

bnkapi - Построение multicloud CloudNative приложений на платформе OpenShift.

Развертывание проекта в OpenShift

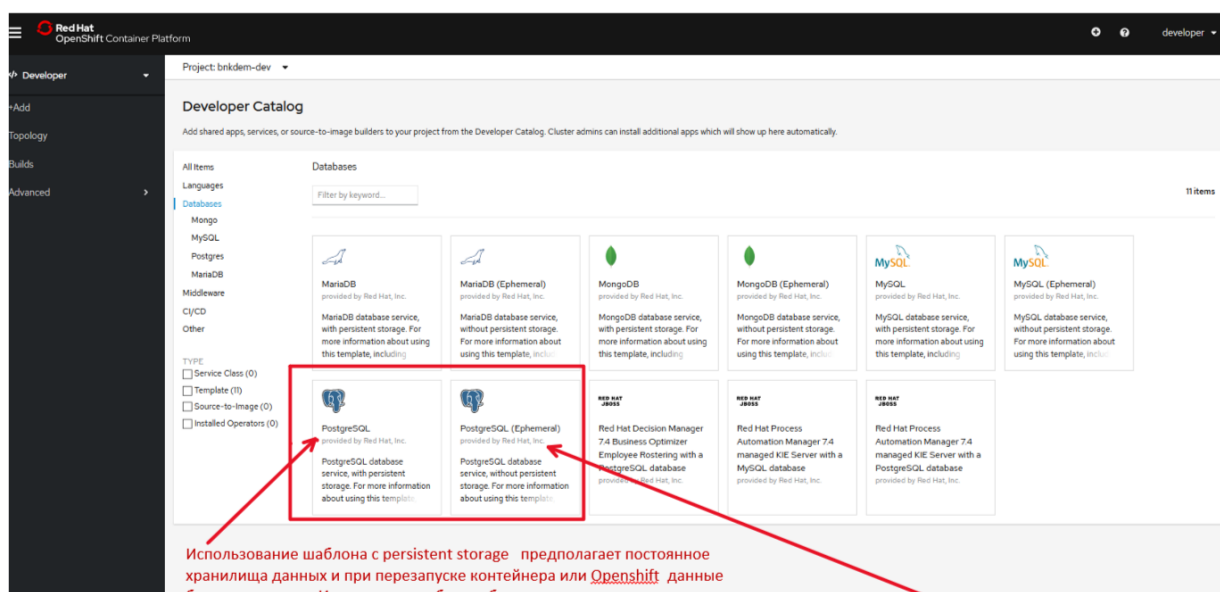
- 1. Цель работы
- 2. Выполнение лабораторной работы
- 2.1. Создать проект в openshift: настройка переменных
- 2.2. Создать проект в openshift: Запуск скрипта на выполнение
- 2.3. Удалить проект в openshift: Запуск скрипта на выполнение
- 2.4. Создать прикладные компоненты проект в openshift: назначение файлов
- 2.5. Создать прикладные компоненты проект в openshift: настройка переменных
- 2.6. Создать прикладные компоненты проект в openshift: Запуск deployment прикладных компонентов
- 2.7. Создать прикладные компоненты проект в openshift: Создание схемы в БД postgres

Цель работы

Целью данной работы является демонстрация возможностей развертывания многокомпонентных приложений на платформе OpenShift.

В лабораторной работе будут выполнены такие этапы:

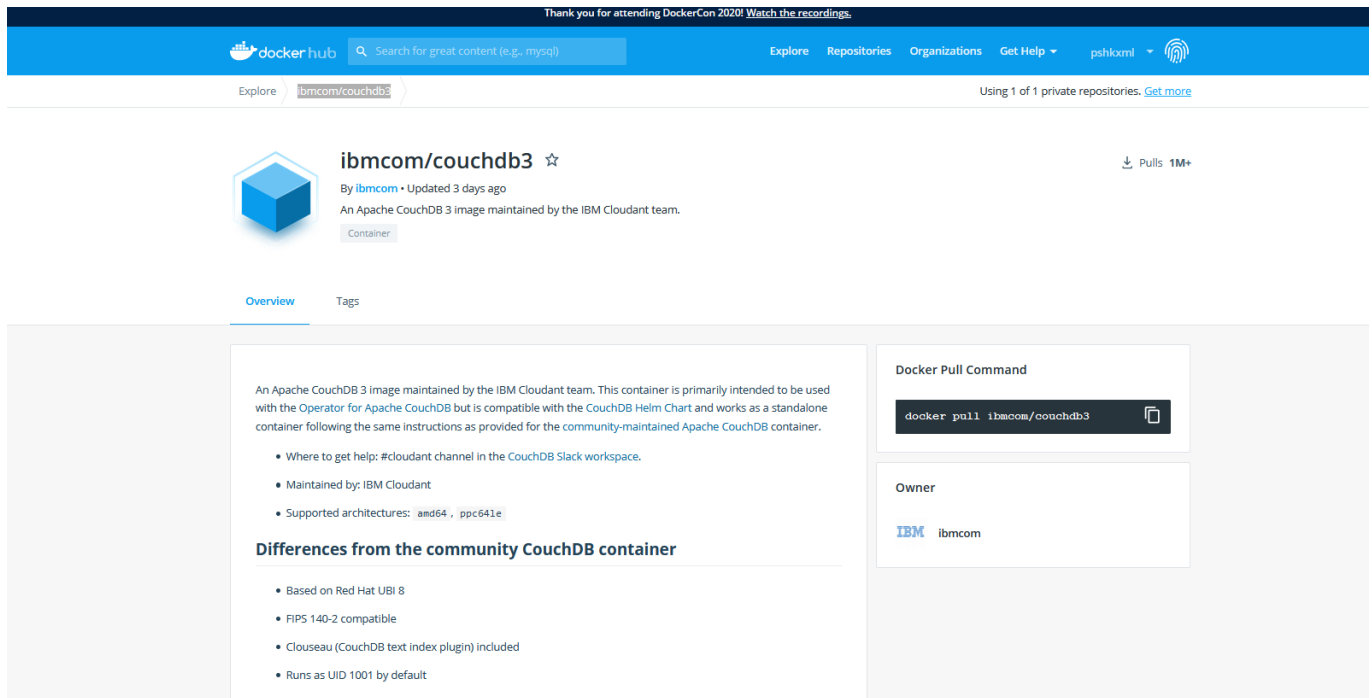
- создание проекта и его подготовка для deployment в него многокомпонентных приложений с помощью `oc cli`.
- deployment реляционной базы данных postgres с предустановленного каталога шаблонов, который присутствует в openshift с помощью `oc cli`, `pic-1`.



