

K8S 30: Kubernetes. Resources.

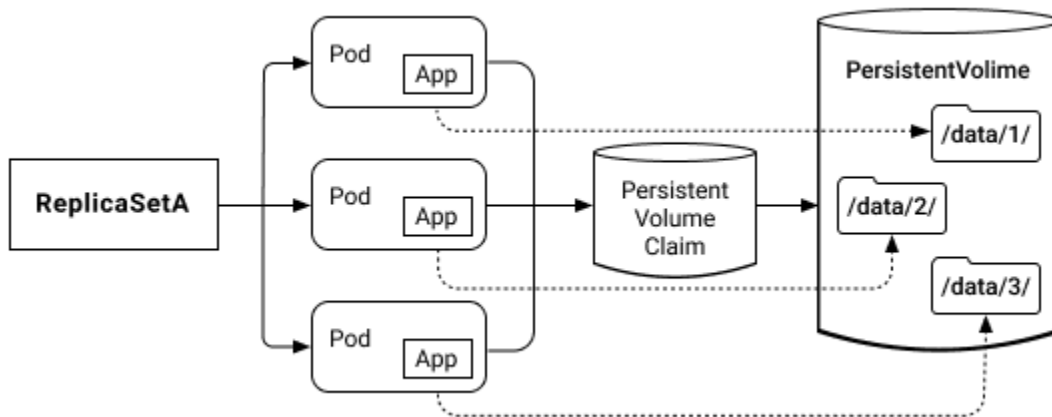
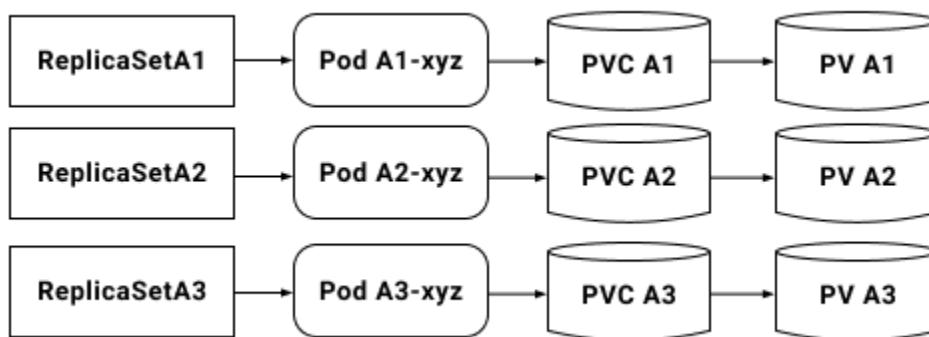
StatefulSet

Описание:

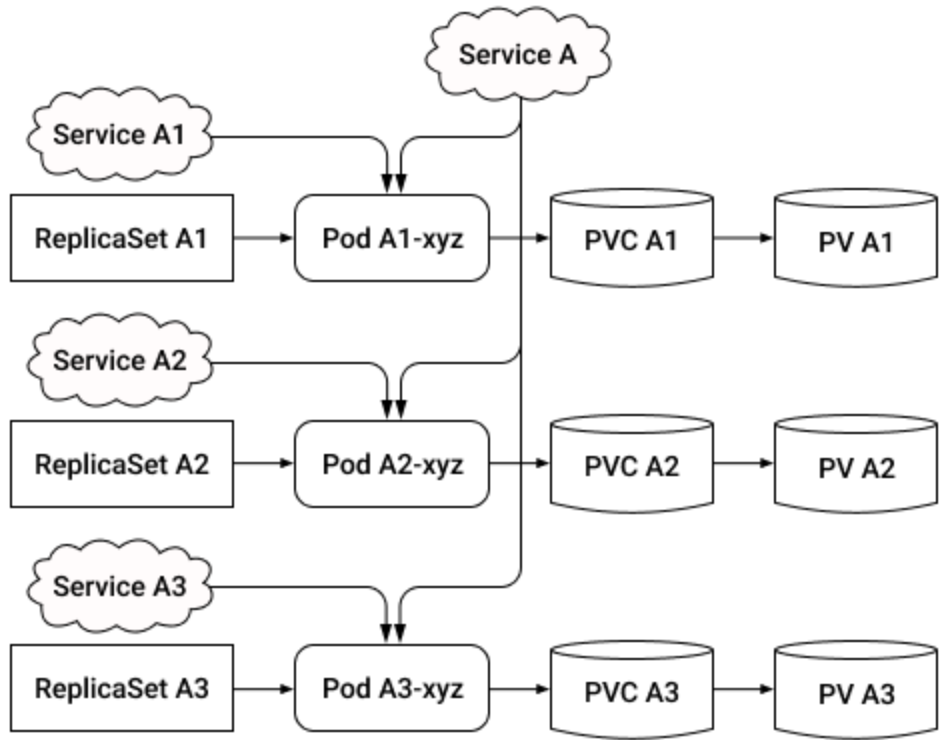
[The basics of stateful applications in Kubernetes](#)

