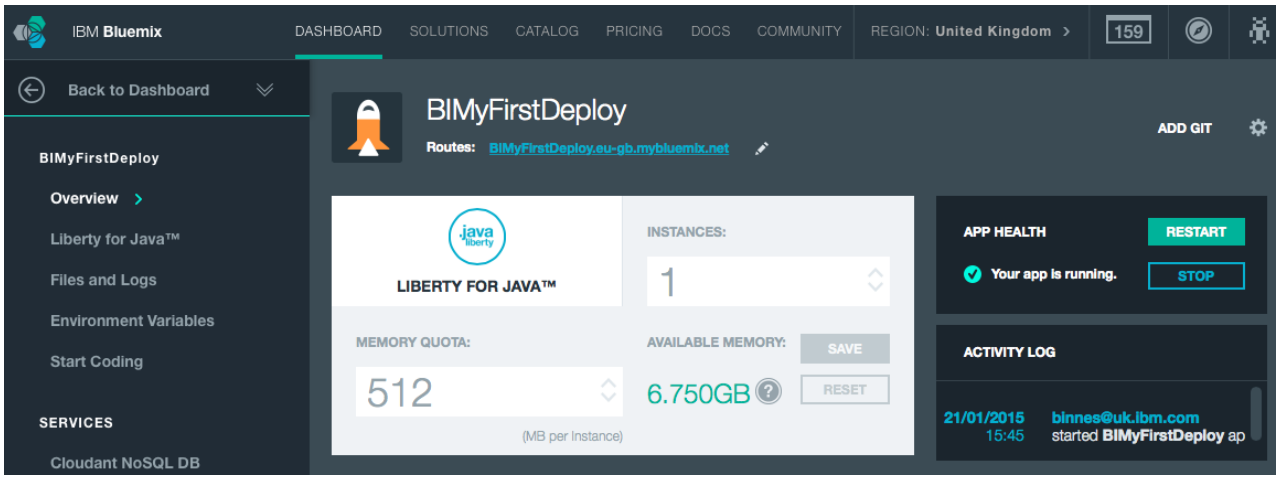
Rapidly Building Apps on the Cloud

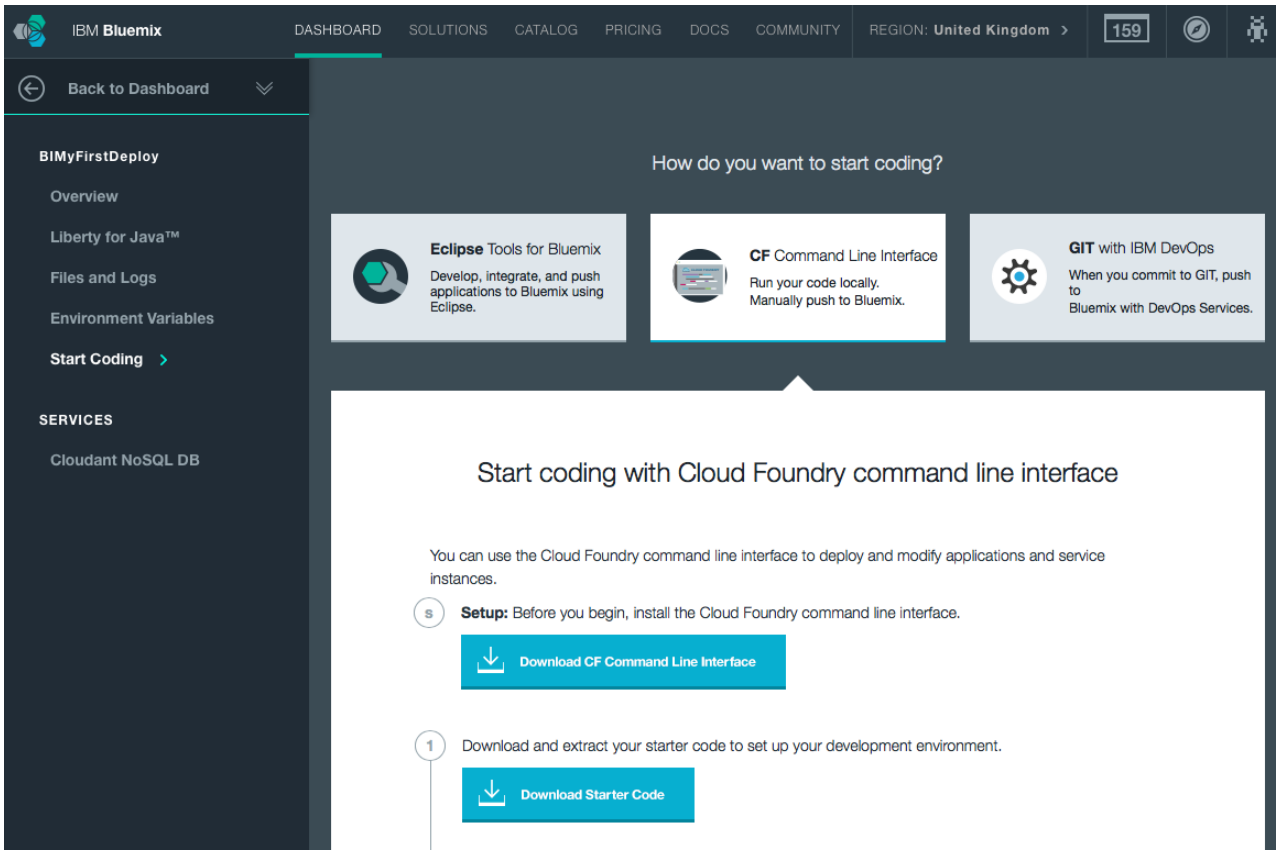
udemy.com/building-highly-scalable-apps-on-the-cloud/learn/v4/t/lecture/2953910

