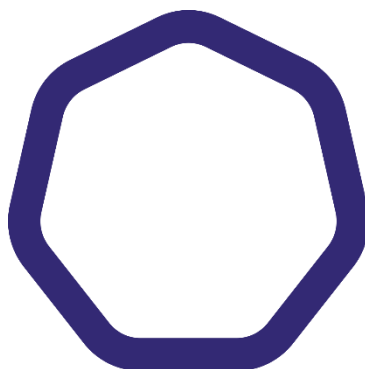


Министерство цифрового развития, связи и массовых коммуникаций
Российской Федерации
Ордена Трудового Красного Знамени федеральное государственное
бюджетное образовательное учреждение высшего образования
«Московский технический университет связи и информатики»
(МТУСИ)



Кафедра «Сетевые информационные технологии и сервисы»

Лабораторная работа №1
на тему «Создание кластера Kubernetes»

Выполнил:
Студенты 1 курса
Группы М092401(75)
Цыганков Р.О.
Проверил:
к.т.н., Шалагинов А. В.

Москва 2024 г.

Цель работы: изучение структуры Kubernetes, исследование ее абстракции в представлении кластера и получение более глубокое понимания Kubernetes.

Выполнение работы:

Для выполнения лабораторных работ необходимо:

1. Зайти на сайт: <https://learning.kasten.io/profile/>
2. Пройти процесс регистрации.
3. Перейти в свой профиль.
4. В разделе «Больше лабораторий для изучения» выбрать лабораторную работу «Создание резервной копии вашего приложения Kubernetes».
5. Нажать на кнопку «Start Lab».

Данный курс состоит из двух частей:

- Теоретический блок, на основании которого созданы вопросы из теста.
- Практический блок.

Теоретический блок:

- 1) Кластер Kubernetes состоит из узла панели управления Kubernetes и одного или нескольких управляемых узлов
- 2) Плоскость управления Kubernetes выполняет следующие функции:
 1. Постоянно сравнивает текущее состояние с желаемым состоянием;
 2. Выполняет инструкции по изменению состояния кластера.
- 3) Самым маленьким разворачиваемым объектом Kubernetes является модуль (Pod)
- 4) При горизонтальном масштабировании изменяется количество модулей (Pods)
- 5) Развертывание (Deployment) управляет созданием и обновлением модулей (Pods), управлением ReplicaSet
- 6) Сервис (Service) – абстракция, определяющая набор модулей (Pods)

7) Пространства имен обеспечивают возможность использования кластера различными пользователями и разделения имен (имена должны быть уникальны только внутри одного пространства имен)

8) Том (Volume) в Kubernetes - каталог для данных, доступных для контейнеров в модуле (Pod)

9) Работы (Job) в Kubernetes работает до завершения

10) Типичные варианты использования DaemonSet – службы (daemons) мониторинга, хранилища, сбора лог-файлов

11) StatefulSets используются для описания порядка запуска модулей (Pods), обеспечения стабильных и уникальных сетевых идентификаторов

Практический блок

Практическая часть лабораторной работы выполнялась на 3 виртуальных машинах (далее ВМ) в локальной сети с установленной на всех Ubuntu 18.04. Одна виртуальная машина является панелью управления.

1. На первом шаге необходимо на каждую ВМ загрузить открытый ключ подписи Google Cloud, добавить Kubernetes apt репозиторий, обновить индекс пакета apt и установить необходимые утилиты:

- Kubelet: основной агент, который работает на узле kubernetes и взаимодействует с панелью управления, чтобы отдавать команды узлу и сообщать о его состоянии;
- Kubeadm: инструмент, который выполняет действия, необходимые для создания кластера Kubernetes;
- Kubectl: Инструмент командной строки, который позволяет вам управлять кластерами Kubernetes и отдавать команды API-серверу;
- Docker Engine: контейнерный движок, используемый Kubernetes для запуска контейнеров.

```
instruqt >_ Control Plane >_ kube-02 >_ kube-03

root@kube-01:~# curl -fsSL /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg
root@kube-01:~# echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main
root@kube-01:~# apt update
Hit:1 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://europe-west1.gce.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu bionic-security InRelease
Get:4 https://packages.cloud.google.com/apt kubernetes-xenial InRelease [9383 B]
Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 Packages [59.4 kB]
Fetched 68.8 kB in 1s (109 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
183 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@kube-01:~# apt install -y kubelet=1.20.0-00 kubeadm=1.20.0-00 kubectl=1.20.0-00 docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libnuma1
Use 'apt autoremove' to remove it.
The following additional packages will be installed:
  bridge-utils conntrack containerd cri-tools kubernetes-cni pigz runc socat ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils conntrack containerd cri-tools docker.io kubeadm kubectl kubelet kubernetes-cni pigz runc socat
```

