

Kubernetes

Причины появления. Команда kubectl

Кирилл Касаткин



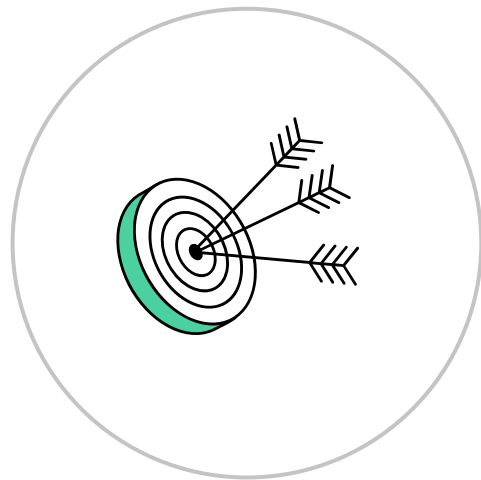


Кирилл Касаткин

DevOps-инженер, Renue

Цели занятия

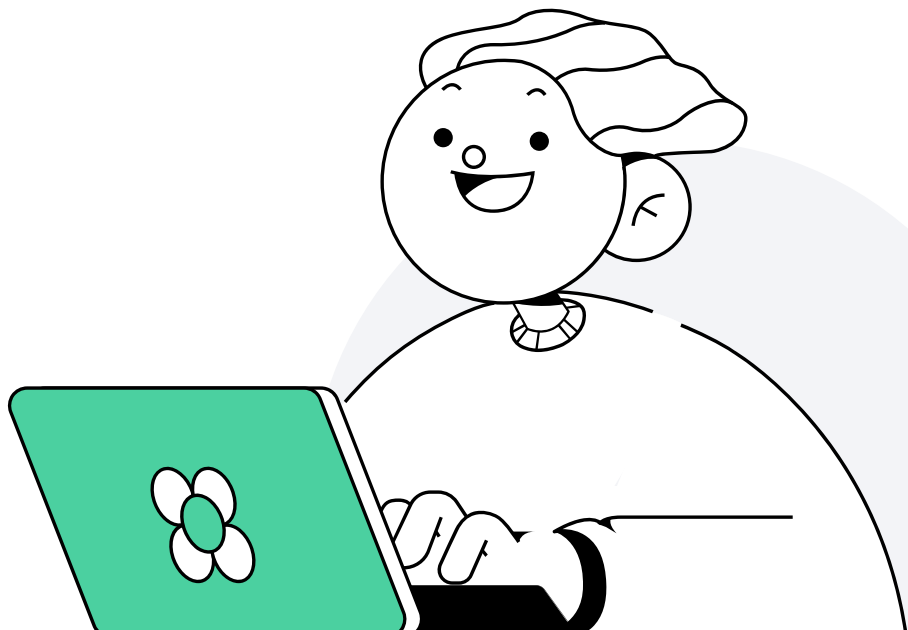
- Познакомиться с причинами появления Kubernetes
- Обсудить проблемы развития приложения:
 - версионирование
 - масштабирование
- Познакомиться с командой kubectl и научиться получать информацию о кластерах Kubernetes



План занятия

- 1 Что такое Kubernetes
- 2 Решения для запуска на локальной машине
- 3 Команда kubectl
- 4 Демонстрация работы
- 5 Итоги
- 6 Домашнее задание

