МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное

бюджетное образовательное учреждение

высшего образования

**«Московский технический университет связи и информатики»**

**(МТУСИ)**

Кафедра

«Сетевые информационные технологии и сервисы»

Лабораторная работа 5

По дисциплине «Архитектура ЦОД»

«Хранилище и приложения Kubernetes»

Выполнили студенты М092301(75):

Михайлов М.Р.

Чернышов Д.В.

Леонов Н.Н.

Проверил:

Ст.пр. Фатхулин Т.Д.

Москва 2024

**1. Цель лабораторной работы.**

Целью данной лабораторной работы является узнать о различных классах хранилищ, типах томов и моментальных снимках в Kubernetes, а также о том, как работает резервное копирование и восстановление приложений Kubernetes с помощью Kasten K10 от Veeam.

**2. Выполнение лабораторной работы.**

Прохождение тестовой части:

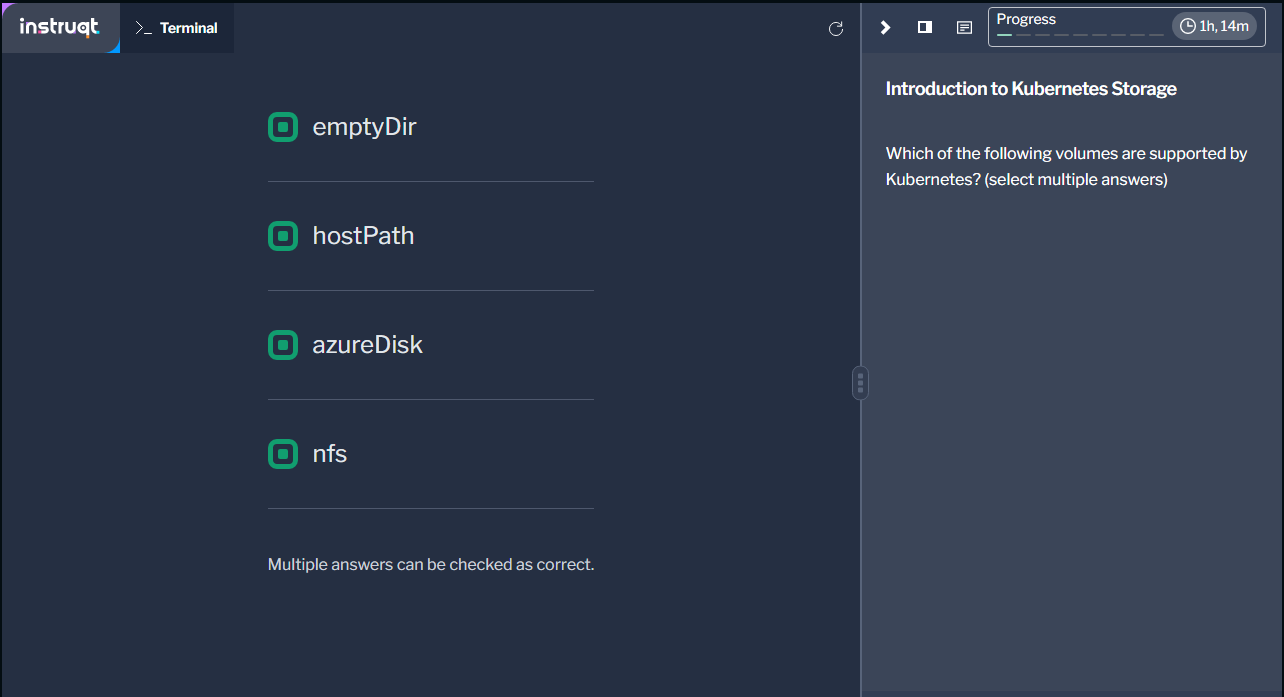


Рисунок 1 – Вопрос 1

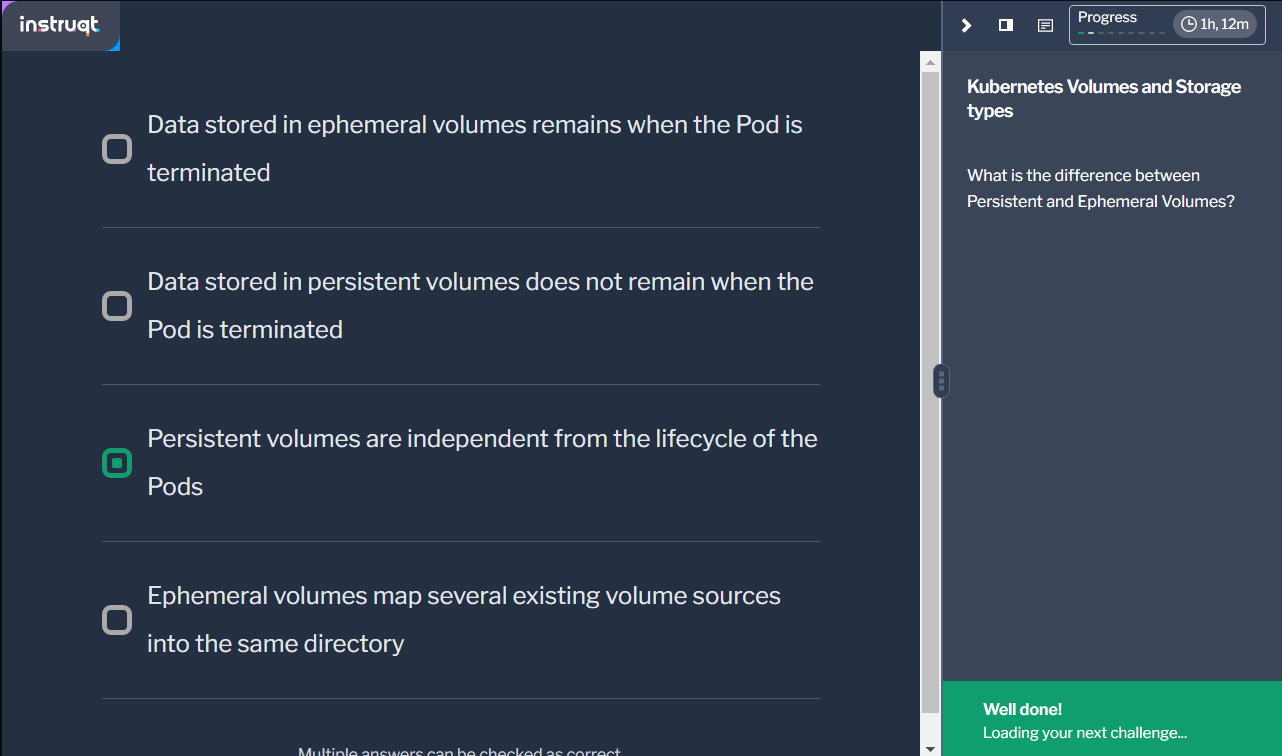


Рисунок 2 – Вопрос 2

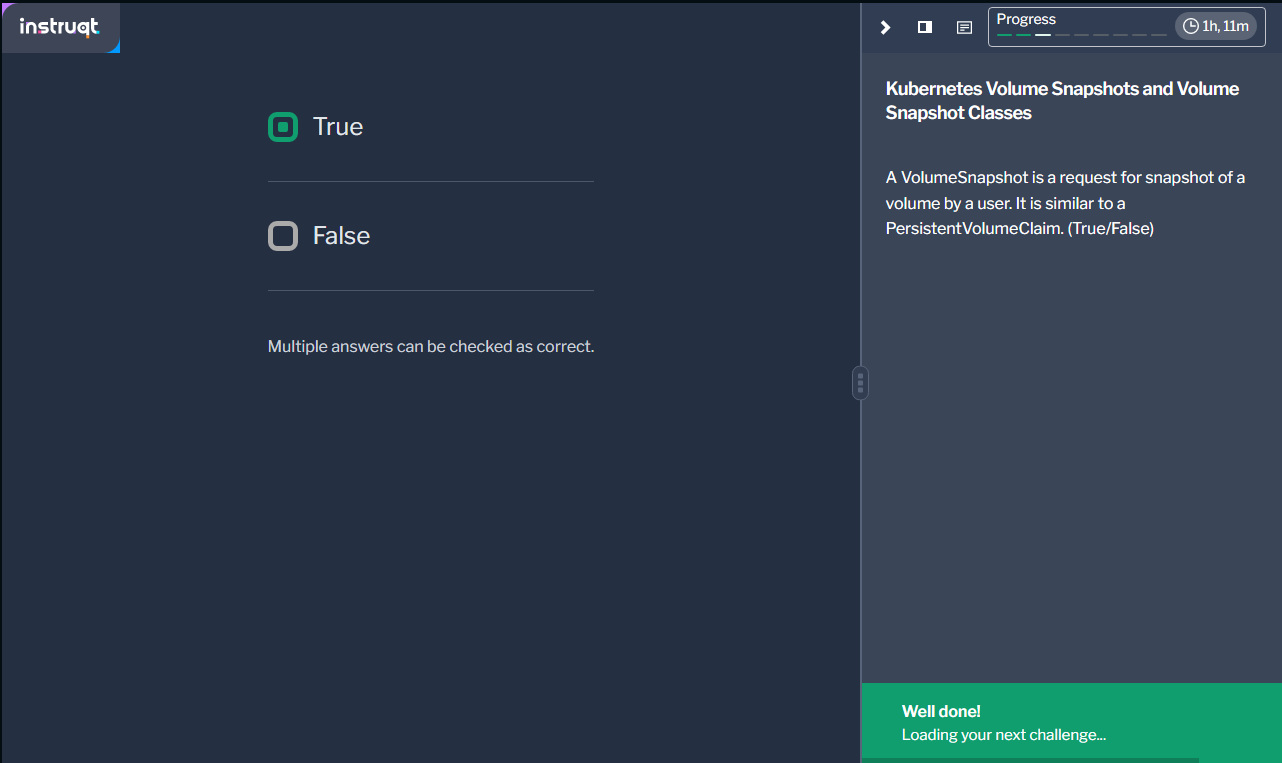


Рисунок 3 – Вопрос 3

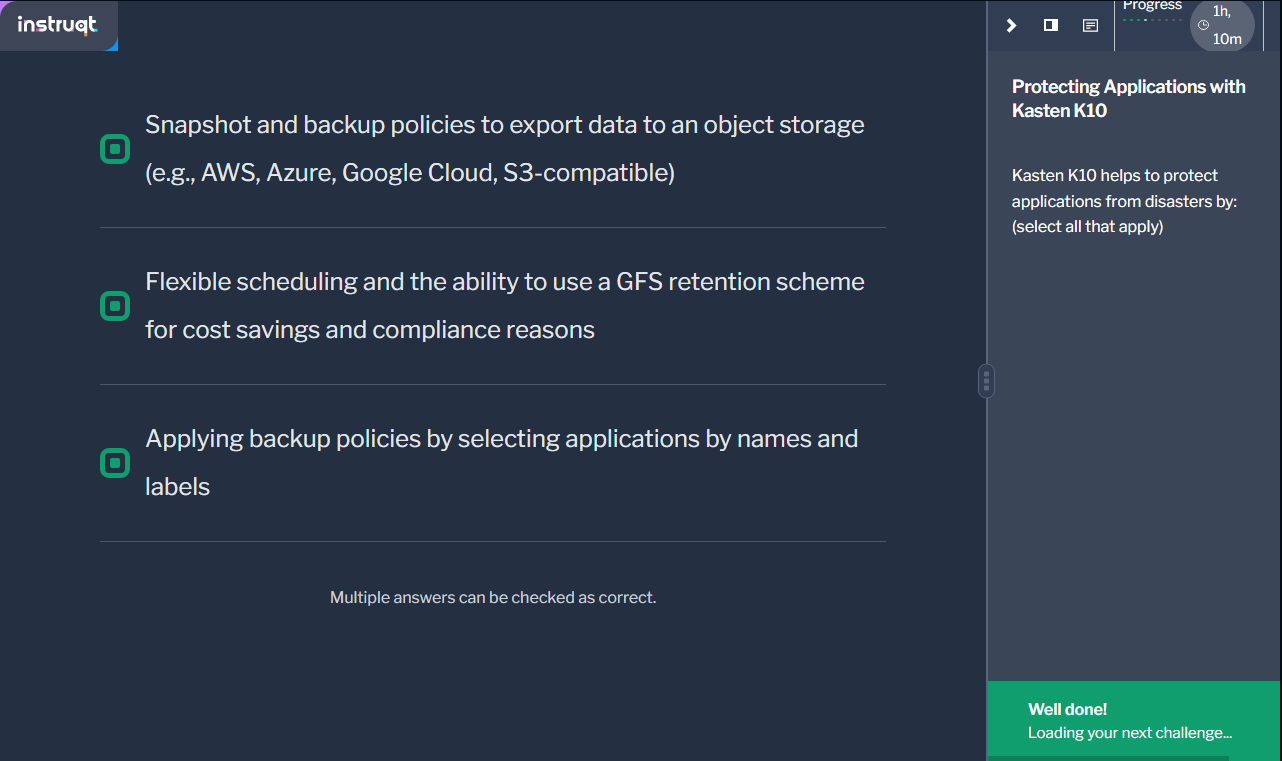


Рисунок 4 – Вопрос 4

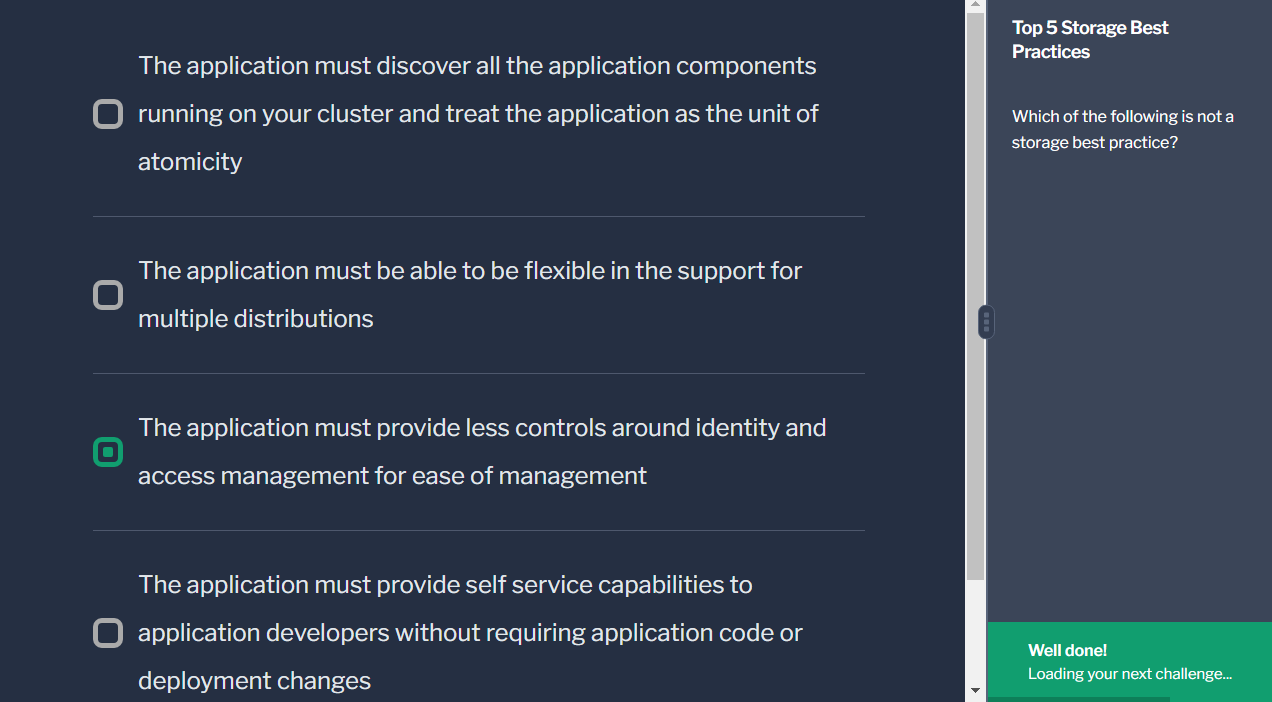


Рисунок 5 – Вопрос 5

Выполнение практической части:

Создание эфемерного хранилища emptyDir.

Монтирование тома происходит таким образом, чтобы теперь веб-сервер обслуживал контент, сгенерированный генератором контента. Тома ephemeral следуют за жизненным циклом пода и создаются, и удаляются вместе с подом (рис. 6):

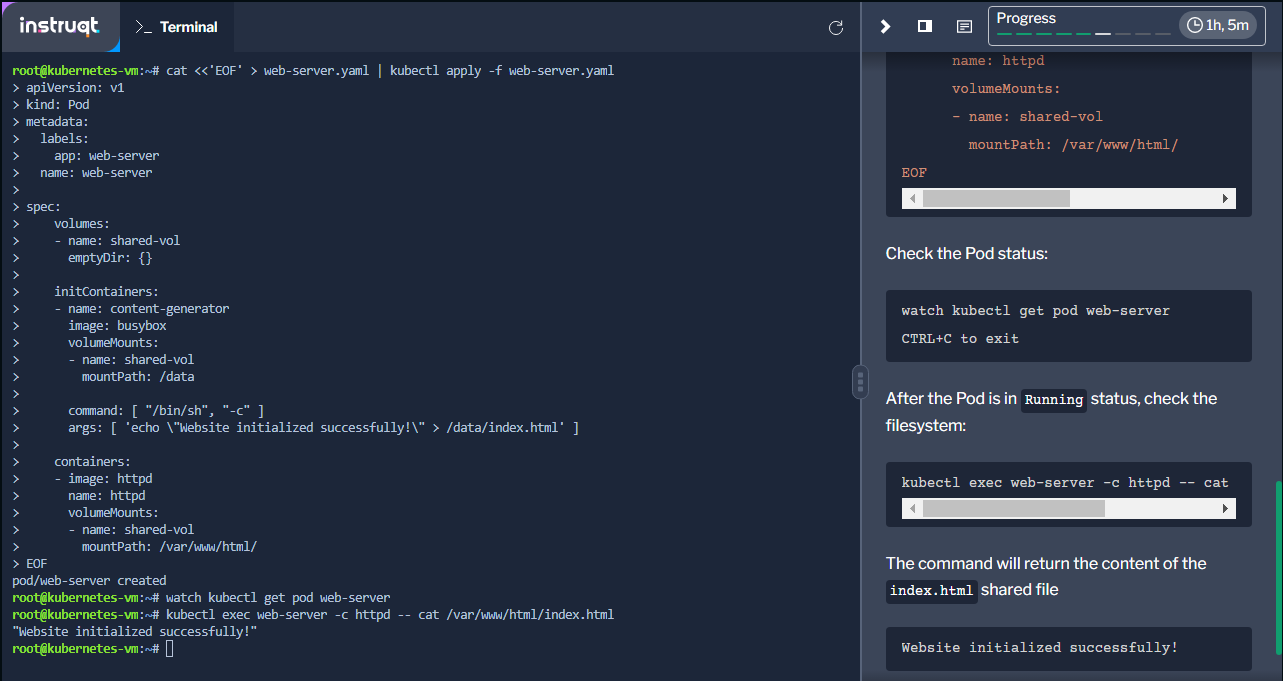


Рисунок 6 - Создание пода с хранилищем emptyDir

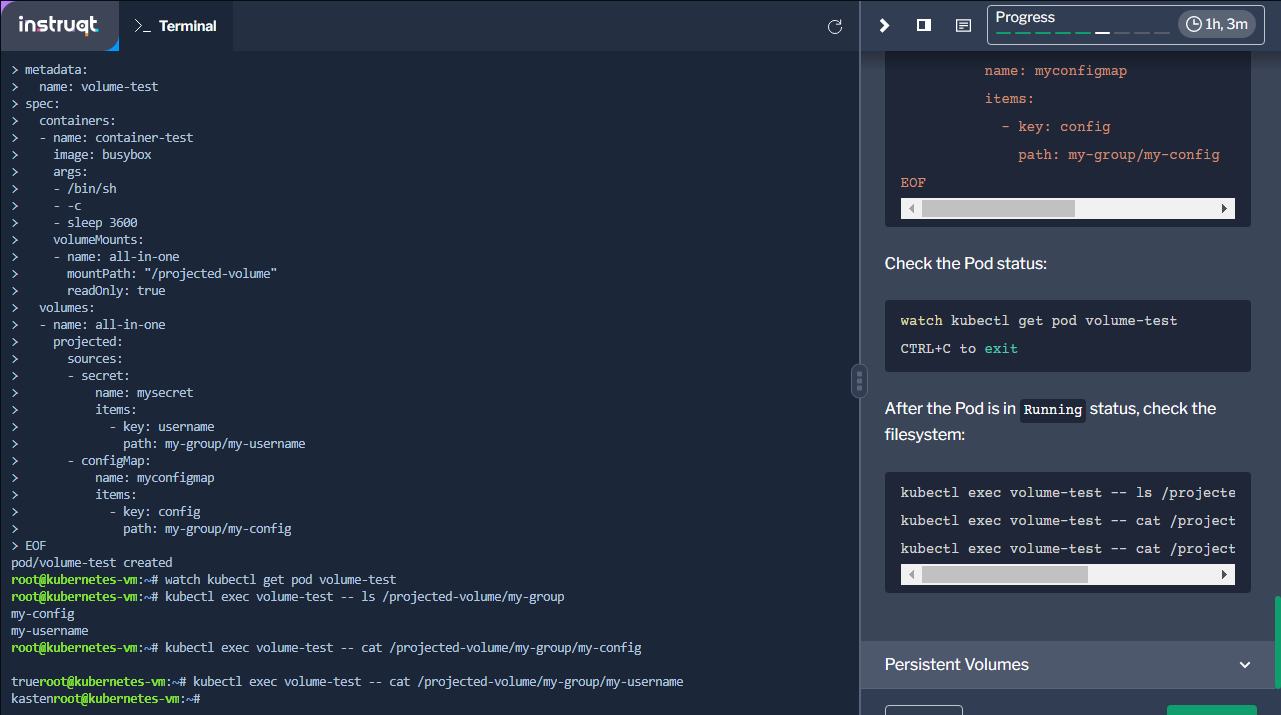


Рисунок 7 - Создание пода с projected-хранилищами

PersistentVolume (PV) — это часть хранилища в кластере, которая была предоставлена ​​администратором или динамически предоставлена ​​с использованием классов хранения. Это ресурс в кластере и его жизненный цикл не зависит от любого отдельного модуля, который использует PV.

