КомпонентыKubernetes

Control plane, worker nodes

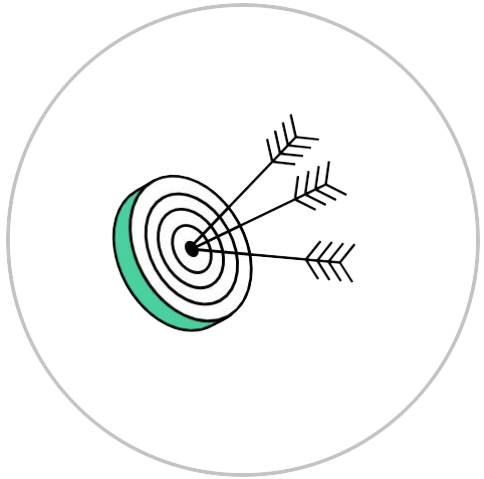
Кирилл Касаткин
DevOps-инженер, Renue

Кирилл Касаткин

DevOps-инженер, Renue

Цели занятия

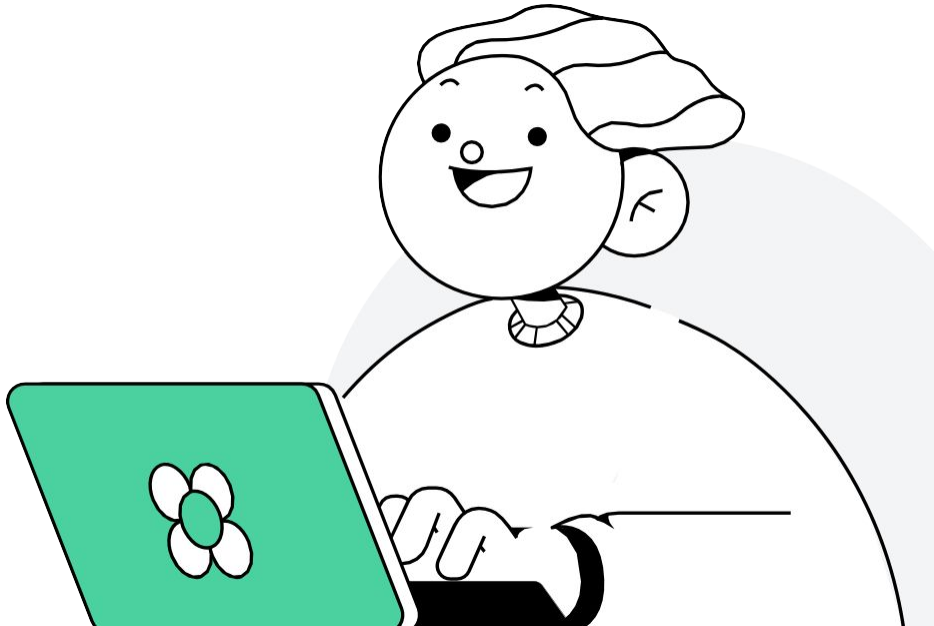
- Познакомиться с компонентами Kubernetes:
  - компоненты control plane (master nodes)
  - компоненты worker nodes
- Разобраться, как происходит взаимодействие указанных компонентов и управление ими



План занятия

1. Архитектура кластера K8s
2. Компоненты control plane
3. Компоненты worker nodes
4. Итоги
5. Домашнее задание

*Нажми на нужный раздел для перехода



Вспоминаем прошлое занятие

Вопрос: что такое кластер?

Вспоминаем прошлое занятие

Вопрос: что такое кластер?

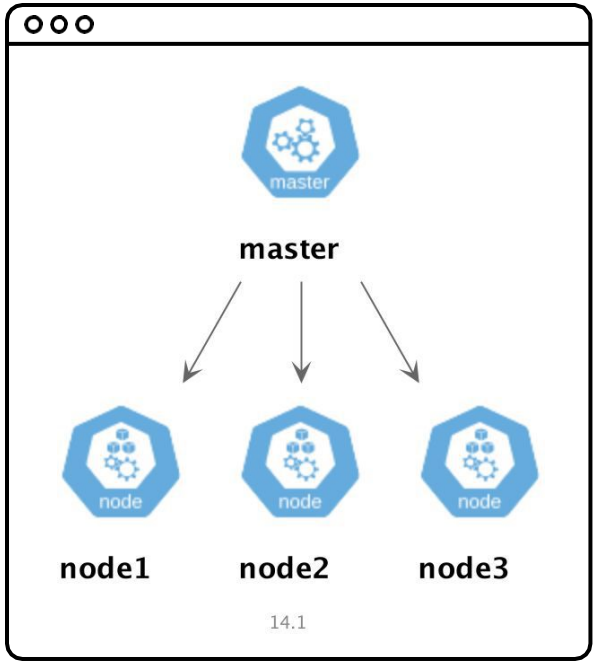
Ответ: группа объединённых компьютеров или нод (node), представляющая единый аппаратный ресурс

Архитектура кластера K8s

1

Архитектура кластера K8s

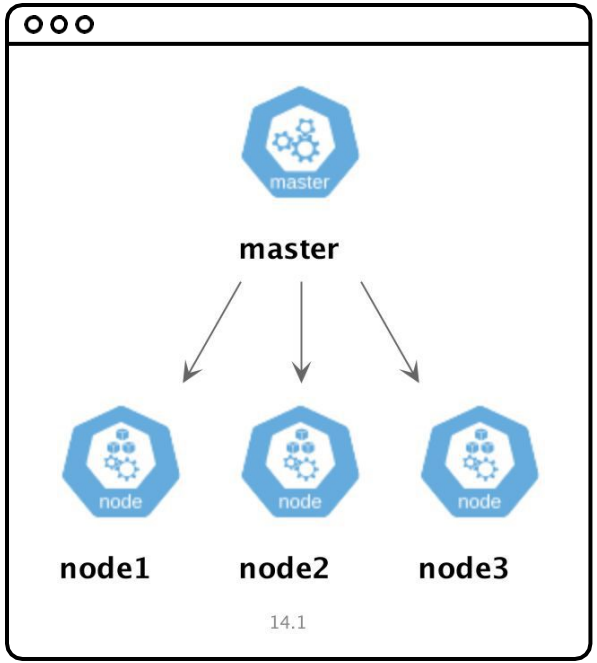
На концептуальном уровне кластер K8s состоит из двух типов рабочих машин — master nodes и worker nodes.



- Управление всем кластером, кроме исполнения подов, происходит с помощью **control plane**

Архитектура кластера K8s

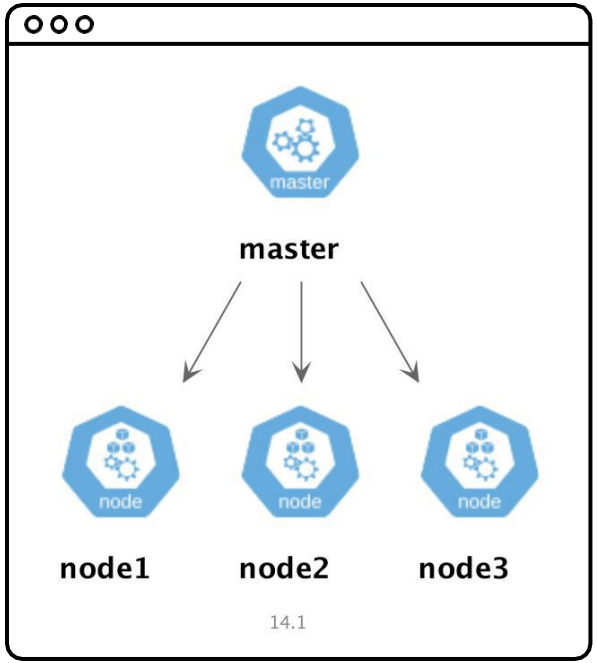
На концептуальном уровне кластер K8s состоит из двух типов рабочих машин — master nodes и worker nodes.