*Нажми на нужный раздел для перехода




Что такое Kubernetes

Какие проблемы решает и какие задачи выполняет



1



**Kubernetes — это open-source-
платформа контейнерной
оркестрации**



Что такое K8S

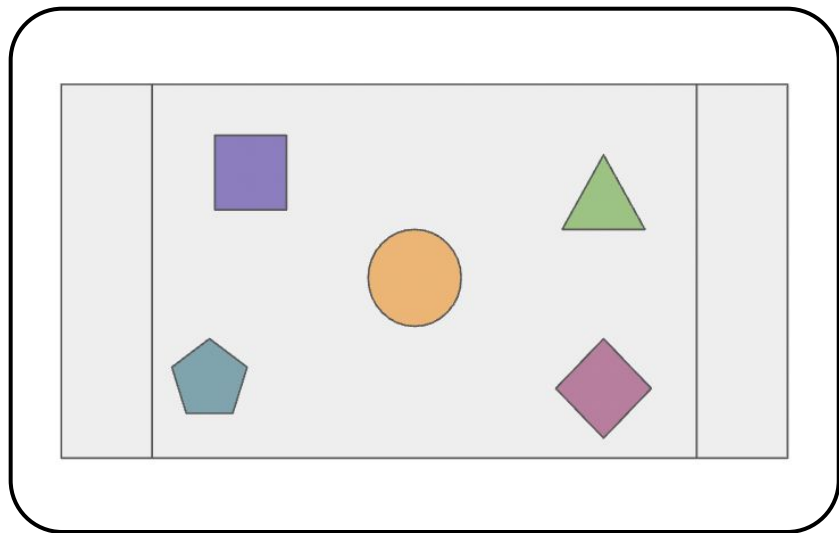


Что такое K8S

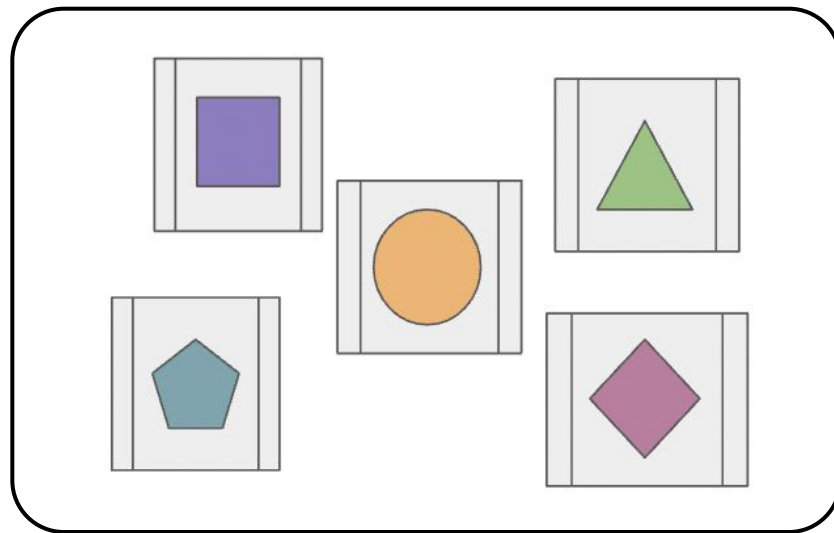
8 — количество букв между K и S в слове «Kubernetes». Иногда для простоты вместо Kubernetes используют K8S

Вспомним предыдущие занятия

Переход от монолитной к микросервисной архитектуре



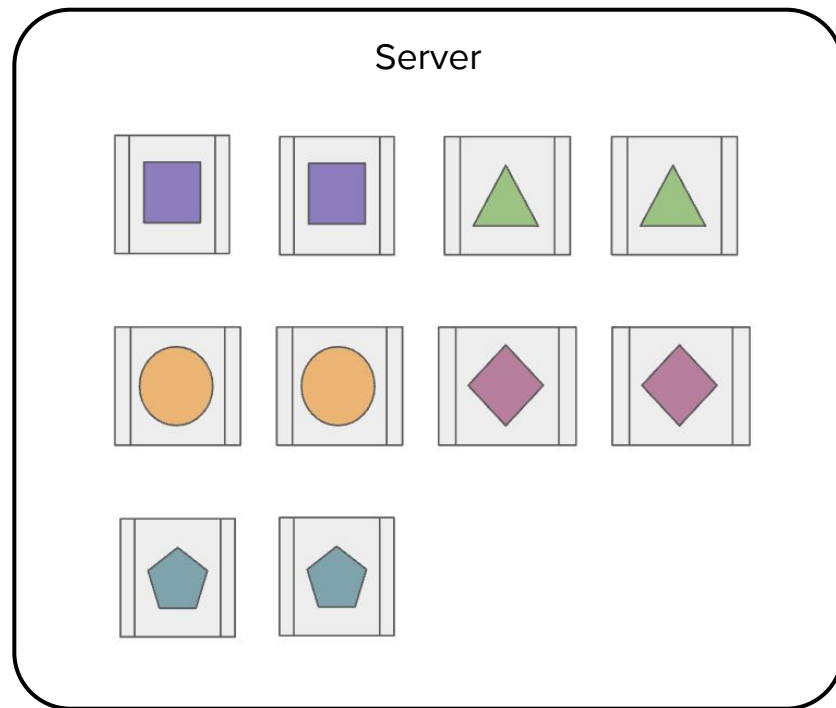
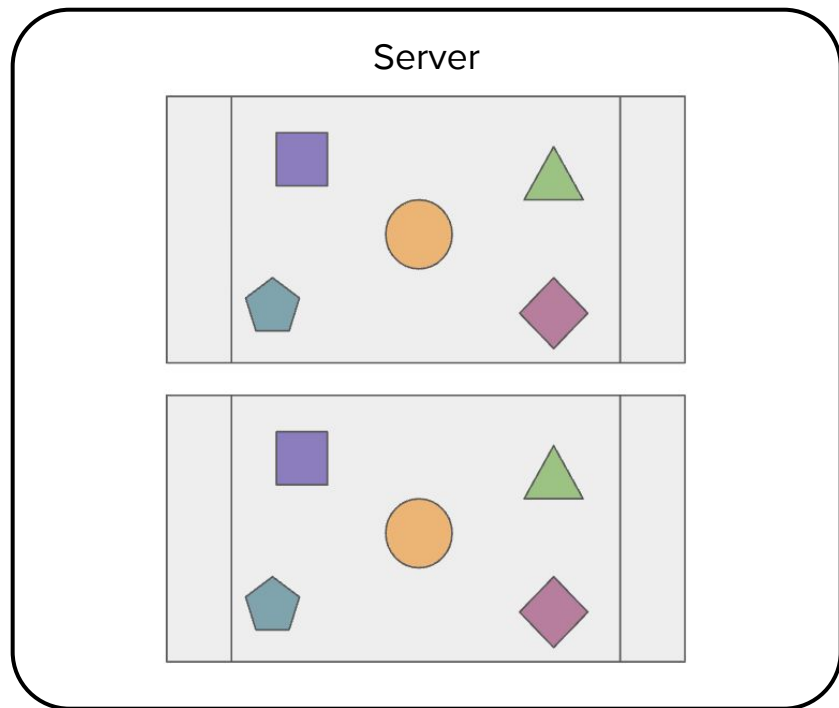
Монолитное приложение содержит все бизнес-функции в **одном процессе**



Микросервисы распределяют бизнес-функции по разным независимым сервисам, запущенным в **отдельных процессах**

