

Запускаем Pod

1. Перейдите в домашнюю папку. Скопируйте репозиторий с практикой в `adminbox` и перейдите в директорию: `1.3.pod`

```
cd ~
```

```
git clone git@gitlab.slurm.io:edu/slurm.git
```

```
cd ~/slurm/practice/1.kube-basics-lecture/1.3.pod
```

2. Запустите pod из файла `pod.yaml` с помощью команды:

```
kubectl create -f pod.yaml
```

3. Проверьте, что pod запустился:

```
kubectl get pod
```

NAME	READY	STATUS	RESTARTS	AGE
my-pod	1/1	Running	0	22s

В Kubernetes для большинства объектов есть сокращения. Так например вместо полного `get pod`, можно использовать `get po`

4. Попробуйте запустить еще один instance приложения. Для этого:

1. Откройте файл `pod.yaml`

2. Измените строчку `name: my-pod` на `name: my-pod-1`

3. Запустите новый pod:

```
kubectl create -f pod.yaml
```

5. Отправьте вывод команды `kubectl get pod` в качестве ответа к этому шагу.

6. Удалите запущенные поды после выполнения задания:

```
kubectl delete pod --all
```

Самостоятельная работа

1. Запустите в кластере pod из образа `busybox:latest` с командой `sh -c 'while true; do echo New random number is $(((RANDOM % 100) + 1)); sleep 2; done'` и именем `hello`

Для указания команды в pod'е используется поле `command`. Оно очень похоже на аналогичное в `docker-compose`.

`command` указывается внутри описания контейнера. На одном уровне с `image` и `name`.

Пример:

```
command: ['bash', '-c', 'echo $HOSTNAME']
```

2. Посмотрите логи pod'а командой:

```
kubectl logs hello
```

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

4. После выполнения задания удалите pod:

```
kubectl delete pod hello
```

+2