

Apache Kafka Quick Start Guide

Apache Kafka 2.0

入门与实践

[美] 劳尔·埃斯特拉达 著 张华臻 译

TP316.4/07

清华大学出版社

Apache Kafka 2.0 入门与实践

[美] 劳尔·埃斯特拉达 著
张华臻 译

清华大学出版社
北 京

内 容 简 介

本书详细阐述了与 Apache Kafka 2.0 相关的基本解决方案，主要包括配置 Kafka、消息验证、消息增强、序列化、模式注册表、Kafka Streams、KSQL、Kafka Connect 等内容。此外，本书还提供了相应的示例、代码，以帮助读者进一步理解相关方案的实现过程。

本书既可作为高等院校计算机及相关专业的教材和教学参考书，也可作为相关开发人员的自学教材和参考手册。

Copyright © Packt Publishing 2018. First published in the English language under the title *Apache Kafka Quick Start Guide*.

Simplified Chinese-language edition © 2019 by Tsinghua University Press. All rights reserved.

本书中文简体字版由 Packt Publishing 授权清华大学出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字：01-2019-4825

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目 (CIP) 数据

Apache Kafka 2.0 入门与实践 / (美) 劳尔·埃斯特拉达著；张华臻译. —北京：清华大学出版社，2019

书名原文：Apache Kafka Quick Start Guide

ISBN 978-7-302-53495-2

I. ①A… II. ①劳… ②张… III. ①分布式操作系统 IV. ①TP316.4

中国版本图书馆 CIP 数据核字 (2019) 第 179553 号

责任编辑：贾小红
封面设计：刘超
版式设计：文森时代
责任校对：马军令
责任印制：杨艳

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>, <http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座

邮 编：100084

社 总 机：010-62770175

邮 购：010-62786544

投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者：三河市国英印务有限公司

经 销：全国新华书店

开 本：185mm×230mm 印 张：9.25

字 数：225 千字

版 次：2019 年 9 月第 1 版

印 次：2019 年 9 月第 1 次印刷

定 价：69.00 元

产品编号：084070-01

译者序

Kafka 是由 Apache 软件基金会开发的一个开源流处理平台，是一种高吞吐量的分布式发布订阅消息系统，它可以处理消费者在网站中的所有动作流数据。时至今日，Apache Kafka 可用于采集数据、执行实时数据分析，并执行实时数据流处理。另外，Kafka 还可用于向复杂事件处理（CEP）架构中输入事件、部署于微服务架构中，并可实现于物联网架构中。

本书是一本适用于数据工程师、软件开发人员和数据架构师的快速入门指南，详细阐述了与 Apache Kafka 2.0 相关的基本解决方案，主要包括配置 Kafka、消息验证、消息增强、序列化、模式注册表、Kafka Streams、KSQL、Kafka Connect 等内容。本书注重于编程实现过程，并提供了相应的示例、代码，以帮助读者进一步理解相关方案的实现过程。

在本书的翻译过程中，除张华臻外，王辉、刘璋、刘晓雪、张博、刘祎等人也参与了部分翻译工作，在此一并表示感谢。

由于译者水平有限，难免有疏漏和不妥之处，恳请广大读者批评指正。

译者

前 言

自 2011 年以来，Kafka 突然呈爆炸式增长。超过三分之一的财富 500 强公司均采用了 Apache Kafka。这些公司包括旅游公司、银行、保险公司和电信公司。

同时，Uber、Twitter、Netflix、Spotify、Blizzard、LinkedIn 和 PayPal 等公司每天也采用 Apache Kafka 处理大量的消息。

时至今日，Apache Kafka 可用于采集数据、执行实时数据分析，并执行实时数据流处理。另外，Kafka 还可用于向复杂事件处理（CEP）架构中输入事件、部署于微服务架构中，并可实现于物联网架构中。

在数据流领域中，Kafka Streams 也面临着一些竞争对手，例如 Apache Spark、Apache Flink、Akka Streams、Apache Pulsar 和 Apache Beam。尽管如此，Apache Kafka 仍拥有一个无法替代的优势，即易于使用。Kafka 易于实现和维护，其学习曲线也相对平缓。

本书是一本实用的快速入门指南，并侧重于展示实际用例，且较少涉及 Kafka 架构方面的理论知识。本书旨在讲解 Apache Kafka 使用者所面临的日常问题。

适用读者

- 本书是一本适用于数据工程师、软件开发人员和数据架构师的快速入门指南。
- 本书注重于编程实现过程，同时详细介绍了 Apache Kafka 2.0 方面的知识。
- 书中的全部示例均采用 Java 8 实现，因此建议读者具有与 Java 8 相关的背景知识，这也是阅读本书的唯一前提条件。

本书内容

第 1 章：配置 Kafka。介绍了 Apache Kafka 2.0 的基础知识，包括如何安装、配置和运行 Kafka。此外，本章还讨论了如何利用 Kafka 代理和 topic 执行基本的操作。

第 2 章：消息验证。探讨了如何针对企业服务总线进行数据验证，其间还会涉及如何从输入流中过滤消息。

第 3 章：消息增强。将考查消息增强机制，对于企业服务总线来说，这也是一项较为重要的任务。消息增强可将附加信息整合至系统消息中。

第 4 章：序列化。将讨论如何构建序列化器和反序列化器，进而写入、读取或转换消息（二进制、原始字符串、JSON 或 Avro 格式）。

第 5 章：模式注册表。将利用 Kafka Schema Registry 验证、序列化、反序列化并维护消息的历史版本。

第 6 章：Kafka Streams。将阐述如何获取与消息分组相关的信息（消息流），以及如何获取附加信息。例如，如何利用 Kafka Streams 处理消息的聚合和合成操作。

第 7 章：KSQL。将讨论如何在 Kafka Streams 的基础上利用 SQL 操控事件流。

第 8 章：Kafka Connect。将介绍快速数据处理工具，以及如何与 Apache Kafka 结合使用以生成数据处理管线。其中，相关工具包括 Apache Spark 和 Apache Beam。

背景知识

在阅读本书时，建议读者具有 Java 8 方面的编程背景知识。

执行本书示例所需的最低配置是 Intel[®]Core i3 处理器、4GB RAM 和 128GB 磁盘空间。这里推荐使用 Linux 或 macOS 操作系统，因为相关示例并不完全支持 Windows 操作系统。

资源下载

读者可访问 <http://www.packtpub.com> 并通过个人账户下载示例代码文件。另外，在 <http://www.packtpub.com/support> 中注册成功后，我们将以电子邮件的方式将相关文件发与读者。

读者可根据下列步骤下载代码文件：

- 利用电子邮件地址和密码登录或注册我们的网站 www.packtpub.com。
- 单击 SUPPORT 选项卡。
- 单击 Code Downloads & Errata。
- 在 Search 文本框中输入书名。

当文件下载完毕后，确保使用下列最新版本软件解压文件夹：

- Windows 系统下的 WinRAR/7-Zip。
- Mac 系统下的 Zipeg/iZip/UnRarX。
- Linux 系统下的 7-Zip/PeaZip。

另外，读者还可访问 GitHub 获取本书的代码包，对应网址为 <https://github.com/PacktPublishing/Apache-Kafka-Quick-Start-Guide>。

此外，读者还可访问 <https://github.com/PacktPublishing/> 以了解丰富的代码和视频资源。

本书约定

本书通过不同的文本风格区分相应的信息类型。下面通过一些示例对此类风格以及具体含义的解释予以展示。

代码块如下所示。

```
{
  "event": "CUSTOMER_CONSULTS_ETHPRICE",
  "customer": {
    "id": "14862768",
    "name": "Snowden, Edward",
    "ipAddress": "95.31.18.111"
  },
  "currency": {
    "name": "ethereum",
    "price": "RUB"
  },
  "timestamp": "2018-09-28T09:09:09Z"
}
```


当某个代码块希望引起读者的足够重视时，一般会采用黑体表示，如下所示。

```
dependencies {
  compile group: 'org.apache.kafka', name: 'kafka_2.12',
    version: '2.0.0'
  compile group: 'com.maxmind.geoip', name: 'geoip-api',
    version: '1.3.1'
  compile group: 'com.fasterxml.jackson.core', name: 'jackson-core',
    version: '2.9.7'
}
```

命令行输入或输出则采用下列方式表达：

```
> <confluent-path>/bin/kafka-topics.sh --list --ZooKeeper
localhost:2181
```

 图标表示较为重要的说明事项。

 图标则表示提示信息和操作技巧。

读者反馈和客户支持

欢迎读者对本书的建议或意见予以反馈。

对此，读者可向 feedback@packtpub.com 发送邮件，并以书名作为邮件标题。若读者对本书有任何疑问，均可发送邮件至 questions@packtpub.com，我们将竭诚为您服务。