Если нагрузка увеличилась

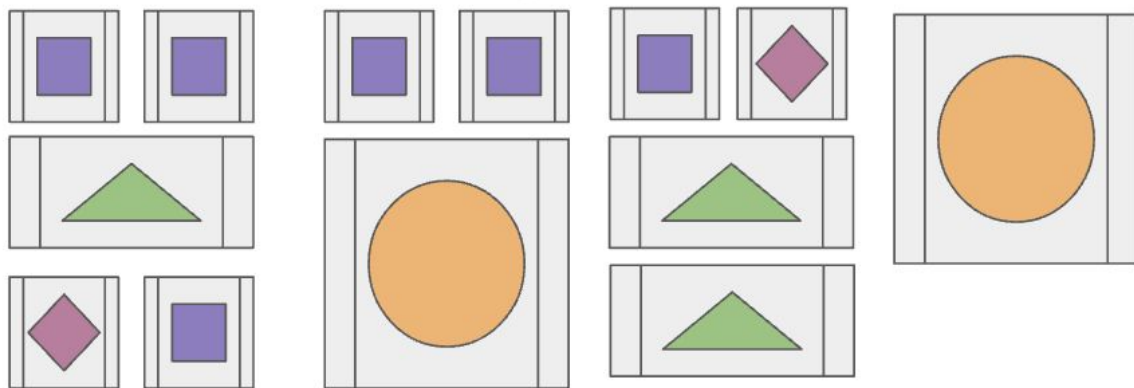
Масштабируемость приложений



Один большой сервер. Проблемы

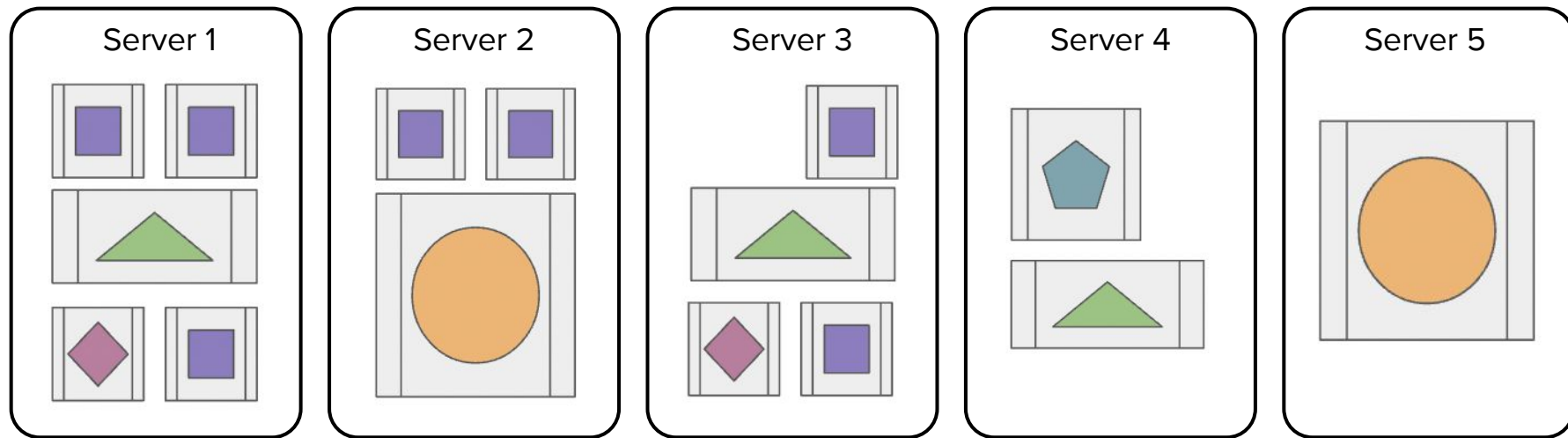
- 1 Единая точка отказа
- 2 Дорого
- 3 Сложно масштабировать (особенно вниз)

Big Server



Несколько серверов

Сервисы можно упаковать в контейнеры и распределить между серверами в зависимости от потребностей



Несколько серверов. Проблемы

- 1 Сложно поддерживать отдельные серверы для разных приложений:
CPU, RAM, Storage
- 2 Неоптимально по стоимости и производительности
- 3 Сложно масштабировать сервисы вручную по отдельным серверам



**Кластер — группа компьютеров или нод (node),
объединённых высокоскоростными каналами связи,
представляющая единый аппаратный ресурс, с точки
зрения пользователя**

Преимущества Kubernetes

- 1 **High Availability (HA)** — высокая доступность приложения, запущенного в нескольких экземплярах
- 2 **Scalability** — масштабируемость за счёт скорости развёртывания контейнеров при увеличении нагрузки
- 3 **Disaster Recovery (DR)** — отказоустойчивость за счёт распределённого количества серверов кластера

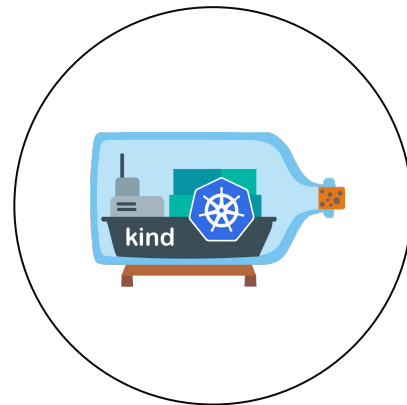
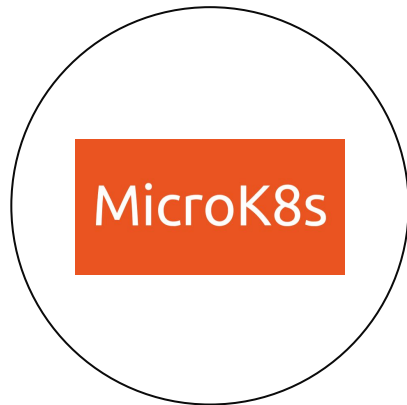
Решения для запуска на локальной машине