Данный контроллер служит для запуска приложений с состоянием (stateful). Хороший пример — kafka или база данных. Каждому контейнеру нужно знать имена соседних узлов и хранить свои данные. Такого поведения сложно добиться с ReplicaSet. Если в описании ReplicaSet есть PersistentVolumeClaim, то все pod будут запрашивать один PVC. К тому же при перезапуске pod пропадет его IP address и hostname.

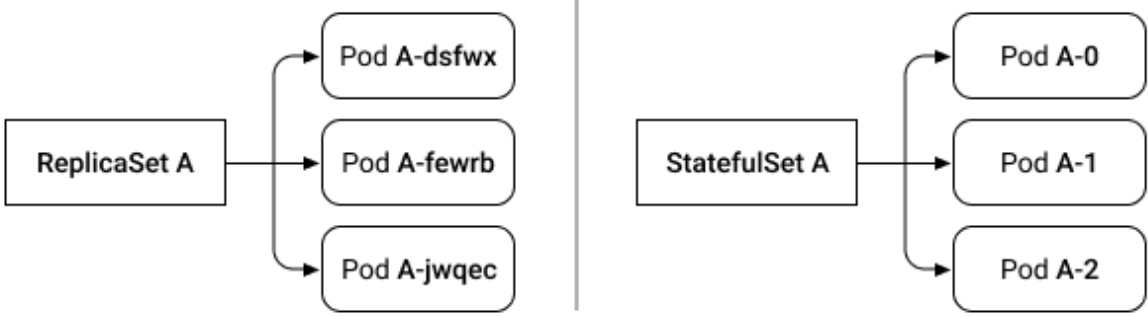
Конечно, можно обойтись ReplicaSet, но посмотрите, сколько объектов придется поддерживать и следить за их состоянием.



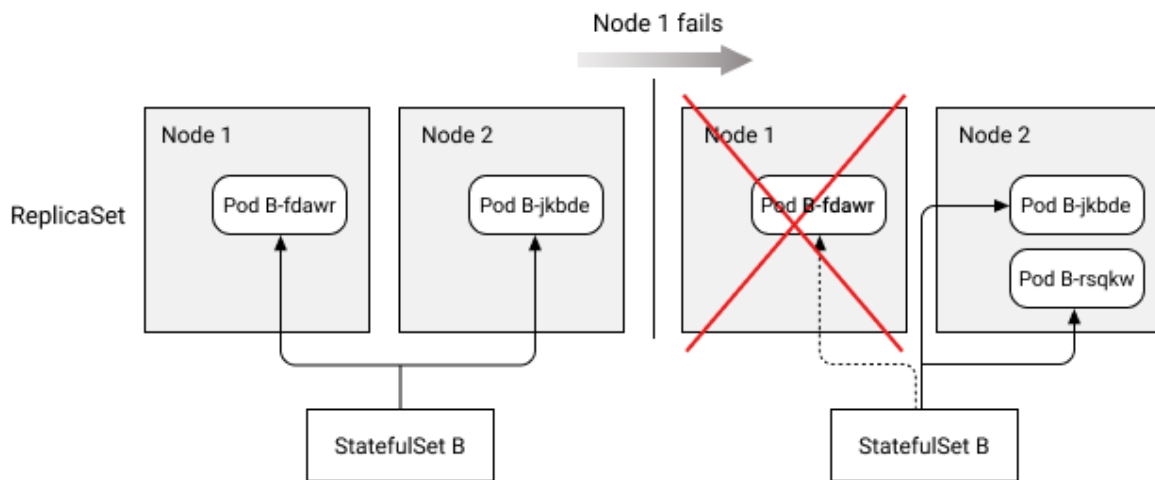
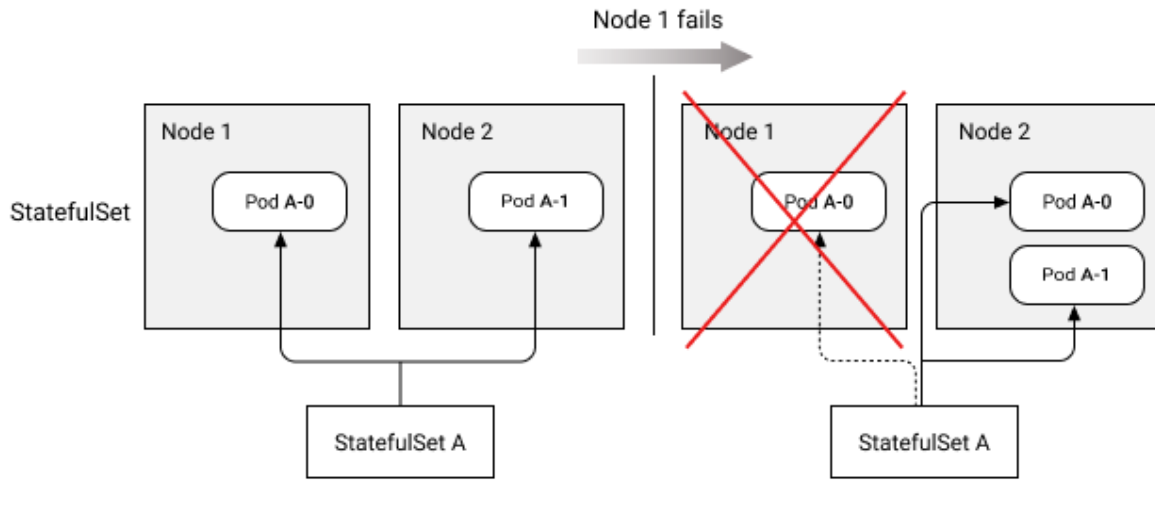
Также придется сделать service для каждого ReplicaSet отдельно и общий для всех.



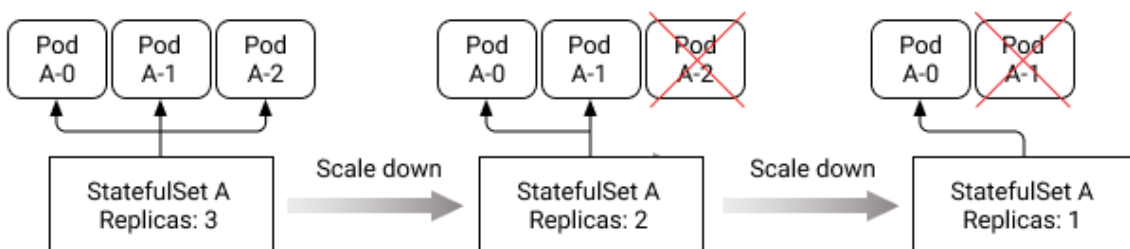
Чтобы не испытывать страдания при запуске подобных приложений, сделали контроллер StatefulSet.