若读者针对某项技术具有专家级的见解，抑或计划撰写书籍或完善某部著作的出版工作，则可访问 www.packtpub.com/authors。

勘误表

尽管我们在最大程度上做到尽善尽美，但错误依然在所难免。如果读者发现谬误之处，无论是文字错误抑或是代码错误，还望不吝赐教。对此，读者可访问 <http://www.packtpub.com/submit-errata> 选取对应书籍，单击 Errata Submission Form 超链接，并输入相关问题的详细内容。

版权须知

一直以来，互联网上的版权问题从未间断，Packt 出版社对此类问题异常重视。若读者在互联网上发现本书任意形式的副本，请告知网络地址或网站名称，我们将对此予以处理。关于盗版问题，读者可发送邮件至 copyright@packtpub.com。

问题解答

若读者对本书有任何疑问，均可发送邮件至 questions@packtpub.com，我们将竭诚为您服务。

目 录

第 1 章 配置 Kafka	1
1.1 Kafka 简介	1
1.2 安装 Kafka	2
1.2.1 在 Linux 中安装 Kafka	4
1.2.2 在 macOS 中安装 Kafka	5
1.2.3 安装 Confluent Platform	7
1.3 运行 Kafka	8
1.4 运行 Confluent Platform	8
1.5 运行 Kafka 代理	10
1.6 运行 Kafka topic	13
1.7 命令行消息生产者	16
1.8 命令行消息消费者	17
1.9 使用 kafkacat	18
1.10 本章小结	19
第 2 章 消息验证	20
2.1 企业服务总线	20
2.2 事件建模	21
2.3 配置项目	23
2.4 从 Kafka 中读取数据	26
2.5 向 Kafka 中写入数据	29
2.6 运行处理引擎	31
2.7 验证器的 Java 编码	33
2.8 运行验证	35
2.9 本章小结	38
第 3 章 消息增强	39
3.1 获取地理位置	40
3.2 增强消息	42
3.3 析取货币价格	44

3.4	利用货币价格充实消息	46
3.5	运行引擎	48
3.6	析取天气数据	51
3.7	本章小结	52
第 4 章	序列化	53
4.1	Kafka 物联网公司 Kioto	53
4.2	项目配置	54
4.3	Constants 类	56
4.4	HealthCheck 消息	58
4.5	Java PlainProducer 类	59
4.6	运行 PlainProducer	62
4.7	Java PlainConsumer 类	63
4.8	Java PlainProcessor 对象	64
4.9	运行 PlainProcessor	67
4.10	自定义序列化器	68
4.11	Java CustomProducer 类	69
4.12	运行 CustomProducer	70
4.13	自定义反序列化器	71
4.14	Java CustomConsumer 类	72
4.15	Java CustomProcessor 类	73
4.16	运行 CustomProcessor	75
4.17	本章小结	76
第 5 章	模式注册表	77
5.1	Avro 简介	77
5.2	定义模式	78
5.3	启动 Schema Registry	79
5.4	使用 Schema Registry	80
5.4.1	在值主题下注册一个新的模式版本	80
5.4.2	在键主题下注册一个新的模式版本	81
5.4.3	将现有的模式注册至新的主题中	82
5.4.4	列出全部主题	82
5.4.5	通过全局唯一 ID 查询模式	82

5.4.6	列出注册于 healthchecks-value 主题下的全部模式版本	83
5.4.7	查询注册于 healthchecks-value 主题下的模式版本 1	83
5.4.8	删除注册于 healthchecks-value 主题下的模式版本 1	83
5.4.9	删除最近注册于 healthchecks-value 主题下的模式	83
5.4.10	删除注册于 healthchecks-value 主题下的全部模式版本	84
5.4.11	检测模式是否已注册于 healthchecks-key 主题下	84
5.4.12	模式兼容性测试	84
5.4.13	获取顶级兼容性配置	84
5.4.14	全局更新兼容性需求条件	85
5.4.15	更新 healthchecks-value 主题下的兼容性需求条件	85
5.5	Java AvroProducer	85
5.6	运行 AvroProducer	89
5.7	Java AvroConsumer 类	91
5.8	Java AvroProcessor 类	92
5.9	运行 AvroProcessor	94
5.10	本章小结	95
第 6 章	Kafka Streams	96
6.1	Kafka Streams 简介	96
6.2	项目配置	97
6.3	Java PlainStreamsProcessor 类	98
6.4	运行 PlainStreamsProcessor	101
6.5	Kafka Streams 扩展	102
6.6	Java CustomStreamsProcessor 类	103
6.7	运行 CustomStreamsProcessor	104
6.8	Java AvroStreamsProcessor 类	105
6.9	运行 AvroStreamsProcessor	107
6.10	延迟事件处理	108
6.11	基本场景	109
6.12	延迟事件的生成	110
6.13	运行 EventProducer	112
6.14	Kafka Streams 处理程序	113
6.15	运行流处理程序	115
6.16	流处理程序分析	116

