



Методическое пособие
по выполнению домашнего
задания курса
**Инфраструктурная платформа на
основе Kubernetes**

Volumes, StorageClass, PV, PVC

Содержание

| | | |
|----|---|----|
| 1. | Введение | 3 |
| 2. | Цели домашнего задания | 4 |
| 3. | Описание домашнего задания | 5 |
| 4. | Пошаговая инструкция выполнения домашнего задания | 7 |
| 5. | Сдача задания | 8 |
| 6. | Критерии оценки | 9 |
| 7. | Рекомендуемые источники | 10 |

1. Введение

Persistent Volumes (PV)

Ресурсы хранения в кластере Kubernetes, которые используют поды. PV существуют независимо от подов — их цикл жизни определяет Политика Возврата (Reclaim Policy). В зависимости от Политики возврата, PV может быть сохранён или удалён после удаления пода, который был привязан к этому PV.

Persistent Volume Claim (PVC)

Заявка на получение определенного ресурса хранилища (PV). После создания привязывается к подходящему по параметрам ресурсу PV, либо PV создается автоматически под конкретную заявку.

В задании мы рассмотрим второй сценарий.

StorageClass

Определяет классы хранилищ, поддерживаемые в кластере. Также описывает механизм provisioner для создания объектов PV автоматически.

Мы рассмотрим в задании создание собственного класса хранилища.

2. Цели домашнего задания



- 1) Научиться создавать в кластере объекты, описывающие персистентные хранилища и научиться подключать их к подам.
- 2) Научиться создавать объект ConfigMap и монтировать его как volume
- 3) Получить представление об объекте StorageClass и механизме provisioning для PV



3. Описание домашнего задания



В данном домашнем задании мы научимся работать с **volume**, **PV** и **PVC**.

Подготовка к выполнению домашнего задания

- Создайте branch `kubernetes-volumes`. Данное домашнее задание будет выполняться в этой ветке.
- Создайте папку `kubernetes-volumes`. Скопируйте в нее манифесты из предыдущего ДЗ. Все новые файлы, которые у вас получатся во время выполнения данного ДЗ необходимо также поместить в эту папку.

Рекомендуемые источники



- Документация по volumes
- Документация по PV и PVC
- Документация по storageClass
- Документация по configMap
- Для задания с * и создания кастомного storageClass можно взять за образец storageClass по-умолчанию. Посмотреть его конфигурацию можно например так `kubectl get sc standard -o yaml`

4. Пошаговая инструкция выполнения домашнего задания



- Создать манифест `pvc.yaml`, описывающий PersistentVolumeClaim, запрашивающий хранилище с storageClass по-умолчанию
- Создать манифест `cm.yaml` для объекта типа configMap, описывающий произвольный набор пар ключ-значение
- В манифесте `deployment.yaml` изменить спецификацию volume типа emptyDir, который монтируется в init и основной контейнер, на pvc, созданный в предыдущем пункте
- В манифесте `deployment.yaml` добавить монтирование ранее созданного configMap как volume к основному контейнеру пода в директорию `/homework/conf`, так, чтобы его содержимое можно было получить, обратившись по url `/conf/file`

Задание с *

- Создать манифест `storageClass.yaml` описывающий объект типа storageClass с provisioner <https://k8s.io/minikube-hostpath> и reclaimPolicy Retain
- Изменить манифест `pvc.yaml` так, чтобы в нем запрашивалось хранилище созданного вами storageClass-а

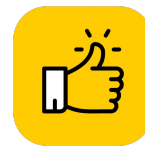


5. Сдача задания



- Добавьте все получившиеся файлы в ветку **kubernetes-volumes**
- Создайте Pull Request к ветке master
- Заполните описание PR по шаблону
- **Не мерджите** PR самостоятельно
- Если у вас возникли вопросы при выполнении ДЗ и необходима консультация преподавателей – добавьте к PR метку **Review Required**
- В личном кабинете Otus сдайте ДЗ на проверку, указав ссылку на Pull Request

6. Критерий оценивания



- 0 баллов – задание не выполнено или выполнено не полностью
- 1 балл – выполнены полностью все основные задания
- 2 балла – выполнены полностью также все задания с *

7. Рекомендуемые источники



- Документация по volumes
- Документация по PV и PVC
- Документация по storageClass
- Документация по configMap
- Для задания с * и создания кастомного storageClass можно взять за образец storageClass по-умолчанию. Посмотреть его конфигурацию можно например так `kubectl get sc standard -o yaml`