Министерство Образования, Культуры и Исследований

**Молдавский Государственный Университет**

**Факультет Математики и Информатики**

**Департамент Информатики**

**Лабораторная работа №1**

**Базовые команды K8s**

Научный руководитель: А. Prepelita

Автор: студент группы I2102 (РУ), Juracovschi Vladislav

**Кишинев, 2023**

Оглавление

[Часть 1 1](#_Toc843)

[Часть 2 5](#_Toc9164)

[Создание сервиса NodePort и развертывания 5](#_Toc29833)

[Создание двух пространств имен 6](#_Toc28169)

[Создание подов в разных пространствах имен 7](#_Toc30485)

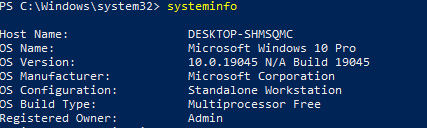
[Интерактивный запуск подов 8](#_Toc4410)

[Вывод 9](#_Toc8478)

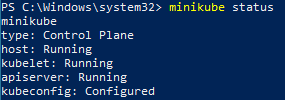
[Библиография 9](#_Toc7146)

# Часть 1

**Systeminfo** - команда , которая предоставляет подробную информацию о компьютере, включая версию операционной системы, конфигурацию процессора, объем памяти и другие системные характеристики.



**minikube status** - команда для проверки текущего статуса Minikube-кластера, который используется для разработки и тестирования Kubernetes приложений.



**kubectl create deployment kubernetes-bootcamp --image=gcr.io/google-samples/kubernetes-bootcamp:v1** - создает в Kubernetes новый деплоймент с именем kubernetes-bootcamp и указывает образ контейнера, который будет развернут в этом деплойменте.



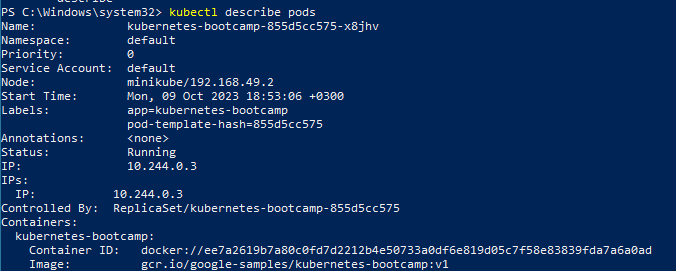
**kubectl get deployments** - это команда для получения списка всех развернутых (deployed) приложений в кластере Kubernetes.



**kubectl get pods** - команда в Kubernetes для получения списка всех подов (контейнеров) в текущем кластере Kubernetes.



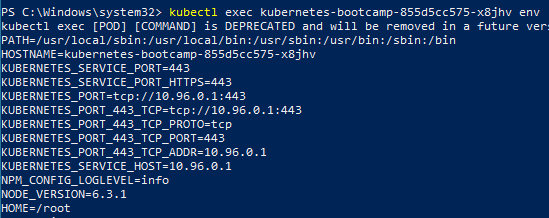
**kubectl describe pods** - это команда в Kubernetes, которая предоставляет подробную информацию о состоянии и конфигурации одного или нескольких pods (контейнеров) в кластере.



**kubectl logs kubernetes-bootcamp-855d5cc575-x8jhv** - это команда для вывода журналов (логов) контейнера, который работает внутри пода с именем "kubernetes-bootcamp-855d5cc575-x8jhv" в Kubernetes кластере



**kubectl exec kubernetes-bootcamp-855d5cc575-x8jhv env** - позволяет просмотреть переменные среды, установленные внутри этого контейнера.



**kubectl exec -ti kubernetes-bootcamp-855d5cc575-x8jhv bash** - команда для выполнения интерактивной командной оболочки (bash) внутри контейнера.



**curl localhost:8080** - это команда для отправки HTTP-запроса на локальный сервер



**Exit** - закрытие bash сессии



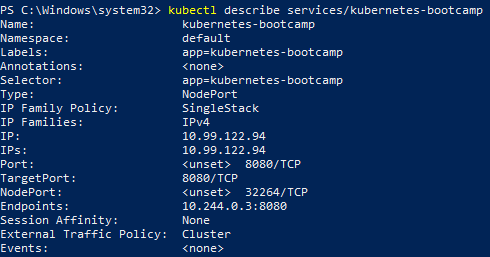
**kubectl get services** - команда для получения списка всех служб в кластере Kubernetes



**kubectl expose deployment/kubernetes-bootcamp --type="NodePort" --port 8080** используется для создания сервиса в Kubernetes, который позволяет доступ к приложению "kubernetes-bootcamp" через NodePort.



**kubectl describe services/kubernetes-bootcamp** - это команда для получения подробной информации о Kubernetes сервисе с именем kubernetes-bootcamp



**kubectl get rs -** отображает информацию о текущем состоянии и количестве реплик каждого ReplicaSet в кластере, позволяя отслеживать и контролировать развертывание приложений.



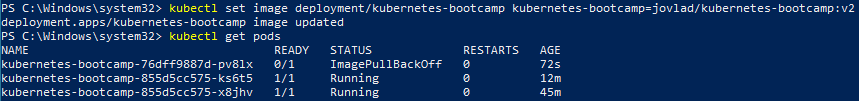
**kubectl scale deployment kubernetes-bootcamp --replicas=2 -** используется для изменения количества реплик (копий) приложения, развернутого в Kubernetes.







**kubectl set image deployment/kubernetes-bootcamp kubernetes-bootcamp=jovlad/kubernetes-bootcamp:v2** - она обновляет образ контейнера с названием "kubernetes-bootcamp" в развёртывании "kubernetes-bootcamp" на новый образ "jovlad/kubernetes-bootcamp:v2"



**kubectl rollout status deployment/kubernetes-bootcamp -** позволяет узнать, успешно ли прошел процесс обновления или развертывания приложения в кластере и сообщает о текущем статусе



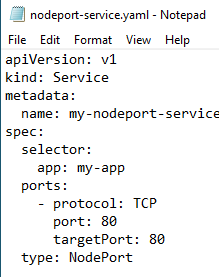
**kubectl rollout undo deployment/kubernetes-bootcamp** - это команда для отката (отмены) обновления внутри Kubernetes-деплоймента с именем "kubernetes-bootcamp"



# Часть 2

Создание сервиса NodePort и развертывания:

Создал YAML-файл , nodeport-service.yaml для создания сервиса NodePort



**kubectl apply -f nodeport-services.yaml** - команда для Kubernetes, которая применяет конфигурацию, описанную в файле nodeport-services.yaml.



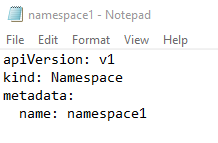




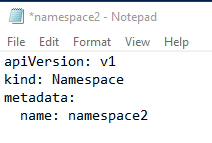
**kubectl apply -f deployment.yaml** - команда для Kubernetes, которая применяет конфигурацию, описанную в файле deployment.yaml.

Создание двух пространств имен:

Создадим файл namespace1.yaml для первого пространства имен:



Создадим файл namespace2.yaml для второго пространства имен:



Применим их с помощью команд:

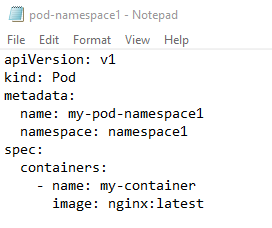
kubectl apply -f namespace1.yaml

kubectl apply -f namespace2.yaml

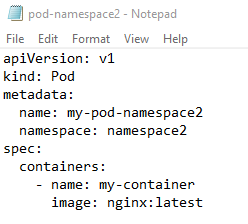


Создание подов в разных пространствах имен:

Создадим файл pod-namespace1.yaml для пода в первом пространстве имен:



Создадим файл pod-namespace2.yaml для пода во втором пространстве имен:



Применим их с помощью команд:

kubectl apply -f pod-namespace1.yaml

kubectl apply -f pod-namespace2.yaml



Интерактивный запуск подов:

Применим интерактивного запуска подов с помощью команд:

kubectl run -it --rm --namespace=namespace1 my-interactive-pod-namespace1 --image=nginx:latest -- bash

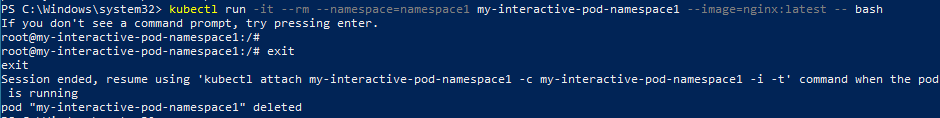
kubectl run -it --rm --namespace=namespace2 my-interactive-pod-namespace1 --image=nginx:latest -- bash

**kubectl run -** создает временный под с образом nginx:latest в пространстве имен namespace1 и открывает интерактивную оболочку Bash внутри этого пода.

После выполнения команды exit, интерактивная сессия завершается, и под удаляется из-за флага --rm.

Появляется сообщение о завершении сессии и предложение использовать команду kubectl attach, чтобы повторно подключиться к поду, если он еще существует и работает.

В конечном итоге, сообщение о том, что под "my-interactive-pod-namespace1" был удален, указывает на завершение жизненного цикла временного пода после использования.



# Вывод

Цель выполнения перечисленных команд и заданий заключается в овладении навыками управления и развертывания приложений в среде Kubernetes. Это важный инструмент для разработчиков и администраторов, позволяющий управлять контейнеризированными приложениями в кластере. Создание развертываний, сервисов и манипуляции с подами и образами обеспечивает понимание ключевых аспектов работы с Kubernetes. Это также включает в себя создание и использование конфигурационных файлов YAML, что является стандартным способом определения состояния кластера. Понимание этих концепций помогает в управлении и масштабировании современных приложений в среде контейнеров.

# Библиография

1. <https://minikube.sigs.k8s.io/docs/start/>
2. <https://kubernetes.io/docs/concepts/workloads/pods/#:~:text=Pods%20are%20the%20smallest%20deployable,how%20to%20run%20the%20containers.>
3. <https://kubernetes.io/docs/tasks/access-application-cluster/list-all-running-container-images/>