PersistentVolumeClaim (PVC) — это запрос пользователя на хранение. Он похож на под. Поды потребляют ресурсы узлов, а PVC — ресурсы PV. Поды могут запрашивать определенные уровни ресурсов (ЦП и памяти). PVC могут запрашивать определенный размер и режимы доступа.

Далее создан PersistentVolume размером 10Gi с названием myvolume. Проверим его статус с помощью команды -kubectl get pv myvolume (рисунок 8):

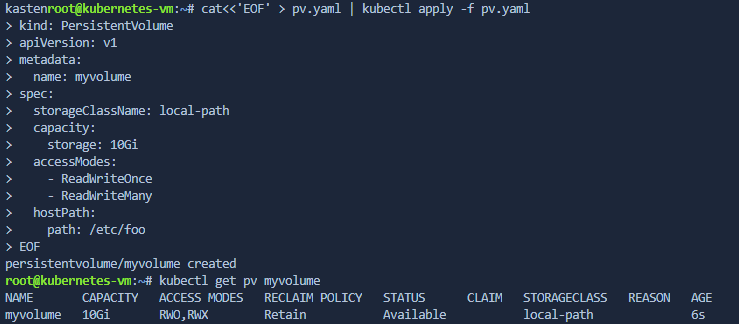


Рисунок 8 – Создание PV

Создадим PVC и проверим статус с помощью команды

-kubectl get pvc mypvc

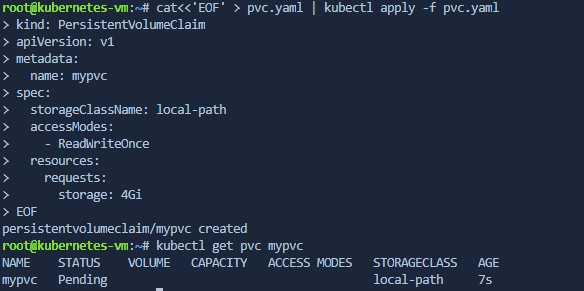


Рисунок 9 – Создание PVC и проверка статуса

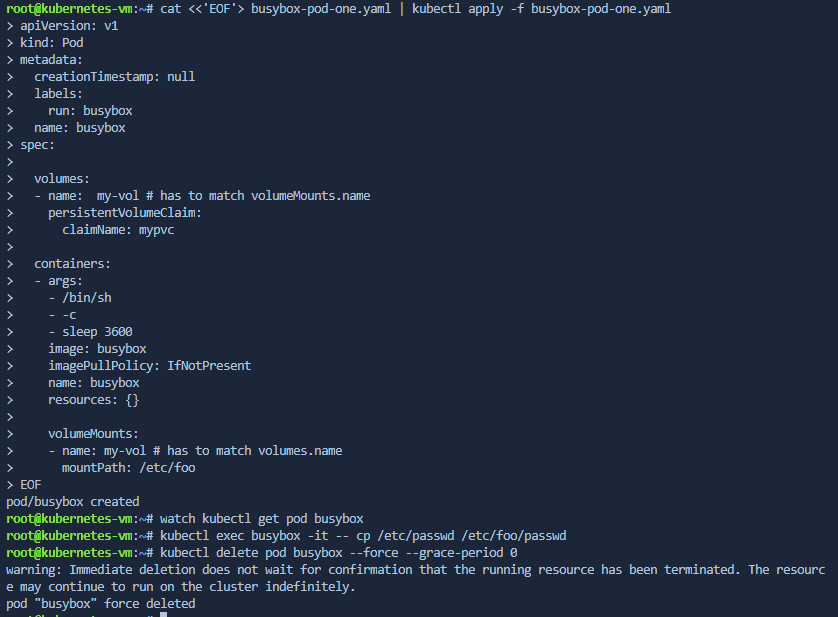


Рисунок 10 - Создание пода и привязка PVC, добавление данных и удаление пода

Далее на рисунке 11 показано создание идентичного пода. Можно убедится в том, что данные появились в нем.

