|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ  Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

Институт комплексной безопасности и специального приборостроения

Кафедра КБ-14 «Интеллектуальные системы информационной безопасности»

### Администрирование баз данных

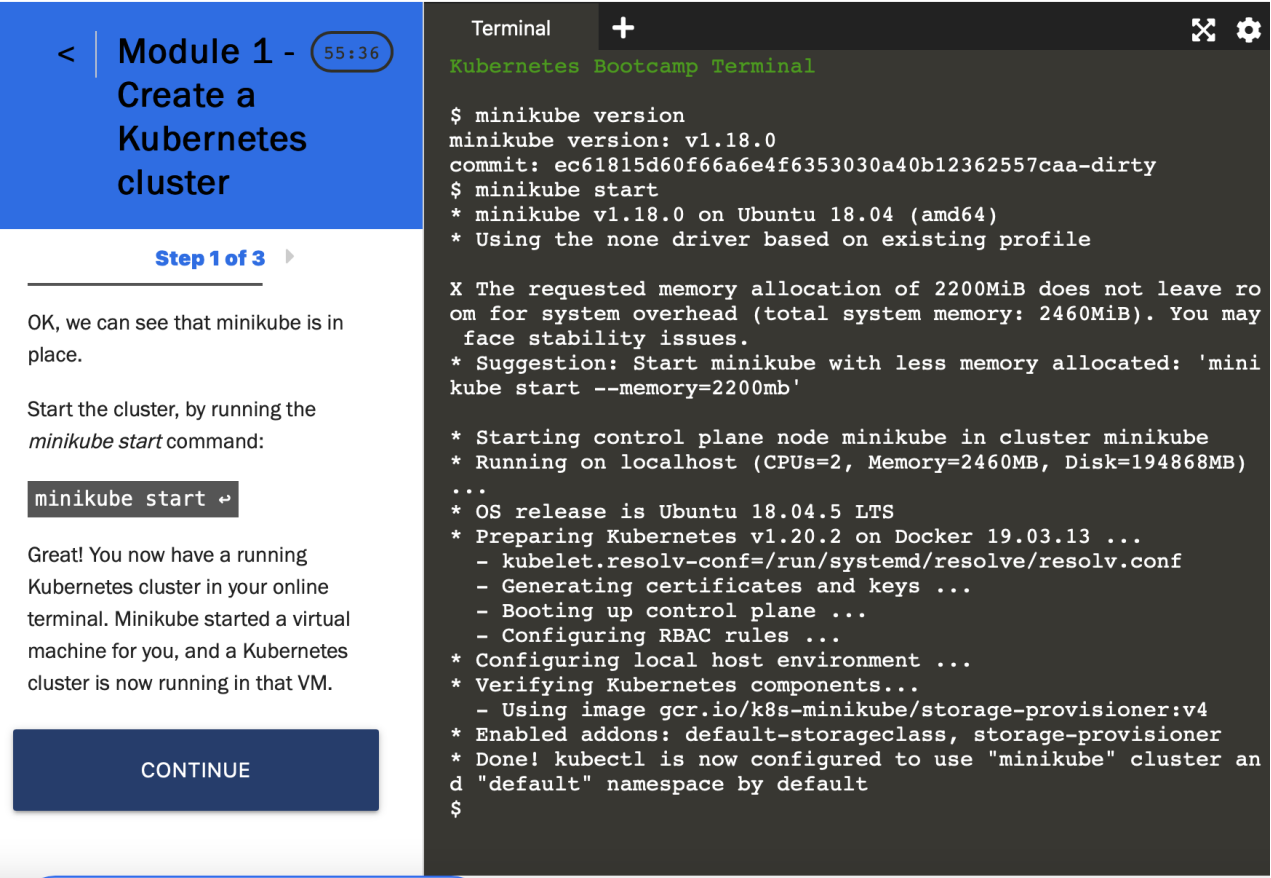
### Практическая работа № 5

# ОТЧЁТ

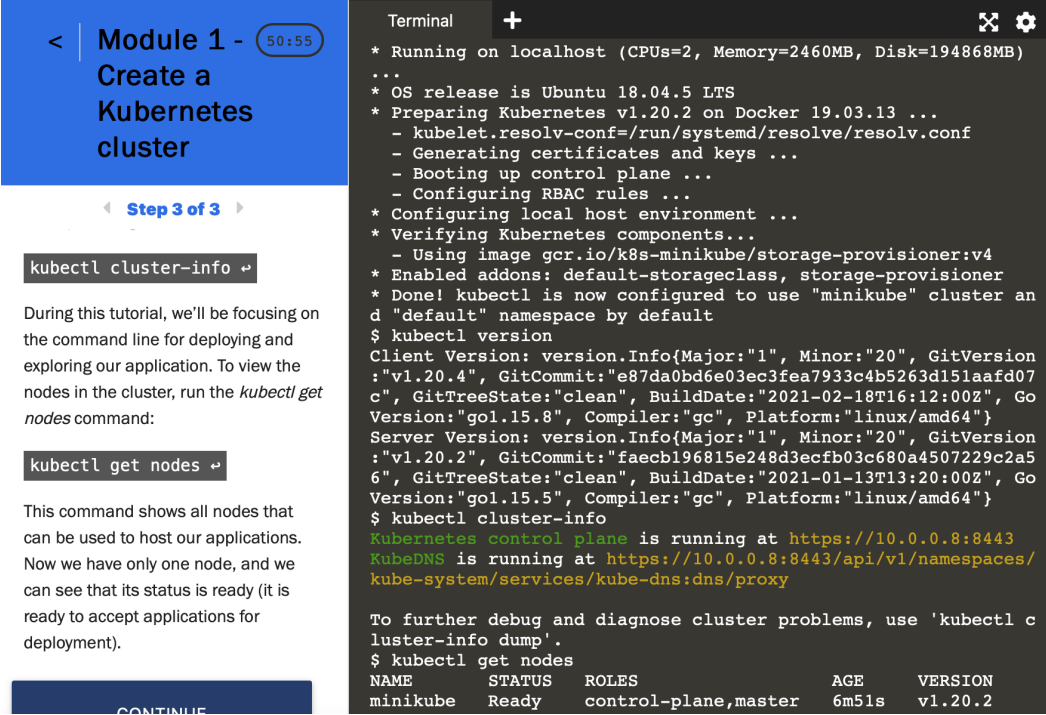
# Выполнил студент группы БСБО-07-20 Любовский С.В.

# Выполнение задания.

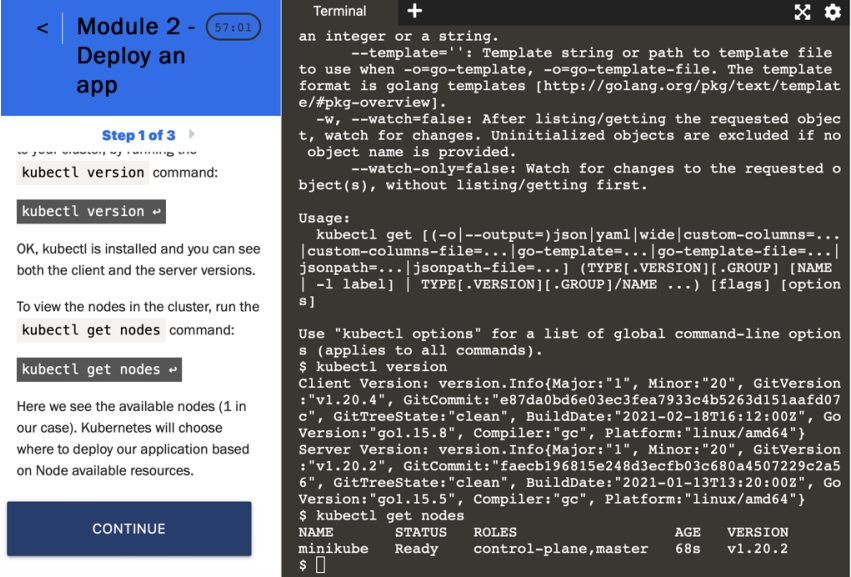
1. Запуск кластера осуществился с помощью команды minikube strart



Команда kubectl get nodes показала все узлы, которые можно использовать для размещения наших приложений. Теперь у нас есть только одна нода, и мы видим, что ее статус готов.



