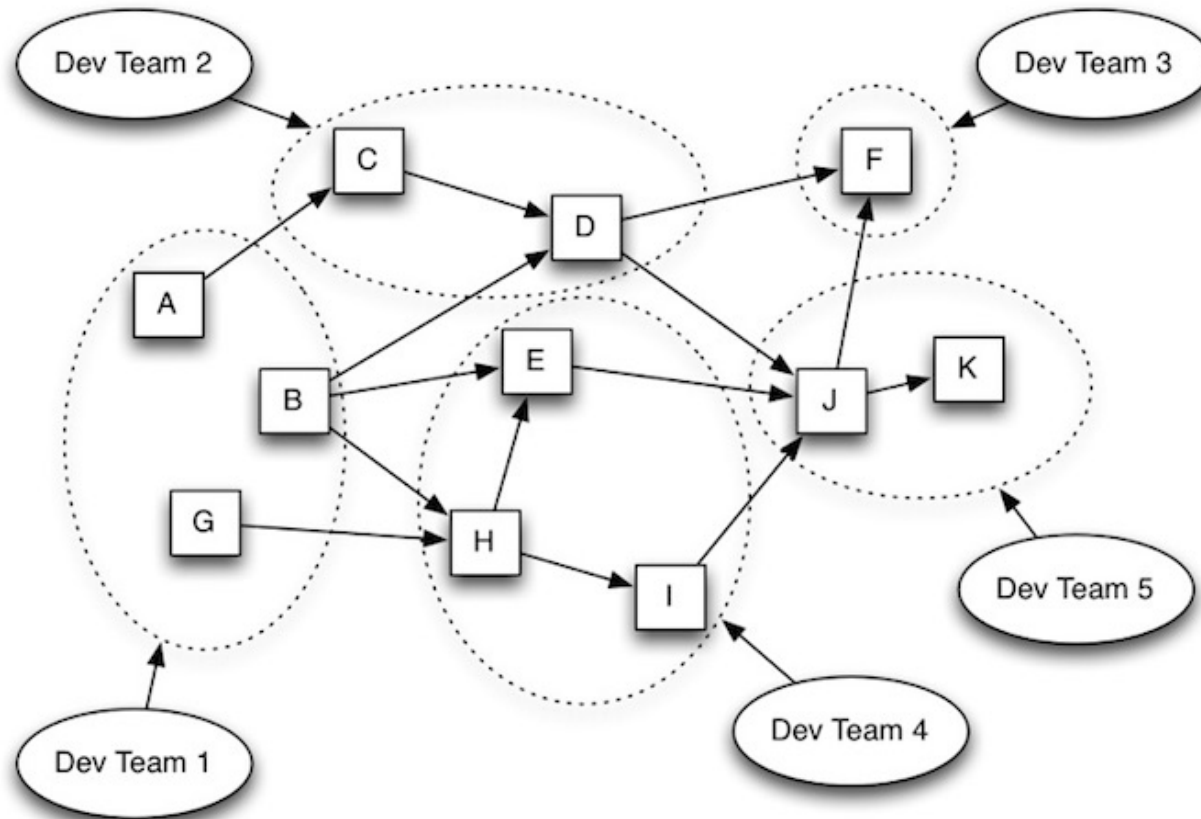


# Kubernetes

# Устройство проекта | Простой случай

- Несколько сервисов/приложений
- Просто деплоить и управлять конфигурацией
- Инженеры принимают решение куда деплоить
- Инженеры хранят знания о том, сколько ресурсов нужно

# Микросервисы



# Kubernetes

- Он же “кубер”, он же `k8s`
- Переосмысление проекта **Borg** от Google в контексте контейнеров
- >**65 000** звезд на GitHub
- >**90 000** коммитов
- Более 4 лет в Open Source
- Версия 1.0 была выпущена 21 июля 2015
- Тогда же Kubernetes присоединился к CNCF и стал первым **выпускником**

# Kubernetes

## Фреймворк для создания платформы

Входит в комплект	Легко добавить
Автомасштабирование	Мониторинг
Автовосстановление	Логирование
Service Discovery	CI/CD
Управление инфраструктурой (Storage, Networks)	
Автоматическое распределение нагрузки*	

# OpenShift



## OPENSIFT

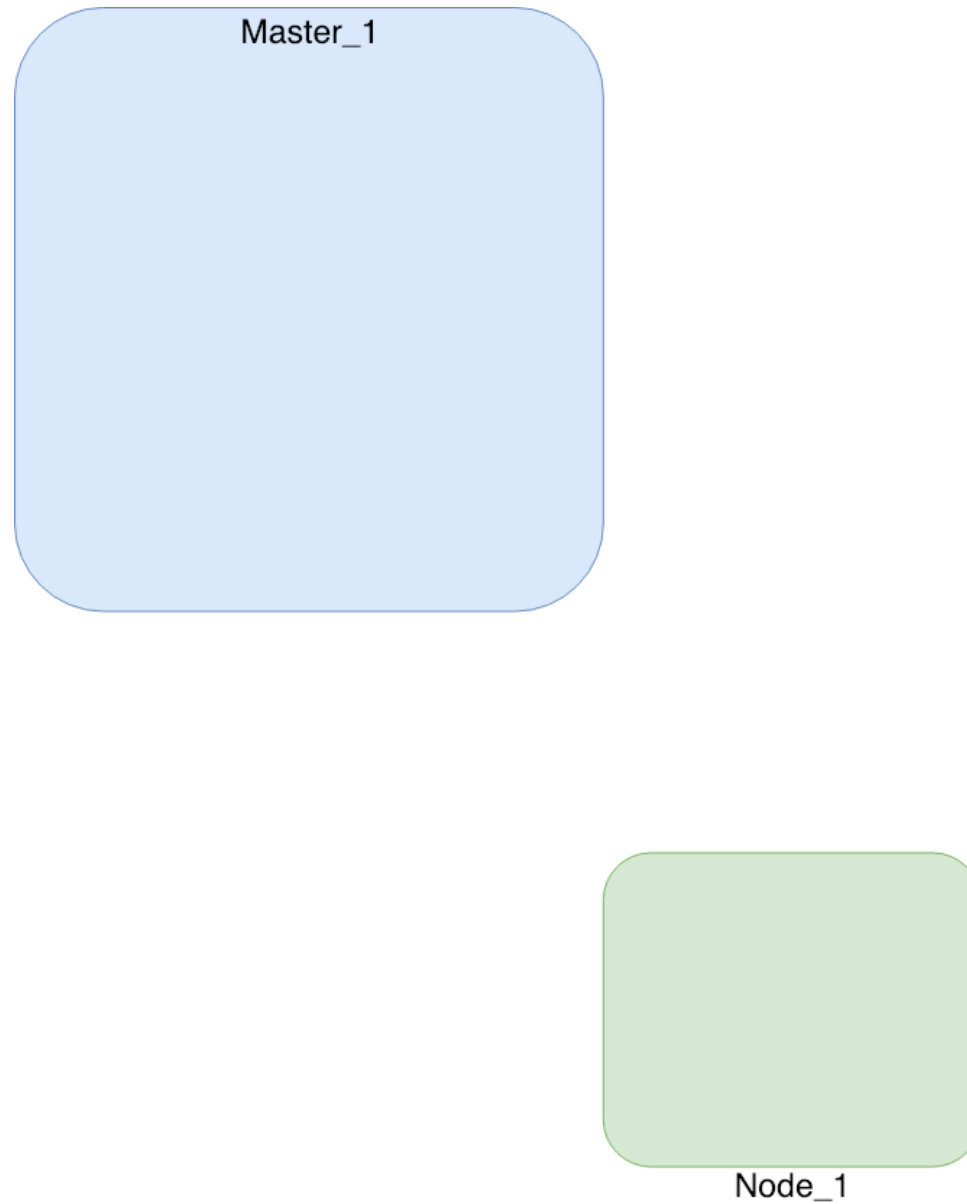
LIFECYCLE AUTOMATION  
(Self-service, CI/CD, Image Stream)

CONTAINER MANAGEMENT  
(Monitoring, Capacity, Policies)

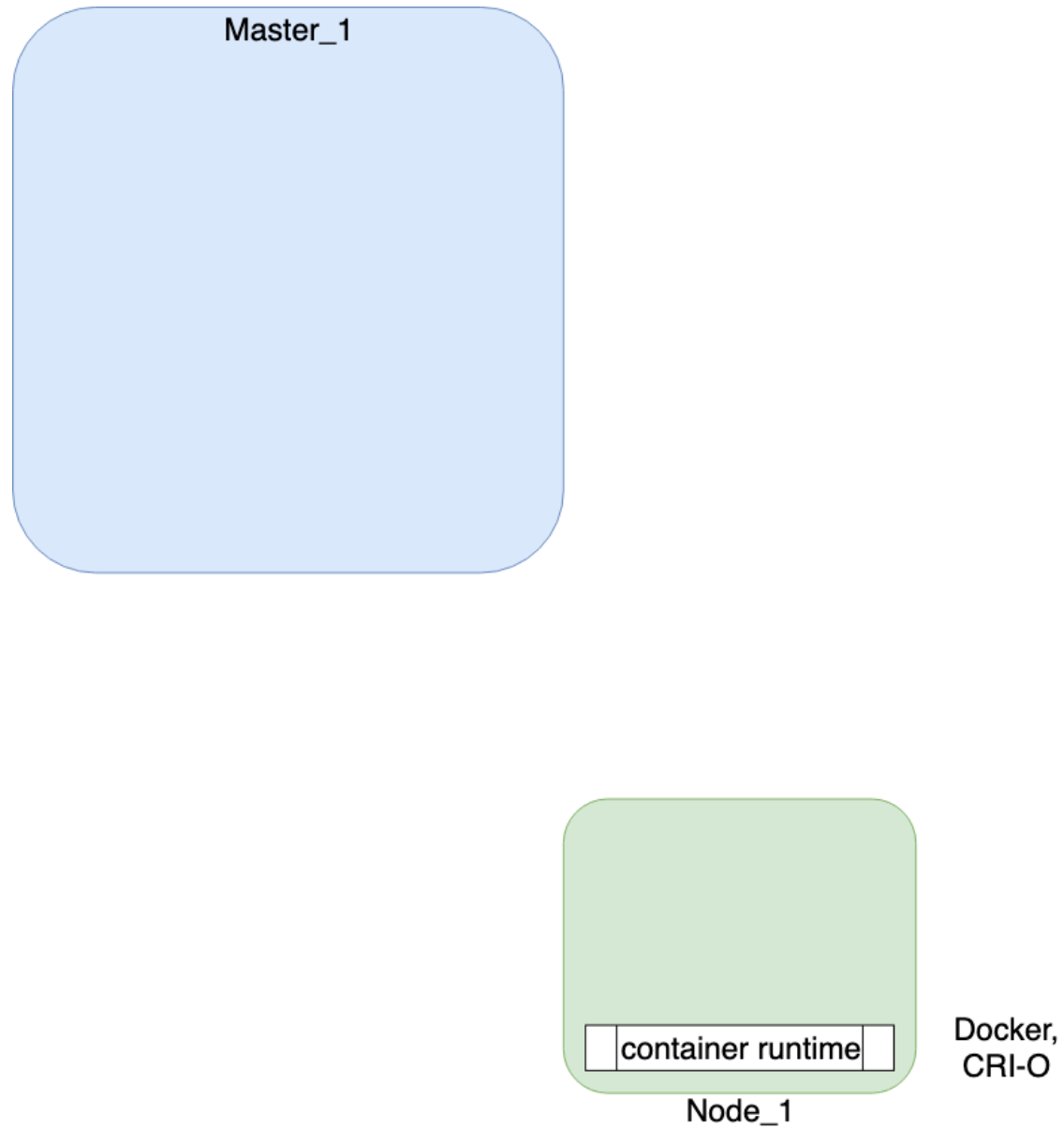
CONTAINER INFRASTRUCTURE  
(Orchestration & Scheduling, Storage, Registry, Security, Networking)

# Архитектура Kubernetes

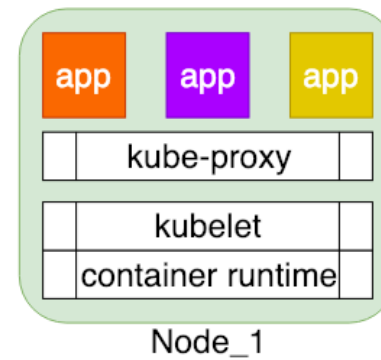
# Архитектура Kubernetes



# Архитектура Kubernetes



# Архитектура Kubernetes



# Архитектура Kubernetes

## kubelet

`kubelet` отвечает за управление Pod

- Общается с API-Server
- Разворачивает Pods согласно PodSpec (описание Pod в формате YAML или JSON)
- Взаимодействует с Container Runtime, сетевыми и storage-плагинами для создания и настройки изолированного окружения

# Архитектура Kubernetes

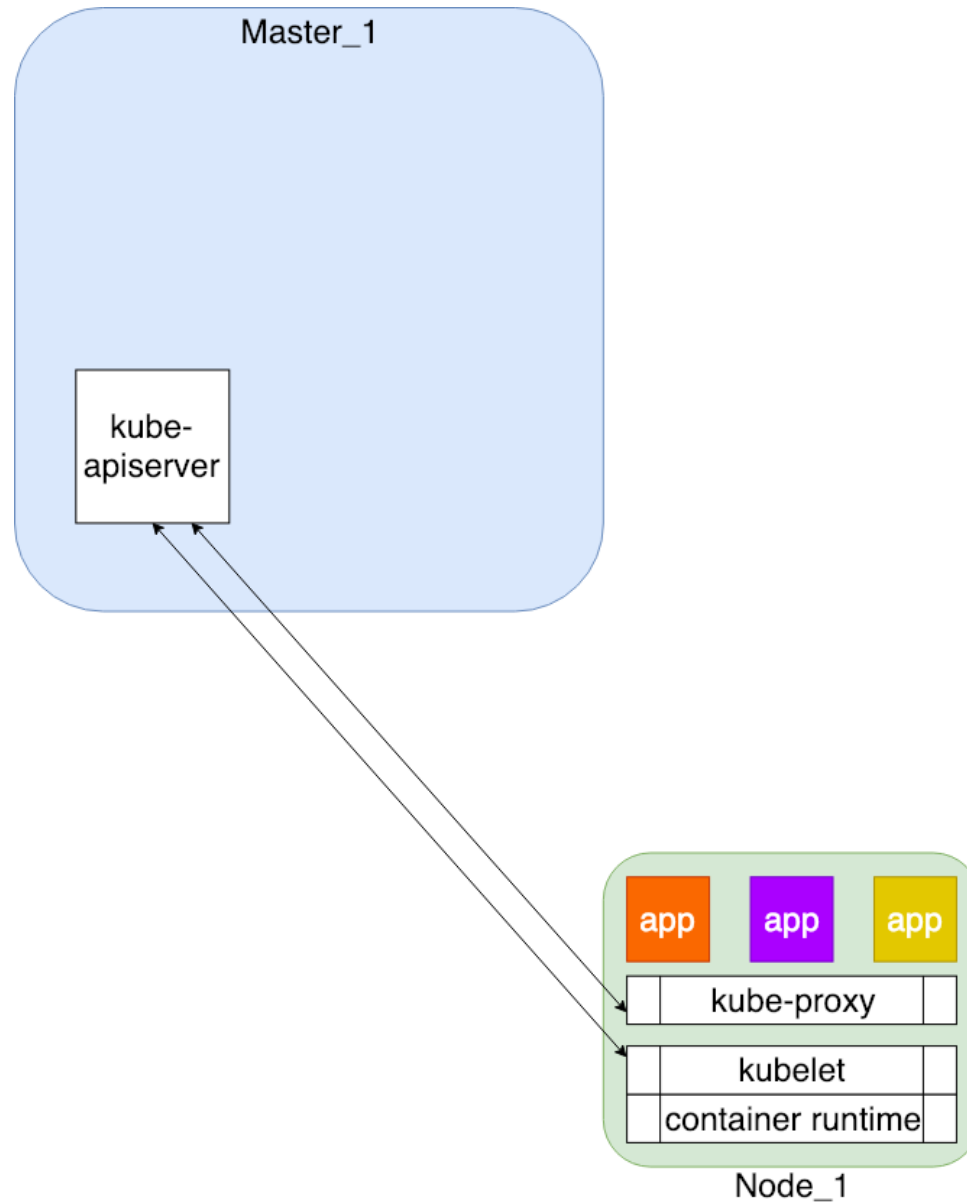
## kube-proxy

`kube-proxy` - это компонент, который обеспечивает:

- Работу с сервисами Kubernetes
- Балансировку нагрузки и управление отказами
- Когда-то давно проксировал через себя TCP/UDP соединения (не HTTP)

Под капотом он управляет `iptables` и `IPVS`

# Архитектура Kubernetes

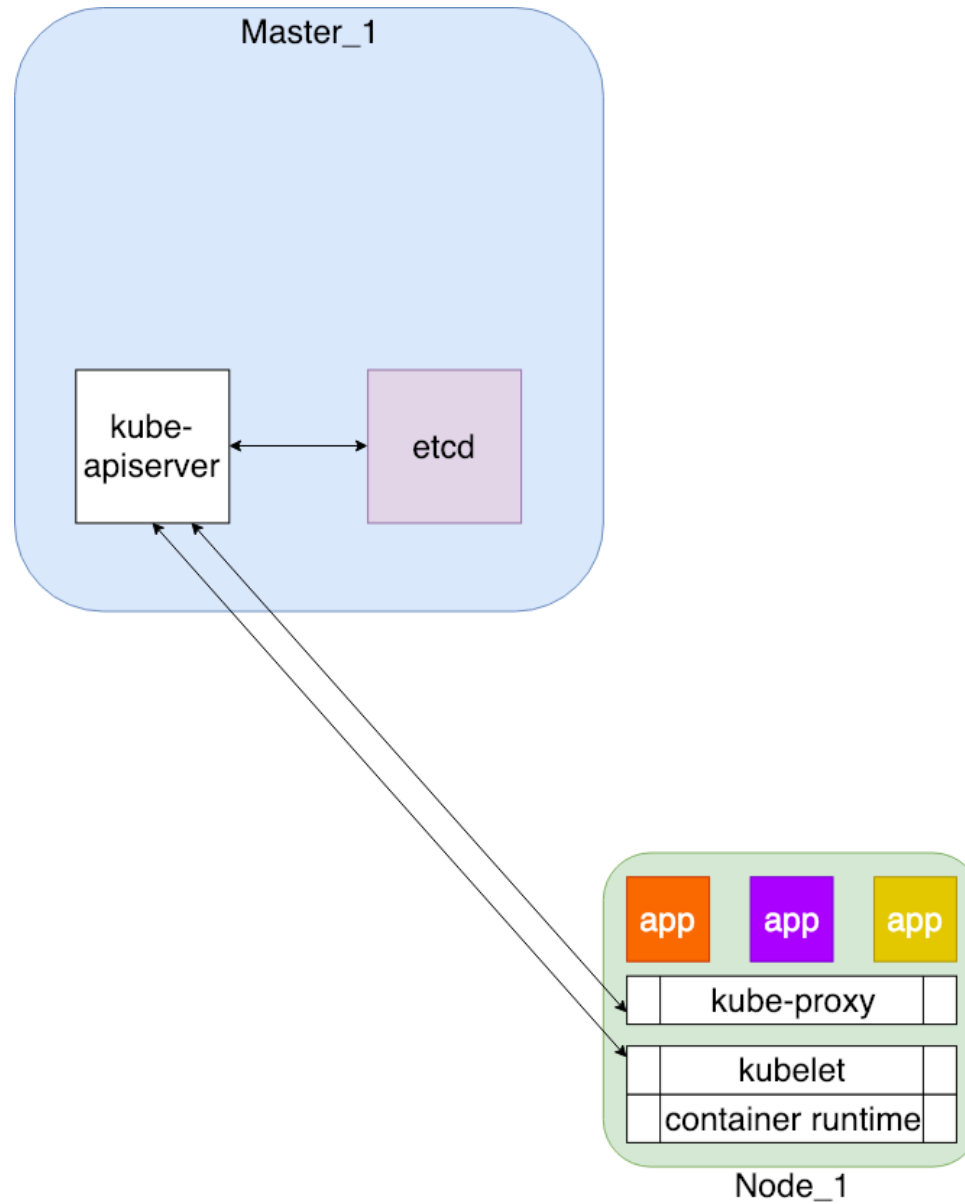


# Архитектура Kubernetes

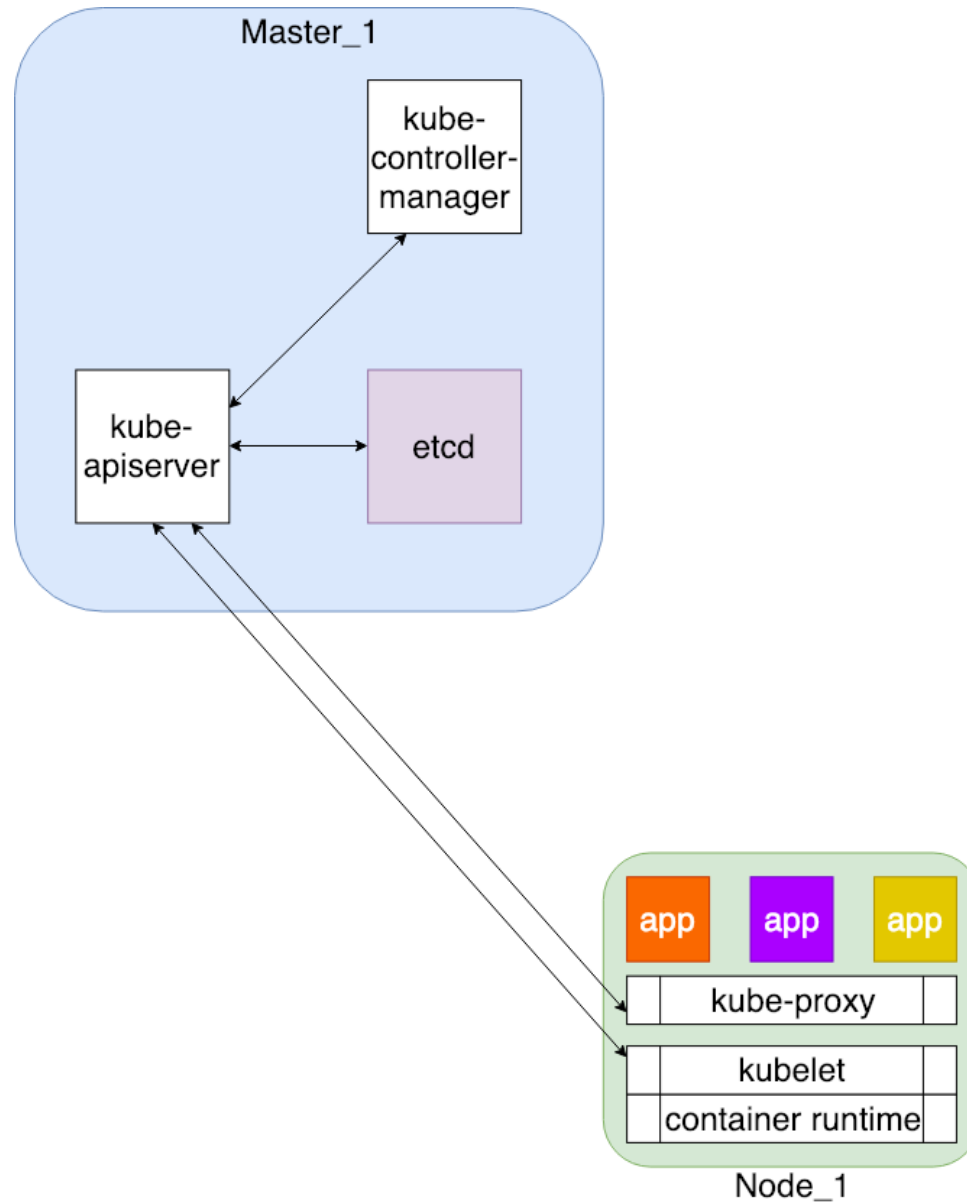
## kube-apiserver

- API-шлюз - точка входа для всех взаимодействий компонентов Kubernetes
- Предоставляет REST-сервис (работа над состояниями объектов)
- Проверяет и конфигурирует API-объекты (pods, services, ...)
- Сохраняет состояние в etcd

# Архитектура Kubernetes



# Архитектура Kubernetes

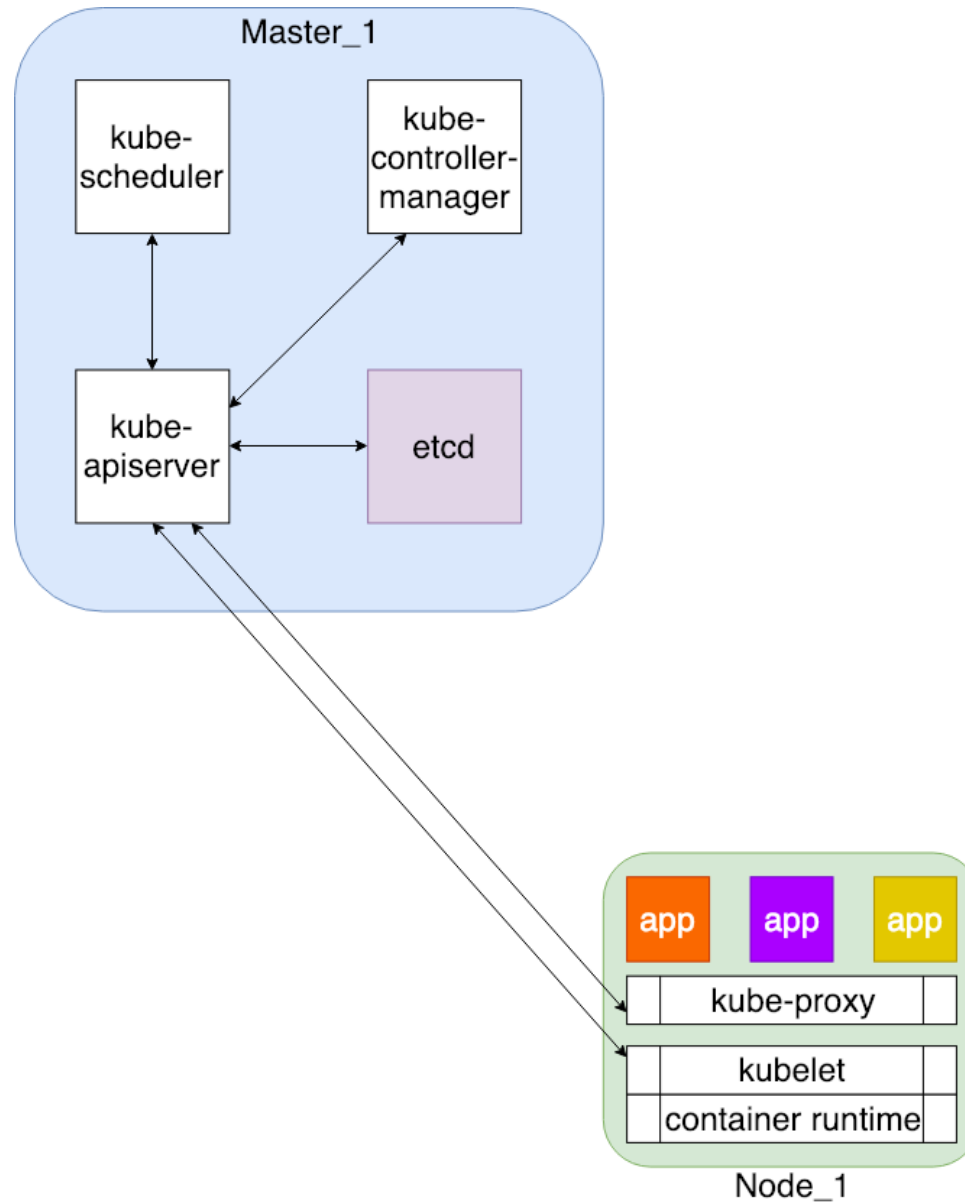


# Архитектура Kubernetes

## kube-controller-manager

- Отслеживает через API изменение состояний объектов
- Набор различных контроллеров:
  - ReplicaSet controller
  - Endpoints controller
  - Namespace controller
  - ...
- Контроллеры сравнивают состояние кластера с описаниями объектов и устраняют несоответствия

# Архитектура Kubernetes



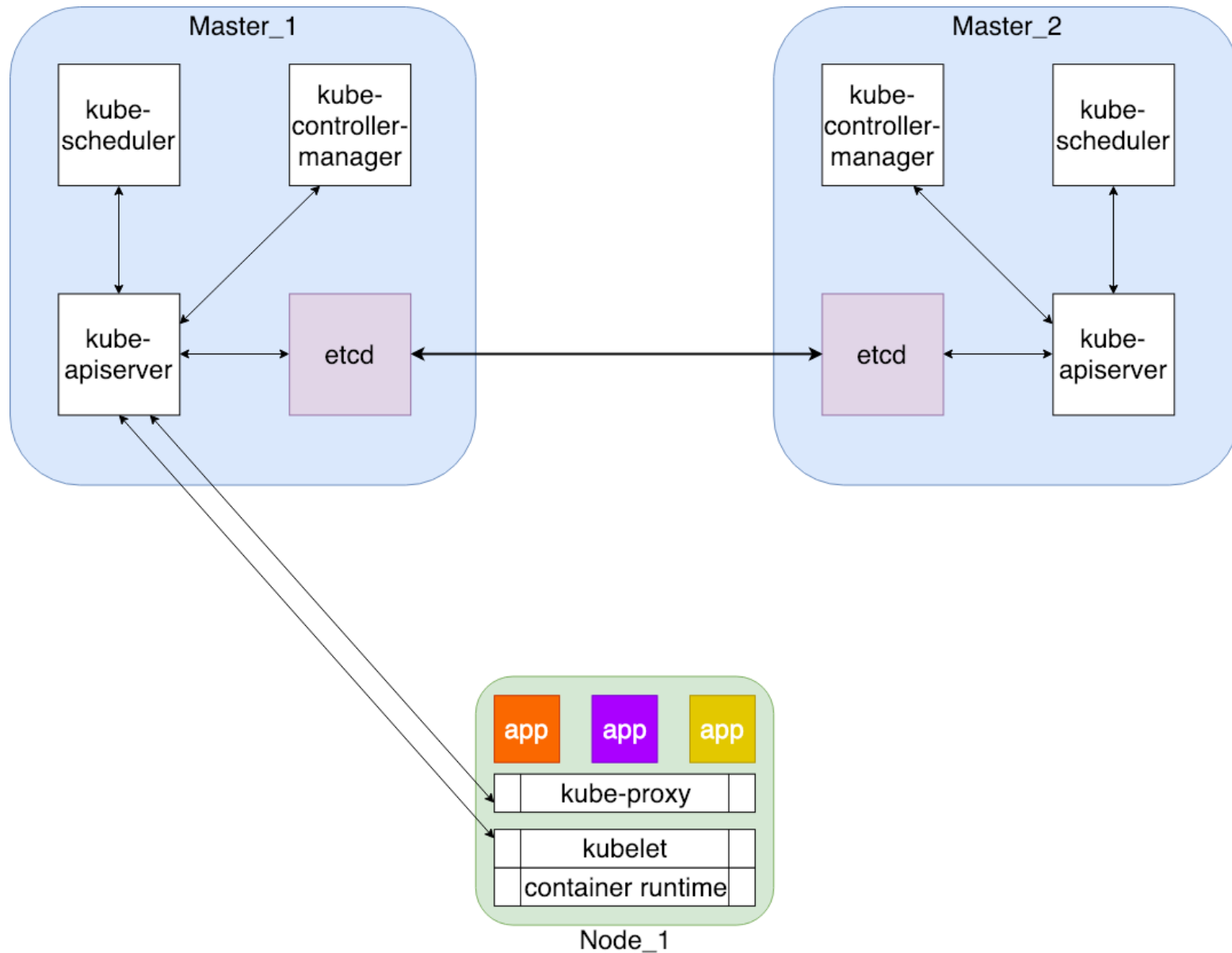
# Архитектура Kubernetes

## kube-scheduler

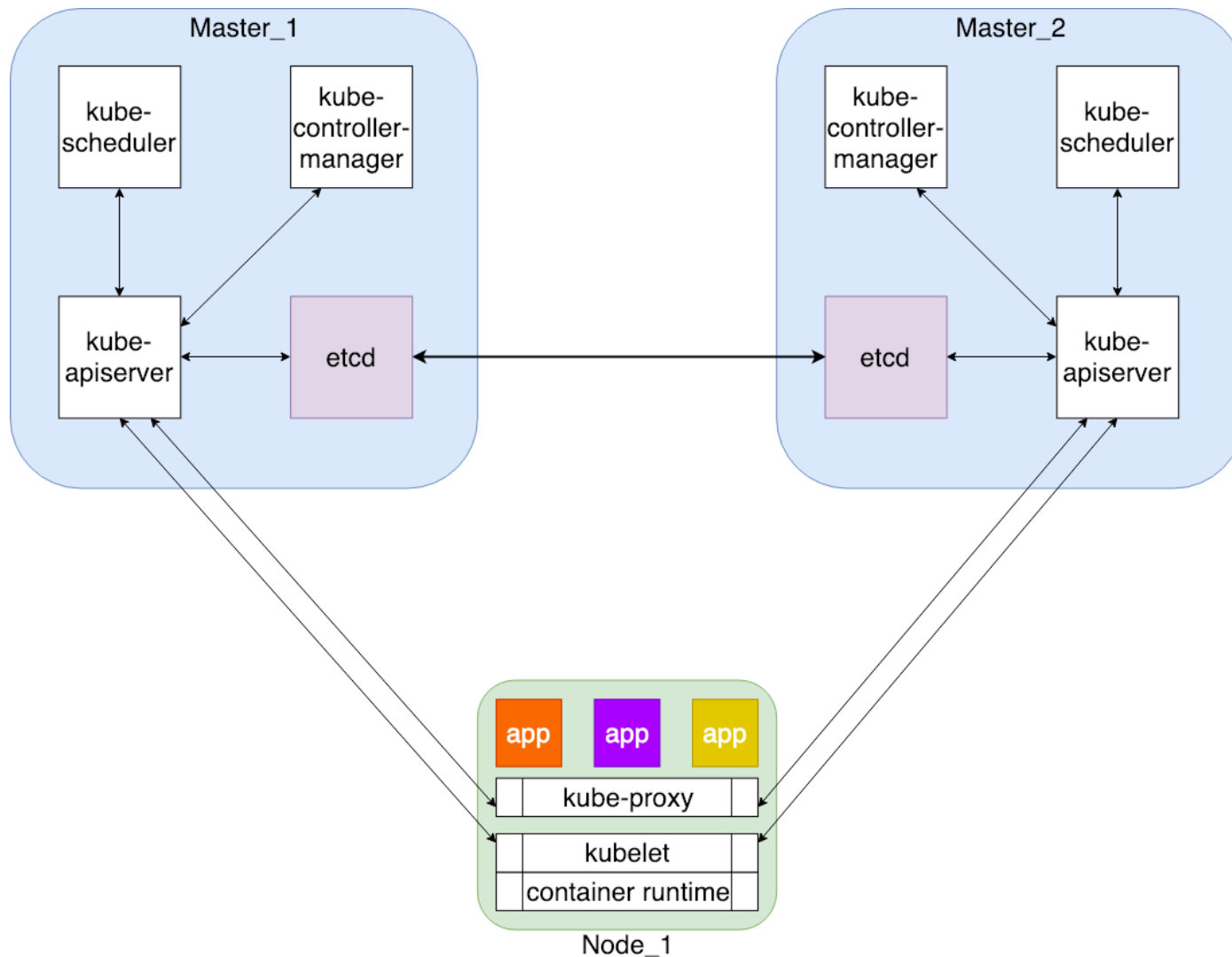
- Учет имеющихся ресурсов
- Учет требований к ресурсам
- Учет ограничений по ресурсам (affinity, anti-affinity)
- Решение где запустить контейнеры (на основе ограничений и требований)

```
1  while True:  
2      pods = get_all_pods()  
3      for pod in pods:  
4          if pod.node == nil:  
5              assignNode(pod)
```

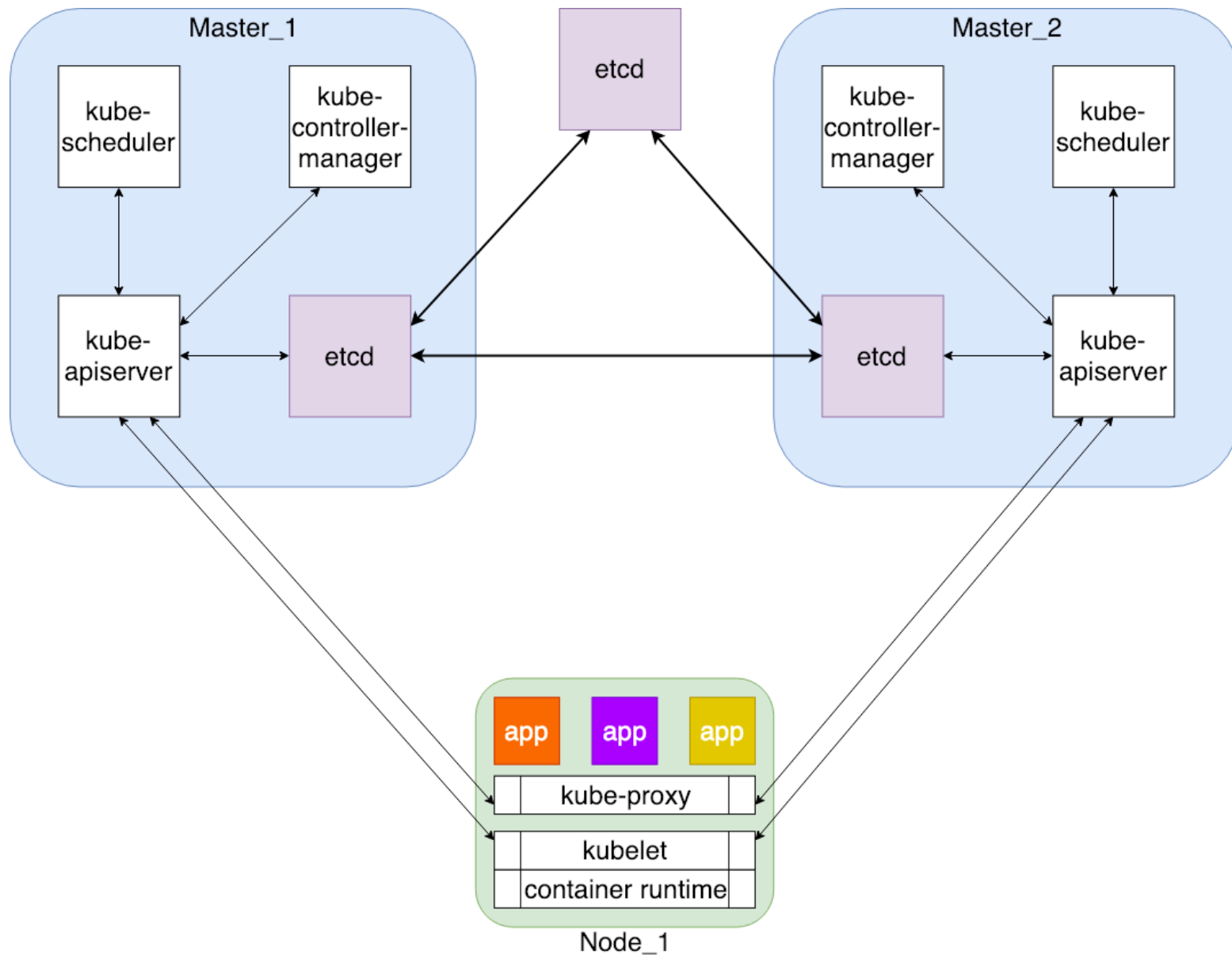
# Архитектура Kubernetes



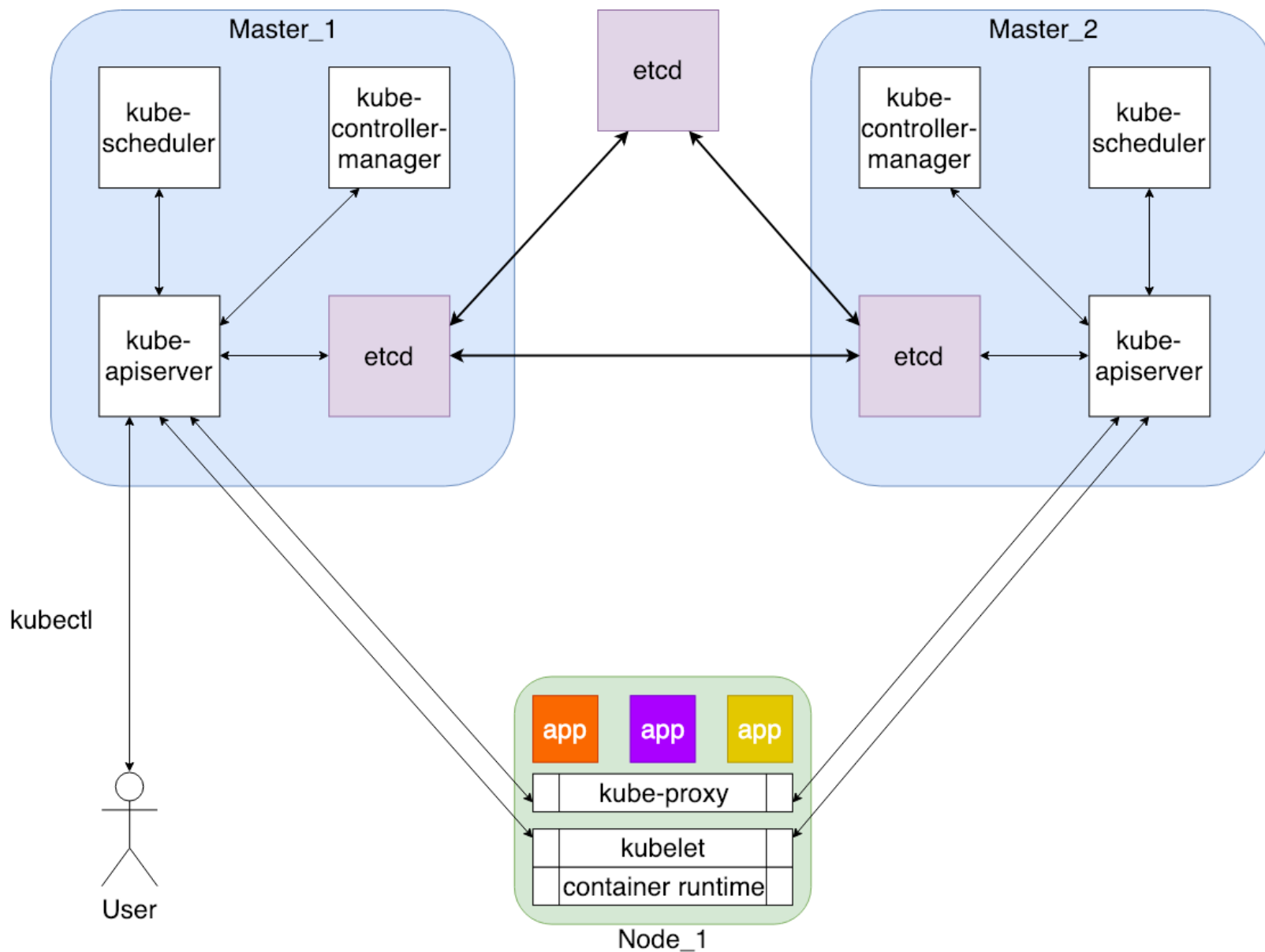
# Архитектура Kubernetes



# Архитектура Kubernetes



# Архитектура Kubernetes



# Архитектура Kubernetes

## Addons

- Расширяют функционал Kubernetes
- Запускаются также, как и остальные сервисы и поды
- Взаимодействуют с Kubernetes через API

## Примеры

- kube-dns/CoreDNS
- Сетевые плагины (Weave, Calico, Flannel)
- Dashboard/Weave Scope
- ...

Обзор и список на официальном сайте

# Kubernetes

- Мотивирует к IaC
- Простое создание идентичных окружений (динамичных, постоянных)
- Отдельно взятый сервер не важен