

**Московский государственный технический университет  
им. Н. Э. Баумана**

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5. Курс «Автоматизация разработки и эксплуатации ПО»**

**Отчет по лабораторной работе №4**

Выполнил:

Студентка группы ИУ5-71Б

Ноздрова Валентина

Проверил:

Дата: 29.12.2022

Подпись:

Дата:

Подпись:







Москва, 2022 г.

## План и задачи лабораторной работы:

1. Подготовка рабочего окружения
2. Знакомство с kubernetes
3. Работа с kubernetes

## Ход выполнения работы:

### 1. Подготовка рабочего окружения

```
Creating cluster "kind" ...  
✓ Ensuring node image (kindest/node:v1.25.3)   
✓ Preparing nodes   
✓ Writing configuration   
✓ Starting control-plane   
✓ Installing CNI   
✓ Installing StorageClass 
```

```
user@devopsiu5:~$ kubectl get ns  
NAME                STATUS   AGE  
default             Active   24m  
kube-node-lease     Active   24m  
kube-public         Active   24m  
kube-system         Active   24m  
local-path-storage  Active   21m
```

### 2. Знакомство с kubernetes

```
user@devopsiu5:~$ kubectl get pods -n kube-system  
NAME                                     READY   STATUS    RESTARTS   AGE  
coredns-565d847f94-2vmvn               1/1     Running   0           36m  
coredns-565d847f94-8768l               1/1     Running   0           36m  
etcd-kind-control-plane                 1/1     Running   0           39m  
kindnet-qw6qt                           1/1     Running   1 (2m22s ago)  36m  
kube-apiserver-kind-control-plane        0/1     Running   1 (2m30s ago)  39m  
kube-controller-manager-kind-control-plane 1/1     Running   4 (4m9s ago)   39m  
kube-proxy-8mk2m                        1/1     Running   0           36m  
kube-scheduler-kind-control-plane        1/1     Running   3 (33m ago)    39m
```

Coredns – DNS-сервер кластера

Etcd – хранилище всех данных кластера

Kindnet – сетевой интерфейс контейнеров

Kube-apiserver – клиентская часть панели управления

Kube-controller-manager – запускает процессы контроллера

Kube-proxy – сетевой прокси, конфигурирует правила сети на узлах

Kube-scheduler – планирует развертывание подов на нодах

### 3. Работа с kubernetes

```
user@devopsiu5:~$ kubectl get pods  
NAME     READY   STATUS    RESTARTS   AGE  
nginx    1/1     Running   0           117s  
user@devopsiu5:~$ kubectl apply -f netshoot_pod.yml  
pod/netshoot created
```

```
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

```
nginx      1/1      Running    0           176m    10.244.0.7    kind-control-plane    <
none>      <none>
nginx      1/1      Running    0           29m     10.244.0.9    kind-control-plane    <n
one>      <none>
```

При пересоздании пода его ip изменился

Добавление лейбла к поду:

```
user@devopsiu5:~$ kubectl label pod nginx app=nginx
pod/nginx labeled
```

```
user@devopsiu5:~$ kubectl get svc
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP  PORT(S)    AGE
kubernetes           ClusterIP   10.96.0.1     <none>       443/TCP    4d22h
nginx-service        ClusterIP   10.96.227.51  <none>       80/TCP     2m3s
nginx-service-headless ClusterIP   None          <none>       80/TCP     6s
```

```
bash-5.2# curl nginx-service
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

```
bash-5.2# host nginx-service
nginx-service.default.svc.cluster.local has address 10.96.227.51
bash-5.2# host nginx-service-headless
nginx-service-headless.default.svc.cluster.local has address 10.244.0.9
```

Nginx-service – укороченное имя сервиса, будет работать в рамках default namespace.

Nginx-service использует clusterip – постоянный ip, а nginx-service-headless – ip пода.

```
user@devopsiu5:~$ kubectl get svc --all-namespaces
NAMESPACE   NAME                 TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
default     kubernetes           ClusterIP  10.96.0.1    <none>        443/TCP          5d
default     nginx-service        ClusterIP  10.96.227.51 <none>        80/TCP           136m
default     nginx-service-headless ClusterIP   None         <none>          80/TCP           134m
ingress-nginx ingress-nginx-controller NodePort    10.96.165.249 <none>          80:31195/TCP,443:30997/TCP 5d
ingress-nginx ingress-nginx-controller-admission ClusterIP    10.96.42.251 <none>          443/TCP           5d
kube-system kube-dns             ClusterIP  10.96.0.10   <none>        53/UDP,53/TCP,9153/TCP 5d

ingress.networking.k8s.io/minimal-ingress created
user@devopsiu5:~$ curl localhost:80
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>

① localhost
```

## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org/).  
Commercial support is available at [nginx.com](http://nginx.com/).