Использование шаблона с persistent storage предполагает постоянное хранилища данных и при перезапуске контейнера или Openshift данные будут сохранены. Именно этот шаблон и будем использовать

Использование шаблона "Ephemeral" не предполагает постоянного хранилища данных и при перезапуске контейнера или Openshift данные пропадут. Удобно для разработки

- deployment NoSql базы данных CouchDB (известная в IBM Cloud как CloudantDB) с Docker репозитория и назначения ей Persistence storage. [ibmcom/couchdb3](#) pic-2.



Thank you for attending DockerCon 2020! Watch the recordings.

docker hub Search for great content (e.g., mysql)

Explore Repositories Organizations Get Help pshxaml

Explore **ibmcom/couchdb3** Using 1 of 1 private repositories. [Get more](#)

ibmcom/couchdb3 ☆

By **ibmcom** • Updated 3 days ago

An Apache CouchDB 3 image maintained by the IBM Cloudant team.

Container

Overview Tags

An Apache CouchDB 3 image maintained by the IBM Cloudant team. This container is primarily intended to be used with the Operator for Apache CouchDB but is compatible with the CouchDB Helm Chart and works as a standalone container following the same instructions as provided for the community-maintained Apache CouchDB container.

- Where to get help: #cloudant channel in the CouchDB Slack workspace.
- Maintained by: IBM Cloudant
- Supported architectures: amd64, ppc64le

Differences from the community CouchDB container

- Based on Red Hat UBI 8
- FIPS 140-2 compatible
- Clouseau (CouchDB text index plugin) included
- Runs as UID 1001 by default

Docker Pull Command

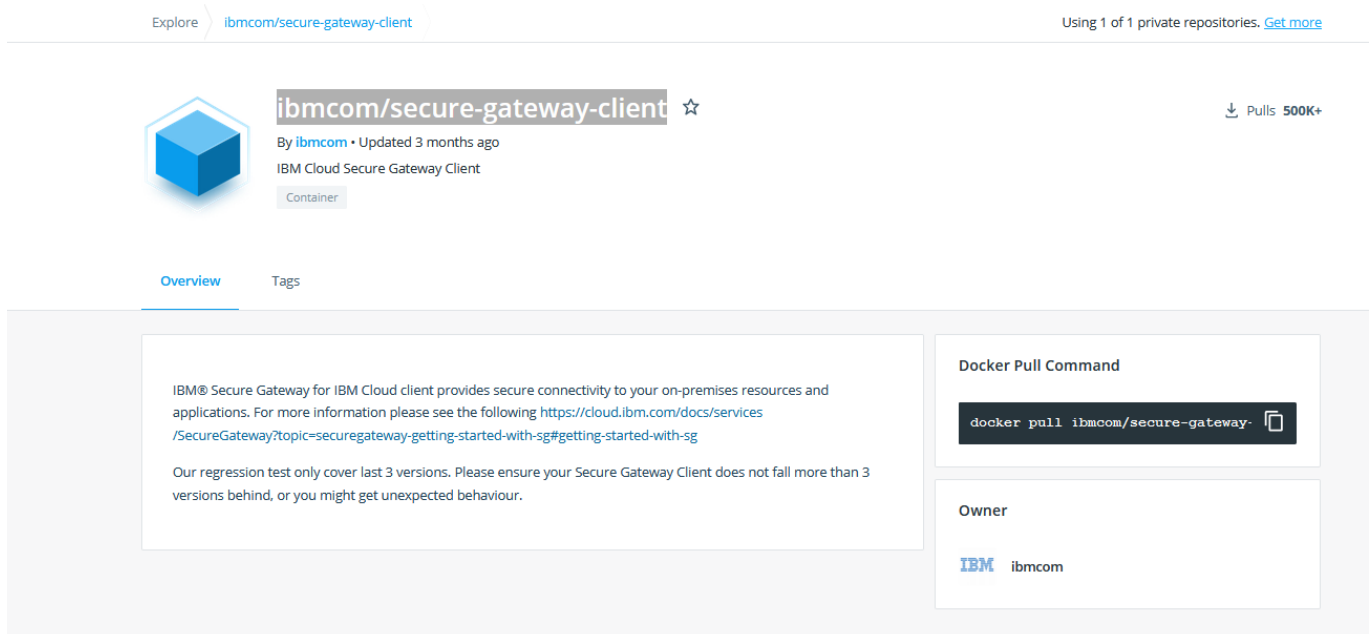
```
docker pull ibmcom/couchdb3
```

Owner

IBM ibmcom

pic-2

- deployment Rest API разработанного на базе Node.js express из исходного кода, размещенного в GitHub.
- deployment клиентской части IBM Secure Gateway с Docker репозитория как контейнер [ibmcom/secure-gateway-client](#).



