

.NET Core On Kubernetes Quick Start

ZhenYu, Liu

- 容器技术回顾
- Kubernetes技术揭秘
- .NET Core在K8S下设计与实战

容器技术回顾

容器在生活领域

- 背包
- 集装箱
- 行李箱

容器在技术领域

- Dictionary
- IIS
- Docker

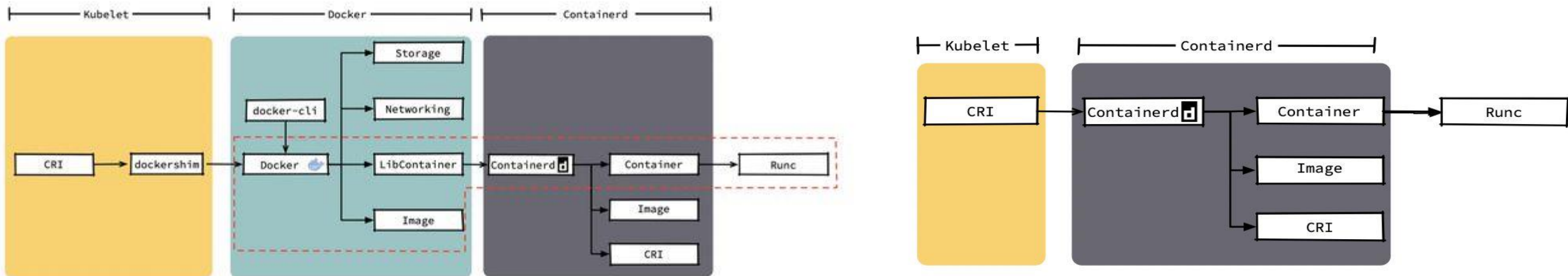
Docker 是什么

Docker 是一个开放源代码软件，是一个开放平台，Docker允许用户将基础设施（Infrastructure）中的应用单独分割出来，形成更小的颗粒（容器），从而提高交付软件的速度。

Docker容器与虚拟机类似，但二者在原理上不同。容器是将操作系统层虚拟化，虚拟机则是虚拟化硬件，docker用Linux核心中的资源分离机制，例如cgroups，以及Linux核心名字空间（namespaces），来创建独立的容器（containers）。这可以在单一Linux实体下运作，避免引导一个虚拟机造成的额外负担。

