

# 部署超级账本 Fabric v1.0 多节点集群（1）

张海宁、陈家豪

VMware 中国研发中心

## 一、概述

在千呼万唤之后，犹抱琵琶的超级账本 1.0 GA 版即将揭开面纱，翘首以待的社区用户将广泛使用这个版本。本文将介绍如何使用 Docker 容器技术来建立起一个多节点 Fabric 集群，并且描述在集群上如何进行基本的操作，如 chaincode 的生命周期维护等。文中采用 Fabric 1.0 beta 的端到端（e2e）范例作为基础来说明原理。本文提供是为手动配置的方法，今后将介绍利用容器平台（如 K8s 等）自动部署超级账本的方式。

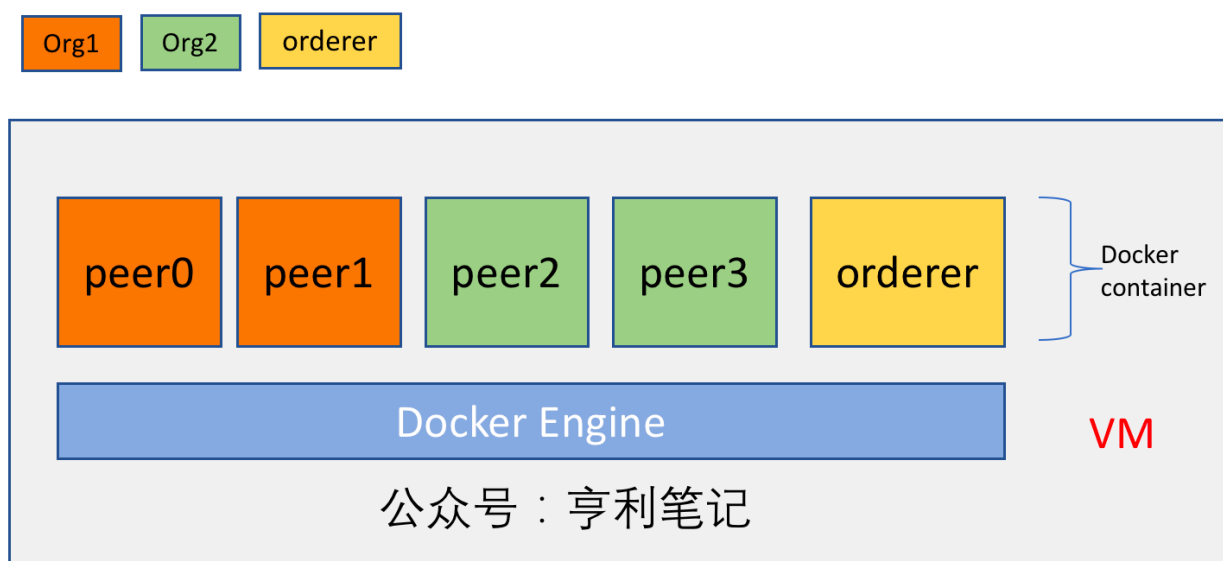
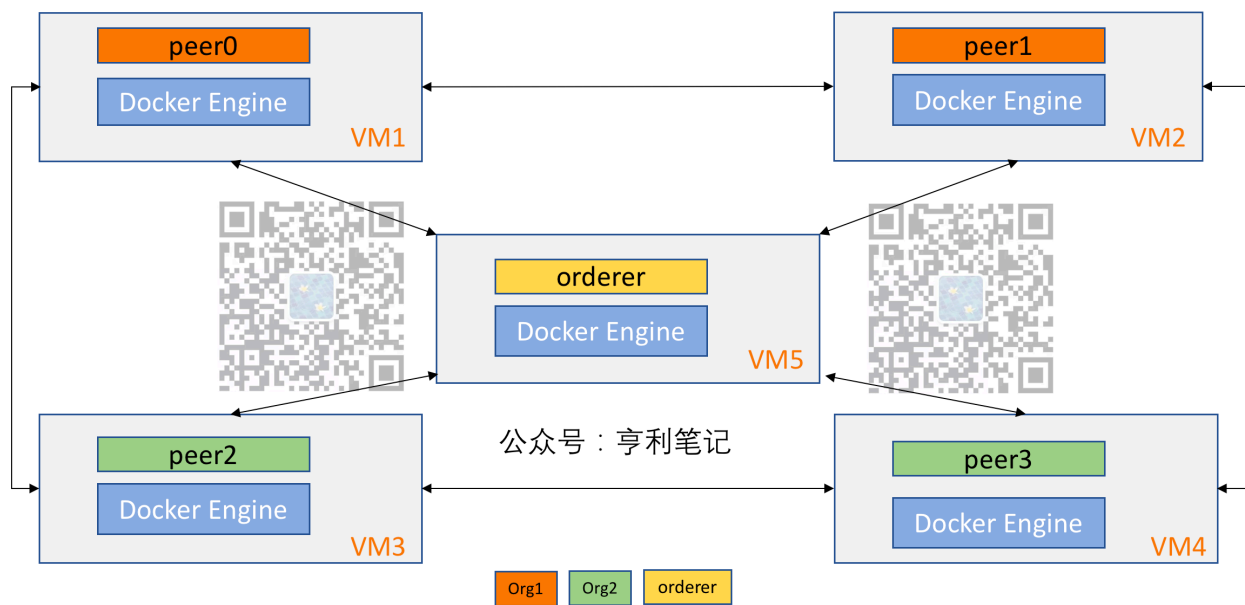


