部署一个生产网络

本部署指南是对设置生产 Fabric 网络组件的适当的顺序的整体概述，此外还有最佳做法和部署时要记住的一些注意事项。

部署 Fabric 网络的过程很复杂，需要了解公钥基础设施和管理分布式系统。如果你是智能合约或应用开发者，在部署生产级别 Fabric 网络时，你不应该需要这种级别的专业技能。然而，你可能需要了解网络是如何部署的，以便开发有效的智能合约和应用程序。

如果你只需要一个开发环境来测试链码、智能合约和应用程序，请查看 使用Fabric的测试网络。它包括两个组织，每个组织拥有一个 peer 节点，以及一个拥有单个排序节点的排序服务组织。**该测试网络并不打算为部署生产组件提供蓝图，也不应该这样使用，因为它会做出生产部署不会做出的假设和决策。**

本指南会向你概述设置生产组件和生产网络的步骤：

步骤一：选定你的网络配置

区块链网络结构必须按照用例来决定。这些基本的业务决策将根据你的用例的改变而改变，但是让我们考虑一些场景。

与开发环境或概念证明相反，在生产环境中操作时，安全性、资源管理和高可用性成为优先考虑的事项。你需要多少个节点来满足高可用性，以及你希望在哪些数据中心部署它们来确保灾难恢复和数据驻留的需求得到满足？你将如何确保你的私钥和信任根保持安全状态？

除了以上提到的，这有一个在部署组件之前你需要做出决定的案例：

- **证书颁发机构配置。** 作为整体决策的一部分，你必须决定你的 peer 节点（有多少，每个通道有多少等等）和你的排序服务（有多少节点，谁将拥有它们），你还必须决定你的组织的 CA（译者注：证书颁发机构，下同）节点如何被部署。生产网络应该使用传输层安全性（TLS），这将需要设置一个 TLS CA，并使用它来生成 TLS 证书。此 TLS CA 需要在你登录 CA 之前部署。我们将更多在 步骤三：设置你的 CA（译注：证书颁发机构，下同）节点 讨论这点。
- **使用或者不使用组织的单位？** 一些组织可能会发现有必要建立组织单位，以便在特定身份和由单一认证机构创建的 MSP（成员服务提供商）之间建立分离。
- **数据库类型。** 网络中的一些通道可能需要所有数据以某种 使用 CouchDB 作为状态数据库 能理解的方式进行建模，然而其他优先考虑速度的网络可能会决定所有 peer 节点将使用 LevelDB。注意通道中不应该有同时使用 CouchDB 和 LevelDB 的 peer 节点，因为两个数据库类型模型数据稍有不同。

- **通道和私有数据。** 一些网络可能认为 通道 是确保某些特定交易的隐私性和隔离性的最佳方式。其他网络可能会认为一个单一通道连同 私有数据 能更好地服务于他们的隐私需求。
- **容器编排。** 不同用户也可能做出针对他们的容器编排的不同决定，为他们的 peer 进程创建单独的容器，为 peer 进程、CouchDB、gRPC 通信以及链码记录日志；而其他用户可能会决定组合这些进程中的一些。
- **链码部署方式。** 用户现在可以选择使用内置构建结合运行支持（自定义构建和使用 外部构建器和启动器 来运行），或者使用 将链码作为外部服务。
- **使用防火墙。** 在产品部署中，属于某个组织的组件可能需要访问其他组织的组件，使用防火墙和高级网络配置成为必须。例如，使用 Fabric SDK 的应用程序需要访问所有组织的所有背书 peer 节点以及所有通道的排序服务。类似地，peer 节点需要在他们接收区块的通道上访问排序服务。

无论你的组件何时以何种方式部署的，为了有效地运行你的网络，你将需要在你选择的管理系统（如 Kubernetes）中拥有高度的专业知识。类似地，网络结构必须被设计成适用于业务用例以及网络所运行行业的任何相关法律法规。