Docker 容器更多的用于表示 软件的一个标准化单元。由于容器的标准化，因此它可以无视基础设施（Infrastructure）的差异，部署到任何一个地方。

Docker与Containerd



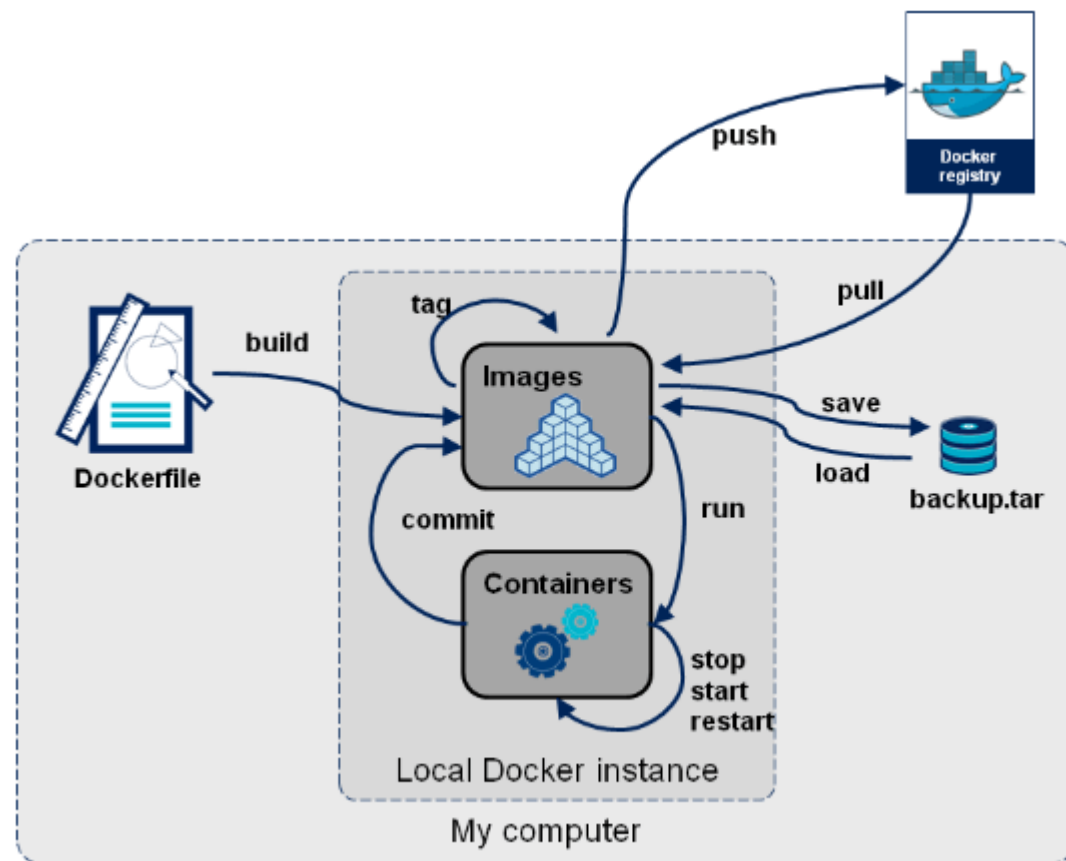
Docker 作为 k8s 容器运行时，调用关系如下：

kubelet --> docker shim （在 kubelet 进程中） --> dockerd --> containerd

Containerd 作为 k8s 容器运行时，调用关系如下：

kubelet --> cri plugin --> containerd

Docker架构



*Kubernetes*技术揭秘

Kubernetes是什么

Kubernetes（常简称为**K8s**）是用于自动部署、扩展和管理“容器化（containerized）应用程序”的开源系统。该系统由Google设计并捐赠给Cloud Native Computing Foundation（今属Linux基金会）来使用。

它旨在提供“跨主机集群的自动部署、扩展以及运行应用程序容器的平台”。它支持一系列容器工具, 包括Docker, Containerd, cri-o。

Kubernetes集群搭建的三种方式

Kubeadm

- 是一个工具，用于快速搭建kubernetes集群，目前应该还是比较方便和推荐的，简单易用
- kubeadm init 以及 kubeadm join 这两个命令可以快速创建 kubernetes 集群