Рисунок 1

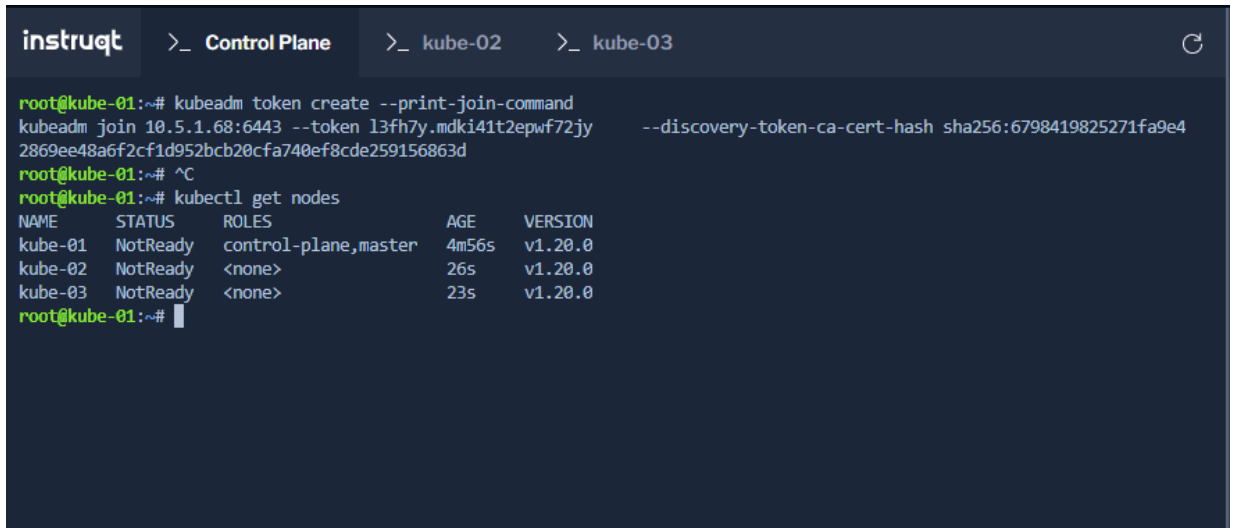
2. На втором шаге происходит настройка панели управления Kubernetes. На каждой ВМ необходимо установить Docker, инициализировать панель управления, создать файл конфигурации

```
instruqt >_ Control Plane >_ kube-02 >_ kube-03

root@kube-03:~# mkdir /etc/docker
mkdir: cannot create directory '/etc/docker': File exists
root@kube-03:~# cat <<EOF | sudo tee /etc/docker/daemon.json
> {
>   "exec-opts": ["native.cgroupdriver=systemd"],
>   "log-driver": "json-file",
>   "log-opts": {
>     "max-size": "100m"
>   },
>   "storage-driver": "overlay2"
> }
> EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
root@kube-03:~#
root@kube-03:~# systemctl enable docker
root@kube-03:~# systemctl daemon-reload
root@kube-03:~# systemctl restart docker
```