In this exercise you will use the Command Line Interface tool to work with Bluemix. You use this tool in a terminal or command window on your workstation.

We will work with the same sample application we used in exercise 3.a. From the dashboard view in Bluemix select the application to enter the application Overview



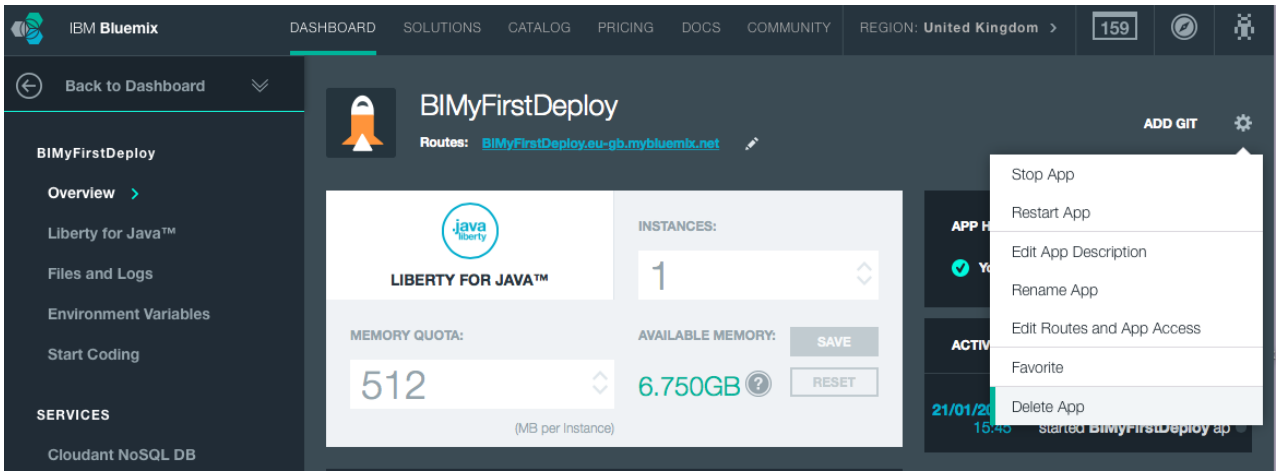
Select 'Start Coding' then 'Download Starter Code'



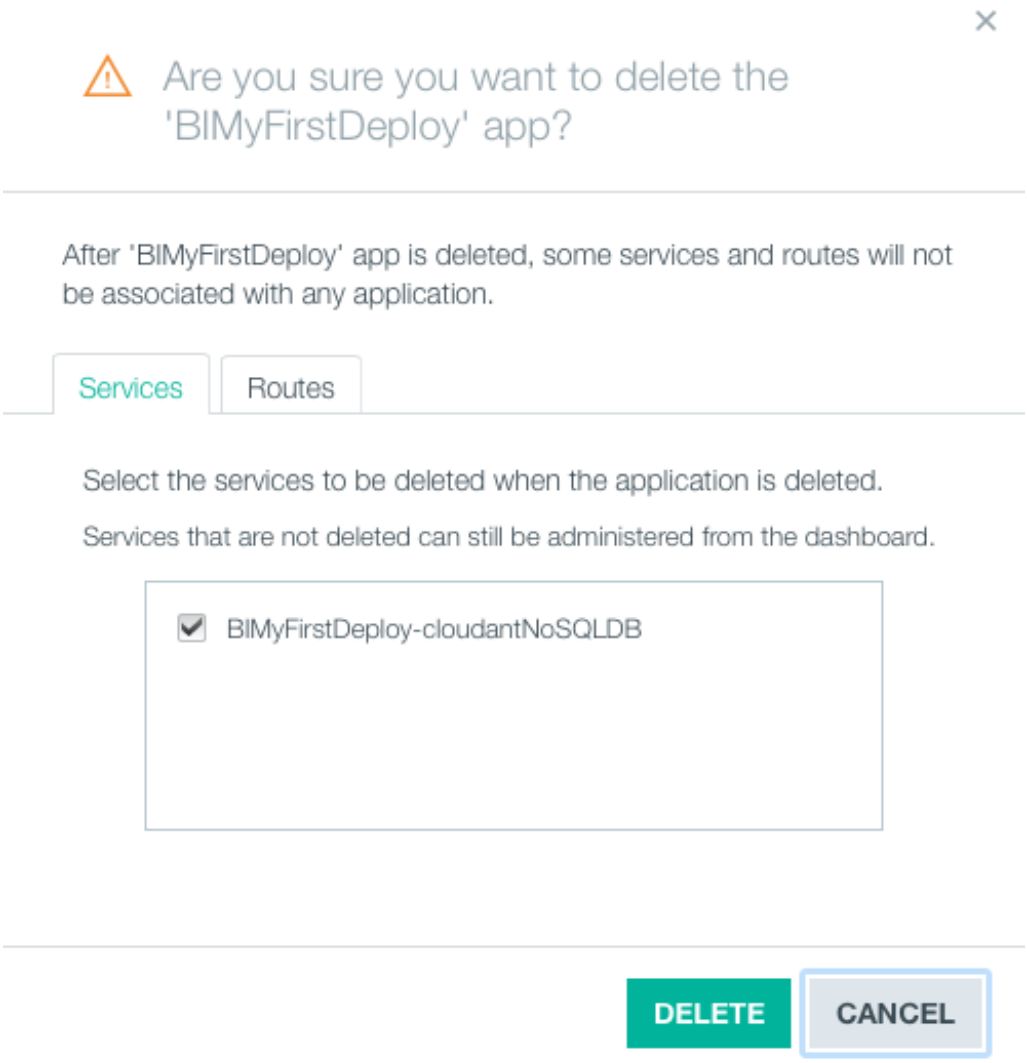
Once the starter package has been downloaded move it to a directory on your workstation where you want to work, such as Bluemix directory in your Documents folder. Then unzip it (double clicking or right-click and select to

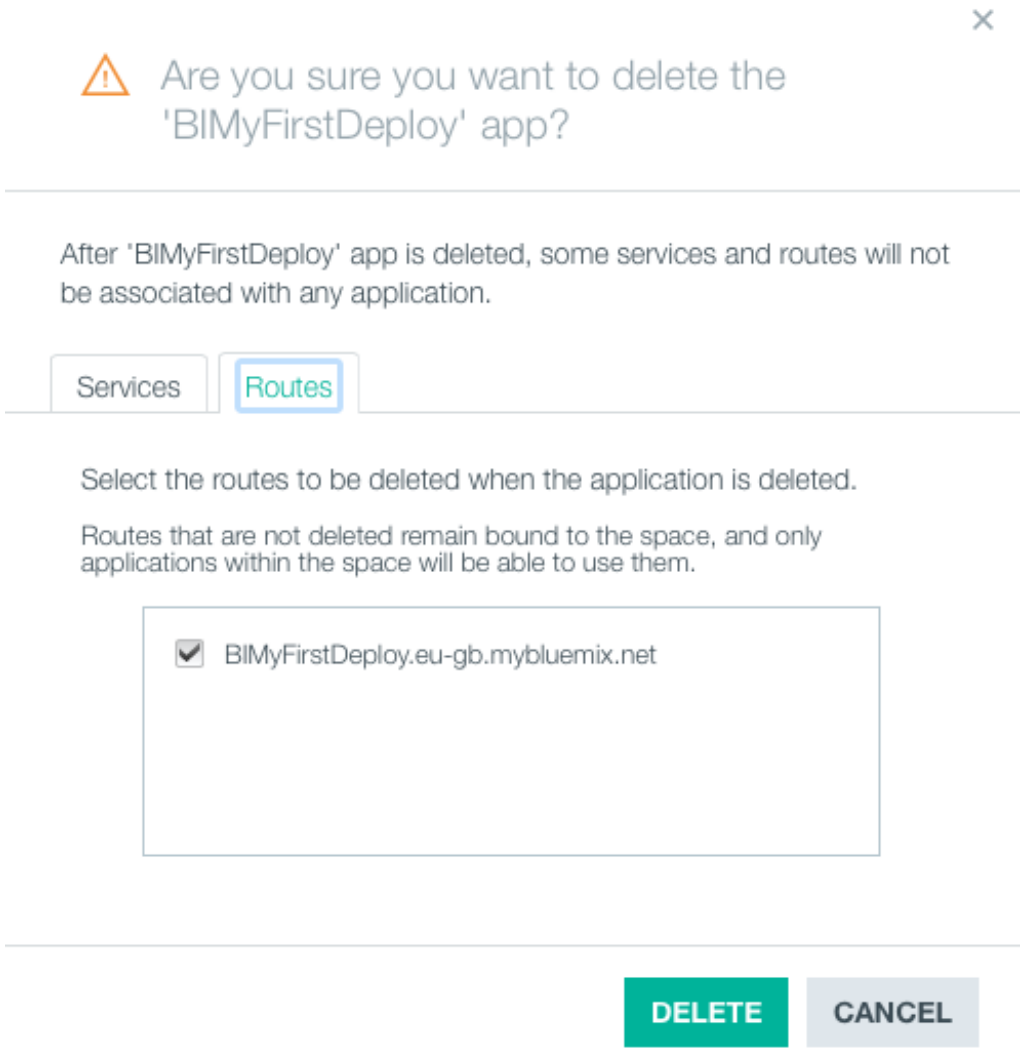
unarchive) . Don't delete the zip file – we will need it in Exercise 3.c

You can delete the deployed application so we can deploy it from the command line. Select Overview page for the application then the gear wheel in the application then select 'Delete App'



You want to delete the Service and the Route with the application, so select the checkbox in the services tab and the Routes tab





Select OK to delete the application.

Open up a command or Terminal window and change directory to the location you unzipped the downloaded sample application.

We need to log in to Bluemix so issue one of the following commands, choose region you have been using in Bluemix UI:

```
cf l -a https://api.ng.bluemix.net (Region: US South)

cf l -a https://api.eu-gb.bluemix.net (Region: United Kingdom)
```

enter your email and password that you use to sign in to the Bluemix Web UI. Select the organization and space you want to work in if prompted.