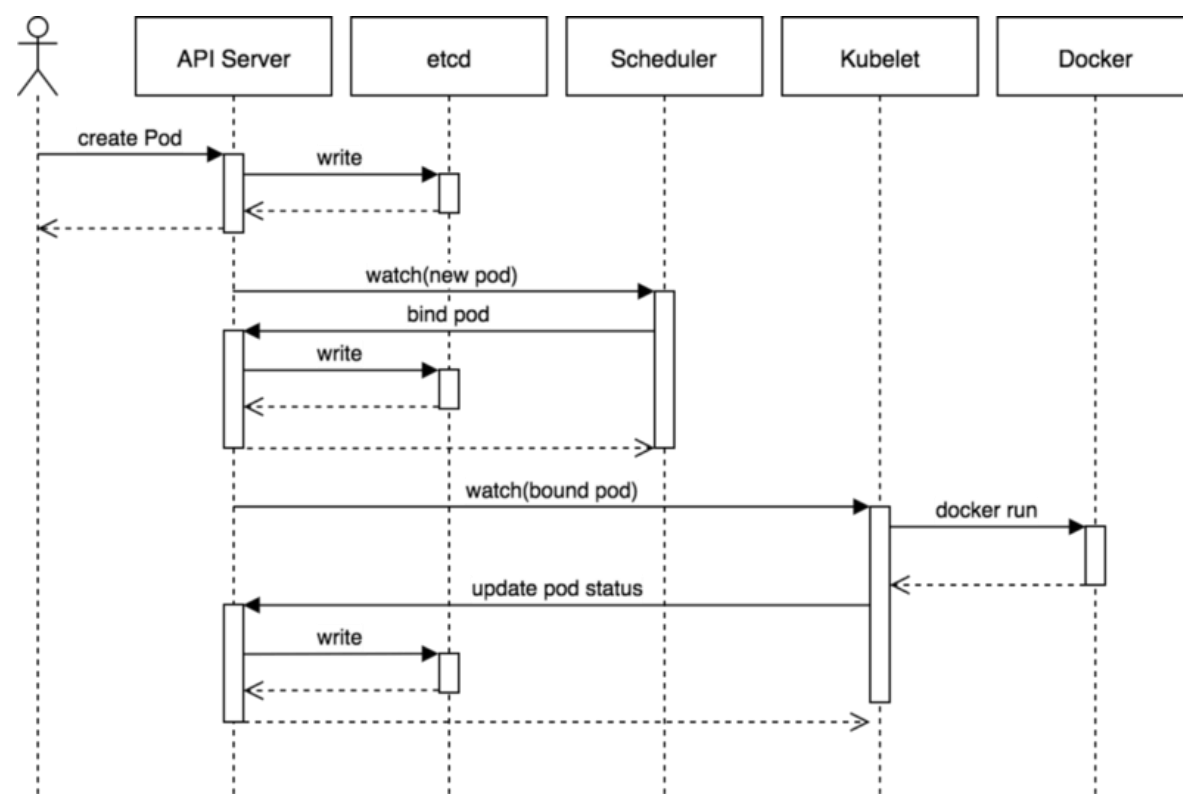
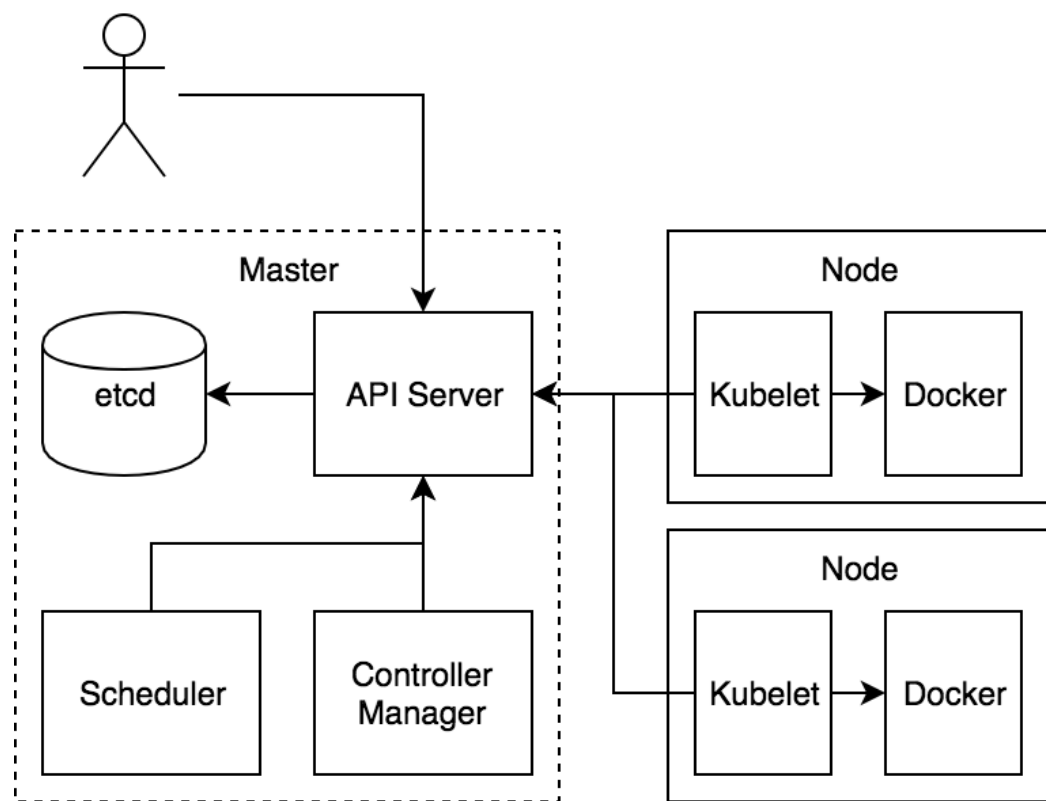
Minikube