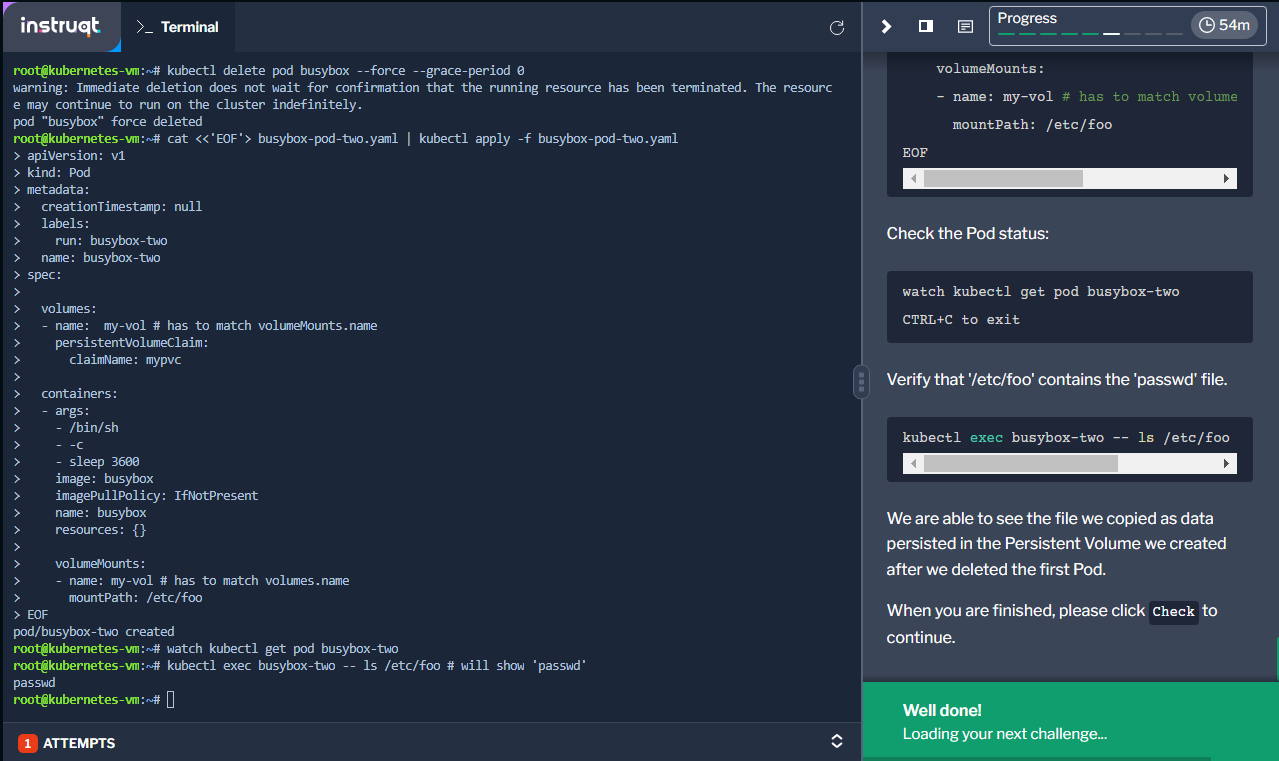


Рисунок 11 - Создание нового пода

Класс VolumeSnapshotClass по умолчанию называется csi-hostpath-snapclass. Это подтверждается выполнением команды (рис. 12):

-kubectl get volumesnapshotclasses

Чтобы получить подтверждение того, что постоянный том создан, нужно использовать команду (рис. 12):

-kubectl get pv

Подтверждение снимка тома осуществляется с помощью команды (рис. 12):

