Lab: Deploy and update the application by using the cf CLI

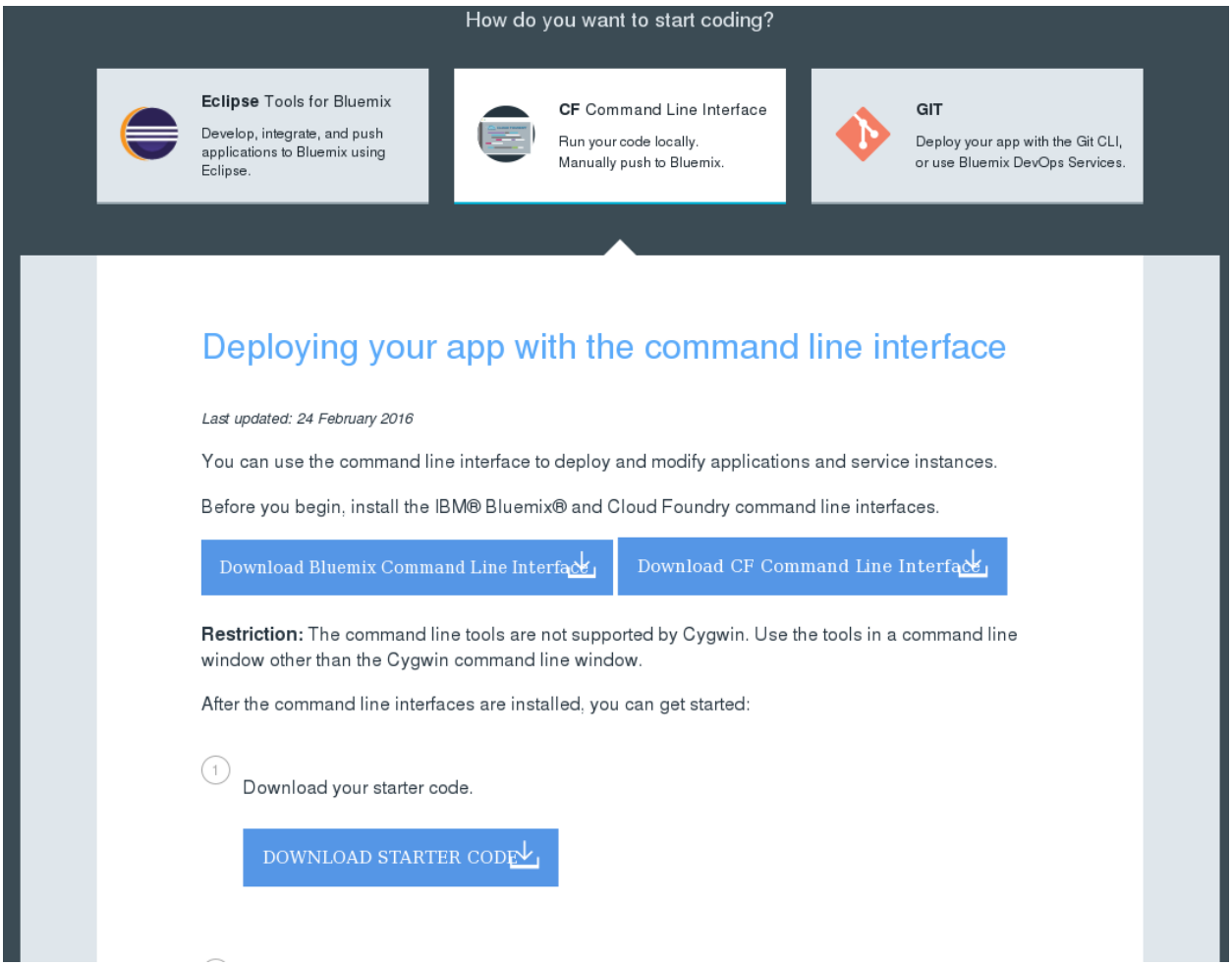
Deploy and then update the application by using the CLI

In this lab, you use the cf command-line interface (CLI) to work with Bluemix. The screenshots are from the Bluemix classic interface and pick up from the previous lab.

The cf CLI is a tool you will use in a terminal or command window on your workstation.

Use the same sample application that was used in the previous lab “Deploy your first application.”