- 一般用于本地开发、测试和学习，不能用于生产环境
- 是一个工具，minikube快速搭建一个运行在本地的单节点的Kubernetes

二进制包

- 上面的两个是工具相当于用程序脚本帮我们装好了集群，属于自动部署，简化部署操作，自动部署屏蔽了很多细节，使得对各个模块感知很少，遇到问题很难排查，如果手动安装，对kubernetes理解也会更全面。
- 目前生产环境的主流搭建方式，已在生产环境验证，kubeadm也可以搭建生产环境

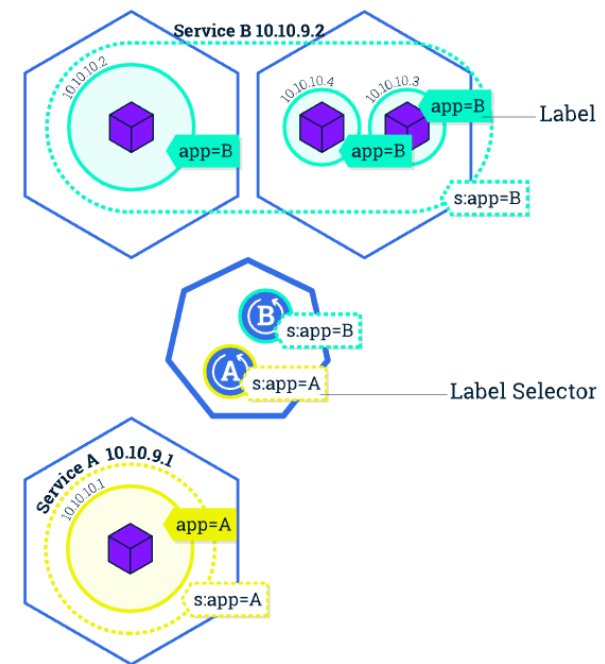
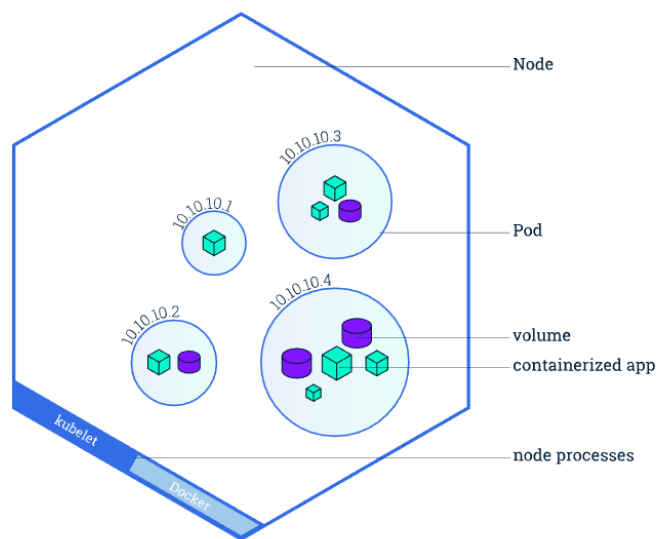
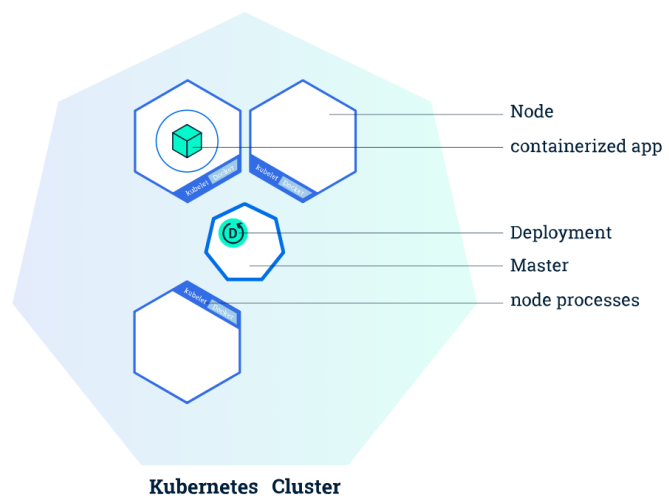
Kubernetes系统架构



