Инструкция по разворачиванию k8s

Все описываемые в руководстве команды выполняются от пользователя с правами root.

ОС Alt Server 10.4

Для редактирования текста используется редактор nano, можно использовать любой другой удобный для вас.

Настройка root прав (пример для пользователя test1):

Добавьте пользователя test1 в группу wheel, используя команду:

usermod -aG wheel test1

отредактируйте файл /etc/sudoers с помощью специальной команды visudo.

Добавьте или раскомментируйте строку:

WHEEL_USERS ALL=(ALL:ALL) ALL

Сохраните изменения и закройте редактор.

В составе тестового кластера 3 ВМ:

Alt-server-1 – Мастер-Нода, nginx
Alt-server-2 – микросервисы, rabbitmq
Alt-server-3 - pgsql,redis-fpo

На всех ВМ k8s выполняем:

Отключение swap:

swapoff -a
sed -i '/ swap / s/^\(.*\)$/#\1/g' /etc/fstab

Установка необходимых пакетов:

apt-get update && apt-get dist-upgrade -y
apt-get install -y kubernetes1.27-kubelet
apt-get install -y cri-tools1.27
apt-get install -y kubernetes1.27-crio
systemctl enable --now kubelet kube-proxy crio

Устанавливаем только на Мастер-Ноде(Alt-Server-1):

apt-get install -y kubernetes1.27-kubeadm

Запуск кластера k8s(выполняется на Мастер-ноде).

На Мастер ВМ (в нашем случае Alt-server-1) выполнить:

kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=SystemVerification

Для добавления ноды в кластер выполнить (пример для ВМ Alt-server-2):

kubeadm join 10.205.0.91:6443 --token fafk6z.4ixtlmhti0z9fuu2 --discovery-token-ca-cert-hash \
    sha256:db10892eab7a1f97242bbdab79e82ef24e809a8bb84692aedc7a22d094a0b8db

Настройка Kubernetes для работы от пользователя:

mkdir ~/.kube
cp /etc/kubernetes/admin.conf ~/.kube/config
chmod 600 ~/.kube/config

Справка по CoreDNS:

Если не запустились CoreDNS сервисы, выполните:

export EDITOR=nano

kubectl edit configmap coredns -n kube-system

При необходимости укажите публичный DNS, например 8.8.8.8

Проверка nod и pod:

Проверка node:

kubectl get nodes -o wide

```
[root@alt-server-1 rnisk8s]# kubectl get nodes -o wide
NAME           STATUS   ROLES           AGE     VERSION    INTERNAL-IP   EXTERNAL-IP
alt-server-1   Ready    control-plane   5h11m   v1.27.16   10.205.0.90   <none>
alt-server-2   Ready    <none>          5h1m    v1.27.16   10.205.0.91   <none>
```

Проверка pod:

kubectl get pods --namespace kube-system

```
[root@alt-server-1 rnisk8s]# kubectl get pods --namespace kube-system
NAME                                   READY   STATUS    RESTARTS   AGE
coredns-5d78c9869d-lmsfd               1/1     Running   0          122m
coredns-5d78c9869d-q9vvk               1/1     Running   0          3h19m
etcd-alt-server-1                      1/1     Running   0          5h14m
kube-apiserver-alt-server-1            1/1     Running   0          5h14m
kube-controller-manager-alt-server-1   1/1     Running   0          5h14m
kube-proxy-24w78                       1/1     Running   0          5h4m
kube-proxy-qbvwx                       1/1     Running   0          5h13m
kube-scheduler-alt-server-1            1/1     Running   0          5h14m
```

Разворачивание сети:

kubectl apply -f https://gitea.basealt.ru/alt/flannel-manifests/raw/branch/main/c10f2/0/25/1/kube-flannel.yml

Установка сервиса etcd на Alt-server-2:

apt-get install etcd

systemctl start etcd

Проверяем что сервис запустился:

systemctl status etcd

Заходим в Конфиг etcd текстовым редактором:

nano /etc/etcd/etcd.conf

Добавляем следующую настройку или меняем существующую:

ETCD_LISTEN_CLIENT_URLS="http://0.0.0.0:2379"

Нажимаем Ctrl+X, Y ,Enter

systemctl restart etcd

на Alt-Server-1 установка Nginx:

apt-get install nginx
systemctl start nginx
systemctl enable nginx

Открываем на редактирование nginx.conf:

nano etc/nginx/nginx.conf

Удаляем содержимое конфига и вставляем содержимое приложенного к руководству nginx.conf

Сохраняем ctrl+x, y , enter

systemctl restart nginx

Установка redis на Alt-Server-3:

apt-get install redis

Запускаем redis:

systemctl start redis
systemctl enable redis

Редактируем конфигурацию чтобы redis слушал всех.

Открываем конфиг nano /etc/redis/redis.conf и добавляем строку:

bind 0.0.0.0 -::1

Установка PostgreSQL(на Alt-Server-3)

Установка пакетов:

apt-get install postgresql15-server
apt-get install postgresql15-contrib
apt-get install postgresql15-postgis

Распаковка архива БД(Прикладывается к инструкции):

Копируем файл dump.sql.gz по пути /var/lib/pgsql/

Распаковываем командой gunzip dump.sql.gz

Восстанавливаем дамп в Postgresql:

psql -U postgres -f /var/lib/pgsql/dump.sql

Инициализация базы:

/etc/init.d/postgresql initdb

systemctl start postgresql

Для создания нужных расширений базы данных создайте файл/var/lib/pgsql/extensions.sql со следующим содержимым:

```
#!/bin/bash
psql -U postgres << EOF
CREATE EXTENSION intarray;
CREATE EXTENSION dblink;
CREATE EXTENSION adminpack;
CREATE EXTENSION pg_stat_statements;
CREATE EXTENSION pgstattuple;
CREATE EXTENSION postgis;
CREATE EXTENSION postgis_topology;
CREATE EXTENSION if not exists "uuid-ossp";
EOF
```

Запустите скрипт:

su - postgres -s /bin/bash -c /var/lib/pgsql/extensions.sql

Редактирование файла pg_hba.conf: Файл находится по пути /var/lib/pgsql/data/.

