

Упражнение 1.

Шаг 1. Войдите в кластер OpenShift.

`oc login -u developer`

Шаг 2. Создайте в своем домашнем каталоге рабочий каталог проекта projects и перейдите в него:

`mkdir projects && cd projects`

Создайте проект:

`oc new-project test-project1`

Шаг 3. Создайте поток образов

Создайте YAML файл `stream.yaml` следующего содержания:

`apiVersion: image.openshift.io/v1`

`kind: ImageStream`

`metadata:`

`name: approved-apache`

`spec:`

`lookupPolicy:`

`local: false`

`tags:`

`- name: "latest"`

`from:`

`kind: DockerImage`

`name: bitnami/apache:latest`

`referencePolicy:`

`type: Source`

Выполните команду:

`oc apply -f stream.yaml`

Упражнение 2. Использование потока образов в качестве исходного образа в сборке

Шаг 1. Создайте Deployment и Service для развертывания в своем проекте:

Выполните команду:

```
oc new-app --image-stream=approved-apache
```

Шаг 2. Проверьте результат

Войдите в веб-консоль и перейдите на вкладку Builds -> ImageStreams. Выберите из выпадающего списка свой проект. Щелкните на созданном потоке «approved-apache». Перейдите на вкладку History.

Затем в веб-консоли перейдите на вкладку Workloads -> Deployments. Посмотрите поле Status

Щелкните на созданном развертывании и посмотрите состояние на вкладках ReplicaSets и Pods

В веб-консоли перейдите в Networking -> Services

Шаг 3. Создайте маршрут к приложению

Выполните команду:

```
oc expose svc approved-apache
```

Затем в веб-консоли перейдите на вкладку Networking -> Routes.

Убедитесь, что маршрут появился.

Прим. Если используется CRC настроенный на разрешение имен через файл hosts, проверьте наличие там соответствующей маршруту записи.

```
oc create secret docker-registry my-docker-secret --docker-server=drrabbit/testrepo --docker-username='drrabbit' --docker-password='Xdc0m$eu911'
```

```
oc secrets link builder my-docker-secret
```

```
oc secrets link default my-docker-secret --for=pull
```

```
oc import-image node-web-app --from=drrabbit/testrepo:node-web-app --confirm
```

```
oc import-image node-web-app --from=drrabbit/testrepo:node-web-app --scheduled --confirm
```

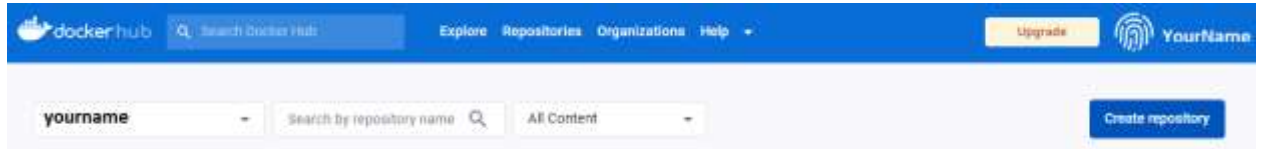
```
Annotations:      openshift.io/image.dockerRepositoryCheck=2023-07-02T11:31:39Z
Image Repository: default-route-openshift-image-registry.apps-crc.testing/test/node-web-app
Image Lookup:     local=false
Unique Images:    1
Tags:             1
latest
  updates automatically from registry drrabbit/testrepo:node-web-app
* drrabbit/testrepo@sha256:b32421cec197f5eb6cee4abb67ff96763768a793a80026c1c07fa9380950552d
  Less than a second ago
```

```
oc describe is/node-web-app
```

Упражнение 2. Развертывание приложения из образа

Шаг 1. Создайте свой тестовый образ на основе существующего:

Создайте свой репозиторий на DockerHub



Загрузите образ

```
docker pull bitnami/apache:latest
```

Подготовьте образ к отправке в собственный репозиторий, используя идентификатор образа. Для этого пометьте образ именем своего репозитория:

```
docker image tag bitnami/apache:latest drrabbit/apache:latest
```

Теперь отправьте образ в свой репозиторий

```
docker push drrabbit/apache:latest
```

Шаг 2. Разверните приложение из образа контейнера

Чтобы развернуть образ контейнера, используйте команду `oc new-app`, указав местоположение образа.

Параметр `--name` предназначен для указания имени развернутого приложения.

Прим. Если имя не указано, то по умолчанию будет использоваться последняя часть имени образа.

`oc new-app drrabbit/apache:latest --name blog`

Проверьте состояние:

`oc status`

Предоставьте доступ к сервису, создав маршрут с помощью команды `oc expose`:

`oc expose svc blog`

Чтобы получить список развернутых подов приложения, используйте команду:

`oc get pods`

Отобразите список ресурсов, созданных для приложения используя команду:

`oc get all -o name --selector app=blog`

Шаг 3. Удалите развертывание

Чтобы убедиться, что вы удаляете ресурсы конкретного приложения, используйте метку, которая была применена командами `oc new-app` и `oc expose` к созданным объектам:

`oc delete all --selector app=blog`

Упражнение 3. развертывание из исходного кода приложения

Шаг 1. Создайте форк репозитория git <https://github.com/MichaelSokoloff/blog-django-py>:

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Owner *

YourName ▾

Repository name *

/ blog-django-py ✓


By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

Sample blog application implemented using Python and Django.

☒ Copy the `master` branch only

Contribute back to openshift-instruqt/blog-django-py by adding your own branch. [Learn more.](#)

 You are creating a fork in your personal account.

Create fork

Шаг 2. Выполните развертывание из исходного кода приложения используя команду:

```
oc new-app --name blog python:latest~ https://github.com/MichaelSokoloff/blog-django-py
```