Kubernetes核心概念

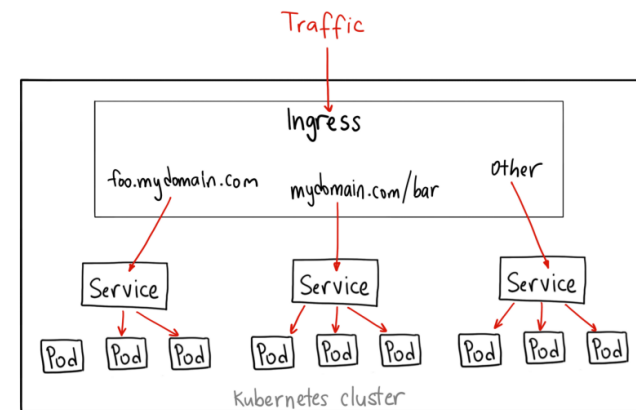
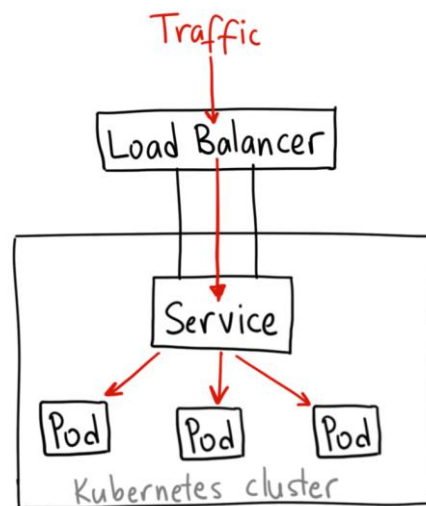
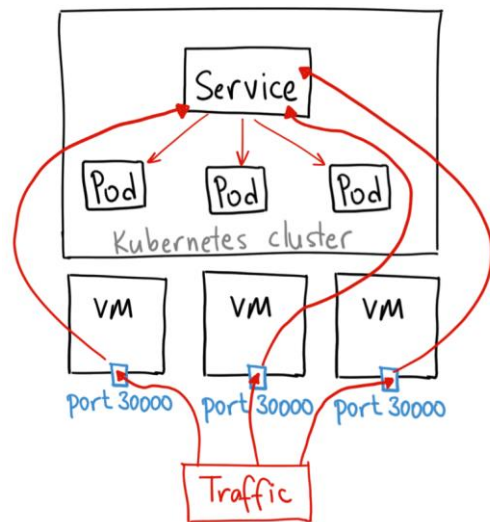
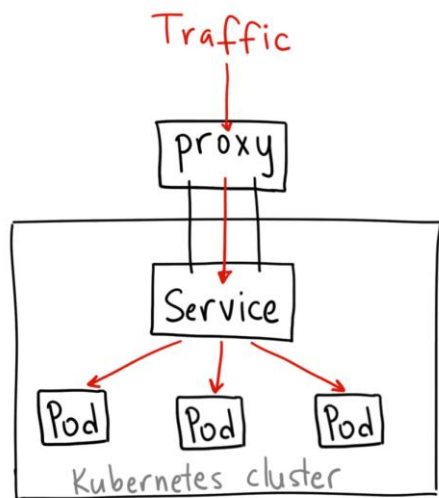
- Node (集群节点)
- Pod (封装了一个或多个应用程序的容器)
- Deployment (部署)
- Services (用来与pod通信)
- Persistent Volumes (数据持久化)
- Namespace (资源隔离)
- Label and selector (资源匹配)