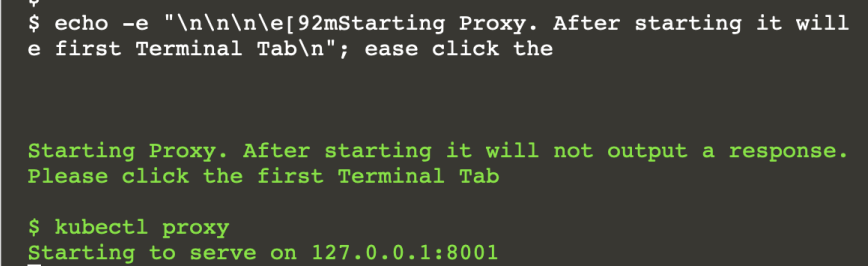
2. Мы видим доступные узлы (в нашем случае 1). Kubernetes выберет, где развернуть приложение, исходя из доступных ресурсов Node.



С помощью команды kubectl create deployment мы развернули приложение. Мы видим, что существует 1 развертывание с одним экземпляром приложения. Экземпляр работает внутри контейнера Docker на нашем узле.



Команда kubectl может создать прокси-сервер, который будет перенаправлять сообщения в частную сеть всего кластера.

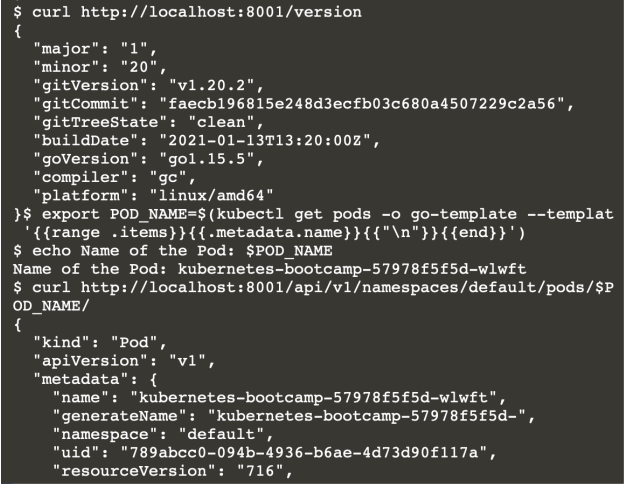


Теперь у нас есть соединение между нашим хостом и кластером Kubernetes. Мы можем запросить версию через API:

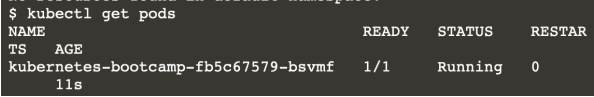
curl http://localhost:8001/version

Можно получить доступ к поду через API, запустив:

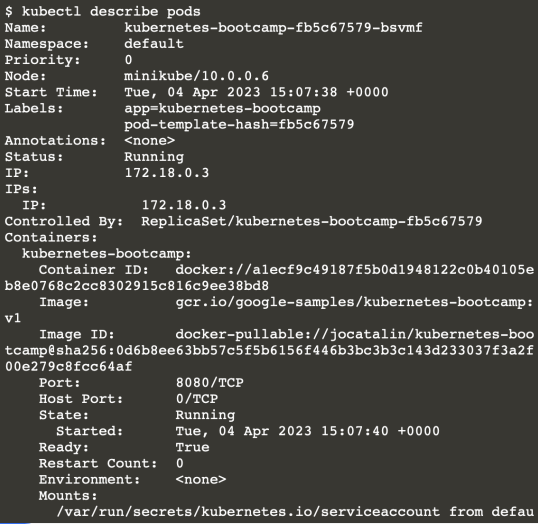
curl http://localhost:8001/api/v1/namespaces/default/pods/$POD\_NAME/



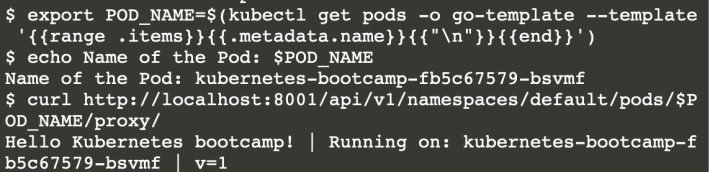
3. Чтобы найти существующие поды, запускает kubectl get pods



