Распределенная микросервисная архитектура приложений в оркестраторе kubernetes

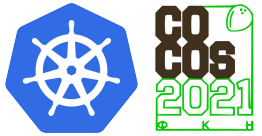
Научный руководитель

Панфилов Петр Борисович
Профессор, Высшая школа бизнеса

Докладчик

Волков Андрей Андреевич
Студент 4 курса, «Бизнес-информатика»

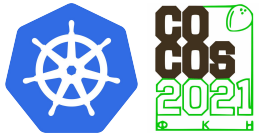
Мотивация



Мотивация

Мартин Фаулер — автор ряда книг и статей по архитектуре ПО.

Мотивация



Преимущества микросервисной архитектуры

Мартин Фаулер — автор ряда книг и статей по архитектуре ПО.

Мотивация

Преимущества микросервисной архитектуры

Сильные границы модулей: микросервисы усиливают модульную структуру, что особенно важно для больших команд.

Мартин Фаулер — автор ряда книг и статей по архитектуре ПО.

Мотивация

Преимущества микросервисной архитектуры

Сильные границы модулей: микросервисы усиливают модульную структуру, что особенно важно для больших команд.

Независимое развертывание: простые сервисы легче разворачивать, и, поскольку они автономны, они с меньшей вероятностью вызовут сбои системы, если что-то пойдет не так.

Мартин Фаулер — автор ряда книг и статей по архитектуре ПО.

Мотивация

Преимущества микросервисной архитектуры

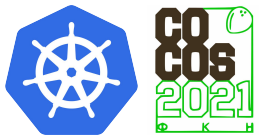
Сильные границы модулей: микросервисы усиливают модульную структуру, что особенно важно для больших команд.

Независимое развертывание: простые сервисы легче разворачивать, и, поскольку они автономны, они с меньшей вероятностью вызовут сбои системы, если что-то пойдет не так.

Разнообразие технологий: с помощью микросервисов вы можете сочетать несколько языков программирования, сред разработки и технологий хранения данных.

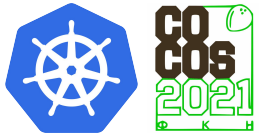
Мартин Фаулер — автор ряда книг и статей по архитектуре ПО.

Мотивация



Kubernetes (K8s) - это открытое программное обеспечение для автоматизации развёртывания, масштабирования и управления контейнеризированными приложениями.

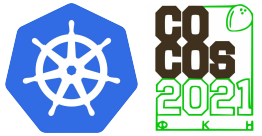
Мотивация



Kubernetes (K8s) - это открытое программное обеспечение для автоматизации развёртывания, масштабирования и управления контейнеризированными приложениями.

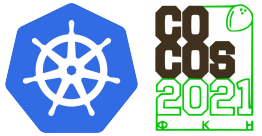
Kubernetes поддерживает модульную структуру, обладает инструментами для независимого развертывания и поддержки микросервисов, работает с контейнеризированными приложениями.

Цель



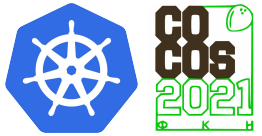
Исследовать архитектуру оркестратора контейнеров Kubernetes и экспериментально рассмотреть процессы развертывания, горизонтального и вертикального масштабирования микросервисов.

Задачи



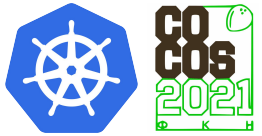
- исследовать архитектуру кластера со стороны DevOps специалиста
- исследовать абстракции Kubernetes со стороны разработчика
- экспериментально исследовать процесс развертывания приложения, горизонтального и вертикального масштабирования в Kubernetes
- провести связь между архитектурой и используемыми абстракциями в Kubernetes

Задачи



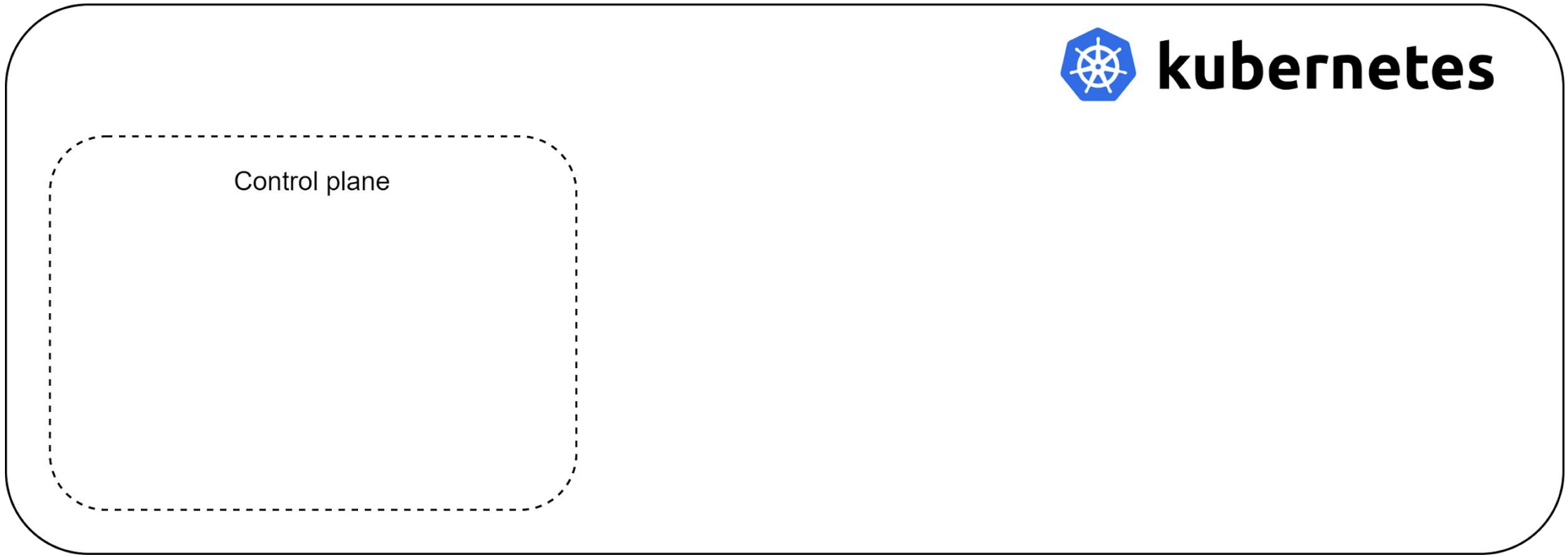
- исследовать архитектуру кластера со стороны DevOps специалиста
- исследовать абстракции Kubernetes со стороны разработчика
- экспериментально исследовать процесс развертывания приложения, горизонтального и вертикального масштабирования в Kubernetes
- **провести связь между архитектурой и используемыми абстракциями в Kubernetes**

Архитектура Kubernetes

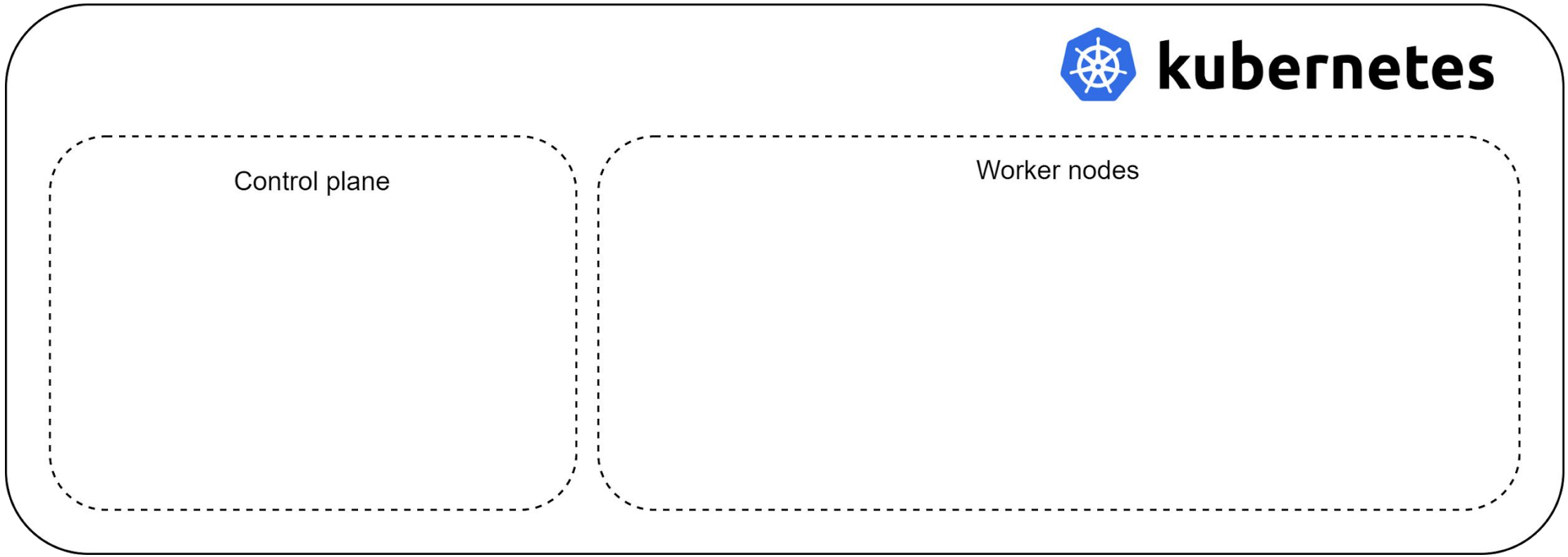


Архитектура Kubernetes

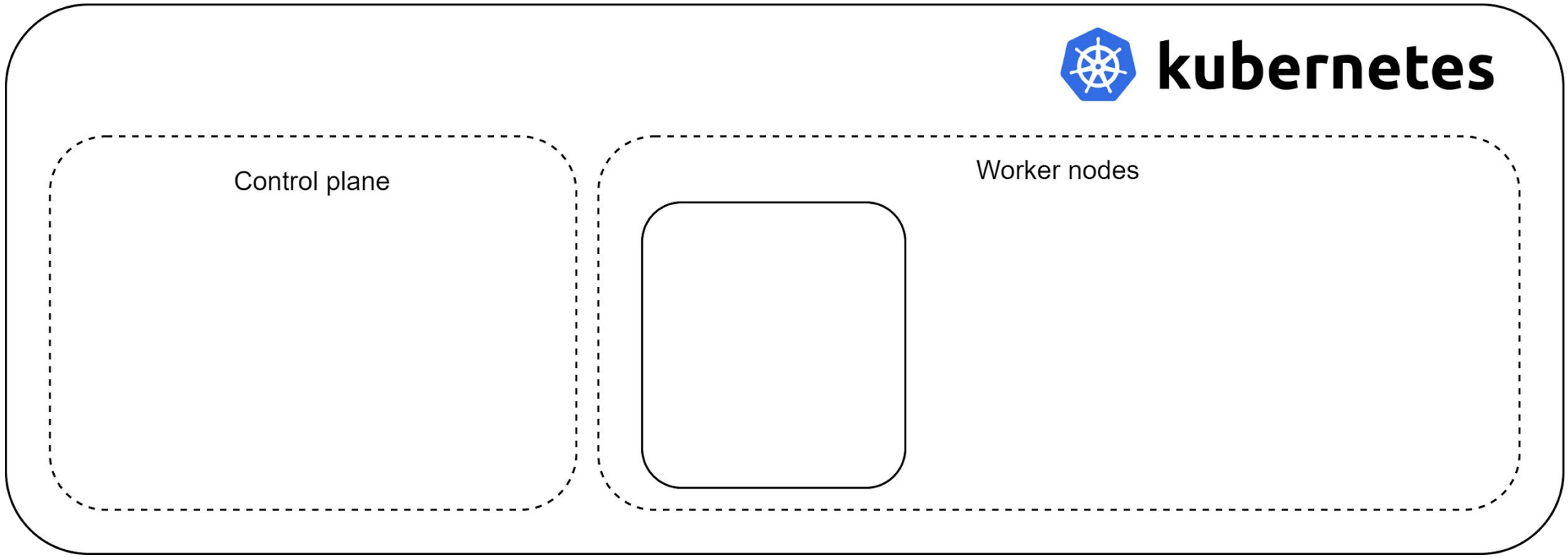
Архитектура Kubernetes



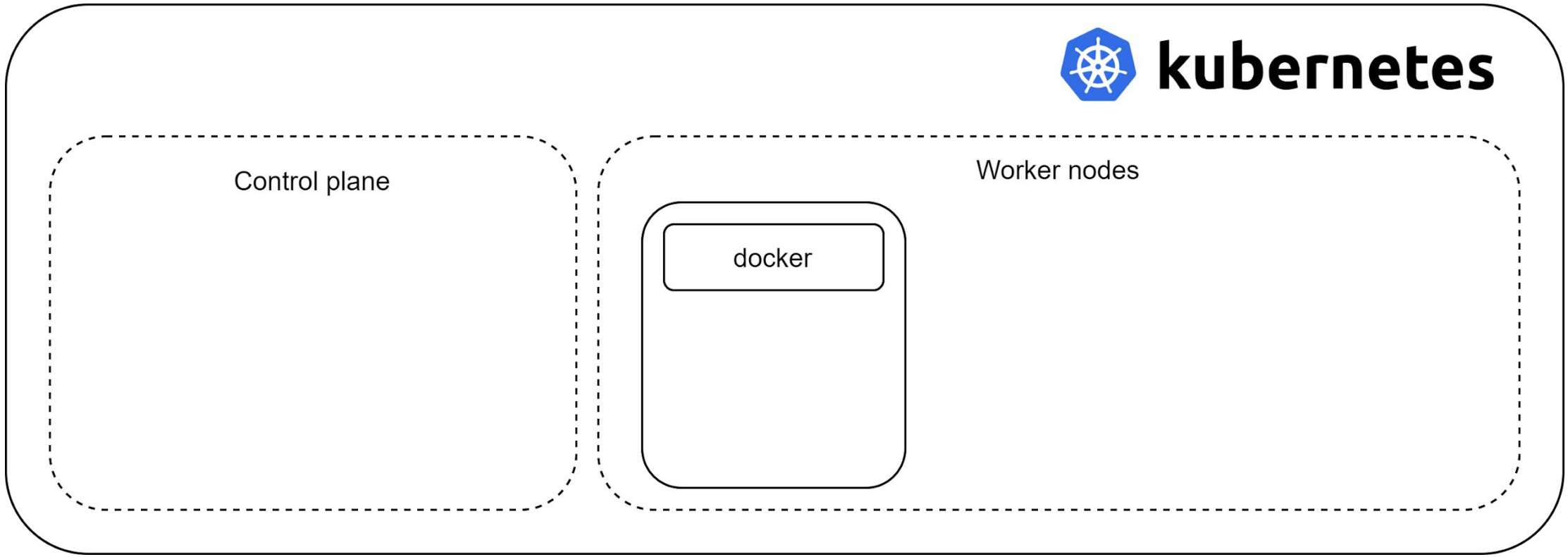
Архитектура Kubernetes



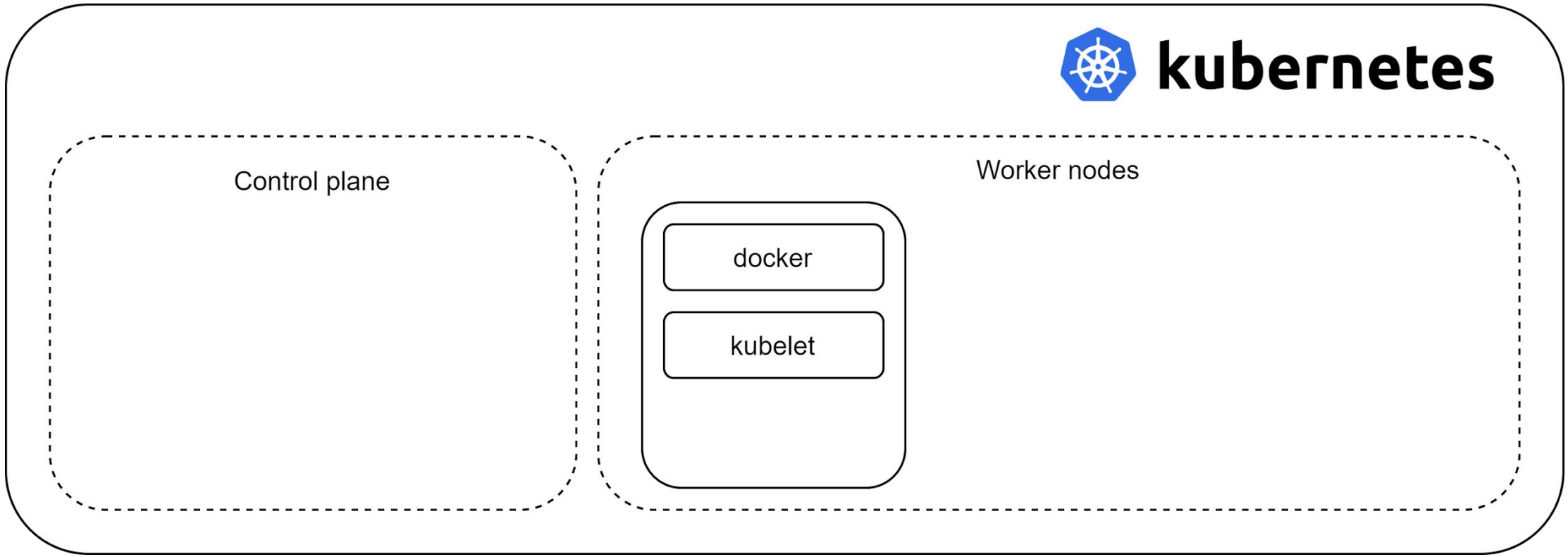
Архитектура Kubernetes



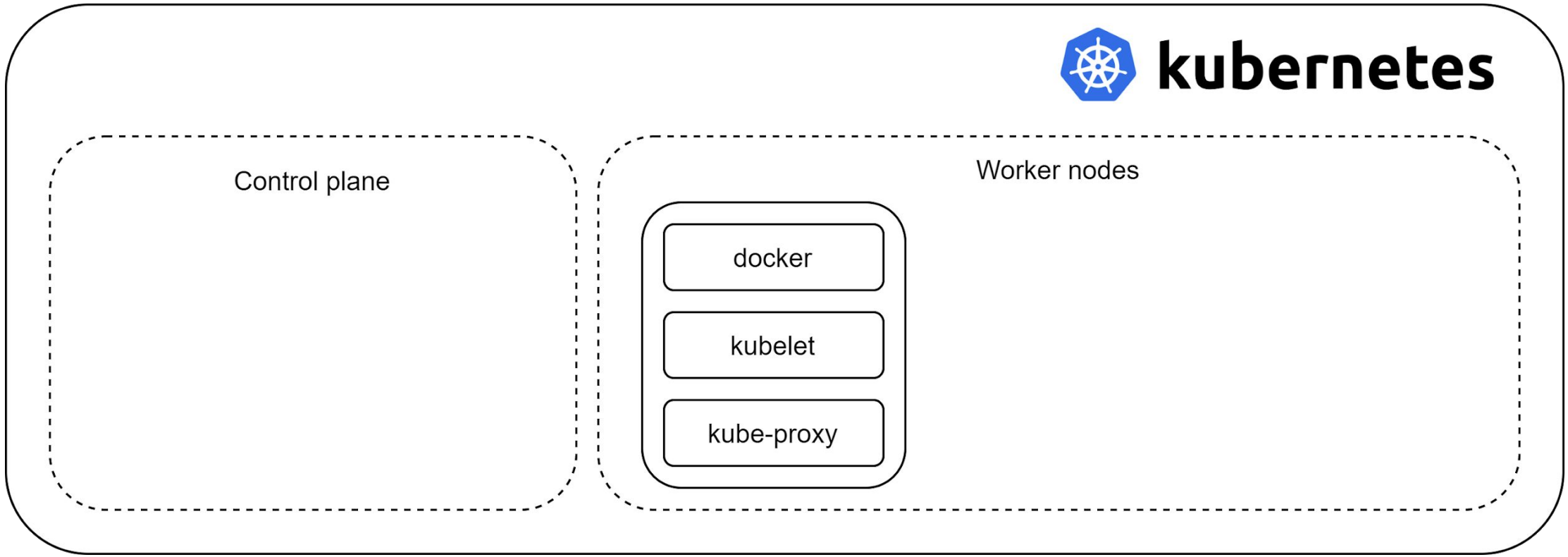
Архитектура Kubernetes



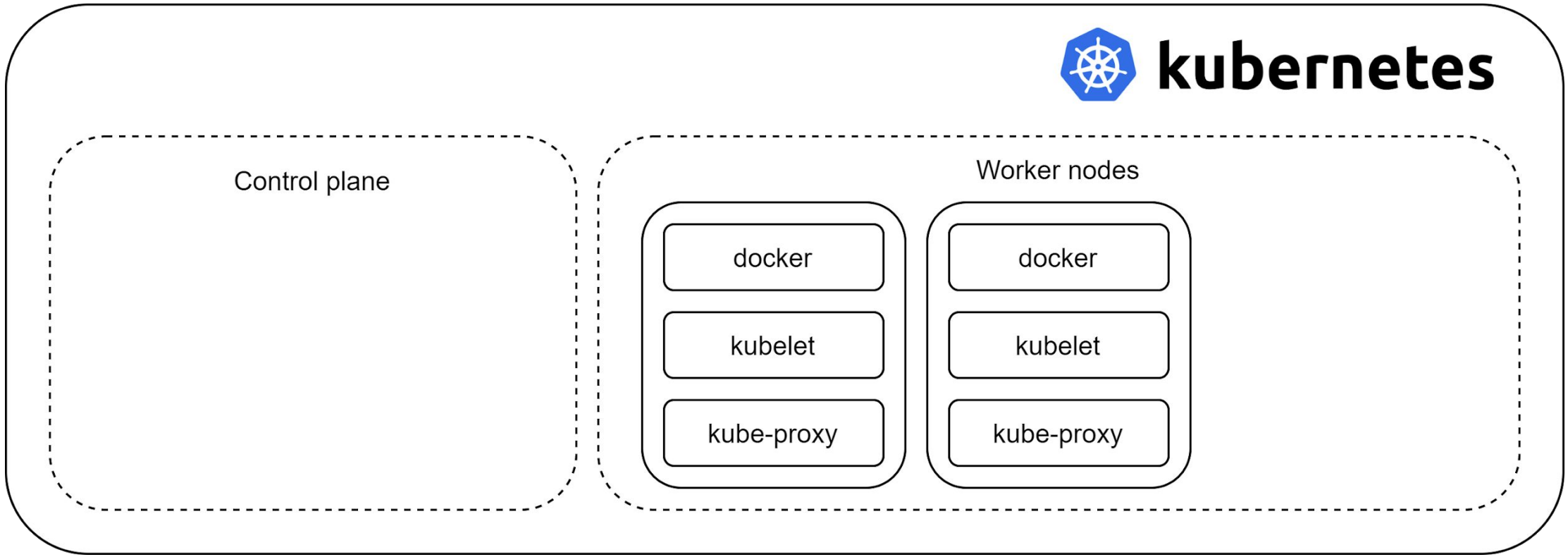
Архитектура Kubernetes



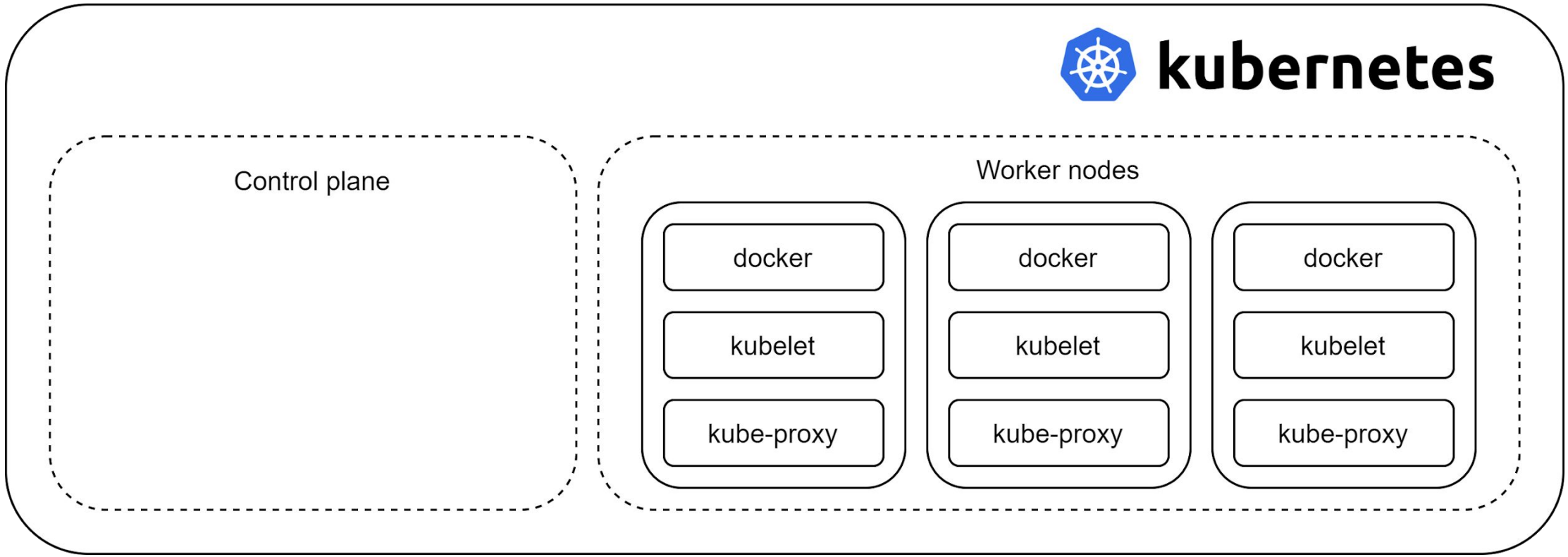
Архитектура Kubernetes



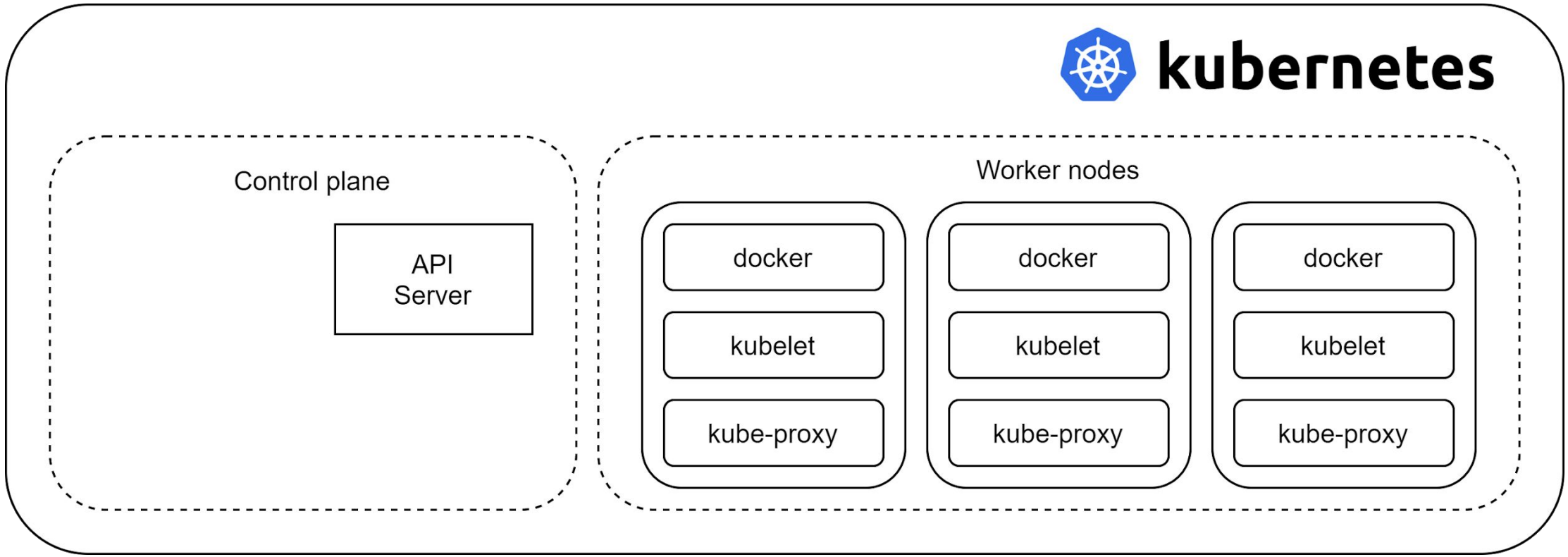
Архитектура Kubernetes



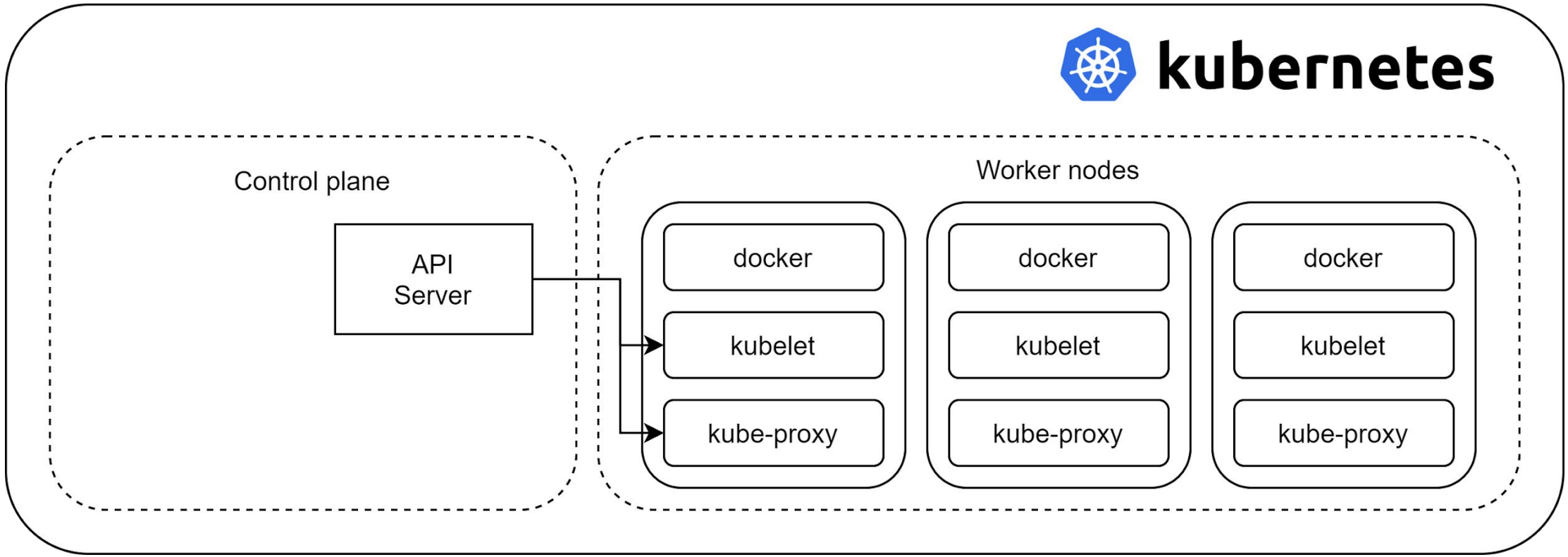
Архитектура Kubernetes



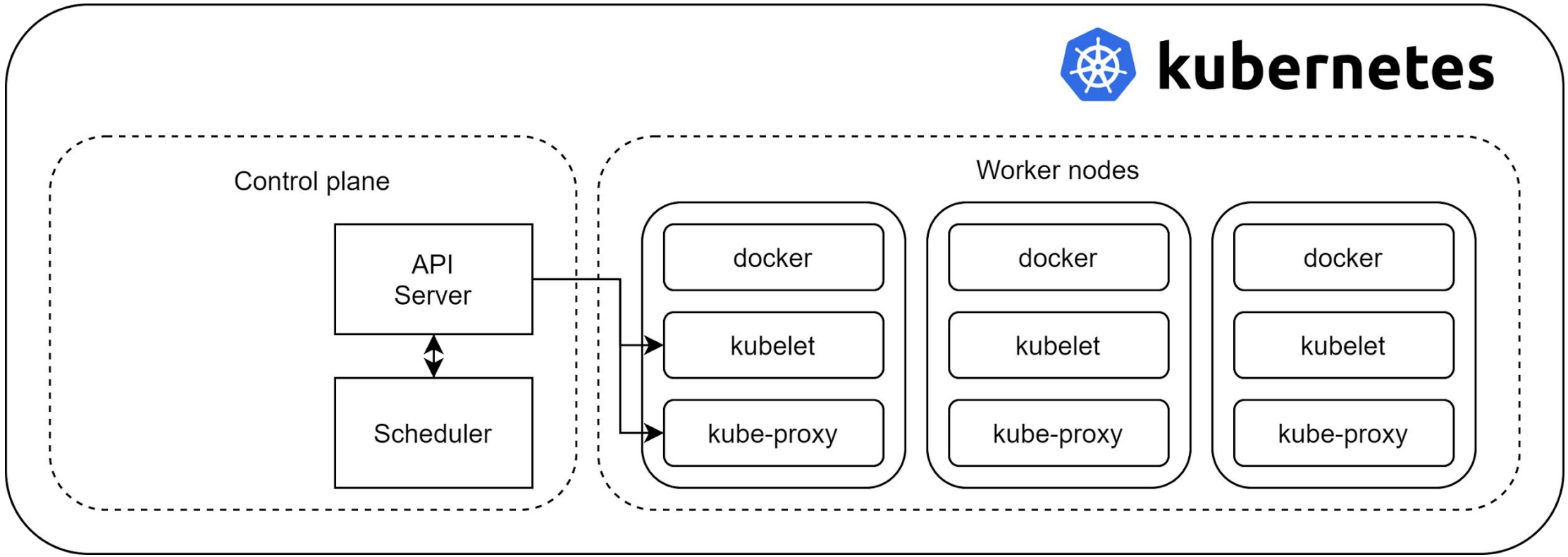
Архитектура Kubernetes



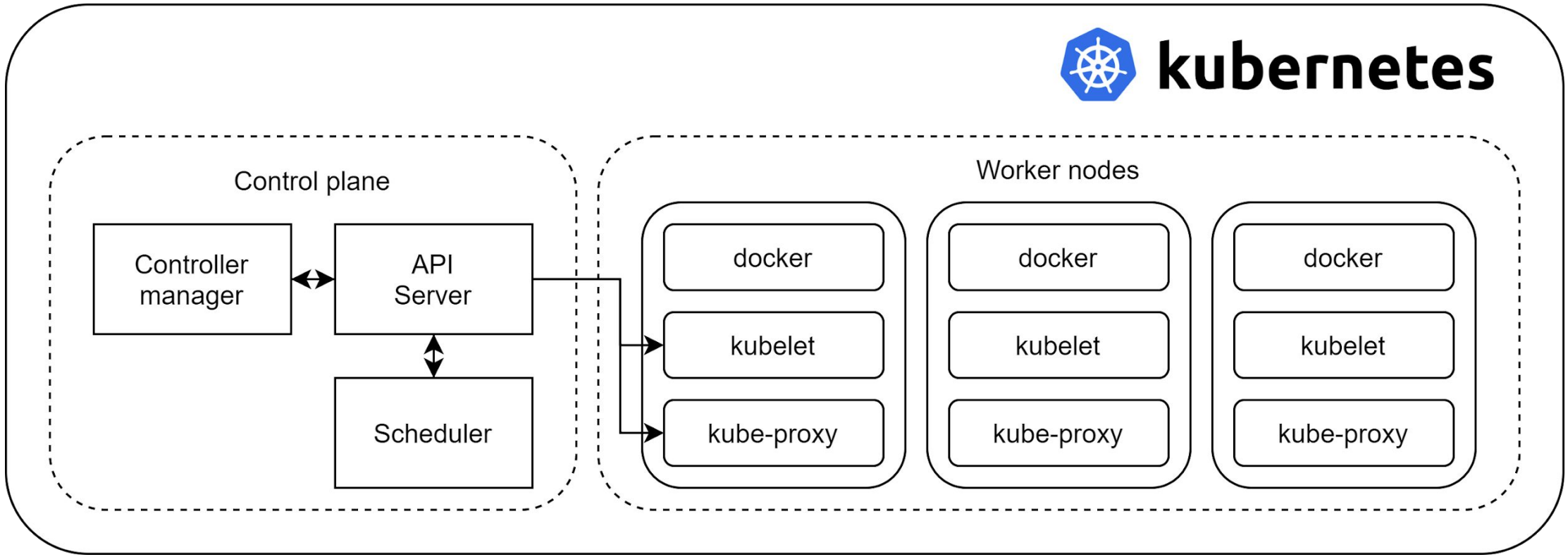
Архитектура Kubernetes



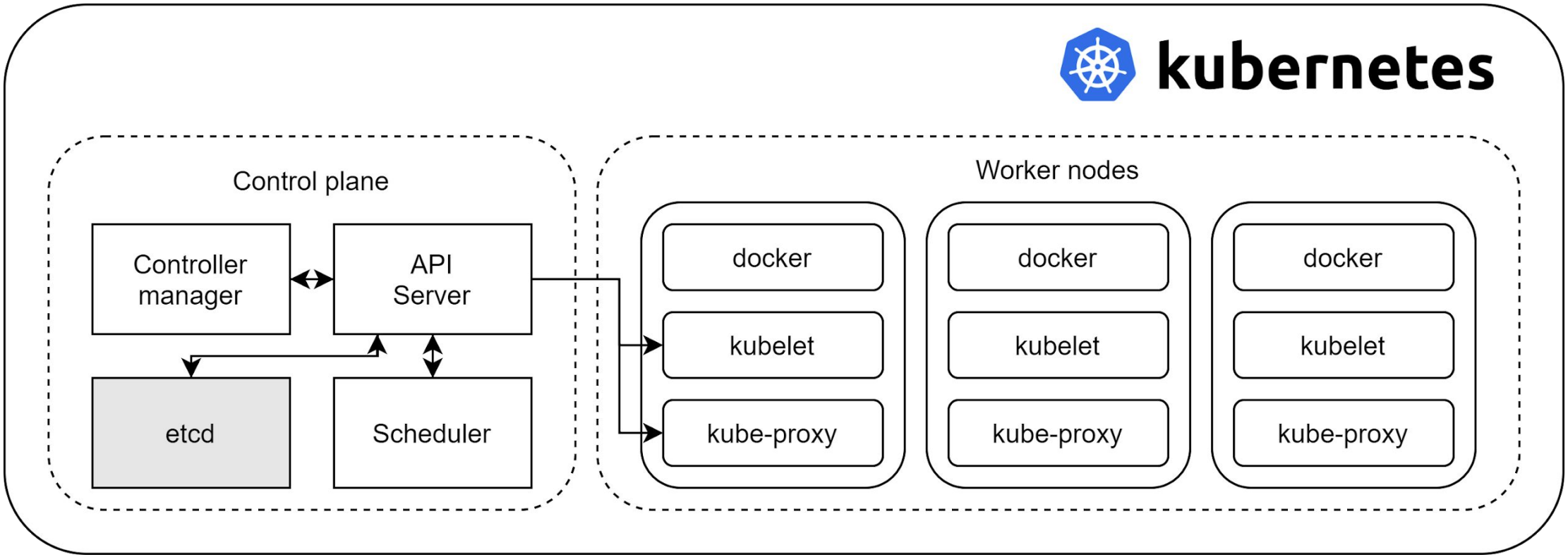
Архитектура Kubernetes



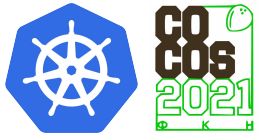
Архитектура Kubernetes



Архитектура Kubernetes



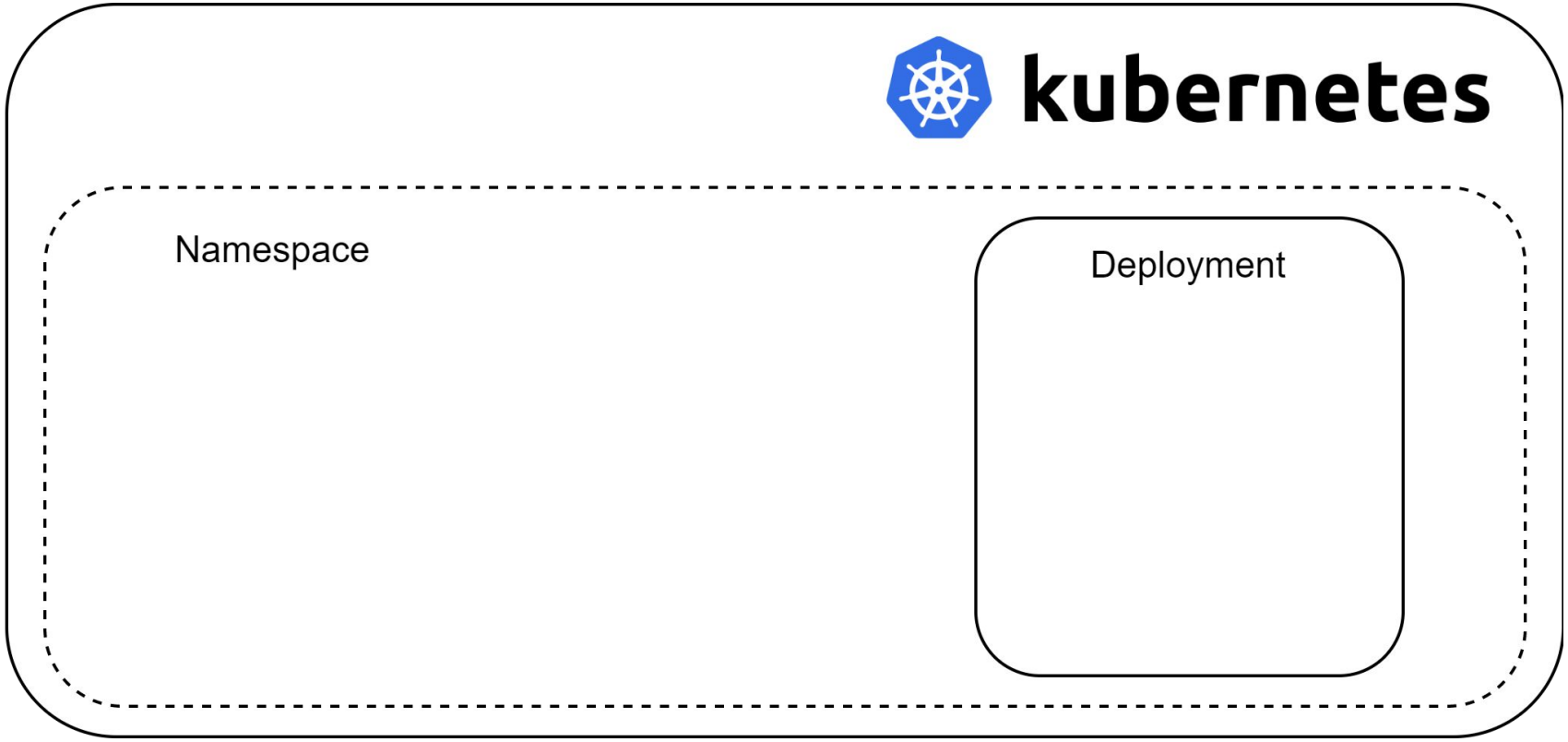
Объекты Kubernetes



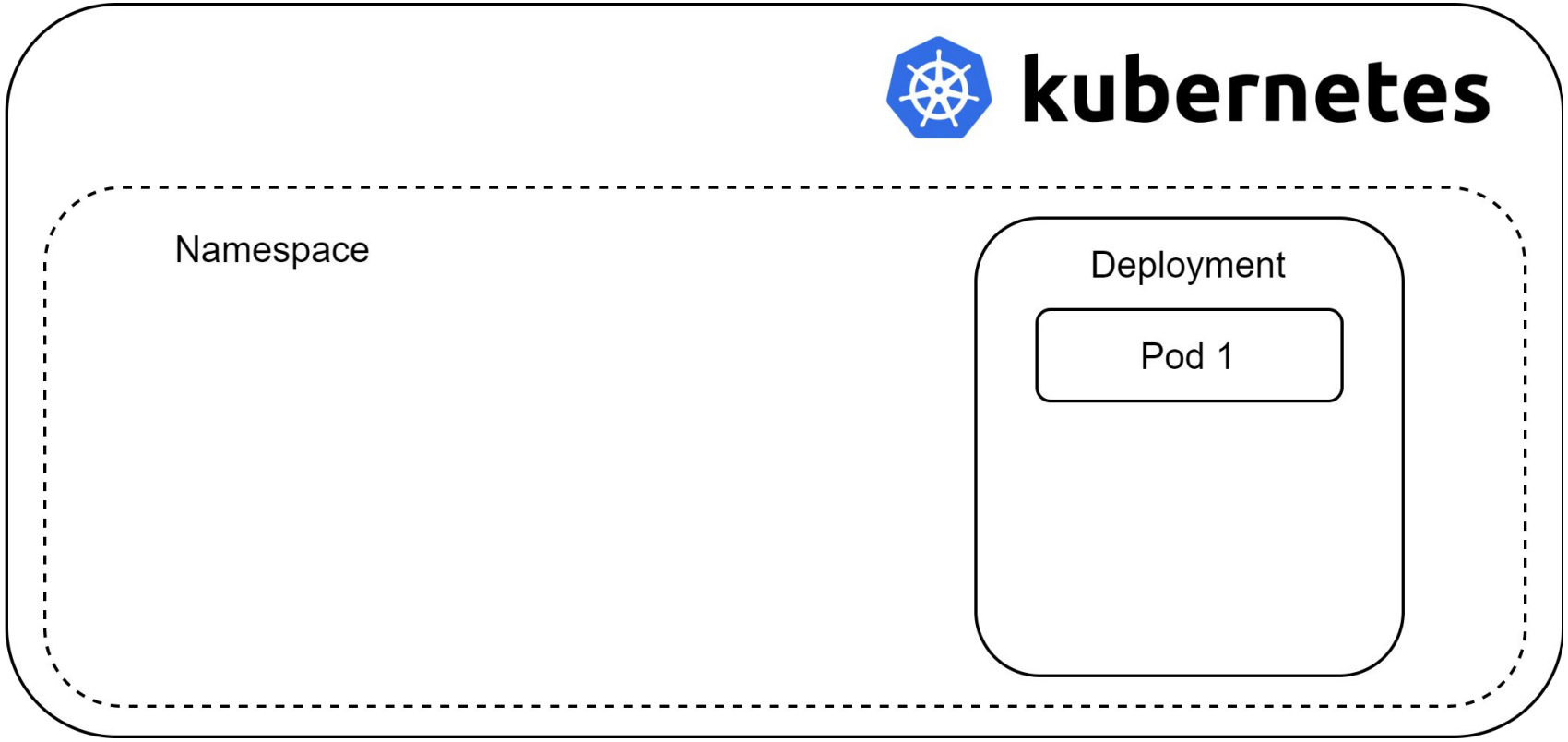
Объекты Kubernetes

Объекты Kubernetes

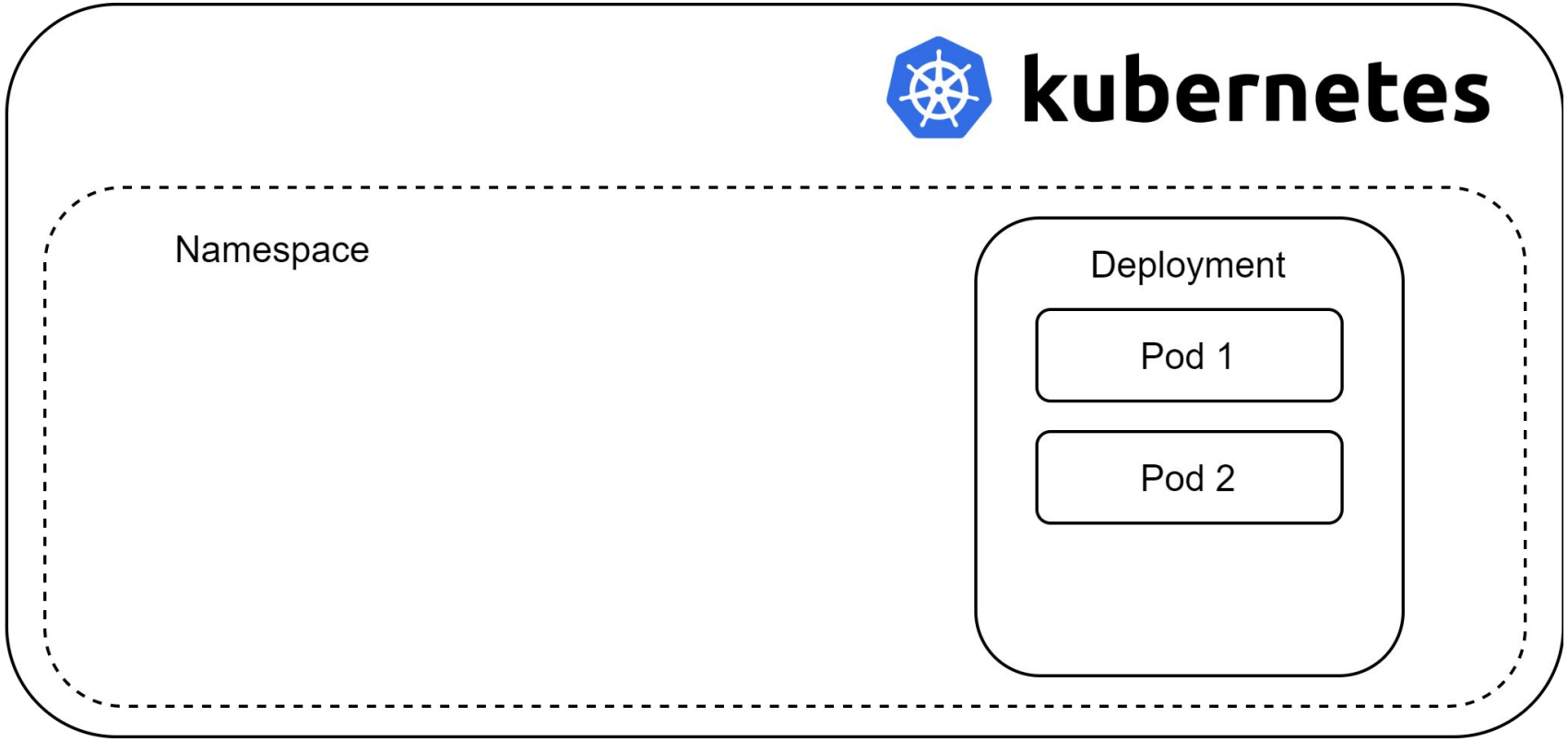
Объекты Kubernetes



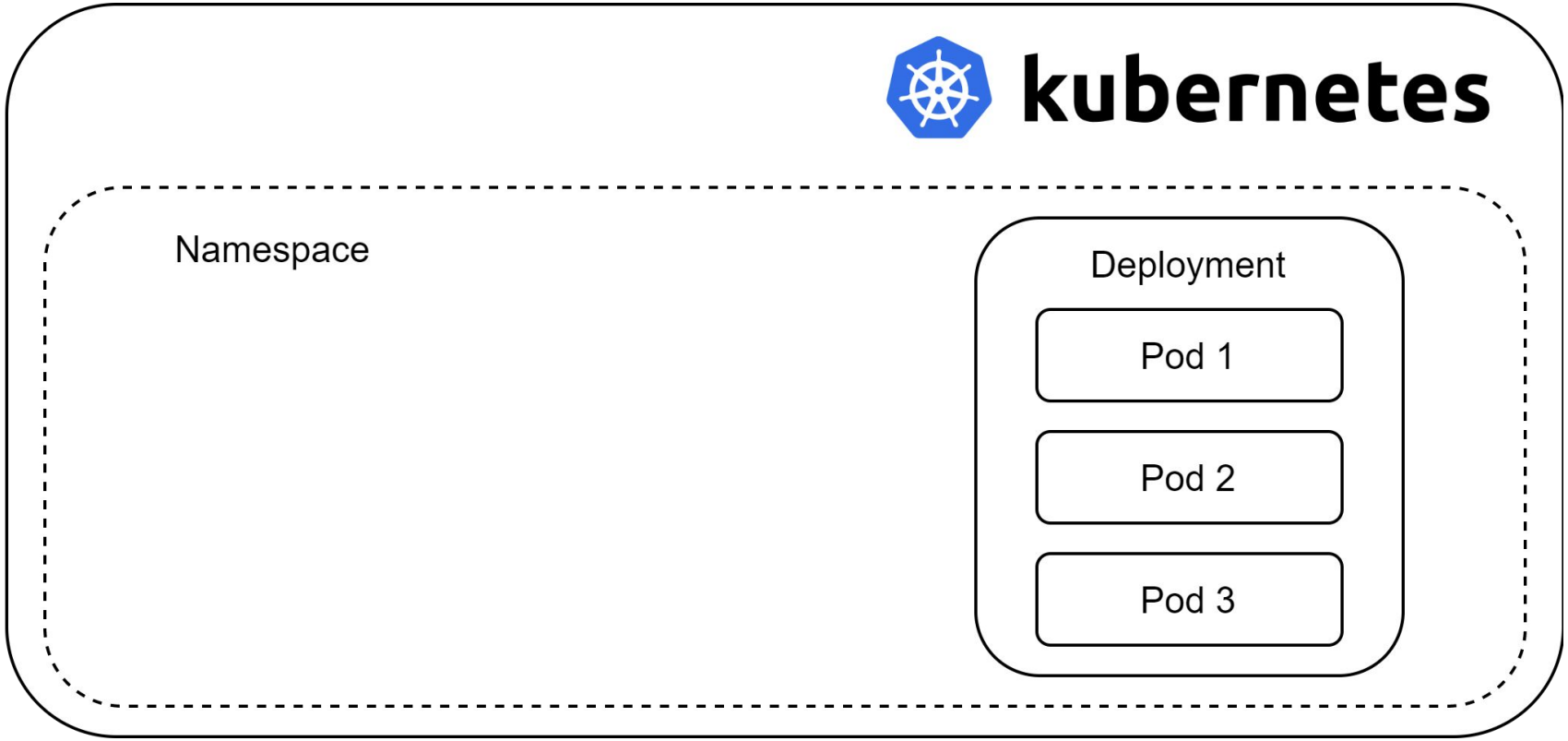
Объекты Kubernetes



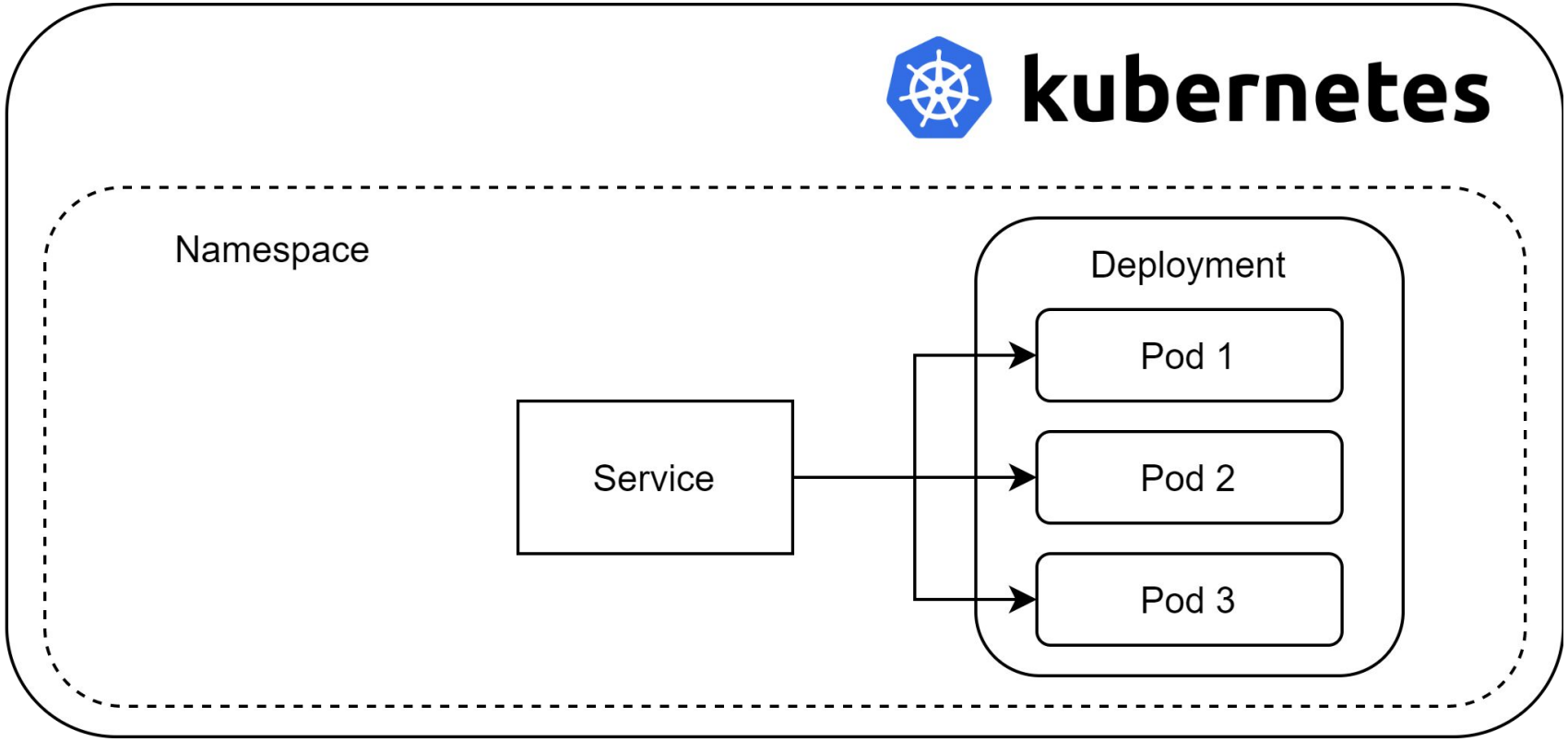
Объекты Kubernetes



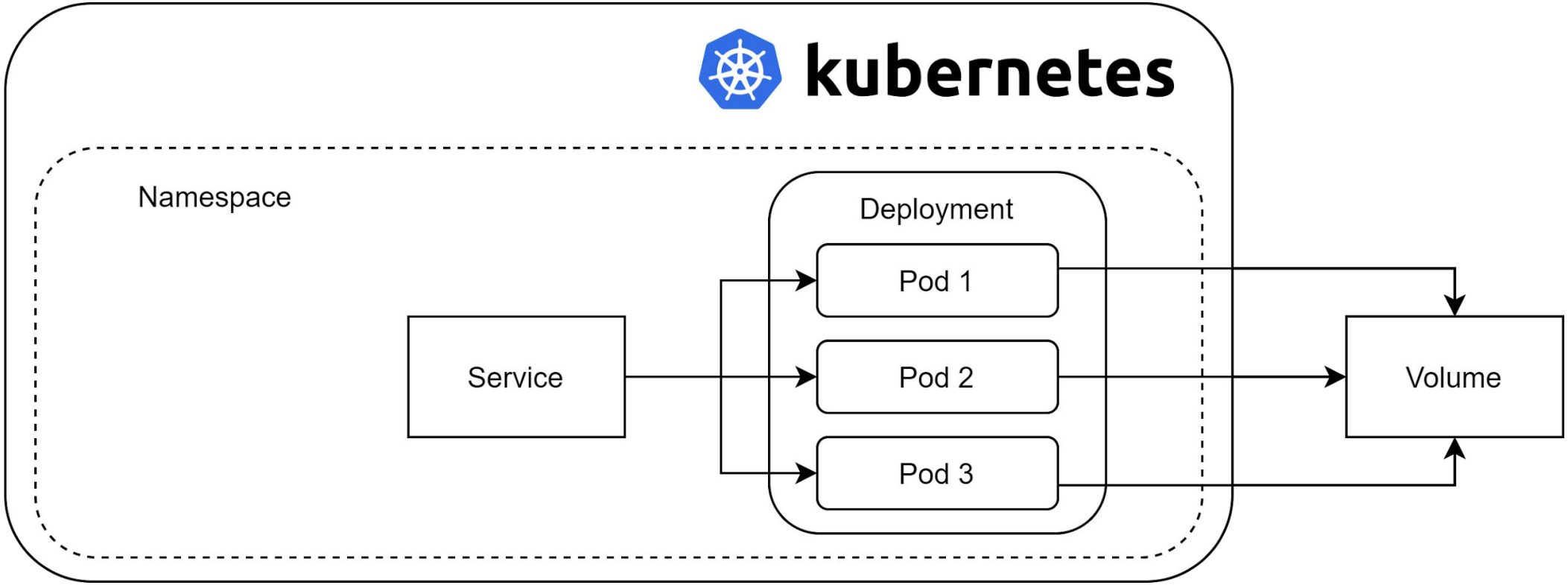
Объекты Kubernetes



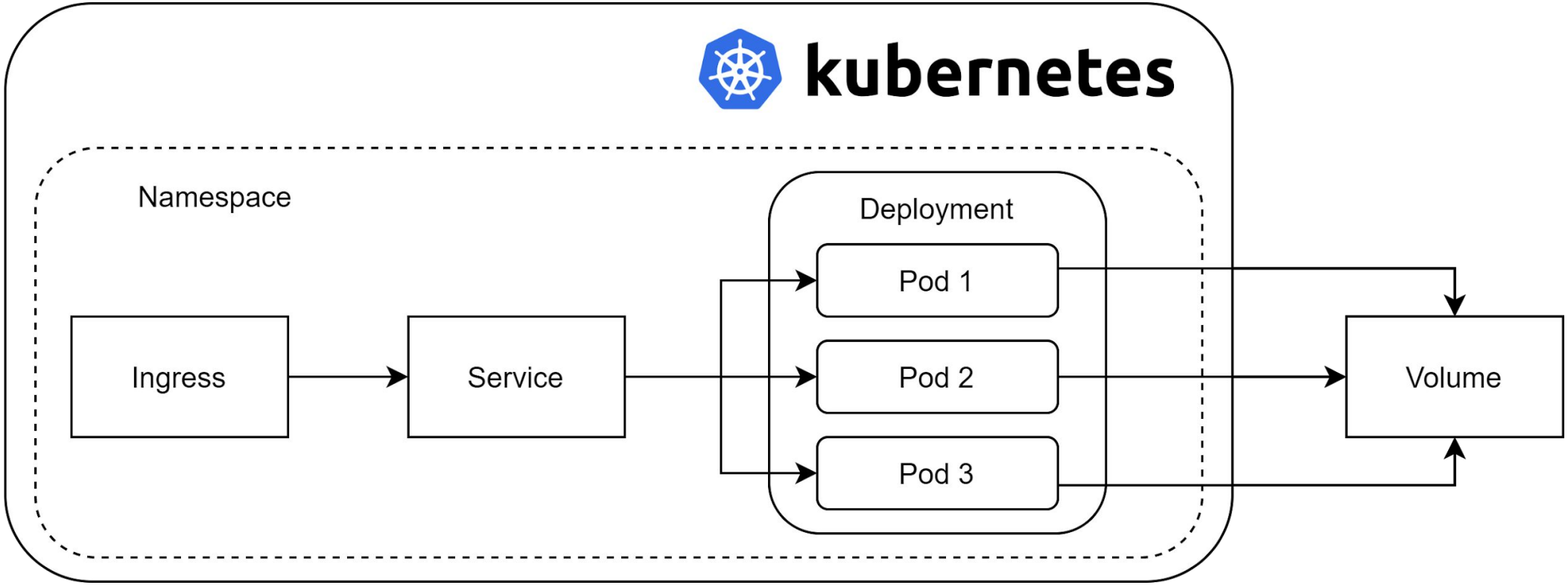
Объекты Kubernetes



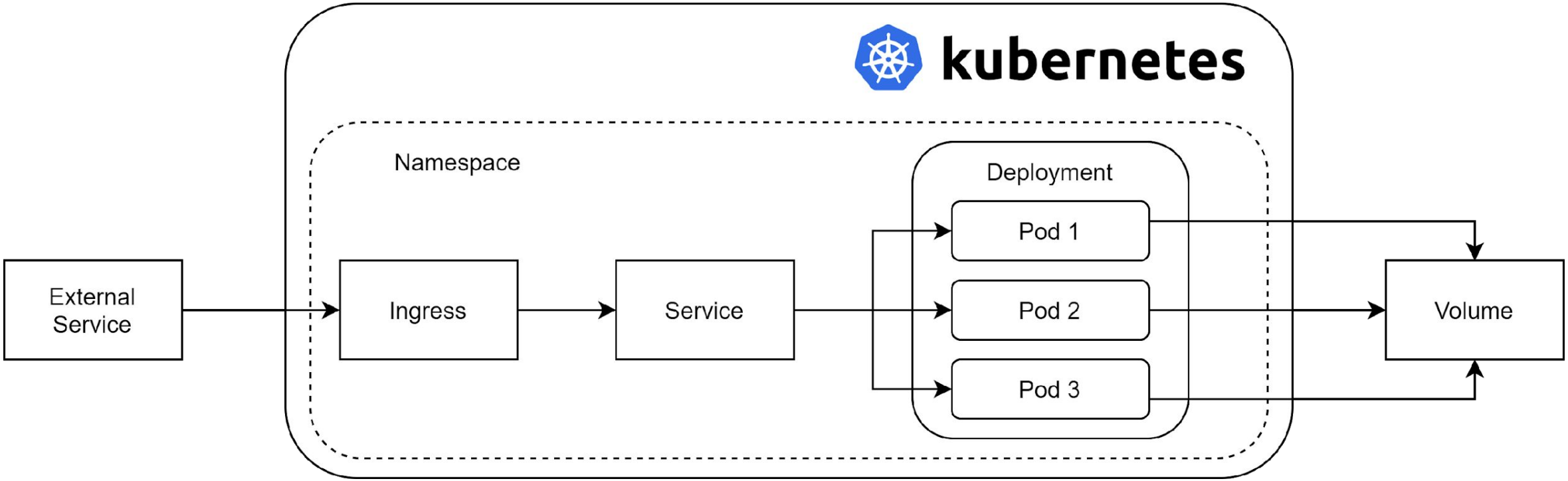
Объекты Kubernetes



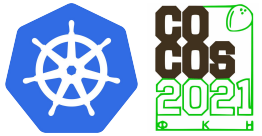
Объекты Kubernetes



Объекты Kubernetes



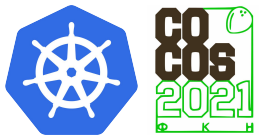
Тестирование Kubernetes



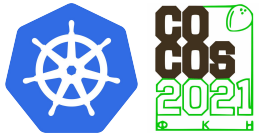
В рамках тестирования была развернута следующая инфраструктура:

- кластер Kubernetes
- сервис для сбора метрик Prometheus
- сервис для отображения метрик Grafana

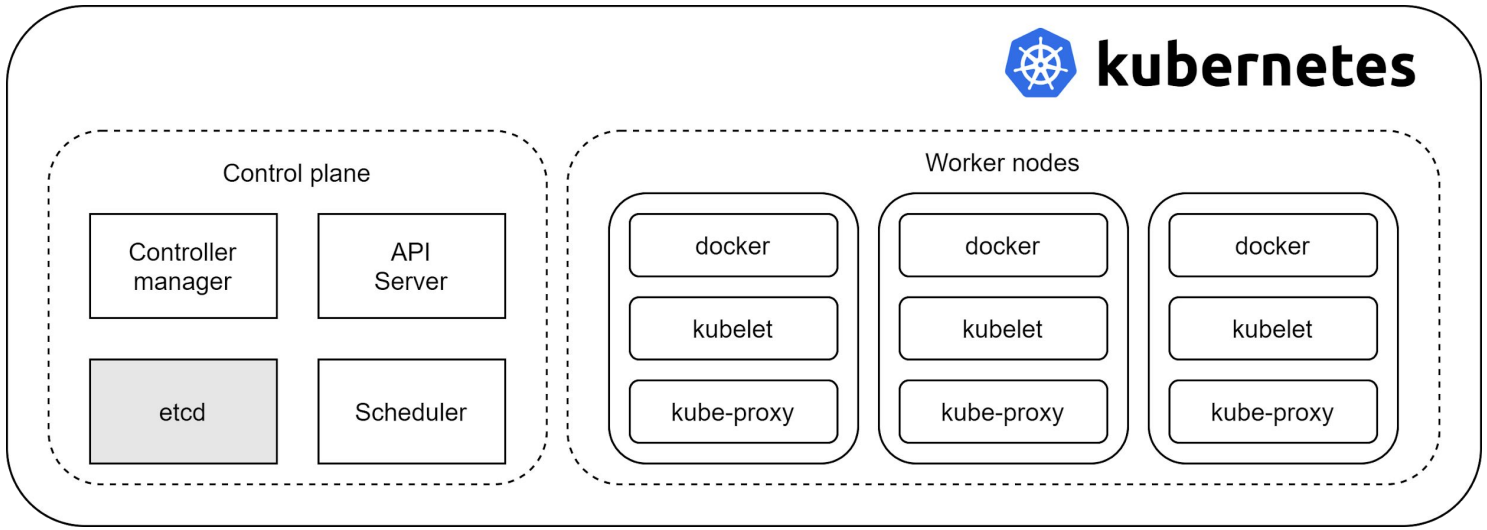
Развертывание

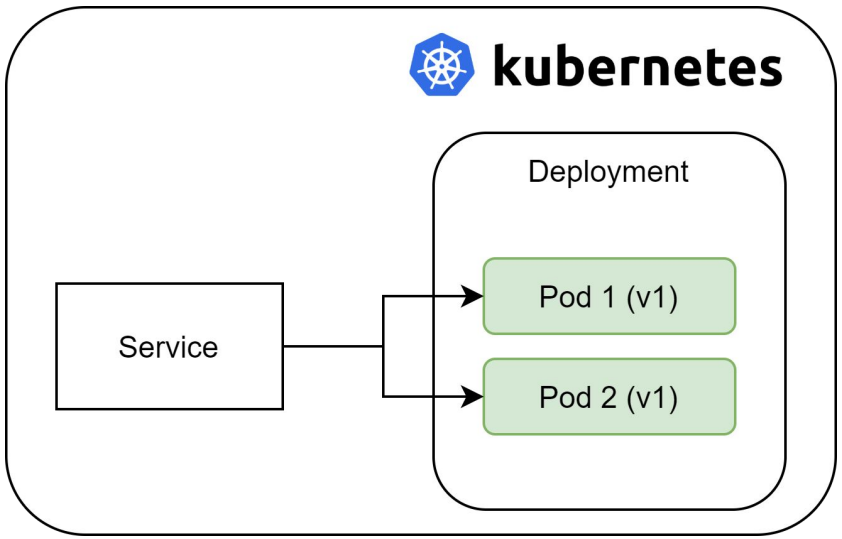


Развертывание

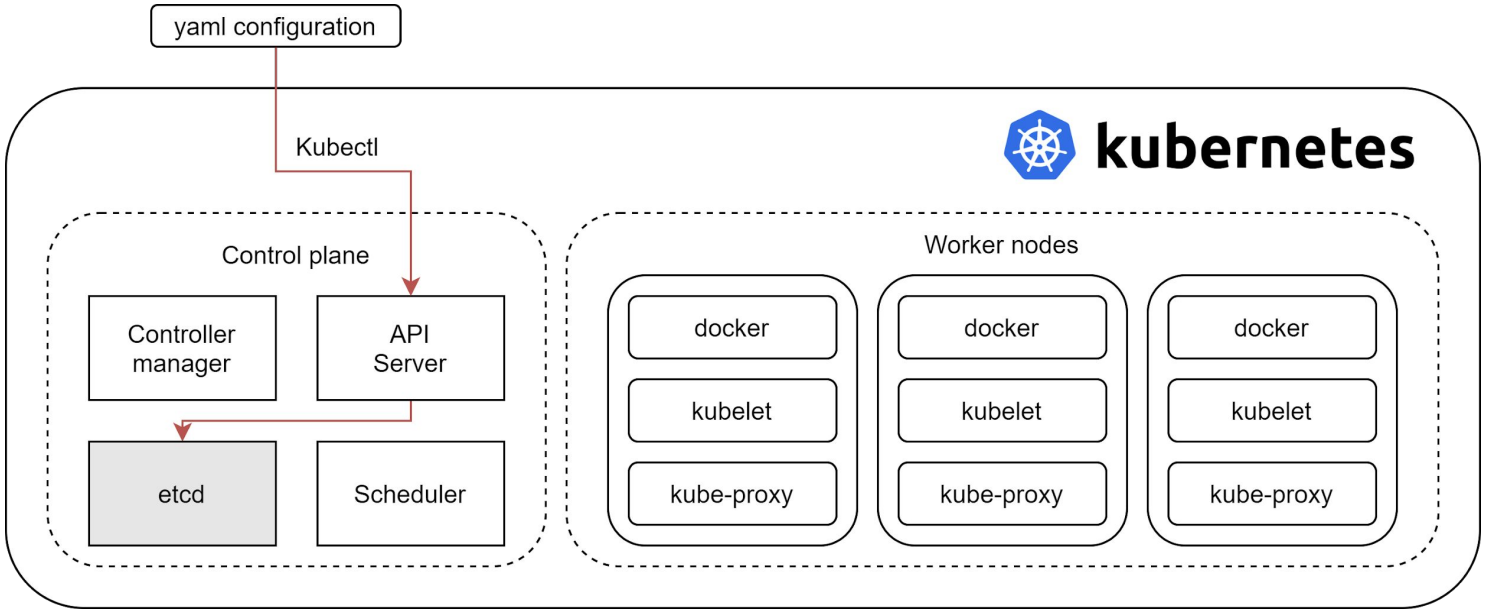


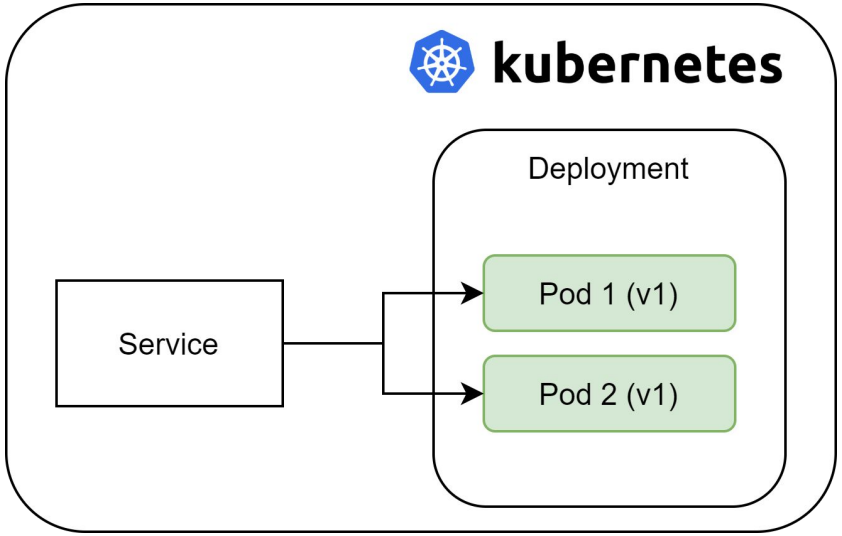
Набор действий, которые выполняет оркестратор с целью сделать новую версию приложения готовой к использованию. От методов развертывания будет зависеть время простоя приложения.


kubernetes
Control plane
Controller manager
API Server
etcd
Scheduler
Worker nodes
docker
kubelet
kube-proxy
docker
kubelet
kube-proxy
docker
kubelet
kube-proxy

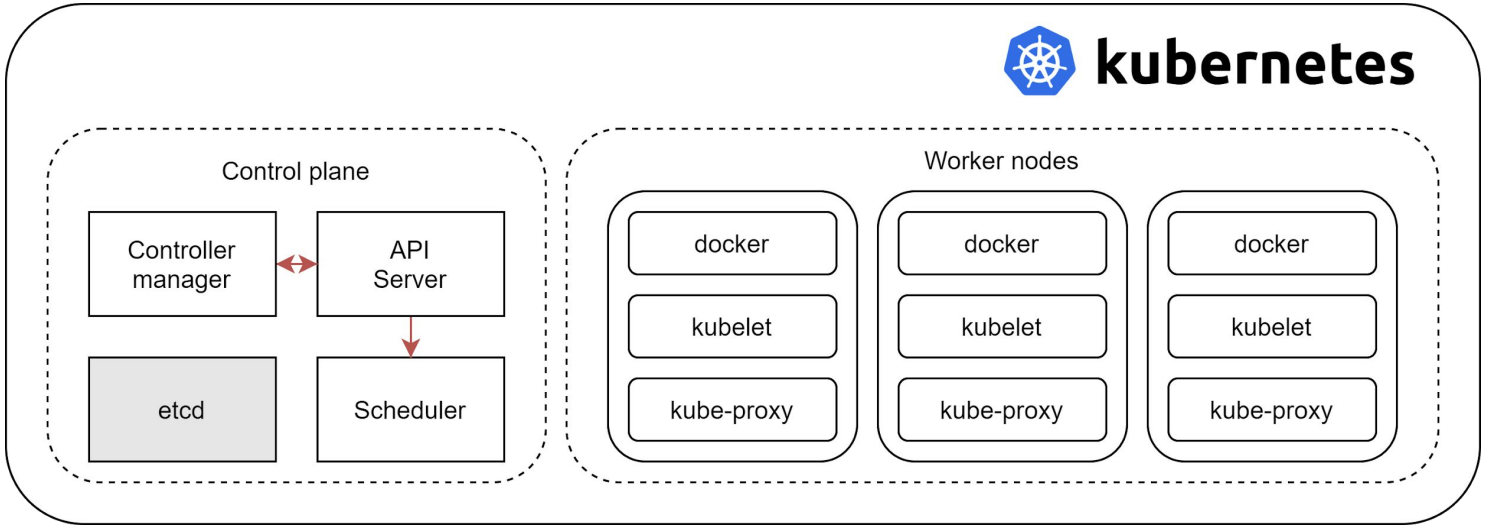

kubernetes
Deployment
Service
Pod 1 (v1)
Pod 2 (v1)

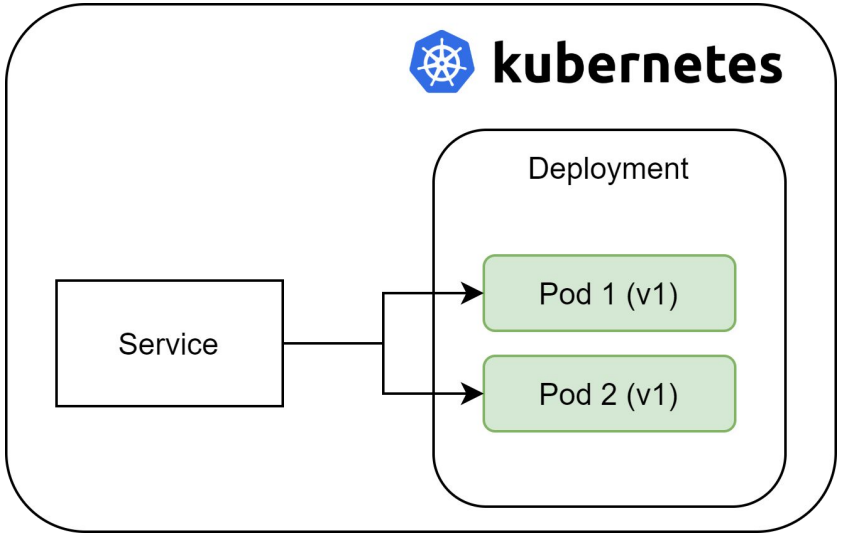
Изменение версии приложения в конфигурации



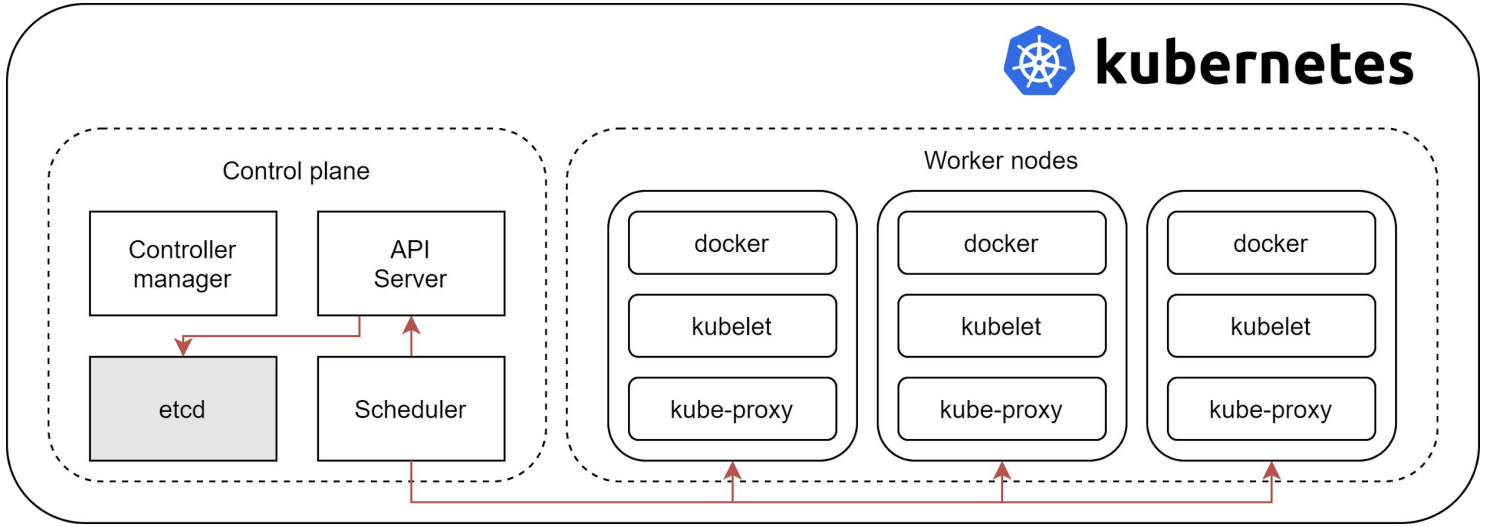


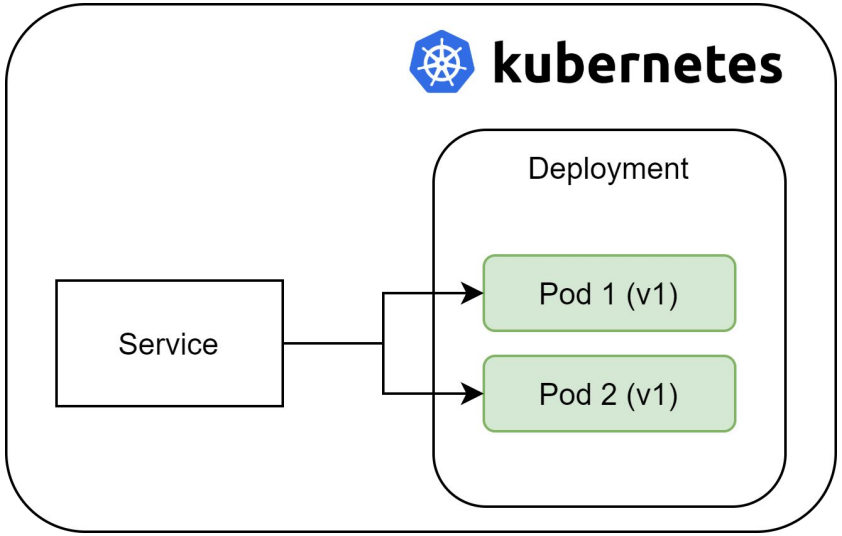
Controller manager замечает изменение в конфигурации



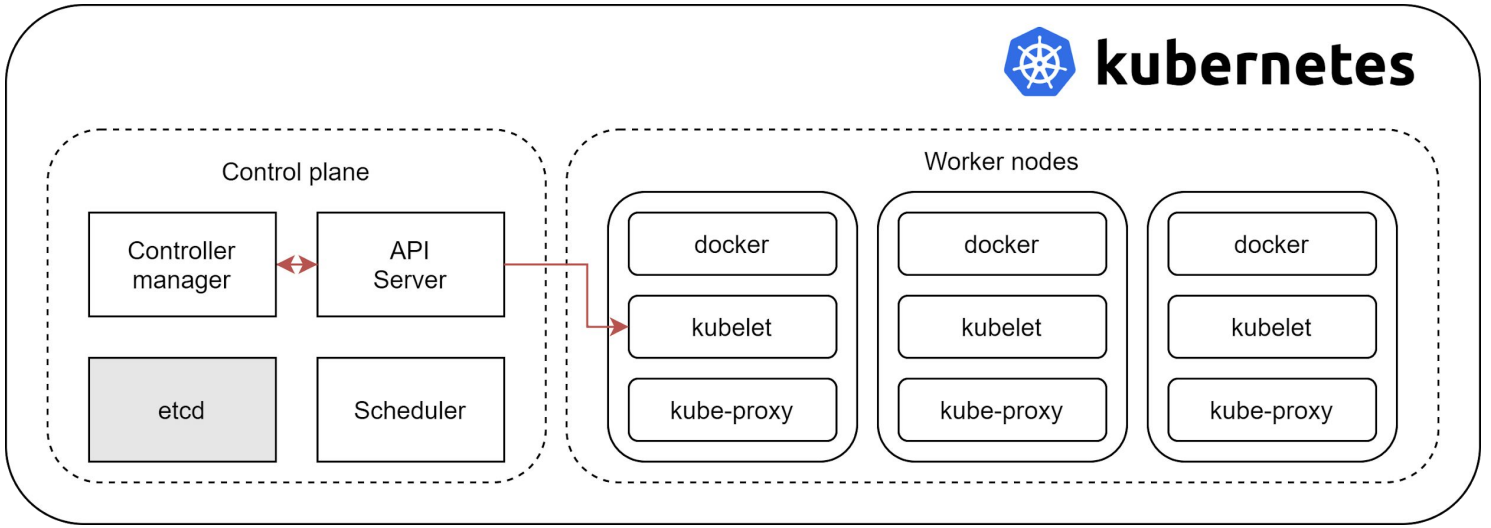


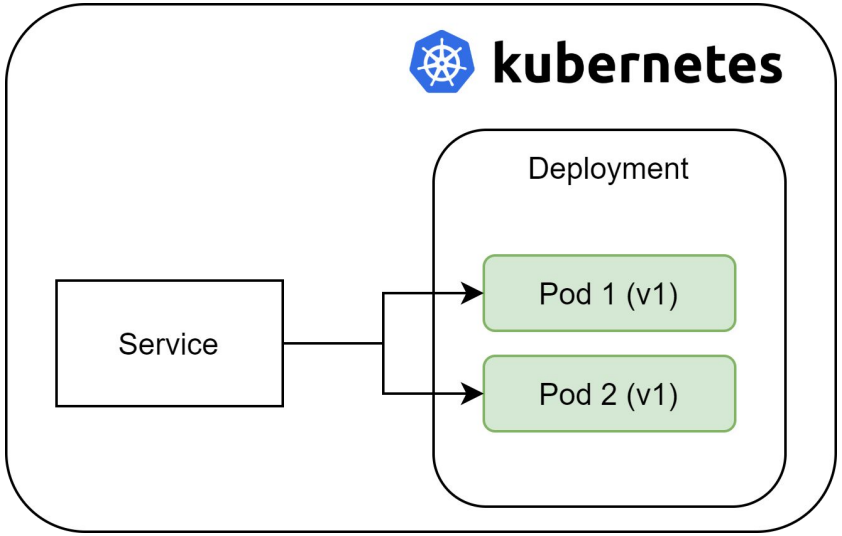
Scheduler принимает решение о том, на каком узле должна быть запущена новая версия приложения



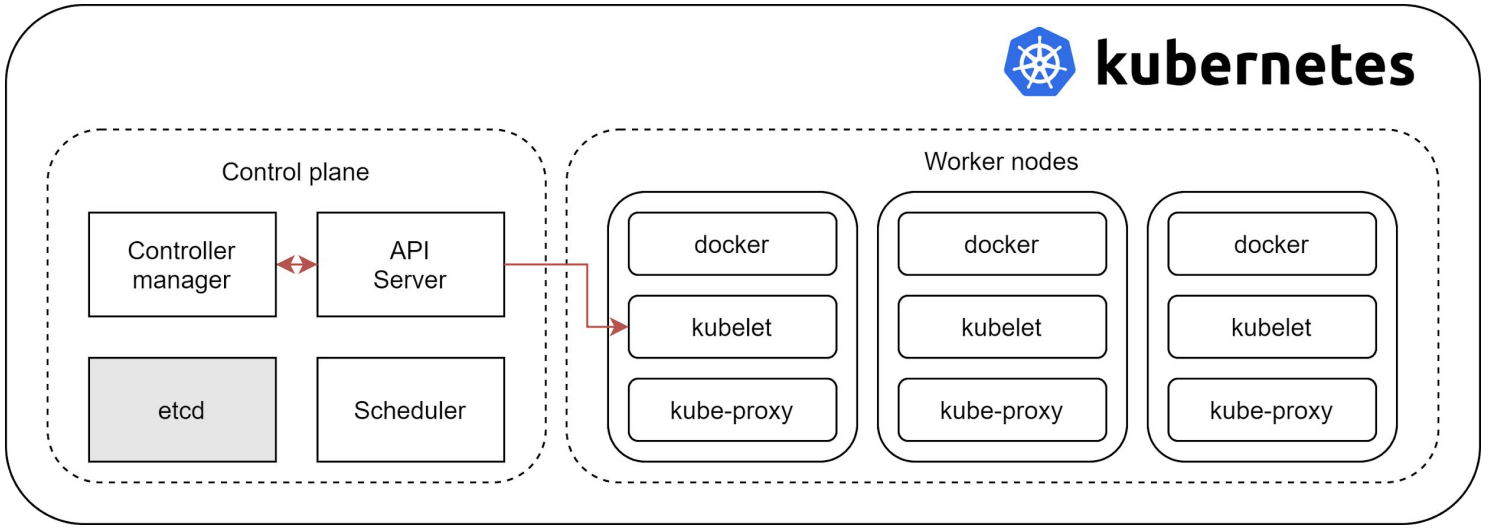


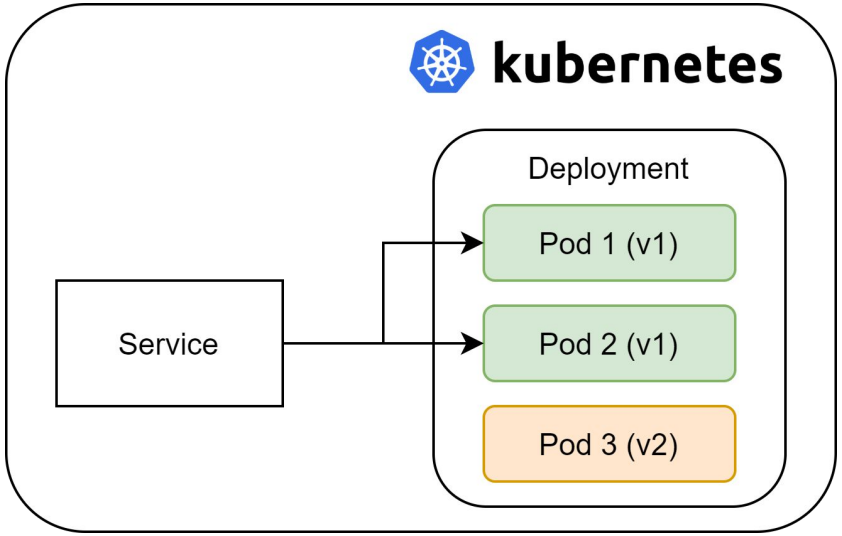
Controller manager через kubelet запускает процесс развертывания



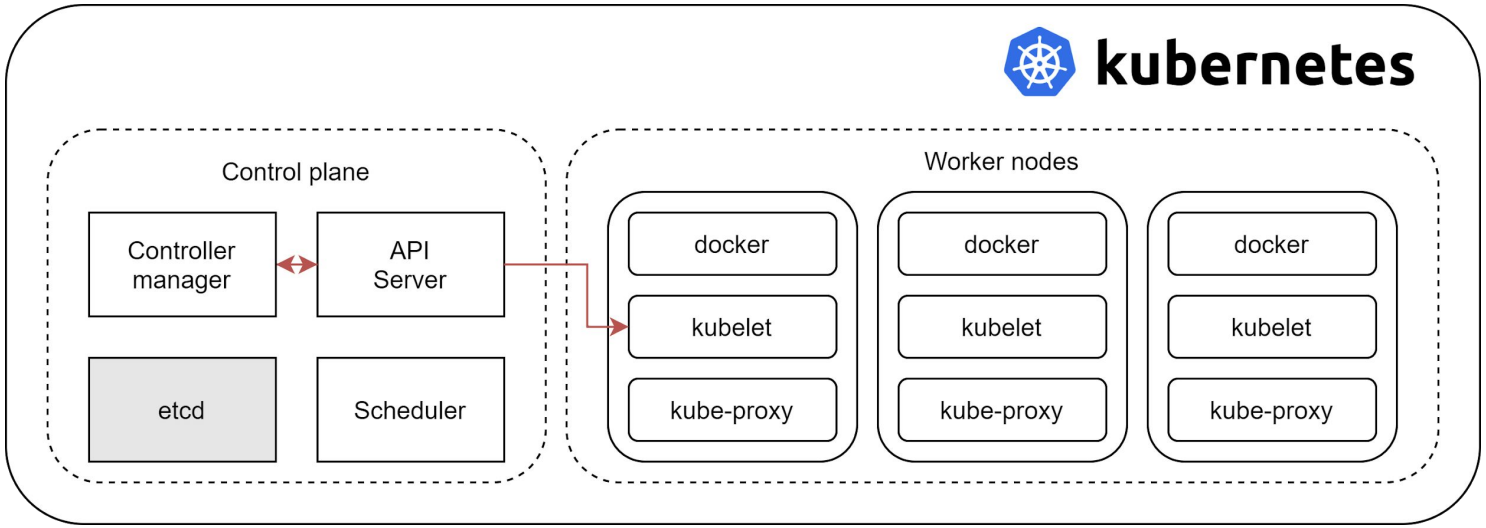


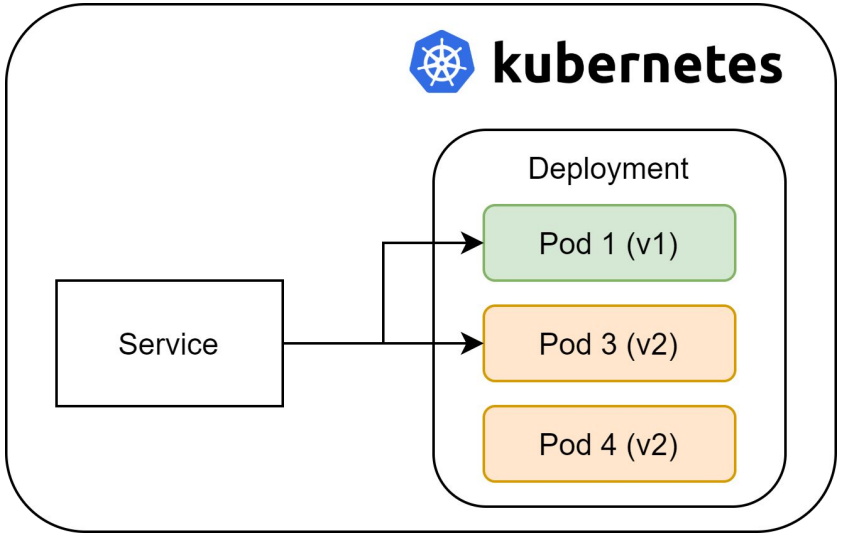
Создается новый под с новой версией приложения



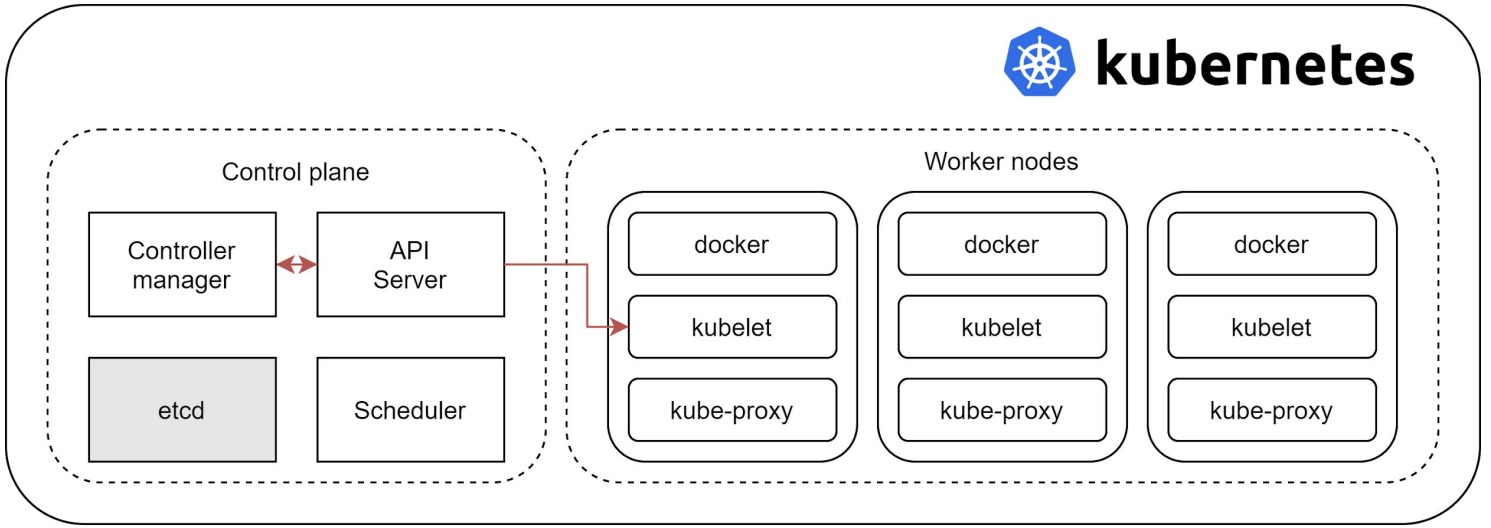


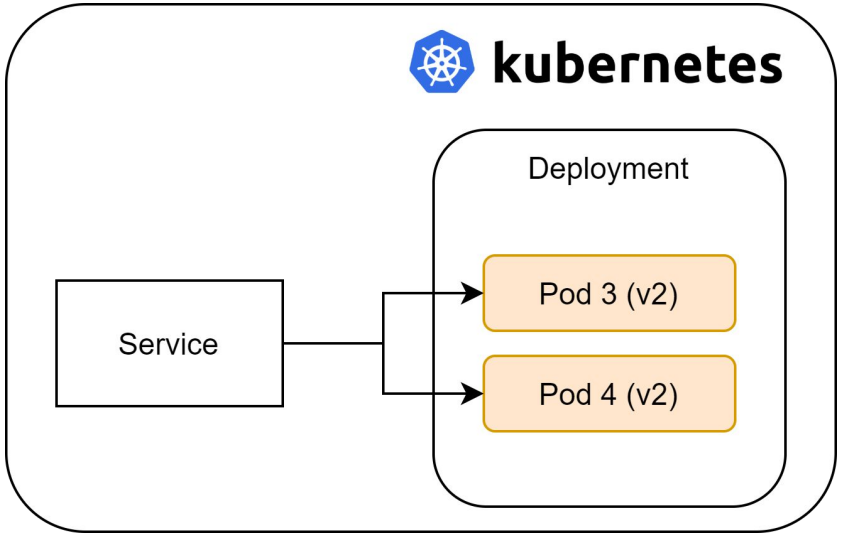
Создается новый под с новой версией приложения, один из подов с предыдущей версией удаляется



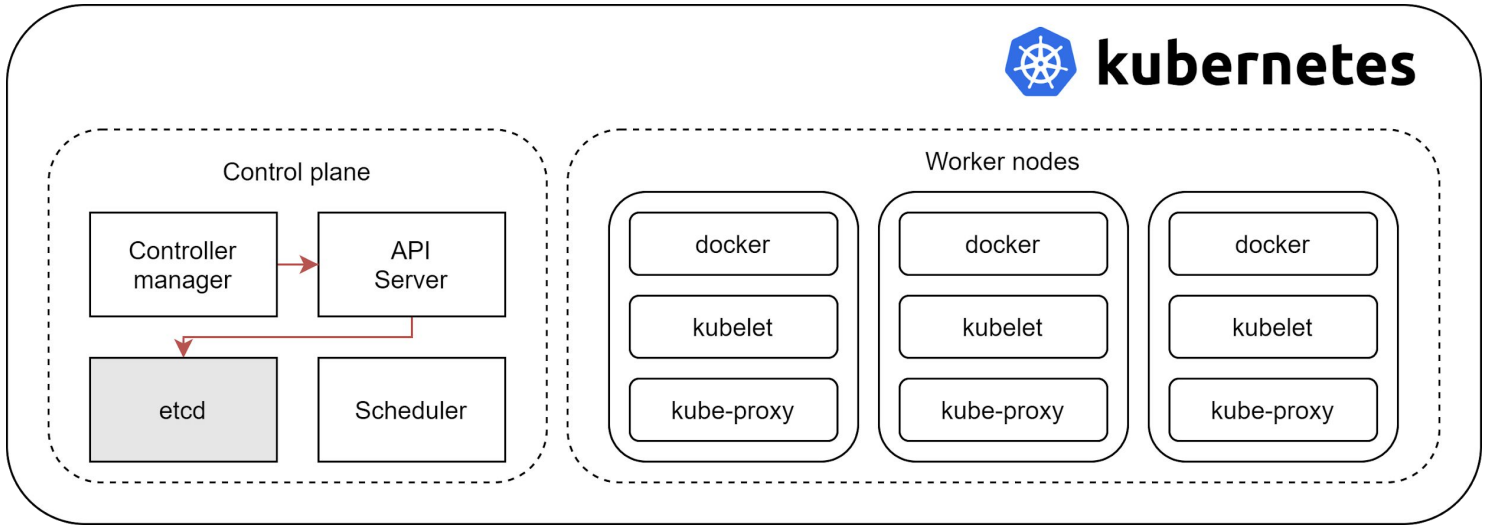


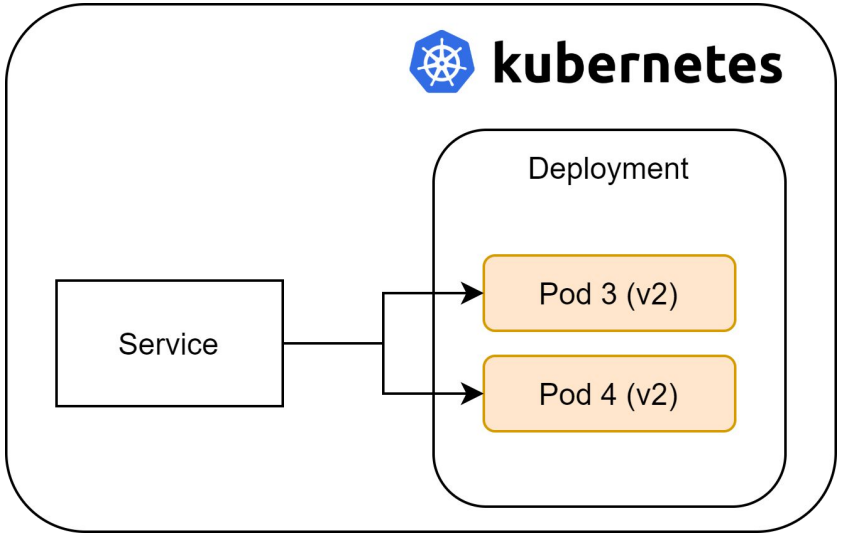
Оба пода прошли проверки и приложение работает с новой версией



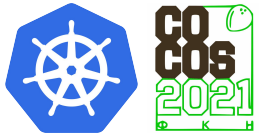


Controller manager сохраняет текущее состояние в etcd

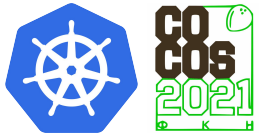




Горизонтальное масштабирование

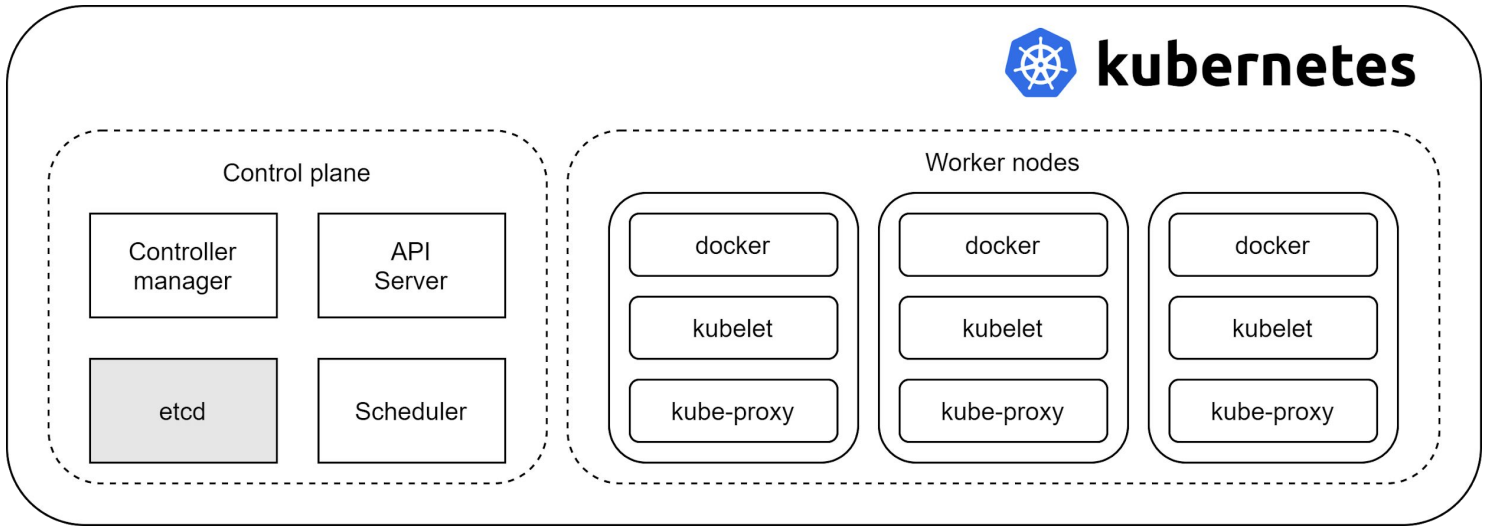


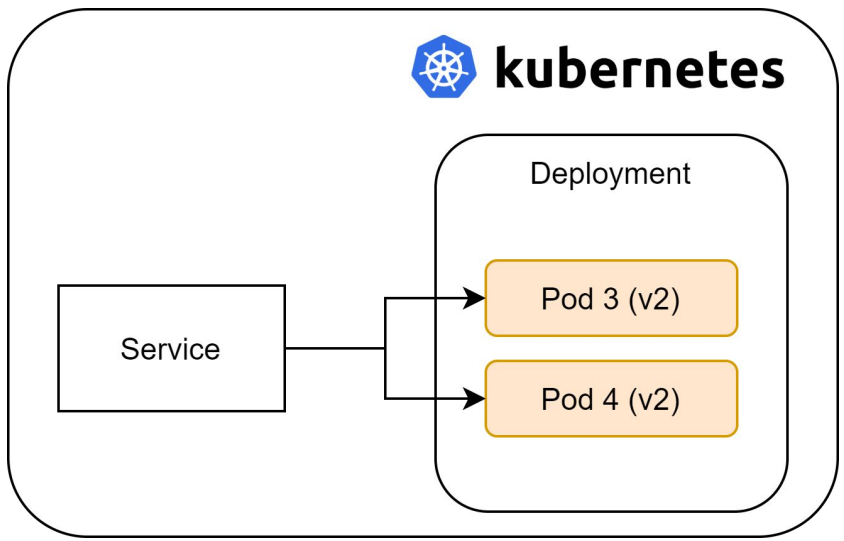
Горизонтальное масштабирование



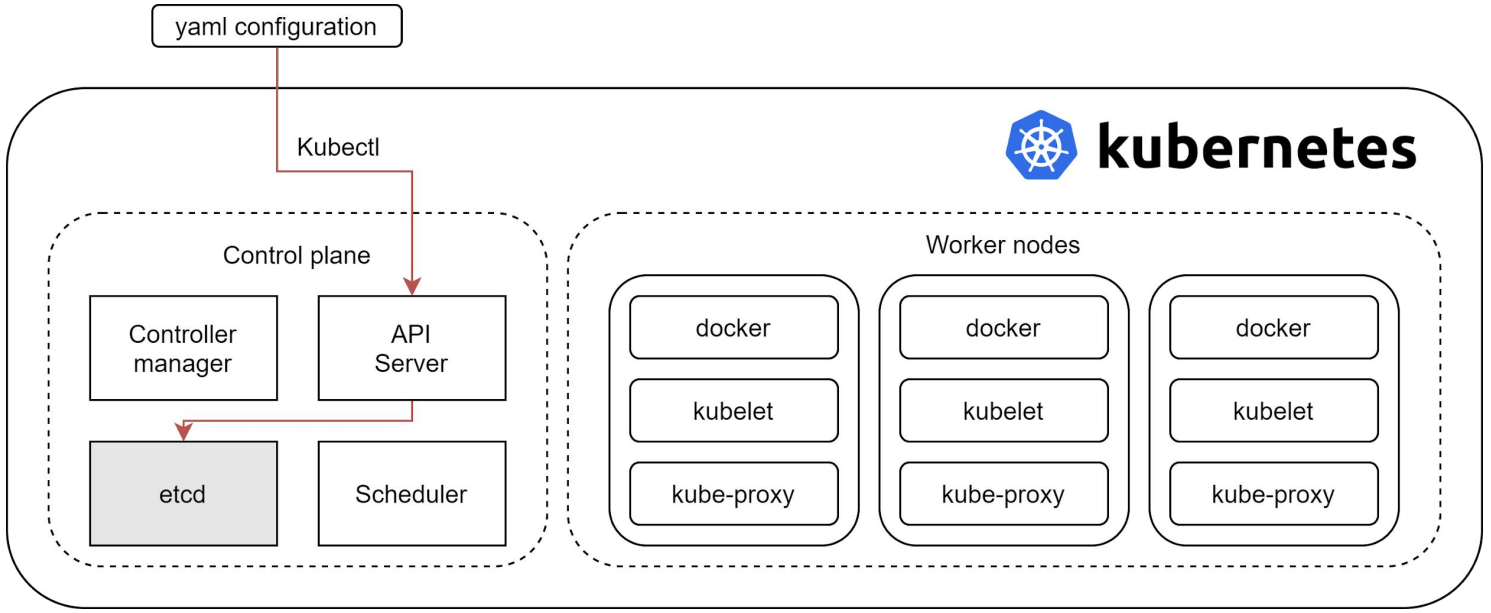
Горизонтальное масштабирование – это способ масштабирования приложений за счет увеличения кол-ва реплик приложения (подов). В Kubernetes такой подход реализован через два механизма:

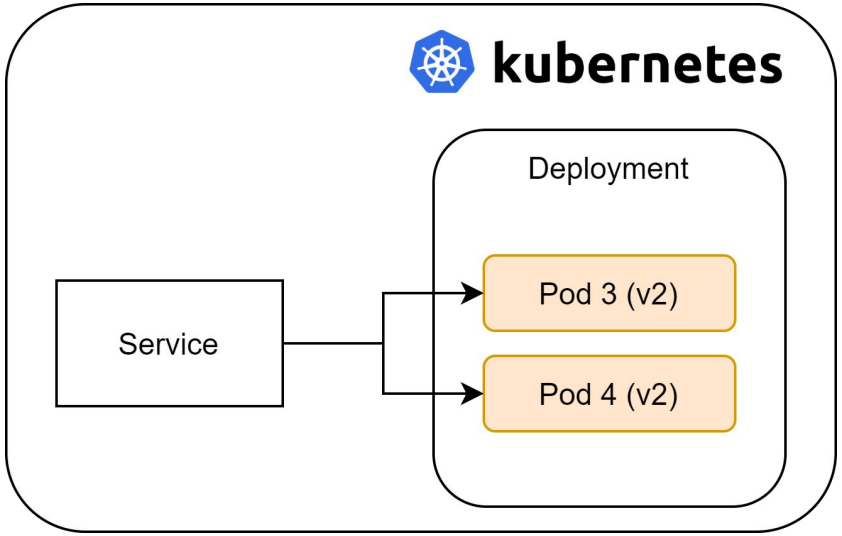
- «Автоматическое горизонтальное масштабирование подов» (англ. Horizontal Pod Autoscaler)
- «Автомасштабирование кластера» (англ. Cluster autoscaling).


kubernetes
Control plane
Controller manager
API Server
etcd
Scheduler
Worker nodes
docker
kubelet
kube-proxy
docker
kubelet
kube-proxy
docker
kubelet
kube-proxy

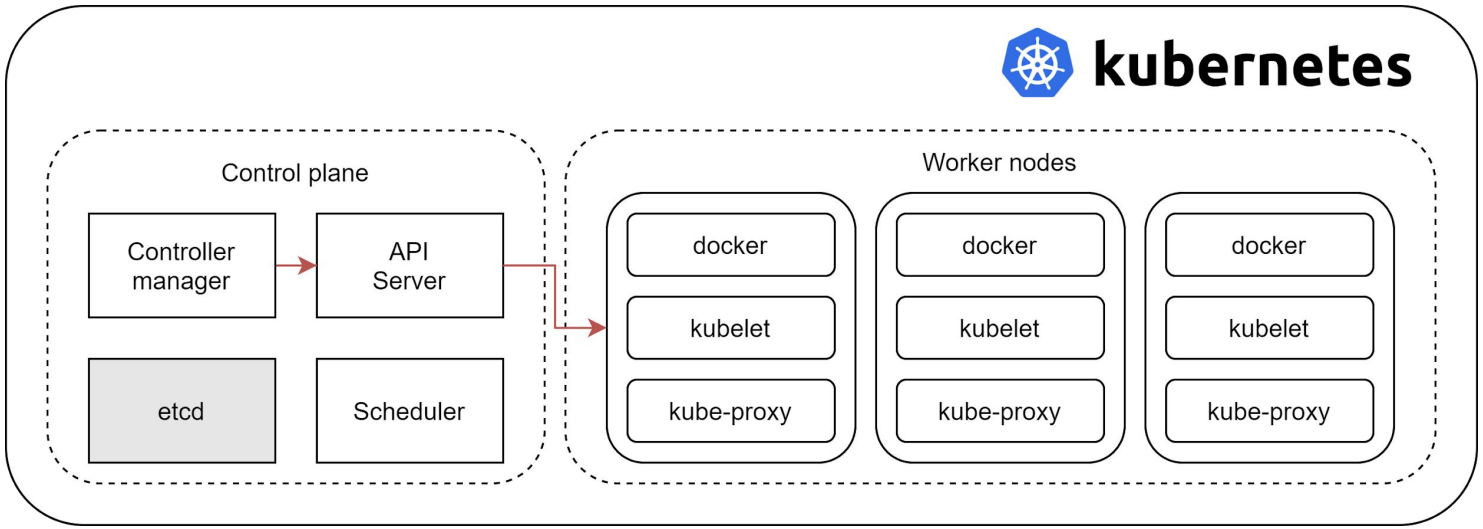

kubernetes
Deployment
Service
Pod 3 (v2)
Pod 4 (v2)

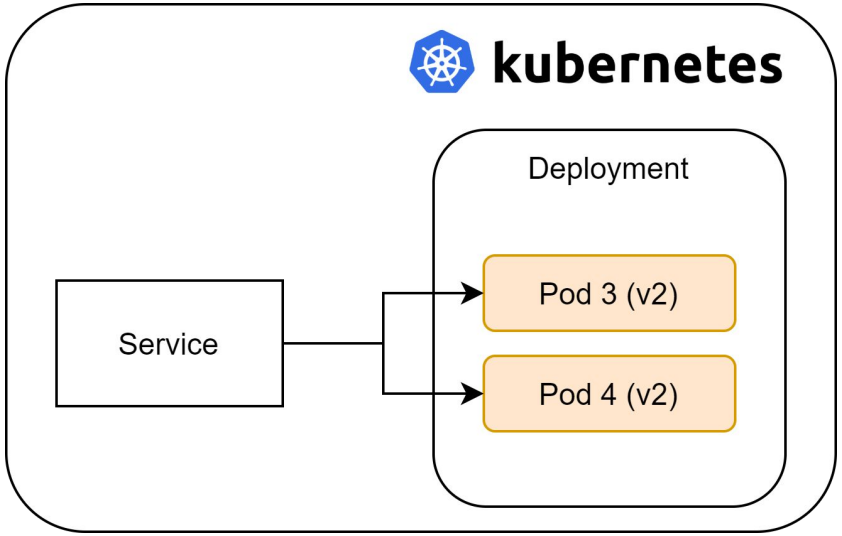
Создание конфигурации для горизонтального масштабирования



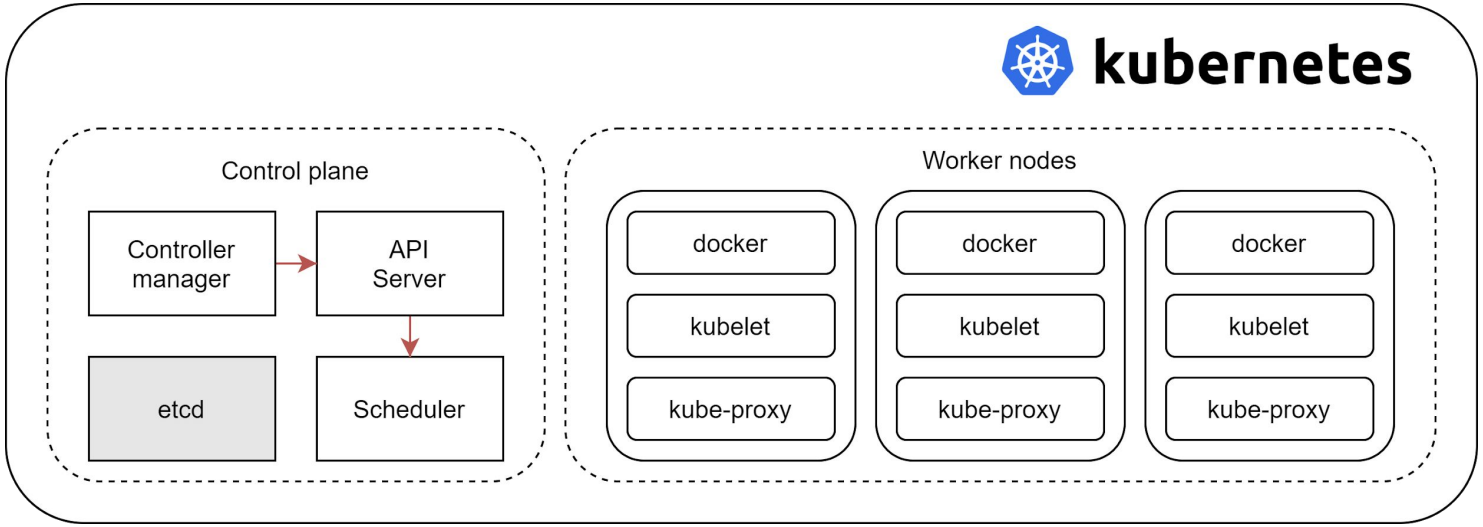


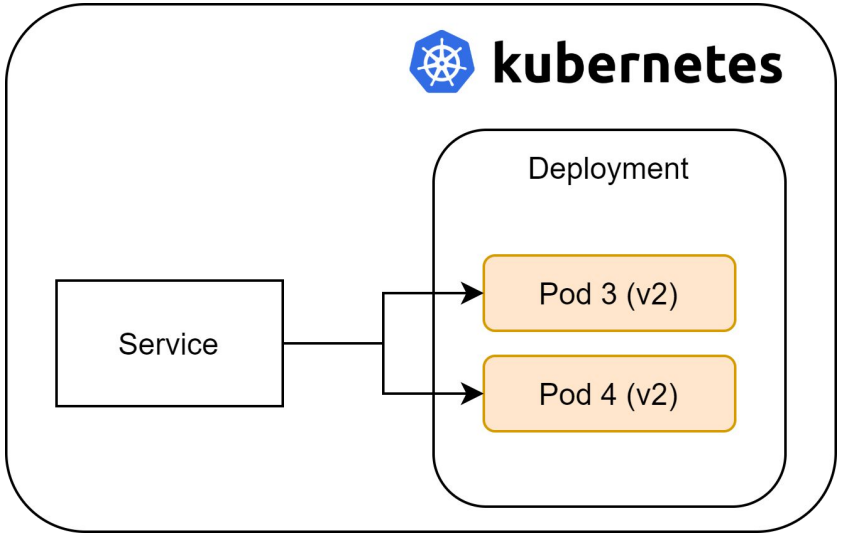
Controller manager запрашивает информацию по утилизации ресурсов в рамках deployment-а



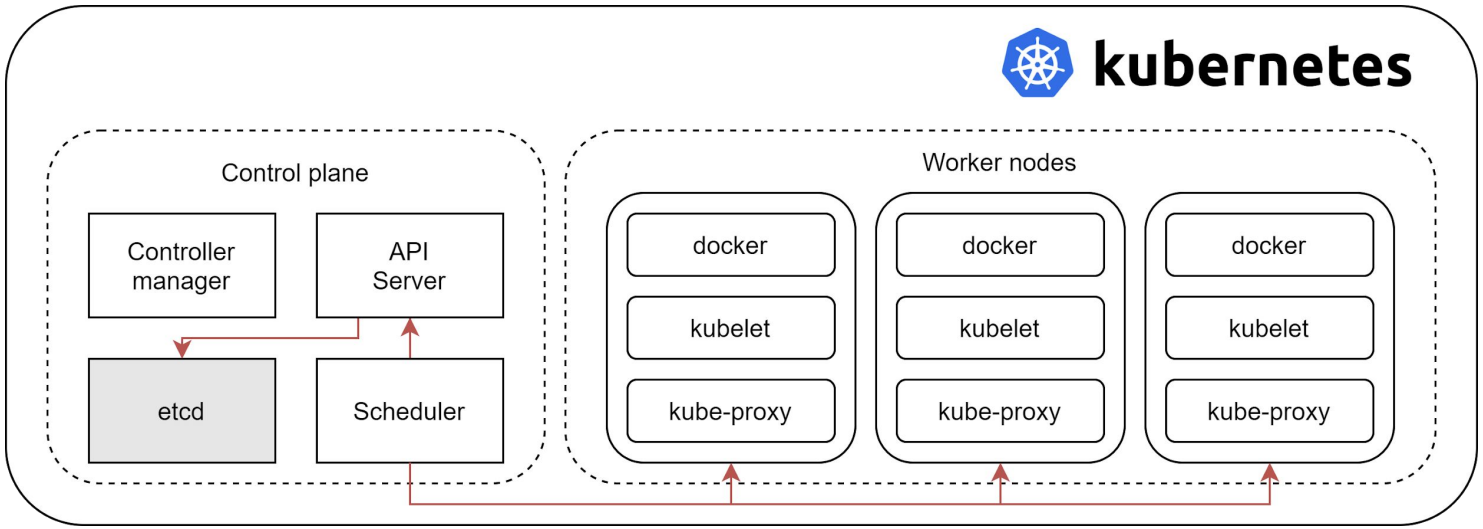


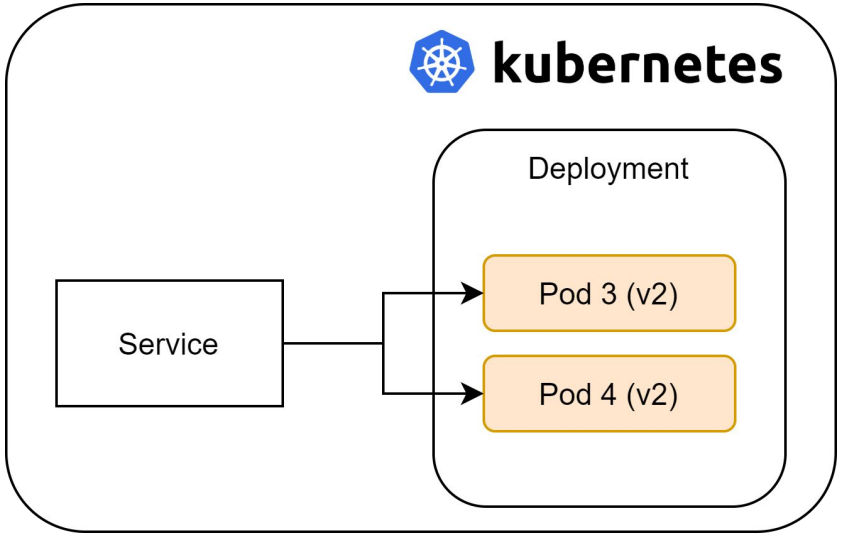
Контрольное значение утилизации ресурсов превышает пороговое



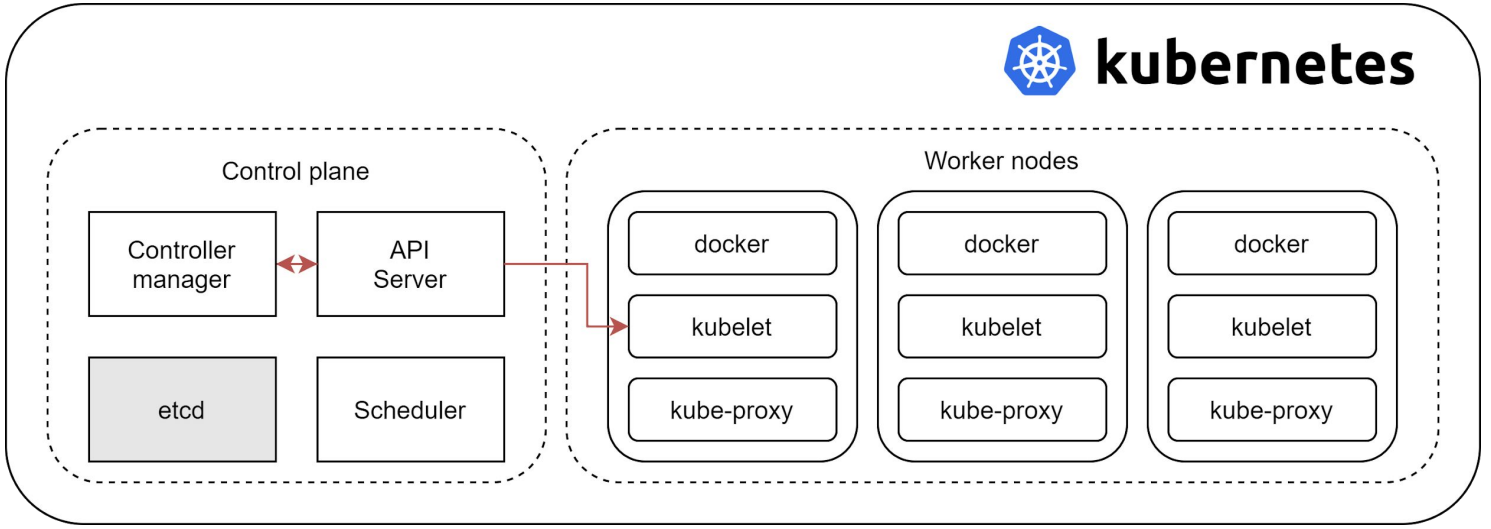


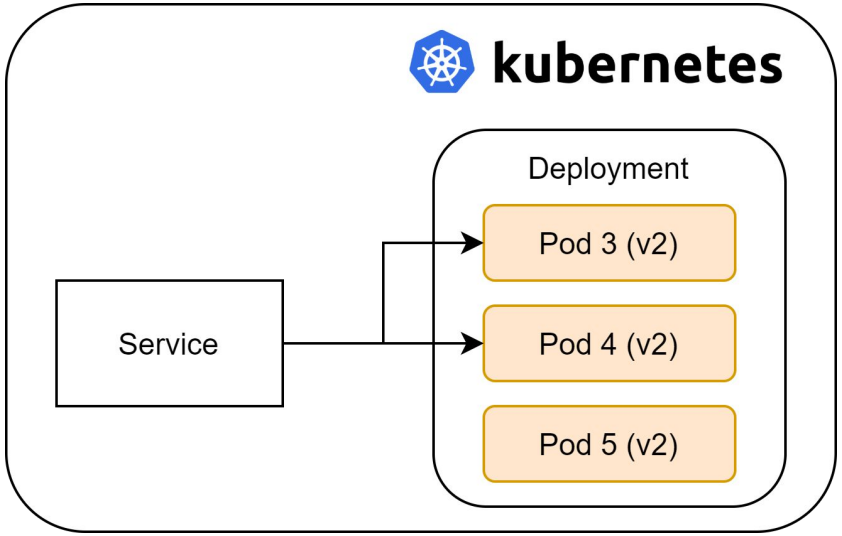
Scheduler принимает решение о том, на каком узле должен быть запущен новая под приложения



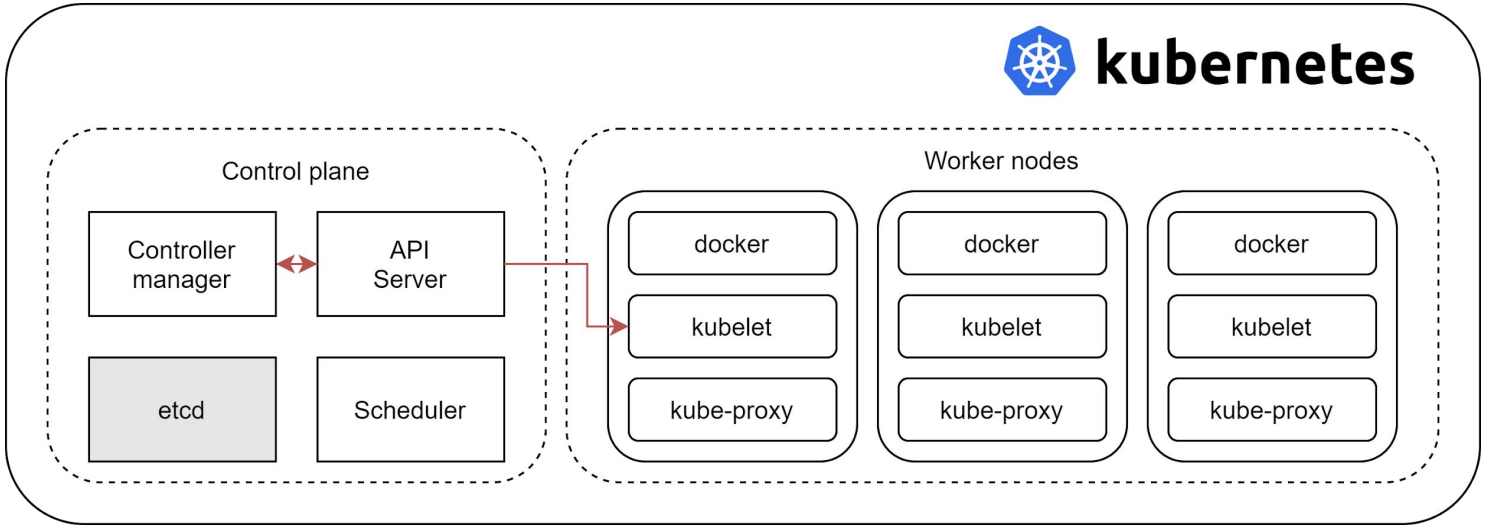


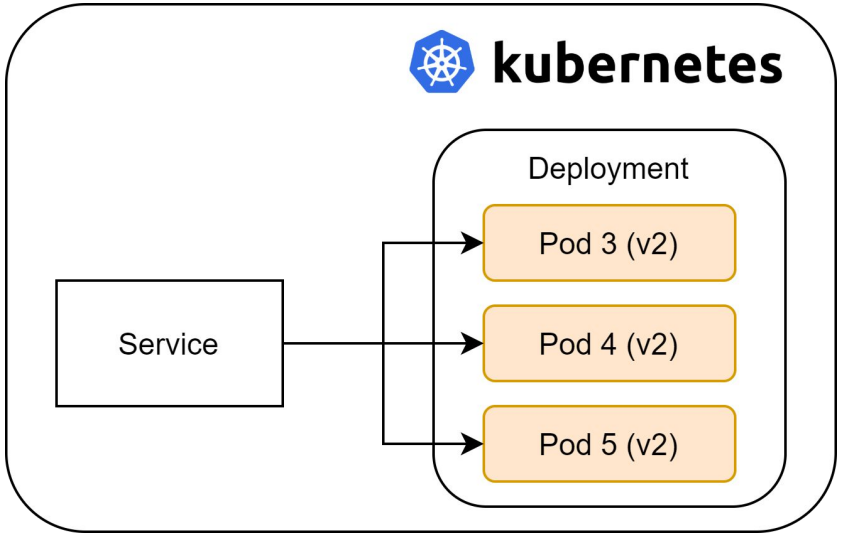
В рамках развертывания создается новый под



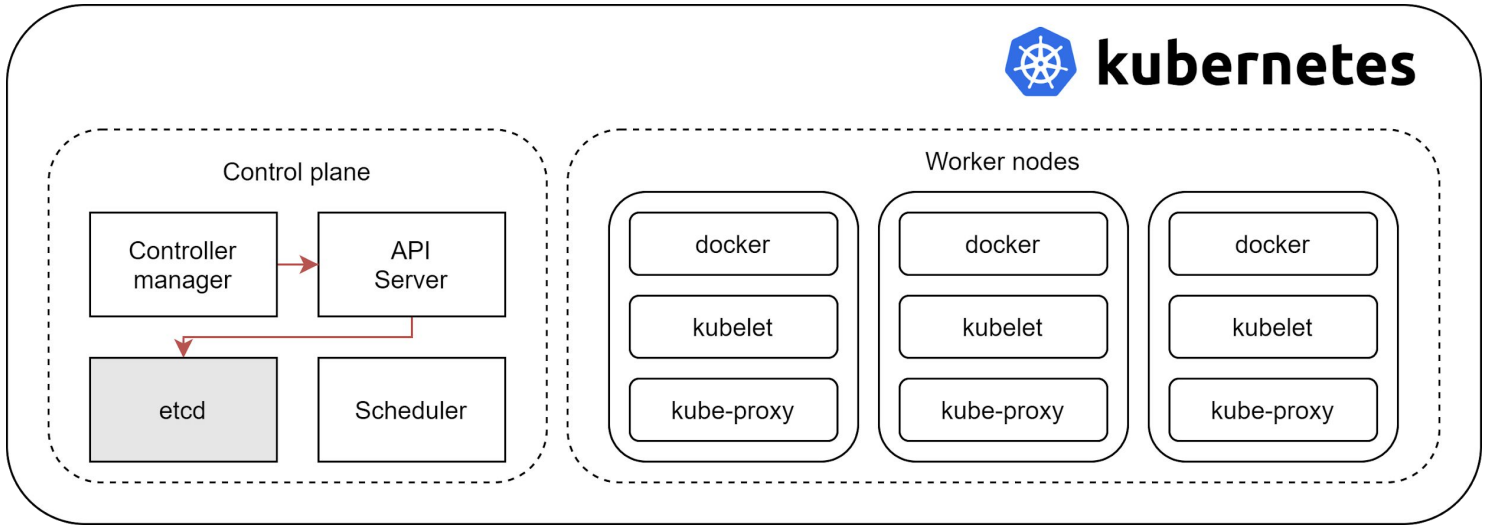


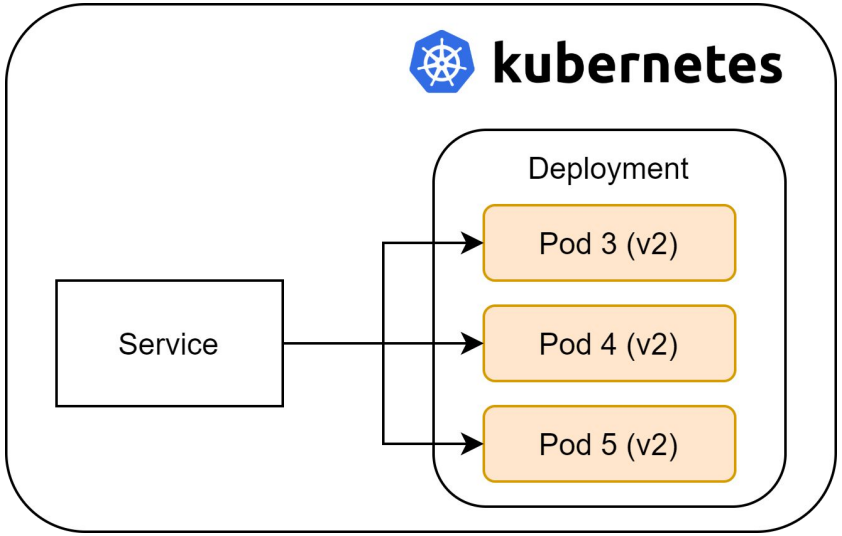
Новый под добавляется в балансировку на уровне сервиса



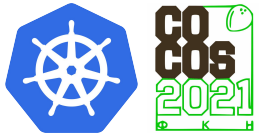


Controller manager сохраняет текущее состояние в etcd

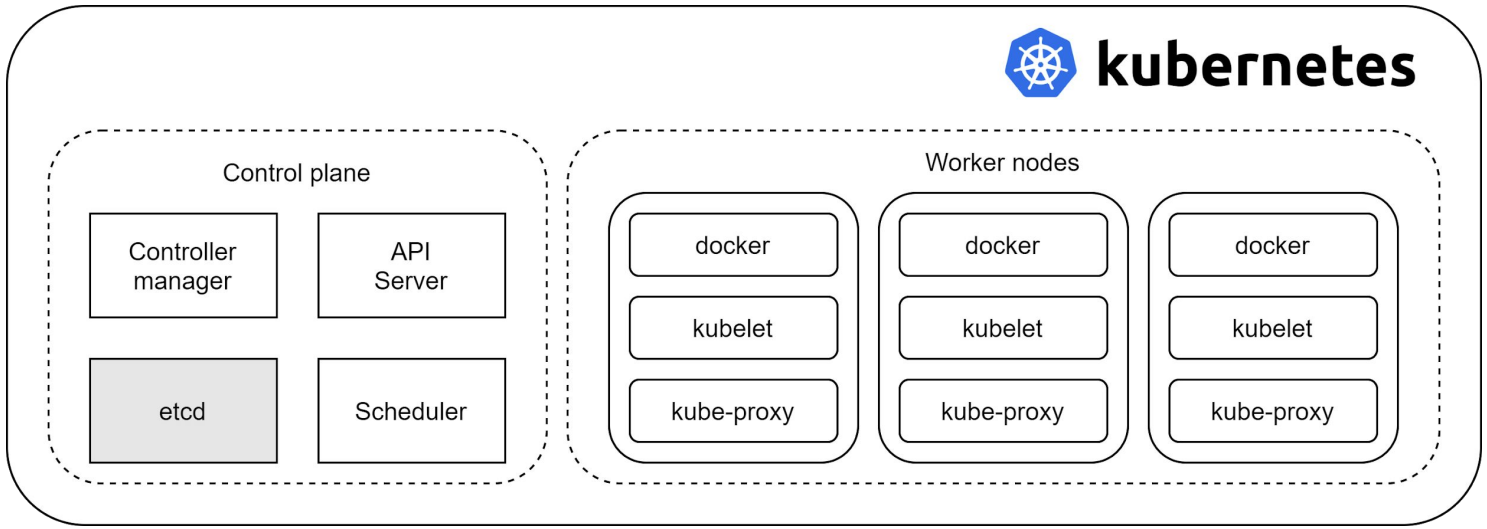


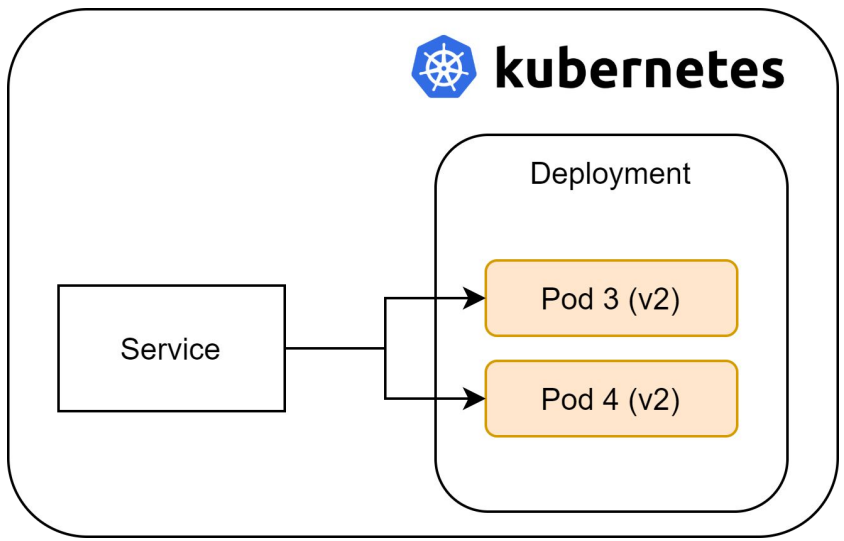


Вертикальное масштабирование

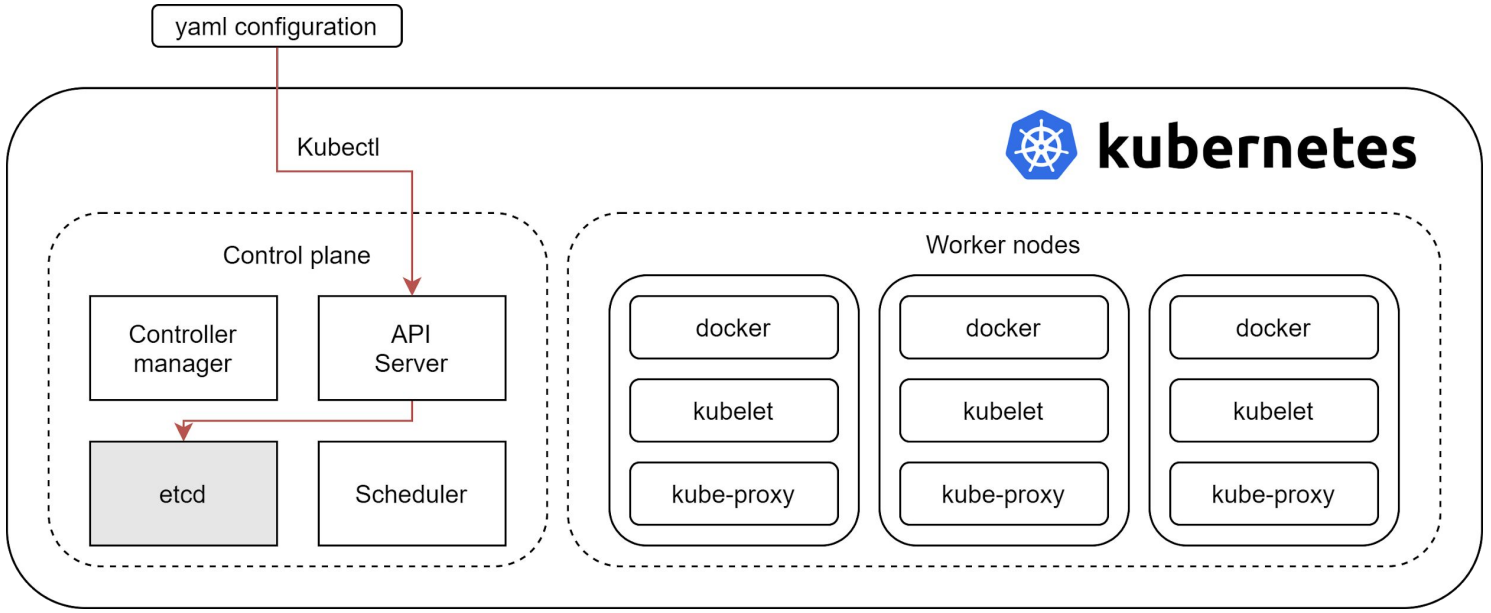


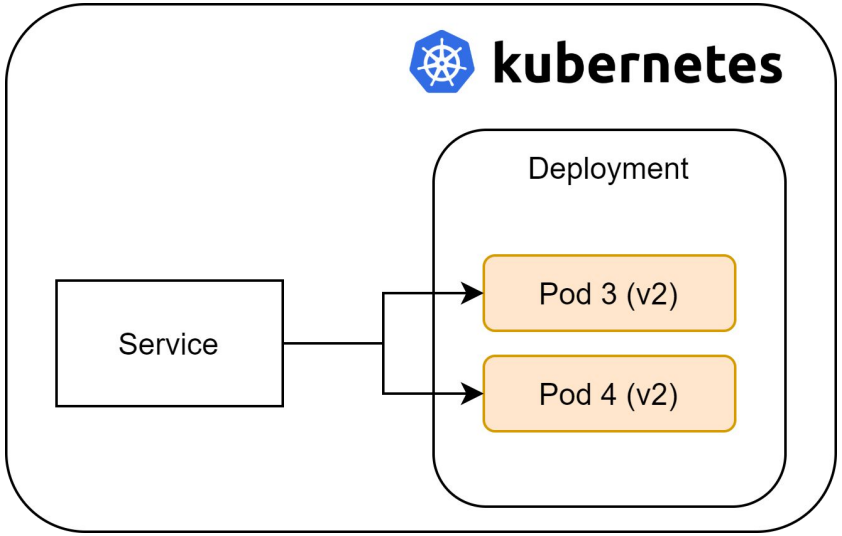
Вертикальное масштабирование – это способ масштабирования приложений за счет увеличения кол-ва ресурсов в существующих подах сервиса. В Kubernetes такой подход называется «Автоматическое вертикальное масштабирование» (англ. Vertical Pod autoscaling).


kubernetes
Control plane
Controller manager
API Server
etcd
Scheduler
Worker nodes
docker
kubelet
kube-proxy
docker
kubelet
kube-proxy
docker
kubelet
kube-proxy

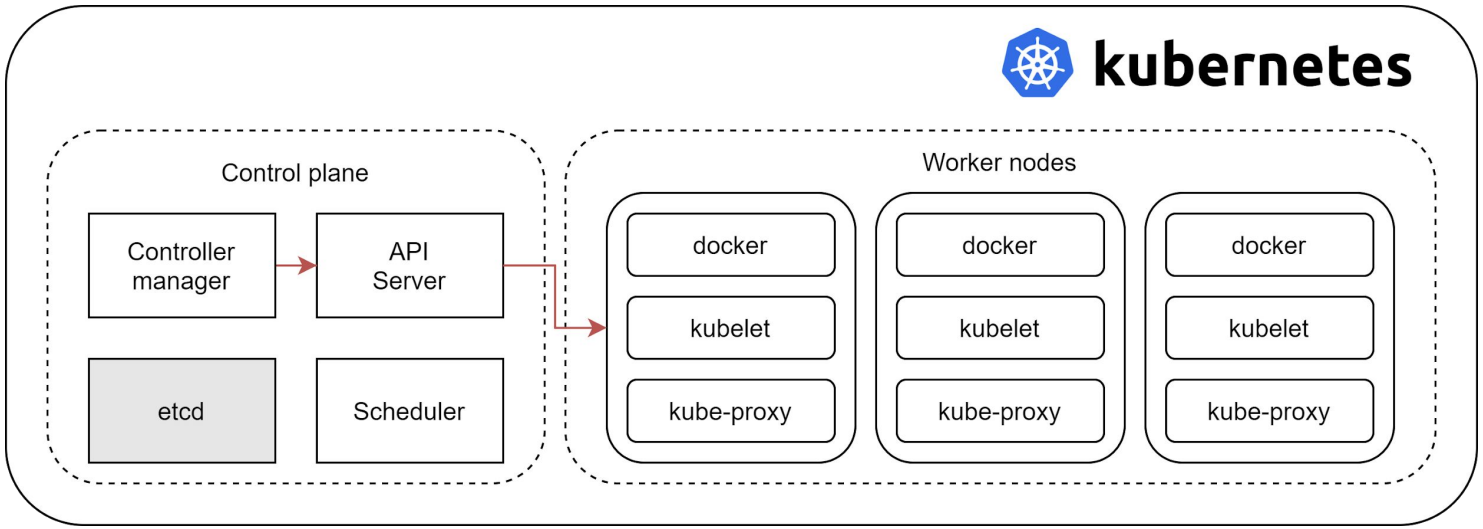

kubernetes
Deployment
Service
Pod 3 (v2)
Pod 4 (v2)

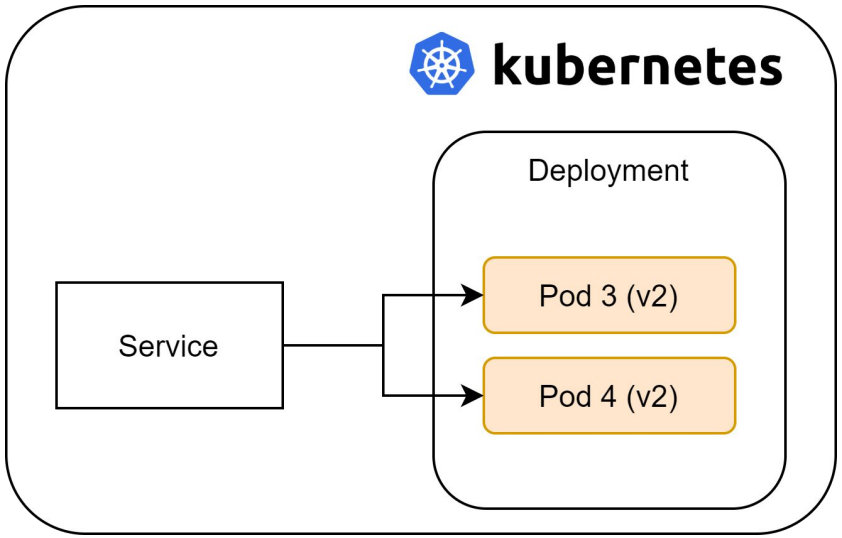
Создание конфигурации для вертикального масштабирования



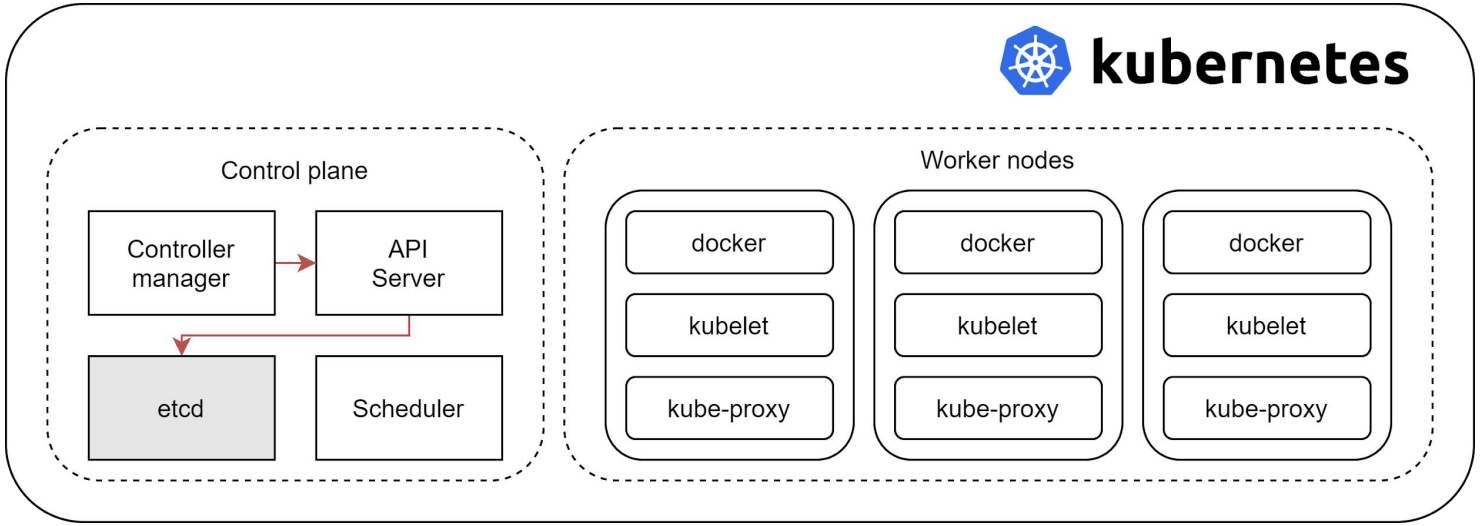


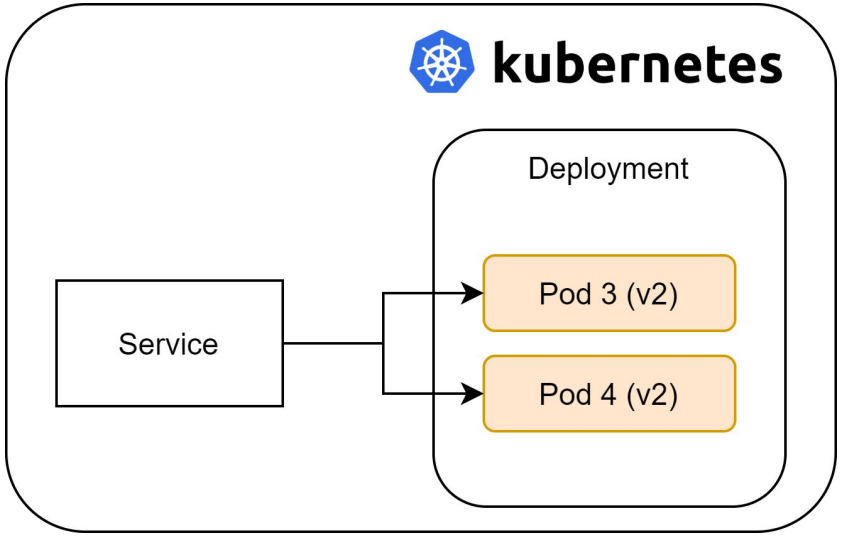
Controller manager запрашивает информацию по утилизации ресурсов в рамках deployment-а



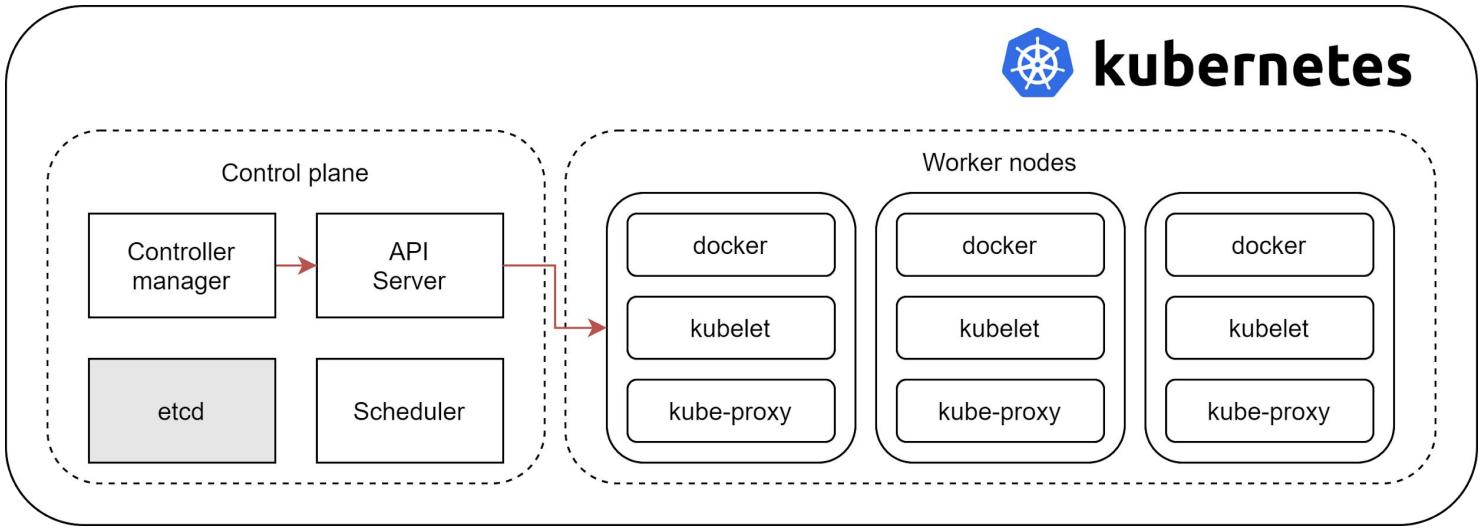


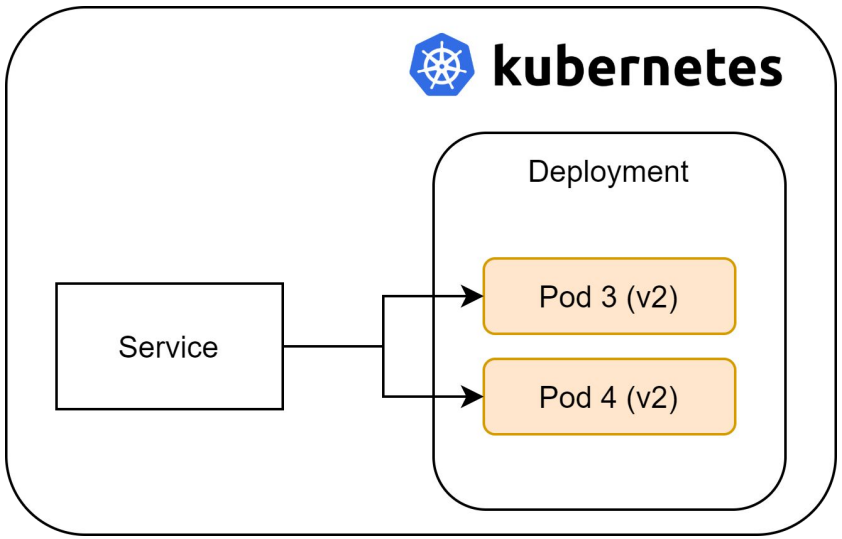
Controller manager добавляет желаемые значения ресурсов для развертывания



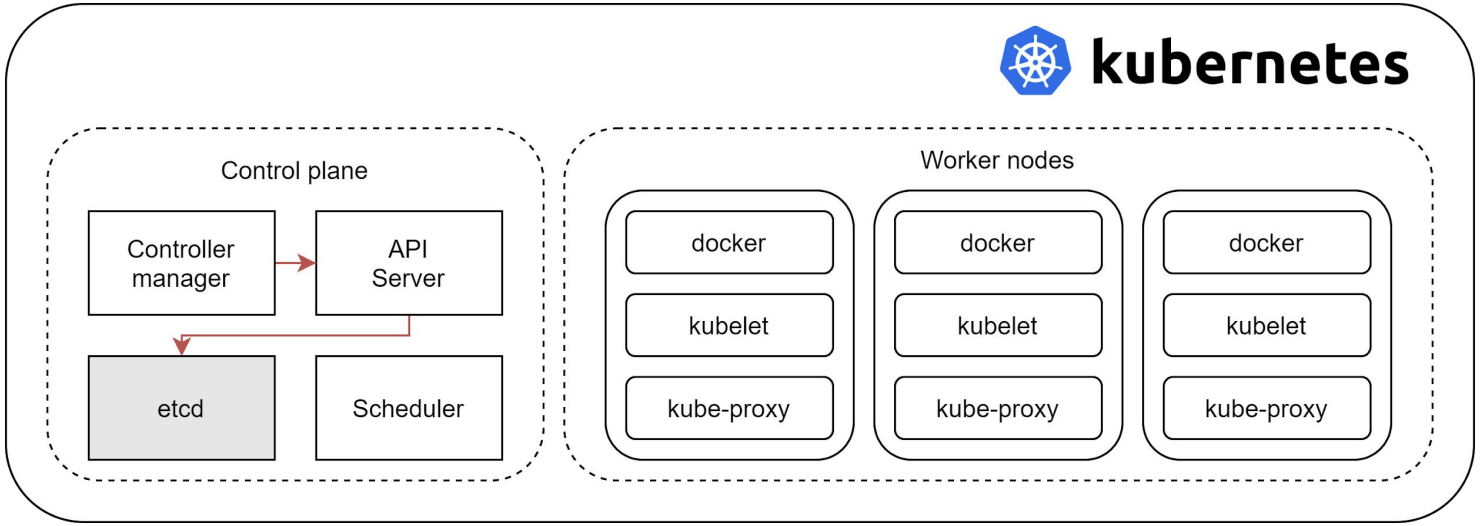


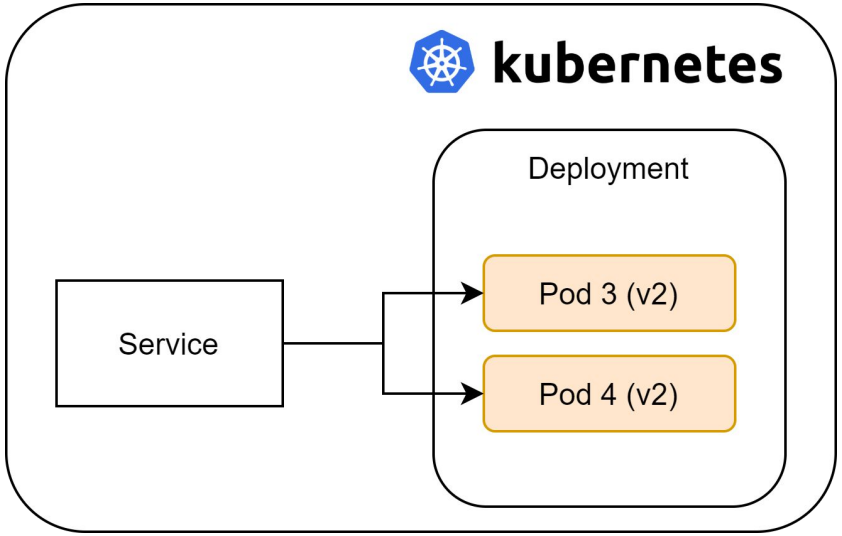
Controller manager изменяет кол-во ресурсов для данных подов



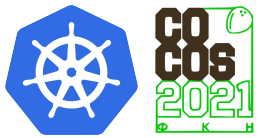


Controller manager сохраняет текущее состояние в etcd



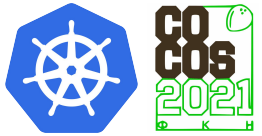


Дальнейшие исследования



1. Исследовать собранные метрики утилизации ресурсов Kubernetes при выполнении ользовании процессов развертывания, горизонтального и вертикального масштабирования.
2. Составить бенчмарки для анализа утилизации ресурсов в кластере Kubernetes и сравнить с другими системами оркестрации приложений (Nomad, Rancher).

Заключение



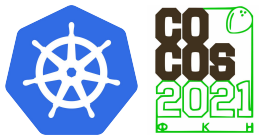
На сегодняшний день Kubernetes является самым популярным инструментом для оркестрации контейнеризированных приложений. Благодаря своей архитектуре и удобному API, он позволяет очень **гибко работать с сущностями для определения состояния кластера**.

Kubernetes позволяет разработчикам **управлять своими приложениями в облаке без глубокого понимания принципов** развертывания и специальных знаний в области системного администрирования.

В рамках данной работы была **проведена связь между компонентами и абстракциями кластера**, что должно улучшить понимание устройства кластера в целом.

Список источников

1. Leila Abdollahi Vayghan, Mohamed Aymen Saied, Maria Toeroe, Ferhat Khendek. Deploying Microservice Based Applications with Kubernetes: Experiments and Lessons Learned // IEEE 11th International Conference on Cloud Computing. 2018. С. 970–973.
2. James Lewis, Martin Fowler. Microservices // [Электронный ресурс]: Martin Fowler: personal website. – Режим доступа: https://martinfowler.com/articles/microservices.html, свободный. (дата обращения: 05.04.21).
3. Leila Abdollahi Vayghan, Mohamed Aymen Saied, Maria Toeroe, Ferhat Khendek. Microservice Based Architecture: Towards High-Availability for Stateful Applications with Kubernetes // IEEE 19th International Conference on Software Quality, Reliability and Security (QRS). 2019. С. 176–185.
4. Leila Abdollahi Vayghan, Mohamed Aymen Saied, Maria Toeroe, Ferhat Khendek. Kubernetes as an Availability Manager for Microservice Applications Journal of Network and Computer Applications. 2019.
5. The Kubernetes Authors. Kubernetes Components // [Электронный ресурс]: Kubernetes official documentation website. – Режим доступа: https://kubernetes.io/docs/concepts/overview/components/, свободный. (дата обращения: 05.04.21).
6. Microservices Research [Электронный ресурс] // Github/VolkovTech – Режим доступа: https://github.com/VolkovTech/microservices-research, свободный. (дата обращения: 05.04.21)



Спасибо за внимание!