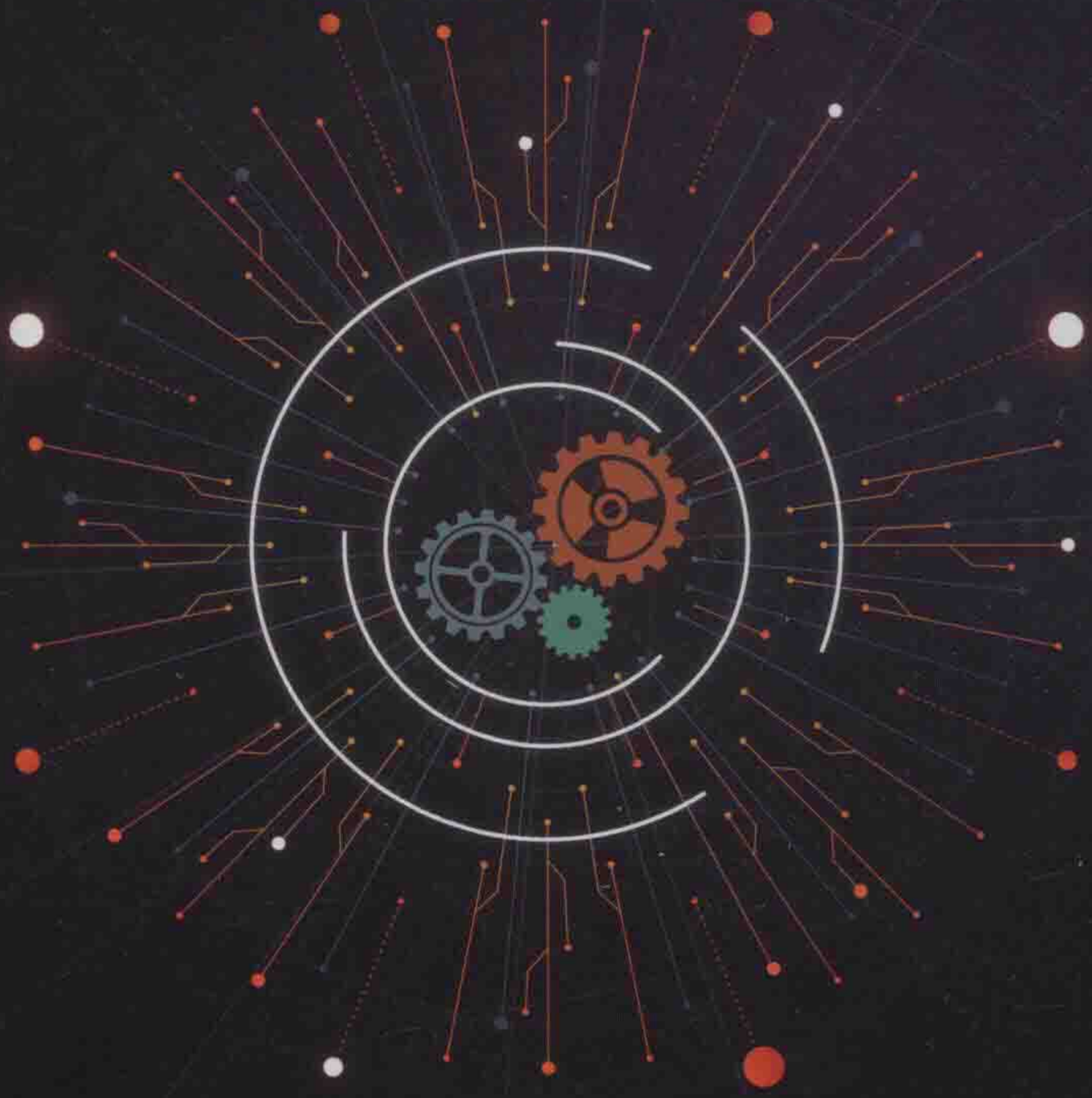
深入解析HyperLedger Fabric区块链平台使用方案
手把手教你快速部署应用服务，助力开发成功落地