图 1.1 单节点下的 Fabric 网络结构图

Fabric 源码中包含一个简单的 e2e 单机部署示例，方便用户理解、研究和开发应用。如图 1.1 所示，在单个机器节点上通过 docker-compose 建立了 5 个节点的 Fabric 网络，每个节点都是由单独的 Docker 容器来模拟。其中 peer0 和 peer1 是同属于 org1 的节点，peer2 和 peer3 是同属于 org2 的节点，它们都加入了相同的 channel 中，并在该 channel 中进行交易，而 orderer 则为该 channel 中的交易提供排序服务。



e2e 的示例虽然简单，但它把多个节点混合部署在一起，无法区分哪些配置对应哪个节点。另外，在实际场景中，Fabric 节点可能会由不同的组织分别拥有和维护，peers 和 orderer 必然会分布在不同的物理节点上，因此多节点的 Fabric 部署成为需要解决的问题，图 1.2 是多节点 Fabric 集群拓扑图。

下面是把单节点 e2e 范例改为多节点的大致步骤：

## 1. 准备环境

运行 Fabric 节点需要依赖以下工具：

- Docker：用于管理 Fabric 镜像以及运行 peer 和 orderer 等组件
- Docker-compose：用于配置 Fabric 容器
- Fabric 源码：源码提供了用于生成证书和配置 channel 的工具和测试代码
- Go 语言开发环境：源码的工具编译依赖于 Go 语言

## 2. 配置多节点 Fabric 集群

在单节点 e2e 示例中，所有节点部署在同一个 docker-compose 的内部网络中，通过容器的 7051 端口进行通信。但是在多节点的情况下，容器之间不能进行直接通讯，因此需要把容器的 7051 端口映射到宿主机上，通过各个宿主机的 7051 端口来实现节点间通信。我们在每个节点中修改 docker-compose.yaml 中的 service 定义，在不同节点只启动需要的 service。例如，在节点 1 中只启动 peer0 的 service，在节点 5 中仅启动 orderer 等。

### 3. 启动多节点 Fabric 集群

在各个节点上配置好 Fabric 的启动环境后，需要依次登录到节点上通过 `docker-compose up` 的方式启动 Fabric 节点。由于启动环境有依赖关系，如 `peer1` 以 `peer0` 作为发现节点，因此需要先启动 `peer0` 再启动 `peer1`。

### 4. 配置 channel

在 Fabric 中，channel 代表了一个私有的广播通道，保证了消息的隔离性和保密性，它由 `orderer` 来管理。channel 中的成员共享该 channel 的账本，并且只有通过验证的用户才能在 channel 中进行交易，与一个 channel 相关的属性记录在该 channel 的初始区块中，可通过 `reconfiguration` 交易进行更改。channel 的初始区块由 `create channel` 交易生成，`peer` 向 `orderer` 发送该交易时会带有的 `config.tx` 文件，该文件定义 channel 的相关属性。

### 5. 发布 chaincode

chaincode 是开发人员按照特定接口编写的智能合约，通过 SDK 或者 CLI 在 Fabric 的网络上安装并且初始化后，该应用就能访问网络中的共享账本。

chaincode 的生命周期如下：

#### a. install（安装）

chaincode 要在 Fabric 网络上运行，必须要先安装在网络中的 `peer` 上，安装同时注明版本号保证应用的版本控制。

#### b. instantiate（实例化）

在 `peer` 上安装 chaincode 后，还需要实例化才能真正激活该 chaincode。在实例化的过程中，chaincode 就会被编译并打包成容器镜像，然后启动运行。若 chaincode 在实例化的过程中更新了数据状态，如给某个变量赋予初始值，则该状态变化会被记录在共享账本中。每个应用只能被实例化一次，实例化可在任意一个已安装该 chaincode 的 `peer` 上进行。

#### c. invoke 和 query（调用和查询）

chaincode 在实例化后，用户就能与它进行交互，其中 `query` 查询与应用相关的状态（即只读），而 `invoke` 则可能会改变其状态。

#### d. upgrade (升级)

在 chaincode 添加新功能或出现 bug 需要升级时，可以通过 upgrade 交易来实现。这时需要把新的代码通过 install 交易安装到正在运行该 chaincode 的 peer 上，安装时需注明比先前版本更高的版本号，接下来只需要向任意一个安装了新代码的 peer 发送 upgrade 交易就能更新 chaincode，chaincode 在更新前的状态也会得到保留。

(未完待续)

扫码关注公众号：亨利笔记，获取更多区块链和云计算等方面科技文章。



<https://github.com/hainingzhang/articles>

#### VMware 公司招聘区块链实习生和外包开发工程师

VMware 公司为超级账本 Hyperledger 项目创始成员，中国研发中心现在招募区块链方向实习生和外包开发工程师，地点：北京知春里。

**实习生：**要求在读研究生，计算机相关专业，懂 Java 或 Go 开发语言，能够实习 3 个月以上，熟悉区块链技术优先。欢迎自荐或推荐。

**外包软件开发工程师：**3 年以上软件开发经验，熟悉 Java 或 Go 开发语言，熟悉分布式系统、Docker，了解区块链技术优先。

有兴趣者发简历到：[harbor@vmware.com](mailto:harbor@vmware.com)