- Управление всем кластером, кроме исполнения подов, происходит с помощью **control plane**
- Control plane состоит из нескольких компонентов, которые, как правило, расположены в **master nodes**

Архитектура кластера K8s

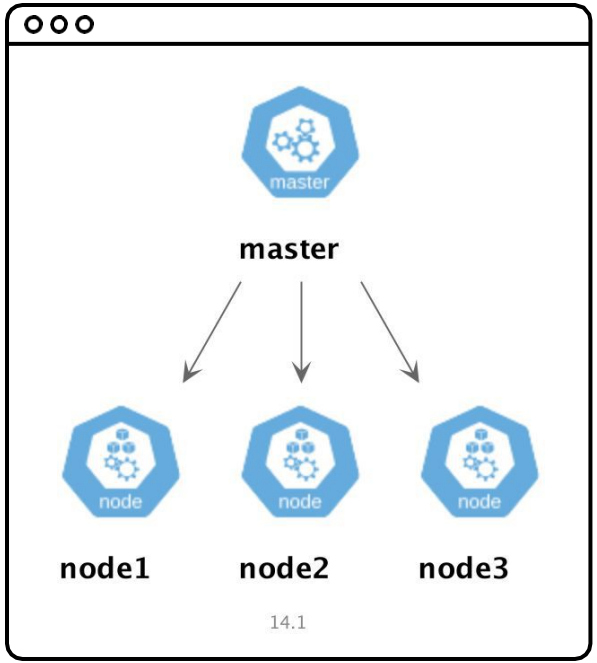
На концептуальном уровне кластер K8s состоит из двух типов рабочих машин — master nodes и worker nodes.



- Управление всем кластером, кроме исполнения подов, происходит с помощью **control plane**
- Control plane состоит из нескольких компонентов, которые, как правило, расположены в **master nodes**
- Приложения обычно запущены в отдельных машинах — **worker nodes**

Архитектура кластера K8s

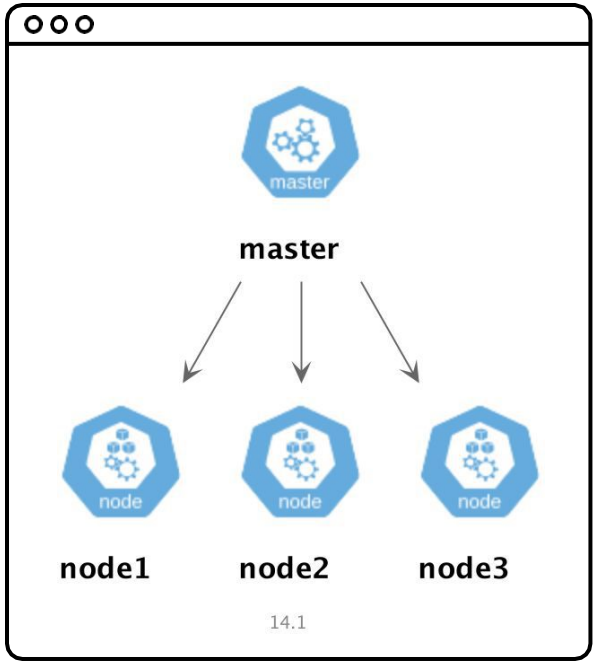
На концептуальном уровне кластер K8s состоит из двух типов рабочих машин — master nodes и worker nodes.



- Управление всем кластером, кроме исполнения подов, происходит с помощью **control plane**
- Control plane состоит из нескольких компонентов, которые, как правило, расположены в **master nodes**
- Приложения обычно запущены в отдельных машинах — **worker nodes**
- Master node, как и worker node, может быть несколько

Архитектура кластера K8s

На концептуальном уровне кластер K8s состоит из двух типов рабочих машин — master nodes и worker nodes.

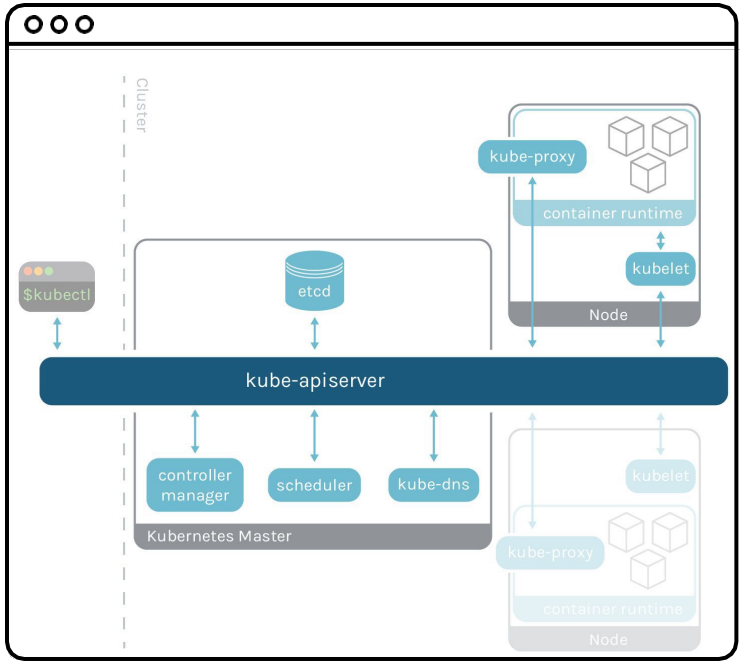


- Управление всем кластером, кроме исполнения подов, происходит с помощью **control plane**
- Control plane состоит из нескольких компонентов, которые, как правило, расположены в **master nodes**
- Приложения обычно запущены в отдельных машинах — **worker nodes**
- Master node, как и worker node, может быть несколько
- На всех нодах должен быть запущен **container runtime**, который не управляется и не зависит от K8s. Взаимодействие происходит через CRI — **container runtime interface**

Компонентыcontrol plane

Компоненты control plane: API-сервер

kube-apiserver — точка входа в кластер. Этот компонент предоставляет REST API и отвечает по HTTPS. Всё управление кластером происходит через kube-apiserver.

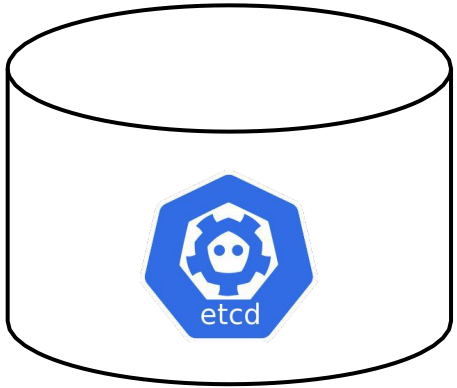


kube-apiserver — это связующее звено для всех компонентов системы, именно благодаря ему осуществляется общение.
Можно представить этот компонент как **шину**, к которой подключены другие компоненты

Источник

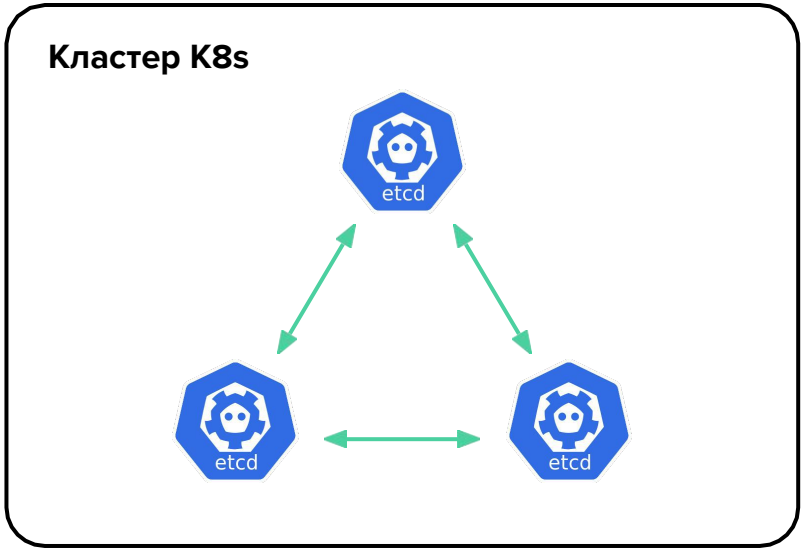
Компоненты control plane: etcd

Распределённое key-value-хранилище.
Для отказоустойчивости рекомендуется нечётное количество экземпляров (RAFT- алгоритм)



Компоненты control plane: etcd

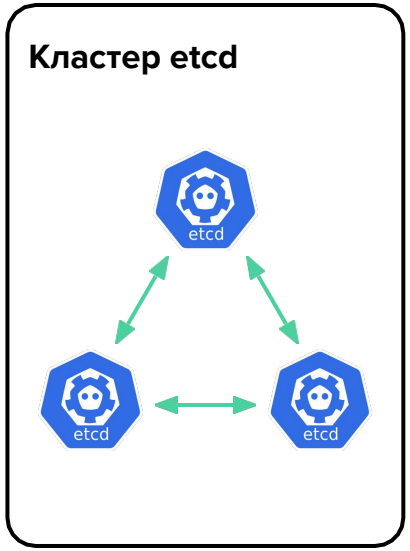
Распределённое key-value-хранилище. Для отказоустойчивости рекомендуется нечётное количество экземпляров (RAFT-алгоритм).

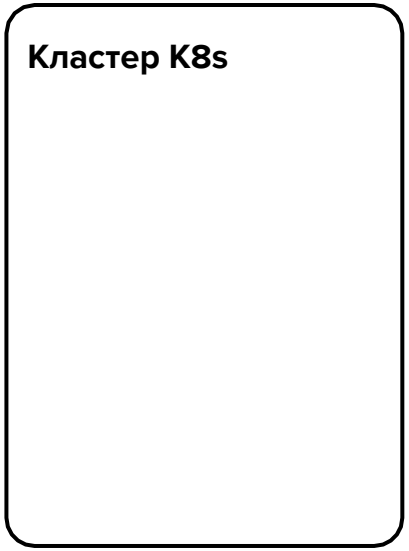


- В простейшем случае etcd может быть запущен на мастер-нодах в виде подов или сервисов

Компоненты control plane: etcd

Распределённое key-value-хранилище. Для отказоустойчивости рекомендуется нечётное количество экземпляров (RAFT-алгоритм).

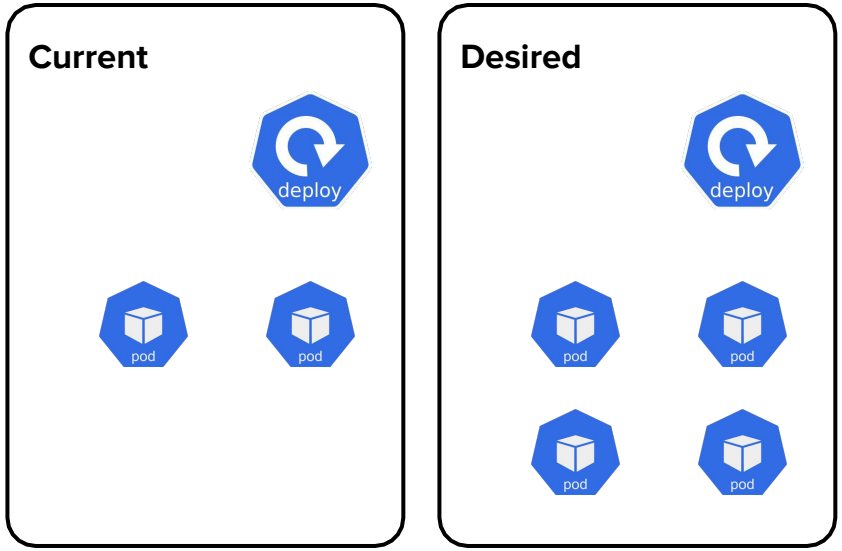




- В простейшем случае etcd может быть запущен на мастер-нодах в виде подов или сервисов
- Для обеспечения максимальной производительности и отказоустойчивости может быть отдельный кластер **etcd** на выделенных серверах

Компоненты control plane: etcd

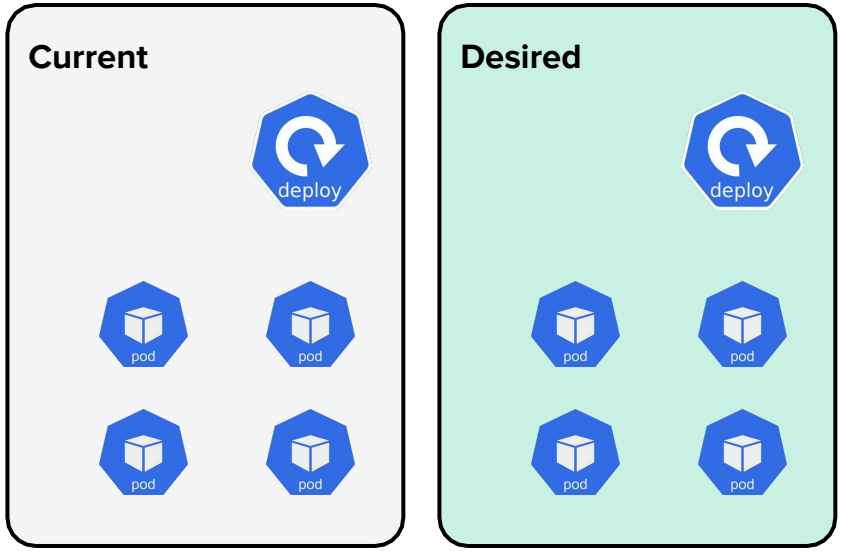
Распределённое key-value-хранилище. Для отказоустойчивости рекомендуется нечётное количество экземпляров (RAFT-алгоритм).



- В простейшем случае etcd может быть запущен на мастер-нодах в виде подов или сервисов
- Для обеспечения максимальной производительности и отказоустойчивости может быть отдельный кластер **etcd** на выделенных серверах
- В этой БД одновременно сохраняется несколько версий одного и того же значения: current state — desired state

Компоненты control plane: etcd

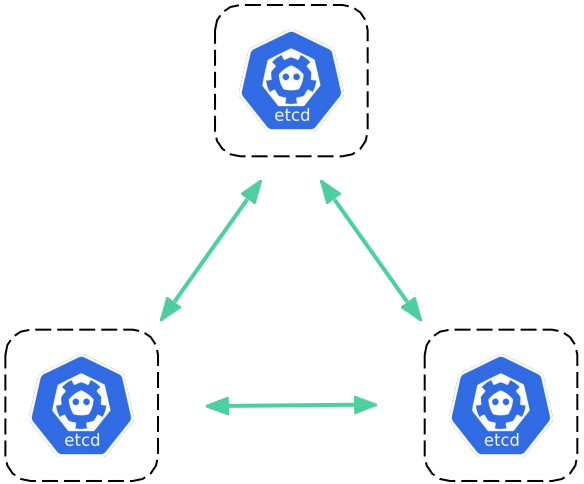
Распределённое key-value-хранилище. Для отказоустойчивости рекомендуется нечётное количество экземпляров (RAFT-алгоритм).



- В простейшем случае etcd может быть запущен на мастер-нодах в виде подов или сервисов
- Для обеспечения максимальной производительности и отказоустойчивости может быть отдельный кластер **etcd** на выделенных серверах
- В этой БД одновременно сохраняется несколько версий одного и того же значения: current state — desired state

Компоненты control plane: etcd

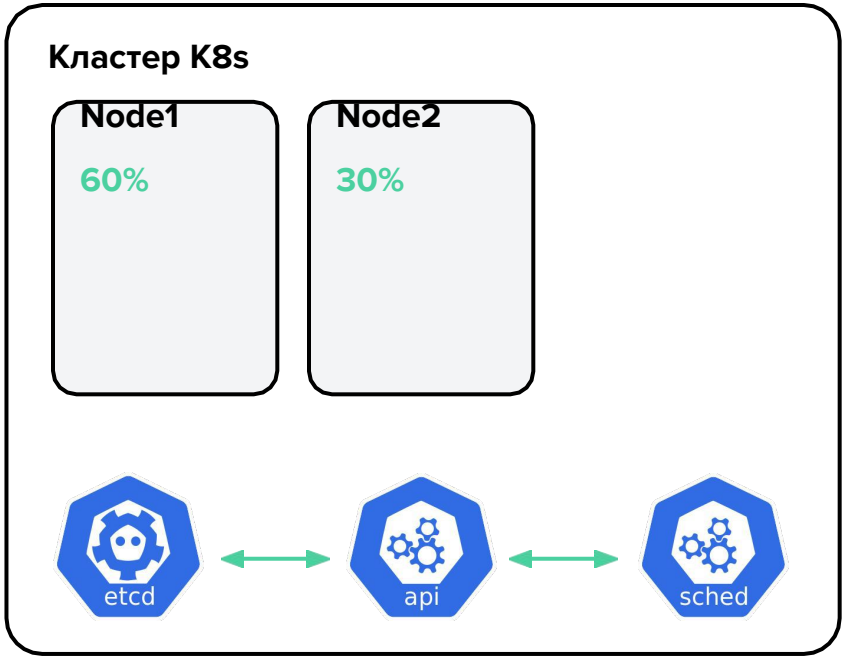
Распределённое key-value-хранилище. Для отказоустойчивости рекомендуется нечётное количество экземпляров (RAFT-алгоритм).



- В простейшем случае etcd может быть запущен на мастер-нодах в виде подов или сервисов
- Для обеспечения максимальной производительности и отказоустойчивости может быть отдельный кластер **etcd** на выделенных серверах
- В этой БД одновременно сохраняется несколько версий одного и того же значения: current state — desired state
- В этой БД не хранятся данные приложений

Компоненты control plane: kube-scheduler

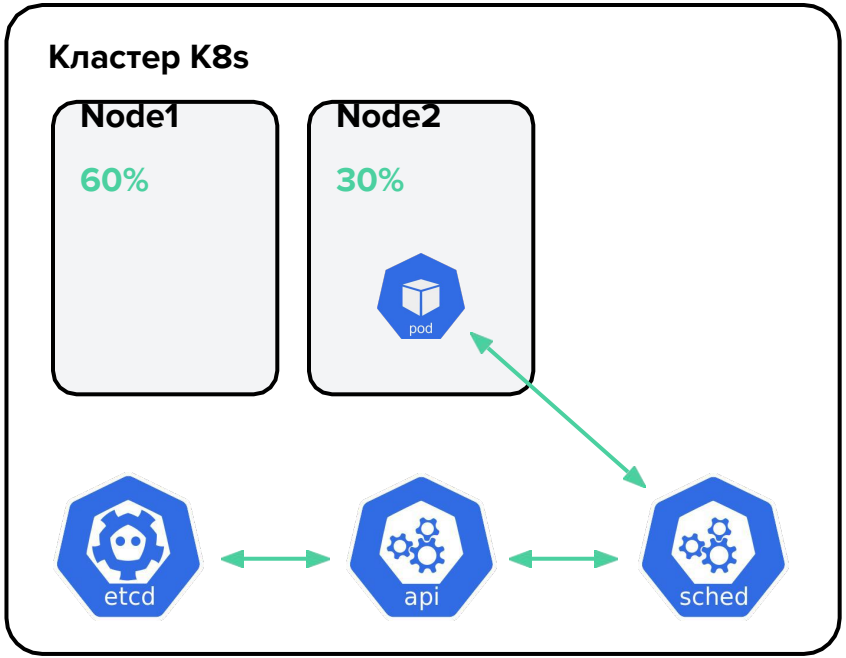
kube-scheduler отслеживает вновь созданные поды, которым не назначена нода, и выбирает наиболее подходящую ноду для разворачивания пода.



- При выборе ноды учитывает текущие ресурсы — записи о ресурсах хранятся в etcd

Компоненты control plane: kube-scheduler

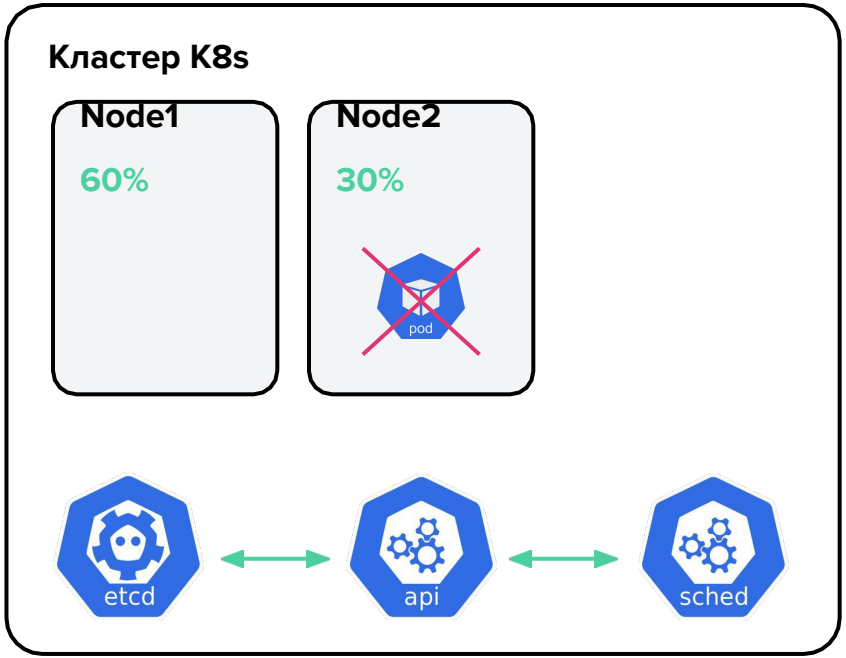
kube-scheduler отслеживает вновь созданные поды, которым не назначена нода, и выбирает наиболее подходящую ноду для разворачивания пода.



- При выборе ноды учитывает текущие ресурсы — записи о ресурсах хранятся в etcd
- kube-scheduler определяет, на какую ноду разместить под, через запись etcd, развёртывает kubelet

Компоненты control plane: kube-scheduler

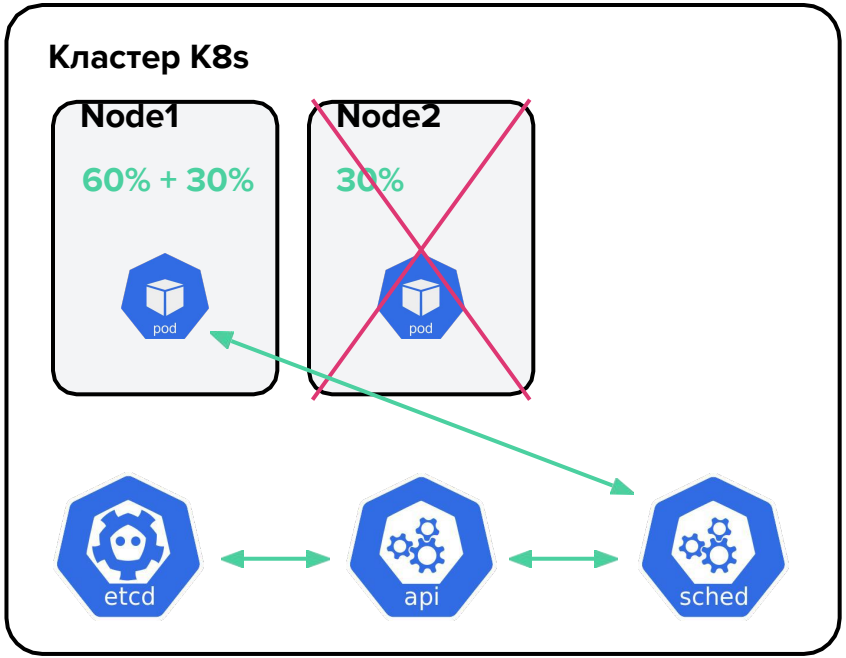
kube-scheduler отслеживает вновь созданные поды, которым не назначена нода, и выбирает наиболее подходящую ноду для разворачивания пода.



- При выборе ноды учитывает текущие ресурсы — записи о ресурсах хранятся в etcd
- kube-scheduler определяет, на какую ноду разместить под, через запись etcd, развёртывает kubelet
- При пересоздании пода происходит вычисление его нового положения

Компоненты control plane: kube-scheduler

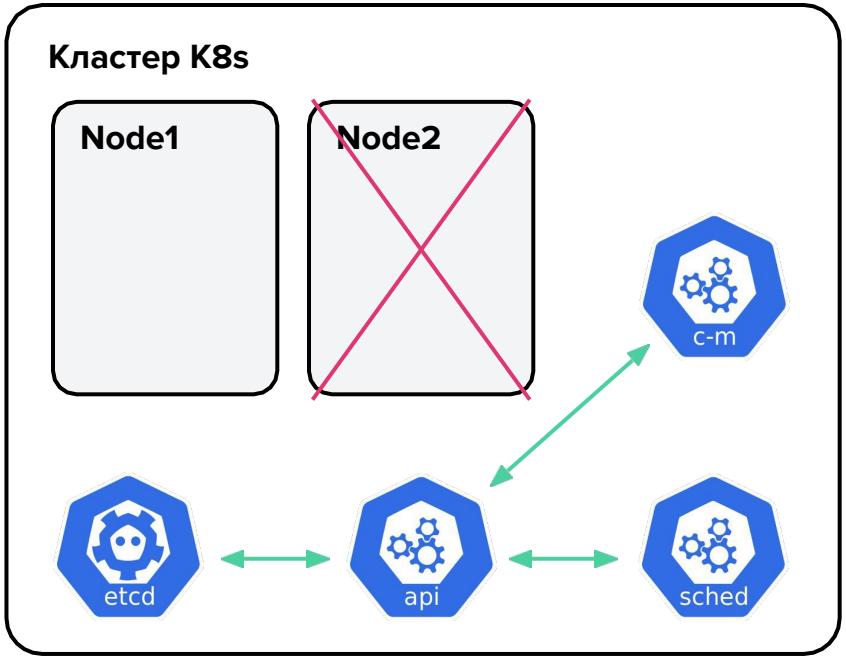
kube-scheduler отслеживает вновь созданные поды, которым не назначена нода, и выбирает наиболее подходящую ноду для разворачивания пода.



- При выборе ноды учитывает текущие ресурсы — записи о ресурсах хранятся в etcd
- kube-scheduler определяет, на какую ноду разместить под, через запись etcd, развёртывает kubelet
- При пересоздании пода происходит вычисление его нового положения
- Если произойдёт потеря ноды, то все поды, расположенные на этой ноде, должны быть перераспределены на другие ноды.
  По этой причине необходимо резервировать ресурсы на рабочих нодах, чтобы обеспечить отказоустойчивость

Компоненты control plane: kube-controller-manager

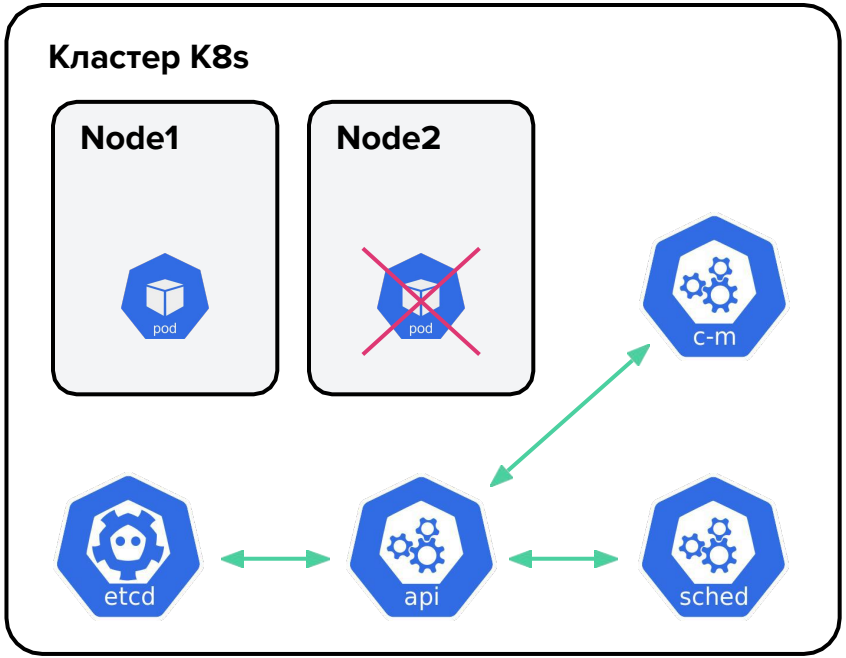
kube-controller-manager состоит из нескольких компонентов, которые запущены в одном процессе. Они выполняют ряд специфических задач:



- **node controller** — опрашивает ноды на предмет их работоспособности, записывает статусы нод в etcd

Компоненты control plane: kube-controller-manager

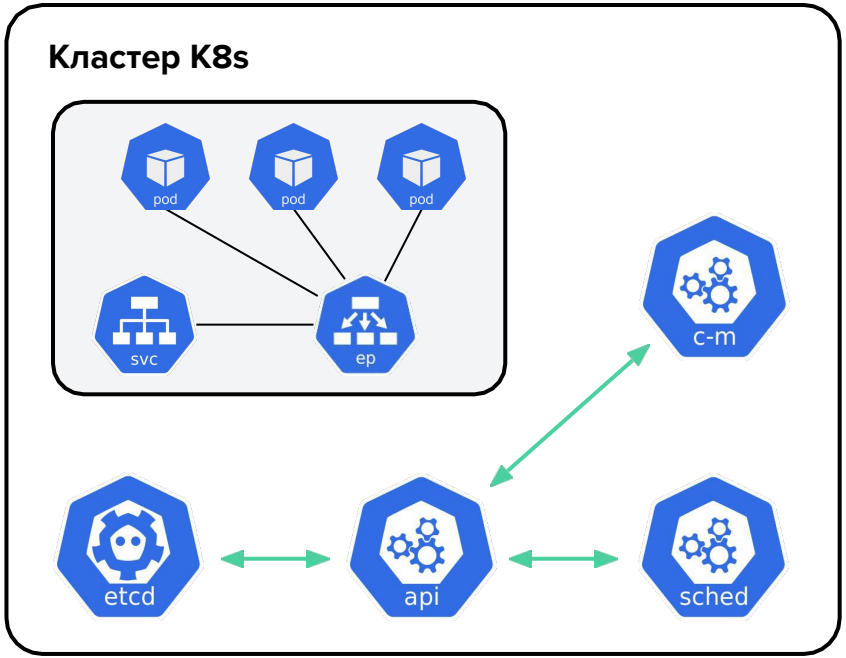
kube-controller-manager состоит из нескольких компонентов, которые запущены в одном процессе. Они выполняют ряд специфических задач:



- **node controller** — опрашивает ноды на предмет их работоспособности, записывает статусы нод в etcd
- **job controller** — отслеживает объекты job, создаёт поды для выполнения назначенных задач

Компоненты control plane: kube-controller-manager

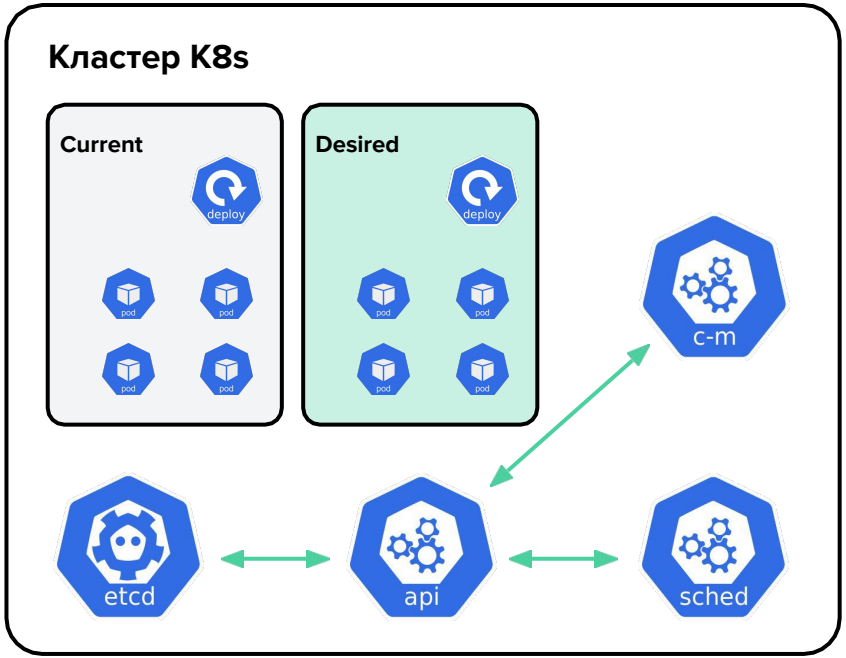
kube-controller-manager состоит из нескольких компонентов, которые запущены в одном процессе. Они выполняют ряд специфических задач:



- **node controller** — опрашивает ноды на предмет их работоспособности, записывает статусы нод в etcd
- **job controller** — отслеживает объекты job, создаёт поды для выполнения назначенных задач
- **endpoints controller** — связывает сервисы и поды

Компоненты control plane: kube-controller-manager

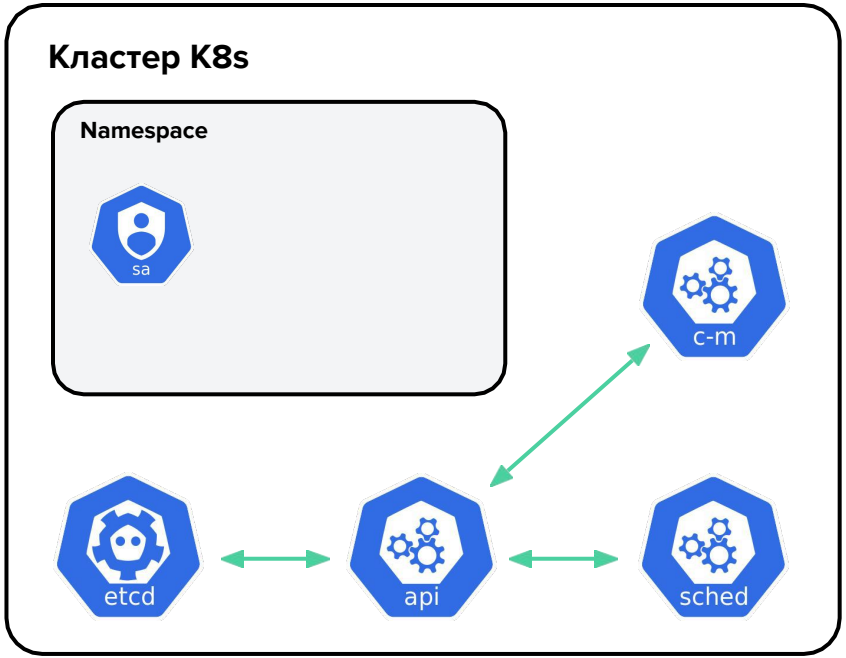
kube-controller-manager состоит из нескольких компонентов, которые запущены в одном процессе. Они выполняют ряд специфических задач:



- **node controller** — опрашивает ноды на предмет их работоспособности, записывает статусы нод в etcd
- **job controller** — отслеживает объекты job, создаёт поды для выполнения назначенных задач
- **endpoints controller — связывает сервисы и поды**
- **replication controller** — отслеживает желаемое количество реплик

Компоненты control plane: kube-controller-manager

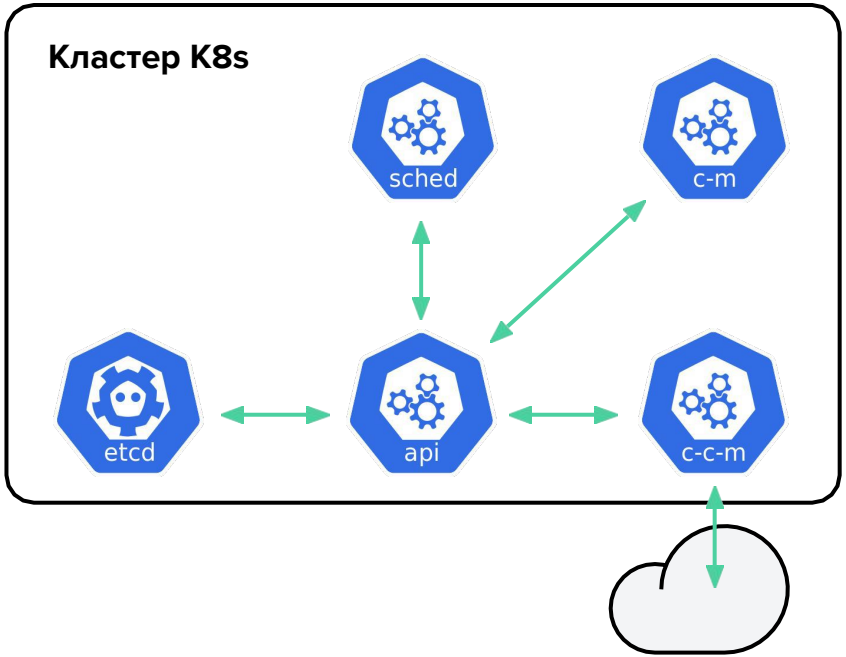
kube-controller-manager состоит из нескольких компонентов, которые запущены в одном процессе. Они выполняют ряд специфических задач:



- **node controller** — опрашивает ноды на предмет их работоспособности, записывает статусы нод в etcd
- **job controller** — отслеживает объекты job, создаёт поды для выполнения назначенных задач
- **endpoints controller** — связывает сервисы и поды
- **replication controller** — отслеживает желаемое количество реплик
- **service account controller** — создаёт сервисные аккаунты для новых namespace

Компоненты control plane: cloud-controller-manager

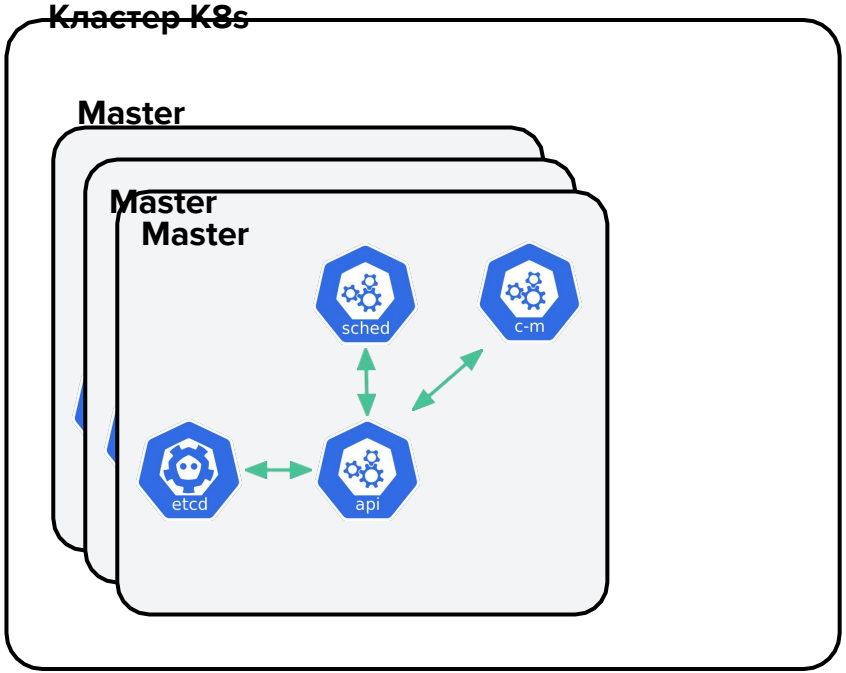
cloud-controller-manager — специфический компонент для облачного провайдера, связывающий кластер K8s с API облачного провайдера:



- **node controller** — для проверки облачного провайдера: была ли удалена нода после того, как она перестала отвечать
- **route controller** — для настройки маршрутов в облачной инфраструктуре
- **service controller** — для создания, обновления и удаления балансировщиков нагрузки в облачной инфраструктуре

Компоненты control plane: HA

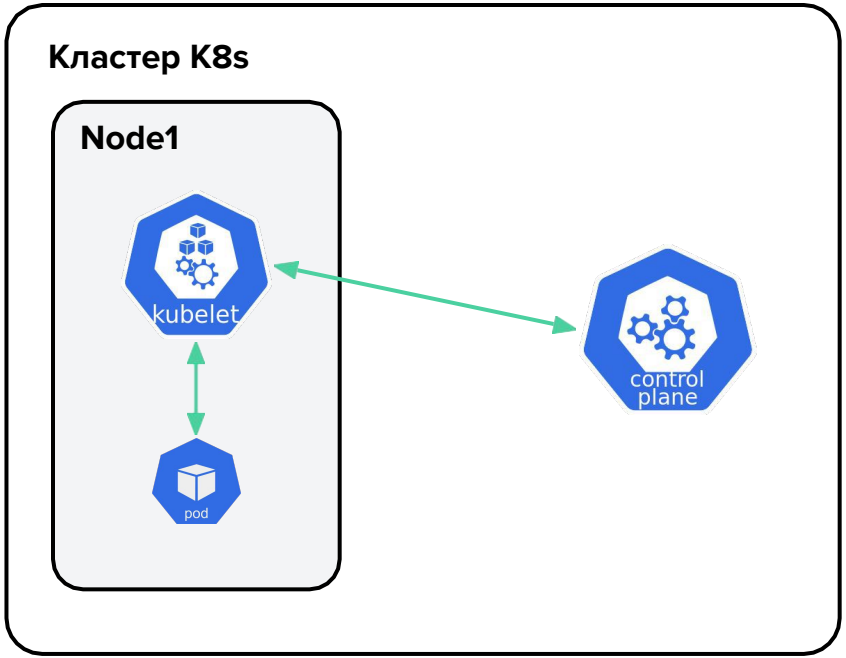
Для отказоустойчивости системы рекомендуется дублировать все компоненты control plane путём выделения отдельных master node и увеличения их количества



Компонентыworker nodes

Компоненты worker nodes: kubelet

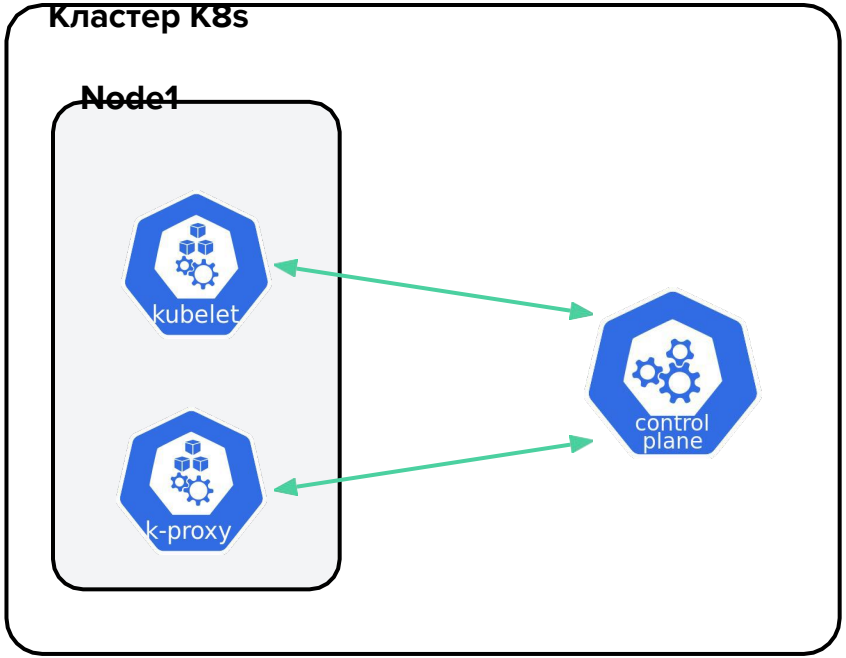
Агент, запущенный на каждой ноде.
Обеспечивает запуск и контроль работы подов на ноде.



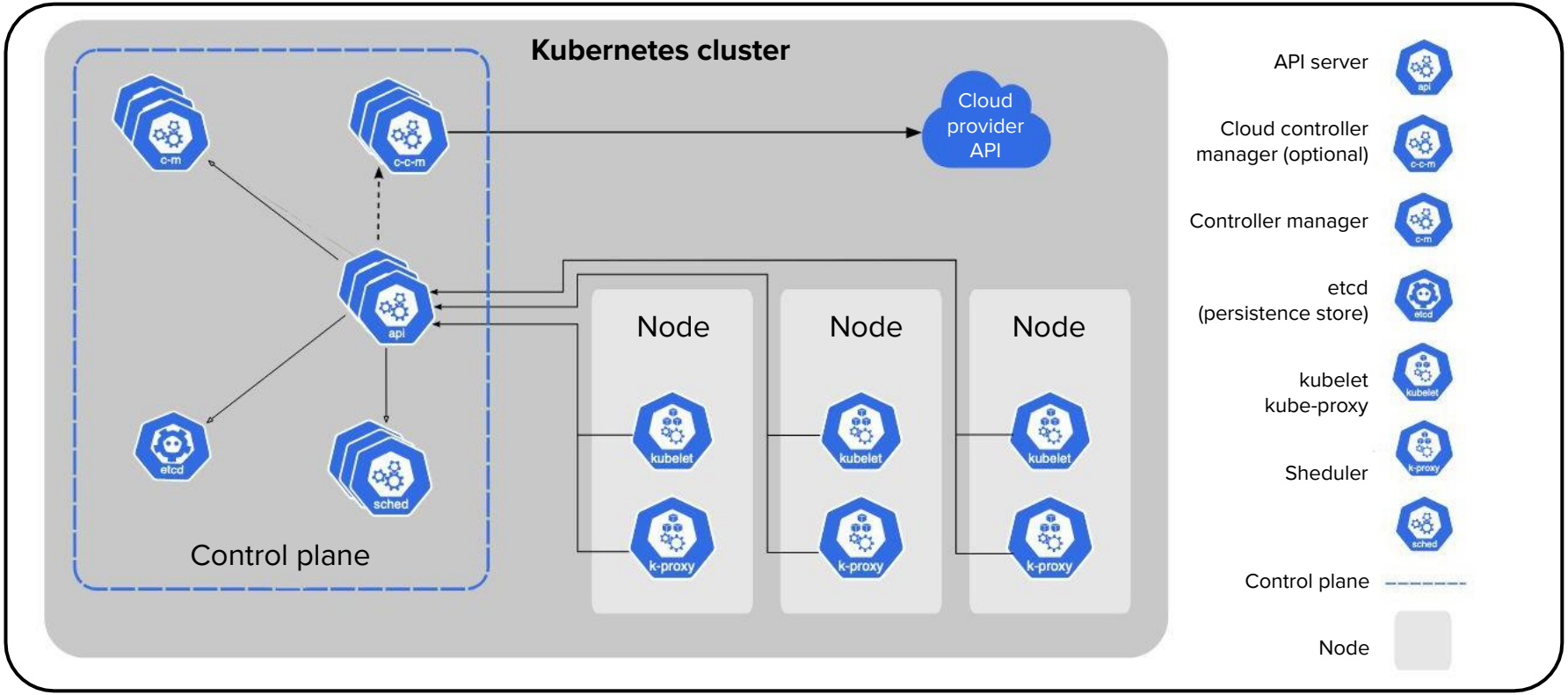
- На входе получает спецификации подов из control plane
- На основании спецификации происходит запуск пода
- Сообщает control plane текущий статус подов

Компоненты worker nodes: kube-proxy

Сетевой прокси, который запущен на каждой ноде. Поддерживает работу сетевых правил на нодах, обеспечивает связность и маршрутизацию трафика



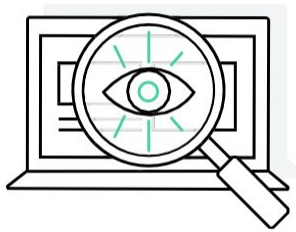
Архитектура кластера K8s: HA



Источник

Демонстрация работы

Компоненты кластера



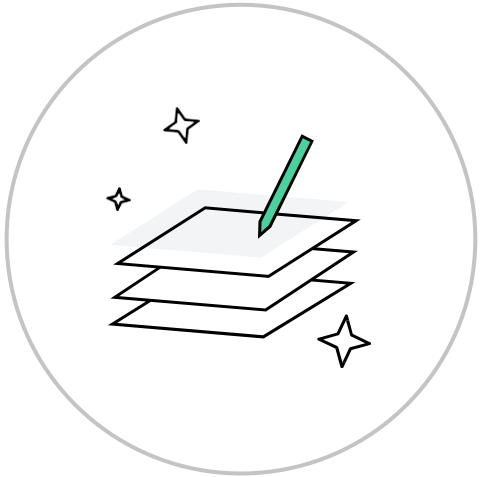
Итоги

- Узнали, из каких компонентов состоит кластер K8s
- Выяснили, какие типы нод существуют
- Поняли, как происходит взаимодействие control plane и worker nodes
- Попробовали подключиться к кластеру и посмотреть в работе объекты, изученные на занятии

Домашнее задание

Давайте посмотрим ваше домашнее задание

1. Вопросы по домашней работе задавайте в чате группы
2. Задачи можно сдавать по частям
3. Зачёт по домашней работе ставят после того, как приняты все задачи



Задавайте вопросы и пишите отзыв о лекции

Герман Никонов
Руководитель DevOps команды