

Хранение в K8s. Часть 1

Volumes, Shared volumes

Кирилл Касаткин
DevOps-инженер, Renue



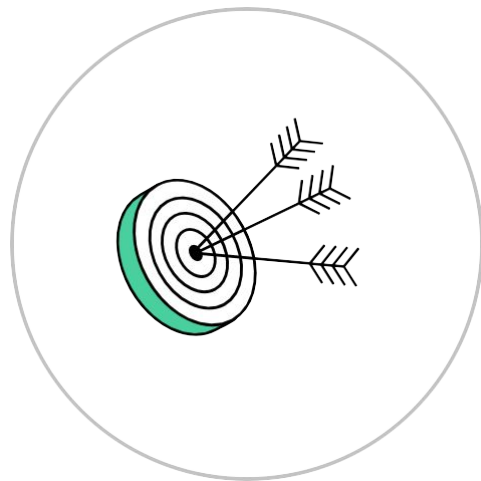
Кирилл Касаткин

DevOps-инженер, Renuе



Цели занятия

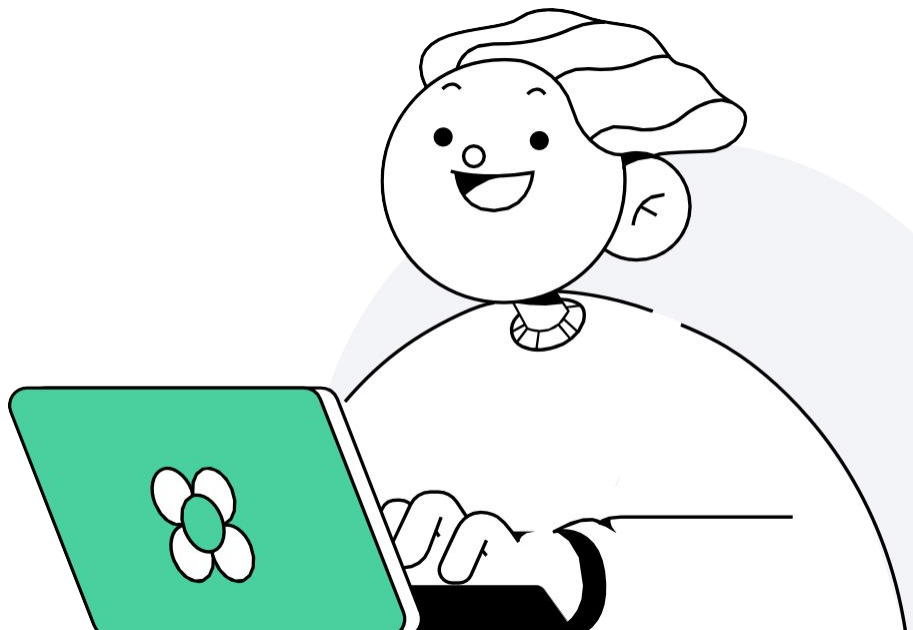
- Узнать:
 - что такое volume (том) и для чего он необходим
 - где фактически находятся сохраняемые контейнерами файлы
- Познакомиться с типами volume
- Понять, как можно использовать общие данные между контейнерами внутри пода
- Разобрать примеры манифестов объектов K8s



План занятия

- 1 Файловая система контейнера
- 2 Volume
- 3 Shared volumes
- 4 Типы volume
- 5 Итоги
- 6 Домашнее задание