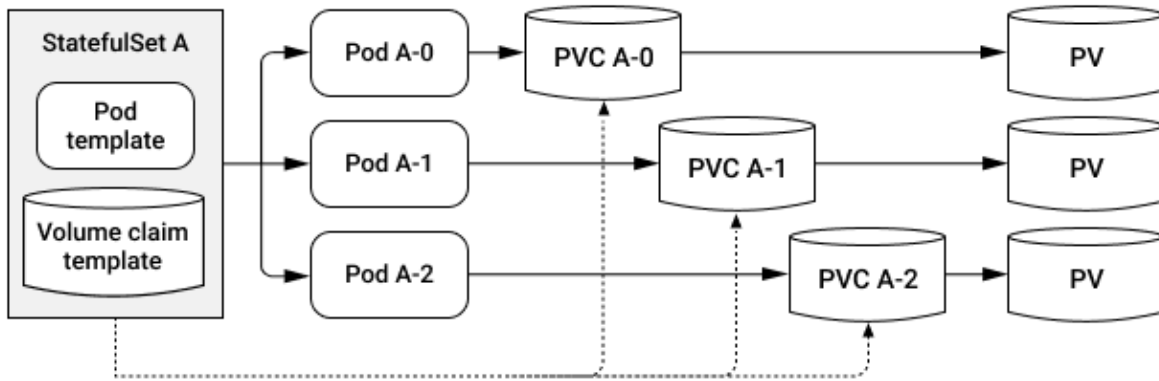
Каждый pod имеет имя, но только поды StatefulSet доступны по DNS и имеют предсказуемое имя.



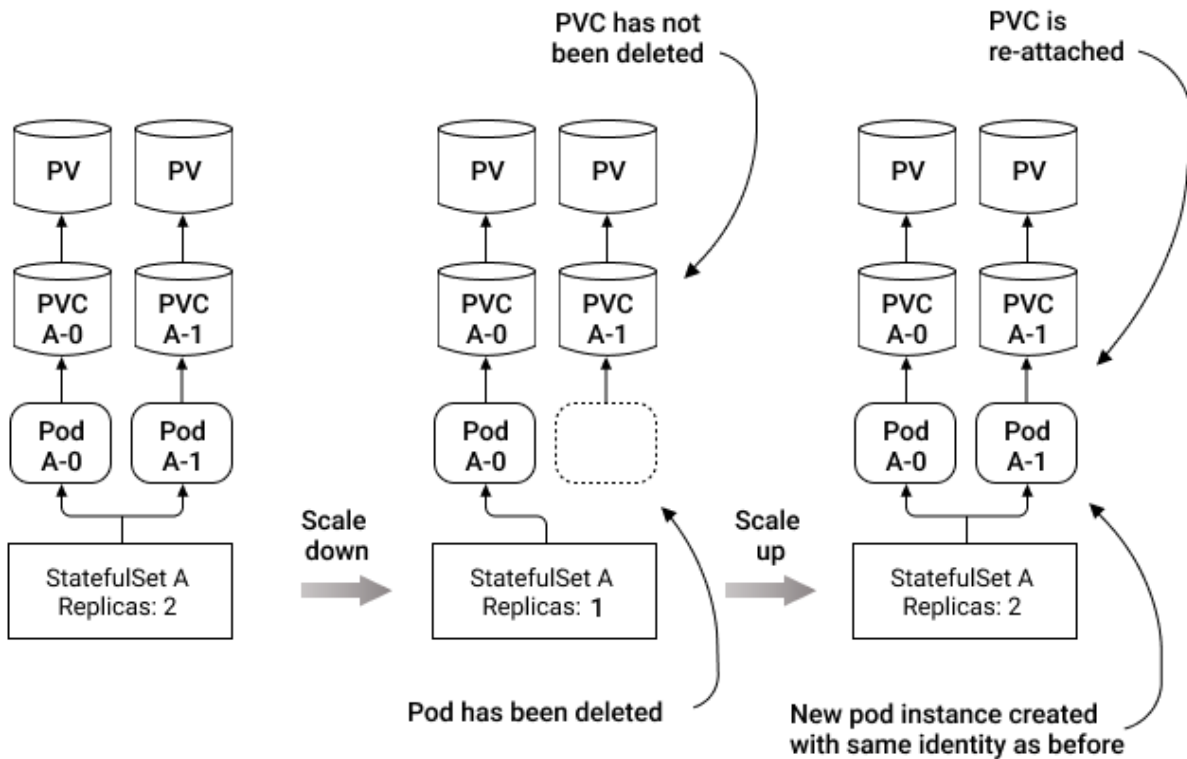
Имя сохраняется при перезапуске. Немного по-иному работает масштабирование StatefulSet. В отличие от ReplicaSet, оно происходит по порядку. ReplicaSet просто запускает дополнительные копии.



Точно также создаются отдельные PVC для каждого pod из replicaset.



Но при масштабировании вниз PVC не удаляется.



```

Пример описания StatefulSet:
apiVersion: apps/v1beta1
kind: StatefulSet
metadata:
  name: mysts
spec:
  serviceName: mysts
  replicas: 2
  template:

```

```

metadata:
  labels:
    app: mysts
spec:
  containers:
  - name: mysts
    image: luksa/mysts-pet
    ports:
    - name: http
      containerPort: 8080
    volumeMounts:
    - name: data
      mountPath: /var/data
  volumeClaimTemplates:
  - metadata:
    name: data
    spec:
      resources:
        requests:
          storage: 1Mi
        accessModes:
        - ReadWriteOnce

```

Заметьте, дополнительно описывается volumeClaimTemplates. Посмотрите описание созданного pod, попробуйте удалить его, изменить количество реплик и понаблюдать за созданием/удалением pod.

Параметр spec.serviceName особенный. При запуске statefulset он обязателен. Создает запись в DNS с SRV-записями по всем pod из данного statefulset. StatefulSet при падении узла кластера ведет себя иначе, чем replicaset.

Предположим, что на трех узлах кластера запущено три pod из statefulset. Что произойдет при падении узла с нулевым pod из statefulset? При просмотре `kubectl get sts` и `kubectl get po` мы увидим `status=unknown` у данного pod. Почему он не перезапустился? Дело в том, что kubelet удаляет поды только по изменению состояния etcd и сообщениям с мастера. Если попробовать удалить pod руками `kubectl delete po sts-pod-0`, то ничего не изменится. Он будет удален только после перезапуска узла кластера. Либо можно принудительно удалить pod: `--force --grace-period=0`. В этом случае pod будет перезапущен на другом узле кластера.

Как правило, в связке с StatefulSet используется Headless Service, что позволяет на уровне приложения заниматься обнаружением экземпляров сервисов при помощи DNS-запросов (мы обсуждали этот тип Service ранее).

Полезные ссылки:

- [StatfulSets \(official docs\)](#)
- [StatefulSet Basics tutorial \(official docs\)](#)

Задание:

1. Используя репозиторий [rabbitmq/rabbitmq-peer-discovery-k8s \(github\)](#), запустите в вашем кластере RabbitMQ кластер из 3 нод.
2. Выведите список подов, созданных StatefulSet (команды и вывод сохранить).
3. Изнутри контейнера получите список нод при помощи команды `rabbitmqctl cluster_status` (команды и вывод сохраните).
4. Увеличьте количество реплик до 5 (команды и вывод сохраните).
5. Выведите список подов, созданных StatefulSet (команды и вывод сохранить).
6. Изнутри контейнера получите список нод при помощи команды `rabbitmqctl cluster_status` (команды и вывод сохраните).
7. Напишите манифест для Headless Service для подов RabbitMQ.
8. Примените написанный манифест (команды и вывод сохранить).
9. Изнутри контейнера получить DNS-запись Headless Service (команды и вывод сохранить).
10. На проверку отправьте манифест Headless Service, сохраненные команды и их выводы.