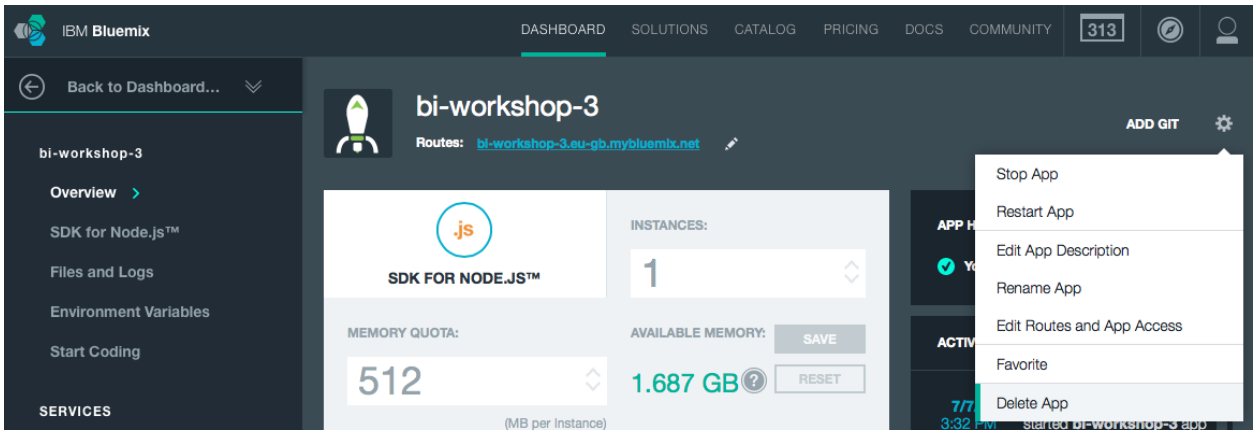
1. Click **Start Coding** and then click **Download Starter Code**.



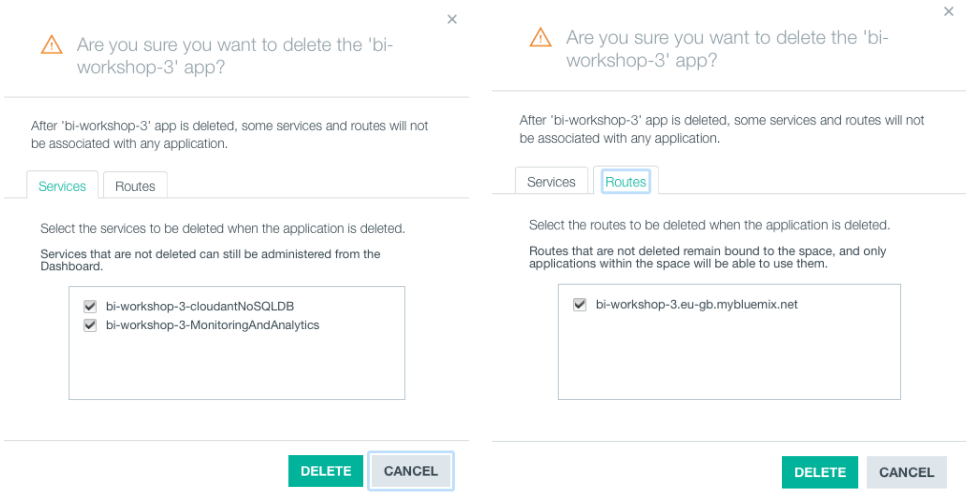
2. After the starter package is downloaded, move it to a directory on your workstation where you want to work, such as the `Bluemix` directory in your `Documents` folder.

3. Extract the package by double-clicking or right-clicking and click **Extract** or **Unarchive** or use a command line tool. Do *not* delete the ZIP file: you will need it in the next lab “Working with Eclipse.”

4. Delete the deployed application so that you can deploy it from the command line. Click the **Overview** page for the application, click the gear wheel in the application, and then click **Delete App**.



5. You may confirm that the service(s) and the route for the application will be deleted in the **Services** tab and the **Routes** tab. By default, they will be checked:



6. Click **DELETE** to delete the application.

7. Open a command or terminal window and change the directory to the location where you extracted the downloaded sample application. (The file `package.json` should be in your current directory.) Note that the cf CLI tool is not supported in a Cygwin bash shell on Windows.

