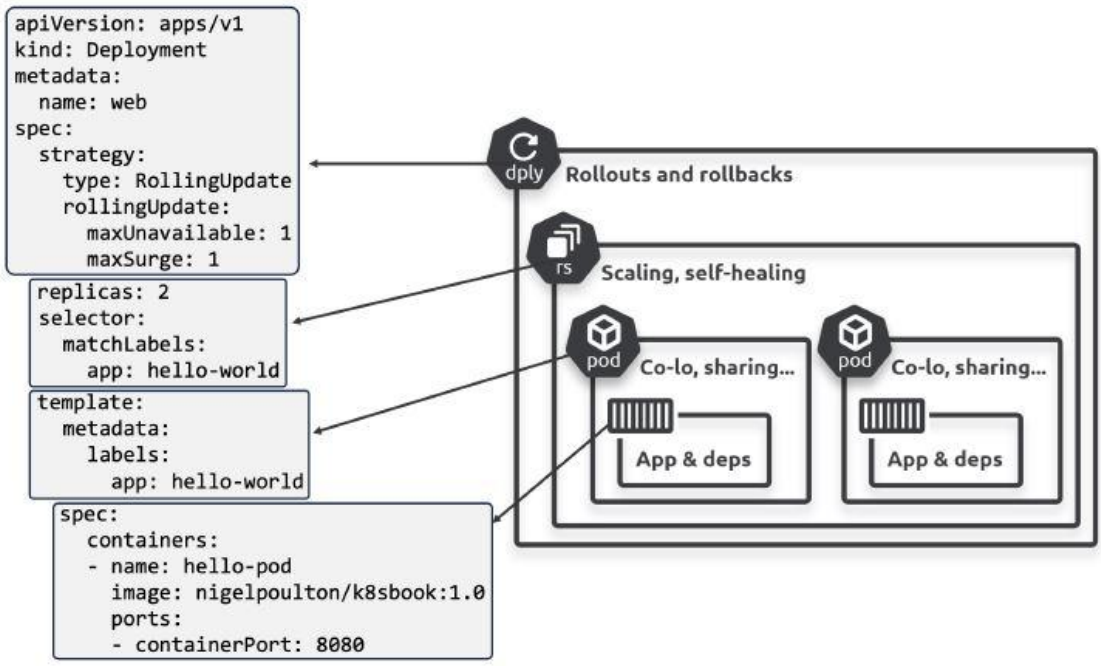
Deployment

Deployments наиболее популярный способ для запуска приложений без сохранения состояния. Это добавляет проверку состояния, масштабирование, восстановление.

Реализовано через deployment контроллер. Каждый контроллер управляет одним или несколькими одинаковыми подами.



Масштабирование (Scalling)

Существуют несколько типов

Тип	Описание
Horizontal Pod Autoscaler	Масштабирование количества подов, наиболее часто используется.
Vertical Pod Autoscaler	Масштабирование ресурсов, потребляемых подами. Не установлен по умолчанию. Редко используется
Cluster Autoscaler	Добавляет/удаляет ноды. По умолчанию, часто используется.

Например, указываем кол-во подов от 2 до 10. Нагрузка повысилась, и HPA запрашивает еще 2 пода. Они запускаются. Но нагрузка растет, и запрашивается еще 2 пода. Однако на существующем кластере нет возможности запустить еще 2 пода, и они переходят в статус Pending. CA определяет Pending поды и увеличивает количество нодов, запуская там поды. И наоборот.

Масштабирование связано с понятием текущего состояния (state). Есть необходимое состояние и наблюдаемое состояние. При неравенстве контроллер запускает процесс изменений.

Важно: архитектура приложения должна поддерживать возможность масштабирования. Микросервисы должны взаимодействовать только через API. При увеличении количества, добавляется новый под.

Реплики

ReplicaSets - набор настроек и подов с одной версией конфигурации. При обновлении yaml создается вторая ReplicaSet и один новый под. Из старой ReplicaSet удаляется один под. И так далее до полного обновления. Но конфигурация сохраняется. Можно вернуть к старым настройкам.

Структура YAML файла

Верхний уровень

Параметр	Описание
kind	Тип, в данном случае Deployments
spec	Спецификация

spec

Параметр	Описание
strategy	Стратегия восстановления
replicas	кол-во реплик
selector	правила выбора меток
template	описание шаблона (все аналогично описанию пода)

Примеры

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-deploy
spec:
  replicas: 10
```

```
  selector:
    matchLabels:
      app: hello-world
  revisionHistoryLimit: 5
  progressDeadlineSeconds: 300
  minReadySeconds: 10
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 1
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
      - name: hello-pod
        image: nigelpoulton/k8sbook:1.0
        ports:
        - containerPort: 8080
        resources:
          limits:
            memory: 128Mi
            cpu: 0.1
```

Пример сервиса для данного приложения

```
apiVersion: v1
kind: Service
metadata:
  name: lb-svc
  labels:
    app: hello-world
spec:
  type: LoadBalancer
  ports:
  - port: 8080
    protocol: TCP
  selector:
```

```
    app: hello-world
```

Основные команды

Команда	Доп. пар.	Описание
kubectl get deploy dep_name		статус
kubectl describe deploy dep-name		Расширенная информация
kubectl get rs		Список реплик
kubectl scale	deploy dep_name --replicas count	Императивное масштабирование. Нежелательно.
kubectl rollout status deployment dep_name		Статус обновления подов
kubectl rollout pause deploy dep_name		Приостановка обновления
kubectl describe deploy dep_name		Отображает в частности список роллбеков
kubectl rollout history deployment dep_name		История роллбеков
kubectl rollout undo deployment hello-deploy --to-revision=1		Возврат. Быстро, но не рекомендуется. Лучше через загрузку старого файла из репозитория и обновление.

Revision #3
Created 21 March 2025 14:17:34 by Admin
Updated 22 March 2025 12:14:43 by Admin