Kubernetes核心概念

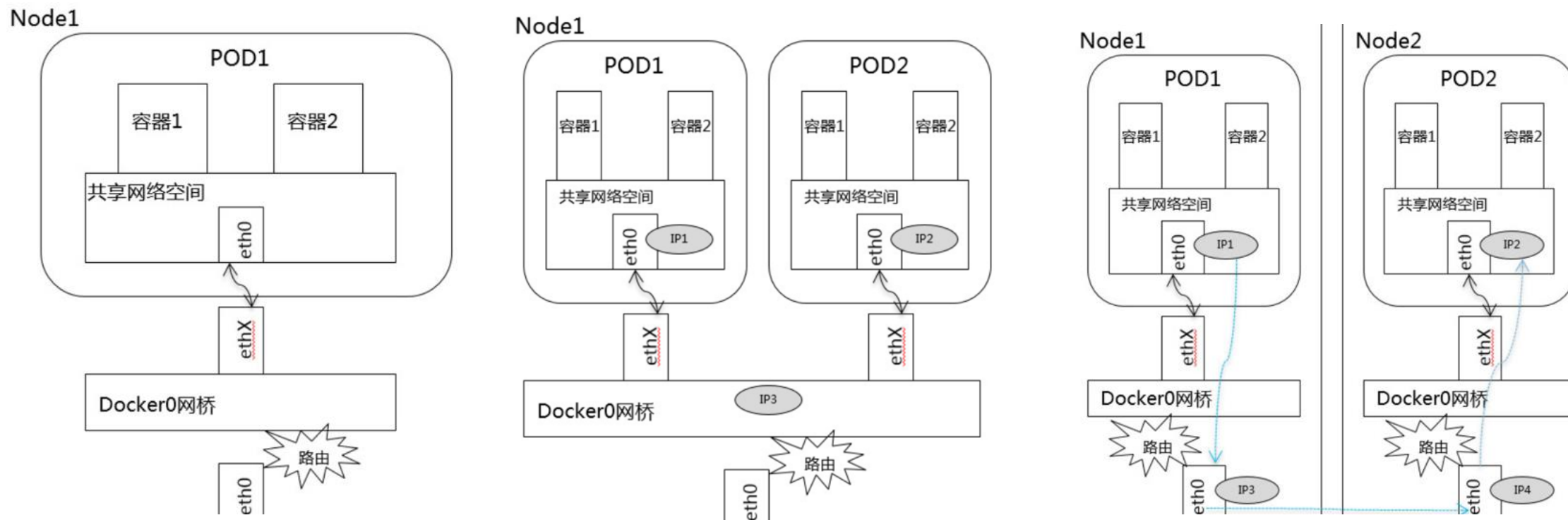


Service的三种网络方案

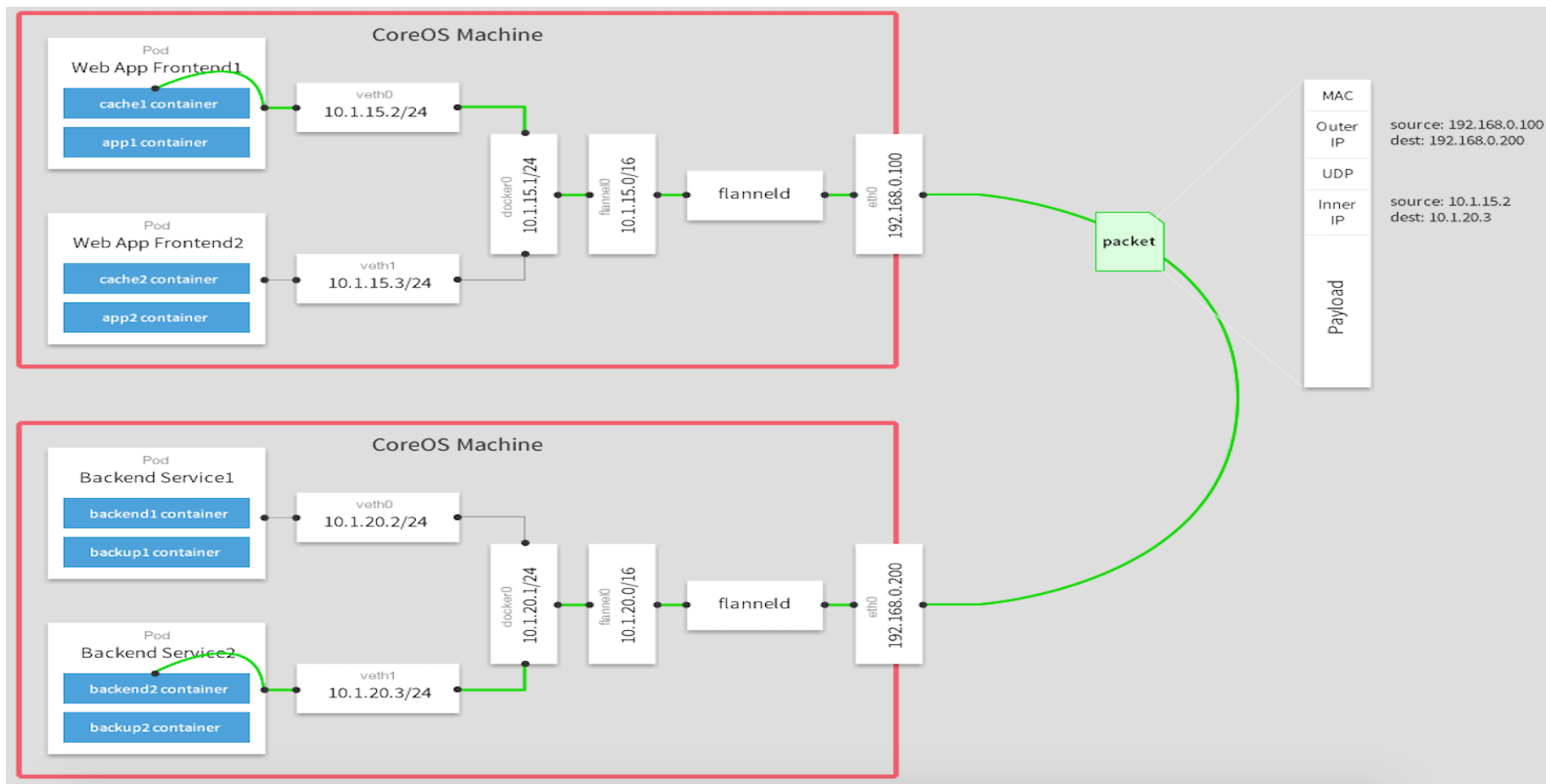
- Cluster IP
- Node Port
- Load Balancer
- (Ingress)



Kubernetes网络通信



Kubernetes Flannel不同Node中Pod通信



.NET Core在K8S下设计与实战

Thanks

E-mail	liuzhenyulive@live.com
Instagram	liuzhenyulive
Wechat	liuzhenyulive
Github	liuzhenyulive