-kubectl get volumesnapshot

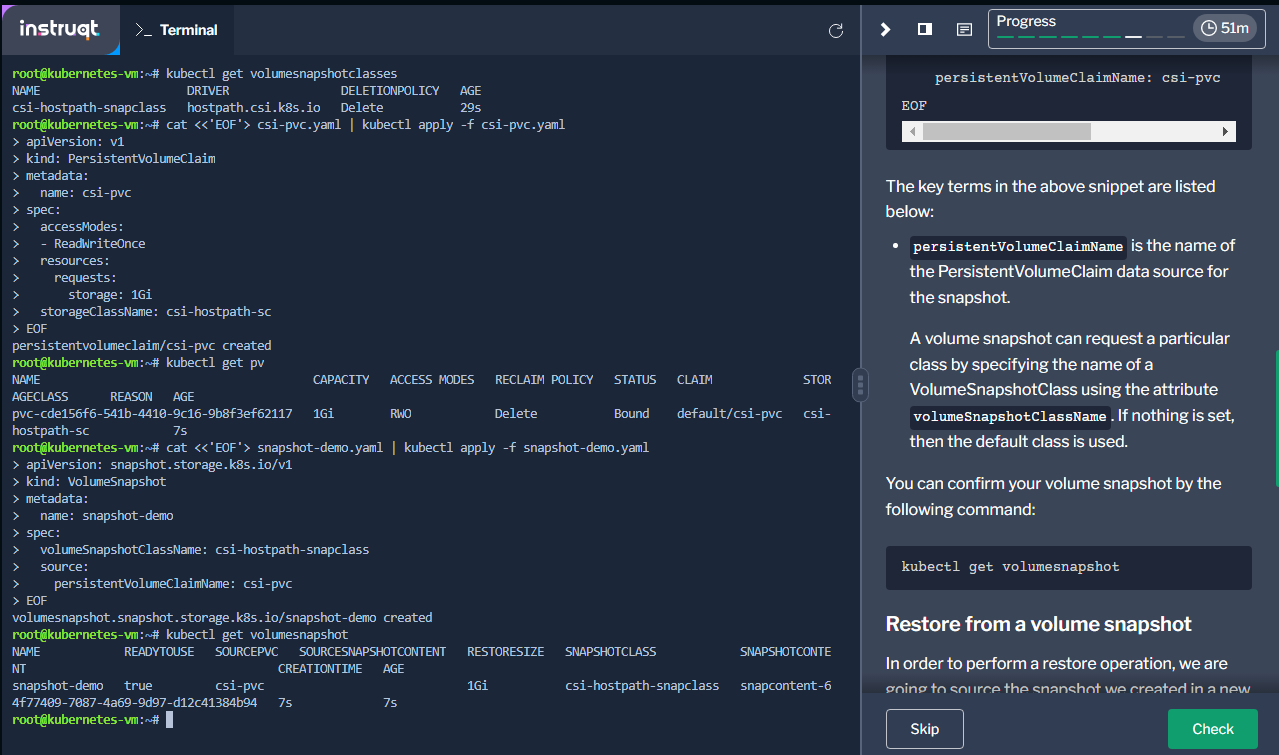


Рисунок 12

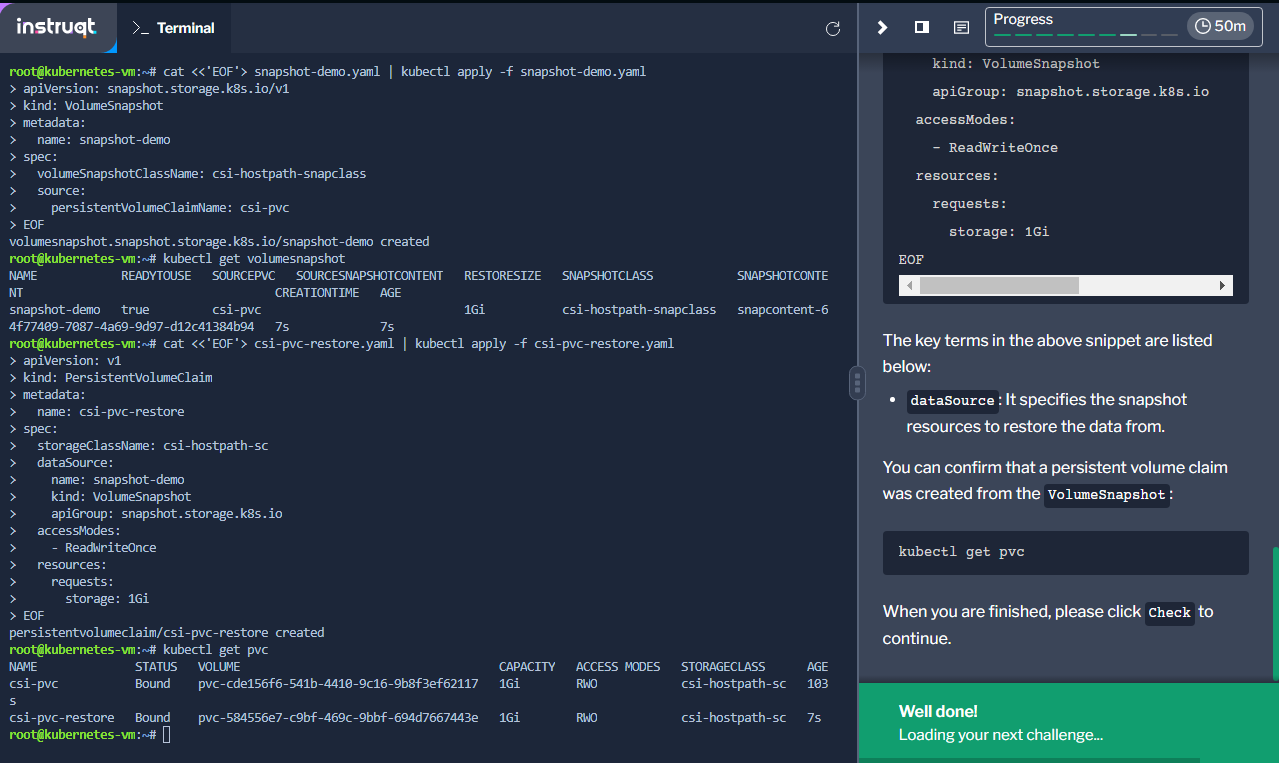


Рисунок 13 - Создание PVC для восстановления и его вывод

Создадим новое пространство имен для размещения приложения с помощью команды (рис. 14): -kubectl create namespace kasten-demo

Развернем в нем приложение, осуществим копирование данных и проверку успешности копирования (рис. 14)

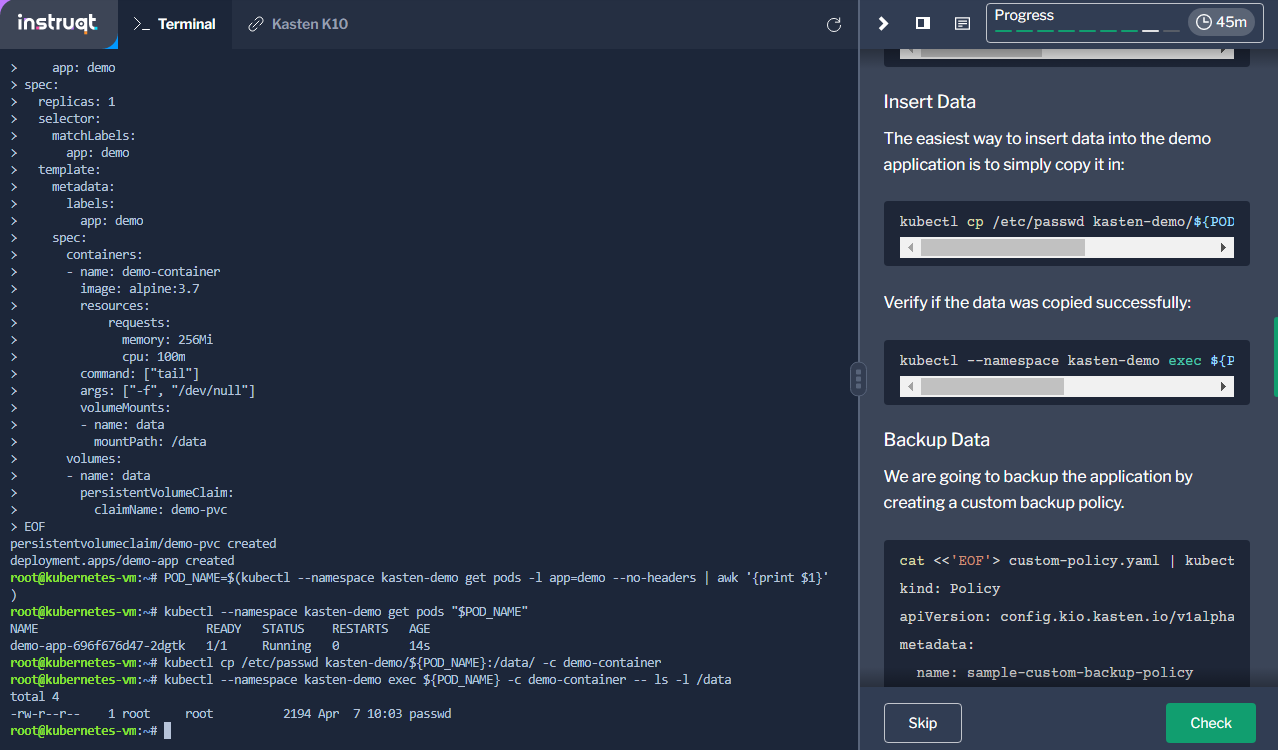


Рисунок 14

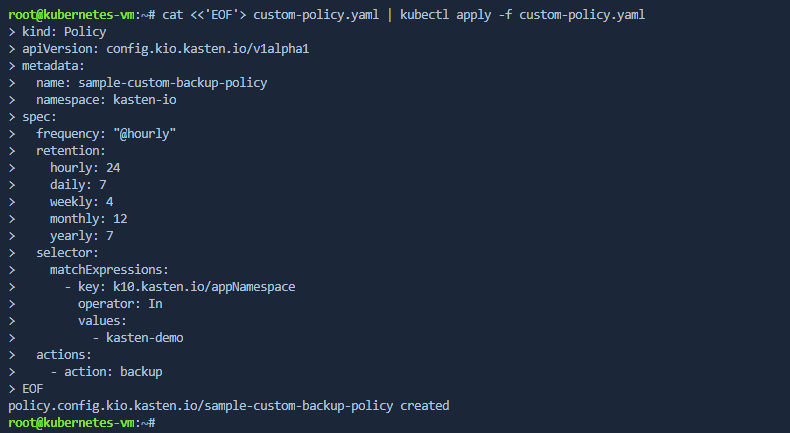


Рисунок 15 – Создание политики на восстановление

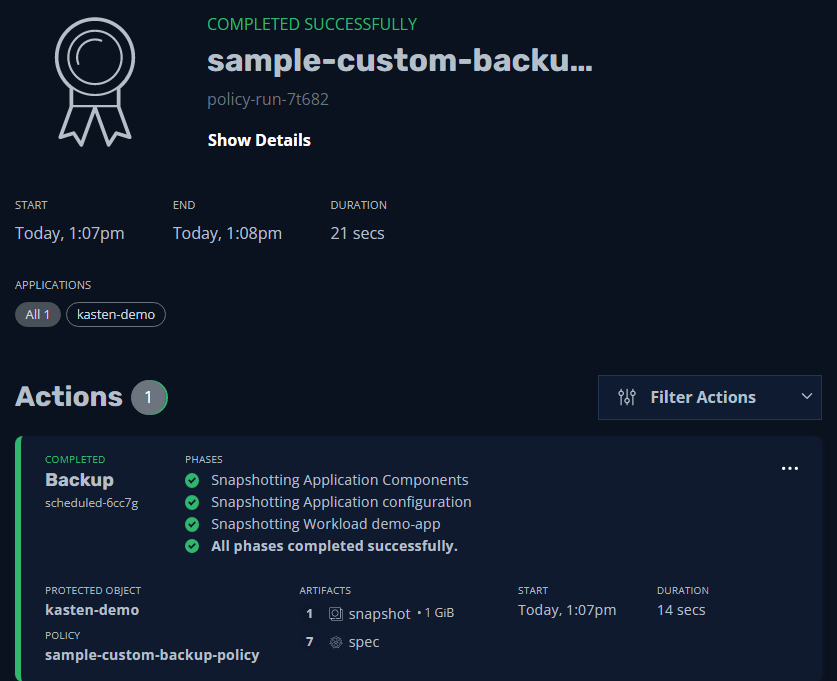


Рисунок 16 – Запуск политики

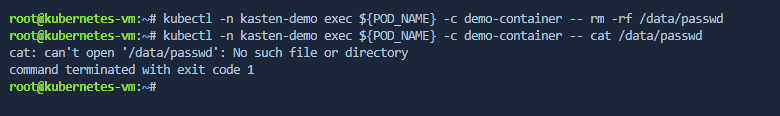


Рисунок 17 – Удаление данных

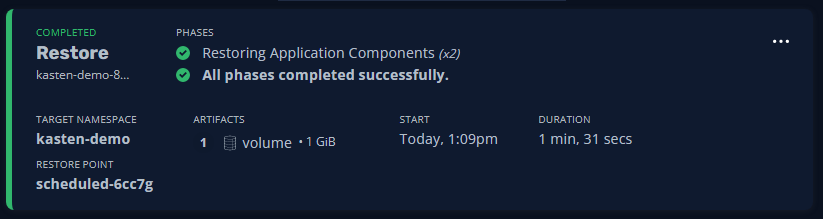


Рисунок 18 – Восстановление данных

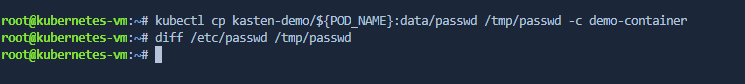


Рисунок 19 – Проверка восстановленных данных