

Задание номер 12:
«Установка и настройка кластера Kubernetes»

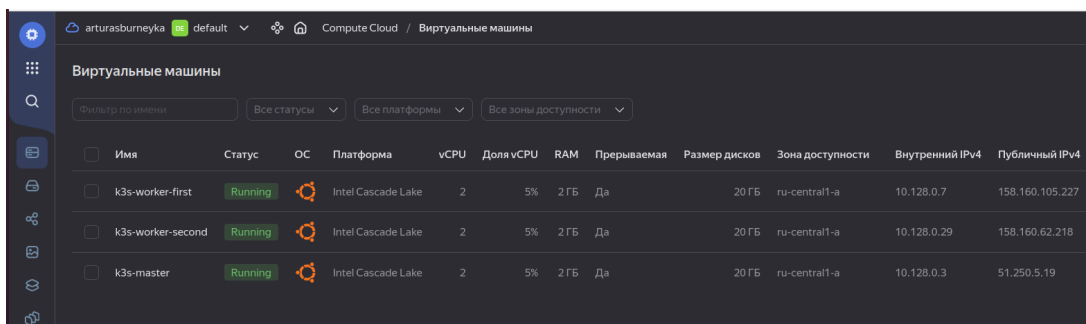
Выполнил:

Слушатель курса «OTUS». «Инфраструктура высоконагруженных систем»
Бурнейка Артурас Витаутасович

Для целей настоящей работы в Yandex Cloud создаются 3 (три) VM:

- k3s-master
- k3s-worker-first
- k3s-worker-second

С помощью Terraform создаётся указанная инфраструктура. Результаты развёртывания представлены на рисунке:



Имя	Статус	ОС	Платформа	vCPU	Доля vCPU	RAM	Прерываемая	Размер дисков	Зона доступности	Внутренний IPv4	Публичный IPv4
k3s-worker-first	Running	Linux	Intel Cascade Lake	2	5%	2 ГБ	Да	20 ГБ	ru-central1-a	10.128.0.7	158.160.105.227
k3s-worker-second	Running	Linux	Intel Cascade Lake	2	5%	2 ГБ	Да	20 ГБ	ru-central1-a	10.128.0.29	158.160.62.218
k3s-master	Running	Linux	Intel Cascade Lake	2	5%	2 ГБ	Да	20 ГБ	ru-central1-a	10.128.0.3	51.250.5.19

В качестве дистрибутива kubernetes выбран сертифицированный дистрибутив k3s (начинал разрабатываться компанией Rancher, в настоящий момент поддерживается Colud Native Computer Foundation).

Дистрибутив является легковесным и прекрасно подходит (но не ограничивается) для развёртывания на одной ноде, для интернета вещей (IoT), для CI.

Привожу ссылку на официальный сайт и документацию:

<https://k3s.io/>

<https://docs.k3s.io/>

Для установки мастера (первого или единственного) и инициализации кластера k3s применена команда:

```
sudo curl -sL https://get.k3s.io | sh -s - server \
--token "c5c77161-d5a6-4c77-87bd-2a5ca9e2706e" \
--write-kubeconfig-mode 644 \
--cluster-init
```

В качестве токена использован UUID, сгенерированный с помощью онлайн-сервиса <https://www.uuidgenerator.net/>.

Бинарный файл k3s расположен по пути:
/usr/local/bin/k3s

Инструмент командной строки (kubectl) расположен по пути:
/usr/local/bin/kubectl

Скрипты для удаления всех объектов (абстракций) кластера `kubernetes` и удаления самого кластера расположены по пути:

```
/usr/local/bin/k3s-killall.sh
/usr/local/bin/k3s-uninstall.sh
```

