



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения
(ИиППО)

ПРАКТИЧЕСКАЯ РАБОТА №5

по дисциплине «Технологии виртуализации клиент-серверных приложений»

Студент группы ИКБО-10-19

Дараган Федор Алексеевич

Руководитель практической работы

(подпись студента)
ассистент Мельников Д.А.

(подпись руководителя)

Работа представлена

«___» _____ 2022 г.

Допущен к работе

«___» _____ 2022 г.

Москва 2022

Оглавление

Практическая работа № 5: «Начало работы с Kubernetes».....	3
Цель работы.....	3
Ход работы.....	4
Вывод.....	9
Ответы на вопросы к практической работе.....	9
1. Назовите виды контроллеров в Kubernetes.....	9
2. Как называется командная строка в Kubernetes?.....	9
3. Что такое под?.....	9
4. Назовите 2 типа ресурсов, из которых состоит кластер Kubernetes.....	10
5. Чем Kubernetes отличается от Docker Swarm?.....	10
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	11

Практическая работа № 5: «Начало работы с Kubernetes»

Цель работы

Вам необходимо выполнить все указанные в задании пункты и отразить в отчете в формате снимков экрана:

1. Для начала работы необходимо установить и запустить minikube в соответствии с установленной ОС с официального сайта [Kubernetes](https://kubernetes.io).
2. Необходимо создать deployment используя локальный docker образ с сервером:
 - название deployment: Фамилия-НомерГруппы (ivanov-ikbo-99-99)
 - используемый образ: Фамилия-НомерГруппы-Образ (ivanovikbo-99-99-obraz)
3. Необходимо посмотреть информацию о Deployment при помощи команды `kubectl get deployments`.
4. Далее необходимо посмотреть информацию о поде при помощи команды `kubectl get pods`.
5. После этого нужно посмотреть события кластера при помощи команды `kubectl get events`.
6. Затем необходимо посмотреть `kubectl` конфигурацию при помощи команды: `kubectl config view`.
7. Потом нужно сделать под с deployment Фамилия-НомерГруппы доступным для публичной сети Интернет с помощью команды `kubectl expose`:
 - сервис должен быть виден вне кластера;
 - порт: 8080.
8. После чего необходимо посмотреть только что созданный сервис `kubectl get services`.
9. Далее нужно запустить сервис `hello-node`: `minikube service ivanov...`
10. Затем требуется отобразить текущие поддерживаемые дополнения и включить дополнение, например `ingress`: `minikube enable ingress`.

11. После этого нужно посмотреть Pod и Service, которые вы только что создали и отключить ingress.
12. После отключения необходимо включить dashboard
 - Далее откройте во вкладке deployments созданный под и опишите в отчете отображаемые параметры.
13. После выполнения освободите ресурсы созданного вами кластера и остановите Minikube.

Ход работы

На листинге 1 показан сервер, на листинге 2 докер-файл для запуска этого сервера.

Листинг 1 — Сервер на js

```
const http = require('http');

const handleRequest = (request, response) => {
  console.log(`Получен запрос на URL: ${request.url}`);
  response.writeHead(200);
  response.end("Привет мир!");
}

const server = http.createServer(handleRequest);
server.listen(8080);
```

Листинг 2 — Докер-файл

```
FROM node:19.2-alpine
EXPOSE 8080
COPY server.js .
CMD node server.js
```

На рисунке 1 показан процесс запуска minikube.

```
laefad@laefad-pc: /media/laefad/Data/Учеба/Виртуалки/Практическая работа 5/code$ minikube start
minikube v1.28.0 на Ubuntu 22.04
Используется драйвер docker на основе существующего профиля
Запускается control plane узел minikube в кластере minikube
Скачивается базовый образ ...
> gcr.io/k8s-minikube/kicbase: 0 В [ ] ?% ? p/s 59s
Перезагружается существующий docker container для "minikube" ...
Подготавливается Kubernetes v1.25.3 на Docker 20.10.20 ...
Компоненты Kubernetes проверяются ...
  • Используется образ gcr.io/k8s-minikube/storage-provisioner:v5
  • Используется образ docker.io/kubernetesui/dashboard:v2.7.0
  • Используется образ docker.io/kubernetesui/metrics-scraper:v1.0.8
Some dashboard features require the metrics-server addon. To enable all features please run:

    minikube addons enable metrics-server

Включенные дополнения: storage-provisioner, default-storageclass, dashboard
Готово! kubectl настроен для использования кластера "minikube" и "default" пространства имён по умолчанию
```

Рисунок 1. Скриншот запуска minikube

На рисунке 2 показана сборка докер-файла и загрузка его в Docker Hub, чтобы Minikube мог его скачать.

```
laefad@laefad-pc: /media/laefad/Data/Учеба/Виртуалки/Практическая работа 5/code$ docker build -f Dockerfile -t laefad/daragan-ikbo-10-19-obraz .
[+] Building 2.3s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 31B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/node:19.2-alpine
=> [auth] library/node:pull token for registry-1.docker.io
=> [internal] load build context
=> => transferring context: 31B
=> [1/2] FROM docker.io/library/node:19.2-alpine@sha256:80844b6643f239c87fcea51e6540eeb054fc7114d979703770ec75250dcd03b
=> CACHED [2/2] COPY server.js .
=> exporting to image
=> => exporting layers
=> => writing image sha256:17e6df5e6123a3915e6e133ec7bde68e6c00dfc8b3f22b2f072c6c03a6abf
=> naming to docker.io/laefad/daragan-ikbo-10-19-obraz

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
laefad@laefad-pc: /media/laefad/Data/Учеба/Виртуалки/Практическая работа 5/code$ docker push laefad/daragan-ikbo-10-19-obraz
Using default tag: latest
The push refers to repository [docker.io/laefad/daragan-ikbo-10-19-obraz]
2d479909c4e1: Layer already exists
5957ac8a40f1: Layer already exists
7ae7e09dbb66: Layer already exists
a3ecf5a1df9e: Layer already exists
e5e13b0c77cb: Layer already exists
latest: digest: sha256:00086ea184b1004f31ef0e0f48280086b0e374c42901b89691508a5fe88eb20e size: 1365
```

Рисунок 2. Скриншот сборки и загрузки образа в Docker Hub

На рисунке 3 показано создание deployment, а также списки существующих deployment и pod.

```
laefad@laefad-pc: /media/laefad/Data/Учеба/Виртуалки/Практическая работа 5/code$ kubectl create deployment daragan-ikbo-10-19 --image=laefad/daragan-ikbo-10-19-obraz
deployment.apps/daragan-ikbo-10-19 created
laefad@laefad-pc: /media/laefad/Data/Учеба/Виртуалки/Практическая работа 5/code$ kubectl get deployments
NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
daragan-ikbo-10-19                 1/1     1             1           6s
laefad@laefad-pc: /media/laefad/Data/Учеба/Виртуалки/Практическая работа 5/code$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
daragan-ikbo-10-19-6c44f8b4b4-hmlfq 1/1     Running   0           10s
```

Рисунок 3. Скриншот создания deployment

На рисунке 4 показан конец вывода команды `kubectl get events`, поскольку полный список событий достаточно большой и относится к «неудачным» запускам.

46m	Normal	Scheduled	pod/daragan-ikbo-10-19	Successfully assigned default/daragan-ikbo-10-19 to minikube
46m	Normal	Pulled	pod/daragan-ikbo-10-19	Container image "daragan-ikbo-10-19-образ" already present on machine
46m	Normal	Created	pod/daragan-ikbo-10-19	Created container daragan-ikbo-10-19
46m	Normal	Started	pod/daragan-ikbo-10-19	Started container daragan-ikbo-10-19
16m	Normal	Killing	pod/daragan-ikbo-10-19	Stopping container daragan-ikbo-10-19
16m	Normal	ScalingReplicaSet	deployment/daragan-ikbo-10-19	Scaled up replica set daragan-ikbo-10-19-6c44f8b4b4 to 1
13s	Normal	ScalingReplicaSet	deployment/daragan-ikbo-10-19	Scaled up replica set daragan-ikbo-10-19-6c44f8b4b4 to 1
44s	Normal	Starting	node/minikube	Starting kubelet.
44s	Normal	NodeHasSufficientMemory	node/minikube	Node minikube status is now: NodeHasSufficientMemory
44s	Normal	NodeHasNoDiskPressure	node/minikube	Node minikube status is now: NodeHasNoDiskPressure
44s	Normal	NodeHasSufficientPID	node/minikube	Node minikube status is now: NodeHasSufficientPID
44s	Normal	NodeAllocatableEnforced	node/minikube	Updated Node Allocatable limit across pods
38s	Normal	Starting	node/minikube	
27s	Normal	RegisteredNode	node/minikube	Node minikube event: Registered Node minikube in Controller
51m	Normal	Scheduled	pod/pod	Successfully assigned default/pod to minikube
50m	Normal	Pulled	pod/pod	Container image "daragan-ikbo-10-19-образ" already present on machine
50m	Normal	Created	pod/pod	Created container pod
50m	Normal	Started	pod/pod	Started container pod
50m	Warning	BackOff	pod/pod	Back-off restarting failed container

Рисунок 4. Скриншот конца списка событий `kubectl`

На рисунке 5 показан текущая конфигурация `kubernetes`.

```
laefad@laefad-pc: /media/laefad/Data/Учеба/Виртуалки/Практическая работа 5/code$ kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /home/laefad/.minikube/ca.crt
    extensions:
    - extension:
        last-update: Sat, 10 Dec 2022 15:19:38 MSK
        provider: minikube.sigs.k8s.io
        version: v1.28.0
        name: cluster_info
    server: https://127.0.0.1:37639
    name: minikube
contexts:
- context:
    cluster: minikube
    extensions:
    - extension:
        last-update: Sat, 10 Dec 2022 15:19:38 MSK
        provider: minikube.sigs.k8s.io
        version: v1.28.0
        name: context_info
    namespace: default
    user: minikube
    name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: minikube
  user:
    client-certificate: /home/laefad/.minikube/profiles/minikube/client.crt
    client-key: /home/laefad/.minikube/profiles/minikube/client.key
```

Рисунок 5. Скриншот конфигурации `kubernetes`

На рисунке 9 показан результат установки расширения ingress.

```
Laefad@laefad-pc: /media/laefad/Data/Учеба/Виртуалки/Практическая работа 5/code$ minikube addons enable ingress
ingress is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
  ▪ Используется образ k8s.gcr.io/ingress-nginx/controller:v1.2.1
  ▪ Используется образ k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.1.1
  ▪ Используется образ k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.1.1
Verifying ingress addon...
The 'ingress' addon is enabled
Laefad@laefad-pc: /media/laefad/Data/Учеба/Виртуалки/Практическая работа 5/code$
```

Рисунок 9. Скриншот установки ingress в kubernetes

На рисунке 10 показан вывод информации о ingress.

```
Laefad@laefad-pc: /media/laefad/Data/Учеба/Виртуалки/Практическая работа 5/code$ kubectl get pod,svc -n kube-system

NAME                                READY   STATUS    RESTARTS   AGE
pod/coredns-565d847f94-9v6wp        1/1     Running   1 (86m ago) 43h
pod/etcd-minikube                    1/1     Running   1 (86m ago) 43h
pod/kube-apiserver-minikube          1/1     Running   1 (86m ago) 43h
pod/kube-controller-manager-minikube 1/1     Running   1 (86m ago) 43h
pod/kube-proxy-m6c7v                1/1     Running   1 (86m ago) 43h
pod/kube-scheduler-minikube          1/1     Running   1 (86m ago) 43h
pod/storage-provisioner              1/1     Running   3 (85m ago) 43h

NAME                                TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
service/kube-dns                    ClusterIP      10.96.0.10   <none>         53/UDP,53/TCP,9153/TCP 43h
```

Рисунок 10. Скриншот вывода информации о подах, созданных ingress

На рисунке 11 показана панель управления minikube. На ней указано имя, пространство имен, дата создания, время существования и уникальный идентификатор для выбранного развертывания. Указана стратегия обновления, время до запуска и количество записей в истории. Указано количество работающих подов.