Далее просматриваем, какие контейнеры находятся внутри этого пода и какие image используются для создания этих контейнеров



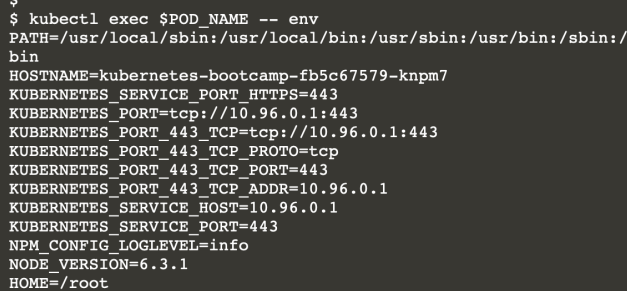
Запускаем прокси во втором терминале, после выводим POD\_NAME и вывод приложения



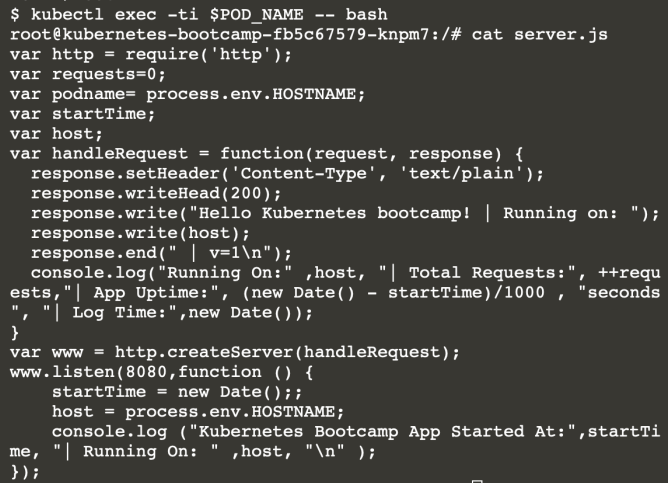
Получение логов



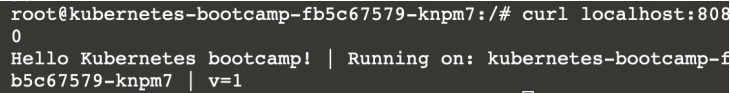
Выполнение команды непосредственно в контейнере после запуска пода



Исходный код приложения находится в файле server.js



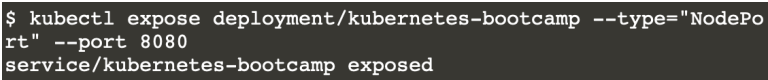
Проверить, запущено ли приложение



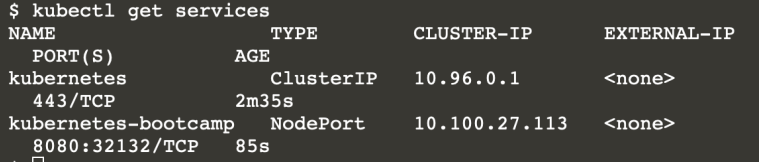
4. Сервисы кластера



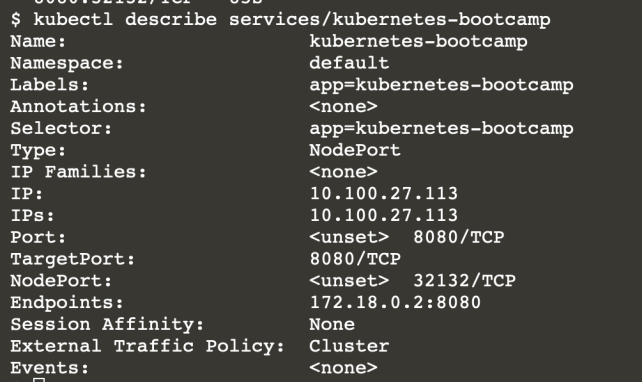
Чтобы создать новый сервис и открыть его для внешнего трафика, используем команду expose с NodePort в качестве параметра