8. Log in to Bluemix by issuing one of the following commands. Use the same region that you used in the Bluemix web UI:

   ```
   cf l -a https://api.ng.bluemix.net          (Region: US South)
   cf l -a https://api.eu-gb.bluemix.net       (Region: United Kingdom)
   cf l -a https://api.au-syd.bluemix.net      (Region: Sydney)
   ```

9. Enter the email and password that you used to log in to the Bluemix web UI. If prompted, select the organization and space that you want to work in.

10. Before you deploy the application, deploy a Cloudant database. View the available services by running this command (this command will take a little while to run as it collects all catalog entries):

    ```
    cf marketplace
    ```

11. In the list of services, find the cloudantNoSQLDB service.

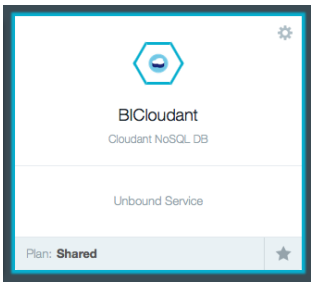
```
WorkloadScheduler          free                           Use the Workload Scheduler service to create and schedule repeata
ble business processes to make applications production ready. Trigger your processes to run based on an event or according to a
 schedule
blazemeter                 free-tier                      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
```

12. Create the service by running this command:

    ```
    cf cs cloudantNoSQLDB Shared BICloudant
    ```

    - `CloudantNoSQLDB` is the name of the service from the cf marketplace command.
    - `Shared` is the name of the service plan that you want to use from the `cf marketplace` command.
    - `BICloudant` is the name of the service instance that you want to use. Enter your own name rather than `BICloudant`. You will use this new name when connecting (binding) the service to the application.

13. Refresh your web UI to you see the deployed service.



14. Deploy the application.

Push the application to Bluemix by entering the following command. Change the application name to your unique name:

```
cf push BI-MyFirstDeploy-3 -c "node app.js" -m 128M --no-manifest --no-
start
```

- `BI-myFirstDeploy-3` is your unique application name and host name.
- `-c` specifies the command to start the application.
- `-m` specifies the amount of memory to allocate to each application instance. The default is 1 GB.
- `--no-manifest` instructs to CLI tool to ignore the supplied manifest file. This will allow the Cloudant database instance that you just created to be linked to the application.
- `--no-start` instructs to CLI tool not to automatically start the application.

The reason not to automatically start is because it needs a database to run. You must link the Cloudant database instance to the application before you start the application. In Cloud Foundry, the action of linking is described as binding the service instance.

15. Link the database and application by using the following command. Substitute the application name and service instance names that you used previously:

```
cf bs BI-MyFirstDeploy-3 BICloudant
```

- `BI-myFirstDeploy-3` is the unique application name used to deploy.
- `BICloudant` is the service instance name used when the service is deployed.

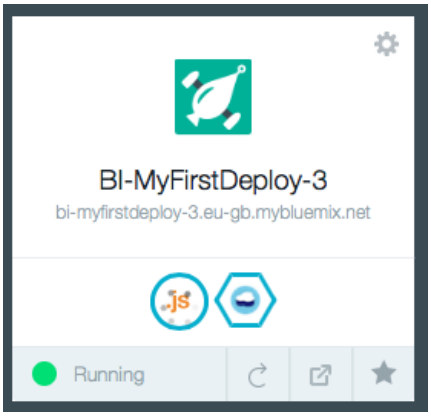
If you refresh the web UI, you see that the application and service are linked, but the application is still stopped.

16. Start an application by running the following command. Substitute the name of your application:

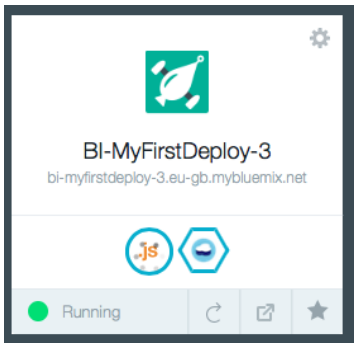
```
cf start BI-MyFirstDeploy-3
```

- `BI-myFirstDeploy-3` is the application that you want to start.

If you refresh the web UI, you should see the application running. If not, you can start the application from the Dashboard.