Before we deploy the application we need to deploy a Cloudant database, so we can look at the available services using

```
cf marketplace
```

Your will get a list of all the services, the one we are interested in is the cloudantNoSQLDB

```
WorkloadScheduler          free                           Use the Workload Scheduler service to create and schedule repeata
ble business processes to make applications production ready. Trigger your processes to run based on an event or according to a
 schedule
blazemeter                 free-tier                      The JMeter Load Testing Cloud
cleardb                    spark                          Highly available MySQL for your Apps.
cloudamqp                  lemur                          Managed HA RabbitMQ servers in the cloud
cloudantNoSQLDB            Shared                         Cloudant NoSQL DB provides access to a fully managed NoSQL JSON d
ata layer that's always on. This service is compatible with CouchDB, and accessible through a simple to use HTTP interface for
mobile and web application models
elephantsql                turtle                         PostgreSQL as a Service
erservice-beta1            free                           IBM Embeddable Reporting for Bluemix provides a mechanism to conn
ect to relational data sources, create reports/dashboard, and embed this service within your application.
loadimpact                 lifree                         Automated and on-demand performance testing
memcachedcloud             25mb                           Enterprise-Class Memcached for Developers
mongodb                    100                            MongoDB NoSQL database
mongolab                   sandbox                        Fully-managed cloud MongoDB
mqlight                    Default                        Develop responsive, scalable applications with a fully-managed me
ssaging provider in the cloud. Quickly integrate with application frameworks through easy-to-use APIs.
mysql                      100                            MySQL database
newrelic                   standard                       Manage and monitor your apps
```

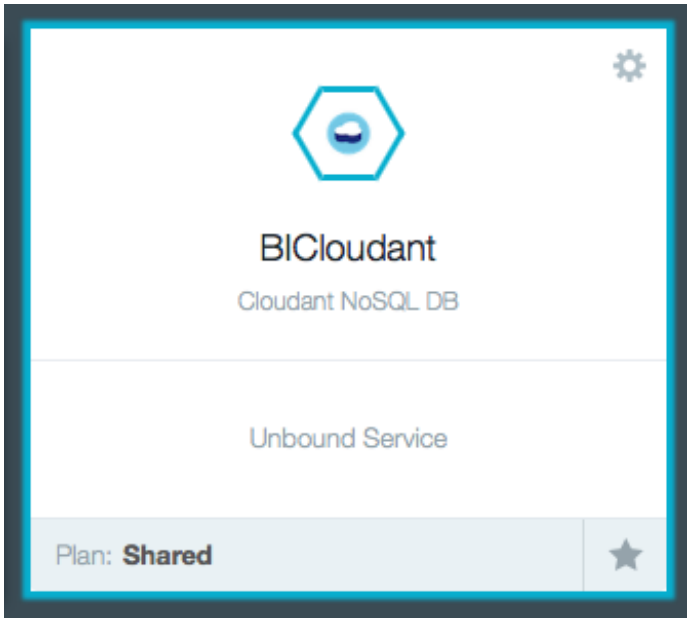
to create the service use command

```
cf cs cloudantNoSQLDB Shared BICloudant
```

where:

- CloudantNoSQLDB is the name of the service from the cf marketplace command
- Shared is the name of the service plan we want to use from the cf marketplace command
- BICloudant is the name of the service instance we want to use – please choose your own name rather than BICloudant – you will need to use this name when connecting (binding) the service to the application.

If you refresh your Web UI you will now see the deployed service:



We can now deploy the application.

Ensure you are in the directory for your application you should have the following files and directories:

JavaCloudantDB.war WebContent build.xml instructions.md src README.txt

bin dep-jar manifest.yml

Enter the following command – changing the application name to a unique name

```
cf push BI-MyFirstDeploy -p
JavaCloudantDB.war -m 512M --no-manifest
--no-start
```

where:

- BImyFirstDeploy will be the application name and hostname
- -p specifies the path or file (war file) containing the application
- -m specifies the amount of memory to allocate each application instance (1GB is default)
- --no-manifest instructs to CLI tool not to use the supplied manifest (will be explained later)
- --no-start instructs to CLI tool not to automatically start the application

We don't want the application to automatically start because it needs a database to run – we need to link the Cloudant database instance to the application before we want the application to start.

To link the database and application we use the following command – substitute the application name and service instance names you used:

```
cf bs BI-MyFirstDeploy BICloudant
```

where:

- BImyFirstDeploy is the application name used when deloying the application
- BICloudant is the service instance name used when deploying the service

If you refresh the Web UI you should see the application and service now linked, but the application is still stopped.

To start an application use the following command – substitute the name of your application:

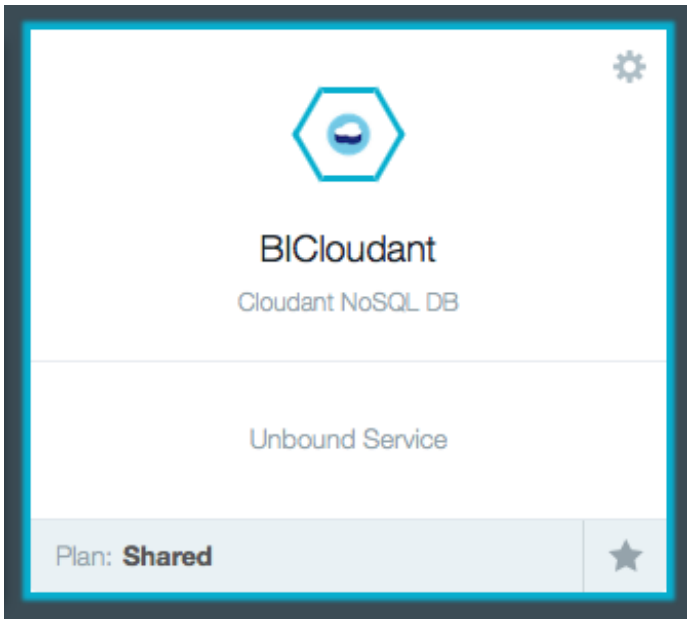
```
cf start BI-MyFirstDeploy
```

where:

- BImyFirstDeploy is the application name you want to start

If you refresh the Web UI you should see the application running. You can launch the application from the Dashboard view

In a text editor open up file src/example/nosql/ResourceServlet.java and modify the name of the file and file content (at the time of this writing, lines 346 and 349)




```
//attachment#1
File file = new File("MySample.txt");
file.createNewFile();
PrintWriter writer = new PrintWriter(file);
writer.write("This is my sample file...");
writer.flush();
writer.close();
```

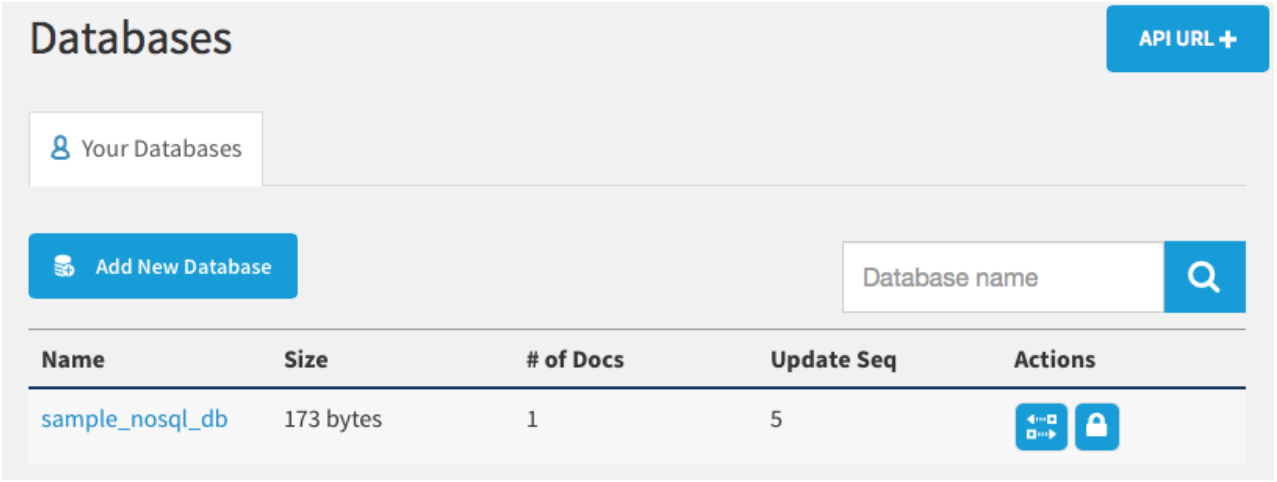
rebuild the WAR file by issuing the ant command in the root directory of the project (contains build.xml)

```
ant  buildfile: /Users/binnes/work/Cloud/WW_IICs/BlueMix/Workshop V2/Lab 3/BImyFirstDeploy/build.xml  clean:       [delete] Deleting directory /Users/binnes/work/Cloud/WW_IICs/BlueMix/Workshop V2/Lab 3/BImyFirstDeploy/bin
```

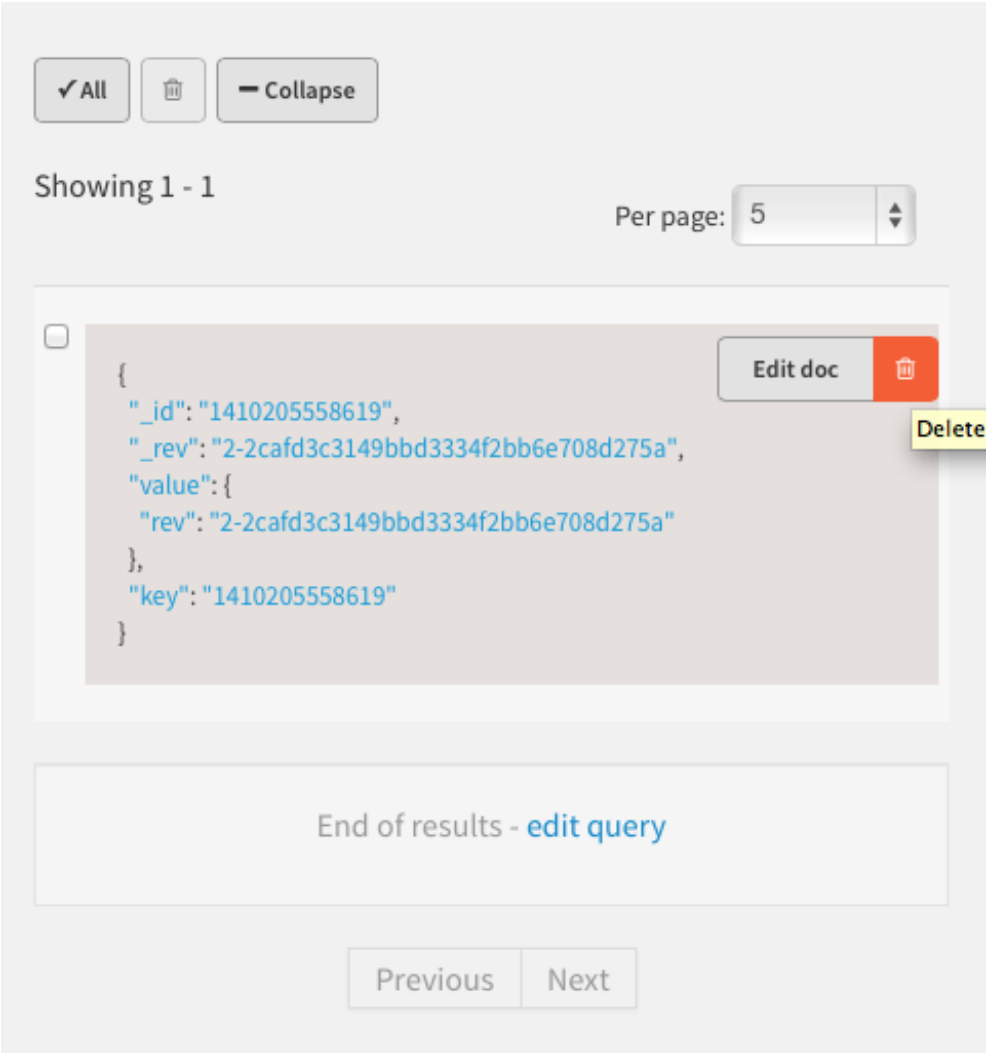
```
[delete] Deleting: /Users/binnes/work/Cloud/WW_IICs/BlueMix/Workshop V2/Lab
3/BImyFirstDeploy/JavaCloudantDB.war  init:       [mkdir] Created dir:
/Users/binnes/work/Cloud/WW_IICs/BlueMix/Workshop V2/Lab 3/BImyFirstDeploy/bin
build-project:       [echo] Java Cloudant DB Web Starter:
/Users/binnes/work/Cloud/WW_IICs/BlueMix/Workshop V2/Lab 3/BImyFirstDeploy/build.xml
[javac] Compiling 1 source file to /Users/binnes/work/Cloud/WW_IICs/BlueMix/Workshop
V2/Lab 3/BImyFirstDeploy/bin       [javac] warning: [options] bootstrap class path
not set in conjunction with -source 1.5       [javac] Note:
/Users/binnes/work/Cloud/WW_IICs/BlueMix/Workshop V2/Lab
3/BImyFirstDeploy/src/example/nosql/ResourceServlet.java uses unchecked or unsafe
operations.       [javac] Note: Recompile with -Xlint:unchecked for details.
[javac] 1 warning      build-war:       [war] Building war:
/Users/binnes/work/Cloud/WW_IICs/BlueMix/Workshop V2/Lab
3/BImyFirstDeploy/JavaCloudantDB.war      build:      BUILD SUCCESSFUL
```

We need to remove the sample file from the database to allow it to be populated again, so in the Bluemix Web UI select the Cloudant Service instance then launch the Cloudant Dashboard.

You should see a single DB



select it, then select to delete the document

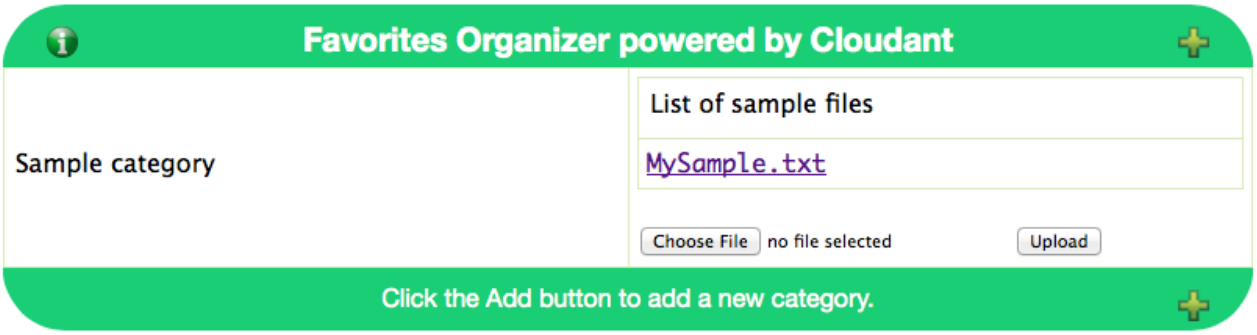


confirm the delete when prompted.

Redeploy the updated WAR file with the push command – this time no need to include the --no-start or memory parameter

```
cf push BI-MyFirstDeploy -p JavaCloudantDB.war --no-manifest
```

Once the application has restarted test to ensure your changes are now running.



We will finish this exercise by deleting the application and service.

```
cf d BI-MyFirstDeploy -r
```

where:

- BImyFirstDeploy is the application name to be deleted
- -r instructs Bluemix to also delete the routes attached to the application

```
cf ds BICloudant
```

where

- BICloudant is the name of the service instance to be deleted

Note : you will be asked to confirm the delete of the application and service answer y to confirm you want to delete