*Thank you for using nginx.*

```
user@devopsiu5:~$ kubectl delete pod/nginx
pod "nginx" deleted
user@devopsiu5:~$ curl localhost:80
<html>
<head><title>503 Service Temporarily Unavailable</title></head>
<body>
<center><h1>503 Service Temporarily Unavailable</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

При удалении пода с nginx получим ошибку, так как контейнер с приложением запускается в поде.

```
user@devopsiu5:~$ kubectl get replicaset
NAME                               DESIRED   CURRENT   READY   AGE
nginx-deployment-7fb96c846b        3         3         3       60s

NAME                               DESIRED   CURRENT   READY   AGE
nginx-deployment-7fb96c846b-c7r2l  1/1      Running   0       18s
nginx-deployment-7fb96c846b-rtv6b  1/1      Running   0       18s
nginx-deployment-7fb96c846b-tbrh6  1/1      Running   0       18s

NAME                               READY    UP-TO-DATE   AVAILABLE   AGE
nginx-deployment 3/3      3             3           107s
```

При создании deployment созданся replicaset из трех контейнеров с nginx

```
user@devopsiu5:~$ kubectl set image deploy/nginx-deployment nginx=nginx:iam-not-exists
deployment.apps/nginx-deployment image updated
```

NAME	READY	STATUS	RESTARTS	AGE
netshoot	1/1	Running	0	17m
nginx-deployment-5cb9d77c68-j2l9w	0/1	ErrImagePull	0	9s
nginx-deployment-7fb96c846b-c7r2l	1/1	Running	0	3m18s
nginx-deployment-7fb96c846b-rtv6b	1/1	Running	0	3m18s
nginx-deployment-7fb96c846b-tbrh6	1/1	Running	0	3m18s

Events:	Type	Reason	Age	From	Message
	Normal	Scheduled	2m35s	default-scheduler	Successfully assigned default/nginx-deployment-5cb9d77c68-j2l9w to kind-control-plane
	Normal	Pulling	57s (x4 over 2m34s)	kubelet	Pulling image "nginx:iam-not-exists"
	Warning	Failed	53s (x4 over 2m31s)	kubelet	Failed to pull image "nginx:iam-not-exists": rpc error: code = NotFound desc = failed to pull and unpack image "docker.io/library/nginx:iam-not-exists": failed to resolve reference "docker.io/library/nginx:iam-not-exists": docker.io/library/nginx:iam-not-exists: not found
	Warning	Failed	53s (x4 over 2m31s)	kubelet	Error: ErrImagePull
	Warning	Failed	39s (x6 over 2m31s)	kubelet	Error: ImagePullBackOff
	Normal	BackOff	25s (x7 over 2m31s)	kubelet	Back-off pulling image "nginx:iam-not-exists"

```
deployment.apps/nginx-deployment rolled back
```

```
user@devopsiu5:~$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
netshoot	1/1	Running	0	21m
nginx-deployment-7fb96c846b-c7r2l	1/1	Running	0	6m25s
nginx-deployment-7fb96c846b-rtv6b	1/1	Running	0	6m25s
nginx-deployment-7fb96c846b-tbrh6	1/1	Running	0	6m25s

```
> containers:
>   - name: nginx
>     image: nginx:1.14.2
>     ports:
>       - containerPort: 80
>     livenessProbe:
>       httpGet:
>         path: /healthz
>         port: 8080
>       initialDelaySeconds: 3
>       periodSeconds: 3
> EOF
deployment.apps/nginx-deployment configured
user@devopsiu5:~$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
netshoot	1/1	Running	0	24m
nginx-deployment-6d845d9d87-8t4zf	0/1	CrashLoopBackOff	5 (36s ago)	2m19s
nginx-deployment-6d845d9d87-tjvkm	0/1	CrashLoopBackOff	5 (33s ago)	2m16s
nginx-deployment-6d845d9d87-v9rlw	0/1	CrashLoopBackOff	5 (30s ago)	2m12s

```

user@devopsiu5:~$ cat <<EOF | kubectl apply -f-
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
        livenessProbe:
          httpGet:
            path: /healthz
            port: 80
          initialDelaySeconds: 120
          periodSeconds: 3
EOF

```

NAME	READY	STATUS	RESTARTS	AGE
netshoot	1/1	Running	0	26m
nginx-deployment-6c77b67cdc-6kt9k	1/1	Running	2 (7s ago)	32s
nginx-deployment-6c77b67cdc-c88bt	1/1	Running	2 (4s ago)	29s
nginx-deployment-6c77b67cdc-mn6bh	1/1	Running	2 (10s ago)	34s

```

user@devopsiu5:~$ curl localhost:80
С новым годом :D user@devopsiu5:~$

```

```

root@nginx-deployment-9488bb684-s2bsl:/# cat /etc/nginx/conf.d/student-server.conf
server {
    listen      80 default_server;

    location / {
        return 200 'С новым годом :D';
    }
}

```

На подах конфиг находится в /etc/nginx/conf.d, как указано в mountPath деплоймента

## Контрольные вопросы:

1. Что такое kubernetes и зачем он нужен?

Инструмент контейнерной оркестрации

2. Расскажите про основные компоненты kubernetes и покажите их в kube-systems

```
user@devopsiu5:~$ kubectl get pods -n kube-system
NAME                                READY   STATUS    RESTARTS   AGE
coredns-565d847f94-2vmvn           1/1     Running   0           36m
coredns-565d847f94-8768l           1/1     Running   0           36m
etcd-kind-control-plane             1/1     Running   0           39m
kindnet-qw6qt                      1/1     Running   1 (2m22s ago) 36m
kube-apiserver-kind-control-plane   0/1     Running   1 (2m30s ago) 39m
kube-controller-manager-kind-control-plane 1/1     Running   4 (4m9s ago)  39m
kube-proxy-8mk2m                   1/1     Running   0           36m
kube-scheduler-kind-control-plane   1/1     Running   3 (33m ago)  39m
```

CoreDNS – DNS-сервер кластера

Etcd – хранилище всех данных кластера

Kindnet – сетевой интерфейс контейнеров

API-server – клиентская часть панели управления

Controller manager – запускает процессы контроллера

Proxy – сетевой прокси, конфигурирует правила сети на узлах

Scheduler – планирует развертывание подов на nodes

3. Расскажите про создание пода из yaml, основные поля.

Версия API – версия kube-api

Тип ресурса – pod, service, deployment и т.д.

Metadata – системные настройки, в т.ч. название

Спецификация ресурса – имя, образ, порты контейнеров

Также могут быть liveness, readiness пробы, ограничения по ресурсам, подключение томов

4. Расскажите про сервисы ClusterIp и Headless

В случае ClusterIP на сервис выделяется постоянный ip, в случае Headless будут выделяться ip доступных подов

5. Расскажите про сервисы NodePort и LoadBalancer

NodePort связывает порт контейнера с портом сервера, LoadBalancer связывает выделенный ip и поды напрямую

6. Расскажите про Ingress

Демон, который управляет внешним доступом к сервисам

## 7. Расскажите про Deployment, StatefulSet и DaemonSet

Deployment позволяет поднимать группу подов, поддерживать их численность и автоматически обновлять.

StatefulSet позволяет дополнительно сохранять идентичность подов.

DemonSet запускает на каждой ноде один экземпляр приложения.

## 8. Расскажите про Job и CronJob

Job создает под для ограниченной по времени задачи, CronJob периодически создает Job

## 9. Расскажите про ConfigMap и Secret

ConfigMap – ресурс с конфигурацией для пода

Secret дополнительно хранится в зашифрованном виде

## 10. Перечислите основные команды kubectl, рассмотренные в лабе

kubectl create cluster – создание кластера

kubectl get <имя ресурса> -n <namespace> – список ресурсов в namespace

kubectl apply – создание ресурса из yaml

kubectl -n <namespace> <под> - логи пода

kubectl edit <ресурс> - изменение конфигурации ресурса

kubectl delete <ресурс> - удаление ресурса

kubectl describe <ресурс> - описание ресурса

kubectl label <под> <название лейбла>=<лейбл> - добавить лейбл к поду

kubectl exec -it <под> <команда> - запустить команду в поде