17. Launch the application by clicking the route in the web UI.



18. In a text editor, open the file `app.js` and modify the name of the file, the file description, and the value (lines 335, 336 and 310):
    - Line 335: Change the docName from `'sample_doc'` to `'test_doc'`
    - Line 336: Change the docDesc from `'A sample Document'` to `'A test Document'`
    - Line 339: Change the value from `'A sample Document'` to `'A test Document'`

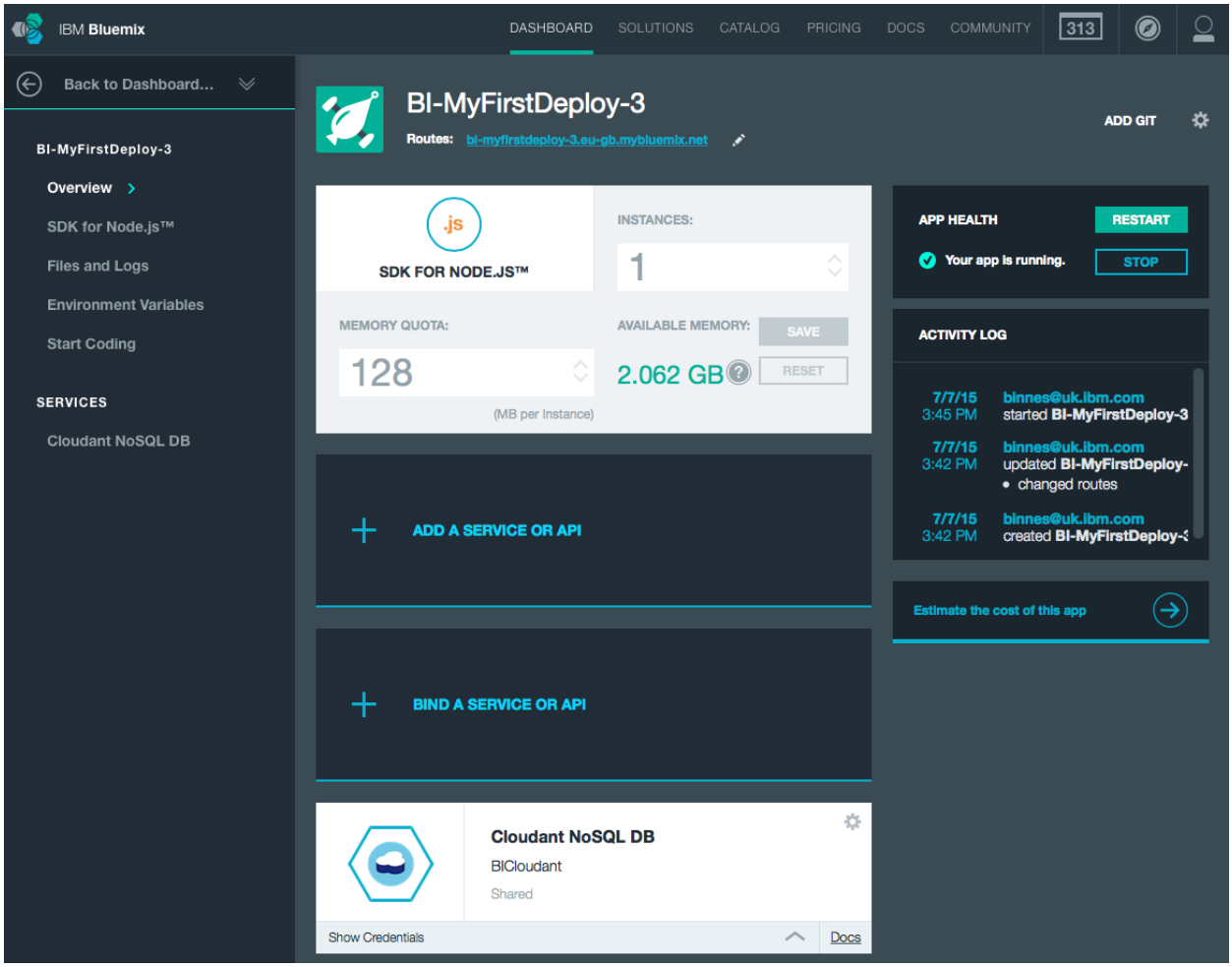
    Save the file when you're finished editing.

```
320
321 app.get('/api/favorites', function(request, response) {
322
323         console.log("Get method invoked.. ")
324
325         db = cloudant.use(dbCredentials.dbName);
326         var docList = [];
327         var i = 0;
328         db.list(function(err, body) {
329                 if (!err) {
330                         var len = body.rows.length;
331                         console.log('total # of docs -> '+len);
332                         if(len == 0) {
333                                 // push sample data
334                                 // save doc
335                                 var docName = 'sample_doc';
336                                 var docDesc = 'A sample Document';
337                                 db.insert({
338                                         name : docName,
339                                         value : 'A sample Document'
340                                 }, '', function(err, doc) {
```

When the application starts for the first time, it creates a sample document in the database.

