

Kubernetes

Попов Матвей

План

- Что это такое?
- Как оно устроено?
- Какие есть примитивы?
- Как они работают?
- Пробуем запустить локально
- Дополнительные инструменты



kubernetes

Что же это такое?

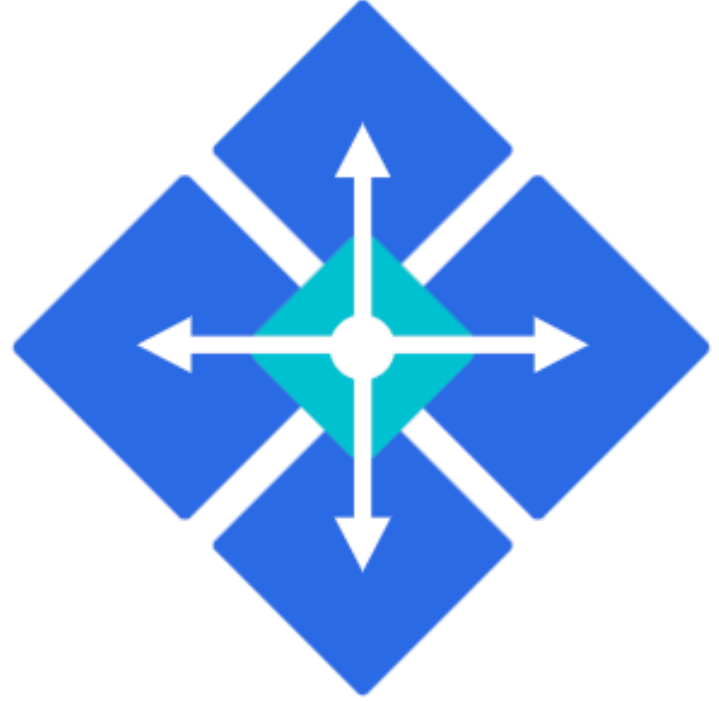
Kubernetes

Это портативная расширяемая платформа с открытым исходным кодом для управления контейнеризованными рабочими нагрузками и сервисами, которая облегчает как декларативную настройку, так и автоматизацию

Краткие факты:

- Управляет контейнерами
- Разрабатывалась Google с 2008 года
- Опубликована в 2014
- С 2015 имеет стабильную версию
- Написан на Go

Еще факты

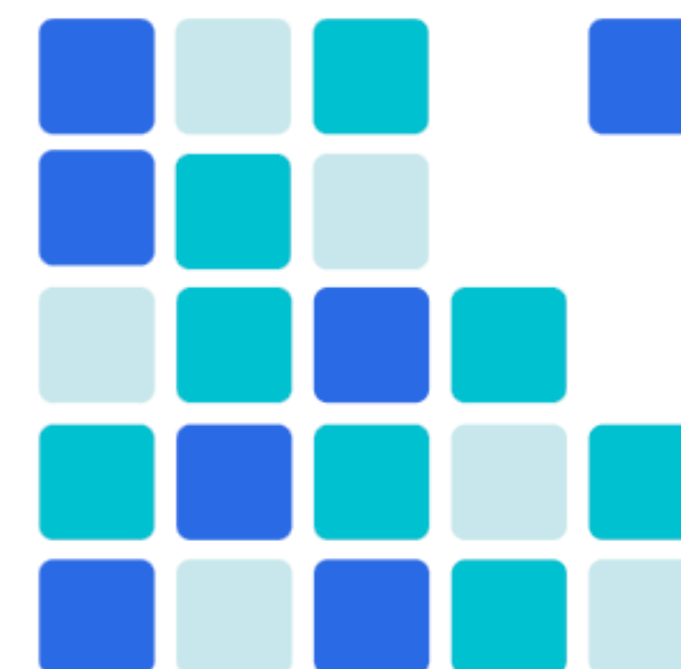


Масштабируемость любого уровня

Разработанный на тех же принципах, которые позволяют Google запускать миллиарды контейнеров в неделю, Kubernetes может масштабироваться без увеличения вашей технической команды.

Бесконечная гибкость

Будь то локальное тестирование или работа в корпорации, гибкость Kubernetes растёт вместе с вами, обеспечивая бесперебойную и простую доставку приложений независимо от сложности ваших потребностей.



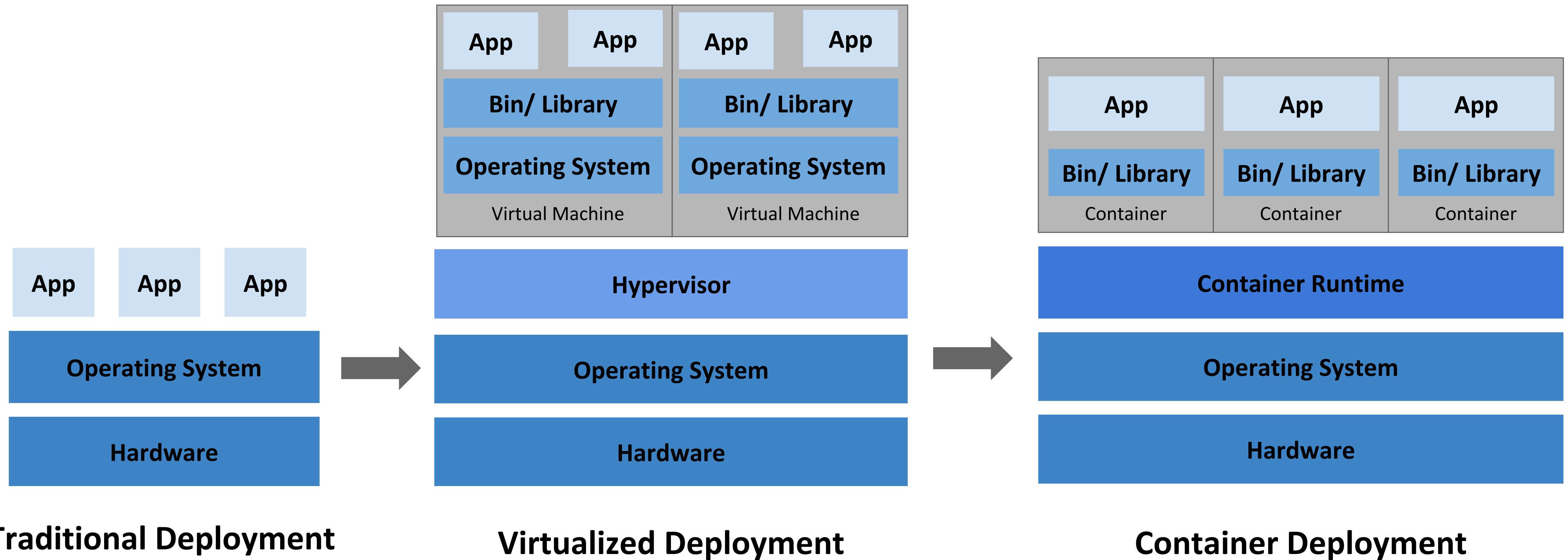


Может быть запущен где угодно

Kubernetes — это проект с открытым исходным кодом, который даёт вам полную свободу воспользоваться преимуществами локальной, гибридной или публичной облачной инфраструктуры, позволяя без усилий перераспределять рабочую нагрузку по мере необходимости.

Посмотрим в прошлое

Как менялась организация запуска приложения с течением времени?



Traditional Deployment:

- Запуск на физических серверах
- Нет способа ограничить границы ресурсов для приложения
- Запуск каждого нового приложения на отдельном сервере
- Поддержка большего количества физических серверов

Virtualised Deployment:

- Запуск нескольких виртуальных машин на одном физическом сервере
- Изоляция приложений между VM
- Эффективное использование ресурсов
- VM - полноценная машина

Container Deployment:

- Похожи на виртуальные машины
- Совместное использование ресурсов
- Очень гибкие, легкие и переносимые

Еще факты про контейнеры:

- Их нужно мониторить
- Между ними нужно распределять нагрузку
- Нужно автоматически развертывать и откатывать
- Нужно управлять конфигурацией

**Вот тут Kubernetes приходит
на помощь!**



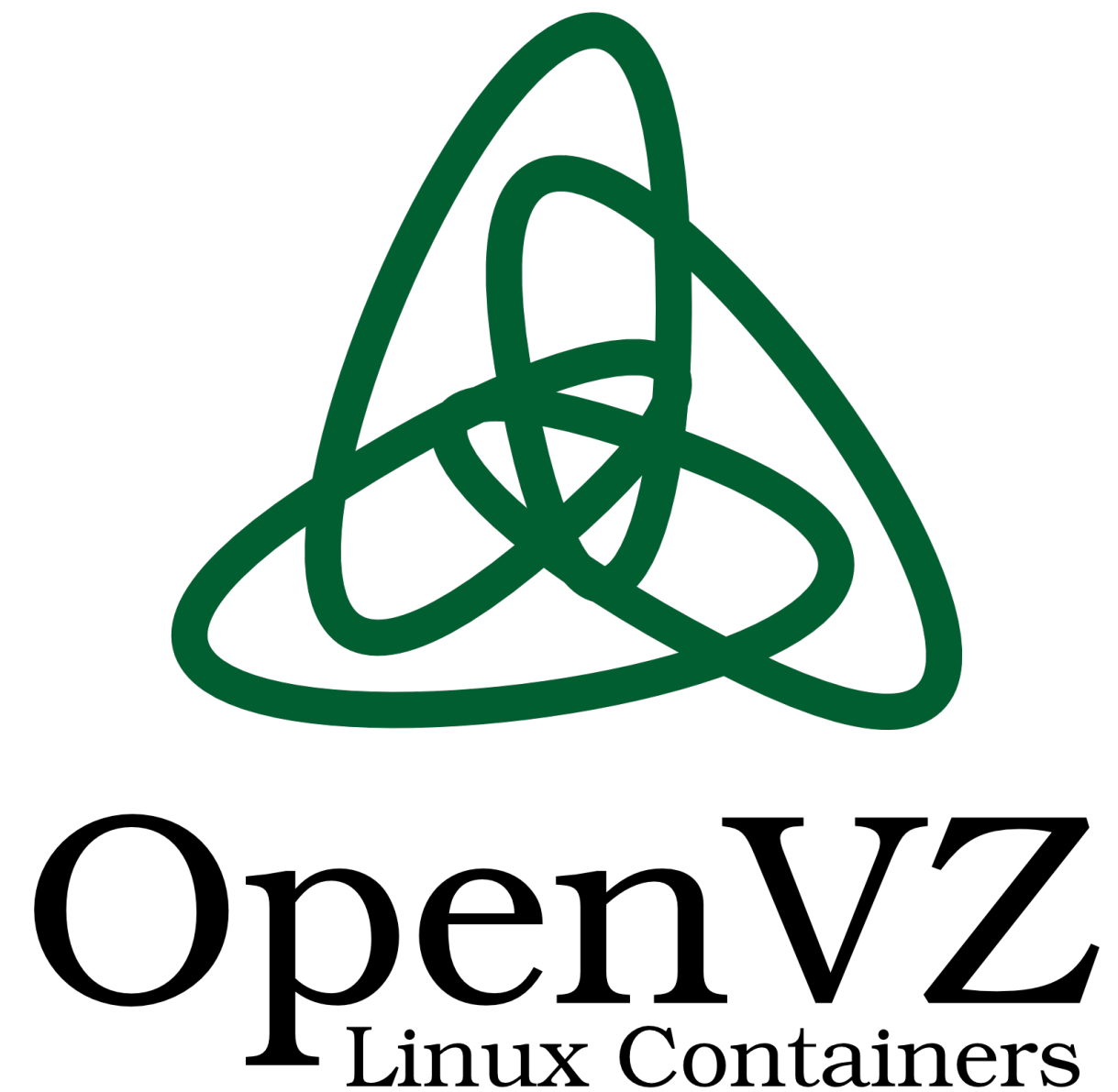
А что такое контейнеры?



Контейнеризация

Метод **виртуализации**, при котором ядро операционной системы поддерживает **несколько** экземпляров **пространства** пользователя вместо одного.

Какие есть реализации?



**Рассмотрим самое
популярное решение**



Краткий(очень) экскурс в Docker

Что это?

Очень простыми словами:

- ПО для виртуализации
- Упрощает разработку приложений и их развертывание
- Упаковывает приложения со всеми необходимыми зависимостями, конфигурациями, системными инструментами и runtime

Контейнер

Стандартизированная абстракция, которая имеет все что необходимо для запуска приложения

Посмотрим на базовые примитивы

Docker Image

- Описание как и с чем должен быть запущен контейнер
- Содержит описание ОС
- Все необходимые зависимости
- Описание как запустить приложение

**Как можно описать
инструкцию по запуску?**

С помощью Dockerfile

```
1 # базовый образ
2 >> FROM ubuntu:20.04
3
4 # установка желаемых пакетов и зависимостей
5 RUN apt-get update
6 RUN apt-get install -y wget && echo "We installed wget!"
7
8 # создание каталога /mytest
9 WORKDIR /mytest
10
11 # создание файла testfile
12 RUN touch testfile
13
14 # ls и echo будут выполняться при запуске контейнера
15 CMD ls && echo "Show me mytest"
16
```

Docker Container

- Работающий экземпляр Docker Image
- Из одного Docker Image может быть запущено много Docker Container

Итог

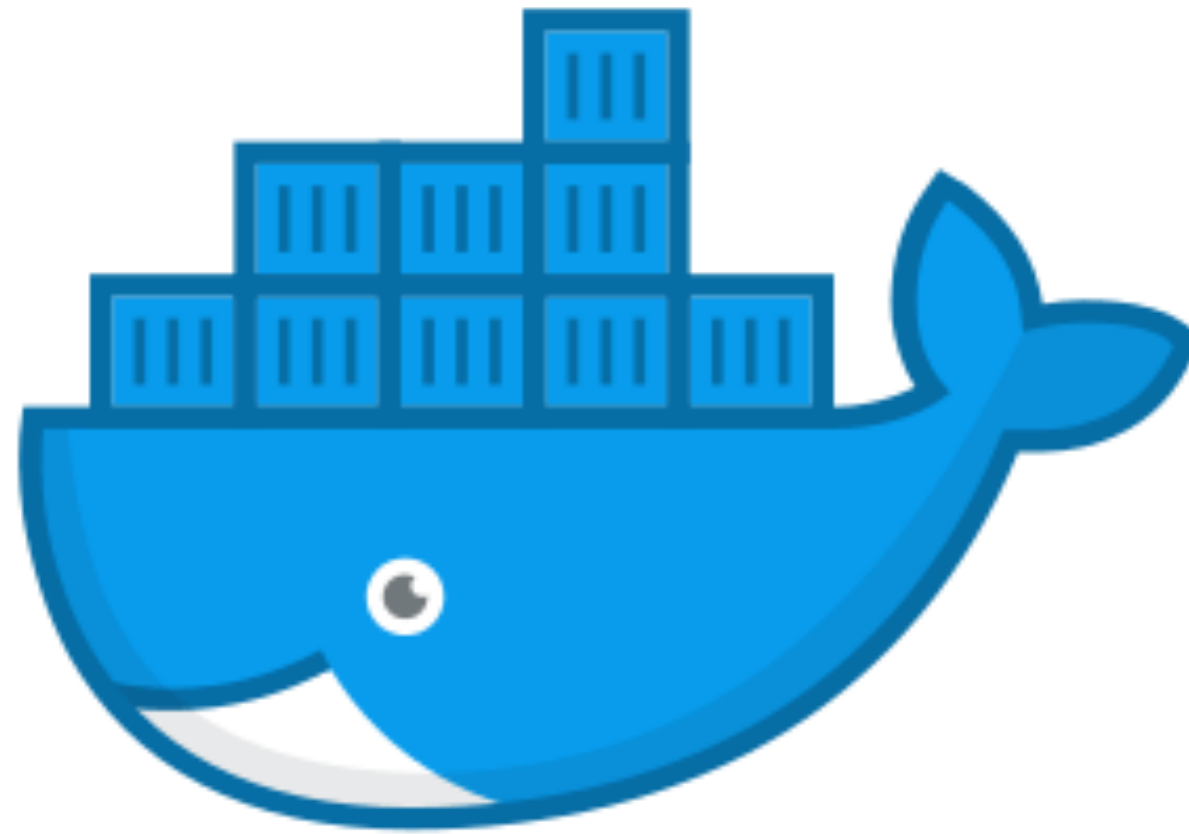
Dockerfile



Build



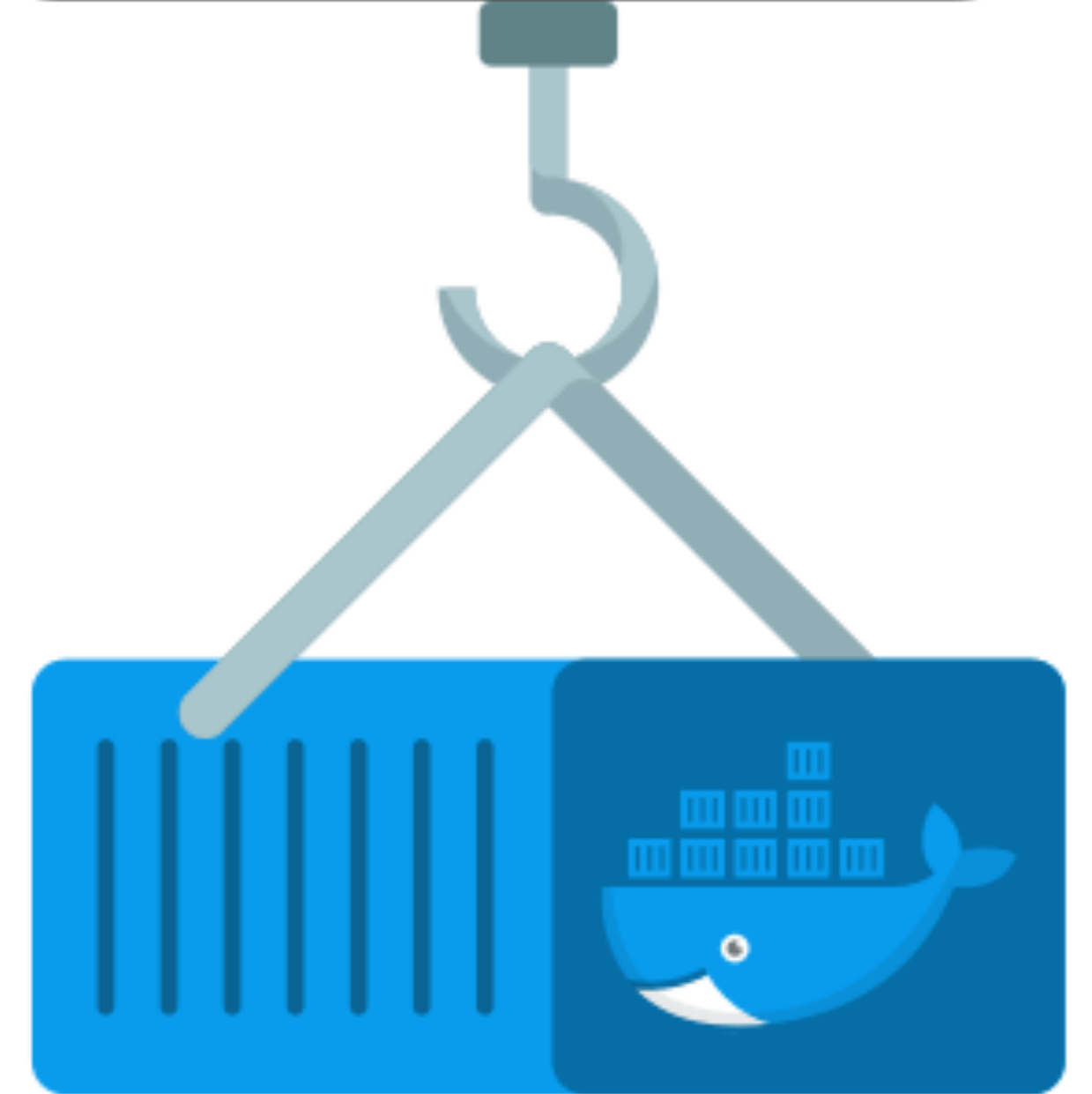
Docker Image



Run



Docker Container



Docker Registry

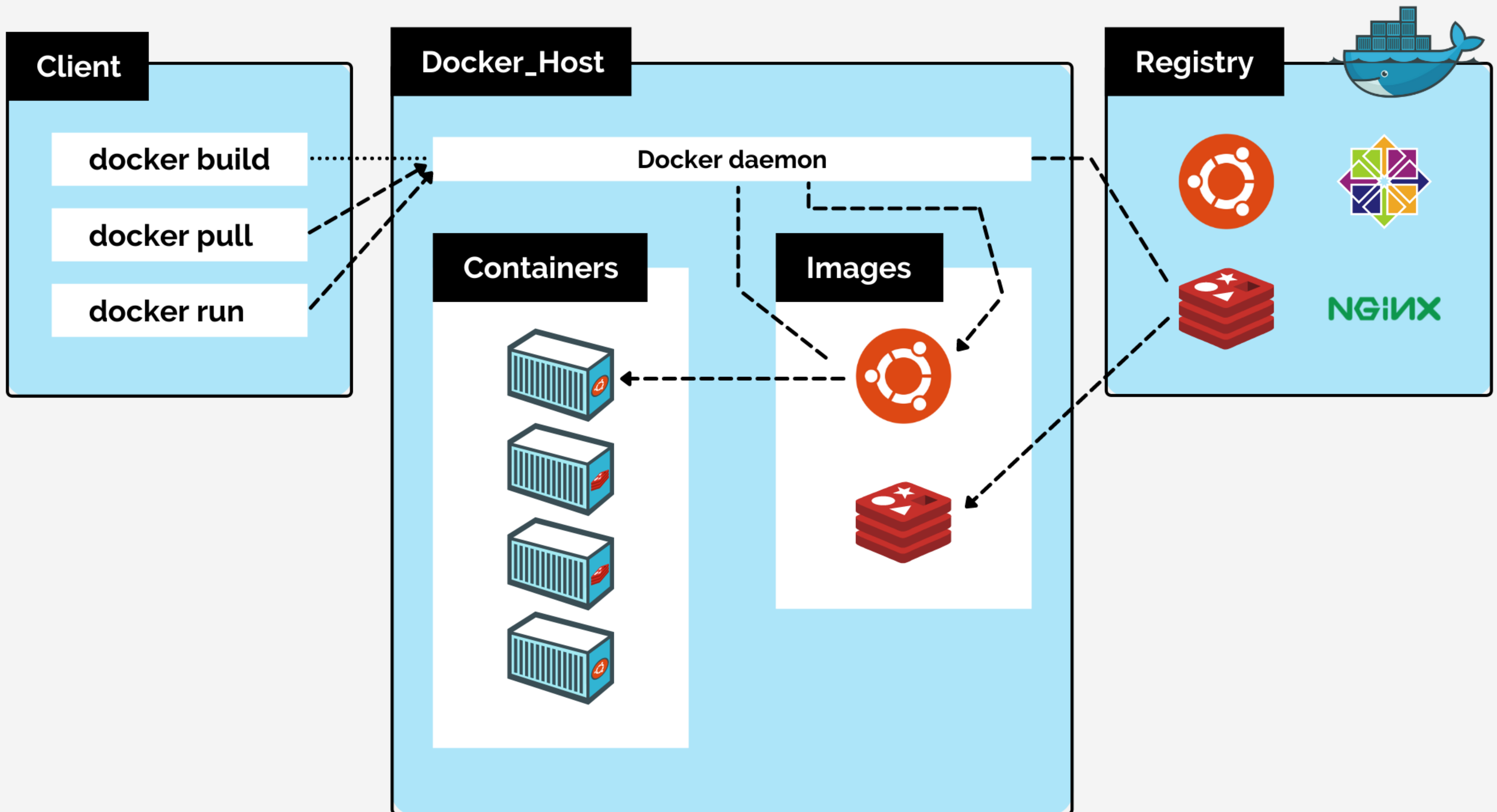
- Мы знаем как получить Docker Container из Docker Image
- Но как получить Docker Image?

**Они хранятся в Docker
Registries!**

**Docker Registry - хранилище с
версионированными докер-образами**

Как это может быть полезно?

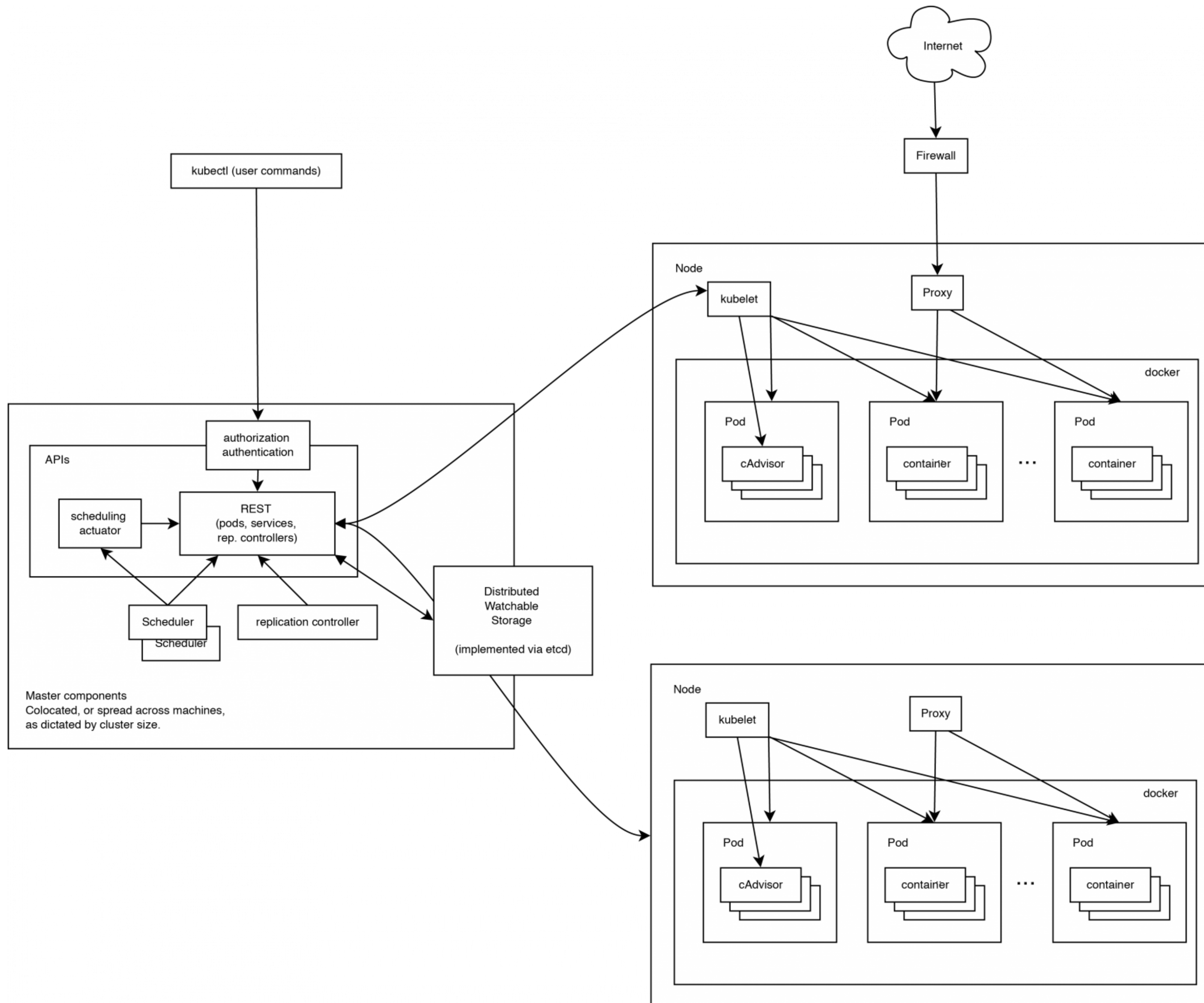
- Собираем на рабочем ПК образ и отправляем в Registry
- На сервере забираем образ из Registry и запускаем



**С полученными знаниями
вернемся к Kubernetes**

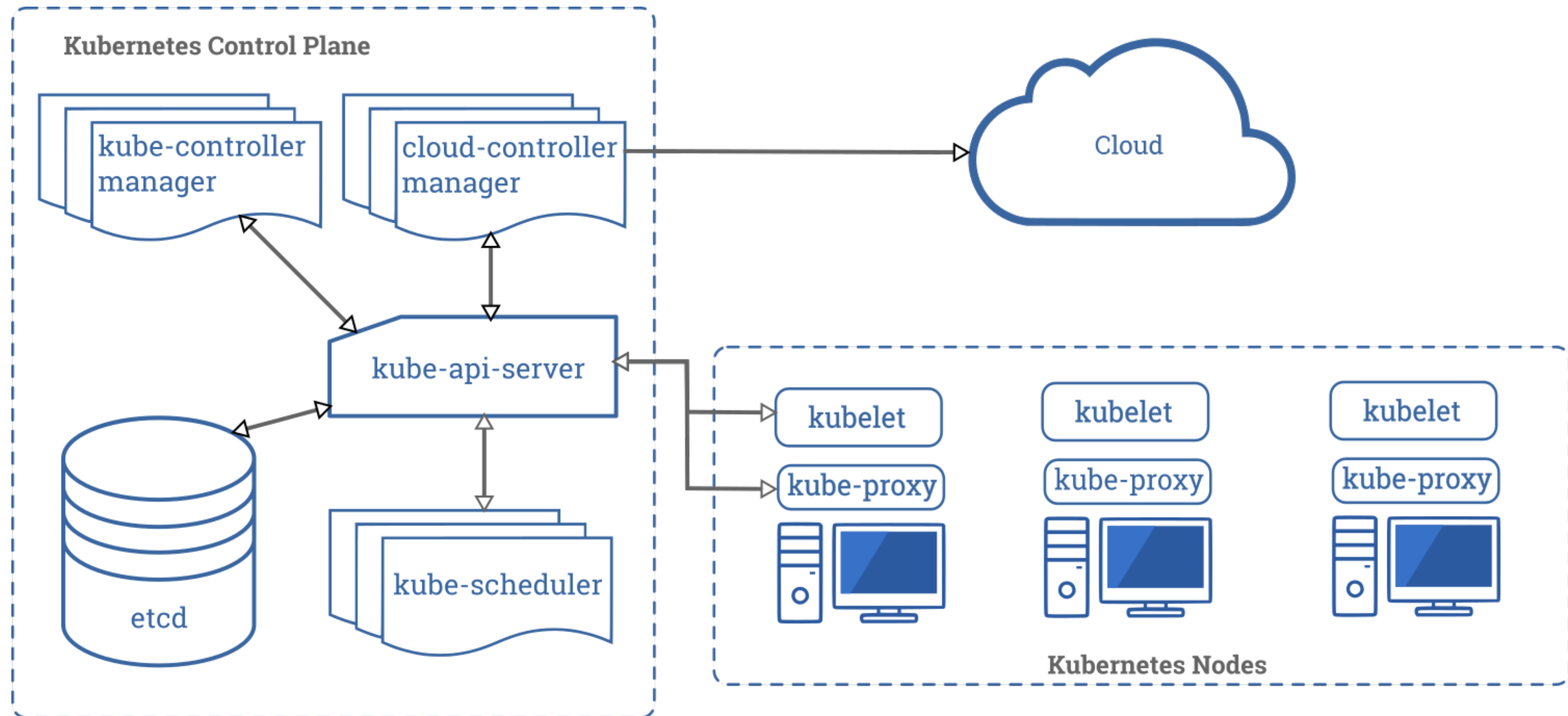
Какие есть компоненты у k8s?

- Основной компонент это Cluster
- Каждый Cluster состоит из Node
- Node существуют двух типов:
 - Worker Node - сервер на котором запускаются и работают контейнеры
 - Master Node - сервер который управляет всеми Worker Node
- Все команды взаимодействия от клиентов всегда идут к Master Node

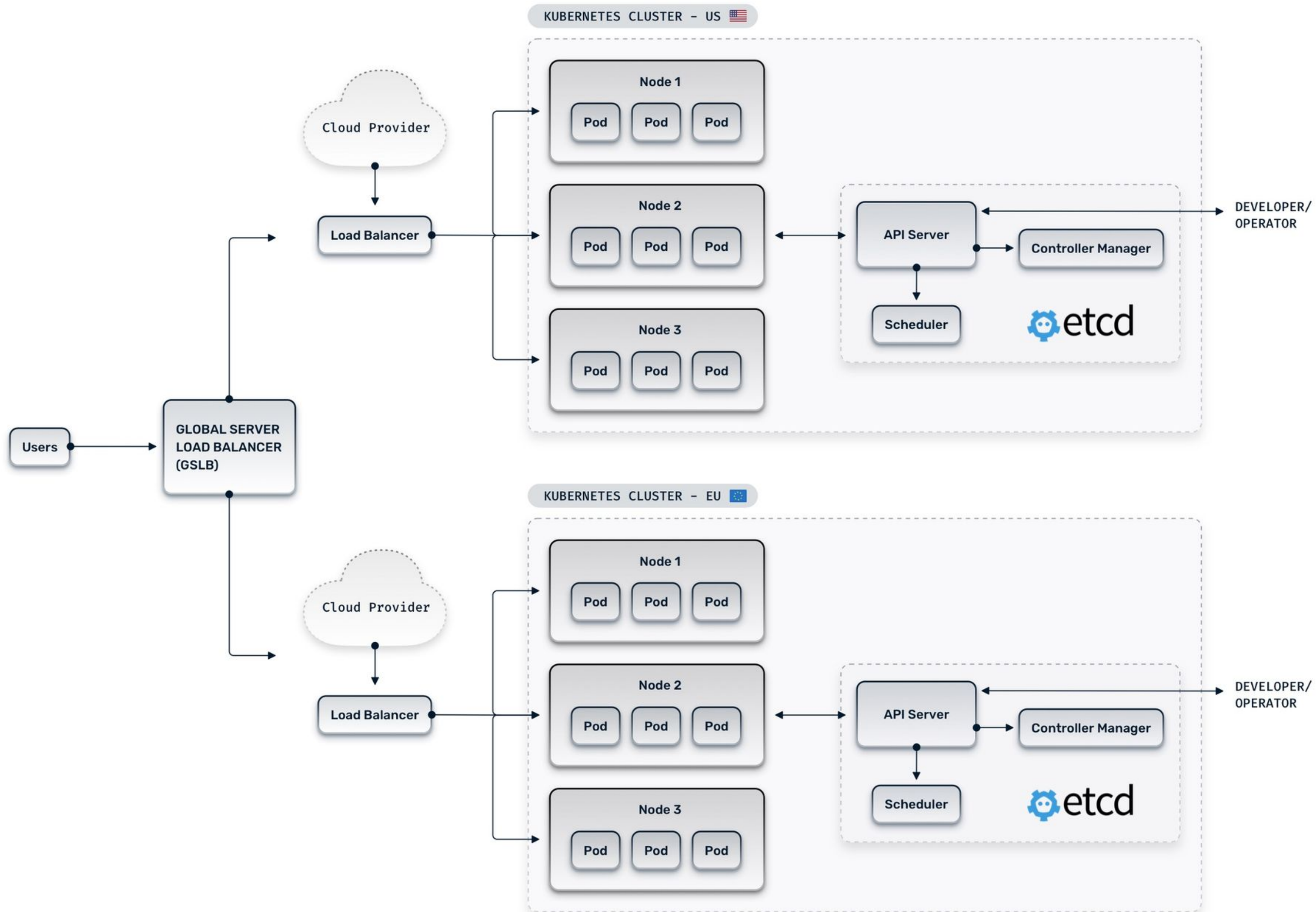


Как-то сложно...

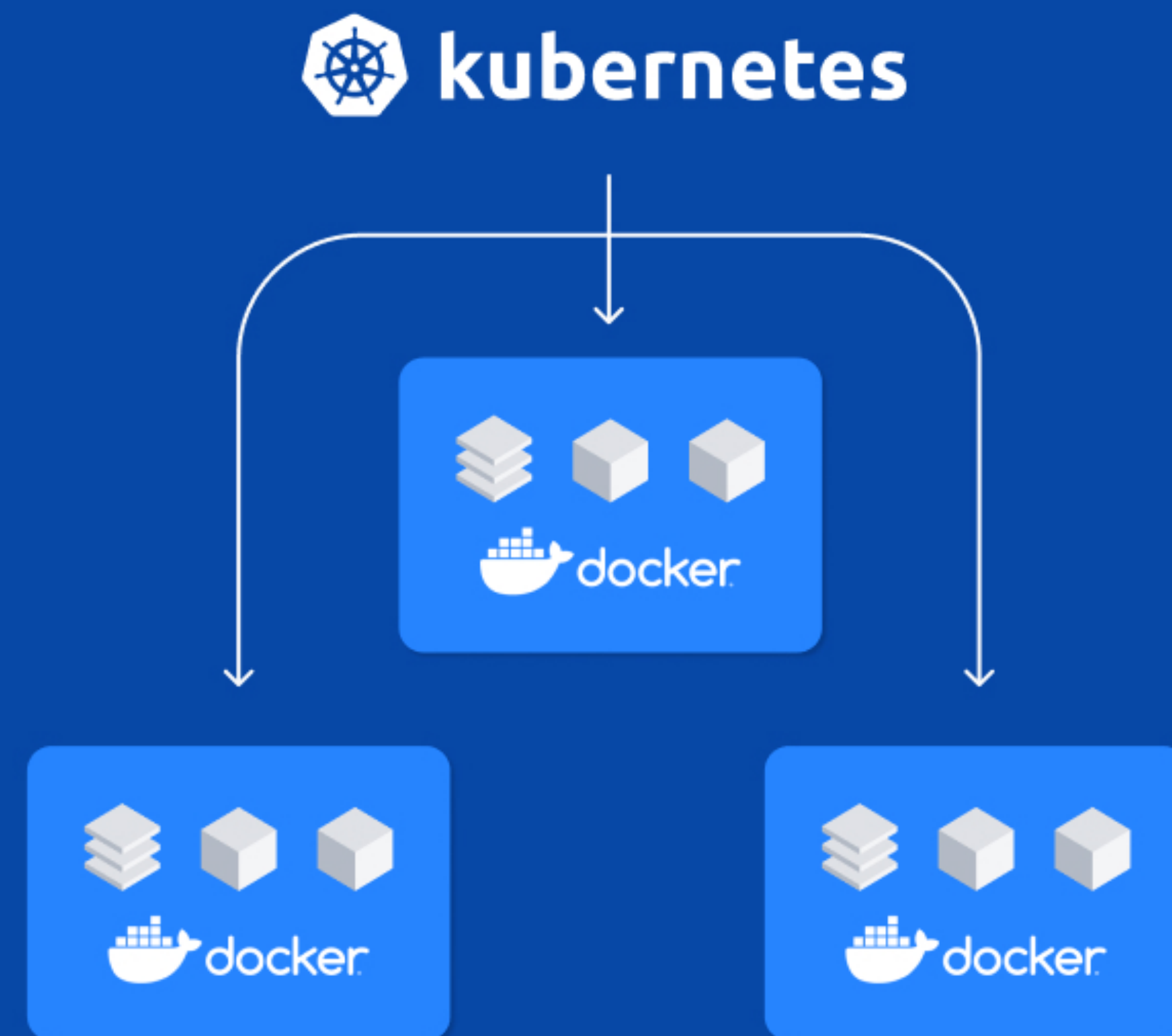
Есть схема попроще



Как обычно выглядит Cluster?



**Можно провести аналогию с
различными Deployment**



Virtualized deployment

Container deployment

Kubernetes deployment

**Какие есть пользовательские
компоненты?**

Самые основные:

- Container
- Pod
- Deployment
- Service
- Ingress
- Nodes
- Cluster

A также:

- DaemonSet
- StatefulSet
- ReplicaSet
- DeploymentSet
- Secret
- Job
- CronJob
- PV
- SVC
- LoadBalancer
- ConfigMaps
- Vertical Pod AutoScaler
- Horizontal Pod AutoScaler
- ...

Начнем с Pod

Что это такое?

Pod представляет собой **запрос на запуск одного или более контейнеров** на одном узле(Node)

**Pod - это один и единственный
объект в k8s, который приводит к
запуску контейнеров**

Deployment

Service

Ingress

**А теперь давайте запустим k8s
локально!**

**Какие есть еще технологии
рядом с k8s?**

Итоги

Вопросы?

Материалы

Будут в README

Спасибо за внимание!