Systemd unit файл для k3s расположен по пути:
/etc/systemd/system/k3s.service

После применения команды на установку и инициализацию кластера на мастере сервис k3s запущен. Результаты представлены на рисунке:

```

about@k3s-master:~$ sudo systemctl status k3s.service
● k3s.service - Lightweight Kubernetes
   Loaded: loaded (/etc/systemd/system/k3s.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2024-04-13 16:16:03 UTC; 9min ago
     Docs: https://k3s.io
  Process: 1449 ExecStartPre=/bin/sh -xc /usr/bin/systemctl is-enabled --quiet nm-cloud-setup.service 2>/dev/null (code=exited, status=0/SUCCESS)
  Process: 1451 ExecStartPre=/sbin/modprobe br_netfilter (code=exited, status=0/SUCCESS)
  Process: 1457 ExecStartPre=/sbin/modprobe overlay (code=exited, status=0/SUCCESS)
 Main PID: 1459 (k3s-server)
    Tasks: 89
   Memory: 1.2G
      CPU: 1min 48.806s
   CGroup: /system.slice/k3s.service
           └─1459 /usr/local/bin/k3s server"
           └─1513 containerd
           └─2144 /var/lib/rancher/k3s/data/3fcd4fc3ae2ba4d577d4ee08ad7092538cd7a7f0da701efa2a8807d44a25f66/bin/containerd-shim-runc-v2 --namespace k8s.
           └─2157 /var/lib/rancher/k3s/data/3fcd4fc3ae2ba4d577d4ee08ad7092538cd7a7f0da701efa2a8807d44a25f66/bin/containerd-shim-runc-v2 --namespace k8s.
           └─2254 /var/lib/rancher/k3s/data/3fcd4fc3ae2ba4d577d4ee08ad7092538cd7a7f0da701efa2a8807d44a25f66/bin/containerd-shim-runc-v2 --namespace k8s.
           └─3212 /var/lib/rancher/k3s/data/3fcd4fc3ae2ba4d577d4ee08ad7092538cd7a7f0da701efa2a8807d44a25f66/bin/containerd-shim-runc-v2 --namespace k8s.
           └─3261 /var/lib/rancher/k3s/data/3fcd4fc3ae2ba4d577d4ee08ad7092538cd7a7f0da701efa2a8807d44a25f66/bin/containerd-shim-runc-v2 --namespace k8s.

```

Инструмент управления кластером kubernetes доступен. Состав кластера после установки первого мастера представлен на рисунке:

```
ubuntu@k3s-master:~$ kubectl version
Client Version: v1.29.3+k3s1
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
Server Version: v1.29.3+k3s1
ubuntu@k3s-master:~$ kubectl get nodes

```

NAME	STATUS	ROLES	AGE	VERSION
k3s-master	Ready	control-plane,etcd,master	16m	v1.29.3+k3s1

```
ubuntu@k3s-master:~$
```

Далее будет производиться присоединение агентов (worker nodes) к единственному мастеру.

Для этого потребуется токен, сгенерированный при установке и инициализации первого мастера.

Токен расположен по пути:
/var/lib/rancher/k3s/server/token

Содержимое токена представлено на рисунке:

```
cat: /var/lib/rancher/k3s/server/token: Permission denied
ubuntu@k3s-master:~$ sudo cat /var/lib/rancher/k3s/server/token
K10ba24ee13b76e2299479ea84bb048f1aa627b935662de572f10578ed78fb7e1e::server:c5c77161-d5a6-4c77-87bd-2a5ca9e2706e
ubuntu@k3s-master:~$
```

На виртуальных машинах k3s-worker-first и k3s-worker-second применена команда для присоединения к мастеру (k3s-master):

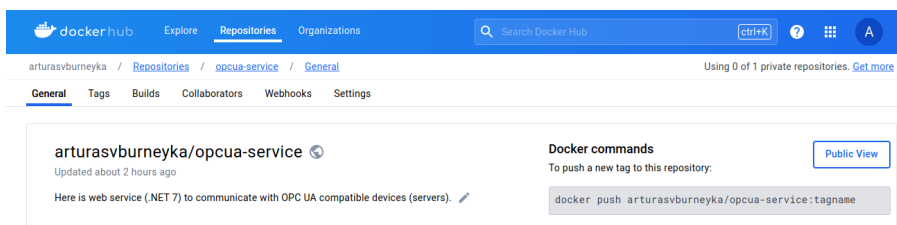
```
sudo curl -sL https://get.k3s.io | sh -s - agent \
--server https://10.128.0.3:6443 \
--token
"K10ba24ee13b7eb2299479ea84bb048f1aa627b9356662de572f10578ed78fb7e1e::s
erver:c5c77161-d5a6-4c77-87bd-2a5ca9e2706e"
```

В результате применённых команд кластер kubernetes в составе одного мастера и двух рабочих узлов установлен и настроен. Состав узлов кластера представлен на рисунке:

```
ubuntu@k3s-master:~$ kubectl get nodes
NAME                STATUS    ROLES                  AGE   VERSION
k3s-master          Ready    control-plane,etcd,master   37m   v1.29.3+k3s1
k3s-worker-first    Ready    <none>                   3m38s v1.29.3+k3s1
k3s-worker-second   Ready    <none>                   65s   v1.29.3+k3s1
ubuntu@k3s-master:~$
```

На следующем этапе будет произведена настройка pod и service для обеспечения доступа к бэкэнд сервису.