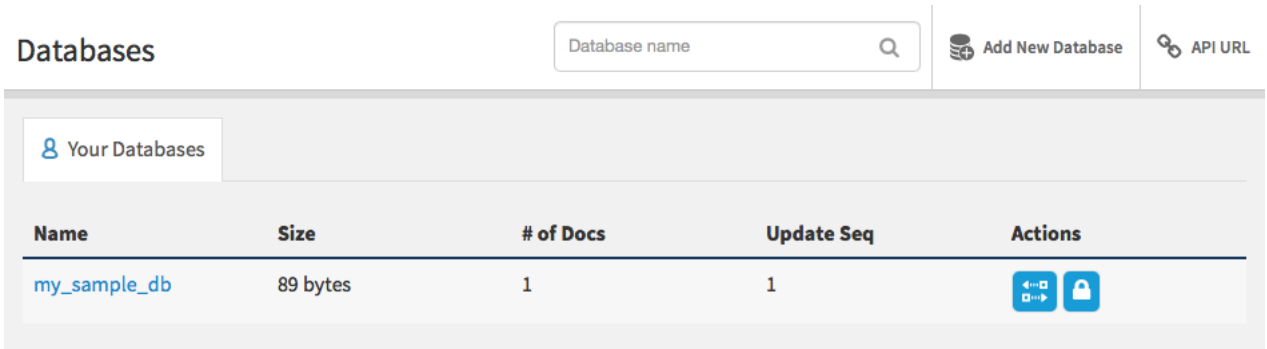
We have just modified the code that creates the sample document in the database. Now you will delete the document from the database and then restart the application to allow the database to be populated with the modified document.

