行文件。

设置或者修改环境变量的具体操作步骤是：首先在 Windows 桌面右击“我的电脑”，在弹出的快捷菜单中选择“属性”命令，在弹出的对话框中选择“高级”选项，然后在弹出的对话框中单击“环境变量”按钮，在弹出的对话框中设置用户变量或者系统变量，最后再设置环境变量 `PATH` 的值。

如果是用 Windows 7 以上的操作系统，可能找不到“我的电脑”快捷图标，其实打开桌面上“我的电脑”，就是运行资源管理器。打开资源管理器的另外一种方法是：按住键盘上的窗口键不放，然后再按 `E` 键之后选择“属性”标签，后面的操作相同，不再赘述。

环境变量设置完成后，需要重新启动命令行才能设置生效。为了检查环境变量是否已设置正确，可以在命令行中显示指定环境变量的值，需要用到 `echo` 命令。`echo` 命令用来

显示一段文字。例如：

```
C:\Users\Administrator>echo Hello
```

执行上面的命令后，将在命令行输出：

```
Hello
```

如果要引用环境变量的值，可以用前后两个百分号把变量名包围起来，如“%变量名%”。例如，使用 `echo` 命令显示环境变量 `PATH` 中的值：

```
C:\Users\Administrator>echo %PATH%
```

## 1.2.2 在 Windows 下使用 Java

本节首先介绍如何安装 JDK，然后介绍如何在命令行开发 Java 程序。Java 开发环境简称 JDK (Java Development Kit)，JDK 包括 Java 运行环境 (Java Runtime Environment)、一堆 Java 工具和 Java 基础类库。可以从 Java 官方网站 <http://www.oracle.com/technetwork/java/index.html> 下载得到 JDK，注意不是 <http://www.java.com> 下的 Java 虚拟机。

进入官网后，选择下载 Java SE，也就是 Java 的标准版本，然后选择 Latest Release 也就是最新发布的安装程序，完整的 JDK 版本号中包括大版本号和小版本号。例如 1.7.0 中的大版本号是 7，小版本号是 0，而 1.8.22 的大版本号是 8，小版本号是 22。因为可以在 Windows 或 Linux 等多种操作系统环境下开发 Java 程序，所以有多个操作系统的 JDK 版本可供选择。

因为 JDK 是有版权的，所以需要接受许可协议 (Accept License Agreement) 后才能下载。如果是在 Windows 环境下开发，就选择 Windows x86，这样会下载类似 `jdk-8u121-windows-i586.exe` 这样的文件，下载完毕后，使用默认方式安装 JDK 即可。

JDK 相关的文件都放在一个叫做 `JAVA_HOME` 的根目录下。JDK 根目录的命名格式是：`C:\Program Files\Java\jdk1.8.0_<version>`最后以一个数字类型的版本号结尾，如 10 或者 21 等。

因为一台计算机上可以安装多个 JDK 和 JVM，为了避免混乱，可以新增环境变量 `JAVA_HOME`，指定一个默认使用的 JDK。

使用 `echo` 命令检查环境变量 `JAVA_HOME`：

```
>echo %JAVA_HOME%  
C:\Program Files\Java\jdk1.8.0_10
```

Eclipse 集成开发环境只需要 `JAVA_HOME` 这一个环境变量即可。如果要检查 `JAVA_HOME` 是否已经正确设置，使用如下命令后显示虚拟机的版本号就表示设置正确了。

```
>"%JAVA_HOME%\bin\java -version  
java version "1.8.0_10-rc"  
Java(TM) SE Runtime Environment (build 1.8.0_10-rc-b28)  
Java HotSpot(TM) Client VM (build 11.0-b15, mixed mode, sharing)
```

如果还需要在 Windows 控制台下执行 Java 程序，则需要访问编译源代码的 `javac.exe` 或者执行 class 文件字节码的 `java.exe`。环境变量 `PATH` 指定了从哪里找 `java.exe` 这样的可

执行文件，可以通过多个路径查找可执行文件，这些路径以分号隔开。如果想在命令行运行 Java 程序，还可以修改已有的环境变量 PATH，增加 Java 程序所在的路径。例如，C:\Program Files\Java\jdk1.8.0\_10\bin。

然后检查环境变量 PATH：

```
>echo %PATH%
```

如果要检查 PATH 是否已经正确设置，只要在任何路径下输入 javac 命令都能显示 javac 的用法，就表示设置正确了，也可以用第一个 Java 程序试验一下。

新建一个 Java 项目后，在这个项目的 src 路径下新建一个叫做 Search 的 Java 类：

```
public class Search {
    public static void main (String args[]) {
        System.out.println("Hello Search!");
    }
}
```

运行结果如下：

```
>javac Search.java
>java Search
```

看运行结果是否显示 Hello Search!