本部署指南不会覆盖每一次迭代和潜在的网络配置，但给出了需要考虑的通用指南和规则。

步骤二：为你的资源设置一个集群

一般来说，Fabric 与部署和管理它的方法无关。例如，可能在一台笔记本电脑上部署和管理一个 peer 节点。因为种种原因，这可能不是可取的，但 Fabric 中没有任何东西可以禁止它。

只要你有能力部署容器，无论是在本地（或防火墙后），还是在云端，都应该有可能建立组件并将它们相互连接。然而，Kubernetes 具有许多有用的工具，使其成为部署和管理 Fabric 网络的普及容器管理平台。需了解更多关于 Kubernetes 的信息，请查看 the Kubernetes documentation。本主题将主要将其范围限制在二进制文件中，并提供在使用 Docker 部署或 Kubernetes 时可以应用的说明。

无论你选择以什么方式，在什么地方部署你的组件，你都需要确保有足够的资源让组件有效运行。你需要的大小很大程度上取决于你的用例。如果你计划将单个 peer 节点加入几个高容量通道，它将比一个用户计划加入单个通道的 peer 节点需要更多的 CPU 和内存。粗略计算，计划分配给 peer 节点的资源大约是你计划分配给单个排序节点资源（如下所述，推荐部署至少三个，最好是五个节点到排序服务）的三倍。类似地，对于一个 CA，你应该需要相当于 peer 节点十分之一的资源。你也需要添加存储到你的集群（一些云提供商可能会提供存储），因为如果没有先与云提供商一起设置存储，你就无法配置持久卷和持久卷声明。

通过部署概念证明网络并在负载下测试它，你将更好地了解你需要的资源。

管理你的基础设施

你用来管理你的后端的确切方法和工具将取决于你选择的后端。然而，这里有一些值得注意的事项。

- 使用机密对象在集群中安全地存储重要的配置文件。有关 Kubernetes 机密的信息，请查看 Kubernetes secrets。你也可以选择使用强化安全模块加密持久卷 （PVs）。类似的路线，在部署完 Fabric 组件后，你可能想在你自己的后端链接一个容器，例如在 Docker Hub 这样的服务中使用私有仓库。你需要以 Kubernetes 密码的形式对登录信息进行编码，并在部署组件时将其包含在 YAML 文件中。

- 集群注意事项和节点大小。在上面的第2步中，我们讨论了如何考虑节点大小的一般概要。你的用例，以及一个健壮的开发阶段，是你真正了解你的 peer 节点、排序节点和 CA 节点需要多大的唯一方法。
- 你选择如何挂载你的卷。最佳做法是将与节点相关的卷挂载在部署节点的外部。这将允许你稍后引用这些卷（例如，重新启动已崩溃的节点或容器），而不必重新部署或重新生成你的密码材料。
- 你如何监控你的资源。通常关键是要建立一个策略和方法来监视我们个人节点使用的资源和部署到集群的资源。随着你加入你的 peer 节点到更多的通道，你可能需要增加它的 CPU 和内存分配。同样，你需要确保你的状态数据库和区块链有足够的存储空间。

步骤三：设置你的 CA（译注：证书颁发机构，下同）节点

Fabric 网络中必须部署的第一个组件是 CA。这是因为在节点本身能被部署之前，节点相关证书（不仅是节点本身的证书，还有识别谁可以管理节点的证书）必须被创建。虽然不是必需使用 Fabric CA 来创建这些证书，但 Fabric CA 还创建了组件和组织要正确定义所需的 MSP 结构。如果用户选择使用 Fabric CA 以外的 CA，则必须自己创建 MSP 文件夹。