Docker образ с бэкэндом предварительно был загружен в регистр Docker Hub и доступен для публичного доступа и скачивания. Реквизиты образа приведены на рисунке:



Pod успешно запущен на узле k3s-worker-first (10.128.0.7). Результаты представлены на рисунке:

```
ubuntu@k3s-master:~$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
opcu-client-pod     0/1     ContainerCreating   0          17s
ubuntu@k3s-master:~$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
opcu-client-pod     1/1     Running      0          18s
ubuntu@k3s-master:~$ kubectl describe pods opcu-client-pod
Name:               opcu-client-pod
Namespace:          default
Priority:            0
Service Account:    default
Node:               k3s-worker-first/10.128.0.7
Start Time:         Sun, 14 Apr 2024 14:23:14 +0000
Labels:             <none>
Annotations:        <none>
Status:             Running
IP:                 10.42.1.11
IPs:                10.42.1.11
Containers:
  opcu-client:
    Container ID:   containerd://347dcd0dd7dc6a524c3af6f8e64f876f2dbd567829ee83a09b8ca8395e4fba6
    Image:          arturasvburneyka/opcu-service:latest
    Image ID:       docker.io/arturasvburneyka/opcu-service@sha256:06e13668658b206677051c08ec87954c0d175f75d2af1980b91df16d722c432f
    Port:          <none>
    Host Port:     <none>
    State:         Running
      Started:     Sun, 14 Apr 2024 14:23:32 +0000
    Ready:         True
    Restart Count:  0
    Environment:   <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-cknbz (ro)
```

Из описания пода `opcua-client-pod` следует, что сервис доступен по внутреннему IP адресу `10.42.1.11` кластера `kubernetes`.

Для публичного доступа к сервису будет создан объект `Service`.

Для решения обеих задач совместно реализован манифест для создания ресурсов (`Pod` + `Service`).

Файл манифеста: `opcua-client.yaml` (находится в репозитории)

После применения манифеста оба ресурса успешно созданы. Результаты представлены на рисунке:

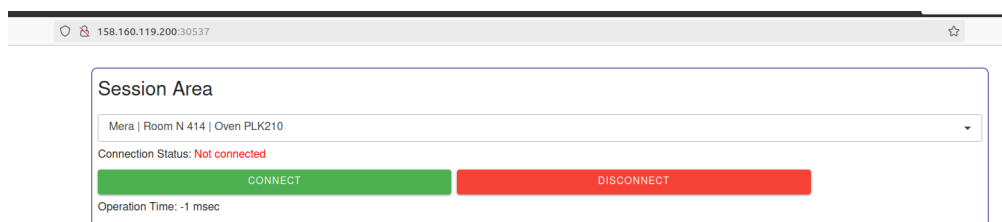
```
ubuntu@k3s-master:~$ kubectl apply -f opcua-client.yaml
pod/opcua-client-pod created
service/opcua-client-service created
ubuntu@k3s-master:~$ kubectl get nodes,pods,deployments,services
NAME                                STATUS    ROLES                                AGE    VERSION
node/k3s-master                     Ready     control-plane,etcd,master           23h    v1.29.3+k3s1
node/k3s-worker-first               Ready     <none>                              22h    v1.29.3+k3s1
node/k3s-worker-second              Ready     <none>                              22h    v1.29.3+k3s1

NAME                                READY    STATUS    RESTARTS   AGE
pod/opcua-client-pod               1/1     Running   0          5s

NAME                                TYPE     CLUSTER-IP    EXTERNAL-IP  PORT(S)          AGE
service/kubernetes                 ClusterIP  10.43.0.1      <none>        443/TCP          23h
service/opcua-client-service       NodePort  10.43.239.234 <none>        6537:30537/TCP   5s
```

Тип созданного сервиса – `NodePort`, что означает публичную доступность к сервису на указанном порту (`30537`) виртуальной машины в `Yandex Cloud`.

Произведена проверка доступности развёрнутого бэкэнд сервиса. Сервис в публичном доступе. Результаты представлены на рисунке:



В указанном примере доступ к сервису произведен по публичному адресу мастера (`k3s-master`) – `158.160.119.200`.

Также развёрнутый в кластере `kubernetes` сервис доступен по публичным адресам обеих рабочих узлов (`k3s-worker-first`, `k3s-worker-second`).

Выводы: кластер `kubernetes` в составе одного мастера и двух рабочих узлов развёрнут; подготовлен и опубликован образ с целевым сервисом (`OPC UA Service`); создан `Pod` и связанный с ним `Service`; бэкэнд доступен публично по любому из адресов узлов кластера.

Спасибо OTUS за интересное задание и новый навык!