Необходимо предоставить доступ ко всем БД для всех пользователей в вашей сети, пример:

В самом конце файла pg_hba.conf добавляем строку:

```
host    all             all             10.0.0.0/24             trust
```

После systemctl restart postgresql

На Alt-Server-1 создаем namespace:

kubectl create namespace rnis

Установка Helm:

apt-get install helm

Установка микросервисов при помощи helm(выполнять команды на Alt-Server-1):

Создаем папку /etc/rnisk8s/:

mkdir /etc/rnisk8s/

Переходим в эту папку:

cd /etc/rnisk8s/

*все дальнейшие действия по установке сервисов выполняются, не выходя из этой папки***.**

Распаковываем предоставленный архив системы РНИС :

tar -xzvf rnisk8s.tar.gz

В папке rnisk8s должна появится папка fpo

Разворачивание rabbitmq в контейнере:

helm upgrade --install rabbitmq -n rnis ./fpo/rabbitmq/

Необходимо создать пользователей в rabbitmq для микросервисов,список необходимых пользователей :

auth

api

dictionary

organizational-units

system

Заходим контейнер с rabbitmq командой:

kubectl exec -it rabbitmq-7b8f8d9c6f-xyz12 -n rnis -- /bin/bash

apt update

apt install vim(можно любой другой текстовый редактор)

Создаем файл users.sh и копируем в него скрипт:

```
#!/bin/bash

[[ $EUID -ne 0 ]] && { echo "Рекомендуется запуск с правами root"; exit 1; }

command -v rabbitmqctl &>/dev/null || { echo "    rabbitmqctl не найден"; exit 1; }

USERS=(auth api dictionary organizational-units system guest)


echo "    Создание пользователей RabbitMQ..."

for u in "${USERS[@]}"; do

  rabbitmqctl add_user "$u" "$u" && \

  rabbitmqctl set_user_tags "$u" administrator && \

  rabbitmqctl set_permissions -p / "$u" ".*" ".*" ".*" && \

  echo "    Создан: $u/$u"

done


echo -e "\n    Список пользователей:"; rabbitmqctl list_users

echo -e "\nГотово!"
```

Сохраняем файл**.**

Делаем скрипт исполняемым:

```
chmod +x users.sh
```

Запускаем скрипт

./users.sh

Создание Alias на redis

Редактирование файла конфигураций сервис redis:

nano /etc/rnisk8s/fpo/redis/templates/endpoints.yaml

В файле нужно прописать Адрес ВМ где установлен Redis в нашем примере это Alt-Server-3:

```
---
apiVersion: v1
kind: Endpoints
metadata:
  name: redis
  namespace: rnis
subsets:
  - addresses:
      - ip: 10.205.0.92 #ip адрес сервера Alt-server-3
    ports:
      - port: 6379
```

Список минимальных сервисов необходимых для работы системы РНИС:

etcd
features
rbac
api
auth
dictionary
ogranizational_units
nginx-frontend

Выполняем команды для установки необходимых микросервисов:

helm upgrade --install features -n rnis ./fpo/features/

helm upgrade --install rbac -n rnis ./fpo/rbac/

helm upgrade --install api -n rnis ./fpo/api/

helm upgrade --install auth -n rnis ./fpo/auth/

helm upgrade --install dictionary -n rnis ./fpo/dictionary/

helm upgrade --install ogranizational_units -n rnis ./fpo/ogranizational_units/

helm upgrade --install nginx-frontend -n rnis ./fpo/nginx-frontend/

Проверяем что сервисы развернуты:

kubectl get pod -n rnis

```
[root@alt-server-1 rnisk8s]# kubectl get pod -n rnis
NAME                                          READY   STATUS    RESTARTS        AGE
api-deployment-84886675-x7ppz                 1/1     Running   1 (4d20h ago)   4d20h
auth-deployment-8b485cccb-brfjc               1/1     Running   0               26h
dictionary-deployment-7975945964-rxbbx        1/1     Running   0               5d21h
features-deployment-6f6c68d49d-9stfm          1/1     Running   5 (5d22h ago)   5d22h
nginx-frontend-deployment-86b547cd7b-fzlfj    1/1     Running   0               25h
organizational-units-deployment-847d6f569c-qnxdd   1/1     Running   0          5d21h
rabbitmq-deployment-64688c96c4-vbzgt          1/1     Running   0               5d22h
rbac-deployment-cb56f4559-b7d9b               1/1     Running   0               5d20h
system-deployment-699fdf786d-sbsfv            1/1     Running   0               4d21h
```

Настройка работы интерфейса системы РНИС

В комплекте с инструкцией идет файл settings. json, его нужно положить на Alt-server-2 , по пути:

var/lib/containers/front/

Пробуем подключиться к интерфейсу:

Описание для ОС WIndows. Добавляем в hosts(находится по пути C:\Windows\System32\ drivers\etc) адрес ВМ на которой развернут nginx, пример:

10.205.0.90 test.t1-group.ru

После этого пробуем зайти через браузер по адресу http://test.t1-group.ru/

Если вы видите следующую картинку,значит все получилось успешно: