

Запускаем ReplicaSet

1. Перейдите в директорию: `1.4.replicaset`

```
cd ~/slurm/practice/1.kube-basics-lecture/1.4.replicaset
```

2. Запустите `replicaset` из файла `replicaset.yaml` командой:

```
kubectl apply -f replicaset.yaml
```

3. Проверьте что два `pod`'а запустились.

В дальнейшем уже известные команды дублироваться не будут.

Самостоятельно подберите нужную команду.

4. Увеличьте количество реплик в `replicaset`'е до трёх. Для этого воспользуйтесь командой:

```
kubectl edit replicaset my-replicaset
```

Данная команда откроет манифест `replicaset`'а, хранящийся в Kube API, на редактирование.

В этом файле нужно найти поле `replicas` и изменить 2 на 3. После этого сохранить файл и выйти из редактора.

5. Проверьте что теперь запущено три `pod`'а.

6. Удалите один из `pod`ов командой:

Обратите внимание, что вместо `placeholder`'а `<ИМЯ POD'а>` Вам нужно подставить реальное имя одного из `pod`'ов из команды `kubectl get pod`

```
kubectl delete pod <ИМЯ POD'а>
```

7. Проверьте, что `pod`'ов все еще так же 3.

8. Обновите версию образа в `replicaset`'е:

```
kubectl set image replicaset my-replicaset nginx=nginx:1.13
```

9. Посмотрите какая версия образа используется теперь в replicaset'e:

```
kubectl describe replicaset my-replicaset
```

```
Name:          my-replicaset
Namespace:     default
Selector:      app=my-app
Labels:        app=my-app
Annotations:   <none>
Replicas:      2 current / 2 desired
Pods Status:   2 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=my-app
  Containers:
    nginx:
      Image:          nginx:1.13 <----- Вот она
      Port:           80/TCP
      Host Port:      0/TCP
      Environment:   <none>
      Mounts:         <none>
      Volumes:        <none>
```

Events:

Type	Reason	Age	From	Message
Normal	SuccessfulCreate	13m	replicaset-controller	Created pod: my-replicaset-g4bsk
Normal	SuccessfulCreate	13m	replicaset-controller	Created pod: my-replicaset-npwgn

10. С помощью команды `kubectl describe pod <ИМЯ POD'а>` проверьте версию образа на запущенных pod'ах replicaset'a.

11. Удалите один из pod'ов и проверьте версию образа в pod'e, созданном вместо удаленного. +2

12. Скопируйте и отправьте в качестве ответа на данном шаге результат выполнения команды:

```
kubectl get po <ИМЯ НОВОГО POD'а> -o jsonpath='{.spec.containers[*].image}'
```

Ключ `-o jsonpath` позволяет получить не весь объект целиком, а только содержимое конкретных полей.

Он крайне полезен при написании скриптов для автоматизации задач в

Kubernetes.

Почитать про его возможности подробнее можно [тут](#) и [тут](#).

13. После выполнения задания удалите replicaset:

```
kubectl delete replicaset --all
```

Самостоятельная работа

1. Запустите в кластере replicaset с одной репликой из образа `debian:jessie` с командой `sh -c 'while true; do echo "Hello, ${USERNAME}!"; sleep 2; done'` и именем `hello`

Для указания команды см. пример в разделе Pod

Значение для переменной окружения `${USERNAME}` должно передаваться при запуске replicaset'a из `env`.

Для указания переменных окружения используется поле `env`. Оно указывается внутри описания контейнера. На одном уровне с `image` и `name`.

Пример:

```
env:  
- name: VAR_NAME  
  value: var-value
```

2. Посмотрите логи pod'a replicaset'a.

3. Отправьте одну строчку лога в качестве ответа на этот шаг.

4. После выполнения задания удалите replicaset:

```
kubectl delete replicaset hello
```