

K8S 02: Kubernetes. Basics. Minikube

Описание:

Kubernetes можно установить во множестве конфигураций, мы начнем с самой простой — All-in-one. Для этого воспользуемся minikube. Инсталляционные пакеты доступны для всех популярных ОС. Руководство по запуску вы сможете найти здесь — [Minikube - Getting Started](#).

Итак, давайте запустим minikube на ОС ubuntu.

Для запуска нам необходима любая система виртуализации. Мы воспользуемся VirtualBox. В руководстве по запуску minikube указано, что версия VirtualBox должна быть ≥ 5.2 . У нас в системе, без подключения дополнительных репозиторийев, доступна именно эта версия 5.2.32-dfsg-0~ubuntu18.04.1 (актуально для ubuntu 18.04).

Устанавливаем:

```
sudo apt install virtualbox
# Устанавливаем minikube
curl -LO
https://storage.googleapis.com/minikube/releases/latest/minikube_1.5.2
.deb \
  && sudo dpkg -i minikube_1.5.2.deb

# Проверяем, что на нашем хосте доступна и активирована виртуализация
egrep -q 'vmx|svm' /proc/cpuinfo && echo yes || echo no

# Запускаем minikube
minikube start --vm-driver=virtualbox
```

Отлично!

Теперь совершим обзорную экскурсию по кластеру. Для доступа к кластеру из консоли потребуется утилита kubectl.

Данная утилита доступна для всех современных ОС. Инсталляционные пакеты есть на [официальном сайте](#).

Взглянем на конфигурацию кластера:

```
kubectl config view
kubectl cluster-info
```

Также информацию о кластере можно получить через веб-интерфейс. Запустим его:

```
minikube dashboard
```

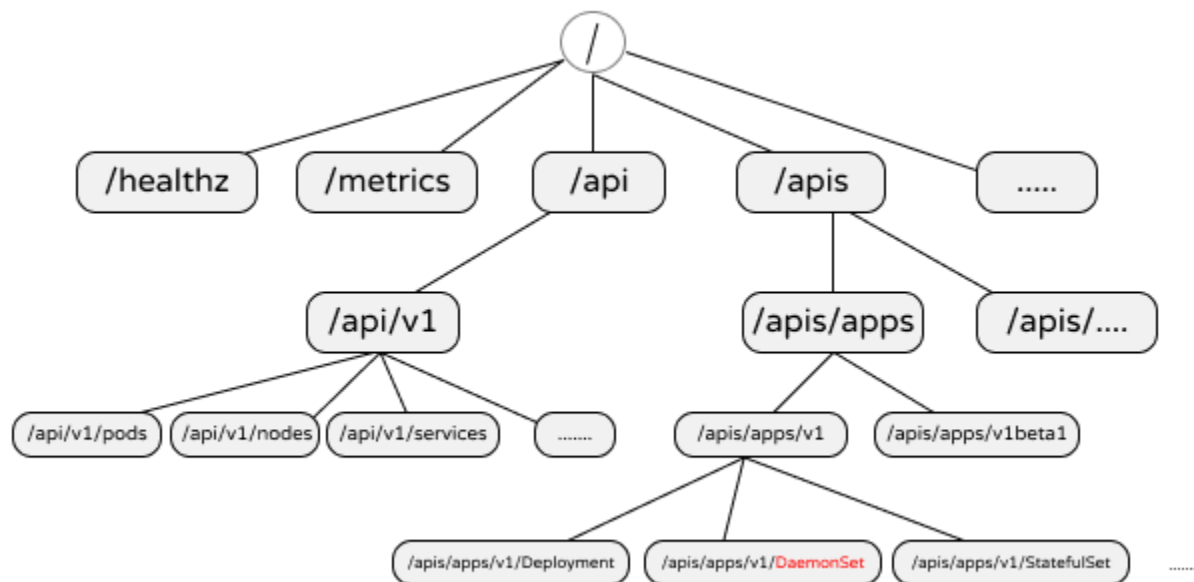
Доступ можно получить через встроенный в kubectl прокси-сервер:
`kubectl proxy`

Затем перейдем на страницу
<http://127.0.0.1:8001/api/v1/namespaces/kube-system/services/kubernetes-dashboard:/proxy/#!/overview?namespace=default>

В этом случае мы получаем доступ к API-серверу мастер-узла через прокси.
Посмотрим, что отдаст нам прокси при запросе корня:

```
curl http://localhost:8001/
```

О том, что такое Kubernetes API и из каких элементов он состоит, можно узнать в следующем видео:



- `/api/v1` - корневая группа API, там живут все основные объекты.
- `/apis/$name/$version` - именованные API, тут живут различные объекты, не входящие в корневую группу. Они могут иметь разный функционал, разную версию и стабильность работы (alpha/beta/stable). Например, `/apis/networking.k8s/v1`.
- `/healthz`, `/logs`, `/metrics` - общие API.

Полезные ссылки:

- [что такое API?](#)
- [Installing Kubernetes with Minikube](#)

- [kubectl config view \(docs\)](#)
- [Accessing the Kubernetes API](#)
- [Minikube in Kubernetes](#)
- [Kubectl in Kubernetes](#)
- [Setting Up Kubernetes Dashboard](#)
- [Kubernetes APIs tutorial and how to use the APIs](#)

Задание:

1. Для выполнения данного задания вам нужна VM. Создайте ее при помощи API или terraform. Доступы к аккаунтам опубликованы во вкладке Инфраструктура. Что бы ваша VM не была удалено через 3 дня, и вы не перенастраивали сервер постоянно, нужно добавить в параметрах метку "task: minikube" или "minikube" в зависимости от облачного провайдера. VM будет удалена через 20 дней. Можно также воспользоваться собственными аккаунтами или запустить локально на вашем ПК.
2. Запустите minikube на запущенной VM.
3. Найдите адрес API-сервера в выводе команды `kubectl config view`.
4. Составьте команду для получения адреса API-сервера из вывода `kubectl config view` - и запишите его в переменную `$APISERVER`. (команду сохраняем).
5. Выполните запрос с помощью утилиты `curl` к API-серверу по пути `$APISERVER/api` (команду и вывод сохраняем).
6. Вы увидите сообщение о том, что доступ запрещен. Получите токен для аутентификации с помощью команды `kubectl get secrets` и запишите его в переменную `$TOKEN`. (команду сохраняем).
7. Снова выполните запрос к API-серверу, но теперь уже с использованием токена. (команду и вывод сохраняем).
8. В качестве ответа приложите сохраненные команды и выводы.