- 一个 CA（或者更多，如果你正在使用中间 CA — 以下有关于中间 CA 的更多信息）用于生成（通过一个称为“登录”的过程）组织管理员的证书、该组织的 MSP 以及该组织拥有的任何节点。此 CA 还将为任何其他用户生成证书。由于它在“登录”身份中的作用，这个 CA 有时被称为“登录 CA”或“ecert CA”。
- 其他 CA 生成保护传输层安全（TLS）通信的证书。因此，这个 CA 通常被称为“TLS CA”。这些 TLS 证书被附加到防止“中间人”攻击的活动中。请注意，TLS CA 仅用于为节点颁发证书，当该活动完成时可以关闭。用户可以选择使用单向（仅客户端）TLS 以及双向（服务器和客户端）TLS，后者也称为“相互（mutual）TLS”。因为指定你的网络使用 TLS（推荐使用） 应该在部署“登录” CA（指定此 CA 配置的 YAML 文件有一个启用 TLS 的字段）之前，所以你应该先部署 TLS CA，并在引导登录 CA 时使用其根证书。当链接到登录 CA 为用户和节点登录身份时候，这份 TLS 证书也会被 `fabric-ca client` 使用。

虽然与组织相关的所有非 TLS 证书都可以由单个“根”CA（即其自身信任根的 CA）创建，但对于添加的安全组织来说，可以决定使用证书由根 CA（或最终导向根 CA 的另一个中间 CA）创建的“中间”CA。由于根 CA 的泄露导致其整个信任域（管理员、节点和它为其生成的任何证书的 CA）崩溃，中间 CA 是一种限制根 CA 暴露的有用方法。你是否选择使用中间 CA 将取决于用例的需要。并非强制使用。请注意，对于那些已经采取此实现方式，并且不想在现有基础设施中添加身份管理层的企业，还可以配置轻量级目录访问协议（LDAP）来管理 Fabric 网络上的身份。

在生产网络中，建议每个组织至少部署一个用于注册目的 CA，另一个用于 TLS 的 CA。 例如，如果你部署了三个与某个组织相关联的 peer 节点和一个与排序组织相关联的排序节点，则至少需要四个 CA。两个 CA 是用于 peer 组织（为 peer 节点、管理员、通信以及代表组织的 MSP 目录结构生成登录和 TLS 证书），其他两个 CA 将用于排序组织注意，用户通常只用登录 CA 来注册和登录，然而节点需要用登录 CA（在登录 CA 中，当节点尝试对其操作进行签名时，它将获得标识它的签名证书）和 TLS CA（在那里它将获得用于验证其通信的 TLS 证书）来注册和登录。

对于如何设置组织 CA 和 TLS CA 并登录其管理员身份的示例，请查看 Fabric CA 部署指南。本部署指南使用 Fabric CA 客户端来注册和登录需要设置 CA 的身份。

步骤四：用 CA 来创建身份和 MSP

创建CA后，可以使用它们为与组织相关的身份和组件（由 MSP 表示）创建证书。对于每个组织，你至少需要：

- **注册和登录管理员身份并创建 MSP**。在创建了与组织关联的 CA 之后，可以使用它先注册一个身份，然后登录它。在第一步中，身份的用户名和密码由 CA 的管理员分配。属性和从属关系也可以被赋予身份（例如，`admin` 的 `role`，这是组织管理员所必需的）。身份注册后，可以使用用户名和密码来登录。该CA将为该身份生成两个证书-网络其他成员已知的公共证书（也称为签名证书）和用于身份签名操作的私钥（存储在 `keystore` 文件夹中）。该 CA 也会生成一个包含 CA 颁发证书的公有证书的 MSP 文件以及 CA 的信任根（这可能是也可能不是同一个 CA）。此 MSP 可以被认为是定义与管理员身份相关的组织。对于这个过程细节的例子，请查看 管理员如何登录的例子。如果组织的管理员也是节点的管理员（这将是典型的），**你必须在创建本地节点的 MSP 之前，创建组织管理员身份，因为当创建本地 MSP 时，节点管理员证书必须被使用**。
- **注册和登录节点身份**。就像一个组织管理员身份被注册和登录一样，节点的身份必须用登录的 CA 和 TLS CA 执行注册和登录操作。因此，登录 CA 和 TLS 共享数据库（允许节点身份仅注册一次并由每个 CA 服务器单独登录)可能是有用的，尽管这是一个可选的配置选项。当用登录 CA 注册节点身份时，不给节点 `admin` 或 `user` 的角色，而是给它一个 `peer` 或 `orderer` 的角色。与管理员一样，此身份的属性和从属关系也可以分配。节点的 MSP 结构称为 “本地 MSP”，因为分配给身份的权限仅与本地（节点）级别相关。此 MSP 是在创建节点身份时创建的，并在引导节点时使用。在你用 TLS CA 登录后并将组织加入到通道（当登录管理员身份时，此证书必须添加到创建的 orgMSP 中）时，以及使用 peer 二进制作为 CLI 客户端向其他 peer 节点（如 `peer chaincode invoke`）或排序节点（如 `peer channel fetch`）进行调用时（因为没有 `orderer` 的 CLI），你将使用生成的TLS 根证书。没有必要将TLS根证书添加到节点的本地MSP中，因为这些证书包含在通道配置中。