*Нажми на нужный раздел для перехода



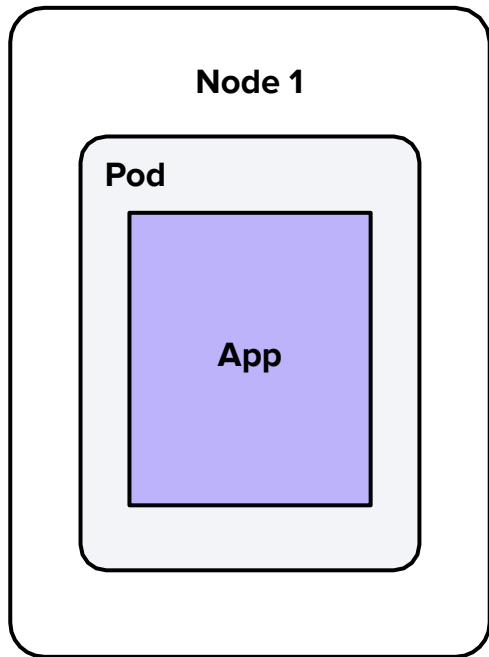
Файловая система контейнера



1

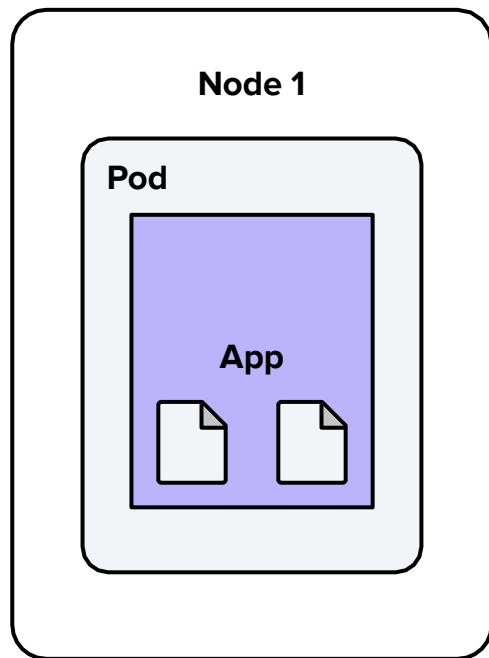
Файловая система контейнера в Pod

Файловая система контейнера **эфемерна**.



Файловая система контейнера в Pod

Файловая система контейнера **эфемерна**.

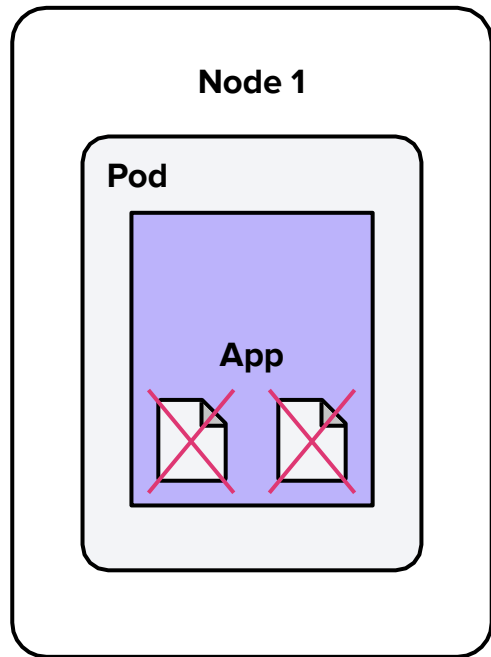


→ Файлы в контейнере существуют, пока существует контейнер



Файловая система контейнера в Pod

Файловая система контейнера **эфемерна**.



- Файлы в контейнере существуют, пока существует контейнер
- С удалением или перезапуском контейнера все изменения файловой системы исчезнут



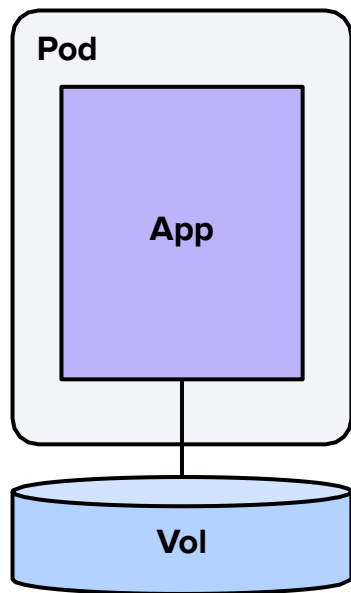
Volume



2

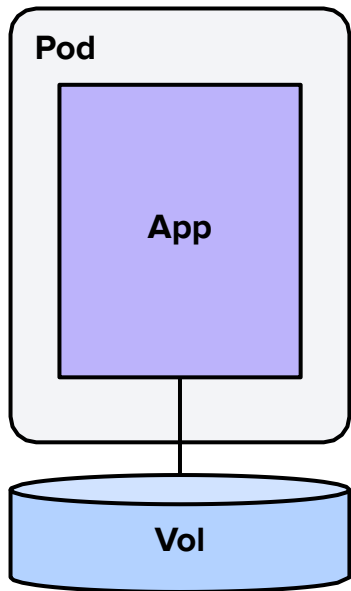
Volume

Volume (том) — директория, хранящаяся за пределами пода.



Volume

Volume (том) — директория, хранящаяся за пределами пода.

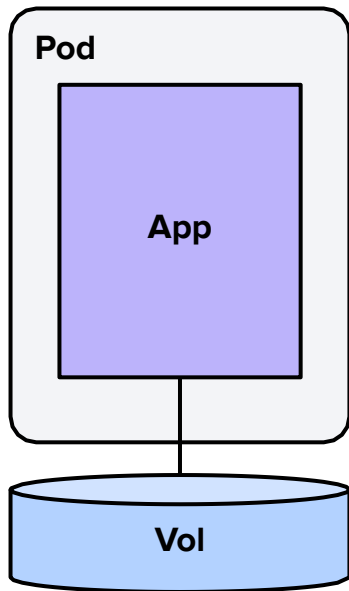


- Том объявляется на уровне пода и монтируется внутрь нужного контейнера



Volume

Volume (том) — директория, хранящаяся за пределами пода.

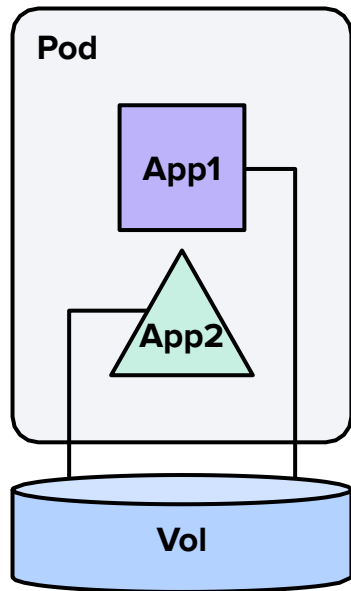


- Том объявляется на уровне пода и монтируется внутрь нужного контейнера
- Путь монтирования объявляется на уровне отдельного контейнера



Volume

Volume (том) — директория, хранящаяся за пределами пода.



- Том объявляется на уровне пода и монтируется внутри нужного контейнера
- Путь монтирования объявляется на уровне отдельного контейнера
- Если в поде несколько контейнеров, то этот путь для каждого из них может различаться



Пример конфигурации

Пример конфигураций Pod с подключением volume

```
apiVersion: v1
kind: Pod
metadata:
  name: volume-pod
spec:
  containers:
    - name: app1
      image: busybox
      command: ['sh', '-c', 'echo Success! > /output/success.txt']
      volumeMounts:
        - name: vol
          mountPath: /output
  volumes:
    - name: vol
      hostPath:
        path: /var/data
```



Пример конфигурации

Пример конфигураций Pod с подключением volume

```
apiVersion: v1
kind: Pod
metadata:
  name: volume-pod
spec:
  containers:
    - name: appl
      image: busybox
      command: ['sh', '-c', 'echo Success! > /output/success.txt']
      volumeMounts:
        - name: vol
          mountPath: /output
  volumes:
    - name: vol
      hostPath:
        path: /var/data
```



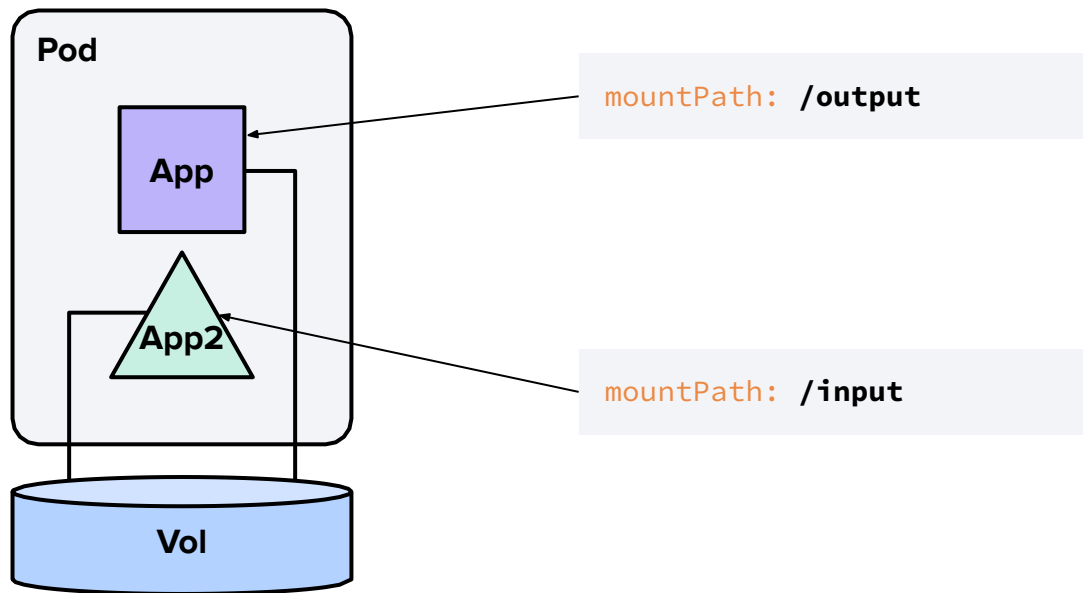
Shared volumes



3

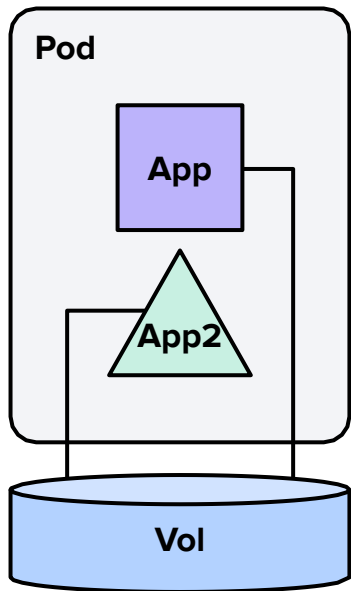
Shared volume между контейнерами в поде

Volume можно использовать как общую папку между контейнерами внутри пода



Shared volume между контейнерами в поде

Volume можно использовать как общую папку между контейнерами внутри пода



```
apiVersion: v1
kind: Pod
metadata:
  name: volume-pod
spec:
  containers:
  - name: app1
    image: busybox
    command: ['sh', '-c', 'echo Success! > /output/success.txt']
    volumeMounts:
    - name: vol
      mountPath: /output
  - name: app2
    image: busybox
    command: ['sh', '-c', 'while true; do cat /input/success.txt']
    volumeMounts:
    - name: vol
      mountPath: /input
  volumes:
  - name: vol
    hostPath:
      path: /var/data
```



Типы volume



4

Типы volume

Существует множество типов volume. Условно их можно разделить на:

- **локальные**, расположенные непосредственно на ноде, где находится под
- остальные, определяемые с помощью Persistent Volume (будет рассмотрено в следующей лекции)



Типы volume

Существует множество типов volume. Условно их можно разделить на:

- **локальные**, расположенные непосредственно на ноде, где находится под
- остальные, определяемые с помощью Persistent Volume (будет рассмотрено в следующей лекции)

Наиболее часто встречаемые **локальные типы**:

- emptyDir
- hostPath



Тип volume emptyDir

Создаётся и удаляется вместе с подом, но при сбое и его дальнейшем самовосстановлении (перезапуске) данные из этого volume не удаляются.

При этом, если под был удалён администратором и затем автоматически перезапущен, данные из volume исчезнут.

```
volumes:  
- name: my-volume  
  emptyDir: {}
```



Тип volume emptyDir

Создаётся и удаляется вместе с подом, но при сбое и его дальнейшем самовосстановлении (перезапуске) данные из этого volume не удаляются.

При этом, если под был удалён администратором и затем автоматически перезапущен, данные из volume исчезнут.

Обычно используется для:

- данных, которые необходимо сохранить при сбое в работе контейнера: например, кеш-файлов
- обмена файлами между несколькими контейнерами в поде

```
volumes:  
- name: my-volume  
  emptyDir: {}
```



Тип volume hostPath

Используется, когда необходимо предоставить доступ к локальной файловой системе ноды. Определяет путь, где будут храниться данные

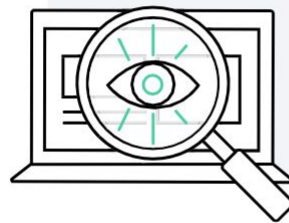
Файлы на хостах доступны для записи только подам, запущенным с правами пользователя root

```
volumes:  
- name: my-volume  
  hostPath:  
    path: /var/data
```



Демонстрация работы

Работа с volume в Pod



Итоги

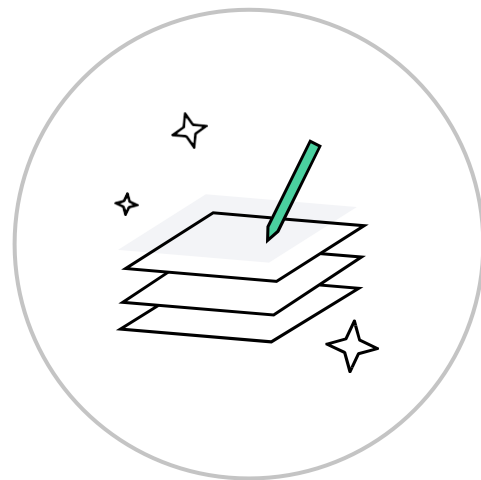
- 1 Узнали, что такое volume
- 2 Выяснили, какие типы volume бывают
- 3 Поняли, как можно использовать общее хранилище между контейнерами в поде
- 4 Рассмотрели примеры манифестов объектов K8s
- 5 Попробовали подключиться к кластеру и посмотреть в работе объекты, изученные на занятии



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

Кирилл Касаткин
DevOps-инженер, Renue