最简单的方法是可以使用 javac 构建出 class 文件，对于复杂的项目，一般是使用工具构建项目源代码。Gradle 就是一个可用于构建 Java 项目的工具，Elasticsearch 本身也是使用 Gradle 构建。可以下载二进制文件来安装 Gradle，网址如下：

<https://services.gradle.org/distributions/gradle-3.5-bin.zip>。

在 Windows 上自动设置 Gradle 环境变量的脚本如下：

```
set input=F:\soft\gradle-3.5
echo gradle 路径为%
set gradlePath=%
::创建 GRADLE_HOME
wmic ENVIRONMENT create
name="GRADLE_HOME",username="<system>",VariableValue="%javaPath%"
call set xx=%Path%;%gradlePath%\bin
::echo %xx%
::将环境变量中的字符重新赋值到 path 中
wmic ENVIRONMENT where "name='Path' and username='<system>'" set
VariableValue="%xx%"
pause
```

打开控制台并运行 gradle -v 命令以显示版本来验证安装是否成功，例如：

```
C:\Users\Administrator>gradle -v
```

显示如下输出：

```
-----
Gradle 3.5
-----
```

```
Build time:   2017-04-10 13:37:25 UTC
Revision:    b762622a185d59ce0cfc9cbc6ab5dd22469e18a6
```

```
Groovy:      2.4.10
Ant:         Apache Ant(TM) version 1.9.6 compiled on June 29 2015
JVM:        1.8.0_121 (Oracle Corporation 25.121-b13)
OS:         Windows Server 2008 6.0 x86
```

可以在 Gradle 构建中使用标准和定制的 Ant 任务,就像在 Ant 自身中使用一样。另外,可以导入现有的 Ant 脚本,就像下面这样简单:

```
ant.importBuild 'build.xml'
```

## 1.2.3 Linux 终端

虽然使用 Linux 操作系统办公的人不多,但是很多大数据应用都运行在 Linux 操作系统下。

首先在 Windows 下安装 Chrome 浏览器,然后可以通过网址 <http://sshy.us/> 登录 Linux 服务器。如果是用 root 账户登录,则终端提示符是“#”,否则终端提示符是“\$”。

如果有现成的 Linux 服务器可用,可以使用支持 SSH 协议的终端仿真程序 SecureCRT 连接到远程 Linux 服务器上,因为可以保存登录密码,所以比较方便。除了 SecureCRT,还可以使用开源软件 PuTTY (<http://www.chiark.greenend.org.uk/~sgtatham/putty>),以及可以保存登录密码的 PuTTY Connection Manager。

使用 VMware、Linux 可以运行在 Windows 系统下,VMware 可以让 Linux 运行在虚拟机中,而且不会破坏原来的 Windows 操作系统。

首先要准备好 VMware,当然仍然需要 Linux 光盘文件。就好像华山派有剑宗和气宗,Linux 也有很多种版本,例如 RedHat、Ubuntu 及 SUSE,这里选择 CentOS (<http://www.centos.org/>)。

也可以在 Windows 下安装 Cygwin,使用它来练习 Linux 的常用命令。

如果需要安装软件,可以下载 RPM 安装包,然后使用 RPM 安装。但操作系统对应的 RPM 安装包找起来比较麻烦,一个软件包可能依赖其他的软件包,为了安装一个软件可能需要下载多个它所依赖的软件包。

为了简化安装操作,可以使用 Yum (Yellow dog Updater, Modified) 来安装,也称其为黄狗升级管理器。Yum 会自动计算出程序之间的相互关联性,并且计算出完成软件包的安装需要哪些步骤,这样在安装软件时,不会再被那些关联性问题所困扰。