Broadview
www.broadview.com.cn

HyperLedger Fabric 开发实战

快速掌握区块链技术

杨毅 编著



中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

HyperLedger Fabric
开发实战
快速掌握区块链技术

杨毅 编著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书系统地介绍了超级账本 HyperLedger Fabric v1.1 架构的设计和应用方法，包括环境及源码部署、Solo 多机部署、Kafka 集群部署、智能合约编写等。同时，针对第三方可插拔式插件 CouchDB 实战应用，Java-SDK 的应用、编写方案和具体接口执行策略进行了详细讲解。另外，本书以搭建一个反欺诈区块链平台项目为例进行了实战演练，读者可以快速掌握区块链技术。

本书适合区块链系统开发人员阅读，需要有一定的面向对象语言的基础，也可供对开发区块链系统感兴趣的高校师生参考。

图书在版编目（CIP）数据

HyperLedger Fabric 开发实战：快速掌握区块链技术 / 杨毅编著. —北京：电子工业出版社，2018.6
ISBN 978-7-121-34173-1

Ⅰ. ①H… Ⅱ. ①杨… Ⅲ. ①电子商务－支付方式－研究 Ⅳ. ①F713.361.3

中国版本图书馆 CIP 数据核字（2018）第 099665 号

责任编辑：宋亚东
印　　刷：天津千鹤文化传播有限公司
装　　订：天津千鹤文化传播有限公司
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱　邮编 100036
开　　本：787×980　1/16　印张 18.5　字数：388 千字
版　　次：2018 年 6 月第 1 版
印　　次：2018 年 6 月第 1 次印刷
定　　价：79.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819，faq@phei.com.cn。

前　言

HyperLedger Fabric 最初是由 Digital Asset 和 IBM 公司贡献的、由 Linux 基金会主办的一个超级账本项目，它是一个目前非常流行并广为人知的区块链网络框架的实现方案。作为一种基于模块化架构开发应用程序或解决方案的基础，HyperLedger Fabric 支持如共识和会员服务等即插即用的组件。HyperLedger Fabric 利用容器技术来运行称为“chaincode”的智能合约，该合约包含了系统的应用程序逻辑。

为什么写作本书

区块链由于去中心化、开放性、自治性、信息不可篡改及匿名性等特征而受到广泛关注，且目前正处在上升势态。抛开炒作的代币项目，应用于行业联盟链或直接搭建私链的项目，采用 HyperLedger Fabric 作为底层平台无疑是最好的选择之一。

本人从接触 HyperLedger Fabric 项目以来，经历了其 0.6 版本到 1.1 版本的数次迭代。因为早期中文资料稀缺，并且 0.6 版本到 1.0 版本是一个跨度非常大的迭代，导致早期的大部分部署和应用经验失去作用，不得已再次从 1.0 版本开始从头学习。当时中文资料极为有限，且大多数都是单篇翻译或纯粹的概念讲解，导致我一直没有找到入门的头绪，只能不断地从官方文档中汲取知识，并成功搭建了基于 Kafka 类型的集群网络。

有了集群的经验，加深了自己对 HyperLedger Fabric 整个网络事务流程的理解，并以此为基础顺利搭建了基于 Fabric-SDK-Java 的客户端项目。也就在这个时候，开始有了写相关博客的想法，并在博客园上发布了第一篇博客，也是从零开始系列文章的第一篇，开始介绍自己的开发历程和部署经验，希望能通过这样的方式帮助更多的开发伙伴加入 HyperLedger Fabric 大家庭。再后来又建立了“区块链学习分享”的微信订阅号，也通过微信号建立了一个纯技术讨论分享的 HyperLedger Fabric 等区块链相关交流群，并在这样的机缘下结识了电子工业出版社宋亚东老师，并正式开始编写本书。

在写书之前我一直在梳理博客的内容，为了真实地还原生产场景，还自费租赁了 15 台服务器用于测试。在已有博文的基础上，外加后来编写的新文档，我比较顺利地完成了本书的编写工作，且书中的项目都依托于所租赁服务器来完成演练，每一步都基于真实情况和场景中的操作。在此过程中，自己对 HyperLedger Fabric 也有了新的认识和理解。

本书主要以 HyperLedger Fabric 案例为引，层层深入，从单机单节点到 Solo 多机组网再到 Kafka 集群部署，其中穿插文档讲解加深读者的理解。与一些偏概念性质的区块链教程类书籍不同，我希望能通过本书帮助读者实现基于 HyperLedger Fabric 的区块链实践落地。