Рисунок 2

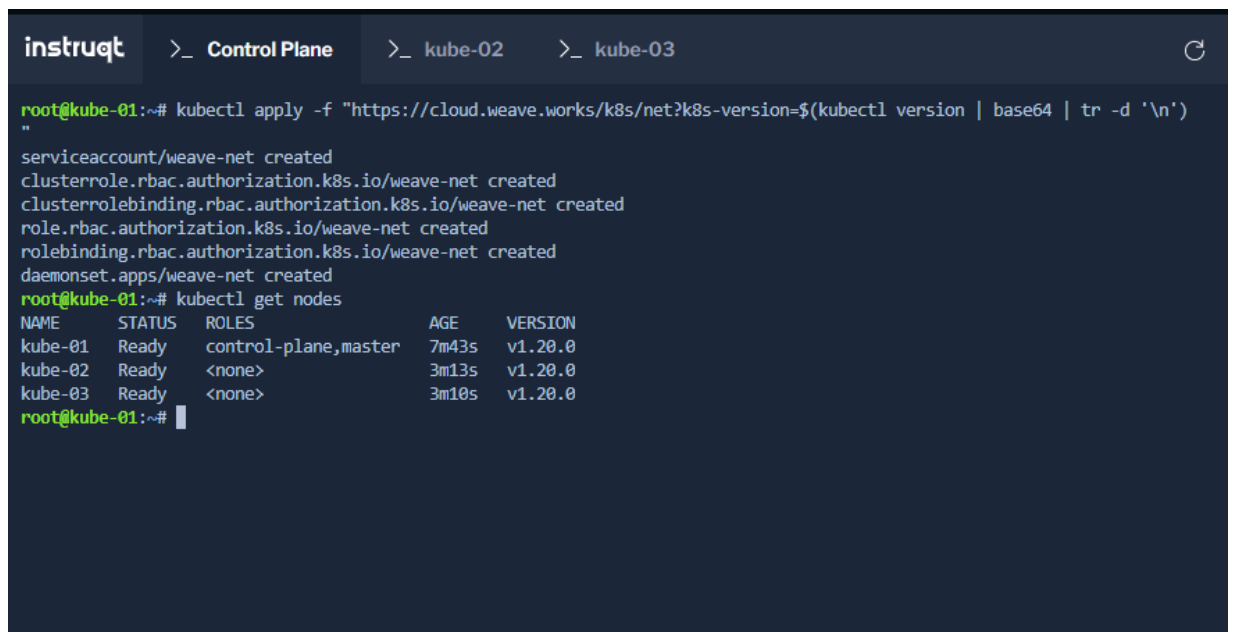
3. На третьем шаге происходит подключение узлов к панели управления.



```
instruqit >_ Control Plane >_ kube-02 >_ kube-03
root@kube-01:~# kubeadm token create --print-join-command
kubeadm join 10.5.1.68:6443 --token l3fh7y.mdk141t2epwf72jy --discovery-token-ca-cert-hash sha256:6798419825271fa9e42869ee48a6f2cf1d952bcb20cfa740ef8cde259156863d
root@kube-01:~# ^C
root@kube-01:~# kubectl get nodes
NAME        STATUS    ROLES                  AGE      VERSION
kube-01     NotReady  control-plane,master   4m56s    v1.20.0
kube-02     NotReady  <none>                  26s      v1.20.0
kube-03     NotReady  <none>                  23s      v1.20.0
root@kube-01:~#
```

Рисунок 3

4. На четвертом шаге происходит сетевая настройка кластера Kubernetes с помощью Weave-net — подключаемого модуля CNI, который предоставляет виртуальную сеть, соединяющую док-контейнеры на нескольких хостах.



```
instruqit >_ Control Plane >_ kube-02 >_ kube-03
root@kube-01:~# kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')
"
serviceaccount/weave-net created
clusterrole.rbac.authorization.k8s.io/weave-net created
clusterrolebinding.rbac.authorization.k8s.io/weave-net created
role.rbac.authorization.k8s.io/weave-net created
rolebinding.rbac.authorization.k8s.io/weave-net created
daemonset.apps/weave-net created
root@kube-01:~# kubectl get nodes
NAME        STATUS    ROLES                  AGE      VERSION
kube-01     Ready     control-plane,master   7m43s    v1.20.0
kube-02     Ready     <none>                  3m13s    v1.20.0
kube-03     Ready     <none>                  3m10s    v1.20.0
root@kube-01:~#
```

Рисунок 4

5. На пятом шаге происходит настройка демонстрационная настройка микросервиса «Магазин носков» с помощью копирования репозитория и развертывания в Kubernetes.

Заключение

В процессе лабораторной работы был создан кластер Kubernetes. Было показано как использовать инструмент командной строки Kubernetes kubectl. Затем было развернуто приложение микросервисов Weave Socks Shop в качестве демонстрации