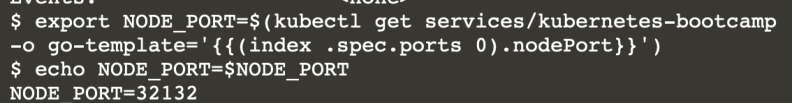
Теперь у нас есть сервис под названием kubernetes-bootcamp



Чтобы узнать, какой порт был открыт, запустим команду describe



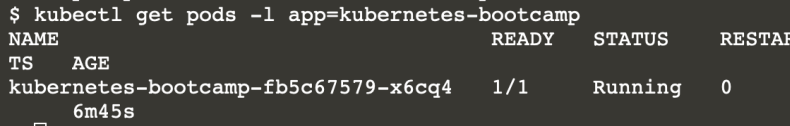
Создает переменную среды с именем NODE\_PORT, которой присвоено значение порта узла



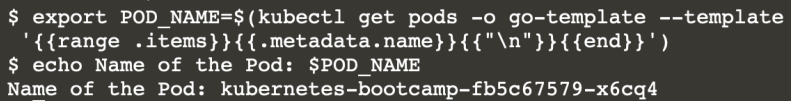
Теперь мы можем проверить, доступно ли приложение за пределами кластера



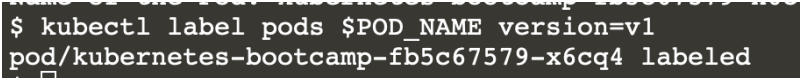
Используем label, вызвав description, для запроса списка подов



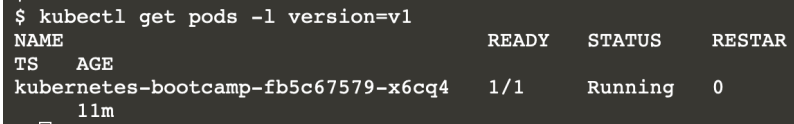
Получите имя пода и сохраните его в переменной POD\_NAME



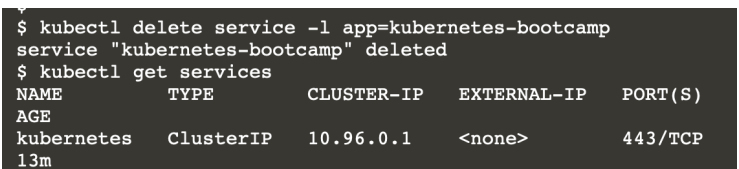
Чтобы применить label



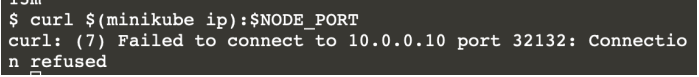
Теперь можно запросить список подов, используя label