6.17	本章小结	117
第 7 章	KSQL	118
7.1	KSQL 简介	118
7.2	运行 KSQL	119
7.3	使用 KSQL CLI	120
7.4	利用 KSQL 处理数据	122
7.5	写入 topic 中	123
7.6	本章小结	126
第 8 章	Kafka Connect	127
8.1	Kafka Connect 简介	127
8.2	项目配置	128
8.3	Spark 流处理程序	129
8.4	从 Spark 中读取 Kafka	130
8.5	数据转换	131
8.6	数据处理	133
8.7	从 Spark 中写入至 Kafka	133
8.8	运行 SparkProcessor	135
8.9	本章小结	136

第 1 章 配置 Kafka

本章主要介绍 Kafka 的具体含义，以及与该技术相关的概念，包括 broker、topic、生产者和消费者。除此之外，本章还将讨论如何采用命令行构建简单的生产者和消费者，以及如何安装 Confluent Platform。本章可视为后续章节的基础内容。

本章主要涉及以下主题：

- Kafka 简介。
- 安装 Kafka（Linux 和 macOS 环境）。
- 安装 Confluent Platform。
- 运行 Kafka。
- 运行 Confluent Platform。
- 运行 Kafka 代理。
- 运行 Kafka topic。
- 命令行消息生产者。
- 命令行消息消费者。
- 使用 kfkcat。

1.1 Kafka 简介

Apache Kafka 是一个开源流平台。当读者正在阅读本书时，可能已经了解到 Kafka 在不牺牲速度和效率的前提下具有良好的水平伸缩性。

Kafka 的核心采用 Scala 编写，而 Kafka Stream 和 KSQL 则采用 Java 编写。另外，Kafka 服务器可在多种操作系统中运行，例如 Unix、Linux、macOS，甚至是 Windows。考虑到 Kafka 一般在 Linux 服务器的生产环境中运行，因而本书中的示例是为在 Linux 环境中运行而设计的。另外，本书中的示例也兼顾到 bash 环境中的应用。

本章将详细阐述如何安装、配置和运行 Kafka。作为一本快速入门指南，本书并未涉及太多的理论细节。当前，读者有必要理解以下 3 项内容：

- Kafka 是一个服务总线：为了连接异构应用程序，需要实现一种消息发布机制，并在其间发送和接收消息。相应地，消息路由器也称作消息代理。Kafka 是一个消息代理，同时也是一种快速处理客户端之间路由消息的解决方案。
- Kafka 架构包含两个指令：第一个指令不会阻塞生产者；第二个指令将隔离生产

者和消费者。生产者不应知道对应的消费者是谁，因此 Kafka 采用了哑代理和智能模型。

- ❑ Kafka 是一个实时消息系统：除此之外，Kafka 还是一个带有发布-订阅模型的软件解决方案，具备开源、分布式、分区、复制和日志提交等特性。

Apache Kafka 中的其他一些概念和术语还包括：

- ❑ 集群：表示一组 Kafka 代理。
- ❑ Zookeeper：表示为一个集群协调器——也可包含不同服务的工具，这些服务是 Apache 生态系统中的一部分内容。
- ❑ 代理 (broker)：这是一个 Kafka 服务器，同时也代表了 Kafka 服务器进程自身。
- ❑ topic：表示为一个队列（包含日志分区）；一个代理可运行多个 topic。
- ❑ 偏移：表示每个消息的标识符。
- ❑ 分区：这是一个不可变的、有序的记录序列，持续附加到结构化提交日志中。
- ❑ 生产者：表示为一个程序并向 topic 发布数据。
- ❑ 消费者：表示为一个程序，处理来自 topic 中的数据。
- ❑ 保存期：保持消息可用的时间。

在 Kafka 中，存在 3 种集群类型，具体如下：

- ❑ 单一节点：单一代理。
- ❑ 单一节点：多个代理。
- ❑ 多个节点：多个代理。

在 Kafka 中，仅包含 3 种消息传递方式，具体如下：

- ❑ 永不重发：一旦发送后，此类消息将不再被重新发送，因而消息可能会丢失。
- ❑ 可以重新发送：如果未接收到消息，可再次发送消息，因而消息永远不会丢失。
- ❑ 仅发送一次：消息仅发送一次，这也是最为困难的发送方式。由于消息仅发送一次，且不会被重发，因而消息的损失率为 0。

消息日志可以通过两种方式进行压缩，具体如下：

- ❑ 粗粒度：按时间压缩的日志。
- ❑ 细粒度：按消息压缩的日志。

1.2 安装 Kafka

Kafka 环境的安装包含以下 3 种方式：

- ❑ 下载可执行文件。
- ❑ 使用 brew（在 macOS 操作系统中）或 yum（在 Linux 操作系统中）。

□ 安装 Confluent Platform。

针对上述 3 种方式，第一步是安装 Java。具体来说，当前需要安装 Java 8。读者可访问 <http://www.oracle.com/technetwork/java/javase/downloads/index.html>，并下载最新版本的 JDK 8。

在本书编写时，Java 8 JDK 的最新版本为 8u191。

对于 Linux 用户，需要执行下列各项步骤。

(1) 将文件模式调整为可执行文件，如下所示。

```
> chmod +x jdk-8u191-linux-x64.rpm
```

(2) 访问 Java 安装目录，如下所示。

```
> cd <directory path>
```

(3) 利用下列命令运行 rpm 安装程序：

```
> rpm -ivh jdk-8u191-linux-x64.rpm
```

(4) 将 JAVA_HOME 变量添加到环境中。下面的命令将 JAVA_HOME 环境变量写入至/etc/profile 文件。

```
> echo "export JAVA_HOME=/usr/java/jdk1.8.0_191" >> /etc/profile
```

(5) 验证 Java 的安装结果，如下所示。

```
> java -version
java version "1.8.0_191"
Java(TM) SE Runtime Environment (build 1.8.0_191-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.191-b12, mixed mode)
```

在编写本书时，最新的 Scala 版本为 2.12.6。当在 Linux 中安装 Scala 时，可执行下列步骤。

(1) 访问 <http://www.scala-lang.org/download> 并下载最新版本的 Scala 二进制文件。

(2) 解压下载后的文件 scala-2.12.6.tgz，如下所示。

```
> tar xzf scala-2.12.6.tgz
```

(3) 向当前环境中添加 SCALA_HOME 变量，如下所示。

```
> export SCALA_HOME=/opt/scala
```

(4) 向 PATH 环境变量中加入 Scala bin 目录，如下所示。

```
> export PATH=$PATH:$SCALA_HOME/bin
```

(5) 当验证 Scala 安装结果时，可执行下列命令。

```
> scala -version
Scala code runner version 2.12.6 -- Copyright 2002-2018,
LAMP/EPFL and Lightbend, Inc.
```

当在机器上安装 Kafka 时，应确保设备上至少具有 4GB 的 RAM；针对 macOS 用户，对应的安装目录应为 `/usr/local/kafka/`；对于 Linux 用户，对应的安装目录应为 `/opt/kafka/`。用户应根据具体的操作系统创建此类目录。