2

Решения для запуска на локальной машине

Запуск на локальной машине позволяет легко начать работу с Kubernetes. Можно создавать и тестировать кластер Kubernetes, не беспокоясь о трате облачных ресурсов и квотах



Kubectrl



3

Программное обеспечение kubectl

kubectl — это инструмент командной строки для управления кластерами Kubernetes.

kubectl может быть установлен отдельно на локальной машине. Поддерживается Linux, macOS, Windows

По умолчанию используется файл *config* в директории *\$HOME/.kube*, который содержит всю необходимую информацию для подключения к кластеру

Подробная [инструкция](#) по установке, [обзор](#) возможностей и краткая [шпаргалка](#)

Структура файла config

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority: certs/cm.crt or base64-encoded-certificate
    server: https://address_to_cluster:6443
    name: prod
contexts:
- context:
    cluster: prod
    namespace: processing
    user: admin
    name: prod
current-context: stage
kind: Config
preferences: {}
users:
- name: admin
  user:
    client-certificate: certs/admin.crt or base64-encoded-certificate
    client-key: certs/admin.key or base64-encoded-certificate
```

Структура файла config

В файле с конфигурацией есть три интересные секции:

- **clusters**
Информация о кластерах, включая сертификаты доступа к кластеру
- **users**
Информация о пользователях, включая сертификаты доступа
- **contexts**
Это сочетание cluster, user и namespace

Команды для работы с конфигом (kubectl config)

Список кластеров

```
kubectl config get-clusters
```

Текущий контекст

```
kubectl config current-context
```

Список пользователей

```
kubectl config get-users
```

Переключение текущего контекста

```
kubectl config use-context name
```

Список контекстов

```
kubectl config get-contexts
```

Установка доступов

```
kubectl config set-credentials
```

Получение информации о ресурсе кластера

Получение информации о всех нодах кластера

```
kubectl get nodes
```

Получение дополнительной информации о всех нодах кластера

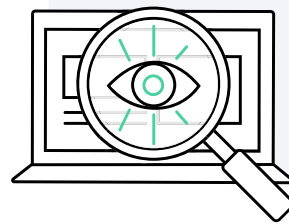
```
kubectl get nodes -o wide
```

Получение детальной информации о конкретной ноде

```
kubectl describe nodes node_name
```

Демонстрация работы

Работа с kubectl на примере microK8S и реального кластера K8S



Итоги

Сегодня мы

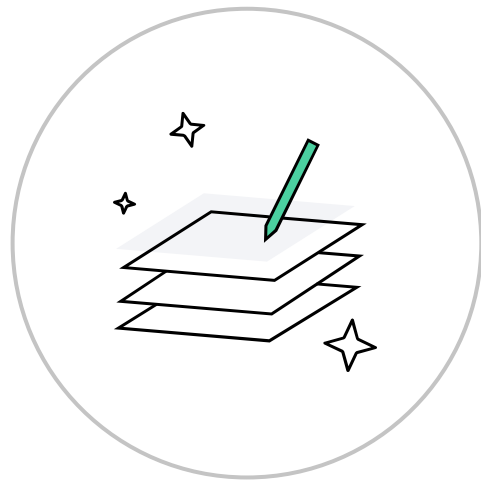
- 1 Узнали, что такое K8S
- 2 Разобрались, с помощью каких механизмов обеспечивается масштабируемость приложений
- 3 Рассмотрели варианты решений для запуска песочницы
- 4 Попробовали подключиться к кластеру и посмотреть информацию о нём



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



Дополнительные материалы

- [Инструкция](#) по установке MicroK8s
- Подробная [инструкция](#) по установке kubectl
- [Обзор](#) возможностей kubectl
- Краткая [шпаргалка](#) по kubectl



Задавайте вопросы и пишите отзыв о лекции

Кирилл Касаткин

