

K8S 08: Kubernetes. Resources.

Введение

Описание:

Kubernetes имеет очень богатую модель объектов. Эти объекты представляют собой приложения, запросы на ресурсы, политики, правила и многое другое. С описанием каждого объекта мы сообщаем кластеру требуемое состояние посредством описания спецификации. А controller-manager, в свою очередь, инициирует изменения (создание/удаление ресурсов) в тех случаях, когда желаемое состояние не соответствует фактическому, непрерывно наблюдая за состоянием кластера, сравнивая результат с ожидаемым. Примеры объектов: Pod, ReplicaSet, Deployment, Namespace. Обычно спецификация объектов описывается в YAML, но можно послать JSON API-серверу. Вот пример описания объекта в YAML:

```
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:1.7.9
          ports:
            - containerPort: 80
```

Назначение всех полей мы изучим позже, сейчас разберем основные. Первая строка apiVersion задает требуемое API для обработки данного объекта. Вспомните структуру

API, которую отдавал API-сервер. Kind — задает тип объекта, а metadata — добавляет информацию к данному объекту, например, имя. Заметьте, есть два разных spec: spec и spec.template.spec. В spec задается желаемое состояние объекта. В данном случае мы запросили 3 реплики, которые будут созданы по описанию в spec.template.spec. Шаблон в spec.template.spec описывает состояние дочернего объекта (если таковой имеется). В текущем примере — это контейнер nginx:1.7.9. В последующих заданиях объекты Kubernetes мы рассмотрим более подробно.

Полезные ссылки:

- [Kubernetes API overview](#)

Задание:

1. Переведите YAML, указанный в описании задания, в JSON и отправьте его API-серверу через curl.
2. В ответе отправьте команду curl с json и полученный ответ сервера.