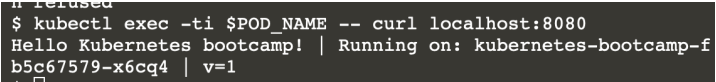
Чтобы удалить сервер используется



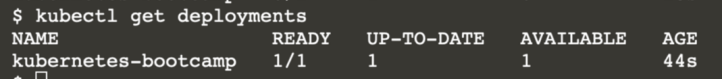
Больше нельзя подключиться



Приложение все еще работает внутри пода



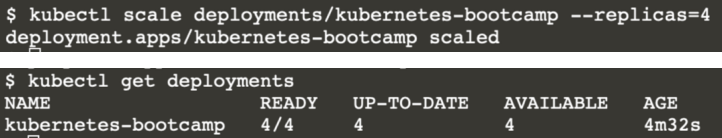
5. Список развернутых подов



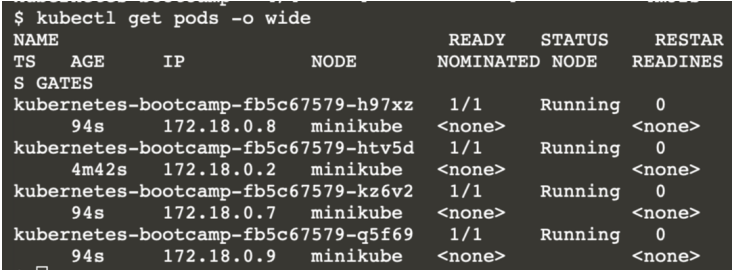
Чтобы увидеть ReplicaSet



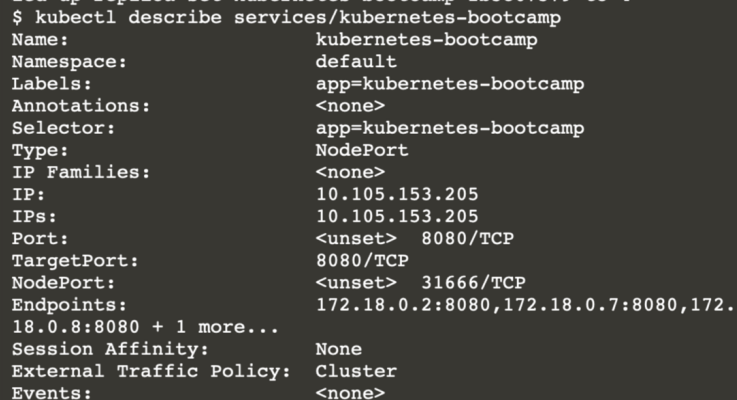
Масштабирование до 4 реплик



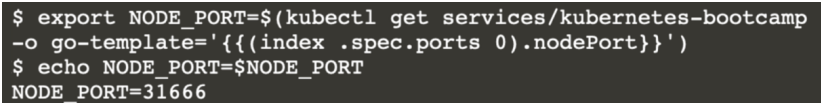
Теперь есть 4 пода с разными IP



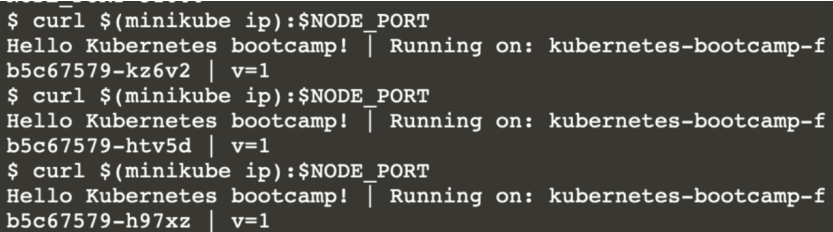
Находим открытые IP и порты



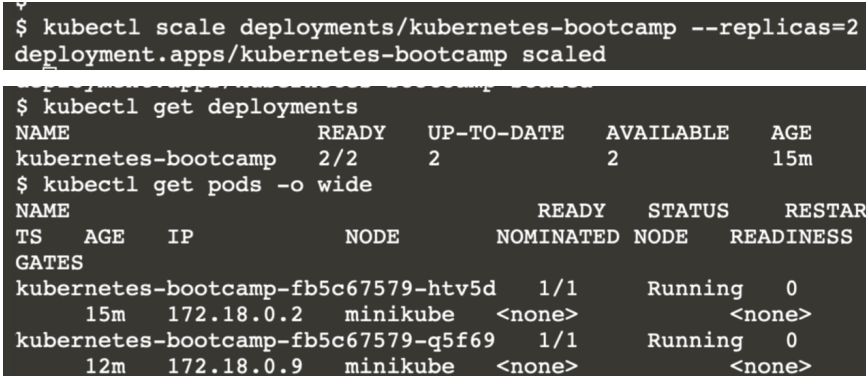
Создание переменной среды с именем NODE\_PORT, которая имеет значение порта узла



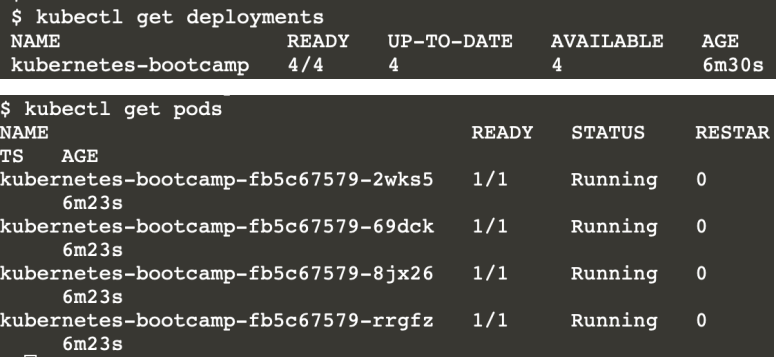
С каждым запросом мы попадаем в разные поды. Это свидетельствует о том, что балансировка нагрузки работает



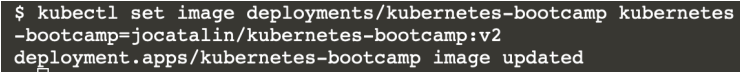
Уменьшение масштаба до 3 реплик



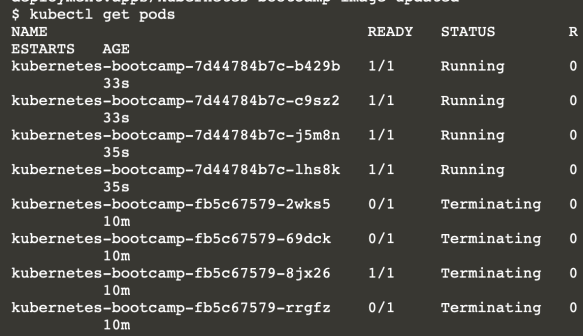
6. Развернутые поды



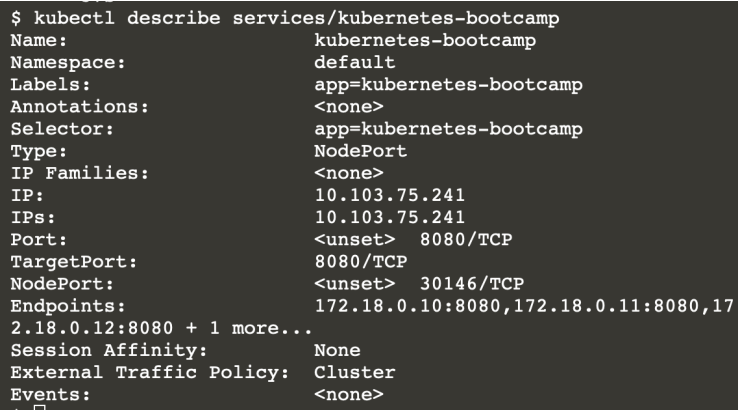
Обновление image



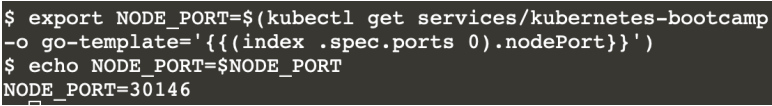
Статус новых подов и старых, завершающих работу



IP и ports



Создание переменной среды с именем NODE\_PORT, которой присвоено значение порта узла



Curl



Каждый раз, когда запускается команда curl, мы попадаем в другой под. На всех подах установлена последняя версия (v2). Выполнив команду rollout status, подтверждаем обновление



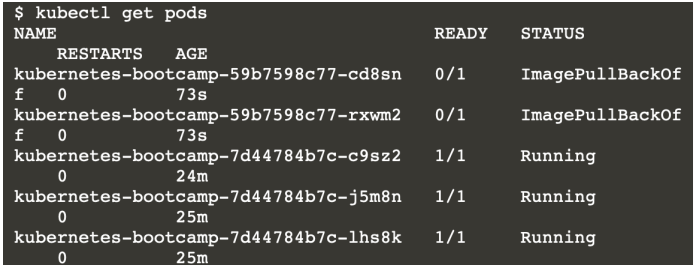
Апдейт до 10 версии



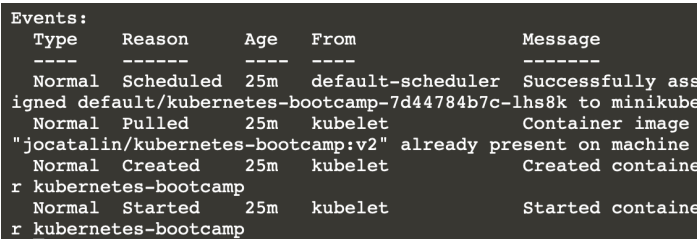
Не все поды были загружены



Некоторые поды имеют статус ImagePullBackOff



Версия образа v10 не существует в репозитории



Откатываем изменения



Rollback прошел успешно