Yum 会自动从网络上下载并安装软件,有点类似于 360 软件管家,但是不会有商业倾向的推销软件。例如,安装支持 wget 和 rzsz 命令的软件有:

```
#yum install wget
#yum install lrzsz
```

可以使用 Nodepad++ 自带的插件 NppFTP 编辑 Linux 下的文件。有些生产环境的集群通过跳板机才能接触到。为了方便在服务器端管理和开发 Elasticsearch 相关应用,可以采用 Micro (<https://github.com/zyedidia/micro>) 这样的终端文本编辑器。

可以使用 DNF 安装 Micro，在安装 DNF 前，必须先安装并启用 epel-release 依赖。使用 Yum 安装 epel-release 的命令如下：


```
# yum install epel-release
```

如果没有 DNF 安装工具软件，也可以直接安装 Micro 的预编译版本。使用 wget 下载 Micro：

```
# wget https://github.com/zyedidia/micro/releases/download/nightly/micro-1.3.4-67-linux64.tar.gz
# tar -xvf ./micro-1.3.4-67-linux64.tar.gz
```

编辑/etc/profile 配置文件，增加 Micro 所在的路径到 PATH 环境变量/home/soft/micro-1.3.4-67。

```
# ./micro /etc/profile
```

说明：和 Windows 不同，Linux 操作系统下的路径名之间用“/”分开。./micro-1.3.4-67-linux64.tar.gz 表示当前路径下的 micro-1.3.4-67-linux64.tar.gz 文件。

增加如下命令：

```
export PATH=/home/soft/micro-1.3.4-67:$PATH
```

可以使用 Micro 来编辑配置文件：

```
# micro /etc/security/limits.conf
```

保存文件后，按 Ctrl+Q 组合键退出。很多 Linux 环境都带有 Python，如果版本太旧，读者可以自行安装。

```
# yum install python34
```

下面先看下当前版本安装在了哪个目录下：

```
# which python
```

输出结果如下：

```
/usr/bin/python
```

一般使用 Bash 将用户可读的命令转换成计算机可理解的命令，并控制命令执行。

Bash 脚本中使用的特殊字符有：

```
#:Comments
~:home directory
```

在屏幕上打印“Hello”：

```
echo "Hello"
```

将 ABC 分配给 a：

```
a=ABC
```

输出 a 的值：

```
echo $a
```



在屏幕上打印 ABC。

将 ABC.log 分配给 b:

```
b=$a.log
```

输出 b 的值:

```
# echo $b
```

在屏幕上输出:

```
ABC.log
```

把文件 ABC.log 中的内容写入 testfile:

```
# cat $b > testfile
```

这里把 cat 命令的输出重定向到 testfile。

我们可以把重复执行的 Shell 脚本写入一个文本文件中。和 Windows 不同，Linux 不以文件后缀名作为系统识别文件类型的依据，但是可以作为我们识别文件的依据，因此我们可以将脚本文件以 .sh 结尾。

可以使用 Micro 创建一个类似 script.sh 的文件 micro script.sh，创建好脚本文件后就可以在文件内用脚本语言要求的格式编写脚本程序了。此外，还可以使用 touch 命令先创建一个空文件 touch script.sh。

在创建的脚本文件中输入以下代码并保存退出。

```
#!/bin/bash  
echo "hello world!"
```

然后添加脚本文件的可执行运行权限:

```
# chmod +x script.sh
```

运行文件 ./script.sh，结果如下:

```
hello world!
```

Shell 脚本中用 “#” 表示注释，相当于 C 语言的 “//” 注释。但如果 “#” 位于第一行开头并且是 “#!”（称为 Shebang）则例外，它表示该脚本使用后面指定的解释器/bin/sh 解释执行。每个脚本程序必须在开头包含 Shebang 语句。

例如，使用参数 n 检查语法错误:

```
# bash -n ./test.sh
```

如果 Shell 脚本中有语法错误，则会提示错误所在行；否则不输出任何信息。

智能系统需要根据不同的外部情况做出不同的处理，所以需要使用流程控制语句。下面简单介绍一下 if 和 case 语句。

if 语句的语法如下:

```
if [ condition ] then  
    command1
```

elif#和 else if 等价: