



Введение в Kubernetes

DevOps Bootcamp: Урок 1

🎯 Цель: Понять зачем нужен Kubernetes и запустить свой первый Pod.



Проблема, которую решает Kubernetes

Docker хорош, но...

- Контейнеры нужно запускать вручную
- Нет автоматического восстановления после сбоев
- Невозможно легко масштабировать
- Сложно управлять сетью и конфигурацией

… «А что, если у нас сотни контейнеров на десятках серверов?»

👉 Нужен оркестратор контейнеров — система, которая управляет ими автоматически.



Что такое Kubernetes

- Open Source система для управления контейнерами
- Разработан Google, сейчас поддерживается CNCF
- Позволяет:
 - Автоматически развертывать контейнеры
 - Балансировать нагрузку
 - Масштабировать приложения
 - Обновлять версии без простоя
 - Восстанавливать после сбоев

 Kubernetes = «операционная система для dataцентра»



Архитектура Kubernetes

Cluster = набор узлов (**Nodes**)

Каждый Node — обычный сервер (VM или физический)

- **Control Plane (Master)**

Управляет кластером

- kube-apiserver
- etcd
- kube-scheduler
- controller-manager

- **Worker Nodes**

Выполняют контейнеры

- kubelet



Логика работы

1. Ты создаёшь Pod через `kubectl apply`
2. `kube-apiserver` сохраняет описание в `etcd`
3. `scheduler` выбирает подходящую Node
4. `kubelet` на этой Node запускает контейнер
5. `kube-proxy` организует сетевой доступ

 Всё — декларативно: **ты описываешь состояние, а Kubernetes его поддерживает.**



Основные объекты Kubernetes

Объект	Назначение
Pod	Минимальная единица запуска (1+ контейнера)
ReplicaSet	Поддерживает нужное количество Pod'ов
Deployment	Управляет ReplicaSet и обновлениями
Service	Сеть и балансировка между Pod'ами
Namespace	Логическая изоляция ресурсов
ConfigMap / Secret	Конфигурации и секреты
Ingress	Внешний HTTP/HTTPS-доступ



Аналогия

- **Pod** — квартира
 - **Node** — дом
 - **Cluster** — город
 - **Deployment** — управляющая компания
 - **Service** — система адресации
 - **Ingress** — ворота в город
- 📦 Kubernetes управляет всем «городом» контейнеров.



Практика: Разворачиваем кластер с Kind

1. Устанавливаем инструменты

```
sudo apt install docker.io -y
curl -LO "https://dl.k8s.io/release/v1.30.0/bin/linux/amd64/kubectl"
chmod +x kubectl && sudo mv kubectl /usr/local/bin/
go install sigs.k8s.io/kind@latest
```

2. Создаём кластер

```
kind create cluster --name demo
kubectl get nodes
```



Запускаем первый Pod

1. Создание Pod:

```
kubectl run nginx --image=nginx --port=80  
kubectl get pods
```

2. Проверка:

```
kubectl describe pod nginx  
kubectl port-forward pod/nginx 8080:80  
curl localhost:8080
```

 Если видишь HTML nginx — всё работает!



YAML-манифест Pod

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
    - name: nginx
      image: nginx:latest
      ports:
        - containerPort: 80
```

Разберём:

- `apiVersion` : версия API
- `kind` : тип ресурса
- `metadata` : имя и метки



Работа с kubectl

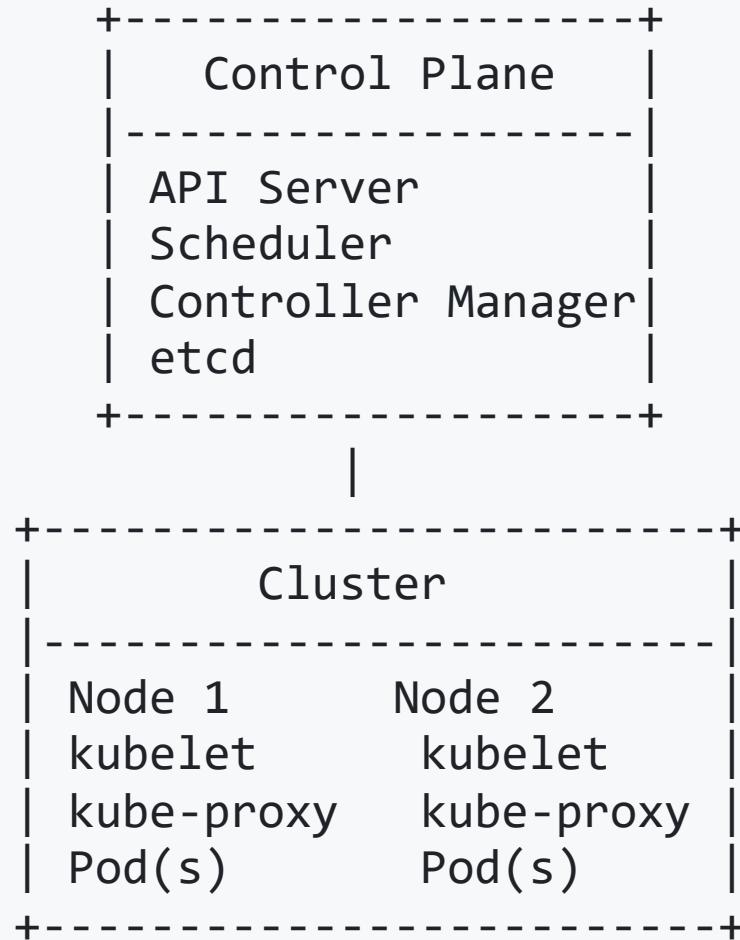
Команда	Назначение
<code>kubectl get pods</code>	Список Pod'ов
<code>kubectl describe pod nginx</code>	Подробности Pod
<code>kubectl logs pod-name</code>	Логи контейнера
<code>kubectl delete pod nginx</code>	Удалить Pod
<code>kubectl apply -f file.yaml</code>	Применить манифест
<code>kubectl explain pod.spec</code>	Подсказка по структуре



`kubectl` — твой интерфейс к кластеру.



Визуализация архитектуры





Резюме урока

Сегодня ты узнал:

- Что такое Kubernetes и зачем он нужен
 - Из чего состоит кластер
 - Что такое Pod и как его запустить
 - Как читать YAML-манифест
 - Основные команды `kubectl`
-  Kubernetes — это не просто инструмент, а философия управления инфраструктурой.



Домашнее задание

1. Установить Minikube или Kind
2. Запустить свой Pod с образом `httpd` или `redis`
3. Создать YAML и применить его:

```
kubectl apply -f mypod.yaml
```

4. Проверить:

```
kubectl get pods  
kubectl describe pod mypod
```

🌟 Сделай скрин с выводом `kubectl get pods` и отправь в чат буткемпа.



Дополнительные ресурсы

- [🌐 kubernetes.io/docs/home](https://kubernetes.io/docs/home)
- [🎥 YouTube: *TechWorld with Nana — Kubernetes in 5 mins*](#)
- [🧪 Katacoda Kubernetes labs](#)
- [📘 Книга: *The Kubernetes Book — Nigel Poulton*](#)



Итог

Kubernetes — это способ управлять хаосом контейнеров.
Он автоматизирует всё, что раньше приходилось делать вручную.

⚡ На следующем уроке: **Deployment, ReplicaSet и Service** —
разберём, как управлять обновлениями и балансировкой.