本书主要内容

本书以干货为主，文档为辅，基于 HyperLedger Fabric v1.1 版本进行讲解。总计 10 章，每章主要内容介绍如下：

第 1 章是基本环境部署，包括内网和外网的不同方案，以及内核处理等。

第 2 章是 HyperLedger Fabric 及环境部署，先是用文档讲述了 Fabric 相关的介绍和主要功能点，接着分析了源码部署和镜像处理方面的问题。

第 3 章带着读者一步步跑通官方的 e2e_cli 案例，并在随后对该案例进行了分析。

第 4 章开始，手动部署一次单机多节点网络。

第 5 章跟随前章的脚步，手动部署一次 Solo 多机网络环境。

第 6 章继续深入，搭建基于 Kafka 的集群网络。

第 7 章以文档为主，着重讲解如何编写智能合约及有关智能合约的用法。

第 8 章详细介绍 CouchDB 的使用，并推荐使用 CouchDB。

第 9 章讲解在 Fabric 发布 1.0 版本之后的对外客户端调用方式、客户端对 SDK 的使用和相关源码。

第 10 章以一个简单的案例做演练，在数据链上用到了智能合约，对数据提取则提供了另一种思路。

读者对象

这是一本基础讲解类的书，写本书是为了帮助更多的新人入门。所以，本书适合正在寻求 HyperLedger Fabric 入门的新人阅读，也适合中、高级开发人员用作工具书参考。

在阅读本书之前，读者需要具备以下基础知识：

- 具有一定的 Linux 操作系统基本命令的常识。
- 有 Java/Go 等面向对象语言的基础，其中智能合约用 Go 语言编写，SDK 则用到了 Java。如果有这方面的基础，则有助于阅读本书。

致谢

在入门及编写本书的时候，有许多人给予了我鼓励和支持。

首先感谢我的妻子，我在开始学习 HyperLedger Fabric 时遇到了很多困难，经常熬夜加班，她始终体谅我，鼓励并支持我。在我写书的时候她不遗余力地帮我查阅相关资料以便我能

够更顺利地完成书稿。

我还要感谢我的领导，也是我的好友王海林，正是他给了我研究学习 HyperLedger Fabric 的机会，并在我遇到困难时帮我逐条分析，厘清思路。他也给予了我在公司最大限度的研发条件和时间，让我在最短的时间里完成了一次自我蜕变。

我还要感谢电子工业出版社的宋亚东老师，感谢他一直以来对我的支持和鼓励，感谢在我早期编写书稿的过程中对内容的编排和规范给予了很多帮助。也感谢电子工业出版社所有参与本书出版工作的老师，是你们的辛勤付出让本书成功出版。

最后，我要感谢我博客的读者及微信群里的朋友们，正是与你们一次次地沟通和探讨，让我不断提升自我，也鞭策我不断前行。

由于我水平有限，书中不足及错误之处在所难免，敬请专家和读者给予批评指正。

杨毅

2018 年 5 月

读者服务

轻松注册成为博文视点社区（www.broadview.com.cn）用户，扫码直达本书页面。

- **下载资源：**本书提供示例代码及资源文件，均可在“下载资源”处下载。
- **提交勘误：**您对书中内容的修改意见可在“提交勘误”处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **交流互动：**在页面下方“读者评论”处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：http://www.broadview.com.cn/34173

目　　录

第1章　基本环境部署

HyperLedger Fabric 是一个基于模块化架构的分布式账本解决方案平台，它拥有深度加密、便捷扩展、部署灵活及可插拔等特性。它的设计初衷是支持不同组件的可插拔实现，并适应整个经济生态系统中存在的复杂性和高精度性。

与其他区块链平台解决方案相比，HyperLedger Fabric 提供了一种独特的扩展便捷和灵活部署的架构。它更多地适用于联盟链形式，即适合企业级之间的区块链联盟方向，建立在可信任的基础上。如果是企业级区块链部署，建议采用 HyperLedger Fabric 提供的方案。

首次接触 HyperLedger Fabric 开发的用户可以从下文的具体环境部署开始，以了解自己今后所要用到的技术和待加强学习的部分。

在本书中用到的宿主机环境是 Centos，版本为 Centos.x86_64 7.4，内核版本为 4.15，相关配置如下。

```
Distributor ID:    CentOS
Description:  CentOS Linux release 7.4.1708 (Core)
Release:      7.4.1708
Codename:     Core

Linux VM_139_63_centos 4.15.11-1.el7.elrepo.x86_64
```

上述选择并非唯一，如常用的 RedHat 及 Ubuntu 的发行版，包括 MacOS 都可以作为 Fabric 的运行环境支持。

1.1　环境整理

Fabric 的节点通过 Docker 容器来运行，启动 Fabric 网络中的节点需要预先安装 Docker、Docker-Compose 和 Go 语言环境，然后在网上拉取相关的 Docker 镜像，再通过配置 Compose 文件来启动各个节点。

如果想让 Docker 在服务器上运行，内核版本不能低于 3.10。如果内核版本不够，则部分功能（如 overlay2 存储层驱动）无法使用，并且部分功能可能不太稳定。

为了成功安装，请首先将 Linux 内核升级到 4.x，并更新本地依赖，以满足 Docker 运行。

当更新本地依赖时，一般执行如下命令即可：

```
sudo yum update
```

如果是在公司服务器内网环境下进行部署，就需要申请 YUM 源 IP 及端口号，具体如下：

```
115.28.122.210:80
112.124.140.210:80
```

该源地址的实际访问域名为 http://mirrors.aliyun.com，此域名 IP 及端口号相对稳定，但也会出现变更的情况。当无法访问阿里 YUM 源所申请对应的 IP/Port 时，请尝试该域名访问，查看其最新 IP/Port 并更新阿里 YUM 的访问权限。

随后可以更新本地 YUM 源，具体操作步骤如下。

步骤 1：备份原来的 YUM 源。

```
sudo cp /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-Base.repo.bak
```

步骤 2：设置阿里 YUM 的源。

```
sudo wget -O /etc/yum.repos.d/CentOS-Base.repo
http://mirrors.aliyun.com/repo/Centos-7.repo
```

步骤 3：清理缓存并生成新的缓存。

```
sudo yum clean all
sudo yum makecache
```

步骤 4：更新 YUM 库。

```
sudo yum update
```

本操作是为了更新所有的内置库到最新版，因为 Docker 最新版本的安装需要所对应的依赖都是相对较新的版本。为了避免安装依赖的麻烦，故此操作很重要。

注意：很多人会犯这个错误，以为可以不断地通过手动方式来将各种依赖导入进来，但根本行不通。

在 Linux 环境下如果缺少依赖，可以在 Linux Packages Search 下载并安装，地址为 https://pkgs.org/。

1.2 Docker安装

Docker 是一个开源的应用容器引擎，也是一个提供混合云上的每个应用程序的容器平台解决方案。如今的企业面临着数字化转型的压力，但它们受到现有应用和基础设施的限制，同

时对日益多样化的云、数据中心和应用程序架构进行了合理化调整。

Docker 在应用程序和基础设施和开发人员之间实现了真正的独立性，并为挖掘其潜力创造了一个更好的协作和创新的模式。让开发者可以将他们的应用及依赖包打包到一个可移植的容器中，然后发布到任何流行的 Linux 机器上，也可以实现虚拟化。容器完全使用沙箱机制，相互之间不会有任何接口。

一个完整的 Docker 由以下几个部分组成：

- Docker Client 客户端。
- Docker Daemon 守护进程。
- Docker Image 镜像。
- Docker Container 容器。

1.2.1　卸载旧版本

首先需要对服务器进行清理，如果之前安装过 Docker，需要先执行卸载操作，具体命令如下：

```
sudo yum remove docker \
                docker-client \
                docker-client-latest \
                docker-common \
                docker-latest \
                docker-latest-logrotate \
                docker-logrotate \
                docker-selinux \
                docker-engine-selinux \
                docker-engine
```

注意：如果 YUM 提示这些包都没有安装也没有问题。

接下来安装 Docker，而 Docker 的安装需要执行两大步骤，第一步是设置仓库，第二步是安装 Docker CE。

1.2.2　在线安装Docker CE

接下来安装所需要的包。yum-utils 提供的 yum-config-manager、device-mapper-persistent-data 和 lvm2 是设备/存储驱动程序所需要的基础应用。具体执行命令如下：

```
sudo yum install -y yum-utils \
device-mapper-persistent-data \
```

```
lvm2
```

使用下面的命令来设置稳定存储库：

```
sudo yum-config-manager \
    --add-repo \
https://download.docker.com/linux/centos/docker-ce.repo
```

可以选择性地启用 edge 和测试存储库，这些存储库包含在 Docker 中，在默认情况下是禁用的。具体执行命令如下：

```
sudo yum-config-manager --enable docker-ce-edge
sudo yum-config-manager --enable docker-ce-test
```

也可以通过使用禁用标志来运行 yum-config-manager 命令，以禁用 edge 或测试存储库。要重新启用它，可以使用-enable 标志。下面的命令禁用 edge 存储库：

```
sudo yum-config-manager --disable docker-ce-edge
```

最后，可执行如下命令安装最新版本的 Docker CE：

```
sudo yum install docker-ce
```

还能通过如下命令安装指定版本的 Docker CE：

```
sudo yum install docker-ce-<VERSION STRING>
```

执行查询 Docker 版本号，看是否安装成功：

```
docker --version
```

正常情况下会出现如下情况：

```
Client:
 Version: 18.03.0-ce-rc4
 API version: 1.37
 Go version:  go1.9.4
 Git commit:  fbedb97
 Built:   Thu Mar 15 07:40:24 2018
 OS/Arch: linux/amd64
 Experimental:false
 Orchestrator:swarm

Server:
 Engine:
  Version:     18.03.0-ce-rc4
  API version:1.37 (minimum version 1.12)
  Go version: go1.9.4
```

```
Git commit: fbedb97
Built:  Thu Mar 15 07:44:03 2018
OS/Arch:    linux/amd64
Experimental:   false
```

1.2.3　离线安装Docker CE

如果是在公司内网环境下进行部署，则无法通过在线方式进行 Docker CE 的安装，需要从官方下载对应的离线包。

注意：本书在编写时 Docker 官方的最新版为 18.03.0.ce-1.el7.centos.x86_64，读者可以在官方下载页面下载最新版，地址如下：

https://download.docker.com/linux/centos/7/x86_64/stable/Packages/

选择下载指定的版本，可将最新版下载至/tmp/docker/docker 目录下。随后执行如下命令进行安装：

```
cd /tmp/docker/docker
yum install docker-ce-18.03.0.ce-1.el7.centos.x86_64.rpm
y
```

安装完成后，按第 1.2.1 节所述执行查询 Docker 版本号，可以检查是否安装成功。

更多安装方面的细节可以参阅 Docker 官方文档进行操作，地址为 https://www.docker.com/。

1.2.4　Docker启动及常用命令

Docker CE 安装完成后，需要启动它并设置为开机启动。

Docker 启动命令如下：

```
service docker start
```

Docker 开机自启动命令如下：

```
chkconfig docker on
```

Docker 常用命令如下。

杀死所有正在运行的容器：

```
docker kill $(docker ps -a -q)
```

删除所有已经停止的容器：

```
docker rm $(docker ps -a -q)
```

删除所有镜像：

```
docker rmi $(docker images -q)
```

强制删除所有镜像：

```
docker rmi -f $(docker images -q)
```

1.3 Docker-Compose安装

Compose 是定义和运行多容器 Docker 应用程序的工具，可以使用 YAML 文件来配置应用服务。然后，通过单个命令可以从配置中创建并启动所有服务。

```
要了解更多关于组合的特性，请参阅特性列表：
https://docs.docker.com/compose/overview/#features
```

使用 Compose 基本上是一个三步骤的过程：

（1）用 Dockerfile 定义应用程序的环境，这样它可以在任何地方复制。

（2）通过 docker-compose.yml 在服务中定义所启动的各个应用，这些应用将在相互隔离的环境中同时运行。

（3）运行 docker-compose up，启动 Compose 并运行整个应用程序。

1.3.1 在线安装Docker-Compose

安装 Docker-Compose 需要服务器支持 curl 命令，如果服务器不支持 curl，需要执行如下操作安装 curl 依赖：

```
yum install curl
```

通过 https://github.com/docker/compose/releases 网址，可以查看 Docker Compose 最新发行版的动向。

注意：本书所使用的 Docker Compose 的版本为 1.20.1。

执行如下操作下载 Docker Compose：

```
sudo curl -L https://github.com/docker/compose/releases/download/1.20.1/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
```

该下载目录为/usr/local/bin/docker-compose，且权限已经给出，再执行 docker-compose --