1.2.1 在 Linux 中安装 Kafka

访问 Apache Kafka 下载页面 <http://kafka.apache.org/downloads>，如图 1.1 所示。

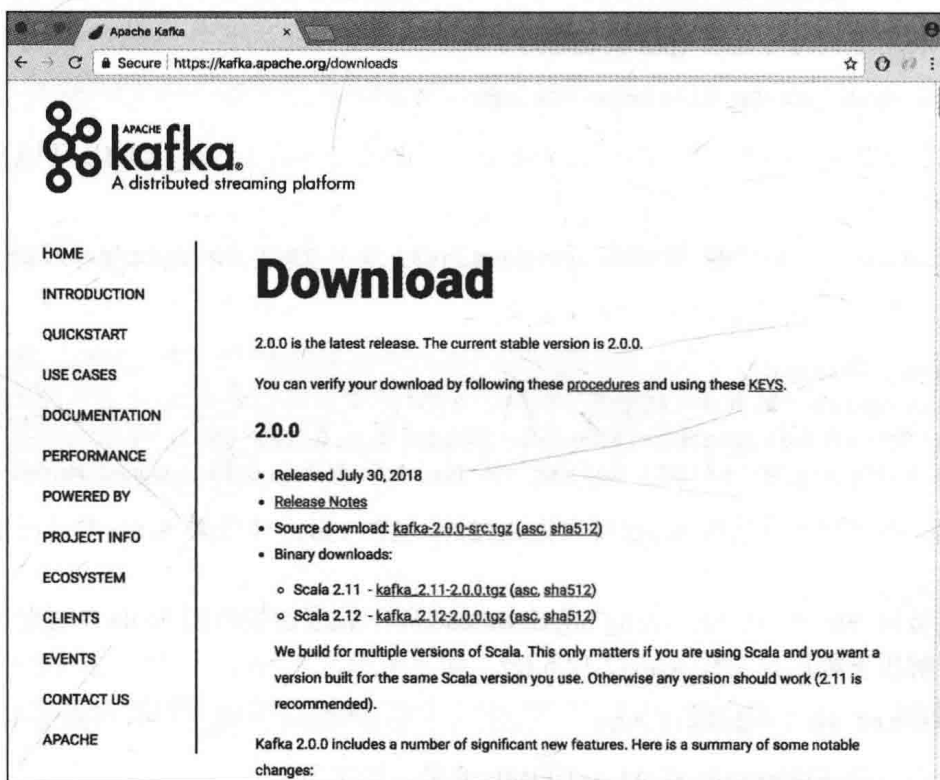


图 1.1 Apache Kafka 下载页面

在编写本书时，当前 Apache Kafka 的稳定版本为 2.0.0。需要注意的是，自版本 0.8.x 以来，Kafka 并未实现向后兼容。因此，不能将当前版本替换为 0.8 之前的版本。在下载了最新的可用版本后，下面继续安装。

TIP 提示：对于 macOS 用户，可利用 `/usr/local` 替换 `/opt/`。

当在 Linux 中安装 Kafka 时，可遵循下列各项步骤。

(1) 在 `/opt/` 目录中解压 `kafka_2.11-2.0.0.tgz` 文件，如下所示。

```
> tar xzf kafka_2.11-2.0.0.tgz
```

(2) 创建 `KAFKA_HOME` 环境变量，如下所示。

```
> export KAFKA_HOME=/opt/kafka_2.11-2.0.0
```

(3) 向 `PATH` 变量中添加 Kafka bin 目录，如下所示。

```
> export PATH=$PATH:$KAFKA_HOME/bin
```

至此，Java、Scala 和 Kafka 均已安装完毕。

当采用命令行方式执行上述命令时，对于 macOS 用户来说，存在一个功能强大的工具，即 `brew`（等同于 Linux 中的 `yum`）。

1.2.2 在 macOS 中安装 Kafka

当在 macOS 中安装 Kafka 时（之前需要安装 `brew`），可执行下列各项步骤。

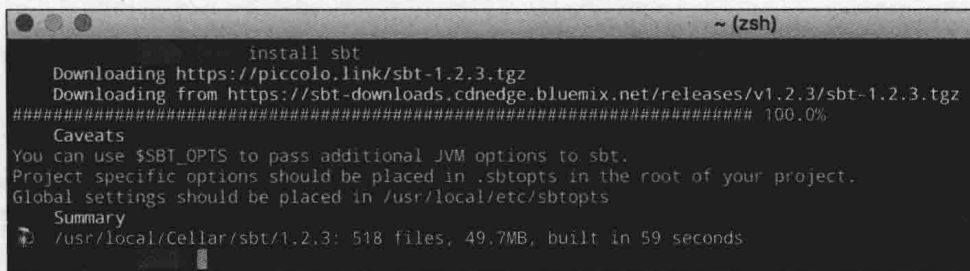
(1) 利用 `brew` 安装 `sbt`（即 Scala 构建工具），如下所示。

```
> brew install sbt
```

如果已在当前环境中安装了 `sbt`，可运行下列命令对其加以更新：

```
> brew upgrade sbt
```

对应的输出结果如图 1.2 所示。



```
install sbt
Downloading https://piccolo.link/sbt-1.2.3.tgz
Downloading from https://sbt-downloads.cdndedge.bluemix.net/releases/v1.2.3/sbt-1.2.3.tgz
##### 100.0%
Caveats
You can use $SBT_OPTS to pass additional JVM options to sbt.
Project specific options should be placed in .sbt_opts in the root of your project.
Global settings should be placed in /usr/local/etc/sbt_opts
Summary
/usr/local/Cellar/sbt/1.2.3: 518 files, 49.7MB, built in 59 seconds
```

图 1.2 Scala 构建工具的安装输出结果

(2) 利用 `brew` 安装 Scala，如下所示。

```
> brew install scala
```