Explore **ibmcom/secure-gateway-client** Using 1 of 1 private repositories. [Get more](#)

ibmcom/secure-gateway-client ☆

By **ibmcom** • Updated 3 months ago

IBM Cloud Secure Gateway Client

Container

Overview Tags

IBM® Secure Gateway for IBM Cloud client provides secure connectivity to your on-premises resources and applications. For more information please see the following <https://cloud.ibm.com/docs/services/SecureGateway?topic=securegateway-getting-started-with-sg#getting-started-with-sg>

Our regression test only cover last 3 versions. Please ensure your Secure Gateway Client does not fall more than 3 versions behind, or you might get unexpected behaviour.

Docker Pull Command

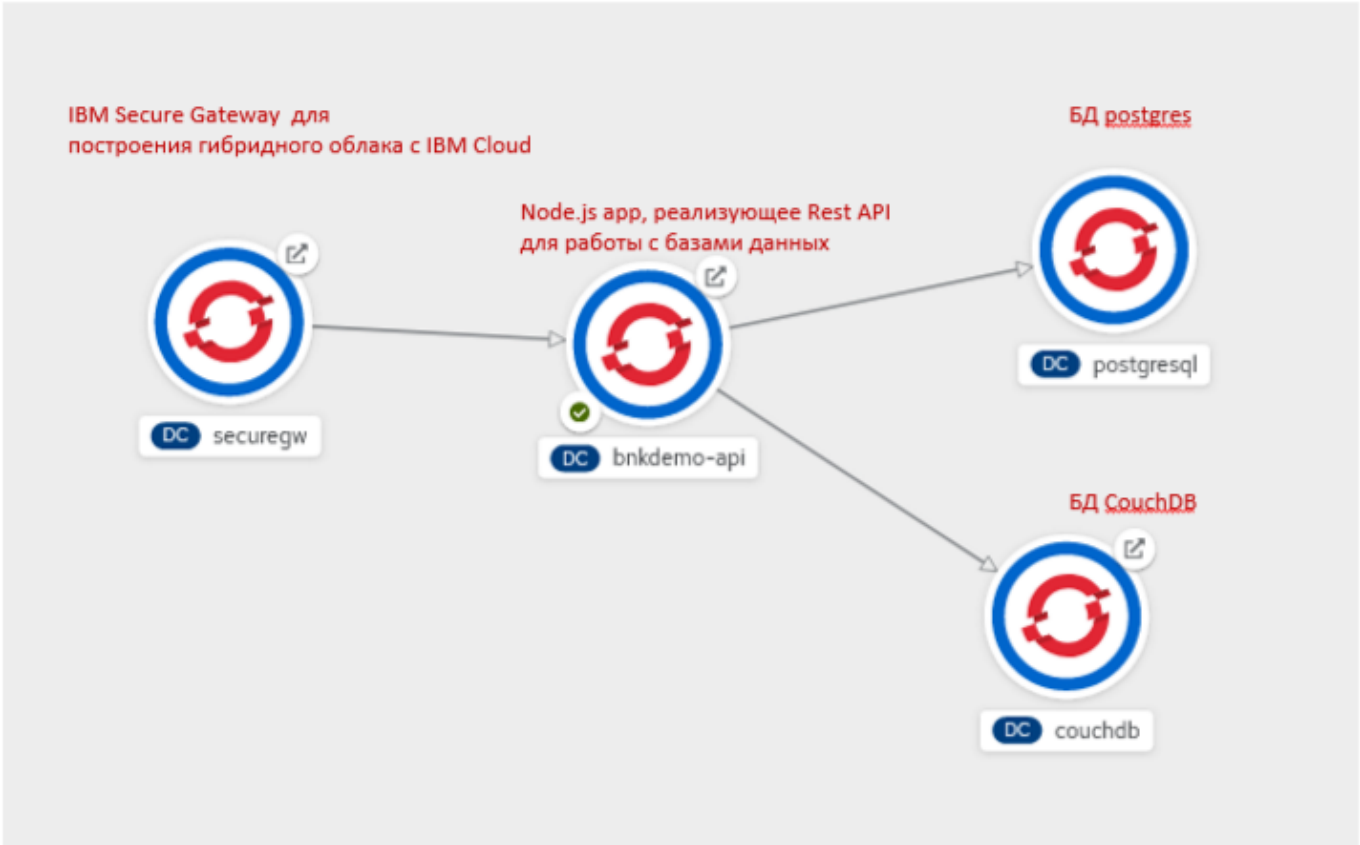
```
docker pull ibmcom/secure-gateway-client
```

Owner

IBM ibmcom

pic-3

В результате выполнения работы будет построена топология компонентов приложения, показанная на pic-4



pic-4

Выполнение лабораторной работы

Создать проект в openshift: настройка переменных

В корне каталога ./openshift находится набор .cmd файлов для создания проекта. Основная цель этого пакета файлов - создать проекты в зависимости от среды и создать дополнительные сущности, которые называются "secrets", для автоматического подключения к gitHub-репозиториям, хранения параметров подключения к базам данных. Ниже - перечень этих файлов с пояснениями

- prj_env.cmd Содержит настройку на проекты и в openshift для разных сред. Предполагается что может быть 3 проекта, в зависимости от сред: Имя проекта для разных сред состоит из непосредственно имени и суффикса, обозначающего идентификатор среды.

Для нашего случая название проекта: "bnkdem", а суффиксы берутся из таблицы. Таким образом, в зависимости от переданного параметра среды, будут создаваться названия проектов как в колонке 3.

идентификатор среды	Наименование среды	Названия проектов
dev	Среда разработки	bnkdem-dev
int	Среда тестирования	bnkdem-int
prod	Среда продуктивная	bnkdem-prod

Фрагмент файла, где формируется имя проекта показан ниже:

```
set BUILD_ENV=%1

echo SET Project parameters

set PRJ-NAME=bnkdem
set PRJ-DISP=bnkdem
set PRJ-DESCR=bnkdem
echo SET ENV marker
set PRJ-NAME=%PRJ-NAME%-%BUILD_ENV%
set PRJ-DISP=%PRJ-DISP%-%BUILD_ENV%
set PRJ-DESCR=%PRJ-DESCR%-%BUILD_ENV%
```

- login.cmd

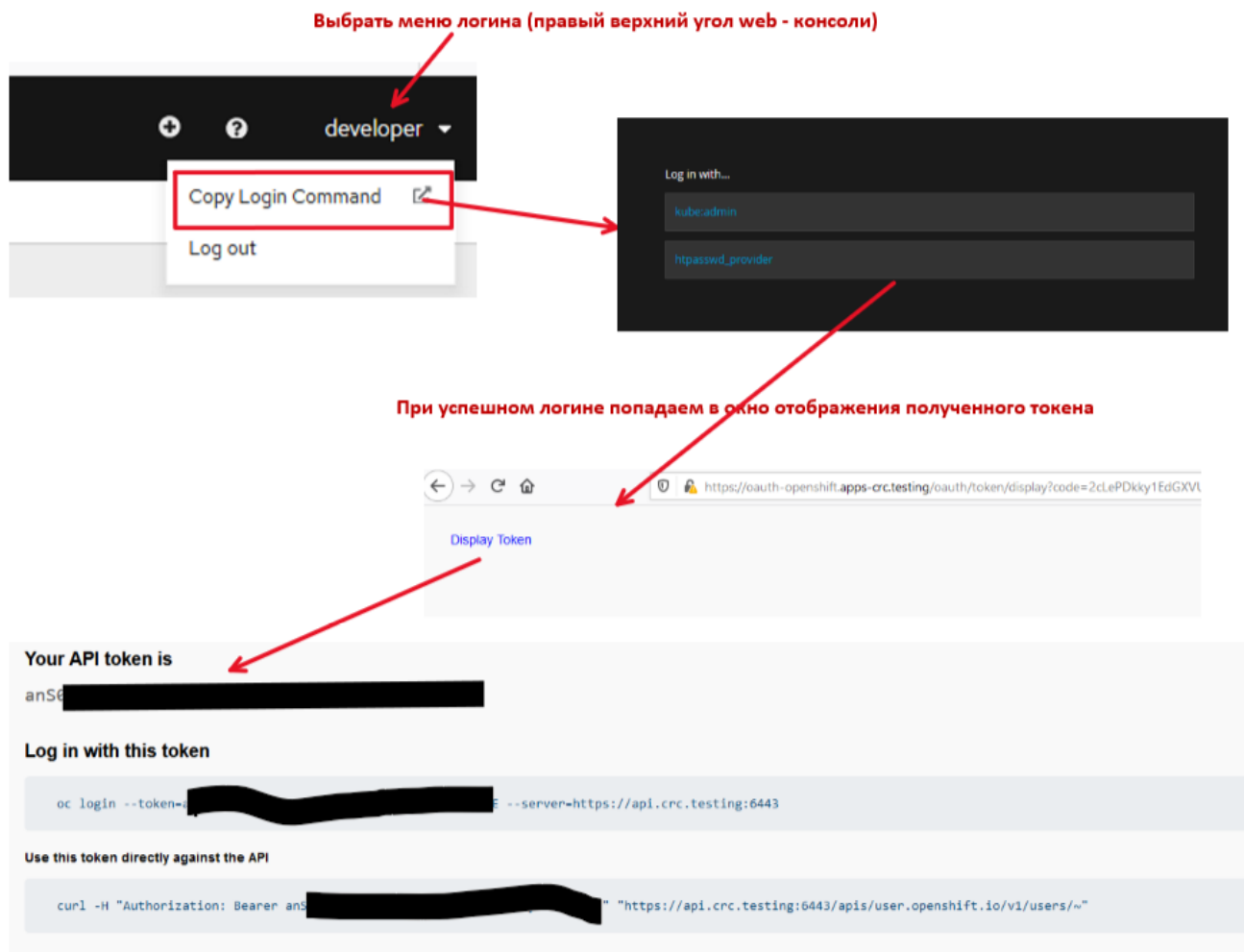
Файл описывающий логин с openshift кластер. в данном случае используется команда:

```
oc login
```

Можно залогиниться по логину и паролю, а лучше по токenu

```
oc login --your token --server=your-claster-api-url
```

На pic-5 показано, как получить логин с токеном



pic-5

- `crt_prj_opshift.cmd` Выполняет пакетное создание проекта и всех секретов. Приложения создаются в отдельном пакете файлов, в зависимости от среды.
- `del_prj_opshift.cmd`
Выполняет пакетное удаление проекта и всех его приложений
- `grn_prj_opshift.cmd` Дать права на проект другим разработчикам
- `rev_prj_opshift.cmd` забрать права на проект
- `crt_prj_secret.cmd` Создать секреты внутри проекта. Ниже в квадратных скобках указано [какие реквизиты для подключения к gitHub] нужно внести:

```
echo *****
echo *   create secret to gitLab
echo *
echo *****

oc create secret generic sinc-gitlab-pvx-1 --from-literal=username=[your
gitlab login] --from-literal=password=[your gitlab password]

oc secrets link deployer sinc-gitlab-pvx-1
```

```
oc secrets link builder sinc-gitlab-pvx-1

oc annotate secret sinc-gitlab-pvx-1 "build.openshift.io/source-secret-
match-uri-1=[your gitlab login uri https://github.com/[you login] ]
```

В данном случае создание секрета к github выполняется полностью набором команд oc. Далее необходимо создать секреты для подключения к базам данных. Поэтому используется другой, универсальный механизм создания объектов в openshift с использованием конфигурационного *.yaml файла.

```
echo *****
echo *      create secret to couchDB
echo *
echo *****
oc create -f couchdb-secret.yaml

echo *****
echo *      create secret to postgresDB
echo *
echo *****
oc create -f postgresdb-secret.yaml
```

- couchdb-secret.yaml
Создание секрета для couchDB. Необходимо внести логин и пароль пользователя для подключения к couchDB

```
stringData:
  database-password: [password]
  database-user: [login]
```

- postgresdb-secret.yaml Создание секрета для postgres DB.

```
stringData:
  database-name: [database name]
  database-password: [password]
  database-user: [login]
```

Создать проект в openshift: Запуск скрипта на выполнение

Для создания проетка на среде int нужно запустить указанный ниже командный файл

```
crt_prj_opshift.cmd int
```

В результате создания сгенерируется такой лог:

```
C:\PSHDEV\PSH-WorkShops\THINK2020\repo\wa\think2020-
bnkapi\openshift>crt_prj_opshift.cmd int
Active code page: 65001
*****
*                CREATE PROJECT ON OPENSIFT
*****
set project and login
=====
SET  Environment variables for project
=====
SET Project parameters
SET ENV marker
PRJ-NAME=bnkdem-int
PRJ-DISP=bnkdem-int
PRJ-DESCR=bnkdem-int
*****
*                Login script on OPENSIFT
*****
* oc login --server --token=
* oc login --server -u -p
*****
*****
* Openshift CLI URL=
*****
Logged into "https://api.crc.testing:6443" as "developer" using the token
provided.

You have access to the following projects and can switch between them with 'oc
project <projectname>':

* bnk-dev
  bnkdem-dev
  bnkdem-prod

Using project "bnk-dev".
create project
Now using project "bnkdem-int" on server "https://api.crc.testing:6443".

You can add applications to this project with the 'new-app' command. For example,
try:

    oc new-app ruby~https://github.com/sclorg/ruby-ex.git

to build a new example application in Python. Or use kubectl to deploy a simple
Kubernetes application:
```

```
kubectl create deployment hello-node --image=gcr.io/hello-minikube-zero-
install/hello-node
```

NAME	DISPLAY NAME	STATUS
bnk-dev		Active
bnkdem-dev	bnkdem-dev	Active
bnkdem-prod	bnkdem-prod	Active

Already on project "bnkdem-int" on server "https://api.crc.testing:6443".

```
*****
*   create secrets
*****
*****
*   create secret to  git
*
*****
secret/sinc-gitlab-pvx-1 created
secret/sinc-gitlab-pvx-1 annotated
secret/sinc-github created
secret/sinc-github annotated
*****
*   create secret to couchDB
*
*****
secret/couchdb created
*****
*   create secret to postgresDB
*
*****
secret/postgresql created
*****
*   grant access to developers
*****
*****
*   Проставить другому разработчику права администратора проекта
*
*****
C:\PSHDEV\PSH-WorkShops\THINK2020\repo\wa\think2020-bnkapi\openshift>
```

Удалить проект в openshift: Запуск скрипта на выполнение

Для удаления проетки на среде int нужно запустить указанный ниже командный файл

```
crt_prj_opshift.cmd int
```

Ниже показан лог раобты комнады удаления


```

C:\PSHDEV\PSH-WorkShops\THINK2020\repo\wa\think2020-
bnkapi\openshift>del_prj_opshift.cmd int
Active code page: 65001
*****
*          DELETE PROJECT ON OPENSIFT
*****
set project and login
=====
SET  Evironment variables for project
=====
SET Project parameters
SET ENV marker
PRJ-NAME=bnkdem-int
PRJ-DISP=bnkdem-int
PRJ-DESCR=bnkdem-int
*****
*          Login script on OPENSIFT
*****
* oc login --server --token=
* oc login --server -u -p
*****
*****
* Openshift CLI URL=
*****
Logged into "https://api.crc.testing:6443" as "developer" using the token
provided.

You have access to the following projects and can switch between them with 'oc
project <projectname>':

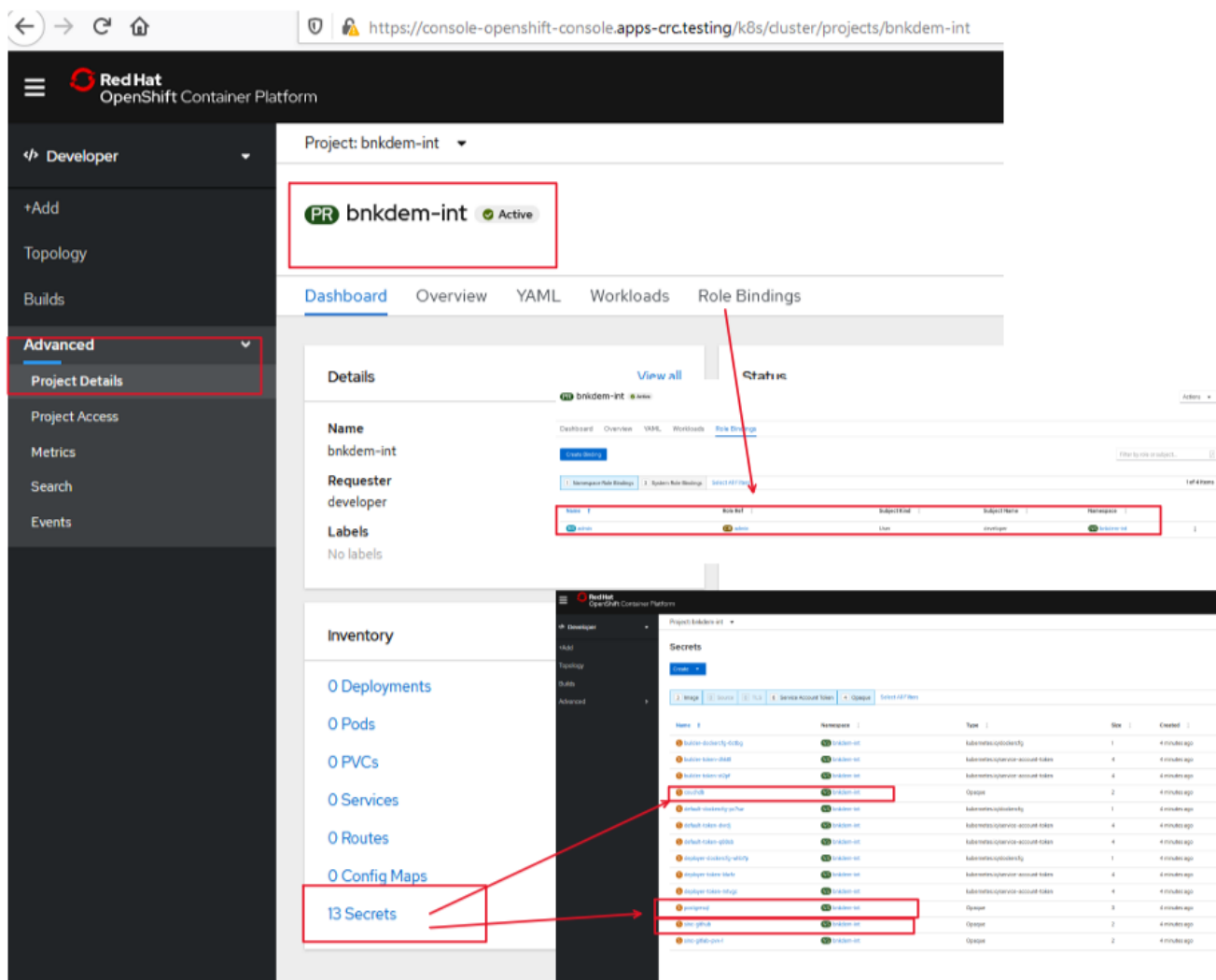
    bnk-dev
    bnkdem-dev
*   bnkdem-int
    bnkdem-prod

Using project "bnkdem-int".
project.project.openshift.io "bnkdem-int" deleted
NAME          DISPLAY NAME    STATUS
bnk-dev              Active
bnkdem-dev    bnkdem-dev     Active
bnkdem-int     bnkdem-int     Terminating
bnkdem-prod    bnkdem-prod    Active

C:\PSHDEV\PSH-WorkShops\THINK2020\repo\wa\think2020-bnkapi\openshift>

```

Зайдя в web-console openshift можем проверить результаты создания проекта (pic-6)



pic-6

Создать прикладные компоненты проект в openshift: назначение файлов

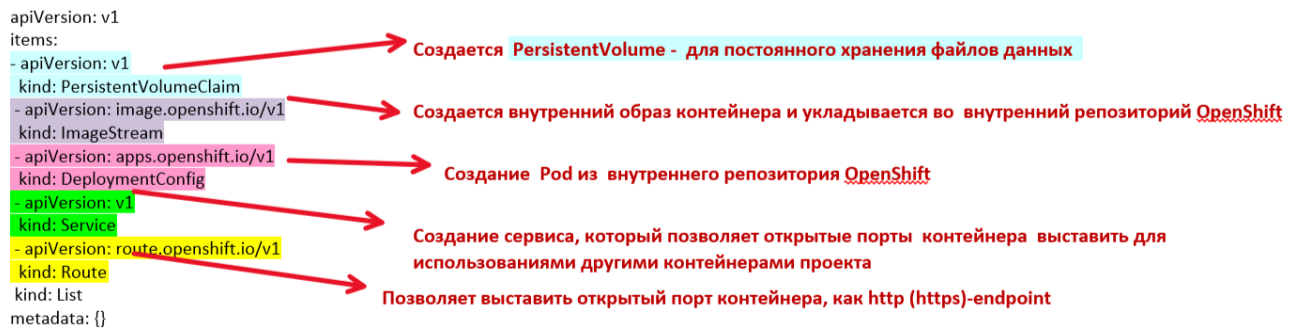
В зависимости от среды в подкаталогах:

```
./dev
./int
.prod
```

находится набор пакетных файлов и настроек переменных среды для создания прикладных компонентов проекта, индивидуальных для каждой среды. В каждом каталоге находится одинаковый набор файлов. Разница может быть в настройках переменных среды.

- `crt_app.cmd` Общий скрипт для последовательного пакетного создания всех прикладных компонентов
- `1-crt_app.cmd` Создать один компонент, который передается в виде параметра.
- `crt_app_couchdb.cmd`, `crt_app_couchdb.yaml` Командный файл создания couchDB из docker-контейнерного. Файл `crt_app_couchdb.yaml` - содержит описание компонентов, которые будут

созданы в процессе выполнения. На pic-7 показаны блоки yaml-файла с пояснениями выполняемых функций



pic-7

А на pic-8 показаны фрагменты, показывающие подключение стораджа и переменных среды.

```

- apiVersion: apps.openshift.io/v1
kind: DeploymentConfig
metadata:
  annotations:
    openshift.io/generated-by: OpenShiftNewApp
  creationTimestamp: null
  labels:
    app: couchdb
    name: couchdb
spec:
  replicas: 1
  selector:
    app: couchdb
    deploymentconfig: couchdb
  strategy:
    resources: {}
  template:
    metadata:
      annotations:
        openshift.io/generated-by: OpenShiftNewApp
      creationTimestamp: null
      labels:
        app: couchdb
        deploymentconfig: couchdb
    spec:
      containers:
        - env:
            - name: COUCHDB_PASSWORD
              valueFrom:
                secretKeyRef:
                  key: database-password
                  name: couchdb
            - name: COUCHDB_USER
              valueFrom:
                secretKeyRef:
                  key: database-user
                  name: couchdb
          - name: method
            value: new-app
          image: ibmcom/couchdb3
          name: couchdb
          ports:
            - containerPort: 5984
              protocol: TCP
            - containerPort: 9100
              protocol: TCP
            - containerPort: 4369
              protocol: TCP
          resources: {}
          volumeMounts:
            - mountPath: /opt/couchdb/data
              name: couchdb1v
          volumes:
            - name: couchdb1v
              persistentVolumeClaim:
                claimName: couchdb1v
        test: false
        triggers:
          - type: ConfigChange

```

В этой части yaml файла показано, как из секрета **couchdb** вычитываются логин и пароль в переменные среды: **COUCHDB_PASSWORD, COUCHDB_USER**

Подключение хранилища

pic-8

- `crt_app_pgdb.cmd` `crt_app_pgdb.yaml` Командный файл для создания postgres DB. Файл `crt_app_pgdb.yaml` содержит описание компонентов, которые будут созданы в процессе выполнения.
- `crt_app_bnkdemo-api.cmd`, `bnkdemo-api.env` Командный файл, для создания Node.JS приложения реализующее REST API для работы с базами данных. Файл `bnkdemo-api.env` содержит перечень переменных среды, которые будут установлены в процессе развертывания.

Тут приложение создается командой **oc new-app**

```

oc new-app https://github.com/pavlo-shcherbukha/bnkapidemo.git#master --context-dir=/src/bnkdemo-be --name="bnkdemo-api" --env-file ./bnkdemo-api.env --strategy=source --source-secret=sinc-github --image-stream=openshift/nodejs:10-SCL

```

```
-l app=bnkdemo-api
```

При этом параметризуется: URL github репозитория, с которого будут доставаться исходники Node.js, через знак "#" указывается имя git branch

Относительная папка, с которой взять исходники, указывается в ключе --context-dir

```
--context-dir=/src/
```

Ключ **--env-file** ссылается на файл с переменными среды, которые будут созданы в процессе deployment

```
--env-file ./bnkdemo-api.env
```

Ключ **--strategy=source** уточняет, что развертывание приложения будет выполнено из репозитория исходного кода, а не из docker образа

Ключ **--source-secret=sinc-github** указывает на секрет с параметрами подключения к github для клонирования репозитория.

Ключ **--image-stream=openshift/nodejs:10-SCL** указывает на базовый образ, в который будут копироваться исходники и потом собираться итоговый образ приложения.

По линку [Container images offer lightweight and self-contained software to enable deployment at scale.](#) можно найти документацию и описание использования контейнеров. В данном случае, так как используется однопользовательская версия используется образ: <https://github.com/nodeshift/centos7-s2i-nodejs>

После создания самого компонента следует следующая команда, по созданию http - роутера, чтобы компонент стал доступным снаружи проекта.

```
echo *****
echo *      create bnkdemo-api Router
echo *
echo *****
oc expose svc/bnkdemo-api --hostname="bnkapi-bnkdem-dev.apps-crc.testing" --
name="bnkapi-bnkdem-dev.apps-crc.testing" --port 8080 -l app=bnkdemo-api
```

В данном случае, хочется обратить внимание, что --hostname состоит из вполне определенных реквизитов: "bnkapi-bnkdem-dev.apps-crc.testing"

- .apps-crc.testing DNS кластера
- bnkdem-dev наименование проекта
- bnkapi наименование компонента в проекте.

следуя этим правилам, можно обеспечить не повторяемость роутеров.

- `crt_app_securegw.cmd securegw.env`

Создание клиентской части (не облачной) IBM Secure GateWay.

`a name="p2-5">`

Создать прикладные компоненты проект в openshift: настройка переменных

Необходимо указать URL DNS openshift кластера в файлах:

- `crt_app_bnkdemo-api.cmd`
- `crt_app_securegw.cmd`
- `crt_app_couchdb.yaml` в разделе Route

```
kind: Route
metadata:
  creationTimestamp: null
  labels:
    app: couchdb
  name: db-bnkdem-dev.apps-crc.testing
spec:
  host: db-db-bnkdem-dev.apps-crc.testing
```

Указать параметры подключения к базам данных в файле: `bnkdemo-api.env`.

Создать прикладные компоненты проект в openshift: Запуск `deployemnt` прикладных компонентов

Для запуска `deployment` нужно запустить набор пакетных файлов с указанием среды `deployment`

```
cd int
crt_app.cmd int
```

Лог создания приложений показан ниже

```
C:\PSHDEV\PSH-WorkShops\THINK2020\repo\wa\think2020-bnkapi\openshift>cd int

C:\PSHDEV\PSH-WorkShops\THINK2020\repo\wa\think2020-
bnkapi\openshift\int>crt_app.cmd int
*****
*
*          CREATE APP IN OPENSIFT
```

```

*****
set project and login
=====
SET  Evironment variables for project
=====
SET Project parameters
SET ENV marker
PRJ-NAME=bnkdem-int
PRJ-DISP=bnkdem-int
PRJ-DESCR=bnkdem-int
*****

*          Login script on OPENSIFT
*****

* oc login --server --token=
* oc login --server -u -p
*****
*****

* Openshift CLI URL=
*****

Logged into "https://api.crc.testing:6443" as "developer" using the token
provided.

You have access to the following projects and can switch between them with 'oc
project <projectname>':

    bnk-dev
    bnkdem-dev
*   bnkdem-int
    bnkdem-prod

Using project "bnkdem-int".
Already on project "bnkdem-int" on server "https://api.crc.testing:6443".
*****

*   create CouchDB
*
*   ./couchdb.env - contains database admin and password
*   docker pull ibmcom/couchdb3
*   https://hub.docker.com/r/ibmcom/couchdb3
*   имитация --output=yaml --dry-run
*****

No resources found
persistentvolumeclaim/couchdb1v created
imagestream.image.openshift.io/couchdb created
deploymentconfig.apps.openshift.io/couchdb created
service/couchdb created
route.route.openshift.io/db-bnkdem-dev.apps-crc.testing created
*****

*   create Postgres DB from template
*
*****

No resources found
service/postgresql created
persistentvolumeclaim/postgresql created
deploymentconfig.apps.openshift.io/postgresql created

```

```

*****
*   Router не нужен БД работает по своему протоколу
*   Нужно сделать port forward для работы с psql
*   oc port-forward postgresql-1-97dtc 15432:5432
*
*****
*****
*   create bnkdemo-api
*
*****

No resources found
warning: Cannot check if git requires authentication.
--> Found image 4b4fcd7 (5 months old) in image stream "openshift/nodejs" under
tag "10-SCL" for "openshift/nodejs:10-SCL"

Node.js 10
-----
Node.js 10 available as container is a base platform for building and running
various Node.js 10 applications and frameworks. Node.js is a platform built on
Chrome's JavaScript runtime for easily building fast, scalable network
applications. Node.js uses an event-driven, non-blocking I/O model that makes it
lightweight and efficient, perfect for data-intensive real-time applications that
run across distributed devices.

Tags: builder, nodejs, nodejs10

* The source repository appears to match: nodejs
* A source build using source code from https://github.com/pavlo-
shcherbukha/bnkapidemo.git#master will be created
* The resulting image will be pushed to image stream tag "bnkdemo-
api:latest"
* Use 'oc start-build' to trigger a new build
* WARNING: this source repository may require credentials.
    Create a secret with your git credentials and use 'oc set build-
secret' to assign it to the build config.
* This image will be deployed in deployment config "bnkdemo-api"
* Port 8080/tcp will be load balanced by service "bnkdemo-api"
* Other containers can access this service through the hostname "bnkdemo-
api"

--> Creating resources with label app=bnkdemo-api ...
imagestream.image.openshift.io "bnkdemo-api" created
buildconfig.build.openshift.io "bnkdemo-api" created
deploymentconfig.apps.openshift.io "bnkdemo-api" created
service "bnkdemo-api" created
--> Success
Build scheduled, use 'oc logs -f bc/bnkdemo-api' to track its progress.
Application is not exposed. You can expose services to the outside world by
executing one or more of the commands below:
'oc expose svc/bnkdemo-api'
Run 'oc status' to view your app.
*****
*   create bnkdemo-api Router
*

```



```
*****  
route.route.openshift.io/bnkapi-bnkdem-dev.apps-crc.testing exposed  
Press any key to continue . . .  
  
C:\PSHDEV\PSH-WorkShops\THINK2020\repo\wa\think2020-bnkapi\openshift\int>
```

Создать прикладные компоненты проект в openshift: Создание схемы в БД postgres

Инструкция по запуску на выполнение DDL-скриптов находится в файле <src/ddl-bnk/readme.md>