有关基于Fabric的块链网络中的身份和权限的更多概念信息，请查看 身份 and 成员服务提供者 (MSP).

要了解如何使用 CA 来生成一个管理员身份和 MSP，查看 Enroll Org1's Admin.

要了解如何用登录 CA 和 TLS CA 来为节点生成证书，查看 Setup Org1's Peers.

步骤五：部署节点

一旦你收集完你需要的所有证书和 MSP，你几乎准备好创建一个节点了。如上所述，有很多合法的方式来部署节点。

创建一个 peer 节点

在创建 peer 节点之前，你需要为 peer 节点定制配置文件。在 Fabric 中，这个文件叫做 `core.yaml`。你可以找到一个示例 `core.yaml` 配置文件 在 Hyperledger Fabric sampleconfig 目录.

正如你在文件中看到的，有相当多的参数，你可以选择设置，或者需要设置节点才能正常工作。一般情况下，如果你不需要更改变化值，就不要管它。但是，你可能需要调整各种地址、指定要使用的数据库类型以及指定节点的 MSP 所在位置。

你主要有两个选择更改配置。

1、编辑和二进制文件绑定的 YAML 文件。

2、在部署时，使用环境变量来重写。

3、在 CLI 命令中指定标识。

选项1的优点是，每当你将节点关闭并又恢复启动时，会持久化你的更改。缺点是，升级到新的二进制版本时，必须将你定制的选项移植到新的 YAML 文件（升级到新版本时，应该使用最新的 YAML）。

无论哪种方式，这有一些在 `core.yaml` 中你必须检查的值。

- `peer.localMspID`：这是你的 peer 组织的本地 MSP 的名称。在此MSP中，将列出你的 peer 组织管理员以及 peer 组织的根 CA 和TLS CA 证书。
- `peer.mspConfigPath`：peer 节点的本地 MSP 的所在地。注意，将此卷挂载到容器外部是最佳做法。这确保即使容器被停止（例如，在维护周期中），MSP也不会丢失，并且必须重新创建。
- `peer.address`：代表同一组织中的其他 peer 节点的终端，这是在组织内建立 gossip 通信的一个重要注意事项。
- `peer.tls`：当你将 `enabled` 值设置为 `true`（应该在生产网络中完成）时，你将必须指定相关 TLS 证书的位置。请注意，网络中的所有节点（peer 节点和排序节点）都必须启用或不启用 TLS。对于生产网络，强烈建议启用TLS。与你的MSP一样，将此卷挂载到容器外部是最佳做法。
- `ledger`：用户可以做许多关于其账本的决定，包括状态数据库类型（例如，LevelDB 或 CouchDB）以及其位置（通过 `fileSystemPath` 指定）。请注意，对于 CouchDB 来说，在 peer 节点外部（例如，在一个单独的容器中）操作你的状态数据库是一种最佳实践，因为你将能够以这种方式更好地将特定资源分配给数据库。出于延迟和安全原因，将 Couch DB 容器放在与 peer 节点服务器相同的服务器上是最佳做法。对 CouchDB 容器的访问应该仅限于 peer 节点容器。
- `gossip`：在设置 Gossip 数据传播协议 时，有许多配置选项要考虑，包括 `externalEndpoint`（它使 peer 节点可被其他组织的 peer 节点发现）以及 `bootstrap` 地址（通过它在 peer 自己的组织中识别一个 peer）。
- `chaincode.externalBuilders`：当使用 将链码作为外部服务 时，设置这个字段很重要。

当你对如何配置 peer 节点、如何挂载卷以及后端配置感到得心应手时，可以运行命令启动 peer 节点（此命令将取决于后端配置）。

创建一个排序节点

与创建 peer 节点不同，你将需要创建一个创世纪块（或者引用已经创建的块，如果将排序节点添加到现有的排序服务中），并在启动排序节点之前指定其路径。

在Fabric中，这个用于排序节点的配置文件称为 `orderer.yaml`。你可以在 Hyperledger Fabric 的 sampleconfig 目录中 找到一个示例配置文件 `orderer.yaml`。请注意，`orderer.yaml` 与排序服务的“genesis block”不同。该区块包括排序系统通道的初始配置，必须在创建排序节点之前创建，因为它用于引导节点。

与 peer 节点一样，你将看到有相当多的参数，你要么可以选择设置，要么需要设置节点才能正常工作。一般情况下，如果您不需要更改变化值，就不要管它。

你有两个主要选项来更改你的配置。

1、编辑和二进制文件绑定的 YAML 文件。

2、 在部署时，使用环境变量重写。

3、 在 CLI 命令中指明标签。

选项1的优点是，每当你将节点关闭并又恢复启动时，会持久化你的更改。缺点是，升级到新的二进制版本时，必须将你定制的选项移植到新的 YAML 文件（升级到新版本时，应该使用最新的 YAML）。

无论如何，在 `orderer.yaml` 中有一些值你必须检查。你将发现这些字段中的一些是和 `core.yaml` 中的一样的，只是名称不同。

- `General.LocalMSPID`：通过排序组织的 CA 生成的本地 MSP 的名称。
- `General.LocalMSPDir`：排序节点所在地的本地 MSP。注意，把此卷挂载在容器外面是一种最佳做法。
- `General.ListenAddress` 和 `General.ListenPort`：代表相同组织中的其他排序节点的终端。
- `FileLedger`：虽然排序节点没有状态数据库，但它们仍然都携带区块链的副本，因为这允许它们使用最新的配置块来验证权限。因此，应该用正确的文件路径定制账本字段。
- `Cluster`：这些值对于与其他排序节点通信的排序服务节点非常重要，例如在基于 Raft 的排序服务中。
- `General.BootstrapFile`：这是用于引导排序节点的配置块的名称。如果此节点是在排序服务中生成的第一个节点，则必须生成此文件，并将其称为“创始块”。
- `General.BootstrapMethod`：给出引导块的方法。目前，这只能是 `文件`，其中指明了 `BootstrapFile` 中的文件。从2.0开始，你可以通过指定 `none` 来简单地在不引导的情况下启动排序节点。
- `Consensus`：确定共识插件（支持并推荐 Raft 排序服务）允许的键值对，用于写头日志（`WALDir`）和快照（`SnapDir`）。

当你对如何配置排序节点、如何挂载卷以及后端配置感到得心应手时，可以运行命令启动排序节点（此命令将取决于后端配置）。

下一步

区块链网络都是关于连接的，所以一旦你部署了节点，你显然会想把它们连接到其他节点！如果你有一个 peer 组织和一个 peer 节点，你需要加入你的组织到一个联盟，并加入 通道。如果你有一个排序节点，你需要添加 peer 组织到你的联盟。你还将需要学习如何开发链码，你可以在以下主题中了解到 场景 and 链码开发者教程。

连接节点和创建通道的部分过程将涉及修改策略以适应业务网络的用例。有关策略的更多信息，请查看 策略。

Fabric 中的一个常见任务是编辑现有的通道。有关该过程的教程，请查看 更新通道配置。一个常见的的通道更新操作是向现有通道添加一个组织。有关该特定过程的教程，请查看 向通道添加组织。有关部署后升级节点的信息，请查看 Upgrading your components。

Deploying a Production CA