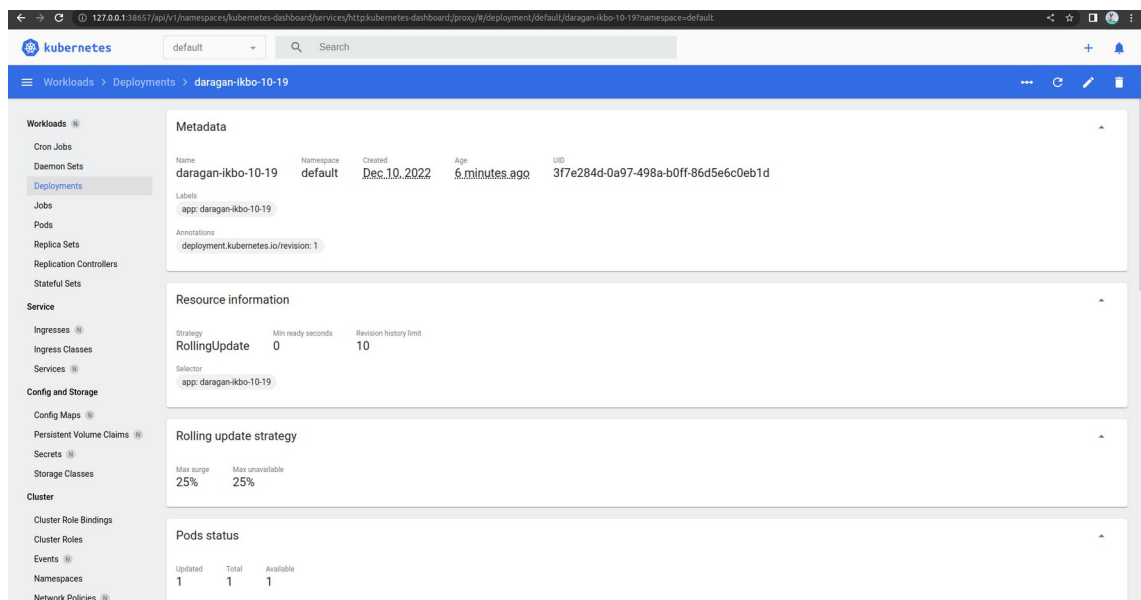


Рисунок 11. Скриншот панели управления minikube для созданного ранее пода

Вывод

В результате выполнения четвертой практической работы были получены навыки работы с Kubernetes и minikube , были созданы поды и сервисы для их работы.

Ответы на вопросы к практической работе

1. Назовите виды контроллеров в Kubernetes.

Deployments — контроллер, который управляет состоянием развертывания подов, которое описывается в манифесте, следит за удалением и созданием экземпляров подов. Управляет контроллерами ReplicaSet.

ReplicaSet — гарантирует, что определенное количество экземпляров подов всегда будет запущено в кластере.

StatefulSets — так же, как и Deployments, управляет развертыванием и масштабированием набора подов, но сохраняет набор идентификаторов и состояние для каждого пода.

DaemonSet — гарантирует, что на каждом узле кластера будет присутствовать экземпляр пода.

Jobs — создает определенное количество подов и смотрит, пока они успешно не завершат работу. Если под завершился с ошибкой, повторяет создание, которое мы описали определенное количество раз. Если под успешно отработал, записывает это в свой журнал.

CronJob — запускает контроллеры Jobs по определенному расписанию.

2. Как называется командная строка в Kubernetes?

Kubectl — это инструмент командной строки для управления кластерами Kubernetes. kubectl ищет файл config в директории \$HOME/.kube. Вы можете указать другие файлы kubeconfig, установив переменную окружения KUBECONFIG или флаг --kubeconfig.

3. Что такое под?

Pods или поды — это абстрактный объект в кластере K8S, который состоит из одного или нескольких контейнеров с общим хранилищем и сетевыми ресурсами, а также спецификации для запуска контейнеров.

Это главный объект в кластере, в нем прописаны, какие контейнеры должны быть запущены, количество экземпляров или реплик, политика перезапуска, лимиты, подключаемые ресурсы, узел кластера для размещения.

- kube-scheduler планирует размещение пода на узлах кластера
- kubelet на рабочем узле кластера запускает под

4. Назовите 2 типа ресурсов, из которых состоит кластер Kubernetes.

Сам кластер K8S состоит из, барабанная дробь, рабочих узлов. В узлах или нодах (Nodes, Worker nodes), помимо контейнеров компонентов самого кластера, размещаются контейнеры наших проектов и сервисов.

Worker nodes состоит из компонентов:

- kubelet — сервис или агент, который контролирует запуск компонентов (контейнеров) кластера
- kube-proxy — конфигурирует правила сети на узлах

Плоскость управления (Master nodes) управляет рабочими узлами и подами в кластере. Там располагаются компоненты, которые управляют узлами кластера и предоставляют доступ к API.

Control plane состоит из компонентов:

- kube-apiserver — предоставляет API кубера
- etcd — распределенное key-value хранилище для всех данных кластера. Необязательно располагается внутри мастера, может стоять как отдельный кластер
- kube-scheduler — планирует размещение подов на узлах кластера
- kube-controller-manager — запускает контроллер
- kubelet — сервис или агент, который контролирует запуск основных компонентов (контейнеров) кластер

5. Чем Kubernetes отличается от Docker Swarm?

Kubernetes уже включает множество инструментов оркестровки, например, автомасштабирование.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Установка Kubernetes с помощью Minikube — Текст: электронный [сайт]. — URL: <https://kubernetes.io/ru/docs/setup/learningenvironment/minikube/>
2. K8S для начинающих. Первая часть — Текст: электронный [сайт]. — URL: <https://habr.com/ru/post/589415/>
3. Kubernetes или с чего начать, чтобы понять что это и зачем он нужен — Текст: электронный [сайт]. — URL: <https://habr.com/ru/company/otus/blog/537162/>
4. Основы Kubernetes — Текст: электронный [сайт]. — URL: <https://habr.com/ru/post/258443/>
5. Kubernetes — Текст: электронный [сайт]. — URL: <https://kubernetes.io/ru/>