19. In the Bluemix web UI, select the Cloudant Service instance and then start the Cloudant Dashboard.

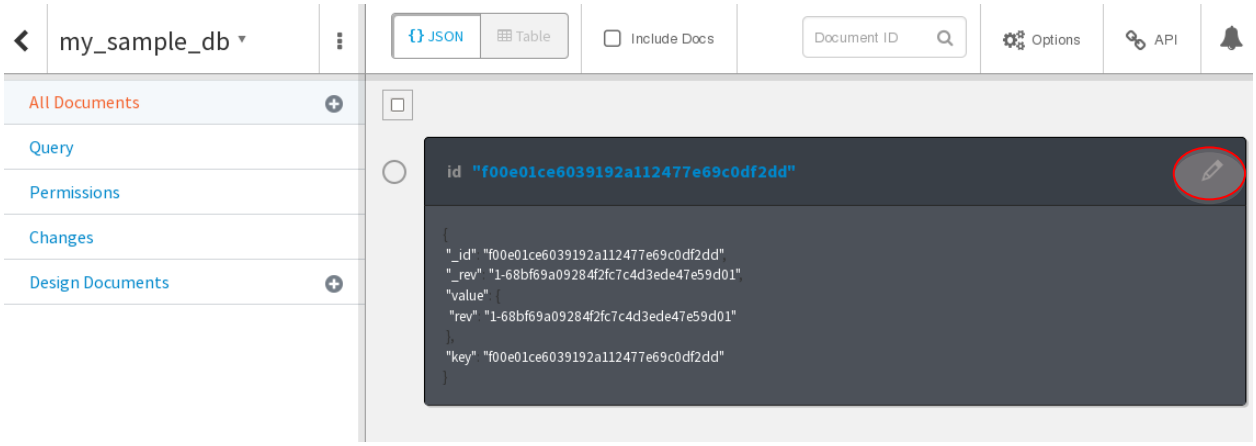


20. Launch the Cloudant console.

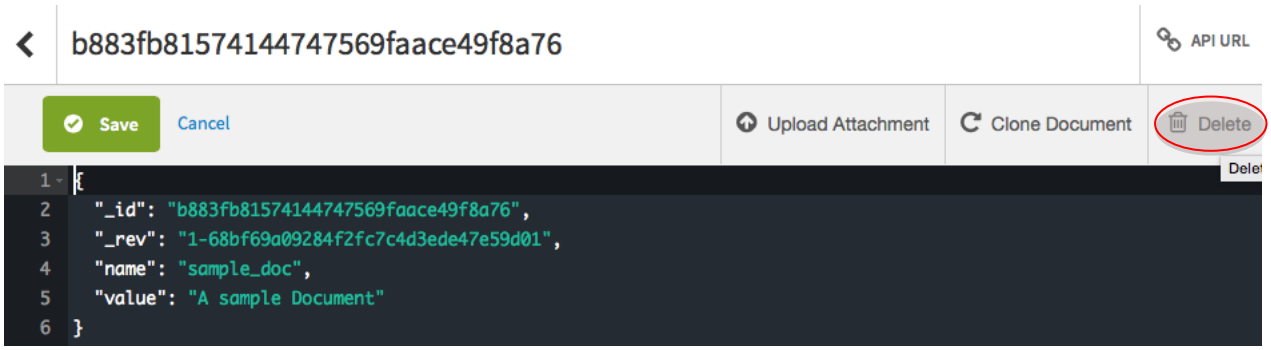
You now see a single database. Select the database by clicking on the name:



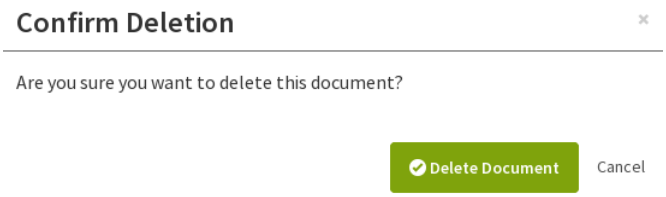
21. Edit the database document by clicking on the pencil icon:



22. Click on the **Delete** button:



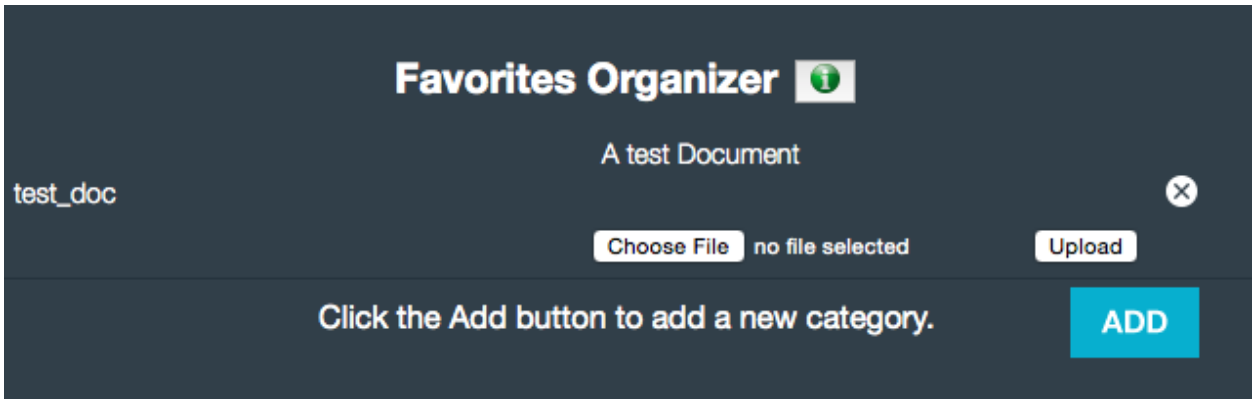
23. Confirm the deletion when prompted.



24. Redeploy the updated application with the `push` command. This time, you don't need to include the `--no-start` or `-m` parameters.

```
cf push BI-MyFirstDeploy-3 -c "node app.js" --no-manifest
```

25. After the application has restarted, test it to ensure that your changes are now running.



After the application is tested to confirm that the modified code is running, the application can be deleted to release resources for the next lab.

26. Delete the application and service and confirm the deletion when prompted by running the following two commands:

    Delete the application: `cf d BI-MyFirstDeploy-3 -r`

    - `BI-myFirstDeploy-3` is the application name to be deleted.
    - `-r` instructs Bluemix to also delete the routes attached to the application.

    Delete the service: `cf ds BICloudant`

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

    Confirm the deletion of the application and service by checking the dashboard in the Bluemix web UI.